



**Politécnico  
de Viseu**

Escola Superior  
de Tecnologia  
e Gestão de Viseu

# **Simulação em computador da Célula Flexível de Fabrico**

Jorge Rosa Lourenço

## **Dissertação**

Mestrado em Engenharia Eletrotécnica - Energia e Automação Industrial

Trabalho efetuado sob a orientação de  
Professor Doutor António Manuel Pereira Ferrolho

Maio de 2023



**Politécnico  
de Viseu**

Escola Superior  
de Tecnologia  
e Gestão de Viseu

# **Simulação em computador da Célula Flexível de Fabrico**

Jorge Rosa Lourenço

## **Dissertação**

Mestrado em Eng.<sup>a</sup> Eletrotécnica - Energia e Automação Industrial

Trabalho efetuado sob a orientação de

Professor Doutor António Manuel Pereira Ferrolho

Maio de 2023

“A vida é feita de desafios, o que nós podemos fazer é superá-los.”

Teen Wolf



## **AGRADECIMENTOS**

Aqui, expresso toda a minha gratidão às pessoas que concorreram para, que se tornasse possível, a realização da presente dissertação. Por tudo isto, não quero deixar de agradecer a todos aqueles que contribuíram, de alguma forma, para o meu sucesso.

Ao meu orientador, Professor Doutor António Manuel Pereira Ferrolho, pelo imprescindível, valioso e incondicional apoio prestado em todas as fases de elaboração desta dissertação, pelos momentos de aprendizagem, durante todo o meu percurso de formação, por todo o incentivo dado ao longo da realização deste trabalho, por me ter aconselhado, nos momentos mais difíceis, fazendo-me ver que o desistir não é a melhor opção. O meu muito obrigado.

A todos os professores que conheci e se empenharam em conduzir-me neste percurso de formação.

Aos meus pais, que sempre primaram pela minha educação, um agradecimento especial por serem quem são, pelo carinho e dedicação que sempre me deram, pelas palavras encorajadoras e de compreensão e de total ajuda na superação dos obstáculos. Obrigado, pela herança que me deixaram.

À minha irmã pelo apoio irrestrito que me deu e por me ter “aturado”, tantas vezes, nos meus momentos mais difíceis.

Aos meu colegas e amigos, pela camaradagem, pela ajuda, pelos conselhos, pelos momentos divertidos, contribuindo, assim, este período de doces memórias.

A todos, um profundo agradecimento!



## RESUMO

A indústria tem sofrido uma evolução galopante, na história da Humanidade. O uso da robótica tem tido um aumento exponencial, dado que a introdução dos robôs permitiu a muitas empresas rentabilizar as linhas de fabrico, torná-las mais flexíveis e adaptativas, produzir com celeridade e precisão. Esta constatação conduz-nos a fragilidades no domínio da mão humana, porém o homem nunca poderá, pelo menos para já, ser descartável, dado que a sua presença é necessária em certas áreas da produção. A empresa necessita da mão e da destreza intelectual do homem em pontos onde os robôs não conseguem obter resultados.

Atualmente, as empresas têm optado, cada vez mais, pela utilização de processos flexíveis, sendo estes sistemas rápidos e com um tempo de reposta, por norma, muito curto. Esta metodologia satisfaz as necessidades dos clientes, contudo para que todas estas tecnologias funcionem é necessária a sua instalação e programação, para cada produto a ser obrado. Assim, impõe-se a necessidade executar a intervenção nos sistemas robóticos, ora, a produção não pode parar, visto conduzir as empresas a incumprimento de contratos e à perda de competitividade no mercado.

Para ultrapassar o grande entrave sentido nas Células Flexíveis de Fabrico (CFF), foi criado um software, que modela o sistema robótico de uma qualquer empresa conseguindo satisfazer todas as alterações que se pretendem em modo offline, passando, apenas, a intervenção pela necessidade de se fazer o update da nova programação na CFF real.

Posto isto, a realização desta dissertação, exigiu um estudo aprofundado de todos os elementos que constituem a CFF que se encontra na Escola Superior de Tecnologia e Gestão de Viseu (ESTGV) no laboratório de Robótica do Departamento de Engenharia Eletrotécnica (DEE), seguidamente foi desenvolvido um modelo virtual de toda a célula, no software da ABB, o RobotStudio, conduzindo ao levantamento das medições de cada elemento da CFF, para assim ser criado um modelo virtual. Ao longo do trabalho, foram realizados testes para averiguar a veracidade da correspondência, na fase final do trabalho foi desenvolvida toda a programação offline da CFF, seguindo-se a execução da simulação virtual, em computador, da célula real.

**Palavras-chave:** Células Flexíveis de Fabrico; Programação Offline; Robótica; RobotStudio.



## ABSTRACT

The industry has undergone a galloping evolution in the history of humanity. The use of robotics has seen an exponential increase, given that the introduction of robots has allowed many companies to monetize their manufacturing lines, make them more flexible and adaptable, and produce with speed and precision. This finding leads us to weaknesses in the domain of the human hand, but man can never, at least for the moment, be disposable, given that his presence is necessary for certain areas of production. The company needs man's hands and intellectual dexterity when robots cannot get results.

Companies have increasingly opted for using flexible processes, these systems being fast and with a response time, as a rule, very short. This methodology satisfies customers' needs, and however, for all these technologies to work, it is necessary to install and program them for each product to be built. Thus, there is a need to intervene in robotic systems. However, production cannot stop, as it leads companies to breach contracts and lose competitiveness in the market.

In order to overcome the main obstacle felt in the Flexible Manufacturing Cells (FMC), software was created that models the robotic system of any company, managing to satisfy all the changes that are intended in offline mode, passing only the intervention for the need of if you update the new programming in the real FMC.

That said, the realization of this dissertation required an in-depth study of all the elements that make up the FMC that is located at the Superior School of Technology and Management of Viseu (ESTGV) in the Robotics laboratory of the Department of Electrotechnical Engineering (DEE). A virtual model of the entire cell was developed in the ABB software, RobotStudio, leading to the necessity of measuring each FMC element to create a virtual model. Throughout the work, tests were carried out to verify the integrity of the correspondence. In the final phase of the work, the entire offline programming of the CFF was developed, followed by the execution of the virtual simulation, in computer, of the actual cell.

**Keywords:** Flexible Manufacturing Cells; Programming Offline; Robotics; RobotStudio.



# ÍNDICE GERAL

<b>ÍNDICE DE TABELAS / QUADROS</b> .....	xii
<b>ÍNDICE DE FIGURAS</b> .....	xiv
<b>LISTA DE SIGLAS / ABREVIATURAS</b> .....	xviii
<b>1. INTRODUÇÃO</b> .....	1
1.1. Enquadramento .....	1
1.2. Objetivos da investigação .....	1
1.3. Estrutura da dissertação .....	2
<b>2. REVISÃO HISTÓRICA</b> .....	3
2.1. Robótica Industrial .....	3
2.1.1. História Da Robótica.....	3
2.1.2. O Aparecimento Dos Robôs Industriais .....	5
2.1.3. Crescimento da Robótica Industrial .....	8
2.1.4. Constituição de um Robô industrial.....	11
2.1.5. Tipos de programação dos Robôs Industriais .....	13
2.1.5.1. Programação on-line .....	13
2.1.5.2. Programação off-line.....	14
2.2. Sistemas Flexíveis de Fabrico .....	16
2.2.1. Resumo da História .....	16
2.2.2. Definição de um SFF.....	17
2.2.3. Componentes de um SFF.....	17
2.2.4. Diferentes configurações de um SFF .....	18
2.2.5. Vantagens e Desvantagens de um SFF.....	19
<b>3. ELEMENTOS DA CÉLULA FLEXÍVEL DE FABRICO</b> .....	21
3.1. Composição da CFF .....	21

3.1.1.	Setor de Armazenamento .....	22
3.1.1.1.	Robô IRB1400 .....	23
3.1.1.2.	Ferramenta de trabalho do IRB1400 .....	24
3.1.2.	Setor de Transporte .....	25
3.1.3.	Setor de Fabrico .....	26
3.1.3.1.	Robô IRB140 .....	27
3.1.3.2.	Ferramenta de Trabalho .....	28
3.1.3.3.	Torno CNC.....	28
3.1.3.4.	Fresadora CNC.....	29
3.1.3.5.	Buffers.....	30
3.1.4.	Sinalização .....	31
4.	TRABALHO DESENVOLVIDO .....	33
4.1.	Software RobotStudio.....	33
4.1.1.	Smart Components .....	34
4.1.2.	Simulação .....	35
4.1.3.	Modelação .....	36
4.1.4.	Controlador.....	37
4.1.5.	Separadores de Programação .....	38
4.2.	Desenvolvimento da Célula da ESTGV em modo virtual.....	39
4.2.1.	Modelação da Célula Flexível de Fabrico .....	39
5.	ANÁLISE DOS RESULTADOS.....	51
5.1.	Programação e simulação do Setor de Armazenamento .....	52
5.2.	Programação e simulação do Setor de Fabrico.....	58
5.3.	Programação e simulação do Setor de Transporte.....	64
6.	CONCLUSÕES E SUGESTÕES DE TRABALHOS FUTUROS .....	67
6.1.	Conclusões.....	67

6.2. Conclusão Geral .....	68
6.3. Trabalhos Futuros .....	69
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>71</b>
<b>APÊNDICES</b> .....	<b>75</b>
7. APÊNDICE A, B e C .....	76
8. PROGRAMAÇÃO DO IRB1400, IRB140 E DO PLC .....	76
<b>ANEXOS</b> .....	<b>77</b>
9. ANEXO A: INTERFACE ROBÓTICO .....	78

## ÍNDICE DE TABELAS / QUADROS

Tabela 2-1: Softwares de programação de robôs industriais.....	14
Tabela 2-2: Linguagens de Programação de Robôs .....	15
Tabela 3-1: Estados da Pinça.....	24



## ÍNDICE DE FIGURAS

Figura 2-1: Leis de Asimov [5].	4
Figura 2-2: Robô Unimate [6].	5
Figura 2-3: Robô Versatran [8].	6
Figura 2-4: (a) Robô Puma, (b) Stanford Arm e (c) Robô IRB6 [11], [12] e [13].	7
Figura 2-5: Robô SCARA [14].	7
Figura 2-6: Robô Delta, instalado na Demarex [15].	8
Figura 2-7: Instalações anuais de robôs industriais [16].	9
Figura 2-8: Maiores mercados de instalações de robôs industriais em 2021 [16].	9
Figura 2-9: Principais fabricantes de robôs industriais em 2017 [17].	10
Figura 2-10: Top 5 das aplicações de robôs para uso profissional [18].	11
Figura 2-11: Analogia do robô com o braço humano [19].	11
Figura 2-12: Exemplo de diferentes controladores de robôs ABB [24].	12
Figura 2-13: Teach-pendant e manual leadtrough [25].	13
Figura 3-1: Organização da CFF [38].	21
Figura 3-2: Setor de armazém da CFF.	22
Figura 3-3: Rampa PA.	22
Figura 3-4: Robô IRB1400 da ABB [40] e [41].	23
Figura 3-5: Pinça Pneumática.	24
Figura 3-6: Setor de transporte.	25
Figura 3-7: Paletes e sensores do setor de transporte.	25
Figura 3-8: Setor de Fabrico.	26
Figura 3-9: Robô IRB140 da ABB [42].	27
Figura 3-10: Pinça do robô IRB140.	28
Figura 3-11: Torno EMCO Concept Turn 55.	28
Figura 3-12: Fresadora CNC EMCO Concept MILL 155.	29

Figura 3-13: Buffers. ....	30
Figura 3-14: Baliza de sinalização. ....	31
Figura 4-1: Ambiente de trabalho do RobotStudio. ....	33
Figura 4-2: Exemplo de AND. ....	34
Figura 4-3: Grupo de Sensores. ....	34
Figura 4-4: SC do sensor de colisão. ....	34
Figura 4-5: SC Attacher. ....	35
Figura 4-6: SC Movimento Linear. ....	35
Figura 4-7: Exemplo de ações criadas para a entrada AS1. ....	35
Figura 4-8: Sequência de uma ação para a entrada AS1. ....	36
Figura 4-9: Modelação de objetos no RobotStudio. ....	36
Figura 4-10: Importar ficheiro desenhado em CAD. ....	37
Figura 4-11: Criação dos sinais no controlador do robô. ....	37
Figura 4-12: Consola de programação virtual do RobotStudio. ....	38
Figura 4-13: Pastas de programação de trajetórias. ....	38
Figura 4-14: Editor <i>offline</i> do programa RAPID. ....	38
Figura 4-15: Pinça Pneumática virtual. ....	39
Figura 4-16: Mecanismo da pinça com 3 posições. ....	39
Figura 4-17: Modelo 3D das Matérias-Primas. ....	40
Figura 4-18: Modelo do Armazém. ....	40
Figura 4-19: Molde da rampa de descarga. ....	41
Figura 4-20: Base do IRB1400. ....	41
Figura 4-21: Setor de Armazenamento em modo virtual. ....	41
Figura 4-22: Modelo 3D da estrutura do setor de transporte. ....	42
Figura 4-23: Sensores indutivos e Stopper. ....	42
Figura 4-24: Tipos de paletes de MP e PA. ....	43

Figura 4-25: Códigos de cada Palete.....	43
Figura 4-26: Setor de transporte em modo virtual.....	44
Figura 4-27: Buffers em modo virtual.....	44
Figura 4-28: Modelo virtual da estrutura e do torno. ....	45
Figura 4-29: Ação criada para abrir a porta do torno. ....	45
Figura 4-30: Configurações do Torno CNC.....	46
Figura 4-31: Modelo virtual da Fresadora CNC.....	46
Figura 4-32: Ação criada para a abrir a porta da fresadora. ....	47
Figura 4-33: Configurações da Fresadora CNC. ....	47
Figura 4-34: Combinação da fresadora em maquinação. ....	48
Figura 4-35: Base do robô IRB140. ....	48
Figura 4-36: Setor de fabrico em modo virtual. ....	49
Figura 4-37: Célula virtual, vista do lado da fresadora. ....	49
Figura 4-38: Célula virtual, vista do lado do torno. ....	50
Figura 5-1: Robô IRB 1400, a trabalhar no armazém 1. ....	52
Figura 5-2: Código para o robô mudar de coluna na linha 1.....	53
Figura 5-3: Código para a escolha do armazém 1 ou 2.....	53
Figura 5-4: Código para gerar o ficheiro de texto “ArmazemTorno.txt”.....	54
Figura 5-5: Dados do ficheiro de texto “ArmazemTorno.txt”. ....	54
Figura 5-6: Rotina de controlo da palete. ....	54
Figura 5-7: Rotina de controlo da peca 1 e 2.....	55
Figura 5-8: a) Controlo da palete e b) Mensagem de controlo.....	55
Figura 5-9: Ficheiro de texto, Palete ”NOK”. ....	56
Figura 5-10: Ficheiro de texto, Palete ”OK”. ....	56
Figura 5-11: a) Palete “NOK” e b) Paleta ”OK”.....	56
Figura 5-12: Input da PPAT quando vai a 1.....	57

Figura 5-13: Descarga do PA da PPAT.....	57
Figura 5-14: a) PMPF e b) PMPT .....	58
Figura 5-15: Rotinas para descarregar a MP. ....	58
Figura 5-16: a) Descarga da PMPF e b) Carregamento do BMPF.....	59
Figura 5-17: a) Descarga da PMPT e b) Carregamento do BMPT. ....	59
Figura 5-18: Buffers de MP carregados. ....	60
Figura 5-19: Código para o carregamento das máquinas CNC.....	60
Figura 5-20: Etapas do Torno.....	61
Figura 5-21: Etapas da Fresadora. ....	61
Figura 5-22: Código para descarregar as máquinas CNC. ....	62
Figura 5-23: Buffers durante a maquinação da MP.....	62
Figura 5-24: Programação para o carregamento das paletes de PA. ....	63
Figura 5-25: BPAF vazio e PPAF carregada.....	63
Figura 5-26: Detecção da paleta .....	64
Figura 5-27: Movimento Linear da Paleta PMPT no lado do IRB 1400.....	65
Figura 5-28: Função Lógica para realizar uma rotação.....	65
Figura 5-29: SmartComponent de uma lâmpada de sinalização. ....	65

## LISTA DE SIGLAS / ABREVIATURAS

ABB	Asea Brown Boveri
AGV	Automated Guided Vehicle
ASEA	Allmänna Svenska Elektriska Aktiebolaget
BMPF	Buffer Matéria-Prima Fresadora
BMPT	Buffer Matéria-Prima Torno
BPAF	Buffer Produtos Acabados Fresadora
BPAT	Buffer Produtos Acabados Torno
CFF	Célula Flexível de Fabrico
CMU	Célula de Máquina Única
CNC	Computer Numerical Control
DEE	Departamento de Engenharia Eletrotécnica
DI	Digital Input
DO	Digital Output
ESTGV	Escola Superior de Tecnologia e Gestão de Viseu
FIFO	First-In-First-Out
FMC	Flexible Manufacturing Cells
FMS	Flexible Manufacturing Systems
IFR	Internacional Federation of Robotis
IPV	Instituto Politécnico de Viseu
ISO	International Organization for Standardization
IRB	Industrial Robot
MP	Matéria-Prima
MPF	Matéria-Prima Fresadora
MPT	Matéria-Prima Torno

PA	Produto Acabado
PAF	Produto Acabado Fresadora
PAT	Produto Acabado Torno
PLC	Programmable Logic Controller
PMPF	Paleta Matéria-Prima Fresadora
PMPT	Paleta Matéria-Prima Torno
PPAF	Paleta Produtos Acabados Fresadora
PPAT	Paleta Produtos Acabados Torno
PUMA	Programmable Universal Machine for Assembly
RAPID	Linguagem de Programação da ABB
SCARA	Selective Compliance Assembly Robot Arm
SA	Setor de Armazenamento
SF	Setor de Fabrico
SFF	Sistema Flexível de Fabrico
ST	Setor de Transporte
TCP	Tool Center Point
TIA	Totally Integrated Automation
VGT	Veículos Guiados por Trilhos



# 1. INTRODUÇÃO

## 1.1. Enquadramento

O projeto desenvolvido ao longo desta dissertação prendeu-se com a “Simulação em computador da Célula Flexível de Fabrico” enquadrando-se na área de Automação e Robótica Industrial. Atualmente, a maioria das empresas utilizam sistemas robotizados nas suas linhas de montagem e fabrico. Deste modo, há uma melhor gestão dos recursos humanos, deslocando-os para outras áreas de produção, onde são não só úteis como de difícil substituição, isto porque as mais recentes tecnologias ainda não conseguem superar o ser humano em todos os aspetos.

Os mercados, por norma, procuram cada vez mais produtos diversificados e originais, onde a questão da flexibilidade pode ser a chave do sucesso a estas exigências do mercado. Para uma resposta criativa e inovadora às necessidades do mercado é necessário ter sistemas de Automação e Robótica Industriais altamente flexíveis e perfeitamente interligados, que permitam ter sempre resposta para os, variadíssimos, tipos de exigência requeridas pelos consumidores.

Neste sentido, o trabalho que foi proposto pretende estudar os ambientes de fabrico flexíveis e simular em computador a CFF que se encontra no laboratório de Automação e Robótica do DEE.

## 1.2. Objetivos da investigação

Como referido, no ponto anterior, o principal objetivo deste projeto foi realizar toda a simulação da célula real, em computador, para que isso fosse possível foi necessário modelar toda esta CFF, tendo sido estabelecidos alguns objetivos necessários ao desenvolvimento destes projetos, tais como:

- Estudar os ambientes de fabrico flexíveis;
- Simular no *software RobotStudio* cada um dos setores da CFF do DEE (setor de fabrico, de transporte e de armazenamento);
- Integrar, controlar e sincronizar os vários setores existentes na CFF de forma a simular o funcionamento real da mesma.

### 1.3. Estrutura da dissertação

Esta dissertação encontra-se estruturada em seis capítulos, sendo descrito todo o trabalho prático desenvolvido, os resultados obtidos e as conclusões.

No primeiro capítulo, “Introdução”, operou-se o enquadramento do projeto a realizar, os objetivos pretendidos na efetuação deste trabalho e definida a estrutura da dissertação.

No segundo capítulo, “Revisão Histórica” é apresentada toda a história da robótica, o aparecimento dos robôs, o crescimento, ao longo dos anos, da robótica industrial, do seu começo e da sua evolução até às novas tecnologias, a constituição de um robô, os tipos de programação e, por fim, uma abordagem aos sistemas flexíveis de fabrico.

No terceiro capítulo, “Elementos da Célula Flexível de Fabrico”, descreve-se de uma forma resumida toda a CFF existente no DEE, servindo de base para o desenvolvimento do restante do trabalho.

No quarto capítulo, “Trabalho Desenvolvido”, foi descrito uma parte introdutória do *RobotStudio*, sendo é apresentado o *software*. Encontra-se, ainda, neste capítulo toda a modelação dos três setores da CFF virtual.

No quinto capítulo, “Análise dos Resultados”, acham-se destacados as partes mais relevantes da célula em funcionamento, assim como os pontos mais importantes da programação de cada setor.

No sexto capítulo, “Conclusões e Sugestões de Trabalho Futuros”, é apresentado a conclusão do trabalho desenvolvido, uma conclusão geral, sobre a dissertação e por fim as sugestões para futuros trabalhos/projetos, de forma a melhorar a CFF.

## **2. REVISÃO HISTÓRICA**

### **2.1. Robótica Industrial**

Neste capítulo aborda-se, de uma forma resumida, um pouco da história da robótica industrial, o seu desenvolvimento e evolução dos robôs, o papel importante que estes tiveram desde o seu aparecimento até à atualidade, nas vastas áreas, e, por fim, os tipos de programação existentes.

Com o avançar do tempo, a robótica tem estado cada vez mais presente no dia-a-dia do ser humano, quer no setor industrial, quer na saúde, podendo afirmar-se que se tornou, praticamente, impossível retirar a robótica de qualquer área de trabalho de alto risco, de grande esforço físico, ou mesmo, de trabalhos repetitivos que conduzam ao desgaste acentuado do operador/trabalhador.

#### **2.1.1. História Da Robótica**

A história do desenvolvimento robótico é um exemplo interessante de como um mito se pode transformar na realidade, ficando, assim, marcada por um mundo de fantasia que forneceu a inspiração para motivar o ser humano a tornar a ficção numa realidade, atingindo assim os objetivos nunca antes conseguidos, intelectualizados e materializados [1], [2] e [3].

Em 1921 o romancista e jornalista checo Karel Capek, criou uma obra de ficção científica designada por “Rossum’s Universal Robots”, onde usou pela primeira vez a palavra Robota, que advém do termo eslavo que “significa trabalhos forçados ou escravos” [1] e [4]. Embora, no final desta obra os robôs pudessem ser vistos como “seres malignos”, visto que no desenrolar das suas ações, estes tentam acabar com a vida humana. O propósito da sua invenção não estaria longe da realidade atual da palavra.

Mais tarde, o escritor de ficção científica, Isaac Asimov, decidiu escrever as leis da robótica, tendo estas sido publicadas por volta de 1950.

Leis da robótica de Asimov:

- 1ª Lei: um robô não pode ferir um ser humano ou, por inação, permitir que um ser humano sofra algum mal;
- 2ª Lei: um robô deve obedecer às ordens dadas por seres humanos exceto nos casos em que tais ordens entrem em conflito com a primeira lei;
- 3ª Lei: um robô deve proteger sua própria existência desde que tal proteção não entre em conflito com a primeira ou a segunda lei [5].

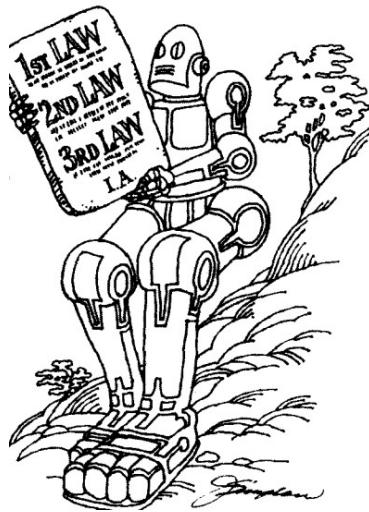


Figura 2-1: Leis de Asimov [5].

Com isto, os robôs surgem pela vontade de criar equipamentos capazes de reproduzir movimentos humanos, para mais tarde estes os poderem substituir os operadores nas funções de elevados riscos, ou por serem demasiado desgastantes. Existem, assim, várias definições de um robô, segundo a norma da ISO 8373 (International Organization for Standardization), define-se o conceito de robô industrial como “um manipulador automaticamente controlado, reprogramável e multifuncional com 3 ou mais eixos” [2], já o “Robot Institute of America”, designa o robô como “um manipulador reprogramável e multifuncional projetado para mover materiais, peças, ferramentas ou dispositivos especiais através de movimentos programados para a execução de uma variedade de tarefas” [2].

Entre outras definições, a mais inspiradora é oferecida por Merriam Webster, que afirmou que um robô é uma máquina com semelhanças às de um ser humano, que realiza várias ações complexas que, normalmente, são atribuídas ao ser humano. Estas definições permitem-nos, distinguir o que realmente pode ser considerado um robô industrial.

### 2.1.2. O Aparecimento Dos Robôs Industriais

Foi em 1954 que se verificou a evolução da robótica industrial quando George Charles Devol, deu o nome *Unimation* à sua nova invenção, uma máquina de uso geral com uma vasta diversidade de aplicações. Mais tarde Joseph Engelberger, nascido em 1925, em New York, desenvolveu um fascínio pelo robô no mundo da ficção científica, em particular a que Isaac Asimov escreveu. Foi então, por volta de 1956 que o destino cruzou Devol e Engelberger. Estas duas personalidades conheceram e conversaram sobre a matriz que tinham em comum, decidindo criar uma empresa em conjunto que levou ao desenvolvimento do robô industrial *Unimate*, apresentado na Figura 2-2, em 1961 foi utilizado para descarregar peças de uma máquina a altas temperaturas. Esta inovação só em 1975 gerou lucros à *Unimation* e se tornou líder mundial em robótica. Os lucros remontaram a 70 milhões de libras em 1983, tendo assim Joseph Engelberger sido considerado o “Pai da Robótica” [2].



Figura 2-2: Robô Unimate [6].

Com o passar dos anos, empresas como a *Ford* e a *General Motors* começaram a investir fortemente nos robôs industriais para alcançarem os seus objetivos definidos, havendo assim um aumento repentino das encomendas dos robôs, o que deu a oportunidade a muitas outras empresas de criarem o seu próprio dispositivo, como foi o caso da *AMF Corporation* que em 1962, fabricou o novo robô, o *Versatran*, apresentado na Figura 2-3, trata-se de um robô cilíndrico encomendado pela Ford para as suas linhas de produção em Canto (Ohio, EUA), o *Versatran* foi também o primeiro robô a ser instalado no Japão em 1967 [7] .

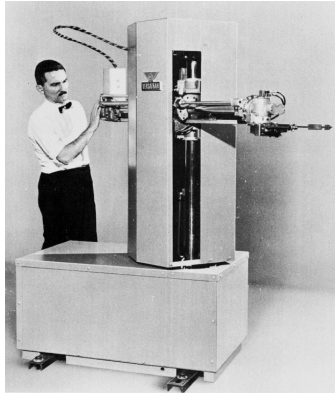


Figura 2-3: Robô Versatran [8].

Em 1969, a *Unimation* cedeu as suas licenças de produção de robôs à *Kawasaki Heavy Indus Trys Ltd.*, para o mercado japonês e asiático, tendo estes incrementado o *Kawasaki-Unimate 2000* o primeiro robô a ser produzido no Japão, desde aí o Japão tornou-se um dos grandes líderes no mundo da robótica industrial.

Na Europa os primeiros robôs foram instalados em 1967, na empresa *Svenska Metallverken*, localizada na Suécia. Dois anos depois, na Noruega, um fabricante de carrinhos de mão, devido à escassez da mão de obra, instalou-os numa fábrica para estes pintarem o seu engenho [9].

Em 1969, um estudante de engenharia mecânica da Universidade de Stanford, chamado Victor Scheinman, desenvolveu o Stanford Arm, apresentado na Figura 2-4 b), um robô com braço robotizado com seis juntas criado, exclusivamente, para ser controlado por computador. Mais tarde em 1973 este estudante veio a formar uma empresa a *Vicarm Inc.* produzindo o *Vicarm* que era um robô elétrico destinado à linha de montagem [9]. Esta empresa acabou por vir a ser comprada pela *Unimation*, sendo utilizada a tecnologia do *Vicarm* como base para desenvolver o *PUMA*. Posto isto, outras empresas começaram a investir, também, no seu robô industrial, como por exemplo a *Kuka* que desenvolveu em 1973 o *Famulus*, cujo nome em latim significa “servo”.

Em 1974, a empresa sueca *ASEA*, agora tão conhecida como *ABB*, introduziu no mercado o primeiro robô industrial de atuação 100% elétrica, o *IRB-6*, apresentado na Figura 2-4 c), este é caracterizado pela sua cor laranja, sendo um manipulador com 5 graus de liberdade, controlado por um microprocessador e com uma capacidade máxima de carga de 6 Kg [10], foi aplicado em locais produtivos para tarefas complexas, isto pela sua capacidade de se mover suavemente por caminhos contínuos, tornando-se ideal para realizar tarefas de soldadura em arco.

No ano de 1978, a *Unimation* que anteriormente já tinha desenvolvido o *PUMA*, apresentado na Figura 2-4 a), resolveu, neste ano juntamente com a *General Motors* lançá-lo no mercado, tendo sido considerado o arquetípico para os robôs antropomórficos, tendo a sua cinemática sido tomada como exemplo em vários livros de robótica, nas universidades de todo o mundo.



(a)



(b)



(c)

Figura 2-4: (a) Robô Puma, (b) Stanford Arm e (c) Robô IRB6 [11], [12] e [13].

No mesmo ano, surgiu outro marco importante na historia da robôs industriais, quando no Japão na Universidade Yamanashi, o professor Hiroshi Makino, criou o robô *SCARA* (Selective Compliance Assembly RobotArm), apresentado na Figura 2-5, este foi projetado com juntas de revolução que possuíam eixos verticais, tendo sido muito utilizado em aplicações, atuava verticalmente sobre a orientação de um peça horizontal, com menos movimento executava as operações rapidamente, com alta precisão e velocidade, desenvolvia de forma suave trabalhos de polimento, montagem, embalagem, entre outras.

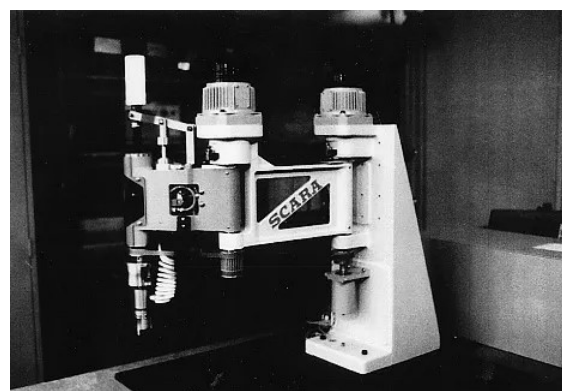


Figura 2-5: Robô SCARA [14].

Outro ponto relevante na história, dos robôs industriais, foi a mudança de serial para máquinas em paralelos, sendo estes robôs capazes de realizar tarefas de alta velocidade, com rigor de precisão. O exemplo mais importante do robô paralelo foi o robô *Delta*, apresentado na Figura 2-6, desenvolvido pela empresa suíça *Demaurex*, que em 1992, utilizou numa instalação denominada “Presto” (“em breve” em italiano), com o objetivo de realizar funções *pick and place* [9].



Figura 2-6: Robô Delta, instalado na Demarex [15].

### 2.1.3. Crescimento da Robótica Industrial

Com o andar dos tempos, os robôs industriais tiveram uma evolução bastante rápida, devido à sua grande capacidade de adaptação ao desenvolvimento de tarefas pesadas e ou perigosas para o ser humano, ao crescimento económico, pois o mesmo equipamento é capaz de desenvolver múltiplas funções, substituindo vários equipamentos distintos, estes fatores conduziram a que as empresas fizessem, fortemente, nesta área.

Através da *International Federation of Robotics* (IFR), é possível, verificar o aumento do número de robôs instalados nos últimos anos, sendo que a expectativa para o futuro seja ainda maior, como se pode observar através da Figura 2-7, é possível divisarmos o recorde de 517000 robôs instalados em fábricas em todo mundo, em 2021, pode, ainda observar-se que durante a pandemia as vendas cresceram 31%, graças ao enorme desenvolvimento do mercado da China, compensando as reduções do resto do mundo, tendo sido apontado como o ano de maior sucesso de vendas na história da indústria, superando anos de pré-pandemia.

Segundo Mariana Bill, presidente da *IFR*, “o uso da robótica e automação está a crescer a uma velocidade de cortar a respiração” [16], pois em seis anos, as instalações anuais dos robôs cresceram exponencialmente.

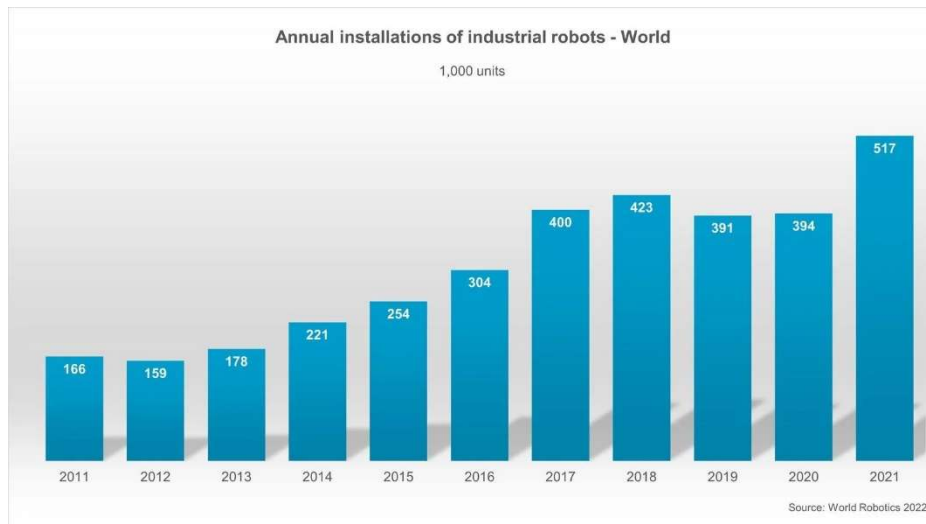


Figura 2-7: Instalações anuais de robôs industriais [16].

Como referido anteriormente, foi graças ao mercado da China, que houve um grande aumento de instalações de robôs industriais, crescendo em 51% as unidades instaladas, porque a China liderou a recuperação global após a pandemia, o que levou a este crescimento repentino foi o facto do governo chinês apoiar, financeiramente, a automação e a robótica, muito devido ao envelhecimento da população, reduzindo, assim, escassez de mão de obra.

Em segundo lugar, surge o Japão, com um aumento de 22% de instalações em 2021, sendo líder no fabrico de robôs. Na Figura 2-8, pode observar-se que também os outros mercados, tanto americano como europeu, aumentaram o nível de crescimento nesta área.

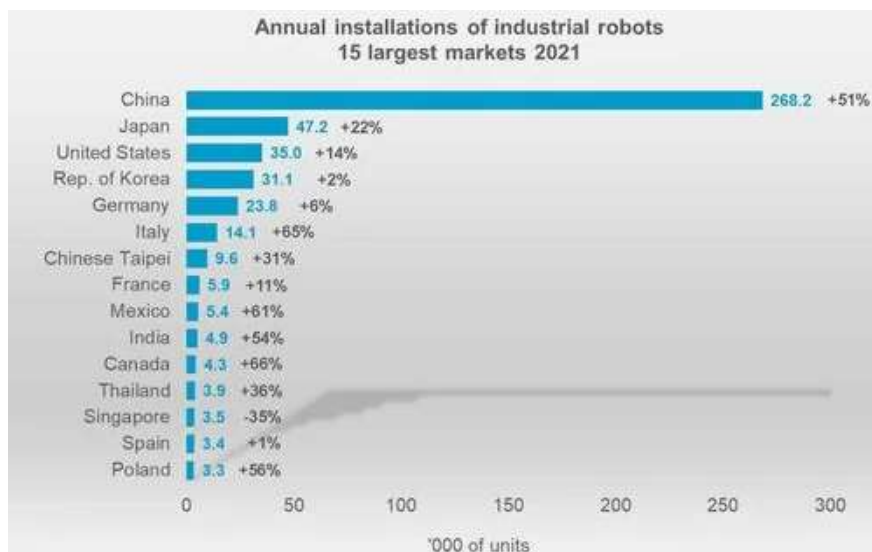


Figura 2-8: Maiores mercados de instalações de robôs industriais em 2021 [16].

Estes aumentos de instalações em grandes mercados só foram possíveis graças aos múltiplos fabricantes de robôs, que produziram vários tipos de robôs para diversas áreas de aplicação, investindo em novas tecnologias para combater o mercado que cada vez se torna mais competitivo.

Na Figura 2-9 pode constatar-se os maiores fabricantes de robôs no ano de 2017, destacando-se, em primeiro lugar, a marca japonesa *FANUC*, em seguida a *YASKAWA*, também ela, japonesa e para fechar o top 3 a empresa sueca *ABB*, os restantes seguiam na disputa para ultrapassar as 100000 unidades de robôs instalados.

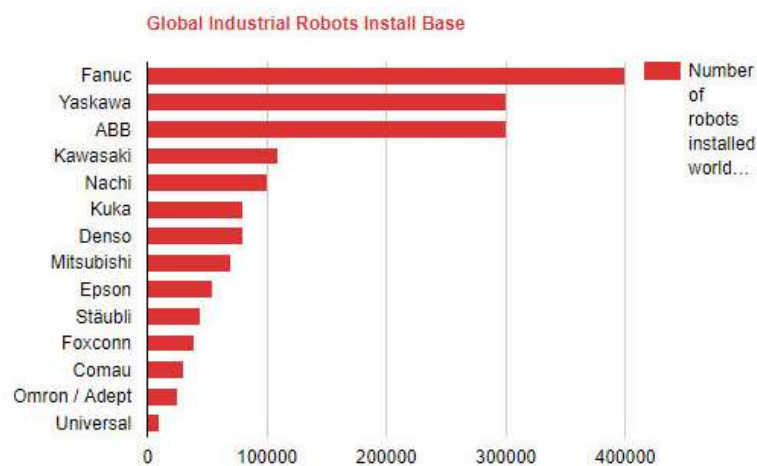


Figura 2-9: Principais fabricantes de robôs industriais em 2017 [17].

Segundo a *IFR* o tipo de indústria que mais robôs instalou, no ano de 2020, foi a indústria eletrónica, tornando-se a assim a maior consumidora de robôs, ultrapassando, desta forma, a indústria automóvel.

Para além destes dados podemos verificar, na Figura 2-10, que para uso profissional em 2021, um em cada três robôs fabricados foram direcionados para o transporte de mercadorias, sendo usados estes em fábricas, já os restantes são usados em hospitais para melhorar a qualidade do serviço prestado, nomeadamente, em blocos operatórios, em serviços de limpeza, e na agricultura. Já a nível de robôs para uso doméstico, os mais vendidos são utilizados para desenvolver tarefas domésticas, serviços de jardinagem, entre outras utilidades.

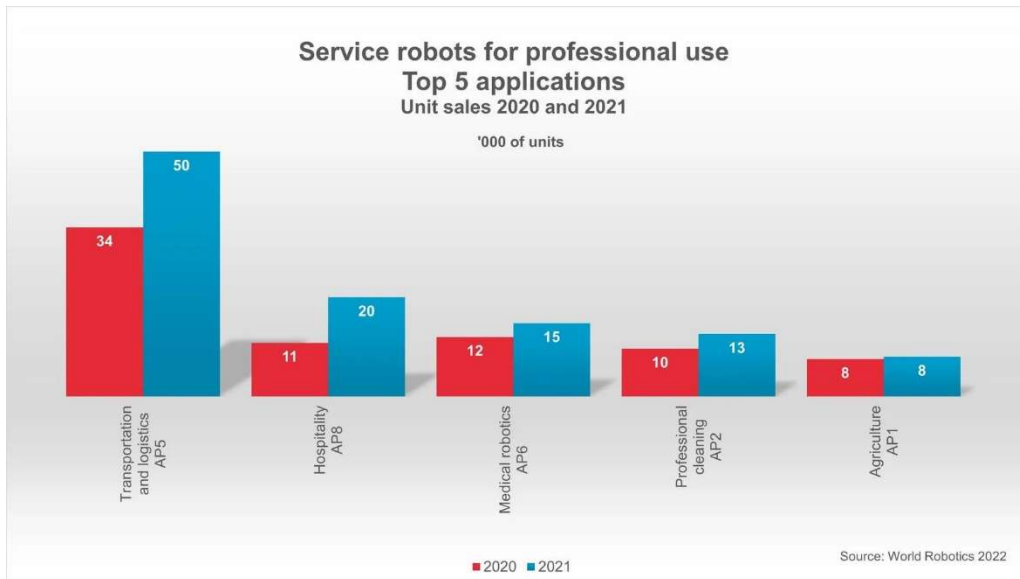


Figura 2-10: Top 5 das aplicações de robôs para uso profissional [18].

#### 2.1.4. Constituição de um Robô industrial

A Figura 2-11, mostra a analogia entre o robô industrial e o braço do ser humano.

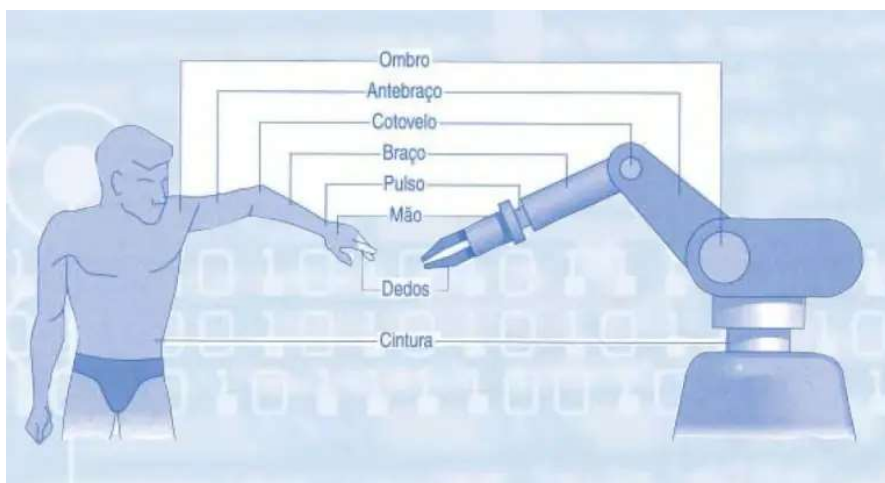


Figura 2-11: Analogia do robô com o braço humano [19].

O braço corresponde à parte do robô que está associada ao posicionamento cartesiano  $(x,y,z)$ , já o punho está direcionado para a orientação da *gripper*/pinça.

Os sensores podem dividir-se em sensores internos e externos, os internos determinam os valores dos ângulos ou posições de cada uma das articulações como, por exemplo, os *resolvers* e os *encoders* incrementais, enquanto os externos, são os responsáveis por conhecerem o ambiente que rodeia o robô, ou seja, sensores que permitem obter informação do

meio externo sobre o qual o robô opera. Estes sensores externos têm de converter as informações do exterior, para que o controlador as consiga processar como, por exemplo, para sabermos o estado do *gripper* (aberto ou fechado), sendo o motivo pelo qual os robôs vêm equipados com alguns sensores diferentes, variando de acordo com o fabricante, o tipo de robô, o modelo e até mesmo o local onde vai ser aplicado. Os principais sensores externos utilizados são: sensores indutivos, sensores capacitivos, sensores óticos, sensores tácteis, sensores de visão, entre outros [20].

Um outro elemento, são os atuadores que representam o papel dos “músculos” do robô [21], sendo estes responsáveis por gerar e impor movimento a qualquer parte mecânica, através de energia vinda dos motores elétricos, cilíndricos pneumáticos ou hidráulicos. Ao contrário dos sensores, os atuadores transformam a informação digital do controlador numa atuação do mundo real.

O controlador é o “cérebro” do robô industrial, apresentado na Figura 2-12, isto é, a unidade capaz de gerar informação de sensores, assim como gerar a informação para ativar os atuadores, tendo também como funções o controlo dos eixos de movimento e outros dispositivos que possam estar associados ao robô [22] e [23] .



Figura 2-12: Exemplo de diferentes controladores de robôs ABB [24].

Para o homem e o controlador terem interação, o controlador tem uma consola portátil, que é diferente de marca para marca, tendo estes uma linguagem diferente, específica para cada fabricante, este tipo de programação pode ser feita *on-line*, através da consola, ou até *off-line*, no computador, sendo que no *on-line* possibilita a interação física, o que não possível *off-line*.

### 2.1.5. Tipos de programação dos Robôs Industriais

Relativamente à programação dos robôs, esta consiste em ensinar ao robô a tarefa que ele deve executar, onde se define um conjunto de pontos, as trajetórias, as velocidades de movimento, bem como a interação do robô com outros equipamentos através da presença de sensores externos. Na indústria moderna existem dois tipos de programação que se destacam, a programação *on-line*, que requer a utilização de um robô físico e de uma consola para manobrar o robô e a programação *off-line*, que necessita da utilização de um *software* de simulação para o robô, que difere em cada fabricante, estes permitem programar e desenvolver o ambiente de trabalho de cada célula.

#### 2.1.5.1. Programação on-line

A programação *on-line* é realizada pelo operador, onde ensina os movimentos e as orientações ao robô, através da movimentação do *joystick* da consola portátil (*teach-pendant*) ou do manual *leadtrough*, como apresentado na Figura 2-13. Ao utilizar-se a consola consegue controlar-se o robô de forma precisa, através do *joystick*, movimentar o robô para as posições desejadas, memorizar as trajetórias e pontos na memória do controlador. A utilização do manual *leadtrough* exige a deslocação direta do robô por parte do programador, este move, manualmente, o braço robótico para o ponto desejado, guardando durante o movimento, os dados relativamente às posições e orientações do *Tool Center Point* (TCP), na sua memória interna.

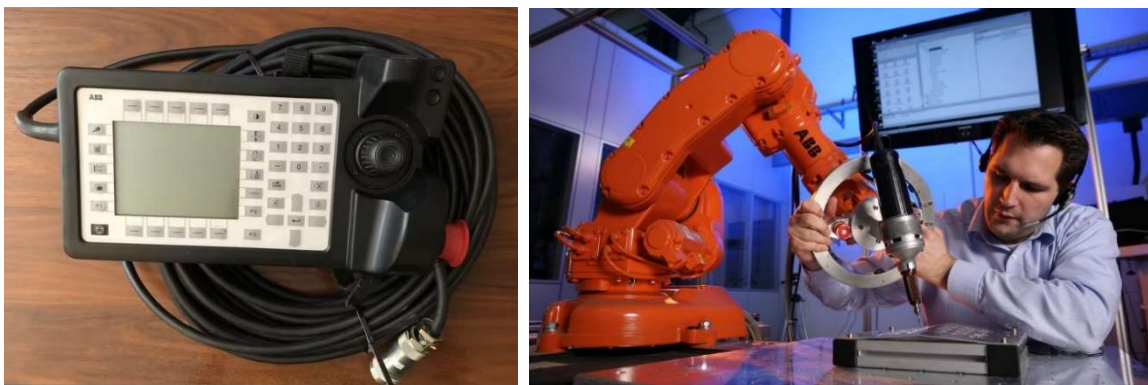


Figura 2-13: Teach-pendant e manual leadtrough [25].

A programação *on-line* tem como vantagem o facto de ser de fácil aprendizagem para qualquer programador, isto é, a simplicidade de criar os pontos necessários para o programa a desenvolver, sendo depois apenas necessário definir na programação, o tipo de movimento, as

velocidades, as zonas de proximidade e as ferramentas a utilizar. As desvantagens na programação *on-line* prendem-se com o facto de haver necessidade de estar em presença do robô, para executar a alteração no programa. Assim, acarreta inconvenientes como o de uma fábrica ter de parar a produção, sendo outro ponto negativo a segurança do programador, que a qualquer distração pode aumentar o risco de acidentes.

#### 2.1.5.2. Programação off-line

A programação *off-line* pode ser considerada como o processo pelo qual os programas são desenvolvidos, parcial ou totalmente, sem a necessidade do uso direto do robô [26], isto veio combater as desvantagens da programação *on-line*, reduzindo o tempo de paragem dos robôs, a retirada do programador de ambientes potencialmente perigosos, permitindo assim conseguir desenvolver o programa sem interferir no trabalho de uma empresa/fábrica.

Os novos investimentos no mercado operam-se ao nível do desenvolvimento das plataformas de simulação, que permitem fazer a modelização do robô, o ambiente gráfico de trabalho em 3D, as ferramentas utilizadas pelo robô para desenvolver a tarefa e as peças a manipular pelo robô. Estes novos *softwares* podem ser específicos para cada fabricante ou alguns até mesmo genéricos, como podemos ver na Tabela 2-1.

Tabela 2-1: Softwares de programação de robôs industriais.

Software Proprietário	Robôs Compatíveis
RobotStudio	ABB
RoboGuide	FANUC
MotoSim EG	MOTOMAN
Software Genérico	Robôs Compatíveis (ex.)
Workspace	ABB
Octopuz	KUKA
Delmia	FANUC

Porém, para além destes novos *softwares* é necessário a recriação da célula flexível de fabrico e para isso são utilizados ficheiros neutros como (STL, IGS, STEP, SAT), onde cada um corresponde a um determinado objeto/produto, criados em *softwares* 3D, que por sua vez são implementados no programa de simulação para ser possível criar a célula de trabalho virtual, para desenhar peças mais simples em que não seja necessário o modelo 3D.

Os próprios *softwares*, permitem também fazer a modelação dos objetos que se estimem essenciais desenvolver.

As etapas que na programação *off-line* são necessárias realizar implicam a criação do modelo 3D da célula robotizada, a calibração do modelo simulado, a criação dos programas de operação do robô, a verificação de colisões e, por fim, o envio dos programas para o controlador virtual para ser testado. Após a verificação da simulação em funcionamento, pode, então, avançar-se para a utilização do programa no robô real, havendo a obrigatoriedade de o testar em modo manual, para verificar se existem erros ao nível de pontos criados ou a nível das movimentações, que não se conseguem ver em simulação.

A nível do desenvolvimento do programa, atualmente temos vários tipos de linguagem, como apresenta a Tabela 2-2. Em 1973 foi desenvolvida a primeira linguagem de programação dos robôs, a *WAVE*, tem sido considerada uma linguagem experimental, sendo que a primeira linguagem comercial disponível foi o *VAL*, usada pelo robô *PUMA* em 1979, sendo, por isso, considerada linguagem de 1º geração. A *VAL* foi revista em 1984, dando origem à *VAL II*, tornando-se na linguagem de 2º geração.

Hoje cada fabricante tem uma linguagem própria do sistema de simulação ou, eventualmente, idêntica à do robô em uso, o que permite realizar procedimentos similares aos utilizados na programação *on-line*.

Tabela 2-2: Linguagens de Programação de Robôs

Linguagem	Fabricante
RAPID	ABB
KAREL	FANUC
IRL	REIS
KRL	KUKA
INFORM II	MOTOMAN
AS	KAWASAKI
MRL	MITSUBISHI
BAPS	BOSH
VAL, VAL II	UNIMATION, ADEPT

As características fundamentais das linguagens de programação de robôs passam pelo controlo de movimentos do manipulador, a manipulação de entradas e saídas, digitais e/ou analógicas, capacidade de modificação do comportamento do robô, capacidade de comunicação, de processamento de dados e de execução de tarefas em simultâneo (*multitask*).

Os problemas deste tipo de programação são os erros nos sistemas de movimentos, erros de modelização, das posições relativas do robô às peças a manipular e *bugs* que o próprio *software* tem, em que muitas das vezes mostra que desenvolveu determinada tarefa, mas olhando para a simulação vemos que isso não aconteceu.

## **2.2. Sistemas Flexíveis de Fabrico**

### **2.2.1. Resumo da História**

Por volta de 1960, o mercado começou a tornar-se mais competitivo, dado que inicialmente a única preocupação das empresas era o custo de produção, mais tarde, essa prioridade passou a ser a qualidade do produto, mas à medida que o mercado foi evoluindo, a velocidade de entrega do produto foi uma exigência imposta pelo cliente ao fabricante. Esta imposição, obrigou as empresas a reformular uma nova estratégia de adaptação, para concretizarem uma resposta, estas tinham de se tornar mais flexíveis nas suas operações e intervir em diferentes mercados, isto é, pela primeira vez na história, os fabricantes tiveram de ter em conta a eficiência, qualidade e flexibilidade para se manterem competitivos no mercado, passando, assim, a estar em diferentes setores de mercado [27], [28] e [29].

Posto isto, a inovação dos Sistemas Flexíveis de Fabrico (SFF), passou a estar relacionada com a vantagem competitiva, sendo que a palavra em voga, nessa altura, era a “agilidade”. Para além de um fabricante ser mais célere na chegada com os produtos ao mercado, estes operavam cada produto que produziam com um custo menor, conseguindo, assim, satisfazer o cliente e captar através disso novos parceiros [27] e [30].

### 2.2.2. Definição de um SFF

Atualmente, os SFF são os sistemas mais automatizados e sofisticados para combater as exigências do mercado, tais como, aumento da concorrência do mercado a nível internacional e a necessidade de redução do tempo e dos custos de produção [27], [31] e [32].

Para que um sistema seja considerado flexível, tem de ser capaz de distinguir diferentes produtos, tem de ser rápido nas suas transições, bem como ser capaz, de em pouco tempo, mudar as operações que está a desenvolver para outras novas que lhe foram aplicadas [27] e [33].

### 2.2.3. Componentes de um SFF

Os componentes básicos de um SFF são:

- Estações de trabalho, sendo por norma constituídas por máquinas CNC, responsáveis por executar diferentes operações de maquinação, como centros de desbaste e perfuração, torneamento, estações automáticas de ferramentas, estações de montagem, de carga e descarga, contudo estas estações de trabalho cada vez mais estão a ser projetadas para desempenhar operações de controlo e inspeções dos produtos;
- Sistemas automatizados e de armazenamento de produtos, por norma feito por um robô que faz movimentos aleatórios e independentes de cada peça de trabalho entre estações, executa cargas e descargas dos materiais, para depois o transportador ser capaz de fazer a deslocação dos produtos entre estações;
- Sistemas computadorizados que controlam estações e transportadores, por um computador central, onde este é utilizado para coordenar cada atividade executada, para além de controlar cada estação, distribui tarefas, faz o controlo de produção, controlo do tráfego e de transporte, a monitorização das peças de trabalho, o controlo da ferramenta, entre outras funções [27], [31], [33] e [34].

Dentro de cada um destes componentes básicos existem, robôs manipuladores; *Automated Guided Vehicle* (AGV); Veículos Guiados por Trilhos (VGT); autómatos programáveis; sistemas de visão; máquina CNC; transportadores; *buffer*; , armazéns [28].

Com o passar dos anos, as empresas começaram a perceber através dos sistemas que controlam as estações, que muitos dos progressos tecnológicos ao nível da automação dos sistemas de fabrico se têm centrado em níveis inferiores (próximos ou muito próximos do *hardware*), o que resultou no aparecimento de máquinas mais sofisticadas, com menores tempos de maquinação, e menores tempos de *setup*. Devido a estes desenvolvimentos tecnológicos, começou a dar-se menos importância ao tempo de execução das operações de produção, isto porque se passou a valorizar cada vez mais a eficiência com que os materiais são transportados entre estações de trabalho e entre estas e os armazéns [35] e [36].

#### **2.2.4. Diferentes configurações de um SFF**

Um sistema flexível de fabrico, consoante o número de estações de produção é caracterizado em categoria diferente, tais como:

- Célula de máquina única (CMU), onde estas células contêm apenas uma estação de trabalho, composta por uma máquina CNC, um sistema de transporte e de armazenamento;
- Célula flexível de fabrico (CFF) que é composta por diferentes equipamentos de diferentes fabricantes, possui entre duas a três estações de trabalho, onde a função é produzir as mesmas peças ou diferentes consoante a necessidade de produtividade de cada produto;
- Sistema flexível de fabrico (SFF), composto por quatro ou mais estações de trabalhos, onde as peças são transportadas automaticamente para outro processo, estes sistemas destacam-se pela flexibilidade e geralmente possuem estações de produção, transportador e sistemas para monitorizar, controlar e inspecionar os produtos fabricados [27], [28] e [37].

### **2.2.5. Vantagens e Desvantagens de um SFF**

Atualmente, a grande vantagem da implementação dos SFF, como já referido é a flexibilidade e adaptabilidade aos novos produtos a fabricar, mas para além desta vantagem existem outras, a saber: tempo reduzido de paragem das máquinas; a melhor gestão e controlo; a redução do espaço de produção; aumento da qualidade e da produtividade. A evolução tecnológica conduzirá certamente, num futuro próximo, a novos avanços do mercado como lasers, sistemas de visão artificial para análise de falhas, entre outras [36].

Relativamente às desvantagens verifica-se a complexidade do sistema que implica custos dos equipamentos, custos elevados da manutenção, custos elevados de formação, custos de automatização e custos do produto, sendo este um das maiores inconvenientes dos SFF, nas empresas.



### 3. ELEMENTOS DA CÉLULA FLEXÍVEL DE FABRICO

Neste capítulo é apresentado a Célula Flexível de Fabrico (CFF), que se encontra presente no laboratório de Robótica e Automação do Departamento de Engenharia Eletrotécnica (DEE) na Escola Superior de Tecnologia e Gestão de Viseu (ESTGV).

#### 3.1. Composição da CFF

A CFF do DEE, como se pode constatar na Figura 3-1, é um exemplo de uma célula que pode encontrar-se num ambiente industrial. Esta célula, como referido no Capítulo 1, é dividida em três setores, a saber:

- O setor de armazenamento, constituído por um robô da *ABB IRB1400* e por cinco armazéns;
- O setor de fabrico, constituído por um robô *ABB IRB140*, por duas máquinas CNC, uma fresadora e um torno, e *buffers* de matérias-primas e produtos acabados;
- O setor de transporte, composto por um tapete retangular (*conveyor*) cuja sua função é transportar peças entre o setor de armazenamento e o setor de fabrico. Existe um PLC na CFF que envia e recebe dados para que as paletes levem a matéria-prima para o setor de fabrico e tragam os produtos acabados até ao local de descarga (rampa de PA), no setor de armazém.

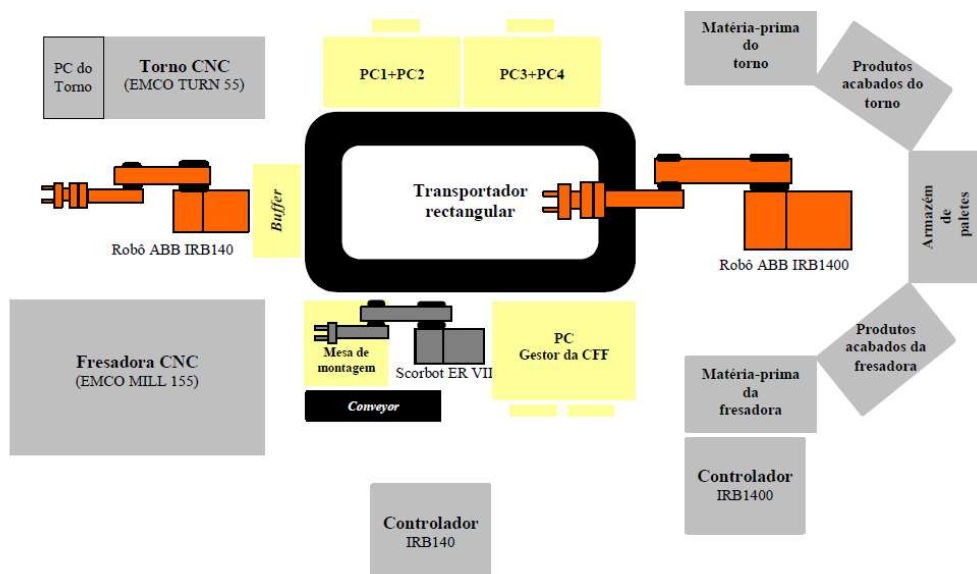


Figura 3-1: Organização da CFF [38].

### 3.1.1. Setor de Armazenamento

O setor de armazenamento, como pode perceber-se na Figura 3-2, tem como função o armazenamento da matéria-prima (MP) e a descarga dos produtos acabados (PA). Sendo constituído por um robô da *ABB*, modelo *IRB1400*, por armazéns que contêm a matéria-prima e por uma rampa para a colocação dos produtos acabados.

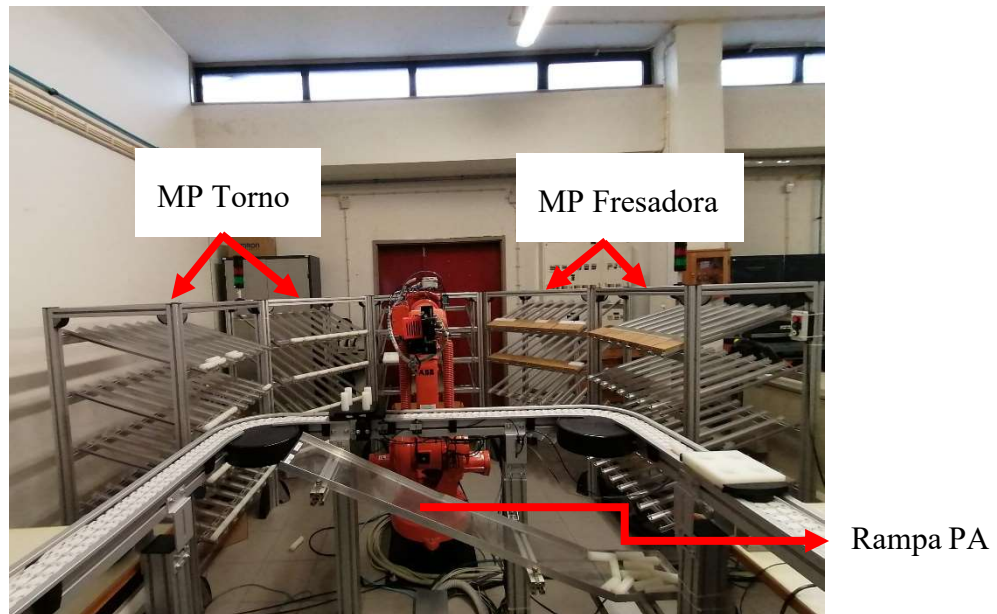


Figura 3-2: Setor de armazém da CFF.

Este setor está dividido em duas partes, o lado esquerdo é destinado ao armazenamento da matéria-prima do torno, o lado direito ao armazenamento da matéria-prima da fresadora. Cada armazém, é constituído por 4 linhas e 8 colunas em cada linha. Ainda neste setor, como referido anteriormente, a rampa de descarga, como apresentado na Figura 3-3, serve para a colocação dos produtos acabados provenientes do torno e da fresadora.

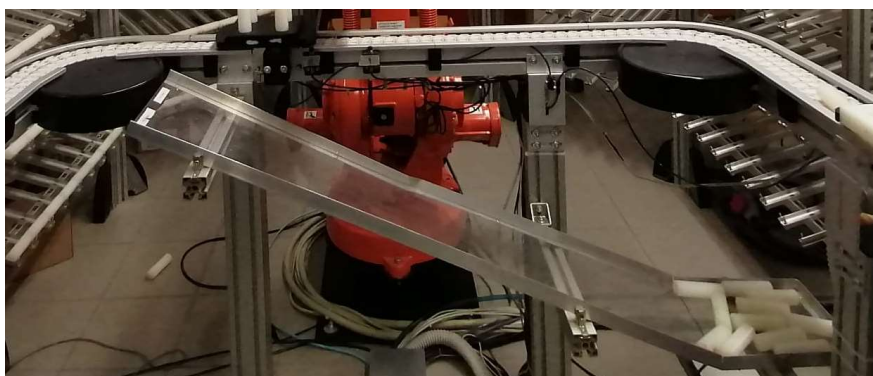


Figura 3-3: Rampa PA.

### 3.1.1.1. Robô IRB1400

O robô *IRB1400* da *ABB*, já referido, utilizado no setor do armazém, apresenta as seguintes características:

- seis graus de liberdade/seis eixos de rotação;
- estrutura antropomórfica;
- um punho esférico ( $\theta_4$  - *roll*,  $\theta_5$  - *pitch*,  $\theta_6$  - *roll*);
- alcance máximo de 1,44m;
- capacidade de carga máxima de 5 Kg;
- repetibilidade de  $\pm 0,05$ mm.

Na Figura 3-4, pode observar-se o volume de trabalho do robô e algumas das características referidas [39].

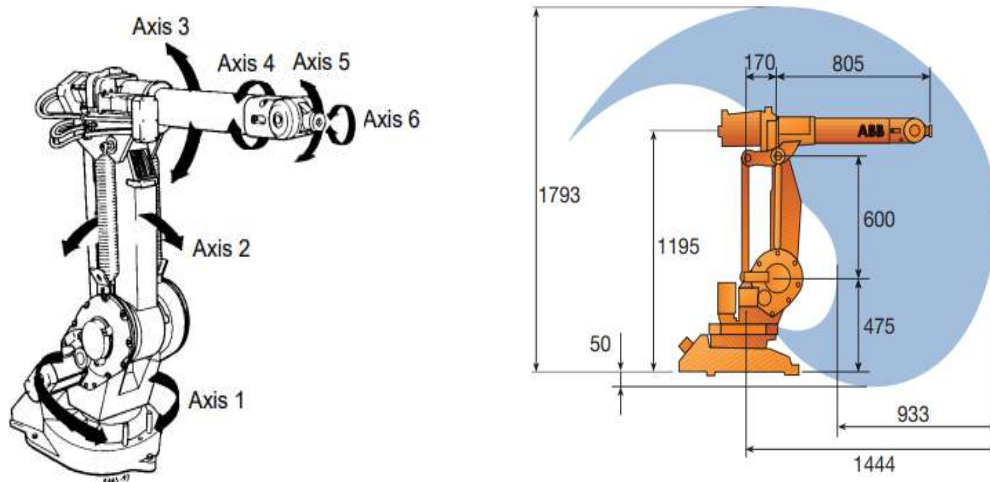


Figura 3-4: Robô IRB1400 da ABB [40] e [41].

### 3.1.1.2. Ferramenta de trabalho do IRB1400

Para o robô desempenhar corretamente as suas funções no armazém, foi necessário instalar uma pinça pneumática no punho do robô, como ilustra a Figura 3-5, estando esta preparada para o transporte de dois tipos de peças de trabalho completamente distintas, paralelepipedais e cilíndricas.

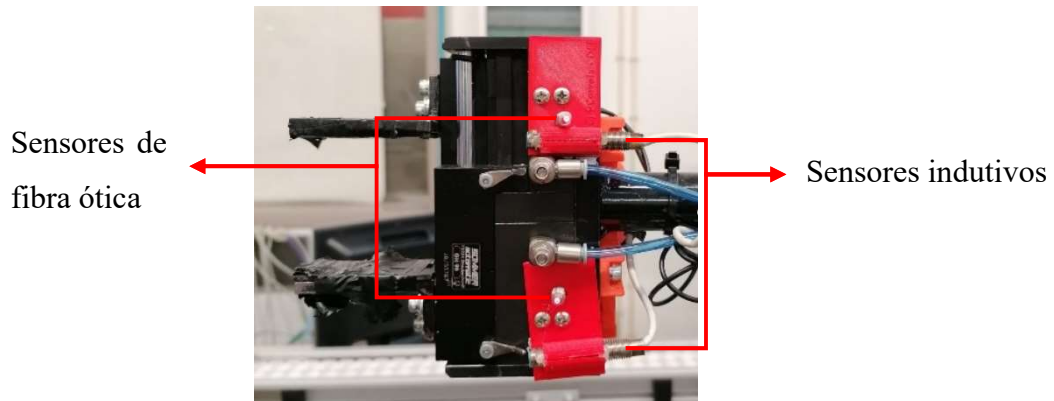


Figura 3-5: Pinça Pneumática.

Esta pinça pneumática possui ainda dois sensores indutivos para verificar se os “dedos” estão abertos ou fechados.

Através destes dois sensores indutivos também é possível saber se o robô ao fechar os “dedos” pegou numa peça ou não. A Tabela 3-1, mostra as várias combinações dos sinais emitidos por estes dois sensores. Esta ferramenta de trabalho possui também dois sensores óticos, como ilustra a Figura 3-5, tendo estes a função de verificar se as peças, colocadas pelo robô nas paletes, estão bem posicionadas ou não. Desta forma é possível obter a informação de *OK* ou *NOK* para as respetivas paletes.

Tabela 3-1: Estados da Pinça.

Estado do gripper	Sensor indutivo 1 (DI10_10)	Sensor indutivo 2 (DI10_11)
Aberto	1	0
Fechado com peça	0	1
Fechado sem peça	0	0

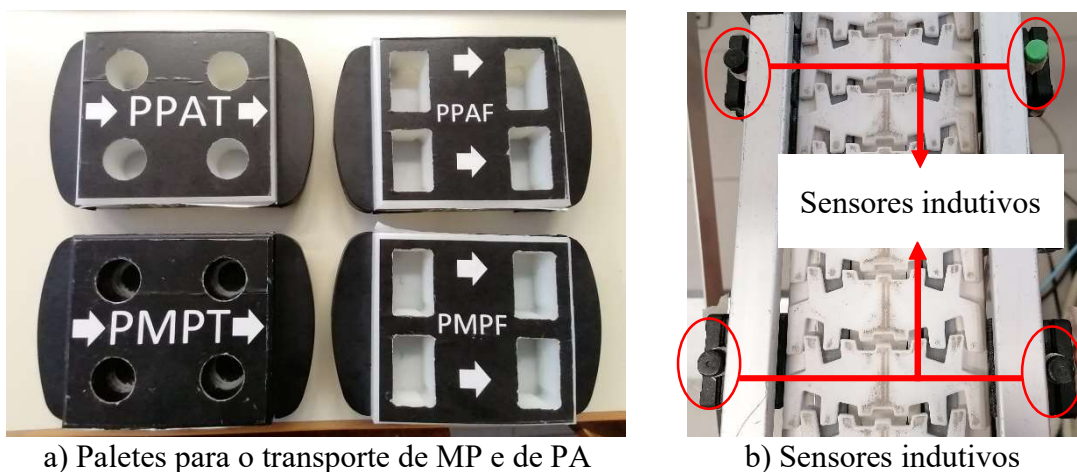
### 3.1.2. Setor de Transporte

O setor de transporte está situado no meio da CFF, como apresentado na Figura 3-6, tem uma configuração retangular, sendo constituído por um tapete rolante de forma a deslocar as paletes pelos vários setores, sempre no sentido anti-horário.



Figura 3-6: Setor de transporte.

A função deste setor é realizar o transporte das matérias-primas e produtos acabados, sendo este transporte possível, pela existência das quatro paletes, que circulam entre o setor de armazenamento e o setor de fabrico, desta forma cada paleta tem uma função diferente dentro da CFF, uma leva a matéria-prima do torno (PMPT), outra leva a matéria-prima da fresadora (PMPF), as outras duas, uma está destinada a trazer os produtos acabados do torno (PPAT) a outra da fresadora (PPAF), como apresentado na Figura 3-7 a).



a) Paletes para o transporte de MP e de PA

b) Sensores indutivos

Figura 3-7: Paletes e sensores do setor de transporte.

Na parte inferior de cada uma destas paletes, estão alguns elementos metálicos posicionados em pontos diferentes, para que quando as paletes se aproximarem dos sensores indutivos, como podemos notar na Figura 3-7 b), posicionados junto ao setor de armazenamento e também no setor de fabrico, estes as consigam identificar o tipo de palete, e transmitir a respetiva informação aos robôs.

### 3.1.3. Setor de Fabrico

O setor de fabrico da CFF, é constituído por um robô manipulador da *ABB*, do modelo *IRB140*, uma fresadora *EMCO MILL 155*, um torno *EMCO MILL 55* e os *buffers*, como apresentado na Figura 3-8. A função deste setor é o processamento das matérias-primas vindas do setor de armazenamento, sendo estas colocadas temporariamente nos *buffers*.

Cada máquina está programada para maquinar um tipo de peça de trabalho, o torno para maquinar peças cilíndricas e a fresadora para maquinar peças prismáticas.



Figura 3-8: Setor de Fabrico.

### 3.1.3.1. Robô IRB140

O robô *IRB140* da *ABB*, utilizado no setor de fabrico, apresenta as seguintes características:

- seis graus de liberdade/seis eixos de rotação;
- estrutura antropomórfica;
- punho esférico ( $\theta_4$  - *roll*,  $\theta_5$  - *pitch*,  $\theta_6$  - *roll*);
- alcance máximo de 0,81m;
- capacidade de carga máxima de 6 Kg;
- precisão de trajetória de  $\pm 0,02\text{mm}$ ;
- repetibilidade de  $\pm 0,03\text{mm}$ .

Na Figura 3-9, pode observar-se o volume de trabalho do robô e algumas das características referidas [42].

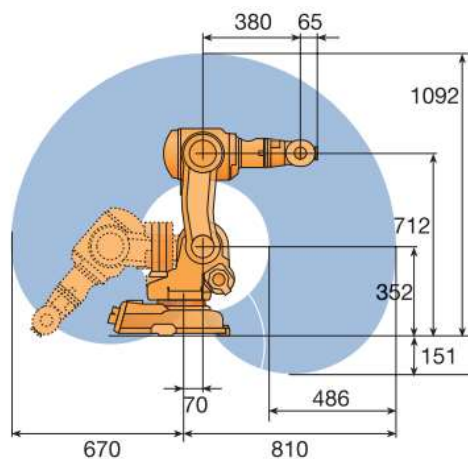


Figura 3-9: Robô IRB140 da ABB [42].

O robô *IRB140* tem como função carregar/descarregar as máquinas CNC (torno e fresadora) no setor de fabrico da CFF. Outra das funções que este robô tem é efetuar a transferência das peças (matérias-primas e produtos acabados) para os *buffers* e para as paletes que circulam no transportador.

### 3.1.3.2. Ferramenta de Trabalho

O funcionamento da pinça pneumática, apresentada na Figura 3-10, é semelhante à pinça instalada no robô *IRB1400* do setor de armazenamento, apresentado no subcapítulo 3.1.1.2. Contudo, esta ferramenta não tem as duas fibras óticas iguais às que estão na ferramenta do robô *IRB1400*.

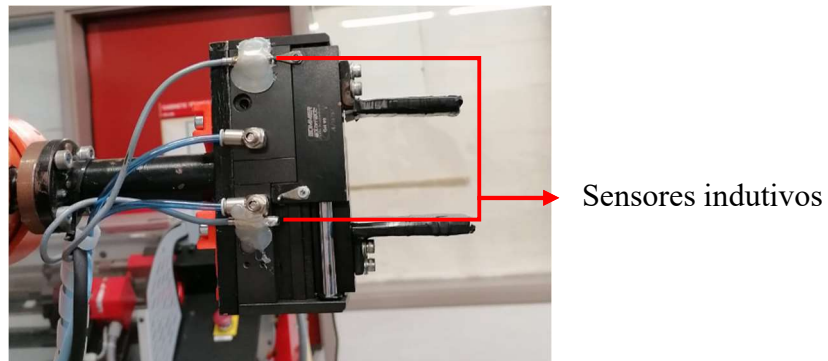


Figura 3-10: Pinça do robô IRB140.

### 3.1.3.3. Torno CNC

O torno CNC que se encontra no setor de fabrico da CFF, como apresentado na Figura 3-11, efetua operações em dois eixos (x e z) e é uma máquina que executa operações de contorno, desbaste e perfuração, tendo como função na CFF, a maquinação de peças cilíndricas.



Figura 3-11: Torno EMCO Concept Turn 55.

Este pode ser automatizado individualmente, como também se pode interligar a uma CFF através de uma interface robótica. O torno *EMCO Concept Turn 55*, contém um *chuck* pneumático, que permite que a peça a ser maquinada fique fixa e uma porta automática.

A interface robótica desenvolvida permite o controlo e a integração do torno CNC no setor de fabrico da CFF, por exemplo, no caso da porta automática, esta recebe informação para abrir, quando se encontra completamente aberta o torno envia um sinal a confirmar que a porta se encontra nessa posição, podendo assim o robô deslocar-se para o torno em segurança.

#### 3.1.3.4. Fresadora CNC

A fresadora CNC que se encontra no setor de fabrico da CFF, como apresentado na Figura 3-12, efetua operações em três eixos (x, y e z), é constituída por um sistema de controlo automatizado, por um *Vice* pneumático que é o responsável para fixar a peça a maquinar. O controlo e a integração da fresadora CNC na CFF só foi possível graças à interface robótica desenvolvida. Um dos exemplos desta interface, é o caso do *Vice*, este recebe informação para abrir, de seguida a fresadora envia um sinal, a confirmar que o *Vice* abriu corretamente, podendo, assim, o robô em segurança executar o movimento da colocação da peça a maquinar no respetivo ponto.



Figura 3-12: Fresadora CNC EMCO Concept MILL 155.

### 3.1.3.5. Buffers

No centro do setor de fabrico, para otimizar o processo de produção, foram instalados numa mesa quatro *buffers*, destinados ao armazenamento de matérias-primas e produtos acabados, como se pode verificar na Figura 3-13. Do lado esquerdo surgem os *buffers* que correspondem à fresadora do lado direito os correspondentes ao torno.

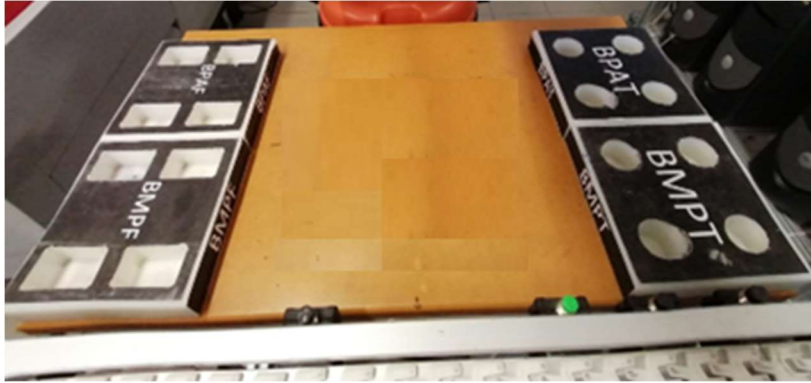


Figura 3-13: Buffers.

A designação de cada um destes *buffers* visualizados na figura anterior são:

- BPAF ---- Buffer Produtos Acabados Fresadora;
- BMPF ---- Buffer Matéria-Prima Fresadora;
- BPAT ---- Buffer Produtos Acabados Torno;
- BMPT ---- Buffer Matéria-Prima Torno.

Os *buffers* das matérias-primas, têm como objetivo ter sempre disponíveis peças para serem maquinadas, rentabilizando, desta forma, a utilização das máquinas, já os *buffers* dos produtos acabados servem para armazenar, temporariamente, as peças maquinadas nas máquinas.

### 3.1.4. Sinalização

O sistema de sinalização na CFF é constituído por duas balizas de sinalização, compostas por elementos luminosos e sonoros, permitindo, assim, ao operador observar se tudo funciona corretamente nos setores de fabrico e armazenamento. Na Figura 3-14, pode perceber-se uma das balizas de sinalização utilizadas na CFF.



Figura 3-14: Baliza de sinalização.



## 4. TRABALHO DESENVOLVIDO

Neste capítulo é feita uma introdução ao *software* utilizado para desenvolver este projeto, o *RobotStudio* da *ABB*. Explicar-se-à como se criam as diversas *Smart Components*, como pode fazer-se a modelação dos vários objetos da CFF, bem como realizar a programação *off-line* e como usar o controlador para que este permita simular o ambiente real.

### 4.1. Software RobotStudio

O *RobotStudio* é um *software* que nos permite, fazer a simulação de células, a modelação e a programação *off-line*, sendo este, exclusivamente, usado para simulação dos robôs da marca *ABB*. A grande vantagem deste *software* é poder simular células reais, sem precisar de estar fisicamente no espaço em que se encontram, pois, o *software* tem incorporado um controlador, *IRC5*, que nos permite controlar o robô virtual, tal como acontece num ambiente físico.

Este *software* tem vários modelos de robôs da *ABB*, sendo apenas necessário escolher o modelo adequado ao nosso trabalho, tem ainda uma biblioteca com alguns equipamentos mais usados nestes ambientes de CFF. Na Figura 4-1, pode visualizar-se o ambiente de trabalho deste *software* com um dos robôs escolhido.

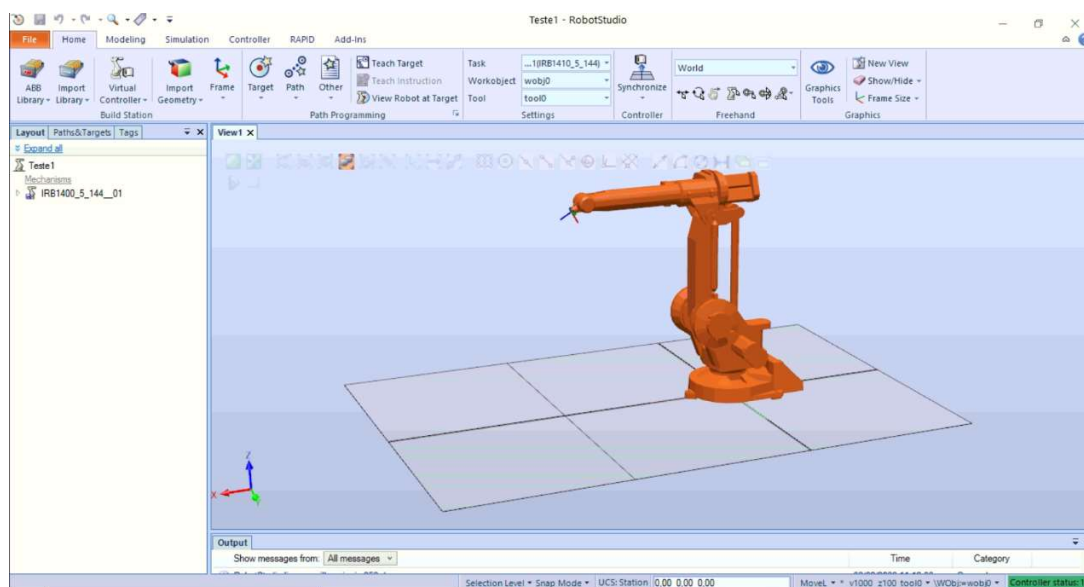


Figura 4-1: Ambiente de trabalho do RobotStudio.

O programa possui ainda mais funcionalidades para operações específicas, que são úteis para o desenvolvimento deste tipo de trabalhos, tais como: importação de ficheiros CAD; criação de objetos CAD; geração automática de trajetórias; verificação de alcances; deteção de colisões; integração com um controlador virtual; criação de *Add-ins* e criação de *Smart componentes* (SC), sendo estes possíveis de criar através de programação em C#, ou através da integração de elementos SC base, como foi realizado neste trabalho.

#### 4.1.1. Smart Components

Como referido, anteriormente, a criação de SC, no trabalho, foi elaborada por integração de elementos SC, elementos estes que já vêm definidos de base, já que o objetivo destes é executar funções lógicas. Os conjuntos base de SC, estão divididos por grupos, sendo que os mais utilizados foram: Sinais e Propriedades, Sensores, Ações, Manipuladores. Além destes SC, também foram utilizados outros, mas não foram determinantes no projeto.

No grupo Sinais e Propriedades encontramos SC, que permitem fazer operações e expressões lógicas através das propriedades já existentes com a junção dos sinais, sendo possível, também, converter variáveis analógicas em digitais e vice-versa, ainda se observam, neste grupo disponível, temporizadores e sinais de relógio para definir intervalos. Na Figura 4-2 apresenta-se um exemplo de uma SC do grupo Sinais e Propriedades.



Figura 4-2: Exemplo de AND.

No grupo dos Sensores, há SC que fornecem informação das interações, estes sensores podem ser de colisão, de linha, de plano e até mesmo de volume, como apresentado Figura 4-3. Por exemplo, para o funcionamento do sensor de colisão é necessário escolher dois objetos e quando estes se intersetarem o sensor é ativo, como se pode verificar na Figura 4-4.



Figura 4-3: Grupo de Sensores.

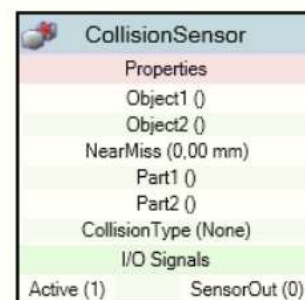


Figura 4-4: SC do sensor de colisão.

No respetivo grupo de ações, os SC mais importantes são os que nos possibilitam anexar e desanexar objetos, através do SC *Attacher* e do SC *Dettacher*. Na Figura 4-5, pode ver-se o bloco SC *Attacher*.

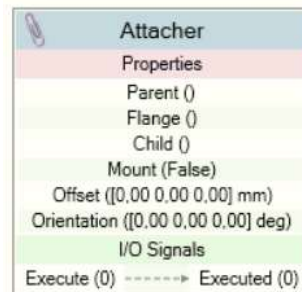


Figura 4-5: SC Attacher.

Nos manipuladores, existem SC que permitem recriar movimentos lineares, em rotação e em posição, ou seja, escolhemos um objeto e indica-se o movimento que ele vai realizar, como apresentado na Figura 4-6.

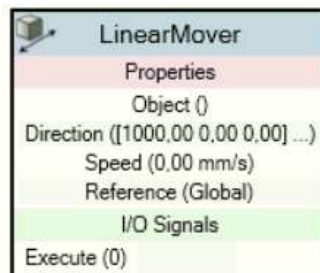


Figura 4-6: SC Movimento Linear.

#### 4.1.2. Simulação

No separador da simulação encontra-se o *Event Manager* que permite criar ações, através de um evento, isto é, quando é preciso realizar-se uma determinada ação para uma entrada ou uma saída do robô. Por exemplo, quando a entrada AS1 for a 1 pretende-se que o objeto fique visível. Para o objeto ficar oculto, é necessário colocar a entrada AS1 a 0, como observado na Figura 4-7.

Activation	Trigger Type	Trigger System	Trigger Name	Trigger Parameter	Action Type	Action System	Action Name	Action Parameter
On	I/O	IRB1410_5_144	AS1	1	Show/Hide ...	Show/Hide Object	Show/Hide Object	Show: IRB1400_5_144__01
On	I/O	IRB1410_5_144	AS1	0	Show/Hide ...	Show/Hide Object	Show/Hide Object	Hide: IRB1400_5_144__01

Figura 4-7: Exemplo de ações criadas para a entrada AS1.

Dentro da ação criada para a entrada AS1, quando esta for a 1 é possível fazer uma sequência de mais ações, ou seja, para além do objeto estar visível, ainda é possível anexar outro objeto ao já existente. A Figura 4-8 mostra a sequência de uma ação para a entrada AS1.

Seq. No	Action
1	Show/Hide Object
2	Attach Object
3	Move Object

Figura 4-8: Sequência de uma ação para a entrada AS1.

### 4.1.3. Modelação

O *software* do *RobotStudio* permite fazer a modelação de geometrias de diferentes objetos de uma célula, como apresentado na Figura 4-9, sendo que alguns objetos, mais complexos, têm de ser modelados no *software* de CAD.

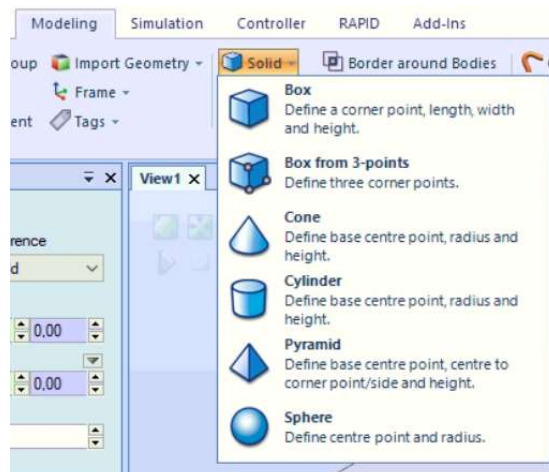


Figura 4-9: Modelação de objetos no RobotStudio.

As modelações obtidas noutros programas de desenho tridimensional podem ser importadas para o *RobotStudio*, como observado na Figura 4-10.

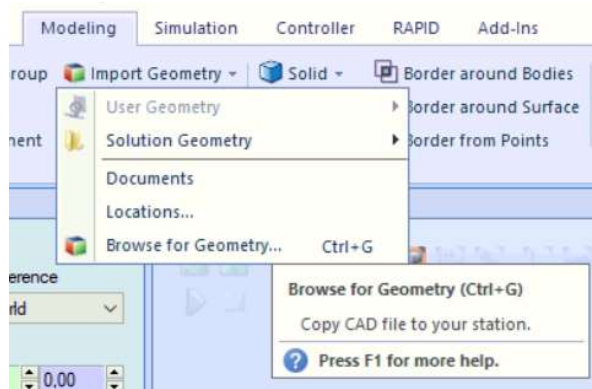


Figura 4-10: Importar ficheiro desenhado em CAD.

Para além de ser possível criar objetos de trabalho, é possível interseccionar, subtrair e unir objetos, para que a partir de dois objetos seja criado apenas um único objeto. Ainda neste ambiente de modelação, conseguimos fazer medições, mover cada peça adicionada ao *layout*, criar grupos de componentes, bem como criar mecanismos como, por exemplo uma pinça pneumática.

#### 4.1.4. Controlador

No separador do controlador é possível, adicionar o controlador para o robô escolhido, fazer as configurações das entradas e das saídas do robô, criam-se todos os sinais que o robô recebe e envia, podendo estes ser digitais ou analógicos. A Figura 4-11 mostra a forma de criar sinais no controlador do robô.

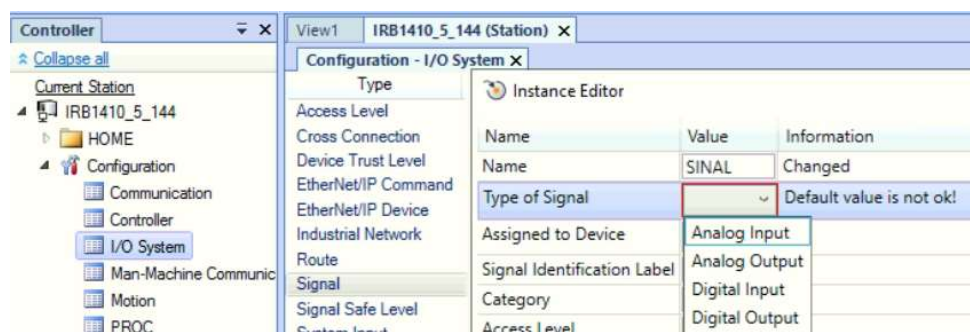


Figura 4-11: Criação dos sinais no controlador do robô.

Ainda neste separador, é possível abrir a consola virtual a *IRC5 FlexPendant*, como apresentado na Figura 4-12, que permite, em modo *offline*, modificar certos dados que foram criados durante a programação e que precisam de ser alterados.

Com a consola virtual é possível movimentar o robô virtualmente, gravar pontos para serem utilizadas nas trajetórias do robô, entre outras coisas.

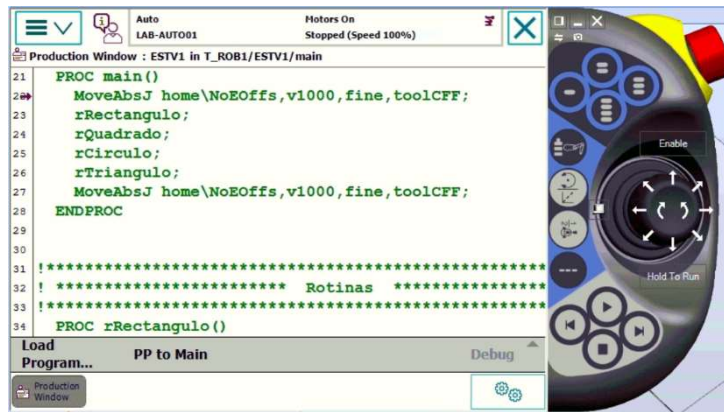


Figura 4-12: Consola de programação virtual do RobotStudio.

#### 4.1.5. Separadores de Programação

No *RobotStudio* são criadas no separador principal as pastas de programação, concebem-se os pontos e as trajetórias que se pretende que o robô execute, sendo possível observar a trajetória desenhada enquanto estamos a programar. A Figura 4-13 apresenta as pastas de programação das trajetórias do robô.

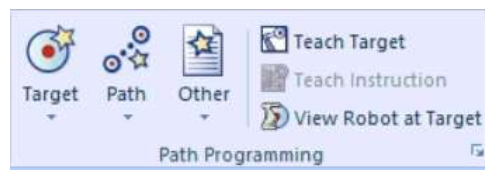


Figura 4-13: Pastas de programação de trajetórias.

Na parte do separador do *RAPID*, ver Figura 4-14, conseguimos desenvolver todo o código necessário para o crescimento de um projeto, sendo possível verificar se existe algum erro no programa criado, bem como correr o programa completo, ou partes do mesmo.

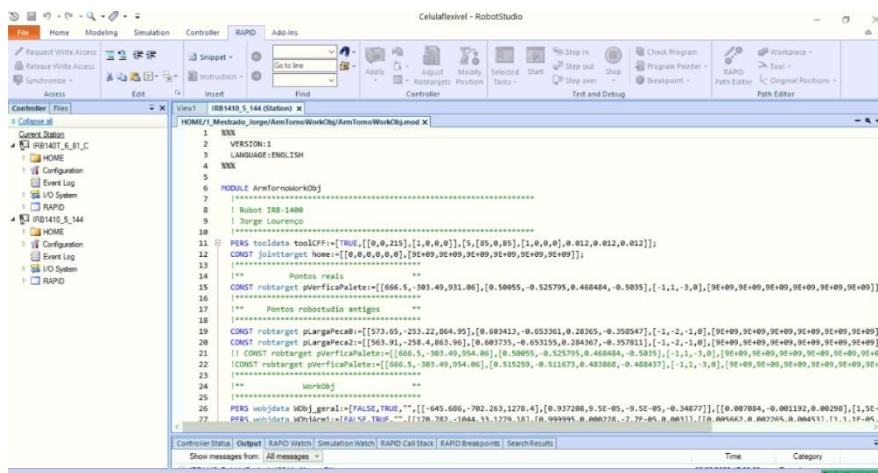


Figura 4-14: Editor *offline* do programa RAPID.

## 4.2. Desenvolvimento da Célula da ESTGV em modo virtual

Neste subcapítulo é apresentado o objetivo principal deste trabalho, que se destaca pela modelização de toda a CFF do DEE, descrita no capítulo 3, e pela simulação da mesma. Para simular e modelar toda a CFF, foi utilizado o *software RobotStudio* da ABB.

### 4.2.1. Modelação da Célula Flexível de Fabrico

O trabalho prático desenvolvido, iniciou-se pela parte da modelação de todo o ambiente real existente na CFF, para ser possível visualizar, em modo virtual, todo o funcionamento da célula. Para ser exequível, teve de ser feito um levantamento de todas as medidas dos diversos componentes existentes no laboratório de Automação e Robótica do DEE, isto para se obter uma modelação praticamente perfeita, sendo que tudo foi desenhado à escala de 1, ou seja, as medidas virtuais correspondem exatamente às reais.

Posto isto, começou-se por desenvolver todo o setor de armazenamento, onde se criou o modelo da pinça pneumática, como apresentado na Figura 4-15.

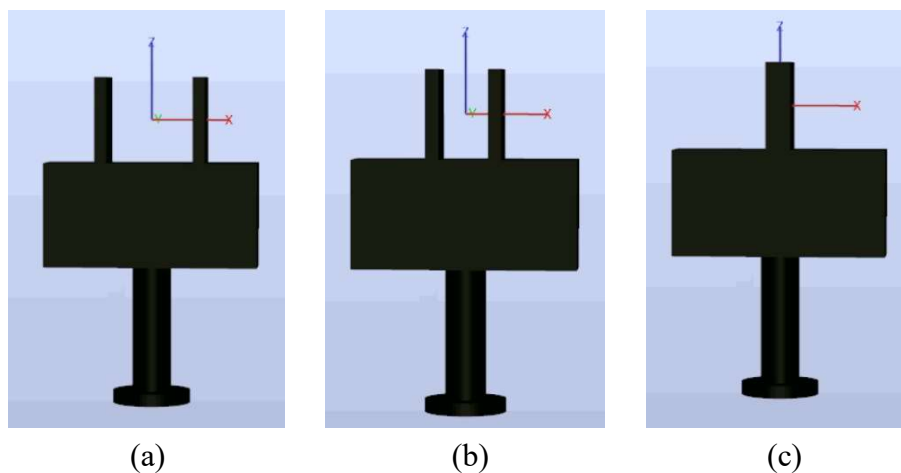


Figura 4-15: Pinça Pneumática virtual.

Para além da modelação da pinça pneumática, foi criado um mecanismo para que fosse praticável a pinça ter três posições, como apresentado na Figura 4-16, sendo elas *HomePosition* (a), *ComPeca* (b) e *SyncPosition* (c), como pode visualizar-se na Figura 4-15.

Edit poses - Pinça

Name	Joint values
HomePosition	27,50 27,50
SyncPosition	0,00 0,00
ComPeca	15,00 15,00

Figura 4-16: Mecanismo da pinça com 3 posições.

Com a pinça concluída foi desenvolvido o modelo 3D das duas peças de matérias-primas, a peça cilíndrica e a peça paralelepipedal, como apresentado na Figura 4-17, que serviram para desenvolver todo o processo criado, sendo estas colocadas, inicialmente nos armazéns.

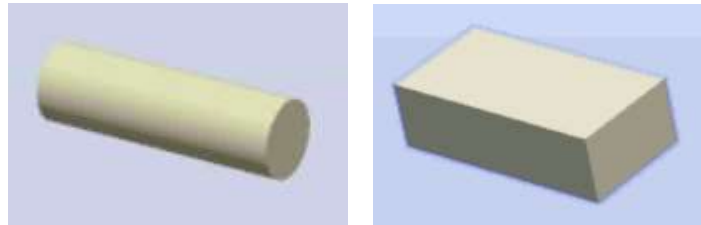


Figura 4-17: Modelo 3D das Matérias-Primas.

Em todo o progresso da estrutura dos armazéns houve o cuidado com cada pormenor desde as inclinações em cada fileira, até ao espaçamento entre as várias colunas de cada fila, isto para que os pontos de todo o armazém da CFF real correspondessem ao modelo virtual.

Na Figura 4-18 a) pode ver-se o armazém ainda em fase de modelação, existindo apenas o suporte da estrutura que foi desenvolvido no programa *3D Builder*, a partir de um molde do perfil 45\*45. Depois de exportado o suporte da estrutura para o *RobotStudio* em forma de geometria, foi executado neste a modelação de cada linha do armazém. O resultado completo do armazém é apresentado na Figura 4-18 b), já com alguma matéria-prima distribuída pelo armazém.

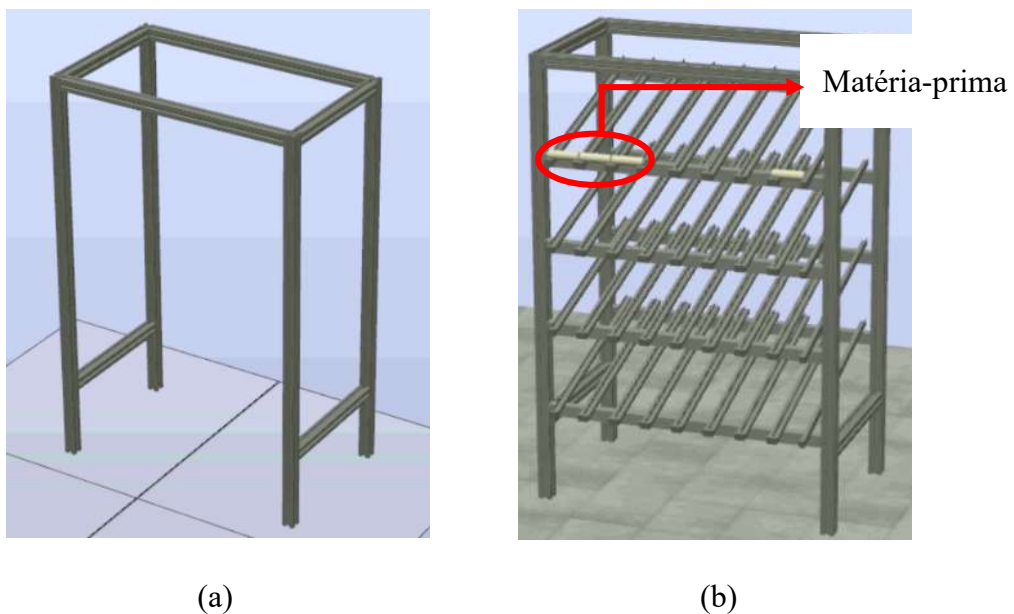


Figura 4-18: Modelo do Armazém.

No setor de armazenamento encontram-se, também, a rampa de descarga, que serve para a colocação dos produtos acabados, provenientes do torno e da fresadora, para tal, foi desenhada a rampa apresentada na Figura 4-19, que irá estar afixada ao setor de transporte.

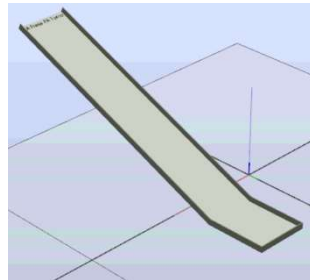


Figura 4-19: Molde da rampa de descarga.

Para concluir o setor de armazenamento, foi desenhada uma base para o robô ser afixado exatamente à mesma altura, daquela que está no laboratório.

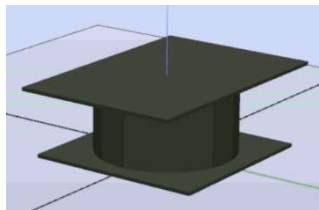


Figura 4-20: Base do IRB1400.

Estando o setor todo modelado, foram, de seguida, adicionados todos os componentes à célula virtual, sendo que o robô *IRB1400* e a lâmpada de sinalização, já fazem parte das bibliotecas que se encontram disponíveis no *RobotStudio*. Pode observar-se na Figura 4-21 o setor de armazenamento, sendo este uma cópia fiel daquele que se encontra no laboratório, como apresentado na Figura 3-2, sendo possível ver as semelhanças do resultado.

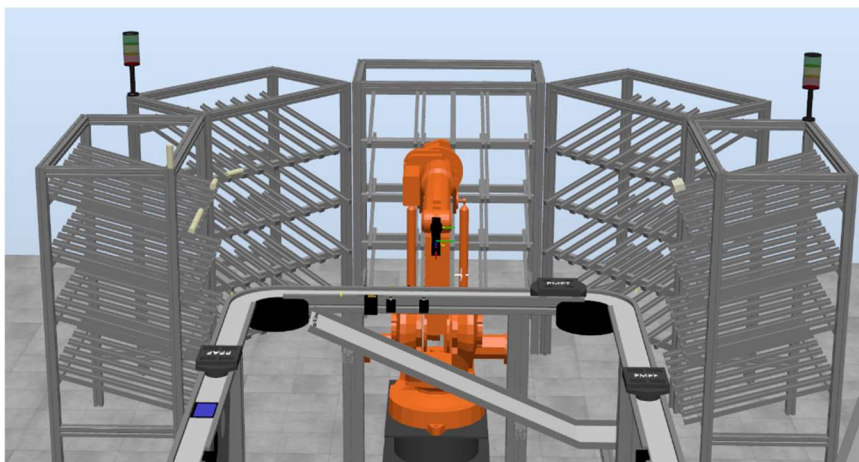


Figura 4-21: Setor de Armazenamento em modo virtual.

Com o setor de armazenamento concluído, decidiu-se avançar para o desenvolvimento do setor de transporte, para se poder fazer os testes de carregamento das paletes da matéria-prima. Este setor começou por ser modelado num programa 3D a partir de um molde do perfil 45\*45, onde foi esquematizado o suporte da estrutura do transportador, com se pode ver na Figura 4-22. De seguida, este modelo foi exportado para o *RobotStudio*, para se poder utilizar as funções deste programa e efetuar todo o transportador.

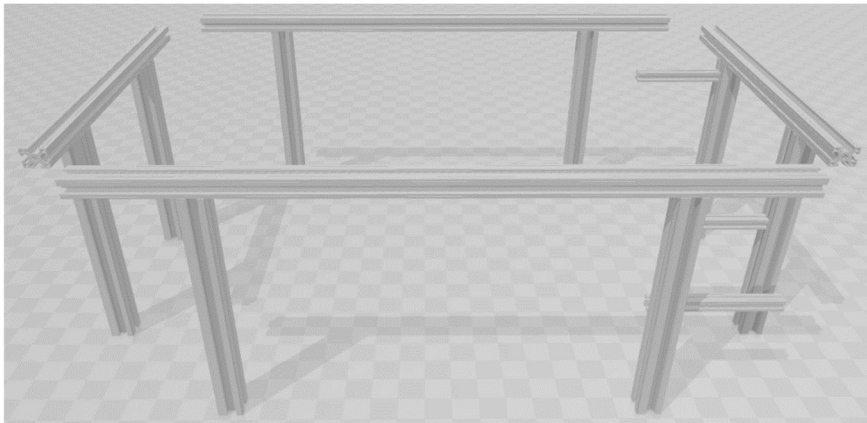


Figura 4-22: Modelo 3D da estrutura do setor de transporte.

Exportando o modelo para o *RobotStudio*, a partir deste, foram criadas todas as calhas e tapetes para que as paletes pudessem rodar neste setor. Foram desenhados sensores indutivos para identificarem o tipo de palete que circula no transportador. As paletes são compostas por umas peças metálicas na parte inferior, conferindo um código diferente a cada palete. Através deste código, as paletes são identificadas pelo sistema. O *stopper* permite a paragem da paleta para o robô *IRB1400* efetuar o carregamento/descarregamento da mesma. Na Figura 4-23 podemos verificar o *stopper* e os sensores indutivos utilizados neste projeto.

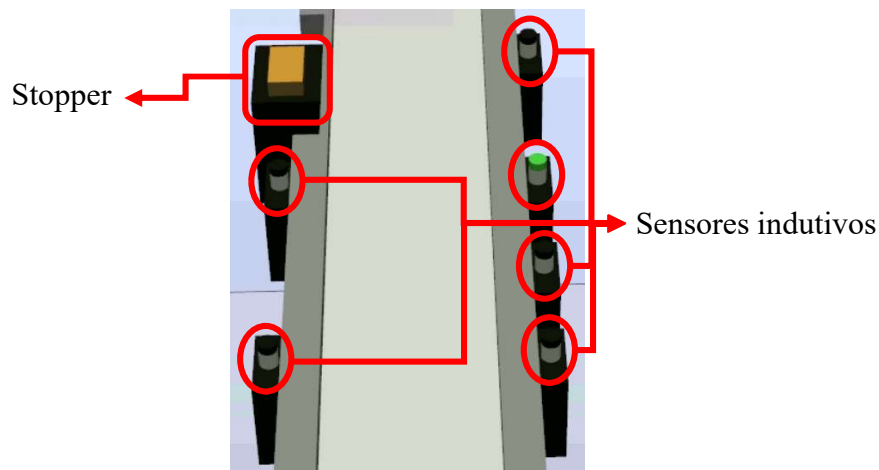


Figura 4-23: Sensores indutivos e Stopper.

Foram criadas quatro paletes para fazerem o transporte da MP e dos PA, como se pode observar na Figura 4-24. Para as paletes serem fáceis de identificar foi atribuído um nome a cada uma delas no *software 3D*.

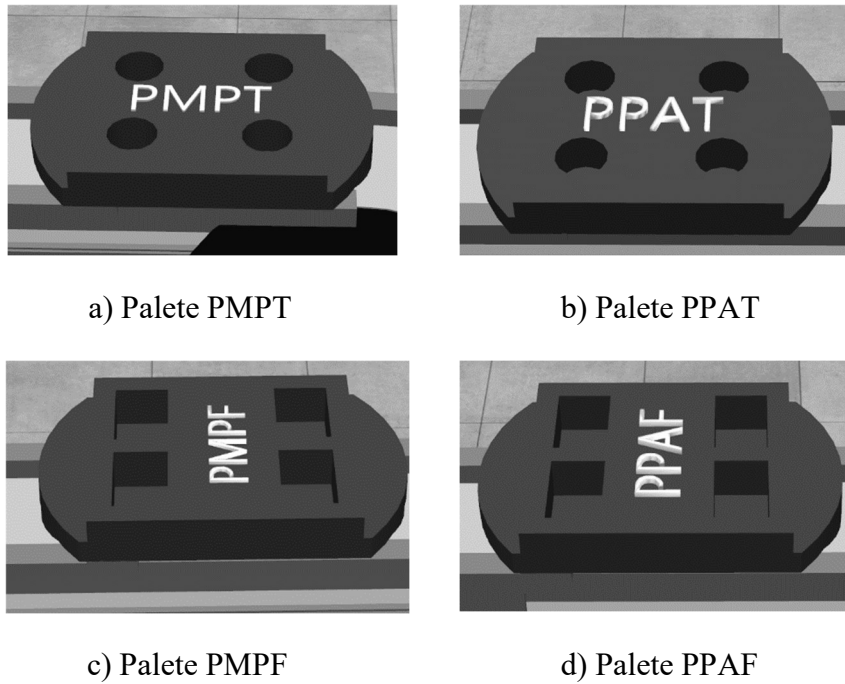


Figura 4-24: Tipos de paletes de MP e PA.

Para os robôs conseguirem identificar cada pallet representada na Figura 4-24, foi adicionado o código de peças metálicas a cada uma delas, como apresentado na Figura 4-25. A Pallet PPAF tem apenas uma peça, a PMPF tem duas peças, a pallet PPAT tem três peças e, por fim, com as quatro peças metálicas a pallet PMPT.

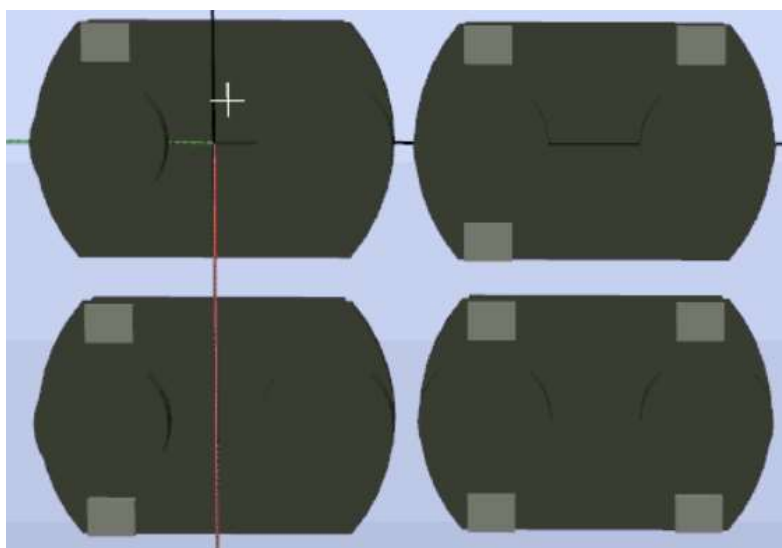


Figura 4-25: Códigos de cada Pallet.

Com todos os componentes do setor de transporte modelados, estes foram adicionados, à estrutura inicial, bem como a rampa PA do setor de armazenamento. O resultado obtido para este setor pode ser observado na Figura 4-26, estando este muito semelhante ao real, como apresentado na Figura 3-6.

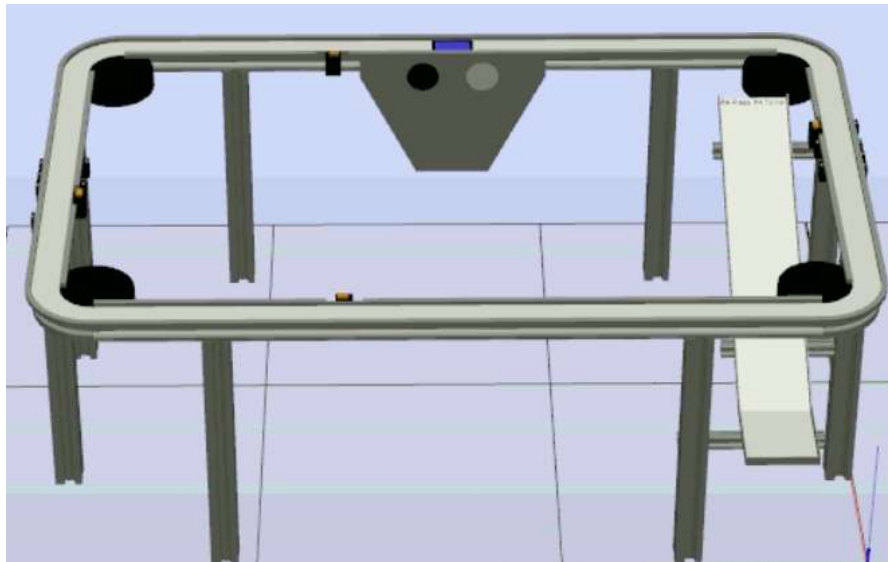
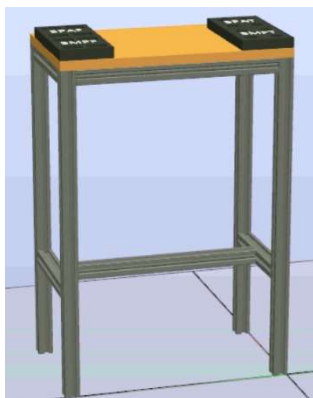


Figura 4-26: Setor de transporte em modo virtual.

Com a CFF praticamente desenvolvida em modo virtual, para esta ficar idêntica à real, falta modelar o setor de fabrico onde se encontram as máquinas CNC e os *buffers*. Assim, começou-se por dar prioridade aos *buffers*, que servem para o armazenamento de MP e dos PA. Foi desenhado no *software 3D Builder*, uma mesa para servir de estrutura aos buffers, como se pode ver na Figura 4-27 a), de seguida, exportou-se para o *RobotStudio* sendo desenhada a tampa da mesa e os quatro *buffers*, como apresentado na Figura 4-27 b), tendo cada um dos *buffers* uma determinada função.



a)



b)

Figura 4-27: Buffers em modo virtual.

Posteriormente, foi modelado o torno CNC, tendo sido criada uma estrutura para servir de base, à máquina. O torno CNC tem um *chuck* que abre e fecha. O *chuck* é o dispositivo que segura a peça durante a fase de maquinação da mesma. Também foi desenhada uma porta para o torno, semelhante à que existe na máquina que se encontra no laboratório. Foram adicionadas umas luzes de sinalização para, quando forem feitos os testes *offline*, deste modo, percebe-se se a máquina naquele momento está a trabalhar ou parada. A Figura 4-28 apresenta o modelo virtual desenvolvido para o torno CNC.

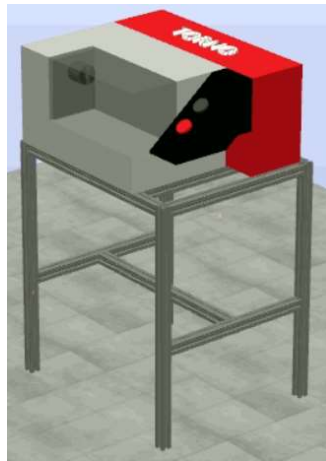


Figura 4-28: Modelo virtual da estrutura e do torno.

Para o torno executar todas as suas funcionalidades, foram criados vários eventos. De acordo com o sinal de cada funcionalidade, o evento criado faz com que a porta abra ou feche, que a luz do torno de *start* e *stop* altere, bem como o *chuck* abrir e fechar. Na Figura 4-29 pode ver-se o exemplo da ação criada para a entrada (DO10\_4). Quando esta saída digital for a um, a porta do torno vai para a posição de abertura. Este processo foi repetido para todas as restantes funcionalidades do torno.

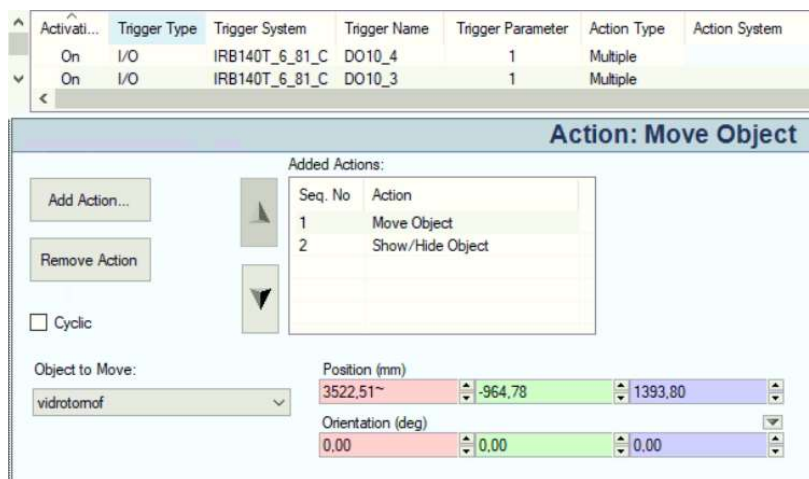


Figura 4-29: Ação criada para abrir a porta do torno.

Todas as ações criadas para mostrar as funcionalidades do torno podem ser visualizadas na Figura 4-30. Na Figura 4-30 a) o torno encontra-se parado, na Figura 4-30 b) observa-se o torno com a porta e *chuck* abertos, estando a máquina pronta a maquinar. Por último, na Figura 4-30 c) temos o torno com o *chuck* e a porta fechados, estando a máquina a maquinar a peça.

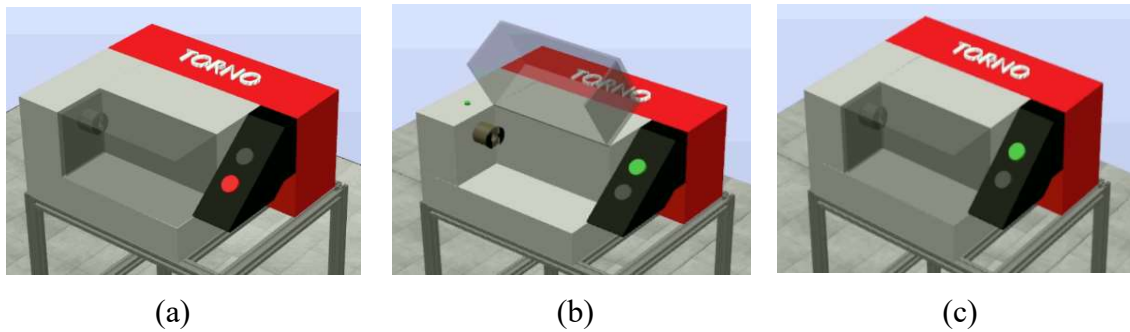


Figura 4-30: Configurações do Torno CNC.

Concluída a primeira máquina CNC, foi de seguida modelada a segunda máquina CNC da CFF, a fresadora. Foi modelada toda a estrutura da mesma, o *vice* onde vai ser colocada a peça para ser maquinada, a porta de segurança da máquina e o *display* onde se pode fazer o controlo de abertura e fecho do *vice* e da porta, quando a máquina está parada ou quando está a maquinar. A Figura 4-31 a) apresenta o modelo virtual da fresadora CNC no *software RobotStudio*. Na Figura 4-31 b) pode observar-se a parte mais importante da fresadora para este trabalho.

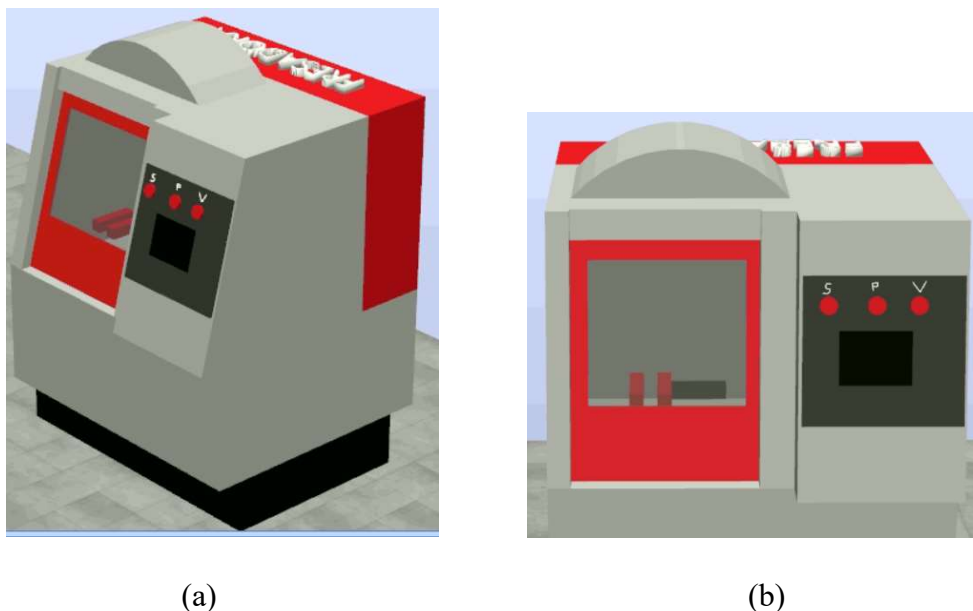


Figura 4-31: Modelo virtual da Fresadora CNC.

Tal como torno, na fresadora também houve a necessidade criar vários eventos. De acordo com o sinal de cada funcionalidade, o evento criado abrir e fechar a porta, liga e desliga a luz de *start* e *stop*, bem como abrir e fechar o *vice*. Na Figura 4-32 pode ver-se o exemplo da ação criada para a entrada (DO10\_16). Quando esta entrada for a um, a porta da fresadora vai abrir. Para as restantes funcionalidades da fresadora, foram implementadas ações equivalentes a esta.

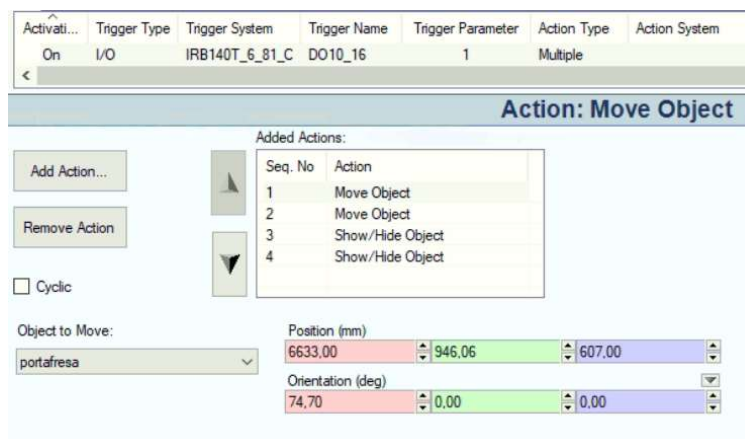


Figura 4-32: Ação criada para a abrir a porta da fresadora.

Todas as ações produzidas para mostrar as funcionalidades da fresadora podem ser visualizadas na Figura 4-33, sendo que para se entender as diferenças de cada imagem existe uma combinação de cores no *display* da fresadora, como apresentado na Figura 4-34. O primeiro botão corresponde *start/stop*, ficando com a cor verde quando a máquina está em *start* e a vermelho quando está em *stop*. O segundo botão (P), quando se encontra a verde significa que a porta está aberta e a vermelho quando está fechada. O terceiro botão (V) está associado ao *vice*, a vermelho quando este está fechado e a verde quando se encontra aberto.

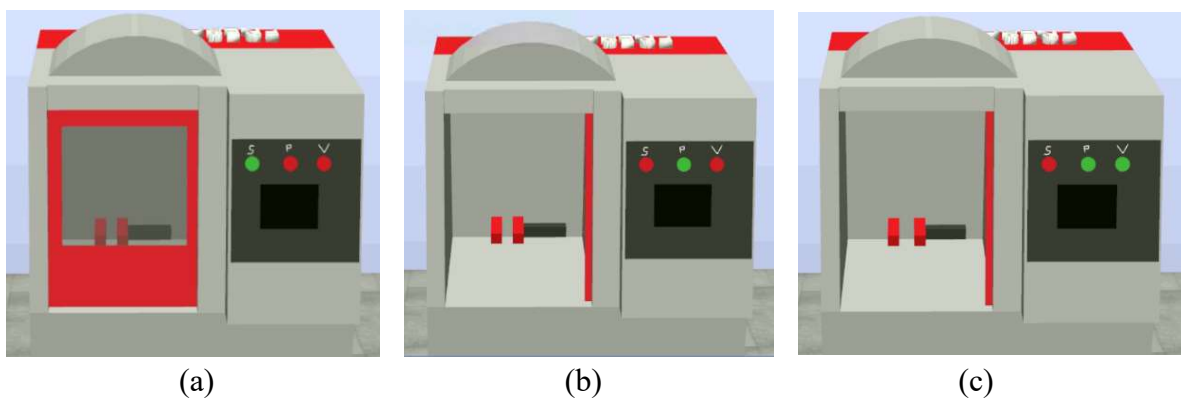


Figura 4-33: Configurações da Fresadora CNC.

A Figura 4-33 a) apresenta a fresadora em maquinação, a Figura 4-33 b) representa a fresadora parada (com a porta aberta e o *vice* fechado), por último, na Figura 4-33 c), observa-se a fresadora parada (com o *vice* e a porta abertos).

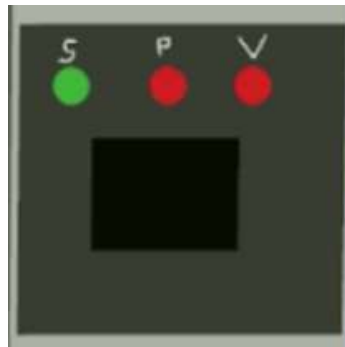


Figura 4-34: Combinação da fresadora em maquinação.

Para o setor ficar completamente modelado, foi ainda necessário desenhar uma base, como apresentado na Figura 4-35. Assim, o robô *IRB140* ficará exatamente à mesma altura, daquela que o robô real se encontra na realidade.

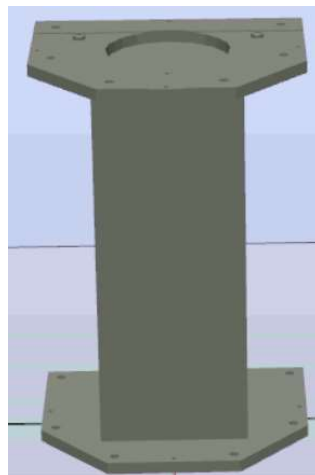


Figura 4-35: Base do robô IRB140.

Estando o setor de fabrico todo modelado, foram adicionados todos os componentes à célula virtual, sendo que o robô *IRB140* e a lâmpada de sinalização, já fazem parte das bibliotecas que se encontram no *RobotStudio*. Para a pinça pneumática foi criada uma biblioteca, sendo apenas necessário agrupá-la ao *IRB140*. A Figura 4-36 apresenta o setor de fabrico, tendo este as mesmas medidas da CFF real, como apresentado na Figura 3-8, onde é possível ver as semelhanças do resultado real com o modelo criado no *software*.

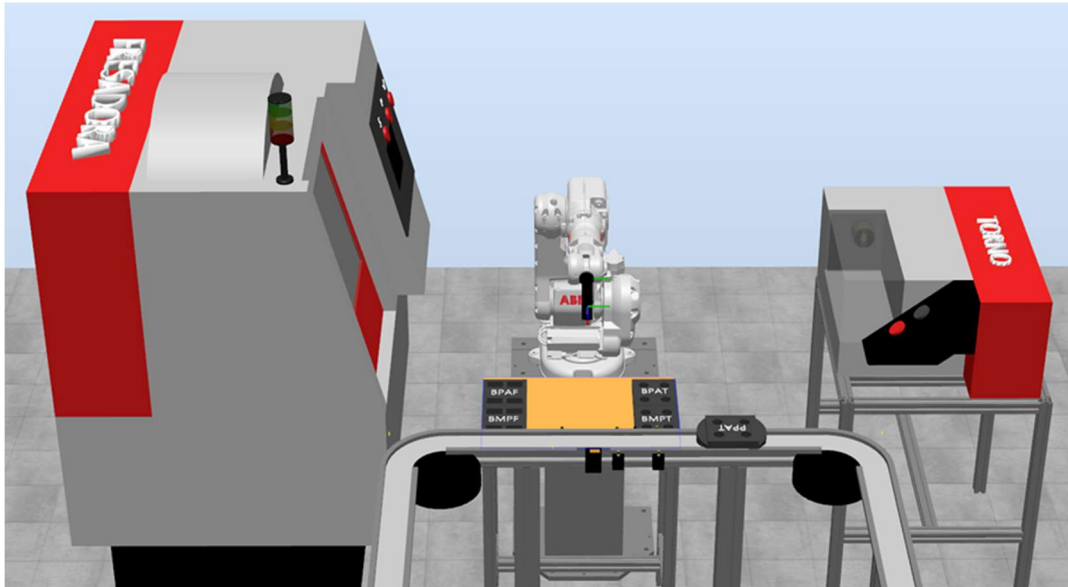


Figura 4-36: Setor de fabrico em modo virtual.

Concluído todo o processo de modelação da CFF, foi criada a célula flexível de fabrico virtual, pelo que foram adicionados os três setores num único ambiente de trabalho. Assim, conseguimos alcançar o objetivo principal, desenvolver a CFF do DEE em modo virtual no *RobotStudio*. As Figuras 4-37 e 4-38 mostram as duas vistas diferentes da célula virtual criada no *RobotStudio*.

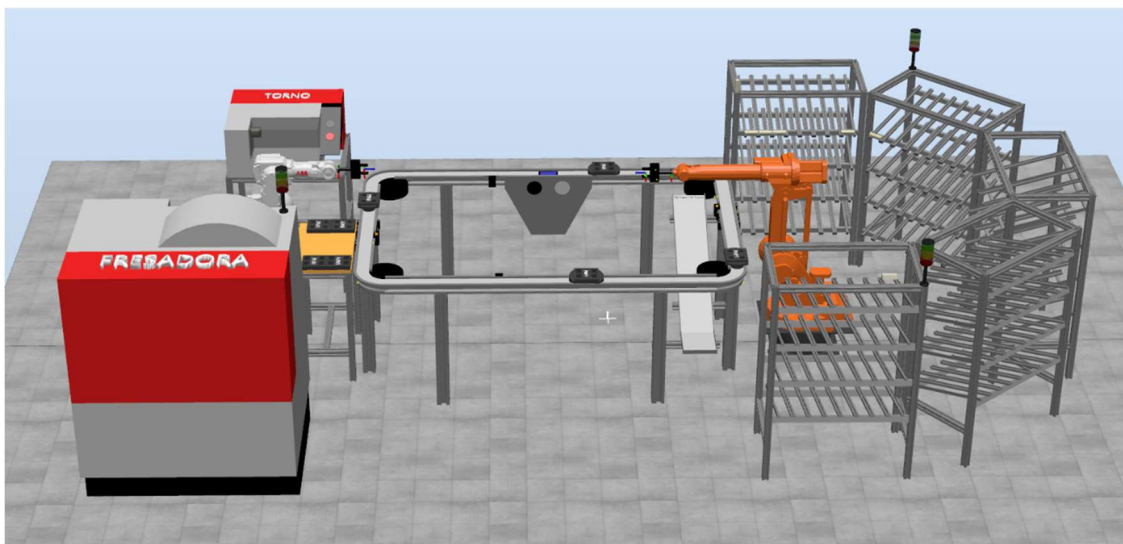


Figura 4-37: Célula virtual, vista do lado da fresadora.

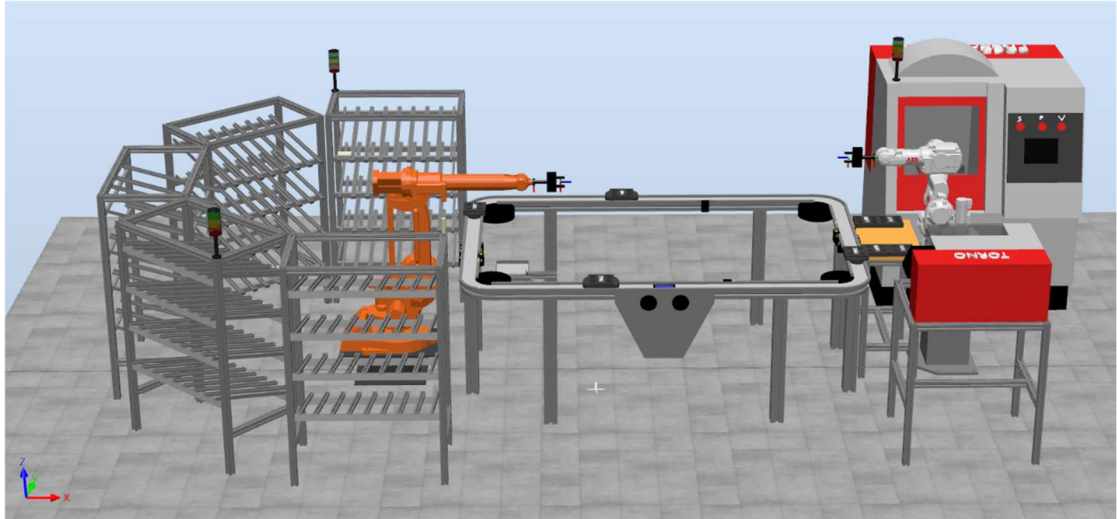


Figura 4-38: Célula virtual, vista do lado do torno.

## 5. ANÁLISE DOS RESULTADOS

Neste capítulo serão explicadas as partes mais importantes da programação dos três setores da CFF. No setor de armazenamento e fabrico foi desenvolvido e implementado um programa, desenvolvido em *RAPID*. No setor de transporte foram criadas funções lógicas para este funcionar e conseguir-se interligar aos outros dois setores [43].

Por norma, quando se utiliza o *software RobotStudio* da *ABB*, para se desenvolverem os movimentos necessários para o robô executar determinada função, são criadas no separador principal, como se pode ver na Figura 4-13, as chamadas pastas de programação. Dentro de uma pasta são adicionados os pontos para onde se pretende que o robô se movimente, escolhendo o movimento que queremos que este execute, tempos de espera, entre outras ações. Posto isto, é criada uma trajetória automaticamente onde se consegue ver a trajetória por onde o robô passa. Para se poder simular o código, tem de se recorrer ao separador do *RAPID* e adicionar a pasta criada a um ficheiro para o controlador o conseguir ler.

Assim, chega-se à conclusão que este tipo de programação não será a mais aconselhada, visto ter de se realizar alterações para o modelo programado, para o controlador conseguir executar determinado programa na célula real. Com isto, decidiu-se criar toda a programação semelhante àquela que é feita para os robôs reais, sem a necessidade de criar as pastas para determinadas trajetórias. Assim, a programação desenvolvida funciona tanto na célula virtual como na real.

### 5.1. Programação e simulação do Setor de Armazenamento

Neste setor foi desenvolvido um programa na linguagem *RAPID*, o robô recebe a informação da paleta de MP para carregar e para carregar a paleta de MP do torno apenas se movimenta nos armazéns 1 e 2, e para a MP da fresadora o robô utiliza os armazéns 3 e 4.

No caso de ser a entrada (DI10\_12) ir a 1, significa que paleta da MP do torno está em condições de ser carregada, posto isto, o robô inicia o código que corresponde ao carregamento desta paleta. Assim, o robô *IRB1400* vai se deslocar para o armazém 1, começando a procura das peças na primeira fila. Caso este detete uma peça, vai carregá-la na paleta, como pode observar-se na Figura 5-1 a), se não detetar nenhuma peça, como apresentada a Figura 5-1 b), este passará para a segunda coluna da primeira fila e assim, repetidamente até ao fim da fila.

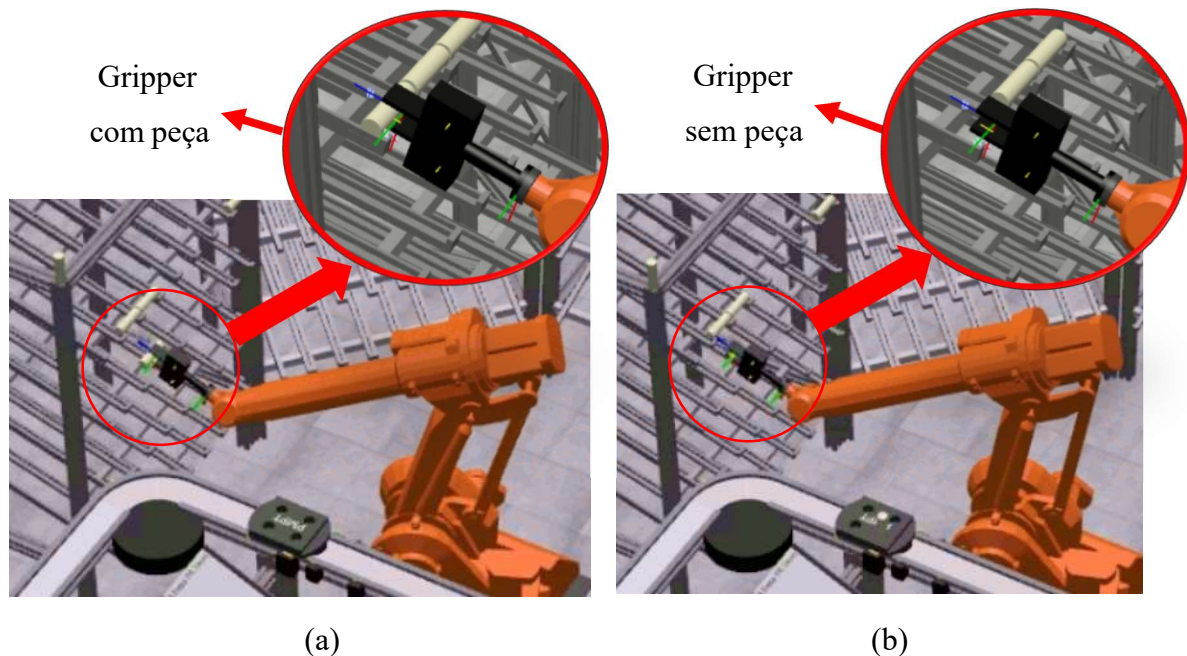


Figura 5-1: Robô IRB 1400, a trabalhar no armazém 1.

O código desenvolvido para a situação de falta de peças na coluna 1 do armazém 1, como ocorreu na Figura 5-1 b), encontra-se na Figura 5-2. Quando há falta de peças na coluna 1 é incrementado uma unidade ao valor na variável (Coluna), passando o robô a procurar peças na coluna 2.

```

PROC rRobotSemPeca()
  rAbrirDedos;
  Coluna:=Coluna+1;
  IF (Linha>=1 AND Coluna=2) X:=-105*(Coluna-1);
  IF (Linha>=1 AND Coluna=3) X:=-105*(Coluna-1);
  IF (Linha>=1 AND Coluna=4) X:=-105*(Coluna-1);
  IF (Linha>=1 AND Coluna=5) X:=-105*(Coluna-1);
  IF (Linha>=1 AND Coluna=6) X:=-105*(Coluna-1);
  IF (Linha>=1 AND Coluna=7) X:=-105*(Coluna-1);
  IF (Linha>=1 AND Coluna=8) X:=-105*(Coluna-1);
  IF (Linha=1 AND Coluna>=1) THEN
    MoveJ pAproxArm,v200,z10,toolCFF\WObj:=WObj_geral;
    Z:=70;
    pAproxArm:=Offs(pPegaPecaArm1,X,120,0);
    pPegaPecaArm:=Offs(pPegaPecaArm1,X,0,0);
    pSobePecaArm:=Offs(pPegaPecaArm1,X,0,70);
  ENDIF

```

Figura 5-2: Código para o robô mudar de coluna na linha 1.

Caso o armazém não tenha nenhuma peça na primeira linha o robô passa para a segunda linha. Se chegar ao fim do armazém e ainda faltarem peças para carregar a paleta, aparecerá uma mensagem para o operador escolher se pretende avançar para o armazém 2, para dar continuação ao seu trabalho. A Figura 5-3, apresenta o código para o operador escolher entre o armazém 1 ou 2.

```

PROC rEscolhaArm()
  lbInicio:
    TPErase;
    TPWrite "*****";
    TPWrite "**      Objectos de trabalho:      **";
    TPWrite "**          1 - Armazem 1          **";
    TPWrite "**          2 - Armazem 2          **";
    TPWrite "*****";
    TPreadNum Esc,"Escolha uma opcao:";
    IF Esc=1 THEN
      Armazem:=1;
      pPegaPecaArm:=pPegaPecaArm0;
      pPegaPecaArm1:=pPegaPecaArm0;
      WObj_geral:=WObjArm1;
      pLargaPeca1:=pLargaPeca0;
      pLargaPeca:=pLargaPeca0;
    ENDIF
    IF Esc=2 THEN
      Armazem:=2;
      pPegaPecaArm:=pPegaPecaArm2;
      pPegaPecaArm1:=pPegaPecaArm2;
      WObj_geral:=WObjArm2;
      pLargaPeca1:=pLargaPeca2;
      pLargaPeca:=pLargaPeca2;
    ENDIF

```

Figura 5-3: Código para a escolha do armazém 1 ou 2.

Para o operador ter um registo em ficheiro do que foi acontecendo, ao longo do carregamento de uma palete, foi desenvolvida a rotina, “rFaltaPECA()”, como podemos ver na Figura 5-4.

```

PROC rFaltaPeca()
Open "home:"\File:="1_Mestrado_Jorge\Ficheiros/ArmazemTorno.txt",file\Append;
IF (Armazem=1) THEN
Write file,"===== ARMAZEM TORNO =====";
Write file,"===== FALTA DE PECAS =====";
Write file,"Armazem"\NoNewLine;
Write file," "\NoNewLine;
Write file,"Data"\NoNewLine;
Write file," "\NoNewLine;
Write file,"Horas";
ENDIF
Write file," "\NoNewLine;
Write file,"Num:=Armazem\NoNewLine;
Write file," "\NoNewLine;
Write file,CDate()\NoNewLine;
Write file," "\NoNewLine;
Write file,CTime();
Close file;
ENDPROC
ENDMODULE

```

Figura 5-4: Código para gerar o ficheiro de texto “ArmazemTorno.txt”.

Neste ficheiro de texto, “ArmazemTorno.txt”, será registada a seguinte informação: qual o armazém onde ocorreu a falta de peças, em que data e hora isso aconteceu. A Figura 5-5 mostra a informação registada no ficheiro “ArmazemTorno.txt”.

Armazem	Data	Horas
1	2022-05-27	02:11:19
2	2022-05-27	02:13:57

Figura 5-5: Dados do ficheiro de texto “ArmazemTorno.txt”.

Foi criada uma rotina, “rControloPalete()” para verificar se a palete ficou bem carregada com as 4 peças. Nesta rotina são utilizadas outras duas rotinas diferentes. Como pode observar-se na Figura 5-6, a rotina “rControloPECA1E2()” verifica as peças 1 e 2 e a “rControloPECA3E4()” verifica as peças 3 e 4.

```

PROC rContoloPalete()
rAbrirDedos;
WaitTime 2;
MoveJ pVerficaPalete,v200,z10,toolCFF;
WaitTime 3;
rContoloPECA1E2;
WaitTime 2;
pVerficaPalete1:=Offs(pVerficaPalete,-75,0,0);
MoveL pVerficaPalete1,v200,z10,toolCFF;
WaitTime 3;
rContoloPECA3E4;
WaitTime 2;
ENDPROC

```

Figura 5-6: Rotina de controlo da palete.

As rotinas “rControloPECA1E2()” e “rControloPECA3E4()” utilizam as fibras óticas (DI10\_6 e a DI10\_7), que se encontram aplicadas na pinça pneumática. Na Figura 5-7, pode verificar-se o código desenvolvido para a rotina “rControloPECA1E2()”, os sensores de fibras óticas detetam se as paletes têm peça ou não. No caso destes ganharem sinal, a peça está colocada corretamente na paleta, por exemplo, a variável “bPeca1” neste caso, vai alterar o seu estado para “TRUE”. No caso de os sensores óticos não ganharem sinal, significa que as paletes não contêm peça, logo a variável “bPeca2”, passa a “FALSE”. De seguida, o controlo das peças 3 e 4 é realizado da mesma forma, alterando o valor das variáveis “bPeca3” e “bPeca4”, consoante o valor lido pelos sensores de fibras óticas.

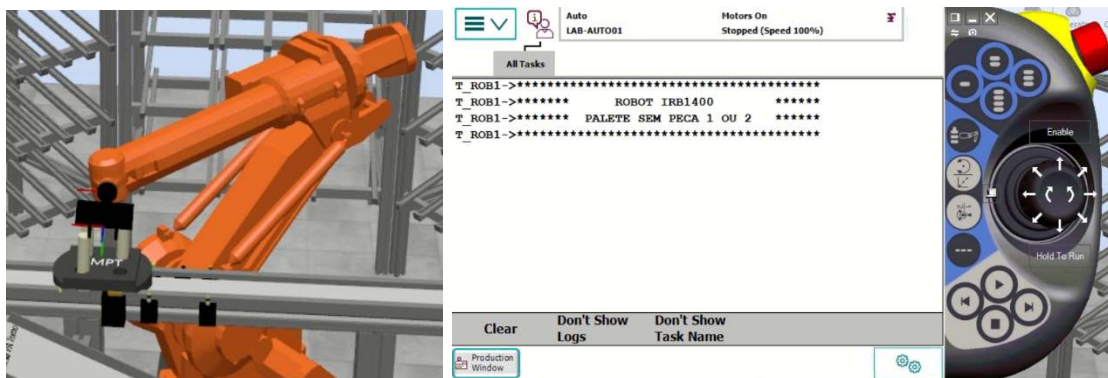
```

PROC rContoPECA1E2()
  IF (DI10_6=1 OR DI10_7=1) THEN
    TPErase;
    TPWrite "*****";
    TPWrite "*****  ROBOT IRB1400  *****";
    TPWrite "*****  PALETE COM PECA  *****";
    TPWrite "*****";
    IF DI10_7=1 bPeca1:=TRUE;
    IF DI10_6=1 bPeca2:=TRUE;
  ENDIF
  IF (DI10_6=0 OR DI10_7=0) THEN
    TPErase;
    TPWrite "*****";
    TPWrite "*****  ROBOT IRB1400  *****";
    TPWrite "*****  PALETE SEM PECA 1 OU 2  *****";
    TPWrite "*****";
    IF DI10_7=0 bPeca1:=FALSE;
    IF DI10_6=0 bPeca2:=FALSE;
  ENDIF
ENDPROC

```

Figura 5-7: Rotina de controlo da peça 1 e 2.

Como referido anteriormente, consoante informação das fibras óticas, como apresentado na Figura 5-8 a), as entradas podem estar a 1 ou a 0. Durante a fase de controlo, aparece uma mensagem na consola a informar o operador se a paleta tem as 4 peças, ou se falta alguma peça na paleta, como se pode ver na Figura 5-8 b).



(a)

(b)

Figura 5-8: a) Controlo da paleta e b) Mensagem de controlo.

Após o controlo da paleta, é gerado o ficheiro de texto que fornece a informação da paleta, a data e hora em que foi feito o controlo, o tempo que o robô demorou a carregar a paleta (tempo de ciclo), se a paleta está completamente carregada (*OK*) ou não (*NOK*) e quais as peças em falta. Como pode observar-se na Figura 5-9, a paleta não foi carregada corretamente (*NOK*), pois a peça 2 e 4 estão em falta.

```

===== PALETE TORNO =====
===== PALETES NO TORNO =====
Paleta      Data      Horas      Tempo De Ciclo (min)  OK/NOK  Peças em Falta
  1         2022-05-19   16:40:21    2.87745             NOK      0 2 0 4
  
```

Figura 5-9: Ficheiro de texto, Paleta "NOK".

Se a paleta estiver carregada corretamente, aparecerá a informação "*OK*" e nas peças em falta teremos a combinação (0 0 0 0), ou seja, não existe nenhuma peça em falta como pode notar na Figura 5-10.

```

===== PALETE TORNO =====
===== PALETES NO TORNO =====
Paleta      Data      Horas      Tempo De Ciclo (min)  OK/NOK  Peças em Falta
  1         2022-05-19   15:08:29    3.59265             OK       0 0 0 0
  
```

Figura 5-10: Ficheiro de texto, Paleta "OK".

A Figura 5-11 a) mostra uma paleta sem as peças 2 e 4 e a Figura 5-11 b) apresenta uma paleta completamente carregada.

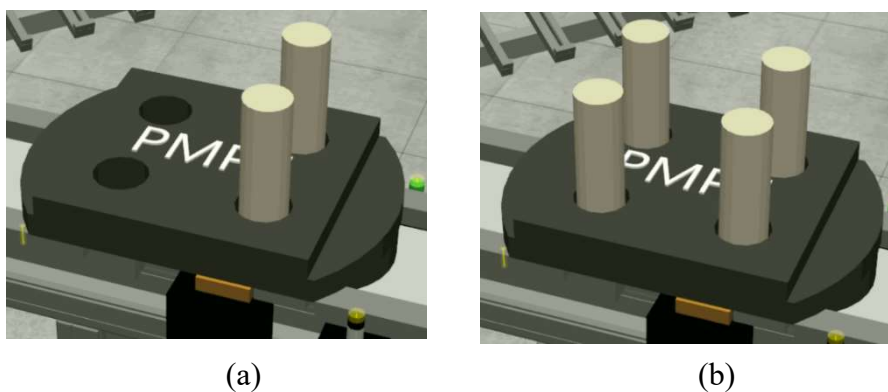


Figura 5-11: a) Paleta "NOK" e b) Paleta "OK".

Quando as paletes da MP estiverem carregadas, estas vão deslocar-se para o setor de fabrico. O robô *IRB1400*, além da função de carregar as paletes de MP, descarrega a paletes dos PA, provenientes da fresadora e do torno. Para o robô executar o descarregamento destas paletes, têm de receber um sinal emitido pelos sensores indutivos, dando a informação de que as paletes do PA, estão no local de descarga. Como se pode observar na Figura 5-12, a entrada “Descarregar” vai a 1, o que leva o robô a simular o excerto do código correspondente a esta entrada. Neste ponto, enquanto a “PecaTF for <=4”, o robô vai simular a rotina “rDescargaT”, retirando as 4 peças da paleta e colocando-as na rampa dos PA.

```

IF (Descarregar=1) THEN
  PecaTF:=1;
  MoveAbsJ home\NoEOffs,v1500,fine,toolCFF;
  WHILE PecaTF<=4 DO
    rDescargaT;
    PecaTF:=PecaTF+1;
  ENDWHILE
  SetDO DESCARCONC,1;
  WaitTime 2;
  SetDO DESCARCONC,0;
  MoveAbsJ home\NoEOffs,v1500,fine,toolCFF;
ENDIF

```

Figura 5-12: Input da PPAT quando vai a 1.

A Figura 5-13 mostra o robô *IRB1400* a descarregar a paleta PPAT e a colocar os PA na rampa dos PA. Estes vão ficar armazenados no final da rampa, até o operador os retirar. Os PA da fresadora, quando chegarem a este ponto, são descarregados da mesma forma.

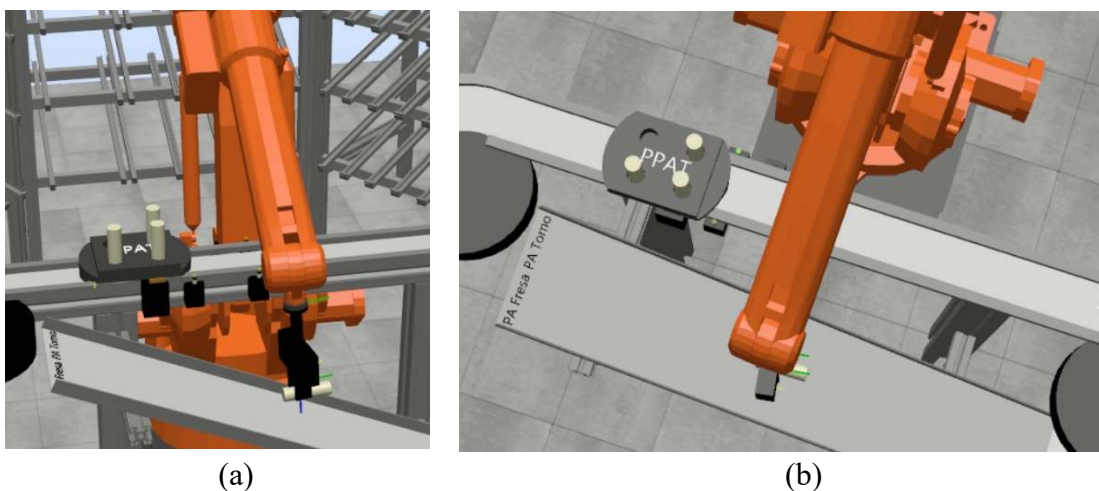


Figura 5-13: Descarga do PA da PPAT

## 5.2. Programação e simulação do Setor de Fabrico

Neste setor, tal como no setor de armazenamento, foi utilizado o mesmo método de programação. Foi desenvolvido, um código onde o robô carrega as paletes dos PA e descarrega as paletes da MP, provenientes do setor de armazenamento. Também é programado o carregamento e descarregamento das duas máquinas CNC.

A Figura 5-14 mostra as paletes posicionadas no *stoppers*, prontas para serem descarregadas pelo robô *IRB140*. O robô descarrega a PMPT e PMPF por ordem de chegada ao setor de fabrico. As entradas (DI12\_9 e DI12\_10) informam o robô da chegada das paletes ao setor de fabrico.

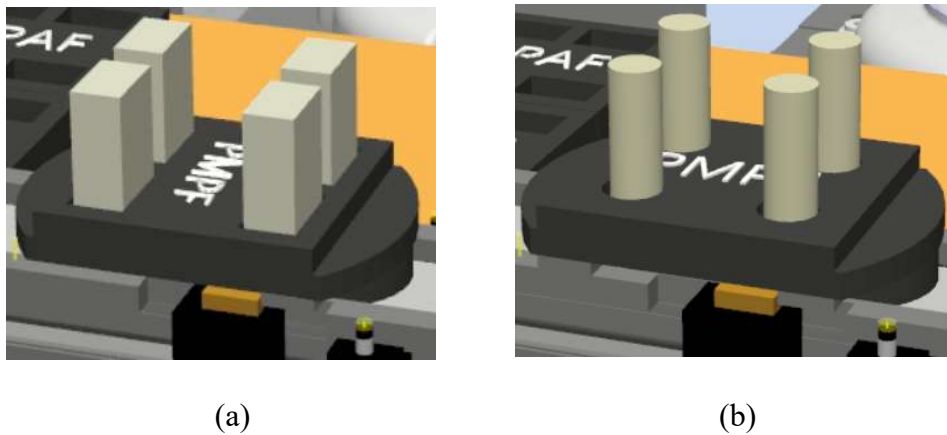


Figura 5-14: a) PMPF e b) PMPT

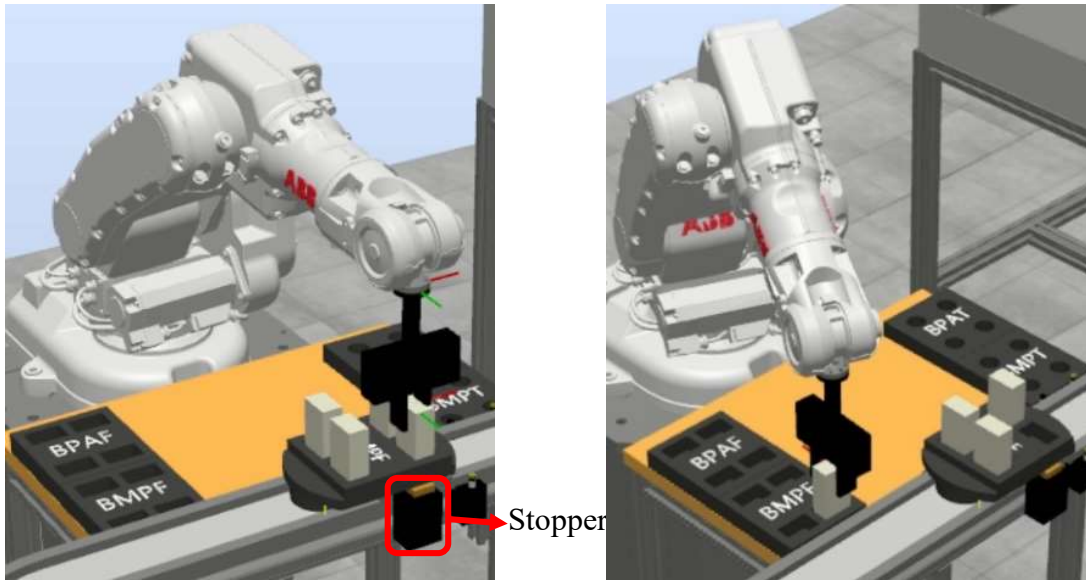
Com a chegada das paletes de MP, o robô descarrega as 4 peças para os respetivos *buffers* da MP. Para isso, é executado o código das rotinas “rFRESADORA()” e “rTORNO()”, apresentado na Figura 5-15.

```
PROC rFRESADORA()
! IF (DI12_10=1) THEN
  rAbrirDedos;
  MoveAbsJ home\NoEOffs,v1500,fine,toolCFF;
  Peca:=1;
  WHILE Peca<=4 DO
    rLevaPeca;
    Peca:=Peca+1;
  ENDWHILE
  bBufferCheioFresa:=TRUE;
  MoveAbsJ home\NoEOffs,v1500,fine,toolCFF;
! ENDIF
ENDPROC

PROC rTORNO()
  IF (DI12_9=1) THEN
    rAbrirDedos;
    MoveAbsJ home\NoEOffs,v1500,fine,toolCFF;
    Peca:=1;
    WHILE Peca<=4 DO
      rLevaPeca;
      Peca:=Peca+1;
    ENDWHILE
    bBufferCheioTorno:=TRUE;
    MoveAbsJ home\NoEOffs,v1500,fine,toolCFF;
  ! ENDIF
  ENDPROC
```

Figura 5-15: Rotinas para descarregar a MP.

Na Figura 5-16, pode observar-se o robô *IRB140* a executar a rotina “rFRESADORA()”. Na Figura 5-16 a), podemos observar a PMPF parada no *stopper* e o robô a iniciar o processo de descarga da paleta. A Figura 5-16 b) mostra o robô a colocar a primeira peça no BMPF. O processo repete-se até a paleta estar completamente descarregada.

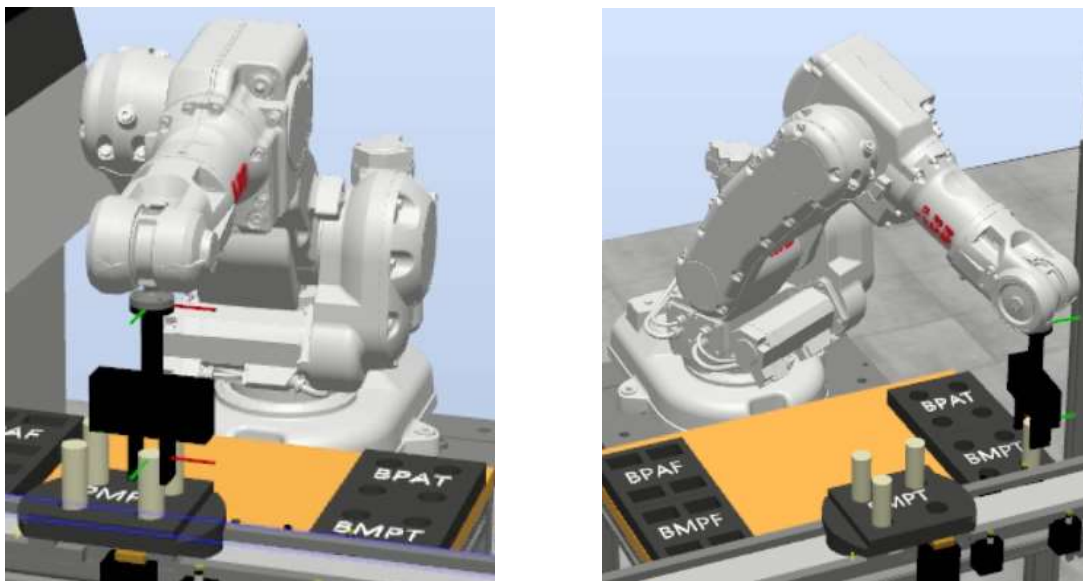


(a)

(b)

Figura 5-16: a) Descarga da PMPF e b) Carregamento do BMPF.

O mesmo processo acontece com a chegada da PMPT, onde o robô descarrega as 4 peças cilíndricas, mas demonstra a Figura 5-17.



(a)

(b)

Figura 5-17: a) Descarga da PMPT e b) Carregamento do BMPT.

Concluídas as rotinas para descarregar a MP, após o ciclo de 4 peças de cada uma das duas paletes, os *buffers* da MP ficam completamente carregados, como pode observar-se na Figura 5-18. Estando os BMPF e BMPT cheios, as variáveis “bBufferCheioTorno” e “bBufferCheioFresa” passam a “TRUE”.

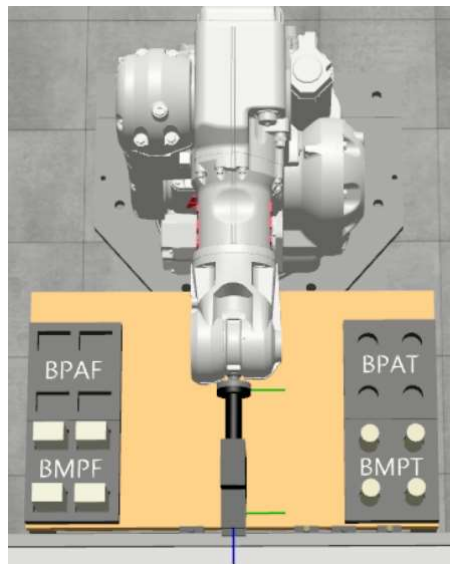


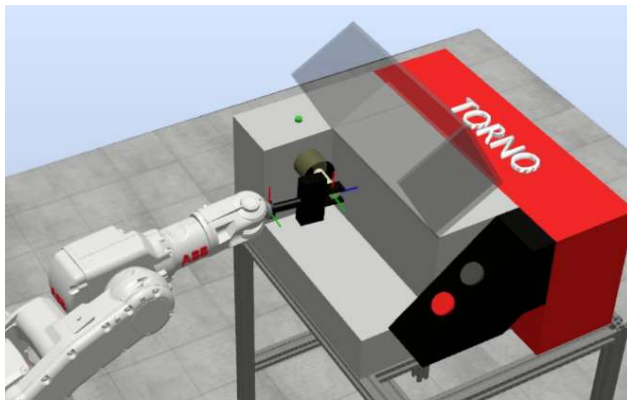
Figura 5-18: Buffers de MP carregados.

A partir deste ponto os *buffers* estão cheios “TRUE” e o robô tem peças para carregar as duas máquinas CNC, podendo assim executar cada um dos seguintes “IF’S”, como pode ver-se na Figura 5-19. Sempre que haja peças disponíveis nos *buffers*, o robô verifica se as CNC estão a maquinar ou se estão paradas. Se estiverem paradas, as variáveis “bTornoLivre ” e “bFresaLivre” vão estar a “TRUE” e serão executadas as rotinas “rCarregaTorno” e “rCarregaFresa”. Se estiverem a maquinar, as variáveis estarão a “FALSE” e o robô vai ter de esperar para que uma das CNC conclua a sua tarefa.

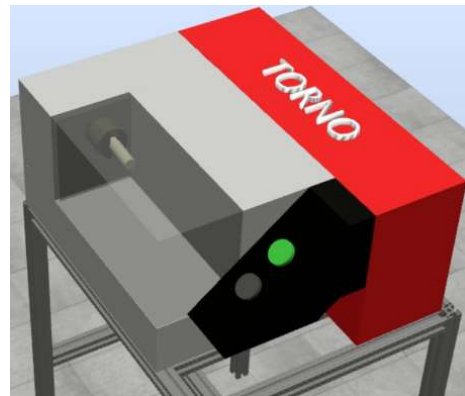
```
IF bBufferCheioTorno=TRUE THEN
  IF (bTornoLivre=TRUE) THEN
    rCarregaTorno;
  ENDIF
ENDIF
IF bBufferCheioFresa=TRUE THEN
  IF (bFresaLivre=TRUE) THEN
    rCarregaFresa;
  ENDIF
ENDIF
```

Figura 5-19: Código para o carregamento das máquinas CNC.

Por exemplo, se ambas as máquinas estiverem paradas, o robô começa por carregar o Torno. Para isso, pega na peça cilíndrica no BMPT e espera pela informação de que a porta e o *chuck* do Torno estejam abertos. De seguida, o robô coloca a peça no *chuck*, como pode ver-se na Figura 5-20 a), espera que este feche, recua para o ponto “Home”, a porta do torno fecha e a máquina recebe a ordem para começar a maquinar, como apresentado na Figura 5-20 b).



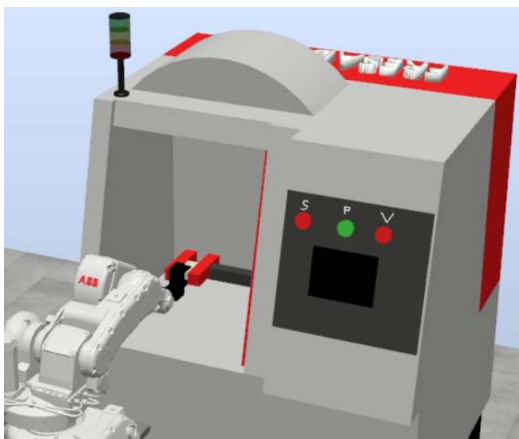
(a) Robô a carregar torno



(b) Torno a maquinar

Figura 5-20: Etapas do Torno.

Enquanto o Torno está a maquinar, e para não haver tempos mortos, o robô inicia o carregamento da Fresadora. Para isso, o robô vai ao BMPF, pega numa peça, verifica se a porta e o *vice* da fresadora estão abertos. De seguida, o robô coloca a peça no *vice*, espera que este feche, como apresentado na Figura 5-21 a). Depois do *vice* fechar, o robô volta ao ponto de “Home”, a porta recebe ordem para fechar e de seguidamente a Fresadora inicia a maquinação, como se pode verificar na Figura 5-21 b).



(a) Robô a carregar a fresadora



(b) Fresadora a maquinar

Figura 5-21: Etapas da Fresadora.

Concluído o carregamento das CNC, o valor das variáveis da “bFresaLivre” e da “bTornoLivre” passa a “FALSE”. Na Figura 5-22 pode observar-se o código desenvolvido para o robô descarregar as máquinas CNC. Foram criadas duas rotinas do tipo TRAP: “rTrapDescFresa” e “rTrapDescTorno”. Estas rotinas são executadas através dos *interrupts* “int\_DescFresa” e “int\_DescTorno”. O robô dará sempre prioridade ao descarregamento das máquinas, uma vez que as rotinas do tipo TRAP têm prioridade sobre as restantes.

```

IF (bFresaLivre=TRUE) THEN
  IDelete int_DescFresa;
  CONNECT int_DescFresa WITH rTrapDescFresa;
  ISignalDI\Single, DI10_12,1,int_DescFresa;
ENDIF

IF (bTornoLivre=TRUE) THEN
  IDelete int_DescTorno;
  CONNECT int_DescTorno WITH rTrapDescTorno;
  ISignalDI\Single, DI10_1,1,int_DescTorno;
ENDIF

```

Figura 5-22: Código para descarregar as máquinas CNC.

Como se pode ver na figura anterior, para que o robô execute as rotinas TRAP, os valores das entradas, “DI10\_12” (caso da Fresadora) e “DI10\_1” para o (Torno) têm de passar a 1, correspondendo ao fim da maquinação das máquinas. Como as rotinas do tipo TRAP geram automaticamente uma fila *FIFO* (First-In-First-Out), o robô *IB140* descarrega as máquinas por ordem de finalização da maquinação. Os PA do Torno e da Fresadora, depois de serem maquinados, são inseridos nos *buffers* BPAF e BPAT, como o apresentado na Figura 5-23.

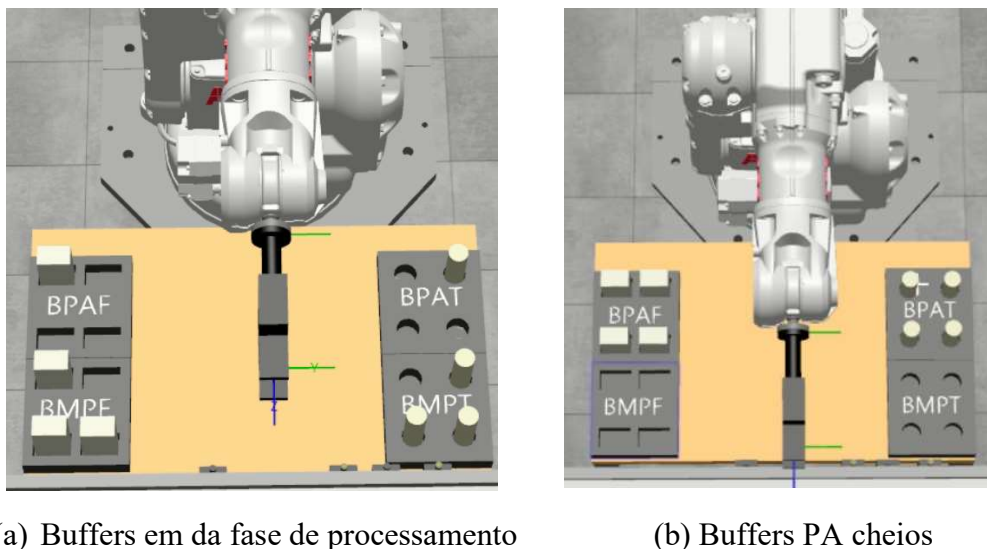


Figura 5-23: Buffers durante a maquinação da MP.

Quando os *buffers* dos PA estão no seu limite máximo, como representado na Figura 5-23 b), as variáveis “bBProdCheioTorno” e “bBProdCheioFresa” passam a “TRUE”. Assim, quando as paletes PPAF e PPAF forem detetadas pelos sensores indutivos instalados no setor de transporte, o robô recebe o sinal nas entradas “DI12\_11” ou “DI12\_12”. Desta forma, o robô executa o carregamento dos produtos acabados de acordo com a paleta que acabou de chegar. Se for a paleta PPAF é executado o código da Figura 5-24 a). Caso seja a paleta PPAF o robô carrega os produtos acabados da fresadora, e executa a parte do programa representado na Figura 5-24 b).

```

IF (DI12_11=1) THEN
  PecaTA:=1;
  WaitTime 1;
  WHILE PecaTA<=4 DO
    IF bBProdCheioTorno=TRUE rCARREGAPRODACABTORNO;
    PecaTA:=PecaTA+1;
    WaitTime 0.2;
  ENDWHILE
  WaitTime 0.4;
  SetDO DO12_11,1;
  WaitTime 2;
  SetDO DO12_11,0;
  MoveAbsJ home\NoEOffs,v1500,fine,toolCFF;
  WaitTime 2;
ENDIF

```

(a)

```

IF (DI12_12=1) THEN
  PecaFA:=1;
  WaitTime 1;
  WHILE PecaFA<=4 DO
    IF bBProdCheioFresa=TRUE rCARREGAPRODACABFRESA;
    PecaFA:=PecaFA+1;
    WaitTime 0.2;
  ENDWHILE
  WaitTime 0.4;
  SetDO DO12_12,1;
  WaitTime 2;
  SetDO DO12_12,0;
  MoveAbsJ home\NoEOffs,v1500,fine,toolCFF;
  WaitTime 2;
ENDIF

```

(b)

Figura 5-24: Programação para o carregamento das paletes de PA.

A Figura 5-25 apresenta a PPAF cheia e o BPAF vazio, depois do robô ter efetuado a transferência das 4 peças. De seguida, é dada a ordem para a paleta se poder deslocar para o setor de armazenamento, para o robô efetuar a descarga do PA, como demonstrado no subcapítulo 5.1 na Figura 5-13.

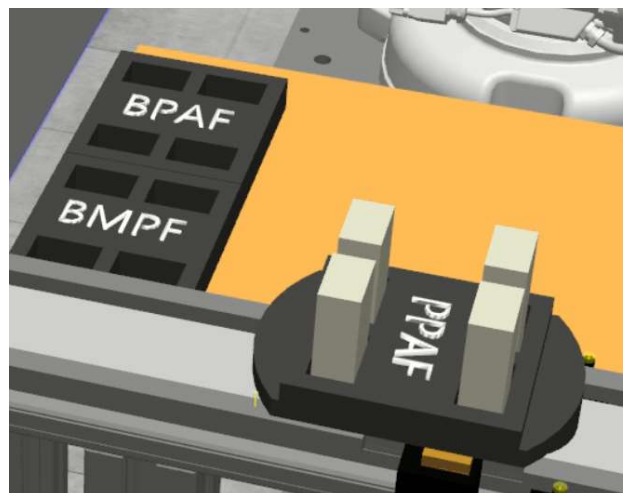


Figura 5-25: BPAF vazio e PPAF carregada.

### 5.3. Programação e simulação do Setor de Transporte

No setor de transporte da célula real foi necessário desenvolver um programa no *TIA PORTAL V16*, utilizando o *LADDER*, para o *PLC S7-1200* da Siemens. O *PLC* recebe e envia informações ao robô, lê valores dos sensores e comanda os *stoppers*, para que as paletes possam rodar entre os setores, referidos anteriormente. Desenvolvido e testado este programa em ambiente real, avança-se para o desenvolvimento da programação do setor de transporte no *RobotStudio*. Pelo facto de serem dois softwares muito pesados, foi aproveitada uma das características do *RobotStudio*, as *Smart Components*. Assim, foram criadas funções lógicas, que permitem fazer o controlo do transportador, exatamente, como acontece na realidade, através destas funções foram também criados os movimentos/percursos das paletes.

Para as paletes se moverem no setor de transporte ver Figura 4-26, foram criados movimentos lineares e de rotação para as quatro Paletes executarem um percurso. Sempre que uma paleta passar nos sensores de deteção das paletes, estes vão dizer ao programa qual a paleta que se encontra naquele local. Na Figura 5-26 pode visualizar-se que quando os sensores (I0.0, I0.1, I0.2, I0.3) forem a 1, o *LogicGate\_3* vai colocar o *output* a 1, dizendo ao robô *IRB1400* que a paleta está no local.

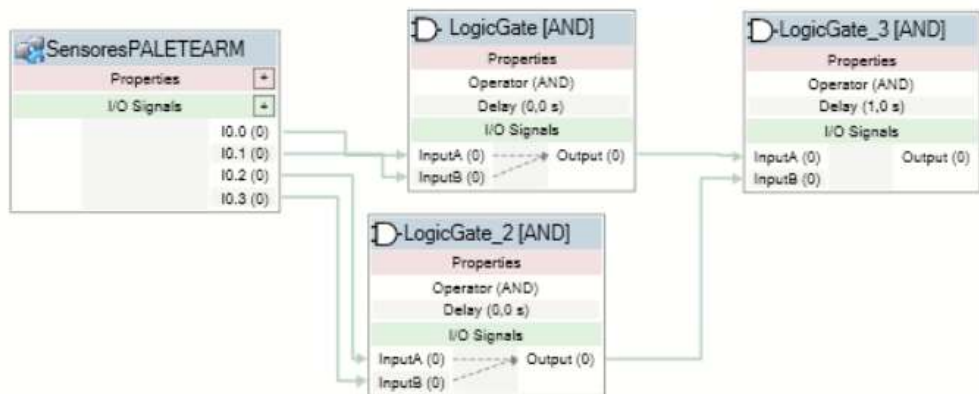


Figura 5-26: Deteção da paleta .

Este sinal de *output* vai ativar o *LineSensor* da Figura 5-27. Este ao ser ativado vai fazer *set* ao movimento da PMPT, a paleta vai parar. Para a paleta se poder movimentar outra vez, o *LogicSRLatch\_8* vai esperar que o *IRB1400* conclua o carregamento. Quando este enviar o sinal de que a PMPT está carregada, é feito o *Reset* e o movimento da paleta é executado novamente.

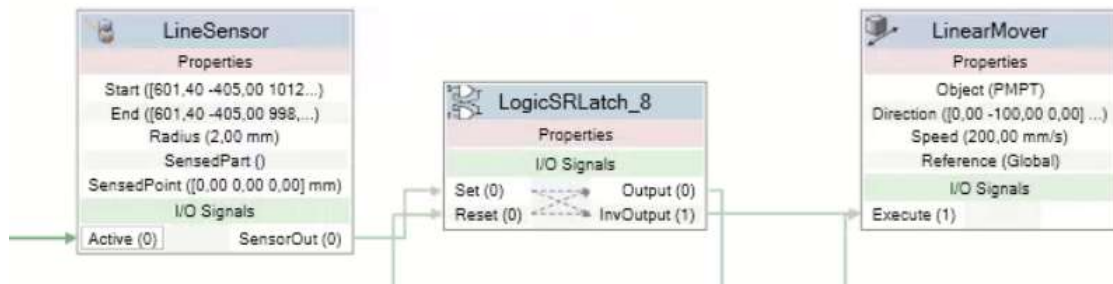


Figura 5-27: Movimento Linear da Paleta PMPT no lado do IRB 1400.

Todos os outros movimentos lineares são executados da mesma maneira que o anterior. No entanto, para as paletes fazerem a parte curvilínea do transportador, foram criadas rotações em cada uma das quatro curvas do tapete, como aponta a Figura 5-28. Por exemplo, a paleta PMPT quando tem uma colisão com o *LineSensor\_6* faz uma rotação de 90 graus, seguindo logo o seu movimento linear programado para o restante percurso.

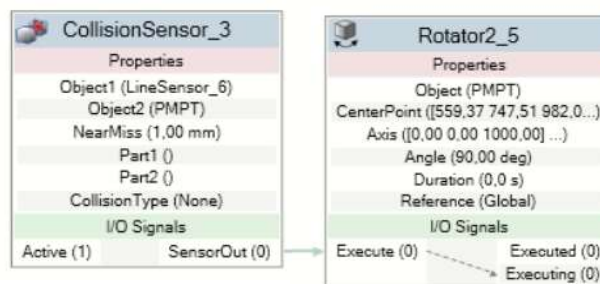


Figura 5-28: Função Lógica para realizar uma rotação.

Para além das várias funções criadas, também foi criada uma *SmartComponent*, de forma a sinalizar que está tudo em ordem para uma CNC começar a trabalhar. No caso do robô *IRB1400*, esta *SmartComponent* sinaliza que o robô está pronto para executar os movimentos pedidos. Como pode observar-se na Figura 5-29, a *SmartComponent* tem um sinal que vai a 1 e faz com que a lâmpada de sinalização fique com a cor verde (*OK*). No caso das CNC estarem paradas, é emitido um sinal e as lâmpadas ficam a vermelho (*ERROR*).



Figura 5-29: SmartComponent de uma lâmpada de sinalização.



## 6. CONCLUSÕES E SUGESTÕES DE TRABALHOS FUTUROS

### 6.1. Conclusões

Com a realização deste trabalho prático, pode concluir-se que todos os objetivos propostos inicialmente foram cumpridos. O objetivo principal desta dissertação foi criar a CFF do DEE em modo virtual. Ao longo do trabalho desta dissertação foi estudada a viabilidade da programação *offline* como alternativa ao método convencional da programação *online*. Para tal, foi utilizado o *software RobotStudio* para a programação e simulação de toda a célula flexível de fabrico. Também foi efetuado um estudo das vantagens e desvantagens da programação *offline*.

No decorrer desta dissertação foi possível averiguar que na realização da parte prática deste trabalho estavam associadas algumas dificuldades, como a correta modelação e programação de toda a CFF da ESTGV.

Com o avanço da realização deste projeto, uma das maiores dificuldades sentidas foi, sem dúvida, a elaboração da parte escrita, nomeadamente o capítulo 2, a *Revisão Histórica*. Foi imperativo recolher muita informação e imagens de artigos científicos, de plataformas *online*, livros, entre outros materiais de pesquisa.

No aperfeiçoamento da parte prática uma das grandes dificuldades sentidas foi em conseguir que os pontos da célula real e virtual fossem semelhantes, sendo que o maior obstáculo a ultrapassar foi, sem dúvida, colocar as paletes no transportador a funcionarem corretamente como acontece na célula do DEE.

Por fim, todos os programas desenvolvidos para o funcionamento da célula foram testados na célula virtual e também na célula real, para se poder demonstrar que o que foi desenvolvido *offline* também funciona na realidade. A programação *offline* teria sido muito útil no período de isolamento devido à COVID 19, uma vez que permitiria o teste *offline* dos programas.

## 6.2. Conclusão Geral

No decorrer da dissertação, adquiri conhecimentos na área da robótica, sendo que, a pesquisa para a realização do capítulo da “Revisão Histórica”, proporcionou-me um pouco mais sobre a história dos robôs, sobre a sua constituição e os vários métodos de programação. A programação *offline* tem vindo, cada vez mais, a destacar-se em relação à programação *online*. Ao longo do trabalho realizado, também foram aprofundados os conhecimentos na área dos sistemas flexíveis de fabrico.

No terceiro capítulo “Elementos da Célula Flexível de Fabrico”, foi feito um estudo aprofundado da CFF, existente no laboratório da Automação e Robótica do DEE da ESTGV.

No quarto capítulo “Trabalho Desenvolvido”, foi imperativo aprender a trabalhar com o *software RobotStudio*. A utilização deste *software* permitiu a modelação dos 3 setores existentes na CFF. Foram retiradas as medidas exatas de todos os equipamentos/componentes da CFF real para ser elaborada uma réplica da mesma no *software RobotStudio*

Por último, no quinto capítulo “Análise dos Resultados”, foram destacados todos os pontos importantes dos 3 setores, desde a programação que foi desenvolvida ao resultado de cada parâmetro, salientando-se o controlo da paleta e as possíveis falhas, carregamento e descarregamento de paletes e máquinas CNC e a programação feita para as paletes se moverem no setor de transporte.

### 6.3. Trabalhos Futuros

Tendo em conta que os objetivos da dissertação foram alcançados, existem sempre aspetos a ser melhorados, sendo que um deles é o melhoramento de alguns pontos da modelação, a programação dos movimentos das paletes nos pontos curvilíneos e a programação para as peças descerem no armazém através da força da gravítica. De seguida, são apresentadas algumas sugestões de trabalhos futuros:

- Implementar novos sensores para a deteção de paletes, pois no modelo virtual os sensores presentes demoram muito a atuar e por vezes detetam mal as paletes;
- Promover novos *buffers* no setor de fabrico para diminuir os tempos mortos e para não existir falta de matéria-prima;
- Aumentar a variedade de matéria-prima, para os robôs desempenharem novas tarefas;
- Implementar a visão artificial no RobotStudio;
- Desenvolver em *RobotStudio* do *IRB1600* e do sistema de palatização, de modo a usufruir-se todo o ambiente do laboratório do DEE, para além da CFF, modelizado em *RobotStudio*.



## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] K. C. Kim, Ed., *Robotics in General Surgery*. New York, NY: Springer New York, 2014.
- [2] T. R. Kurfess, Ed., *Robotics and automation handbook*. Boca Raton: CRC Press, 2005.
- [3] P. J. Mckerrow, *Introduction to Robotics*, Singapore: Addison Wesley, 1991.
- [4] V. M. F. Santos, "Robótica Industrial", p.166, 2003.
- [5] R. Clarke, "Asimov's Laws of Robotics: Implications for Information Techno", p. 10, 1994.
- [6] "Unimate - The First Industrial Robot".[Online]. Disponível em: <https://www.automate.org/a3-content/joseph-engelberger-unimate> (Acedido 1-nov-2022).
- [7] A. Gasparetto e L. Scalera, "A Brief History of Industrial Robotics in the 20th Century", *Adv. Hist. Stud.*, vol. 08, n.º 01, pp. 24–35, 2019.
- [8] "1958-62 - "VERSATRAN" Industrial Robot - Harry Johnson & Veljko Milenkovic", 2013. [Online]. Disponível em: <http://cyberneticzoo.com/early-industrial-robots/1958-62-versatran-industrial-robot-harry-johnson-veljko-milenkovic/> (Acedido 8-nov-2022).
- [9] A. Gasparetto e L. Scalera, "From the Unimate to the Delta Robot: The Early Decades of Industrial Robotics", em *Explorations in the History and Heritage of Machines and Mechanisms*, B. Zhang e M. Ceccarelli, Eds., em *History of Mechanism and Machine Science*, vol. 37. Cham: Springer International Publishing, 2019, pp. 284–295.
- [10] J. P. M .D. Antunes, "Programação de robôs industriais em operações de maquinagem", M.S. Thesis, FEUP, PT, Porto, 2015.
- [11] "Robótica Experimental CS225A". [Online]. Disponível em: <https://khatib.stanford.edu/teaching/cs225a-2014-15-fall/projects.html> (Acedido 8-nov-2022).
- [12] S. University, "Stanford's robotics legacy», *Stanford News*", 2019. [Online]. Disponível em: <https://news.stanford.edu/2019/01/16/stanford-s-robotics-legacy/> (Acedido 8-nov-2022).
- [13] "Histórico da Informação". [Online]. Disponível em: <https://www.historyofinformation.com/image.php?id=6896> (Acedido 8-nov-2022).
- [14] D. Feng, "Executando robôs SCARA | Laboratório de arquitetura interativa". [Online]. Disponível em: <http://www.interactivearchitecture.org/performing-scara-robots.html> (Acedido 8-nov-2022).
- [15] "Robô paralelo Delta - a história de sucesso". [Online]. Disponível em: <https://www.parallelic.org/Reviews/Review002.html> (Acedido 9-nov-2022).
- [16] IFR, "World Robotics Report: "All-Time High" with Half a Million Robots Installed in one Year", 2021. [Online]. Disponível em: <https://ifr.org/ifr-press-releases/news/wr-report-all-time-high-with-half-a-million-robots-installed> (Acedido 9 -nov-2022).
- [17] B. Dias, «Ranking dos 14 maiores fabricantes mundiais de robôs industriais», *ServNews Robótica*, 19 de julho de 2017. <https://servnews.com.br/site/ranking-dos-14-maiores-fabricantes-mundiais-de-robos-industriais/> (acedido 11 de novembro de 2022).
- [18] IFR, «Sales of Robots for the Service Sector Grew by 37% Worldwide», *IFR International Federation of Robotics*. <https://ifr.org/ifr-press-releases/news/sales-of-robots-for-the-service-sector-grew-by-37-worldwide> (acedido 12 de novembro de 2022).

- [19] "Classificação dos robôs". [Online]. Disponível em: <https://tegruposete7.wordpress.com/classificacao-dos-robos/> (Acedido 15-nov-2022).
- [20] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*. McGraw-Hill, cop.1987.
- [21] A.M.P. Ferrolho, "Apontamentos das aulas de Robótica Industrial," Viseu, 2022. [Online] disponível em: Plataforma ESTGV (Acedido: 15-nov-2022 ).
- [22] H. Asada and J. J. E. Slotine, *Robot Analysis and Control*, John Wiley & Sons, Inc., 1986.
- [23] J. J. Craig, *Introduction to Robotics Mechanics & Control*, Addison Wesley, 1986.
- [24] "IRC5 - Controlador de robô compacto by ABB Robotics | DirectIndustry". [Online]. Disponível em: <https://www.directindustry.com/pt/prod/abb-robotics/product-30265-169114.html> (Acedido: 15-nov-2022).
- [25] "How to benefit from the robotic skill shortage - Robohub".[Online]. Disponível em: <https://robohub.org/how-to-benefit-from-the-robotic-skill-shortage/> (Acedido 16-nov-2022).
- [26] S. Y. Nof, *Handbook of Industrial Robotics*. John Wiley & Sons, 1999.
- [27] H. K. Shivanand, "*Flexible Manufacturing System*". New Age International, 2006.
- [28] F. D. M. Almeida, "Visão artificial no controlo da Célula Flexível de Fabrico da ESTGV", M.S. Thesis, DEE, IPV, PT, Viseu, 2022.
- [29] Viswanadham and Y. Narahari, *Performance Modeling of Automated Manufacturing Systems*, Prentice-Hall, INC., 1992.
- [30] Kusiak, *Modelling and Design of Flexible Manufacturing Systems*, Elsevier Science Publishers, 1986.
- [31] A. I. R. de Carvalho, "Desenvolvimento e melhoria da Célula Flexível de Fabrico da ESTGV", M.S. Thesis, DEE, IPV, PT, Viseu, 2016.
- [32] M. F. M. Lima, "Desenvolvimento de Células de Fabrico Flexível: Simulação e Controlo", M.S. Thesis, DEEC, UTL, PT, Lisboa, 1996.
- [33] M. Kaighobadi and K. Venkatesh, "Flexible Manufacturing Systems: An Overview", *International Journal of Operations & Production Management*, vol.14, no. 4, pp 26-49,1994.
- [34] Blazewicz, J Dileep R. Sule, *Industrial Scheduling*, PWS Publishing Company, 1996.
- [35] Mikell P. Groover, "Automation, Production, Systems and Computer Integrated Manufacturing", Prentice HallInternational.
- [36] J. B. Waldner, *CIM, Principles of Computer Integrated Manufacturing*, John Wiley & Sons, 1992.
- [37] R. A. Wysk, N. S. Yang, and S. Joshi, "Detection of deadlocks in flexible manufacturing cells", *IEEE Trans. Robot. Autom.*, vol. 7, n.º 6, pp. 853–859, 1991.
- [38] A.M.P. Ferrolho, "Apontamentos das aulas de Sistemas Flexíveis de Fabrico," Viseu, 2023. [Online] disponível em: Plataforma ESTGV (Acedido: 25-jan-2023 ).
- [39] *ABB IRB1400 Users Manual*. ABB Robotics AB, 2003.

- [40] ABB Robotics, "3HAC9376-1\_revE\_en\_library.pdf". [Online]. Disponível em [https://library.e.abb.com/public/21c32e5bcd2708c5c12576cb00528e5d/3HAC9376-1\\_revE\\_en\\_library.pdf](https://library.e.abb.com/public/21c32e5bcd2708c5c12576cb00528e5d/3HAC9376-1_revE_en_library.pdf) (Acedido: 27-jan-2023).
- [41] ABB, "IRB1400\_R3-US 02\_05.pdf", 2005. [Online]. Disponível em: [https://library.e.abb.com/public/99bb3fb8ff6495cfc1257b130056d120/IRB1400\\_R3-US%2002\\_05.pdf](https://library.e.abb.com/public/99bb3fb8ff6495cfc1257b130056d120/IRB1400_R3-US%2002_05.pdf) (Acedido: 27-jan-2023).
- [42] ABB, "IRB140\_R5-US 02\_05.pdf.", 2005. [Online]. Disponível em: [https://library.e.abb.com/public/6e9e7cc2610b19e6c1257b130056d10f/IRB140\\_R5-US%2002\\_05.pdf](https://library.e.abb.com/public/6e9e7cc2610b19e6c1257b130056d10f/IRB140_R5-US%2002_05.pdf) (Acedido: 1-fev-2023).
- [43] *ABB Manual de Referência RAPID*. ABB Robotics AB, 2003.



## APÊNDICES

## **7. APÊNDICE A, B e C**

## **8. PROGRAMAÇÃO DO IRB1400, IRB140 E DO PLC**

A programação desenvolvida, para os robôs e PLC, encontra-se disponível num ficheiro à parte da dissertação.

## **ANEXOS**

## 9. ANEXO A: INTERFACE ROBÓTICO

### Entradas e saídas na Fresadora CNC

POSIÇÃO NO IRB140	DESCRIÇÃO
DI10_9=1 DI10_10=0	ALARM ACTIVE
DI10_9=1 DI10_10=1	EMERGENCY
DI10_12	PROGRAM STOP
DI10_13	VICE CLOSED
DI10_14	VICE OPEN
DI10_15	DOOR CLOSED
DI10_16	DOOR OPEN
DO10_9	CLOSE DOOR
DO10_13	NC START
DO10_14	CLOSE VICE
DO10_15	OPEN VICE
DO10_16	OPEN DOOR

### Entradas e saídas na Torno CNC

POSIÇÃO NO IRB140	DESCRIÇÃO
DI10_1	PROGRAM STOP
DI10_2	CHUCK OPEN
DI10_3	CHUCK CLOSED
DI10_4	DOOR OPEN
DI10_5	DOOR CLOSED
DI10_8	ALARM ACTIVE
DO10_2	PROGRAM START
DO10_3	CLOSE DOOR
DO10_4	OPEN DOOR
DO10_7	CLOSE CHUCK
DO10_8	OPEN CHUCK

### Outras entradas e saídas

IRB140	DESCRIÇÃO
DI12_1	Sensor óptico número 1
DI12_6	Sensor óptico número 2
DI12_3	Botão OK do operador
DI12_2	Botão “Parar produção”
DO12_1	Baliza de sinalização (luz verde)
DO12_2	Baliza de sinalização (luz vermelha + alarme)
DO10_1Pinça	<i>Gripper</i> (Pinça) do robô (0 – abre, 1 – Fecha)

PLC S7-200 PLC S7-1200	Robô IRB 1400	Descrição
I1.0	DO10 9	Paleta MP torno carregada
I1.1	DO10 10	Paleta MP fresadora carregada
Q0.4	DI10 12	Carregar paleta MP torno
Q0.5	DI10 13	Carregar paleta MP fresadora

PLC S7-200 PLC S7-1200	Robô IRB 140	Descrição
I1.2	DO12 9	Paleta MP torno descarregada
I1.3	DO12 10	Paleta MP fresadora descarregada
Q0.6	DI12 9	Descarregar paleta MP torno
Q0.7	DI12 10	Descarregar paleta MP fresadora
Q1.0	DI12 11	Carregar Paleta PA torno
Q1.1	DI12 12	Carregar Paleta PA fresadora
I1.4	DO12 11	Paleta PA torno carregada
I1.5	DO12 12	Paleta PA fresadora carregada

PLC S7-200 PLC S7-1200	Transportador – setor do armazém
I0.0	S1 no setor do armazém
I0.1	S2 no setor do armazém
I0.2	S3 no setor do armazém
I0.3	S4 no setor do armazém
MPT	I0.0=1; I0.1=1; I0.2=1; I0.3=1
MPF	I0.0=1; I0.1=0; I0.2=0; I0.3=1
PAT	I0.0=1; I0.1=0; I0.2=1; I0.3=1
PAF	I0.0=0; I0.1=0; I0.2=0; I0.3=1

PLC S7-200 PLC S7-1200	Transportador – setor de fabrico
I0.4	S1 no setor de fabrico
I0.5	S2 no setor de fabrico
I0.6	S3 no setor de fabrico
I0.7	S4 no setor de fabrico
MPT	I0.4=1; I0.5=1; I0.6=1; I0.7=1
MPF	I0.4=1; I0.5=0; I0.6=0; I0.7=1
PAT	I0.4=1; I0.5=0; I0.6=1; I0.7=1
PAF	I0.4=0; I0.5=0; I0.6=0; I0.7=1

PLC S7-200 PLC S7-1200	Transportador – <i>Stoppers</i>
Q0.0	<i>Stopper</i> no setor de fabrico
Q0.1	<i>Stopper</i> no setor do armazém
Q0.2	<i>Stopper</i> no setor auxiliar
Q0.3	<i>Stopper</i> no setor de montagem