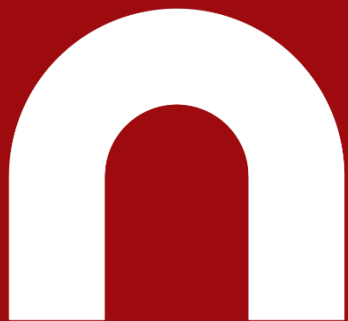




isec
Engenharia

MESTRADO EM ENGENHARIA
ELETROTÉCNICA

**Aplicações de automação na indústria de
cerâmica**

João Pedro Godinho Mendes

Orientadores

Professor Doutor Nuno Miguel Fonseca Ferreira

Professor Coordenador com Agregação

Doutor Victor Daniel Neto dos Santos

Professor Coordenador

INSTITUTO POLITÉCNICO DE
COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Coimbra, março de 2023



isec **Engenharia**

DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA

Aplicações de automação na indústria de cerâmica

**Relatório de Estágio de Natureza Profissional para a obtenção do
grau de Mestre em Engenharia Eletrotécnica**

**Especialização em Automação e Comunicações em Sistemas
Industriais**

Autor

João Pedro Godinho Mendes

Orientadores

Professor Doutor Nuno Miguel Fonseca Ferreira

Professor Coordenador com Agregação

Doutor Victor Daniel Neto dos Santos

Professor Coordenador

Supervisor na empresa Centro Tecnológico de Cerâmica e Vidro

Mestre Victor Francisco

INSTITUTO POLITÉCNICO DE
COIMBRA

Coimbra, março de 2023

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

AGRADECIMENTOS

Gostaria de agradecer aos meus familiares, pelo apoio recebido em todas as fases da minha vida, que se revelou crucial para o meu desenvolvimento pessoal e profissional. Agradeço-lhes pela compreensão e motivação constante ao longo de todo o percurso académico.

Ao meu orientador, Professor Doutor Nuno Ferreira, por toda a disponibilidade, preocupação, motivação, incentivo e ajuda no decorrer de todo o estágio curricular e durante o processo de escrita de este relatório de estágio.

Ao coorientador Doutor Victor Daniel Neto dos Santos, por toda a disponibilidade, preocupação, motivação, incentivo e ajuda durante o processo de escrita do presente relatório de estágio.

Ao supervisor da empresa responsável I&D CTCV, Mestre Victor Francisco, pela motivação, confiança e oportunidade para a realização do estágio curricular, agradeço-lhe a boa disposição, conselhos e a total disponibilidade durante a realização deste trabalho.

Ao André Santos e ao Rogério Torres, que me acompanharam no decorrer do estágio curricular, por toda a presença, motivação, disponibilidade e companheirismo durante esta etapa.

À Filipa Nunes, pela constante motivação, compreensão, ajuda, presença e carinho durante todo o período do estágio curricular e o período da elaboração do presente relatório de estágio.

Ao Rui Gouveia, por toda a dedicação, ensinamento, apoio, motivação, confiança e companheirismo durante esta etapa.

RESUMO

O presente relatório surge com o objetivo de apresentar o Estágio Curricular desenvolvido no Centro Tecnológico de Cerâmica e Vidro (CTCV), a fim de obtenção do grau de Mestre em Engenharia Eletrotécnica com especialização em Automação e Comunicações em Sistemas Industriais.

Apresenta vários sistemas, criados para a inserção da automação na indústria de cerâmica, desde processos de vidragem por mergulho, a inspeção autónoma na tarefa de localização automática e a avaliação de defeitos na superfície de peças cerâmicas.

Consta de uma explicação/apresentação do processo de automação, que recorre a sistemas robóticos, inteligência artificial, redes de comunicação, entre outros, criando processos mais autónomos e menos dependentes de mão de obra.

A inclusão da automação tem um papel fundamental, uma vez que utiliza um conjunto de técnicas destinadas a tornar automática a realização de tarefas.

Palavras chaves: Automação, Robótica, Visão Artificial, Simulação, Indústria de Cerâmica

ABSTRACT

The main purpose of this report is to present the Curricular Internship developed at the Technological Center of Ceramics and Glass, in order to obtain a master's degree in Electrical Engineering with a specialization in Automation and Communications in Industrial Systems.

It presents several systems, created for the insertion of automation in the ceramic industry, from dip glazing processes, autonomous inspection in the automatic location task, and the evaluation of defects on the surface of ceramic pieces.

It consists of an explanation/presentation of the automation process, which requires robotic systems, artificial intelligence, and communication networks, among others, creating more autonomous and less labor-dependent processes.

The inclusion of automation has a fundamental role since it uses a set of techniques aimed at making the execution of tasks automatic.

Keyword: Automation, Robotics, Artificial Vision, Simulation, Ceramic Industry

ÍNDICE

RESUMO.....	iii
ABSTRACT	iv
ÍNDICE DE FIGURAS	viii
ÍNDICE DE TABELAS	xii
ABREVIATURAS E SIGLAS.....	xiii
1. INTRODUÇÃO	1
1.1. Estrutura do documento	2
2. ESTADO DA ARTE.....	3
2.1. Automação Industrial.....	3
2.1.1. Evolução da automação industrial.....	6
2.1.2. Automação fixa.....	13
2.1.3. Automação programável.....	13
2.1.4. Automação flexível.....	14
2.1.5. Automação integrada.....	15
2.2. Robótica Industrial.....	15
2.2.1. Elementos de um robô industrial	17
2.2.2. Tipos de robôs industriais.....	19
2.2.3. Robótica colaborativa	24
2.3. Sistemas de visão presentes na automação	24
2.4. PLC - Programmable Logic Controller	26
2.4.1. Classificação dos PLC	28
2.4.2. Controlo de entradas e saídas	29
2.4.3. Arquitetura do autómato	29
2.4.4. Programação do autómato	31
2.5. Human Machine Interface (HMI)	35
3. DESENVOLVIMENTO DE PROJETOS NA INDÚSTRIA DE CERÂMICA.....	39
3.1. Guia Simulador MotoSimEG-VRC	39
3.1.1. Criar projeto e ambiente envolvente.....	40
3.1.2. Criar Ferramenta.....	48

3.1.3.	Editar/Definir dados da ferramenta	50
3.1.4.	Criar um <i>JOB</i>	52
3.1.5.	Programar o robô	53
3.1.6.	Entradas e saídas.....	55
3.1.7.	Alterar a Programação do Robô	56
3.1.8.	Adicionar uma passadeira transportadora.....	57
3.1.9.	Adicionar uma prensa	58
3.1.10.	Movimentar dispositivos externos (Passadeira/ Prensa).....	60
3.1.11.	Programar dispositivos externos	61
3.1.12.	Scripts	62
3.1.13.	Simulação.....	64
3.1.14.	Exemplo criado para o CTCV	64
3.2.	Automação nos processos de vidragem	75
3.2.1.	Vidragem manual por mergulho.....	75
3.2.2.	Descrição do processo desenvolvido	76
3.2.3.	Sistema de visão	80
3.2.4.	Sistema de manipulação	82
3.2.5.	Sistema de repouso	84
3.2.6.	Interligação e controlo dos diversos sistemas.....	85
3.2.7.	Programação	89
3.3.	Automação nos processos de inspeção	90
3.3.1.	Estudo do protótipo	91
3.3.2.	Programação PLC.....	95
3.3.3.	Interface homem-máquina.....	96
3.4.	Feiras e Artigos	98
4.	CONCLUSÕES E TRABALHO FUTURO	99
	REFERÊNCIAS	101
	ANEXOS	107

ÍNDICE DE FIGURAS

Figura 1 – Linha automatizada na indústria de cerâmica. Fonte: Própria	3
Figura 2 – Paletização de pratos de cerâmica. Fonte: Própria	5
Figura 3 – Evolução da indústria. Fonte: (Disner & Cipullo, 2021)	6
Figura 4 – Tear mecânico Fonte: (Nunes, 2015)	7
Figura 5 – Circuito sequencial com recurso a relés Fonte: (Souza, 2021)	8
Figura 6 – Pirâmide níveis de automação Fonte: (Hammari, 2021)	9
Figura 7 – Principais tecnologias na indústria 4.0 Fonte:(Tecnicon, 2020)	12
Figura 8 – Principais objetivos da indústria 5.0 (Xu, 2021, p. 0).....	13
Figura 9 – Robô <i>Unimate</i> Fonte: (Engelberger, 1979).....	16
Figura 10 – Robôs indústrias <i>KUKA</i> Fonte: (<i>Robôs industriais</i> , n.d.)	17
Figura 11 – Diversos controladores <i>KUKA</i> Fonte: (<i>KUKA KR C4</i> , n.d.)	18
Figura 12 – Robôs articulados Fonte: (Silveira, 2019)	20
Figura 13 – Robôs cartesianos Fonte: (Silveira, 2019)	20
Figura 14 – Configuração robô cilíndrico Fonte: (Silveira, 2019).....	21
Figura 15 – Robô esférico Fonte: (Silveira, 2019).....	22
Figura 16 – Robô <i>SCARA</i> Fonte: (<i>Epson</i> , n.d.)	22
Figura 17 – Robô Delta. Fonte: Silveira, 2019	23
Figura 18 – Robô colaborativo Fonte: (mercado, 2019)	24
Figura 19 – Ciclo das etapas de um sistema de visão. Fonte: Própria.	25
Figura 20 – Sistema de inspeção <i>INSPECTWARE</i> Fonte: Própria	26
Figura 21 – Aplicação do Sistema de visão Fonte: (Riku, 2006).....	26
Figura 22 – PLC <i>NJ301</i> Fonte: Própria	27
Figura 23 – Arquitetura de um PLC Fonte: Própria	30
Figura 24 – Exemplo de programação em <i>ladder</i> Fonte: Própria	32
Figura 25 – Exemplo de programação em <i>Structured Text</i> Fonte: (Duranso, 2021)	33
Figura 26 – Exemplo de programação em <i>Instruction List</i> Fonte: (Budimir, 2019)	34
Figura 27 – Exemplo de programação em <i>Sequential Function Chart</i> Fonte: (Antonsen, 2022)	34
Figura 28 – Exemplo de programação em <i>Function Block Diagram</i> Fonte: (Romanov, 2019).....	35
Figura 29 – HMI Centro de Inspeção Fonte: Própria.....	36
Figura 30 – HMI segurança Fonte: Própria.....	37
Figura 31 – Janela principal do MotoSim EG-VRC Fonte: Simulador MotoSimEG-VRC	41
Figura 32 – Menu <i>Controller</i> do MotoSim EG-VRC Fonte: Simulador MotoSimEG-VRC	41
Figura 33 – Janela “ <i>New Controller</i> ” Fonte: Simulador MotoSimEG-VRC	42
Figura 34 – Janela “ <i>New Controller</i> ” Fonte: Simulador MotoSimEG-VRC	42

Figura 35 – Janela de modo de manutenção de controlador Fonte: Simulador MotoSimEG-VRC	43
Figura 36 – Controlador Virtual do Robô Fonte: Simulador MotoSimEG-VRC	43
Figura 37 – Robô <i>Gp4</i> e Controlador Virtual Fonte: Simulador MotoSimEG-VRC	44
Figura 38 – <i>Cad Tree</i> Robô GP4 Fonte: Simulador MotoSimEG-VRC.....	44
Figura 39 – Janela para adicionar novos modelos Fonte: Simulador MotoSimEG-VRC.....	45
Figura 40 – Exemplo da criação de um suporte para o robô com <i>BOX2</i> Fonte: Simulador MotoSimEG-VRC	46
Figura 41 – Alteração de coordenadas de origem Fonte: Simulador MotoSimEG-VRC	47
Figura 42 – Exemplo da criação da célula presente no CTCV Fonte: Simulador MotoSimEG-VRC	47
Figura 43 – Criação de um modelo integrado numa hierarquia Fonte: Simulador MotoSimEG-VRC	48
Figura 44 – Definir atributos do modelo Fonte: Simulador MotoSimEG-VRC	49
Figura 45 – Parâmetros da ferramenta criada Fonte: Simulador MotoSimEG-VRC	49
Figura 46 – Ferramenta criada através dos parâmetros inseridos Fonte: Simulador MotoSimEG-VRC ...	50
Figura 47 – Menu [<i>Controller</i>] onde se situa a ferramenta [<i>Tool Data</i>] Fonte: Simulador MotoSimEG-VRC	50
Figura 48 – Parâmetros inseridos para o <i>gripper</i> gerado Fonte: Simulador MotoSimEG-VRC.....	51
Figura 49 – Menu [<i>Home</i>] onde se localiza a ferramenta [<i>Frame Display</i>].....	51
Figura 50 – Controlador virtual Fonte: Simulador MotoSimEG-VRC.....	52
Figura 51 – Janela no controlador para um novo programa Fonte: Simulador MotoSimEG-VRC	52
Figura 52 – Localização do ícone do teclado do controlador virtual Fonte: Simulador MotoSimEG-VRC	53
Figura 53 – Exemplo de movimento de juntas com velocidade 100% Fonte: Simulador MotoSimEG-VRC	54
Figura 54 – Localização da ferramenta “ <i>OLP</i> ” Fonte: Simulador MotoSimEG-VRC	54
Figura 55 – Localização da ferramenta “ <i>I/O Monitor</i> ” Fonte: Simulador MotoSimEG-VRC	55
Figura 56 – Localização da ferramenta “ <i>I/O Connection Manager</i> ” Fonte: Simulador MotoSimEG-VRC	56
Figura 57 – Inserir passadeira transportadora Fonte: Simulador MotoSimEG-VRC	57
Figura 58 – Janela “ <i>Conveyor Installation</i> ” Fonte: Simulador MotoSimEG-VRC.....	57
Figura 59 – Ilustração de ambiente com passadeira transportadora Fonte: Simulador MotoSimEG-VRC	58
Figura 60 – Adicionar uma prensa Fonte: Simulador MotoSimEG-VRC	59
Figura 61 – Janela [<i>Press</i>] Fonte: Simulador MotoSimEG-VRC.....	59
Figura 62 – Configuração da prensa Fonte: Simulador MotoSimEG-VRC	59
Figura 63 – Representação da prensa no simulador Fonte: Simulador MotoSimEG-VRC	60
Figura 64 – Localização da ferramenta [<i>Position Panel</i>] Fonte: Simulador MotoSimEG-VRC	60
Figura 65 – Ferramenta de movimentação de dispositivos externos Fonte: Simulador MotoSimEG-VRC	61
Figura 66 – Painel de programação de dispositivos externos Fonte: Simulador MotoSimEG-VRC	61

Figura 67 – Janela “ <i>Model Script Editor</i> ” Fonte: Simulador MotoSimEG-VRC	62
Figura 68 – Exemplo de comando “ACT” Fonte: Simulador MotoSimEG-VRC.....	63
Figura 69 – Menu de simulação Fonte: Simulador MotoSimEG-VRC	64
Figura 70 – Imagens reais da despaletização efetuado por recuso humano Fonte: Própria.....	65
Figura 71 – Vista global da linha de montagem criada Fonte: Própria.....	66
Figura 72 – Disposição criada no simulador dos azulejos Fonte: própria.....	67
Figura 73 – Especificações do robô GP-180-120 Fonte: (Yaskawa, n.d.)	68
Figura 74 – Garra do robô 1 Fonte: própria.....	69
Figura 75 – Espaço de trabalho do Robô 1 Fonte: própria.....	69
Figura 76 – Garra robô 2 mecanismo secundário A) aberto B) fechado Fonte: própria	70
Figura 77 – Robô 2 a transportar azulejos com o uso do sistema de ventosas Fonte: própria	71
Figura 78 – Robô 2 a transporta as caixas com o uso das duas ferramentas Fonte: própria	71
Figura 79 – Caixa com os azulejos antes da celagem Fonte: própria.....	72
Figura 80 – Caixa com os azulejos apos da celagem.....	72
Figura 81 – Mecanismo de paletização A) Aberto B) Fechado Fonte: própria.....	73
Figura 82 – Robô 3 a executar a paletização Fonte: própria.....	74
Figura 83 – Paletização final A) real B) virtual Fonte: própria	74
Figura 84 – Funcionário a vidrar um prato por mergulho Fonte: Própria.....	76
Figura 85 – Robô na posição de repouso Fonte: Própria	77
Figura 86 – Imagem captada e processamento da mesma Fonte: Própria	77
Figura 87 – Robô a recolher o prato	78
Figura 88 – Ilustração da vidragem do prato	78
Figura 89 – Movimento de uniformização do vidrado	79
Figura 90 – Robô a posicionar os pratos Fonte: Própria	80
Figura 91 – Câmara <i>Mako</i> e luminária Fonte: Própria	81
Figura 92 – Ligação do sistema de visão.....	82
Figura 93 – Características de alcance do robô GP4 Fonte: Própria	83
Figura 94 – Garra projetada anexada ao robô Fonte: Própria	83
Figura 95 – Regulador de pressão Fonte Própria.....	84
Figura 96 – Sistema de repouso Fonte: Própria.....	85
Figura 97 – Esquema de interligação dos diversos componentes Fonte: Própria.....	86
Figura 98 – Reles de proteção Fonte: Própria	86
Figura 99 – <i>Arduino</i> , acoplado a <i>Stepper Motor Shield</i> e dois <i>Chips A4988</i> Fonte: Própria.....	87
Figura 100 – Placa com função de informar o funcionamento do sistema Fonte: Própria	87
Figura 101 – Placa GPIO Fonte: Própria.....	88

Figura 102 – Diagrama da placa GPIO Fonte: “<i>Install and Wiring</i>”	88
Figura 103 – Fluxograma da descrição do processo de vidragem Fonte: Própria.....	90
Figura 104 – Exemplos típicos de defeitos encontrados nas peças cerâmicas Fonte: Própria	91
Figura 105 – Esquema da rede de comunicação Fonte: Própria	92
Figura 106 – a) Estrutura da célula b) Sistema de manipulação Fonte: Própria	93
Figura 107 – Sistema de aquisição: a) projetado, b) realizado Fonte: Própria.....	93
Figura 108 – a) Fluxograma de instruções b) Grafcet de instruções Fonte: Própria	94
Figura 109 – Programação em <i>Ladder</i> do Sistema de Iluminação do Sistema de Inspeção Fonte: Própria.	96
Figura 110 – Botoeira principal Fonte: própria	97
Figura 112 – HMI Sistema de inspeção Fonte: Própria	97
Figura 113 – Logotipo da conferência ICMA²SC’22.....	98
Figura 114 – Presença na feira 360 <i>Tech Industry</i> 2022.	98

ÍNDICE DE TABELAS

Tabela 1 – Funções das ligações e respetivos pins conectores Fonte: própria.....	89
---	-----------

ABREVIATURAS E SIGLAS

3D	Três Dimensões
A	Ampere
AC	<i>Alternating Current</i>
AOUT	<i>Analog Output</i>
AP	Autómato Programável
C#	Linguagem de programação C#
C++	Linguagem de programação C++
CAD	<i>Computer-Aided Design</i>
CNC	<i>Computer Numerical Control</i>
COM	Comum
CPU	<i>Central Process Unit</i>
CTCV	Centro Tecnológico de Cerâmica e Vidro
D	<i>Duty Cycle</i>
DC	<i>Direct Current</i>
DEE	Departamento de Engenharia Eletrotécnica
DIN	<i>Digital Input</i>
DLL	<i>Dynamic Link Library</i>
E/S	Entradas e Saídas
ERP	<i>Enterprise Resource Planning</i>
FBD	<i>Function Block Diagram</i>
GPIO	<i>General Purpose Input/Output</i>

GPS	<i>Global Positioning System</i>
GUI	<i>Graphical User Interface</i>
HMI	<i>Human-Machine Interface</i>
IA	Inteligência Artificial
IDE	<i>Integrated Development Environment</i>
IEC	<i>International Electrotechnical Commission</i>
IIoT	<i>Industrial Internet of Things</i>
IL	<i>Instruction List</i>
IN	<i>Input</i>
IoT	<i>Internet of Things</i>
ISEC	Instituto Superior de Engenharia de Coimbra
LD	<i>Ladder</i>
LED	<i>Light Emitting Diode</i>
MES	<i>Manufacturing Execution System</i>
mm	Milímetros
OUT	<i>Output</i>
PC	<i>Personal Computer</i>
PLC	<i>Programmable Logic Controller</i>
POE	<i>Power over Ethernet</i>
PWM	<i>Pulse Width Modulation</i>
QR	<i>Quick Response</i>
RTU	<i>Remote Terminal Unit</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>

SCARA	<i>Selective Compliance Assembly Robot Arm</i>
SFC	<i>Sequential Flow Chart</i>
SO	Sistema Operativo
SoC	<i>System-on-a-Chip</i>
ST	<i>Structured Text</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
TwinCAT	<i>The Windows Control and Automation Technology</i>
V	<i>Volt</i>
W	<i>Watt</i>
Ω	<i>Ohm</i>

1. INTRODUÇÃO

A evolução nos sistemas de produção deve-se à competição entre as empresas, ao aumento das exigências do consumidor e ao desenvolvimento das tecnologias, que envolvem cada vez mais microprocessadores, sistemas robóticos, inteligência artificial, redes de comunicação, entre outras, criando assim cada vez processos mais autónomos e menos dependentes de mão de obra. A inclusão da automação tem um papel fundamental, recorrendo a um conjunto de técnicas destinadas a tornar automáticas a realização de tarefas. A automação industrial consiste em recriar vários processos, substituindo o trabalho humano por diversos equipamentos, garantindo ganhos de quantidade e qualidade da produção e, simultaneamente, oferecendo preços mais competitivos para os consumidores. Esta evolução está ligada, em grande parte, ao avanço recente da microeletrónica, que invadiu os setores produtivos apesar de oferecer grandes progressos na produção, substituindo os recursos humanos por sistemas robóticos, propiciando enormes ganhos de produtividade ao integrar tarefas distintas, nomeadamente a produção (Sioma, 2020).

A automação e a robótica podem ser empregues em diversos contextos, em que o objetivo principal se foque na otimização de processos industriais. A Indústria da Cerâmica em Portugal representa um número de empresas bastante significativo, sendo o maior exportador europeu de louça cerâmica de uso doméstico. Não obstante, este tipo de indústria emprega um elevado número de pessoas. Tendo em conta esta realidade, torna-se imperativo rentabilizar e melhorar vários processos, de modo a garantir soluções cada vez mais otimizadas e eficientes. Grande parte da investigação e desenvolvimento nesta área de conhecimento tem sido realizada no Centro Tecnológico da Cerâmica e do Vidro (CTCV) (*QUEM SOMOS - CTCV*, n.d.).

O CTCV localizado no iParque em Antanhol (Coimbra), foi fundado em 1987, é composto por uma equipa pluridisciplinar, com grande experiência acumulada ao longo dos anos, e com excelente formação especializada. Oferece um conjunto de soluções integradas que passam pela consultoria especializada, auditoria, formação profissional, medição, ensaio e soluções na área da Cerâmica. Nos últimos 10 anos o CTCV participou em mais de 30 projetos de I&D, envolvendo cerca de 100 parceiros nacionais e internacionais. (*QUEM SOMOS - CTCV*, n.d.)

Recentemente, nos novos edifícios situados no *iParque*, originaram-se novos projetos de investigação, sendo um deles na área da Robótica e Automação. Através do desenvolvimento nesta área no CTCV, surgiu a oportunidade de criar uma parceria com o Instituto Superior de Engenharia de Coimbra (ISEC), permitindo a aquisição de conhecimentos essenciais nesta área tecnológica, e na promoção futura de projetos de I&D (Miranda, 2020).

1.1. Estrutura do documento

Este relatório está estruturado de forma a evidenciar as competências adquiridas, bem como o trabalho desenvolvido ao longo do período de projeto/estágio. O documento está dividido em quatro capítulos, incluindo o atual procede à introdução do tema e da instituição de acolhimento (Capítulo 1).

No Capítulo 2 encontra-se explanado o Estado da Arte, onde se abordará a automação industrial, nomeadamente a sua evolução, desde a indústria 1.0 até à indústria 4.0. Concomitantemente, estará descrita a Robótica Industrial, com vários elementos e tipos detalhadamente mencionados. De seguida, e ainda no mesmo capítulo, revelar-se-ão os Sistemas de Visão presentes na automação, explorados minuciosamente, sendo elucidados diversos componentes. Posteriormente, eleva-se um outro subcapítulo inerente ao segundo capítulo, onde se aborda o *Programmable Logic Controller* (PLC) e, posteriormente, o *Human-Machine Interface* (HMI).

O Capítulo 3 consiste no Desenvolvimento do Projeto na Indústria Cerâmica, onde está apresentado o Guia Simulador MotoSimEG-VRC. Neste guia, será possível observar o método de criação do projeto e do meio envolvente, das ferramentas, como editar/definir dados da ferramenta, como criar um programa e alterar a programação de um robô, adicionar passareiras transportadoras e prensas, movimentar e programar dispositivos externos. Na parte final do mesmo guia, abordar-se-ão as entradas e as saídas, os *scripts* e a simulação.

Ainda no mesmo capítulo – o terceiro – encontra-se descrita a Automação nos Processos de Vidragem e nos Processos de Inspeção. Adicionalmente serão mencionadas as Feiras nas quais participei e os artigos científicos escritos, que se revelam de extrema importância para o desenrolar do presente relatório.

Por fim, o último e quarto Capítulo estará assente nas conclusões e no raciocínio crítico quanto ao trabalho futuro.

2. ESTADO DA ARTE

Atualmente, devido ao avanço tecnológico, existe uma ampla variedade de sistemas automatizados aplicados na indústria e nas suas áreas envolventes, contribuindo para a realização de tarefas repetitivas e nocivas para a saúde dos trabalhadores. A utilização destes sistemas aumenta também a produção, reduz custos e melhora tanto a qualidade do serviço como a do produto (Magalhães & Ferreira, 2022).

2.1. Automação Industrial

A Automação consiste em criar um sistema automático com recurso a mecanismos que verificam o seu próprio funcionamento, efetuando medições e introduzindo correções, sem a necessidade da interferência do homem. A automação pode ser definida como uma tecnologia que utiliza comandos programados para operar um dado processo, combinados com o uso de informação para determinar que os comandos sejam executados corretamente, utilizada em processos antes operados por seres humanos. É a aplicação de técnicas computadorizadas ou mecânicas, para diminuir o recurso de mão de obra em inúmeros processos, utilizando vários recursos como o PLC – sensores, atuadores, robôs, entre outros aplicados nas linhas de produção. A automação tem diversas vantagens, contudo o objetivo maior compreende a diminuição dos custos e o aumento da velocidade da produção (Magalhães & Ferreira, 2022). Na Figura 1 é possível observar a inclusão de sistemas de visão, bem como sistemas robotizados a efetuar um processo de inspeção.

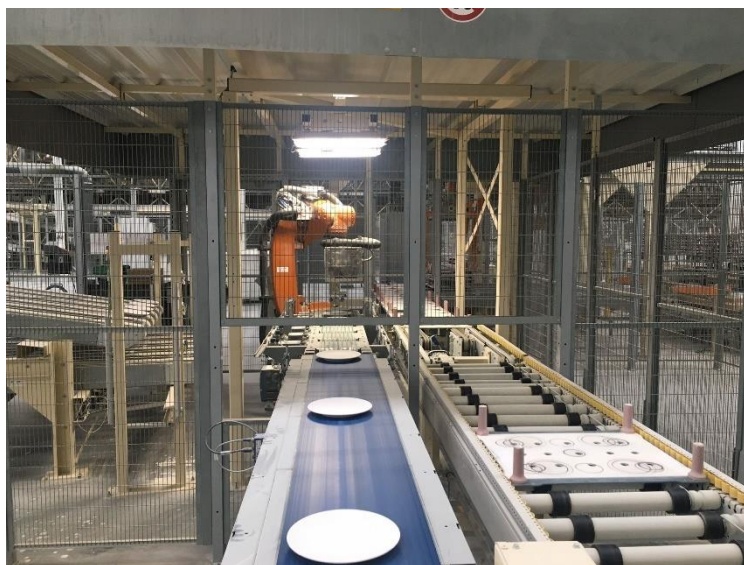


Figura 1 – Linha automatizada na indústria de cerâmica. Fonte: Própria

Um sistema automatizado contribui para o aumento da competitividade de uma empresa pelo meio de:

- Redução de recursos humanos, com a automatização das linhas de produção e o controlo da empresa, logo serão necessários menos operários. Por outro lado, aumentam os custos de implementação das soluções de automação sendo necessária uma maior qualificação humana para a operação dos diversos equipamentos;
- Aumento da qualidade dos produtos, sendo que um sistema automatizado oferece maior precisão que o homem, logo ter-se-á melhores características de repetibilidade e garantia de qualidade constante;
- Redução de custos de *stocks*, pois como a produtividade é aumentada, não há necessidade de grandes *stocks*;
- Redução do número de produtos perdidos;
- Menor tempo gasto no projeto e fabricação de novos produtos caso os equipamentos obtidos sejam programáveis e aptos a desempenhar diferentes operações;
- Modificações no produto são facilmente implementadas;
- Resposta rápida às solicitações do mercado (Godoi, 2022).

A automação não necessita de recorrer a robôs, dado que esta pode recorrer a sistemas inteligentes de supervisão ou produção, controlo de qualidade e muitos outros. É responsável pela integração de diversos e complexos sistemas (Godoi, 2022).

No que concerne às suas vantagens, a automação inclui maiores taxas de produção, maior produtividade, uso mais eficiente de materiais, melhor qualidade do produto, segurança, semanas de trabalho mais curtas para a mão de obra e prazos de entrega de fábrica reduzidos. Os principais motivos para a introdução da automação compreendem uma maior produção e maior produtividade. Apesar da alta versatilidade dos seres humanos, os sistemas automatizados executam o processo de fabricação com menos variabilidade do que os trabalhadores humanos, resultando em maior controlo e consistência da qualidade do produto. Com a automatização do processo e uso de materiais mais eficientes, verifica-se uma diminuição dos desperdícios por defeito (Godoi, 2022).

Na Figura 2 verifica-se um sistema totalmente automatizado de paletização de pratos na indústria de cerâmica.



Figura 2 – Paletização de pratos de cerâmica. Fonte: Própria

Outro motivo para proceder à automatização é a segurança do trabalhador. Os sistemas automatizados geralmente removem os trabalhadores do local de trabalho, protegendo-os assim contra os perigos do ambiente da fábrica. Com a automatização de processos, pode remover-se a interação de mão de obra humana em ambientes nocivos para a saúde humana.

Outra vantagem da automação é a redução do número de horas trabalhadas em média por semana pelos operários. A mecanização e a automação acarretam um papel significativo nessa redução. Finalmente, o tempo necessário para processar uma ordem de produção típica na fábrica é, geralmente, reduzido com a automação (Paulo, 2022).

No que respeita às desvantagens da automação, a mais frequente reflete-se na deslocação do trabalhador. Apesar dos benefícios sociais que podem resultar da reconversão dos trabalhadores deslocados para outros empregos, em quase todos os casos o trabalhador cujo emprego foi assumido por uma máquina, passa por um período de *stress* emocional.

Outras desvantagens do equipamento automatizado incluem o alto dispêndio de capital necessário para investir em automação, visto que um sistema automatizado pode custar milhões de euros para projetar, fabricar e instalar. Para além disso, exige um nível mais alto de manutenção comparativamente com uma máquina operada manualmente e, geralmente, menor grau de flexibilidade em termos dos produtos possíveis, que diverge de um sistema (Lima, 2017).

A automação da indústria é uma grande vantagem tanto para a indústria como para os trabalhadores, sendo que as operações consideradas perigosas e que exigem um esforço físico do homem estão a ser executadas por robôs. Não obstante, a automação promove também a inclusão de trabalhadores portadores de deficiência física, que seria impossível num ambiente convencional (Nascimento, 2018).

2.1.1. Evolução da automação industrial

A automação industrial está em constante evolução. No passado, os trabalhos eram executados através de recursos humanos, todavia, atualmente, a maioria dos trabalhos na indústria são executados com recurso a máquinas automatizadas. Antes da indústria ser automatizada existia menos qualidade de vida, os trabalhos eram mais pesados, cansativos e as remunerações baixas, sendo que o Homem sempre procurou automatizar. Automatizar significa, genericamente, dispensar o Homem ou outros seres vivos da realização de tarefas (Hitomi, 1994).

Esta ideia passou por diversas evoluções ao longo do tempo. A Figura 3 ilustra a evolução da indústria desde o surgimento do conceito de industrialização, nos finais do século XVIII, (Indústria 1.0), até ao aparecimento do conceito de Indústria 5.0.

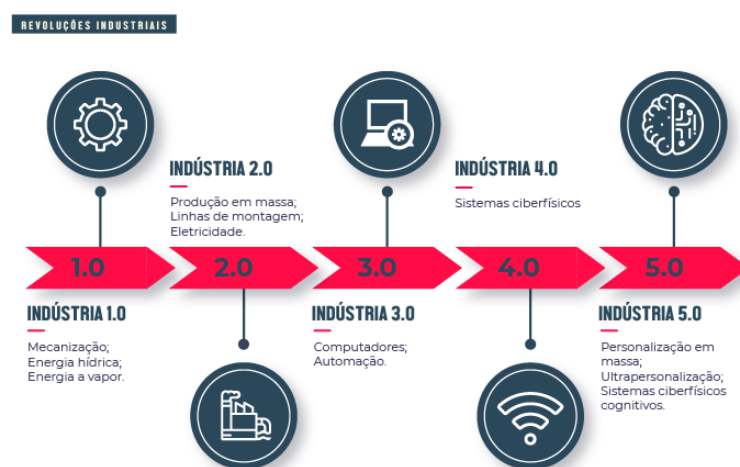


Figura 3 – Evolução da indústria. Fonte: (Disner & Cipullo, 2021)

2.1.1.1. Indústria 1.0

A Indústria 1.0, conhecida como a 1ª revolução industrial, coincide com o surgimento das máquinas a vapor, sendo esta a principal forma de automação da época. A principal indústria na época era a indústria têxtil, logo a primeira máquina a ser alimentada a vapor de água foi um tear mecânico, em 1784, como representado na Figura 4.

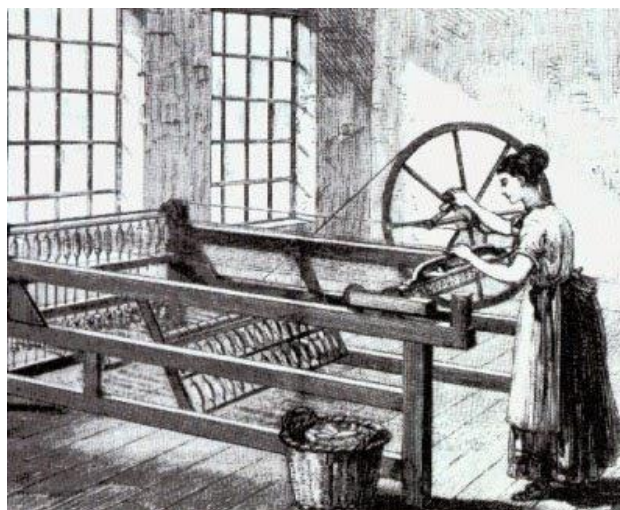


Figura 4 – Tear mecânico Fonte: (Nunes, 2015)

Nesta época, as unidades industriais existentes eram escassas e não concorriam entre si, levando a que as instalações fossem destinadas a fabricar um dado produto durante muitos anos, ao fim do qual se impunha uma total reconfiguração do espaço, com custos muito elevados para fabricar um novo produto. Por outro lado, a mão de obra era barata, abundante, sem capacidade de reivindicar por melhores salários, horários ou condições de trabalho, o que fazia com que não houvesse necessidade de automatizar (Vinitha, 2020).

2.1.1.2. Indústria 2.0

Na indústria 2.0 o uso de energia elétrica, o uso do motor à explosão, dos corantes sintéticos, a produção do aço e do alumínio em escala, e a invenção do telégrafo estipularam a exploração de novos mercados e a aceleração do ritmo industrial.

Surgiu também o início ao Fordismo, termo criado por Henry Ford em 1914, que se refere aos sistemas de produção em massa e ao consumo em massa. Este modelo revolucionou a indústria automobilística a partir de janeiro de 1914, quando Ford introduziu a primeira linha de montagem automatizada.

Para conseguir criar os sistemas de produção em massa, desenvolveram-se os primeiros automatismos, sistemas que permitem a realização automática de operações. Estes foram implementados recorrendo a uma lógica cablada, baseada no funcionamento de relés. Na Figura 5 está representado um exemplo de um circuito sequencial com recurso a relés (Sakurai & Zuchi, 2018).

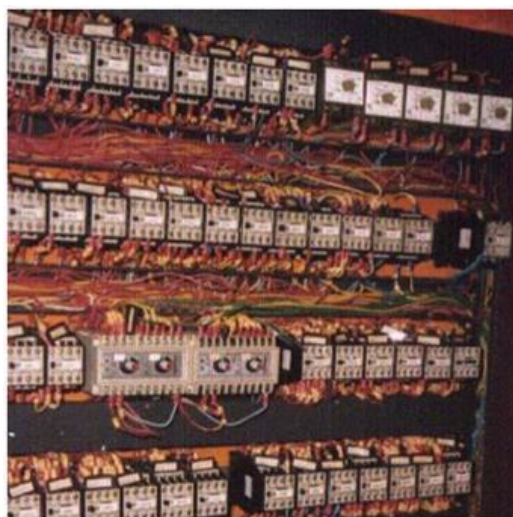


Figura 5 – Circuito sequencial com recurso a relés Fonte: (Souza, 2021)

Para o funcionamento dos automatismos é determinada a ligação dos diversos condutores entre os acionadores, como por exemplo relés, temporizadores, entre outros. Contudo, a lógica cablada tem alguns problemas relacionados com a modificação do sistema, qualquer alteração no funcionamento do automatismo implica modificar a cablagem, e inserir ou trocar componentes. Além disso, para executar essa alteração é obrigatório parar o processo de fabrico, o que terá grande impacto para custo final, visto que não é apenas a alteração do automatismo. Na maioria dos casos, a única forma de modificar estes sistemas após a sua instalação passa por voltar ao ponto de partida, por isto esta automação é designada por automação fixa (Souza, 2021).

2.1.1.3. Indústria 3.0

A Terceira Revolução Industrial, também conhecida por Revolução Técnico-Científica e Informacional, é um processo de inovação tecnológica marcado pelos avanços no campo da Informática, da Robótica, das Telecomunicações, dos Transportes, da Biotecnologia e Química, além da Nanotecnologia. Apesar de não haver consenso entre os especialistas sobre o seu início, a maioria dos autores data a década de 1970 como determinante para alavancar esse período no mundo da indústria.

A indústria 3.0 foi estabelecida com a entrada dos microprocessadores na indústria e dos primeiros protocolos de comunicações industriais.

O primeiro grande impacto desta indústria foi o surgimento do Autómato Programável ou Controlador Lógico Programável – PLC, que será abordado no subcapítulo. 2.4.

A tecnologia industrial neste ponto, dividiu-se em 5 níveis diferentes:

- **Nível 0:** É constituído pelos sensores e atuadores presentes na indústria;
- **Nível 1:** Este nível é responsável pelo controlo efetuado pelo PLC, este recebe informações dos sensores nas suas entradas e emite sinais para os atuadores pelas suas saídas. Para além de ser capaz de enviar e receber dados do nível inferior, também é capaz de comunicar com o nível acima. Para efetuar essas comunicações foram desenvolvidos diversos protocolos de comunicação que permitiram recolher os dados referentes aos níveis 0 e 1, e partilhá-los com outro PLC (comunicação dentro do mesmo nível) ou com computadores industriais (nível 2).
- **Nível 2:** Responsável pela supervisão, é constituído por computadores industriais integrados em HMIs Interfaces Humano/Máquina, que será uma tecnologia abordada num subcapítulo futuro. Estes conseguem comunicar diretamente com os níveis inferiores e funcionam como *gateway* para comunicar com os níveis superiores.
- **Nível 3:** Ou sistemas de execução da produção *Manufacturing Execution System* (MES): trata-se de um *software* que faz a gestão e monitoriza o que decorre numa fábrica.
- **Nível 4:** Responsável pelo planeamento dos recursos da empresa *Enterprise Resource Planning* (ERP) é o *software* que permite o planeamento e logística do negócio de uma ou mais unidades de produção (Tantawi, 2019).

Na Figura 6 abaixo estão representados os diferentes níveis anteriormente mencionados.

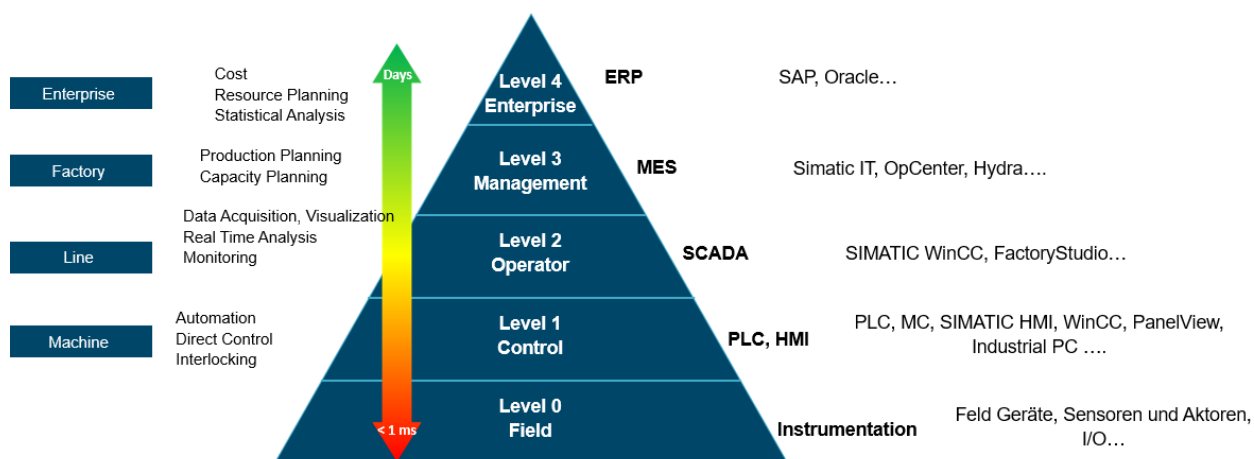


Figura 6 – Pirâmide níveis de automação Fonte: (Hammari, 2021)

Em resumo, na indústria 3.0, com a introdução de todas as tecnologias foi possível efetuar muitas alterações ao processo de fabrico, permitindo ainda alterações em tempo real e sem a necessidade de haver uma paragem na linha. Este conceito é denominado por automação flexível, e é possível devido aos próprios controladores e aos robôs, capazes de se adaptarem às necessidades.

2.1.1.4. Indústria 4.0

A Quarta Revolução Industrial denominada como Indústria 4.0 é um conceito que representa a automação industrial e a integração de diferentes tecnologias, como a inteligência artificial, robótica, Internet das coisas e computação em nuvem. Este conceito surge com o objetivo de promover a digitalização das atividades industriais, melhorando os processos e aumentando a produtividade. De seguida apresentam-se, detalhadamente, as diversas tecnologias introduzidas na indústria 4.0:

- **Inteligência artificial:** Consiste na aplicação de análise avançada e técnicas baseadas em lógica, incluindo aprendizagem de máquinas, para interpretar eventos, analisar tendências e comportamentos de sistemas, apoiar e automatizar decisões, e, por fim, realizar ações.
- **Computação em nuvem:** É a distribuição de serviços de computação – servidores, armazenamento, bases de dados, redes, *software*, análises, inteligência – pela Internet, com utilização de memória, capacidade de armazenamento e cálculo de computadores e servidores instalados em *datacenters*, proporcionando recursos flexíveis e economias de escala. A computação em nuvem permite às empresas aceder a recursos computacionais abundantes como um serviço a partir de distintos dispositivos remotos.
- **Big data:** É uma abordagem para atuar em dados com maior variedade e complexidade, que chegam em volumes crescentes e com velocidade, usados para resolver problemas de negócios. São utilizadas técnicas estatísticas e de inteligência artificial para extrair informações relevantes aos negócios, inferências e tendências.
- **Cyber segurança:** É um conjunto de infraestruturas de *hardware* e *software* voltado para a proteção dos ativos de informação, por meio do tratamento de ameaças que põem em risco a informação que é processada, armazenada e transportada pelos sistemas de informação que estão interligados.
- **Internet das coisas:** Interconexão entre objetos por meio da Internet. Estes objetos (sensores, *softwares* e outras tecnologias), são organizados em redes que comunicam e interagem, podendo ser remotamente monitorados e/ou controlados, resultando em ganhos de eficiência.
- **Robótica avançada:** Robôs autónomos capazes de comunicar com diferentes dispositivos.

- **Manufatura digital:** É o uso de um sistema integrado, baseado num computador, que consiste em simulação, visualização 3D, análises e ferramentas de colaboração para criar definições de processos de manufatura e produto (Siemens, n.d.).
- **Manufatura aditiva:** É um método de manufatura que permite produzir artigos de forma imediata, com um modelo digital e sem necessidade de moldes. Associada à impressão 3D, essa tecnologia chegou para revolucionar a indústria com processos mais eficientes (Veit, 2018).
- **Integração de sistemas:** União de diferentes sistemas de computação e aplicações de *hardware*, para atuar como um todo coordenado, e que possibilita a troca de informações entre os diferentes sistemas. Permite a empresas um olhar abrangente sobre o seu negócio. As informações em tempo real sobre o processo produtivo influenciam a tomada de decisões de gestão mais rapidamente, bem como decisões estratégicas sobre o negócio da empresa, de modo a serem mais facilmente implantadas no processo de produção. Somente a instalação de pacotes ERP não se enquadram, mas a sua integração a sistemas de controlo da produção industrial sim.
- **Sistemas de simulação:** Utilização de computadores e conjunto de técnicas para gerar modelos digitais que descrevem ou exibem a interação complexa entre várias variáveis dentro de um sistema, imitando processos do mundo (Hammari, 2021).
- **Digitalização:** Consiste em tecnologias digitais para transformar processos de produção, de desenvolvimento de produtos e/ou modelos de negócios, visando a otimização e eficiência nos processos. A transformação digital abrange: projeto e implementação de plano de digitalização, instalação de sensores, aquisição e tratamento de dados (Demir & Cicibaş, 2017).

A Figura 7 representa a interligação das diversas tecnologias acima referidas.



Figura 7 – Principais tecnologias na indústria 4.0 Fonte:(Tecnicon, 2020)

Deste modo, as vantagens e desafios da indústria 4.0 são a de tornar a produção mais flexível e customizada, aumentar a produtividade, aumentar a eficiência na utilização de recursos e de energia, diminuir desperdícios, ligar vários dispositivos de diferentes processos logísticos, otimizar as decisões do processo, identificar e prevenir problemas com a manutenção preventiva, facilitar novas oportunidades de negócio e melhorar as condições laborais (Bahrin, 2016).

2.1.1.5. Indústria 5.0

A Quinta Revolução Industrial é um aperfeiçoamento da quarta revolução, onde o foco está no desenvolvimento da inteligência artificial, sustentabilidade e personalização mais humana, aspetos representados na Figura 8.

Ao contrário da indústria 4.0, que tem como objetivo principal maximizar os lucros sem ter em conta a justiça social ou até mesmo a sustentabilidade, esta indústria promove a utilização da tecnologia para a obtenção de objetivos sociais que vão para além do aumento de lucros e manutenção de empregos.



Figura 8 – Principais objetivos da indústria 5.0 (Xu, 2021, p. 0)

Assim sendo, esta indústria recorre ao uso de robôs colaborativos de modo a reintroduzir o trabalho humano na indústria, possibilitando a criação de customização em massa. A mão de obra passa assim a ser vista como um investimento, e não como um custo extra (Xu, 2021, Demir & Cicibaş, 2017).

2.1.2. Automação fixa

A automação fixa realiza um único conjunto de tarefas sem desvios. Este tipo de automação industrial é caracterizado pela rigidez da configuração do equipamento, ou seja, uma vez criada uma determinada configuração de controlo, não é possível alterá-la posteriormente sem realizar um novo projeto.

2.1.3. Automação programável

A automação programável é executada por meio de comandos entregues através de um *software* programável. Isso significa que os processos resultantes podem variar muito com a mudança de instruções dadas. No entanto, como os esforços de programação não são triviais, os processos, nomeadamente as tarefas, não se alteram significativamente. Esse tipo de automação é comum em configurações de produção em massa que forjam tipos semelhantes de produtos, que utilizam muitas das mesmas etapas e ferramentas, como em fábricas de papel ou laminadoras de aço.

O processo de fabricação compatível com este sistema seria a fabricação repetitiva, em que os mesmos produtos são produzidos por um longo período de tempo e em grandes lotes, com necessidades reduzidas de supervisão humana. São normalmente usados na fabricação de automóveis e máquinas.

A configuração inicial do equipamento de automação programável pode exigir um alto custo. Não obstante, como os processos são contínuos e relativamente imutáveis, tendem a ser mais baratos a longo prazo (Admin, 2019).

2.1.4. Automação flexível

A automação flexível é uma extensão da automação programável, sendo a sua definição exata ainda alvo de atualizações.

Este sistema de automação fornece o equipamento de controlo automático, que oferece uma grande flexibilidade para modificações no *design* do produto. Essas mudanças podem ser realizadas de forma ágil, através de comandos fornecidos via código, pelos operadores.

Um sistema flexível de produção tem a capacidade de gerar uma variedade de produtos, sem perda significativa de tempo de produção para ajustes, em eventos de introdução de novos produtos (Admin, 2019).

Assim, o sistema pode produzir várias combinações de produtos sem a necessidade de organizá-los em lotes separados.

Os processos de fabricação compatíveis com este sistema seriam:

- **Fabricação Discreta:** Permite variações dentro do processo de fabricação, embora limitada, por exemplo, em embalagens de alimentos ou na indústria têxtil;
- **Manufatura Job Shop:** Ocorre dentro de áreas de produção definidas e é mais intensiva em mão de obra, em comparação com outras formas de manufatura. Um exemplo seria fazer máquinas personalizadas;
- **Processo de Fabricação de Lotes:** As matérias-primas movem-se pela linha de produção em lotes, de modo que haja uma pausa entre cada etapa à medida que um lote se move (por exemplo, na indústria farmacêutica e na fabricação de tintas);
- **Processo de Fabricação Contínuo:** Oferece um processamento consistente, em que o processo de fabricação não sofre alterações em nenhum momento. Este tipo de fabricação é comumente usado na fabricação de alimentos e bebidas, bem como na fabricação de petróleo e gás (Zimm, 2020).

2.1.5. Automação integrada

A automação integrada envolve a automação total das fábricas, pois é totalmente manipulada por computadores e processos de controlo, com o mínimo de envolvimento humano. Os computadores podem projetar as peças necessárias, testar os projetos e fabricar as peças. A automação integrada, tal como a automação flexível, é compatível tanto com a fabricação de processo em lote quanto com a fabricação de processo contínuo (Zimm, 2020).

As tecnologias que usam esse tipo de automação incluem:

- Planeamento de processos auxiliado por computador;
- Projeto e fabricação suportados por computador;
- Máquinas-ferramentas de controlo numérico computadorizado;
- Controlo informatizado de produção e programação;
- Sistemas automáticos de armazenamento e recuperação;
- Sistemas de máquinas flexíveis;
- Sistemas automatizados de manuseamento de materiais (por exemplo: robôs);
- Correias transportadoras e guindastes automatizados.

2.2. Robótica Industrial

A robótica é um ramo da engenharia que envolve a conceção, projeto, fabricação e operação de robôs. O objetivo do campo da robótica é criar sistemas autónomos que possam auxiliar os humanos de várias maneiras.

Um robô industrial é um sistema de robô usado para fabricação. São automatizados, programáveis e capazes de movimento em três ou mais eixos.

As aplicações típicas de robôs incluem robôs de soldadura, pintura, montagem, desmontagem, *pick and place*, embalagem e rotulagem, paletização, inspeção de produtos e testes. Importa realçar que todas estas aplicações são executadas com alta persistência, velocidade e precisão.

A robótica pode assumir várias formas. Um robô pode-se assemelhar a um humano ou pode apresentar-se sob a forma de uma aplicação robótica, como a automação de processos robóticos - *Robotic Process Automation* (RPA), que simula como os humanos se envolvem com o *software* para executar tarefas repetitivas e baseadas em regras.

O termo ‘robótico’ é uma extensão da palavra ‘robô’. Um de seus primeiros usos veio do escritor checo *Karel Čapek*, que usou a palavra na sua peça “Robôs Universais de Rossum”, em 1920.

No entanto, foi o autor de ficção científica *Isaac Asimov* que recebeu crédito por ser a primeira pessoa a usar o termo na década de 1940, pelo *Oxford English Dictionary*.

Na história de *Asimov*, ele sugeriu três princípios para orientar o comportamento de robôs autônomos e máquinas inteligentes (Gasparetto & Scalera, 2019).

As Três Leis da Robótica de *Asimov* sobreviveram até ao presente:

1. Os robôs nunca devem prejudicar os seres humanos;
2. Os robôs devem seguir as instruções dos humanos sem violar a primeira regra;
3. Os robôs devem-se proteger sem violar as outras regras.

No entanto, foi apenas algumas décadas depois, em 1961 - com base em *designs* dos anos 50 - que o primeiro robô programável, *Unimate*, representado na Figura 9, foi criado para mover peças de metal de uma máquina de fundição (Segovia, 2013, Engelberger, 1979).



Figura 9 – Robô *Unimate* Fonte: (Engelberger, 1979)

Embora o campo da robótica e a exploração dos potenciais usos e funcionalidades dos robôs tenham crescido substancialmente no século XX, o conceito não é recente. A robótica desenvolve máquinas que podem substituir humanos e replicar ações humanas. Os robôs podem ser empregues em diversas situações e para diversos propósitos, contudo, atualmente, grande parte deles são usados em ambientes considerados perigosos (incluindo inspeção de materiais radioativos, detecção e desativação de bombas), processos de fabricação ou onde os seres humanos não podem sobreviver (por exemplo: no espaço, debaixo de água, em altas temperaturas e para efetuar limpeza e contenção de materiais perigosos e radiação).

Os robôs podem assumir qualquer forma, mas alguns são concebidos se apresentarem similares aos humanos, na sua aparência. Este facto coadjuva a aceitação de robôs em certos comportamentos replicativos, que geralmente são realizados por pessoas (Gasparetto & Scalera, 2019).

2.2.1. Elementos de um robô industrial

As quatro partes principais de um robô industrial são o manipulador, o controlador, o dispositivo de interface humana e a fonte de alimentação.

2.2.1.1. Manipulador

Um manipulador industrial é uma combinação de elementos estruturais rígidos (corpos ou elos), conectados entre si através de articulações (juntas), este último composto por braços, sensores, atuadores.

Os braços dos robôs, representados na Figura 10 (especificamente o modelo *KUKA*), são constituído por partes mecânicas rígidas denominadas por elos, que são interligados por juntas. Através da cinemática, estas definem a posição cartesiana (x, y, z, a, b, c) do robô. As juntas são de rotação - (R), ou prismática/de translação - (P).



Figura 10 – Robôs indústrias *KUKA* Fonte: (*Robôs industriais*, n.d.)

Os sensores são responsáveis por obter informações do meio envolvente, e transmiti-las de forma que seja possível ao controlador processar. Por este motivo, os robôs apresentam-se equipados com diferentes sensores de série, que variam consoante o tipo de robô, o modelo, a aplicação e o fabricante.

No entanto, existem sensores que são comuns a todos os robôs, como por exemplo os *encoders* absolutos, em cada um dos seus eixos. Os *encoders* absolutos permitem ao controlador do robô medir o seu ângulo, velocidade e posição, que se revelam elementos-chave para o cálculo da cinemática do robô. Existem ainda entradas disponíveis para se conectar sensores externos ao controlador do robô, permitindo que estes transmitam informação de algo externo ao sistema robótico, nomeadamente o estado da sua garra - aberta ou fechada (Santos, 2003).

Os atuadores também estão presentes nos punhos dos robôs e, tal como referido anteriormente, estes representam os músculos dos robôs. Ao contrário dos sensores, os atuadores transformam a informação digital do controlador numa atuação no mundo real. Tal como os sensores, os atuadores presentes em cada robô variam consoante o tipo de robô, o modelo, a aplicação e o fabricante, sendo possível usar atuadores elétricos, pneumáticos e hidráulicos (Gruver, 1984).

2.2.1.2. Controlador

O controlador é a unidade capaz de interpretar a informação de sensores, assim como gerar informação para ativar os atuadores do robô. Nalguns casos, não existe uma separação entre a unidade de potência do robô, sendo que esta fica acoplada no controlador, fornecendo assim energia aos restantes componentes do mesmo.

Atualmente, muitos destes controladores consistem em computadores com um *software* específico para o robô (*KUKA KR C4*, n.d.). Na Figura 11, abaixo, estão representados vários controladores *KUKA*.



Figura 11 – Diversos controladores *KUKA* Fonte: (*KUKA KR C4*, n.d.)

2.2.1.3. HMI de um robô

Os HMIs de um robô são dispositivos portáteis que se apresentam, usualmente, com ou sem fio, normalmente incluídos no sistema de controlo de um robô industrial. Estes podem conter vários botões ou interruptores, ou apresentar concomitantemente uma tela sensível ao contacto, como é o caso dos modelos mais recentes. Não obstante, apresentam um *display* que ilustra os comandos do robô, e permite a edição desses comandos. Além disso, o visor pode ser usado para recuperar o histórico de comandos do robô. Os dispositivos HMI utilizam um teclado para entrada de tarefas e uma programação facilitadora. Outra característica comum dos HMI é a existência de um botão vermelho, que é o botão de emergência do sistema robótico. É geralmente usado em caso de mau funcionamento do robô industrial e interrompe imediatamente a operação mediante a sua ativação, garantindo a segurança de todos os trabalhadores e/ou equipamentos de fabricação ao redor do robô (Santos, 2003).

2.2.2. Tipos de robôs industriais

Os seis principais tipos de robôs industriais mais conhecidos atualmente são: o *Cartesiano*, o *SCARA*, o *Cilíndrico*, o *Delta*, o *Esférico* e o *Verticalmente Articulado*. Estes estão disseminados na indústria, e enfatizam a automatização de tarefas, o aumento da segurança do trabalhador e o aumento da produção em geral, reduzindo custos operacionais e o desperdício.

2.2.2.1. Articulados

Os robôs com esta configuração estão ilustrados na Figura 12, e contém pelo menos três juntas de rotação (RRR). O eixo de movimento da junta de rotação da base é ortogonal às outras duas juntas de rotação, que são simétricas entre si. Esta configuração permite maior mobilidade aos robôs.



Figura 12 – Robôs articulados Fonte: (Silveira, 2019)

Esta configuração de robôs recorre exclusivamente a eixos de movimento rotativo. A sua programação é complexa e apresenta uma expedita manobrabilidade. Não obstante, atinge velocidades elevadas, um considerável alcance (dependendo das dimensões do mesmo) e fácil acesso ao espaço frontal, lateral, superior e retaguarda. Além disso, tem a possibilidade de utilização de cinemática aberta e de cinemática fechada.

Esta configuração é a mais utilizada na indústria devido a sua versatilidade, e tem diferentes tipos de aplicações como: embalagem de alimentos, soldadura, mobilização de materiais, montagem automotiva, paletização, manipulação de vidro, entre outras (Silveira, 2019).

2.2.2.2. Cartesianos

O Robô Cartesiano (presente na Figura 13) possui três juntas prismáticas (PPP), e ostenta um movimento composto por três translações, cujos eixos de movimento são coincidentes com um sistema de coordenadas de referência cartesiano. Uma variante deste tipo de robô é a configuração tipo Pórtico.



Figura 13 – Robôs cartesianos Fonte: (Silveira, 2019)

É geralmente usado para tarefas de *pick and place*, devido à sua alta precisão, facilidade de programação, ser capaz de se deslocar ao longo de superfícies significativas, e com capacidade de carga deveras elevada. (Silveira, 2019)

2.2.2.3. Cilíndricos

Na configuração cilíndrica, representada na Figura 14, os eixos de movimento podem ser descritos no sistema de coordenadas de referência cilíndrica. É formado por duas juntas prismáticas e uma de rotação (PPR), arquitetando movimentos de duas translações e uma rotação.

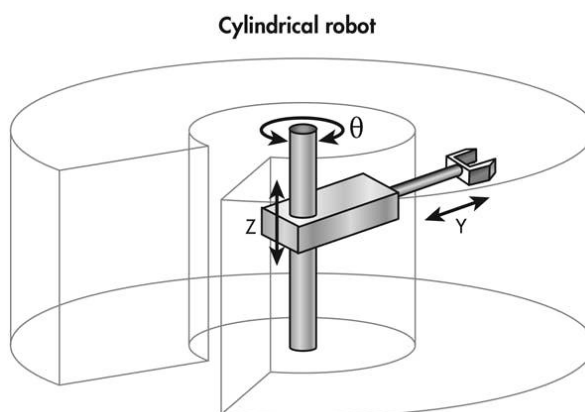


Figura 14 – Configuração robô cilíndrico Fonte: (Silveira, 2019)

Com a combinação de dois eixos ortogonais de movimento linear com um eixo de movimento rotativo, este sistema oferece fácil controlo, programação simples, boa precisão, velocidades de operação elevadas, boa acessibilidade frontal e lateral, e uma estrutura simples que proporciona uma elevada fiabilidade.

2.2.2.4. Esféricos

Os eixos de movimento de um Robô Esférico (Figura 15) formam um sistema de coordenadas de referência polar, através de uma junta prismática e duas de rotação (PPR), compondo movimentos de uma translação e duas rotações. O volume de trabalho gerado é aproximadamente esférico.

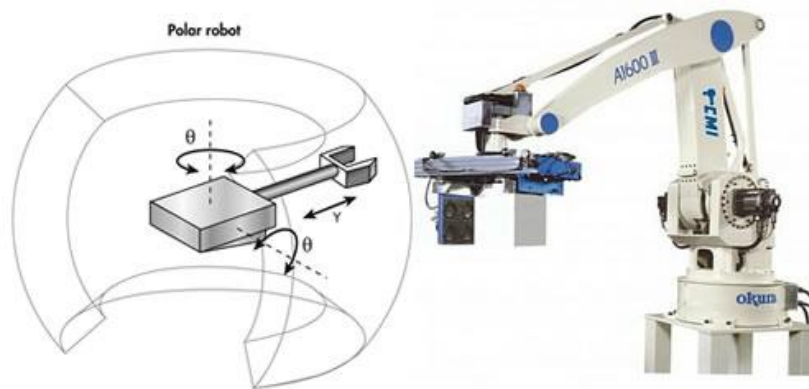


Figura 15 – Robô esférico Fonte: (Silveira, 2019)

Os robôs esféricos agregam dois movimentos de rotação com um movimento linear, num sistema de coordenadas polar, asseverando um sistema de controlo e programação simples, elevado espaço de trabalho, velocidades de operação elevadas, elevada capacidade de carga, de precisão e de repetibilidade.

2.2.2.5. SCARA

Os robôs *Selective Compliance Assembly Arm* (SCARA), representados na Figura 16, retratam duas juntas de rotação dispostas em paralelo, concretizando um movimento num plano, e uma junta prismática, perpendicular a este plano (PRR), apresentando, portanto, uma translação e duas rotações. O volume de trabalho é aproximadamente cilíndrico (Silveira, 2019).



Figura 16 – Robô SCARA Fonte: (Epson, n.d.)

Estes robôs apresentam as seguintes vantagens:

- Estrutura rígida na direção vertical;
- Elevada manobrabilidade;
- Elevada velocidade de trabalho;
- Precisão e repetibilidade elevadas.

Esta configuração de robôs é geralmente utilizada em tarefas de montagem automatizada.

2.2.2.6. Delta

Os Robôs Delta (Figura 17) apresentam configuração do tipo plataforma e mecanismos em forma de cadeia cinemática fechada. Possui três juntas prismáticas (PPP) ou três juntas de rotação (RRR).



Figura 17 – Robô Delta. Fonte: Silveira, 2019

Os robôs Delta, também denominados de robôs paralelos, têm uma morfologia tipicamente aracnoide, possuindo vários elementos de ligação agregados a uma placa principal, tal como se encontra representado na figura acima. Estes são comumente usados em processos de *pick and place* que carecem de uma velocidade muito elevada e alta precisão. Porém, este tipo de robô logra uma operação complexa, que implica um controlador dedicado a este tipo de programação (Silveira, 2019).

2.2.3. Robótica colaborativa

A robótica colaborativa é uma tecnologia recente, que permite que os robôs partilhem o espaço de trabalho com os operadores, de forma colaborativa. Isto só é possível pois os robôs possuem sensores que outorgam a detecção de contacto com qualquer tipo de obstáculo, privilegiando a segurança.

O impacto na indústria é impetuoso, uma vez que esses robôs acarretam o potencial de realizar a mesma tarefa que os recursos humanos, sem necessidade de recorrer a alteração da configuração do ambiente fabril. Assim sendo, as fabricas aumentam a produtividade com uma quantidade similar de funcionários (Sherwani, 2020).

2.2.3.1. Robôs colaborativos

Os Robôs Colaborativos, também denominados de *Cobots* (refletidos na Figura 18), são a vigente tecnologia em robótica, e converteram consideravelmente a automação. Garantem a segurança dos trabalhadores, sendo, portanto, colaborativos. São equipados com sensores sensíveis, que aumentam a sensibilidade do robô. Se o robô sofrer uma interrupção no decorrer da sua operação, este entrará em modo de segurança. Os robôs industriais não possuem esta capacidade, pelo que a possibilidade de ocorrência de acidentes aumenta (Sherwani, 2020).



Figura 18 – Robô colaborativo Fonte: (mercado, 2019)

2.3. Sistemas de visão presentes na automação

Os sistemas de visão refletem-se na tecnologia responsável por substituir ou complementar as inspeções manuais e medições, com câmaras digitais e processamento de imagens. A tecnologia é utilizada em diversas indústrias, para automatizar a produção, aumentar a velocidade de produção e rendimento, e melhorar a qualidade do produto. Na automação, a operação por qual o sistema de visão pode ser descrito por um fluxo de quatro etapas, representadas na Figura 19.

Numa primeira etapa, o sistema inicializa o seu procedimento através da aquisição de uma imagem, por intermédio de um sensor de imagem. Na segunda etapa, e assegurada a obtenção da imagem, segue-se o processamento da imagem e a análise da mesma. Este processamento e análise difere conforme o fim desejado. Na terceira fase, alcançados os resultados pretendidos, procede-se à comunicação do resultado para o sistema que controla o processo. Numa última etapa, desenrola-se a ação, dependendo do resultado do sistema de visão.

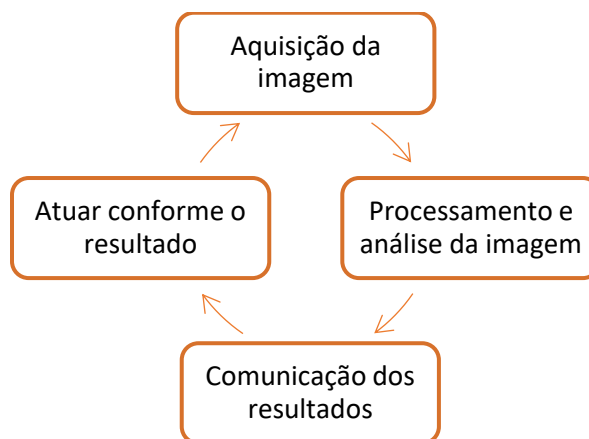


Figura 19 – Ciclo das etapas de um sistema de visão. Fonte: Própria.

Os sistemas de visão presentes em ambientes industriais, categorizam-se em quatro tipos de aplicações, nomeadamente: localizar, medir, inspecionar e identificar.

Na localização, o desígnio do sistema de visão repercute-se no ato de situar o objeto e reportar sua posição e orientação. Esta categoria é bastante comum em aplicações robotizadas, dado que para estas aplicações, a câmara atinge uma referência no objeto. Serve como exemplo: quando o ponto de referência é o centro de uma peça de cerâmica, em seguida, envia as informações para um robô que pega o objeto (Riku, 2006).

Nas aplicações de medição, o objetivo do sistema de visão é medir as dimensões físicas do objeto. Na indústria de cerâmica esta aplicação pode ser utilizada para medir o diâmetro, a curvatura, a área, a altura e o volume de peças de cerâmica.

Em aplicações de inspeção, a finalidade do sistema de visão é validar determinadas características, como por exemplo defeitos numa peça de cerâmica (excesso de vidrado, deformações, rachas, grânulos, fendas, poros, enrolamentos, entre outros).

A Figura 20, apresenta um exemplo de uma aplicação de um sistema de visão, na indústria da cerâmica presente no CTCV.

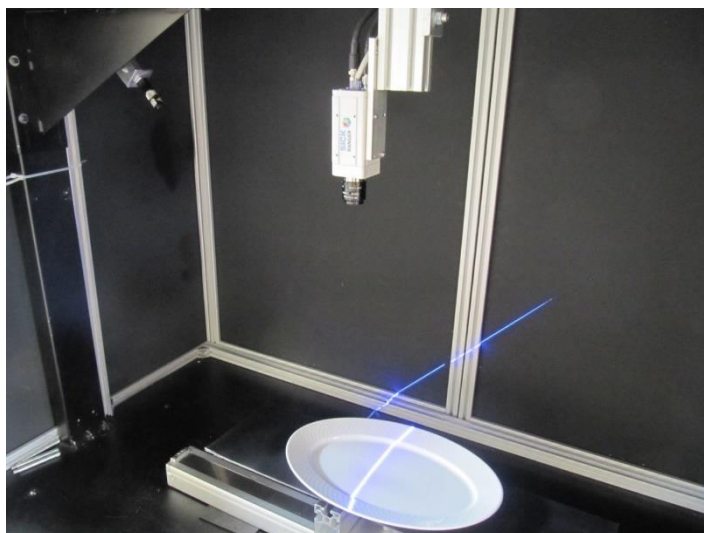


Figura 20 – Sistema de inspeção INSPECTWARE Fonte: Própria

Numa aplicação de identificação, o sistema de visão pode ser programado com o intuito de ler vários códigos, nomeadamente códigos QR e caracteres alfanuméricos (texto e números). A Figura 21, representa uma câmara que lê a data de validade de um alimento, num pacote.

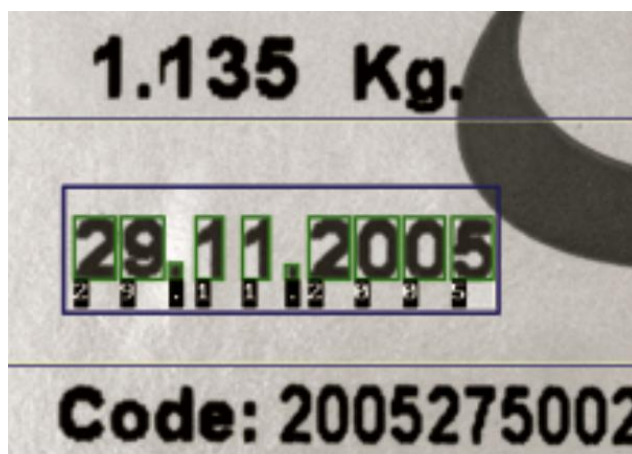


Figura 21 – Aplicação do Sistema de visão Fonte: (Riku, 2006)

2.4. PLC - Programmable Logic Controller

Um *Programmable Logic Controller* (PLC) é um equipamento eletrónico digital, compatível com aplicações industriais. O primeiro PLC surgiu em 1968, tendo sido concebido pela *General Motors*.

Os PLCs surgiram com a evolução dos antigos painéis elétricos, cuja lógica fixa tornava impraticável qualquer mudança extra do processo. A tecnologia do PLC é exequível, mediada por circuitos integrados, e com o acréscimo da evolução da lógica digital.

Este equipamento evidencia vantagens elementares:

- a) Fácil diagnóstico durante o projeto;
- b) Fácil de implementar em quadros devido ao seu tamanho reduzido;
- c) Podem ser reprogramados sem interromper o processo produtivo;
- d) Possibilidade de guardar programas em memória;
- e) Baixo consumo de energia;
- f) Manutenção reduzida;
- g) Liberdade para adicionar expansões de entradas e saídas;
- h) Capacidade de comunicação com diversos equipamentos, entre outras.

Os PLC são computadores industriais usados para controlar diferentes processos eletromecânicos para uso em manufatura, fábricas ou outros ambientes de automação, exemplo observável na Figura 22.



Figura 22 – PLC NJ301 Fonte: Própria

Variam em tamanho e são amplamente utilizados numa variedade de indústrias, visto que são rápidos, fáceis de operar e são considerados fáceis de programar. Não obstante, têm a potencialidade de ser programados de várias maneiras, desde a lógica *ladder*, que é baseada em relés eletromecânicos, até linguagens de programação especialmente adaptadas de BASIC e C. Atualmente, a maioria dos PLC usa uma das cinco linguagens de programação mencionadas de seguida: Diagrama *Ladder*; Texto Estruturado; Diagrama de Blocos Funcionais; Lista de Instruções ou Gráficos Sequenciais de Funções.

Os PLC atuam como interfaces físicas entre dispositivos presentes na fábrica, e um sistema SCADA ou HMI. Os PLC comunicam, monitorizam e controlam processos automatizados, como linhas de montagem, funções de máquinas ou dispositivos robóticos.

As funções de um PLC são divididas em três categorias principais: entradas, saídas e *Central Process Unit* (CPU). Os PLC capturam dados de um processo monitorizando as entradas, às quais as máquinas os dispositivos estão conectados. Os dados de entrada são então processados pela CPU, que aplica lógica aos dados, com base no estado das entradas. A CPU executa a lógica do programa criada pelo utilizador, e envia os dados ou comandos para as máquinas e dispositivos aos quais está conectada.

Existem dois tipos principais de entradas: entradas de dados de dispositivos e máquinas, e entradas de dados facilitadas por humanos. Os dados de entrada dos sensores e das máquinas são enviados ao PLC. As entradas podem incluir estados ‘liga/desliga’ para interruptores mecânicos, botões e codificadores. Os estados altos/baixos determinam temperaturas, sensores de pressão e detetores de nível de líquido. Contrariamente, os estados abertos/fechados atuam para bombas e valores.

As entradas para interação com os humanos incluem botões, interruptores e sensores de dispositivos como teclados, telas sensíveis ao toque, controlos remotos ou leitores de cartão. As saídas são as ações físicas ou resultados visuais que são baseados numa lógica de PLC, em resposta a essas entradas. As saídas físicas incluem motores de arranque, acender uma luz, drenar uma válvula, aumentar o aquecimento ou desligar uma bomba. As saídas visuais são enviadas para dispositivos como impressoras, projetores ou monitores (*What Is a PLC?*, 2020).

2.4.1. Classificação dos PLC

Os PLC podem ser classificados segundo a sua capacidade: nano e micro (possuem até 16 entradas e saídas). Normalmente são compostos por um único módulo com capacidade de memória máxima de 512 passos. Por outro lado, existem concomitantemente PLCs de médio porte, com capacidade de entrada e saída de até 256 pontos, digitais e analógicas. Permitem até 2048 passos de memória. Não obstante, evidenciam-se os PLC de grande porte, com construção modular, com CPU principal e auxiliares, especificamente módulos de entrada e saída digitais e analógicas, módulos especializados, módulos para redes locais. Permitem a utilização de até 4096 pontos. A memória pode ser otimizada para o tamanho requerido pelo utilizador.

2.4.2. Controlo de entradas e saídas

Para a criação de um sistema autónomo são necessários diversos equipamentos, nomeadamente no que concerne ao controlo dos mesmos. Estes são divididos em entradas e saídas, onde o controlo pode ser analógico ou discreto. As entradas e/ou saídas digitais são os elementos discretos, as entradas e/ou saídas analógicas são os elementos variáveis, baseados em valores conhecidos de tensão ou corrente. Para o controlo de entradas e saídas existem díspares tipos, como:

- **Entradas Discretas:** Fornecem apenas um pulso ao controlador (PLC), ou seja, possuem apenas a funcionalidade de ‘ligar’ ou ‘desligar’ (nível alto ou nível baixo), remontando a álgebra booleana que trabalha com uns e zeros.
- **Entradas Analógicas:** Medem as grandezas de forma analógica, o que requer que os controladores tenham conversores Analógico-Digitais (A/D). Exemplos de medidas que espelham a forma analógica são a temperatura e pressão.
- **Saídas Discretas:** Estas, através de um pulso do controlador, determinam o seu funcionamento, como é tipo de válvulas electropneumáticas.
- **Saídas Analógicas:** Dispõem de um sinal analógico, que varia continuamente ao longo do tempo, sendo necessário um conversor digital para analógico (D/A) para operar com este tipo de saída. Os exemplos mais comuns são: válvula proporcional, acionamento de motores DC e *displays* gráficos.

2.4.3. Arquitetura do autómato

Ao citar um PLC, normalmente faz-se referência a múltiplos componentes que se interligam entre si e que formam um PLC funcional. Entre esses componentes, existem cinco que se encontram presentes em todos os PLC, esquematizados na Figura 23, sendo eles:

- **Unidade Central de Processamento (CPU):** Igualmente designada como microprocessador, é o ‘cérebro’ do autómato, uma vez que realiza operações aritméticas e lógicas, bem como funções de controlo. Tem ainda a seu cargo a gestão dos periféricos e o diagnóstico dos defeitos que possam ocorrer internamente. Este também é o responsável por ler os valores lógicos das entradas presentes na memória, executar as instruções do programa e atualizar os valores das saídas na memória (Francisco, 2015);

- **Memória do Autômato:** Conjunto de memórias que asseguram o funcionamento do PLC. Entre elas existem:
 1. Memória de sistema: onde se encontra o sistema operativo do autômato;
 2. Memória de programa: que contém o registo de toda a programação feita pelo utilizador e o que é pretendido ao autômato realizar;
 3. Memória de dados: onde são guardados os valores de dados das entradas/saídas, os resultados das operações realizadas pelo CPU e os dados necessários à execução do programa (Francisco, 2015).
- **Entradas/Saídas (E/S):** “As E/S asseguram a integração direta do autômato no seu ambiente de trabalho” (Francisco, 2015). As entradas trazem as informações do mundo exterior até ao processador e as saídas que levam a informação processada pelo PLC até ao mundo exterior;
- **Fonte de Alimentação:** Responsável por manter o autômato ligado, com as tensões necessárias para o funcionamento do autômato, proporcionadas a partir da tensão da rede elétrica. Existem dois tipos de autômatos: os autômatos alimentados a 24 VDC com fonte de alimentação externa, e a 230 VAC com fonte de alimentação interna (Francisco, 2015);
- **Periféricos:** Estes são dispositivos que se conectam ao autômato através das suas portas de comunicação. O periférico mais comum de se utilizar, atualmente, é o computador, dado que é o meio mais comum para a programação do autômato;

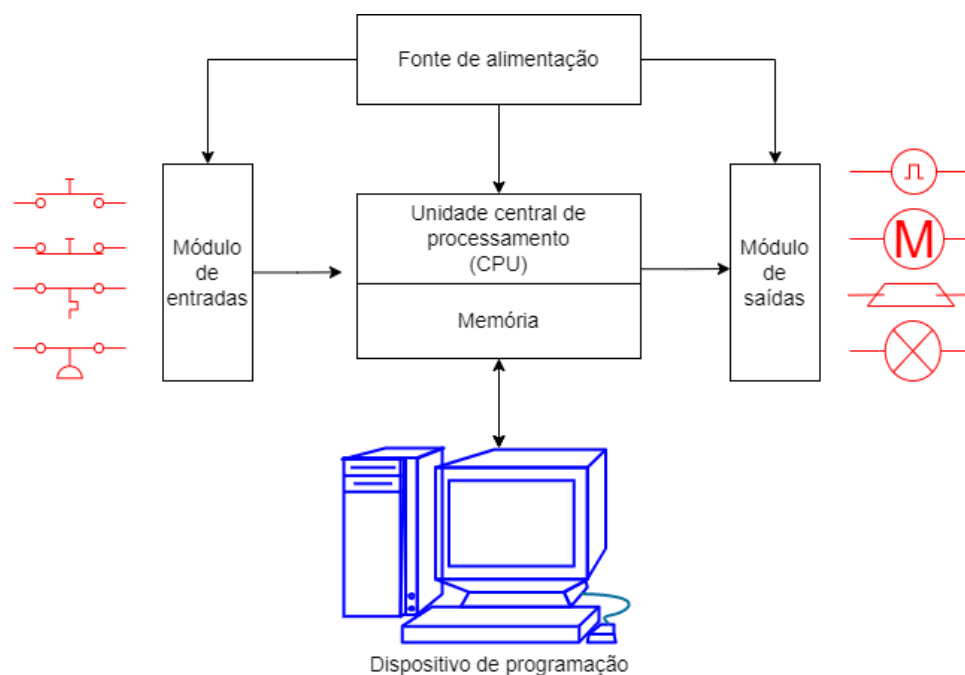


Figura 23 – Arquitetura de um PLC Fonte: Própria

2.4.4. Programação do autómato

Como já referido, os PLC foram desenvolvidos para oferecer flexibilidade à indústria, sendo que estes evoluíram com o tempo de uma forma em que muitos PLC modernos, e de alto desempenho, são na verdade computadores a correr os mais recentes sistemas operativos juntamente com *software* de resposta em tempo-real. Citam-se exemplos como: os computadores, da *Beckhoff - TwinCat 3*, que operam sobre os sistemas operativos, *Windows 10*, *Windows 7* e *Windows CE* da *Microsoft* (Beckhoff Automation, 2021).

No entanto, no início do surgimento dos PLC, era notória uma vultuosa barreira, sendo esta a falta de padronização dos sistemas. Com a diversidade de linguagens de programação no mercado e os múltiplos fabricantes a desenvolver PLC sem que houvesse uma padronização na linguagem de programação a ser usada, passou a existir uma grande curva de aprendizagem dentro das tecnologias de cada fabricante, uma vez que os conhecimentos desenvolvidos para o PLC de um fabricante, podiam não ser de forma alguma compatíveis com o PLC de outro fabricante. Outro problema que surgiu foi o facto de que quem fazia a automação industrial na época, ou seja, quem executava os circuitos de automação em lógica cablada (baseada em relés), eram os eletricitistas, que não tinham conhecimento em programação. Para pôr fim a isto, o *International Electrotechnical Commission* (IEC) desenvolveu uma coleção de normas com a intenção de criar uma experiência comum a todos os programadores de automação industrial (Gruver, 1984).

A padronização das linguagens de programação dos PLC, atualmente conhecida como IEC 61131-3, foi gradualmente criada com a união de várias padronizações de diferentes fabricantes ao longo de muitos anos. A mesma permite que todas as arquiteturas e modos de funcionamento de todos os PLC sejam semelhantes. No entanto, nem todos os fabricantes produtores de PLC implementam todas as padronizações definidas pela norma.

2.4.4.1. Elementos comuns entre as linguagens de programação definidas pelo IEC 61131-3

A norma IEC, na terceira parte, tem por objetivo principal fornecer metodologias de construção de lógicas de programação, de forma estruturada e modular, permitindo a fragmentação dos programas em partes coordenáveis.

Tem também como objetivo definir cinco linguagens de programação, cada uma com as suas características e especificidades, com a intenção de envolver a maioria das necessidades de controlo.

De entre as linguagens utilizadas, refira-se:

- *Structured Text* (ST) - Texto Estruturado;
- *Instruction List* (IL) - Lista de Instruções;
- *Ladder* (LD) - Linguagem *ladder*;
- *Function Block Diagram* (FBD) - Diagrama de Bloco;
- *Sequential Flow Chart* (SFC) - Diagrama de Fluxo.

A norma IEC permite o uso de outras linguagens de programação, desde que obedecidas as mesmas formas de chamadas e trocas de dados (*Visual Basic*, *Flow Chart*, *C++*, entre outras).

2.4.4.2. Ladder Logic (LD)

A *Ladder Logic*, é uma linguagem de blocos, sendo que esta é de fácil sintaxe, intuitiva e o seu funcionamento é baseado no funcionamento dos relés. Desta forma, facilmente os eletricitistas, já experientes na área, com uma baixa curva de aprendizagem, conseguiam evoluir da lógica cablada para a programação de PLC com mais facilidade. De facto, esta linguagem de programação é tão simples que, ainda hoje, é a mais usada dentro da automação industrial para programar PLC. Para se programar nesta linguagem é necessário colocar blocos seguidos, conectados entre si por uma linha, linha esta que é representativa de um cabo que transmite energia. Desta forma, alguns blocos comportam-se como interruptores do circuito, consoante a sua variável, impedindo que a energia chegue aos blocos mais à frente, enquanto outros blocos escrevem nas variáveis (Gruver, 1984).

Na Figura 24, abaixo, apresenta-se um exemplo de instruções em *ladder*.

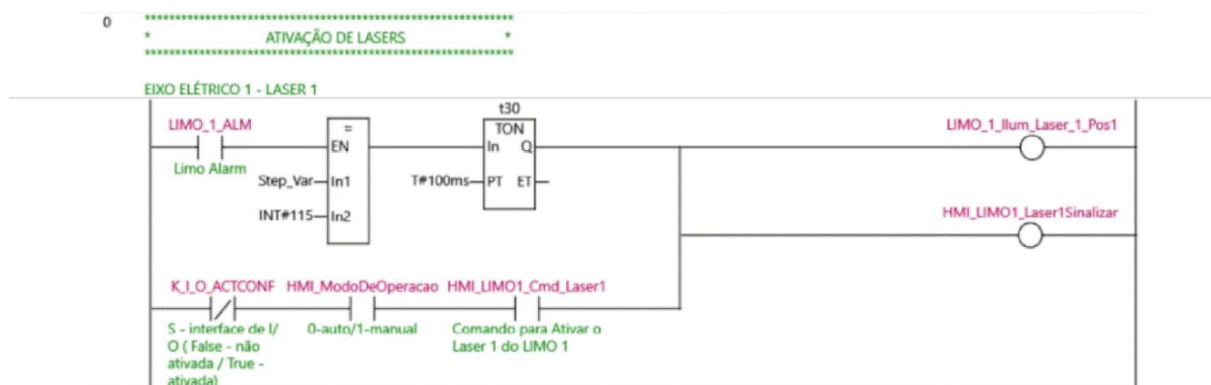


Figura 24 – Exemplo de programação em *ladder* Fonte: Própria

2.4.4.3. Structured Text (ST)

O *Structured Text* é uma linguagem de programação, em texto, baseada em linguagens de alto nível, como BASIC ou C. A programação ST faz parte da norma IEC 61131. Na Figura 25, é observável um exemplo de programa em ST. O programa denomina-se *Main* e é definido mediante as instruções [PROGRAM] e [END_PROGRAM].

```
IF #ControlValve1_Closed = false AND #ControlValve1_Open = True THEN
    #Pump_Start:= TRUE;
ELSIF #ControlValve1_Closed = True OR #ControlValve1_Open = False THEN
    #Pump_Start:= False;
END_IF;
```

Figura 25 – Exemplo de programação em *Structured Text* Fonte: (Duranso, 2021)

A linguagem é composta por declarações escritas, separadas por ponto e vírgula. As instruções usam indicações predefinidas para alterar variáveis. As variáveis podem-se apresentar através de valores definidos explicitamente, variáveis armazenadas internamente, ou entradas e saídas. Os espaços podem ser usados para separar declarações e variáveis, embora não sejam necessários. O texto estruturado não diferencia maiúsculas de minúsculas, mas pode ser útil reduzir as variáveis maiúsculas e maiúsculas. O indentação e os comentários também devem ser usados para aumentar a legibilidade e documentar o programa (Gruver, 1984) (Duranso, 2021).

2.4.4.4. Instruction List (IL)

A programação da lista de instruções (IL) é definida como parte do padrão IEC 61131. Isto é, usa instruções muito simples semelhantes às linguagens de programação mnemónicas originais desenvolvidas para PLC. Importa realçar que esta linguagem é baseada na programação em linguagem *assembly*. É o nível mais fundamental da linguagem de programação - todos as outras linguagens de programação podem ser convertidas em programas IL. Esta linguagem é complexa de programar, sendo normalmente usada quando existe uma necessidade de que o código tenha um alto desempenho (Hitomi, 1994) (Gruver, 1984).

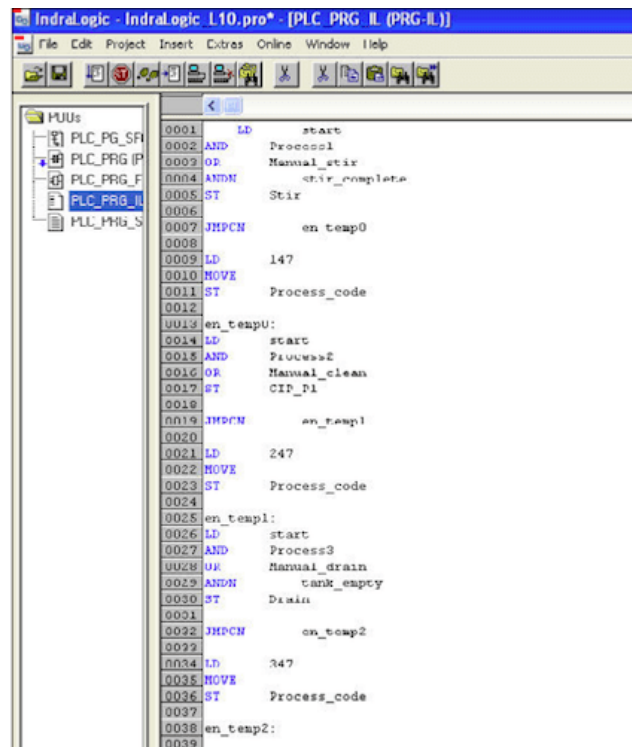


Figura 26 – Exemplo de programação em *Instruction List* Fonte: (Budimir, 2019)

2.4.4.1. Sequential Function Chart (SFC)

Todos os métodos anteriores são adequados para processos que possuem um único estado ativo a qualquer momento. Isso é adequado para máquinas e processos mais simples, contudo é geralmente implementado em máquinas complexas, que são projetadas para realizar operações simultâneas. Isso requer um controlador que seja capaz de processamento simultâneo, o que significa que estará mais do que um estado ativo, a qualquer momento, o que pode ser alcançado com vários diagramas de estado ou com técnicas, como gráficos de funções sequenciais, tal como se pode verificar na Figura 27 (Antonsen, 2022).

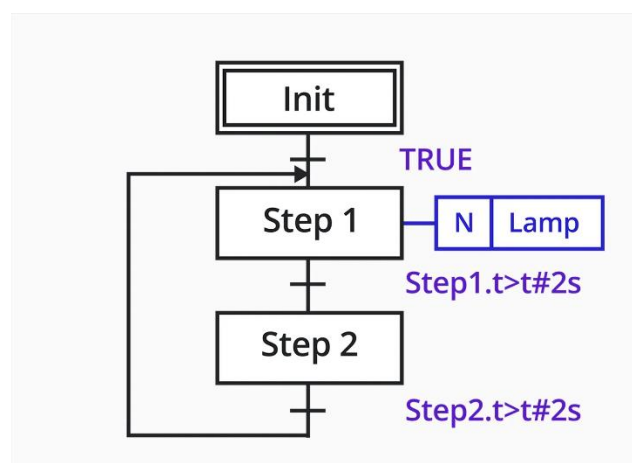


Figura 27 – Exemplo de programação em *Sequential Function Chart* Fonte: (Antonsen, 2022)

Os gráficos de SFC são uma técnica gráfica para escrever programas de controlo simultâneo. Evidencia-se que os mesmos são igualmente denominados de *Grafcet* ou IEC 848.

2.4.4.2. Function Block Diagram (FBD)

O *Function Block Diagram* faz parte da norma IEC 61131-3. O conceito principal agregado a um FBD é o fluxo de dados. Neste tipo de programas, os valores fluem das entradas para as saídas, através de blocos funcionais. Denota-se, na Figura 28, um exemplo de FBD.

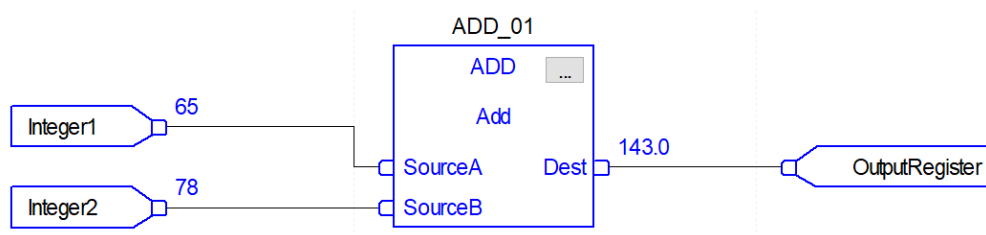


Figura 28 – Exemplo de programação em *Function Block Diagram* Fonte: (Romanov, 2019)

Um programa FBD é construído através de blocos de funções, que estão conectados entre si para definir a troca de dados. As linhas de conexão abrangem um tipo de dados, que deve ser compatível em ambas as extremidades. As entradas e saídas dos blocos de função podem ser invertidas.

2.5. Human Machine Interface (HMI)

Os *Human Machine Interface* (HMI), exemplificados na Figura 29, são usados como um painel de controlo do operador para PLC e *Remote Terminal Units* (RTU). Os HMI substituem interruptores, mostradores e outros controlos ativados manualmente, por representações gráficas do processo de controlo e controlos digitais, para influenciar esse processo (Knapp, 2011).



Figura 29 – HMI Centro de Inspeção Fonte: Própria

Os HMI permitem que os operadores iniciem e parem ciclos, determinem pontos de ajuste e executem outras funções necessárias para ajustar e interagir com um processo de controlo. Como os HMI são baseados em *software*, substituem fios e controlos físicos, por parâmetros de *software*, permitindo que sejam adaptados e ajustados com muita facilidade (Knapp, 2011).

Os HMI são aplicações de *software* modernas, executadas em sistemas operacionais atuais e, como tal, são capazes de executar diversas funções. Atuam como uma ponte entre o operador humano e a lógica complexa de um ou mais PLCs, permitindo que o operador funcione no processo, ao invés da lógica subjacente que executa a função e controla inúmeras funções em processos distribuídos e potencialmente complexos de uma localização centralizada. Para o concretizar, a interface do utilizador irá representar graficamente o processo que está a ser controlado, incluindo os valores do sensor e outras medições, bem como a representação visível dos estados de saída (que motores estão ligados, que bombas estão ativas, entre outros) (Knapp, 2011).

Os humanos interagem com o HMI por intermédio de uma consola de computador e, como tal, devem-se autenticar no sistema HMI, através de uma senha de acesso. Como os HMI fornecem dados de supervisão (representação visual do estado e valores atuais de um processo de controlo), bem como controlo (ou seja, alterações de ponto de ajuste), o acesso do utilizador pode ser bloqueado por funções específicas para utilizadores específicos, garantindo segurança, tal como se pode verificar na Figura 30 (Knapp, 2011).

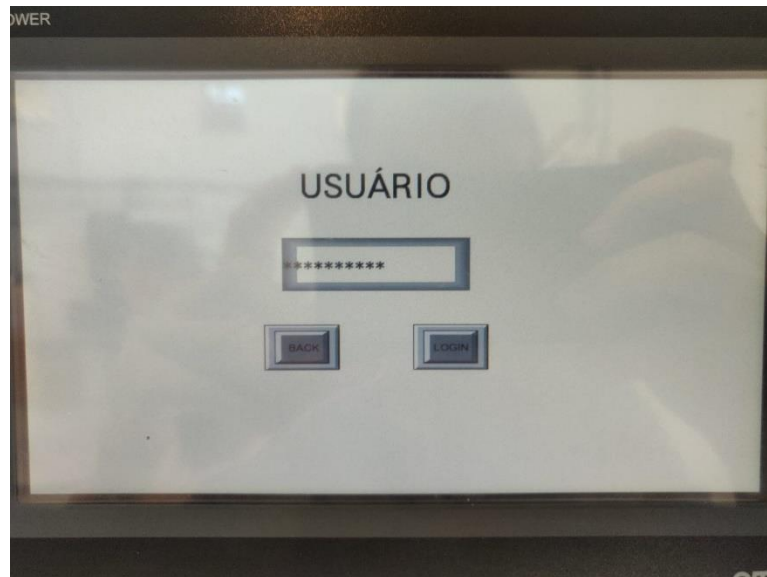


Figura 30 – HMI segurança Fonte: Própria

3. DESENVOLVIMENTO DE PROJETOS NA INDÚSTRIA DE CERÂMICA

Apesar de bastante automatizada, a indústria de cerâmica apresenta diversos desafios no que concerne à automação dos processos, nomeadamente o manuseamento de peças a cru, a vidragem, a inspeção de peças para deteção de defeitos, entre outros. No decorrer do estágio, uma das ferramentas mais utilizadas foi o simulador MotoSimEG-VRC, o qual se revelou de extrema importância para o estudo/ilustração de diversos projetos.

3.1. Guia Simulador MotoSimEG-VRC

Este capítulo descreve assim o funcionamento básico do MotoSimEG-VRC, apresentando exemplos práticos para utilizadores iniciantes deste sistema. O MotoSimEG-VRC reflete um enorme potencial de grande complexidade e, como tal, foi criado um breve guia para facilitar a compreensão do mesmo.

O simulador MotoSimEG-VRC é um *software* de programação *offline* para robôs *Yaskawa*, usado para criar simulações 3D de sistemas complexos. Esta ferramenta possui diversas aplicações e é bastante importante para o estudo de diversos projetos, sendo possível programar os diversos robôs conseguindo adequar e alterar os seus componentes sem custos, sendo esta completamente indispensável para indústrias no ramo da robótica que utilizem robôs da *Yaskawa*.

O *software* de simulação usa o mesmo modelo cinemático do controlador do robô e da linguagem INFORM, possibilitando a criação de trabalhos do robô em modo *offline*.

O referido simulador é utilizado para:

- Execução *offline* de programas para robôs *Yaskawa*;
- Estudos de alcance e acessibilidade de robôs, periféricos e equipamentos;
- Análises de tempos de ciclo;
- Programação *offline* de trajetórias;
- Deteção de colisões;
- Configuração do sistema;
- Configuração da Unidade de Segurança Funcional;
- Acesso remoto ao controlador do robô real.

Apresenta diversas vantagens, designadamente:

- Elevada taxa de precisão;
- Redução de tempos de programação;
- Redução de tempos de instalação do robô/célula;
- Redução do tempo de inatividade e consequente aumento de atividade da linha de produção;
- Capacidade de programar novas peças antes do início da produção;
- Capacidade de aumentar a eficiência modificando qualquer programa atual do robô e simplificar a programação com um gráfico detalhado que fornece a trajetória do robô;
- Suporta simulação de vários robôs e múltiplos controladores;
- Ferramenta de apoio no desenvolvimento de projetos;
- Ensaio de novas aplicações ou novas ferramentas.

Verificando todas as funcionalidades e vantagem deste simulador, e tendo em conta que é o primeiro ano do CTCV na área da robótica foi proposto o desenvolvimento de um pequeno guia do simulador para que este possa ser utilizado na empresa no futuro.

3.1.1. Criar projeto e ambiente envolvente

Inicialmente, procede-se à instalação e a ativação da licença do Simulador MotoSim EG-VRC sendo necessário, para tal, a criação de um projeto. Abaixo apresenta-se uma sequência dos procedimentos, evoluindo sistemática e detalhadamente, desde a criação da célula até a definição do robô.

1. Selecionar em [INICIAR] no menu da barra de tarefas e clicar em [PROGRAM] - [Motoman] - [Moto Sim EG-VRC] - [MotoSim EG-VRC], para executar o MotoSim EG-VRC.
2. Quando a janela principal aparecer, pressionar no botão MotoSim EG-VRC (imagem de um braço robótico realçada na Figura 31). Posteriormente, selecionar o menu [New] e no submenu escolher [New].

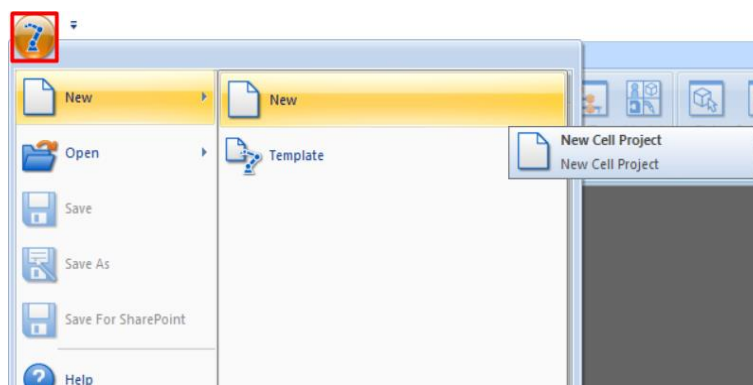


Figura 31 – Janela principal do MotoSim EG-VRC Fonte: Simulador MotoSimEG-VRC

3. Após a caixa de diálogo “*New Cell*” aparecer, digitar o nome de célula desejada e a respetiva diretoria. De seguida, pressionar [OK].
4. A célula gerada contém apenas o modelo de piso. Para adicionar um robô, adicionar previamente o controlador na célula, selecionando o menu [Controller]. No subgrupo de instruções [Setup], clicar no botão [New], como se verifica na Figura 32.

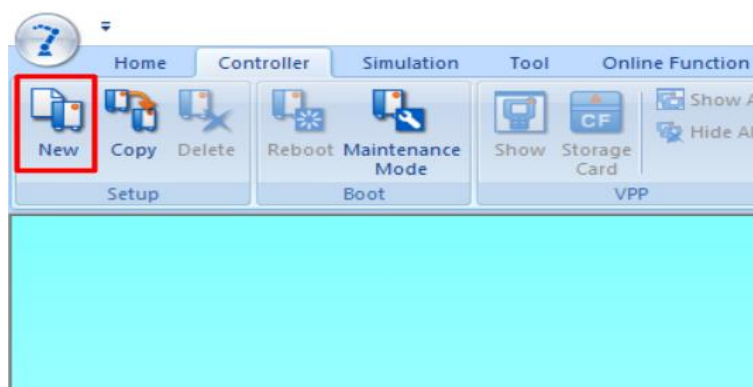


Figura 32 – Menu *Controller* do MotoSim EG-VRC Fonte: Simulador MotoSimEG-VRC

5. Na janela “*Create Controller*” serão exibidas 3 opções:
 - Opção 1: *New VRC Controller (no file)* – Controladores disponibilizados pelo software.
 - Opção 2: *VRC Controller (using file)* – Escolher um ficheiro de um controlador.
 - Opção 3: *VRC Controller (network)* – Caso o PC se encontre na mesma rede que o controlador, é possível carregar as suas definições.

Selecionar a opção desejada e pressionar o botão [OK].

6. Após visualização da janela “*New Controller*”, procede-se à seleção do tipo de controlador pretendido. Para tal, terá que se digitar o nome desejado e selecionar a

versão do sistema. Na Figura 33 está ilustrada a referida janela para o Controlador *YRC1000micro*, como o nome “Controlador CTCV” e a versão *YBS2.44-00*.

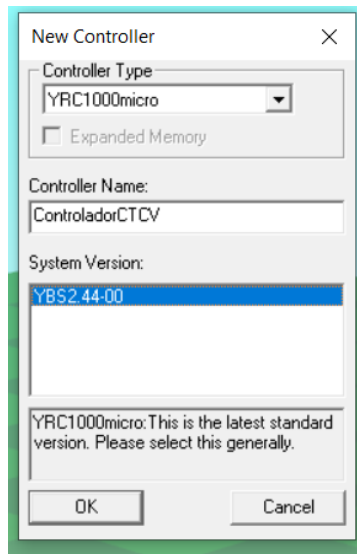


Figura 33 – Janela “New Controller” Fonte: Simulador MotoSimEG-VRC

7. Após selecionar o controlador, escolher-se-á o robô a utilizar e a aplicação do mesmo. É de ressaltar que determinados robôs necessitam de controladores específicos, como se pode verificar na Figura 34. O robô selecionado foi o robô disponibilizado pelo CTCV, cujo controlador pode ser executado no modo de manutenção ou no modo predefinido.

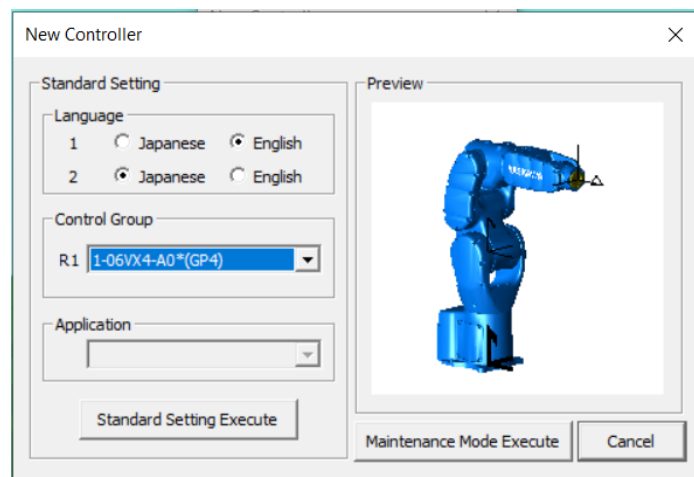


Figura 34 – Janela “New Controller” Fonte: Simulador MotoSimEG-VRC

8. Finalizadas todas as configurações anteriores, procede-se à inicialização do controlador, conforme a opção escolhida no passo anterior. O controlador virtual irá iniciar com a versão de sistema selecionada. Quando a inicialização estiver concluída, a consola do robô será exibida no modo de manutenção, simultaneamente com o guia “Instruções do Modo de Manutenção do Controlador”.

Quando todas as etapas de inicialização estiverem concluídas, pressionar o botão [Complete] do diálogo do guia de instruções, para reiniciar o controlador no modo de operação normal.

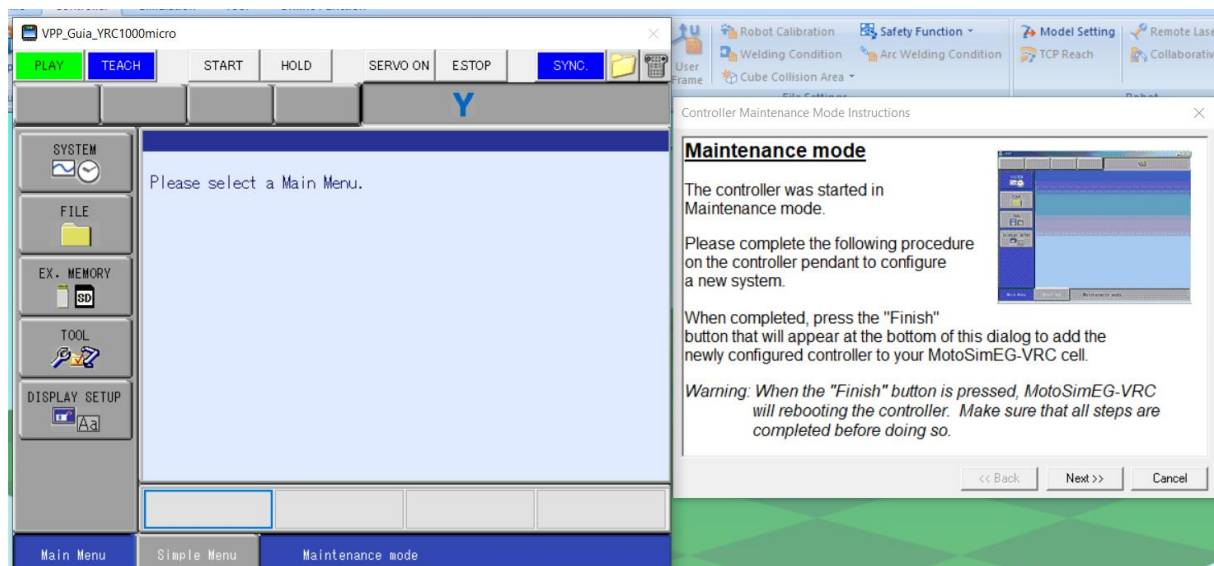


Figura 35 – Janela de modo de manutenção de controlador Fonte: Simulador MotoSimEG-VRC

9. Após o reinício, o controlador de programação virtual surge em modo normal. Posteriormente, deve-se selecionar o nome do robô pretendido (na Figura 36 abaixo o nome escolhido é *RoboGP4-CTCV*) e pressionar o botão [OK].

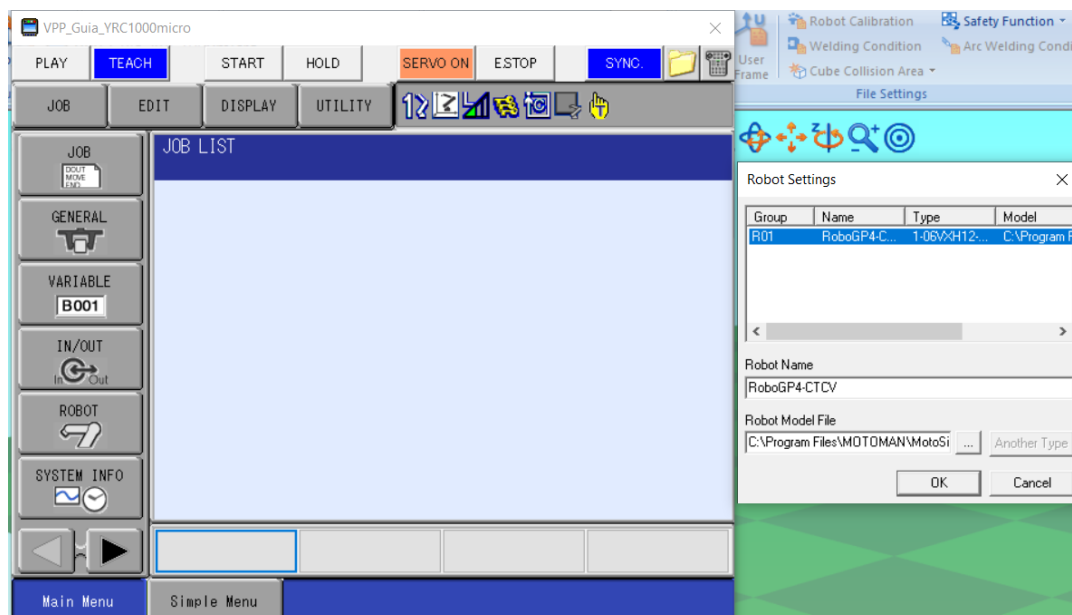


Figura 36 – Controlador Virtual do Robô Fonte: Simulador MotoSimEG-VRC

10. Após a conclusão dos passos anteriores, o modelo do robô e o controlador devem aparecer na tela da célula, tal como é possível observar na Figura 37.

Nota: O robô pode ser exibido abaixo do plano de origem (chão). Isso ocorre porque o valor de deslocamento da origem operacional do robô e o piso da coordenada central foi definido como 0 (valor inicial). Verificar “Ponto 15” para definir a posição do robô.

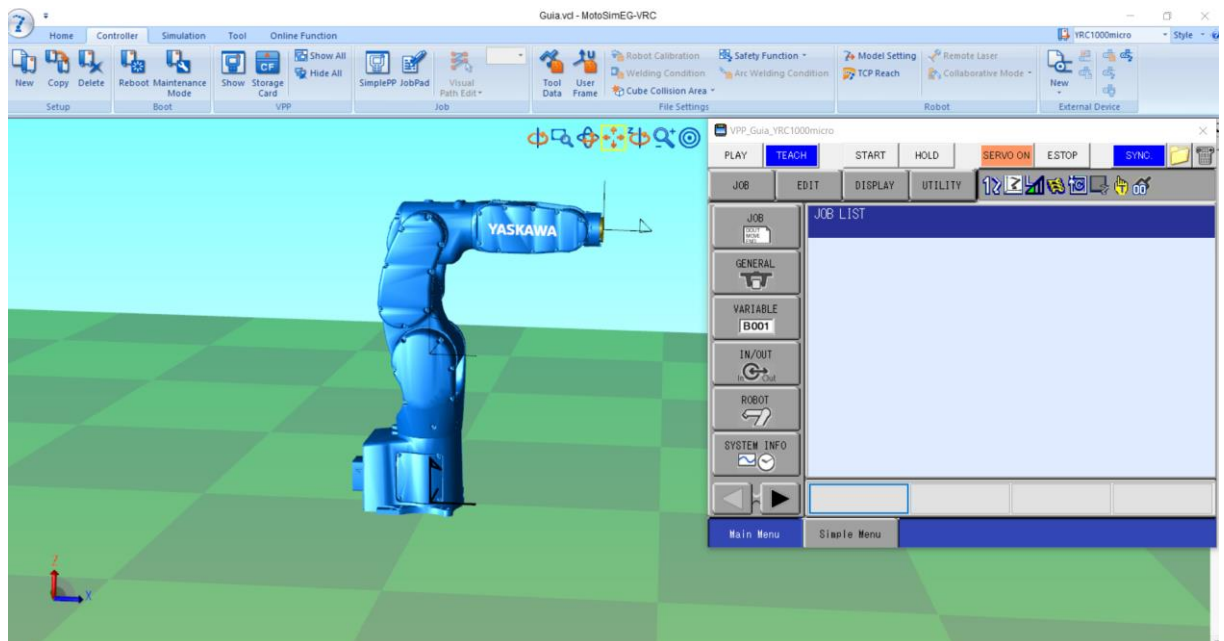


Figura 37 – Robô Gp4 e Controlador Virtual Fonte: Simulador MotoSimEG-VRC

11. Para criar peças para construir o meio envolvente, clicar no menu [Home], no grupo [Model], e selecionar [CadTree]. Esta janela contém a hierarquia dos objetos presentes no ambiente. Na Figura 38 abaixo, está descrito o Modelo *Dummy* nomeado por *world*, que representa o eixo do mundo, ou seja, todas as criações desenhadas neste projeto serão em função do mesmo. O objeto *FLOOR* é a ilustração do plano $Z=0$ (chão).

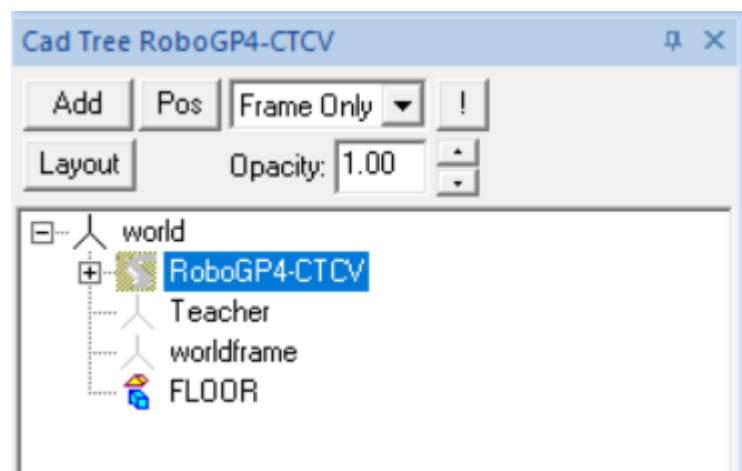


Figura 38 – Cad Tree Robô GP4 Fonte: Simulador MotoSimEG-VRC

12. Para criar o meio envolvente, procede-se à criação de peças através da ferramenta disponibilizada pelo MOTOSIM EG, ou à importação de ficheiros 3D (criados por *softwares* 3D, nos formatos compatíveis).

Para criar objetos, selecionar [*World*] (caso o objeto não esteja dependente do robô ou exista outro objeto criado). De seguida, adicionar o novo objeto selecionando [*Add*], tal como espelhado na figura abaixo. Posteriormente, introduzir e pressionar [*OK*]. Para introduzir o nome no objeto selecionado, clicar no campo [*File Name*] na diretoria do ficheiro.

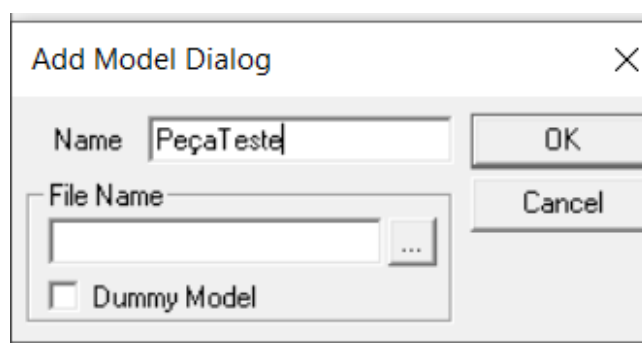


Figura 39 – Janela para adicionar novos modelos Fonte: Simulador MotoSimEG-VRC

13. Tendo adicionado o objeto selecionado, procede-se à escolha das formas/sólidos geométricos desejados. As opções fornecidas são:
- BOX* – Paralelepípedo com origem no seu centro;
 - BOX2* – Paralelepípedo com origem no seu vértice;
 - CYLINDER* – Cilindro com origem na sua superfície inferior;
 - CONE2* – Cone com origem na sua superfície inferior;
 - SPHERE* – Esfera com origem no seu centro;
 - PIPE2* – Tubo com origem na sua superfície inferior;
 - AXIS6* – Conjunto de eixos que podem ser usados como pontos de destino ou para configuração de eixos externos. Pode ser editado no *Frame*;
 - LINE* – Várias linhas que unem os diferentes pontos definidos, podendo formar um triângulo (por exemplo);
 - LINE2* – Linha definida por dois pontos que une os mesmos;
 - CUBE* – Poliedro com quatro pontos ou mais, onde o ponto de referência é o primeiro ponto definido;
 - FLOOR* – Plano perpendicular ao eixo Z do mundo, com a aparência parecida com um tabuleiro de xadrez, onde se pode definir independentemente o número de quadrados em X e em Y;

1. *FACE* – Face definida por pelo menos 3 pontos;
14. Através das ferramentas acima mencionadas, para criar o meio envolvente é necessário escolher os parâmetros do mesmo. Os parâmetros existentes, e que se encontram representados na Figura 40, são: a altura, a largura, o comprimento e a localização do ponto de origem e as respetivas rotações, cor, entre outros.

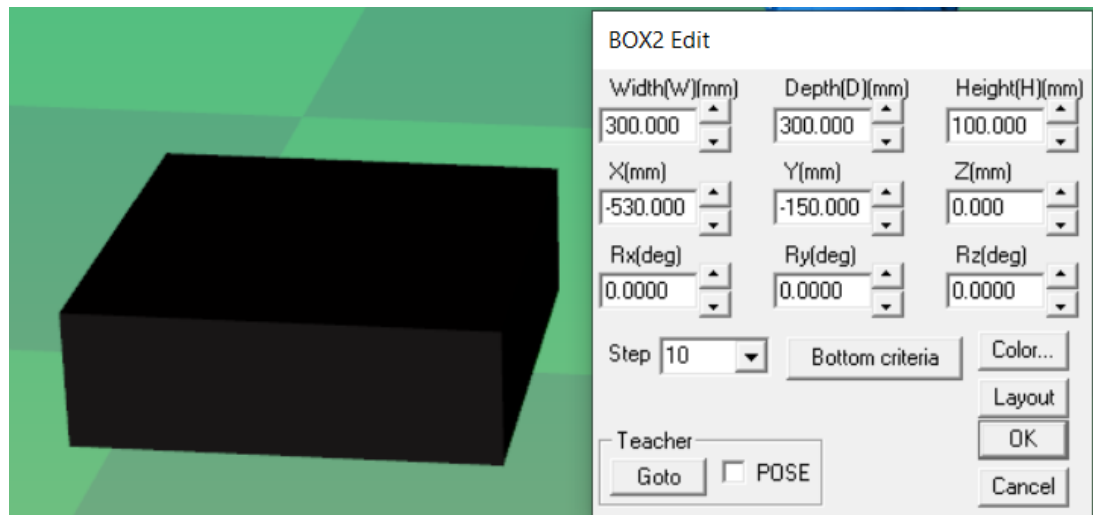


Figura 40 – Exemplo da criação de um suporte para o robô com *BOX2* Fonte: Simulador MotoSimEG-VRC

15. Após a criação dos objetos, é possível executar a alteração da sua disposição e da sua posição de origem, através do botão [*Pos*] na [*Cad Tree*]. Inserir em X, Y e Z os valores pretendidos para modificar a localização do modelo. Na Figura 41 pode-se constatar a possibilidade de alterar a posição do robô de forma a criar uma base.

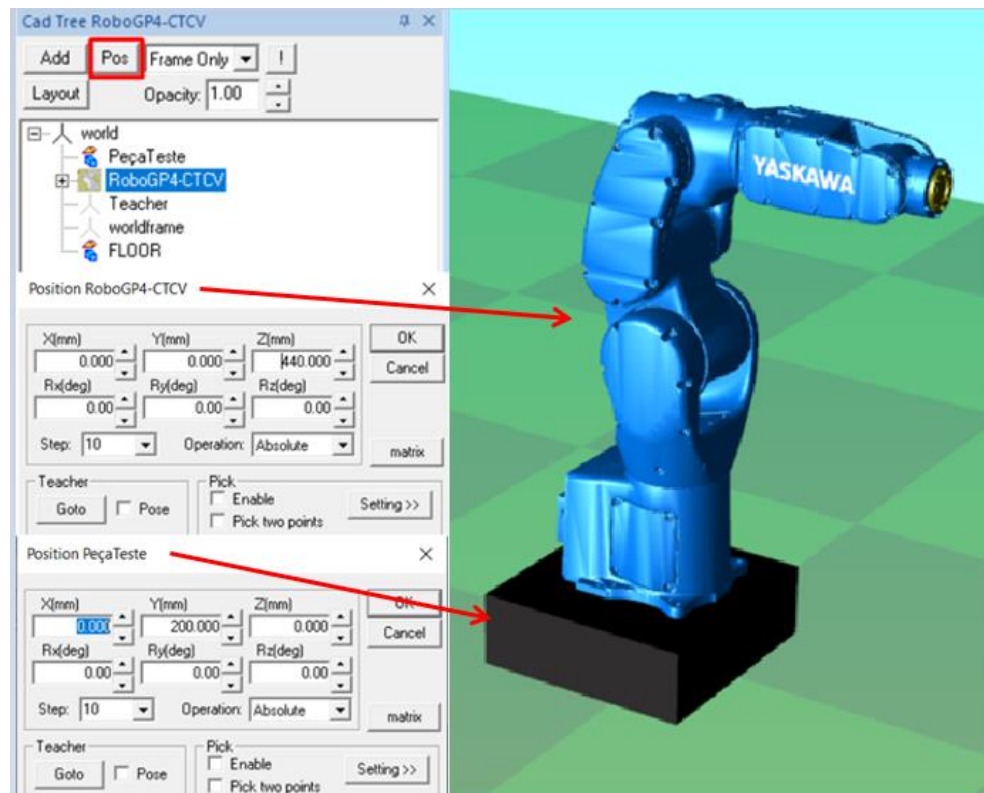


Figura 41 – Alteração de coordenadas de origem Fonte: Simulador MotoSimEG-VRC

Com as instruções anteriores é possível criar o meio envolvente para o desenvolvimento de diversos projetos. O exemplo apresentado na Figura 42, foi empregue na célula presente no CTCV.

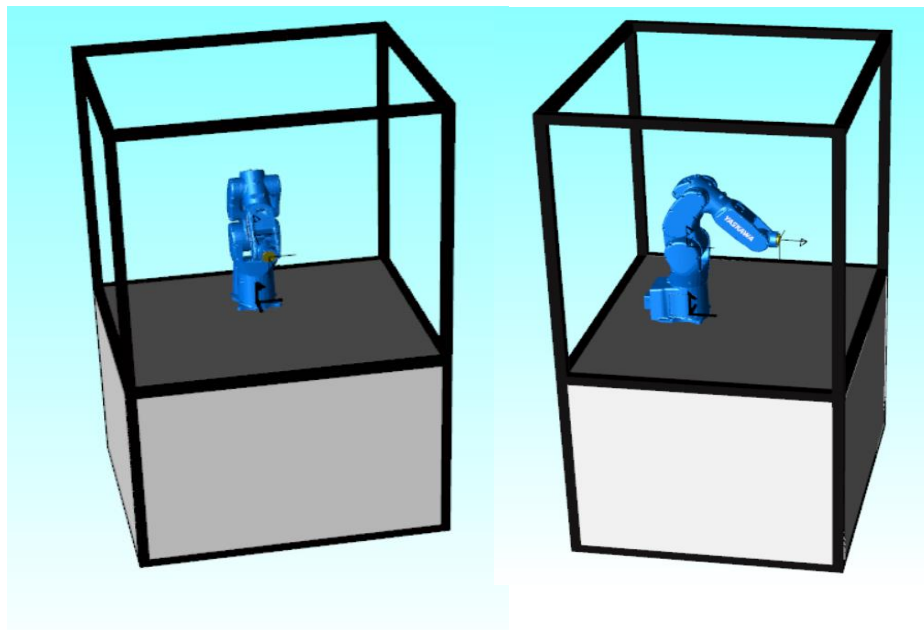


Figura 42 – Exemplo da criação da célula presente no CTCV Fonte: Simulador MotoSimEG-VRC

3.1.2. Criar Ferramenta

Para elaborar qualquer tipo de interação do robô com outros objetos ou o meio envolvente, torna-se imperativa a criação de uma ferramenta e de um parâmetro. De seguida, abordar-se-ão em diversos pontos, sequencialmente, como se cria uma garra contida no robô.

1. Recorrer ao [Cad Tree] para criar um objeto com nome desejado. No separador da hierarquia do robô, selecionar o “NomeDoRobo_flange” e criar um modelo para que o modelo com hierarquia superior, seja o modelo de ferramenta flange do robô.
2. Se o modelo não estiver definido corretamente, para alterar basta selecionar com o botão do lado direito do rato [Model Attribute - Set Parent], conforme é possível observar na Figura 43 e na Figura 44.

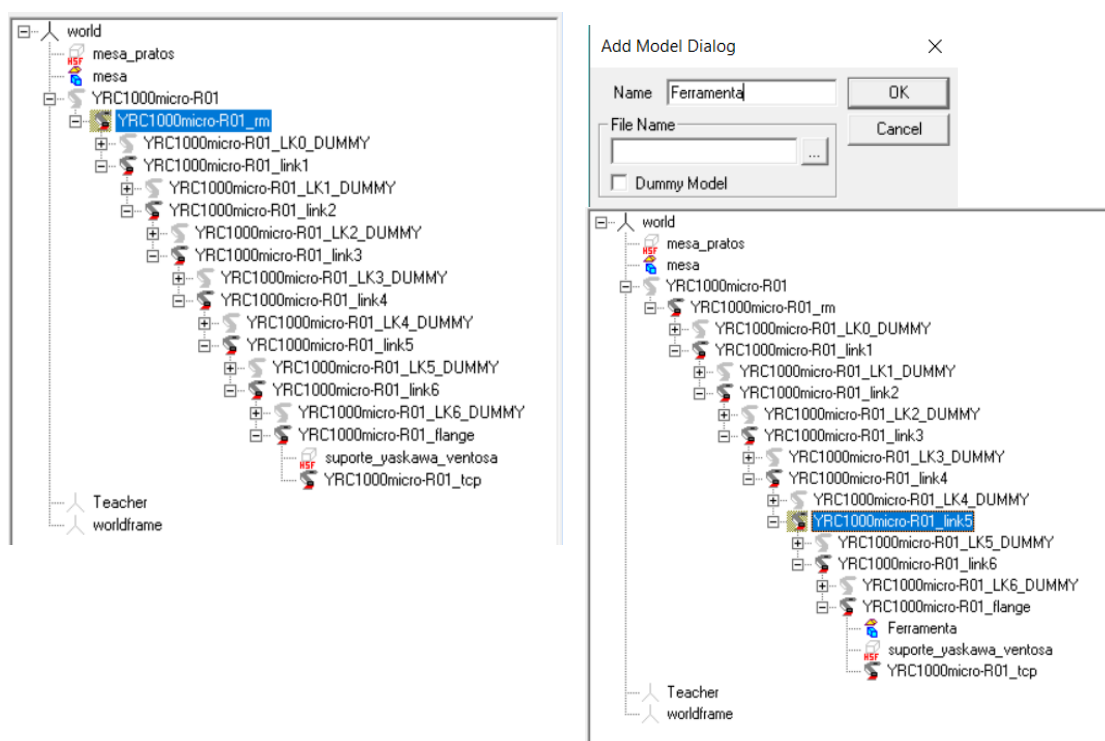


Figura 43 – Criação de um modelo integrado numa hierarquia Fonte: Simulador MotoSimEG-VRC

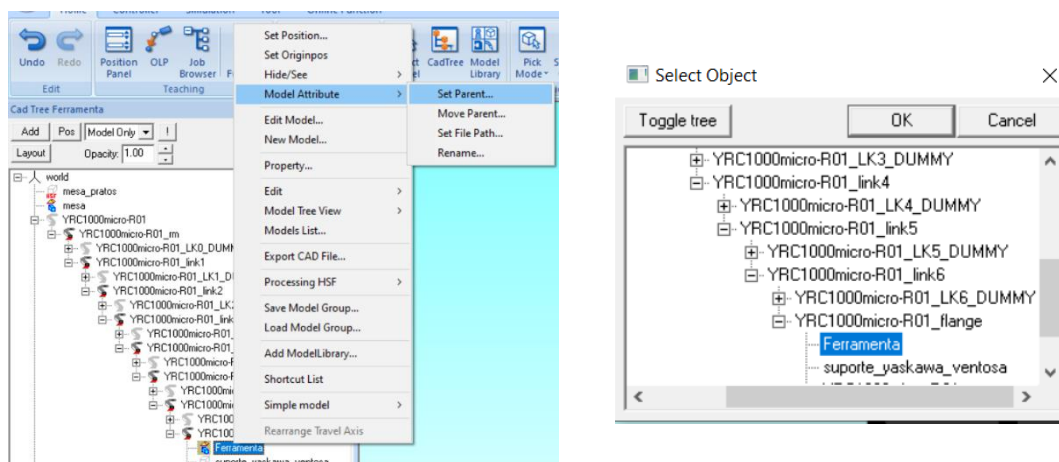


Figura 44 – Definir atributos do modelo Fonte: Simulador MotoSimEG-VRC

- Com o modelo definido na hierarquia correta, gera-se a parte gráfica da garra selecionada, através dos passos descritos anteriormente. A garra criada, no exemplo que se segue, é composta por um suporte que contém uma ventosa alimentada por ar comprimido. Para a recriar no simulador, faz-se uso das formas geométricas anteriormente introduzidas, com o objetivo de ilustrar a mesma.

Na Figura 45 é possível ver todos os atributos das formas escolhidas para gerar a ferramenta da Figura 46.

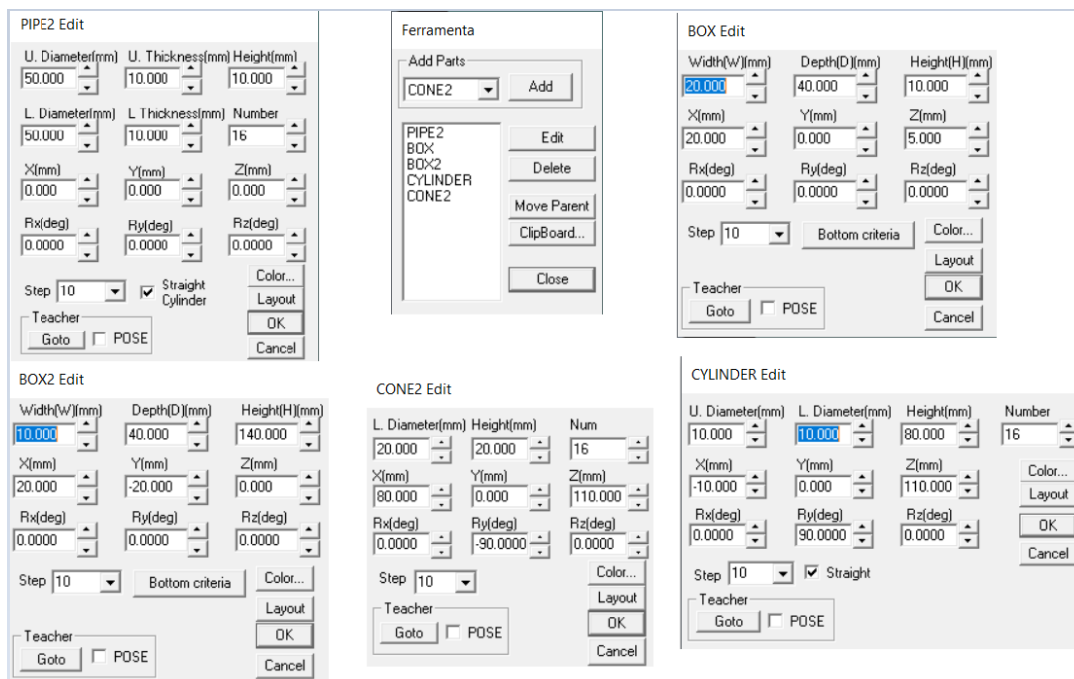


Figura 45 – Parâmetros da ferramenta criada Fonte: Simulador MotoSimEG-VRC

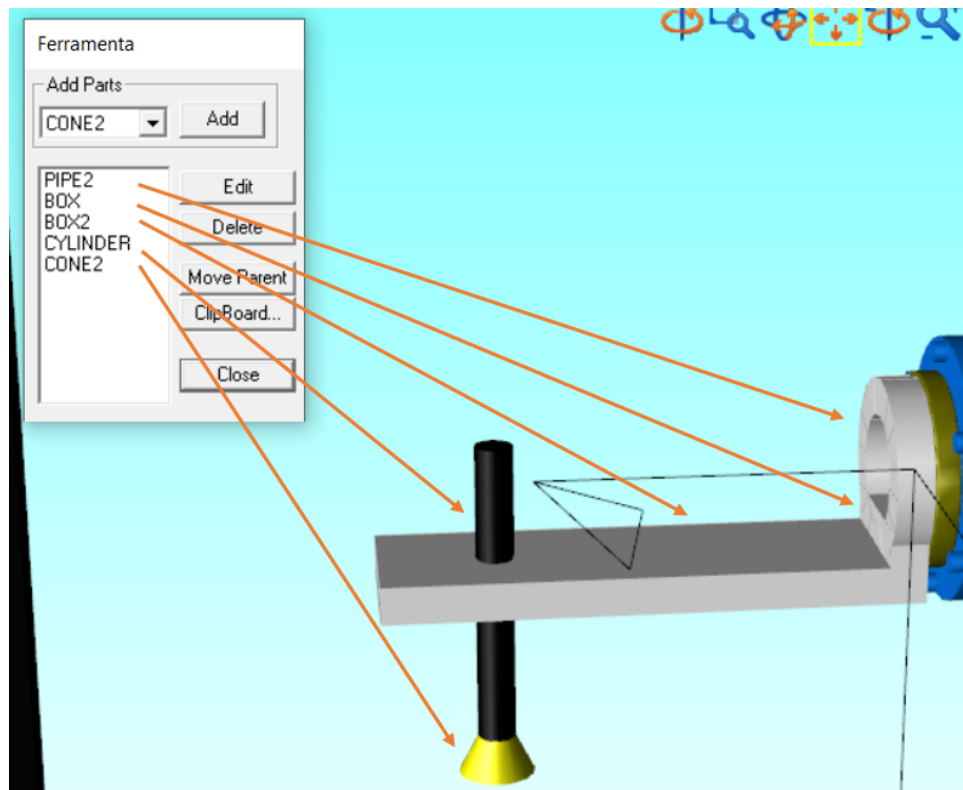


Figura 46 – Ferramenta criada através dos parâmetros inseridos Fonte: Simulador MotoSimEG-VRC

3.1.3. Editar/Definir dados da ferramenta

Nesta seção abordar-se o modo de editar/definir os dados da ferramenta. A ferramenta que irá ser criada é uma garra de manipulação de pratos, alimentada a ar comprimido.

Não obstante, e antes de definir os dados da ferramenta, é fundamental identificar a localização e as medidas correspondentes do ponto de contacto da ferramenta com os objetos.

1. No menu [Controller] no grupo [File Settings], seleccionar [Tool Data], tal como se encontra ilustrado na Figura 47.



Figura 47 – Menu [Controller] onde se situa a ferramenta [Tool Data] Fonte: Simulador MotoSimEG-VRC

2. De seguida, na janela, deve-se introduzir os valores correspondentes à rotação e translação do eixo pretendido.

A Figura 48 contém a operação requerida para o exemplo sugerido. Para auxílio, pode-se ativar no menu [Home] no grupo [Screen], a ferramenta [Frame display], tal como é apresentado na Figura 49. É importante salientar que com esta ferramenta ativa, será visível o eixo da ferramenta.

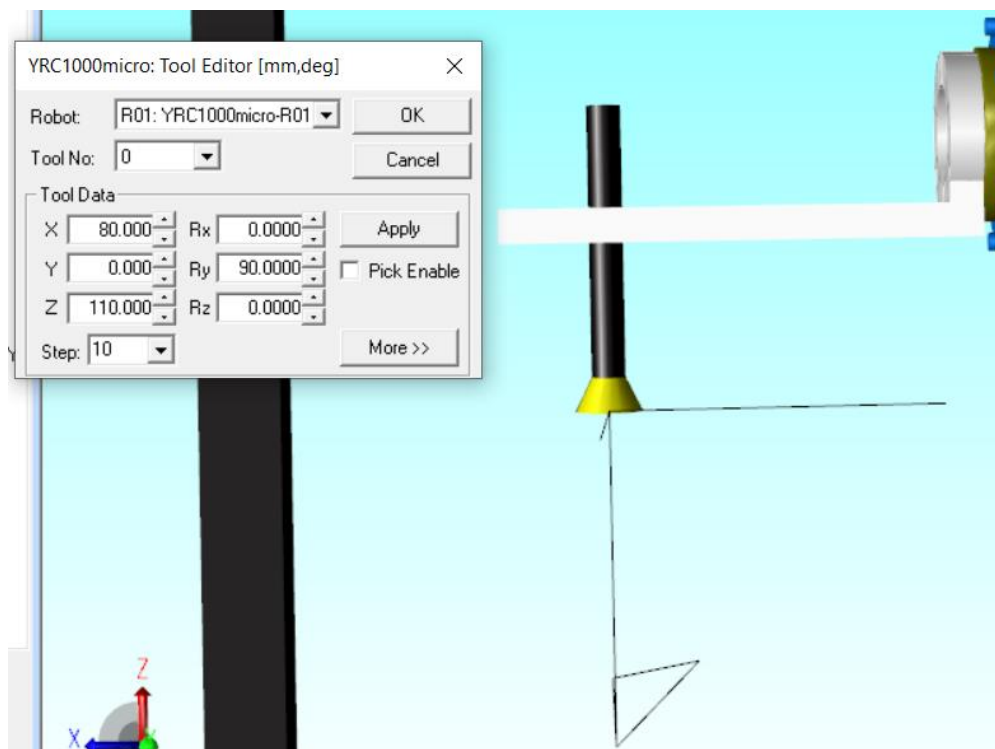


Figura 48 – Parâmetros inseridos para o *gripper* gerado Fonte: Simulador MotoSimEG-VRC

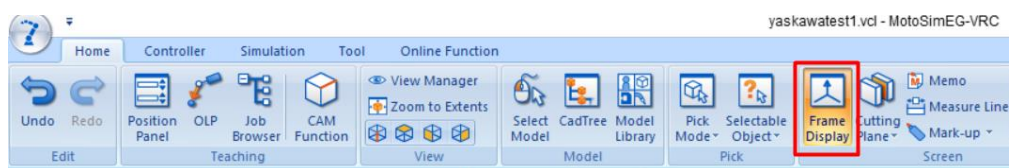


Figura 49 – Menu [Home] onde se localiza a ferramenta [Frame Display]

3.1.4. Criar um JOB

1. No menu principal do controlador virtual, seleccionar *JOB - CREATE NEW JOB*. (Figura 50)



Figura 50 – Controlador virtual Fonte: Simulador MotoSimEG-VRC

2. Na janela apresentada na Figura 51 no campo “*JOB NAME*”, deve-se introduzir o nome desejado para o trabalho, para este exemplo, utilizou-se o nome "GUIA".

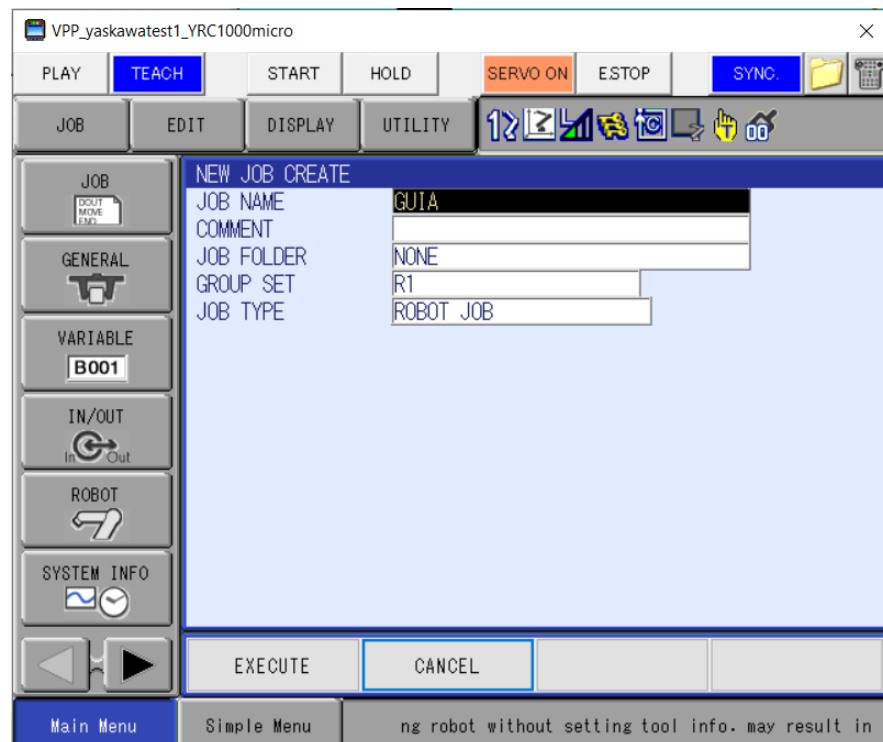


Figura 51 – Janela no controlador para um novo programa Fonte: Simulador MotoSimEG-VRC

3. Para concluir a criação do *JOB*, basta selecionar [*EXECUTE*] ou pressionar a tecla [*ENTER*]

3.1.5. Programar o robô

A Consola Virtual é empregue na programação do robô, sendo a mesma equivalente à programação de um robô físico. No anexo I, segue-se um guia Yaskawa, com algumas utilidades de programação através da consola.

As ferramentas específicas de programação encontram-se no simulador. Seguem abaixo os passos descritos, detalhadamente, para o implementar:

1. Pressionar o botão [*Servo On*]. Assim que a alimentação do servo for ativada, mover o robô para a posição desejada, usando as teclas correspondentes ao eixo, no controlador virtual. Para exibir ou ocultar o teclado do controlador, pressionar a tecla [*~*] ou clicar no ícone com a localização ilustrada na Figura 52.

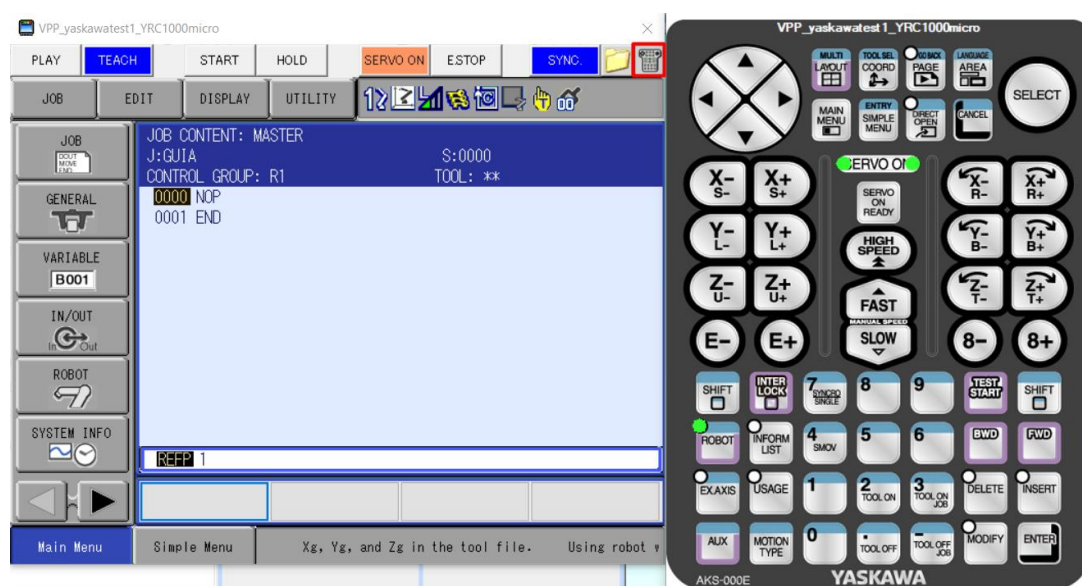


Figura 52 – Localização do ícone do teclado do controlador virtual Fonte: Simulador MotoSimEG-VRC

2. Para selecionar o tipo de movimento, faz-se uso da tecla [*Motion Type*] do controlador. Exemplo: Para selecionar o movimento de juntas, usar “*MOVJ*”.
3. Posteriormente, para definir a velocidade do movimento, seleciona-se a linha de código. De seguida, move-se o cursor para o valor de velocidade, elegendo o valor da velocidade através de [*SHIFT*], com recurso às setas “para cima” e “para baixo”. No exemplo abaixo, definiu-se o tipo de movimento “*MOVJ*” e uma velocidade de 100%. Ilustrado o resultado na Figura 53.



Figura 53 – Exemplo de movimento de juntas com velocidade 100% Fonte: Simulador MotoSimEG-VRC

4. Para registrar a posição, pressiona-se a tecla [ENTER].
5. No menu [Home], no grupo [Teaching], carregar no botão [OLP] para exibir a janela da ferramenta “OLP”. A localização da ferramenta OPL está presente na Figura 54.



Figura 54 – Localização da ferramenta “OLP” Fonte: Simulador MotoSimEG-VRC

6. *A posteriori*, na janela “OLP”, marca-se a caixa de seleção [Pick enable]. Após essa seleção, determina-se o objeto para o robô agarrar, e seleciona-se um ponto arbitrário do mesmo. Na ferramenta “OLP” prime-se sobre o objeto, para o robô se colocar na posição de recolha do mesmo, evitando a manipulação eixo a eixo.

3.1.6. Entradas e saídas

Para adicionar uma instrução de entradas/saídas no controlador virtual, pressiona-se o botão *[INFORM LIST]*. Quando ativo, é acionado um *LED* nesse mesmo botão. De seguida, no menu aberto pela ação anterior, seleciona-se a instrução desejada:

No menu:

1. DOUT:
 - OT#(7) ON/OFF - Acionar a saída correspondente (saída 7)
 - OG#(1) 5 - (Grupo de 8 saídas)
 - OGH#(3) B000 - (Grupo de 4 saídas)
2. DIN:
 - B000 IN#(9), IG#(8), IGH#(12), SIN#(2) - Guarda o estado das entradas para a variável B000
 - B00 OT#(4), OG#(7), OGH#(5), SOUT#(23) - Guarda o estado das saídas para a variável B000
3. WAIT:
 - IN#(9), IG#(8), IGH#(12), SIN#(2) = B001 T = 0,01 - Aguardar a entrada ou o tempo
 - OT#(4), OG#(7), OGH#(5), SOUT#(23) = B002 - Aguardar saídas
 - BOOO = 5 - Aguardar o valor de uma variável
4. PULSE:
 - OT#(15) T= 0,01 - Ativar uma saída durante 0,01s
5. AOUT:
 - AO#(3) 5,00 Envio de uma tensão entre $\pm 5,0$ v.

O *MotoSim* apresenta ainda várias ferramentas como “*I/O Monitor*”, para monitorização de entradas e saídas. Na Figura 55 encontra-se a localização da ferramenta “*I/O Monitor*”.

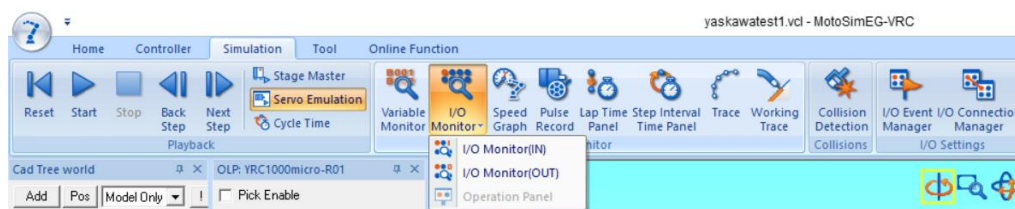


Figura 55 – Localização da ferramenta “*I/O Monitor*” Fonte: Simulador MotoSimEG-VRC

Não obstante, apresenta também a ferramenta “*I/O Connection Manager*”, Figura 56, sendo esta a responsável pela ligação entre as entradas e saídas de diversos robôs/dispositivos externos.

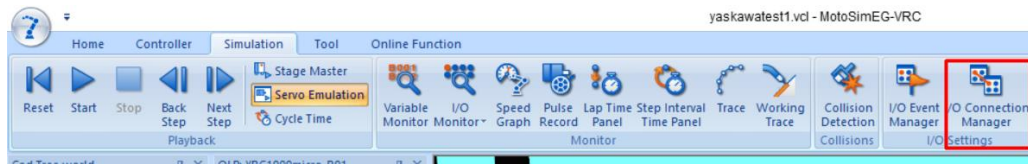


Figura 56 – Localização da ferramenta “*I/O Connection Manager*” Fonte: Simulador MotoSimEG-VRC

3.1.7. Alterar a Programação do Robô

As linhas de programação criadas são visíveis no controlador virtual. Para tal, desloca-se o cursor do “*JOB*” para a primeira etapa, pressionando a tecla [*FWD*] (no teclado do controlador virtual), movendo assim o robô para uma certa posição. No momento em que o robô atingir a posição definida é interrompido o movimento e o cursor do “*JOB*” deixará de ‘piscar’. Prontamente, solta-se a tecla [*FWD*] e pressiona-se novamente a mesma, para seguir para a próxima etapa do trabalho.

Um “*JOB*” pode ser editado com o seguinte procedimento:

- Editar Instruções anteriormente criadas - Mover o cursor para o passo a ser modificado (com o robô na posição definida), e pressionar a tecla [*MODIFY*] do teclado do controlador, seguindo da tecla [*ENTER*].
- Adicionar Etapas - Mover o cursor para a etapa anterior ao ponto de inserção. Subsequentemente, mover o robô para a posição eleita, usando o controlador virtual. De seguida, definir o tipo de movimento e a velocidade do movimento, pressionar a tecla [*ADD*] do teclado do controlador, e logo após a tecla [*ENTER*].

Excluir Etapas ou Instruções - Mover o cursor para a instrução a ser excluída (caso a instrução seja de movimento, é necessário mover o robô para a posição que se deseja eliminar). Para tal, prime-se a tecla [*FWD*]. Concluídos os passos anteriores, pressiona-se a tecla [*DELETE*] do teclado pendente, e depois [*ENTER*].

3.1.8. Adicionar uma passadeira transportadora

As passadeiras transportadoras são dispositivos que podem ser configurados com 1 a 3 eixos lineares (X, Y, Z). Para adicionar um tapete à célula, dispõem-se os seguintes passos:

1. No menu [Controller], no grupo “External Device”, clicar no botão [New] e seleccionar “Conveyor”. Instruções recriadas na Figura 57.



Figura 57 – Inserir passadeira transportadora Fonte: Simulador MotoSimEG-VRC

2. Na janela aberta, Figura 58, elege-se o número do controlador e o número de juntas, com limite máximo de adição de 6 tapetes transportadores. O transportador é registado como controlador, que é determinado pela seleção do número do transportador e não pode ser alterado.

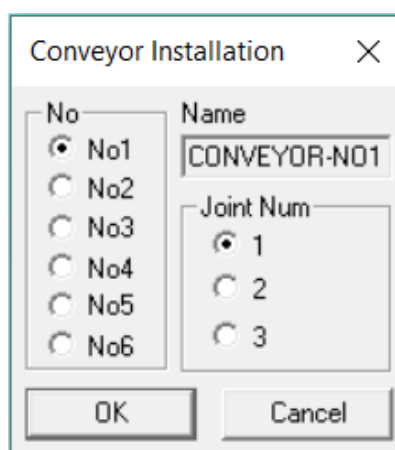


Figura 58 – Janela “Conveyor Installation” Fonte: Simulador MotoSimEG-VRC

3. Na janela “Conveyor Settings”, Figura 59, é exequível a definição das dimensões, da velocidade e da faixa de operação do transportador, através da edição das informações do transportador. Esta caixa de diálogo é exibida automaticamente quando um novo transportador é instalado. Também pode ser exibido posteriormente, para modificar as configurações do transportador, seleccionando no menu [Controller], no grupo “External Device”, no botão [Conveyor Setting].

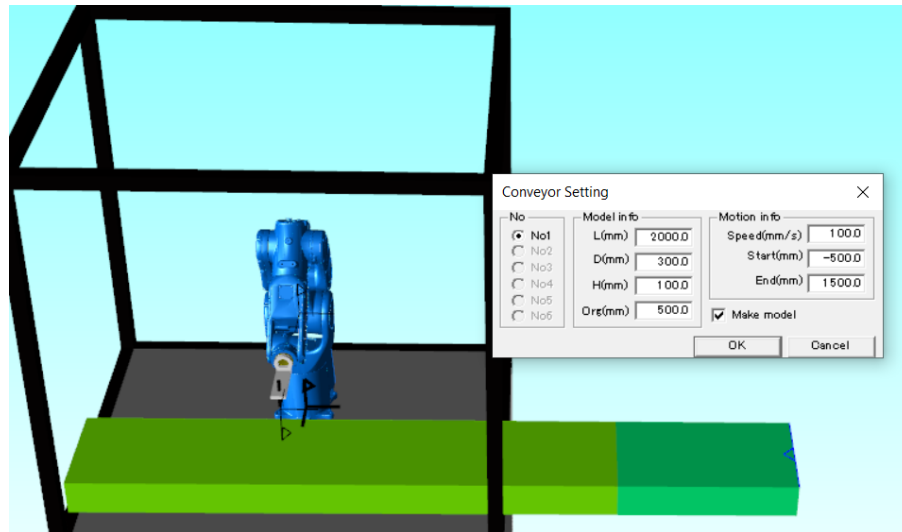


Figura 59 – Ilustração de ambiente com passadeira transportadora Fonte: Simulador MotoSimEG-VRC

Na janela da passadeira transportadora, os parâmetros que podem ser alterados são:

- L (mm) - Comprimento total do transportador;
- D (mm) - Largura do transportador;
- H (mm) - Espessura do transportador;
- Org (mm) - Posição para o início do transportador;
- $Speed$ (mm/s) - Velocidade de operação do transportador;
- $Start$ (mm) - Posição inicial da operação do transportador;
- End (mm) - Posição final da operação do transportador.

3.1.9. Adicionar uma prensa

As prensas são dispositivos que possuem 3 eixos lineares (X, Y, Z), embora as mais usuais apenas utilizem o eixo Z. Para adicionar uma prensa à célula, seguem-se as instruções detalhadamente:

1. No menu [Controller], no grupo [External Device], premir o botão [New] e seleccionar [Press], na Figura 60 encontra se a localização da função para adicionar uma prensa.



Figura 60 – Adicionar uma prensa Fonte: Simulador MotoSimEG-VRC

2. Na janela gerada, Figura 61, selecionar [*Make model*] e pressionar [*OK*].

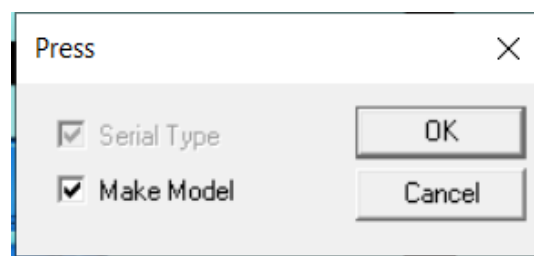


Figura 61 – Janela [*Press*] Fonte: Simulador MotoSimEG-VRC

3. Digitar o nome na caixa de diálogo “*Device Install*”, conforme mostrado na Figura 62 abaixo, e pressionar o botão [*OK*].

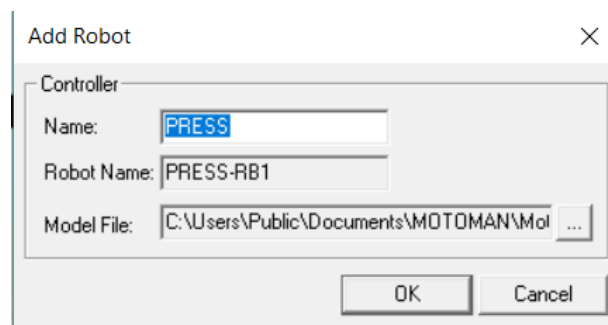


Figura 62 – Configuração da prensa Fonte: Simulador MotoSimEG-VRC

4. Terminados os passos anteriores, a prensa será adicionada, tal como é possível observar na Figura 63 seguinte:

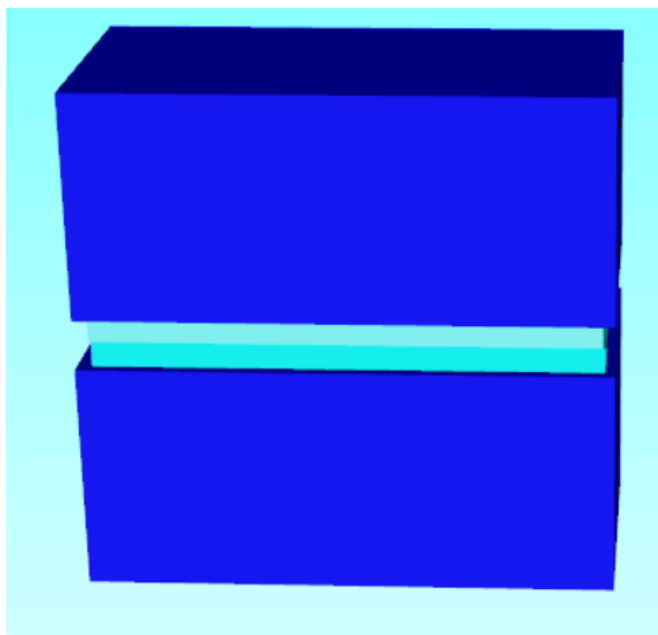


Figura 63 – Representação da prensa no simulador Fonte: Simulador MotoSimEG-VRC

3.1.10. Movimentar dispositivos externos (Passadeira/ Prensa)

Os dispositivos podem ser movidos usando o Pannel de Posição, da Figura 64. No menu [*Home*], no grupo [*Teaching*], clica-se no botão [*Position Panel*]. A janela exibida é responsável pela movimentação dos dispositivos externos.



Figura 64 – Localização da ferramenta [*Position Panel*] Fonte: Simulador MotoSimEG-VRC

Ao contrário de outros robôs, os dispositivos externos só podem ser movidos usando o sistema de coordenadas de “*Pulse*” ou “*Joint*”. A utilização de “*Joint*” é o equivalente a um sistema de coordenadas retangulares (coordenadas X, Y, Z em milímetros). Na Figura 65 está presente a janela da ferramenta para movimentação de dispositivos externos.

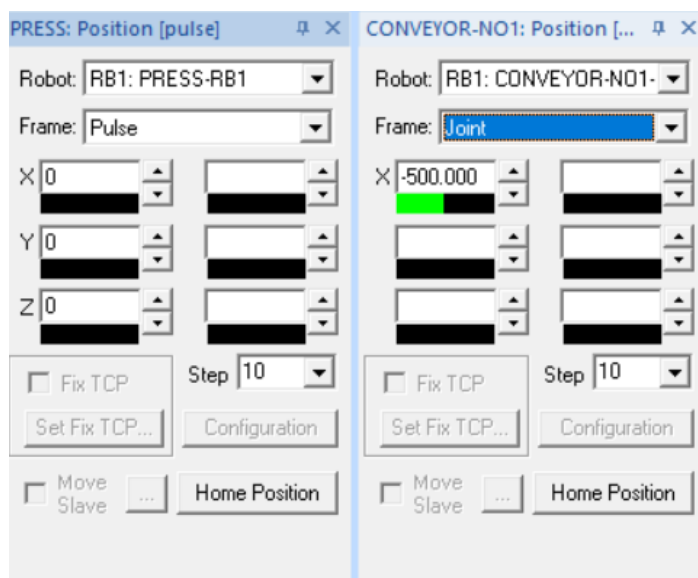


Figura 65 – Ferramenta de movimentação de dispositivos externos Fonte: Simulador MotoSimEG-VRC

3.1.11. Programar dispositivos externos

Os dispositivos podem ser programados para se moverem de forma similar a um controlador de robô, criando trabalhos e depois reproduzindo-os.

O painel “Job”, Figura 66, é usado para exibir o trabalho e permitir a programação do mesmo. No menu [Controller], no grupo [External Device], clica-se no botão [Job Panel]. A caixa de diálogo gerada é onde serão inseridas as instruções para os respectivos dispositivos externos.

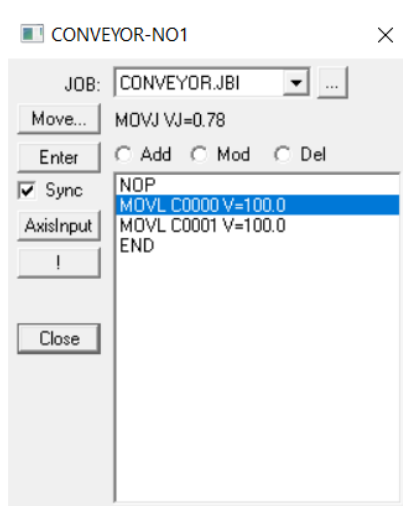


Figura 66 – Painel de programação de dispositivos externos Fonte: Simulador MotoSimEG-VRC

A janela representada acima, responsável pelo funcionamento do dispositivo externo, em que com a passadeira transportada é possível executar o seguinte:

- [JOB] - Selecionar o programa na lista ou através do botão [...]
- [Sync] – Caso a *checkbox* esteja selecionada, o dispositivo desloca-se sincronizado, de acordo com a etapa selecionada.
- [Add], [Mod], [Del] – Selecionar caso o objetivo seja adicionar, modificar ou apagar uma linha de código, respetivamente.
- [Enter] - Executa o comando selecionado entre [Add], [Mod], [Del].
- [Move] - Exibe janela de interpolação onde é necessário especificar o tipo de movimento, velocidade e nível de posição para a operação de reprodução. Clica-se em [OK] e será gerada uma instrução de movimento.

3.1.12. Scripts

Os *scripts* permitem manipular modelos através de comandos. Para exibir o editor de *scripts*, Figura 67, seleciona-se o menu [Simulation], no grupo [Model Simulation], o ícone [Model Script Manager].

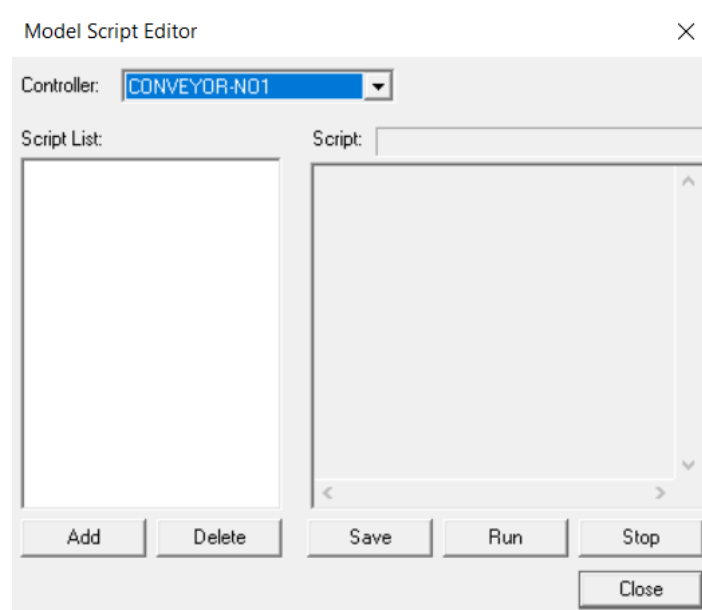


Figura 67 – Janela “Model Script Editor” Fonte: Simulador MotoSimEG-VRC

Os comandos que podem ser usados no *script* para manipular modelos são:

- SEE M1 - Exibe o modelo M1.
- HID M1 - Oculta o modelo M1.

- MOV M1 M2 - Move o modelo M1 na *Cad Tree* para M2, sem alterar a sua localização do modelo relativamente ao mundo.
- AXIS6 M1=10,20,30,0,0,0 - Move o modelo M1 para (10,20,30,0,0,0).
- DDX6 M1=10,20,30,0,0,0 - Move o modelo M1 adicionando (10,20,30,0,0,0) a sua posição atual.
- DUP M1 M2 - Copia o modelo M1 para criar M2.
- REF M1 M2 – Copia a referência do modelo M1 para M2.
- DEL M1 - Exclui o modelo M1.
- ACT M1 D S=P1 E=P2 T=T1, T2 – O modelo M1 move se de P1 para P2 na direção D entre os T1 e T2.
 - D: Direção do movimento (X, Y, Z, Rx, Ry ou Rz)
 - P1: Posição inicial do movimento
 - P2: Posição final do movimento
 - T1: Tempo de início do movimento após a execução deste *script* (milissegundos)
 - T2: Tempo de fim do movimento após a execução deste *script* (milissegundos)

Script input assistance

model instruction: ACT

Model:

Direction: X

StartPos: 0.000 mm EndPos: 0.000 mm

Move time: 0 ms - 0 ms

☐ I want to set the start time to end time of the previous

step: 10

Add Change Cancel

Figura 68 – Exemplo de comando “ACT” Fonte: Simulador MotoSimEG-VRC

3.1.13. Simulação

Para aceder às funções de simulação, Figura 69, de todos os componentes criados, recorre-se ao menu [Simulation] no grupo “Payback”. Neste grupo estão as principais funções de simulação.

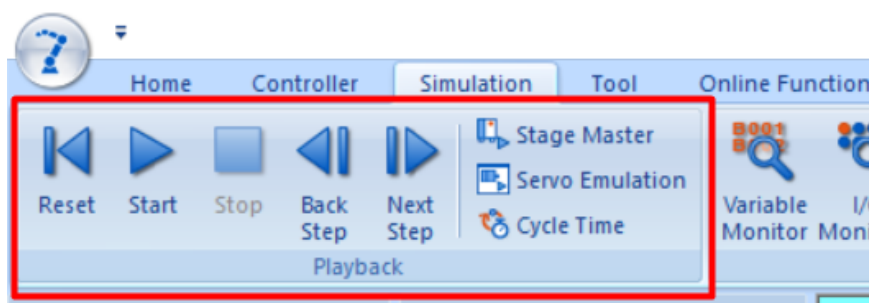


Figura 69 – Menu de simulação Fonte: Simulador MotoSimEG-VRC

A ferramenta descrita acima tem a capacidade de simular todo o processo, após a conclusão do projeto, através da tecla [START].

3.1.14. Exemplo criado para o CTCV

No decorrer do estágio, foi possível observar diversas linhas de montagem, no ramo da indústria da cerâmica. A tarefa solicitada compreendeu a simulação de diversas partes do processo de fabrico, com o desígnio de elucidar o cliente do CTCV dos recursos e equipamentos necessários para implementar na prática as referidas linhas de montagem e os custos envolvidos.

De seguida, abordar-se-á um exemplo que teve origem numa proposta de simulação de uma linha de produção, composta por três robôs, de características similares. A simulação delineada permitiu verificar o potencial do programa *MotoSim EC-VRC*, uma vez que foi possível constatar e simular um ambiente industrial, com sincronização total.

Salienta-se que o exemplo descrito abaixo é meramente representativo e contextualizado à área em estudo, sendo o programa *MotoSim EC-VRC* detentor de infinitas possibilidades de simulação.

O Projeto foi desenvolvido para a empresa Aleluia Cerâmicas, S.A., sediada em Aveiro, dedicada exclusivamente à produção e decoração de revestimentos e pavimentos cerâmicos, sendo uma referência no mundo cerâmico (SobreNós-Aleluia Cerâmicas, 2022). Este projeto consiste na automatização de uma linha de azulejos, em que o primeiro passo do processo é a despaletização de peças de cerâmica.



Figura 70 – Imagens reais de despaletização manual em linha Fonte: Própria

Na Figura 70 estão representadas imagens num contexto real, na empresa Aleluia Cerâmica, sendo possível observar a despaletização realizada com mão de obra humana. Os azulejos têm dimensões passíveis de ser manobradas por um robô, com força, alcance e outras características posteriormente selecionadas para o efeito. Os azulejos observados na Figura 70 têm a forma de um paralelepípedo, com 1000 mm de comprimento, 400 mm de largura e 30 mm de altura.

Os azulejos encontram-se numa palette, dispostos nas laterais da mesma, alternados, tal como se pode visualizar acima, na Figura 70 à esquerda.

À direita, está ilustrado o momento em que as paletes são mobilizadas e os azulejos condicionados, com recurso humano. A substituição da mão de obra por um robô tem inúmeros benefícios e vantagens, nomeadamente a nível da saúde, uma vez que elimina a necessidade de transporte de materiais pesados, e a nível da rentabilidade de tempo e produtividade para a própria entidade empregadora. O uso de um robô simplificará a tarefa, tornando a possibilidade de ocorrência de erros diminuta, como sejam danificação da peça ou acondicionamento pouco seguro.

Com a descrição da totalidade do processo, é de extrema importância ilustrá-lo como um todo. O primeiro robô (visível no canto superior esquerdo da Figura 71 abaixo), é responsável pela despaletização, colocando as peças de azulejo numa passadeira que os irá transportar até ao robô seguinte. Esta passadeira executa vários processos, que por motivos de confidencialidade, não foram descritos pela empresa.

À chegada ao robô número dois, este agrupa os azulejos três a três e coloca-os numa caixa de cartão, com recurso à garra número um. A garra número dois do segundo robô, eleva posteriormente os azulejos condicionados e transporta-os na passadeira número três. À medida que as caixas de cartão são deslocadas, é emitido um sinal pelo próprio robô, realçando a

disponibilidade de recepção de nova caixa. No decurso da mobilização das caixas na terceira passareira, estas são fechadas através de um dispositivo automatizado externo.

O robô número três realiza a paletização das caixas de azulejos, para posterior encaminhamento ao consumidor final.

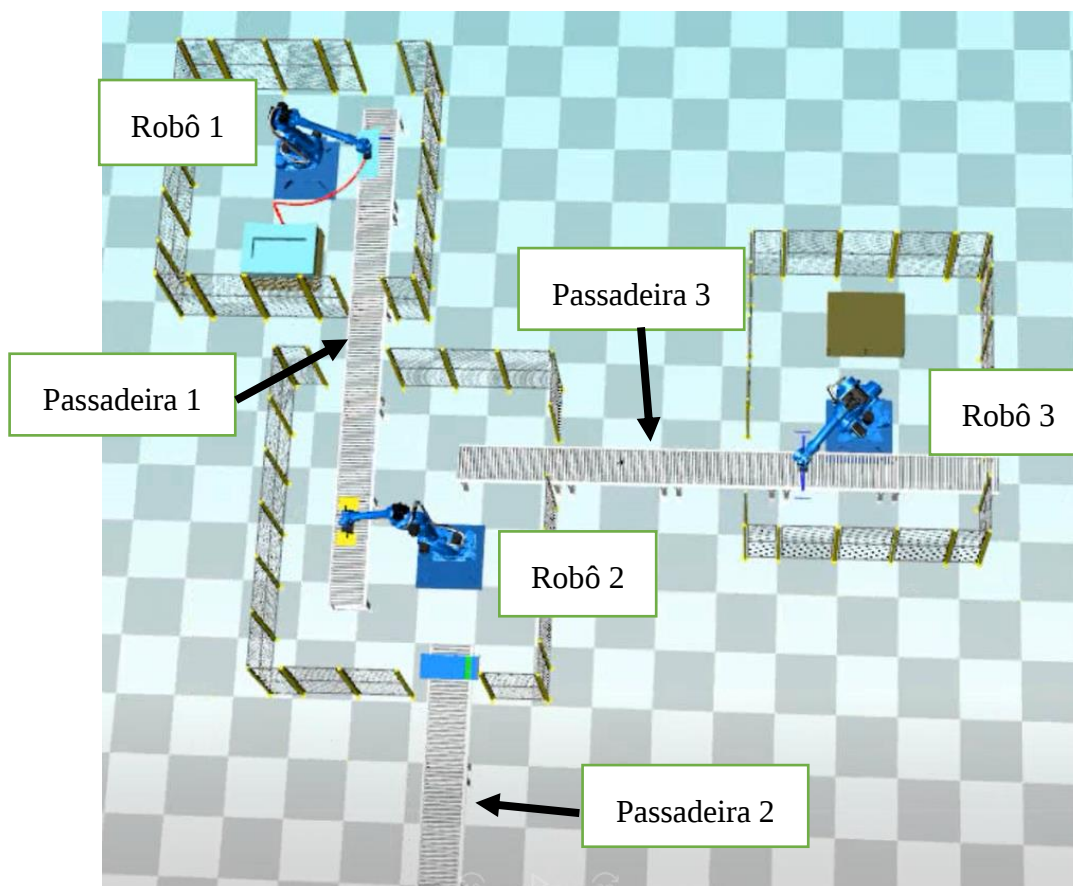


Figura 71 – Vista global da linha de montagem criada Fonte: Própria

Com recurso ao *MotoSim EG-VRC*, inicializou-se o processo de criação/desenho da paleta visualizada na Figura 72, com a respetiva disposição dos azulejos, seguindo os passos descritos no Guia do Simulador *MotoSim EG-VRC* (Criar projeto e ambiente envolvente).

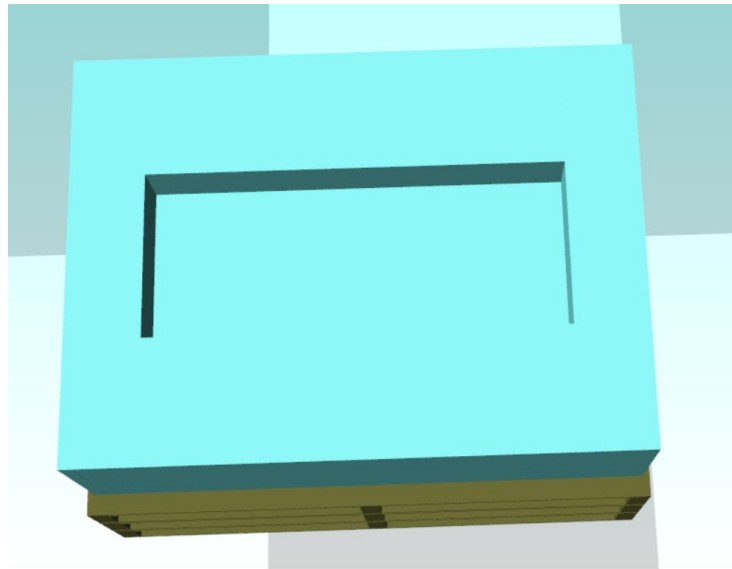


Figura 72 – Disposição criada no simulador dos azulejos Fonte: própria

Tendo em conta as dimensões e o peso dos azulejos, procedeu-se à escolha do robô mais adequado para dar resposta à tarefa pretendida. O robô eleito denomina-se GP180-120, que tem capacidade de carga de 120 kg, com controlador YRC1000, alcance horizontal de 3,058 m e alcance vertical de 4,105 m.

A Figura 73 , apresenta detalhadamente, as características do robô selecionado, com vista a complementar a informação mencionada anteriormente.

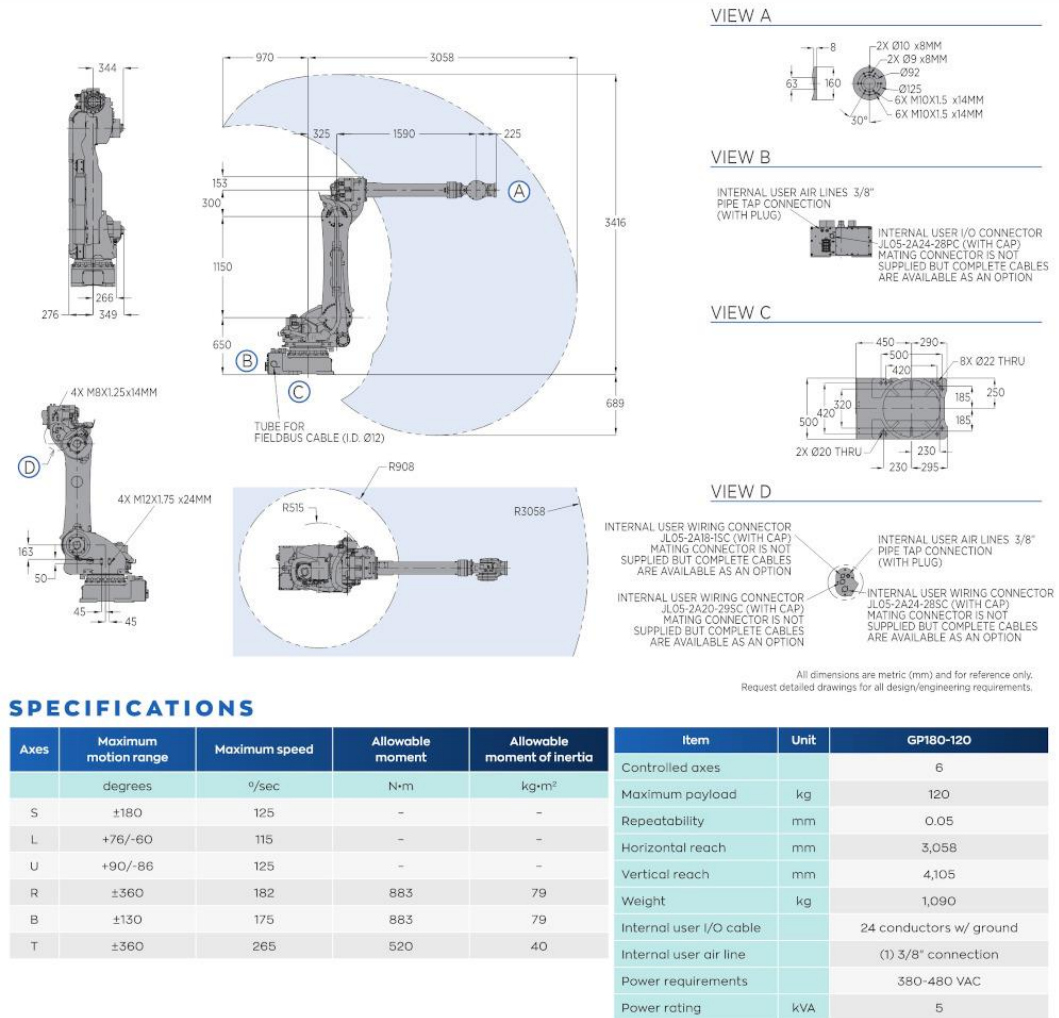


Figura 73 – Especificações do robô GP-180-120 Fonte: (Yaskawa, n.d.)

Findada a eleição do robô, o passo seguinte prende-se à seleção do tipo de garra mais ajustada para garantir o transporte e manipulação das peças de azulejo. A garra é composta por dez ventosas, de forma a certificar que o azulejo é deslocado em segurança, através de um sistema de vácuo. A ilustração da garra esta representada na Figura 74.

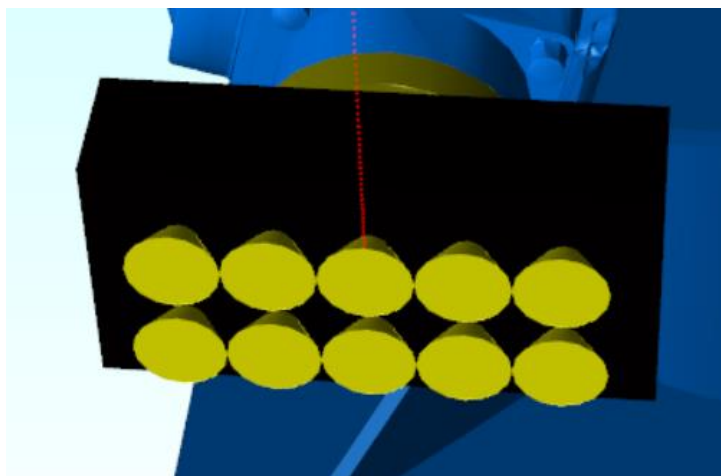


Figura 74 – Garra do robô 1 Fonte: própria

Assegurado o posicionamento e após a definição da ferramenta pré-selecionada, o primeiro robô inicia, na sua programação, a execução da despaletização. Este processo encontra-se concomitantemente explicitado no Guia Simulador *MotoSim EG VRC*, na secção “Criar Ferramenta”.

Para a concretizar a despaletização, o robô irá recolher um azulejo e colocá-lo na passadeira (assim que esta se encontre desimpedida). Procede-se à repetição do processo descrito acima, até garantir que a despaletização se encontra completa. Na Figura 75, é possível observar o procedimento caracterizado acima. Enaltece-se que a programação total do robô se encontra presente no Anexo I.

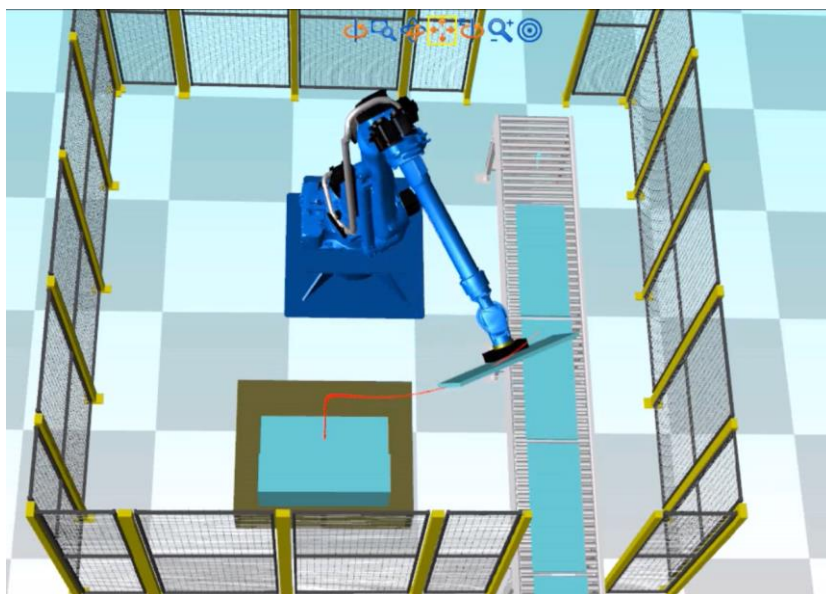


Figura 75 – Espaço de trabalho do Robô 1 Fonte: própria

Posteriormente, e terminada a etapa em que o primeiro robô executa a sua função na mobilização dos azulejos, estes seguem na passadeira nº1, com o desígnio de se apresentarem disponíveis e aptos para seguirem para o armazém e posteriormente para o consumidor.

Para consumir a entrega, os azulejos devem encontrar-se dispostos dentro de uma caixa, contendo três azulejos cada uma. De seguida, através de um sistema automatizado, as caixas são fechadas, de forma a serem transportadas e colocadas numa nova passadeira, com diferentes funções.

A otimização desta metodologia é alcançada com recurso à nomeação de uma ferramenta composta por dois mecanismos díspares. O primeiro mecanismo assenta no manuseamento dos azulejos de forma similar à ferramenta projetada para o primeiro robô. De outro modo, o segundo mecanismo prende-se à manipulação e mobilização das caixas onde constam os três azulejos este mecanismo está representado na Figura 76 abaixo.

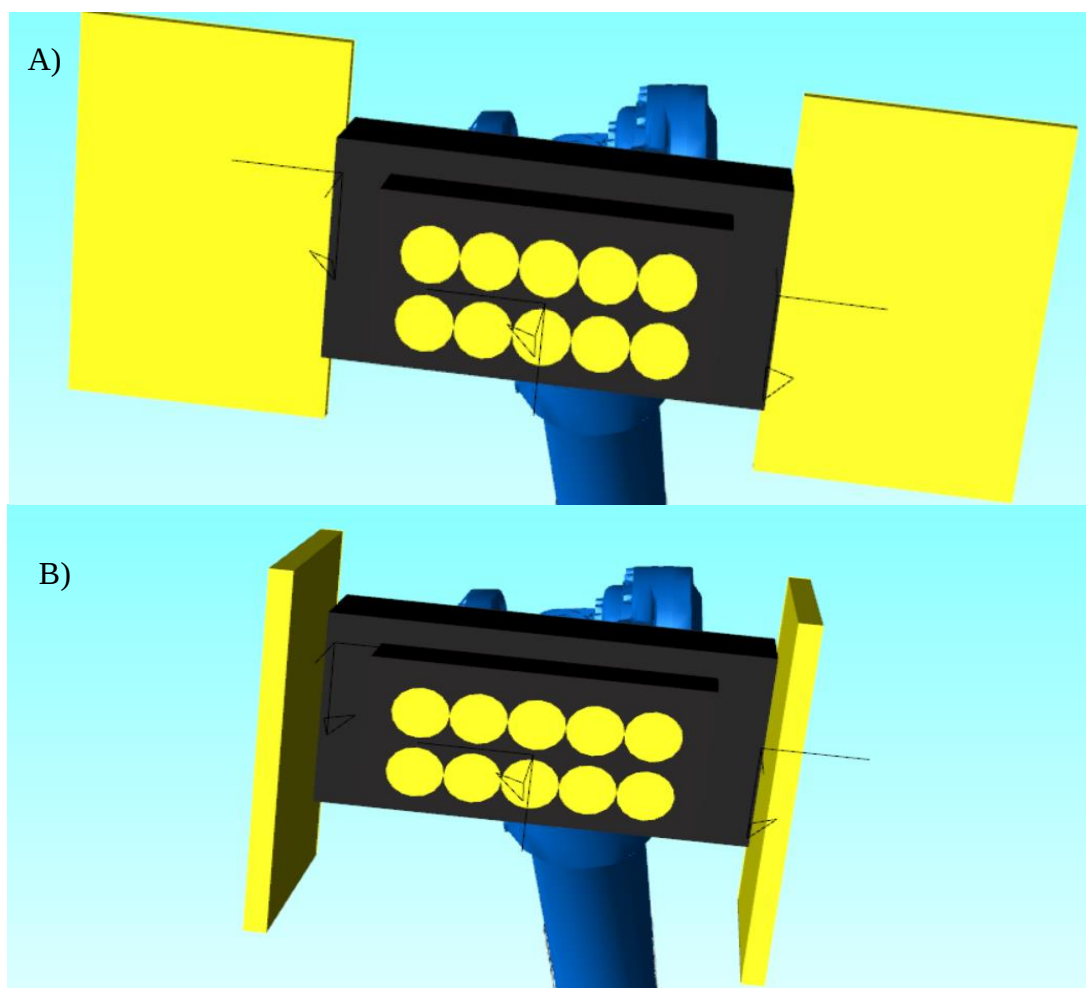


Figura 76 – Garra robô 2 mecanismo secundário A) aberto B) fechado Fonte: própria

Importa realçar que, é com recurso à ferramenta do sistema de ventosas, que o robô, Figura 77, retira da passadeira nº1 os três azulejos e os acondiciona numa caixa (de cartão). Esta mobilização é assegurada graças a um sensor que está apto para visualizar que os mesmos se encontram na posição correta.

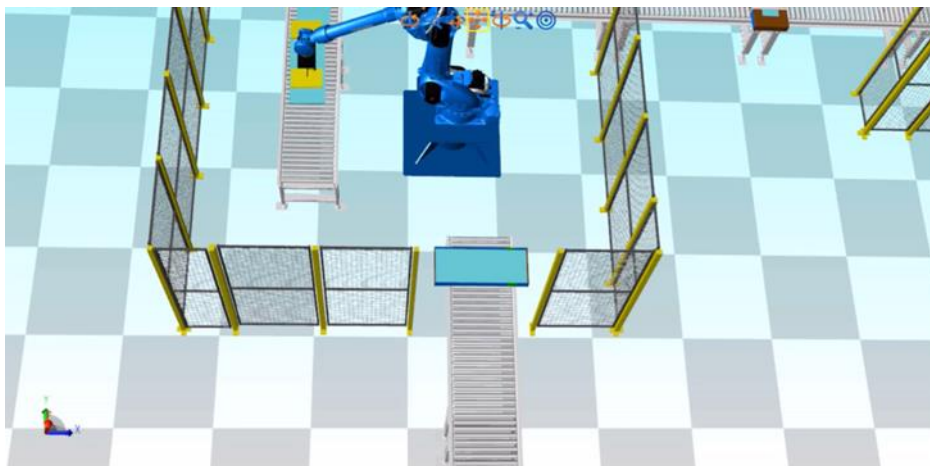


Figura 77 – Robô 2 a transportar azulejos com o uso do sistema de ventosas Fonte: própria

Decorrida a fase inicial da primeira tarefa, e após autenticação de que as caixas com azulejos se encontram devidamente fechadas, o robô inicia o transporte das mesmas. Esta transferência é alcançada com a intervenção de duas garras que, em simultâneo, transportam os azulejos devidamente empacotados para a passadeira nº3, processo ilustrado na Figura 78.

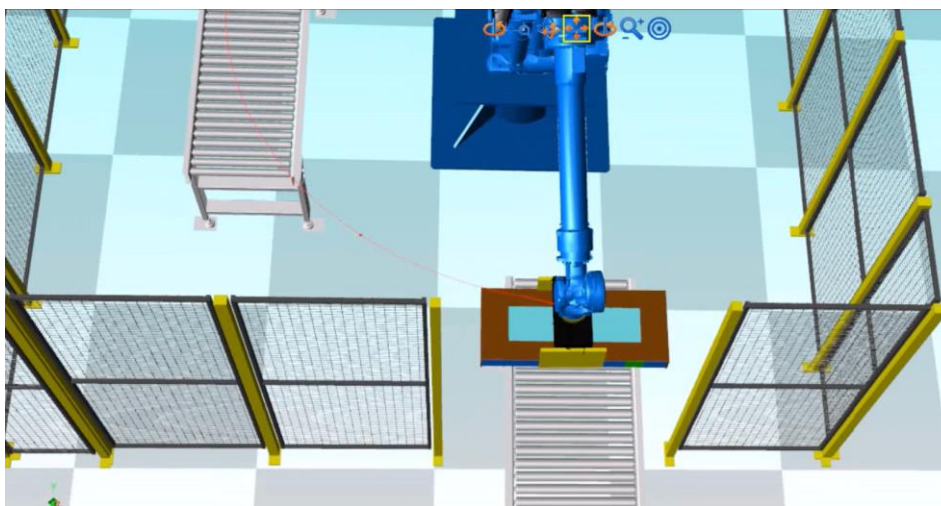


Figura 78 – Robô 2 a transporta as caixas com o uso das duas ferramentas Fonte: própria

Na terceira passadeira, Figura 79, que se encontra entre o segundo e o terceiro robô, as caixas que contém os azulejos, e que foram previamente fechadas, são finalmente seladas longitudinalmente, com recurso a uma fita de cinta. Por fim, as embalagens, ficam aptas a serem entregues ao consumidor.

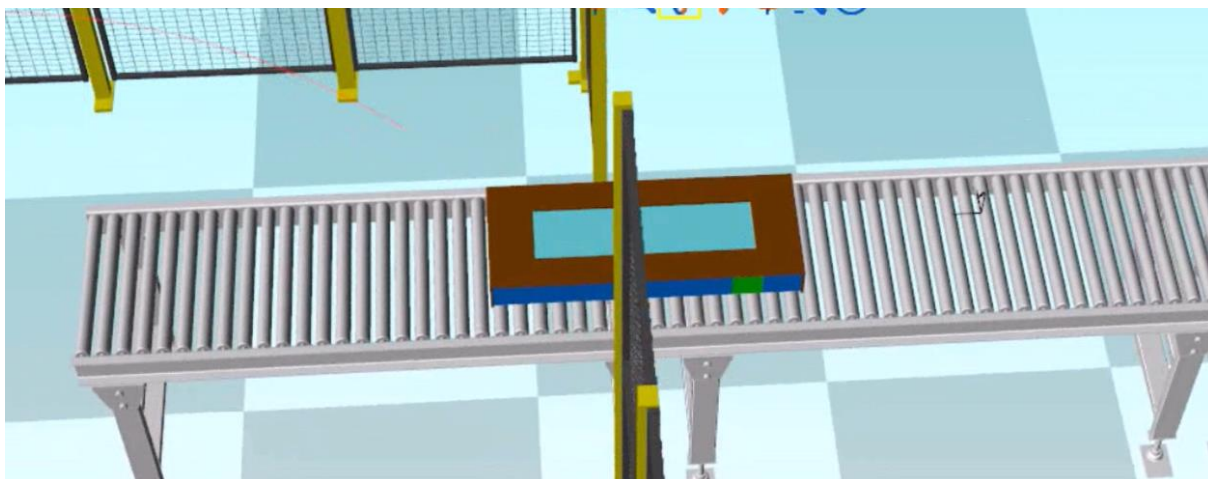


Figura 79 – Caixa com os azulejos antes da celagem Fonte: própria

Na Figura 79 acima, observa-se a fase em que as caixas se encontram seladas e preparadas para a paletização.

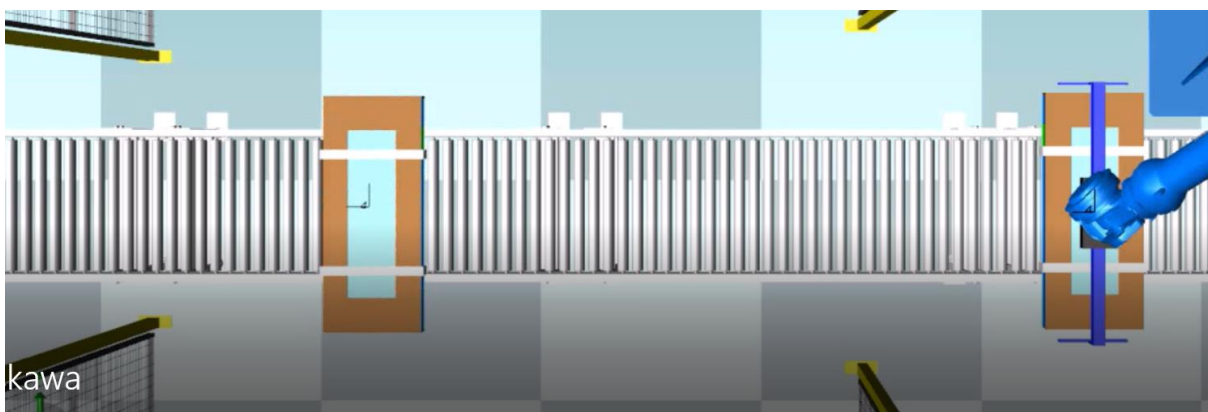


Figura 80 – Caixa com os azulejos apos da celagem

Tendo em conta a disposição das caixas na paletização, tornou-se necessária a criação de uma garra adequada e ajustada às mesmas. Foi realizado um estudo acerca do processo de paletização, tendo-se concluído que a manipulação das caixas com azulejos apenas poderia concretizar-se através da zona superior das mesmas (topo).

Atendendo ao facto mencionado, foi concebida uma garra mecânica, auxiliada por um sistema de vácuo no centro, de modo a oferecer estabilidade ao produto, e não exercer força em demasia, dada a fragilidade das peças.

Abaixo, Figura 81, verifica-se o funcionamento/procedimento acima narrado.

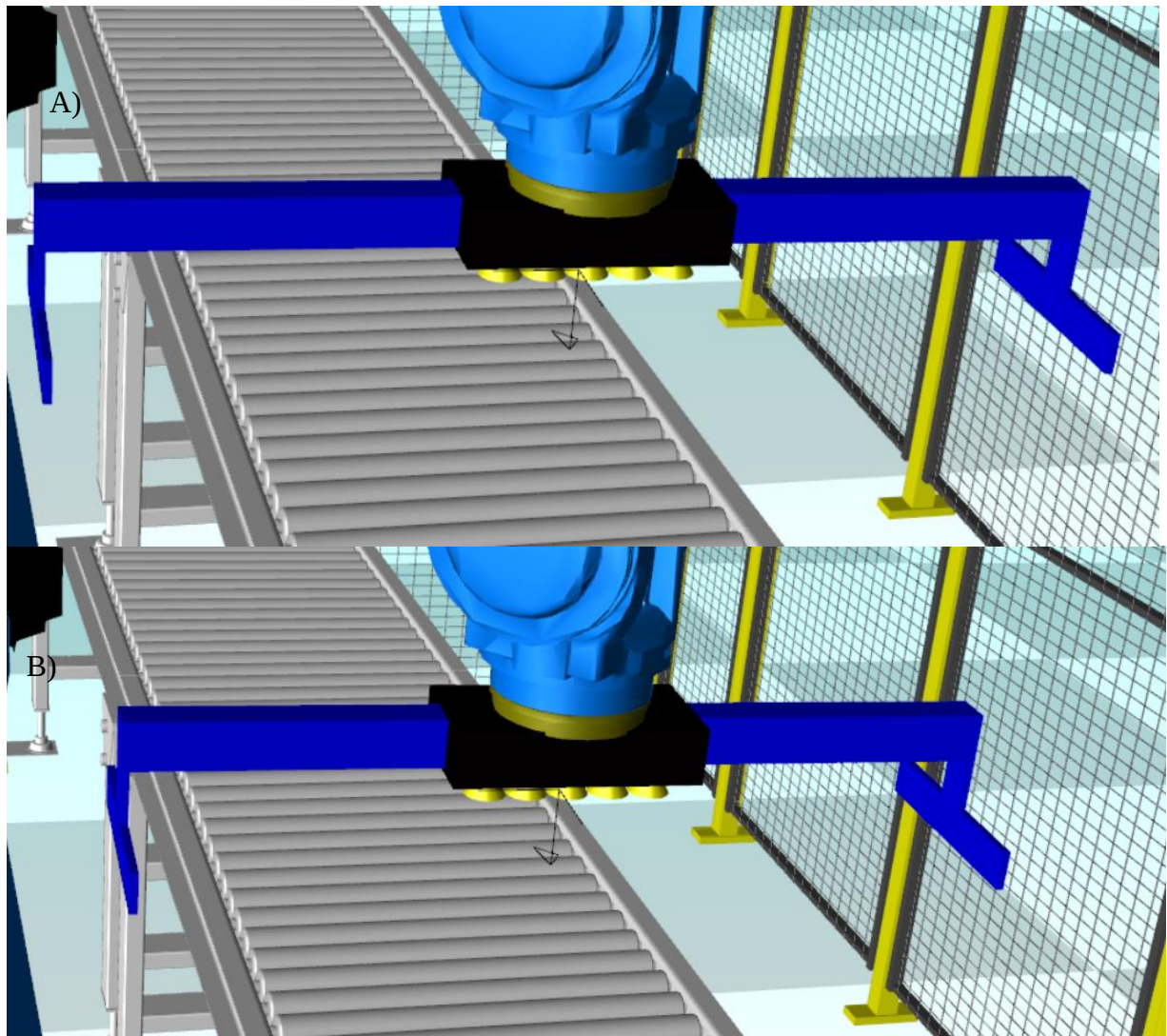


Figura 81 – Mecanismo de paletização A) Aberto B) Fechado Fonte: própria

Após concluídas as configurações e o estudo da paletização, o robô inicia o procedimento segundo a programação efetuada.

Abaixo, na Figura 82, visiona-se a ação do robô quando este se encontra a executar a paletização.

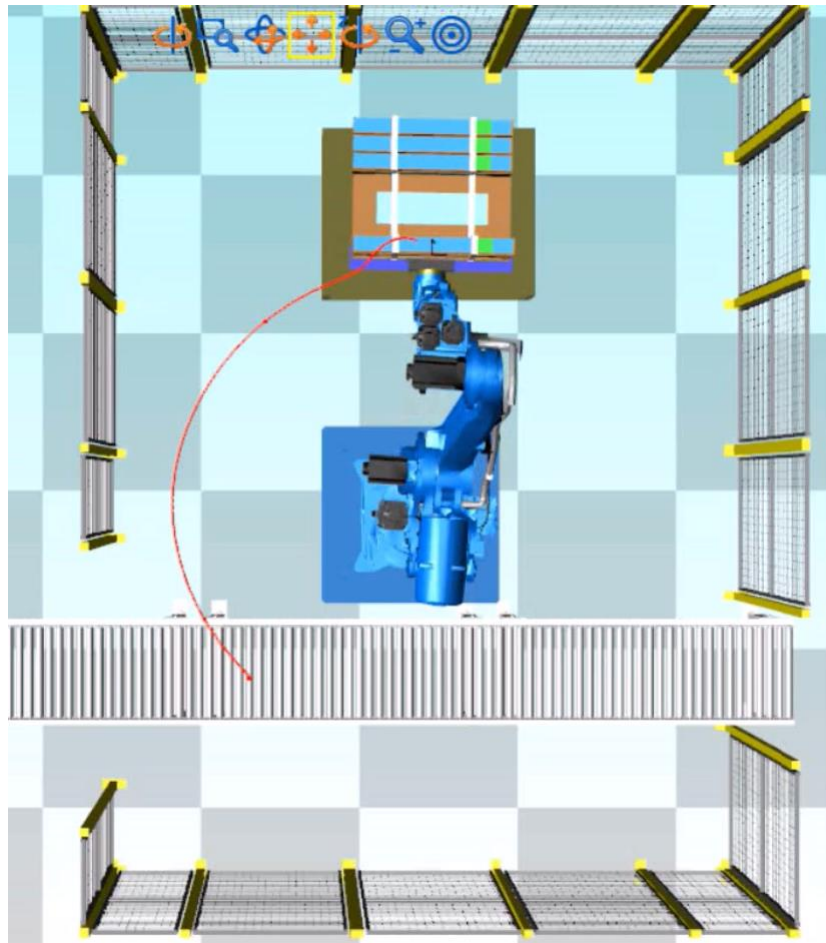


Figura 82 – Robô 3 a executar a paletização Fonte: própria

Na Figura 83, visualiza-se o momento final da paletização, no contexto real (A) - à esquerda), e na simulação (B)- à direita). Reitera-se então, as semelhanças patentes na paletização executada através de mão de obra, com recurso humano, *versus* a simulação por intermédio de um robô.

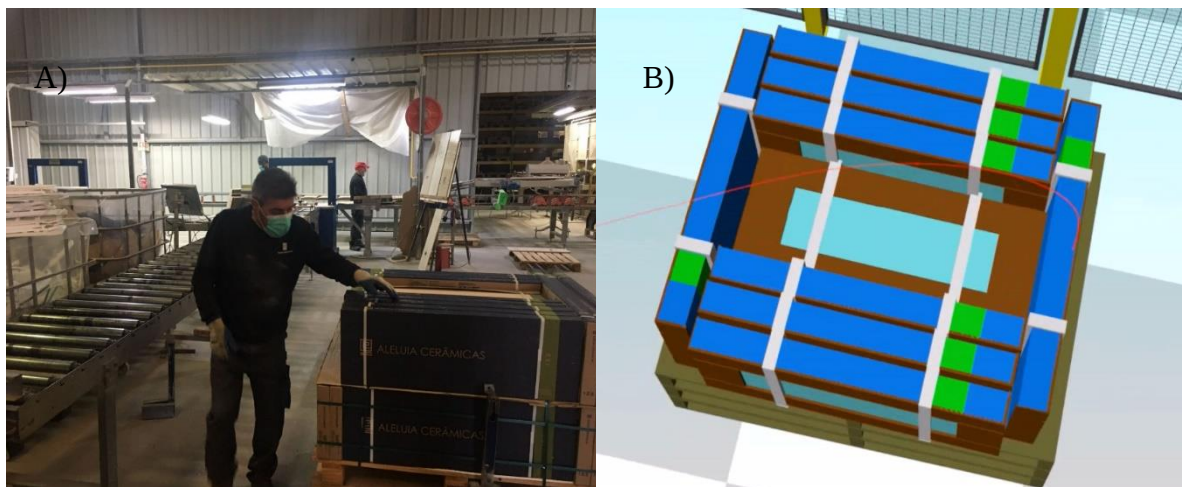


Figura 83 – Paletização final A) real B) virtual Fonte: própria

A sequência retratada reflete um dos exemplos concebidos para o CTCV, nomeadamente uma linha completa de processos industriais, de uma fábrica de cerâmica. É de extrema importância enaltecer que o delineamento, execução, programação e conclusão deste processo de despaletização e paletização evidencia o potencial do simulador, não só no que concerne à ilustração como também no que diz respeito ao próprio planeamento.

Não obstante, e após uma reflexão crítica com base nos ganhos e nas vantagens do uso de um robô, conclui-se que há a possibilidade de realizar um estudo acerca dos custos e do material necessário para a implementação da solução descrita, uma vez que é notória a utilidade de um robô nesta aplicação.

3.2. Automação nos processos de vidragem

Os processos de vidragem são indispensáveis na indústria da cerâmica, sendo vastas as hipóteses elegíveis, nomeadamente a vidragem automatizada (de pistola). Este tipo de vidragem poderia ser útil, contudo confere um grau elevado de desperdícios, não sendo possível a sua implementação em diversas tipologias de peças. Posto isto, desenvolveu-se um processo automatizado de vidragem por mergulho, contudo dependente de mão de obra humana. Este facto reflete-se numa desvantagem para este tipo de processo, tendo em conta a carência de recursos humanos. Não obstante, a vidragem por mergulho é um procedimento repetitivo e nocivo para a saúde, o que se reflete num aumento de consequências nefastas para o ser humano. Todavia, na presença de investimento, é possível alcançar elevados níveis de qualidade correspondente às expectativas.

Para melhor compreensão da importância da vidragem na indústria da cerâmica, será realizada uma breve descrição do conceito. A vidragem cerâmica consiste no revestimento a uma substância vítrea, previamente sujeita a aumento de temperatura, de modo a fundir-se. Esta vidragem surge com finalidade decorativa e de melhoria das propriedades mecânicas e de impermeabilização.

3.2.1. Vidragem manual por mergulho

Tal como mencionado acima, o processo de vidragem por mergulho é executado com recurso a mão de obra. A Figura 84 é ilustrativa de parte do procedimento, em que a peça cerâmica é submersa pelo trabalhador, num recipiente que contém a substância líquida do vidrado. No seguimento desta sequência, a peça cerâmica é mobilizada, com o objetivo de garantir que o vidrado preenche uniformemente todo o objeto.

Finalizado o mergulho da peça em produção, e após a aplicação homogênea do vidrado, esta deve permanecer imóvel e num local seguro, com o intuito de assegurar uma correta e eficaz secagem.



Figura 84 – Funcionário a vidrar um prato por mergulho Fonte: Própria

3.2.2. Descrição do processo desenvolvido

O processo desenvolvido e preconizado para implementar no contexto prático no CTCV, com recurso à disponibilidade de robôs, é uma réplica do processo manual, todavia totalmente automatizado.

Na presente secção será explanado, detalhadamente e com recurso a figuras, todo o processo sequencial da automatização da vidragem em peças cerâmicas. Numa fase inicial, o sistema do robô executa diversos preparativos, nomeadamente a calibração da câmara e a preparação da garra. Para tal, coloca-se o robô em movimento para executar o preenchimento de um copo do sistema de vácuo, com a substância líquida para a vidragem, integrado na garra do robô.

Posteriormente, o robô desloca-se para a posição de repouso para se proceder à calibração da mesa, Figura 85, assegurando que os pratos (peças de cerâmica) se encontram na área de captura de imagem. A realização da calibração está dependente de uma câmara, uma vez que o sistema desenvolvido não possui passadeira e, conseqüentemente, é impedido de recorrer a sensores para identificar as peças. Assim, a câmara irá auxiliar na calibração através da captura periódica e repetida da superfície onde se encontram as peças, caracterizando-as (em número, cor e tamanho).



Figura 85 – Robô na posição de repouso Fonte: Própria

Após a identificação de uma peça (prato) na área de trabalho, essa informação é transmitida ao robô que se irá mobilizar consoante as indicações recolhidas, e a posição de repouso em que se encontra. Importa realçar que as informações remetidas para o robô, seguem através de um sistema de visão, sistema esse que já se encontra detalhado ao longo do presente relatório (em 3.2.3), assim como outros.

Na Figura 86 é possível verificar a imagem captada e o processamento da mesma para determinar as coordenadas do centro do prato, a verde são as coordenadas reais em pixéis e a vermelho a coordenadas em mm para o robô.

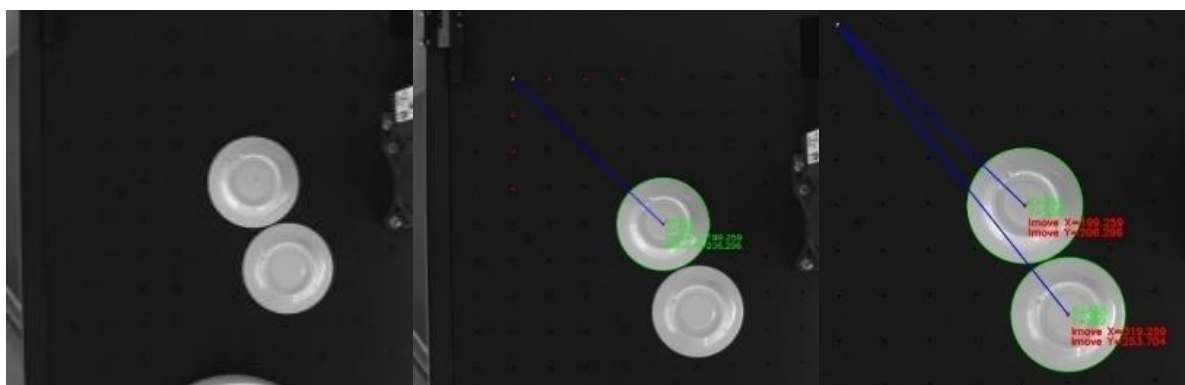


Figura 86 – Imagem captada e processamento da mesma Fonte: Própria

Assim que o controlador do robô tem conhecimento das informações necessárias e inerentes ao começo do processo de vidragem, Figura 87, o robô avança da posição de descanso e movimenta-se para a posição eleita, mediante a referência fornecida pelo sistema de visão, onde vai agarrar o prato por ativação do vácuo contido na garra projetada.



Figura 87 – Robô a recolher o prato

Com a garra do robô fixa à peça de cerâmica (prato), o mesmo é elevado e deslocado para o reservatório do líquido de vidragem, para proceder à técnica de vidragem por mergulho. Tal como reforçado anteriormente, a eficácia deste processo encontra-se dependente de um mergulho uniforme e homogêneo, com o propósito de certificar que o vidrado se encontra em quantidade similar em toda a extensão do prato. Esta homogeneidade é obtida com recurso a um motor integrado na garra, que se ativa no decorrer da submersão da peça, durante aproximadamente cinco segundos. A Figura 88 ilustra o processo descrito anteriormente.

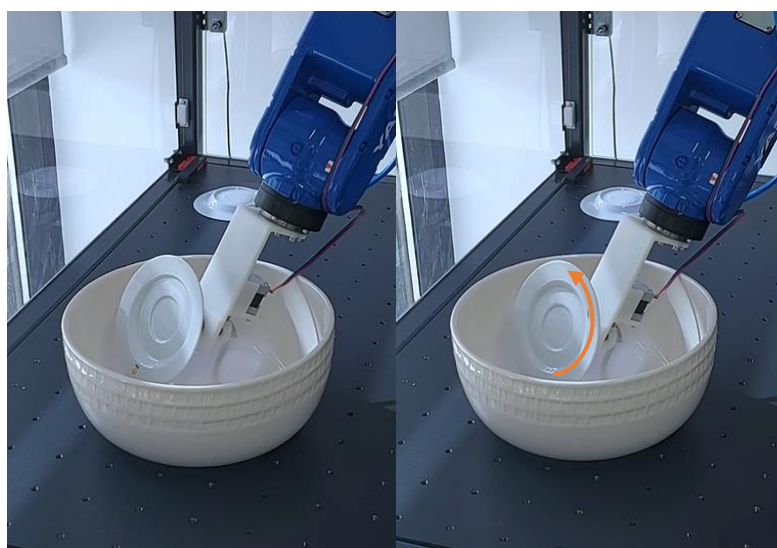


Figura 88 – Ilustração da vidragem do prato

Quando o prato cerâmico já se encontra com o líquido de vidragem aplicado, este é retirado do reservatório sendo de seguida alvo de movimentos giratórios e circulares, combinados com a rotação do próprio prato, evitando desperdícios e contribuindo uma vez mais, para uma vidragem uniforme, processo ilustrado na Figura 89.

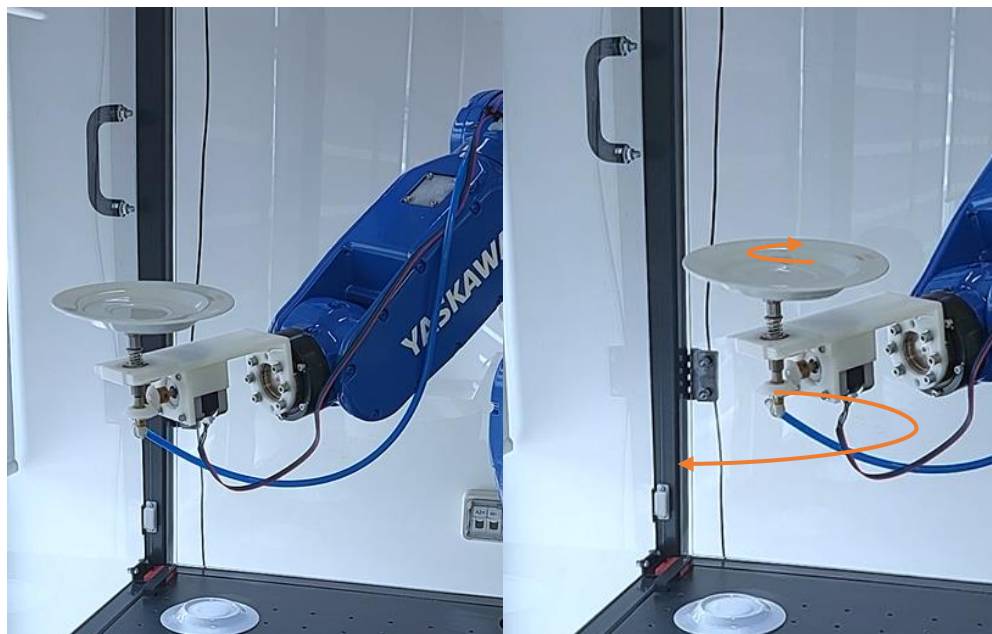


Figura 89 – Movimento de uniformização do vidrado

Após conter a substância do vidrado, o produto de cerâmica (prato) deve ser sujeito a uma secagem exaustiva, sendo imperativo assegurar que o mesmo permanece em descanso e imóvel, sobre uma superfície lisa e plana, como por exemplo uma mesa. Para possibilitar e sustentar que o prato se mantém na mesma posição, o mesmo é colocado em repouso, exposto continuamente a um sistema rotativo.

A Figura 90, reflete a repetição da sequência de vidragem por mergulho, até à mesa de apoio se encontrar totalmente preenchida por pratos de cerâmica.

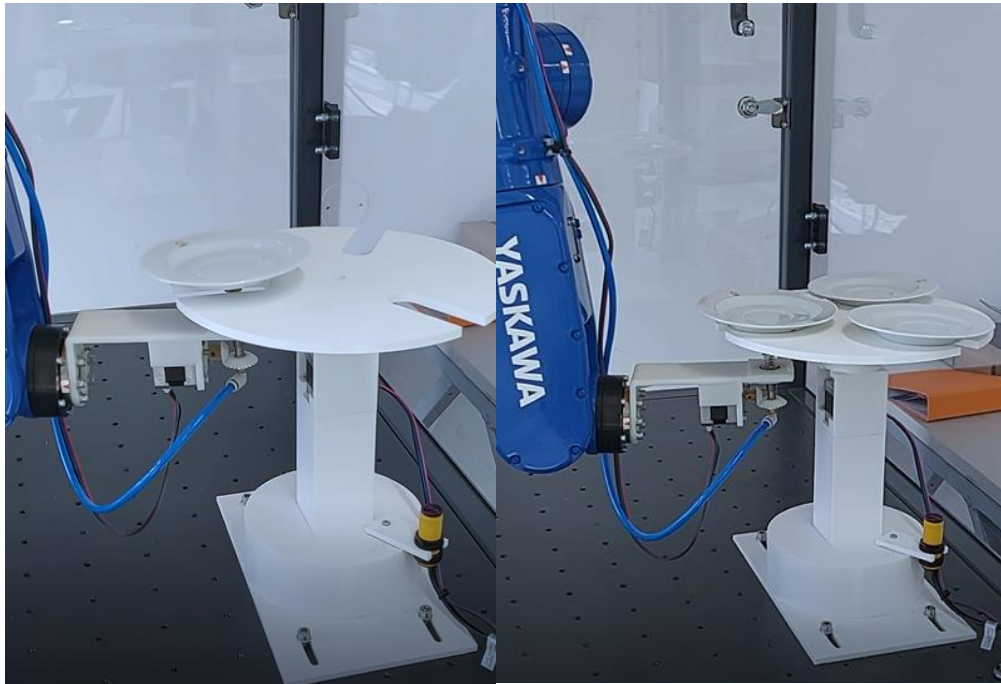


Figura 90 – Robô a posicionar os pratos Fonte: Própria

É de extrema importância realçar que, quando a mesa de apoio se encontra totalmente preenchida por pratos de cerâmica, o robô obedece à ordem de repouso, permanecendo imobilizado. Reinicia o processo após ser assegurada disponibilidade da superfície de apoio (mesa), para dispor novos elementos de cerâmica.

3.2.3. Sistema de visão

O Sistema de Visão tem como finalidade detetar o objeto na área de trabalho, relatando em seguida a sua posição e orientação. Para alcançar essa localização, a câmara deve localizar a referência do objeto, sendo esta o centro do mesmo. Imediatamente, são remetidas as informações recolhidas para o próximo robô, que tem como desígnio realizar a manipulação do objeto.

O processo empregue no projeto implementado no CTCV é composto por um sistema de visão que acarreta uma câmara *mako*, observável na Figura 91, responsável pela aquisição/captura da imagem. Esta câmara encontra-se conectada a um *Power Over Ethernet* (POE), que tem como função fornecer energia em corrente contínua (DC), eliminando assim a necessidade de utilizar fontes de alimentação convencionais, e tomadas separadas.



Figura 91 – Câmara *Mako* e luminária Fonte: Própria

O POE, que foi mencionado anteriormente conectado a uma câmara de características específicas, tem um funcionamento singular. Este encontra-se conectado a um *switch* (para criar uma rede *Ethernet*), que por sua vez está acoplado a um computador que é responsável pelo processamento da imagem obtida, e consequentemente pela transmissão dos dados adquiridos para o controlador do robô.

Urge enfatizar que o computador utilizado possui um sistema operativo denominado *Ubuntu*, funcionando por intermédio da API da *Allied Vision* (*software* intermediário que permite que dois aplicativos comuniquem). Realça-se que o programa adotado foi desenvolvido na linguagem *Python*.

Na Figura 92 encontra se um esquema de ligação dos diversos componentes do sistema de visão.

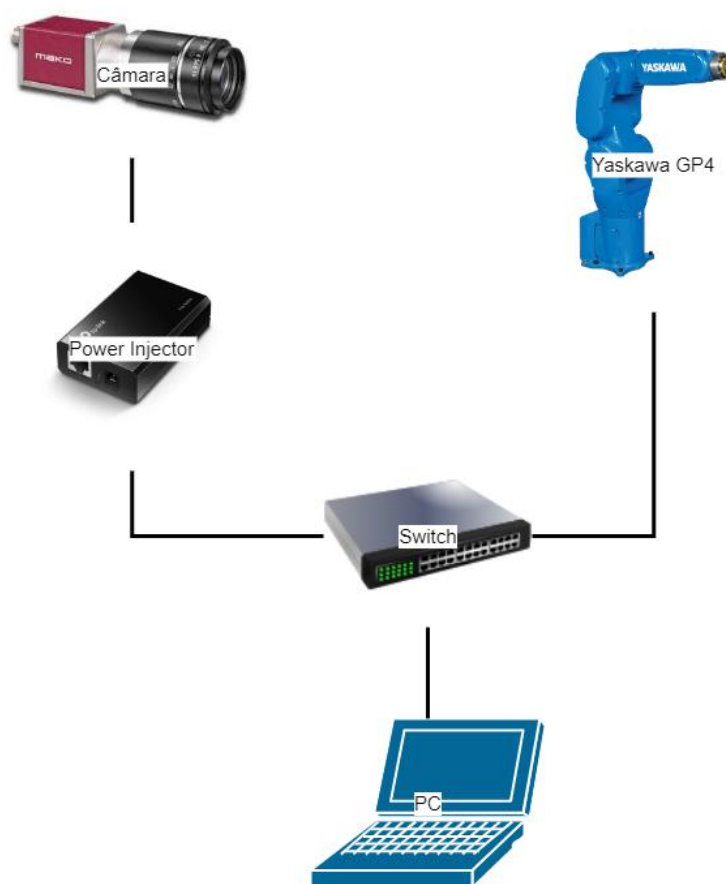


Figura 92 – Ligação do sistema de visão

3.2.4. Sistema de manipulação

A respeito dos processos de manuseamento de peças cerâmicas, a cru, nomeadamente no que diz respeito a objetos como um prato (de baixo peso e com uma superfície regular), as ventosas utilizadas devem cumprir equitativamente determinados critérios. Isto deve-se ao facto de ser de extrema importância garantir que o produto não é sujeito a nenhum tipo de deformação, ou alteração da sua forma original, ou mesmo fragmentar-se. Estas possibilidades revelam-se dificuldades substanciais na indústria da cerâmica.

Posto isto, tornou-se impreterível o desenvolvimento de um protótipo onde foi possível regular a força exercida pela ventosa, no decorrer do manuseamento e mobilização das peças de cerâmica, devendo esta serem trabalhadas manualmente.

O sistema de manipulação é composto por seis eixos, da *Yaskawa*, com as características mencionadas abaixo, na Figura 93.

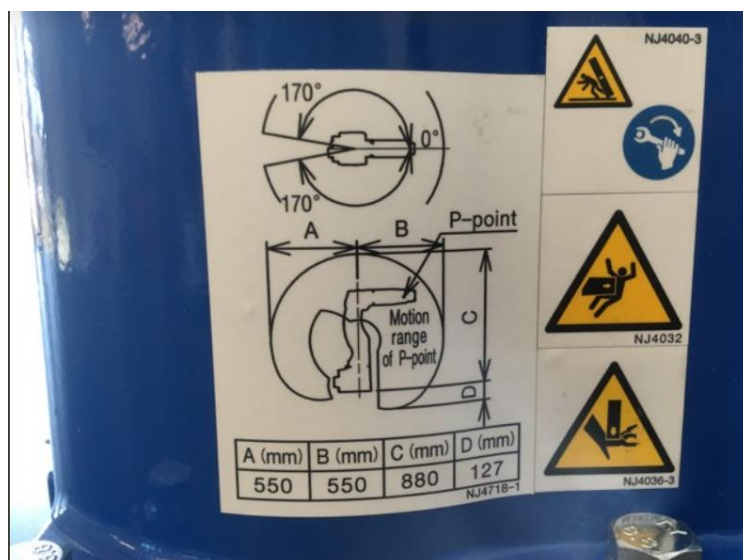


Figura 93 – Características de alcance do robô GP4 Fonte: Própria

O sistema de manipulação, Figura 94, confere uma ampla liberdade no que concerne à manipulação das peças. Para o devido efeito, foi desenvolvida uma garra, personalizada, que contém um sistema de vácuo, que permite a manipulação do prato. Agregado a este sistema, existe um motor que efetua a rotação do referido prato.

A garra foi concebida especificamente para esta aplicação, criada com a ajuda da impressão em 3D. Ademais, importa realçar que foi projetada e programada para ser acionada, atendendo ao circuito da mesma.

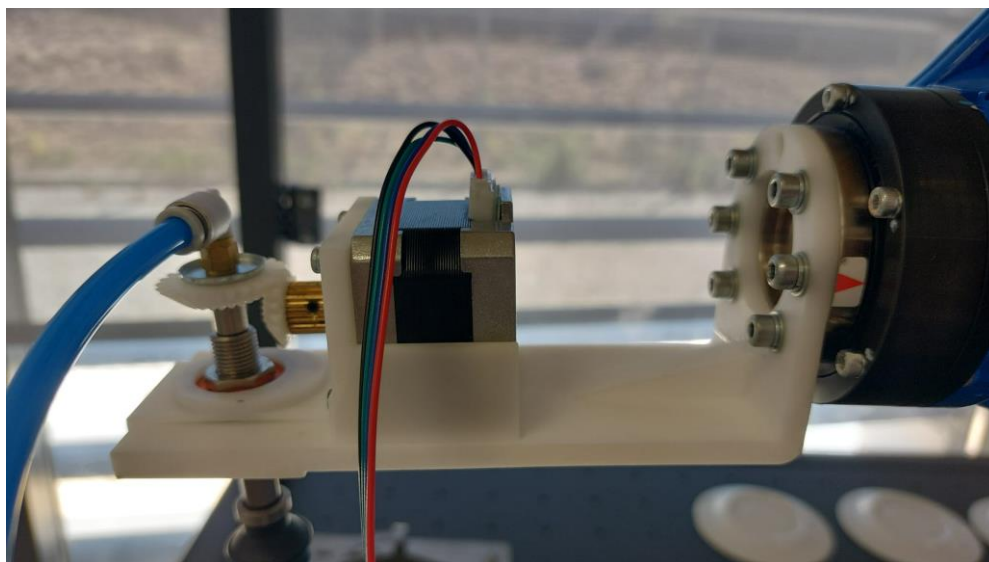


Figura 94 – Garra projetada anexada ao robô Fonte: Própria

É impreterível sublinhar que, independentemente de a garra estar conectada ao *Arduino*, as funções ‘ligar e ‘desligar’ são executadas pelo controlador.

Uma parte integrante do sistema de manipulação tem como objetivo acoplar o prato, sendo a mesma alimentada através de ar comprimido. Surgiu a necessidade de adicionar um regulador de pressão desse mesmo ar comprimido, Figura 95, com a finalidade de monitorizar a força que é exercida sobre o objeto.



Figura 95 – Regulador de pressão Fonte Própria

De igual forma, a velocidade do objeto pode ser determinada e limitada. Não obstante, esta apenas é passível de ser regulada com recurso ao *Arduino*, alterando o *software*.

3.2.5. Sistema de repouso

A eficácia e sucesso do processo da vidragem pode ser determinada pela secagem do vidrado, revelando-se como uma das fases de súpera magnitude, sendo que para tal foi gerado um sistema de repouso. Esse sistema consiste numa mesa, projetada e desenvolvida em formato 3D, Figura 96, possuindo um mecanismo rotativo, facilitador da deteção de espaços livres. Ainda assim, agregado a este mecanismo, existem sensores e uma placa, desenhada com o intuito de ser responsável pelo processamento e envio de informação, respeitante à sua localização.

O sistema de repouso encontra-se conectado ao controlador do robô com vista à adoção de tarefas que visem desimpedir a mesa, perante a probabilidade de está se encontrar totalmente preenchida com peças cerâmicas, por meio de partilha/troca de informação.

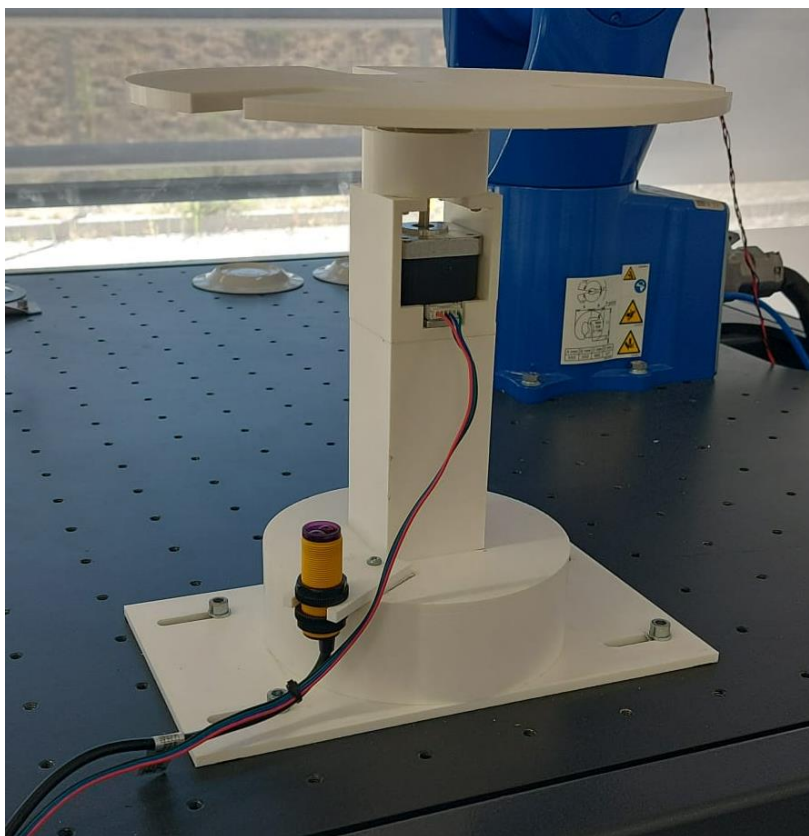


Figura 96 – Sistema de repouso Fonte: Própria

A mesa, em 3D, possui três ranhuras, com o propósito de providenciar o acesso ao robô, visto que este agarra os pratos através de secção inferior do mesmo. Não obstante, as ranhuras são de igual forma fundamentais para o alinhamento dos materiais, pois abaixo das mesmas encontram-se os sensores de distância de infravermelhos, capacitados para detetar objetos entre 30 a 80 milímetros. Através da informação recolhida pelos sensores, é possível concluir se o espaço/posição acima se encontra ou não livre.

3.2.6. Interligação e controlo dos diversos sistemas

Apesar dos diferentes sistemas estarem interligados entre si, Figura 97, e de cada um, individualmente, ser responsável pela execução de uma determinada função, o controlo de uma tarefa é regido pelo controlador do próprio robô. Posto isto, o uso de autómato é fundamental.

Contudo, é recorrente a existência de um défice associado ao controlador do robô, sendo necessário a inserção de um *Arduino*. Todavia, o acréscimo deste revela-se obsoleto com a utilização de um autómato.

Em oposição, perante um protótipo e devido aos custos inerentes serem elevados, a opção mais económica compreende a seleção de um *Ardiuno*, com um circuito projetado para executar

funções análogas, mesmo apesar de não ser ideal devido à velocidade de resposta e à segurança dos equipamentos.

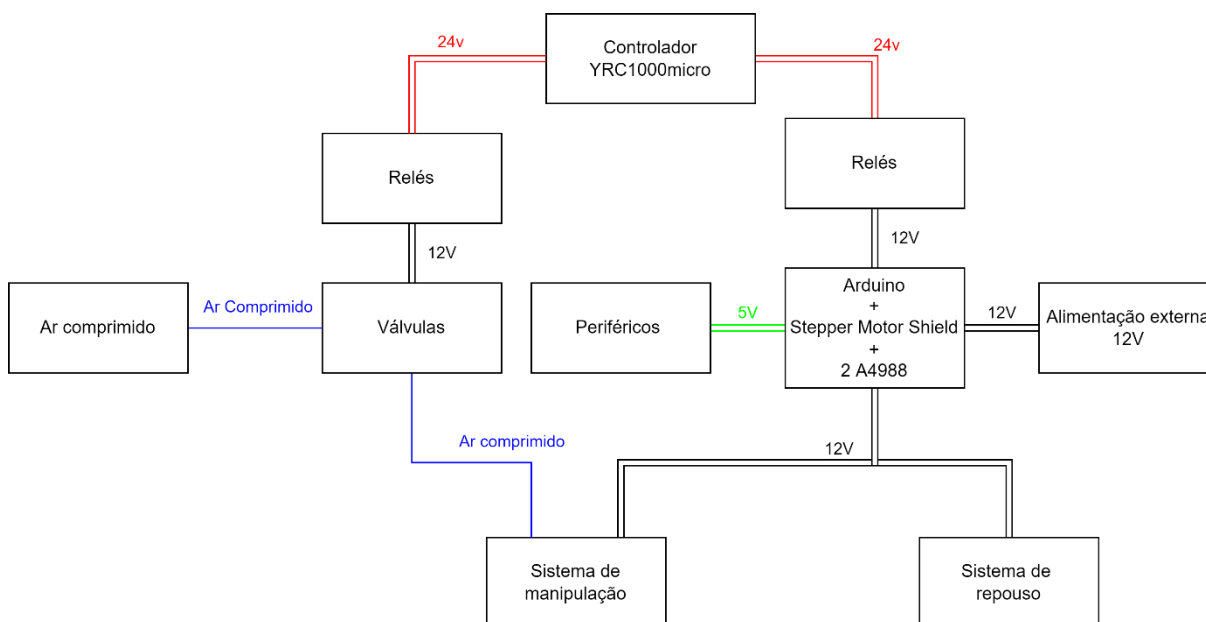


Figura 97 – Esquema de interligação dos diversos componentes Fonte: Própria

O relé foi empregue para garantir que a interligação do controlador entre o *Arduino* e a válvula de ativação do ar comprimido não é comprometida pelos diferentes níveis de tensão, entre o próprio controlador e o *Arduino*, assegurando simultaneamente o bom funcionamento, a proteção e o isolamento dos equipamentos.



Figura 98 – Relés de proteção Fonte: Própria

A combinação do uso de um *Arduino*, um *Stepper Motor Shield* e dois *Chips A4988*, Figura 99, permite o controlo dos motores, um inserido na garra com a função de girar a parte de contacto com as peças anexadas à garra e na mesa do sistema de repouso responsável por rodar o tampo para auxiliar o robô na sua tarefa.

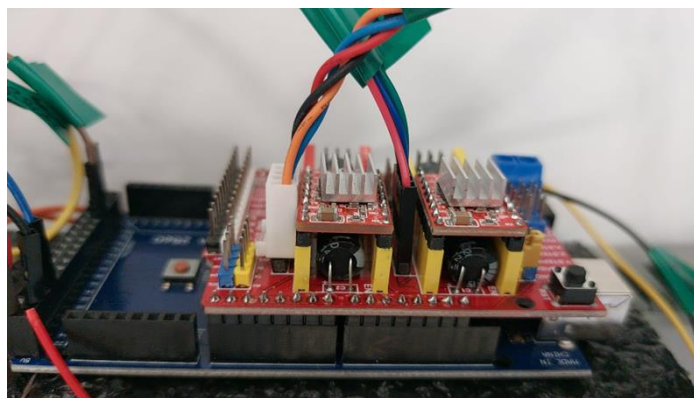


Figura 99 – *Arduino*, acoplado a *Stepper Motor Shield* e dois *Chips A4988* Fonte: Própria

O *Arduino* possui a capacidade de ser responsável pela rotação de ambos os motores, tendo sido programado para o efeito. De igual modo, recebe informações do sensor, acerca da posição em que a mesa se encontra, e processa de acordo com a necessidade de rotação da mesma.

Contíguo ao *Arduino* existe uma placa de circuito com determinadas resistências, Figura 100, bem como dois LEDs, que acendem de acordo com o funcionamento do motor. Estes LEDs são fundamentais na deteção de erros e na verificação do correto funcionamento do motor.

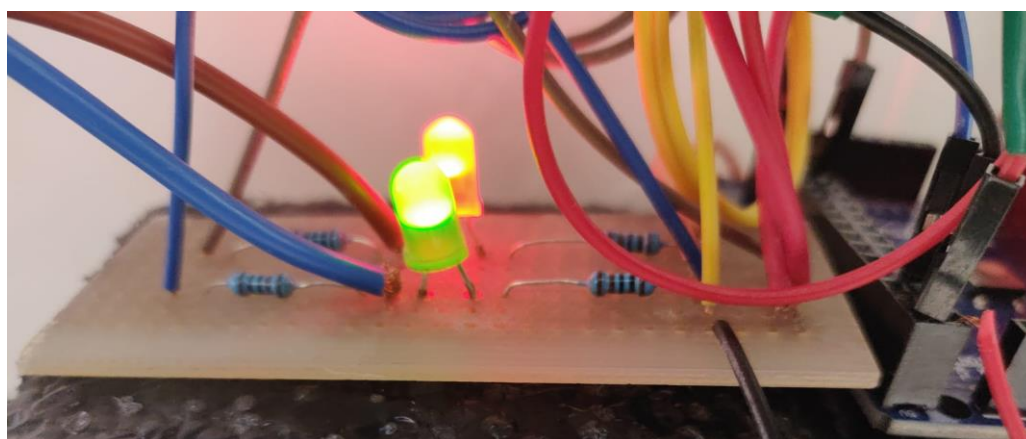


Figura 100 – Placa com função de informar o funcionamento do sistema Fonte: Própria

A articulação dos componentes referidos com o controlador, é efetuada pela placa *General Purpose Input/Output* (GPIO), representada na Figura 101, a qual é responsável pela interligação entre o controlador e os periféricos.

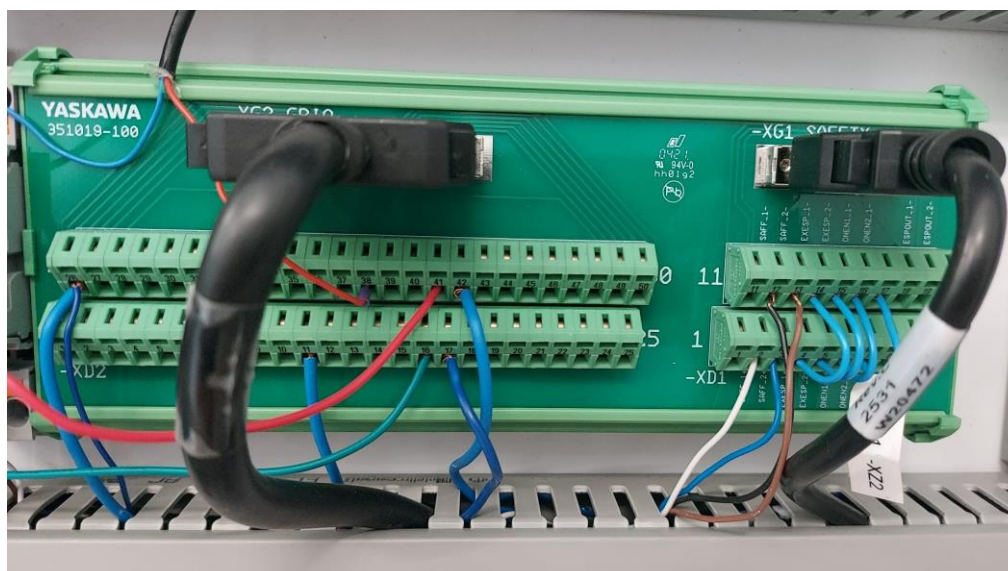


Figura 101 – Placa GPIO Fonte: Própria

Para compreender o funcionamento da porta GPIO, recorreu-se a um manual intitulado “*Install and Wiring*”, e verificou-se o diagrama de conexões da placa. Esse diagrama encontra-se ilustrado detalhadamente abaixo, na Figura 102.

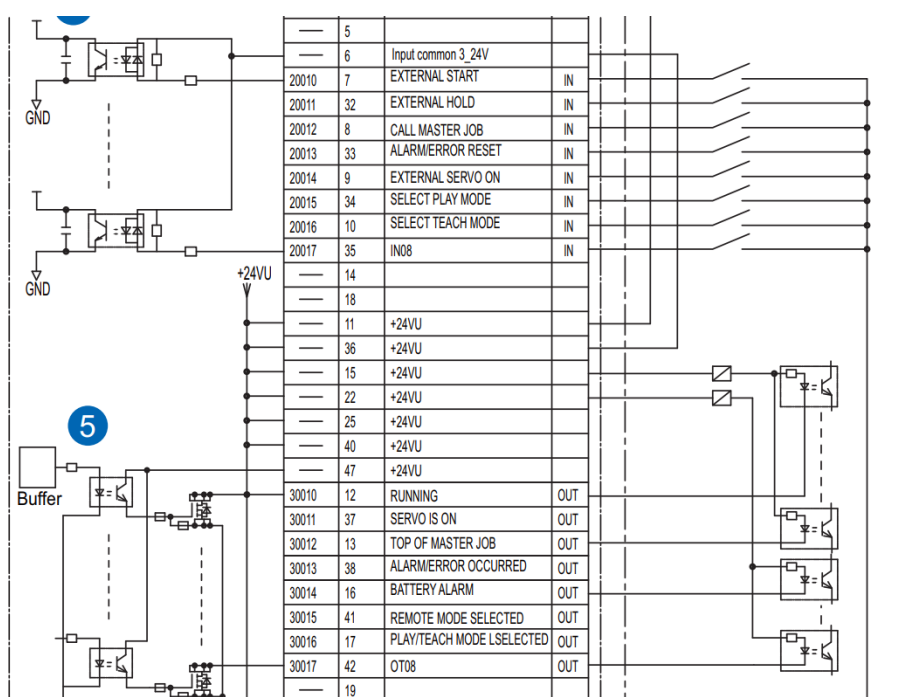


Figura 102 – Diagrama da placa GPIO Fonte: “*Install and Wiring*”

Através do diagrama é possível determinar quais as ligações a efetuar, bem como o respetivo número lógico atribuído no robô. Exemplificando, para o número lógico 30014, a posição na placa é o número 16, que tem como função “*Battery Alarm*”.

Como o robô utilizado é de ensino, foi necessário proceder a alteração do *ladder* do controlador para concretizar as ligações escolhidas, devido ao facto de este apenas ter uma entrada e uma

saída. A alteração do *ladder* é realizada pelo *Teach Pendant*, conectado em modo de manutenção. No Anexo I está presente um guia de como operar o robô.

Para facilitar a compreensão dos procedimentos utilizados, foi realizada a Tabela 1 que de uma forma concisa e sucinta, reflete as alterações realizadas e respetivas funções.

Tabela 1 – Funções das ligações e respetivos pins conectores Fonte: Própria

Placa GPIO	Número lógico	Nome User	Função
42	30017	OT8	Garra Vácuo (Sistema de manipulação)
17	30016	OT7	Garra Ejetor (Sistema de manipulação)
41	30015	OT6	Controlo do motor (sistema de repouso)
16	30014	OT5	Controlo do motor (Sistema de manipulação)
38	30013	OT4	Controlo da luminária (Sistema de visão)
11	-	-	24V
26	-	-	GND

Na Tabela 1, na primeira coluna, que corresponde ao número da Placa GPIO, consta o número correspondente ao PIN conector da mesma. Na segunda coluna, respeitante ao Número Lógico, detalham-se os números da entrada/saída para o controlador. Por conseguinte, o nome exibido na terceira coluna, é o Nome Atribuído pelo Utilizador, concomitantemente com a respetiva entrada/saída. Por último, e no que concerne à Função, é nesta coluna que se espelha a função externa que o utilizador atribui a cada entrada/saída.

Após concluir as ligações necessárias, é necessário proceder à programação do robô. Segue, no Anexo II, o guia de programação e a programação efetuada nos diversos equipamentos.

3.2.7. Programação

Finalizadas as conexões e a programação de todos os periféricos, procedeu-se à programação do robô, sendo este o elo principal e o responsável pelo funcionamento de todo o processo.

Na Figura 103 está representado um fluxograma, que espelha e apresenta de forma sucinta a estrutura de pensamento empregue na execução da programação. Importa realçar que a programação efetuada, num todo, está visível no Anexo III.

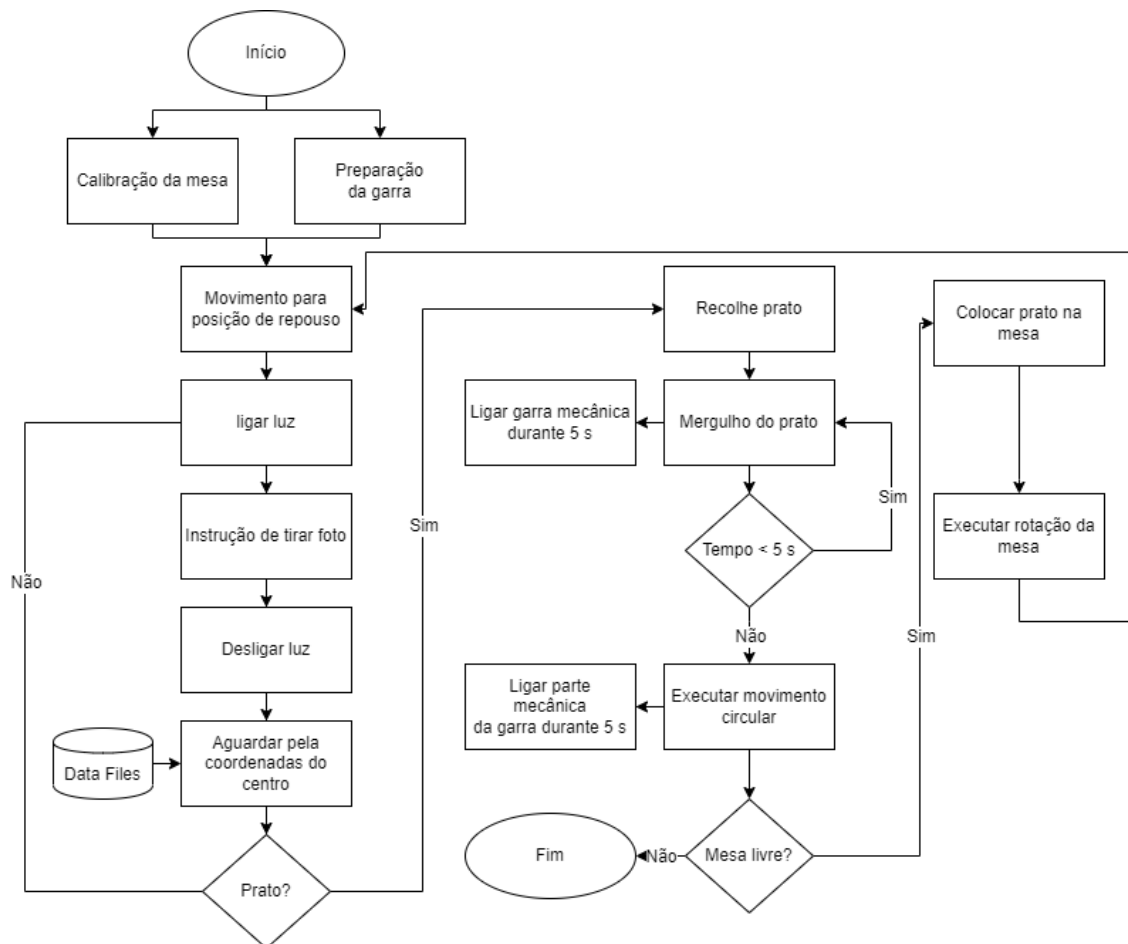


Figura 103 – Fluxograma da descrição do processo de vidragem Fonte: Própria

3.3. Automação nos processos de inspeção

Na indústria da cerâmica, apesar dos avanços tecnológicos, o processo de controlo de qualidade, na sua grande maioria, é realizado através de inspeção visual humana. Para este tipo de controlo recorre-se a pessoas com elevado grau de experiência na deteção de defeitos, tal é a sua especificidade e variedade. Com a automatização de outros processos e com o aumento resultante de produtividade e competitividade das empresas, o processo de controlo de qualidade requer maior rapidez e fiabilidade. A diminuição de recursos humanos com formação específica, e o aumento de produtividade desejado pelas empresas, traduz-se num desgaste biológico do corpo humano o qual é um dos principais fatores que diminuem o controlo de qualidade. A automação não serve apenas para processos repetitivos e de fácil execução, esta também é capaz de tomar decisões previamente programadas, ou até mesmo criar sistemas que vão desenvolvendo inteligência. Posto isto, foi desenvolvido um processo totalmente autónomo, para efetuar a inspeção e deteção de defeitos de peças de cerâmica.

3.3.1. Estudo do protótipo

O protótipo utilizado teve o apoio de diversas entidades/empresas. Financiado pela empresa Costa Verde, o protótipo denomina-se *InspectWare*, sendo que a automação e a visão do mesmo foram asseguradas por outras empresas: RobotSol e Enermeter, respetivamente.

Todavia, e apesar do protótipo se ter mantido funcional durante 50 horas, este não se apresentava funcional à chegada do CTCV. Foram encontrados diversos erros no PLC e no próprio robô, bem como em diversas ligações que se apresentavam desligadas e/ou corrompidas.

Este protótipo é composto por um sistema mecânico robotizado, que permite manipular os objetos de louça utilitária a inspecionar. Agrega um módulo de aquisição e pré-processamento (sistema de aquisição de imagem) adequado às geometrias das superfícies cerâmicas, bem como um módulo de *software*/algoritmo de visão por computador e futuramente com inteligência artificial, que permite a deteção de um conjunto de defeitos visíveis em iluminação difusa. Simultaneamente, efetua um processamento tridimensional de informação para deteção de defeitos de geometria, sendo todos estes elementos controlados automaticamente, com recurso a automação dividindo assim em diferentes sistemas.



Figura 104 – Exemplos típicos de defeitos encontrados nas peças cerâmicas Fonte: Própria

O Sistema de Controlo que é composto por um autómato (responsável por controlar os diferentes dispositivos, e processar a informação), cartas de entrada/saída, *DeviceNet Master* (entre outros), conectados respetivamente aos diversos dispositivos, recorrendo a protocolos de comunicação. De entre os existentes, foram utilizados os seguintes: *EtherCAT EtherNet* e *DeviceNet*. Como se pode verificar o esquema da rede de comunicação na figura abaixo. Simultaneamente foi elaborado um HMI, o qual foi programado para proporcionar interação humana com o controlador.

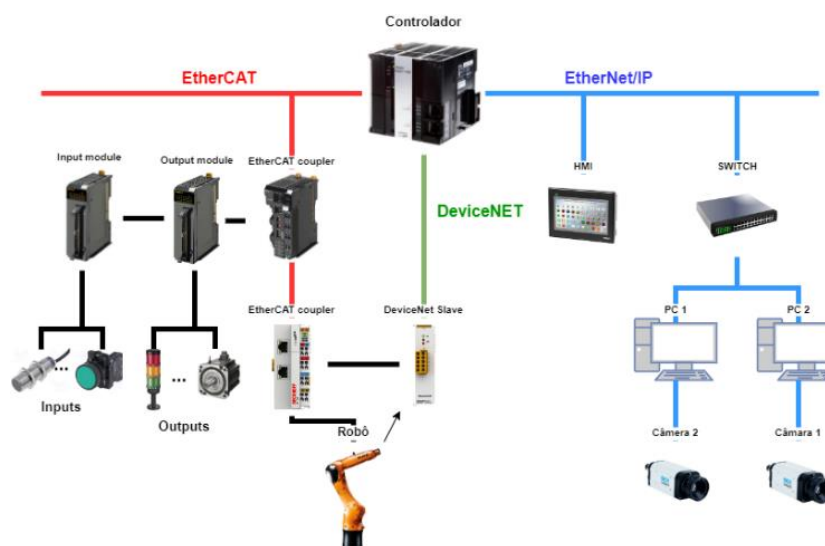


Figura 105 – Esquema da rede de comunicação Fonte: Própria

A tecnologia *Ethernet for Control Automation Technology (EtherCAT)* foi utilizada como suporte ao controlo de entradas e saídas onde o controlador (NJ301) é o mestre sendo composta por: 1 node NX-ECC202 acoplado a um módulo de saídas e a um módulo de entradas e um 2 node composto por um EK1100.

DeviceNET onde o controlador (NJ301) se encontra acoplado ao *Device Net Master CJ1W-DRM21* que se encontrando ligado ao módulo escravo EL6752-0010. O Terminal permite dentro de uma rede de terminais *EtherCAT* a integração de qualquer dispositivo *DeviceNet*.

A rede *EtherNet/IP* está responsável pela comunicação com o HMI, os PCS utilizando um *switch* para garantir a comunicação com ambos, os PCs estão ligados a câmaras, mas utilizam uma rede independente.

O sistema de transporte é composto por três tapetes, sendo um de entrada e dois de saída, onde será realizada uma seleção e separação das peças com e sem defeito.

O Sistema de Manipulação consiste numa célula robotizada, com um alcance máximo de 901.5 mm (com adicional alcance de 140 mm de alcance da garra), estando o robô posicionado no centro do protótipo. O robô tem como funcionalidade retirar os pratos do tapete de entrada, colocá-los nas zonas de inspeções e, posteriormente, encaminhá-los aos respetivos tapetes de saída. Este robô terá nele integrado uma garra, composta por 3 ferramentas alimentadas por ar comprimido, onde cada uma é responsável por colocar, retirar e virar a peça, respetivamente.

Na Figura 106 estão representadas, a estrutura da célula robôtizada – em a) -, e em b) o sistema de manipulação acima mencionado detalhadamente.

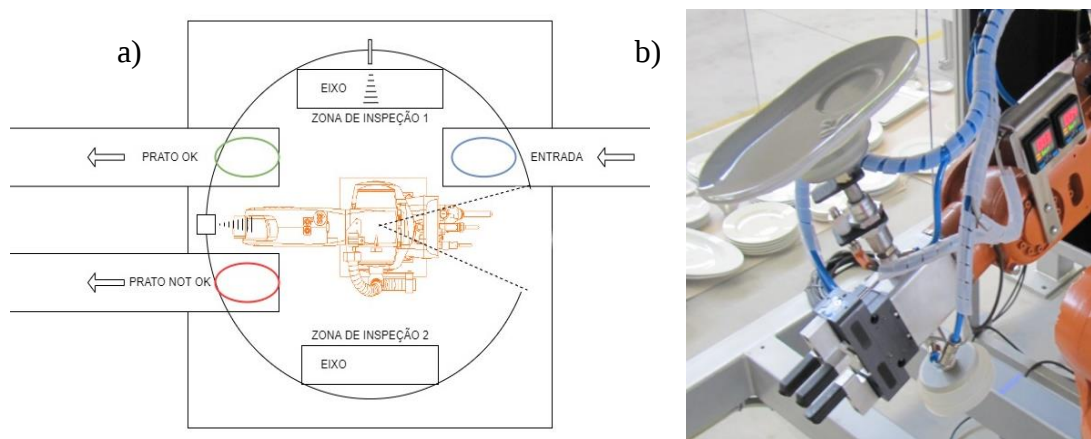


Figura 106 – a) Estrutura da célula b) Sistema de manipulação Fonte: Própria

O Sistema de Aquisição representado na Figura 107, contém uma câmara, um laser e iluminação, encontrando-se posicionado por cima do prato. O prato posiciona-se num suporte, acoplado a um eixo linear, a imagem a processar é adquirida ao efetuar um movimento linear abaixo da câmara.

A visão 2D e a visão 3D são indispensáveis para capturar as imagens necessárias, sendo responsáveis pela deteção de defeitos de brilhos e cores, e defeitos de geometria, respetivamente.

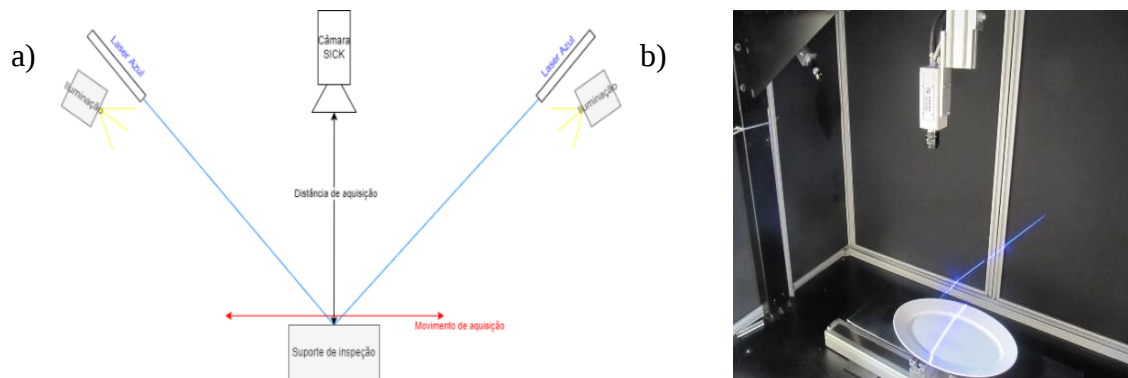


Figura 107 – Sistema de aquisição: a) projetado, b) realizado Fonte: Própria

O protótipo desenvolvido numa fase inicial funciona da seguinte maneira: as peças de cerâmica são encaminhadas até ao tapete de entrada, que possui na sua extremidade um sensor que tem como função enviar um sinal para o autómato. Por conseguinte, o robô – que se encontra em posição de repouso – aguarda a informação de que o objeto (prato) está posicionado na entrada. De seguida, uma câmara 2D captura uma imagem do prato nessa mesma posição. Com estas duas informações, o robô está apto a processá-las e a recolher o prato, com recurso à garra número 1.

Após deter o prato, o robô inicia a verificação de defeitos através de um sistema de inspeção livre, que contém 2 garras. Para executar a inspeção de um prato, o mesmo é colocado sobre uma plataforma que é movida através de um motor linear. Este movimento é realizado através de uma instrução dividida em 3 fases, sendo apenas duas delas iluminadas por 1 laser, contrário ao movimento da plataforma, adquirindo as imagens necessárias. Findadas as 3 fases, repete-se o processo novamente, distinguindo-se da sequência anterior, neste momento, o robô mobiliza o prato (vira-o), expondo a face contrária do prato para o sistema de visão. Posteriormente, é realizado um novo movimento de aquisição, onde após o processo de inspeção do prato e o robô possuir a garra número 2 disponível, desloca o prato para o respetivo tapete de saída.

Posto isto, foi elaborado um projeto composto por inúmeras instruções de diversas linguagens de programação de PLC, com o intuito de interligar os sistemas construídos com o sistema de controlo convertendo fluxograma a) representado no *Grafcet* b) abaixo representado.

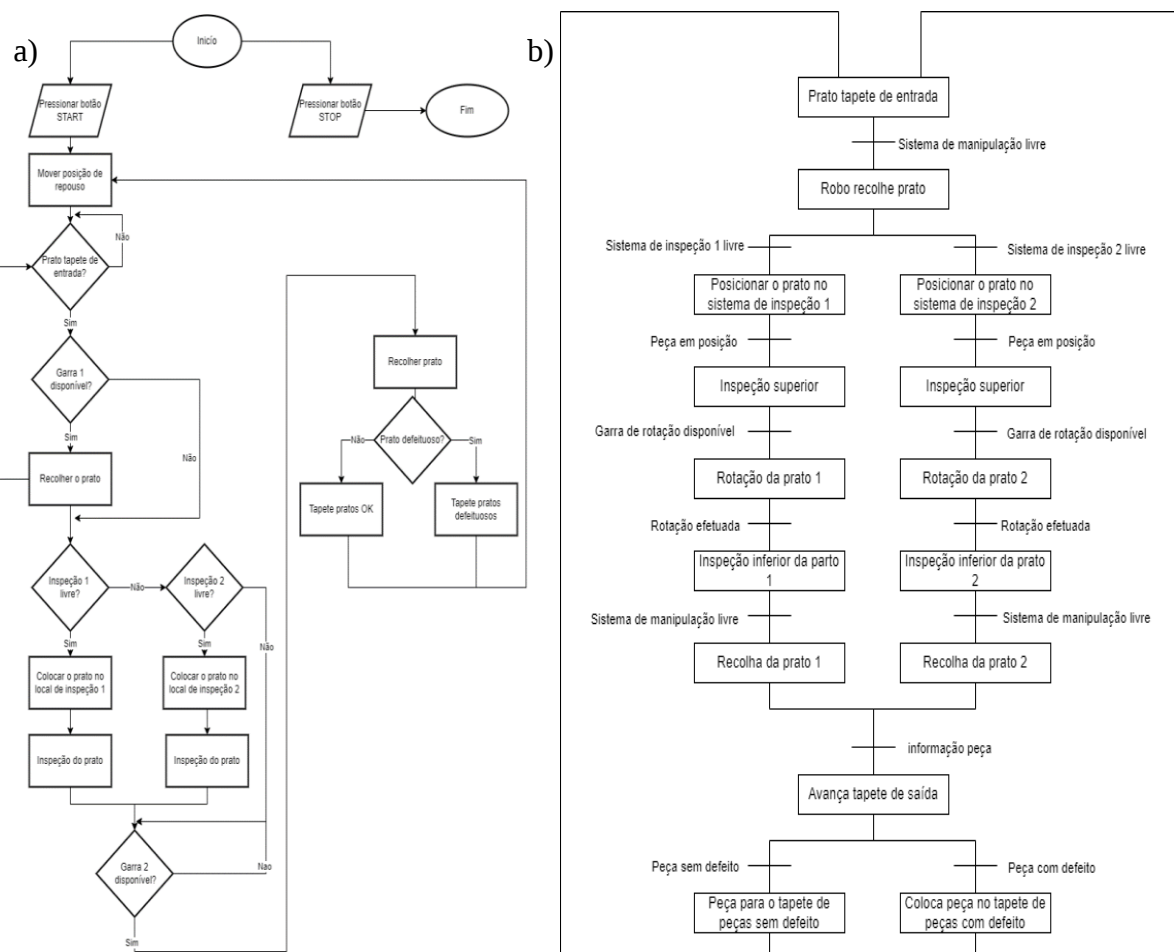


Figura 108 – a) Fluxograma de instruções b) Grafcet de instruções Fonte: Própria

Neste projeto está em desenvolvimento a parte de inteligência artificial a qual permitirá economizar o tempo de processamento de imagem e consequentemente otimizar o processo de análise e inspeção da peça de cerâmica.

3.3.2. Programação PLC

A programação PLC é responsável pela análise das entradas e acionamento das saídas. O programa utilizado denomina-se *Sysmac*, e foi usado o tipo de programação *ladder*.

No Anexo IV, é possível encontrar toda a programação do PLC, detalhadamente. A Figura 109 surge como um exemplo da metodologia da programação da iluminação do sistema de inspeção, anteriormente explicitada no ponto referente ao estudo do protótipo.

Tal como referido acima, o prato posiciona-se num suporte acoplado a um eixo linear. Este eixo linear deve executar três movimentos, numa plataforma, a uma velocidade constante. No decorrer do primeiro movimento, sucede-se a análise do primeiro laser. Abaixo, na Figura 109, é possível constatar que se o alarme não se encontrar ligado e a variável *step* for igual a 115 (variável acionada para o primeiro movimento do motor na direção positiva) durante 30 segundos, a saída do laser liga e fornece essa informação ao HMI. Evidencia-se que o laser pode, de igual forma, ser acionado através do modo manual do HMI.

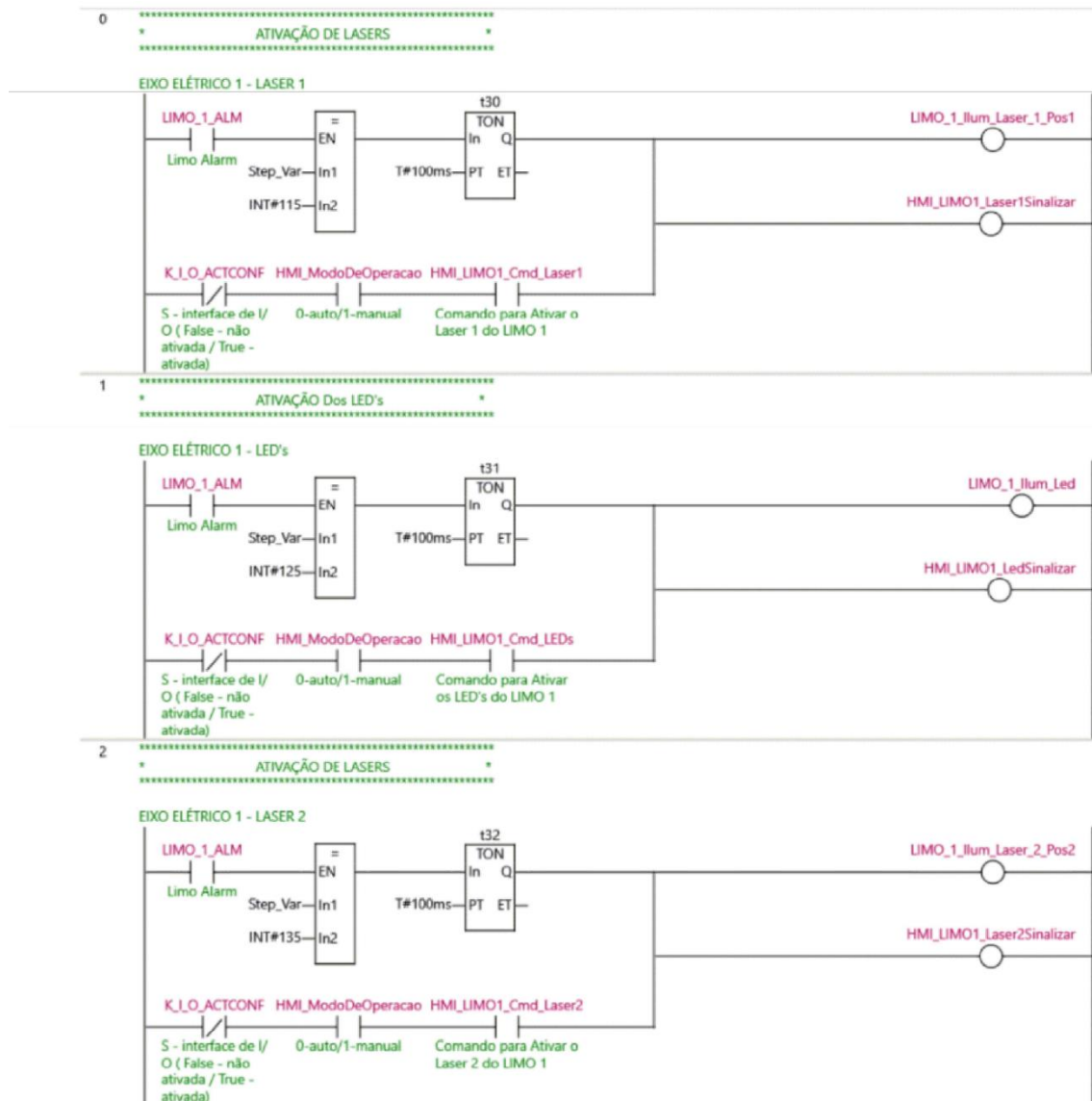


Figura 109 – Programação em *Ladder* do Sistema de Iluminação do Sistema de Inspeção Fonte: Própria

Esta lógica repete-se, novamente, duas vezes. A primeira diz respeito à análise do prato com iluminação com recurso a luzes LED e, por sua vez, a segunda apenas com a presença do laser, com inclinação oposta.

3.3.3. Interface homem-máquina

Com o intuito de assegurar a interação homem-máquina, foi programada uma botoeira que contém diversos botões, com fins distintos. Esta botoeira possui um botão de emergência (vermelho), de maior dimensão – botão ‘cogumelo’ -, um botão para iniciar o processo (verde) e um botão para parar o processo (vermelho), tal como se pode visualizar na Figura 110. Realça-

se que na botoeira constam ainda mais dois botões, cor-de-laranja e azul, que não detêm ainda funções atribuídas.



Figura 110 – Botoeira principal Fonte: própria

Foi programado um HMI para os operadores comunicarem com a máquina, com recurso ao *CX-Programmer* da *OMRON*.

O HMI tem várias funcionalidades, como a monitorização em tempo real, o modo manual para forçar entradas/saídas, a análise para estudo de dados e alarmes. O modo manual encontra-se protegido por uma palavra-passe (1000), para garantir que o uso é restrito apenas a operadores qualificados. Na Figura 111 é possível observar a estrutura do HMI.



Figura 111 – HMI Sistema de inspeção Fonte: Própria

3.4. Feiras e Artigos

Durante o estágio foram publicados dois artigos: 1) João Mendes, André Santos, Rogério Torres, Sandra Carvalho, António Baio Dias, Victor Francisco, N. M. Fonseca Ferreira, “Estudo e demonstração de aplicações de Robótica industrial para a Indústria Cerâmica”, revista Técnica #14: Transformação Digital da indústria,” de Cerâmica e Vidro, publicado em 27 de maio de 2022 e 2) João Mendes, N.M. Fonseca Ferreira, Victor D. N. Santos, “*Inspection in the Ceramics and Glass Industry*”, *International Conference on Mathematical Analysis and Applications in Science and Engineering (ICMA2SC’22)*, ISEP, Porto-Portugal, June 27th- 29th, 2022.



Figura 112 – Logotipo da conferência ICMA²SC’22.

Foi elaborado um poster para a conferência ICMA²SC’22 apresentado em 27 de junho de 2022.

Foi marcada a presença na feira 360 Tech Industry 2022 a representar o CTCV.



Figura 113 – Presença na feira 360 Tech Industry 2022.

4. CONCLUSÕES E TRABALHO FUTURO

Conclui-se assim que a automatização dos processos de inspeção cerâmica é um passo seguro e prático para a modernização do setor da indústria cerâmica, uma vez que sem esta automatização, as empresas cerâmicas ficam dependentes de um conjunto cada vez mais reduzido de trabalhadores especializados na inspeção de peças cerâmicas.

Através destes projetos e dos resultados obtidos, pode-se constatar que a automação oferece inúmeras vantagens e soluções mais viáveis na área industrial. A automação do processo de vidragem por mergulho diminui os efeitos nefastos para a saúde do operador, comparativamente com o processo de vidragem manual. Por outro lado, o processo automatizado de vidragem por pistola acarreta desvantagens por se demonstrar nocivo, visto que a pulverização da tinta em larga escala aumenta o risco de inalação. A vidragem por mergulho garante também resultados consistentes e versatilidade das peças vidradas produzidas.

Para além disto, através do centro de inspeção de defeitos, o procedimento automatizado acaba por ser mais rápido e mais eficaz, quando comparado com a ‘mão de obra’ humana, uma vez que é possível executar duas inspeções em simultâneo. Este facto é demonstrado dado que, enquanto as plataformas de inspeção estiverem ocupadas, o robô capta um terceiro prato proveniente do tapete de entrada, rentabilizando ao máximo o tempo do processo, e permitindo uma melhor gestão dos recursos humanos.

Contudo, a identificação de defeitos cerâmicos é um desafio para a visão artificial, visto que o ambiente na indústria cerâmica poderá não ser o mais propício para a utilização desta tecnologia por apresentar muitas poeiras, que obstruem a realização do processo, requerendo a manutenção do protótipo com mais frequência. Por outro lado, a própria superfície dos cerâmicos apresenta brilho que pode refletir contra as lentes das câmaras de inspeção, incapacitando-as de adquirir uma imagem nítida.

Comparando o sistema de inspeção com os existentes, este apresenta uma grande vantagem sobre os outros sistemas de inspeções, tendo em conta que como a manipulação é efetuada por um braço robótico não será difícil adaptar a novos tipos de louça, além disso para peças que demorem bastante tempo a serem analisadas este sistema pode ter diversos sistemas de inspeção ou até mesmo desenvolver sistema de inspeções com ambientes específicos para tipos de louça diferentes, o que proporciona um diversificado número de aplicações.

Num futuro projeto, a implementação de inteligência artificial no sistema de visão nos processos de vidragem permitirá identificar o centro das peças mais complexas, diminuindo o

tempo de processamento de imagem. Por outro lado, os sistemas de inspeção contribuirão para a detecção de novos defeitos, garantindo uma taxa de erro praticamente nula, o que irá aumentar a fiabilidade e responder às exigências da indústria 4.0.

Conclui-se ainda que os simuladores são bastante benéficos para a indústria, garantindo o estudo de diversos projetos, pois é possível programar vários robôs e adequar/alterar os seus componentes sem custos, sendo completamente indispensável para indústrias no ramo da robótica.

REFERÊNCIAS

- Admin. (2019). What is Industrial Automation & Types of Industrial Automation. *Concept Systems Inc.* <https://conceptsystemsinc.com/what-is-industrial-automation-types-of-industrial-automation/>
- Antonsen, T. M. (2022,). Sequential Function Chart (SFC) Programming for Beginners. *PLC Programming Courses for Beginners | RealPars.* <https://realpars.com/sequential-function-chart/>
- Kamarul Bahrin, M. A. K., Othman, M. F., Azli, N. H. N., & Talib, M. F. (2016). Industry 4.0: A review on Industrial Automation and robotic. *Jurnal Teknologi*, 708(6–13). <https://doi.org/10.11113/jt.v78.9285>
- Beckhoff Automation, G. (2021). *TwinCAT 3 PLC*. Beckhoff Automation. <https://www.beckhoff.com/en-us/products/automation/twincat/tc1xxx-twincat-3-base/tc1200.html>
- Budimir, M. (2019). *What are Instruction Lists (ILs) for PLC programming?* <https://www.motioncontroltips.com/instruction-lists-ils-plc-programming/>
- Demir, K., & Cicibaş, H. (2017). *Industry 5.0 And A Critique Of Industry 4.0*.
- Disner, E., & Cipullo, G. (2021). *Indústria 5.0: Conheça o novo conceito de produção*. Korp ERP. <https://www.korp.com.br/industria-5-0-conheca-o-novo-conceito-de-producao/>
- Duranso, G. (2021). Basics of Structured Text (ST) Programming | Examples & Applications. *PLC Programming Courses for Beginners | RealPars.* <https://realpars.com/structured-text/>
- Engelberger, J. F. (1979). Robotics in 1984. *Industrial Robô: An International Journal*, 6(3), 115–119. <https://doi.org/10.1108/eb004755>
- Epson (n.d.). Obtido 5 de setembro de 2022, de <https://www.epson.pt/produtos/robôs/s%C3%A9rie-scara-t/epson-scara-t3-b401s/p/35895>
- Francisco, A. (2015). *Autómatos Programáveis (5º)*. ETEP - Edições Técnicas e Profissionais.

- Gasparetto, A., & Scalera, L. (2019). A Brief History of Industrial Robotics in the 20th Century. *Advances in Historical Studies*, 8(1), Art. 1. <https://doi.org/10.4236/ahs.2019.81002>
- Godoi, L. (2022). *Automa o de Processos Industriais*. https://www.academia.edu/37215062/Automa_o_de_Processos_Industriais
- Gruver, W. A., Soroka, B. I., Craig, J. J., & Turner, T. L. (1984). Industrial robot programming languages: A comparative evaluation. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-14(4), 565–570. <https://doi.org/10.1109/TSMC.1984.6313327>
- Hammari, J. (2021). *From Industry 3.0 to Industry 4.0—Practice-Oriented Executive Summary | LinkedIn*. <https://www.linkedin.com/pulse/from-industry-30-40-practice-oriented-executive-summary-hammari/>
- Hitomi, K. (1994). Automation—Its concept and a short history. *Technovation*, 14(2), 121–128. [https://doi.org/10.1016/0166-4972\(94\)90101-5](https://doi.org/10.1016/0166-4972(94)90101-5)
- Knapp, E. (2011). Chapter 5—How Industrial Networks Operate. E. Knapp (Ed.), *Industrial Network Security* (pp. 89–110). Syngress. <https://doi.org/10.1016/B978-1-59749-645-2.00005-7>
- KUKA KR C4. (n.d.). KUKA AG. Obtido 25 de outubro de 2022, de <https://www.kuka.com/pt-pt/produtos-servi%c3%a7os/sistemas-de-rob%c3%b4/unidades-de-comando-de-rob%c3%b4/kr-c4>
- Lima, I. T. S. de, Santos, P. V. S. dos, Costa, M. A. da, Santos, P. B. dos, Freitas, T. R. de, Silva, P. V. da, & Lima, M. M. (2017). Automação Como Processo do Aumento da Produtividade. *Revista de Trabalhos Acadêmicos - Universo Recife*, 4(2), Art. 2. <http://revista.universo.edu.br/index.php?journal=1UNICARECIFE2&page=article&op=view&path%5B%5D=4480>
- Magalhaes, P., & Ferreira, N. (2022). Inspection Application in an Industrial Environment with Collaborative Robots. *Automation*, 3(2), Art. 2. <https://doi.org/10.3390/automation3020013>
- Eletronor. (2019). *A Aplicação de Robôs colaborativos Nas Diferentes Indústrias*. Eletronor, from <https://www.eletronor.com.br/blog/aplicacao-de-robos-colaborativos/>

- Miranda, L. P., Adriano. (2020). *A história de uma caneca feita por robôs*. PÚBLICO. <https://www.publico.pt/2020/12/12/economia/noticia/historia-caneca-robos-1942615>
- Nascimento, V. (2018). Automação industrial. *Automação Industrial*. https://www.academia.edu/41607356/Automa%C3%A7%C3%A3o_industrial
- Nunes, E. (2015). *Tear Mecânico*. <http://invencoesseculoxix.blogspot.com/2015/05/tear-mecanico.html>
- Paulo, M. (2022). *A automatização de processos: Benefícios e desvantagens*. PRIMAVERA BSS. <https://pt.primaverabss.com/pt/blog/automatizacao-de-processos/>
- CTCV. (n.d.). *Quem Somos*. CTCV. Obtido 25 de outubro de 2022, de <https://www.ctcv.pt/quemsomos.html>
- Riku, P. (2006). *Machine Vision Introduction2*. 2 web. https://www.academia.edu/38693358/Machine_Vision_Introduction2_2_web
- Robôs industriais*. (n. d.). KUKA AG. Obtido 11 de setembro de 2022, de <https://www.kuka.com/pt-pt/produtos-servi%C3%A7os/sistemas-de-rob%C3%B4s/rob%C3%B4s-industriais>
- Romanov, V. (2019). *Introduction to Function Block Programming in RSLogix 5000*. <https://www.solisplc.com/tutorials/function-block-programming>
- Sakurai, R., & Zuchi, J. D. (2018). As Revoluções Industriais até a Indústria 4.0. *Revista Interface Tecnológica*, 15(2), Art. 2. <https://doi.org/10.31510/inf.v15i2.386>
- Santos, V. M. F. (2021, September 6). *Robótica industrial*. Google Books. Retrieved November 5, 2022, from https://books.google.com/books/about/Rob%C3%B3tica_Industrial.html?id=e1ajzgEACAAJ
- Segovia, V. (2013). *History of Control History of PLC and DCS*. Automate. <https://www.automate.org/a3-content/joseph-engelberger-unimate>
- Sherwani, F., Asad, M. M., & Ibrahim, B. S. K. K. (2020). Collaborative Robots and Industrial Revolution 4.0 (IR 4.0). *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, 1–5. <https://doi.org/10.1109/ICETST49965.2020.9080724>

- Siemens. (n.d.). *Manufatura Digital*. Siemens Digital Industries Software. Obtido 25 de outubro de 2022, de <https://www.plm.automation.siemens.com/global/pt/our-story/glossary/digital-manufacturing/13157>
- Silveira, C. B. (2019). Os 6 Principais Tipos de Robôs Industriais. *Citisystems*. <https://www.citisystems.com.br/tipos-de-robos/>
- Sioma, A. (2020). Automated Control of Surface Defects on Ceramic Tiles Using 3D Image Analysis. *Materials*, 13(5), 1250. <https://doi.org/10.3390/ma13051250>
- Souza, N. de. (2021). Conhecendo o CLP. <https://medium.com/20-21/conhecendo-o-clp-7e04ca1a6648>
- Tantawi, K. H., Sokolov, A., & Tantawi, O. (2019). Advances in Industrial Robotics: From Industry 3.0 Automation to Industry 4.0 Collaboration. *2019 4th Technology Innovation Management and Engineering Science International Conference (TIMES-iCON)*, 1–4. <https://doi.org/10.1109/TIMES-iCON47539.2019.9024658>
- Tecnicon. (2020). *4 exemplos práticos da adoção da Indústria 4.0 nas fábricas*. TECNICON. https://www.tecnicon.com.br/blog/476-4_exemplos_praticos_da_adocao_da_industria_4_0_nas_fabricas
- Veit, D. R. (2018). *Impactos da manufatura aditiva nos sistemas produtivos e suas repercussões nos critérios competitivos*. <http://www.repositorio.jesuita.org.br/handle/UNISINOS/6927>
- Vinitha, K., Ambrose Prabhu, R., Bhaskar, R., & Hariharan, R. (2020). Review on industrial mathematics and materials at Industry 1.0 to Industry 4.0. *Materials Today: Proceedings*, 33, 3956–3960. <https://doi.org/10.1016/j.matpr.2020.06.331>
- What is a PLC? (2020). Inductive Automation. <http://www.inductiveautomation.com/resources/article/what-is-a-PLC>
- Xu, X., Lu, Y., Vogel-Heuser, B., & Wang, L. (2021). Industry 4.0 and Industry 5.0 — Inception, conception and perception. *Journal of Manufacturing Systems*, 61, 530–535. <https://doi.org/10.1016/j.jmsy.2021.10.006>
- Yaskawa. (n.d.). *GP180-120*. Obtido 30 de outubro de 2022, de https://www.yaskawa.eu.com/products/robos/handling-mounting/productdetail/product/gp180-120_705

Zimm, M. (2020). Different Types of Industrial Automation Systems. *One Pull Wire*.
<https://onepullwire.com/news/types-industrial-automated-systems/>

ANEXOS

Índice de anexos

Anexo I. Guia Yaskawa.....	109
Consola Yaskawa	109
Modos de Funcionamento	111
Nível de segurança.....	112
Coordenadas de eixo a eixo	112
Coordenada base.....	113
Coordenadas da ferramenta	113
Coordenadas de utilizador	114
Menus	114
Teclas	115
Gestão de programas	115
Selecionar um programa:.....	115
Criar um programa:	115
Renomear um programa:	115
Copiar um programa:.....	116
Apagar um programa:.....	116
Instruções:.....	116
Criação de pontos:	116
Modificação de um ponto de programa:.....	117
Apagado de um ponto de programa:.....	117
Nível de posição:	117
Lançamento de um programa	117
Manual: Modo TEACH.....	117
Automático: Modo PLAY	118
Controlo de Entradas / Saídas.....	118
Programação:	119
Operadores.....	120
Anexo II. Programa Arduino	122

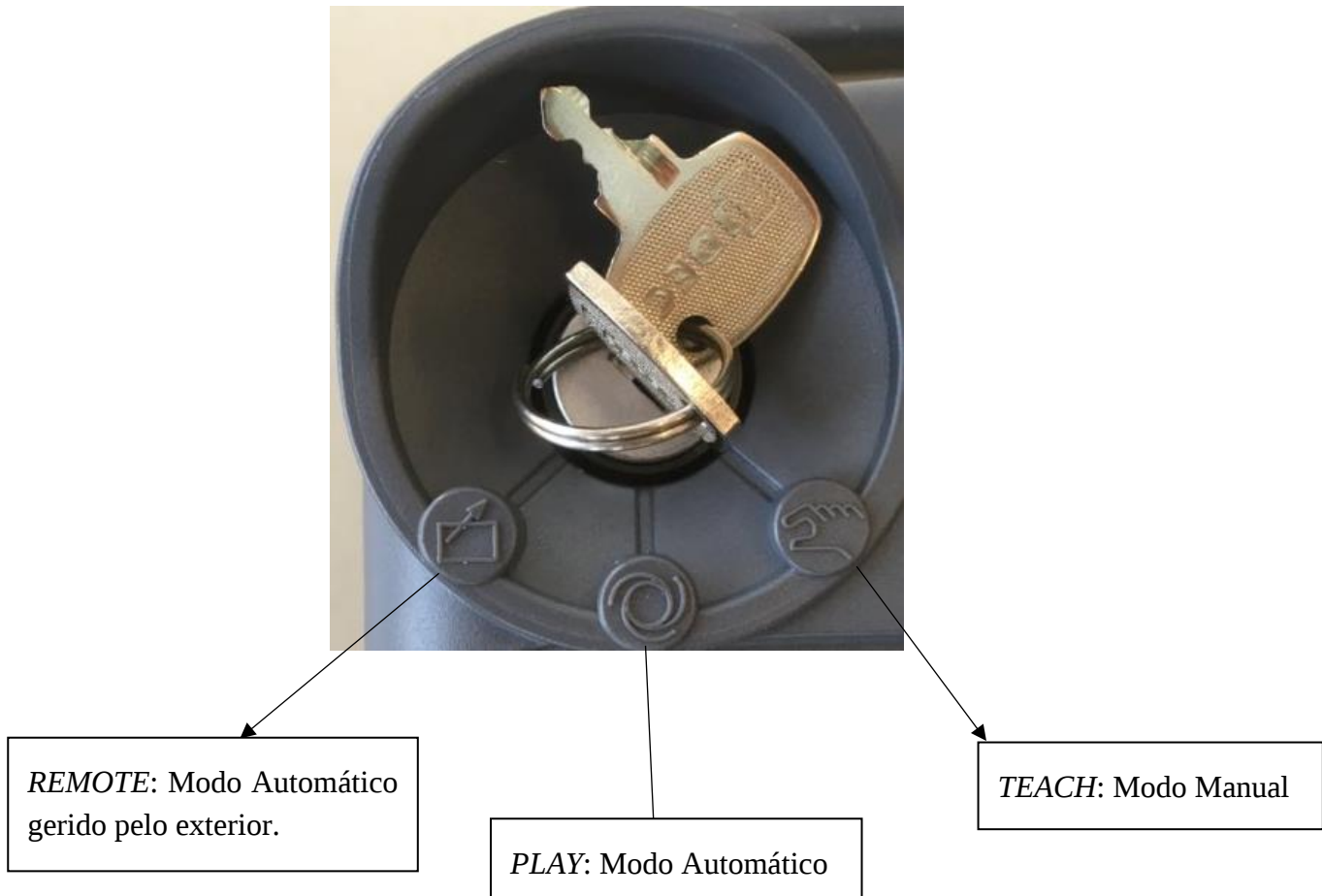
Anexo I. Guia Yaskawa

Consola Yaskawa





Modos de Funcionamento

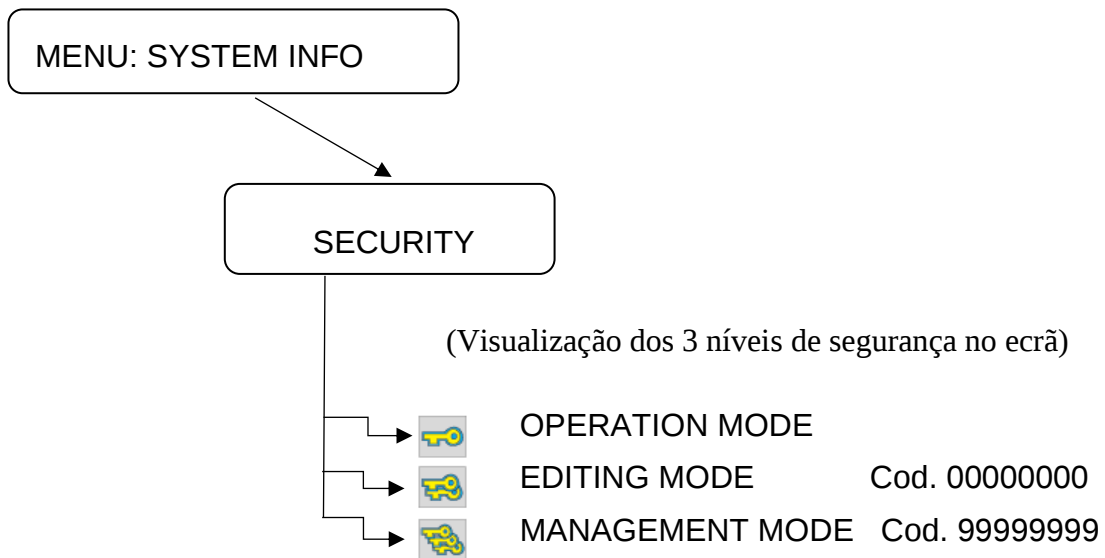


TEACH: O modo manual serve para criar programa ou editar programas existentes.

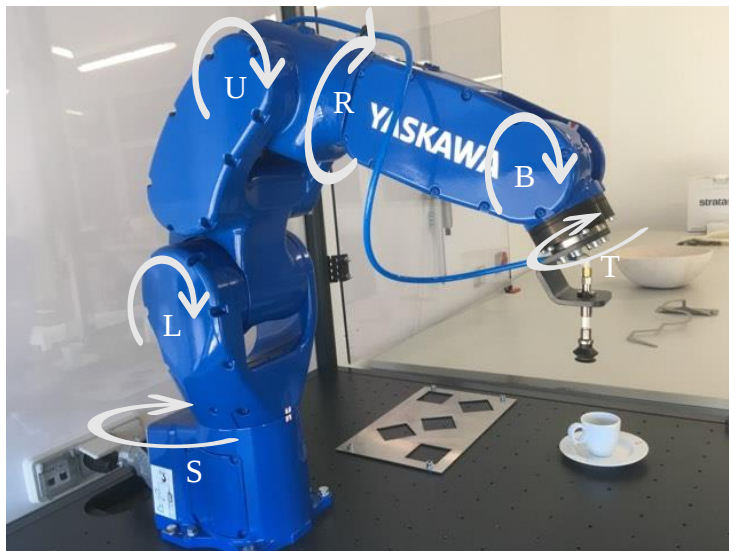
PLAY: O modo automático serve para correr os programas existentes não sendo possível alterar os mesmos.

REMOTE: Modo automático gerido pelo exterior tem como função a mesma que o modo automático, mas este é controlado externamente não sendo necessária o uso da consola para começar/parar/mudar os programas criados.

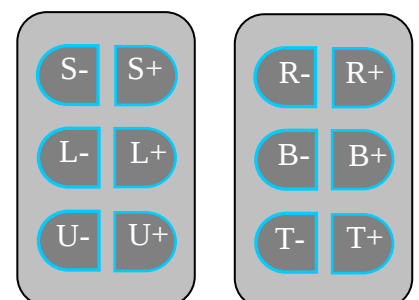
Nível de segurança



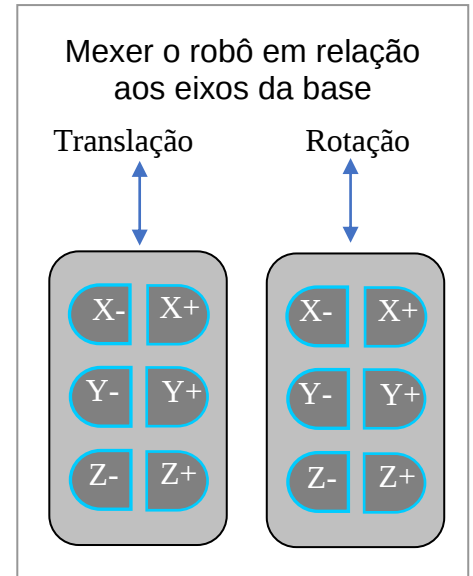
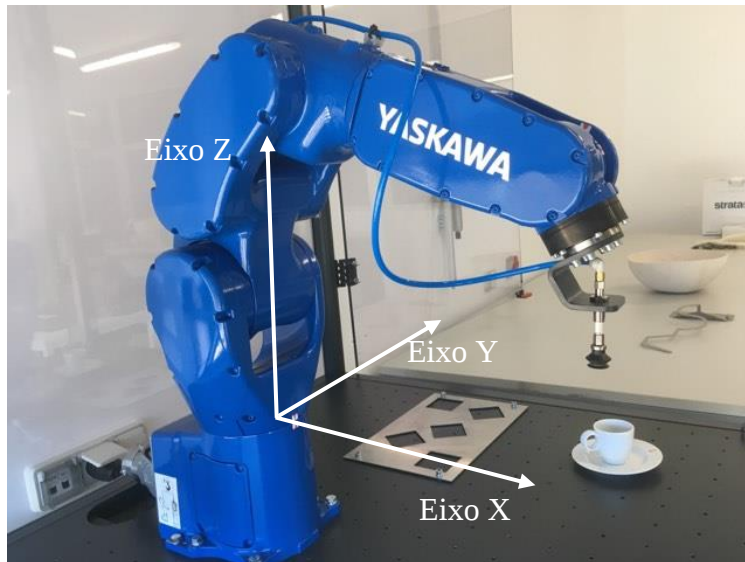
Coordenadas de eixo a eixo



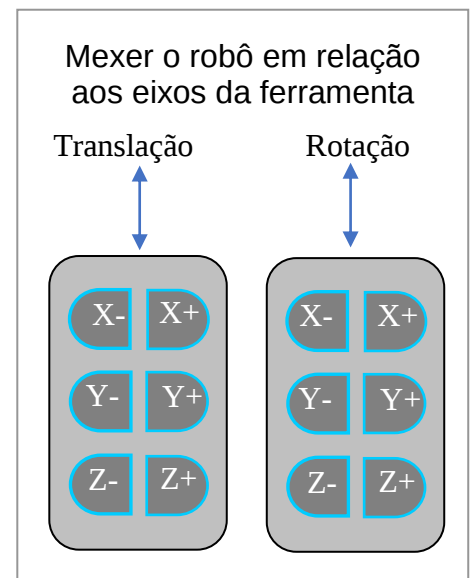
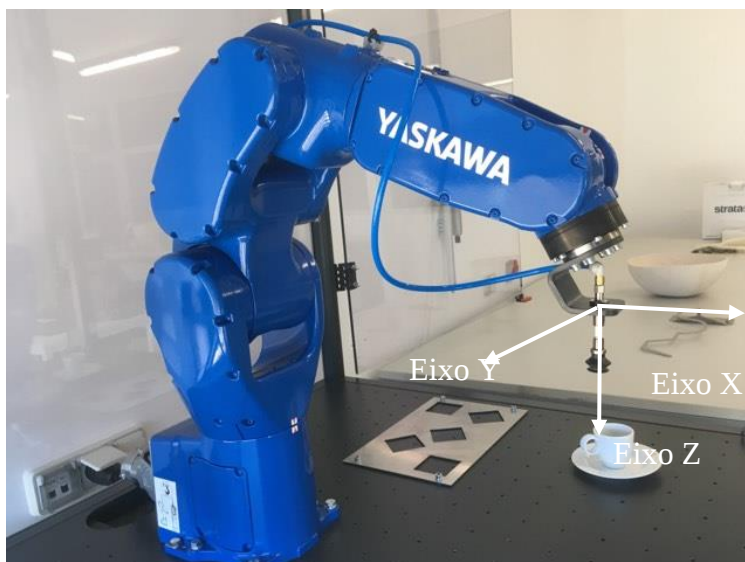
Mexer os 6 eixos do robô independentemente



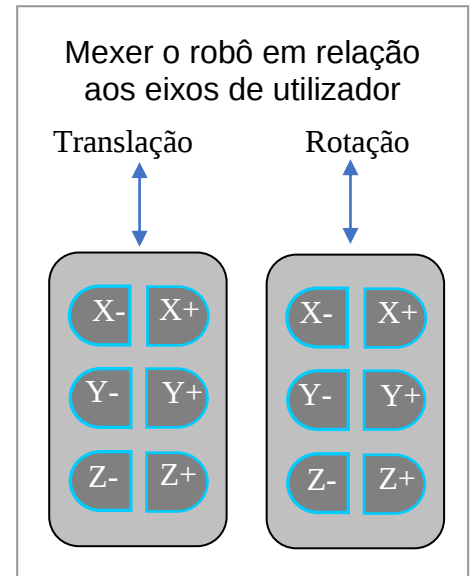
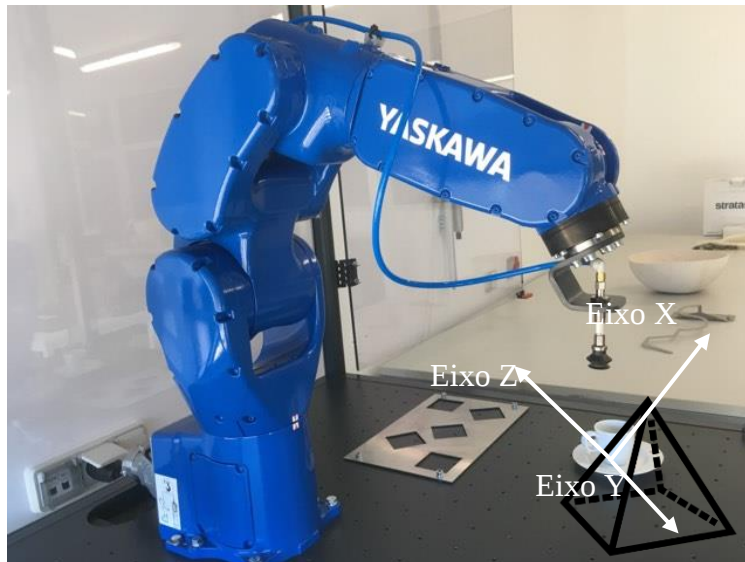
Coordenada base



Coordenadas da ferramenta



Coordenadas de utilizador



Menus

Modo de segurança



PROGRAMA	: Gestão e edição de programas	
SOLD. ARCO	: Aplicação do robô	
VARIABLE	: Gestão e edição de variáveis	
ENT/SAL	: Gestão de Entradas/Saídas	
ROBOT	: Ficheiros e condições do robô	
INFORM. SISTEMA	: Informação do sistema	
FD/CF	: Cópia de segurança	
PARAMETRO	: Parâmetros do robô	
AJUSTE	: Ajustes do robô	
VISUAL. SISTEMA	: Aparência do ecrã	

Teclas

	: Seleção de um Robô	
	: Seleção de um eixo externo	
	: Seleção de coordenadas	
	: Seleção da velocidade em manual	
		
	: Aparece/desaparece a barra do Menu principal	
	: Seleção de área de ecrã	
	: Página seguinte	
	: Acesso direto	
	: Deslocamentos de cursor	
	: Seleção dos menus	
	: Validação	
	: Acesso a funções auxiliares	

Gestão de Programas

Selecionar um programa: (Nível de segurança requerido - )

Para Selecionar um programa seleciona no menu “JOB”, “SELECT JOB”, onde abre uma lista de programas existente na consola. Selecionar o programa pretendido.

Criar um programa: (Nível de segurança requerido - )

Para a criação de um programa no menu “JOB” selecionar “CREATE NEW JOB” introduzir o respetivo no e executar

Renomear um programa: (Nível de segurança requerido - )

Para renomear um programa seleciona-se no menu “JOB”, “SELECT JOB”, onde abre uma lista de programas existente na consola. Selecionar o programa pretendido e escolher a opção “RENAME JOB” preencher com o nome pretendido e selecionar “EXECUTE”

Copiar um programa: (Nível de segurança requerido -)

Para copiar um programa seleciona-se no menu “JOB”, “SELECT JOB”, onde abre uma lista de programas existente na consola. Selecionar o programa pretendido e escolher a opção “COPY JOB”.

Apagar um programa: (Nível de segurança requerido -)

Para copiar um programa seleciona-se no menu “JOB”, “SELECT JOB”, onde abre uma lista de programas existente na consola. Selecionar o programa pretendido e escolher a opção “DELETE”.

Instruções:

MOVJ VJ=100 %: Movimento Joint ou Articular para se deslocar de um ponto para outro em que seja necessário fazer movimentos sem interessar a trajetória, os eixos chegam todos ao seu valor no final do movimento podendo alterar os valores destes mesmos garantindo maior velocidade e maior liberdade de movimento.

MOVL V=9000 cm/min: Movimento Linear é usado quando para entre dois pontos o robô executa uma trajetória linear garantido sempre a mesma orientação da garra.

MOVC V=9000 cm/min; Movimento Circular é usado para fazer um movimento circular, para isso precisa de ter 3 pontos, de modo a fazer uma aproximação de um arco com esses pontos.

MOVS V=9000 cm/min: Movimento Spline / Elíptico pode ser usado para fazer um movimento circular dependendo da posição, ou seja, esta opção vai efetuar a passagem pelos pontos selecionados, mas garantido um movimento uniforme sem mudanças bruscas de direção.

Criação de pontos:

Deslocar o robô à posição desejada situar o cursor na linha anterior a desejada selecionar o tipo de interpolação com o botão *motion type* escolher a velocidade desejada para a movimentação para o ponto, para inserir o ponto antes de carregar no *enter* tem de ser pressionado o botão *insert*.

Modificação de um ponto de programa:

Situar o cursor no número de linha a modificar posicionar o robô no novo ponto pressionar o botão *modify* e de seguida o botão *enter*.

Apagado de um ponto de programa:

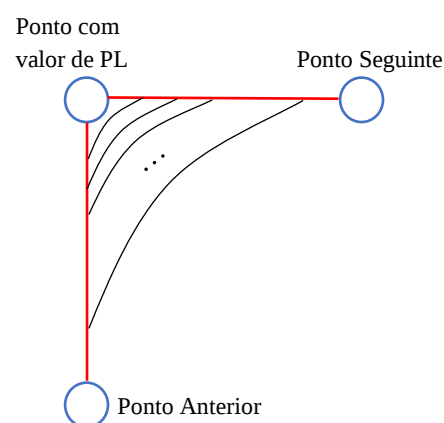
Situar o cursor na linha que se deseja apaga levar o robô para esse ponto físico com a tecla FWD, pressionar delete e de seguida *enter*.

Nível de posição:

MOVJ VJ=75% PL0 a PL8

PL0: Assegura a chegada ao ponto

PL1 a PL8: Arredonda a trajetória nunca chegando ao ponto, quanto maior o valor de PL mais afastada do ponto será a trajetória como está representado na imagem seguinte.



Lançamento de um programa

Selecionar um programa da lista de programas

Manual: Modo TEACH

Linha a linha:

Pressionando o botão FWD executa apenas as instruções de movimento, caso seja pressionado ao mesmo tempo o botão INTERLOCK + FWD executa todas as instruções o botão BWD executa a instrução no sentido inverso do programa.

Em contínuo:

Para correr o programa em modo contínuo apenas tem-se que manter premido os botões *INTERLOCK + TEST START*.

Automático: Modo *PLAY*

Para iniciar o programa em modo automático primeiramente muda-se para modo automático como explicado acima e depois apenas prime-se o botão *START* (botão ilustrado na figura), quando o programa começar a luz do botão permanecerá acesa durante a duração do mesmo.



Para parar a execução apenas é preciso pressionar o botão *HOLD* (botão ilustrado na figura).



Nota: Em ambos os casos, o curso do programa fica condicionado pelos três tipos de operação possíveis:

Step: Passo a Passo



Cycle: Só 1 ciclo



Auto: Ciclo infinito



Controlo de Entradas / Saídas

Para visualizar as Entradas/Saídas, escolher o menu “*IN/OUT*” onde poderá escolher o tipo de entrada e saída desejado:

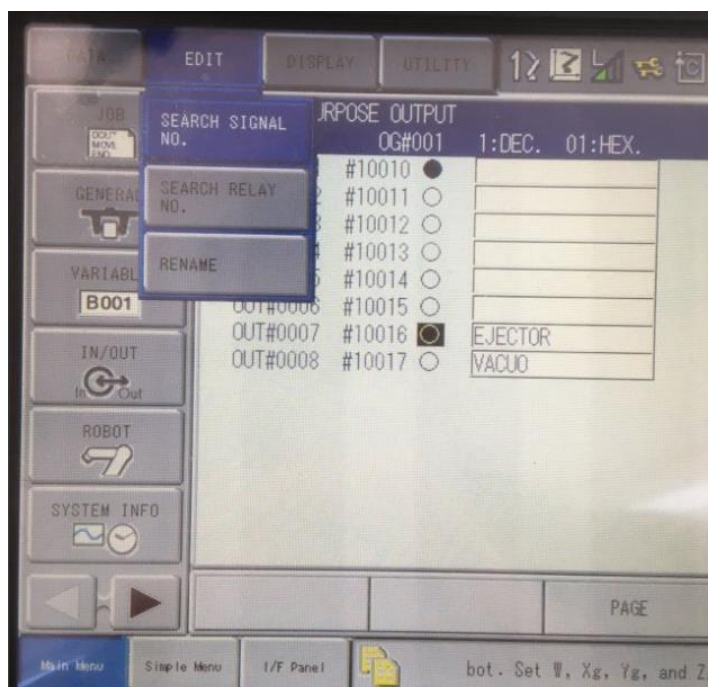
Entrada externa - “*EXTERNAL INPUT*”

Saída externa – “*EXTERNAL OUTPUT*”

Entrada universal – “*GENERAL PURPOSE INPUT*”

Saída universal – “*GENERAL PURPOSE OUTPUT*”

Para encontrar uma Entrada ou Saída seleccionar o menu “*EDIT*”, “*SEARCH SIGNAL NO*” ou “*SEARCH RELAY NO*”



Para forçar a saída apenas selecionar a saída ou entrada desejada e pressionar o botão “INTERLOCK” e o botão “SELECT”

Programação:

Entradas e saídas:

Para adicionar uma instrução de entradas/saídas pressionar o botão “*INFORM LIST*”, este quando está ativo aciona um led nesse mesmo botão, no menu aberto pela ação anterior selecionar a instrução desejada.

Menu:

6. DOUT

- OT#(7) ON/OFF - Acionar a saída correspondente (saída 7)
- OG#(1) 5 - (Grupo de 8 saídas)
- OGH#(3) B000 (Grupo de 4 saídas)

7. DIN

- B000 IN#(9), IG#(8), IGH#(12), SIN#(2) – Guarda o estado das entradas para a variável B000
- B00 OT#(4), OG#(7), OGH#(5), SOUT#(23) - Guarda o estado das saídas para a variável B000

8. WAIT

- IN#(9), IG#(8), IGH#(12), SIN#(2) = B001 T = 0,01 - Aguardar a entrada ou o tempo
- OT#(4), OG#(7), OGH#(5), SOUT#(23) = B002 - Aguardar saídas
- BOOO = 5 - Aguardar o valor de uma variável

9. PULSE

- OT#(15) T= 0,01 - Ativar uma saída durante 0,01s

10. AOUT

- AO#(3) 5,00 Envio de uma tensão entre $\pm 14,0$ v.

Operadores



ARITH

ADD: Soma

SUB: Subtração

MUL: Multiplicação

DIV: Divisão

INC: Incremento (+1)

DEC: Decremento (-1)

AND: AND Lógica

OR: OR Lógica

NOT: NOT Lógica

XOR: XOR Lógica

SIN: Seno

COS: Cosseno

ATAN: Arc tangente

SQRT: Raiz quadrada

MULMAT: Multiplicação matricial

INVMAT: Inversa de matriz

CLEAR <variável> nome / ALL: Pôr a zero várias variáveis

SET: Cópia de uma variável

SETE: Carregar um valor num dos dados de uma variável P

GETE: Recuperar um dado de uma variável P

GETS: Recuperar um dado do sistema

\$PX000 = Posição atual do robô em eixos de robô independentes (S, L, U, R, B, T)

\$PX001 = Posição atual do robô em coordenadas cartesianas (X, Y, Z, Rx, Ry, Rz,)

GETPOS STEP#(2): Recuperar a posição de um passo

CNVRT: Converter uma variável em uma coordenada

MFRAME: Criar uma coordenada de utilizador

SETREG: Carregar um valor a um registo

GETREG: Recuperar um valor de um registo

SETFILE: Altera os dados do ficheiro

GETFILE: Adquire dados do arquivo

Anexo II. Programa Arduino

```
#include <AccelStepper.h>
// Define stepper motor connections and motor interface type. Motor
// interface type must be set to 1 when using a driver:
#define dirPin 5
#define stepPin 2
#define SENSOR 22 // define pint 2 for sensor
const int buttonPin = 46;
#define directioPin 6
#define SteppPin 3
const int botaoPin = 26;
int AUX = 1;
// Create a new instance of the AccelStepper class:
AccelStepper stepper = AccelStepper(AccelStepper::DRIVER, stepPin,
dirPin);
AccelStepper stepper2 = AccelStepper(AccelStepper::DRIVER, SteppPin,
directioPin);
void setup() {

    // Set the maximum speed in steps per second:
    stepper2.setMaxSpeed(1000);
    pinMode(botaoPin, INPUT);

    stepper.setMaxSpeed(1000);
    pinMode(buttonPin, INPUT);
    pinMode(SENSOR, INPUT_PULLUP); // define pin as Input sensor
}
void loop()
{
    stepper.setCurrentPosition(0);
    stepper2.setCurrentPosition(0);
    while(digitalRead(SENSOR) != 0){ // Encontrar a supreficie da mesa
        stepper.setSpeed(400);
        stepper.runSpeed();
    }
    while(digitalRead(SENSOR) == 0){ // Encontrar a abertura da mesa
        stepper.setSpeed(400);
        stepper.runSpeed();
    }

    while(1){

        if(digitalRead(buttonPin)==HIGH && AUX == 1){
            stepper.setCurrentPosition(0);
            while(stepper.currentPosition() != 1072)
            {
                stepper.setSpeed(150);
                stepper.runSpeed();
            }
            AUX++;
        }
        if(digitalRead(buttonPin)==HIGH && AUX == 2){
            while(stepper.currentPosition() != 2128)
            {
                stepper.setSpeed(150);
```

```

        stepper.runSpeed();
    }
    AUX++;
}
if(digitalRead(buttonPin)==HIGH && AUX == 3){
    while(stepper.currentPosition() != 3200)
    {
        stepper.setSpeed(150);
        stepper.runSpeed();

    }
    AUX=1;
}
while(digitalRead(botaoPin)==HIGH) {
    stepper2.setSpeed(300);
    stepper2.runSpeed();
}
}
}

```

Anexo III. Programação Robô Yaskawa

/JOB

//NAME APRESENSEMMESA1

//POS

///NPOS 127,0,0,0,0

///TOOL 0

///POSTYPE PULSE

///PULSE

C00000=-16,10778,-26571,760,34567,81950

C00001=-95666,8064,-28023,5667,33833,80117

C00002=-95666,59571,-18804,3914,67971,82252

C00003=-95666,72620,-10689,3910,71036,82401

C00004=-95666,62503,-17185,3911,68859,82295

C00005=-95666,7578,-27869,5739,33282,80064

C00006=-43297,7578,-27870,5740,33283,80064

C00007=-43292,7577,-27870,5735,33278,162200

C00008=-39073,27668,-2977,7887,-24097,160698

C00009=-39073,41265,-4485,16042,-11562,155969

C00010=-39073,49380,-2769,27689,-6900,149561

C00011=-39073,40597,-4558,15416,-12027,156320

C00012=-36706,6238,-28065,-1547,24457,165596

C00013=-41987,18135,-15138,7237,22064,161876

C00014=-32069,20477,-12321,-9852,21449,169231

C00015=-36706,6238,-28065,-1547,24457,165596

C00016=-41987,18135,-15138,7237,22064,161876

C00017=-32069,20477,-12321,-9852,21449,169231

C00018=-36706,6238,-28065,-1547,24457,165596

C00019=-41987,18135,-15138,7237,22064,161876
C00020=-32069,20477,-12321,-9852,21449,169231
C00021=-36706,6238,-28065,-1547,24457,165596
C00022=-1097,-8335,-35661,1067,26714,162415
C00023=-1071,37224,-36387,525,65772,164059
C00024=14497,46667,-24343,51560,65416,160587
C00025=33282,48872,-17256,36825,60052,159311
C00026=33282,51587,-16188,36631,60953,159824
C00027=33282,56790,-13852,36348,62484,160684
C00028=10625,55199,-21361,54447,67641,162739
C00029=-16,10778,-26571,760,34567,81950
C00030=-126072,11345,-17174,43042,35840,64027
C00031=-126072,65246,-8311,29157,63718,80058
C00032=-126072,72727,-3383,29044,64951,80559
C00033=-126072,65729,-8018,29145,63818,80101
C00034=-126072,14577,-18761,39890,38624,66572
C00035=-43297,7578,-27870,5740,33283,80064
C00036=-43292,7577,-27870,5735,33278,162200
C00037=-39073,27668,-2977,7887,-24097,160698
C00038=-39073,41265,-4485,16042,-11562,155969
C00039=-39073,49380,-2769,27689,-6900,149561
C00040=-39073,40597,-4558,15416,-12027,156320
C00041=-36706,6238,-28066,-1547,24457,165596
C00042=-41987,18135,-15138,7237,22064,161876
C00043=-32069,20477,-12321,-9852,21449,169231
C00044=-36706,6238,-28065,-1547,24457,165596
C00045=-41987,18135,-15138,7237,22064,161876

C00046=-32069,20477,-12321,-9852,21449,169231
C00047=-36706,6238,-28065,-1547,24457,165596
C00048=-41987,18135,-15138,7237,22064,161876
C00049=-32069,20477,-12321,-9852,21449,169231
C00050=-36706,6238,-28065,-1547,24457,165596
C00051=-1097,-8335,-35661,1067,26714,162415
C00052=-1071,37224,-36387,525,65772,164059
C00053=14497,46667,-24343,51560,65416,160587
C00054=33282,48872,-17256,36825,60052,159311
C00055=33282,51587,-16188,36631,60953,159824
C00056=33282,56790,-13852,36348,62484,160684
C00057=10625,55199,-21361,54447,67641,162739
C00058=-16,10778,-26571,760,34567,81950
C00059=-64506,13930,-22517,-29689,37690,93580
C00060=-64506,65396,-10729,-21755,65442,83415
C00061=-64506,72362,-6139,-21703,66694,83058
C00062=-64506,72362,-6139,-21703,66694,83058
C00063=-64506,65396,-10729,-21755,65442,83415
C00064=-64506,13930,-22517,-29689,37690,93580
C00065=-43292,7577,-27870,5735,33278,162200
C00066=-39073,27668,-2977,7887,-24097,160698
C00067=-39073,41265,-4485,16042,-11562,155969
C00068=-39073,49380,-2769,27689,-6900,149561
C00069=-39073,40597,-4558,15416,-12027,156320
C00070=-36706,6238,-28066,-1547,24457,165596
C00071=-41987,18135,-15138,7237,22064,161876
C00072=-32069,20477,-12321,-9852,21449,169231

C00073=-36706,6238,-28065,-1547,24457,165596
C00074=-41987,18135,-15138,7237,22064,161876
C00075=-32069,20477,-12321,-9852,21449,169231
C00076=-36706,6238,-28065,-1547,24457,165596
C00077=-41987,18135,-15138,7237,22064,161876
C00078=-32069,20477,-12321,-9852,21449,169231
C00079=-36706,6238,-28065,-1547,24457,165596
C00080=-1097,-8335,-35661,1067,26714,162415
C00081=-1071,37224,-36387,525,65772,164059
C00082=14497,46667,-24343,51560,65416,160587
C00083=33282,48872,-17256,36825,60052,159311
C00084=33282,51587,-16188,36631,60953,159824
C00085=33282,56790,-13852,36348,62484,160684
C00086=10625,55199,-21361,54447,67641,162739
C00087=-16,10778,-26571,760,34567,81950
C00088=10625,55199,-21361,54447,67641,162739
C00089=33282,56790,-13852,36348,62484,160684
C00090=33282,51587,-16188,36631,60953,159824
C00091=33282,48872,-17256,36825,60052,159311
C00092=14496,46666,-24344,51563,65419,160590
C00093=-16,10778,-26571,760,34567,81950
C00094=-95666,8064,-28023,5667,33833,80117
C00095=-95666,59571,-18804,3914,67971,82252
C00096=-95666,72620,-10689,3910,71036,82401
C00097=-95666,72620,-10689,3910,71036,82401
C00098=-95666,59571,-18804,3914,67971,82252
C00099=-95666,8064,-28023,5667,33833,80117

C00100=-16,10778,-26571,760,34567,81950
C00101=10625,55199,-21361,54447,67641,162739
C00102=33282,56790,-13852,36348,62484,160684
C00103=33282,51587,-16188,36631,60953,159824
C00104=33282,48872,-17256,36825,60052,159311
C00105=14496,46666,-24344,51563,65419,160590
C00106=-16,10778,-26571,760,34567,81950
C00107=-126072,11345,-17174,43042,35840,64027
C00108=-126072,65246,-8311,29157,63718,80058
C00109=-126072,72727,-3383,29044,64951,80559
C00110=-126072,72727,-3383,29044,64951,80559
C00111=-126072,65246,-8311,29157,63718,80058
C00112=-126072,11345,-17174,43042,35840,64027
C00113=-16,10778,-26571,760,34567,81950
C00114=10625,55199,-21361,54447,67641,162739
C00115=33282,56790,-13852,36348,62484,160684
C00116=33282,51587,-16188,36631,60953,159824
C00117=33282,48872,-17256,36825,60052,159311
C00118=14496,46666,-24344,51563,65419,160590
C00119=-16,10778,-26571,760,34567,81950
C00120=-64506,13930,-22517,-29689,37690,93580
C00121=-64506,65396,-10729,-21755,65442,83415
C00122=-64506,72362,-6139,-21703,66694,83058
C00123=-64506,72362,-6139,-21703,66694,83058
C00124=-64506,65396,-10729,-21755,65442,83415
C00125=-64506,13930,-22517,-29689,37690,93580
C00126=-16,10778,-26571,760,34567,81950

```
//INST  
  
///DATE 2022/11/04 14:31  
  
///ATTR SC,RW  
  
///GROUP1 RB1  
  
NOP  
  
DOUT OT#(4) ON  
  
MOVJ C00000 VJ=25.00  
  
MOVJ C00001 VJ=25.00  
  
MOVL C00002 V=250.0  
  
MOVL C00003 V=23.0  
  
DOUT OT#(8) ON  
  
DOUT OT#(7) OFF  
  
TIMER T=1.000  
  
MOVL C00004 V=23.0  
  
MOVL C00005 V=250.0  
  
MOVJ C00006 VJ=25.00  
  
MOVJ C00007 VJ=25.00  
  
MOVJ C00008 VJ=25.00  
  
MOVL C00009 V=93.0  
  
MOVL C00010 V=11.7  
  
DOUT OT#(5) ON  
  
TIMER T=3.000  
  
DOUT OT#(5) OFF  
  
MOVL C00011 V=11.7  
  
MOVJ C00012 VJ=25.00  
  
DOUT OT#(5) ON  
  
MOVC C00013
```

MOVC C00014
MOVC C00015
MOVC C00016
MOVC C00017
MOVC C00018
MOVC C00019
MOVC C00020
MOVC C00021
DOUT OT#(5) OFF
MOVL C00022 V=250.0
MOVJ C00023 VJ=25.00
MOVJ C00024 VJ=25.00
MOVL C00025 V=42.5
MOVL C00026 V=22.7
DOUT OT#(8) OFF
DOUT OT#(7) ON
TIMER T=0.100
DOUT OT#(7) OFF
MOVL C00027 V=22.7
MOVL C00028 V=83.3
MOVJ C00029 VJ=25.00
PULSE OT#(6)
DOUT OT#(4) ON
MOVJ C00030 VJ=25.00
MOVL C00031 V=250.0
MOVL C00032 V=23.0
DOUT OT#(8) ON

DOUT OT#(7) OFF

TIMER T=1.000

MOVL C00033 V=23.0

MOVL C00034 V=250.0

MOVJ C00035 VJ=25.00

MOVJ C00036 VJ=25.00

MOVJ C00037 VJ=25.00

MOVL C00038 V=250.0

MOVL C00039 V=23.0

DOUT OT#(5) ON

TIMER T=3.000

DOUT OT#(5) OFF

MOVL C00040 V=23.0

MOVJ C00041 VJ=25.00

DOUT OT#(5) ON

MOVC C00042

MOVC C00043

MOVC C00044

MOVC C00045

MOVC C00046

MOVC C00047

MOVC C00048

MOVC C00049

MOVC C00050

DOUT OT#(5) OFF

MOVL C00051 V=250.0

MOVJ C00052 VJ=25.00

MOVJ C00053 VJ=25.00
MOVL C00054 V=42.5
MOVL C00055 V=22.7
DOUT OT#(8) OFF
DOUT OT#(7) ON
TIMER T=0.100
DOUT OT#(7) OFF
MOVL C00056 V=22.7
MOVL C00057 V=83.3
MOVJ C00058 VJ=25.00
PULSE OT#(6)
MOVJ C00059 VJ=25.00
MOVL C00060 V=250.0
MOVL C00061 V=23.0
DOUT OT#(8) ON
DOUT OT#(7) OFF
TIMER T=1.000
MOVL C00062 V=23.0
MOVL C00063 V=250.0
MOVL C00064 V=250.0
MOVJ C00065 VJ=25.00
MOVJ C00066 VJ=25.00
MOVL C00067 V=250.0
MOVL C00068 V=23.0
DOUT OT#(5) ON
TIMER T=3.000
DOUT OT#(5) OFF

MOVL C00069 V=23.0

MOVJ C00070 VJ=25.00

DOUT OT#(5) ON

MOVC C00071

MOVC C00072

MOVC C00073

MOVC C00074

MOVC C00075

MOVC C00076

MOVC C00077

MOVC C00078

MOVC C00079

DOUT OT#(5) OFF

MOVL C00080 V=250.0

MOVJ C00081 VJ=25.00

MOVJ C00082 VJ=25.00

MOVL C00083 V=42.5

MOVL C00084 V=22.7

DOUT OT#(8) OFF

DOUT OT#(7) ON

TIMER T=0.100

DOUT OT#(7) OFF

MOVL C00085 V=22.7

MOVL C00086 V=83.3

MOVJ C00087 VJ=25.00

TIMER T=25.000

MOVL C00088 V=83.3

MOVL C00089 V=22.7

DOUT OT#(8) ON

DOUT OT#(7) OFF

TIMER T=1.000

MOVL C00090 V=22.7

MOVL C00091 V=42.5

MOVL C00092 V=42.5

MOVJ C00093 VJ=25.00

PULSE OT#(6)

MOVJ C00094 VJ=25.00

MOVL C00095 V=250.0

MOVL C00096 V=23.0

DOUT OT#(8) OFF

DOUT OT#(7) ON

TIMER T=0.100

DOUT OT#(7) OFF

MOVL C00097 V=23.0

MOVL C00098 V=250.0

MOVJ C00099 VJ=25.00

MOVJ C00100 VJ=25.00

MOVL C00101 V=83.3

MOVL C00102 V=22.7

DOUT OT#(8) ON

DOUT OT#(7) OFF

TIMER T=1.000

MOVL C00103 V=22.7

MOVL C00104 V=42.5

MOVL C00105 V=42.5

MOVJ C00106 VJ=25.00

PULSE OT#(6)

MOVJ C00107 VJ=25.00

MOVL C00108 V=250.0

MOVL C00109 V=23.0

DOUT OT#(8) OFF

DOUT OT#(7) ON

TIMER T=0.100

DOUT OT#(7) OFF

MOVL C00110 V=23.0

MOVL C00111 V=250.0

MOVJ C00112 VJ=25.00

MOVJ C00113 VJ=25.00

MOVL C00114 V=83.3

MOVL C00115 V=22.7

DOUT OT#(8) ON

DOUT OT#(7) OFF

TIMER T=1.000

MOVL C00116 V=22.7

MOVL C00117 V=42.5

MOVL C00118 V=42.5

MOVJ C00119 VJ=25.00

MOVJ C00120 VJ=25.00

MOVL C00121 V=250.0

MOVL C00122 V=23.0

DOUT OT#(8) OFF

DOUT OT#(7) ON

TIMER T=0.100

DOUT OT#(7) OFF

MOVL C00123 V=23.0

MOVL C00124 V=250.0

MOVJ C00125 VJ=25.00

MOVJ C00126 VJ=25.00

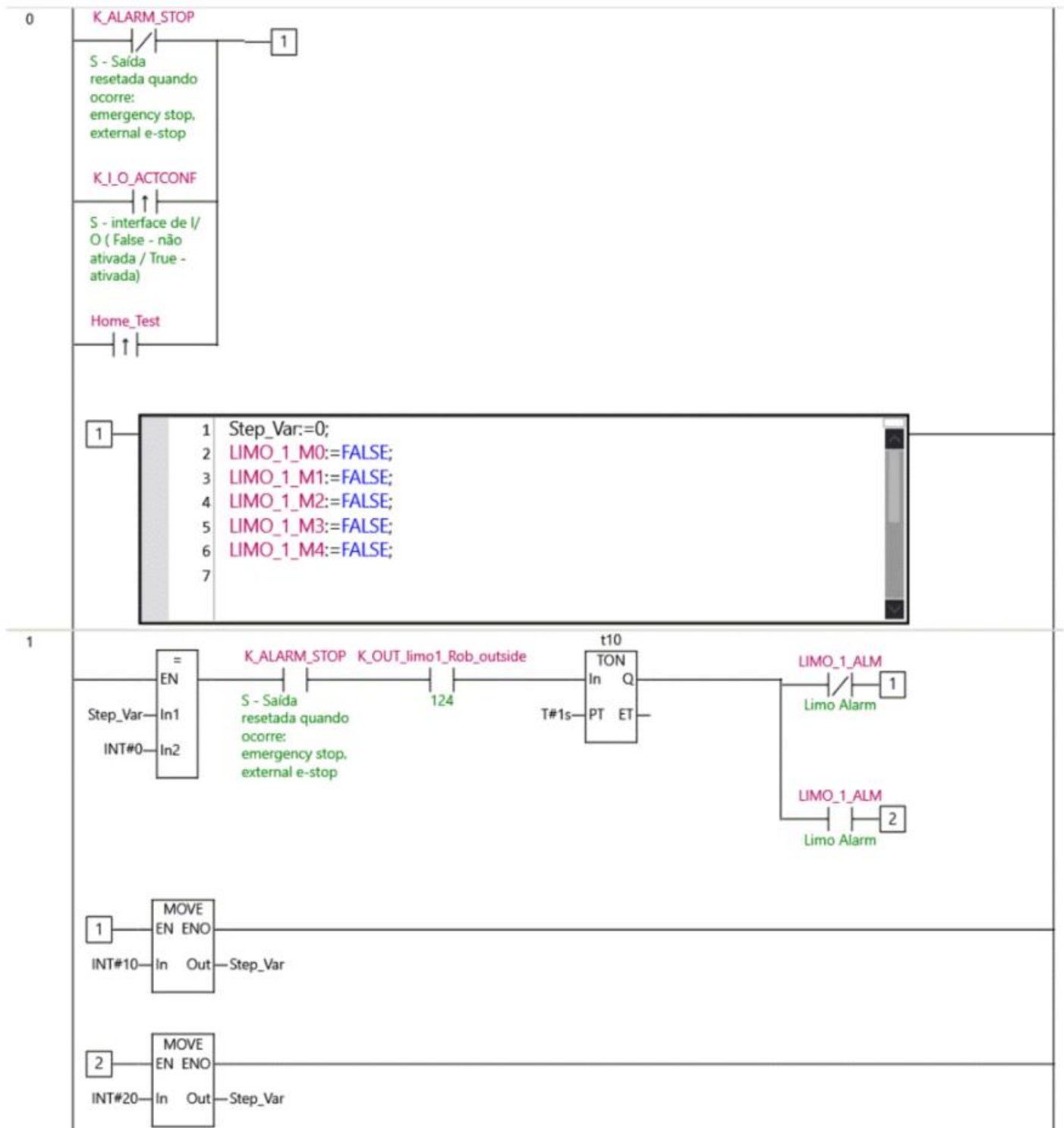
TIMER T=65.000

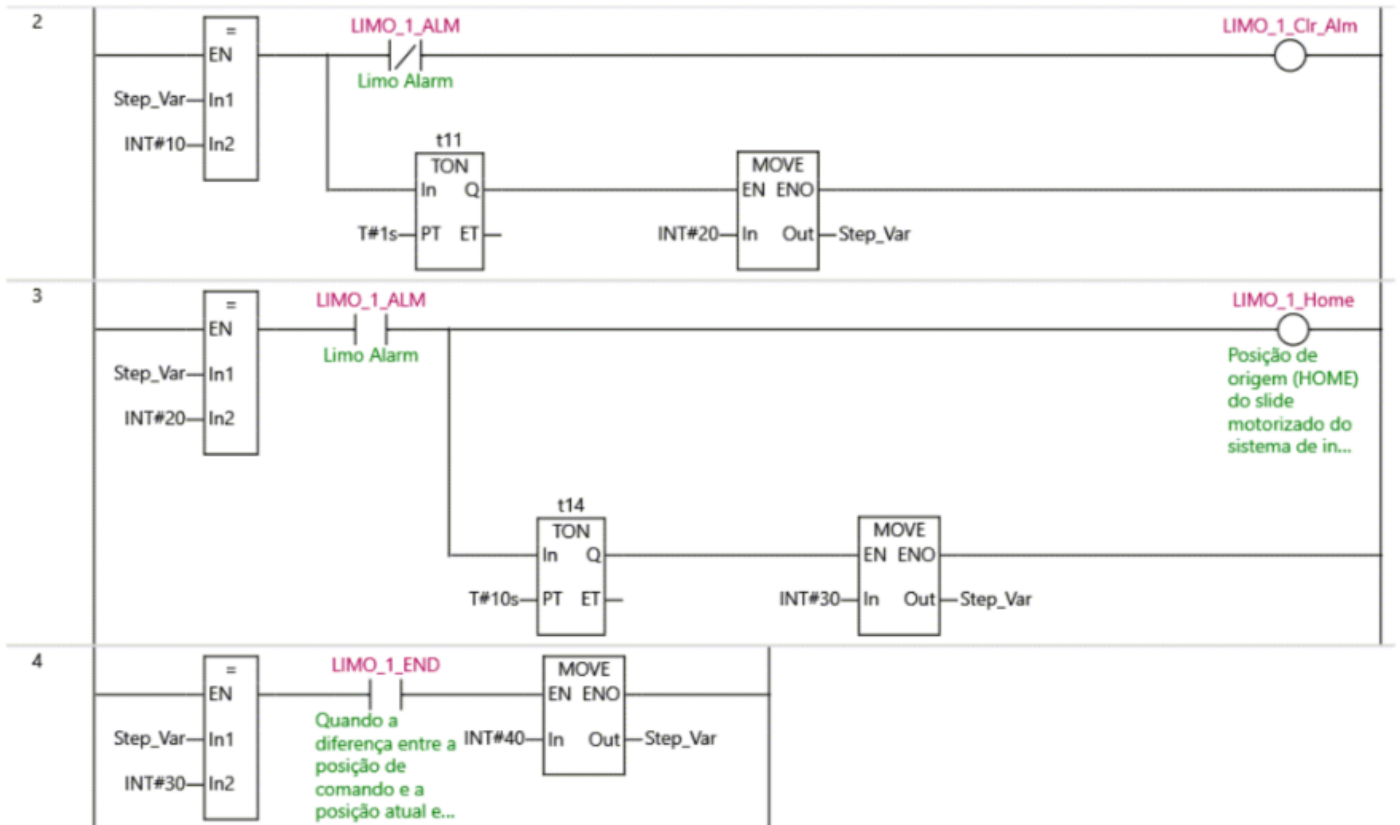
TIMER T=25.000

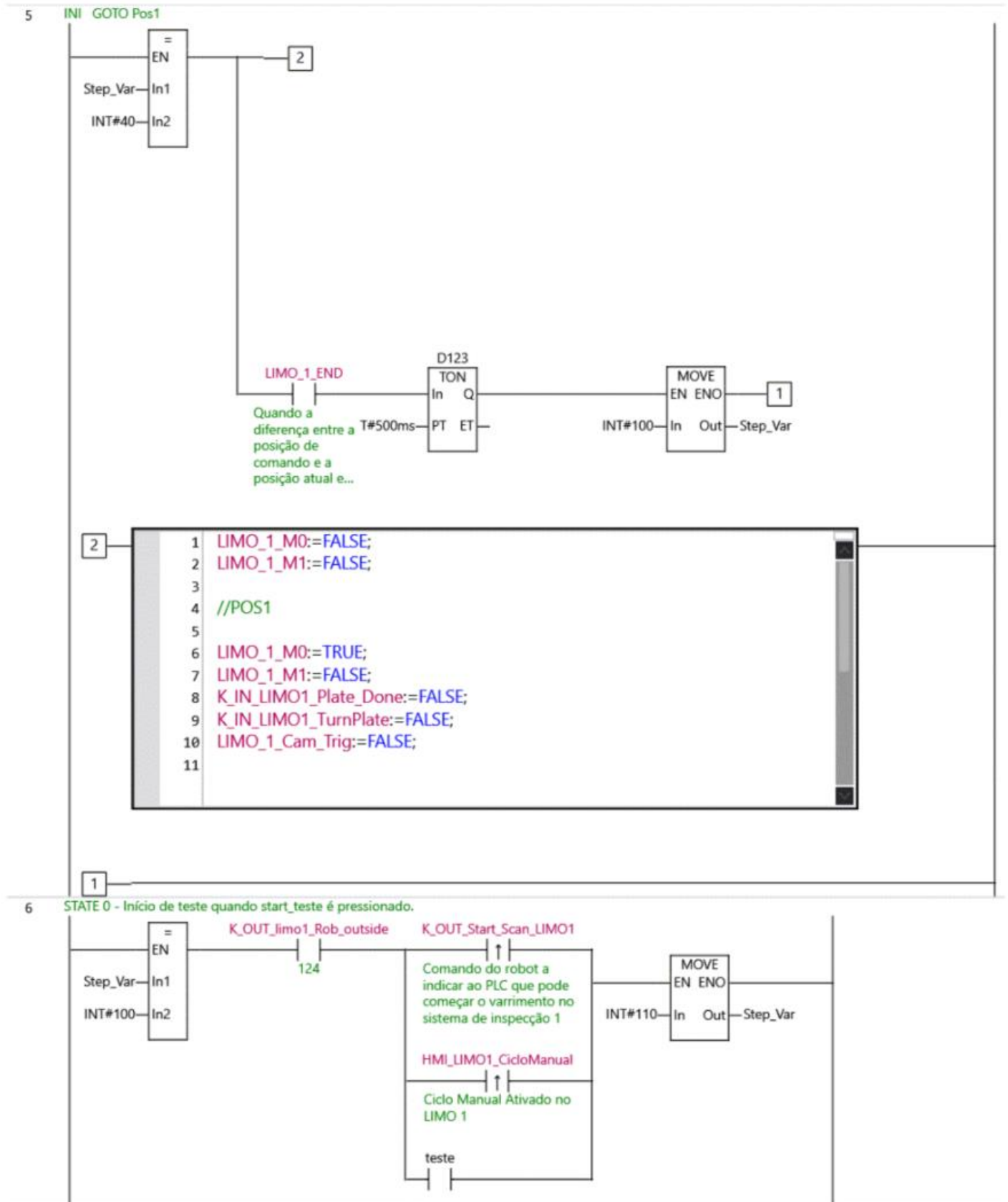
END

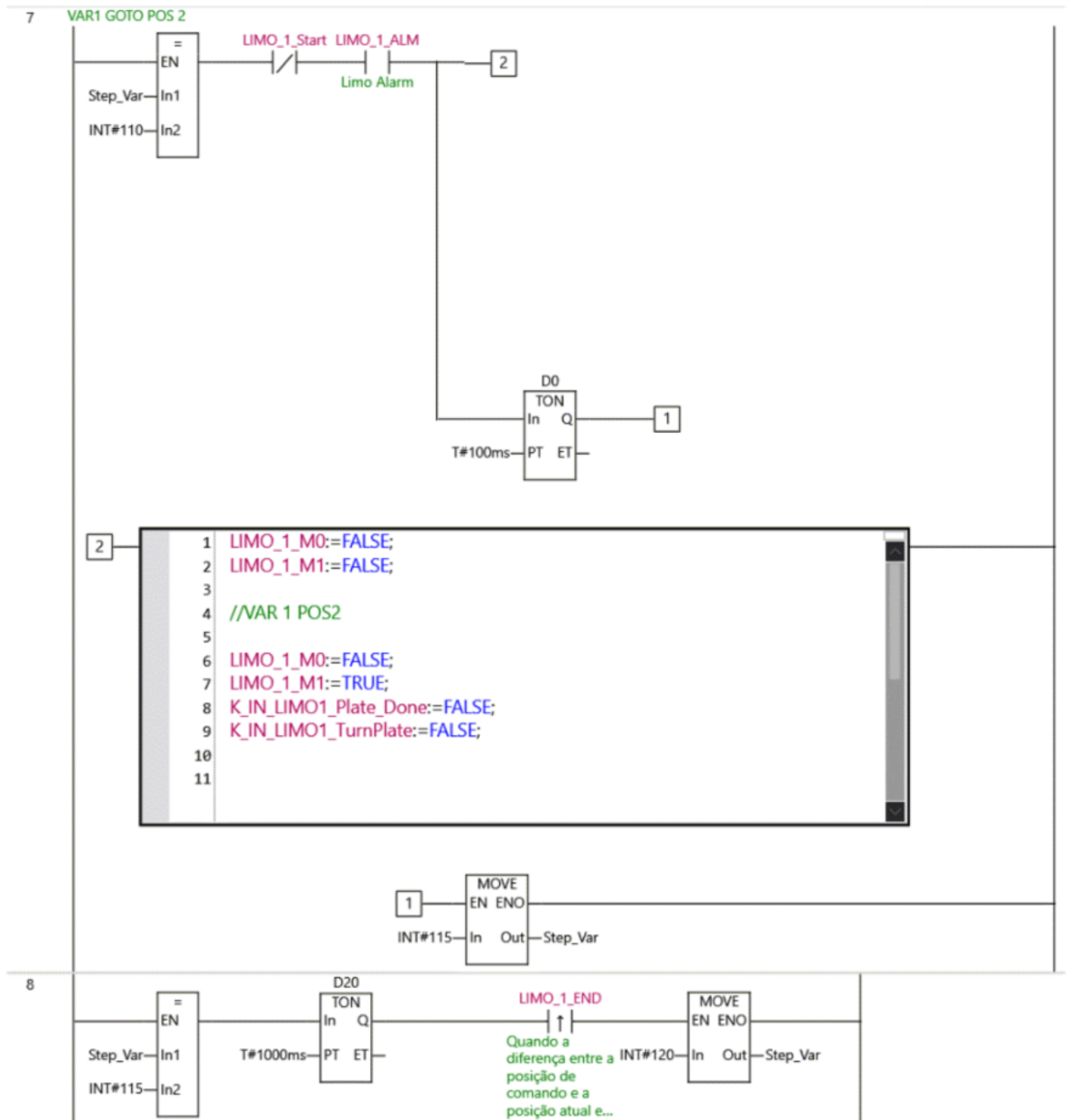
Anexo IV. Programa PLC

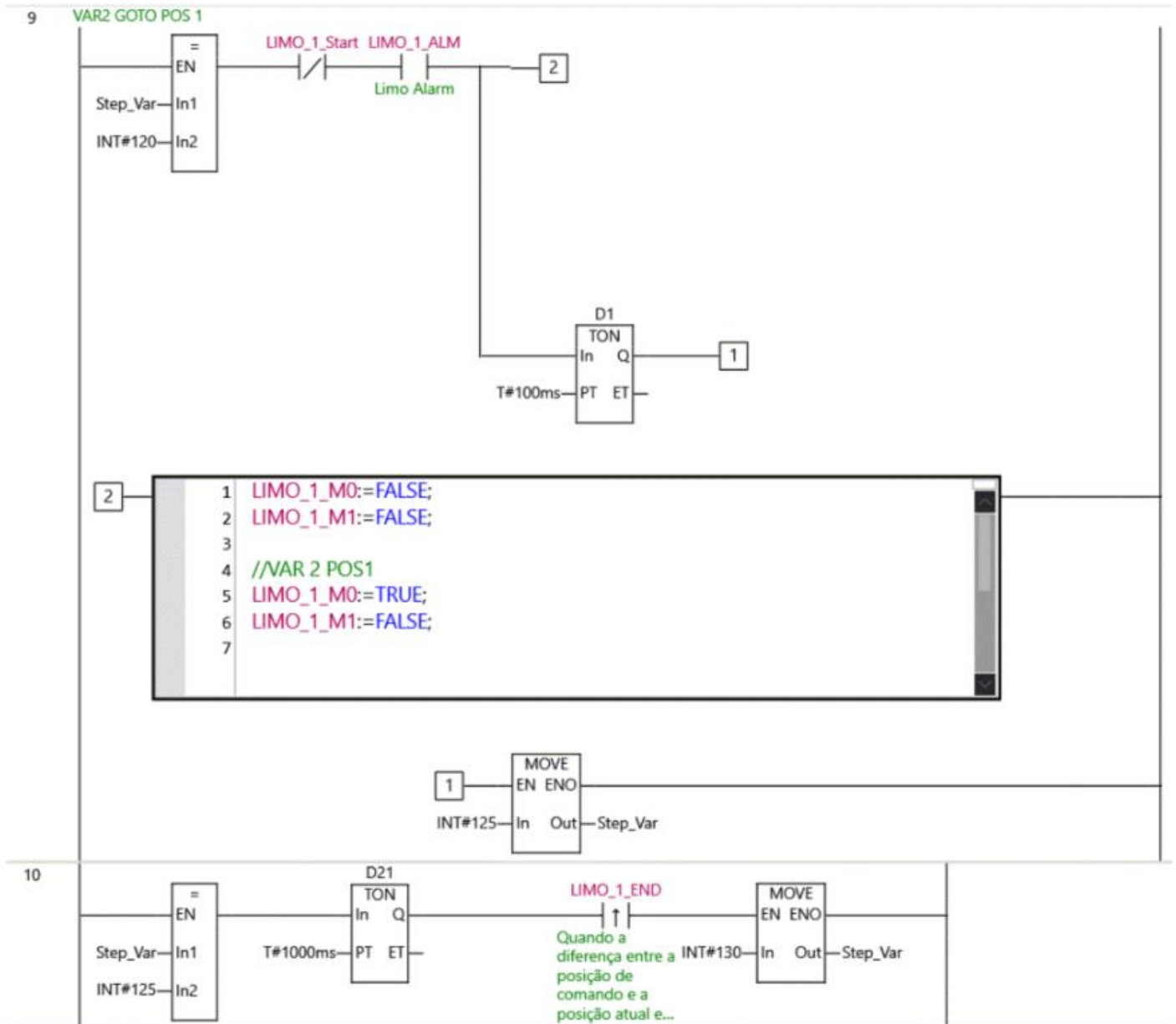
Controlo Limo 1

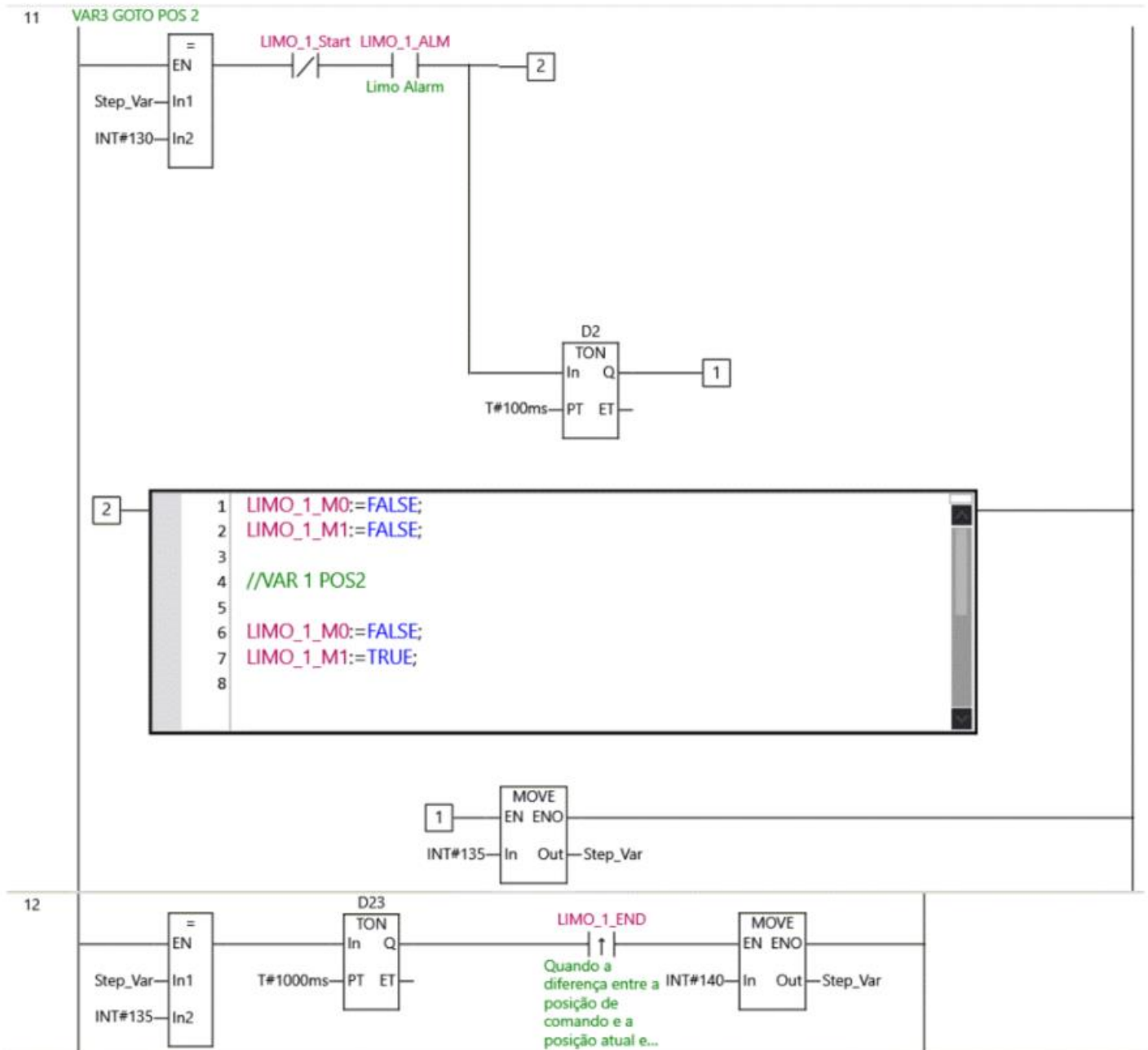


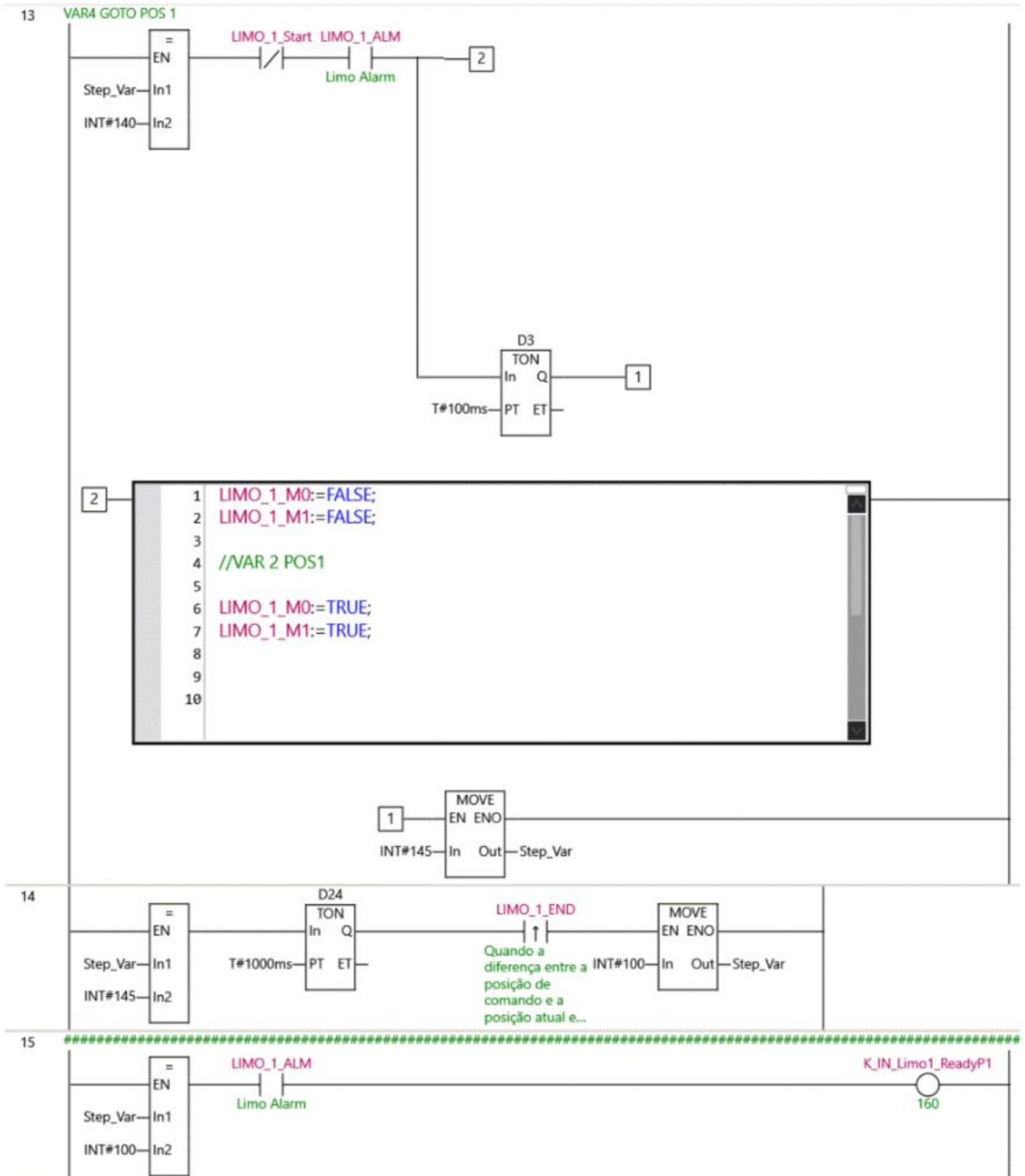


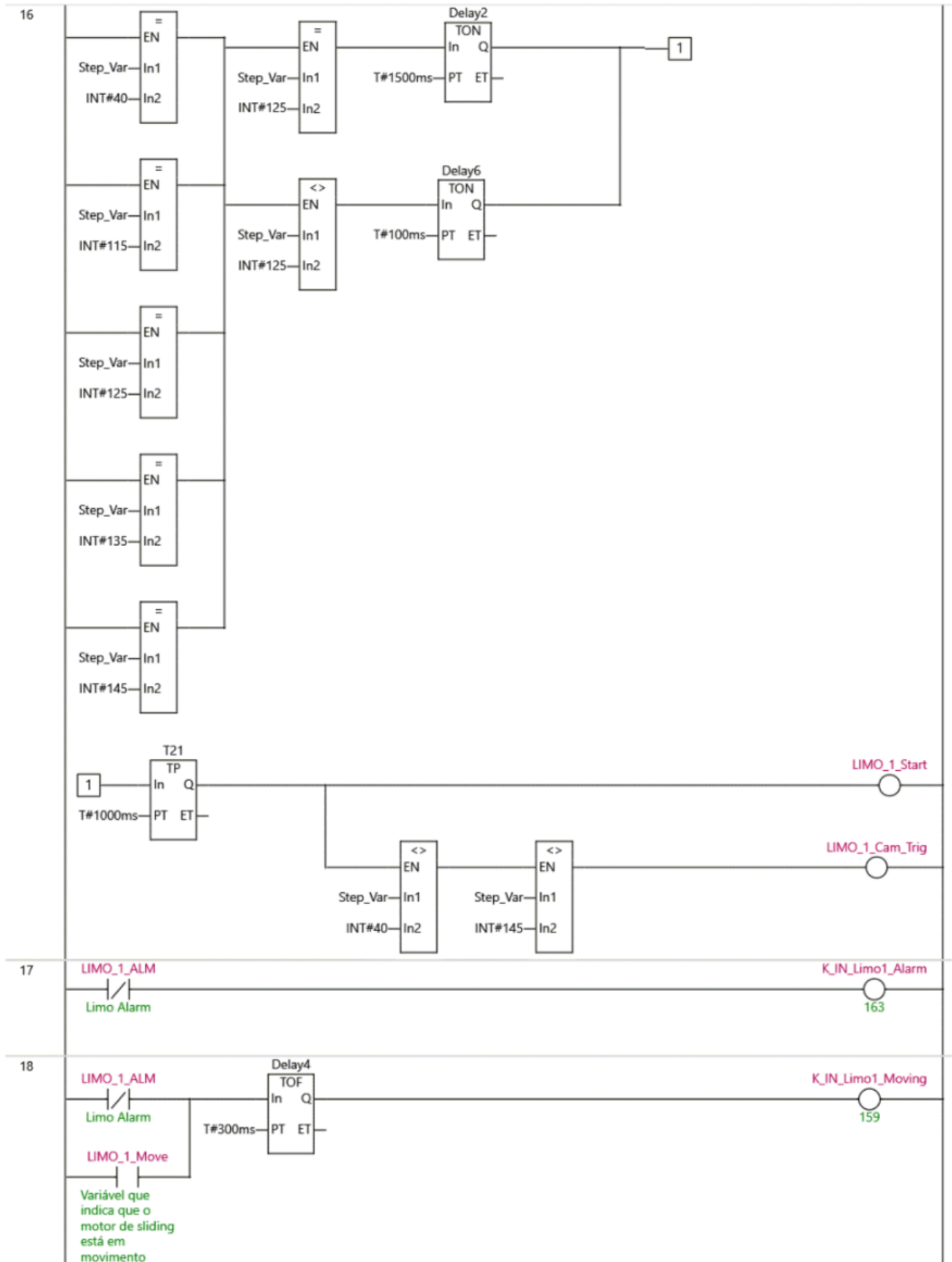












Iluminação Limo 1

