



# Projeto e Montagem de Célula Flexível de Montagem com Colaboração Homem/Robô

Vítor Manuel Ribeiro de Sousa

Dissertação apresentada à Escola Superior de Tecnologia e Gestão de Bragança para obter o grau de Mestre em Engenharia Industrial - ramo Engenharia Eletrotécnica.

Sob orientação de:

Prof. Dr. Paulo Jorge Pinto Leitão

Prof. Dr. José Fernando Lopes Barbosa

Bragança

2018





# Projeto e Montagem de Célula Flexível de Montagem com Colaboração Homem/Robô

Vítor Manuel Ribeiro de Sousa

Dissertação apresentada à Escola Superior de Tecnologia e Gestão de Bragança para obter o grau de Mestre em Engenharia Industrial - ramo Engenharia Eletrotécnica.

Sob orientação de:

Prof. Dr. Paulo Jorge Pinto Leitão

Prof. Dr. José Fernando Lopes Barbosa

Bragança

2018



# Agradecimentos

Gostaria de agradecer aos meus pais por me inculcaram os valores que hoje definem a minha personalidade, acreditaram em mim e lutaram para que eu hoje pudesse estar a evoluir.

À minha namorada por me incentivar a não desistir.

Aos meus orientadores Professor Dr. Paulo Jorge Pinto Leitão e Professor Dr. José Fernando Lopes Barbosa, pela oportunidade cedida e todos os seus conselhos. A todos com quem privei na minha vida estudantil e laboral.

Porque a minha família são os meus amigos um muito obrigado a eles e deixo aqui o meu voto de gratidão, não só pelas pequenas conquistas mas também derrotas que me ajudaram a conquistar, foram essas que me ensinaram a viver e a trabalhar.

À Escola e sobretudo à cidade dos amigos para sempre que tão bem me acolheu e me convenceu a ficar etapa após etapa, ensinou a crescer e fez-me reter lições para a vida.

A todos o meu muito obrigado por me ajudarem a compreender que o mundo não acaba nos livros!

Vítor Manuel Ribeiro de Sousa



# Resumo

Atualmente, os mercados são bastante exigentes e procuram produtos diversificados, de elevada qualidade e com prazos de entrega bastante reduzidos. Estes requisitos impõem novos desafios à indústria mundial, e também portuguesa. Neste âmbito, surge a 4ª revolução industrial onde novas tecnologias de informação e comunicação, assim como de inteligência artificial, são utilizadas para realizar a digitalização dos sistemas produtivos, considerando a elevada quantidade de dados disponíveis para realizar monitorização, previsão e otimização dos processos.

A adoção destes princípios e das tecnologias emergentes associadas encontra-se ainda numa fase precoce. Os motivos deste atraso são essencialmente, o investimento necessário à sua implementação, o conhecimento das possíveis vantagens, a elevada dependência por parte da indústria no uso de processos manuais, exigindo uma elevada especialização de mão-de-obra e pouca automatização de processos. O projeto I4.0@TMAD foca esta temática através da promoção da Indústria 4.0 na Região de Trás-os-Montes e Alto Douro, tendo um papel fundamental em criar abertura no seio das empresas da região para abraçar esses novos conceitos e tecnologias. No âmbito deste projeto, surge então a necessidade de desenvolver uma célula demonstrativa que considere conceitos, equipamentos e abordagens inovadoras. Desta forma, foi construída uma célula robótica colaborativa que demonstra a potencialidade de colaboração entre o humano e o robô. Este trabalho contemplou, entre outros aspetos, o dimensionamento dos componentes, nomeadamente os estruturais, mecânicos e de controlo, a sua implementação, testes e validação da solução desenvolvida. De forma a demonstrar os conceitos de uma forma simplificada e abrangente, optou-se por implementar dois cenários de interação colaborativa homem-robô.

Para finalizar, foram realizadas varias demonstrações a empresários de vários setores da região de Trás-os-Montes e Alto Douro (TMAD).

**Palavras-chave:** industria 4.0, robótica colaborativa, automação de processos, colaboração humano-robô, segurança.

# Abstract

Currently, the markets are quite demanding and are looking for diversified products, high quality and with very short lead times. These requirements pose new challenges to the world industry, as well as portuguese. In this context, the 4th industrial revolution emerges, where new information and communication technologies, as well as artificial intelligence, are used to perform the digitization of production systems, considering the high amount of data available to perform monitoring, forecasting and optimization of processes.

The adoption of these principles and associated emerging technologies is still at an early stage. The reasons for this delay are essentially the investment needed to implement it, the knowledge of the possible advantages, the high dependence of the industry on the use of manual processes, requiring a high labor specialization and little automation of processes. The I4.0@TMAD project focuses on this theme through the promotion of Industry 4.0 in the Trás-os-Montes and Alto Douro Region of Portugal, playing a key role in creating openness within the region's companies to embrace these new concepts and technologies. Within this project, the need arises to develop a demonstration cell that considers innovative concepts, equipment and approaches. In this way, a collaborative robotic cell was built that demonstrates the potential for collaboration between the human and the robot. This work included, among other aspects, the dimensioning of the components, namely structural, mechanical and control, its implementation, testing and validation of the developed solution. In order to demonstrate the concepts in a simplified and comprehensive way, it was decided to implement two scenarios of human-robot collaborative interaction. To conclude, several demonstrations were held for entrepreneurs

from various sectors of the TMAD region.

**Keywords:** industry 4.0, collaborative robotics, process automation, human-robot collaboration, security.

# Conteúdo

<b>Resumo</b>	<b>vii</b>
<b>Abstract</b>	<b>ix</b>
<b>Acrónimos</b>	<b>xix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Objetivos . . . . .	3
1.2 Estrutura do Documento . . . . .	3
<b>2 Estado da Arte</b>	<b>5</b>
2.1 Indústria 4.0 e Suas Premissas . . . . .	7
2.2 Células Industriais . . . . .	9
2.3 Sistemas Robóticos . . . . .	10
2.3.1 Classificação . . . . .	10
2.3.2 Robôs Manipuladores . . . . .	12
2.3.3 Sistemas de Controlo . . . . .	15
2.4 Sistemas Robóticos Colaborativos . . . . .	16
2.5 Tipos de Ferramenta . . . . .	20
2.5.1 Garras . . . . .	20
2.5.2 Ferramentas e Sensores Adicionais . . . . .	21
2.6 Processamento e Simulação . . . . .	22
2.6.1 V-REP . . . . .	23

2.6.2	Robotstudio . . . . .	24
2.6.3	Solidworks . . . . .	25
2.7	Comunicações Industriais . . . . .	26
2.8	Sistemas de Visão . . . . .	27
2.9	Sistemas de Segurança . . . . .	28
2.10	Ergonomia . . . . .	30
<b>3</b>	<b>Requisitos e Conceção do Sistema Colaborativo</b>	<b>33</b>
3.1	Requisitos Funcionais do Sistema . . . . .	33
3.2	Arquitetura do Sistema . . . . .	35
3.3	Enquadramento da Prova de Conceitos . . . . .	38
3.3.1	Cenário Demonstrativo de Tarefa Contínua . . . . .	39
3.3.2	Cenário Demonstrativo de Arrumação de Peças . . . . .	40
3.3.3	Cenário Demonstrativo de Colocação de Peças . . . . .	41
3.4	Conceção Estrutural . . . . .	42
3.5	Sistemas de Apoio ao Funcionamento . . . . .	44
3.5.1	Sistemas de Apoio à Tarefa Contínua . . . . .	44
3.5.2	Sistemas de Apoio ao Cenário de Colocação de Peças . . . . .	45
3.5.3	Sistemas de Apoio ao Cenário de Arrumação de Peças . . . . .	46
3.6	Simulação para Validação Funcional . . . . .	48
<b>4</b>	<b>Desenvolvimento e Instalação da Solução Desenvolvida</b>	<b>51</b>
4.1	Instalação de Equipamentos . . . . .	51
4.1.1	Montagem dos Dispositivos de Robótica . . . . .	52
4.1.2	Montagem do Circuito de Potência . . . . .	54
4.1.3	Montagem do Sistema de Iluminação . . . . .	55
4.1.4	Montagem dos Dispositivos de Emergência . . . . .	57
4.2	Programação e Configurações . . . . .	58
4.2.1	Configurações Iniciais do Robô . . . . .	58
4.2.2	Instalação de <i>Software</i> dos Periféricos do Robô . . . . .	59

4.2.3	Definições de Segurança . . . . .	59
4.2.4	Componente de Comunicação . . . . .	61
4.2.5	Programação do Robô . . . . .	61
4.3	Construção da Aplicação HMI . . . . .	66
<b>5</b>	<b>Testes Experimentais</b>	<b>69</b>
5.1	Testes aos Dispositivos Elétricos . . . . .	69
5.2	Testes aos Dispositivos de Segurança . . . . .	70
5.3	Testes ao Funcionamento de Processos . . . . .	72
5.3.1	Desenvolvimento da Torre de <i>Hanoi</i> . . . . .	73
5.3.2	Desenvolvimento do Cenário de Colocação de Peças . . . . .	74
5.3.3	Desenvolvimento do Cenário de Arrumação de Peças . . . . .	75
<b>6</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>77</b>
6.1	Conclusão . . . . .	77
6.2	Trabalhos Futuros . . . . .	79
<b>A</b>	<b>Esquemas elétricos</b>	<b>85</b>
<b>B</b>	<b>Foto-Descrição do Cenário de Tarefa Contínua</b>	<b>94</b>
<b>C</b>	<b>Código de Programação Robô</b>	<b>96</b>
<b>D</b>	<b>Tutorial de Programação</b>	<b>108</b>
<b>E</b>	<b>Poster do Trabalho Realizado</b>	<b>127</b>

# Lista de Tabelas

4.1	Descrição dos Códigos Escritos na Porta 129 <i>Modbus</i> pelo Computador. . .	63
4.2	Descrição dos Códigos Escritos na Porta 136 <i>Modbus</i> pelo Robô. . . . .	63



# Lista de Figuras

2.1	Primeiro Robô Industrial [2]. . . . .	5
2.2	Retrato das Evoluções Industriais (adaptado de [3]) . . . . .	7
2.3	Tecnologias e Conceitos da Indústria 4.0 [5]. . . . .	8
2.4	Classificação de Robôs [8]. . . . .	11
2.5	Robô StarWars C-3P0 [10]. . . . .	11
2.6	Robô <i>AGB MiR100<sup>TM</sup></i> [11]. . . . .	12
2.7	Robô <i>ABB IRB1410</i> [12]. . . . .	12
2.8	Diferença Entre Precisão e Repetibilidade do Tool Center Point (TCP) [14].	14
2.9	Braço Manipulador IRB 1400 [12]. . . . .	15
2.10	Robô UR5 com Unidade de Controlo e Consola [15]. . . . .	16
2.11	Robô Colaborativo UR3 [15]. . . . .	19
2.12	Garra Mecânica <i>ROBOTIQ 2 Finger</i> [20]. . . . .	20
2.13	Sensor de Força FT300 ROBOTIQ [20]. . . . .	21
2.14	Wrist Câmara ROBOTIQ [20]. . . . .	22
2.15	Ambiente de Simulação Virtual Robot Experimentation Platform (V-REP).	23
2.16	Ambiente de Simulação RobotStudio [24]. . . . .	24
2.17	Ambiente de Simulação Solidworks. . . . .	25
2.18	Constituição da Kinect [28]. . . . .	28
2.19	Célula Robótica Tradicional. . . . .	29
3.1	Pré Requisito para Definição de Zonas de Trabalho. . . . .	34
3.2	Arquitetura do Sistema. . . . .	37

3.3	Torre de <i>Hanoi</i> Posição Inicial e Posição Final [32]. . . . .	39
3.4	Movimentos Possíveis com 3 Discos [32]. . . . .	40
3.5	Identificação das Zonas de Trabalho da Célula. . . . .	40
3.6	Esquema de Funcionamento do Reconhecimento da Zona de Montagem. . .	41
3.7	Projeto Computer Aided Design (CAD) da Célula. . . . .	43
3.8	Projeto CAD Torre de <i>Hanoi</i> . . . . .	44
3.9	Projeto CAD dos Discos da Torre de <i>Hanoi</i> . . . . .	44
3.10	Projeto CAD Peça Cilíndrica. . . . .	45
3.11	Projeto CAD da Rampa de Peças Cilíndricas. . . . .	45
3.12	Tabuleiro para Colocação de Peças. . . . .	46
3.13	Projeto CAD Cabos de Navalha. . . . .	46
3.14	Projeto CAD da Rampa de Cabos de Navalha. . . . .	47
3.15	Imagem da Simulação Realizada em V-REP. . . . .	49
4.1	Calha para Passagem de Cabos e Calha de Fixação de Componentes. . . .	52
4.2	Montagem de Câmara e Garra no Robô [33]. . . . .	53
4.3	Ligações de Câmara e Garra no Controlador do Robô [33]. . . . .	53
4.4	Montagem Final do Quadro Elétrico. . . . .	54
4.5	Zonas de Iluminação Criadas na Bancada. . . . .	55
4.6	Calha de Encaixe para Fita Light Emitting Diode (LED). . . . .	55
4.7	Apoio para Calha de Iluminação. . . . .	56
4.8	Resultado Final da Implementação da Calha de Iluminação. . . . .	56
4.9	Instalação dos Equipamentos de Segurança. . . . .	57
4.10	Ligações no Robô do Botão de Emergência [19]. . . . .	57
4.11	Definição do TCP com a Montagem de Garra e Câmara. . . . .	58
4.12	Quadro de Calibração da Câmara para UR3. . . . .	59
4.13	Configurações das Entradas e Saídas de Segurança. . . . .	60
4.14	Configurações de Parâmetros dos Modos Reduzido e Normal. . . . .	60
4.15	Configurações da Ligação <i>Modbus</i> . . . . .	61

4.16	Esquema de Comunicações Robô-PC. . . . .	64
4.17	Variação da Posição da Garra consoante o Ângulo de Abertura [34]. . . . .	65
4.18	Programação dos Nós na Plataforma Node-RED. . . . .	66
4.19	Aplicação Gráfica a Correr no <i>Tablet</i> . . . . .	67
5.1	Resultado da Implementação da Iluminação. . . . .	70
5.2	Testes de Funcionamento dos Dispositivos de Segurança. . . . .	71
5.3	Testes de Paragem de Proteção do Robô. . . . .	72
5.4	Imagens de Funcionamento da Torre de <i>Hanoi</i> . . . . .	73
5.5	Imagens de Funcionamento do Cenário 1. . . . .	74
5.6	Imagens de Funcionamento do Cenário 2. . . . .	75
6.1	Foto do Evento Demonstrativo de Bragança [37]. . . . .	79

# Acrónimos

**cobot** Robô Colaborativo Industrial.

**3D** Desenho a Três Dimensões.

**ABB** Asea Brown Boveri.

**ACK** Acknowledgement.

**AGV** Autonomous Guided Vehicle.

**CAD** Computer Aided Design.

**DHCP** Dynamic Host Configuration Protocol.

**ESTiG** Escola Superior de Tecnologia e Gestão.

**HMI** Human–Machine Interface.

**I4.0@TMAD** Promoção da Indústria 4.0 na Região de Trás-os-Montes e Alto Douro.

**IoT** Internet das Coisas.

**ISO** International Organization for Standardization.

**LCAR** Laboratório de Controlo Automação e Robótica.

**LED** Light Emitting Diode.

**RGB** Red Green Blue.

**TCP** Tool Center Point.

**TMAD** Trás-os-Montes e Alto Douro.

**UR** Universal Robots.

**USB** Universal Serial Bus.

**V-REP** Virtual Robot Experimentation Platform.

# Capítulo 1

## Introdução

Nos últimos anos a indústria na região TMAD tem assistido a uma estagnação da evolução tecnológica, não sendo somente fruto da desertificação mas principalmente pelo tipo de indústria que a região comporta. As empresas desta região caracterizam-se, maioritariamente, em pequenas e médias empresas de produtos na sua maioria artesanais e personalizáveis por parte dos clientes, o que leva a que a sua produção seja principalmente executada por mão de obra humana [1]. Resulta então a necessidade de desenvolver novas ferramentas e métodos que permitam a evolução do tecido empresarial da região, permitindo construir produtos de uma forma rápida e eficiente, associada a baixos custos e, acima de tudo, que possuam boa qualidade para assim aumentarem a competitividade quer entre setores quer entre regiões ou países.

Razões como estas levam a que a indústria esteja obrigatoriamente sujeita a uma necessidade de melhoria contínua e, neste sentido é criado um consórcio de instituições públicas, liderado pela Escola Superior de Tecnologia e Gestão (ESTiG), que cria o projeto de Promoção da Indústria 4.0 na Região de Trás-os-Montes e Alto Douro (I4.0@TMAD), que visa identificar oportunidades que a indústria 4.0 oferece às empresas, familiarizá-las com novos conceitos e facilitar a passagem de conhecimento acerca de instrumentos tecnológicos existentes no mercado.

Após a realização de um diagnóstico tecnológico inicial com o objetivo de avaliar o grau de preparação das empresas para enfrentarem os desafios da 4<sup>a</sup> revolução industrial

pela região TMAD, foi possível retirar de entre outros aspetos que a maioria das empresas retinha o conceito mas na sua maioria não conseguiam explicar quer o conceito quer as suas principais valências ou tecnologias[1].

Neste contexto, e com o propósito de demonstrar conceitos e funcionalidades foram desenvolvidas cinco bancadas demonstrativas, nomeadamente: Aquisição e Análise de Grande Volume de Dados; Simulação de Sistemas de Automação; Integração do Humano em Ambientes de Automação Industrial; Inspeção Robótica Inteligente e Adaptativa; Sistema de Montagem com a Colaboração Humano-Robô; sendo esta última a principal bancada sobre qual recai o tema desta dissertação.

Na indústria de manufatura, o papel das células robotizadas e, em particular, células robóticas colaborativas e flexíveis, torna-se fulcral no sentido que permite garantir celeridade nos processos de fabrico, acomodar a introdução de alterações regulares nos produtos fabricados e permite que as alterações do produto não acarretem investimentos avultados, assim como garantem a integridade física dos operadores. Estas células aliam a regularidade no funcionamento ao rigor no fabrico, aos baixos custos de produção e a uma elevada qualidade nos resultados das tarefas que desempenham. A introdução deste tipo de tecnologias não passa por promover a substituição da mão de obra humana mas sim procura preservá-la e tomar partido dos benefícios dos dois intervenientes o humano e robô.

Fazem parte integrante das células robotizadas colaborativas, maioritariamente, os robôs manipuladores. Estes podem desempenhar variadíssimas tarefas e garantir a qualidade do desempenho conjugada com a adaptabilidade do operador. No que toca às empresas a utilização de robôs passou para de uma fase em que os robôs eram encarados como um meio para atingir um fim (mão de-obra mais barata) a uma fase em que os objetivos mais importantes são mais qualidade, adaptabilidade de sistemas e segurança para os seus operadores, qualidades estas só possíveis com robôs colaborativos. Do ponto de vista dos operadores esta nova integração de robótica permite libertá-los do desempenho de tarefas pesadas e repetitivas, mas não desprezando as suas qualidades. Assim, cada vez mais, a indústria aposta na automatização robotizada assim como todas as tecnologias

trazidas pela vulgarmente chamada indústria 4.0 de modo a poderem obter produtos de forma mais rápida, com menos desperdícios, de melhor qualidade, com personalização por parte do cliente e com um baixo custo.

## 1.1 Objetivos

O principal objetivo desta dissertação é projetar e desenvolver a montagem de uma célula robótica colaborativa cobrindo todas as fases do desenvolvimento do produto, e visando servir de demonstrador das tecnologias Indústria 4.0 no âmbito do projeto I4.0@TMAD.

Numa primeira fase o trabalho de projeto consiste em aplicar em simulação um processo de montagem simplificado para aferir o dimensionamento da célula e a necessidade de adquirir os equipamentos necessários ao funcionamento para uma gama diversificada de aplicações.

Posteriormente à aquisição dos diversos equipamentos segue-se a fase de montagem e integração dos diversos componentes que compõem o sistema.

Por fim, serão elaboradas aplicações demonstrativas, devidamente enquadradas na finalidade do projeto, permitindo demonstrar o maior número de potencialidades, e constituindo desta forma um atrativo para o público alvo, nomeadamente para aqueles que tomam as decisões na melhoria tecnológica das empresas.

## 1.2 Estrutura do Documento

Este documento encontra-se dividido em cinco capítulos:

- No primeiro capítulo, faz-se uma introdução geral, que serve para explicar a motivação, a necessidade e os objetivos pretendidos com a realização desta dissertação.
- No segundo capítulo são abordadas as principais tecnologias e equipamentos que fundamentam os trabalhos realizados durante a realização do projeto, uma breve

introdução à robótica e às suas aplicações, mas dando principal relevância à sua aplicação no projeto;

- No terceiro capítulo, descrevem-se os requisitos de concepção descrições dos cenários demonstrativos.
- O quarto capítulo diz respeito à implementação das ideias onde está retratado o processo de montagem da célula e instalação dos seus componentes assim como programação;
- No quinto capítulo são apresentados os testes experimentais onde são retratados o funcionamento de cada cenário e as funcionalidades do sistema, arquiteturas aplicadas nos sistemas de visão, testes de segurança do equipamento e análise geral do funcionamento da célula.
- Por fim no ultimo capítulo são apresentadas as conclusões retiradas da dissertação realizada e propostas para trabalhos futuros.
- Em anexo pode ser encontrada documentação técnica referente aos esquemas elétricos, tutorial de programação para introdução ao robô colaborativo e foto-descrição da execução dos cenários pré estabelecidos.

# Capítulo 2

## Estado da Arte

Na robótica a escolha do tipo de tecnologia depende muito dos fatores custo e rentabilidade associada à sua utilização, esta porém está associada à competitividade entre empresas e até a inovação em produtos e requisitos ligados ao volume de produção. O primeiro desenvolvimento de um robô industrial data de 1959 por George Devol e Joseph Engelberger, (Figura 2.1), que pesava duas toneladas e usava atuadores hidráulicos [2].

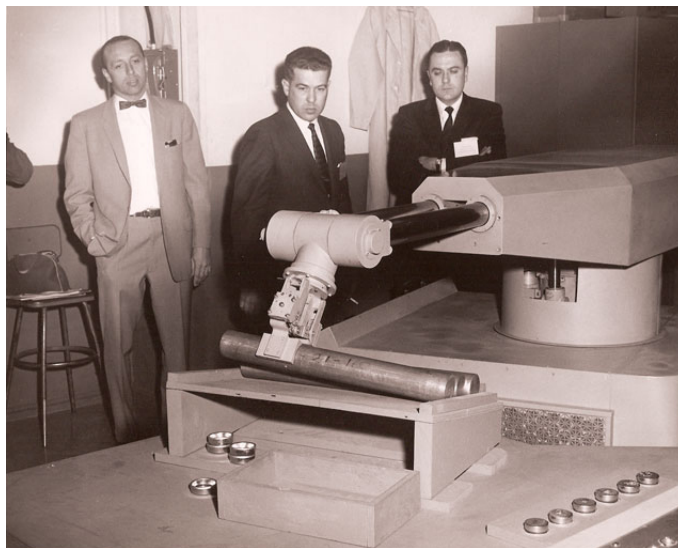


Figura 2.1: Primeiro Robô Industrial [2].

Muito embora hoje em dia as grandes empresas estejam cada vez mais modernizadas e automatizadas para a produção em massa, as empresas de pequena e média dimensão

não o estão, estando sim viradas para a produção de um determinado produto com algum tipo de personalização realizada por parte dos clientes e ainda com critérios de qualidade superior competindo assim com grandes concorrentes. É aí que entra a automatização de processos, neste caso não preocupada com a quantidade, mas sim com a qualidade e diversidade de produtos visto que as empresas não produzem para terem grandes stocks, produzem apenas em pequena ou média escala.

Os desafios de integração e desenvolvimento têm como base a interação dos operadores humanos com os robôs manipuladores, pois estes coabitam em ambiente industrial e para que isso seja possível é necessário criar interfaces que possibilitem a sua comunicação/ interação. Esta nova realidade de colaboração homem máquina acarreta diversos desafios à área de integração e desenvolvimento, pois é necessário criar dispositivos e *software* de Human–Machine Interface (HMI), bastante simples, intuitivos e, acima de tudo, seguros.

A segurança torna-se a primeira e principal preocupação de um instalador pois embora o nível de repetibilidade do sistema automatizado seja fiável, a segurança pode ser sempre posta em causa devido à baixa previsibilidade humana.

Os processos de produção foram investigados e atualizados ao longo dos anos e encontram-se organizados na maioria dos casos em células flexíveis de fabrico, cada uma delas realiza um determinado número de tarefas sobre as matérias-primas ou produtos semi-acabados, até que estes cheguem a ser o produto final. O percurso entre o armazém de matérias-primas até ao armazém de produtos acabados apresenta várias fases, que estão associadas à transição de uma célula flexível para a seguinte.

Na figura 2.2 é descrita de forma sucinta o surgimento das tecnologias e das diferentes eras industriais.

- A indústria 1.0 data do século XVII é retratada desta forma como a primeira revolução industrial impulsionada pela máquina a vapor ou seja pelo aparecimento da produção mecanizada.
- A indústria 2.0 surge já no século XIX com a era da eletricidade onde surgem os motores elétricos, surgem também as primeiras linhas de produção.

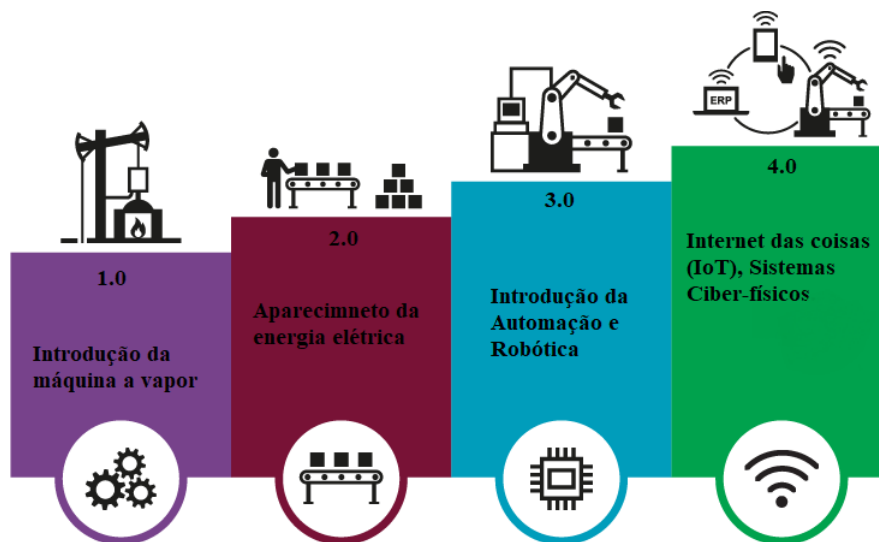


Figura 2.2: Retrato das Evoluções Industriais (adaptado de [3])

- Por indústria 3.0 entende-se a era da automatização já no século XX. É ainda nesta fase que surge a eletrônica e os microprocessadores e são introduzidos os robôs para substituir o homem nas linhas de montagem.
- As evoluções tecnológicas retratadas ao longo das anteriores revoluções trouxeram avanços notáveis a nível mundial no ramo industrial e por isso surge a era da indústria 4.0, onde são introduzidos os sistemas ciber-físicos, onde máquinas tomam as suas próprias decisões comunicando com outras máquinas ou acedendo a diversos tipos de dados, batizada como a era da digitalização.

O mercado nacional não ficou atrás dos demais países e hoje em dia já é possível ouvir por parte dos empresários nacionais referências à introdução de princípios e tecnologias associadas à Indústria 4.0 nos seus processos produtivos.

## 2.1 Indústria 4.0 e Suas Premissas

Com o desenvolvimento da Internet, *software* e *hardware* cada vez mais sofisticado, a capacidade das máquinas aprenderem e colaborarem, iniciou-se uma transformação industrial. Esta transformação foi chamada por Erik Brynjolfsson e Andrew McAfee como

a segunda idade da máquina e em 2013 na feira Industrial de Hannover, na Alemanha, falava-se pela primeira vez no conceito de indústria 4.0 [4].



Figura 2.3: Tecnologias e Conceitos da Indústria 4.0 [5].

Outros termos, como por exemplo “*smart factory*”, “*intelligent factory*”, “*factory of the future*”, são termos similares que descrevem uma visão do que será uma fábrica no futuro inspirada no termo indústria 4.0. A indústria no geral com esta etapa revolucionária será mais inteligente, flexível e dinâmica [4].

Dentro das tecnologias reveladas na Figura 2.3 podem ser realçados alguns dos pilares da indústria 4.0 sendo eles [4]:

- Big Data e Análise de Dados: Capacidade de Interligar dados de máquinas, procurando com isso prever ou adaptar de forma otimizada a produção, prevendo manutenções nos equipamentos e a redução de produtos com defeito.

- Internet das Coisas (IoT) e cyber-segurança: resumidamente este ponto é descrito pela capacidade de interligar equipamentos para que seja possível a partilha de informações para concluir determinadas tarefas, respeitando normas de segurança de dados.
- Impressão 3D : produção de peças desenhadas em CAD, através de fabrico aditivo, excluindo o uso de moldes tradicionais.
- Sistemas colaborativos: *Software* capazes de interagir com outras máquinas e seres humanos, promovendo a inteligência artificial e robótica colaborativa também é um ponto a salientar.
- Simulação Virtual: valência para testar ou viabilizar sistemas com o objetivo de conseguir a máxima otimização de processos e produtos quer na fase de projeto ou por ventura para alterações por exemplo de *layout* e avaliar as consequências.

## 2.2 Células Industriais

Uma célula robotizada é descrita como uma área de trabalho devidamente estudada e projetada com um objetivo de produção comum, onde existem distribuições de tarefas, interligando equipamentos, na qual estará um ou mais robôs e de diversos sistemas de automação permitindo assim a otimização e realização das operações. A utilização de um robô pressupõe a utilização de uma célula industrial pelo que para o seu perfeito funcionamento são necessários diversos sistemas como [6]:

- Robôs industriais e ferramentas - Robô com integração de ferramentas que forma o principal conjunto, uma vez que é este o conjunto a realizar a tarefa principal;
- Sistemas de segurança - referem-se a sistemas que garantam a proteção do equipamento e operadores;
- Interface homem-máquina: responsável por informar o operador e ou a base de dados do estado de operação, ocorrência de falhas entre outros.

- Sistemas de alimentação/remoção de produtos - disponibilizam o produto e após operação, terão de ser retirados por outro sistema, o de remoção, por exemplo rampas de *stock*;
- Sistemas de posicionamento do produto - referem-se à colocação do produto para o robô operar sem haver movimento por exemplo um molde ou fixador;

Podem ainda ser introduzidos sistemas de inspeção ou outros sistemas mais específicos para cada utilização.

## 2.3 Sistemas Robóticos

O uso da robótica industrial vai desde as micro-empresas até às maiores empresas, sendo que, por vezes, estas chegam a ter número semelhante de trabalhadores e de manipuladores instalados. A indústria automóvel é o grande cliente de manipuladores industriais onde realizam tarefas de soldadura e manipulação.

A aplicação da robótica passou por ser um ótimo substituto de mão de obra humana, ou seja, poupança de recursos visto um só robô conseguir trabalhar 24h sobre 24h, baixando o custo de produção e poupando turnos de funcionários.

### 2.3.1 Classificação

Os robôs podem ser classificados utilizando critérios como a autonomia do sistema de controlo, mobilidade da base, estrutura cinemática, forma de acionamento, graus de liberdade, geometria do espaço de trabalho, capacidade de carga, velocidade e precisão [7].

Na Figura 2.4 descrevem-se os diferentes tipos e divisões conforme as características e aplicabilidade de cada robô. Existe logo uma grande divisão referente à mobilidade, a estrutura cinemática e por último ao tipo de ambiente em que o mesmo irá realizar.

- Humanoides: Tal como o nome indica, têm o aspeto do ser humano (Figura 2.5) tem cabeça, troco e membros superiores e inferiores. “Que fã de ficção científica

Sistema de controle	Mobilidade da base	Estrutura Cinemática	Espaço de trabalho
Equipamentos teleoperados	{ Veículo teleoperado Manipulador teleop.		
Robôs	Móveis	{ Aquáticos { Aéreos Terrestres	{ Submarinos Marinos { Pernas Rodas
	Fixos	{ Paralelos Série	{ 3 - 6 GdL Cartesiano Cilíndrico Esférico Articulado SCARA

Figura 2.4: Classificação de Robôs [8].

pode dizer honestamente que, depois de assistir a *Star Wars*, não sonhou em ter um C-3PO” [9].



Figura 2.5: Robô StarWars C-3PO [10].

- Móveis: Os robôs móveis são capazes de se moverem de uma forma autônoma. As movimentações efetuadas são com base no ambiente que os rodeiam, recolhendo sinais através de sensores de vários tipos. Estes robôs podem ter como meio de locomoção rodas ou lagartas [7]. Os robôs móveis estão neste momento no auge da sua existência visto que muitos centros de investigação estão a tentar espremer todas as suas potencialidades. São encontrados na indústria Autonomous Guided Vehicle (AGV) (Figura 2.6), em instalações militares, em ambientes de segurança,

e como produtos de consumo, seja para o entretenimento ou para realizar alguns trabalhos como limpeza ou corte de relva.



Figura 2.6: Robô *AGB MiR100<sup>TM</sup>* [11].

- Manipuladores: São capazes de manusear peças ou ferramentas (Figura 2.7), através de movimentos programados para a realização de uma variedade de tarefas como soldadura, embalagem entre outros, sendo estes abordados com mais detalhe no próximo capítulo [7].



Figura 2.7: Robô *ABB IRB1410* [12].

### 2.3.2 Robôs Manipuladores

Normalmente, um robô industrial, independentemente da sua potencial aplicação, é mecanicamente concebido para posicionar e orientar no espaço o seu eixo terminal: uma garra ou uma ferramenta [13]. A aplicação destes equipamentos em vez da mão de obra humana é aplicada pelos mais variados motivos:

- Trabalho perigoso para seres humanos: quando o ambiente no qual a tarefa é desempenhada é perigosa, insegura ou nociva. (exemplos: fundição, pintura por pulverização ou soldadura)
- Peças de difícil manuseio: objetos com peso elevado ou com grande volume.
- Operação contínua: retorno financeiro é rápido visto um só robô poder ser equivalente a vários trabalhadores, uma vez que pode trabalhar ininterruptamente.
- Ciclo de trabalho repetitivo: por entre algumas vantagens da utilização de um robô industrial esta é principal característica que tende a promover o uso de um robô em detrimento de um operário humano. Se a sequência de elementos no ciclo for a mesma e os movimentos simples, um robô geralmente consegue desempenhar o ciclo de trabalho com mais consistência e repetibilidade que um operador [13].

A totalidade das aplicações destes equipamentos são variadas e a decisão para a utilização são impostas pelos seguintes parâmetros [7]:

- Graus de liberdade – o número eixos de um robô influencia a decisão como por exemplo para alcançar qualquer ponto num plano são necessários pelo menos dois eixos, ou para alcançar um ponto no espaço são necessários pelo menos três eixos. Para controlar completamente a orientação do extremo de um braço, são necessários outros três eixos.
- Cinemática – a disposição dos membros e das juntas determina os possíveis movimentos.
- Área de trabalho – o local onde um robô pode trabalhar.
- Capacidade de carga – quanto peso um robô pode levantar.
- Velocidade precisão e repetibilidade – rapidez e proximidade com que um robô pode posicionar o seu TCP. Dois importantes parâmetros característicos dos robôs são a sua precisão e repetibilidade.

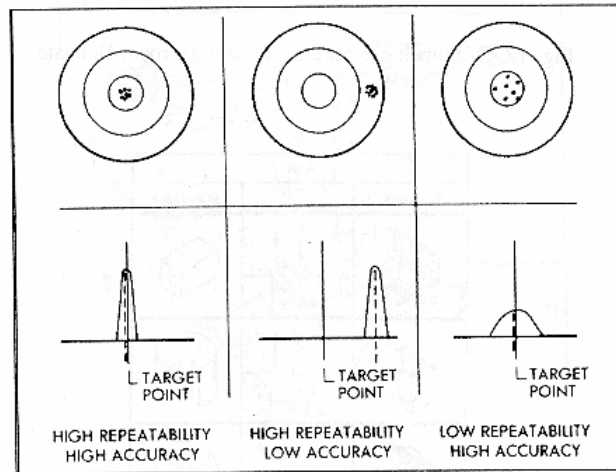


Figura 2.8: Diferença Entre Precisão e Repetibilidade do TCP [14].

Por precisão, entende-se a capacidade do robô de ir a uma posição desejada, com respeito a um sistema de referência fixo (base) e com um erro definido. Já a repetibilidade, entende-se a capacidade do robô de, uma vez conhecida e alcançada uma posição, e partindo-se da mesma condição inicial, voltar a ir ("repetir") novamente a posição com um erro determinado. Ambos os parâmetros são mostrados na (Figura 2.8) [13].

- Controlo dos movimentos – para algumas aplicações o robô precisa apenas executar repetidamente um número limitado de posições pré-programadas. Para aplicações mais específicas como a soldadura devem ser levados em consideração o movimento velocidade e orientação do TCP.
- Fonte de energia – alguns robôs utilizam motores elétricos, enquanto outros utilizam atuadores hidráulicos estando em causa a relação Velocidade versus Força, respetivamente.
- Acoplamento – Tipo de conexão entre as juntas visto estas poderem ser diretamente implementadas com motores ou através de caixas de redução.

Os braços robóticos usados para manipulação de objetos ou posicionamento de ferramentas, foram desenvolvidos para imitar o “braço humano” na realização de tarefas.

Deste modo é possível fazer uma analogia dos sete graus de liberdade do “braço humano” para o braço robótico, uma junta de três graus de liberdade (o ombro), seguida de uma junta com um grau de liberdade (o cotovelo) e por outra com três graus de liberdade (o punho). A grande maioria dos robôs manipuladores tem, geralmente, seis eixos, que lhe permitem atingir qualquer posição/orientação no seu espaço de trabalho, desde que não atinjam os pontos de singularidade [14].

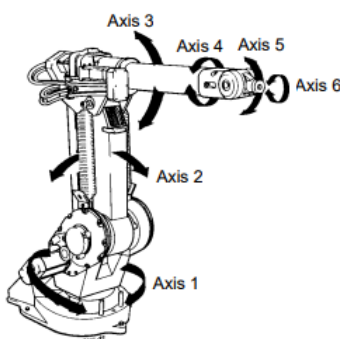


Figura 2.9: Braço Manipulador IRB 1400 [12].

A existência das singularidades representa uma perda de mobilidade no manipulador industrial, mas, desde que conhecidas, estas podem ser evitadas aquando da realização do trabalho. Como no braço humano, os robôs manipuladores industriais usam as primeiras juntas para posicionamento (no homem representam o cotovelo e o ombro) das juntas seguintes. As últimas juntas dos robôs, são utilizadas para pequenos ajustes de posição ou orientação da ferramenta (Figura 2.9) [14]. Também a limitação da rotação pode ser um problema facilmente contornável visto que colocando o robô numa posição diferente, o ponto pode ser alcançado sem problemas.

### 2.3.3 Sistemas de Controlo

Um robô é constituído por um conjunto de, no mínimo, três elementos fundamentais para possibilitar o seu funcionamento e programação são eles o manipulador, controlador e consola de programação (Figura 2.10).

O manipulador é um dispositivo que desenvolve a tarefa para o qual foi programado.



Figura 2.10: Robô UR5 com Unidade de Controlo e Consola [15].

A sua constituição é caracterizada por um número de segmentos metálicos, unidos entre si por juntas. É caracterizada por ter uma base assente numa célula, bancada ou pedestal e a outra extremidade ser móvel, a partir da qual é possível executar determinada tarefa ou acoplar alguns tipos de ferramenta para desenvolverem o trabalho pretendido [16].

O controlador é um dispositivo destinado a controlar todos os movimentos executados pelo manipulador, processa a informação recebida do operador e a adquirida através de sensores é capaz de efetuar operações de cálculo que, depois são enviadas para os atuadores e para o manipulador. Paralelamente a isto, o controlador pode efetuar tarefas como o registo de dados em memória e a gestão das comunicações com o operador ou com outros dispositivos externos [16].

O terceiro componente que se revela imprescindível para a programação do robô é a consola de programação. A consola, para além de possibilitar a programação do robô através de *software* adequado, também tem funções de configuração e monitorização da laboração e de diagnóstico de falhas. É na maioria dos casos extremamente intuitiva e móvel e permite mover o braço robótico de forma manual [16].

## 2.4 Sistemas Robóticos Colaborativos

O conceito Robô Colaborativo Industrial (*cobot*), surgiu pela primeira vez em 1995, como parte de um projeto de investigação realizado pela Fundação General Motors [17]. Um sistema é considerado colaborativo quando existe ou possa existir uma interação, em tempo-real, entre o operador e o robô durante a execução de uma tarefa. Lado a lado

com a robótica colaborativa vem o termo segurança que neste tipo de robôs é levado ao topo da sua inovação. A interação entre o operador e o sistema visa reproduzir os movimentos físicos (membros superiores e/ou inferiores, mãos, dedos) do operador. O termo colaborativo aplicado aos sistemas robóticos permite distinguir entre dois modelos: a telemanipulação e a manipulação colaborativa. Na telemanipulação existe uma interação indireta entre o operador e o sistema robótico [18].

Os *cobots* são destinados à interação direta com um trabalhador humano, lidando com uma carga partilhada. Sendo estes, uma subdivisão dos braços manipuladores industriais convencionais tendo como principal diferença a não necessidade de divisão de áreas de trabalho entre operador e robô e também pela preocupação por parte dos construtores na construção do corpo destes robôs uma vez não tem esquinas nem zonas que possam hipoteticamente ferir ou entalar operadores.

É necessário ainda separar os *cobots* dos robôs teleoperados, nos quais um operador humano controla um robô e uma carga remotamente. Estes equipamentos interagem com as pessoas produzindo “superfícies virtuais” definidas por *software* que limitam e guiam o movimento da carga partilhada, olhando sempre ao fator segurança. Os benefícios ergonômicos e de produtividade resultam da combinação da força e da interface do computador do *cobot* com a percepção e a destreza do trabalhador humano [17].

O objetivo da utilização destes equipamentos é conseguir que seja possível trabalhar com as pessoas lado a lado sem supor nenhum tipo de contratempo. Hoje em dia mais de vinte anos depois da sua aparição, os *cobots* estão presentes nas mais variadas fábricas e também na consciência do público em geral. Entretanto, ainda são muitos os que não conseguem compreender a diferença entre um robô tradicional e um *cobot* [19].

As principais Vantagens de Braços Robóticos Colaborativos são:

- Equipas de humanos e máquina: Os robôs da indústria clássica realizavam o seu trabalho a seguirem um programa fixo, o que incluía não ter em conta as pessoas que trabalhavam com estes. De facto, neste caso, os acidentes evitam-se mantendo espaços de segurança. Estes *cobots* estão desenhados especificamente para trabalhar

com pessoas. Deste modo, em vez de estarem "enjaulados", trabalham num ambiente cooperativo e ajudam com tarefas complexas que podem não ser completamente automatizadas.

- Mais económico: Um dos aspetos mais atrativo dos *cobot* é o seu baixo custo se comparado com o preço de um robô convencional. A possibilidade de serem reutilizados em diferentes funções de maneira rápida e serem de fácil programação permite que a sua instalação seja viável em qualquer tipo de empresa de qualquer sector.
- Comportamento inteligente e seguro: Estes equipamentos estão desenhados para trabalhar na mesma área que os operadores. Com sensores sofisticados, os *cobot* imobilizam ao mínimo contacto, o que permite evitar qualquer perigo para as pessoas. Isso faz com que as áreas fechadas e os espaços de segurança já não sejam necessários.
- Flexíveis e com capacidade de aprendizagem: Os *cobot* são muito fáceis de programar. Por exemplo, um técnico pode realizar um movimento com o braço do *cobot* e este pode reproduzir o mesmo movimento de forma automática, o que reduz consideravelmente qualquer processo de programação.
- Aplicável em qualquer lugar: Os *cobots* não só são fáceis de reprogramar, também são relativamente fáceis de mover e de utilizar noutros pontos de linhas de produção. A maioria dos *cobot* podem ser montados em qualquer orientação, desde o solo até as paredes e tetos.

Como desvantagens podem ser apontadas a perda de velocidade em relação aos robôs convencionais e a falta de robustez que pode ser impeditivo para implementar em algumas áreas específicas.

Exemplo disso é o robô colaborativo UR3 (Figura 2.11) que apresenta as características ideais para tarefas leves de montagem e trabalhos que exigem precisão absoluta. Com rotação de 360 graus em todas as articulações e rotação infinita na junta final. O Universal Robots (UR)3 torna-se num dos robôs mais flexíveis e versáteis do mercado

atualmente, as principais características que sobressaem são a automatização de tarefas até 3 kg e o alcance de raio até 500 mm, importa ainda salientar que embora não seja uma característica fundamental o peso do robô é de apenas 11 kg [15].

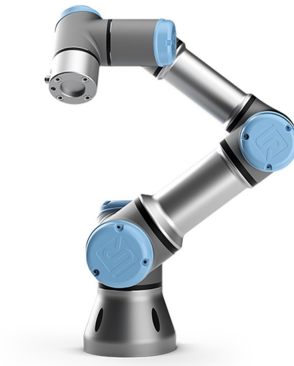


Figura 2.11: Robô Colaborativo UR3 [15].

A empresa UR é especialista na construção de *cobot* e leva em consideração diversos aspectos da sua construção como precisão, velocidade, otimização ou ergonomia. Olhando à qualidade torna-se um produto extremamente fiável e com um preço competitivo e com tempo de recuperação do investimento médio de apenas 195 dias [15]. A oferta da empresa é pequena em termos de variação de modelos existem apenas UR3, UR5 e UR10, sendo a carga útil de cada robô 3kg, 5 kg e 10kg respetivamente.

Aliada a todas as características destes robôs está a definição fundamental de colaboração e de segurança, os robôs UR podem trabalhar lado a lado com operadores, já que a sua sensibilidade de força integrada faz com que os robôs parem de funcionar automaticamente quando se deparam com obstáculos. Além disso, o robô pode ser programado para operar em modo reduzido quando uma pessoa entrar na sua área de trabalho. O conceito de ser um robô colaborativo passa também por ser intuitivo, acessível e de fácil instalação [15].

## 2.5 Tipos de Ferramenta

A ferramenta que um manipulador tem na sua extremidade depende do tipo de tarefa que o robô vai desempenhar. Da mesma forma que existe uma enorme diversidade de aplicações para os robôs desenvolverem trabalho, também existem variadas ferramentas de trabalho adequado.

Ao falar de ferramentas é indispensável dividir em dois grandes grupos: as garras e as ferramentas.

### 2.5.1 Garras

O grupo das garras pode, por sua vez, ser dividido também em dois grupos: as mecânicas e as não mecânicas. As garras mecânicas usam dedos mecânicos ativados por um motor ou ar comprimido, já as garras não mecânicas funcionam através de sucção ou por ventosa.

Para manipulação, geralmente são usadas pinças com dois dedos (Figura 2.12), e ainda casos em que seja necessário utilizar mais que uma ferramenta. A configuração da garra também depende da maneira como se faz o *pick* do objeto.



Figura 2.12: Garra Mecânica *ROBOTIQ 2 Finger* [20].

A utilização de garras *ROBOTIQ* é aconselhada pela UR, até porque para além da sua conceção ter sido inicialmente direcionada para robôs da UR a sua programação é extremamente simplificada e tal como nos robôs são disponibilizados diversos parâmetros internos e é possível programar de diversas formas (exemplo: ler corrente consumida ou

ângulo de abertura). Como vantagens salienta-se que é totalmente elétrica e não precisa de outra conexão externa ao robô, sendo usado uma conexão Universal Serial Bus (USB) como meio de comunicação. A sua construção não tem partes cortantes ou passível de entalamento e está altamente preparada para preservar a integridade dos operadores que trabalharemos com os *cobots*.

Para o caso de garras não mecânicas podem existir sistemas que através da indução permitem a movimentação de peças, com ventosas para fazer o efeito de sucção, com ganchos ou com garras adesivas que torna a sua utilização mais específica.

### 2.5.2 Ferramentas e Sensores Adicionais

No campo das ferramentas, existem inúmeras aplicações e por isso mesmo uma infinidade de soluções existentes, por exemplo para o caso da soldadura, a ferramenta do robô é uma tocha de soldar com um aspeto muito similar às tochas manuais. Nos casos em que a tarefa é de pintura, o robô tem uma pistola de pintar ou sistema equivalente que permite projetar a tinta na superfície pretendida.



Figura 2.13: Sensor de Força FT300 ROBOTIQ [20].

Uma vez que o presente trabalho é sobre “pick and place” a ferramenta a utilizar seria uma garra mas, a integração de um sensor de força (Figura 2.13) que, tornaria por exemplo a precisão de operação muito mais eficaz assim como seria possível executar outro

tipo de atividades como montagem de *motherboards* em carcaças, aumentando assim a fiabilidade do sistema implementado [21].

Outra ferramenta que faz interesse representar é a câmara, (Figura 2.14) também da ROBOTIQ que para além da existência de compatibilidade entre câmara e robô, esta pode ser aplicada diretamente com a garra ou até com sensor de força ampliando assim as funcionalidades de qualquer sistema. Desta forma o robô pode funcionar de forma quase autónoma a identificar objetos, adaptar a sua orientação conforme a posição do objeto tendo para isso já algoritmos criados de tratamento de imagem e por esse motivo a sua implementação está bastante simplificada [20].



Figura 2.14: Wrist Câmara ROBOTIQ [20].

Estes são alguns exemplos de ferramentas, mas como mencionado anteriormente, existe uma grande diversidade de ferramentas dependendo do tipo de aplicação/finalidade visada.

## 2.6 Processamento e Simulação

As empresas durante anos tomaram decisões, pela experiência de profissionais. Com base nesses dados, definiam recursos, projetos, *lay-outs*, entre outros. Durante a laboração do dia a dia, vinham os problemas, os custos desnecessários, as falhas, ou seja, problemas que poderiam ser observados, eliminados ou reduzidos significativamente, mediante a simulação computacional prévia.

Por isso, torna-se fulcral a necessidade de antes da tomada de decisão, levar a cabo,

por meio dos mais variados simuladores de processos, simular quantas vezes for necessário antes da tomada de decisão, nessa condição possa ser feita uma previsão de curto, médio e longo prazo, podendo avaliar os recursos, observar melhores fluxos de processo, analisar volumes de produção, *stocks* entre outros.

### 2.6.1 V-REP

O simulador de robótica V-REP possui um ambiente integrado, em que a arquitetura de controlo consiste em que cada objeto ou modelo Desenho a Três Dimensões (3D) pode ser controlado de forma independente através de um *script*. Isto faz do V-REP uma ferramenta ideal e versátil para aplicações multi-robôs e até multi-equipamentos. Os controladores podem ser escritos em várias linguagens, nomeadamente em C/C++, *Python*, *Java* ou *Lua*. Algumas das aplicações que podem ser executadas pelo V-REP são a simulação de sistemas de automação fabril, monitorização remota, controlo de *hardware*, verificação e prototipagem rápida e monitorização segura.

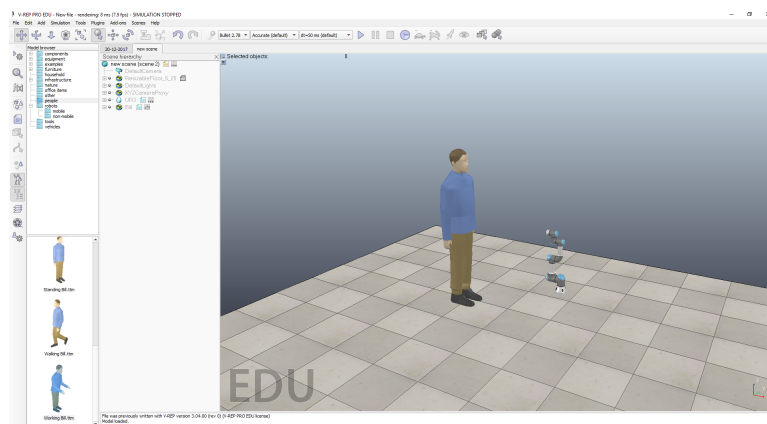


Figura 2.15: Ambiente de Simulação V-REP.

E pode ainda ser articulada com aspetos físicos como simulação da interação de um humano conforme verificado na Figura 2.15. Pode ainda ser integrado com aspetos reais como por exemplo, pode haver a leitura de um sensor e esta ter influência na simulação [22].

## 2.6.2 Robotstudio

O RobotStudio é um sistema de Simulação e *software* de programação *offline* da Asea Brown Boveri (ABB), permite entre outras coisas que a programação do robô seja feita num computador sem parar a produção no caso de ser um robô já em produção ou então estudar o melhor *layout* para o seu funcionamento. Permite fazer toda a programação real de um robô em ambiente virtual e depois enviar o código para o robô físico e fazer *play*.

O RobotStudio é construído no "*Virtual Controller*" da ABB, uma cópia exata do *software* real que roda nos robôs de produção. Isso permite que simulações extremamente realistas sejam realizadas através do uso de programas de robô e arquivos de configuração idênticos aos arquivos usados em produção [23].

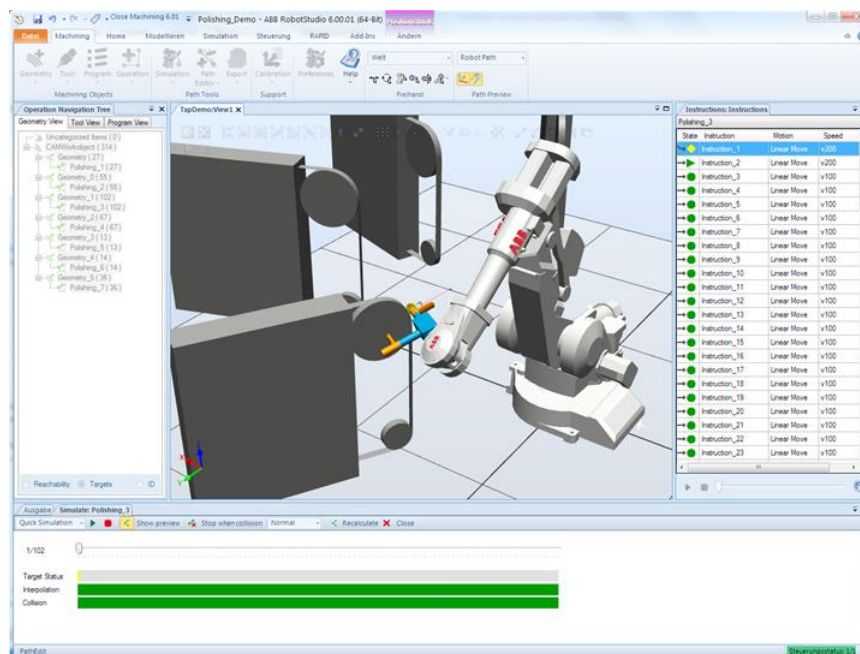


Figura 2.16: Ambiente de Simulação RobotStudio [24].

Este *software*, representado na Figura 2.16, tem apenas um senão, apenas é permitido trabalhar com equipamentos da ABB pelo que se a necessidade de simulação passar pela utilização de um outro robô a sua escolha deve ser reconsiderada.

### 2.6.3 Solidworks

O SolidWorks é um *software* de 3D CAD e surgiu em 1995. Este programa baseia-se em computação paramétrica, criando formas tridimensionais a partir de operações geométricas elementares. No ambiente do programa, a criação de um sólido ou superfície, tipicamente começa com a definição de um *sketch* desenhado a duas dimensões que depois é transformado através de uma operação num modelo tridimensional. A sua grande utilização deve-se sobretudo à facilidade de trabalhar com este *software* e com as compatibilidades do mesmo.

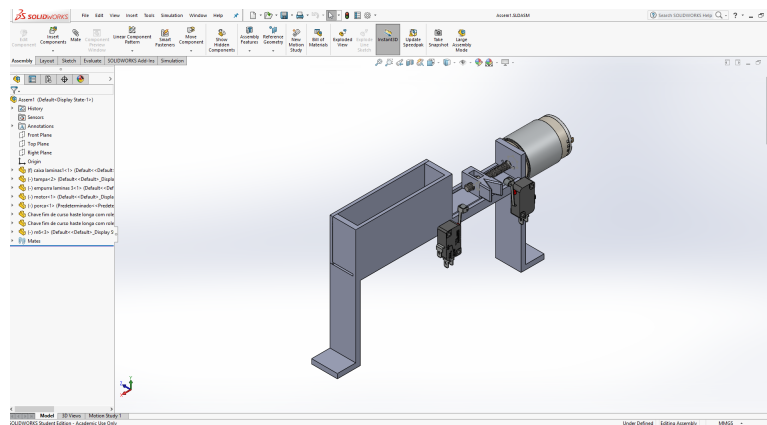


Figura 2.17: Ambiente de Simulação Solidworks.

Este programa, apresentado na Figura 2.17, dispõe de um amplo leque de funcionalidades, incluindo funções específicas para chapa metálica, construção soldada e produção de moldes. A sua utilização prende-se não só à necessidade de virtualização para a construção mas também para a simulação de processos e construções. As suas aplicações passam pela mais variadas áreas mas como em todos os programas deste tipo a sua utilização mantém-se bastante presente na indústria automóvel. Através deste programa é possível, por exemplo, desenvolver uma réplica de um robô real, considerando as suas dimensões e massa reais e posteriormente fazer testes de resistência ou dinâmicos [25].

## 2.7 Comunicações Industriais

A integração de vários sistemas obriga a existência de redes de comunicação. No campo da automação não aparenta ter solução fácil dada a falta de compatibilidade direta entre distintos equipamentos de diferentes fabricantes. Mesmo que seja desenvolvida uma nova rede aberta que permita uniformizar a maioria das soluções existentes já era por si só complicado, visto que a maioria do equipamento instalado não permitir diferentes aplicações, o investimento necessário não seria na maioria dos casos viável [26].

A solução parte por adotar “sistemas abertos”, o objetivo é que os dispositivos de um fabricante possam interagir com produtos de um fabricantes diferentes. Para que tal aconteça, é necessário que um dos fabricantes utilize protocolos *standard*. Se isto acontecer, cada utilizador poderá comprar os dispositivos que entender, tendo em conta o preço e a finalidade a que se destinam. A utilização de protocolos *standard* conduz à redução de preços e ao aumento da flexibilidade no que respeita à compra de dispositivos de comunicação.

Sem protocolos, uma rede de comunicação não funciona, uma vez que estabelece a forma como um programa deve preparar os dados para serem enviados ao nível seguinte do processo de comunicação. Podem ser comparados a idiomas, ou linguagens, que servem para estabelecer a comunicação entre os equipamentos na indústria.

De entre os protocolos existentes, os mais utilizados em sistemas robóticos ou de automação industrial são:

- *CAN*: utilizado maioritariamente na indústria automóvel e em sistemas de automação. A informação é realizada através de comunicação série, tendo as mensagens um formato fixo. Este protocolo funciona com mecanismos de prioridades e possui um sistema de deteção de erros fiável.
- *Ethernet*: encontra-se limitado a 100 metros de extensão e, necessita de *hubs*. É segura e, aquando da comunicação, existe pouca largura de banda.
- *Modbus*: protocolo aberto, desenvolvido como um simples meio de troca de dados

entre controladores e sensores, utilizando mestre-escravo / cliente-servidor. Difundido a larga escala e bem aceite pelo meio industrial, tornou-se num dos protocolos mais implementados em diversos produtos de diversos fabricantes para a área da automação industrial. Não apresenta robustez notável, mas a sua simplicidade permite flexibilidade de aplicação em diferentes problemas industriais faz com que seja o protocolo mais utilizado [27] .

## 2.8 Sistemas de Visão

Em visão por computador para reconhecimento de objetos ou pessoas são utilizadas técnicas de deteção de pontos de interesse em objetos ou superfícies. Isto é feito através da extração de características, que devem ser invariantes em relação à orientação, escala e localização. Há dois aspetos a ter em conta no que diz respeito a uma característica: a deteção de um ponto-chave, que identifica uma área de interesse, e a sua descrição que caracteriza a sua região. Normalmente é identificada uma região que contém uma grande variação de intensidade, como um canto ou uma aresta, o centro do objeto é designado por ponto-chave ou no caso de um humano pelo tom de pele. A sua descrição é geralmente determinada, medindo a orientação dos seus pontos mais próximos, dando origem a um vetor que identifica um dado ponto-chave.

A Microsoft Kinect é um dispositivo, lançado em Novembro de 2010, composto por uma câmara Red Green Blue (RGB), sensores de profundidade, um conjunto de microfones e um motor para inclinação. As principais características deste equipamento são[28]:

- Câmara RGB, com 8 bits por cada canal de cor, uma resolução de 640x480 píxeis com um *framerate* máximo de 30 Hz.
- sensor infravermelho que constituído por um projetor laser combinado com um sensor monocromático, que captura informação 3D.
- Possui um alcance entre os 0.8 e os 4 metros.

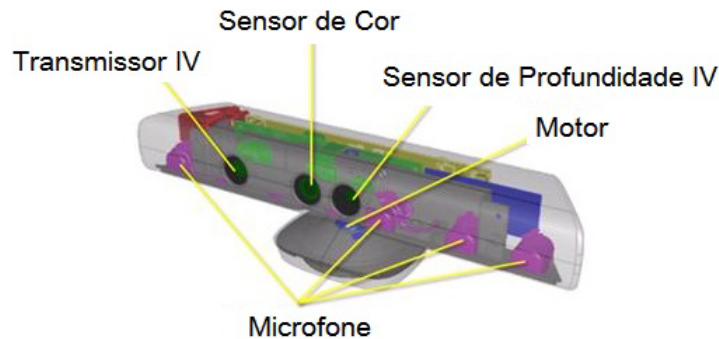


Figura 2.18: Constituição da Kinect [28].

- Possui um campo de visão de 57 graus na horizontal e 43 graus na vertical.

A possibilidade de ligação ao computador impulsionou o desenvolvimento de programas e aplicações em que por exemplo, fazem a diferenciação de objetos e pessoas presentes num determinado espaço. O reconhecimento de seres humanos e a interpretação da sua linguagem corporal constitui atualmente inúmeras aplicações que promovem a interação homem-máquina.

Relativamente aos seus sensores, o sensor de profundidade IV permite a captura de imagens tridimensionais em que cada pixel possui coordenadas X, Y e Z, correspondentes ao referencial da câmara. A coordenada Z representa a distância da câmara ao objeto detetado.

O sensor de cor RGB deteta a cor da luz que atinge a câmara registando para cada píxel da imagem valores de vermelho, verde e azul. O sensor de profundidade da Kinect possui algumas limitações funcionando com normalidade para valores de profundidade entre os 0.8 e os 4 metros.

A utilização de infravermelhos por este sensor faz com que não seja possível a determinação de superfícies.

## 2.9 Sistemas de Segurança

A segurança é o fator comum e de extrema importância em todas as instalações industriais. A falta dela trás muitas elevadas e riscos para operadores e máquinas. Numa célula

robótica convencional, onde se encontram equipamentos em movimento, tem de existir um perímetro de segurança que, proteja e separe os operadores da célula. Esse perímetro é usualmente constituído por barreiras físicas do género “jaula” (Figura 2.19) com uma entrada, estando esta controlada por sensores. Atualmente a utilização de vigilância por laser ou sistemas de visão artificial também tem vindo a ganhar importância.



Figura 2.19: Célula Robótica Tradicional.

Com a utilização de robôs colaborativos na automação industrial faz com que a interação humana seja feita de forma segura e que trabalhem lado a lado com os operários da indústria, através de configurações de segurança avançadas que respeitem também as normas da International Organization for Standardization (ISO) 10218 [29].

Essas configurações de segurança em robôs colaborativos podem ser feitas através de sensores de movimento a laser: quando a máquina deteta a presença humana, ela pára de funcionar ou adapta a sua performance, permitindo que o operador faça o que precisa fazer ao seu lado em total segurança. Um sistema colaborativo normalmente tem também criadas zonas de visão, ou seja, zonas em que é detetada a presença do operador, o sistema

permite que a máquina continue a funcionar. A máquina só irá parar quando o operador se aproximar demais, ou seja entre na zona perigosa e quando ele sair dessa área, retoma o seu normal funcionamento.

Nessas configurações de segurança, os robôs chegam a tocar o operário, mas param de funcionar quando detetam qualquer tipo de toque. Além disso, eles não podem possuir pontas afiadas, motores expostos e são limitados na quantidade de peso que podem carregar, para reduzir o risco de acidentes graves.

O acesso a botões de emergência deve estar como no caso dos robôs convencionais, disponível na consola de programação e se possuir outros meios de ação deveram haver outros botões de emergência disponíveis. Por mais que sejam seguros, os equipamentos industriais ainda são máquinas, e precisam ser tratados com responsabilidade, como tudo o que envolve a automação industrial.

## 2.10 Ergonomia

A ergonomia segundo a norma 9245 da ISO é descrita como a qualidade da adaptação de um posto de trabalho a um operador e há tarefa que este realiza. Esta característica deve ser tida em conta, visto que por mais fácil que seja uma operação e aparentemente inofensiva para a conservação da saúde do operador, esta pode cair por terra visto que o número de repetições pode mudar qualquer previsão. Pode-se dizer que quanto mais adaptado for o processo, maiores serão os níveis de eficácia, eficiência e satisfação alcançados pelo operador durante o uso do sistema, sendo eles [30]:

- Eficácia: a capacidade que os sistemas conferem a diferentes tipos de utilizadores para alcançar os objetivos em número e com a qualidade pretendida.
- Eficiência: a quantidade de recursos (por exemplo, tempo, esforço físico e cognitivo) que os sistemas solicitam aos operadores para realizar a tarefa.
- Satisfação: a emoção que os sistemas proporcionam aos usuários, face aos resultados obtidos e dos recursos necessários para alcançar os objetivos.

Por outro lado, um problema de ergonomia é identificado quando um aspeto da interface está em desacordo com as características dos utilizadores da mesma e da maneira pela qual ele realiza uma determinada tarefa.

A ergonomia está ligada à maneira como os profissionais desempenham as suas atividades quotidianas, tais como, a posição em cadeiras, mesas, bancadas, sequência de movimentos ou tarefas, ruídos, iluminação, entre outros. Estes são pontos fulcrais uma vez que podem evitar problemas de saúde aos operadores como, dores nas costas por estar demasiado baixa ou demasiado alta, ou até pela simples colocação de um ecrã que em posição desconfortável pode causar sequelas na coluna a longo prazo [31].

Portanto, a ergonomia torna-se uma questão bastante abrangente e tem como objetivo fazer com que os colaboradores desempenhem as suas atividades de maneira mais confortável e sem que ofereça risco para a sua saúde visto que vários estudos apontarem que os maiores problemas ergonómicos são causados por mobiliário inadequado problemas de postura, principalmente no que se refere ao uso do teclado e rato.



# Capítulo 3

## Requisitos e Conceção do Sistema Colaborativo

No projeto de conceção de uma célula robótica são tidos em conta diversos aspetos estruturais, ergonómicos, funcionais e sobretudo ressaltar o aspeto de segurança quer de humanos quer de equipamentos. Neste sentido este capítulo delimita os requisitos a seguir quer no projeto da solução final quer na relação dos diversos componentes intervenientes e de-linearização dos cenários demonstrativos.

### 3.1 Requisitos Funcionais do Sistema

A conceção desta célula colaborativa destina-se a ser uma bancada demonstrativa de tecnologias e conceitos pela região TMAD, motivo pelo qual, no dimensionamento físico da célula há a necessidade de mobilidade e portabilidade, a estrutura precisa ser o mais funcional possível sendo que do ponto de vista do operador deve estar enquadrada com a estrutura média de um humano.

A célula deve conter o maior número de partes visíveis, pelo que a sua construção deverá comportar o menor número possível de materiais opacos, para facilitar a observação dos integrantes a funcionar. Em relação à parte de controlo da célula onde se incluem controladores e quadro elétrico, deve estar vedado com material opaco para que o foco

seja a parte demonstrativa da célula. Esta deve ainda ter um aspeto Industrial e revelar uma aplicabilidade abrangente no que toca aos tipos de indústria.

O aspeto de segurança para o operador deverá ser um ponto fundamental pelo que como pré-requisito devem ser criadas pelo menos dois tipos de funcionamento do robô:

- Com presença de operador: O robô deve perceber a presença de operador por meio de equipamentos de segurança externos e com essa informação deverá adaptar a sua performance (reduzir: aceleração, força e velocidade).
- Sem presença de operador: O robô deve executar as suas tarefas com a performance de trabalho normal.

Um outro requisito ligado à segurança sendo esta mais ligada a parte estrutural a célula deve compreender duas zonas de trabalho distintas.

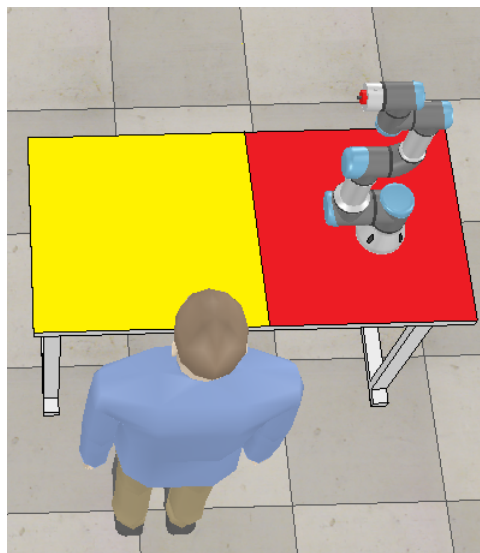


Figura 3.1: Pré Requisito para Definição de Zonas de Trabalho.

Na Figura 3.1, a zona colorida a vermelho representa a área de trabalho exclusiva do robô onde este se encontrará montado, o operador nesta zona deverá ter acesso o mais restrito possível desde que não prejudique a movimentação do robô. Já a parte colorida a amarelo na Figura 3.1 deverá ser uma zona de trabalho comum onde poderão

trabalhar operador e robô independentemente da tarefa que for delineada nos cenários demonstrativos.

As interrupções devem ser feitas por meio de uma HMI, o acesso físico a esta deve ser o menos exigente de movimento possível, intuitivo e acima de tudo que leve adiante a ideia de célula colaborativa e seja o mais possível cooperante com o operador.

O objetivo inicial é delinear uma atividade dita "contínua" em que robô se encontre a desenvolver uma tarefa completamente autônoma sem intervenção humana como por exemplo organizar peças ou movê-las para posições adequadas usando para isso uma câmara ou um conjunto câmara-garra ou outra configuração que seja aplicável.

O passo seguinte deve passar por elaborar dois diferentes cenários que introduzam interrupções ao funcionamento contínuo do robô. Estas interrupções devem ser o mais complexas quanto os integrantes o permitirem e devem demonstrar as potencialidades do funcionamento do conjunto de equipamentos utilizados. Estes cenários devem ainda evidenciar a possibilidade de trabalho colaborativo entre humano e robô, tanto no aspeto de segurança como no aspeto colaborativo, ou seja, o robô deve ser adaptativo o suficiente para reconhecer o trabalho executado pelo operador e auxiliá-lo na resolução das suas tarefas.

## **3.2 Arquitetura do Sistema**

Para esta fase foram realizados vários protótipos e as linhas de construção do projeto foram delimitadas através de debate com os restantes membros de projeto, outras instituições de ensino reconhecidas por este tipo de trabalho e empresas fornecedoras de equipamentos. A escolha dos equipamentos ficou condicionada ao orçamento pelo que este deveria ser o mais contido possível e que abordasse o maior número de tecnologias.

A escolha da estrutura da bancada requer uma solução específica e particular e por esse motivo deverá ser projetada e dimensionada para o efeito visto não existirem no mercado soluções chave na mão com dimensões e aplicações tão flexíveis como o pretendido.

No que toca aos equipamentos a aplicar na bancada, a escolha recai então sobre:

- Robô: A escolha do robô recaiu sobre o braço robótico da UR denominado de UR3. A escolha deste robô colaborativo prendeu-se pelo facto do mesmo ser versátil independentemente do campo de aplicação, ser de peso e dimensões apropriadas para a aplicação numa bancada móvel e ainda por haver variadíssimas ferramentas pré-preparadas para aplicar na sua junta terminal.
- Câmara e garra: Para a escolha destes equipamentos foram consultados diversos fornecedores e integradores pelo que a solução mais robusta e de fácil integração recaiu sobre a garra *Gripper Q85 2 Finger* e pela câmara *Wrist Camera* ambas da marca *ROBOTIQ*.
- Computador: Para tornar o sistema mais robusto e com "aparência" industrial a escolha recaiu então sobre um computador industrial *NEO-MINI-30R*, em que a sua principal característica não se prende um grande poder de armazenamento ou processamento mas sim trazer fiabilidade e confiança ao sistema.
- HMI: Na integração de uma interface homem-robô optou-se por integrar um tablet (samsung galaxy Tab), não só para demonstrar que todos os equipamentos ditos convencionais podem ser aplicados em meios industriais mas também para poder aquando das demonstrações ter uma atitude mais interativa com o público.
- Sistema de segurança: Para a segurança da célula será aplicado um botão de emergência convencional e, como equipamento de verificação de presença de operador, serão aplicadas barreiras fotoelétricas da marca *OMRON* ficando a medida das mesmas pendente até ao dimensionamento estrutural da célula. A utilização de barreiras fotoelétricas pressupõe a utilização ainda de relé de segurança para o seu controlo.
- Material Elétrico: O material elétrico não terá escolha específica pelo que serão utilizados equipamentos convencionais respeitando as normas e regras técnicas deste tipo de equipamentos.
- Iluminação: Para a iluminação será construída uma iluminação adaptada à célula.

Terão que ser adaptados equipamentos existentes no mercado para a célula, posteriormente ao dimensionamento da mesma.

- Periféricos de funcionamento: Todos os periféricos que se tornem cruciais para o funcionamento da célula ou dos cenários, serão elaborados em plataforma de desenho CAD para posteriormente serem materializados em impressora 3D.

Todos estes componentes anteriormente referidos juntos conseguem abraçar um número considerável de tecnologias faltando apenas delinear a relação entre os referidos componentes.

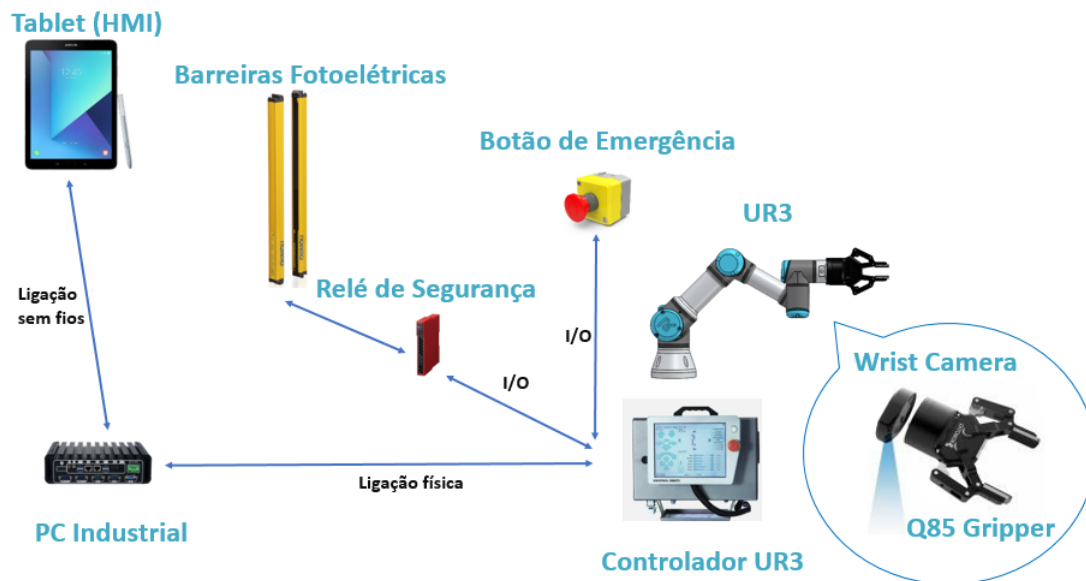


Figura 3.2: Arquitetura do Sistema.

Na Figura 3.2 é possível identificar os principais intervenientes do projeto que estarão montados na bancada de trabalho e como tal importa salientar o tipo de ligações que serão realizadas.

O *tablet* terá a função de HMI, mas servirá também para demonstrar junto do público alvo a aplicabilidade de sistemas convencionais ao ramo industrial (pode passar de mão em mão aquando das demonstrações), e ampliando assim as valências do sistema aplicado. Por outras palavras será o ecrã móvel do computador industrial visto a aplicação principal e a comunicação com o robô estar implementada no dito computador industrial.

Ao controlador do robô estarão ligados os periféricos robóticos (garra e câmara) e o respetivo robô, sendo as ligações já pré definidas pelos fabricantes dos referidos equipamentos. Paralelamente no Controlador do robô estarão também ligados os equipamentos de segurança composto por:

- Botão de emergência externo que liga diretamente num *input* destinado para o efeito no controlador do robô.
- Barreiras fotoelétricas, equipamentos que requerem controlo de um sub-equipamento denominado de relé de segurança que fará o controlo dos sinais que vão para o controlador do robô.

A estas configurações serão acrescentadas outras que possam advir da implementação dos cenários demonstrativos posteriormente delineados.

### **3.3    Enquadramento da Prova de Conceitos**

A decisão sobre a escolha da prova conceitos adequada partiu de vários testes e protótipos para a célula, as ideias pré projeto foram sofrendo alterações muito por causa dos equipamentos que se deveriam aplicar mas também pela necessidade de adaptação de periféricos que já tinham sido adquiridos para o qual era necessário projetar e estudar o enquadramento e a ligação de todos os componentes.

A prova conceito precisa de ser o mais abrangente possível em termos de aplicabilidade a diversos tipos de indústria para não se tornar restritiva no momento da apresentação do projeto, pelo que aplicações muito específicas não serão consideradas. Serão então delimitados 3 cenários diferentes: I Cenário Demonstrativo de Tarefa contínua (desenvolvimento da torre de *Hanoi*); II Cenário Demonstrativo 1 (auxílio ao operador na arrumação de peças); Cenário Demonstrativo 2 ( auxílio do operador na montagem de uma *giftbox* ) ;

### 3.3.1 Cenário Demonstrativo de Tarefa Contínua

Na primeira atividade, o robô estará a desenvolver de forma automática e contínua a torre de *Hanoi* este "quebra-cabeça" que consiste mover discos numa base que contém três pinos, em que num dos quais estão alguns discos uns sobre os outros, em ordem crescente de diâmetro, de cima para baixo (Figura 3.3) [32].



Figura 3.3: Torre de *Hanoi* Posição Inicial e Posição Final [32].

A tarefa desta atividade consiste em passar todos os discos de um pino para outro qualquer, usando um dos pinos como auxiliar, este jogo tem apenas duas regras muito simples sendo elas:

- Apenas se pode movimentar um disco de cada vez.
- Nunca se pode colocar uma peça de tamanho superior em cima de uma com tamanho menor.

O número mínimo de "movimentos" para conseguir transferir todos os discos do primeiro pino para o segundo é:

$$2^n - 1$$

, sendo  $n$  o número de discos. Para o caso de 3 discos, o mínimo de movimentos seria 7, facilmente explicado na Figura 3.4. O movimento começa em qualquer um dos ângulos do triângulo, depois devem apenas ser feitos os movimentos expressos no perímetro da Figura 3.4. No caso de não ser seguida a sequência desse perímetro é possível chegar ao resultado desejável, mas não de forma otimizada, uma vez que o número de movimentos pode ser em muito excedido em relação ao número calculado pela fórmula acima transcrita [32].

No caso deste cenário, a torre de *Hanoi* realizar-se-á com cinco discos obtendo-se então um número mínimo de movimentos de 31. A dificuldade de resolução deste quebra cabeças prende-se pela necessidade de rápido raciocínio para saber qual a peça e para onde a mover demarcando-se assim a utilização de robótica para resolver esta atividade.

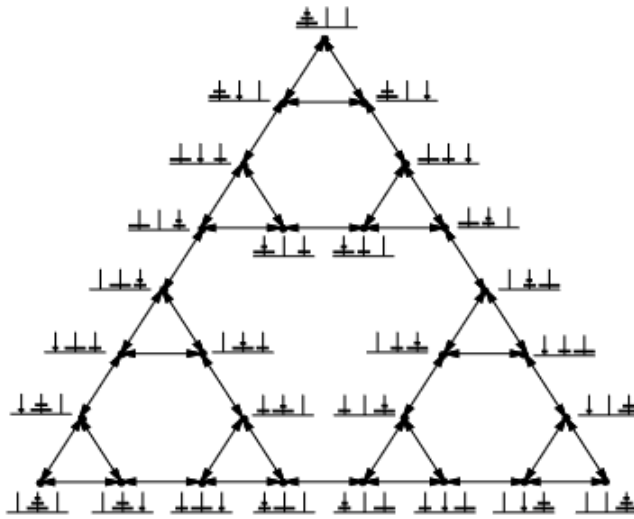


Figura 3.4: Movimentos Possíveis com 3 Discos [32].

### 3.3.2 Cenário Demonstrativo de Arrumação de Peças

No caso deste cenário, para além da tarefa realizada no cenário de tarefa contínua, o robô deverá ser capaz de perceber por meio de processamento de imagem ou por sensores, a presença de operador a executar tarefas de montagem na zona de trabalho seguro ou zona de trabalho comum da célula (Figura 3.5) e com isso adaptar a sua velocidade, força e aceleração. Aquando da retirada do operador, o robô volta ao seu estado normal de trabalho, sem que seja necessário qualquer tipo de rearme.

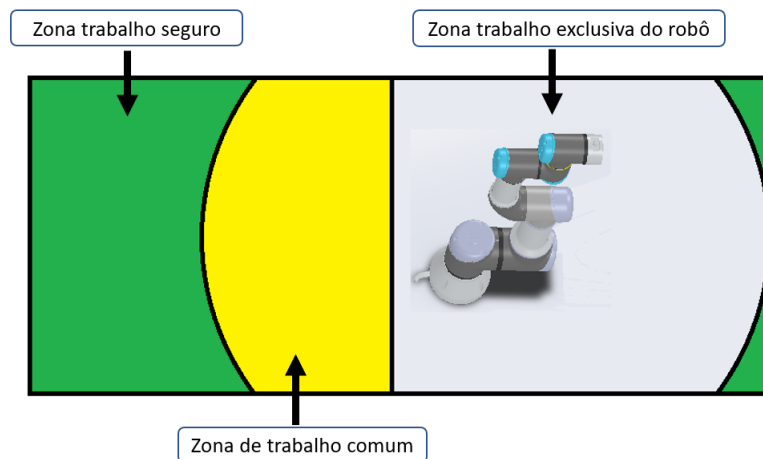


Figura 3.5: Identificação das Zonas de Trabalho da Célula.

Para além destas questões de segurança, o operador terá a tarefa de colocar na área de trabalho, previamente definida pelo campo de visão do robô, n peças cilíndricas ou paralelepípedas com orientação aleatória sendo que no final da colocação deverá premir um botão de chamada do robô. O robô irá para uma posição de reconhecimento visual e posteriormente calcula a posição e orientação de cada peça colocando-as numa rampa de stock. Assim que deixe de detetar qualquer peça, na sua área de visão, o robô volta à sua atividade principal (torre de *Hanoi*) sem que para isso seja necessário o operador realizar qualquer ação.

### 3.3.3 Cenário Demonstrativo de Colocação de Peças

Neste cenário, serão implementados também conceitos colaborativos, mais avançados introduzindo-se novas funcionalidades. Por outras palavras, esta nova tarefa segue o anterior cenário mas de forma inversa, ou seja em vez de o robô ir à posição de reconhecimento localizar a posição da peça e a arrumar, vai localizar o local onde irá colocar a peça. Para tal será criada uma área de trabalho constituída por quadrículas brancas e pretas, dispostas alternadamente em forma de xadrez. O operador terá ao seu dispor, peças numa rampa colocada na zona de trabalho comum e, deve colocar 8 peças nas quadriculas brancas (Figura 3.6).

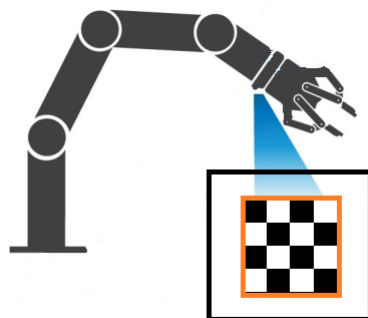


Figura 3.6: Esquema de Funcionamento do Reconhecimento da Zona de Montagem.

Se por qualquer motivo o operador não colocar todas as peças, o trabalho deve ser concluído pelo robô. Para tal, o robô depois de receber ordem do operador, move-se para

a posição de reconhecimento e, é capaz de identificar as quadrículas onde não existem peça, completando assim o trabalho não finalizado pelo operador.

É através de uma aplicação web a operar num *tablet*, sendo esta a única maneira de o operador comunicar com os componentes da célula. Será possível nela escolher qual o cenário a realizar, assim como monitorizar stocks e ainda possíveis falhas de funcionamento. Enquanto a escolha não é feita e posteriormente à realização da tarefa selecionada o robô encontra-se de forma contínua e automática a resolver a torre de *Hanoi*, mantendo as mesmas configurações de segurança do cenário anterior.

### **3.4    Conceção Estrutural**

Considerando os cenários descritos anteriormente e o objetivo demonstrativo da célula, esta deverá ser móvel, ter resistência estrutural para suportar o robô e as suas acelerações, assim como, todos os adereços necessários à implementação dos cenários acima descritos. A sua conceção deverá também levar em conta o objetivo principal que é a demonstração de conceitos.

A utilização de um robô pressupõe também a utilização de um controlador e de uma consola de programação pelo que, ter um suporte devidamente seguro e ergonómico torna-se também indispensável, pelo que no projeto foi tido em consideração. Sendo a finalidade da célula implementação de conceitos e temas atuais, o que hoje é atual daqui a um curto espaço de tempo pode já estar completamente ultrapassado em termos tecnológicos e por esse motivo poderá ser necessário aplicar um diferente robô ou periférico. Para contornar esse problema é necessário criar a flexibilidade de fixação dos diversos componentes em toda a área da célula, com especial atenção para o tampo, onde irão assentar o robô e periféricos, para tal está implementado um tampo em perfil bosch .

À célula foi acrescentada uma longarina superior ajustável à altura, onde poderá ser montado o robô, estando esta perfeitamente dimensionada, acrescentando assim uma valência que futuramente pode ser crucial. Posteriormente à parte estrutural, a célula encontra-se totalmente vedada com acrílico transparente que, de acordo com a aplicação

pode ser retirada, mudando a qualquer momento o *layout* de trabalho. Na parte traseira da célula encontram-se ainda duas janelas também com acrílico que permitem por exemplo repor stocks. Esta vedação prende-se apenas pelo facto de esta célula ser demonstrativa para uma alargada faixa etária, podendo tanto ser para demonstrar conceitos a crianças, empresários ou até idosos.



Figura 3.7: Projeto CAD da Célula.

Na Figura 3.7 é possível ter uma visão melhor das dimensões finais sendo estas 1,7 metros de altura 0.75 metros de largura e 1,2 metros de comprimento, estando estas medidas definidas inferiormente em relação ao tamanho médio de uma porta convencional. Uma vez mais a existência de vedação de acrílico pode ser pensado como segurança a mais, uma vez que o robô colaborativo por si só já o é seguro, mas como referido anteriormente a inovação tecnológica pode levar a que num curto espaço de tempo haja necessidade de colocar outro tipo de tecnologia sendo que nesse caso a célula já se encontra preparada.

## 3.5 Sistemas de Apoio ao Funcionamento

Para a realização de todos os cenários será necessário recorrer a *software* CAD para desenhar e simular dimensões dos diversos sistemas de apoio ao funcionamento de todos os cenários e sistemas de apoio de funcionamento geral.

### 3.5.1 Sistemas de Apoio à Tarefa Contínua

Para a realização da torre de *Hanoi* é necessário por exemplo desenvolver os pinos da torre de que, inicialmente foram testados com madeira e ferro, validando assim a construção do modelo físico.

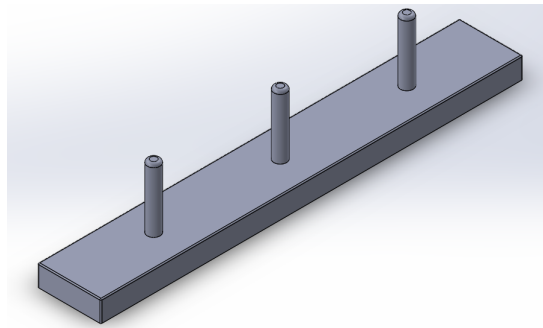


Figura 3.8: Projeto CAD Torre de *Hanoi*.

A referida base para a resolução do "quebra cabeças" será construída em bloco de alumínio. Com as medidas dos pinos de 7 centímetros de altura por 12 milímetros de espessura conforme a Figura 3.8.

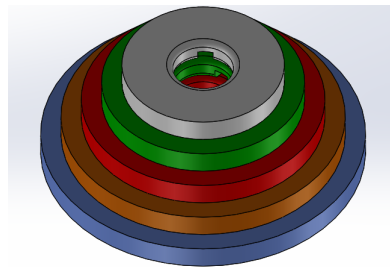


Figura 3.9: Projeto CAD dos Discos da Torre de *Hanoi*.

Continuando na torre de *Hanoi* depois de devidamente dimensionada com diversos testes práticos também serão desenhadas neste caso para serem impressas em plástico

com auxílio de Impressora 3D, cinco discos de diâmetros diferentes (Figura 3.9) variando de disco para disco 10 milímetros de diâmetro .

### 3.5.2 Sistemas de Apoio ao Cenário de Colocação de Peças

Para os sistemas de apoio ao cenário um foi necessário desenvolver vários testes para analisar quais as peças que melhor se adaptam ao desenvolvimento da prova conceito pretendida. O facto de integrar as peças de sistemas já existentes na ESTiG será também um aspeto a ter em consideração, uma vez que abre possibilidades no sentido de futuramente, se necessário, permitir uma integração entre equipamentos e projetos.

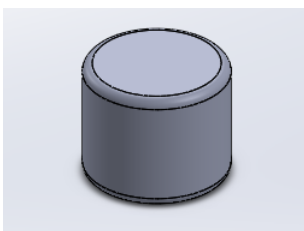


Figura 3.10: Projeto CAD Peça Cilíndrica.

O dimensionamento CAD das peças a serem movidas (Figura 3.10), mantêm a forma e tamanho já existente em outras aplicações robóticas do Laboratório de Controlo Automação e Robótica (LCAR) onde também ficará esta célula depois dos eventos demonstrativos.

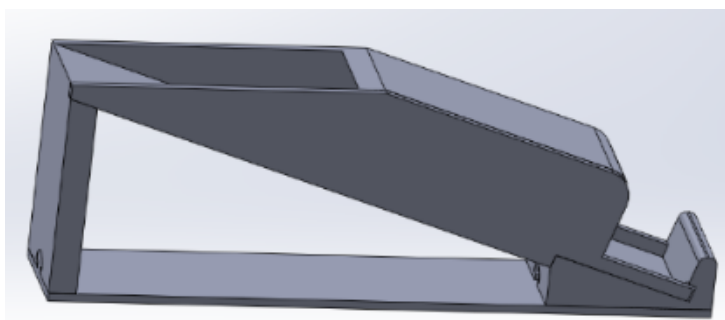


Figura 3.11: Projeto CAD da Rampa de Peças Cilíndricas.

Na rampa de *stock* (Figura 3.11) dimensionada para que suporte o máximo de 8 peças cilíndricas tendo em conta o facto que não precisa de qualquer eletromecatrônica para a sua implementação, ambas as peças são impressas em impressora 3D de material plástico.

V		V	
	V		V
V		V	
	V		V

Figura 3.12: Tabuleiro para Colocação de Peças.

A bandeja de trabalho será realizada em acrílico com sobreposição de vinil para reduzir o brilho que era introduzido pelo flash da câmara ao adquirir a imagem para processar. O aspeto final do plano superior com a forma de uma grelha com símbolos invariáveis à rotação da câmara para que a colocação da peça seja feita sempre com a mesma orientação (Figura 3.12).

### 3.5.3 Sistemas de Apoio ao Cenário de Arrumação de Peças

Nesta tarefa vão ser utilizados dois tipos de peças; utiliza as peças do cenário anterior e ainda uma outra para aumentar as valências da aplicação.

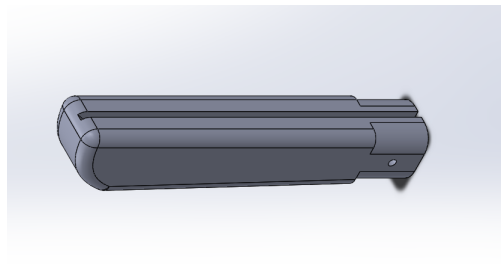


Figura 3.13: Projeto CAD Cabos de Navalha.

No caso do segundo cenário é necessário utilizar peças que dependendo da sua orientação a garra realizará uma adaptação da sua orientação para demonstrar as potencialidades do sistema sendo para isso desenhadas peças com aspeto de cabo de navalha (Figura 3.13) também no sentido de enquadrar este exemplo numa indústria da cutelaria presente no território alvo deste projeto.

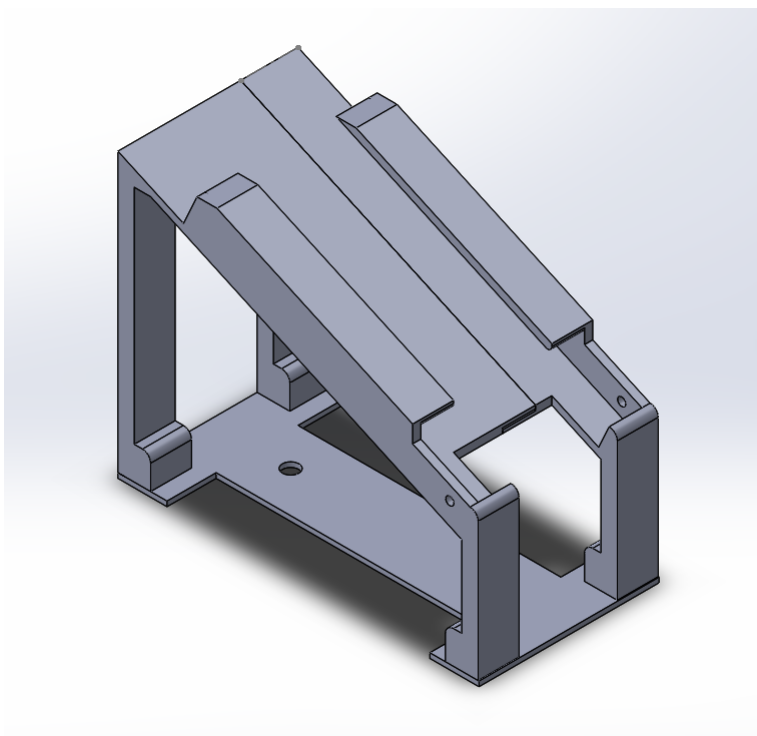


Figura 3.14: Projeto CAD da Rampa de Cabos de Navalha.

Consequentemente terá de ser construída uma rampa para acomodar os referidos cabos (Figura 3.14), rampa esta que não necessita de qualquer eletromecatrônica para funcionamento uma vez que as peças posicionam-se sempre na mesma posição correta por ação da gravidade.

Desta forma os componentes compreendem todas as necessidades pré-estabelecidas, pelo que se conseguem enquadrar as várias tecnologias implementadas neste sistema e também enquadrá-las no ambiente já existente no LCAR, elevando assim as possibilidades de futura integração.

## 3.6 Simulação para Validação Funcional

Com o propósito de elaborar um ambiente virtual que reproduza os processos e tecnologias a demonstrar, foram testadas algumas ferramentas de *software* necessárias para levar a cabo cada um dos cenários propostos. Cada uma das ferramentas selecionadas irão ser aplicada para uma determinada tarefa, entre elas representar modelos em 3D, criar ambientes de simulação, estabelecer parâmetros de comunicação, entre outros.

O funcionamento das ações desenvolvidas no programa de simulação foi comandado por uma interface implementada em Matlab. Esta interface não só permite a interação entre o simulador e o utilizador, mas também permite o acesso à base de dados construída através de uma aplicação web. A simulação foi desenvolvido no *software* V-REP (Figura 3.15).

Na realização de um ambiente virtual que abarque as funcionalidades necessárias, não só é necessária a perceção das ações que se pretendem simular, mas também é necessário ter em consideração todos os materiais e equipamentos que intervêm na sua concretização.

Entre os materiais necessários reproduzir na plataforma do SolidWorks, encontra-se compreendidos a célula propriamente dita, o cabo e o seu dispensador, assim como pequenos objetos para darem suporte à montagem.

Esta simulação foi o único teste feito para aferir ideias do pré projeto em termos ergonómicos e de dimensões assim como acrescentar ou retirar valências. Nesta fase de projeto a ideia era desenvolver uma célula para demonstração de conceitos numa unidade fabril de cutelaria e assim sendo os seguintes cenários envolviam a produção e customização de produtos por parte de um cliente.

- Cenário 1 - Neste cenário, é possível verificar que o robô adapta a sua velocidade de operação de acordo com a presença ou não do operador na zona partilhada. O operador procede há escolha de uma referência e consoante a escolha o robô vai recolher o cabo e a lâmina e entrega-los na bandeja de trabalho para o operador proceder a sua tarefa de montagem. Finalizada a tarefa de montagem o operador faz chamada ao robô para arrumar a peça finalizada no local apropriado.

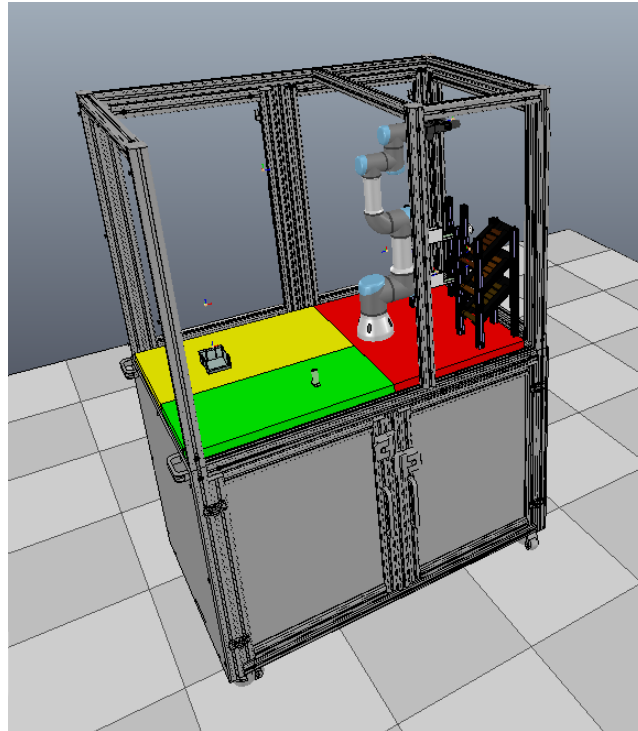


Figura 3.15: Imagem da Simulação Realizada em V-REP.

- Cenário 2 - Sobrepor ao cenário um a utilização do robô para efetuar inspeção à peça final. Caso o produto não possua as especificações corretas será necessário *re-work*, a ser efetuado pelo operador estando esta decisão a cargo do operador. Metodologia usada, quando necessário para a compreensão do relatório;
- Cenário 3 - Numa tentativa de implementar e demonstrar a Internet das coisas e a personalização do produto final por parte do cliente ou seja em qualquer parte do mundo um cliente submete uma encomenda de um determinado produto, esse produto cai na base de dados de pedidos.

O operador quando chega à bancada de trabalho apenas dá sinal ao robô que se encontra pronto para montar a peça final pelo que não sabe a referência que irá fazer, o robô após a chamada do operador consulta a base de dados de encomendas e a lista de stock que tem presente na bancada vai decidir qual a referência a montar e por sua vez quais as peças que vai fornecer ao operador para montar. No final é fechada a encomenda.



# Capítulo 4

## Desenvolvimento e Instalação da Solução Desenvolvida

Neste capítulo é descrito o trabalho de implementação, salientando os pontos mais relevantes da mesma, dificuldades particulares encontradas ou soluções técnicas inovadoras desenvolvidas e aplicadas. Ao longo deste projeto não é abordada a montagem da estrutura da célula uma vez que essa tarefa foi realizada pelo fornecedor da estrutura, tendo sido apenas necessário realizar pequenos ajustes ou posicionamento de componentes.

### 4.1 Instalação de Equipamentos

Nesta secção são descritos os componentes que compõem o sistema e que em conjunto elevam a complexidade do sistema:

- Montagem de dispositivos de robótica.
- Montagem do circuito de potência.
- Montagem do sistema de iluminação.
- Montagem de dispositivos de emergência.

Para dar apoio à montagem dos sub-componentes acima referidos foi implementado um *upgrade* para passagem de cabos quer na parte superior da célula quer no tampo e na parte inferior da mesma (rodeado a vermelho na Figura 4.1).

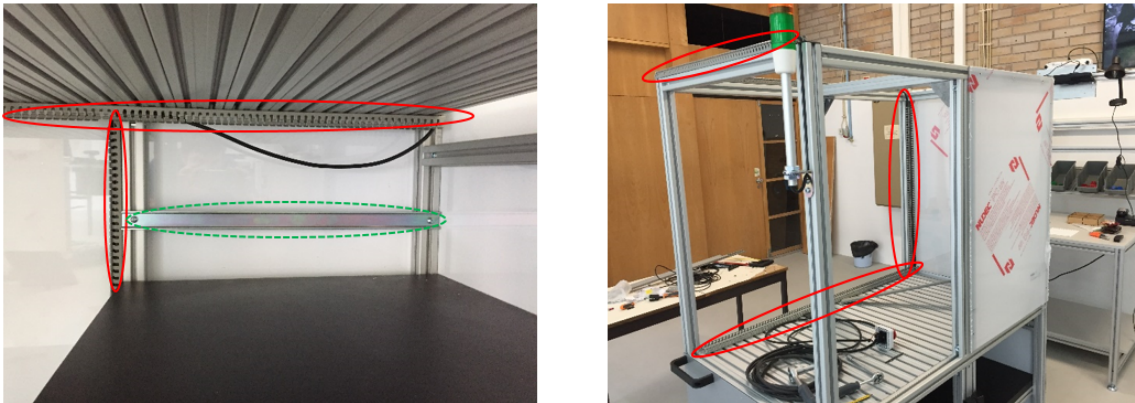


Figura 4.1: Calha para Passagem de Cabos e Calha de Fixação de Componentes.

Já para apoio à fixação de componentes de controlo foi instalada uma calha metálica (rodeado a verde tracejado na Figura 4.1). Estes anexos à construção inicial vêm não só facilitar a montagem dos equipamentos mas também manter a estética original e manter o acesso a qualquer momento a cabos para corrigir e detetar anomalias, assim como introduzir novos equipamentos sem necessitar de intervenção profunda (ex.fazer furos).

#### 4.1.1 Montagem dos Dispositivos de Robótica

A integração dos equipamentos robóticos prende-se pela fixação do controlador na parte inferior da célula e a colocação do robô no tampo da bancada anteriormente projetada, esta tarefa foi em muito simplificada, uma vez que os perfis implementados foram de acordo com a furação do robô podendo este ser montado em qualquer ponto do tampo. No caso da montagem dos periféricos, ou seja, da câmara (*Wrist Camera*) e da garra (*Q85 2-Finger*) está muito facilitada uma vez que foram produzidos para trabalharem em conjunto e de forma integrada com o robô UR3 (Figura 4.2) pelo que, apenas foi necessário colocar na posição indicada os parafusos e fazer a passagem dos cabos inerentes ao funcionamento.

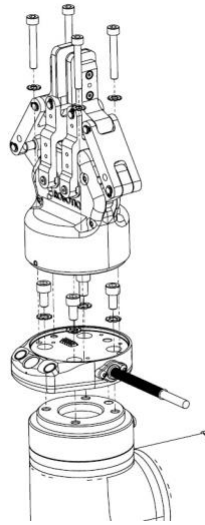


Figura 4.2: Montagem de Câmara e Garra no Robô [33].

Ressalva-se que pelo facto da câmara estar preparada para ser integrada com a garra, esta vem equipada com um sistema de contactos para comunicar com a referida garra. Com esta característica é possível fazer o comando das duas ferramentas com a necessidade apenas de passar um cabo até a junta terminal do robô.

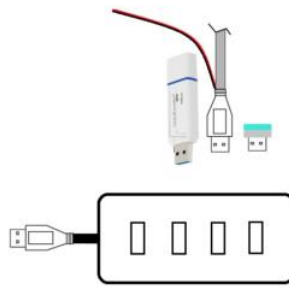


Figura 4.3: Ligações de Câmara e Garra no Controlador do Robô [33].

Nas ligações elétricas a componente robótica do sistema necessita de uma ligação a 24V e uma ligação USB no robô uma vez que a memória do *software*, *dongle* da licença de produto e o USB de comunicação com as ferramentas estão ambos conectados num *hub* USB (Figura 4.3). A utilização destes equipamentos pressupõe ainda a instalação de *software* de utilização na consola do robô sendo esta posteriormente retratada.

### 4.1.2 Montagem do Circuito de Potência

Na distribuição e controlo elétrico da célula foram tidos em conta diversos aspetos como segurança para o utilizador, enquadramento na estrutura assim como a proteção de equipamentos. A escolha dos equipamentos recaiu sobre a aplicação de tomadas em calha para possibilitar alterações a qualquer momento, sem que isso implique fugir à estética inicial nem implicar custos avultados.



Figura 4.4: Montagem Final do Quadro Elétrico.

A célula encontra-se equipada com ligação à terra em todos os equipamentos assim como proteção diferencial de 300 mA (Figura 4.4). Foram ainda criados três circuitos elétricos comandados por um disjuntor unipolar. O primeiro circuito que comanda a ligação do robô é o transformador 230V AC/24V DC que alimenta o sistema de segurança composto pelas barreiras e pelo relé de segurança.

O segundo circuito faz o controlo exclusivamente do comando da iluminação da célula. No terceiro circuito estão penduradas as tomadas de reserva para carregamento do *tablet* do operador assim como a ligação para um computador ou outros periféricos futuramente instalados. Em anexo encontram-se os esquemas elétricos referentes a esta montagem (Anexo A).

### 4.1.3 Montagem do Sistema de Iluminação

O controlo do circuito de iluminação é feito por meio de um disjuntor no quadro elétrico devidamente identificado que comanda um transformador 230V AC /24V DC. A iluminação da célula está dividida em dois circuitos (Figura 4.5): I Circuito da zona de trabalho comum entre robô e operador; II Circuito para a zona de trabalho exclusiva do robô.

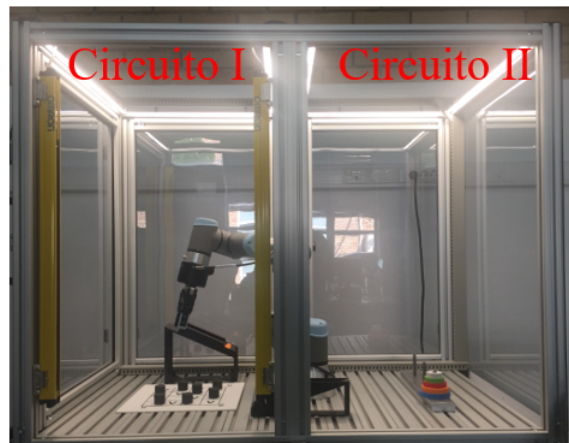


Figura 4.5: Zonas de Iluminação Criadas na Bancada.

A necessidade de criar duas zonas de iluminação distintas prende-se pelo fato de o trabalho realizado pelo robô ser diferente nas duas zonas e poder em aplicações futuras ter diferentes ambientes luminosos.



Figura 4.6: Calha de Encastre para Fita LED.

A decisão de integração de dispositivos de regulação de intensidade luminosa recai sobre o facto da câmara não ter iluminação suficiente em alguns casos e noutros ter iluminação em demasia, obrigando então a integração nos circuitos de regulação de intensidade, os *dimmer*.

A solução de instalação física recaiu então na implementação de perfil de alumínio em U com tampa translúcida (Figura 4.7), onde está embutida a fita LED, fazendo dois retângulos de iluminação, um para cada zona. Solução esta perfeitamente enquadrada com a estrutura respeitando as cores e a robustez do sistema.

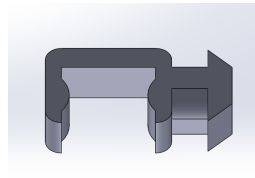


Figura 4.7: Apoio para Calha de Iluminação.

Foram ainda desenhados e impressos em impressora 3D os apoios para o referido perfil de U (Figura 4.6) encaixar no perfil bosch da estrutura na célula.



Figura 4.8: Resultado Final da Implementação da Calha de Iluminação.

Na colocação da iluminação, em fita LED foi tido em conta o aspeto visual da célula sendo necessário implementar uma solução eficaz, robusta e sobretudo funcional, ficando assim a iluminação perfeitamente enquadrada na estrutura da célula (Figura 4.8). Em anexo encontram-se os esquemas elétricos referentes a esta montagem (Anexo A).

#### 4.1.4 Montagem dos Dispositivos de Emergência

Sendo o objetivo da bancada a introdução de um robô colaborativo o aspeto de segurança para com o operador está já subentendido. Ainda assim, para reforçar este aspeto e enaltecer as potencialidades de implementação foram introduzidas barreiras fotoelétricas, um botão de emergência e um indicador luminoso (Figura 4.9).

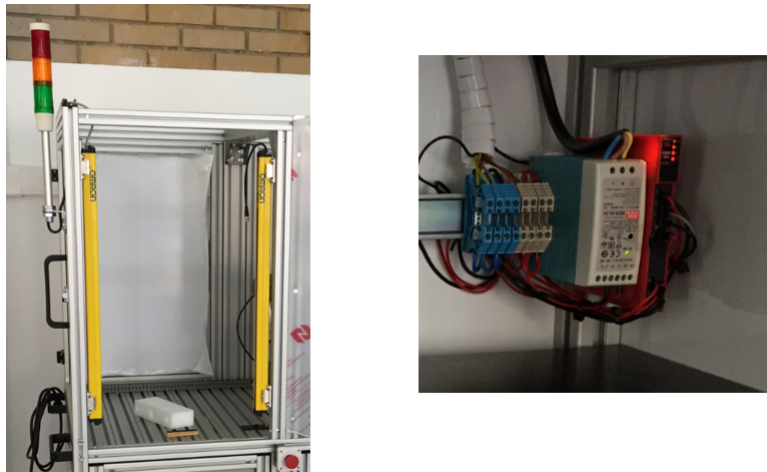


Figura 4.9: Instalação dos Equipamentos de Segurança.

No caso das barreiras fotoelétricas foram utilizadas barreiras da OMRON F3SG-RE (Figura 4.9) e conseqüentemente para a sua aplicação foi necessária a utilização de um relé de segurança da mesma marca com referência G9SE-201. Para ligação deste elemento de segurança ao robô realizam-se as entradas de segurança configuráveis (entradas CI0 e CI1). A utilização de redundância nas ligações prende-se pelo facto das normas de proteção de equipamentos industriais assim o exigirem.

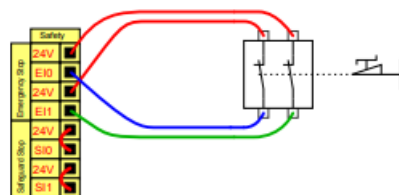


Figura 4.10: Ligações no Robô do Botão de Emergência [19].

Na ligação do botão de emergência exterior da célula foram utilizadas as saídas diretamente do robô (Figura 4.10) conforme aconselhado pelo manual de funcionamento.

## 4.2 Programação e Configurações

No que diz respeito a programação do robô, esta deve ser verificada no que diz respeito a colisões (entre o robô e o restante equipamento da célula) e quanto à necessidade de todas as localizações onde se pretende que o robô efetue operações. Depois de desenvolvidos, os programas e configurações podem ainda ser otimizados quer em relação a tempos de ciclo, otimizações de trajetória e adaptação de parâmetros de movimento.

### 4.2.1 Configurações Iniciais do Robô

Para a definição do TCP foram utilizados dados do fabricante uma vez que os mesmos já estavam definidos para a integração das duas ferramentas (câmara e garra) eliminando assim erros de medida que poderiam ser cometidos. Esta configuração é feita diretamente na consola do robô (Figura 4.11).

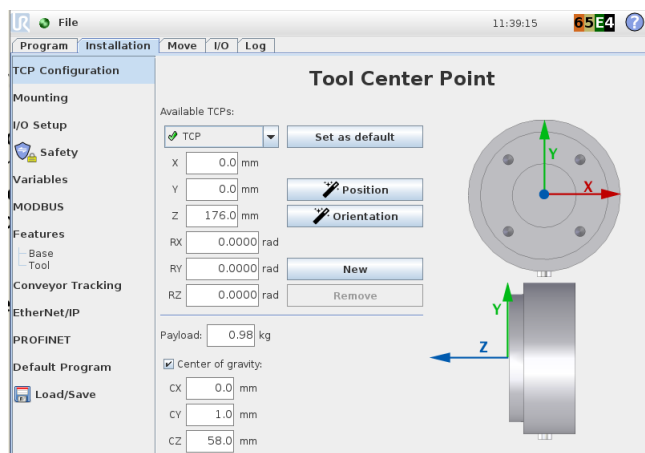


Figura 4.11: Definição do TCP com a Montagem de Garra e Câmara.

Para a combinação implementada da garra *2-finger-85* com *Wrist camera* a massa do conjunto é 0,975 Kg e o centro de massa em X, Y e Z é respetivamente 0,1 e 58 milímetros e o TCP fixa-se nos 176 milímetros em Z.

### 4.2.2 Instalação de *Software* dos Periféricos do Robô

Os periféricos do robô correspondem à garra e à câmera. Foram instalados no controlador do robô ambos os *software* que acompanhavam cada equipamento denominado de *URcap*. Para o funcionamento da garra foi necessário apenas a instalação do referido *software*. No caso da câmera houve a necessidade de acompanhar a instalação do *software* com a instalação de uma *dongle* de chave de produto assim como uma *pen* onde está o *software vision system* programa este responsável pela identificação dos objetos. Para além da instalação do *software* da câmera foi necessário fazer a calibração com um quadro de calibração disponibilizado pelo fornecedor (Figura 4.12).



Figura 4.12: Quadro de Calibração da Câmera para UR3.

Depois de colocado o quadro de calibração e iniciado o processo de calibração o robô realiza movimentos automaticamente para aquisição de 27 imagens em posições e orientações diferentes.

### 4.2.3 Definições de Segurança

Nas definições de segurança existe já o menu pré definido para selecionar as saídas ou entradas. Estas entradas e saídas encontram-se já em ligações emparelhadas (Figura 4.13) não podendo ser alterada a ordem nem utilizar apenas uma só entrada, correspondendo a uma exigência das normas ISO para máquinas e aplicações industriais.

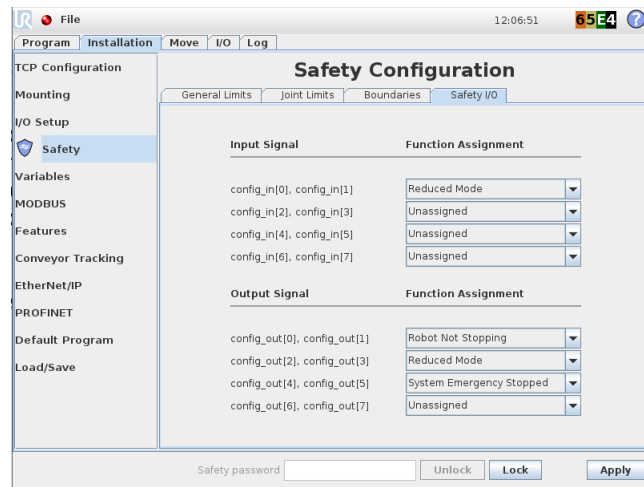


Figura 4.13: Configurações das Entradas e Saídas de Segurança.

Nas entradas é ligado o sinal vindo do relé de segurança pelo que, a entrada é configurada para que o robô entre no modo reduzido (Figura 4.14) entradas CI0 e CI1.

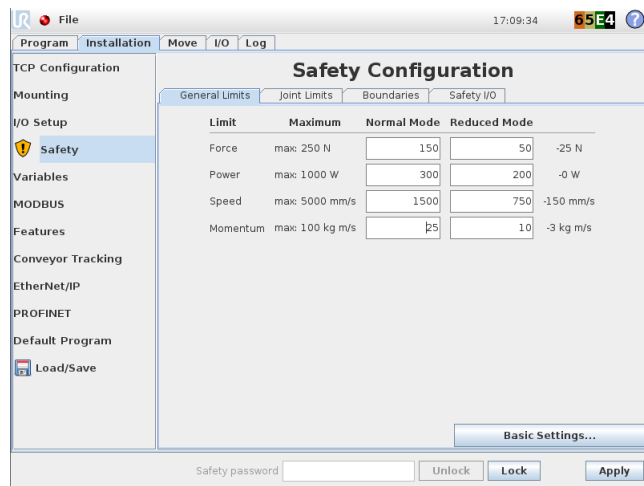


Figura 4.14: Configurações de Parâmetros dos Modos Reduzido e Normal.

Podem ser configurados parâmetros como força, velocidade e aceleração do robô quando ativado qualquer um dos modos de funcionamento. No que toca às saídas de segurança estão conectados o semáforo onde as ligações estão também divididas em canais duplos, estando então atribuídos a luz vermelha às saídas CO4 e CO5, a luz amarela ligada às saídas CO2 e CO3 e por fim a luz verde ligada às entradas CO0 e CO1.

### 4.2.4 Componente de Comunicação

A comunicação é necessária para que haja ligação entre a aplicação a correr no *tablet*, o robô e os seus periféricos. Essa comunicação na parte do robô definiu-se como método de ligação o Dynamic Host Configuration Protocol (DHCP) sendo que a atribuição do endereço fica a cargo do roteador no qual se encontra ligado. O protocolo de comunicação utilizado foi o *Modbus* e a comunicação é feita através da escrita e leitura de registos (Figura 4.15).

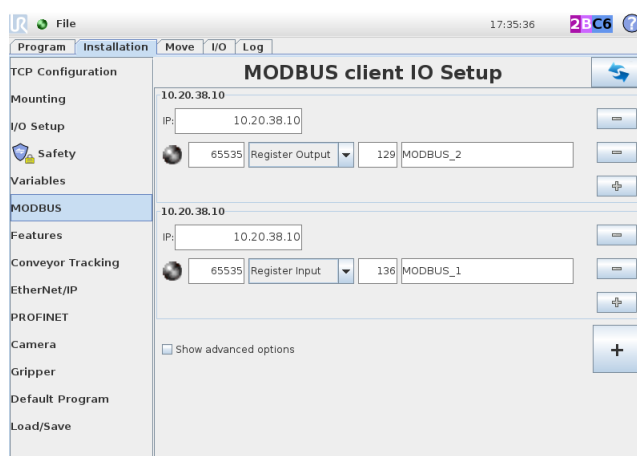


Figura 4.15: Configurações da Ligação *Modbus*.

A comunicação está restrita a um registo de leitura (porta 129) e outro de escrita (porta 136) onde são trocados códigos predefinidos.

### 4.2.5 Programação do Robô

O código de programação do robô pode ser dividido em cinco partes. A maior, onde é feita a programação da torre de *Hanoi*, outras duas partes correspondentes a cada um dos cenários, a parte de comando de operações comunicação e por fim o subprograma de comando da garra.

### Programação da Torre de *Hanoi*

A programação da torre de *Hanoi* foi desenvolvida em cima da função *Switch*. Esta necessidade prendeu-se pelo facto da execução dos cenários obrigar há introdução de interrupções da execução da torre conseguindo assim guardar o número do próximo *case* a processar. Com a utilização deste tipo de função consegue-se também garantir que o robô só irá fazer a interrupção para a execução do cenário, quando terminar o movimento que se encontra a realizar ou seja, se já iniciou o processo de pegar na peça do pino A a interrupção ficará em espera até que o robô volte a deixar a peça no pino B. O programa está construído para desenvolver a torre de *Hanoi* de forma contínua; numa primeira fase mover todas as peças de acordo com as regras do jogo do pino A para B e posteriormente do pino B para o A.

### Programação do Cenário de Completar Tabuleiro

Depois de iniciada a execução deste cenário a programação passa por verificar a zona de trabalho comum através da função *Camera Locate* caso identifique a zona da letra V, desobstruiria, e calcularia as coordenadas para a colocação de peça vai ao *storage* pegar numa peça e por fim coloca-a nas coordenadas anteriormente calculadas. A ordem da colocação de peça, no caso de faltar mais que uma, é feito de forma arbitrária, e de forma independente realizando uma localização de cada vez. O funcionamento da função de deteção da câmara comporta-se segundo a função *if*; por outras palavras são executadas as linhas dentro da função se esta detetar as características definidas, neste caso a letra V. Quando não forem detetadas peças o programa envia sinal por *modbus* a informar que finalizou a tarefa, sendo que também é utilizada a comunicação para informar o operador no caso de não haverem peças no *storage* acabando aí a execução do cenário até que seja novamente chamado para terminar.

### Programação do Cenário de Arrumação de Peças

Para a execução do cenário 2 o robô move-se para uma posição de reconhecimento da zona de trabalho onde executa duas verificações distintas com a função *Camera Locate*, uma para o tipo de peças cilíndricas e outra para o tipo de peças cabos. Foi atribuída prioridade à arrumação das peças cilíndricas pelo que só no fim de arrumar todas as peças deste tipo é que passa para a deteção e arrumação dos cabos. Quando a função da câmara não detetar nenhuma das peças o robô informa, tal como no cenário anterior, o operador da finalização da tarefa.

### Subprograma de Comunicação

Como descrito anteriormente o subprograma de comunicação consiste em estar sempre à escuta do canal de comunicação vindo diretamente do computador ou seja da porta 136 para daí tomar decisões de execução dos cenários conforme a tabela 4.2.

Tabela 4.1: Descrição dos Códigos Escritos na Porta 129 *Modbus* pelo Computador.

Código	Descrição
1	Executar Cenário 1
2	Executar Cenário 2
4	ACK

O sistema utiliza ainda a porta 136 com os códigos descritos na tabela 4.1 para enviar informações como processo concluído ou então códigos de Acknowledgement (ACK).

Tabela 4.2: Descrição dos Códigos Escritos na Porta 136 *Modbus* pelo Robô.

Código	Descrição
99	ACK
100	Cenário concluído
101	Sem peças no Armazém

A comunicação *modbus* entre robô e computador funciona sobre o princípio pergunta-resposta, ou seja, sempre que há um pedido tem que haver a confirmação de receção do pedido (Figura 4.16) .

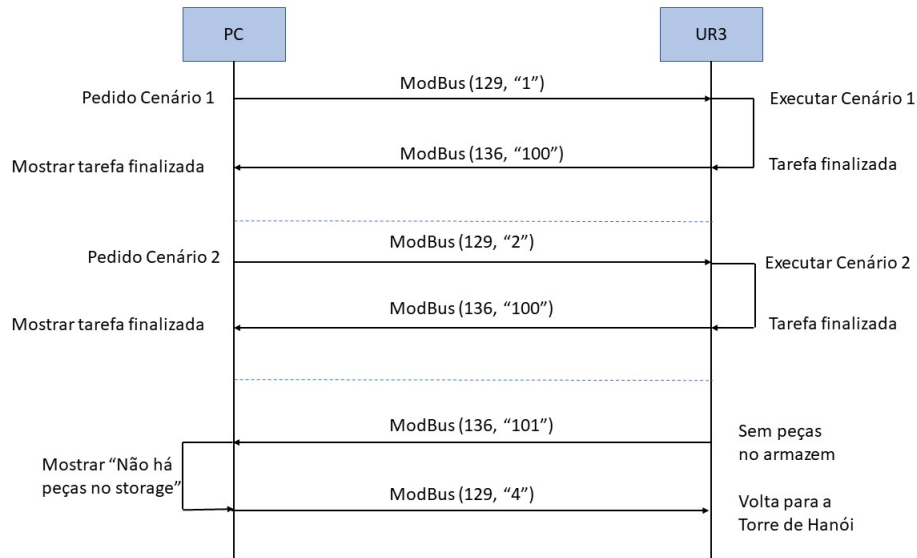


Figura 4.16: Esquema de Comunicações Robô-PC.

### Subprograma de Comando Adaptativo da Garra

A construção de um subprograma para realizar um comando adaptativo da garra partiu da necessidade de corrigir um problema de funcionamento que surgiu aquando da construção do programa da torre de *Hanoi*. Nomeadamente, o problema estava no facto de na mesma posição poderem haver peças de diâmetros diferentes para garra apanhar e consequentemente o TCP variar em altura (i.e. na coordenada Z). Esta situação foi detetada após a marcação das posições para pegar numa determinada peça, uma vez que estas foram efetuadas para uma peça com um determinado diâmetro. Posteriormente, se no decorrer do código, para uma determinada posição, o diâmetro da peça fosse inferior ou superior, à peça com a qual foi marcada a posição, havia o problema de ajuste de *offset*, na coordenada Z, havendo a possibilidade de o robô pegar muito em cima ou muito em baixo.

Esta nota de funcionamento vem retratada no manual de funcionamento da Garra (Figura 4.17) onde é descrita a variação da posição da garra que chega a ter a diferença de 10 mm consoante a variação do ângulo de abertura [34]. Esta situação foi corrigida no código de comando da garra, através da função exibida na imagem acima mencionada.

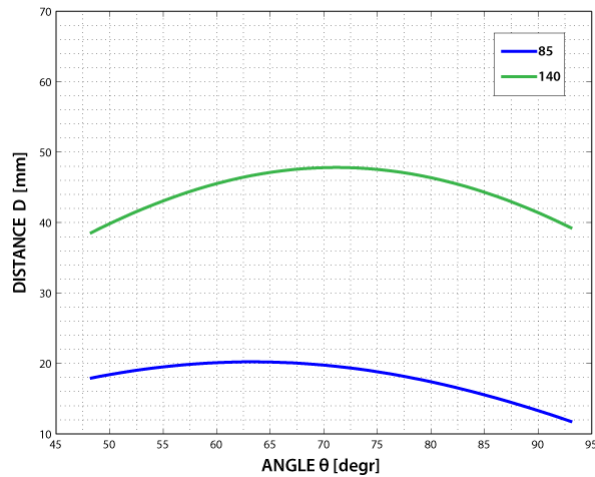


Figura 4.17: Variação da Posição da Garra consoante o Ângulo de Abertura [34].

Este subprograma foi conseguido de forma iterativa pelo que ao fim de vários testes foi possível chegar a uma solução aceitável. Esta solução recai sobre a leitura do consumo de corrente que a garra consumiu até uma determinada posição de fecho e, com essa leitura, fazer ou não o incremento em altura da posição inicial. O valor da corrente gasta para chegar a um determinado ângulo de fecho varia de 0 a 255 pelo que 0 corresponde a garra totalmente aberta e 255 totalmente fechada.

SubProgram\_1

```

pergarra=70
Loop rq_is_object_detected()== False and pergarra < 230
  posgarra=get_actual_tcp_pose()
  possub=pose_add(posgarra , subir)
  rq_move(pergarra)
  movel(pose_add(posgarra , subir) , a=1,v=0.25 , t=0,r=0)
  pergarra=pergarra+40

```

Quando o programa é chamado a garra é fechada até ao valor 70, valor este que corresponde ao consumo de corrente necessário para fechar a garra e prender a peça de maior diâmetro (disco azul). Após o fecho da garra é verificado se a garra deteta alguma peça, caso não detete o valor da corrente é incrementado em mais 40 e o robô é

incrementado na coordenada Z em 2 mm. Este processo é repetido até que a garra detete que tem peça ou então que chegue ao valor máximo de corrente consumida fixado nos 255. Fazendo a soma do número de incrementos do valor de corrente (70 iniciais mas 40 por quatro vezes) verifica-se que apenas se chega ao valor máximo de 230 que é enviado para a garra, deixando uma margem de 25, margem esta que está também calculada para que, no caso de alguma peça não estar corretamente colocada o robô não chega sequer a tocar no pino, uma vez que o valor da corrente consumida por movimento está limitada.

### 4.3 Construção da Aplicação HMI

A aplicação a correr no *tablet* é a única forma de interação com o operador para com a célula e seus integrantes, pelo que tem como função não só receber pedidos do operador, mas também informar estados de funcionamento e falhas.

A construção desta aplicação foi desenvolvida com recurso a plataforma Node-RED [35]. Esta plataforma utiliza uma abordagem de programação gráfica, tornando-se possível desenvolver uma aplicação através da conexão entre blocos. Adicionalmente, esta plataforma disponibiliza diversas categorias de blocos, permitindo a utilização facilitada a diversos protocolos de comunicação e a programação lógica, assim como a criação intuitiva de componentes gráficos [36].

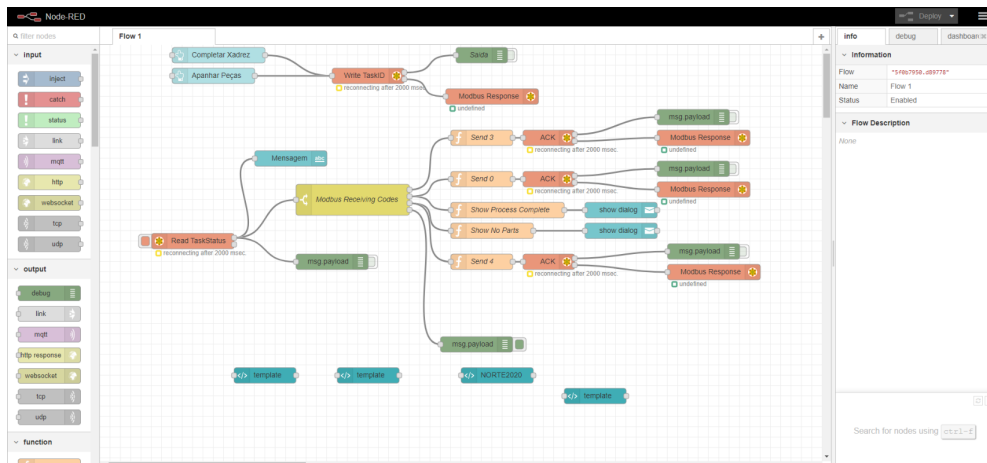


Figura 4.18: Programação dos Nós na Plataforma Node-RED.

Esta aplicação, apresentada na Figura 4.18, faz a conversão dos códigos enviados de e para o controlador do robô (referidos anteriormente na secção 4.2.5) por meio da comunicação *Modbus*.



Figura 4.19: Aplicação Gráfica a Correr no *Tablet*.

De forma sucinta, a programação desenvolvida consiste em blocos de leitura e escrita de endereços definidos no *Modbus* para uma posterior visualização gráfica apelativa e intuitiva (Figura 4.19) [36]. Esta interface gráfica embora aparente ter um grau de dificuldade alto, a sua implementação não obriga o programador a ter grandes conhecimentos de programação.



# Capítulo 5

## Testes Experimentais

A automatização da célula constitui o principal objetivo, pelo que a fase de testes valida todos os pressupostos iniciais de funcionamento. Neste capítulo estão compreendidos todos os testes de validação à solução aplicada, testes estes que podem ser distinguidos em três tipos, elétrico, segurança e o desenvolvimento do processo em si.

### 5.1 Testes aos Dispositivos Elétricos

O teste elétrico tem como principal objetivo verificar o cumprimento da norma IEC 60204-1 (Segurança de máquinas), e deste modo garantir que a célula tem a robustez e as proteções necessárias para o bom funcionamento da estrutura projetada. De entre os testes necessários foram realizados: o teste à tensão residual e o teste ao interruptor diferencial.

- O teste à tensão residual consiste em alimentar o sistema e todos os seus equipamentos, esperar que os equipamentos carreguem os seus condensadores e bobines, cortar o circuito e após 5 segundos do corte de energia à célula, a tensão residual da mesma seja inferior a 60 V.
- O teste ao interruptor diferencial realiza-se após alimentar todos os circuitos presentes na célula, pressionar o botão de teste do interruptor diferencial e em seguida

verificar que a proteção diferencial atuou.

Os testes aos dispositivos elétricos visam a proteção dos seres humanos contra choque elétrico, tendo este ponto ficado garantido. Dentro dos testes elétricos foram ainda verificadas continuidades em todo o circuito, usando um multímetro para o efeito, e analisados os isolamentos da cablagem.



Figura 5.1: Resultado da Implementação da Iluminação.

A iluminação artificial foi implementada para permitir um posterior controlo da luminância de modo a que a captura da imagem pela câmara seja realizada nas melhores condições. Desta forma, esta solução não acusou qualquer falha ficando ainda em perfeito enquadramento estético com a construção da restante estrutura (Figura 5.1).

## 5.2 Testes aos Dispositivos de Segurança

Os testes à segurança da célula compreendem o acionamento de dispositivos de emergência (botões de emergência e barreiras fotoelétricas) e visualização de estados de funcionamento através do semáforo colocado.

O teste ao indicador luminoso passou pela verificação correta de ativação da luz correspondente ao estado atual do funcionamento. Desta forma, verificou-se que as cores

vermelho, amarelo e verde, correspondem respectivamente, a paragem de emergência, robô em movimento normal e robô a operar em modo reduzido.

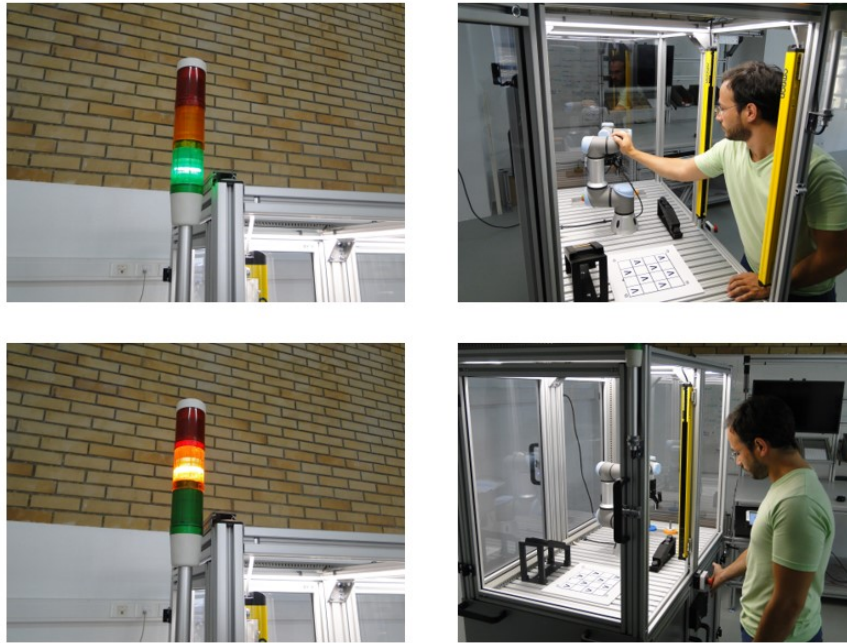


Figura 5.2: Testes de Funcionamento dos Dispositivos de Segurança.

Na Figura 5.2 estão representados os testes do semáforo, acende a luz verde quando o robô se encontra em modo reduzido querendo com isto dizer que o operador interrompeu as barreiras. O significado da cor do semáforo é demonstrativo de que é seguro o operador permanecer na área de trabalho comum. Quando o robô se encontra a trabalhar no seu estado normal, a célula permanece com a luz amarela, indicando que o operador pode entrar na área comum mas tendo cuidado com os movimentos do robô, uma vez que este se encontra a trabalhar em modo normal. Finalmente, é possível verificar que o botão de emergência se encontra num local de fácil acesso e visualização, estando este permanentemente desobstruído. Quando acionado, o semáforo passa ao estado de intermitente vermelho e o robô e os seus periféricos entram em modo de emergência, parando qualquer movimento e/ou ação que estejam a executar. É ainda possível garantir que quando acionado o botão de emergência o mesmo não desarma de forma automática, sendo para tal necessário uma nova e diferente ação manual (girar para libertar).

Outros testes realizados foram por exemplo testar a paragem de proteção perante o toque com o humano durante a permanência de ambos na área de trabalho comum, pelo que se verificou que a força exercida por parte do robô (no estado de funcionamento reduzido) aquando do contacto físico, não era suficiente para causar sequelas físicas.

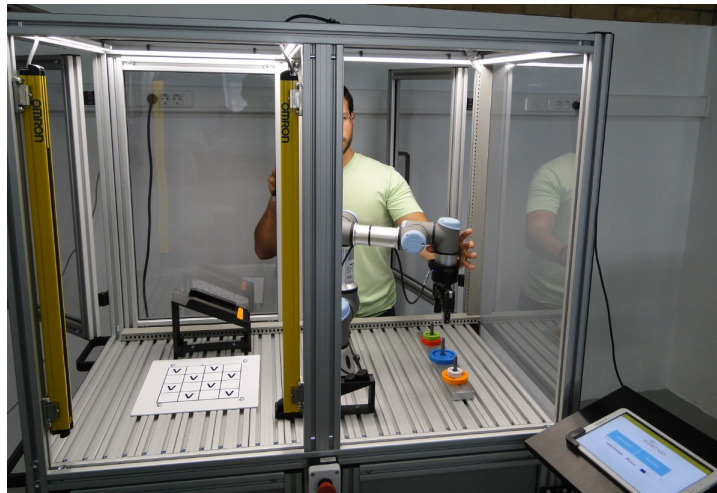


Figura 5.3: Testes de Paragem de Proteção do Robô.

Caso o operador, por qualquer motivo, contrariar as barreiras físicas, ou seja, abrir uma janela de trás da célula e entrar em contacto com o robô (Figura 5.3), embora a força de impacto exercida seja maior comparativamente com o caso anterior (notar que neste caso o robô está a operar em modo normal) de igual forma o operador não sofre danos. É de salientar que este teste foi repetido inúmeras vezes.

### 5.3 Testes ao Funcionamento de Processos

Os testes de processo têm como principais características verificar o funcionamento dos cenários assim como validar as soluções implementadas, qualidade de execução e simular possíveis imprevistos de execução para contemplar todas as falhas possíveis.

### 5.3.1 Desenvolvimento da Torre de *Hanoi*

A realização da torre de *Hanoi* foi testada enumeras vezes, tentando verificar a ocorrência de falhas assim como introduzir inúmeras interrupções ou execuções dos cenários pelo que o sistema revelou robustez suficiente para afirmar que o funcionamento da mesma é fiável.

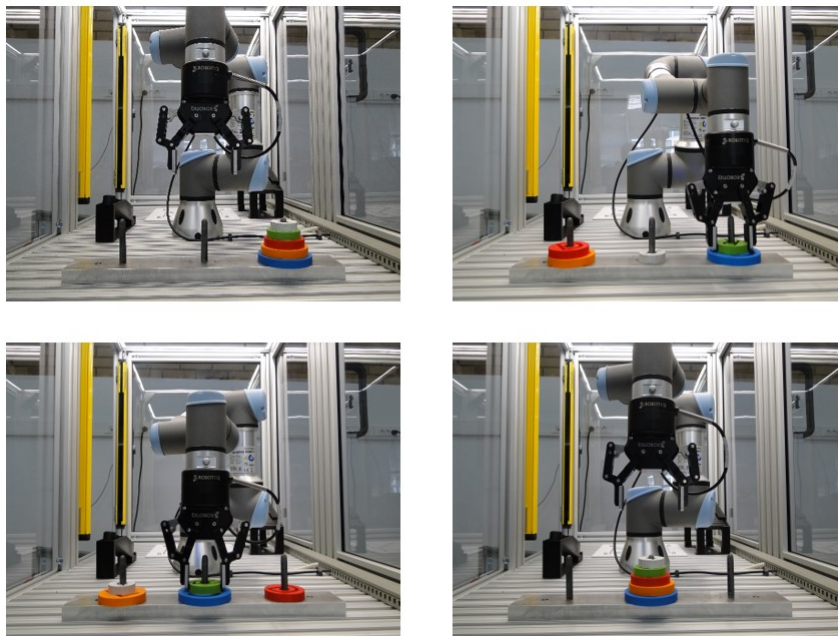


Figura 5.4: Imagens de Funcionamento da Torre de *Hanoi*.

Ao iniciar o programa é solicitado ao operador que verifique ou que coloque na posição inicial as peças. Esta localização e ordem inicial dos discos encontra-se retratada na primeira posição da Figura 5.4 pelo que nas restantes posições estão retratados movimentos intermédios e o final. Caso os discos não se encontrem inseridos nos respetivos pinos o sistema está preparado para não agarrar os respetivos pinos uma vez que, a percentagem de fecho da garra está limitada. No anexo D encontram-se todos os estados do desenvolvimento da torre de *Hanoi* através de uma foto-descrição.

### 5.3.2 Desenvolvimento do Cenário de Colocação de Peças

Na realização do cenário de colocação de peças no tabuleiro os testes realizados mostraram-se totalmente satisfatórios. O teste de falha realizado recaiu sobre colocar menos peças na rampa de *stock* do que as necessárias. Neste caso, o robô reconhece a não presença de peça agarrada e envia para o operador a mensagem de falta de peças, pelo que a não ocorrência desta falha está garantida.

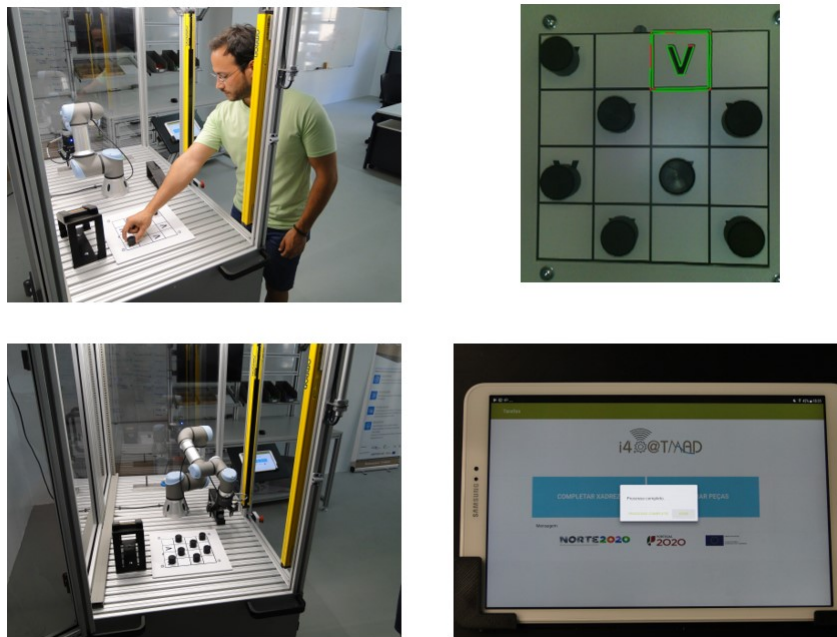


Figura 5.5: Imagens de Funcionamento do Cenário 1.

Na Figura 5.5 é descrito o funcionamento do cenário pelo que, o operador tem a tarefa de colocar as peças nas posições da letra V, quando solicita a colaboração do robô o mesmo desloca-se para a posição de reconhecimento e captura a imagem para processar e identificar o local de descarga compreendido pela letra V e o quadrado em volta. Este procedimento é repetido para cada uma das posições necessárias e, caso o operador deseje colocar mais peças no tabuleiro, o mesmo reconhece-as desde que estejam no local correto, fazendo a sua validação. No final do processo, i.e. quando todas as peças forem colocadas, o robô informa o operador de processo finalizado. O teste foi ainda realizado em diferentes ambientes de luminosidade, pelo que as falhas foram nulas.

### 5.3.3 Desenvolvimento do Cenário de Arrumação de Peças

Como tarefa deste cenário, o operador tem que arrumar as peças despejadas na área comum de trabalho, sendo que, sempre que este precisar de apoio faz a requisição através da aplicação de interface gráfica desenvolvida.

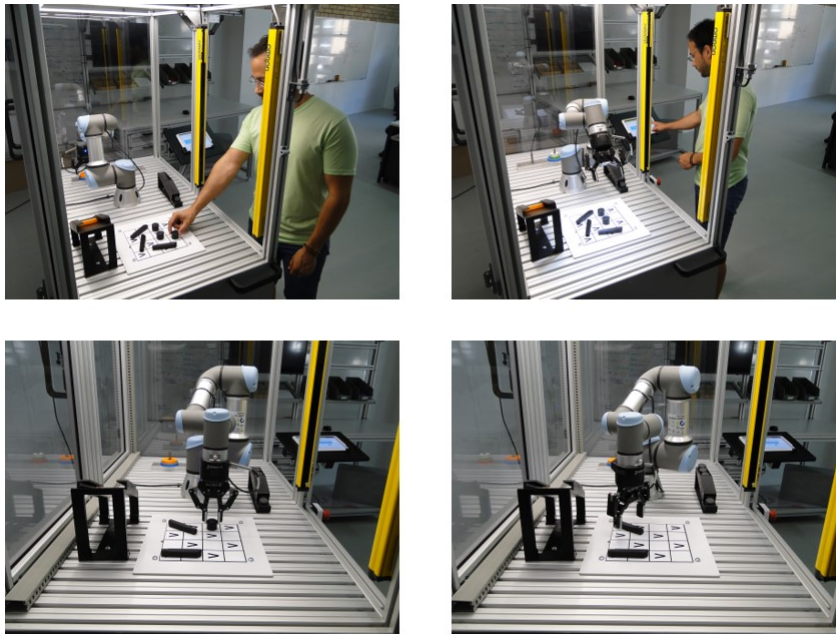


Figura 5.6: Imagens de Funcionamento do Cenário 2.

Para testar o funcionamento do cenário de arrumação de peças, foram colocados diferentes números de peças cilíndricas e de cabos (Figura 5.6), pelo que o funcionamento não revelou qualquer anomalia tanto ao nível de identificação da posição da peça como no tipo de peça identificada. As peças em forma de cabo, mesmo sendo de características mais complexas, foram colocadas de forma arbitrária e a sua identificação foi devidamente realizada assim como o seu *pick* fora realizado da forma pretendida.



# Capítulo 6

## Conclusões e Trabalhos Futuros

Ao longo deste documento, foi apresentado e descrito todo o trabalho realizado no âmbito desta dissertação. Além do levantamento bibliográfico realizado, foi apresentado um relatório detalhado da construção/projeção de uma célula robótica.

### 6.1 Conclusão

As tecnologias inovadoras trazidas pela indústria 4.0, mais propriamente a automação industrial, e neste caso específico a robótica colaborativa, tem vindo a ganhar terreno e é utilizada nas mais diversas áreas, constituindo um fator de aumento de competitividade por parte das empresas. Nas células tradicionais, têm vindo a ser melhorados os processos de monitorização e processamento de dados, gerando informação para posterior análise. Os sistemas de informação têm sido otimizados para o processamento de grandes quantidades de dados gerados por todas os equipamentos e sensores atualmente ligado à rede.

Com a implementação da HMI e da aplicação desenvolvida, é possível dar ao operador, num aspeto muito mais intuitivo, eficiente e prático do controlo de uma máquina que de outro modo seria difícil identificar. Adicionalmente, com o HMI é possível diagnosticar e apresentar erros e alarmes ao operador.

Neste sentido e levando em consideração as premissas do projeto I4.0@TMAD, no

qual o tema desta dissertação se insere, foi possível durante os eventos realizados para empresários, providenciar conceitos e tecnologias inovadoras, pelo que a bancada teve uma grande aceitação, gerando assim novas ideias de melhoria dos processos instalados nas suas empresas por parte do público alvo.

As ideias transmitidas com a proposta desta bancada foram, entre outras, esperar uma diminuição do risco de erro humano, melhor segurança para o operador, uma maior eficiência no processo, redução de custos, inventário permanente e automático. A utilização do conjunto robô/humano foi entendida como uma solução flexível que permite, quando aplicado no meio empresarial, uma rápida adaptação a uma vasta gama de produtos. A célula foi desenhada e construída com o intuito de se vincular o mínimo possível a situações concretas, tendo-se estudado situações o mais genéricas possível.

Na construção da célula foram considerados diversos aspetos de robustez e de aplicabilidade, que noutra enquadramento teriam sido descuidados, sendo esses aspetos justificados pelo facto de a célula ficar exposta num centro de investigação no qual vão ser realizados ensaios e testes. A funcionalidade dos equipamentos não foi levada ao seu nível máximo de desempenho criando assim abertura e base para futuros projetos ou dissertações.

O objetivo central era ter uma bancada que fosse apelativa, robusta e que sobretudo que despertasse interesse e fosse explícita nos conceitos demonstrados pelo foi durante os 3 eventos de demonstração (Figura 6.1) pela região TMAD, o interesse por parte do público alvo foi atingido devidos ao comentários e dúvidas demonstradas.

Posteriormente a cada um dos eventos, ocorreram sessões de geração de ideias e debates de temáticas ligadas às bancadas anteriormente expostas, onde mais uma vez foi possível perceber que as demonstrações influenciaram a maneira como os empresários expunham os seus problemas, usando nessa descrição conceitos de indústria 4.0 para os solucionar.

Em resumo, realizou-se o estudo de uma célula flexível, com a função de demonstrar conceitos e tecnologias inovadoras, na qual foram aplicados conceitos retidos da formação académica em engenharia eletrotécnica e de computadores e da experiência profissional na área da manutenção industrial, assim como dicas dos fornecedores dos equipamentos. Foi possível ao longo do desenvolvimento da dissertação aprender e evoluir em áreas como



Figura 6.1: Foto do Evento Demonstrativo de Bragança [37].

design de produto, impressão 3D assim como a componente comunicativa uma vez que foi necessário aprender conceitos de exposição de ideias e de motivar o mais possível o público alvo.

## 6.2 Trabalhos Futuros

O trabalho desenvolvido teve como principal objetivo a criação de uma bancada demonstrativa de conceitos e valências. Paralelamente a este objetivo foram também criados pontos e bases para alavancar projetos futuros no âmbito académico como outras teses de mestrado e até base para teses de doutoramento. Neste sentido são várias as melhorias e desenvolvimentos que podem valorizar o trabalho efetuado até ao momento, das quais se destacam as seguintes:

- Estudo do comportamento dinâmico da estrutura do sistema robótico e desenvolver otimização de trajetórias;
- Desenvolver a auto-adaptabilidade luminosa da célula seja qual for o ambiente em que a célula esteja inserida.
- Aplicar câmara para deteção e reconhecimento de operador e com isso adaptar a performance do robô (aceleração velocidade e força) para cada operador e adaptar

a luminosidade da célula;

- Utilização de câmara para fazer reconhecimento de presença de operador nas diferentes zonas de trabalho ou seja, criar restrições virtuais e com isso adaptar ou parar, o funcionamento do robô.
- Introduzir traceabilidade nas peças que são movimentadas na célula e gravar não só o momento de produção mas também o operador que a realizou entre outros parâmetros que se achem relevantes.
- Aplicação da célula para resolução de problemas concretos que os empresários colocaram durante as demonstrações.

# Bibliografia

- [1] R. Gama, J. F. Lopes, F. Pires, P. Leitão, J. Barbosa e A. P. Lousada, «Diagnósticos tecnológicos relativos a pelo menos 4 sectores industriais regionais- Relatório interno», 2018.
- [2] International Federation of Robotics, *International Federation of Robotics - Robot History*, 2018. URL: <https://ifr.org/robot-history> (acedido em 06/04/2018).
- [3] Simio, *industry-4.0*. URL: <https://www.simio.com/applications/industry-40/industry-40.png> (acedido em 31/07/2018).
- [4] K. Schwab, «The Fourth Industrial Revolution», rel. téc., 2016. URL: [www.weforum.org](http://www.weforum.org).
- [5] P. D. António Brochado Correia, «PwC “Global Industry 4.0”», rel. téc., 2017. URL: [www.pwc.pt/industria40](http://www.pwc.pt/industria40).
- [6] P. Abreu e A. M. Lopes, «Robótica Industrial Textos de apoio», rel. téc., 2001.
- [7] S. Y. Nof, *Handbook of industrial robotics*. John Wiley, 1999, p. 1348, ISBN: 0471177830.
- [8] L. A. M. Riascos, *Fundamentos de Robótica - manipuladores e robôs móveis*. 2010, p. 211, ISBN: 978.85.7651.158.8.
- [9] Linuxmagazine, «O robô Nao humano», 2012.
- [10] StarWars, *C-3PO (See-Threepio) | StarWars.com*, 2018. URL: <http://www.starwars.com/databank/c-3po> (acedido em 05/04/2018).
- [11] M. I. Robots, *MiR100<sup>TM</sup> | Mobile Industrial Robots*, 2018. URL: <http://www.mobile-industrial-robots.com/en/products/mir100/> (acedido em 05/04/2018).

- [12] ABB, *IRB 1410 - Industrial Robots From ABB Robotics*, 2018. URL: <http://new.abb.com/products/robotics/industrial-robots/irb-1410> (acedido em 06/04/2018).
- [13] A. Mendes Lopes, «MODELAÇÃO CINEMÁTICA E DINÂMICA», 2002.
- [14] T. Freire e U. Federal, *Aplicação de Robôs nas Indústrias*, 2015.
- [15] U. robots, «UR3 Manual», rel. téc., 2018.
- [16] M. D. E. Paralela, «Manipulador Robótico», rel. téc., 2002, pp. 1–41.
- [17] M. Peshkin e J. E. Colgate, «Cobots», *Industrial Robot: An International Journal*, vol. 26, n.º 5, pp. 335–341, jul. de 1999, ISSN: 0143-991X. DOI: 10.1108/01439919910283722.
- [18] F. Manuel, «Sistema Robótico Colaborativo utilizando Restrições Virtuais», 2010.
- [19] Universal Robots, *Braços robóticos industriais colaborativos / Cobots da Universal Robots*, 2018.
- [20] Robotiq, *Advanced Assembly of a Robotiq Universal Controller - YouTube*. URL: [https://www.youtube.com/watch?v=rpJ%7B%5C\\_%7D7EkGjzw](https://www.youtube.com/watch?v=rpJ%7B%5C_%7D7EkGjzw) (acedido em 09/04/2018).
- [21] —, *Start Production Faster - Robotiq*, 2018. URL: <https://robotiq.com/> (acedido em 09/04/2018).
- [22] E. Rohmer, S. P. N. Singh e M. Freese, «V-REP: A versatile and scalable robot simulation framework», em *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, nov. de 2013, pp. 1321–1326, ISBN: 978-1-4673-6358-7. DOI: 10.1109/IR0S.2013.6696520.
- [23] «Technology and applications of ABB RobotStudio», *Industrial Robot: An International Journal*, vol. 36, n.º 6, pp. 540–545, out. de 2009, ISSN: 0143-991X. DOI: 10.1108/01439910910994605.

- [24] ABB, *RobotStudio Machining PowerPac - ABB Robotics - Machining Software (Application Software)*. URL: <https://new.abb.com/products/robotics/application-software/machining/robotstudio-machining-powerpac> (acedido em 30/08/2018).
- [25] O. M. Stavridis, R. L. Kajfez, E. A. Riter e M. Y. Mollica, «Modeling real-world objects: Connecting SolidWorks to Toy Adaptation», em *2016 IEEE Frontiers in Education Conference (FIE)*, IEEE, out. de 2016, pp. 1–7, ISBN: 978-1-5090-1790-4. DOI: 10.1109/FIE.2016.7757744.
- [26] T. Noqueira, «Redes de comunicação para sistemas de automação industrial», rel. téc., 2009.
- [27] Modbusorg, «MODBUS Application Protocol 1 1 b», rel. téc., 2006. URL: <http://www.modbus-ida.org>.
- [28] Microsoft, *Kinect for Windows Sensor Components and Specifications*, 2018. URL: <https://msdn.microsoft.com/pt-pt/library/jj131033.aspx> (acedido em 19/04/2018).
- [29] International Organization for Standardization, *ISO 10218-1:2011 - Robots and robotic devices – Safety requirements for industrial robots – Part 1: Robots*, 2011.
- [30] —, *ISO9241-210:2010(en), Ergonomics of human-system interaction— Part210: Human-centred design for interactive systems*, 2010.
- [31] J. Dul e B. Weerdmeester, *Ergonomia prática*. Edgard Blücher, 2004, ISBN: 9788521203490. URL: <https://www.blucher.com.br/livro/detalhes/ergonomia-pratica-369>.
- [32] C. A. Knoblock, *Abstracting the Tower of Hanoi*, 1990.
- [33] Robotiq, «Manual support robotiq Wrist camera», 2018.
- [34] —, «Manual support Griper 2-finger 85», 2018.
- [35] *Node-RED : About*. URL: <https://nodered.org/about/> (acedido em 23/08/2018).

- [36] Z. Chaczko e R. Braun, «Learning data engineering: Creating IoT apps using the node-RED and the RPI technologies», em *2017 16th International Conference on Information Technology Based Higher Education and Training (ITHET)*, IEEE, jul. de 2017, pp. 1–8, ISBN: 978-1-5386-3968-9. DOI: 10.1109/ITHET.2017.8067827. URL: <http://ieeexplore.ieee.org/document/8067827/>.
- [37] Estação diária 96.8 FM, *Promoção da Indústria 4.0 na Região de Trás-os-Montes e Alto Douro, I4.0@TMAD-Edição Jornal*. URL: <http://estacaodiariajornal.com/projeto-da-estgl-parceria-promove-industria-4-0/>.

# Apêndice A

## Esquemas elétricos



# Esquema Ligações Elétricas

## Livro de documento

Célula Robotica Montagem Colaborativa

Humano - Robot

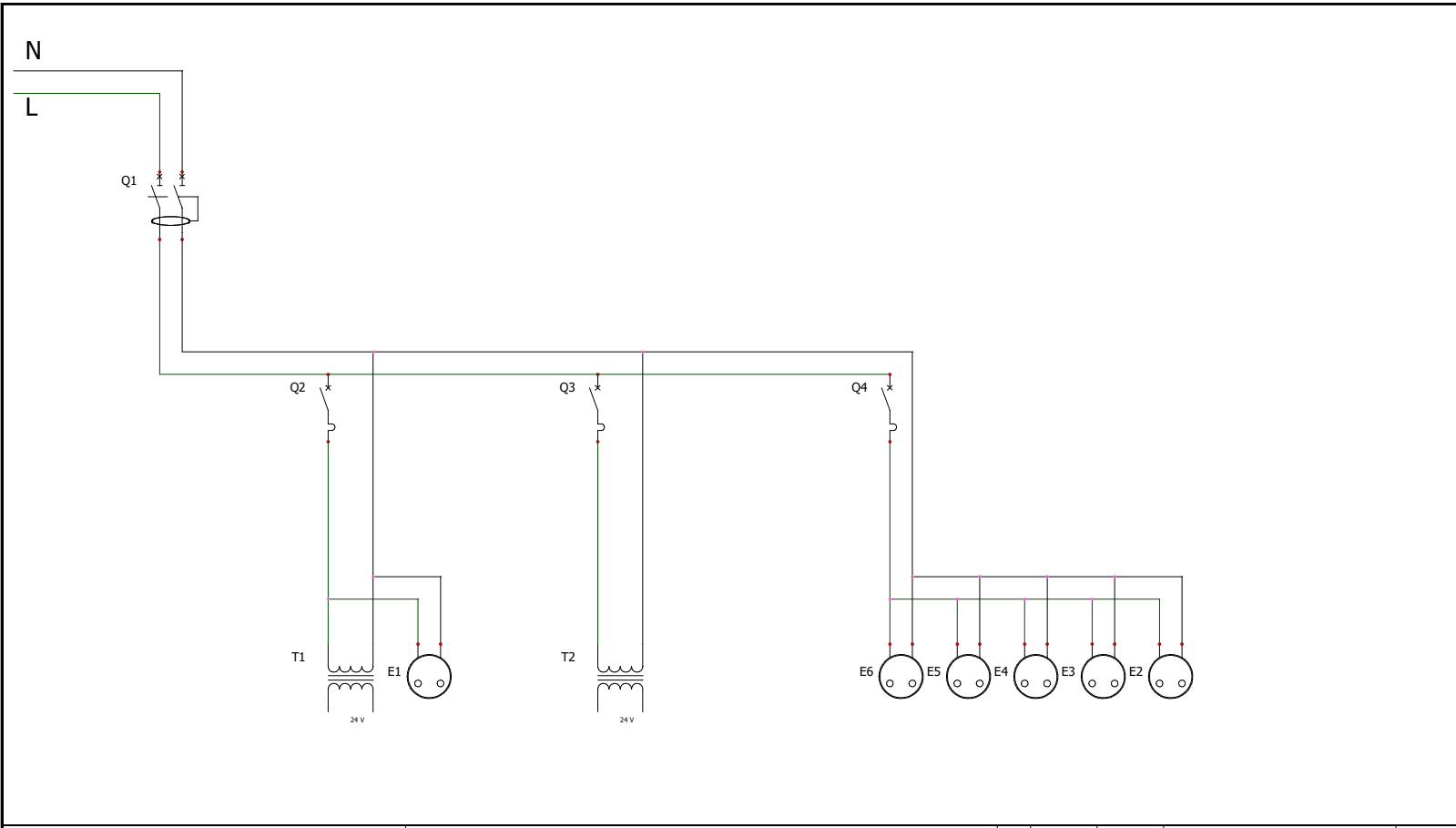
REV.	DATE	NAME	CHANGES	REVISION	SCHEME
0	07/07/2018	Vitor		0	
Vitor de Sousa vitorm.r.s@live.com.pt			CONTRACT:	Dados do usuário 1 Vitor de Sousa Dados do usuário 2 vitorm.r.s@live.com.pt	01

1-Livro de documento

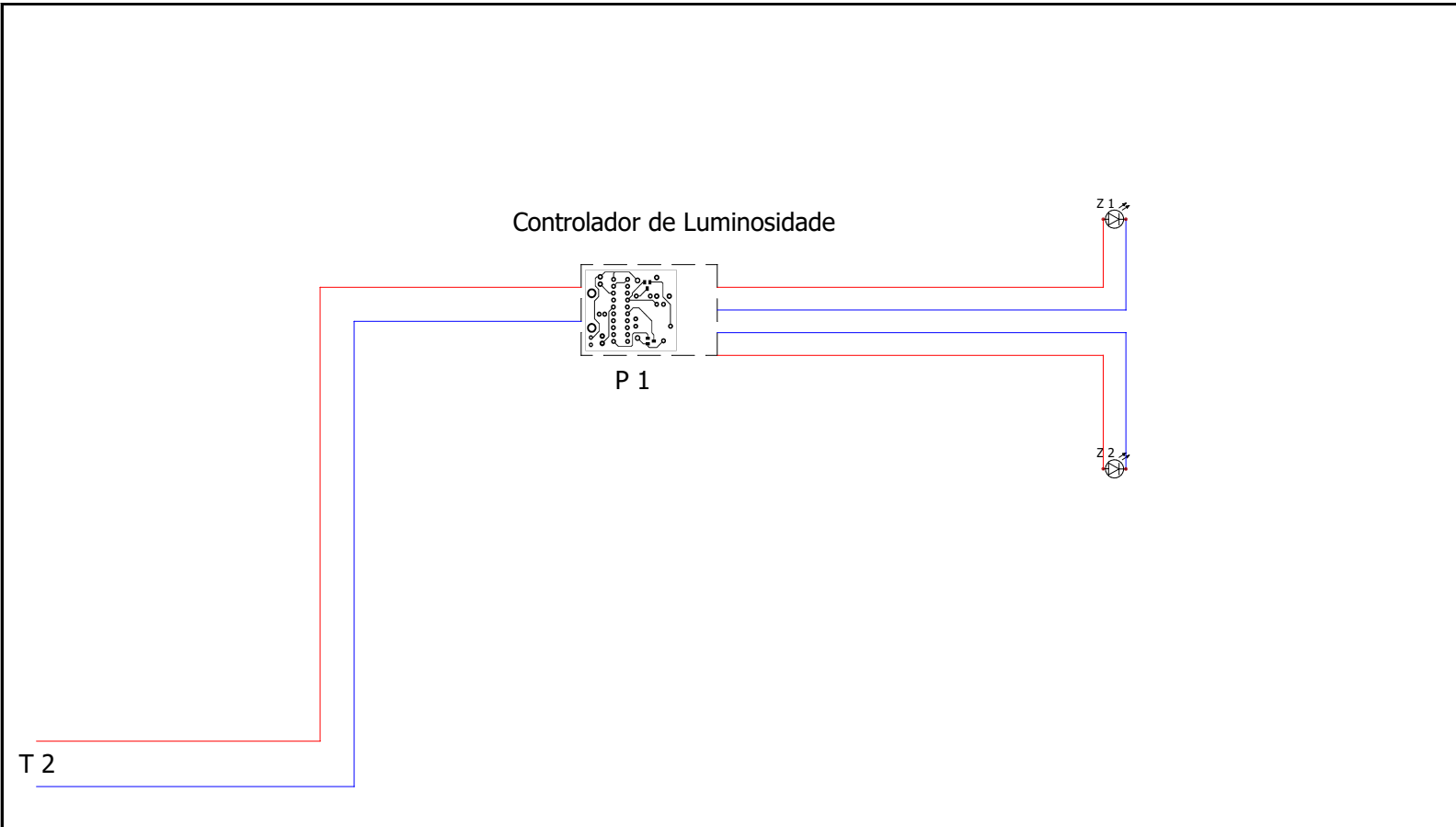
Desenho	Função	Localização	Revisão	Data	Criado por	Descrição do livro	Designação da pasta
01	F1	L1	0	07/07/2018	Vitor	Folha de rosto	
02	F1	L1	0	07/07/2018	Vitor	Lista de desenhos	
03	F1	L1	0	07/07/2018	Vitor	Esquema Quadro Elétrico	
04	F1	L1	0	07/07/2018	Vitor	Esquema Iluminação	
05	F1	L1	0	07/07/2018	Vitor	Esquema de ligação Cortinas e luz	
06	F1	L1	0	07/07/2018	Vitor	Esquema de ligação botão de emergência externo	
07	F1	L1	0	07/07/2018	Vitor	Esquema de ligação Semáforo	
08	F1	L1	0	09/07/2018	Vitor	Legenda	

DesignSpark Electrical

		Lista de desenhos						REVISION
								0
								SCHEME
								02
CONTRACT:	LOCATION:	L1	Painel elétrico principal			Dados do usuário 1	Dados do usuário 2	
						Vitor de Sousa	vitorm.r.s@live.com.pt	

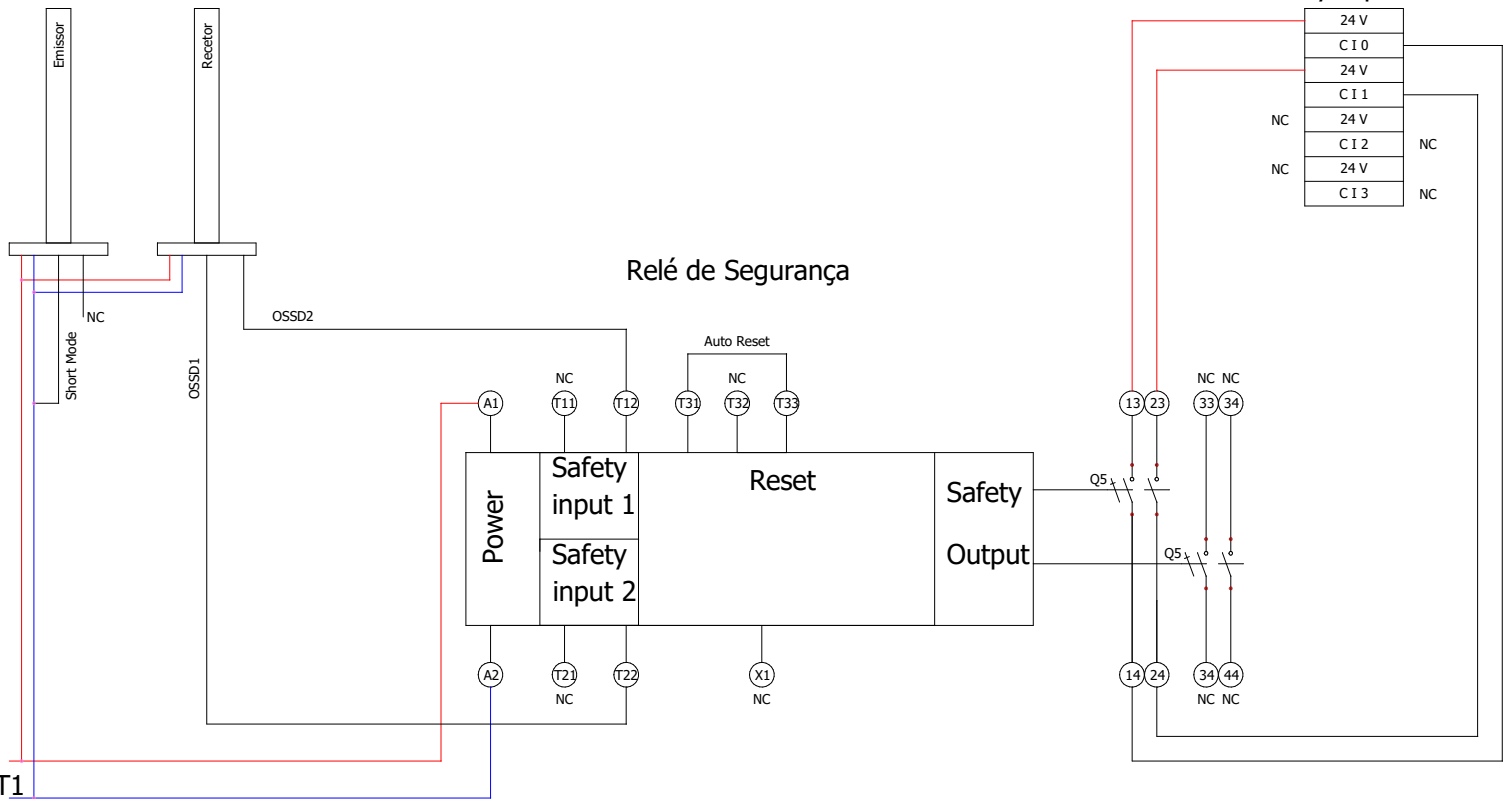


<b>Esquema Quadro Eléctrico</b>		REVISION <b>0</b>																
		SCHEME <b>03</b>																
CONTRACT:	LOCATION: <b>L1</b>	Painel eléctrico principal																
		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">REV.</td> <td style="width: 10%;">DATE</td> <td style="width: 10%;">NAME</td> <td style="width: 70%;">CHANGES</td> </tr> <tr> <td>0</td> <td>07/07/2018</td> <td>Vitor</td> <td></td> </tr> <tr> <td colspan="2">Dados do usuário 1</td> <td colspan="2">Dados do usuário 2</td> </tr> <tr> <td colspan="2">Vitor de Sousa</td> <td colspan="2">vitorm.r.s@live.com.pt</td> </tr> </table>	REV.	DATE	NAME	CHANGES	0	07/07/2018	Vitor		Dados do usuário 1		Dados do usuário 2		Vitor de Sousa		vitorm.r.s@live.com.pt	
REV.	DATE	NAME	CHANGES															
0	07/07/2018	Vitor																
Dados do usuário 1		Dados do usuário 2																
Vitor de Sousa		vitorm.r.s@live.com.pt																



		<b>Esquema Iluminação</b>			REVISION
					0
		REV.	DATE	NAME	CHANGES
CONTRACT:		Dados do usuário 1		Dados do usuário 2	
LOCATION: L1	Painel elétrico principal			Vitor de Sousa	vitorm.r.s@live.com.pt
					SCHEME
					04

# Barreiras Fotoelétricas



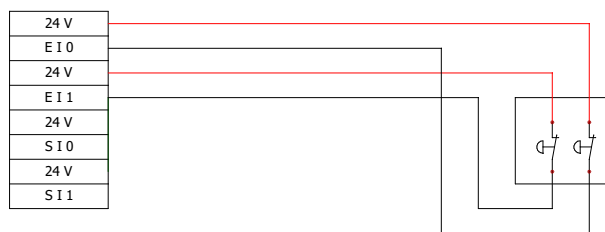
## Esquema de ligação Cortinas e luz

CONTRACT: LOCATION: L1 Painelelctrico principal

REV.	DATE	NAME	CHANGES
0	07/07/2018	Vitor	
Dados do usu1rio 1			Dados do usu1rio 2
Vitor de Sousa			vitorm.r.s@live.com.pt

REVISION	0
SCHEME	05

Safety Input UR3



Botão de Emergência

Esquema de ligação botão de emergência externo

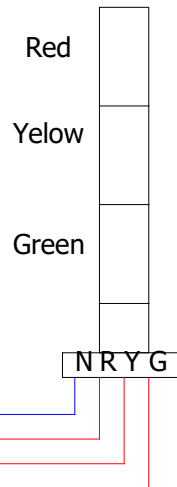
CONTRACT:	LOCATION: L1	Painel elétrico principal	REV. 0 07/07/2018		Vitor	CHANGES	REVISION
			Dados do usuário 1 Vitor de Sousa	Dados do usuário 2 vitorm.r.s@live.com.pt	0		
							SCHEME
							06

### Safety Output UR3

0 V	
D O 0	
0 V	NC
D O 1	NC
0 V	NC
D O 2	NC
0 V	NC
D O 3	NC

0 V	NC
D O 4	
0 V	NC
D O 5	NC
0 V	NC
D O 6	NC
0 v	NC
DO7	NC

### Semáforo



### Esquema de ligação Semáforo

CONTRACT:

LOCATION: L1

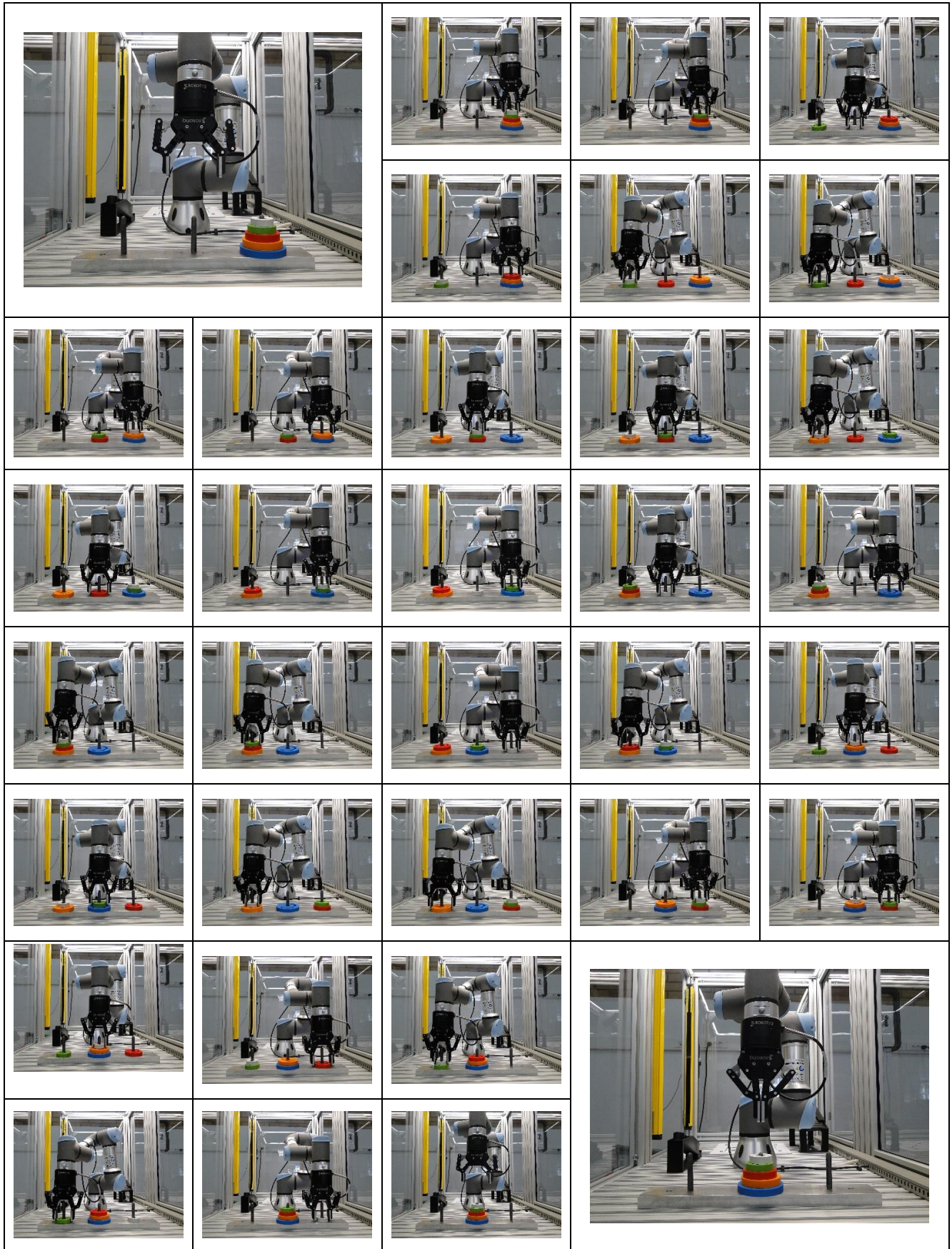
Painel elétrico principal

					REVISION
					0
0	07/07/2018	Vitor			SCHEME
REV.	DATE	NAME	CHANGES		07
Dados do usuário 1			Dados do usuário 2		
Vitor de Sousa			vitorm.r.s@live.com.pt		



## Apêndice B

### Foto-Descrição do Cenário de Tarefa Contínua



# Apêndice C

## Código de Programação Robô

```

1 Program
2 BeforeStart
3   rq_reset ()
4   rq_activate ()
5   rq_open_and_wait ()
6 Robot Program
7   Popup
8   a=1.5
9   v=5
10  close=0
11  estado=1
12  arruma=0
13  xadrez=0
14  tem_pecas=0
15  falta_pecas=0
16  tem_cabos=0
17  Loop
18    If xadrez==1
19      MoveJ
20        repousohanoi
21        up_rampa
22        reconhece
23      Loop falta_pecas==0
24        falta_pecas=1
25        Camera Locate
26          write_port_register(136,98)
27          falta_pecas=0
28          MoveJ
29            ap_stock_2
30            MoveL
31              stock_2
32              Gripper Close (1)
33              ap_stock_2
34            If rq_is_object_detected()== True
35              MoveL
36                aprox_peca_x
37                peca_x
38              Gripper Open (1)
39              aprox_peca_x
40            MoveJ
41              reconhece
42            Else
43              falta_pecas=1
44              write_port_register(136,101)
45              Wait MODBUS_2==4
46              write_port_register(136,0)
47              Gripper Open (1)
48            write_port_register(136,100)
49            Wait MODBUS_2==0
50            write_port_register(136,0)
51            MoveJ
52              up_rampa
53              repousohanoi
54            xadrez=0
55            falta_pecas=0
56          If arruma==1
57            MoveJ
58              repousohanoi
59              up_rampa
60              reconhece
61            Loop tem_pecas==0
62              tem_pecas=1
63              Camera Locate
64                write_port_register(136,98)
65                tem_pecas=0
66              MoveL
67                aprox_peca
68                peca
69              Gripper Close (1)
70              aprox_peca
71            MoveJ
72              ap_stock

```

```

73         MoveL
74             stock
75             Gripper Open (1)
76             ap_stock
77             reconhece
78 tem_pecas=0
79 Loop tem_cabos==0
80     tem_cabos=1
81     Camera Locate
82         write_port_register (136,98)
83         tem_cabos=0
84     MoveL
85         aprox_peca_c
86         peca_c
87         Gripper Close (1)
88         aprox_peca_c
89     MoveJ
90         ap_stock_cabo
91     MoveL
92         stock_cabo
93         Gripper Open (1)
94         ap_stock_cabo
95         reconhece
96 write_port_register (136,100)
97 Wait MODBUS_2==0
98 write_port_register (136,0)
99 MoveJ
100     up_rampa
101     repousohanoi
102     arruma=0
103     tem_cabos=0
104 Switch estado
105     Case 0
106         Wait: 0
107         Call SubProgram_1
108         rq_open_and_wait ()
109         MoveL
110             repousohanoi
111             movel (pose_add (P11, pr) , a=a, v=v, t=0, r=0)
112             P11
113             Popup
114             movel (pose_add (P11, pr) , a=a, v=v, t=0, r=0)
115             movel (pose_add (p21, pr) , a=a, v=v, t=0, r=0)
116             p21
117             Popup
118             movel (pose_add (p21, pr) , a=a, v=v, t=0, r=0)
119             movel (pose_add (P31, pr) , a=a, v=v, t=0, r=0)
120             P31
121             Popup
122             movel (pose_add (P31, pr) , a=a, v=v, t=0, r=0)
123             repousohanoi
124             rq_open_and_wait ()
125         estado=1
126     Case 1
127         MoveL
128             repousohanoi
129             movel (pose_add (P11, pr) , a=a, v=v, t=0, r=0)
130             movel (pose_add (P11, p5) , a=a, v=v, t=0, r=0)
131             Call SubProgram_1
132             movel (pose_add (P11, pr) , a=a, v=v, t=0, r=0)
133             movel (pose_add (p21, pr) , a=a, v=v, t=0, r=0)
134             movel (pose_add (p21, p2) , a=a, v=v, t=0, r=0)
135             rq_open_and_wait ()
136             movel (pose_add (p21, pr) , a=a, v=v, t=0, r=0)
137         estado=2
138     Case 2
139         movel (pose_add (P11, pr) , a=a, v=v, t=0, r=0)
140         movel (pose_add (P11, p4) , a=a, v=v, t=0, r=0)
141         Call SubProgram_1
142         movel (pose_add (P11, pr) , a=a, v=v, t=0, r=0)
143         movel (pose_add (P31, pr) , a=a, v=v, t=0, r=0)
144         movel (pose_add (P31, p2) , a=a, v=v, t=0, r=0)

```

```

145     rq_open_and_wait ()
146     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
147     estado=3
148 Case 3
149     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
150     movel (pose_add (p21,p1) , a=a, v=v, t=0, r=0)
151     Call SubProgram_1
152     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
153     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
154     movel (pose_add (P31,p3) , a=a, v=v, t=0, r=0)
155     rq_open_and_wait ()
156     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
157     estado=4
158 Case 4
159     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
160     movel (pose_add (P11,p3) , a=a, v=v, t=0, r=0)
161     Call SubProgram_1
162     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
163     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
164     movel (pose_add (p21,p2) , a=a, v=v, t=0, r=0)
165     rq_open_and_wait ()
166     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
167     estado=5
168 Case 5
169     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
170     movel (pose_add (P31,p2) , a=a, v=v, t=0, r=0)
171     Call SubProgram_1
172     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
173     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
174     movel (pose_add (P11,p4) , a=a, v=v, t=0, r=0)
175     rq_open_and_wait ()
176     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
177     estado=6
178 Case 6
179     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
180     movel (pose_add (P31,p1) , a=a, v=v, t=0, r=0)
181     Call SubProgram_1
182     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
183     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
184     movel (pose_add (p21,p3) , a=a, v=v, t=0, r=0)
185     rq_open_and_wait ()
186     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
187     estado=7
188 Case 7
189     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
190     movel (pose_add (P11,p3) , a=a, v=v, t=0, r=0)
191     Call SubProgram_1
192     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
193     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
194     movel (pose_add (p21,p4) , a=a, v=v, t=0, r=0)
195     rq_open_and_wait ()
196     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
197     estado=8
198 Case 8
199     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
200     movel (pose_add (P11,p2) , a=a, v=v, t=0, r=0)
201     Call SubProgram_1
202     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
203     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
204     movel (pose_add (P31,p2) , a=a, v=v, t=0, r=0)
205     rq_open_and_wait ()
206     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
207     estado=9
208 Case 9
209     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
210     movel (pose_add (p21,p3) , a=a, v=v, t=0, r=0)
211     Call SubProgram_1
212     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
213     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
214     movel (pose_add (P31,p3) , a=a, v=v, t=0, r=0)
215     rq_open_and_wait ()
216     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)

```

```

217         estado=10
218     Case 10
219         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
220         movel (pose_add (p21,p2) , a=a, v=v, t=0, r=0)
221         Call SubProgram_1
222         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
223         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
224         movel (pose_add (P11,p3) , a=a, v=v, t=0, r=0)
225         rq_open_and_wait ()
226         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
227         estado=11
228     Case 11
229         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
230         movel (pose_add (P31,p2) , a=a, v=v, t=0, r=0)
231         Call SubProgram_1
232         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
233         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
234         movel (pose_add (P11,p4) , a=a, v=v, t=0, r=0)
235         rq_open_and_wait ()
236         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
237         estado=12
238     Case 12
239         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
240         movel (pose_add (p21,p1) , a=a, v=v, t=0, r=0)
241         Call SubProgram_1
242         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
243         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
244         movel (pose_add (P31,p3) , a=a, v=v, t=0, r=0)
245         rq_open_and_wait ()
246         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
247         estado=13
248     Case 13
249         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
250         movel (pose_add (P11,p3) , a=a, v=v, t=0, r=0)
251         Call SubProgram_1
252         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
253         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
254         movel (pose_add (p21,p2) , a=a, v=v, t=0, r=0)
255         rq_open_and_wait ()
256         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
257         estado=14
258     Case 14
259         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
260         movel (pose_add (P11,p2) , a=a, v=v, t=0, r=0)
261         Call SubProgram_1
262         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
263         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
264         movel (pose_add (P31,p4) , a=a, v=v, t=0, r=0)
265         rq_open_and_wait ()
266         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
267         estado=15
268     Case 15
269         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
270         movel (pose_add (p21,p1) , a=a, v=v, t=0, r=0)
271         Call SubProgram_1
272         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
273         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
274         movel (pose_add (P31,p5) , a=a, v=v, t=0, r=0)
275         rq_open_and_wait ()
276         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
277         estado=16
278     Case 16
279         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
280         movel (pose_add (P11,p1) , a=a, v=v, t=0, r=0)
281         rq_move_and_wait (100)
282         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
283         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
284         movel (pose_add (p21,p2) , a=a, v=v, t=0, r=0)
285         rq_open_and_wait ()
286         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
287         estado=17
288     Case 17

```

```

289     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
290     movel (pose_add (P31,p4) , a=a, v=v, t=0, r=0)
291     Call SubProgram_1
292     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
293     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
294     movel (pose_add (P11,p2) , a=a, v=v, t=0, r=0)
295     rq_open_and_wait ()
296     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
297     estado=18
298     Case 18
299     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
300     movel (pose_add (P31,p3) , a=a, v=v, t=0, r=0)
301     Call SubProgram_1
302     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
303     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
304     movel (pose_add (p21,p3) , a=a, v=v, t=0, r=0)
305     rq_open_and_wait ()
306     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
307     estado=19
308     Case 19
309     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
310     movel (pose_add (P11,p1) , a=a, v=v, t=0, r=0)
311     Call SubProgram_1
312     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
313     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
314     movel (pose_add (p21,p4) , a=a, v=v, t=0, r=0)
315     rq_open_and_wait ()
316     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
317     estado=20
318     Case 20
319     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
320     movel (pose_add (P31,p2) , a=a, v=v, t=0, r=0)
321     Call SubProgram_1
322     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
323     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
324     movel (pose_add (P11,p2) , a=a, v=v, t=0, r=0)
325     rq_open_and_wait ()
326     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
327     estado=21
328     Case 21
329     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
330     movel (pose_add (p21,p3) , a=a, v=v, t=0, r=0)
331     Call SubProgram_1
332     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
333     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
334     movel (pose_add (P31,p3) , a=a, v=v, t=0, r=0)
335     rq_open_and_wait ()
336     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
337     estado=22
338     Case 22
339     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
340     movel (pose_add (p21,p2) , a=a, v=v, t=0, r=0)
341     Call SubProgram_1
342     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
343     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
344     movel (pose_add (P11,p3) , a=a, v=v, t=0, r=0)
345     rq_open_and_wait ()
346     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
347     estado=23
348     Case 23
349     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
350     movel (pose_add (P31,p2) , a=a, v=v, t=0, r=0)
351     Call SubProgram_1
352     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
353     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
354     movel (pose_add (P11,p4) , a=a, v=v, t=0, r=0)
355     rq_open_and_wait ()
356     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
357     estado=24
358     Case 24
359     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
360     movel (pose_add (P31,p1) , a=a, v=v, t=0, r=0)

```

```

361         Call SubProgram_1
362         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
363         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
364         movel (pose_add (p21,p3) , a=a, v=v, t=0, r=0)
365         rq_open_and_wait ()
366         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
367         estado=25
368     Case 25
369         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
370         movel (pose_add (P11,p3) , a=a, v=v, t=0, r=0)
371         Call SubProgram_1
372         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
373         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
374         movel (pose_add (p21,p4) , a=a, v=v, t=0, r=0)
375         rq_open_and_wait ()
376         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
377         estado=26
378     Case 26
379         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
380         movel (pose_add (P11,p2) , a=a, v=v, t=0, r=0)
381         Call SubProgram_1
382         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
383         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
384         movel (pose_add (P31,p2) , a=a, v=v, t=0, r=0)
385         rq_open_and_wait ()
386         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
387         estado=27
388     Case 27
389         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
390         movel (pose_add (p21,p3) , a=a, v=v, t=0, r=0)
391         Call SubProgram_1
392         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
393         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
394         movel (pose_add (P31,p3) , a=a, v=v, t=0, r=0)
395         rq_open_and_wait ()
396         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
397         estado=28
398     Case 28
399         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
400         movel (pose_add (P11,p1) , a=a, v=v, t=0, r=0)
401         Call SubProgram_1
402         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
403         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
404         movel (pose_add (p21,p4) , a=a, v=v, t=0, r=0)
405         rq_open_and_wait ()
406         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
407         estado=29
408     Case 29
409         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
410         movel (pose_add (P31,p2) , a=a, v=v, t=0, r=0)
411         Call SubProgram_1
412         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
413         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
414         movel (pose_add (P11,p2) , a=a, v=v, t=0, r=0)
415         rq_open_and_wait ()
416         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
417         estado=30
418     Case 30
419         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
420         movel (pose_add (P31,p1) , a=a, v=v, t=0, r=0)
421         Call SubProgram_1
422         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
423         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
424         movel (pose_add (p21,p5) , a=a, v=v, t=0, r=0)
425         rq_open_and_wait ()
426         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
427         estado=31
428     Case 31
429         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
430         movel (pose_add (P11,p1) , a=a, v=v, t=0, r=0)
431         Call SubProgram_1
432         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)

```

```

433         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
434         movel (pose_add (p21,p6) , a=a, v=v, t=0, r=0)
435         rq_open_and_wait ()
436         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
437         estado=32
438     Case 32
439         MoveJ
440             repousohanoi
441             Gripper Close (1)
442             Gripper Open (1)
443             Gripper Close (1)
444             Gripper Open (1)
445         estado=33
446     Case 33
447         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
448         movel (pose_add (p21,p5) , a=a, v=v, t=0, r=0)
449         Call SubProgram_1
450         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
451         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
452         movel (pose_add (P11,p2) , a=a, v=v, t=0, r=0)
453         rq_open_and_wait ()
454         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
455         estado=34
456     Case 34
457         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
458         movel (pose_add (p21,p4) , a=a, v=v, t=0, r=0)
459         Call SubProgram_1
460         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
461         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
462         movel (pose_add (P31,p2) , a=a, v=v, t=0, r=0)
463         rq_open_and_wait ()
464         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
465         estado=35
466     Case 35
467         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
468         movel (pose_add (P11,p1) , a=a, v=v, t=0, r=0)
469         Call SubProgram_1
470         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
471         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
472         movel (pose_add (P31,p3) , a=a, v=v, t=0, r=0)
473         rq_open_and_wait ()
474         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
475         estado=36
476     Case 36
477         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
478         movel (pose_add (p21,p3) , a=a, v=v, t=0, r=0)
479         Call SubProgram_1
480         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
481         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
482         movel (pose_add (P11,p2) , a=a, v=v, t=0, r=0)
483         rq_open_and_wait ()
484         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
485         estado=37
486     Case 37
487         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
488         movel (pose_add (P31,p2) , a=a, v=v, t=0, r=0)
489         Call SubProgram_1
490         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
491         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
492         movel (pose_add (p21,p4) , a=a, v=v, t=0, r=0)
493         rq_open_and_wait ()
494         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
495         estado=38
496     Case 38
497         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
498         movel (pose_add (P31,p1) , a=a, v=v, t=0, r=0)
499         Call SubProgram_1
500         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
501         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
502         movel (pose_add (P11,p3) , a=a, v=v, t=0, r=0)
503         rq_open_and_wait ()
504         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)

```

```

505         estado=39
506     Case 39
507         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
508         movel (pose_add (p21,p3) , a=a, v=v, t=0, r=0)
509         Call SubProgram_1
510         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
511         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
512         movel (pose_add (P11,p4) , a=a, v=v, t=0, r=0)
513         rq_open_and_wait ()
514         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
515         estado=40
516     Case 40
517         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
518         movel (pose_add (p21,p2) , a=a, v=v, t=0, r=0)
519         Call SubProgram_1
520         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
521         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
522         movel (pose_add (P31,p2) , a=a, v=v, t=0, r=0)
523         rq_open_and_wait ()
524         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
525         estado=41
526     Case 41
527         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
528         movel (pose_add (P11,p3) , a=a, v=v, t=0, r=0)
529         Call SubProgram_1
530         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
531         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
532         movel (pose_add (P31,p3) , a=a, v=v, t=0, r=0)
533         rq_open_and_wait ()
534         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
535         estado=42
536     Case 42
537         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
538         movel (pose_add (P11,p2) , a=a, v=v, t=0, r=0)
539         Call SubProgram_1
540         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
541         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
542         movel (pose_add (p21,p3) , a=a, v=v, t=0, r=0)
543         rq_open_and_wait ()
544         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
545         estado=43
546     Case 43
547         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
548         movel (pose_add (P31,p2) , a=a, v=v, t=0, r=0)
549         Call SubProgram_1
550         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
551         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
552         movel (pose_add (p21,p4) , a=a, v=v, t=0, r=0)
553         rq_open_and_wait ()
554         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
555         estado=44
556     Case 44
557         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
558         movel (pose_add (P11,p1) , a=a, v=v, t=0, r=0)
559         Call SubProgram_1
560         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
561         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
562         movel (pose_add (P31,p3) , a=a, v=v, t=0, r=0)
563         rq_open_and_wait ()
564         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
565         estado=45
566     Case 45
567         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
568         movel (pose_add (p21,p3) , a=a, v=v, t=0, r=0)
569         Call SubProgram_1
570         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
571         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
572         movel (pose_add (P11,p2) , a=a, v=v, t=0, r=0)
573         rq_open_and_wait ()
574         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
575         estado=46
576     Case 46

```

```

577     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
578     movel (pose_add (p21,p2) , a=a, v=v, t=0, r=0)
579     Call SubProgram_1
580     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
581     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
582     movel (pose_add (P31,p4) , a=a, v=v, t=0, r=0)
583     rq_open_and_wait ()
584     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
585     estado=47
586 Case 47
587     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
588     movel (pose_add (P11,p1) , a=a, v=v, t=0, r=0)
589     Call SubProgram_1
590     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
591     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
592     movel (pose_add (P31,p5) , a=a, v=v, t=0, r=0)
593     rq_open_and_wait ()
594     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
595     estado=48
596 Case 48
597     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
598     movel (pose_add (p21,p1) , a=a, v=v, t=0, r=0)
599     rq_move_and_wait (100)
600     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
601     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
602     movel (pose_add (P11,p2) , a=a, v=v, t=0, r=0)
603     rq_open_and_wait ()
604     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
605     estado=49
606 Case 49
607     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
608     movel (pose_add (P31,p4) , a=a, v=v, t=0, r=0)
609     Call SubProgram_1
610     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
611     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
612     movel (pose_add (p21,p2) , a=a, v=v, t=0, r=0)
613     rq_open_and_wait ()
614     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
615     estado=50
616 Case 50
617     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
618     movel (pose_add (P31,p3) , a=a, v=v, t=0, r=0)
619     Call SubProgram_1
620     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
621     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
622     movel (pose_add (P11,p3) , a=a, v=v, t=0, r=0)
623     rq_open_and_wait ()
624     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
625     estado=51
626 Case 51
627     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
628     movel (pose_add (p21,p1) , a=a, v=v, t=0, r=0)
629     Call SubProgram_1
630     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
631     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
632     movel (pose_add (P11,p4) , a=a, v=v, t=0, r=0)
633     rq_open_and_wait ()
634     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
635     estado=52
636 Case 52
637     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
638     movel (pose_add (P31,p2) , a=a, v=v, t=0, r=0)
639     Call SubProgram_1
640     movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
641     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
642     movel (pose_add (p21,p2) , a=a, v=v, t=0, r=0)
643     rq_open_and_wait ()
644     movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
645     estado=53
646 Case 53
647     movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
648     movel (pose_add (P11,p3) , a=a, v=v, t=0, r=0)

```

```

649         Call SubProgram_1
650         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
651         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
652         movel (pose_add (P31,p3) , a=a, v=v, t=0, r=0)
653         rq_open_and_wait ()
654         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
655         estado=54
656     Case 54
657         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
658         movel (pose_add (P11,p2) , a=a, v=v, t=0, r=0)
659         Call SubProgram_1
660         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
661         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
662         movel (pose_add (p21,p3) , a=a, v=v, t=0, r=0)
663         rq_open_and_wait ()
664         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
665         estado=55
666     Case 55
667         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
668         movel (pose_add (P31,p2) , a=a, v=v, t=0, r=0)
669         Call SubProgram_1
670         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
671         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
672         movel (pose_add (p21,p4) , a=a, v=v, t=0, r=0)
673         rq_open_and_wait ()
674         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
675         estado=56
676     Case 56
677         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
678         movel (pose_add (P31,p1) , a=a, v=v, t=0, r=0)
679         Call SubProgram_1
680         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
681         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
682         movel (pose_add (P11,p3) , a=a, v=v, t=0, r=0)
683         rq_open_and_wait ()
684         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
685         estado=57
686     Case 57
687         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
688         movel (pose_add (p21,p3) , a=a, v=v, t=0, r=0)
689         Call SubProgram_1
690         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
691         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
692         movel (pose_add (P11,p4) , a=a, v=v, t=0, r=0)
693         rq_open_and_wait ()
694         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
695         estado=58
696     Case 58
697         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
698         movel (pose_add (p21,p2) , a=a, v=v, t=0, r=0)
699         Call SubProgram_1
700         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
701         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
702         movel (pose_add (P31,p2) , a=a, v=v, t=0, r=0)
703         rq_open_and_wait ()
704         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
705         estado=59
706     Case 59
707         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
708         movel (pose_add (P11,p3) , a=a, v=v, t=0, r=0)
709         Call SubProgram_1
710         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
711         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
712         movel (pose_add (P31,p3) , a=a, v=v, t=0, r=0)
713         rq_open_and_wait ()
714         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
715         estado=60
716     Case 60
717         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
718         movel (pose_add (p21,p1) , a=a, v=v, t=0, r=0)
719         Call SubProgram_1
720         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)

```

```

721         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
722         movel (pose_add (P11,p4) , a=a, v=v, t=0, r=0)
723         rq_open_and_wait ()
724         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
725         estado=61
726     Case 61
727         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
728         movel (pose_add (P31,p2) , a=a, v=v, t=0, r=0)
729         Call SubProgram_1
730         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
731         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
732         movel (pose_add (p21,p2) , a=a, v=v, t=0, r=0)
733         rq_open_and_wait ()
734         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
735         estado=62
736     Case 62
737         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
738         movel (pose_add (P31,p1) , a=a, v=v, t=0, r=0)
739         Call SubProgram_1
740         movel (pose_add (P31,pr) , a=a, v=v, t=0, r=0)
741         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
742         movel (pose_add (P11,p5) , a=a, v=v, t=0, r=0)
743         rq_open_and_wait ()
744         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
745         estado=63
746     Case 63
747         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
748         movel (pose_add (p21,p1) , a=a, v=v, t=0, r=0)
749         Call SubProgram_1
750         movel (pose_add (p21,pr) , a=a, v=v, t=0, r=0)
751         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
752         movel (pose_add (P11,p6) , a=a, v=v, t=0, r=0)
753         rq_open_and_wait ()
754         movel (pose_add (P11,pr) , a=a, v=v, t=0, r=0)
755         estado=64
756     Case 64
757         estado=1
758     SubProgram_1
759         pergarra=70
760         Loop rq_is_object_detected()== False and pergarra<220
761             posgarra=get_actual_tcp_pose ()
762             possub=pose_add (posgarra, subir)
763             rq_move (pergarra)
764             movel (pose_add (posgarra, subir) , a=1, v=0.25, t=0, r=0)
765             pergarra=pergarra+40
766     Event read_port_register (129)
767         If MODBUS_2==2
768             arruma=1
769             write_port_register (136, 99)
770         If MODBUS_2==1
771             xadrez=1
772             write_port_register (136, 99)
773     Event digital_in[0]== True
774         chamada=1
775

```

# Apêndice D

## Tutorial de Programação

## Mini Tutorial de Programação

### UR3, Câmara e Garra



Vitor de Sousa

[vitors@ipb.pt](mailto:vitors@ipb.pt)

Neste mini tutorial o utilizador é convidado a fazer um programa para operar com o conjunto robótico constituído por um robô colaborativo (Universal Robots) uma garra (ROBOTIQ 2-finger 85) e uma câmara (ROBOTIQ wrist camera), estando estes já integrados e devidamente instalados.

O programa será capaz de depois de o humano colocar peças cilíndricas numa mesa, detetar a sua posição e arrumar numa rampa de *stock* previamente instalada na célula robótica, isto tudo de forma automática.

Inicialmente é necessário verificar todos os parâmetros de segurança para não danificar equipamentos nem ferir os utilizadores. Posteriormente passa-se a uma breve explicação de programação dos equipamentos em epígrafe.

Este Tutorial está dividido em dois capítulos, o primeiro refere-se às configurações do sistema e o segundo aborda a programação.

Comece por ligar a consola de programação.

Depois de iniciar a consola aparece um popup onde é solicitada a inicialização do robô, deve seleccionar “Not Now” uma vez que antes de iniciar é necessário fazer algumas configurações cruciais ao funcionamento.

O primeiro passo é iniciar um novo programa vazio. Seleccionar “Program Robot” (Figura 1), inserir a *password* “1” e seleccionar “Empty program”.

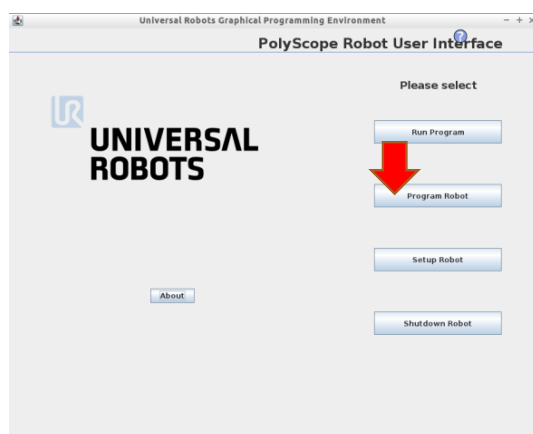


Figura 1-Começar um novo programa vazio.

**Nota:**atribua os nomes propostos para que no final possa comparar com o exemplo

# 1. Configurações

**Passo 1-** Definição de Tool Center Point (TCP). Depois de criado o programa vazio entrar no menu "Installation", selecionar "TCP configuration" e introduzir o incremento em z de 140 mm (Figura 2). Deve também ativar "Center of gravity" e colocar a massa do conjunto Griper Robotq Q85 + Wrist Camera e o respectivos valores do centro de massa (Tabela 1).

Combination	x (mm)	y (mm)	z (mm)	Mass (grams)
-	0	5	9	230
FT 300	0	2	30	530
2-Finger 85	0	1	58	975

Tabela 1-Dados do centro de massa.

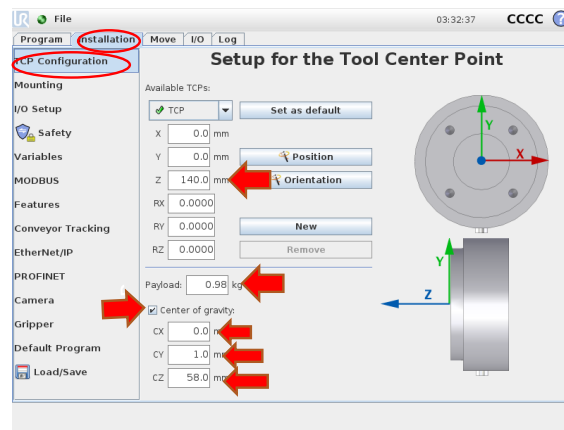


Figura 2--Painel de programação do TCP.

## Passo2- Salvar as configurações de Instalação.

Para guardar deve ir à aba "Installation" depois clicar em "Load/Save" e selecionar em "Save as..." (Figura 3).

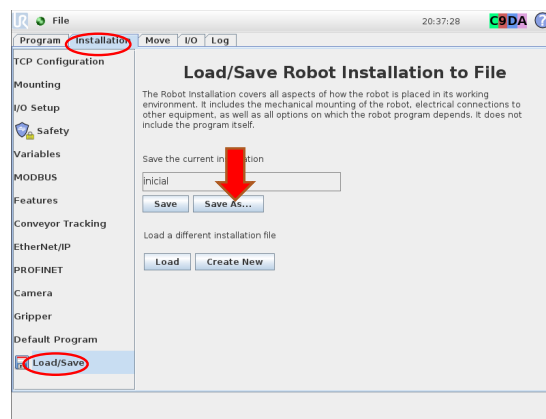


Figura 3-Salvar Instalação.

Agora deve alterar o nome para **“Tutorial”** e depois **“Save”**(Figura 4).

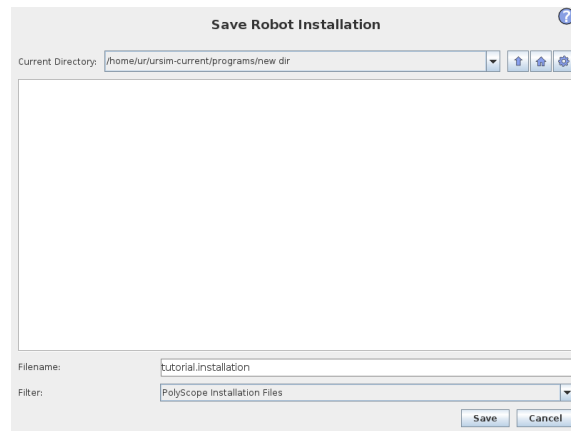


Figura 4- Guardar configurações.

Aparece um *popup* onde é solicitado que sejam aplicadas as alterações que introduziu. Deve clicar em **“Set installation”**.

### **Passo3-** Introdução de um ponto variável.

Depois de criado o programa vazio entrar no menu **“Installation”**, seleccionar **“Features”** e criar um ponto (Figura 5).

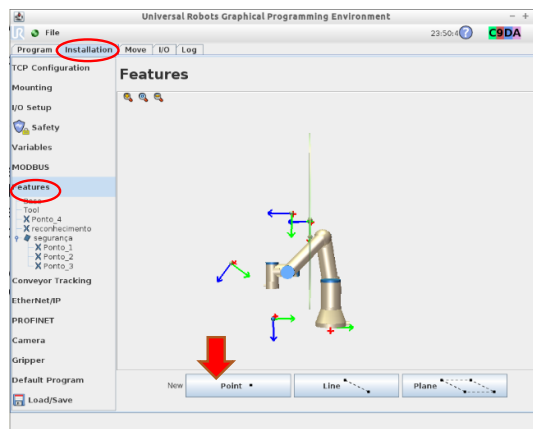


Figura 5-Definir novo **“Feature Point”**.

Seleccionar agora o ponto anteriormente introduzido com o nome **“Point\_1”** e colocar um nome passível de percepção (exemplo: **snapshotpos**) em **“Rename”**. Habilitar **“Variable”** e por fim **“Set Point”** (Figura 6).

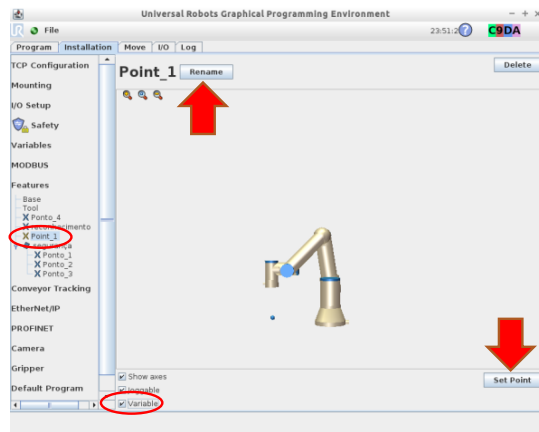


Figura 6-Tornar o ponto variável.

Aparecerá agora o menu de movimento do robô. Aconselha-se a aplicação de uma localização conforme a Tabela 2.

Ao tentar introduzir os valores aparece um *popup* onde é solicitado que inicialize o robô. Deve clicar em “Go to initialization screen”, depois deve clicar em “On” para ligar o robô e por fim em “Start” e verificar se o estado do robô ficou em Normal, depois disso no canto inferior direito clicar em “Ok”. *Verificar botões de emergência.*

Introduza as ângulos (Tabela 2) nas juntas respetivas (Figura 7) e depois “Ok”. De seguida será exibida uma tela para movimento desta posição acima inserida, deve então pressionar continuamente em “Auto” para que o robô vá para a posição definida de forma automática. Verificar trajetória se for bater em alguma parte da estrutura levar robô em manual e no final “ok”.

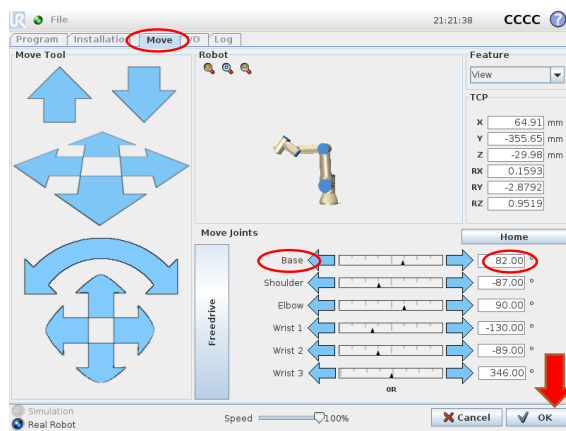


Figura 7-Procedimento para introdução de ponto variável.

Junta	Graus
Base	82
Shoulder	-87
Elbow	90
Wrist 1	-130
Wrist 2	-89
Wrist 3	346

Tabela 2- Ângulos para as 6 juntas do robô.

**Nota:** Se necessário reduzir a velocidade do movimento e se tiver algum obstáculo na sua trajetória deve ser selecionado o modo de movimento manual e assim que a trajetória esteja desimpedida pressionar novamente “Auto”.

**Passo4-** Calibração da Câmara. Para a calibração é necessário ter o quadro de calibração disponibilizado especificamente para UR3 (Figura 8).



Figura 8-Quadro de calibração para UR3.

No menu “Installation” escolher “Camera”, depois a aba de “Snapshot”. Aqui é selecionado o ponto variável **snapshotpos** salvo anteriormente no menu “Features” e pressionar “Define”.

Coloca-se o quadro de calibração o mais centrado possível com a imagem que aparece no menu e seleciona-se “Save position”.

O passo seguinte carece de atenção uma vez que o robô irá realizar movimentos automaticamente. Depois de garantir a segurança do sistema deve selecionar “Calibrate” e o robô irá fazer um *scan* do quadro através da aquisição de 27 imagens em posições e orientações diferentes. No final o *software* ainda corre o algoritmo de calibração em mais nove posições para aferir resultados.

É ainda solicitado ao utilizador que valide a identificação da grelha de calibração devendo para isso clicar em “Accept”(Figura 9). Caso a calibração não seja corretamente efetuada deverá clicar em “Recalibrate”.

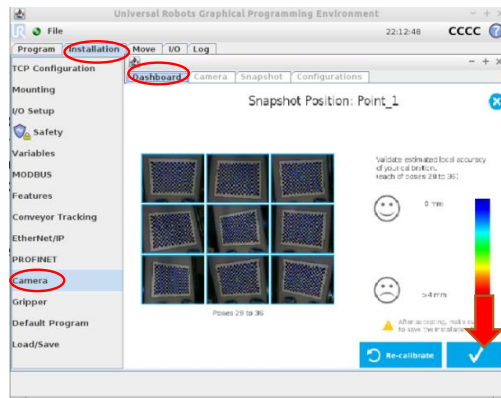


Figura 9-Calibração da câmara.

**Nota:** No caso de falha a meio da calibração é necessário centrar o quadro novamente e clicar em “Calibrate”. Para o bom funcionamento a iluminação abundante e constante é necessária.

**Passo 5-** Salvar as configurações de Instalação novamente. Consultar **passo 2** e fazer “Apply and restart”.

## 2-Programação.

Finalizado o processo de calibração parte-se agora para a construção do código exemplo para deteção automática de peça na mesa, assim sendo seleccione “Program”.

**Passo 1-** Introduzir a instrução para o robô se mover até a posição inicial de reconhecimento para isso seleccionar “<empty>” na trama do programa e depois introduzir a função “MoveJ” presente na aba “Structure/Basic” e clicar em “Move”(Figura 10).

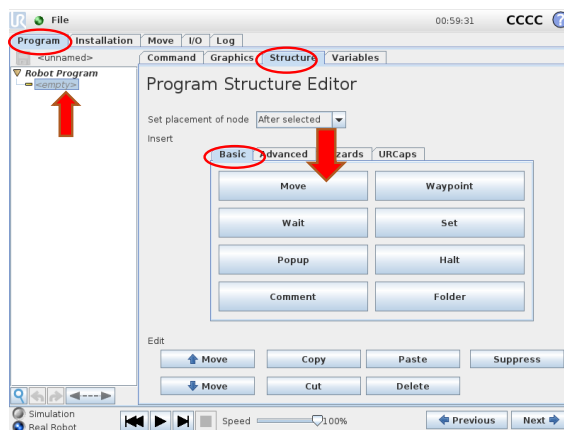


Figura 10- Introdução de MOVEJ.

**Passo 2-** Introduzir o ponto de reconhecimento. Na trama de programação aparece um “MoveJ” e imediatamente a seguir aparece um ponto de nome “waypoint” (Figura 11), deverá selecionar este ponto. Na aba “Comand” alterar o nome em “Rename” (exemplo: **reconhece\_pos**) e por fim selecionar “Set Waypoint” onde deve colocar novamente os valores dos ângulos presentes na Tabela 2.

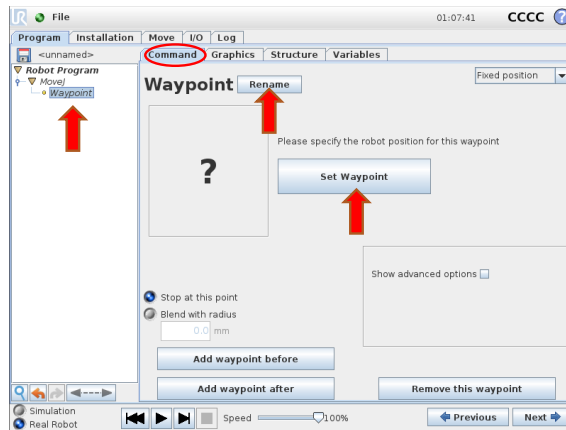


Figura 11-Gravar a posição do ponto de reconhecimento.

**Passo 3-** Introduzir a função “Camera Locate”. Selecionar o “MoveJ” do passo 1 e introduzir a função “Camera Locate” presente em “Structure/URCaps” (Figura 12), função esta responsável por identificar o objeto e retornar à posição que será treinada posteriormente.

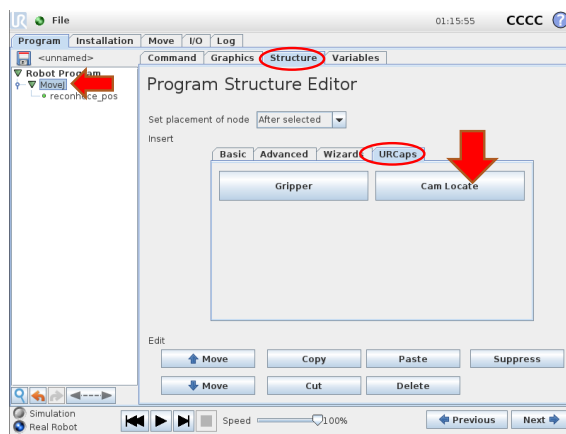


Figura 12-Introdução da função da câmara.

**Passo 4-** Treino do objeto a detetar. Deve seleccionar a função “Camera Locate” no programa. Abrir a aba “Command” e de seguida clicar em “Teach object”(Figura 13).

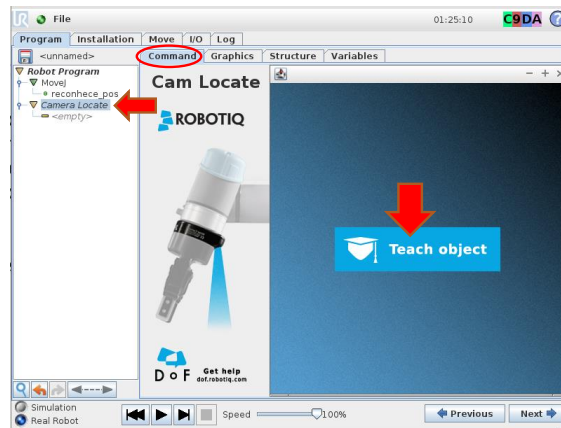


Figura 13-Início da calibração da função "Camera Locate".

De seguida é exibida a página onde se selecciona a calibração anteriormente realizada como “Snapshotpos”. Aparecem então as duas formas possíveis de treinar o objeto a forma automática e a paramétrica, seleccione “Parametric mode” (Figura 14) .

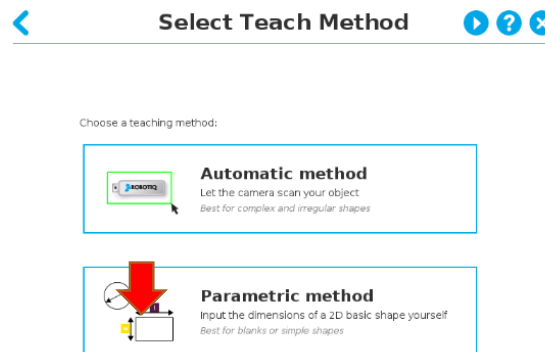


Figura 14- Definição do método de treino da peça.

A página seguinte é onde deve escolher o tipo de objeto e as suas dimensões, para o exemplo da peça a utilizar foi diâmetro (**D**) 30mm altura (**H**) 25mm, deve ser escolhido o círculo como tipo de objeto no final seleccionar “Define”(Figura 15).

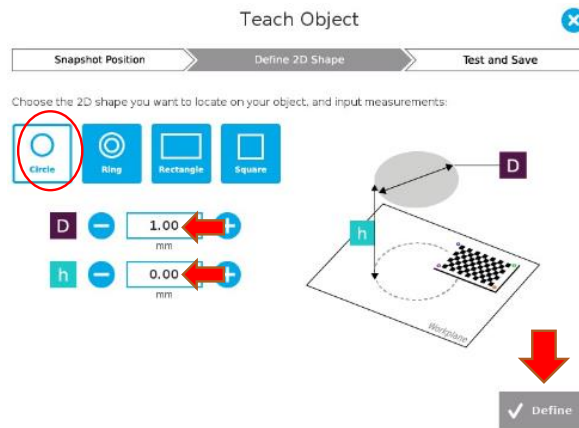


Figura 15-Introdução das medidas do objeto.

Deve retirar agora o quadro de calibração e utilizar o existente na célula.

Posteriormente à introdução das medidas o *software* irá correr um programa de deteção para isso é necessário colocar a peça cilíndrica o mais centrada possível na imagem e seleccionar o icon da câmara. Se a peça for devidamente detetada deve prosseguir para a definição da posição no icon de coordenadas (3º icon vertical) e no final salvar .

**Passo 5-** Introduzir “MoveL” agora em relação à função da câmara. Selecione “<empty>” depois de “Camera Locate” e introduza a função “Move” presente na aba “Structure/Basic” e seleccionar “Move” (consultar passo 1) sendo aqui introduzida a posição de pegar na peça calculada pela função da câmara.

Observe que está definido por defeito “MoveJ” sendo então necessário alterar para um movimento linear. Para isso deve seleccionar o “MoveJ” depois na aba “Command” do lado direito (Figura 16) alterar para MoveL. Posteriormente deve também ser alterado o ponto de referência seleccionado em “Feature” o ponto “snapshotpos\_var”. Neste menu pode também ser alterada a aceleração e velocidade do movimento, mas para a aplicação os valores padrão são mantidos.

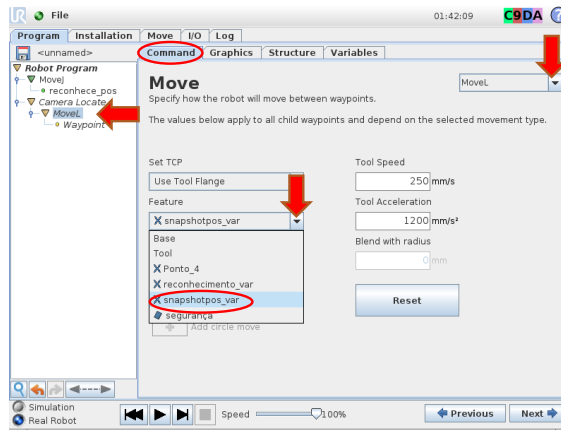


Figura 16-Move L em função do ponto variável.

**Passo 6-** Depois de adicionado o “MoveL” verifica-se que já foi criado o ponto “Waypoint”, deve então selecioná-lo ir a aba “Comand” dar um nome a esse ponto (exemplo:”**app\_targetpos**”) e de seguida “Set Waypoint” (Figura 17).

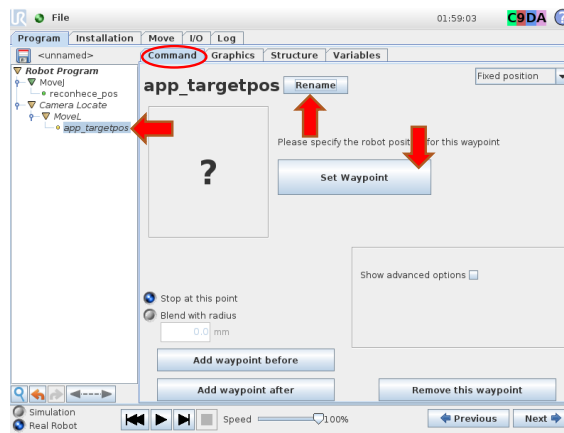


Figura 17-Introdução do ponto de aproximação.

No menu seguinte (Figura 18) deve clicar em “Freedryve” continuamente ou então no botão presente na retaguarda da consola de programação, levar o robô com a mão para uma posição cerca de 10 cm acima da peça, já com a orientação pretendida para depois pegar na peça, e no final clicar em “Ok”.

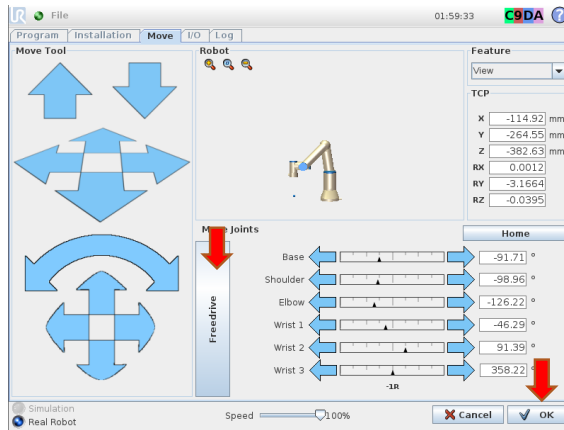


Figura 18-Gravar posição em modo "Freedrive"

Depois de definido o ponto de aproximação falta agora definir o ponto de pegar na peça. Este deve ser adicionado clicando em "Add waypoint after" ( Figura 19).

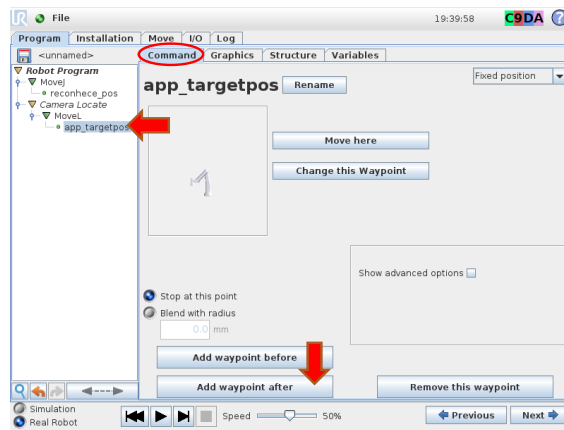


Figura 19- Adicionar novo ponto.

Adicionado o ponto deve também ser alterado o seu nome (exemplo: "target\_pos") e para definir a posição e orientação deve ser o mesmo que na (Figura 18) mas desta vez na posição para pegar na peça. Esta posição deve ficar centrada com a peça e com a distancia da garra à mesa equivalente à altura da peça uma vez que o funcionamento da garra assim o exige.

**Nota:** A garra deve estar totalmente aberta.

Falta finalmente colocar novamente o ponto de aproximação para que o robô depois de pegar na peça volte a subir. Para tal seleccione o ponto anteriormente definido como “app\_targetpos” vá à aba “Structure” seleccione “Copy” (Figura 20).

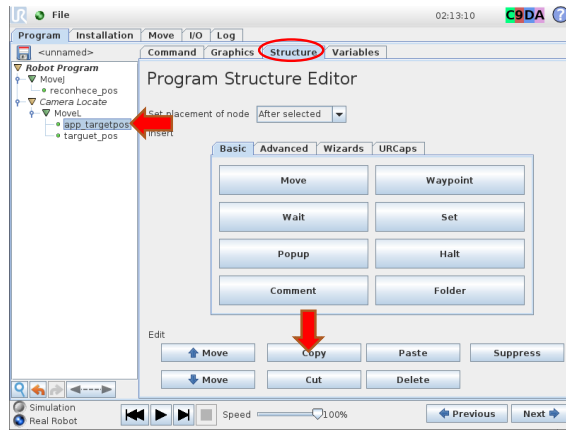


Figura 20-Copiar um ponto.

Depois volte ao programa seleccione o ponto “target\_pos” e clique em “Paste”(Figura 21), ficando assim o Move completo.

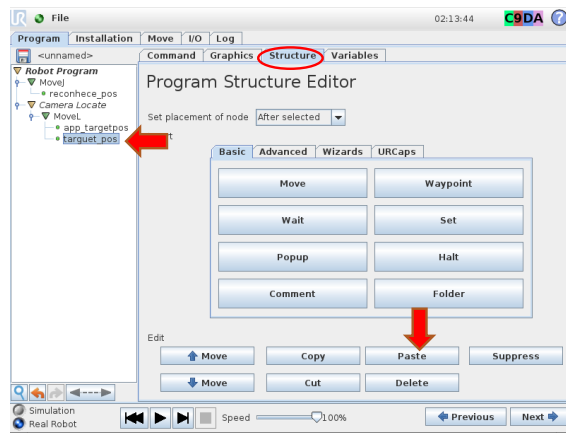


Figura 21-Colar o ponto.

**Nota:** Caso a peça seja movida neste passo entre a definição das posições, basta seleccionar a função “Camera locate” depois “Test/Modify” (Figura 22) e repetir o procedimento do passo 4.

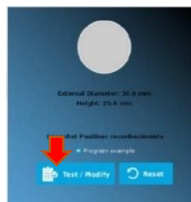


Figura 22- Modificar posição de deteção

**Passo 7-** Introdução da função da garra. Deve seleccionar novamente no programa “target\_pos” e de seguida ir a “Structure/URCaps” e seleccionar “Gripper” (Figura 23).

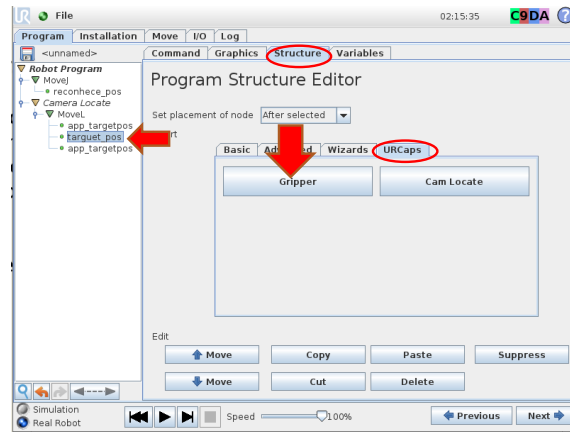


Figura 23-Adicionar a função da garra.

**Passo 8-** Configuração da garra. Para tal seleccione “Gripper Move” na trama do programa depois vá para a aba de “Command” e clique em “Edit action” (Figura 24).

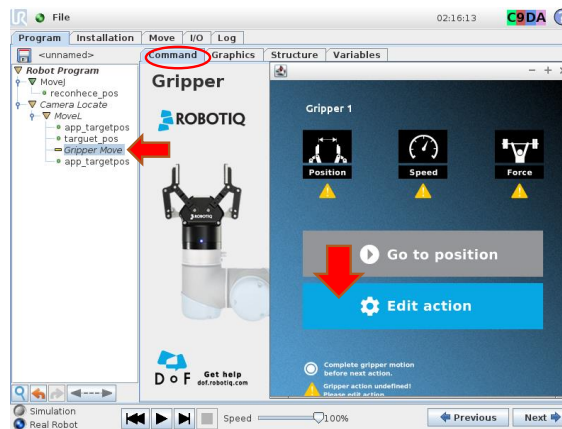


Figura 24- Função da garra.

Aparece agora uma página onde é possível definir parâmetros para abertura ou fecho da garra. Os parâmetros são a força exercida sobre a peça, velocidade de fecho ou abertura e a percentagem de fecho deixando estes valores na definição padrão. Neste passo é apenas solicitado que seleccione “Close” e no final basta salvar a operação em “Save” (Figura 25).

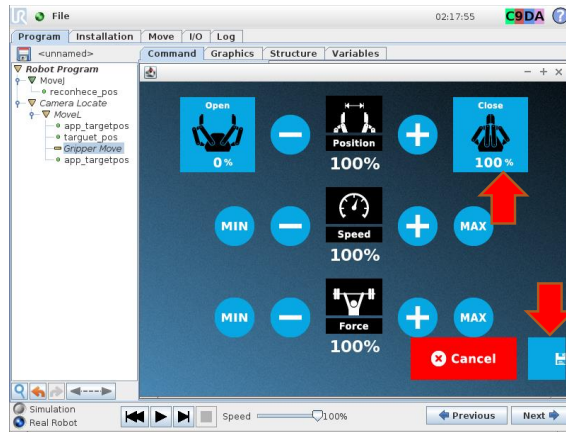


Figura 25-Parametrização da função da garra.

**Passo 9-**Introdução dos movimentos e posições para arrumar as peças no *storage*.  
 Selecione o “Move L”(Figura 26). Na aba “Structure” introduza um “MoveJ” tal como no **passo 1**.

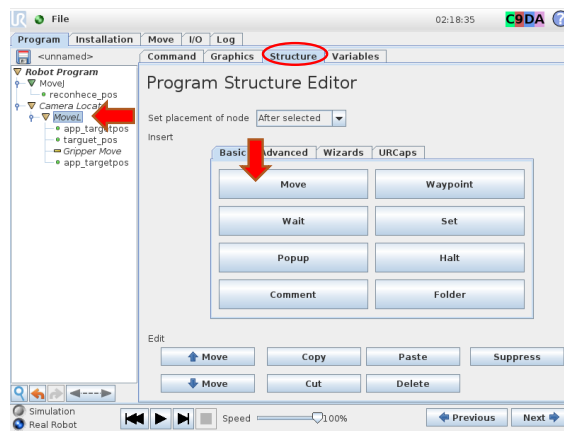


Figura 26-Move J do movimento da peça para o *storage*.

O ponto a adicionar agora será o ponto de aproximação à rampa de stock existente (Figura 27), recomenda-se a definição do ponto cerca de 10 cm acima da localização de descarga sendo sugerido como nome “**app\_stock**” para este ponto. No caso de duvida consultar **passo 6**.

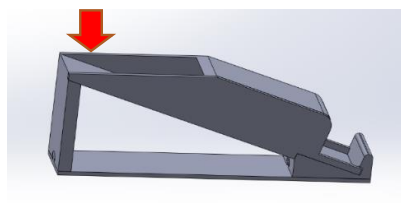


Figura 27-Local de descarga das peças.

**Passo 10-** Introduzir “MoveL” para descarga da peça no *storage*. Deve seleccionar o “app\_stock” ir até “Structure” e introduzir novamente um “Move” (Figura 28).

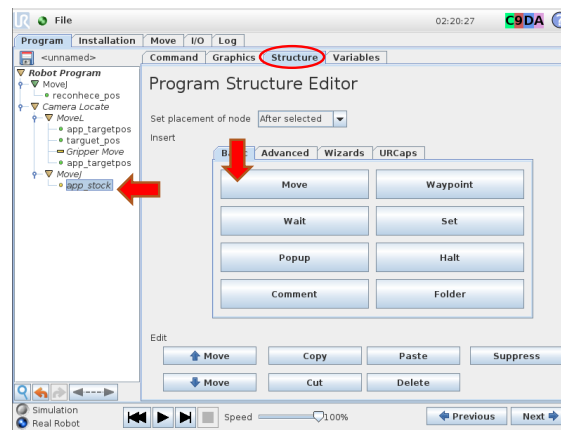


Figura 28-Move L para descarga de peças.

Deve tornar o “Move” anteriormente adicionado num “MoveL”(Figura 29).

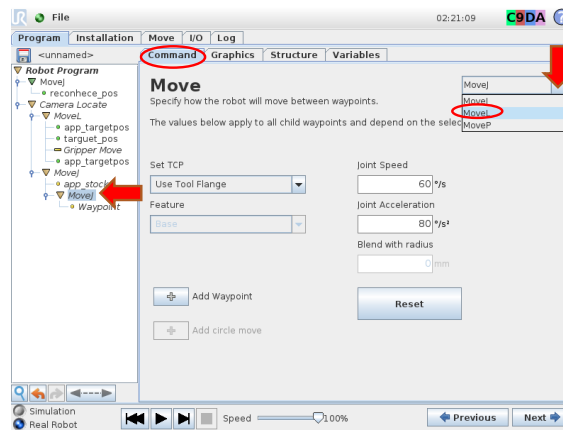


Figura 29-Converter "MoveJ" em "MoveL".

Falta então definir o ponto de descarga na posição apropriada com o nome sugerido de “stock”. Deve ainda copiar o ponto de aproximação do *storage* para a posição seguinte. No caso de dúvidas consultar o **passo 6**.

**Passo 11-** Introdução da função da garra para largar a peça. Deve seleccionar o ponto de “stock” ir a “Structure/URcaps” seleccionar “Griper” (Figura 30).

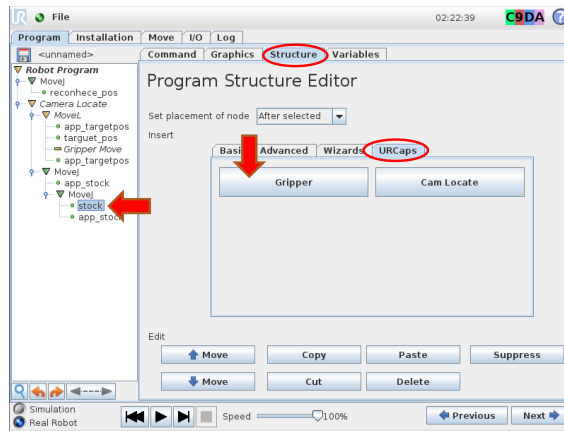


Figura 30-Introdução da Garra para descarga de peça.

Agora deve repetir o **passo 8** mas com a seleção de “Open” e no final “Save” (Figura 31).

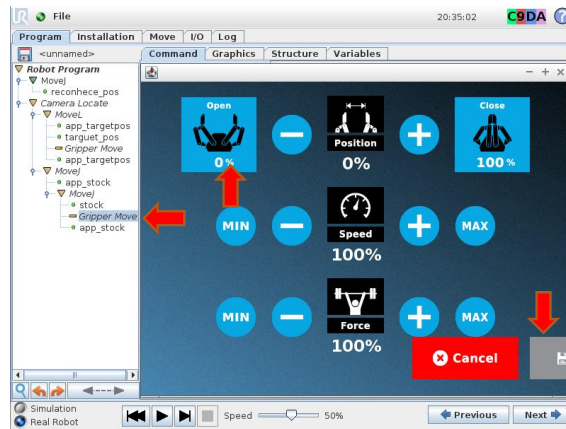


Figura 31-Parameterização para abertura da garra.

O código está agora finalizado, deve ainda verificar se o programa por si realizado está idêntico ao da Figura 32 .Note que a função da câmara funciona como um “if” e por isso as linhas que estão imediatamente abaixo só são processadas se houver algum objeto detetado, assim sendo o robô permanecerá na posição “reconhece\_pos” até que detete alguma peça.

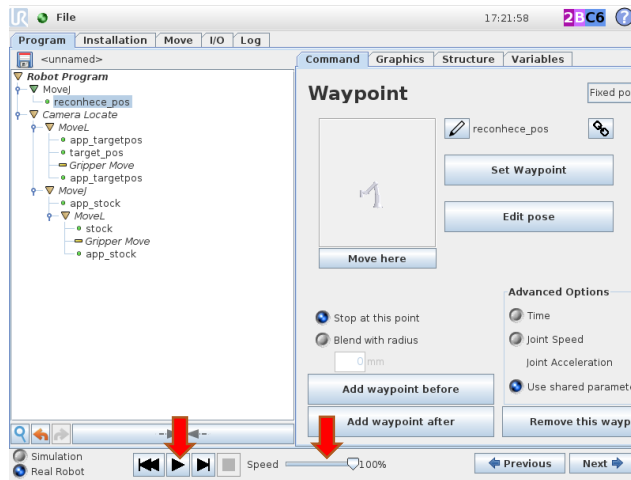


Figura 32-Aspecto final do programa.

Para a reprodução do programa deve garantir novamente toda a segurança e na primeira vez que executar o programa reduza a velocidade para menos de 50% para precaver qualquer erro de trajetórias e depois carregar em *play*.

Ira aparecer uma página onde é solicitado que mova o robô para a primeira posição do código, pode fazê-lo pressionando “Auto” continuamente ou então de forma manual movendo os seus eixos. Depois de colocado na posição original pressione novamente *play* e teste a colocação das peças em diversas localizações.

Para guardar o programa deve faze-lo selecionando “file” no canto superior esquerdo da consola depois clicar em “Save as” escolher um nome e uma localização e fazer “Save”.

# Apêndice E

## Poster do Trabalho Realizado

# Design and Assembly of Cell Flexible Assembly with Collaboration Man / Robot

Vitor Sousa, Jos é Barbosa, Paulo Leitão



## Motivation and Objectives

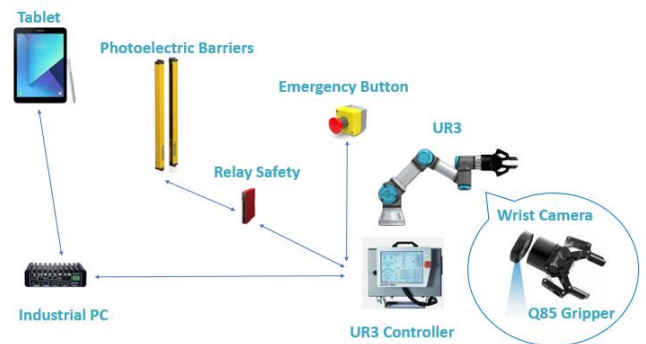
The project I4.0@TMAD (Promoção da Indústria 4.0 na Região de Trás-os-Montes e Alto Douro) has the fundamental role of getting the companies of the region to embrace new concepts and technologies encapsulated by the fourth industrial revolution. In this way, we built a collaborative cell that demonstrates the potential of the human-robot collaboration.

### Description Work Done

This work contemplated, among other aspects:

- ❑ Dimensioning of structural, mechanical and control components;
- ❑ Implementation and programming of all stakeholders;
- ❑ Testing and validation of the developed solution;
- ❑ Demonstrate the potential of the workbench at demonstrative events for entrepreneurs;

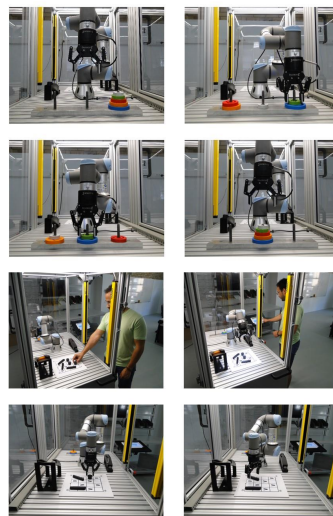
### System Architecture



### Demo Scenarios

Continuing Activity:

Autonomous and continuous development of the Hanoi tower



Scenario 1: Robot helps the operator in storage pieces;



Scenario 2: Robot helps the operator completing the assembly board.



### Demonstrative Workbench



### Demolab Events

