



## **Criação de software autónomo para ilhas de paletização robotizadas**

**ADRIANO SANTOS SILVA**

julho de 2018

# Criação de *software* autónomo para ilhas de paletização robotizadas

Adriano Santos Silva



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2018



Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de  
Disciplina de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia  
Eletrotécnica e de Computadores

Candidato: Adriano Santos Silva, Nº 1130767, 1130767@isep.ipp.pt

Orientação científica: Ramiro Barbosa, rsb@isep.ipp.pt

Empresa: Pentaline - Equipamentos Industriais, Lda

Supervisão: Nuno Almeida, nunoalmeida@pentaline.pt



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

Junho de 2018



---

# Resumo

---

A automação e a robótica são áreas tecnológicas com um elevado nível de aplicação na indústria e, com os requisitos do mercado dos dias de hoje, é impensável conceber soluções industriais complexas sem sistemas de controlo automatizados.

Enquanto que o papel da automação nestas soluções industriais é a simplificação e regularização do controlo, a robótica flexibiliza estas soluções, oferecendo uma melhor adaptação a diferentes produtos e processos. Estes são conceitos que ganham forma nas soluções de paletização robotizada.

As ilhas de paletização apresentam três principais focos: o autómato mestre, a interface de controlo e os robôs presentes na ilha. Apesar destas três componentes, o objetivo do trabalho proposto é o desenvolvimento da interface e da programação dos robôs que integram a ilha, reservando a programação do autómato mestre para trabalho futuro, em equipa, devido à sua grande complexidade.

Analisando estes objetivos pode já ser concluído que a interface exigirá grande capacidade de processamento e de armazenamento para possibilitar uma rápida e intuitiva gestão dos modelos das paletes, impedindo que se use como base para a interface as típicas consolas de automação por limitações no processamento e armazenamento. Para isto é necessária a utilização de um PC industrial e programação em *softwares* de elevado desempenho, sendo disponibilizadas as ferramentas para criação de interfaces das gamas mais avançadas da Siemens, permitindo a programação com *scripts* em Visual Basic ou em C.

Os robôs da ilha de paletização serão programados *offline*, sendo desenvolvida uma solução funcional que permita testar várias ideias que surjam para a sua lógica de trabalho, possibilitando uma perceção das variações na cadência e velocidade sem ser necessário implementar numa ilha real. Para isto terá de ser feita a modelação tridimensional de uma ilha de paletização. A programação será feita em RAPID recorrendo a tecnologias como o *conveyor tracking* utilizando o *software* RobotStudio fornecido pelo mesmo fabricante dos robôs, a ABB.

### ***Palavras-Chave:***

Automação, robótica, flexibilidade, programação, interface, ilha de paletização.



---

# Abstract

---

*Robotics and Automation are technological fields with high level of application in industry and, with nowadays market demandings, is out of mind to develop complex solutions without the automated control available.*

*While automation's role in industrial solutions is to simplify and rule the application's control, robotics role is to flexibilize these solutions, offering better adapting to different products with different processes. These are the important concepts to be considered in the development of robotic palletizing cells.*

*Palletizing cells have three master components: the PLC, the control interface and the robots present inside the cell. Despite the existence of three master components, the proposed work is the interface's development and the robot's programming, reserving the PLC's programming to future team work due to its great complexity.*

*Analyzing these objectives it's already possible to conclude that the interface will demand high processing and storage capacity to allow a fast and intuitive layout management. With this demandings is impossible to use the typical automation consoles due to limitations in processing and storage. This interface will need a industrial PC and a high-performance development software. For this are available a set of Siemens most advanced softwares that allow Visual Basic and C programming.*

*The palletizing cell robots will be offline programmed, and will be developed a functional solution that will allow to test several ideas that arise for the cell's work logic, allowing a perception of the variations in cadence and speed without being necessary to implement in a real cell. For this is necessary to do the cell's three-dimensional modeling. The programming will be done in RAPID using technologies like conveyor tracking, using the software RobotStudio provided by the same manufacturer of robots, ABB.*

**Keywords:**

*Automation, robotics, flexibility, programming, interface, palletizing cell.*



---

# Agradecimentos

---

Aproveito esta secção para agradecer a todos os que, de algum modo, me ajudaram e apoiaram ao longo deste percurso académico, em especial aos meus pais e irmãos.

Agradeço também a todos os professores que me apoiaram e acompanharam, de modo particular ao Eng. Ramiro que, de forma exemplar, acompanhou o desenvolvimento desta TESE de mestrado.

Por fim agradeço também a todos os que fazem parte da Pentaline, que me receberam e integraram na empresa, de modo particular ao Eng. Nuno, ao Eng. André e ao Eng. Daniel que partilharam o seu conhecimento comigo e guiaram o desenvolvimento de todo o projeto.



---

# Conteúdo

---

<b>Conteúdo</b>	<b>xi</b>
<b>Lista de Figuras</b>	<b>xiii</b>
<b>Lista de Tabelas</b>	<b>xvii</b>
<b>Glossário</b>	<b>xix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivações . . . . .	1
1.2 Descrição da Empresa de Estágio . . . . .	2
1.3 Objetivos . . . . .	3
1.4 Plano de Trabalho . . . . .	4
1.5 Organização do Documento . . . . .	4
<b>2 Estado da Arte</b>	<b>7</b>
2.1 Do Início ao Presente . . . . .	7
2.2 A Robótica no Meio Industrial . . . . .	13
2.2.1 O que é um Robô Industrial? . . . . .	14
2.2.2 Manipuladores Robóticos . . . . .	16
2.2.2.1 Elos de Ligação . . . . .	18
2.2.2.2 Juntas . . . . .	18
2.2.2.3 Graus de Liberdade . . . . .	19
2.2.2.4 Desempenho de um Robô . . . . .	21
2.2.2.5 Métodos de Atuação dos Robôs . . . . .	26
2.2.2.6 Classificação dos Robôs . . . . .	27
2.2.2.7 Punho . . . . .	31
2.2.2.8 Atuadores Finais . . . . .	32
2.3 Interfaces Industriais Homem-Máquina . . . . .	33
<b>3 O Processo de Paletização</b>	<b>41</b>

3.1	Robôs de paletização . . . . .	45
3.1.1	Manipuladores . . . . .	45
3.1.2	Atuadores Finais . . . . .	46
3.2	Metodologias de Paletização . . . . .	51
3.3	A Ilha de Paletização . . . . .	55
3.4	Caso de Estudo . . . . .	61
3.4.1	Paletização de garrafas de refrigerante . . . . .	61
3.4.1.1	Desafio . . . . .	61
3.4.1.2	Solução . . . . .	63
<b>4</b>	<b>Desenvolvimento de Interface Homem-Máquina para Ilhas de Paletização</b>	<b>71</b>
4.1	Objetivos . . . . .	71
4.2	Requisitos . . . . .	73
4.3	Alternativas de Solução . . . . .	75
4.4	Descrição da Arquitetura . . . . .	78
4.5	Procedimentos de Teste . . . . .	81
4.6	Protótipo Funcional . . . . .	84
4.6.1	Implementação Realizada . . . . .	84
4.6.1.1	Base de Dados . . . . .	84
4.6.1.2	Interface Homem-Máquina . . . . .	94
4.7	Testes . . . . .	115
<b>5</b>	<b>Programação e Simulação de uma Ilha de Paletização</b>	<b>117</b>
5.1	Objetivos . . . . .	117
5.2	Requisitos . . . . .	118
5.3	Descrição da Arquitetura . . . . .	119
5.4	Procedimentos de Teste . . . . .	121
5.5	Implementação realizada . . . . .	123
5.6	Testes . . . . .	135
<b>6</b>	<b>Conclusão</b>	<b>137</b>
	<b>Bibliografia</b>	<b>141</b>
<b>A</b>	<b>Esquemas das Paletes a Formar na Ilha de Paletização</b>	<b>149</b>
<b>B</b>	<b>Programação dos procedimentos da interface em Visual Basic</b>	<b>153</b>
<b>C</b>	<b>Implementação de Ilha de Paletização no RobotStudio</b>	<b>157</b>

---

# Lista de Figuras

---

1.1	Pentaline . . . . .	2
1.2	Calendarização das tarefas realizadas . . . . .	4
2.1	Painel de relés de uma máquina anterior a 1980 [4]. . . . .	8
2.2	Modelo 084 (à esquerda); Modelo 184 (à direita)[6][7]. . . . .	9
2.3	Modelo de um robô baseado nos desenhos de DaVinci [10]. . . . .	10
2.4	Estrutura de um robô PUMA [13]. . . . .	11
2.5	Evolução da representação da robótica na indústria [14]. . . . .	12
2.6	Relação entre os volumes de produção e o custo de cada solução [15]. . . . .	14
2.7	Estimativa do número de robôs fornecidos à indústria nos últimos 10 anos e até 2020 [16]. . . . .	14
2.8	Principais constituintes de um robô [18]. . . . .	15
2.9	Elos de ligação e juntas num robô [19]. . . . .	16
2.10	Ilustração da deflexão de uma viga de aço para pequenas forças (em cima) e para grandes forças com deformação (em baixo) [21]. . . . .	17
2.11	Exemplificação do <i>backlash</i> num conjunto de engrenagens [23]. . . . .	19
2.12	Ilustração dos vários tipos de juntas [24]. . . . .	20
2.13	Robôs com 4, 5 e 6 graus de liberdade (esquerda para a direita) [25]. . . . .	21
2.14	Diferença entre precisão e repetibilidade [26]. . . . .	22
2.15	Diagrama de carga do robô ABB IRB2400-10 [27]. . . . .	23
2.16	Sensor de forças externas a atuar no atuador final [28]. . . . .	24
2.17	Posições de montagem possíveis para o ABB IRB 140 [29]. . . . .	25
2.18	Classificação dos robôs de acordo com as suas características [30]. . . . .	28
2.19	Ilustração da estrutura de um robô paralelo [32]. . . . .	29
2.20	Ilustração de um robô com estrutura em pórtico [33]. . . . .	29
2.21	Ilustração de um robô com configuração esférica [34]. . . . .	30
2.22	Ilustração de um robô com configuração articulada [35]. . . . .	31
2.23	Ilustração de um robô com configuração SCARA [36]. . . . .	31
2.24	Exemplo de uma garra mecânica de atuação pneumática [39]. . . . .	32

2.25	Tocha de soldadura acoplada a um robô (à esquerda) e robô de pintura na indústria automóvel (à direita) [38]. . . . .	33
2.26	Painel industrial <i>touchscreen</i> SIMATIC IFP1900 produzido pela Siemens [42]. . . . .	34
2.27	Resultados da comparação entre os interfaces mais antigos e os mais recentes [43]. . . . .	35
2.28	Ecrã apresentado durante o arranque que não exibia dados acerca da entrada de fluído, apenas da saída [47]. . . . .	37
2.29	Esquema que ilustra as relações entre os aspetos da consciência situacional [43]. . . . .	37
2.30	Ilustração dos aspetos da consciência situacional [43]. . . . .	38
2.31	Diferença entre a apresentação dos valores brutos e uma representação analógica [43]. . . . .	39
2.32	Principais diferenças entre as versões do WinCC [51]. . . . .	40
3.1	Ilustração dos processos anteriores e subsequentes à paletização [52]. . . . .	42
3.2	Ilustração das principais características de uma palete. . . . .	43
3.3	Exemplo do mosaico adotado para um determinado tipo de artigo. . . . .	44
3.4	<i>Gripper</i> de atuação pneumática com uma chapa fixa [56]. . . . .	47
3.5	<i>Gripper</i> com as funcionalidades de agarrar produtos, paletes e intercalares [57]. . . . .	48
3.6	Manipulação de sacos de cacau em pó [58]. . . . .	49
3.7	<i>Gripper</i> com movimento de abertura linear (à esquerda) e em V (à direita) [59] [60]. . . . .	49
3.8	Garra de sucção a manipular caixas de cartão [61]. . . . .	50
3.9	Garra de sucção com funcionalidades de apanhar intercalares e paletes [57]. . . . .	51
3.10	Implementação da metodologia de paletização à caixa numa ilha de paletização [64]. . . . .	52
3.11	Ilha de paletização que segue a metodologia de paletização à fila [65]. . . . .	54
3.12	Ilha de paletização que segue a metodologia de paletização à camada [67]. . . . .	54
3.13	Ilha de paletização à camada [67]. . . . .	56
3.14	Ilha de paletização à fila [68]. . . . .	59
3.15	Ilha de paletização ao produto [69]. . . . .	60
3.16	Ilha de paletização apresentada como solução ao desafio colocado pela Refrescos Lda. . . . .	63
3.17	Linha de formação de mosaico e <i>picking</i> . . . . .	64
3.18	Robô de formação de mosaico e respetivo controlador, pedestal e <i>gripper</i> . . . . .	65
3.19	Robô de manipulação de camadas e respetivo controlador, pedestal e <i>gripper</i> . . . . .	66

3.20	Robô de manipulação de intercalares e respetivo controlador, pedestal e <i>gripper</i> . . . . .	67
3.21	Armazém automático de paletes e armazém de intercalares. . . . .	68
4.1	Principais focos de informação na interface. . . . .	72
4.2	Origem dos dados para configuração de um mosaico. . . . .	74
4.3	PC industrial Nuvo-3000 [70]. . . . .	76
4.4	Esquema ilustrativo das comunicações entre a interface, a base de dados e o PLC. . . . .	79
4.5	Estrutura dos menus e submenus da interface para ilhas de paletização. . . . .	80
4.6	Esquema ilustrativo da estrutura da base de dados. . . . .	81
4.7	Informação de um produto que o utilizador introduz na interface e linha correspondente na base de dados. . . . .	82
4.8	Diagrama entidade-relação da base de dados. . . . .	86
4.9	Características da tabela dos artigos. . . . .	87
4.10	Características da tabela dos intercalares. . . . .	88
4.11	Características da tabela das paletes. . . . .	88
4.12	Características da tabela das coordenadas. . . . .	89
4.13	Características da tabela dos mosaicos. . . . .	89
4.14	Exemplo de um número recebido no campo "Artigos_por_camada" para um mosaico com sucessão de camadas personalizada . . . . .	90
4.15	Informação recebida na tabela das paletes da base de dados para o exemplo apresentado. . . . .	91
4.16	Informação recebida na tabela dos intercalares da base de dados para o exemplo apresentado. . . . .	91
4.17	Informação recebida na tabela dos artigos da base de dados para o exemplo apresentado. . . . .	92
4.18	Coordenadas atribuídas e grupos formados na construção do mosaico. . . . .	92
4.19	Informação recebida na tabela dos mosaicos da base de dados para o exemplo apresentado. . . . .	93
4.20	Informação da tabela das coordenadas para o exemplo apresentado. . . . .	94
4.21	Base do ecrã da interface Homem-máquina desenvolvida. . . . .	95
4.22	Ecrã apresentado ao utilizador no menu de informações da ilha. . . . .	96
4.23	Ecrã apresentado ao utilizador no menu de alarmes. . . . .	97
4.24	Configurações dos eventos e alarmes já configurados. . . . .	97
4.25	Ecrã apresentado ao utilizador no menu de alarmes. . . . .	98
4.26	Menu que dá acesso à gestão dos itens da base de dados. . . . .	99
4.27	Janelas de criação de artigos, paletes e intercalares (da esquerda para a direita). . . . .	100
4.28	Ecrã de agrupamento de novo artigo no <i>gripper</i> . . . . .	103
4.29	Janelas de eliminação de artigos, paletes e intercalares e mosaicos. . . . .	103
4.30	Ecrã inicial do criador de mosaicos. . . . .	106

4.31	Ecrã apresentado para escolha do artigo. . . . .	106
4.32	Botões apresentados para indicar distribuição de intercalares e paletes. . . . .	107
4.33	Janela apresentada para indicar sucessão de camadas. . . . .	108
4.34	Janela apresentada para criar o <i>layout</i> do mosaico. . . . .	109
4.35	Fluxograma que representa o <i>script</i> de adicionar item. . . . .	112
4.36	Janela apresentada para o menu de comando geral. . . . .	114
5.1	Organização da informações na memória do robô [75]. . . . .	120
5.2	Organização das tarefas do programa da ilha de paletização. . . . .	120
5.3	Principais constituintes do <i>conveyor tracking</i> [76]. . . . .	121
5.4	Relação do PLC com os principais constituintes da ilha. . . . .	122
5.5	Criação do <i>gripper</i> do robô manipulador de camadas. . . . .	123
5.6	<i>Grippers</i> criados para manipulação de <i>packs</i> (à esquerda) e de intercalares (à direita). . . . .	124
5.7	Enquadramento da linha de formação de mosaico com o robô de manipulação de <i>packs</i> . . . . .	125
5.8	Enquadramento da linha de formação de mosaico e do <i>conveyor</i> de paletes com o robô de manipulação de camadas. . . . .	126
5.9	Enquadramento do robô de manipulação de intercalares com o <i>conveyor</i> de paletes e com o armazém de intercalares. . . . .	127
5.10	Área definida para o <i>conveyor tracking</i> . . . . .	128
5.11	Tarefas que cada um dos controladores tem programadas. . . . .	131
5.12	Referencial definido para posicionamento na mesa de formação de mosaico. . . . .	133
5.13	Mosaicos testados na ilha de paletização (Laranja 1.25 4x3 à esquerda, Laranja 1.5 3x2 (método 1) ao meio e Laranja 1.5 3x2 (método 2) à direita). . . . .	136

---

# Lista de Tabelas

---

2.1	Comparação das principais características de cada método de atuação [26]. . . . .	27
3.1	Comparação entre robôs de 4 e 6 eixos [54] [55]. . . . .	46
3.2	Produtos a serem trabalhados na ilha de paletização. . . . .	62
3.3	Paletes e intercalares a utilizar na ilha de paletização. . . . .	62
3.4	Características das paletes a formar na ilha de paletização. . . . .	63
3.5	Comparação entre cadências pedidas e atingidas. . . . .	69
5.1	Entradas e saídas do controlador dos robôs de camada e intercalares. . .	129
5.2	Entradas e saídas do controlador do robô de formação de mosaico. . .	130



---

# Glossário

---

Abreviatura	Descrição	Definição
AGV	<i>Automated guided vehicle</i>	página 27
BP	British Petroleum	página 36
CLP	Controlador Lógico Programável	página 8
CNC	<i>Computer Numeric Control</i>	página 2
CPU	<i>Central Processing Unit</i>	página 15
DC	<i>Direct Current</i>	página 11
E/S	Entradas e Saídas	página 15
IEC	International Electrotechnical Commission	página 35
ISEP	Instituto Superior de Engenharia do Porto	página 2
ISO	International Organization for Standardization	página 35
LASER	<i>Light Amplification by Stimulated Emission of Radiation</i>	página 33
NASA	National Aeronautics and Space Administration	página 11
PC	<i>Personal Computer</i>	página 16
PLC	<i>Programmable Logic Controller</i>	página 2
PUMA	<i>Programmable Universal Machine for Assembly ou Programmable Universal Manipulation Arm</i>	página 11
SCARA	<i>Selective Compliance Assembly Robot Arm</i>	página 28



# Capítulo 1

---

## Introdução

---

Desde o século XVIII, como consequência da revolução industrial, começou a existir a necessidade de transportar os produtos de uma fábrica até aos vários pontos de consumo. Até esta época os artigos consumidos eram produzidos pelos próprios consumidores ou adquiridos diretamente a produtores locais, o que excluía a necessidade de transporte.

No entanto, só em finais do século XIX é que o transporte de quantidades elevadas passou a ser uma necessidade. Com isto, surgem questões de eficiência, o que leva a que se comece a pensar em técnicas de acondicionamento de produtos de forma a que o seu manuseamento e transporte em quantidade seja facilitado.

Desde então, a paletização é um processo indispensável na indústria, tanto para uma melhor organização de *stock*, como para um despacho mais eficaz dos produtos finais.

Até hoje, os métodos de paletização têm sofrido alterações, no sentido de uma melhoria contínua na qualidade e nos custos associados. Esta procura da melhoria, em paralelo com os avanços tecnológicos nas áreas de automação e robótica, está a levar a paletização a um patamar de total autonomia.

### 1.1 Motivações

Devido às diferentes vertentes deste trabalho estão presentes várias motivações ao longo do seu desenvolvimento. As principais motivações vêm no seguimento da ideia de uma melhoria na qualidade e nos custos associados à paletização em que se pretende que os sistemas tradicionais sejam substituídos por **automação industrial e robótica**.

A automação pode ser vista como a evolução dos sistemas mecânicos em que operadores humanos estão em constante contacto com os materiais e os equipamentos para que estes realizem, com sucesso, as funções para as quais foram projetados.

Com a introdução da automação nestes sistemas a ação humana é reduzida ao mínimo imprescindível, e as decisões normalmente tomadas ao longo do processo estão previstas na programação do PLC, sendo transmitidas não só à máquina mas também a um conjunto de dispositivos de monitorização que estejam ligados em rede.

A robótica, que é vista como um ramo da automação, torna os processos mais flexíveis, permitindo que sejam produzidas soluções com grande adaptabilidade à alteração de parâmetros e é, por este motivo, também chamada "Soft Automation"<sup>1</sup> [1].

Em suma, os avanços na tecnologia nos ramos de automação industrial e robótica tornam possível que se concebam sistemas com interfaces simpáticas aos operadores, com grande flexibilidade, capazes de reduzir custos de operação e manutenção e aumentar a produtividade, qualidade e segurança, já que os operadores deixam de ter contacto direto com a maquinaria [2].

## 1.2 Descrição da Empresa de Estágio

A empresa acolhedora do estágio, a Pentaline (ver Figura 1.1), é uma empresa do concelho de Oliveira de Azeméis, situada no Parque Imodelta na freguesia de Pindelo, a cerca de 45 km do ISEP.

A Pentaline dedica a sua atividade ao projeto e construção de equipamentos industriais, com vasta experiência na construção de linhas de produção, que incluem desde transportadores de várias naturezas (convergentes e divergentes) a elevadores e a equipamentos mais específicos (enchimento, etiquetagem, empacotamento e paletização).



Figura 1.1: Pentaline

---

<sup>1</sup>*Soft Automation* - Categoria da automação que se refere a equipamentos reprogramáveis e reconfiguráveis, como os robôs. No lado oposto, a "*Hard Automation*" refere-se a equipamentos construídos para um propósito específico, como as CNC.

Inicialmente, a Pentaline tinha o seu foco de trabalho centrado na produção de transportadores e equipamentos similares, posteriormente foi avançando para a produção de outros tipos de máquinas mais complexas com serviços de automação subcontratados. O passo seguinte foi a produção de ilhas de empacotamento e paletização altamente automatizadas e robotizadas. Recentemente a empresa tem intensificado a sua aposta no desenvolvimento a nível interno das soluções automatizadas e robotizadas, estabelecendo importantes parcerias com empresas destes sectores como a ABB e a Siemens. Com um previsível aumento da carga de trabalho, a Pentaline tem planeada uma mudança de instalações de modo a que possa ser possível um aumento da capacidade de produção.

Com estes objetivos associados ao valor que a Pentaline atribui à formação surgem, então, condições para o decorrer de um estágio nas suas instalações, onde é feito o projeto e o fabrico de todos os seus equipamentos.

### 1.3 Objetivos

Após um produto ter passado por todos os processos de produção que lhe conferem as características desejadas é necessário distribuir o mesmo pelos diversos pontos de venda, agrupados em paletes que por sua vez têm uma organização dos artigos muito particular e que depende das suas características físicas. O processo de paletização contempla diversas variáveis, entre outras é dada especial importância à rapidez do processo, que possibilita um escoamento mais eficaz do produto, e à consistência da paleta, que permite um transporte das mesmas sem problemas de desmoronamento e permite também uma melhor organização nos veículos que as transportam. De modo a melhorar os índices referentes a estes dois fatores a robótica e automação têm vindo a ser cada vez mais aplicadas nos processos de paletização. Como qualquer sistema deste tipo, este necessita de uma interface que permita o controlo das ilhas de paletização. Um dos objetivos principais deste trabalho é o **desenvolvimento de uma interface** que seja capaz de se interligar e adaptar a sistemas robotizados diferentes. A interface deverá solicitar a introdução de todos os dados necessários ao funcionamento, o que despoletará, consequentemente, a total adaptação da ilha robotizada (dimensões do produto, fila, camada, paleta base, paleta dupla, intercalares, cadências/velocidades). Em suma, a interface deve possibilitar a criação e o armazenamento de modelos com produto, paleta, intercalar e mosaico. Posteriormente à criação, deve ser possível enviar um dos modelos criados e armazenados para produção, o que se traduz na comunicação dos dados referentes ao modelo escolhido para o autómato mestre que controla a ilha de paletização.

Este autómato mestre será o responsável pelo controlo e sincronização entre todos os equipamentos existentes na ilha, sendo que uma das funções incluídas nesta responsabilidade que lhe é atribuída é a transmissão dos valores recebi-

dos por parte da interface aos robôs da ilha de paletização. Estes robôs terão o seu próprio processamento, no entanto devem sempre de estar dependentes do PLC para realizar as suas operações. Como tal, outro objetivo deste trabalho é desenvolver a programação dos robôs de uma ilha deste tipo que seja capaz de servir de base para vários projetos diferentes com pequenos ajustes ou alterações. A programação dos robôs para formação de paletes não se pode prender a um mosaico em específico, sendo também objetivo que estes robôs adaptem o seu funcionamento consoante o mosaico que lhes é transmitido pelo PLC. Outro objetivo relacionado com os robôs é que seja feita uma modelação tridimensional de uma ilha de paletização. Ao ter uma ilha modelada no Robot Studio é possível que sejam realizados testes de diferentes métodos de manipulação e observar os seus resultados, como verificação de cadências ou até colisões com outros equipamentos. Pode ainda ser referido que, apesar de grande parte das características de otimização estarem relacionadas com cada projeto em específico, é objetivo que sejam já implementadas algumas técnicas que sejam comuns à grande maioria das ilhas, como é o caso da utilização do *conveyor tracking*.

## 1.4 Plano de Trabalho

Ao longo do desenvolver deste trabalho foram várias as tarefas que foram realizadas, sendo que as mesmas têm uma ordem específica e uma data de realização. O resumo destes dados é apresentado na Figura 1.2 que apresenta a calendarização da atividades relacionadas com o decorrer do projeto.

Nome da Tarefa	Início	Conclusão	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun
<b>1 Adaptação à empresa</b>	11/09/17	22/09/17	■									
<b>2 Interface homem-máquina</b>	25/09/17	23/03/18		■	■	■	■	■	■	■	■	■
2.1 Estudo do problema	25/09/17	03/11/17		■	■							
2.1.1 Softwares de desenvolvimento	25/09/17	16/10/17		■	■							
2.1.2 Bases de dados	17/10/17	03/11/17			■							
2.2 Base de dados SQL	06/11/17	15/12/17			■	■						
2.2.1 Estruturação	06/11/17	17/11/17			■							
2.2.2 Comunicação com WinCC	20/11/17	15/12/17				■	■					
2.3 Desenvolvimento da interface no WinCC	20/11/17	02/03/18				■	■	■	■			
2.4 Testes e Correções	05/03/18	23/03/18							■	■		
<b>3 Desenvolvimento de Ilha de Paletização</b>	06/03/18	20/04/18							■	■	■	
3.1 Estudo dos Equipamentos	06/03/18	08/03/18							■			
3.2 Estudo de tecnologias associadas (Conveyor Tracking, Multitasking, etc)	09/03/18	14/03/18							■			
3.3 Modelação da ilha	15/03/18	21/03/18							■			
3.4 Programação dos robôs	22/03/18	13/04/18								■	■	
3.5 Testes e Correções	16/04/18	30/04/18									■	
<b>4 Desenvolvimento do Relatório</b>	27/11/17	12/06/18				■	■	■	■	■	■	■

Figura 1.2: Calendarização das tarefas realizadas

## 1.5 Organização do Documento

Este documento encontra-se organizado em 6 capítulos, ao longo dos quais é descrito o trabalho realizado. Para que se perceba melhor a estrutura deste relatório são aqui descritos de forma sucinta os conteúdos de cada capítulo.

No primeiro capítulo é feita uma introdução ao tema do trabalho, são descritas as motivações, é apresentada a empresa onde decorreu o estágio e são traçados os objetivos que se pretendem atingir com o trabalho desenvolvido.

No segundo capítulo é feita uma breve introdução histórica da robótica e da automação, uma análise das características e bases da robótica na atualidade e uma exposição da importância das interfaces homem-máquina e as principais linhas a seguir no seu desenvolvimento.

No terceiro capítulo é apresentada uma abordagem à temática da paletização robotizada, analisando que robôs são mais indicados para estes fins, descrevendo as ilhas de paletização ao nível dos equipamentos que as compõe e das metodologias adotadas e, por fim é feita a análise de um caso de estudo que propõe a implementação de uma ilha de paletização robotizada.

No quarto capítulo são detalhados os objetivos que se pretendem atingir com a interface a ser desenvolvida e os requisitos aos quais a mesma deve corresponder. É também apresentado o projeto preliminar do sistema, onde é descrita a arquitetura da interface e as bases sob as quais a mesma assenta. Por fim é apresentado o trabalho desenvolvido, dando especial ênfase às funcionalidades mais complexas que foram desenvolvidas.

No quinto capítulo é abordada a temática do desenvolvimento da modelação tridimensional da ilha e da sua programação, sendo apresentados os objetivos desse trabalho, a arquitetura que se pretende seguir no desenvolvimento da ilha no Robot Studio e é, por fim, apresentado o desenvolvimento e teste da ilha 3D e da programação de cada um dos robôs.

No sexto e último capítulo são apresentadas as conclusões recolhidas ao longo do desenvolvimento deste trabalho sendo também descritos pontos de melhoria a partir dos quais se pode desenvolver trabalho no futuro.



## Capítulo 2

---

# Estado da Arte

---

À semelhança de todas as tecnologias presentes nos dias de hoje, a automação e a robótica são áreas que nasceram da necessidade de simplificar e melhorar atividades ou processos. Após a sua criação foram sendo feitas sucessivas apostas no seu desenvolvimento até que estas tecnologias ganharam a forma que é conhecida no presente, possibilitando a criação de soluções industriais complexas e com grande flexibilidade. Neste capítulo é possível ter uma visão sobre as capacidades destas tecnologias que, apesar do seu elevado grau de desenvolvimento, continuam a ser melhoradas dia após dia.

### 2.1 Do Início ao Presente

A transição cultural da colonização nómada para sedentária é o primeiro grande exemplo da aplicação de um conjunto de novas técnicas para contribuição do aumento do bem estar do Homem. Desde então é possível observar uma contínua procura pela diminuição da participação humana em tarefas repetitivas e fisicamente exigentes. É desta ideia que, milhares de anos depois, surge a automação industrial.

A revolução industrial, que foi o culminar da necessidade de produzir mais e melhor, é o principal marco da substituição da mão de obra humana por máquinas que tornam os processos mais eficientes e aumentam a qualidade dos produtos. Exemplo disto são as linhas de produção no ramo automóvel que, com a aplicação de automação industrial, começaram a produzir em massa e com melhor qualidade [3].

Apesar da melhoria face aos sistemas de produção antigos ser significativa, estes primeiros modelos de automação industrial tinham grandes desvantagens. O facto de serem extremamente inflexíveis, em muito contribuía para este conjunto

de desvantagens, isto é, quando havia necessidade de alterar as características do modelo em produção, ou quando surgia um pedido com especificações especiais, era necessário realizar um complexo ajuste da linha de produção que podia demorar vários dias e gerar grandes atrasos na produção agendada. No entanto este não era o único problema destas soluções. Para o controlo de cada máquina era exigido um painel de relés e, portanto, numa linha de produção existiam vários painéis, resultando em milhares de relés que estavam constantemente sujeitos aos desgastes (ver Figura 2.1). O grau de complexidade da arquitetura dos painéis de relés não permitia uma fácil análise das avarias, isto é, quando um relé avariava, o sistema simplesmente deixava de funcionar e a deteção e reparação destas avarias simples podia demorar muitas horas, gerando muitas vezes grandes atrasos na produção agendada.

Foi para ultrapassar estes problemas de inflexibilidade e logística nas linhas de produção que os engenheiros da General Motors começaram, na década de 60, a associar os recentes avanços no controlo computadorizado às linhas de produção. Os requisitos que estes engenheiros impunham, associados aos conhecimentos de Dick Morley nestas áreas fazem surgir o primeiro Controlador Lógico Programável (CLP, do inglês PLC). Os requisitos impostos a D. Morley para o desenvolvimento deste projeto passavam principalmente por um sistema: semelhante a um computador em termos de flexibilidade, mas com um custo competitivo com o dos painéis de relés; de manutenção fácil e programação semelhante à lógica dos relés; capaz de funcionar num ambiente industrial com pó, humidade, vibrações e ruído eletromagnético; que possibilite a troca de componentes e tenha capacidade de ser expansível, ou seja, um sistema modular [5].

Desde então o pequeno controlador, o Modelo 084, com muito pouca memória e baixa rapidez, mesmo para as aplicações da altura começou o seu caminho de evolução na empresa criada por Dick Morley, a Modicon Company. Apesar de o Modelo 084 já ser um considerável avanço na automação, os PLC só ganharam popularidade com o surgimento do Modelo 184 (ver Figura 2.2), um verdadeiro sucesso por conseguir corrigir as graves lacunas no anterior modelo.

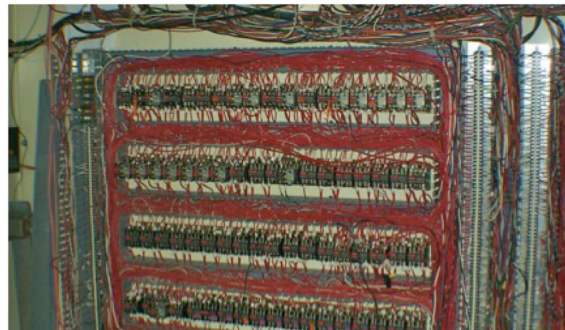


Figura 2.1: Painel de relés de uma máquina anterior a 1980 [4].

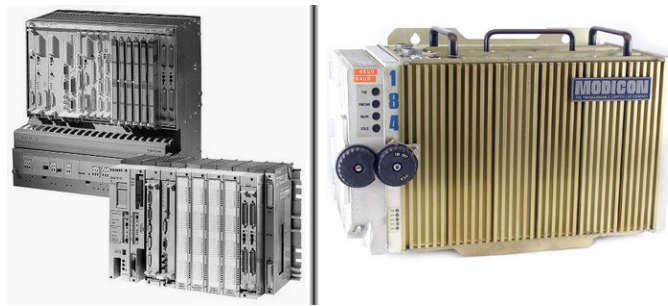


Figura 2.2: Modelo 084 (à esquerda); Modelo 184 (à direita)[6][7].

Os primeiros PLC apenas conseguiam trabalhar com base em entradas e saídas digitais, na sua programação apenas era possível trabalhar com contactos, *timers* e contadores. No entanto as necessidades foram aumentando e mais tarde começou a ser possível usar entradas e saídas analógicas e foram também introduzidos novos tipos de variáveis, exemplo das variáveis de vírgula flutuante.

Outro dos pontos importantes da evolução dos PLC é a sua capacidade de comunicação, inicialmente limitada entre equipamentos do mesmo fabricante e em alguns casos apenas do mesmo modelo. No entanto a evolução neste campo seguiu o caminho da integração de todos sistemas num só, o que implicou desenvolvimentos a nível de compatibilidade entre equipamentos e rapidez nas comunicações.

Ao nível da programação os avanços tecnológicos foram, também, notáveis. Inicialmente a programação de um PLC era necessariamente feita em laboratório e através de complexos equipamentos de programação [5]. Só em finais da década de 70 é que começou a ser viável a programação de equipamentos no seu local de instalação, isto é, fazer uma edição da programação já com todo o sistema instalado, como acontece frequentemente em fases de testes. Para que isto fosse possível os autómatos começaram a ter uma porta de comunicação específica para programação que se ligava a um pequeno programador portátil. Foi no seguimento destes desenvolvimentos que mais tarde, com a popularização de computador, surge o primeiro software de programação de PLC baseado em Windows, o DirectSOFT da Automation Direct. Com os desenvolvimentos neste tipo de *softwares* não só se tornou mais fácil de programar os PLC como também possibilitou uma mais fácil deteção de erros na programação e falhas no sistema.

Noutro campo da automação, a **Robótica**, a história é feita desde as primeiras civilizações. Já nas civilizações gregas eram feitas referências a mecanismos muito básicos com aparência humana e movidas por força da água ou pela ação de pesos. No entanto a evolução destes mecanismos nestas civilizações não foi um ponto de interesse, já que não havia necessidade económica nem prática do aumento da rapidez nas tarefas associadas a estes modelos [8].

Foi com os estudos de Leonardo DaVinci (1452 - 1519) na área da robótica que surgiram os primeiros desenvolvimentos significativos [9]. DaVinci deixou

um vasto material com estudos sobre sistemas mecânicos. Entre estes materiais destacam-se um conjunto de desenhos que sugerem um robô orientado para o combate, um soldado robô, em que os seus movimentos eram feitos por engrenagens e polias (ver Figura 2.3).

No entanto só no século XX, passados mais de 400 anos, é que a palavra robô é atribuída a estes mecanismos [8]. Foi em 1921 que o dramaturgo Karel Čapek (1890 - 1938) apresentou este termo numa das sua peças teatrais, *Rossumovi Univerzální Roboti* (Robôs Universais de Rossum). Nesta peça, uma das personagens era uma máquina com semelhanças humanas que realizava trabalhos pesados, na altura feitos por escravos. Foi notável o sucesso desta obra, 2 anos após o seu lançamento já se encontrava traduzida para mais de 30 línguas.

Mais tarde, Isaac Azimov (1920 - 1992) introduz, através de uma pequena publicação no ano de 1942 intitulada de *"Runaround"*, o termo robótica, a ciência que estuda e utiliza os robôs. Em 1950 publica um conjunto de histórias intituladas de *"I Robot"* e é nestes textos que são pela primeira vez propostas as três leis aplicáveis à robótica. Mais tarde, Azimov, decidiu acrescentar mais uma lei às três que havia criado, a lei zero. Estas 4 leis da robótica podem ser descritas da seguinte forma [11]:

- **Lei 0** - *um robô não pode fazer mal à humanidade e nem, por inação, permitir que ela sofra algum mal. Desse modo, o bem da humanidade é primordial ao dos indivíduos;*
- **Lei 1** - *um robô não pode fazer mal a um ser humano e nem, por inação, permitir que algum mal lhe aconteça;*
- **Lei 2** - *um robô deve obedecer às ordens dos seres humanos, exceto quando estas contrariarem a primeira lei;*
- **Lei 3** - *um robô deve proteger a sua integridade física, desde que com isto não contrarie as duas primeiras leis.*



Figura 2.3: Modelo de um robô baseado nos desenhos de DaVinci [10].

Apesar de intemporais, isto é, mesmo tendo sido criadas há quase 70 anos continuam válidas para os dias de hoje, estas leis são vistas como tendo sido criadas num contexto ficcional pois em 1950 era praticamente impossível prever os enormes desenvolvimentos que acabaram por ocorrer nesta área.

No entanto, mesmo com a presença da ideia dos sistemas robotizados, só foram possíveis desenvolvimentos significativos a partir de 1940, altura em que foram criados os computadores. Reunidas as condições para o desenvolvimento de sistemas robóticos, em 1961, George C. Devol (1912 - 2011) e Joseph Engelberger (1925 - 2015) criam o UNIMATE, um ano depois este robô é aplicado na indústria, na Ford Motor Company, onde era utilizado para descarga de uma máquina de fundição [12].

Anos mais tarde, na década de 60, surge o Shakey que resultou das pesquisas de um grupo de professores no Stanford Research Institute. Este pequeno robô constituiu um marco neste área devido à integração de sistemas de acionamento elétrico e sensores que permitiam o robô perceber o ambiente que o rodeia. Após estes desenvolvimentos se terem tornado realidade é então criado o PUMA, um robô capaz de realizar tarefas de montagem. Desde então, este tipo de robôs tem sido utilizado na indústria tanto para operações de montagem como para operações de soldadura e também há casos de utilização em laboratórios de investigação. A popularidade destes robôs deve-se ao facto de terem um elevado grau de flexibilidade graças à sua arquitetura composta por uma cadeia de membros articulados com juntas rotacionais e prismáticas (ver Figura 2.4) atuadas por servomotores DC acoplados a conjuntos de engrenagens, o que lhes confere a capacidade de se deslocarem no espaço XYZ [12] [13].

Nos anos 70, numa altura em que a tecnologia já se encontrava com bases sólidas, a NASA decide integrar nas suas missões à lua plataformas autónomas

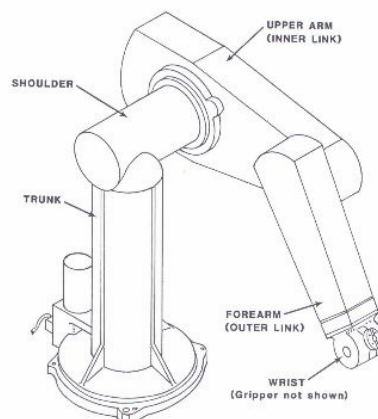


Figura 2.4: Estrutura de um robô PUMA [13].

que tinham de ser capazes de explorar terrenos hostis e é deste modo que as formas robóticas criadas até então foram alvo de aperfeiçoamentos significativos, tornando-as mais ágeis e robustas [12]. É desde estes desenvolvimentos que a robótica ganhou grande popularidade a nível mundial (ver Figura 2.5) e desde então surgiram novas arquiteturas e tipos de atuação, cada um com as suas vantagens, e assim a robótica foi enriquecendo até que hoje em dia pode ser observada uma capacidade de adaptação destes sistemas a quase todos os tipos de indústrias e tarefas mesmo que sejam exigentes em termos de flexibilidade, rapidez, autonomia e ambiente em que se inserem.

Deste modo a inclusão dos robôs prevista por Čapek e Azimov, tornou-se realidade e cada vez os seus recursos são mais utilizados resultando num aumento da qualidade de vida do Homem.

Em suma, os desenvolvimentos na robótica podem ser divididos em 4 etapas. Uma primeira etapa em que era realizada a gestão de tarefas repetitivas com autonomia muito limitada, a segunda etapa foi marcada pela inclusão dos primeiros sistemas de visão artificial, na terceira etapa destacam-se sistemas com mobilidade avançada, e por fim uma etapa que ainda tem vindo a ser desenvolvida, onde se destacam os sistemas com inteligência artificial [12].

Esta capacidade de adaptação levou a que a inclusão dos sistemas robóticos chegasse aos sistemas de paletização e de despaletização. O grande objetivo da inclusão de soluções robotizadas prende-se essencialmente com questões de rendimento embora a qualidade também seja um ponto de melhoria face às soluções convencionais. Nos sistemas em que a paletização é feita por um ou mais colaboradores há limites da capacidade física que limitam a capacidade de repetitividade e cadência, como é exemplo da paletização de *packs* de garrações de água em que o peso de cada item limita a cadência dos operadores, ou na paletização de sacos, em que a deformabilidade dos artigos limita a qualidade das paletes. Foi por estes motivos que surgiram, inicialmente as paletizadoras automáticas mas, por questões de flexibilidade, foram sendo substituídas por soluções robotizadas.

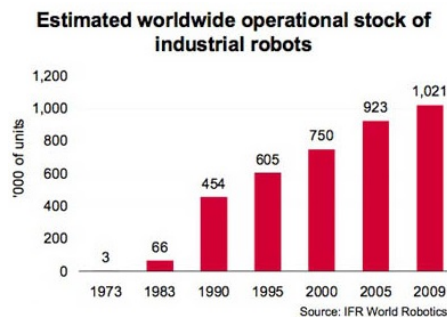


Figura 2.5: Evolução da representação da robótica na indústria [14].

## 2.2 A Robótica no Meio Industrial

Na atualidade, a robótica detém um papel extremamente relevante para a indústria a nível mundial. A conquista desta posição preponderante deve-se às características destes equipamentos, capazes de conferir a um processo elevada qualidade, produtividade, segurança, flexibilidade e eficiência. Além destas vantagens, os robôs são vistos pelos investidores como o fim de alguns problemas relacionados com a mão de obra humana.

O investimento num robô restringe-se ao seu custo de aquisição, programação e manutenção, o que é sempre bem definido a longo prazo num plano de custos. Outra vantagem é o tempo de concessões ao longo do trabalho em que operadores humanos estão claramente em desvantagem devido às pausas que são necessárias para alimentação e descanso. Trabalhos muito monótonos, repetitivos ou fisicamente exigentes não constituem um problema para os robôs, ao contrário de um trabalhador comum, que ficaria esgotado num tempo inferior ao do seu turno e provavelmente refletindo este cansaço na sua saúde. Por fim, a vantagem que normalmente é mais visível é a rapidez de um robô. A capacidade de rapidez destes equipamentos é de tal forma elevada que nenhum operador é comparável. Todas estas vantagens se traduzem num aumento da produtividade das indústrias e num aumento de qualidade de vida da população.

No entanto há ainda aspetos técnicos em que os humanos estão em vantagem, como a capacidade de lidar facilmente com o inesperado e maior capacidade de previsão de necessidades próprias e de sistemas na vizinhança, o que significa que este campo da tecnologia ainda é alvo de estudo e tem margem para ainda mais evolução.

Em termos económicos há indústrias em que a robótica tem uma posição bem definida, para situações em que os robôs são aplicados no processamento de produtos e existe equilíbrio técnico entre os robôs, os humanos, e outros tipos de automação, há uma variável a ter em conta, o volume de produção [15]. Em casos que o volume de produção de cada produto é pequeno, isto é, há uma constante alteração significativa dos produtos a vantagem recai sobre os operadores humanos, já em empresas que os volumes são enormes a vantagem recai sobre tipos de automação menos flexíveis, deixando para a robótica o intervalo entre estes dois extremos (ver Figura 2.6).

O facto de a robótica se implantar no intervalo de volume de produção médio não significa que este seja um grupo reduzido de aplicações, muito pelo contrário. Os mercados atuais têm vindo a desenvolver tendências que favorecem a robótica.

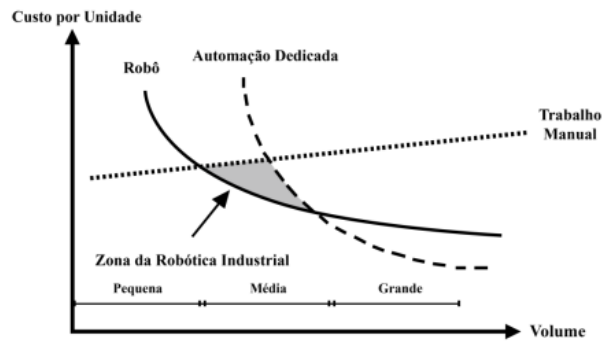


Figura 2.6: Relação entre os volumes de produção e o custo de cada solução [15].

A elevada concorrência, produtos cada vez mais personalizados, o ciclo de vida dos produtos cada vez menor e a exigência de melhor relação qualidade/preço são características do mercado que fazem com que as empresas evitem produzir para *stock*, isto é, produzem principalmente em média escala ao ritmo do surgimento das encomendas [15]. É por estes motivos que pode ser observado um constante aumento de robôs na indústria nos últimos anos, prevendo-se que continue nos próximos anos (ver Figura 2.7).

### 2.2.1 O que é um Robô Industrial?

A palavra robô deriva de uma peça teatral escrita por Karel Capek em 1921, como já foi descrito na secção anterior. Na altura apenas se referia a trabalho feito por máquinas mas hoje esta palavra refere-se a um conjunto de equipamentos. Geralmente é feita confusão entre os termos robô e manipulador, no entanto são palavras com significado completamente diferente. Robô é o conjunto dos equipamentos que compõem um sistema robótico (ver Figura 2.8), ou seja, um robô é composto pelo seu **controlador**, por um ou mais **manipuladores** (estrutura

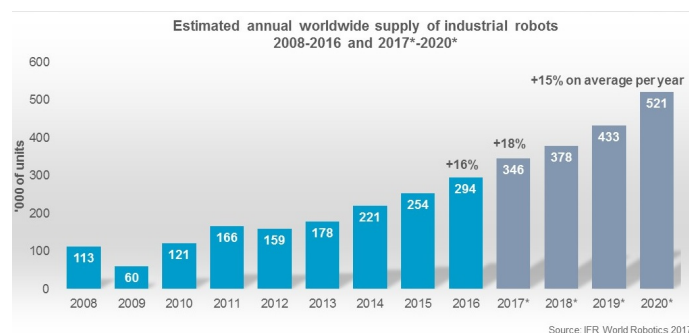


Figura 2.7: Estimativa do número de robôs fornecidos à indústria nos últimos 10 anos e até 2020 [16].

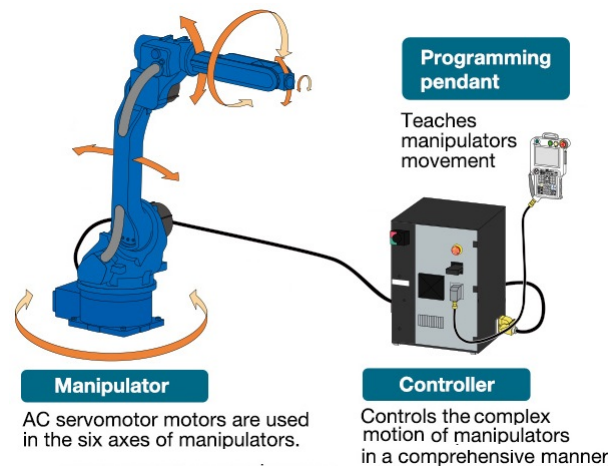


Figura 2.8: Principais constituintes de um robô [18].

mecânica e atuadores) com pelo menos um **atuador final** cada, e opcionalmente pode ser incluída a **consola de controlo e programação** [17] [18]. Neste sistema há ligação física entre todos os componentes e estes comunicam entre si para o correto funcionamento do mesmo.

A unidade de controlo, ou controlador, pode ser descrita como o "cérebro" do sistema pois é deste equipamento que resulta todo o sincronismo e lógica do robô previamente programado [17]. Normalmente este equipamento é um armário que contém no seu interior o CPU, os módulos de comunicação, os módulos de E/S digitais e analógicas, os módulos de segurança e, para a interação com o manipulador possui os *drivers* de potência que forem necessários de acordo com o número de eixos e quantidade de robôs a serem utilizados. Em alguns modelos há ainda suporte para *drivers* adicionais para atuação de motores externos, oferecendo a possibilidade de minimizar ou eliminar o uso de um PLC na solução robotizada.

O manipulador, ou braço robótico, é o componente mais popular destes sistemas principalmente por ser um dos pontos onde mais se concentram as tarefas realizadas pelo robô. É composto por vários sensores de força, *resolvers* (leitura da posição) e atuadores que podem ser de diferentes tecnologias e têm como responsabilidade posicionar o atuador final para que este possa desempenhar a sua tarefa. Este atuador final, por vezes chamado de ferramenta, fica acoplado na extremidade do punho do robô (último eixo) e é o componente que mais varia de solução para solução por ser o ponto de contacto com os produtos a manusear, sendo necessária adaptação ao formato e textura de cada um.

Por fim, a consola de controlo e programação, por vezes também referida como *Teach Pendant*, é o ponto de contacto entre o robô e um operador. Esta pode ser utilizada para programação, memorização de pontos, calibração de ferramentas, consulta de mensagens e estado do sistema e alteração de configurações, no

entanto nem todas as soluções necessitam destas funcionalidades no seu normal funcionamento. Atualmente um programador pode desempenhar estas funções a partir de um PC com o *software* relativo ao robô instalado. Depois de programado o robô e as máquinas envolvidas arrancam apenas com a atuação de um botão, nunca sendo necessárias novas configurações. As mensagens de erro e de estado podem ser reunidas num PC industrial que reúne numa rede de comunicação vários robôs numa fábrica.

### 2.2.2 Manipuladores Robóticos

O manipulador é, como foi descrito anteriormente, o componente do sistema robótico responsável pelo posicionamento e descrição de trajetórias do atuador final. Na grande parte das aplicações os braços robóticos assemelham-se a um braço humano, são constituídos por seis eixos e é por este motivo que um robô com um manipulador deste tipo classifica-se como um “Robô de seis eixos”. No entanto nem todos os manipuladores de seis eixos se assemelham a um braço humano, devendo-se isto ao tipo de juntas que o compõe. Outro ponto a referir é que nem todos os manipuladores possuem 6 eixos, já que existem aplicações que não exigem tanta liberdade no posicionamento e nas trajetórias. Há também variáveis referentes ao tipo de atuação do braço e variáveis que se prendem com a robustez mecânica, pelo que podem ser listados diversos manipuladores com características diferentes. Através da análise dos pontos fortes e fracos de cada um resulta uma indicação das aplicações que são mais favorecidas pelo uso de um manipulador com características específicas.

A estrutura de um manipulador pode ser dividida em dois principais tipos de elementos, as juntas e os elos. A relação entre uns e outros dá-se ao longo do braço em que os elos são unidos pelas juntas (ver Figura 2.9).

São estes elementos que vão caracterizar um manipulador quanto aos seus

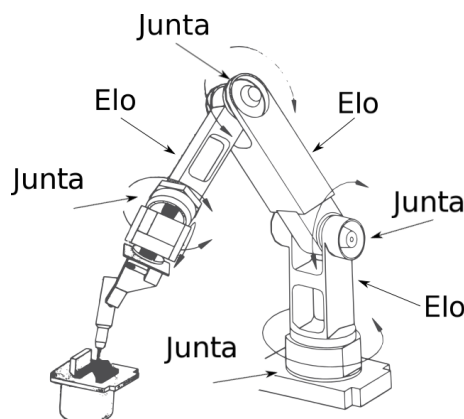


Figura 2.9: Elos de ligação e juntas num robô [19].

graus de liberdade. O método de atuação das juntas tem influência na rapidez, na capacidade de carga, na precisão e na repetibilidade do manipulador. No que toca a estes dois últimos aspetos, não é só o método de atuação que os influencia, há que ter em conta o importante papel desempenhado pelo algoritmo de controlo que, de acordo com a sua complexidade, pode tornar um manipulador mais ou menos preciso ou repetitivo.

Um manipulador, tal como qualquer outro sistema elétrico ou mecânico, tem as suas características físicas, particularidades no algoritmo de posicionamento e imprecisões. Estes três pontos relacionam-se na maneira como o manipulador se irá posicionar no espaço e a qualidade desse posicionamento será tanto melhor quanto essa relação. Estas características intrínsecas ao manipulador prendem-se com as suas limitações físicas [20]. Um robô que esteja a manipular a sua carga máxima não tem as mesmas características de quando se está a posicionar sem carga. Apesar de a robustez dos manipuladores permitir que sejam vistos na generalidade como corpos rígidos na prática nem sempre funciona assim, o peso do próprio manipulador, do atuador final e da carga que está a ser manipulada causam deflexões nos materiais que constituem o braço (ver Figura 2.10).

Este efeito agrava-se em robôs de grande porte, em que as cargas são elevadas e o comprimento dos elos é maior. Apesar de serem projetadas arquiteturas que evitem o agravamento destes efeitos, eles são inevitáveis ao nível milimétrico, pelo que cabe ao algoritmo de posicionamento ter a particularidade de incluir um modelo de deflexão e compensar esse desvio. De forma a melhorar um pouco mais esta compensação podem ainda ser incluídos sensores de deformação capazes de medir cargas e deflexões. Devido às características físicas do manipulador, o algoritmo de posicionamento deve ainda contemplar dados referentes às acelerações nos eixos de forma a serem reduzidas vibrações, principalmente quando o manipulador descreve trajetórias com velocidades elevadas.

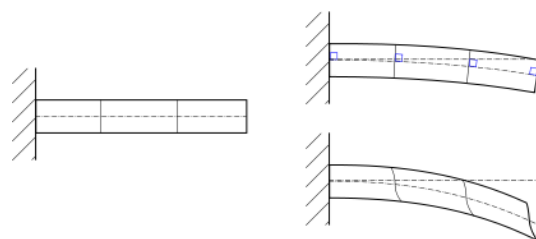


Figura 2.10: Ilustração da deflexão de uma viga de aço para pequenas forças (em cima) e para grandes forças com deformação (em baixo) [21].

### 2.2.2.1 Elos de Ligação

Os elos podem ser caracterizados do ponto de vista de cada junta adjacente. Do ponto de vista de uma junta, o elo que vem da base é chamado de *elo de entrada* e o elo que está mais próximo do atuador final é chamado de *elo de saída* [22].

Apesar dos esforços reunidos no projeto destes elementos para que estes sejam ao mínimo deformáveis, é inevitável que estes apresentem algum grau de flexibilidade quando submetidos a grandes esforços, tanto de flexão como de torção.

Neste sentido são evitadas certas técnicas na conceção dos elos. As ligações no elo com recurso a parafusos, apesar de simplificarem o processo de produção de formas complexas introduzem no elo um ponto muito suscetível à deflexão. Para fabricação dos elos é dada preferência a estruturas soldadas e fundidas, tornando estas peças menos suscetíveis à deformação. No entanto, muitas das vezes é necessário recorrer a processos secundários de produção, como maquinação para retificação da peça e tratamento térmico para remoção de tensões internas, aumento da dureza e resistência mecânica e para melhorar a resistência à corrosão e ao calor. Nestas peças, os materiais mais usados são o alumínio e o aço, mas nestes últimos tempos têm também sido empregues em novos tipos de robôs materiais como a fibra de carbono e de vidro, plásticos reforçados e termoplásticos [20].

### 2.2.2.2 Juntas

Uma junta de um robô é normalmente composta por 4 componentes, a transmissão (ou engrenagens), o atuador, pelo menos um sensor de posição e o eixo. Em alguns robôs, são também incluídos sensores de velocidade e de binário.

Enquanto que o principal ponto fraco dos elos é a deformabilidade, o ponto fraco das juntas é o *backlash*. Esta característica das juntas, também conhecida por "*folga*" pode ser desenvolvida ao longo do tempo, com a evolução do desgaste dos próprios componentes da junta ou pelo facto de o robô trabalhar com cargas superiores às nominais. Noutras situações esta folga já é característica da junta, principalmente em situações em que a precisão não é importante e são usadas juntas com menor qualidade de projeto. Quer numa situação, quer noutra, a presença de folgas nas juntas é um fator determinante na precisão do robô. Grande parte das vezes este *backlash* provém das rodas dentadas presentes nas engrenagens (ver Figura 2.11), mas em caso de haver uso indevido do robô (colisões e excesso de peso) pode ser também motivado por empenos nos eixos das juntas e danos nos rolamentos.

Para a grande maioria dos robôs, são as juntas que permitem os movimentos rotativos ou lineares. Estas, são denominadas articulações rotacionais e prismá-

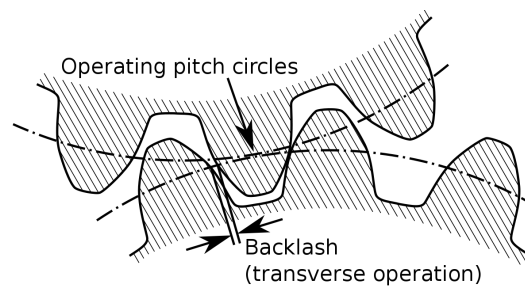


Figura 2.11: Exemplificação do *backlash* num conjunto de engrenagens [23].

ticas, respetivamente e fazem parte do grupo de elementos que classificam um robô. Existem ainda outros tipos de juntas que, embora menos populares que as anteriores, são por vezes utilizadas. São estas a junta esférica (ou bola de encaixe) e a junta de Cardan (ou junta universal/Hooke).

As juntas rotacionais (Tipo R) permitem o movimento de rotação de um elo de saída em relação a um elo de entrada cujos eixos se cruzam num ponto e podem ser divididas em três tipos (ver Figura 2.12):

- Junta Rotacional
- Junta de Torção
- Junta Revolvente

As juntas lineares ou prismáticas (Tipo L ou P) permitem o movimento linear entre os dois elos adjacentes e podem ser divididas em 2 tipos (ver Figura 2.12):

- Junta Linear ou Prismática
- Junta Ortogonal

### 2.2.2.3 Graus de Liberdade

Os graus de liberdade de um robô são uma característica que está relacionada com a capacidade de o manipulador realizar um determinado número de diferentes movimentos independentes uns dos outros e prende-se com o número de juntas com que o robô está equipado. Como tal é possível ter robôs com até sete graus de liberdade [25].

- **Um Grau de Liberdade:** Os robôs com apenas um grau de liberdade só são capazes de movimentar um objeto numa linha reta, normalmente não se vêem muitos robôs com esta característica.

- **Dois Graus de Liberdade:** Os robôs com dois graus de liberdade têm capacidade de trabalhar num plano X,Y, isto é, a duas dimensões. Por exemplo, um robô deste tipo pode pegar e elevar uma peça para depois a baixar e largar num ponto ao lado. Tudo isto realizado num plano a duas dimensões e sem alterar a orientação do objeto.
- **Três Graus de Liberdade:** Com três graus de liberdade os robôs já são capazes de realizar um grande leque de manobras, podendo realizar as mesmas tarefas dos robôs com menos um grau de liberdade mas com a capacidade de se movimentar em profundidade, ou seja, pode situar o atuador final no espaço X,Y,Z sem alterar a orientação do mesmo. Este tipo de robôs é muito utilizado em tarefas de *pick and place* em que não há necessidade de alterar a orientação dos objetos, como por exemplo na paletização camada a camada, em que o robô pega em camadas já apresentadas ao robô na posição que irão ficar na paleta.
- **Quatro Graus de Liberdade:** Adicionando um grau de liberdade aos robôs com três graus de liberdade, o mesmo fica capacitado de orientar o objeto que esteja a manusear sob um dos eixos possíveis (transversal, longitudinal ou vertical) (ver Figura 2.13). São muito populares na paletização por ser uma aplicação em que normalmente apenas há necessidade de rodar os objetos num plano horizontal.
- **Cinco Graus de Liberdade:** A diferença entre estes robôs e os anteriores está na capacidade de orientação do objeto, com mais um grau de liberdade, estes são capazes de orientar o objeto em dois eixos dos possíveis (ver Figura 2.13).
- **Seis Graus de Liberdade:** Tal como os robôs com quatro e cinco graus de liberdade, os de seis graus de liberdade movimentam um objeto sob o espaço X,Y,Z, mas são capazes de o orientar em qualquer um dos eixos, oferecendo

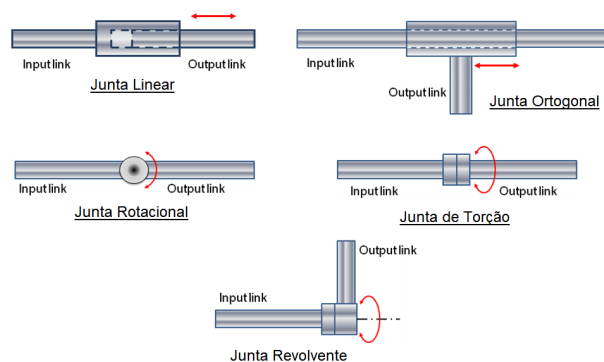


Figura 2.12: Ilustração dos vários tipos de juntas [24].

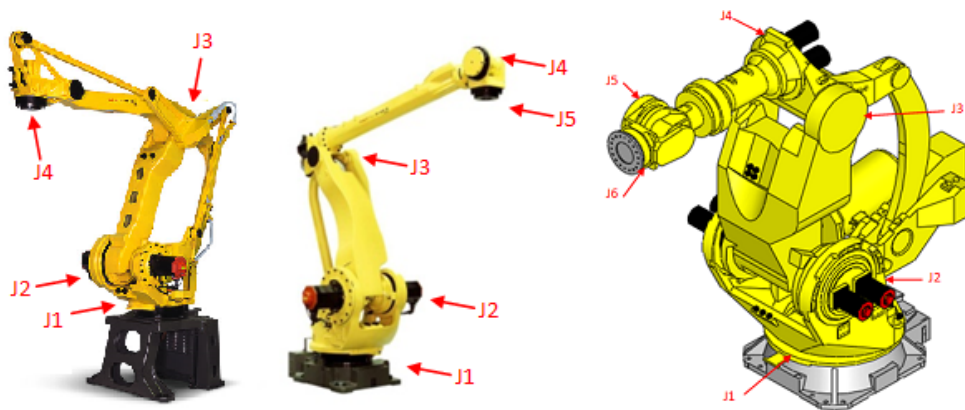


Figura 2.13: Robôs com 4, 5 e 6 graus de liberdade (esquerda para a direita) [25].

total flexibilidade nas suas tarefas (ver Figura 2.13). São os robôs mais populares ao nível industrial pelo facto de permitirem esta flexibilidade que lhes é intrínseca, deste modo, são capazes de orientar o atuador final de tal forma que o mesmo atuador final pode ter várias ferramentas que são utilizadas de acordo com a orientação do punho do robô.

- **Sete Graus de Liberdade:** Os robôs de sete graus de liberdade são semelhantes aos de seis graus de liberdade mas a este é adicionada a capacidade de movimentar o robô sobre um eixo que é normalmente horizontal mas pode também ser vertical. Com este eixo o robô torna-se capaz de se mover de um sítio para outro, oferecendo mobilidade e flexibilidade ao sistema em causa.

#### 2.2.2.4 Desempenho de um Robô

O desempenho de um robô é normalmente descrito através de variáveis que caracterizam o seu funcionamento. Entre outras é descrita a precisão do robô, a repetibilidade, a resolução, a velocidade, a aceleração, a capacidade de carga, a robustez do manipulador e o seu espaço de trabalho [20].

A precisão ou exatidão de um robô refere-se à capacidade de posicionar o robô no ponto do espaço que lhe foi indicado. Esta característica é especialmente importante em tarefas pouco repetitivas, como por exemplo numa tarefa de *picking* de produtos que aparecem num tapete de forma completamente aleatória ou em situações em que os pontos a atingir vão sendo indicados a partir de uma base de dados. Para se obter uma boa precisão é importante um algoritmo de controlo que contemple o máximo de variáveis que possam existir no sistema, como deflexões e pesos a serem manipulados. Por vezes os manipuladores têm um modelo de controlo que trabalha com o modelo de corpo rígido não contemplando algumas variáveis externas, e exigindo mais calibrações para se obter uma

boa precisão. Tipicamente a precisão dos robôs industriais varia entre  $\pm 10$  mm para robôs pouco precisos e  $\pm 0,1$  mm para robôs com modelos de controlo mais complexos [20].

A repetibilidade, por vezes confundida com precisão (ver Figura 2.14), é a capacidade de um robô repetir um posicionamento que foi feito anteriormente. A importância desta característica é dada principalmente em aplicações que incluem trabalhos repetitivos como empacotamentos e alimentação de peças a máquinas. Esta característica pode ser vista como o raio de uma esfera que inclui o conjunto de pontos para os quais o robô retorna depois de enviado da mesma origem, pelo mesmo programa, com a mesma carga e com a mesma configuração do braço. Em certas situações esta esfera pode nem incluir o ponto indicado como destino devido a erros na calibração, a limitações na precisão do próprio robô ou até por erros do controlador no arredondamento de cálculos. Esta especificação está também relacionada com o porte do robô, e varia entre 1 mm e 2 mm em grandes robôs e ronda os 5  $\mu$ m nos robôs de microposicionamento [20].

A resolução de um robô indica o menor incremento possível na sua posição. Esta característica é especialmente importante quando o robô é controlado por sensores que lhe indicam o posicionamento que deve tomar de seguida. Para calcularem os valores referentes a esta característica, alguns fabricantes tomam como base a resolução dos *encoders* ou dos servomotores, dependendo do tipo de atuação. No entanto este tipo de cálculo pode induzir em erro pois nestes cálculos

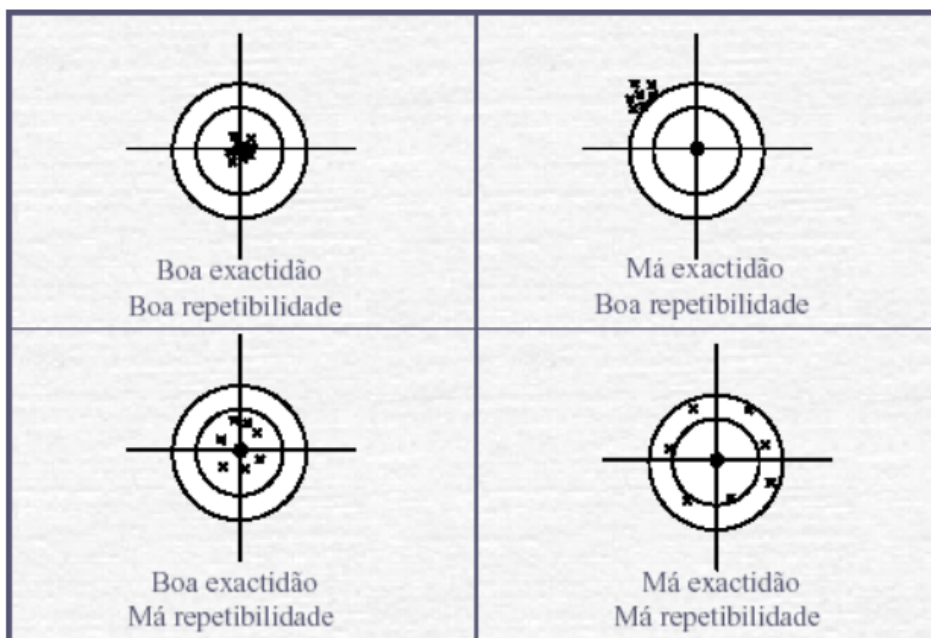


Figura 2.14: Diferença entre precisão e repetibilidade [26].

acabam por não serem contemplados fatores como o atrito entre juntas e elos e o *backlash* que possa existir nas juntas. Normalmente a resolução dos robôs está na faixa dos 0,001 mm a 0,5 mm, o que para a grande maioria das aplicações industriais é mais que o suficiente [20].

A capacidade de carga manipulável pelo robô é um aspeto que deve de ser devidamente analisado pois esta não é uma característica tão linear como as anteriores. A carga que pode ser acoplada ao punho do robô varia com a distância e com o alinhamento do centro de massa em relação ao punho, como é exemplo do ABB IRB 2400-10 que tem uma capacidade de 12 kg mas que varia dependendo da localização do centro de massa da carga (ver Figura 2.15). Nas especificações do robô são apresentados dois parâmetros referentes a este aspeto, a carga nominal e a carga máxima. A carga nominal refere-se ao máximo de carga que o robô é capaz de manipular à velocidade máxima sem perder a precisão indicada. A carga máxima refere-se ao máximo de carga que o robô é capaz de manipular a uma velocidade reduzida sem que se perca a precisão indicada. Grande parte dos robôs são ainda capazes de acoplar no seu braço alguma carga referente a equipamentos suplementares ao atuador final, como por exemplo caixas de eletroválvulas [26].

Normalmente a velocidade máxima do robô refere-se à velocidade máxima a que o atuador se pode deslocar no espaço, esta encontra-se limitada tanto pela capacidade de velocidade dos atuadores das juntas como pela capacidade máxima

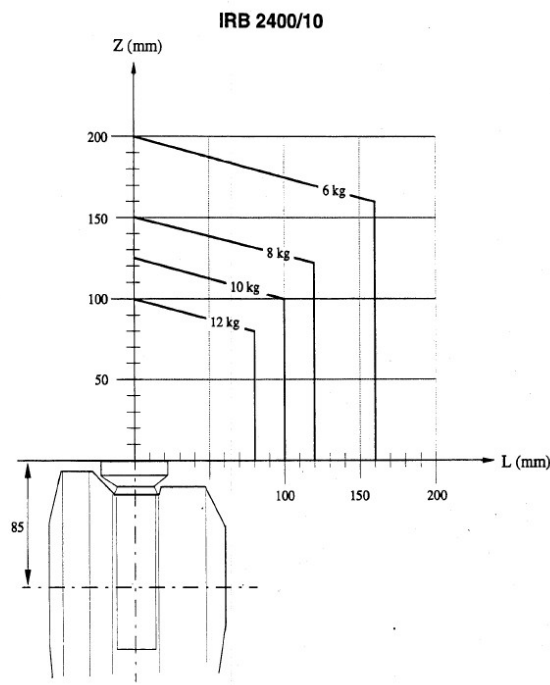


Figura 2.15: Diagrama de carga do robô ABB IRB2400-10 [27].

das cablagens de alimentação. Esta característica é tida como importante em situações em que os objetos manipulados não são frágeis e não há risco de estes se perderem durante o movimento do robô. Normalmente as velocidades máximas do atuador final não ultrapassam os 20 m/s, isto já em robôs de grande porte [20].

Para a grande maioria dos manipuladores a massa manipulada é substancialmente menor que a massa manipuladora, o que significa que o esforço para movimentar o manipulador é superior ao esforço para movimentar a massa, ou seja, a aceleração é um fator que terá de ser tanto maior quanto maior for o manipulador em si. A aceleração é um fator mais capaz de afetar o tempo de ciclo que a velocidade máxima já que a aceleração traduz-se na rapidez com que se atingem velocidades mais elevadas. É por isto que em projetos que se desejem cadências elevadas, manter acelerações elevadas é mais prioritário que ter capacidade de grande velocidade ou carga. No entanto, acelerações mais elevadas têm implicações ao nível físico do robô, exigindo estruturas mais rígidas que preservem a integridade do manipulador. Em alguns robôs de montagem de componentes leves as acelerações podem ultrapassar os 10 G [20].

Quando se refere a robustez do manipulador refere-se também a capacidade do manipulador resistir a colisões. No curso do seu trabalho podem acontecer situações imprevistas como colisões envolvendo o manipulador, as ferramentas ou objetos na sua vizinhança. Por vezes, estes acidentes podem provocar apenas danos no exterior, o que não depende apenas da velocidade do impacto, também depende do *design* do manipulador que poderá ser mais ou menos resistente a colisões. Normalmente estes danos estão relacionados com fraturas ou deformações nos elos, estragos ao nível dos dentes das engrenagens das juntas e outros danos mais particulares, como estragos na cablagem e nos fins de curso. Para evitar estas situações, no projeto dos robôs são incluídos sensores de sobrecarga nas juntas, tanto na força como na alimentação, e sensores de força no punho (ver Figura 2.16). Podem também ser incluídas superfícies almofadadas e desta forma tentar reduzir os danos provenientes de uma colisão [20].

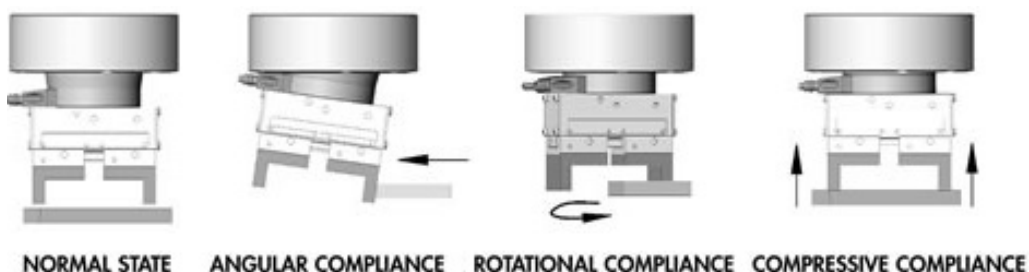


Figura 2.16: Sensor de forças externas a atuar no atuador final [28].

Outra característica dos robôs é o seu espaço de trabalho. Esta, é uma característica que está essencialmente relacionada com a estrutura física do manipulador. O espaço de trabalho, também chamado volume de trabalho é o conjunto dos pontos no espaço que o robô é capaz de atingir. Quanto maior é este conjunto de pontos, maior é a capacidade de o robô se posicionar de maneiras diferentes.

Nos robôs manipuladores comuns de juntas rotacionais, as juntas do ombro e do cotovelo são as que têm um papel preponderante na determinação do volume de trabalho [20]. Já as articulações do pulso têm menor contribuição para o volume de trabalho, na maioria dos casos estas juntas oferecem diferentes orientações para os mesmos pontos dentro do volume de trabalho. Como esta característica está intimamente relacionada com as juntas do robô, o espaço de trabalho será diferente não só de acordo com o tamanho do robô como também pelo tipo de juntas que o compõem. Em suma, o volume de trabalho de um robô depende sempre das características físicas do manipulador, isto é, do tipo de juntas e seus limites e do comprimento dos elos.

Por vezes, ao acoplar um atuador final o volume de trabalho aumenta, motivado por o robô ficar com maior alcance. Noutras situações o volume de trabalho acaba por ser reduzido pelo atuador final, devido a cablagens que não permitem que as juntas ultrapassem certos limites, ou devido ao tamanho do próprio atuador final que limita o movimento no espaço mais próximo ao robô. Existem ainda outros motivos que limitam o espaço de trabalho de um robô, como a proximidade a objetos e equipamentos na sua vizinhança, como transportadores e vedações.

Visto que o volume de trabalho é a zona onde o robô é efetivamente capaz de trabalhar sem limitações físicas este deve ser aproveitado da melhor maneira, adotando uma posição de instalação que favoreça a realização do trabalho numa zona dentro do volume indicado pelo fabricante. Nem todos os robôs oferecem essa possibilidade, mas o ABB IRB 140 é exemplo de um robô que pode ser instalado tanto no chão como na parede ou no teto (ver Figura 2.17).

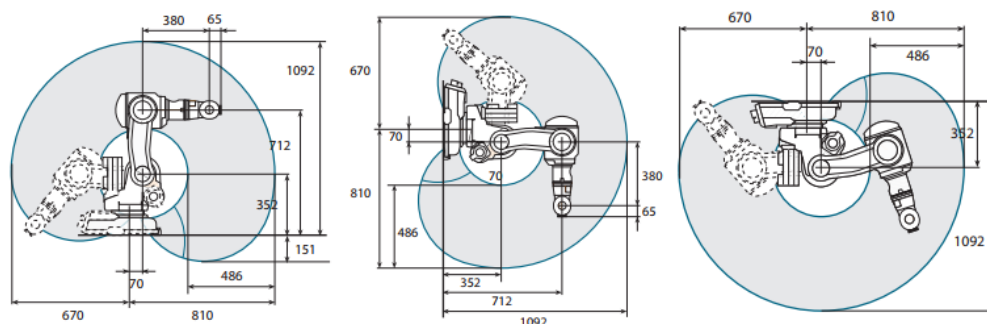


Figura 2.17: Posições de montagem possíveis para o ABB IRB 140 [29].

Atualmente, a indústria tem imposto como requisitos dos robôs elevada precisão, elevada repetibilidade, capacidade de movimento a grandes velocidades com grandes acelerações, volume de trabalho amplo, capacidade de integração do robô com outros equipamentos e sensores já existentes no ambiente fabril e um controle simples o suficiente para ser calibrado e afinado por um operador [26].

#### 2.2.2.5 Métodos de Atuação dos Robôs

Os atuadores dos robôs têm atuação diferente de acordo com as características exigidas pela solução em que são implementados. Existem três principais tipos de atuação: Pneumático, Hidráulico e Eletromagnético [20].

Os atuadores pneumáticos são normalmente encontrados em manipuladores simples principalmente por não haver grande controle no movimento entre dois fins de curso, portanto, o ponto forte destes atuadores é o movimento ponto-a-ponto. A simplicidade no controle e o baixo custo mantêm a pneumática como uma solução viável nestas situações em que os atuadores não são de grande porte [20].

A utilização da atuação hidráulica está atualmente em desuso, e a sua principal vantagem é a capacidade de manipular cargas extremamente elevadas. Neste tipo de sistemas os equipamentos são guiados por solenóides e servoválvulas e a grande desvantagem relaciona-se com o custo de aquisição e o custo de manutenção que são extremamente elevados. Outra desvantagem são fugas que originam perda de fluido e sujidade no ambiente em que estão instalados [20].

A utilização da atuação eletromagnética nos robôs tem-se popularizado nestes últimos anos, acompanhando a evolução dos motores elétricos mais eficientes, de dimensões mais reduzidas e de características mais seguras que permitem a sua utilização em atmosferas perigosas e agressivas. Este tipo de atuadores é rápido e preciso graças à possibilidade de aplicar técnicas de controle avançadas. Outra das vantagens é a ausência de ruído durante o seu trabalho. Como o custo destes equipamentos não é elevado, estes são uma alternativa viável aos dois tipos de atuação anterior. Normalmente, os motores utilizados são os motores passo-a-passo, os motores de íman permanente, e os motores sem escovas (*brushless*) também chamados de servomotores de corrente alternada. Uma das desvantagens destes sistemas prende-se com o facto de os motores não poderem ser aplicados diretamente no robô, há necessidade de utilizar transmissões que podem desenvolver folgas e limitar a precisão. Outra desvantagem é a incapacidade de travamento da junta quando o motor está inativo, sendo exigida a utilização de um travão para esse fim, carregando o manipulador com mais peso [20].

De modo a ser possível uma mais fácil comparação destes três tipos de atuação são confrontadas as suas características na Tabela 2.1.

Tabela 2.1: Comparação das principais características de cada método de atuação [26].

	Método de Atuação		
	<b>Eletromagnético</b>	<b>Hidráulico</b>	<b>Pneumático</b>
Controlo	Fácil	Moderado	Difícil
Velocidade	Alta	Moderada/Alta	Muito alta
Aceleração	Baixa/Moderada	Alta	Baixa
Precisão	Boa (Depende do tipo e estado da transmissão)	Boa	Má (só para nos fins de curso)
Repetibilidade	Boa	Boa	Boa
Capacidade Estática	Má. Depende da integração de travões	Boa	Moderada, devido à compressibilidade do ar
Ambiente de trabalho	Necessários equipamentos especiais para trabalho em atmosferas perigosas e agressivas	Fugas sujam o meio fabril	Poluição sonora das válvulas e compressores
Custo	Moderado	Elevado	Moderado

Além destas três fontes de energia na atuação do robô, pode ser considerada mais uma variável que classifica a atuação como direta ou indireta. A atuação indireta é utilizada devido ao facto de ser necessário compensar a falta de binário e atenuar o excesso de velocidade que são os pontos fracos dos atuadores eletromagnéticos nestas aplicações. Com a aplicação de técnicas de atuação indireta torna-se possível converter o excesso de velocidade em binário e, deste modo, preencher as lacunas que a atuação eletromagnética apresentava. Esta atuação indireta traduz-se, por exemplo, na aplicação de um conjunto de engrenagens ao motor elétrico. A atuação direta é mais comum nos atuadores hidráulicos e pneumáticos e, como o próprio nome diz, não requer adaptação de mais componentes para além dos próprios atuadores.

### 2.2.2.6 Classificação dos Robôs

Existem diversos aspetos pelos quais os robôs podem ser classificados. Podem ser distinguidos pela sua geração, pelas linguagens de programação que suportam, pela capacidade computacional do controlador, pela complexidade e capacidades do modelo de controlo, pela mobilidade da base, pela estrutura cinemática, pelo acionamento, pelos graus de liberdade e pela geometria do espaço de trabalho [26]. Como pode ser observado na Figura 2.18, os robôs podem ser distinguidos a um primeiro nível pela mobilidade da base, podendo ser considerados móveis, caso a movimentação não esteja confinada a um espaço, e são considerados fixos se for verificado o contrário. Como o objetivo desta análise se foca nos robôs fixos, os robôs móveis serão agora deixados de parte tendo em conta que estes robôs são essencialmente veículos guiados de forma autónoma (AGV).

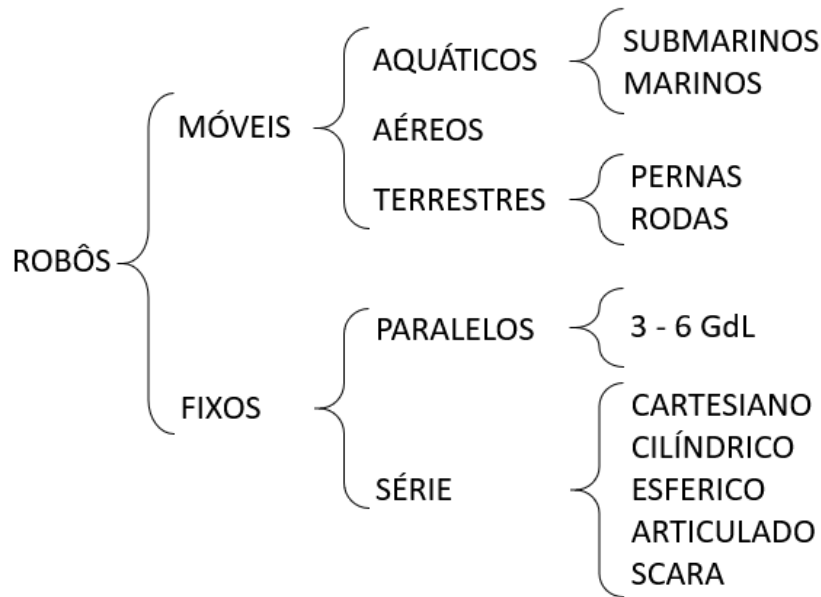


Figura 2.18: Classificação dos robôs de acordo com as suas características [30].

No que toca aos robôs fixos estes são classificados quanto à sua estrutura cinemática, sendo esta paralela ou em série. Os robôs de estrutura em série, por vezes também referidos de robôs de cadeia cinemática aberta, caracterizam-se pelo facto de os seus eixos e as suas juntas se sucederem ao longo do manipulador, isto é, cada junta encontra-se ligada a dois elos, assim como cada elo se encontra ligado a duas juntas, exceto os dos extremos do manipulador. Os robôs de estrutura paralela, por vezes também referidos como robôs de cadeia cinemática fechada, caracterizam-se pelo facto de as juntas não se encontrarem interligadas por elos, ou seja, estão em paralelo umas com as outras.

Estes robôs de estrutura paralela podem ser subdivididos de acordo com os seus graus de liberdade, sendo os mais comuns os de 3 e 4 graus de liberdade mas podem chegar a ter até 6 graus de liberdade. Esta é uma configuração de robô que, recentemente, se tem popularizado nas tarefas de *picking* com provas dadas ao nível das grandes velocidades de trabalho associadas a elevada precisão e repetibilidade. Devido à sua configuração muito particular (ver Figura 2.19) estes robôs são também chamados de "aranhões", normalmente são constituídos por eixos paralelos de rotação no plano vertical, mas podem também existir alguns equipamentos em que as juntas são lineares, embora sejam mais raros. Os mais comuns ao nível industrial são compostos apenas por juntas rotacionais e através destas oferecem ao robô 4 graus de liberdade, 3 para posicionamento no espaço (X,Y,Z) e 1 para a rotação do objeto sobre o eixo vertical, isto é, rodar o objeto.

Os robôs de estrutura em série podem ser divididos de acordo com a geometria do seu espaço de trabalho dando origem a 5 principais grupos, os robôs cartesianos, os cilíndricos, os esféricos, os articulados e os SCARA.

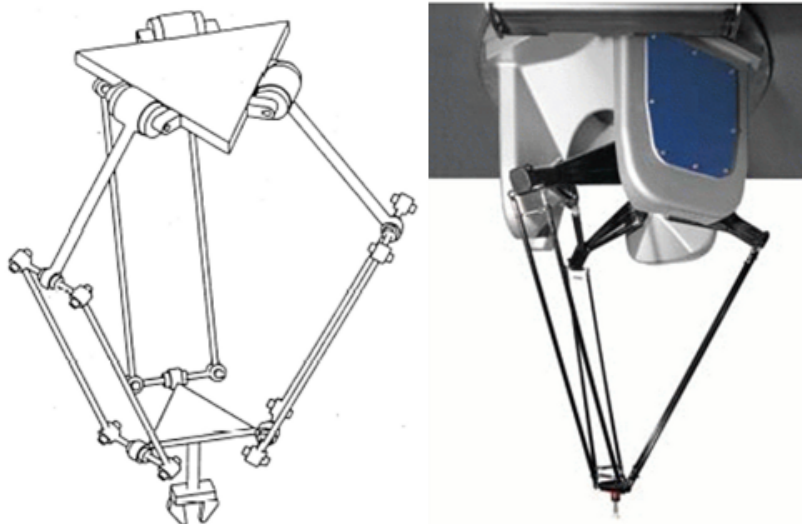


Figura 2.19: Ilustração da estrutura de um robô paralelo [32].

Os robôs de configuração cartesiana caracterizam-se por serem capazes de realizar com precisão movimentos lineares no espaço com um elevado grau de simplicidade no controle, originando um espaço de trabalho com o formato de um prisma retangular [31]. Este tipo de robôs é utilizado para alimentação de peças para máquinas, nomeadamente na colocação de peças em tornos CNC. Por serem compostos por três eixos de movimento linear têm uma configuração LLL. Existe ainda um grupo de robôs que pode ser observado como uma subdivisão destes últimos, os robôs com configuração em pórtico ou Gantry (ver Figura 2.20).

Os robôs com configuração cilíndrica têm, como o próprio nome o diz, um volume de trabalho de forma aproximadamente cilíndrica como resultado da conjugação de duas juntas lineares com uma junta rotacional [31]. Neste tipo de robôs a junta rotacional encontra-se logo depois da base e as seguintes são as do tipo linear, atribuindo a este tipo de robô a configuração RPP.

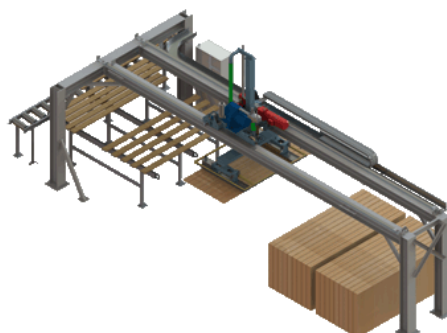


Figura 2.20: Ilustração de um robô com estrutura em pórtico [33].

Substituindo a segunda junta, que é linear, por uma junta rotacional é apresentado o robô com configuração esférica ou polar com uma configuração das juntas do tipo RRP [31]. Como resultado desta configuração é apresentado um volume de trabalho de formato aproximadamente esférico (ver Figura 2.21). Seguindo a ideia de transição entre as configurações anteriores, esta configuração também perde rigidez mecânica em relação à configuração anterior, a configuração cilíndrica, e ganha volume de trabalho em relação à mesma, mas, como é expectável, a complexidade do controlo também aumenta devido à introdução de mais uma junta rotacional [26].

A configuração mais comum a nível industrial é a chamada de configuração articulada ou revoluta [31]. Esta configuração aproxima-se da estrutura de um braço humano. Com uma configuração que tem os seus três eixos principais compostos por juntas rotacionais (RRR) este tipo de robôs apresenta um espaço de trabalho superior a qualquer robô de outro grupo (ver Figura 2.22) mas apresentando uma menor rigidez mecânica. Como resultado da integração de três eixos rotacionais, o controlo destes robôs torna-se complexo. No entanto as vantagens que este tipo de robôs apresenta são de tal forma valorizadas que estes são os robôs com mais aplicações estudadas devido à sua agilidade, sendo capazes de contornar obstáculos no espaço de trabalho, têm uma elevada relação espaço ocupado/volume de trabalho o que torna possível a sua instalação em locais com restrição de espaço e, mesmo assim, manter elevadas velocidades de trabalho [26].

Por fim, a configuração SCARA (*"Selective Compliance Assembly Robot Arm"*) que, como o próprio nome o indica, é um tipo de robô especificamente concebido para tarefas de montagem. Tipicamente composto nos dois primeiros eixos por juntas rotacionais no plano horizontal e no terceiro eixo por uma junta linear no plano vertical, este tipo de robôs tem uma configuração RRP (ver Figura 2.23)

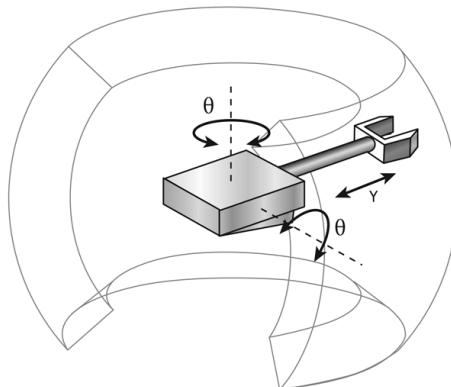


Figura 2.21: Ilustração de um robô com configuração esférica [34].

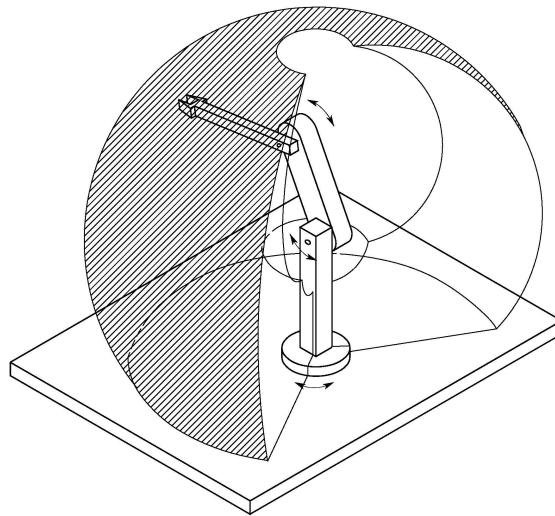


Figura 2.22: Ilustração de um robô com configuração articulada [35].

[31]. Por ter as duas primeiras juntas capazes de lhe conferir movimentos rotacionais no plano horizontal, este robô tem pouca rigidez nas direções X e Y, mas tem a vantagem de ser extremamente rígido em Z [26].

### 2.2.2.7 Punho

O punho do robô tem como função orientar a ferramenta para que esta desempenhe a sua função de modo adequado. Os movimentos possíveis do punho estão relacionados com os graus de liberdade do robô. Tipicamente, nos robôs que possuem total capacidade de orientação do punho, os que têm 6 graus de liberdade, o punho é composto por uma junta de torção e duas juntas de rotação. No eixo longitudinal é feita a rotação em torno do braço, no eixo transversal é feita a rotação do punho para cima e para baixo, por fim, no eixo vertical é feita a rotação do punho para a esquerda e para a direita [26].

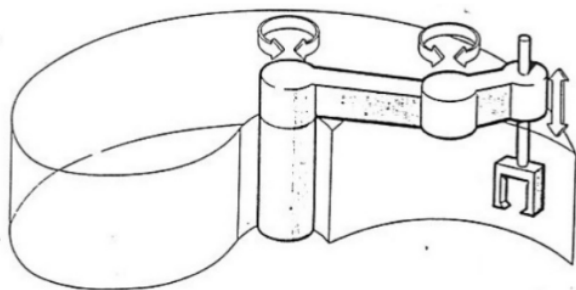


Figura 2.23: Ilustração de um robô com configuração SCARA [36].

### 2.2.2.8 Atuadores Finais

O atuador final é o ponto de contacto entre o robô e o objeto que está a ser manipulado, fazendo com que este elemento seja o que mais se altera de uma solução para outra, com o objetivo de uma melhor adaptação aos objetos que serão manuseados ou às tarefas que serão desempenhadas. Normalmente é feita uma distinção dos atuadores finais de acordo com a tarefa que é realizada, sendo denominados como atuadores finais de garras ou pinças para operações de manipulação e denominados de ferramentas quando a tarefa desempenhada pelo atuador final não envolve manipulação do objeto que está a ser trabalhado.

Os atuadores finais de garras podem ter arquiteturas muito distintas, de acordo com a tarefa que é desempenhada, sendo as mais comuns as garras mecânicas, as garras magnéticas e as garras de sucção. As garras mecânicas são compostas por duas ou mais pinças (ou dedos), capazes de abrir e fechar e de exercer uma força suficiente para segurar os objetos que estão a ser manipulados (ver Figura 2.24) [37] [38]. A sua atuação pode ser pneumática, elétrica ou hidráulica, cada uma com vantagens e desvantagens similares às descritas no subcapítulo 2.2.2.5 “Métodos de Atuação dos Robôs”.

No que toca às garras magnéticas, estas aplicam-se no manuseamento de objetos que tenham partes metálicas a partir da utilização de um eletroímã ou de um ímã permanente. Estas têm a vantagem de se adaptarem facilmente a vários formatos de peça, tendo tempos rápidos para pegar nas mesmas, no entanto por vezes pode haver deslocamento lateral durante as trajetórias do robô, principalmente se forem feitas acelerações agressivas [37] [38].



Figura 2.24: Exemplo de uma garra mecânica de atuação pneumática [39].

Em relação às garras de sucção, estas baseiam o seu funcionamento na criação de vácuo para manusear objetos planos, lisos e limpos e sem perfurações. A sua atuação pode ser feita com base numa bomba de vácuo ou com base no princípio de Venturi [38].

No que toca às ferramentas, estas assemelham-se às usadas por trabalhadores, diferindo nas adaptações necessárias para acoplamento no punho do robô. Deste modo o robô é capaz de realizar diversas operações, como por exemplo rebarbagem e polimento, soldadura por arco e por pontos, pintura, corte de materiais a LASER e a jato de água, furações e rebitagem (ver Figura 2.25) [38]. Aparentemente manusear a ferramenta em vez do objeto é mais fácil, no entanto o desgaste da ferramenta acrescenta grande complexidade no controlo do processo, já que há necessidade de reajustar os pontos de posicionamento do robô a cada ciclo que é concluído. Nem todos tipos de ferramenta estão suscetíveis deste desgaste, no entanto, quando o mesmo está presente é necessário incluir também um mecanismo de troca do material de desgaste, ou fazer paragem de produção para um operador realizar a troca do consumível.

## 2.3 Interfaces Industriais Homem-Máquina

A interface de um sistema é o espaço onde é feita a interação entre um operador e uma ou várias máquinas ou sistemas. Como resultado desta interação pretende-se que seja possível controlar o sistema em questão e que seja dado um *feedback* do seu funcionamento que facilite a tomada de decisões no controlo ou a monitorização do seu funcionamento. Até há relativamente poucos anos, a interface era feita através de botões e sinalizadores, mas a utilização destes elementos tem vindo a diminuir. Atualmente a interface apresenta-se de um modo mais gráfico, com a utilização de grandes painéis *touchscreen*, resultado da evolução tecnológica. Ao contrário dos botões e sinalizadores, estes painéis não se encontram necessariamente instalados na porta de um quadro elétrico, permitindo maior flexibilidade no seu local de instalação (ver Figura 2.26).



Figura 2.25: Tocha de soldadura acoplada a um robô (à esquerda) e robô de pintura na indústria automóvel (à direita) [38].



Figura 2.26: Painel industrial *touchscreen* SIMATIC IFP1900 produzido pela Siemens [42].

Com a utilização de monitores o leque de opções de *design* e de funcionalidades tornou-se extremamente amplo e, à semelhança do que acontece fora do meio industrial, qualquer pessoa que possua um *tablet* consegue fazer uma utilização do dispositivo mesmo sem formação específica. Isto porque as interfaces têm vindo a tornar-se cada vez mais intuitivas permitindo uma rápida aprendizagem e grande facilidade de utilização. No entanto há um fator que distingue uma interface industrial das interfaces comerciais, as consequências da ocorrência de uma falha no processo podem incluir riscos para os operadores, para o ambiente e para o próprio sistema. Posto isto, é importante que a interface seja concebida com metodologias que, além de a tornarem visualmente apelativa, dêem especial atenção à prevenção de falhas e redução dos erros que tenham como base a fraca utilização da interface. O facto de uma interface possuir um ambiente mais *user friendly* permite que seja feita uma análise que facilita a memorização e a perceção do enquadramento dos parâmetros nos seus valores de referência [43].

Os sistemas e máquinas que já possuam outras interfaces mais antigas não são, normalmente, alvo de *upgrade*, muitas vezes por se acreditar que os custos envolvidos não justificam o retorno, mas se for feita uma ponderação das vantagens envolvidas com a adoção de uma interface percebe-se então que, se projetada com base nos padrões descritos anteriormente, há uma maior taxa de deteção atempada de avarias, uma melhor perceção da avaria em questão (diminuindo o tempo de reparação) e uma melhor capacidade de tomar decisões referentes ao sistema com base na interface (ver Figura 2.27) [43].

Do ponto de vista do utilizador, uma interface com baixa qualidade é sinónimo de que o sistema que está a ser controlado é igualmente de baixa qualidade. Apesar de grande parte das vezes esta associação ser falsa, o sistema fica logo

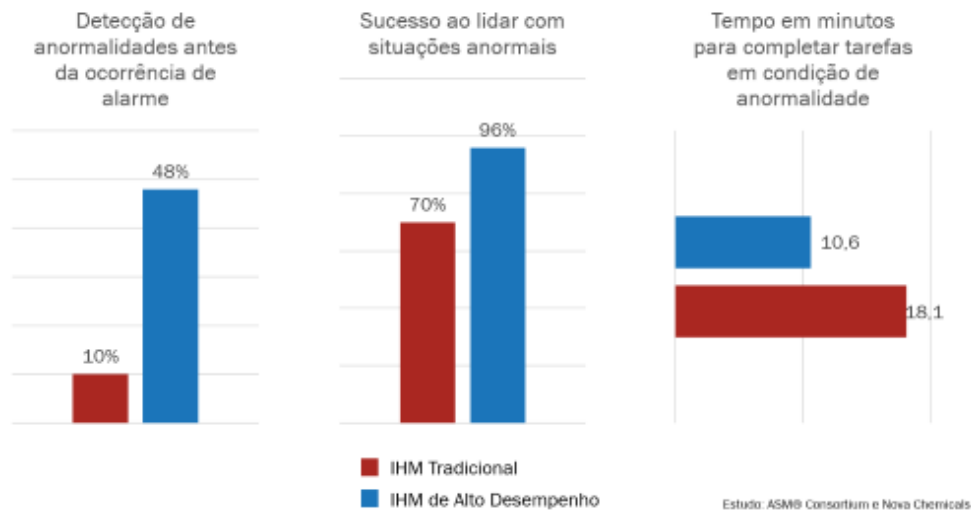


Figura 2.27: Resultados da comparação entre os interfaces mais antigos e os mais recentes [43].

comprometido pela falta de interesse que a interface irá gerar, exigirá também demasiada formação e, durante a sua utilização, irá gerar confusão e induzir em erro o utilizador. Com todas estas desvantagens a produtividade será severamente afetada e o investimento acabará por não gerar o retorno previsto [44].

Para definir a qualidade de uma interface existem quatro critérios que a mesma deve preencher, são eles a usabilidade, a experiência do utilizador, a acessibilidade e a comunicabilidade [44]. Dependendo da finalidade da interface, esta deve dar especial ênfase aos critérios que sejam mais relevantes, nunca esquecendo os outros.

O critério de usabilidade encontra-se interligado com o critério de experiência do utilizador, sendo descrito pelas normas ISO 9241-11 e ISO/IEC 25010 como o grau em que um produto é utilizado para uma finalidade específica para atingir objetivos específicos com eficácia, eficiência e satisfação. Esta última, resulta da avaliação individual da utilização e dos resultados por parte de um conjunto específico de utilizadores. É neste critério que são incluídos fatores como a aprendizagem, a memorização, a segurança no uso e a satisfação do utilizador [45] [46].

No que toca à acessibilidade, esta refere-se à ausência de barreiras que impeçam a boa interação com a interface. Dar ênfase ao critério de acessibilidade significa permitir que o máximo número de pessoas possam interagir com a interface, independentemente das suas características psicomotoras [44]. Nas interfaces industriais são, normalmente, mais relevantes as características psicológicas, principalmente em situações críticas.

O critério de comunicabilidade está dependente de como a interface é projetada, devendo seguir uma lógica que permita o utilizador perceber os eventos

no processo e na interface, na ótica de que se o utilizador perceber as intenções de quem concebeu a interface irá perceber o seu funcionamento e será capaz de fazer uma utilização produtiva da mesma. Por vezes para tornar mais fácil a comunicação são usadas analogias ou objetos gráficos [44].

Para que a interface seja bem desenvolvida o principal foco deve ser mantido no utilizador, o processo é relegado para segundo lugar. Com isto quer-se dizer que, tudo o que está presente na interface deve ser dirigido ao utilizador, à sua boa compreensão e à sua rápida e eficaz ação sobre o processo pois, apesar de no normal funcionamento a utilização de uma interface se revelar eficaz, o mesmo pode não ser observado em situações críticas, de elevado grau de importância, de pressão e de emergência. Em analogia pode ser considerada uma estrada pavimentada com asfalto de baixa qualidade, em que na maioria das situações permite uma utilização eficaz, mas assim que se verifiquem condições adversas surgirão problemas de segurança graves. Assim, a interface torna-se num dos principais focos de acidentes por falha humana e características como facilidade na aprendizagem, facilidade de memorização e taxas de erros reduzidas em muito contribuem para evitar acidentes [43] [44].

Um conhecido exemplo que realça a importância da interface num sistema é o acidente que ocorreu numa refinaria da BP América, no estado do Texas a 23 de março de 2005. Uma série de explosões aconteceram quando era feito o arranque de uma das unidades do processo. Após as investigações, o relatório apresentado indicava, no ponto 3.4, um fraco interface do sistema como uma das causas do sucedido. A interface, que se encontrava distante do sistema a ser controlado, distribuía a informação por vários ecrãs (ver Figura 2.28), não possuindo um resumo geral que permitisse ver o processo como um todo. Como consequência, os operadores não foram capazes de perceber uma falha numa transferência de fluídos assinalada num dos ecrãs, mas não no de arranque (o balanço entre entrada e saída de fluído na unidade não era apresentado, apenas eram mostrados dados do caudal de saída). Um acidente com causas semelhantes ocorreu no Reino Unido em 1994 [47].

De modo a evitar situações de risco, a interface tem também a função de manter o operador com consciência do que está a acontecer, isto é, manter a sua consciência situacional. Manter a consciência situacional é sinónimo de dizer que o operador está ciente do que se está a passar no processo, havendo sintonia entre aquilo que é percecionado e aquilo que está realmente a acontecer. Como indica a Figura 2.29, a consciência situacional envolve três aspetos:

- Perceção
- Compreensão
- Projeção

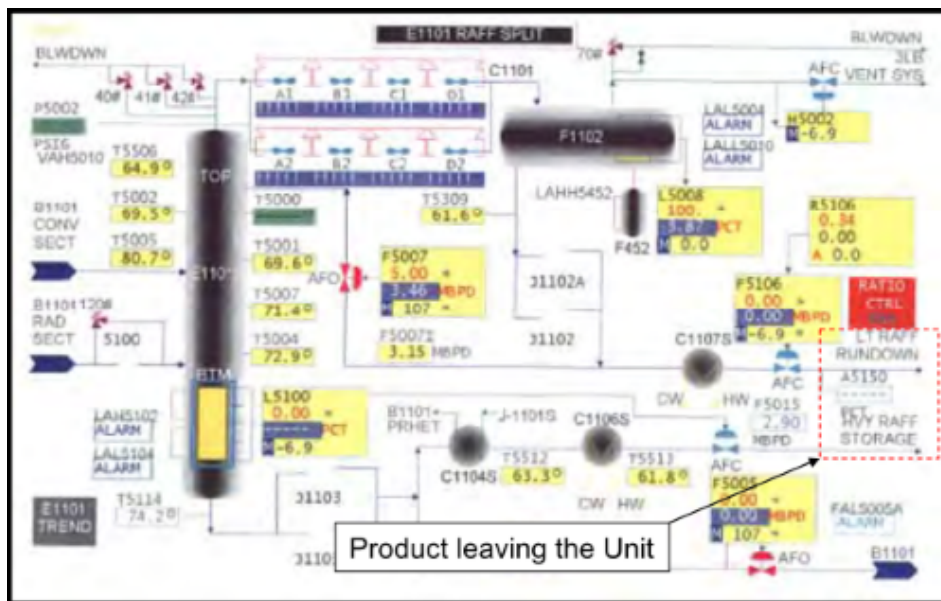


Figura 2.28: Ecrã apresentado durante o arranque que não exibia dados acerca da entrada de fluido, apenas da saída [47].

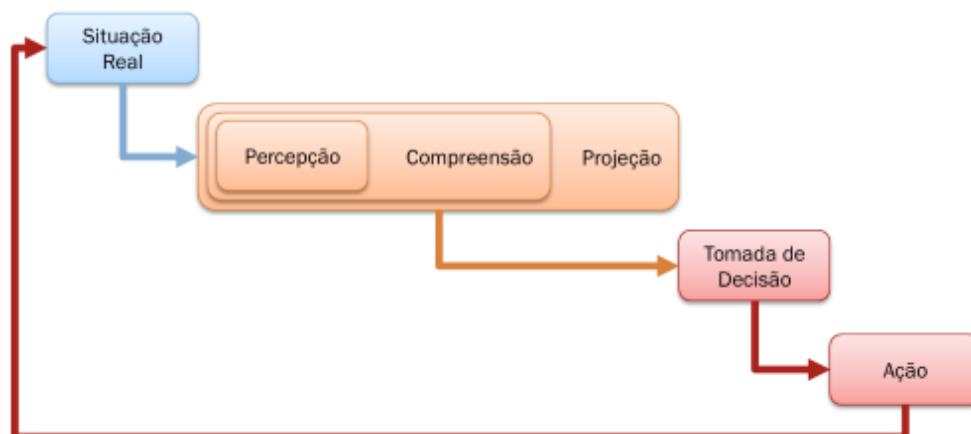


Figura 2.29: Esquema que ilustra as relações entre os aspetos da consciência situacional [43].

A percepção é a fase em que o operador faz a leitura dos dados diretamente da interface. A compreensão é a fase em que o operador compara a leitura com os parâmetros predefinidos para aquele indicador do processo, percebendo se a situação está dentro da normalidade ou não. A projeção é a fase final na qual o operador prevê as consequências resultantes dos dados analisados no seu contexto de modo a agir em conformidade com a situação [43].

A associação destes aspetos pode ser feita pelo operador ou pela interface, isto é, se a interface for capaz de apresentar o enquadramento dos parâmetros nos seus valores de referência e projetar o que está implícito com a atual leitura, os erros que possam resultar desta associação são eliminados. Para que a interface dê a percepção do valor é apresentado o dado bruto, no entanto, como este é o primeiro dos três aspetos, não nos apresenta uma contextualização nem aviso sobre eventuais inconformidades. Para corresponder ao segundo aspeto da consciência situacional, a compreensão, são apresentados na interface os limites entre os quais o parâmetro se deve manter, preferencialmente de um modo mais gráfico, que permita esta percepção logo numa análise inicial. Exemplo disto são os gráficos de barras ou manómetros com faixas de cor. Por fim, para se obter a projeção dos dados podem ser adotados vários métodos, sendo um dos mais populares a utilização de um gráfico que ilustre a evolução no tempo do parâmetro em questão (ver Figura 2.30) [43].

Por fim, é importante realçar um dos pontos que já foi referido, a capacidade de memorização rápida. Este é um importante ponto a explorar pois nem sempre a atenção é uma constante e é também uma mais valia para a realização de um *check up* rápido do sistema. Para este fim, importa inicialmente seleccionar a informação principal que se deseja transmitir pois numa análise rápida o utilizador apenas retém cores, formas e, eventualmente, alguns números e palavras que chamem a atenção. Por este motivo, o ecrã deve evitar excesso de cor e objetos visto que os mesmos irão reduzir a sua eficiência informativa, pelo que, quando não há necessidade de se saber o valor exato de um conjunto de sensores ou processos recomenda-se a utilização de agrupamentos mais analógicos, tornando-se mais fácil de, por exemplo, identificar uma leitura de um sensor que disperse do valor registado pelos seus redundantes, como se pode verificar por observação da Figura 2.31 [43].

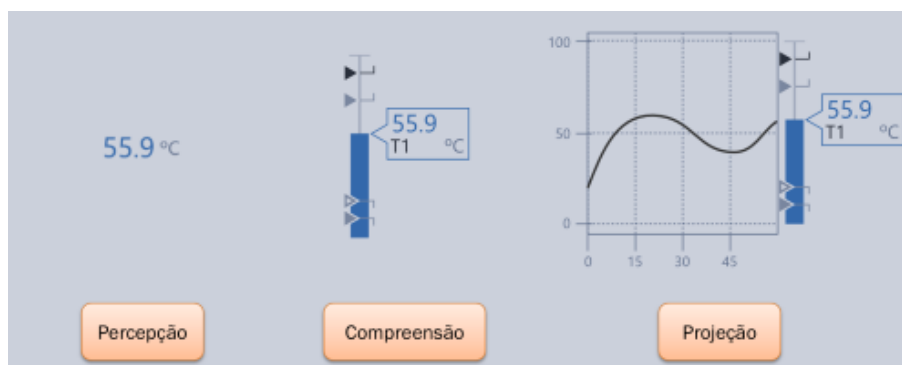


Figura 2.30: Ilustração dos aspetos da consciência situacional [43].

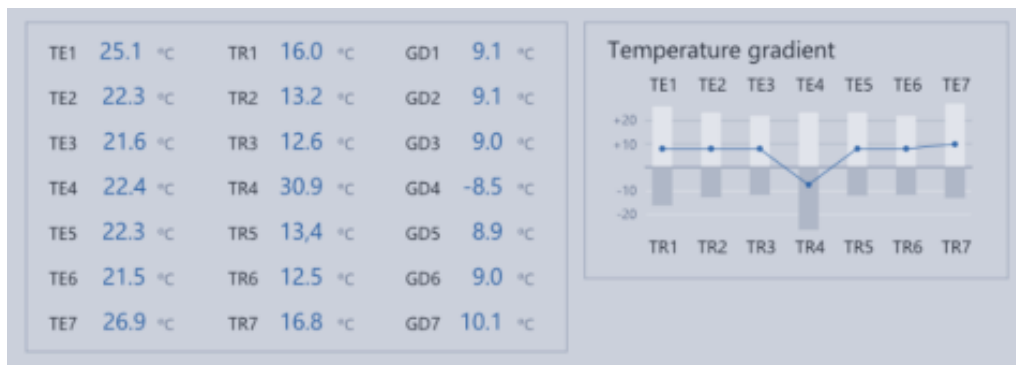


Figura 2.31: Diferença entre a apresentação dos valores brutos e uma representação analógica [43].

Atualmente encontram-se no mercado diferentes modelos de dispositivos que permitem escolher a melhor opção para cada sistema. A Siemens divide os seus dispositivos de interação em 3 gamas: “Basic HMI”, “Advanced HMI Panel-based”, “Advanced HMI PC-based”. A primeira gama disponibiliza dispositivos muito simples e de tamanho reduzido, aplicados principalmente em pequenas soluções. A gama “Advanced HMI Panel-based” já disponibiliza dispositivos de alta *performance* que permitem visualização de vários processos e são capazes de diversas funcionalidades, tais como a integração de vários painéis ao longo de uma ilha e a capacidade de trabalhar com base num PC industrial. Em termos de aspeto, a gama “Advanced HMI Panel-based” é semelhante à anterior, no entanto estes dispositivos já vêm com um PC industrial integrado. Estes dispositivos são recomendados para soluções que sejam particularmente exigentes ao nível de configuração e ao nível da quantidade e do tipo de informação que é processada. Com estes dispositivos é dada a possibilidade de criação de um servidor e de diversos clientes ao longo da ilha, comunicando diretamente entre si [48].

Para o desenvolvimento da interface, a Siemens oferece o *software* WinCC que é tratado como um *add-on* do TIA Portal. O WinCC é o *software* utilizado para o desenvolvimento de todas as aplicações para os dispositivos de interface da Siemens, tanto nos painéis mais simples como nos mais complexos, baseados num computador industrial e com um servidor e vários clientes.

No que toca à programação de sistemas baseados em computador, o WinCC subdivide-se em duas vertentes, a versão Advanced e a versão Professional. A versão Advanced permite, no seu pacote base, controlo por rede, comunicação com bases de dados, alertas, mensagens, relatórios e alarmes configuráveis com possibilidade de serem guardados na memória do PC e, com a aquisição de alguns pacotes de funcionalidades, permite ainda a programação com recurso a *scripts* Visual Basic [49]. A versão Professional possui todas as funcionalidades da anterior, incluindo na sua versão base várias funcionalidades que na versão anterior

são extras (por exemplo a programação com recurso a *scripts* Visual Basic) e, além de disponibilizar um maior leque de ferramentas para a criação da interface, possibilita a criação de sistemas com vários utilizadores ligados por rede, tanto ao longo de uma grande linha de produção como em localizações diferentes com recurso a plataformas *on-line* (ver Figura 2.32) [50].

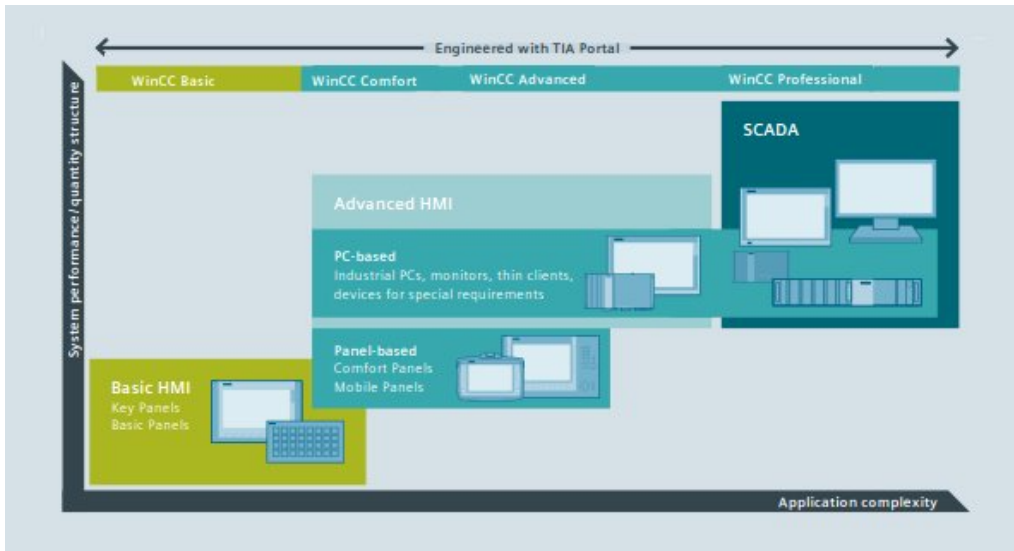


Figura 2.32: Principais diferenças entre as versões do WinCC [51].

## Capítulo 3

---

# O Processo de Paletização

---

Na grande maioria das empresas, a distribuição dos produtos finais é feita através de paletes, onde os produtos são organizados logo após saída da linha de produção e, deste modo, torna-se mais fácil de realizar as transferências para o armazém de produtos acabados, para o transporte e para os retalhistas que os comercializam. A paletização dos produtos não é um processo independente dos processos anteriores, nomeadamente o de produção, nem independente dos processos que lhe sucedem (ver Figura 3.1). O projeto da organização dos produtos na paleta encontra-se condicionado pela produção na medida em que o formato dos produtos e a maneira como os mesmos são embalados condicionam as quantidades que uma paleta é capaz de suportar, influenciando o seu custo de armazenamento e distribuição. Após a paletização há a necessidade de armazenar e transportar as paletes, deste modo as dimensões e o peso das paletes estão dependentes da capacidade dos armazéns, do tipo de meios de transporte utilizados e da capacidade dos mesmos.

Por estes motivos a paletização tem ganho especial importância na indústria, sendo alvo de otimizações tanto ao nível da organização dos produtos como ao nível da criação da paleta, tendo como objetivo a redução dos custos nestes processos, de modo a tornar o preço final de cada produto mais competitivo.

É com o objetivo principal de reduzir custos, mas também de aumentar a qualidade, que se aplicam soluções robotizadas na paletização. No entanto, estas aplicações robotizadas, como tantas outras, são sempre alvo de uma melhoria contínua, e são constantemente estudadas novas metodologias principalmente com vista a melhorar as cadências, tornando esta numa área da robótica industrial em constante evolução.

Numa ilha de paletização podem existir um ou mais robôs, dependendo da metodologia adotada, no entanto algumas tarefas que os robôs desempenham são



Figura 3.1: Ilustração dos processos anteriores e subsequentes à paletização [52].

transversais, como o apanhar e posicionar dos produtos, um de cada vez ou em grupo e o apanhar e posicionar dos intercalares. O apanhar e posicionar da paleta nem sempre é feito por robôs, sendo grande parte das vezes apresentada a paleta na sua posição com recurso a transportadores.

Para formar uma paleta devem ser seguidas as características pretendidas para a mesma (ver Figura 3.2). Uma paleta é caracterizada pelos seguintes fatores:

- **Produto paletizado** - As características do produto que é paletizado, como as dimensões e consistência em muito influenciam as características da paleta. É por isso importante saber os valores do comprimento, da largura, da altura, e do peso.
- **Paleta utilizada** - Em relação à paleta utilizada é importante ter em conta as dimensões e o peso da mesma. É ainda importante saber o peso máximo suportado pela mesma, no entanto este fator não é relevante na criação da paleta.
- **Distribuição de paletes** - Quando se fala de distribuição de paletes refere-se ao número de paletes/bases que compõe a paleta e entre que camadas estas se localizam. Apesar de não ser comum a utilização de mais que uma paleta, por vezes esta característica é importante pois permite uma maior

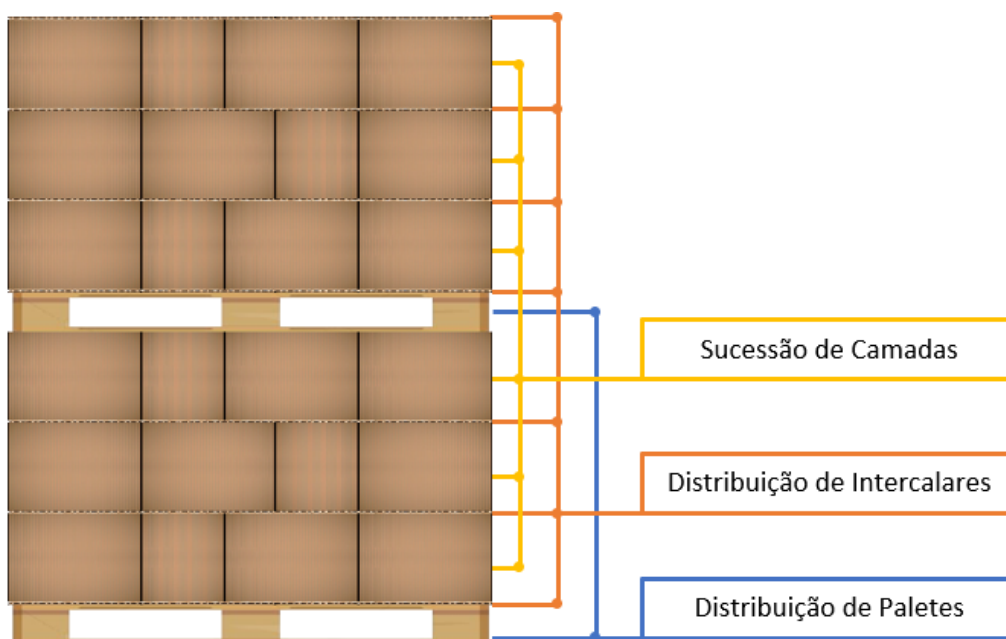


Figura 3.2: Ilustração das principais características de uma paleta.

cadência na ilha e uma maior eficiência na distribuição dos produtos pelos retalhistas.

- **Intercalar utilizado** - A utilização de uma placa ou película que separe uma camada da outra é frequente, permitindo que as camadas assentem melhor umas por cima das outras, resultando numa paleta mais estável. As dimensões dos intercalares, o peso e o material que os compõe têm importância para se perceber a altura que a paleta terá com a inclusão de um determinado número de intercalares, bem como o peso que resultará dos mesmos.
- **Distribuição de intercalares** - A distribuição dos intercalares é uma característica semelhante à distribuição de paletes, no entanto, ao contrário desta, a distribuição de intercalares ao longo das camadas da paleta é muito frequente.
- **Artigos por camada** - O número de artigos numa camada está, normalmente, condicionado pelas dimensões da paleta, no entanto, em casos muito particulares esta característica pode estar condicionada pela capacidade de carga do robô que manipula as camadas.
- **Número de camadas** - O número de camadas está normalmente restrito por um de quatro fatores, a estabilidade da paleta, a altura máxima

permitida pelo armazém, a altura máxima permitida pelo meio de transporte, ou o peso máximo permitido para a paleta.

- **Mosaico das camadas** - O mosaico das camadas é o padrão através do qual os produtos estão organizados ao longo da camada, sendo o objetivo deste mosaico organizar tantos produtos quanto os possíveis na camada mantendo uma estrutura estável para a paleta (ver Figura 3.3). Estes mosaicos nem sempre são iguais ao longo das camadas, existem situações em que a sucessão de camadas segue uma organização par/ímpar espelhada ou rodada 180°. Assim a camada seguinte e a anterior são diferentes da atual segundo este critério, com objetivo de aumentar a robustez da paleta. Existem ainda situações em que se pretendem que todas as camadas sejam diferentes, não seguindo uma lógica de sucessão específica.

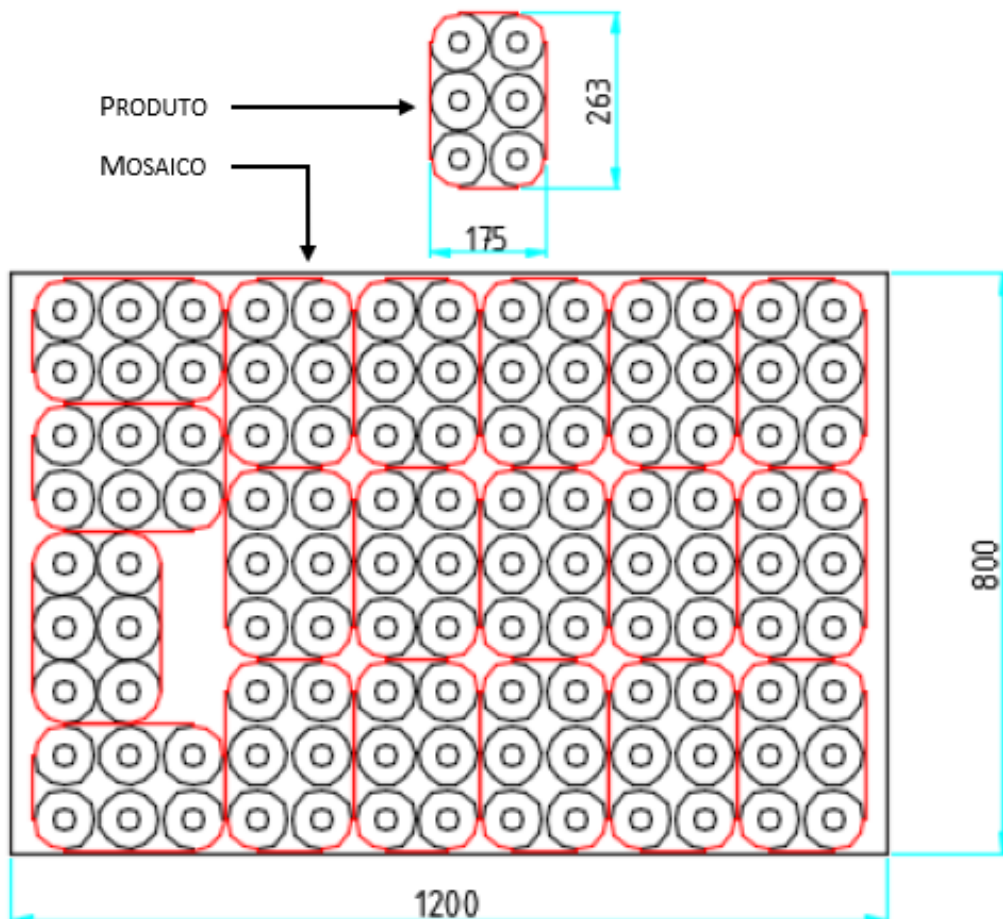


Figura 3.3: Exemplo do mosaico adotado para um determinado tipo de artigo.

## 3.1 Robôs de paletização

### 3.1.1 Manipuladores

Normalmente, numa ilha de paletização os robôs são os focos principais de atenção, sendo estes os responsáveis por grande parte dos ganhos de cadência e flexibilidade da solução. Nestas aplicações os robôs desempenham essencialmente funções de *pick and place* (apanhar e posicionar), tanto de produtos como de intercalares e até paletes, dependendo do projeto.

Partindo do princípio que, na paletização todos os objetos manipulados são apresentados ao robô sob um plano horizontal e têm como destino de posicionamento um plano paralelo ao de *picking* (igualmente horizontal), pode ser dito que as exigências ao nível de graus de liberdade do robô nunca irá ultrapassar os quatro graus de liberdade, já que, para além dos três graus para posicionamento no espaço, apenas será necessário orientar a rotação do objeto sob o eixo longitudinal. É por estes motivos que os fabricantes tipicamente apresentam robôs indicados para soluções de paletização com 4 eixos.

Por vezes são também utilizados robôs de seis eixos, muitas vezes por questões de flexibilidade que são exigidas ao robô, no entanto estas opções entre o número de eixos do robô têm várias implicações. Normalmente é feita uma associação direta: aos robôs de quatro eixos é associada a grande velocidade de trabalho e a obtenção de cadências elevadas; aos robôs de seis eixos é associada a grande flexibilidade para, por exemplo, ser possível utilizar um atuador final com múltiplas funções. No entanto existem outras características que pesam na altura de escolher qual o robô a utilizar (ver Tabela 3.1). Os robôs de seis eixos têm vantagem sobre os de quatro eixos quando se trata da facilidade na programação e alteração do *layout* envolvente, sendo que num robô de seis eixos torna-se mais fácil editar a programação na medida em que estes têm maior capacidade de posicionamento do manipulador, oferecendo a possibilidade de contornarem objetos que estejam dentro do espaço de trabalho. Apesar de menos flexíveis e com menor repetibilidade, os robôs de 4 eixos são capazes de se posicionar com uma precisão mais elevada, fator que, associado às suas capacidades de velocidade e aos menores preços por unidade os tornam na opção mais viável para o *pick and place* dos produtos a paletizar. Os robôs de seis eixos, apesar de continuarem com um preço por unidade mais elevado, têm vindo a apresentar uma tendência de descida à medida que a tecnologia se vai desenvolvendo e aperfeiçoando. A questão do preço é muitas das vezes preponderante, no entanto, assim que houver um equilíbrio de preço entre as diferentes opções, as escolhas serão feitas integralmente com base nas capacidades e características de cada modelo, permitindo respostas que se enquadrem perfeitamente nos requisitos dos clientes [53].

Tabela 3.1: Comparação entre robôs de 4 e 6 eixos [54] [55].




	ABB - IRB 660-180/3.15	ABB - IRB 6640-185/2.80	
Nº de eixos	4	6	
Repetibilidade (mm)	0.1	0.05	
Alcance (m)	3.15	2.80	
Carga máxima (kg)	180	185	
Limite	Eixo1	+180° a -180°	+170° a -170°
	Eixo2	+85° a -42°	+85° a -65°
	Eixo3	+120° a -20°	+70° a -180°
	Eixo4	---	+300° a -300°
	Eixo5	---	+120° a -120°
	Eixo6	-300° a -300°	+306° a -360°
Velocidade	Eixo1	130°/s	100 a -110°/s
	Eixo2	130°/s	90°/s
	Eixo3	130°/s	90°/s
	Eixo4	---	170 a -190°/s
	Eixo5	---	120 a -140°/s
	Eixo6	300°/s	190 a -235°/s

### 3.1.2 Atuadores Finais

Com tanta relevância como o tipo de manipulador existe o atuador final que se encontra acoplado ao punho do manipulador. O *gripper* é o elemento que mais varia de uma solução robotizada para outra, visto que o formato do mesmo tem uma relação muito próxima com as características físicas dos produtos que são manipulados pelo robô. O mais comum nas ilhas de paletização são os *grippers* de garras mecânicas, sendo que a sua atuação pode ser pneumática ou elétrica.

Muitas das vezes são também utilizadas garras de sucção, sendo que estas são mais sensíveis à variação das características dos produtos nomeadamente a textura e o peso. Como foi descrito na secção anterior, acerca dos manipuladores, os robôs de seis eixos oferecem grande flexibilidade e, como tal, é comum serem utilizados atuadores finais com diversos mecanismos de modo a serem desempenhadas tarefas distintas pelo mesmo manipulador. Com robôs de quatro eixos o mesmo pode ser possível, embora não seja tão comum. Nestes atuadores finais com várias funcionalidades podem mesmo existir mecanismos com métodos de atuação diferente, no entanto há sempre uma tentativa de manter todos os mecanismos com um método de atuação comum de modo a poupar na quantidade de cabos e ligações que têm de ser acoplados ao longo do robô até ao atuador final.

Como referido, os *grippers* são desenvolvidos com base no objeto a manipular, no entanto existem várias soluções *standard*. Sempre que possível são utilizadas soluções pré-concebidas com o intuito de poupar nos custos do projeto mas, quando não existe uma boa adaptação a estas soluções, são então desenvolvidos atuadores finais mais específicos. Normalmente, estes atuadores finais projetados propositadamente para uma solução apenas diferem no formato das zonas de contacto com os objetos, seguindo quase sempre a ideia base dos mecanismos mais comuns.

Os *grippers* de garras mecânicas podem adotar dois principais modelos. Para a manipulação dos objetos individualmente ou em grupos reduzidos é comum a utilização de um *gripper* de chapas, em que a atuação do mecanismo que faz com que os objetos sejam agarrados tanto pode ser pneumático como elétrico. Neste modelo de *gripper*, quando a atuação é pneumática, é comum que uma das chapas se movimente e a outra seja fixa (ver Figura 3.4). Nestes casos, o ajuste do



Figura 3.4: *Gripper* de atuação pneumática com uma chapa fixa [56].

*gripper* para diferentes tipos de produtos passa a ser feito manualmente no próprio *gripper*, ajustando a posição da chapa fixa, e muitas das vezes é ainda utilizado um mecanismo numa das chapas que segura o produto por baixo, consequência da ausência de um controlo eficaz da força aplicada. Para se obter um melhor controlo da abertura e do fecho da garra e das forças que são exercidas sob o produto é mais indicada a atuação elétrica, com recurso a servomotores. De modo a melhorar a superfície de contacto do *gripper* com os objetos, muitas das vezes, procede-se ao revestimento das chapas com uma camada de borracha ou de silicone. Deste modo, além de se aumentar o atrito entre a garra e o produto, passa a haver uma tolerância a variações mínimas na dimensão dos produtos. Muitas das vezes, de forma a tornar o robô capaz de realizar várias tarefas diferentes, este tipo de atuador final inclui ainda um mecanismo para apanhar e posicionar paletes e, se necessário, pode ser ainda incluído um sistema de sucção para apanhar e posicionar intercalares, como pode ser observado na Figura 3.5 [57].

A segunda abordagem às garras mecânicas apresenta semelhanças com a anterior no seu método de atuação, no entanto, esta, não utiliza chapas para apanhar os produtos. Neste modelo de *gripper* é utilizado um conjunto de pinças que se fecham por baixo dos objetos que se estão a manusear [57]. Deste modo, este *gripper* é capaz de manusear grandes grupos de produtos com pesos consideravelmente elevados e também demonstra ser capaz de manusear produtos deformáveis com peso elevado, como por exemplo sacos de cacau (ver Figura 3.6). Por reunir estas capacidades, este tipo de *gripper* é muito utilizado em soluções de paletização em que esta é feita à camada e não ao produto. Nestes *grippers*, a inclusão de outros mecanismos para manipulação de intercalares ou paletes não é tão comum

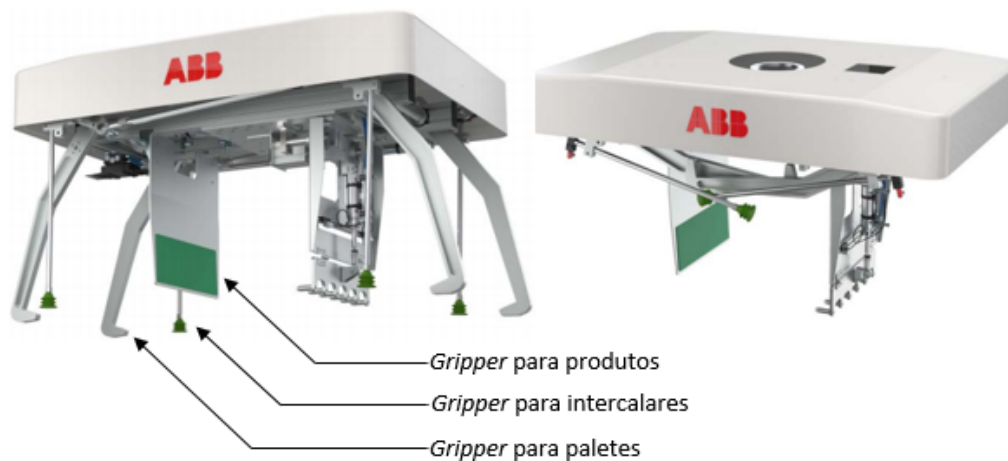


Figura 3.5: *Gripper* com as funcionalidades de agarrar produtos, paletes e intercalares [57].



Figura 3.6: Manipulação de sacos de cacau em pó [58].

como na abordagem anterior, no entanto, em soluções com um menor orçamento podem ser desenvolvidos *grippers* que permitam a realização dessas tarefas pelo mesmo robô [57]. Outra característica que pode variar nestes *grippers* é a linha de abertura, podendo ser linear ou em V, como ilustrado na Figura 3.7. Esta característica depende bastante dos produtos a manusear e se estes estão em grandes grupos. Caso o objetivo seja a manipulação de apenas um produto é comum utilizar-se uma linha de abertura em V, contribuindo para um agarrar mais suave. Caso se queiram manipular grandes grupos é utilizada uma abertura em linha, deste modo a organização dos produtos uns em relação aos outros não

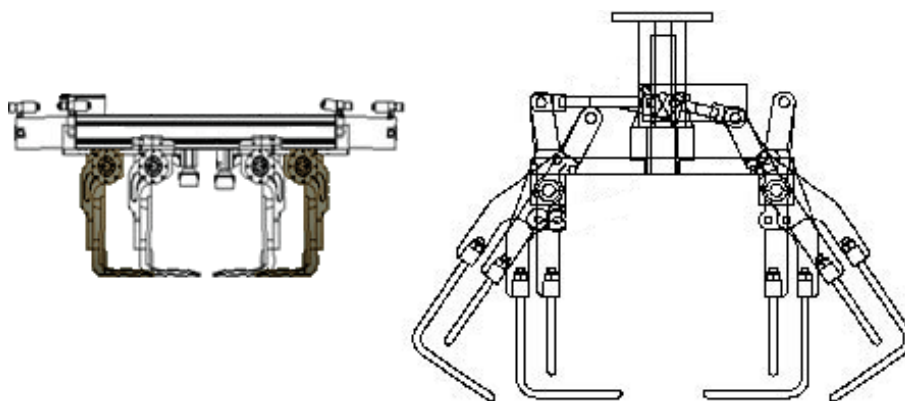


Figura 3.7: *Gripper* com movimento de abertura linear (à esquerda) e em V (à direita) [59] [60].

é afetada. Por fim, há ainda uma implicação que estas garras impõem, a base sobre a qual os produtos são apresentados deve possuir cavidades onde as pinças possam entrar para ficarem por baixo dos produtos, tais como transportadores de rolos ou mesas elevatórias.

Um dos modelos de *gripper* também muito utilizado é a garra de sucção. À primeira vista, estas garras dão a aparência de que na manipulação de produtos pode haver a perda do objeto durante a trajetória, principalmente pelo facto de os objetos apenas serem agarrados pela sua superfície superior (ver Figura 3.8), no entanto, com as ventosas a possuírem propriedades que lhes conferem capacidade de adaptação ao formato dos objetos, a manipulação de produtos é bastante eficaz. A utilização destes mecanismos permite manipular tanto produtos individuais como pequenos conjuntos de produtos e até camadas completas, sem ser necessário proceder a nenhum ajuste manual, cada ventosa pode ser ativada ou desativada individualmente, o que permite a adaptação ao tamanho e número dos produtos a serem manipulados sem perda de força de sucção. Outra das vantagens destes *grippers* é a sua rapidez ao agarrar e ao largar os produtos, permitindo a realização de tempos de ciclo curtos. No entanto estes *grippers* apresentam também um conjunto de desvantagens, nomeadamente a necessidade da inclusão de uma bomba de vácuo para gerar o efeito de sucção nas ventosas e outra das desvantagens é a sensibilidade a detritos que se encontrem na superfície dos objetos a manipular e a poeiras na atmosfera [62]. Apesar de existirem vários tipos de filtros para proteger o sistema dessas poeiras e detritos, quando esse

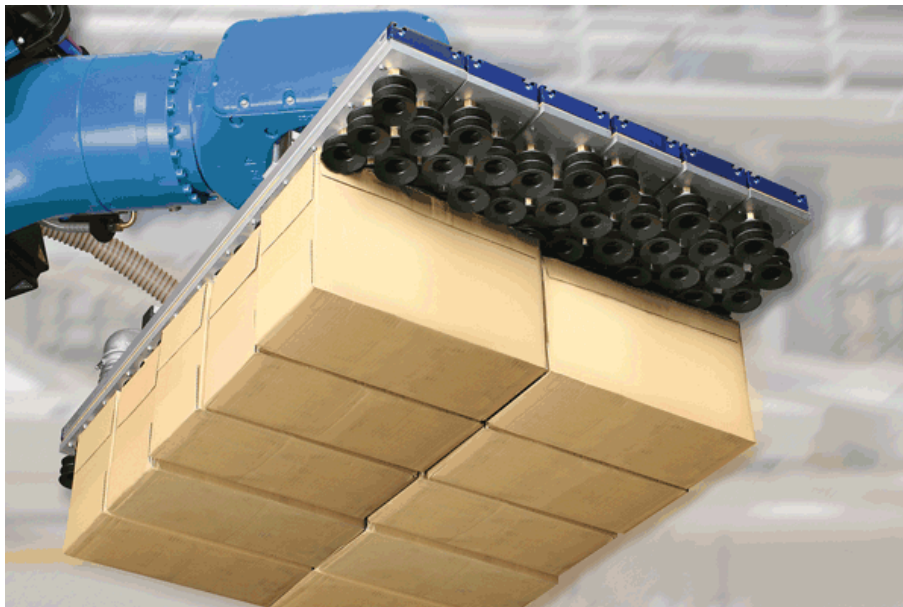


Figura 3.8: Garra de sucção a manipular caixas de cartão [61].

filtro fica saturado é necessário proceder à sua limpeza. Por último existe ainda a desvantagem do desgaste das ventosas, sendo que ao fim de um determinado número de ciclos é necessário trocar as ventosas do *gripper* para que o sistema mantenha o seu bom funcionamento.

Tal como nos *grippers* de chapas, estes também aparecem muitas vezes como *grippers* que associam vários mecanismos para realização de diferentes tarefas, conjugando a manipulação de intercalares, produtos e paletes num só *gripper* (ver Figura 3.9) [57]. De salientar ainda que, uma ilha de paletização que se baseie num robô com um atuador final deste tipo fica com uma cadência restringida pela utilização deste *gripper* multifuncional. No entanto existem situações em que uma opção deste tipo não afeta a cadência máxima de processamento visto que, por vezes, a cadência do processo que alimenta diretamente a ilha de paletização é inferior. Assim conclui-se que estas opções ao nível do *gripper* não dependem só dos objetos a manipular, mas também das cadências pretendidas.

### 3.2 Metodologias de Paletização

Como metodologias de paletização podem ser distinguidas algumas principais técnicas que são seguidas para que entrem na ilha de paletização as paletes (bases), intercalares e produtos, e saiam paletes formadas com os produtos dispostos na forma desejada e com uma melhor cadência, robustez e custo de paletização que as formadas pelos operadores. O fator principal para decisão da metodologia a adotar numa solução é a cadência desejada, ou a cadência necessária para que não haja acumulação de produtos à entrada da ilha. No entanto existem ainda outros fatores, como a disponibilidade financeira, que por vezes exige a redução do número de robôs que são equipamentos caros na solução apresentada, e muitas

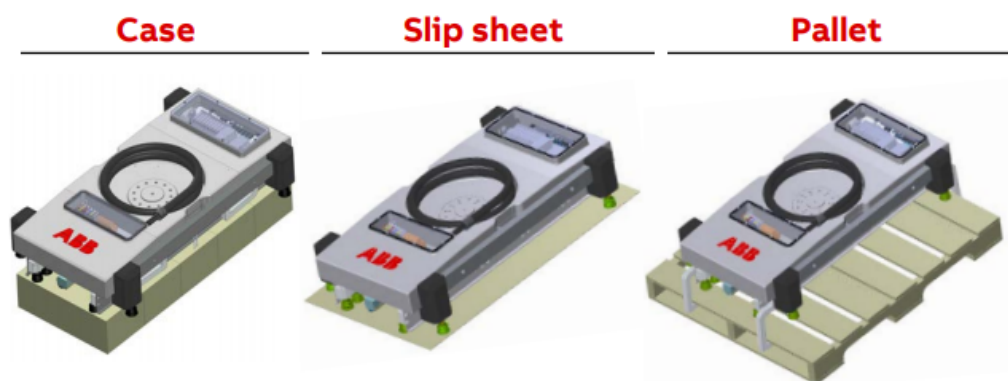


Figura 3.9: Garra de sucção com funcionalidades de apanhar intercalares e paletes [57].

vezes existe também o fator do espaço disponível para instalação da ilha.

Como metodologia mais básica é apresentada a paletização à caixa, ou ao produto. Como o próprio nome indica, nestas soluções o robô realiza ciclos em que nunca manipula mais que um produto em simultâneo, resultando numa formação de palete produto a produto. As ilhas que seguem esta metodologia são conhecidas por serem mais económicas em termos de preço de aquisição, e portanto, exigem uma grande flexibilidade dos equipamentos da ilha, nomeadamente do manipulador e do *gripper*, que precisam de desempenhar diversas funções, como o apanhar e posicionar da paleta, o apanhar e posicionar do intercalares e o apanhar e posicionar dos produtos, como é exemplificado pela ilha de paletização ilustrada na Figura 3.10. Muitas das vezes é ainda acrescentada a capacidade de o robô envolver a paleta em filme protetor [63].

Nas soluções que seguem esta metodologia são conseguidas cadências que rondam os 30 produtos por minuto, tendo tendência a reduzir de acordo com a distribuição das tarefas pelos equipamentos. Caso haja necessidade de colocar na ilha um robô que tenha as tarefas de posicionamento de paletes, intercalares e artigos e ainda a colocação de filme protetor, a cadência irá ser inferior à de uma solução em que o robô esteja com menos tarefas e as restantes distribuídas por outros equipamentos. Embora o *gripper* tenha menor custo de projeto e de produção, já que tem menos funcionalidades, é necessária a inclusão de outros



Figura 3.10: Implementação da metodologia de paletização à caixa numa ilha de paletização [64].

equipamentos, como transportadores. Estas opções trazem vantagens técnicas mas refletem-se no custo final da ilha e, muitas vezes, no espaço ocupado pela mesma, pelo que a ponderação tem de ser feita tendo em conta todos estes fatores [63].

Outra metodologia muitas vezes adotada é a paletização à fila. Tal como na metodologia anteriormente apresentada, nesta o robô coloca os produtos diretamente na sua disposição final na palete, a principal diferença está na capacidade de manipulação de vários artigos em simultâneo. Em termos práticos, esta manipulação de vários artigos em simultâneo restringe a maneira como os artigos são apresentados ao robô e restringe também o mosaico da palete. Nas ilhas que adotam esta metodologia, os produtos podem ser acumulados no final de um transportador uns a seguir aos outros, havendo apenas a necessidade de contar quantos já se encontram agrupados no final do transportador, formando uma fila (ver Figura 3.11). Assim que esteja formado o grupo pretendido o robô apanha o mesmo e posiciona-o na sua disposição final na palete, fazendo com que a orientação dos produtos desse grupo uns em relação aos outros não possa ser alterada. Assim, o mosaico da palete tem de ser desenhado a pensar nos grupos que provêm das filas formadas pelo transportador. No entanto, caso existam situações que exijam a manipulação de apenas um produto, as ilhas que sigam esta metodologia, também são capazes de o fazer, embora este seja um ponto que faz baixar a cadência do sistema [65].

É comum que as células que tenham esta metodologia implantada atinjam cadências entre 30 e 80 produtos por minuto (são capazes de realizar cerca de 10 ciclos com filas entre 3 e 8 produtos), no entanto, este é um indicador que está extremamente dependente das características de cada produto e do número de produtos manipulado por grupo. Caso haja possibilidade de manusear um número elevado de produtos por grupo a cadência será claramente beneficiada. Grande parte das vezes estas células apresentam apenas um robô mas, por vezes, o mesmo robô não está encarregue de fazer toda a palete, deixando o posicionamento de intercalares e de paletes para outros robôs ou outro tipo de mecanismo e desta forma melhorar a cadência da ilha [65].

A última metodologia a ser abordada é chamada de paletização à camada. As ilhas de paletização que seguem esta metodologia são geralmente vistas como soluções para taxas de produção muito elevadas. Nestas ilhas cada produto passa por várias fases até ser colocado na sua posição final na palete. Primeiramente é formada uma camada completa por um ou mais robôs, depois essa camada completa avança para ponto seguinte em que um robô a irá colocar na palete, enquanto isso já está a ser formada uma nova camada pelo primeiro robô ou conjunto de robôs (ver Figura 3.12). Nestas ilhas a colocação de intercalares fica atribuída ao robô que posiciona as camadas ou a um robô que apenas tem essa

finalidade. As paletes são apresentadas ao robô por meio de transportadores. Todos estes elementos a trabalhar em sincronismo são capazes de realizar ciclos que rondam as 7 camadas por minuto, dependendo em parte das características dos produtos e do mosaico que seja necessário formar [66].



Figura 3.11: Ilha de paletização que segue a metodologia de paletização à fila [65].

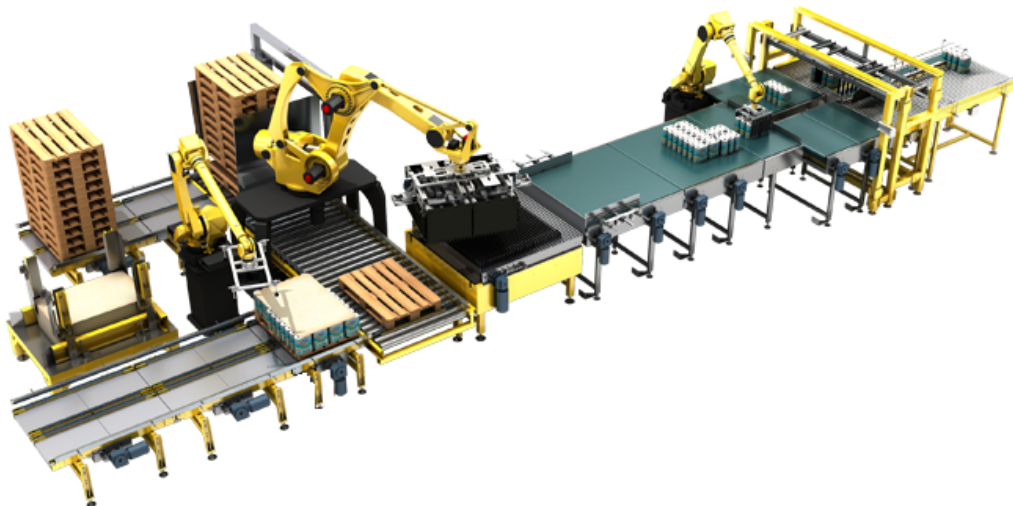


Figura 3.12: Ilha de paletização que segue a metodologia de paletização à camada [67].

Normalmente a aquisição deste tipo de ilhas surge da necessidade de escoar os produtos provenientes do fabrico com mais rapidez, tendo como maior prioridade as cadências elevadas, por isso é comum que o valor do investimento para este tipo de soluções seja bastante elevado dada a sua complexidade e a quantidade de equipamentos necessários para corresponder aos requisitos [66].

Cada metodologia tem as suas características particulares, traduzindo-se em vantagens e desvantagens. Como tal, a empresa que pretende investir num sistema de paletização dificilmente tem capacidade para ponderar todos estes fatores, pelo que, na grande maioria das situações, esta ponderação é feita pela empresa integradora que, com base na sua experiência, saberá melhor qual a metodologia a adotar de acordo com os requisitos que lhes são apresentados.

### 3.3 A Ilha de Paletização

A ilha de paletização é o conjunto de todos os equipamentos que fazem parte da solução automatizada de paletização. Assim, no *layout* de uma fábrica pode-se considerar a ilha de paletização como uma só secção sem qualquer necessidade de intervenção durante o seu normal funcionamento onde entram produtos, paletes e intercalares e saem paletes prontas para o armazenamento e transporte.

Embora existam sempre diferenças de uma ilha para outra, existe um conjunto de ilhas de paletização *standard*. Depois de projetadas estas soluções apenas se fazem ajustes que se relacionem com cada projeto, como o formato das garras, o tamanho dos robôs, e o *layout* dos transportadores. Assim, trabalhando sobre uma base previamente concebida, o desenvolvimento de novas soluções para diferentes clientes torna-se mais rápido e menos propenso a erros de projeto.

Numa ilha de paletização existem elementos que estão presentes independentemente da metodologia, como o robô e os transportadores de entrada e de saída, no entanto existem elementos que só aparecem em ilhas que adotem determinada metodologia. Em termos de equipamentos usados, as ilhas de paletização à camada são mais completas que as outras pelo que a análise dos seus equipamentos será feita primeiro.

Tipicamente, uma ilha de paletização à camada segue um *layout* idêntico ao ilustrado pela Figura 3.13, na qual se encontram numerados os principais equipamentos e pontos mais importantes.

- **1** - O transportador de *packs* para acumulação à entrada da instalação é responsável por trazer o *packs* do exterior para o interior da ilha.
- **2** - O transportador que aparece de seguida ao transportador de *packs* para acumulação à entrada da instalação é o transportador de indexação

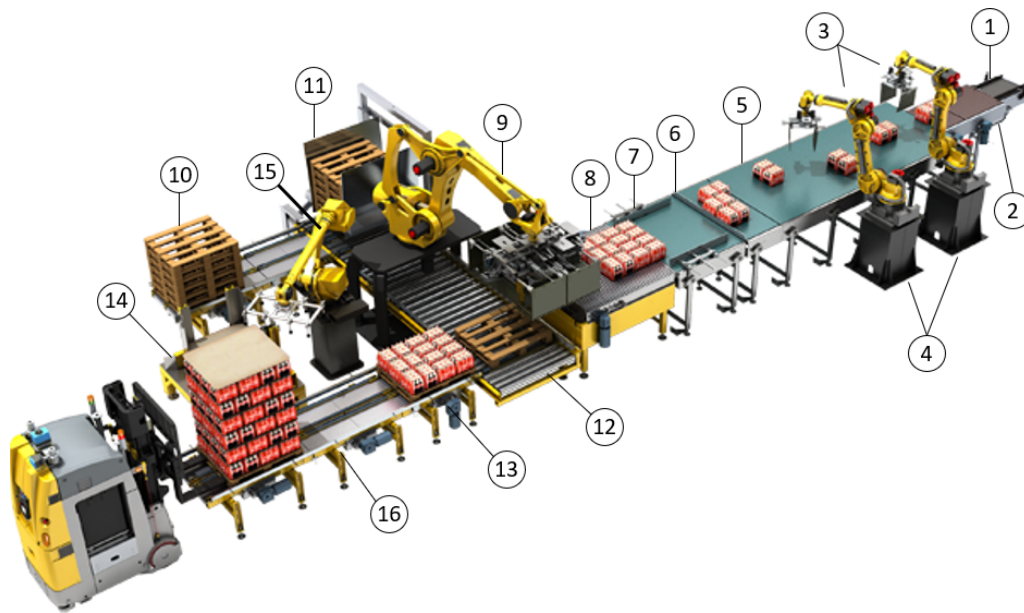


Figura 3.13: Ilha de paletização à camada [67].

de *packs*. Este transportador tem como função agrupar os *packs* nas quantidades que se pretendem de modo a que os robôs possam manipular vários produtos simultaneamente.

- **3** - Para manipular os *packs* é utilizado um ou mais robôs. Neste exemplo são utilizados dois robôs de modo a aumentar a cadência da ilha. Estes robôs, denominados de robôs de formação de mosaico, têm a tarefa de apanhar os *packs* que vêm do transportador de indexação e os posicionar no ponto do transportador que o levará à posição pretendida.
- **4** - Para que a área de trabalho dos robôs seja aproveitada da melhor maneira possível estes são elevados através da utilização de pedestais de aço. Com a utilização de um pedestal, o ponto de alcance máximo do robô fica situado à altura do transportador.
- **5** - O transportador de formação de camada é o transportador sob o qual os robôs de formação de mosaico trabalham. Como se encontra em contínuo movimento estes robôs têm de recorrer a *softwares* de *conveyor tracking*, trabalhando sob um referencial que se movimenta à velocidade do tapete. A única situação em que este transportador pára é quando ocorre um cenário em que os robôs de formação de mosaico não estão a ser capazes de organizar todos os *packs* que estão no transportador, assim que os robôs voltam a acompanhar o ritmo o transportador arranca novamente.

- **6** - Depois do transportador de formação de camada vem a mesa de acumulação de camada. Este transportador encontra-se constantemente em movimento e no seu final existe um batente. Com este batente os produtos que foram ordenados vão sendo acumulados uns a seguir aos outros, e assim que é completa uma camada o batente baixa e os produtos passam todos para o transportador seguinte.
- **7** - Após passarem pela mesa de acumulação de camada, os produtos passam para a mesa de *pré-picking* de camada. Este transportador tem dois batentes pneumáticos instalados nos seus extremos laterais, responsáveis por centrar a camada que vem do transportador anterior. Assim que se deteta que a camada está por completo neste transportador, o mesmo é parado e os batentes são atuados. Assim que os batentes são recolhidos o transportador arranca novamente e a camada já na sua organização final passa para o transportador seguinte.
- **8** - Já com o *layout* da camada formado, os produtos passam para a mesa de *picking*. Este transportador transporta a camada até junto do robô que a irá manipular.
- **9** - Para manipular a camada é utilizado um robô com capacidade de carga muito superior aos outros, já que a sua função é manipular uma camada completa, em muitas situações o peso da mesma está na ordem das centenas de quilogramas. Após apanhar a camada que lhe é apresentada na mesa de *picking* o robô realiza uma trajetória que leva a camada até à sua posição final na paleta. Tal como os primeiros, este robô também é normalmente instalado sob um pedestal de aço de modo a ser feito um melhor aproveitamento do seu espaço de trabalho.
- **10** - O local de entrada de paletes vazias é um ponto que deve de estar devidamente protegido por sistemas de segurança já que é normal existirem pessoas nessa zona. Este local deve também ser particularmente robusto visto que na maioria das vezes as paletes são ali colocadas com um empilhador e os impactos do mesmo com a estrutura podem ser frequentes.
- **11** - O armazém de paletes vazias é o local onde ficam armazenadas as paletes e é composto por um mecanismo capaz de debitar a pedido do sistema uma paleta de cada vez.
- **12** - O transportador de paletes é o responsável por transportar uma paleta vazia desde o ponto em que a mesma sai do armazém de paletes vazias até ao ponto em que é formada a paleta, o que o torna o equipamento responsável pelo correto posicionamento da paleta.

- **13** - O local de formação da palete é o ponto onde a palete é entregue pelo transportador de paletes e onde a mesma se mantém durante e até que a sua construção esteja completa.
- **14** - O armazém de intercalares é outro ponto que deve de estar assegurado por sistemas de segurança, pois também é normal a movimentação de pessoas nesta zona. Aqui os intercalares permanecem até que são utilizados.
- **15** - O robô de colocação de intercalares nem sempre está presente nas ilhas de paletização por desempenhar uma tarefa que pode ser feita pelo robô de manipulação de camada se lhe for adicionado um mecanismo que o capacite de pegar nos intercalares. Se por um lado isto poupa algum investimento, por outro lado esta poupança reflete-se na redução da cadência da ilha.
- **16** - A saída de paletes completas pode estar diretamente ligada a um armazém automático ou pode ser também preparada para descarga de um empilhador, neste caso a segurança é também um ponto de elevada importância já que esta saída representa uma grande abertura na vedação da ilha e que dá acesso direto para o seu interior. Devem ser implementados sistemas que permitem a saída de paletes mas que impeçam a entrada de pessoas com a ilha em funcionamento.

Uma ilha de paletização à fila é, normalmente, menos complexa e alguns dos equipamentos que a compõem são semelhantes aos que compõem a ilha de paletização à camada. Na Figura 3.14 é representada uma ilha de paletização à fila onde os sete pontos mais importantes se encontram referenciados.

- **1** - Tal como na ilha de paletização à camada, esta possui um transportador de *packs* para acumulação à entrada da instalação.
- **2** - Enquanto que na ilha de paletização à camada existia um transportador específico para cadenciar e organizar os *packs*, nesta o transportador de entrada leva os produtos até ao seu final e cria acumulação do número de produtos pretendidos através da contagem com recurso a fotocélulas ou técnicas semelhantes. Assim que o agrupamento está completo o transportador pára de modo a que não se acumulem mais produtos no seu final. Assim que o robô apanha o conjunto, o transportador arranca de novo.
- **3** - À semelhança do que acontece nas ilhas de paletização à camada, existe um armazém de intercalares. No entanto é mais comum que seja o mesmo robô que organiza os produtos a fazer a tarefa de colocar os intercalares, já que esta é uma metodologia que tenta conciliar uma boa cadência com flexibilidade dos equipamentos.

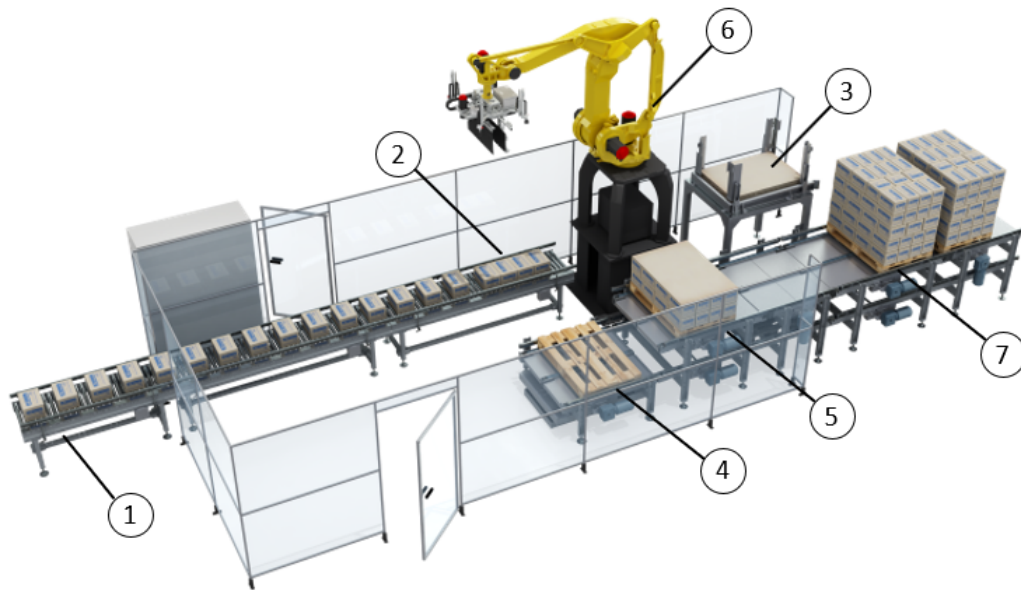


Figura 3.14: Ilha de paletização à fila [68].

- **4** - Existe também um transportador de entrada de paletes vazias que, neste exemplo, se encontra por baixo do transportador de saída. No entanto, este transportador não posiciona as paletes no ponto onde é formada a paleta, ou seja, o robô tem de apanhar a paleta no final deste transportador e colocar a mesma no início do transportador de saída, local onde é formada a paleta.
- **5** - O ponto de formação da paleta, à semelhança da ilha de paletização à camada, situa-se no início do transportador de saída, evitando grandes movimentações da paleta depois de estar formada.
- **6** - Normalmente a ilha é apenas constituída por um robô que realiza todas as tarefas necessárias, exigindo que o seu *gripper* inclua mecanismos para pegar em paletes, intercalares e produtos. Este robô também se encontra montado sob um pedestal de aço para melhor se aproveitar o seu espaço de trabalho.
- **7** - A saída de paletes prontas é feita, neste exemplo, pelo transportador de saída, com capacidade de acumulação. No entanto nestas ilhas é também comum que a paleta seja construída no chão em dois pontos diferentes, assim quando o robô está a formar a paleta num dos pontos, outro está a ser descarregado por um empilhador que pode ser autónomo ou guiado por um operário. Esta última hipótese surge da necessidade de reduzir os custos do investimento em algumas soluções.

Por fim, as ilhas mais básicas são as que seguem a metodologia de paletização à caixa, ou ao produto. Nestas ilhas, a grande maioria dos equipamentos assemelham-se aos da ilha de paletização à fila, como é exemplificado na Figura 3.15, no entanto como as capacidades dos equipamentos são menores o investimento será também inferior.

- **1** - Nestas soluções, o transportador de entrada de produtos na ilha leva o produto até ao ponto onde é apanhado pelo robô. Neste exemplo em específico, no ponto onde o produto é apanhado o transportador é um transportador de rolos de modo a permitir que o produto seja agarrado pela sua parte de baixo com a garra de pinças.
- **2** - O armazém de paletes automático e o transportador permitem que as paletes sejam entregues no ponto de formação das paletes quando é necessário, no entanto em muitas situações o armazém de paletes não é automático pelo que as mesmas se encontram empilhadas num armazém simples e o robô fica responsável pelo posicionamento das mesmas. O mesmo acontece com os intercalares que, apesar de neste exemplo não serem necessários, são muitas vezes também manipulados pelo robô.

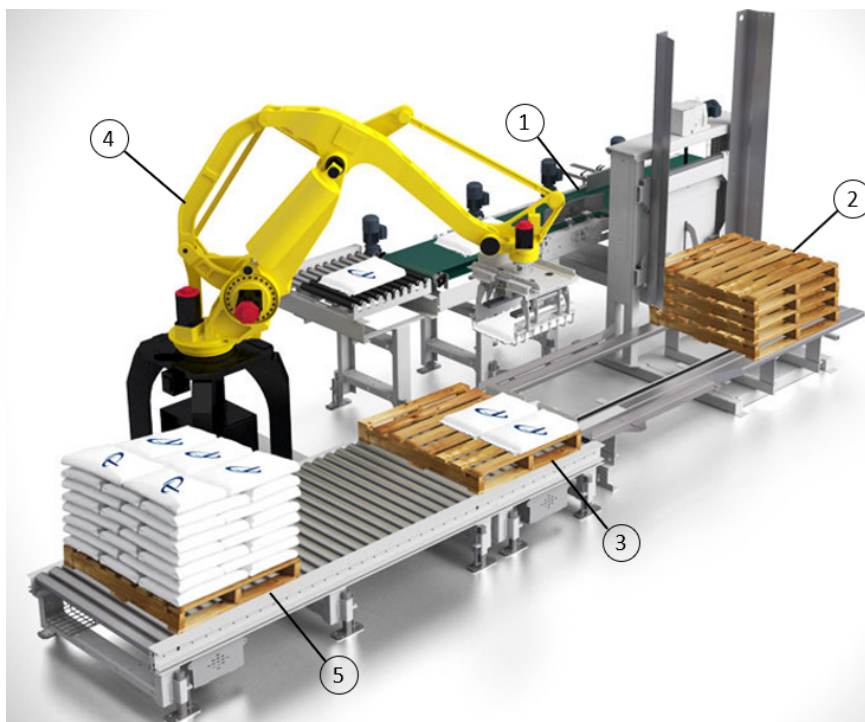


Figura 3.15: Ilha de paletização ao produto [69].

- **3** - É comum que em todas as ilhas de paletização o ponto de formação da palete seja sobre o transportador de saída ou num ponto que evite mais manobras complexas para evacuação da palete.
- **4** - Normalmente, nestas soluções, o robô é mais barato devido ao facto de que a sua capacidade de carga não tem de ser muito elevada, apenas tem de suportar o *gripper* e o objeto a ser manipulado, que nestas situações é apenas um produto, um intercalar ou uma palete.
- **5** - A saída das paletes completas, tal como no exemplo anterior, é feito por um transportador mas pode ser feito de uma maneira mais convencional, com um empilhador.

## 3.4 Caso de Estudo

Nesta secção é feita a análise de um caso de estudo. Esta análise consiste na apresentação do problema e dos requisitos e na apresentação da solução que foi dada a este desafio. Este caso de estudo apresenta uma solução real que se desenvolveu na Pentaline pelo que a identidade do cliente será mantida em confidencialidade, deste modo é-lhe atribuído um nome fictício.

### 3.4.1 Paletização de garrafas de refrigerante

#### 3.4.1.1 Desafio

A empresa Refrescos Lda tem observado um aumento nas encomendas de refrigerantes que lhe são feitas e, para reagir a este aumento de procura, investiu no aumento da sua capacidade de produção. Com o aumento do volume de produção surgiram necessidades de escoamento mais elevado no final da produção, isto é, na paletização e armazenamento dos produtos. Para resolver este problema a Refrescos Lda decidiu investir numa ilha de paletização para que a capacidade de paletização acompanhe o aumento de produção.

A Refrescos Lda necessita de uma solução capaz de paletizar os seus *packs* de 6 e 12 garrafas de 1,25 l ou 1,5 l. No entanto tem perspectivas de que num futuro próximo passe a produzir *packs* de 8 garrafas e também garrafas de 0,5 l.

Como as quantidades de cada palete, o seu *layout* e dimensões encontram-se estudadas e acertadas com os clientes, a solução a desenvolver deve ser capaz de formar paletes com os mosaicos que são seguidos atualmente. Estes encontram-se ilustrados no Apêndice A para possibilitar uma melhor perceção. Em relação ao modelo das paletes utilizadas, este deve-se manter inalterado, com dimensões de 1200x800 mm e a solução pode, opcionalmente, estar preparada para no futuro utilizar paletes com dimensões superiores (1200x1000 mm). Os intercalares a utilizar devem de ser os atualmente consumidos, com dimensões 1200x800 mm e, tal

como as paletes, a solução pode estar preparada para trabalhar com intercalares de dimensões superiores (1200x1000 mm). É ainda referido que na formação de paletes deve ser colocado um intercalar entre todas as camadas exceto no topo.

No que respeita à entrada e alimentação de paletes, a Refrescos Lda pretende que seja instalado um armazém de paletes automático com alimentação por empilhador, e referindo-se à entrada e alimentação de intercalares pretende que os intercalares se encontrem num armazém de intercalares automático com alimentação manual ou por empilhador.

A Refrescos Lda indica ainda que a enchedora tem uma cadência nominal de 24000 garrafas/hora e é a essa cadência que a ilha de paletização tem de corresponder. Por fim, fornece o *layout* do local onde será instalada a ilha de paletização e as máquinas na sua vizinhança.

De modo a facilitar a análise dos dados mais relevantes a informação relativa aos *packs* de produtos, suas características e cadências mínimas exigidas está organizada na Tabela 3.2, a informação relativa às características das paletes e intercalares está organizada na Tabela 3.3, e a informação referente à construção das paletes está organizada na Tabela 3.4.

Como é esperado, a empresa que está à procura de uma solução automatizada de paletização não coloca um requisito que especifique qual a metodologia a adotar na ilha de paletização. Cabe à Pentaline ponderar os requisitos e as capacidades que cada metodologia tem, de modo a que seja feita uma decisão que relacione os requisitos funcionais, os requisitos de espaço e limitações no investimento.

Tabela 3.2: Produtos a serem trabalhados na ilha de paletização.

Referência	Config. Pack	Dimensões CxLxA (mm)	Peso (kg)	Cadência (pack/min)
Laranja 1,25 l	3x2	263x175x308	7,5	67
	4x3	350x263x308	15,0	34
Laranja 1,50 l	3x2	273x182x336	9,0	67
	4x3	364x273x333	18,0	34
Ananás 1,50 l	3x2	276x184x333	9,0	67
	4x3	368x276x333	18,0	34
Futuro 0.5 l	-	-	-	-
	-	-	-	-

Tabela 3.3: Paletes e intercalares a utilizar na ilha de paletização.

Tipo	Altura (mm)	CxL (mm)
Paleta LPR (Euro Pallet)	150	1200x800
Paleta Industrial	150	1200x1000
Intercalar Cartão Liso	1	1200x800
Intercalar Cartão Liso XL	1	1200x1000

Tabela 3.4: Características das paletes a formar na ilha de paletização.

Referência	Config. Pack	Nº Camadas	Palete	Packs/Camada	Peso/Camada (kg)
Laranja 1,25 l	3x2	4	LPR	19	143
	4x3	4	LPR	10	150
Laranja 1,50 l	3x2	4	LPR	19	171
	4x3	4	LPR	10	180
Ananás 1,50 l	3x2	4	LPR	19	171
	4x3	4	LPR	10	180
Futuro 0.5 l	-	-	-	-	-
	-	-	-	-	-

### 3.4.1.2 Solução

Para dar resposta ao desafio colocado pela Refrescos Lda, a Pentaline estudou os requisitos impostos e concluiu que a melhor solução passaria pela implementação de uma ilha de paletização que segue a metodologia de paletização à camada. Para apresentação da solução ao cliente, a Pentaline elaborou uma representação 3D, a qual é apresentada na Figura 3.16.

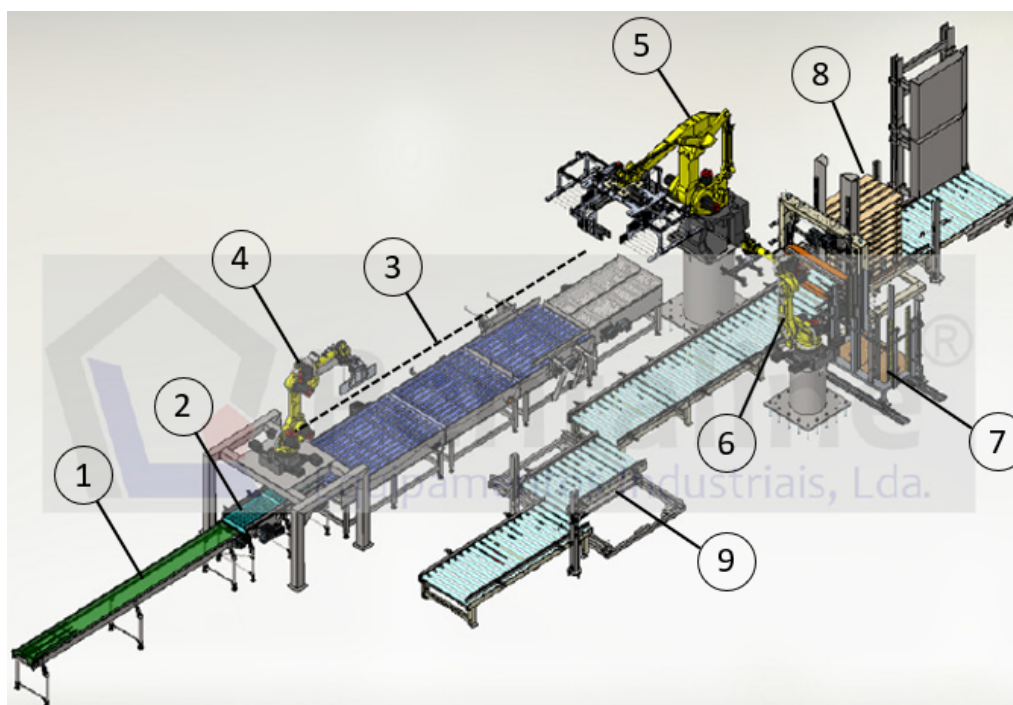


Figura 3.16: Ilha de paletização apresentada como solução ao desafio colocado pela Refrescos Lda.

Para trazer os produtos desde o exterior até ao interior da ilha, esta solução é composta por um transportador de *packs* para acumulação à entrada da instalação, identificado com o número 1 na Figura 3.16. Logo depois deste é instalado um transportador responsável pela indexação de *packs* à entrada da instalação (identificado com o número 2 na Figura 3.16) de modo a que os *packs* entrem para a zona de formação de mosaico agrupados consoante as necessidades impostas pelo mosaico. Ambos os transportadores têm uma largura de 360 mm, o que é o suficiente para qualquer um dos tipos de *packs* já que o que possui a maior largura é o *pack* 4x3 de garrafas de sumo de ananás de 1,50 l.

Depois destes dois transportadores segue-se a linha de formação de mosaico e *picking*, identificada com o número 3 na Figura 3.16 e representada com maior pormenor na Figura 3.17. Nesta figura é possível identificar os seus principais constituintes, sendo identificada com o ponto 1 a mesa de formação de camada, seguindo-se a mesa de acumulação de camada (ponto 2) que é sucedida pela mesa identificada com o ponto 3, a mesa de *pré-picking*, e por último surge a mesa de *picking* identificada com o ponto 4. A largura destas mesas é de 911 mm, possibilitando a formação dos mosaicos pretendidos, tendo na sua totalidade um comprimento de 5200 mm. Tanto o batente no final da mesa de acumulação de camada como o batente no final da mesa de *pré-picking* são de atuação pneumá-

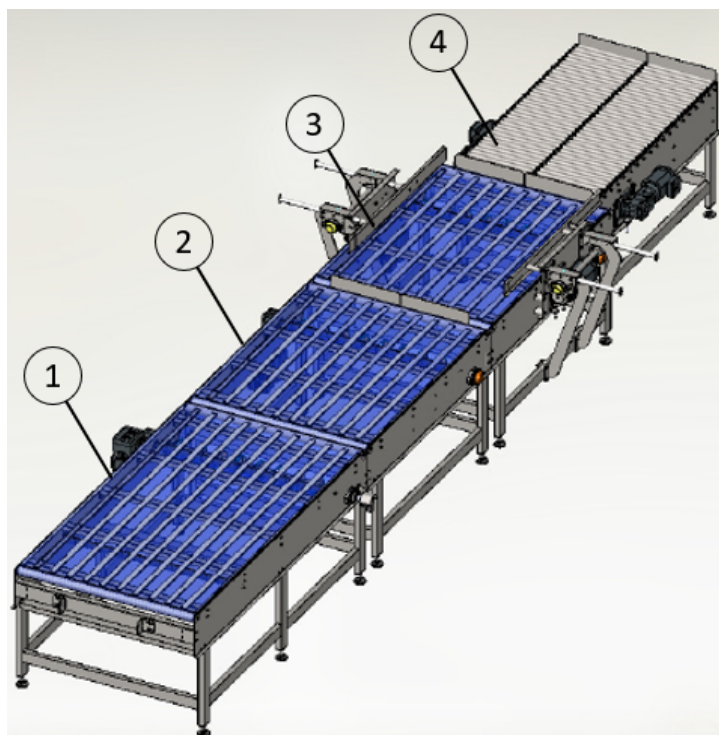


Figura 3.17: Linha de formação de mosaico e *picking*.

tica, já os centradores da mesa de *pré-picking* têm atuação elétrica de modo a permitir um ajuste da posição de centragem da camada que está a ser preparada.

Em relação aos robôs utilizados, foi escolhido para robô de formação de mosaico um FANUC R-1000iA/80H, representado na Figura 3.16 com o número 4. Este é um robô de 5 eixos, com 80 kg de capacidade de carga, com uma repetibilidade que ronda os 0,2 mm e com um alcance máximo de 2230 mm. A razão da escolha de um robô de 5 eixos para esta função está relacionada com o ponto de montagem do robô que, com o objetivo de poupar espaço se encontra instalado por cima do transportador de indexação de *packs*. Assim, o plano em que o robô irá trabalhar terá uma altura inferior à da sua base, pelo que se requer um robô com características que lhe confirmam maior agilidade. Este robô inclui o controlador *standard* R-30iB capaz de controlar robôs com até seis eixos e inclui ainda uma consola táctil onde pode ser feita e consultada a programação. De salientar ainda que este robô trabalha com o sistema *Line Tracking*, que o capacita de trabalhar sobre objetos de trabalho móveis, como os *packs* que se movimentam ao longo da mesa de formação de camada (ver Figura 3.18).

O *gripper* que se encontra acoplado a este robô é projetado tendo em conta



Figura 3.18: Robô de formação de mosaico e respetivo controlador, pedestal e *gripper*.

os objetos que serão manipulados, pelo que foi adotado um mecanismo baseado numa garra de chapas de atuação elétrica. Como o *gripper* é servo-controlado, os dois lados da garra têm uma translação simétrica pelo que o produto a manipular estará sempre centrado no *gripper*.

Para a manipulação de camadas é necessário um robô com capacidade de carga mais elevada sendo que nos mosaicos indicados pela Refrescos Lda o peso máximo de uma camada será de 180 kg, isto para os *packs* 4x3 de garrafas de 1,50 l. Como este robô, ao contrário do robô de formação de mosaico, trabalha sobre planos com pouco desnível em relação ao plano da sua base foi escolhido o robô FANUC M-410iC/500 (identificado na Figura 3.16 com o número 5) com um controlador R-30iB e respetiva consola táctil. Este é um robô de 4 eixos, capaz de manipular até 500 kg no conjunto *gripper*+camada, tem uma repetibilidade próxima dos 0,5 mm e um alcance máximo de 3143 mm.

O *gripper* acoplado a este robô foi projetado para manipular as camadas formadas por *packs* de garrafas, como tal a melhor decisão seria optar por uma garra de pinças de abertura linear de modo a que ao pegar na camada a mesma não fosse deformada (ver Figura 3.19). Este *gripper* é composto por três principais

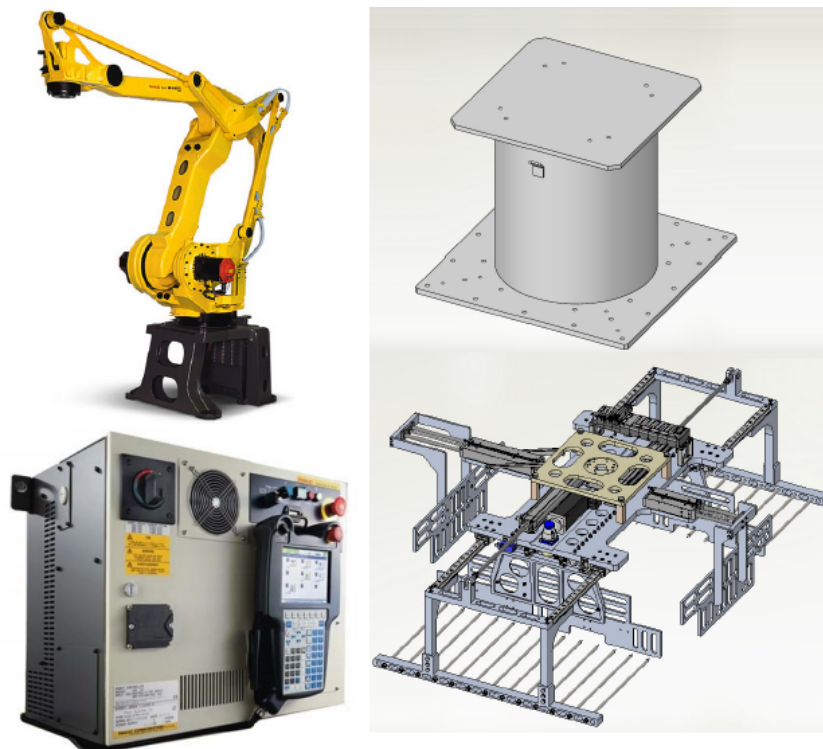


Figura 3.19: Robô de manipulação de camadas e respetivo controlador, pedestal e *gripper*.

mecanismos: um mecanismo composto por quatro chapas responsáveis por centrar a camada, um mecanismo de garfos de translação individual com atuação pneumática responsáveis por segurar a camada pela sua base, por fim existe ainda um batente no topo, também de atuação pneumática, responsável por estabilizar a camada na sua parte superior. No que toca ao mecanismo de centragem, as duas chapas de movimento paralelo ao dos garfos são servocontroladas, as duas restantes têm atuação pneumática.

O último robô na ilha é o robô de manipulação de intercalares, o qual se encontra identificado com o número 6 na Figura 3.16. Para esta função foi escolhido o FANUC M-710iC/20L e o respetivo controlador R-30iB com a consola táctil incluída. Este, é um robô de seis eixos com uma capacidade de carga de 20 kg, repetibilidade na ordem dos 0,15 mm e um alcance máximo de 3110 m. Para este robô é de realçar os requisitos de capacidade de alcance e agilidade, já que os intercalares têm comprimento e largura consideráveis e têm de ser retirados do seu armazém numa trajetória vertical. Este robô é o mais indicado para este fim pois a sua capacidade de carga não tem de ser elevada já que o *gripper* acoplado a este robô é de construção em alumínio (ver Figura 3.20). O *gripper* é composto

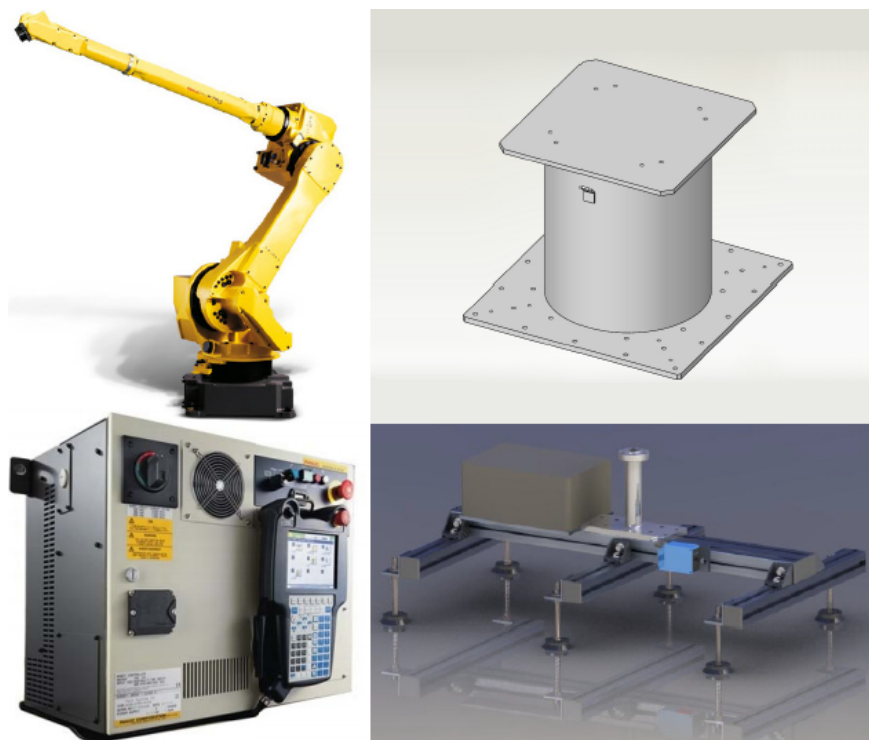


Figura 3.20: Robô de manipulação de intercalares e respetivo controlador, pedestal e *gripper*.

por um sistema de vácuo e um sistema de sopro para a manipulação dos intercalares através de 6 ventosas fixas com uma mola. É ainda incluído um sensor que faz a deteção de intercalares presente no *gripper*. Para a aproximação à pilha de intercalares existe um sensor que monitoriza a altura da pilha de intercalares e a comunica ao controlador do robô.

O armazém de intercalares (identificado com o número 7 na Figura 3.16) encontra-se montado sobre carris e, como são fornecidos dois armazéns torna-se fácil e rápida a alimentação de intercalares. Basta retirar o armazém de intercalares vazio e colocar o que foi cheio por trabalhadores no exterior da ilha. Estes armazéns têm diversos pontos de afinação e, deste modo, é possível colocar intercalares de outros modelos (ver Figura 3.21).

No que respeita às paletes, as mesmas serão armazenadas num armazém automático de funcionamento eletromecânico com capacidade de armazenamento de 16 paletes, identificado na Figura 3.16 com o número 8. Depois de serem dispensadas pelo armazém automático, as paletes seguem em transportadores de rolos com 1428 mm de largura, isto é, capazes de trabalhar com qualquer uma das paletes indicadas pela Refrescos Lda. Depois de construída, a paleta segue até uma mesa repartidora que, por razões de redução do espaço ocupado pela ilha teve de ser incluída. Esta mesa repartidora faz o desvio da paleta completa para o transportador de saída, onde existe uma barreira fotoelétrica de segurança que permite a deteção de intrusos na abertura para saída de paletes completas (ver Figura 3.21).

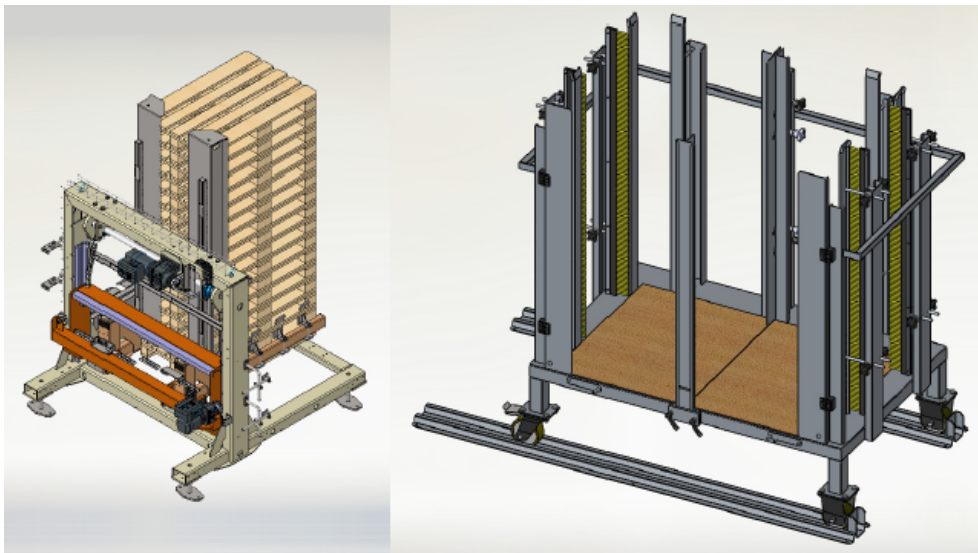


Figura 3.21: Armazém automático de paletes e armazém de intercalares.

Por fim, são apresentadas as cadências que esta ilha é capaz de atingir. Pela análise da Tabela 3.5 pode ser concluído que os principais requisitos de cadência exigidos à Pentaline foram correspondidos e até excedidos, o que resulta numa possibilidade da Refrescos Lda poder continuar a aumentar a sua produção sem exceder a capacidade da ilha de paletização.

Tabela 3.5: Comparação entre cadências pedidas e atingidas.

Referência	Config. Pack	Cadência pedida (pack/min)	Cadência da ilha	
			(pack/min)	(paletes/min)
Laranja 1,25 l	3x2	67	80	1,05
	4x3	34	40	1
Laranja 1,50 l	3x2	67	80	1,05
	4x3	34	40	1
Ananás 1,50 l	3x2	67	80	1,05
	4x3	34	40	1
Futuro 0.5 l	-	-	-	-
	-	-	-	-



## Capítulo 4

---

# Desenvolvimento de Interface Homem-Máquina para Ilhas de Paletização

---

O desenvolvimento de uma interface homem-máquina para ilhas de paletização é tido como objetivo deste trabalho e, ao ser atingido com sucesso resulta num *software* que recolhe, trata e envia dados para o autómato mestre, que funciona como o cérebro de uma ilha de paletização de modo a que esta funcione de acordo com as necessidades do utilizador.

### 4.1 Objetivos

Uma ilha de paletização não é um sistema simples, possui muitos equipamentos nos quais se integram muitos sensores e atuadores, além disso, muitas das vezes, existe mais que um robô. Esta complexidade aumenta ou diminui consoante a metodologia pela qual a ilha trabalha, sendo que nas ilhas de paletização à camada a complexidade e número de equipamentos são substancialmente superiores aos das ilhas de paletização ao produto e à fila.

Um dos objetivos desta interface é que esta seja capaz de se interligar com ilhas de diversas metodologias, sublinhando que a mesma deve ser desenvolvida com vista à sua integração com as ilhas de paletização à camada, de modo a que em situações de necessidade de adaptação a outras ilhas com diferente metodologia apenas seja necessário omitir funcionalidades, já que as ilhas de paletização à camada são as mais completas.

No que toca à introdução do *layout* da camada de cada palete, a complexidade

é menor nas ilhas de paletização ao produto, já que cada produto presente na palete é tratado com uma unidade. Nas ilhas de paletização à fila e à camada é comum que sejam manipuladas quantidades variáveis de produtos em simultâneo, sendo esse um dos pontos que faz diferir a complexidade nos interfaces de ilhas com diferentes metodologias.

O *layout* das camadas das paletes é, na maioria das vezes, estudado e definido em ambientes computacionais de alto desempenho, capazes de gerir a área útil da palete da melhor maneira possível, sem comprometer a robustez da mesma e a integridade dos produtos, tendo em conta as limitações dos meios de transporte e armazenamento. Depois de definido, o mosaico tem de ser posto em produção na ilha. O objetivo desta interface é fazer a ponte entre o estudo e definição do mosaico e a produção de paletes com essa organização dos produtos, funcionando como ferramenta de parametrização do sistema (ver Figura 4.1).

Outro dos objetivos é ter uma introdução de informações relativas aos modelos das paletes que não esteja diretamente relacionado com a ligação da interface à ilha, isto é, mesmo tendo a ilha desligada ou uma falha de comunicação, o *software* continue a permitir a criação e introdução de características de novos modelos. Assim a interface torna-se num bloco que integra a ilha de paletização mas que é capaz de manter diversas funcionalidades quando isolado da mesma, o que não acontece com a maioria das interfaces homem-máquina, já que, tipicamente, estas funcionam como uma extensão do sistema que apenas permite o controlo dos seus equipamentos.

Tendo estes objetivos em conta, pode concluir-se que a interface em questão não é somente um *software* de interação entre o operador e a ilha de paletização. Além das funcionalidades comuns de gestão da instalação tem-se como objetivo a criação de uma interface capaz de receber as características das paletes e o

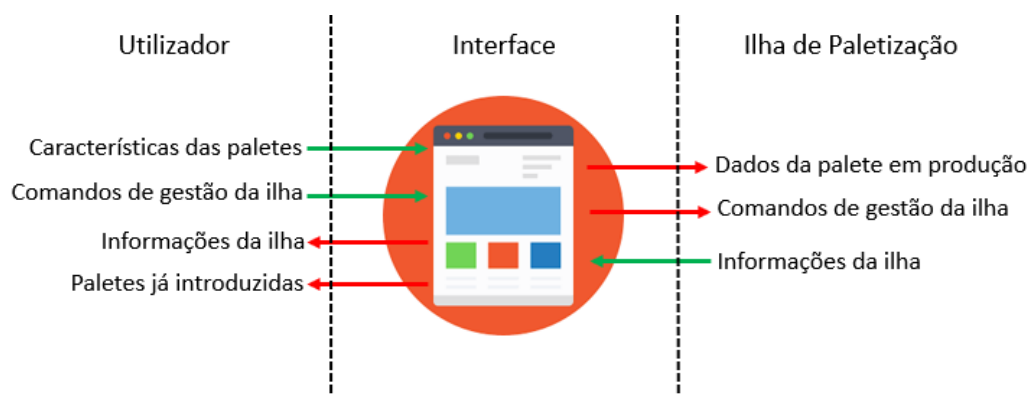


Figura 4.1: Principais focos de informação na interface.

respetivo mosaico das suas camadas, disponibilizando estas informações sempre que sejam necessárias tanto para consulta como para ordenar a sua produção.

## 4.2 Requisitos

O desenvolvimento de uma interface homem-máquina pressupõe um bom desempenho funcional e estético, tal como foi descrito no capítulo dois, sendo que o desempenho funcional se traduz num requisito de que a interface contemple as funcionalidades técnicas que são pretendidas, e a vertente mais visual do desempenho traduz-se no requisito de criar um bom relacionamento entre o utilizador e a interface de forma a haver uma utilização tão fluída quanto possível.

O principal requisito está relacionado com a gestão dos mosaicos e dos seus constituintes. A interface tem de ser capaz de receber dados acerca das paletes, dos produtos e dos intercalares utilizados pela empresa que acolhe a ilha de paletização e esses dados devem de ser guardados em memória para posterior consulta e utilização para criação dos mosaicos. Posteriormente, a interface deve ser capaz de receber os dados referentes aos mosaicos e de os guardar para futura seleção e envio para produção. É ainda importante sublinhar que estas listas, tanto as de intercalares, paletes e produtos como a de mosaicos podem ser muito extensas pelo que a interface deve ser capaz de gerir, de algum modo, uma grande quantidade de informação.

No que toca aos dados inseridos relativamente a produtos, paletes e intercalares estes devem obrigatoriamente contemplar as características base do comprimento, altura, largura e peso, existindo também espaço para atribuição de um nome ou termo identificativo ao item em questão. Opcionalmente pode ainda existir um campo de observações ou informações adicionais tais como cor (da paleta) ou material (do intercalar). De referir ainda que, na criação de um novo artigo, tem de ser dada a possibilidade de visualização das dimensões do novo produto relacionadas com as dimensões do *gripper* de formação de mosaico, permitindo uma perceção da capacidade de agrupamento de produtos no *gripper*.

Relativamente aos mosaicos a introdução de dados é mais extensa e complexa. Existem duas principais fases às quais a interface deve corresponder (ver Figura 4.2), a primeira está relacionada com a introdução de dados base e características principais da paleta a ser construída, a segunda fase é a criação do mosaico, isto é, a construção do *layout* que as camadas da paleta devem apresentar.

Nas informações base do mosaico estão incluídas:

- tipo de paleta, intercalar e produto que fazem parte do novo modelo;
- margem máxima da paleta, isto é, a distância máxima que os produtos podem exceder os limites da paleta;

- número de camadas da paleta;
- número de artigos que serão colocados em cada camada;
- direção de formação da camada, que indica se as camadas serão formadas na linha de formação de mosaico lateralmente ou frontalmente;
- distribuição de intercalares ao longo criação da paleta;
- distribuição de paletes ao longo da criação da paleta;
- sucessão de camadas; este campo indica se as camadas serão todas iguais ou se há uma sucessão par/ímpar;
- um campo opcional para possibilitar introdução de notas e observações referentes ao mosaico em criação.

Na fase de criação do mosaico é criado o *layout* das camadas da paleta, para isto devem ser apresentadas duas opções que possibilitem a criação com base na introdução de coordenadas e a criação com base num ambiente mais criativo, com o posicionamento dos artigos, individualmente ou em grupos de acordo com a capacidade do *gripper*, sobre a paleta, com a representação do estado do *layout* que está a ser construído. Ao longo do processo de criação devem ser detetadas anomalias, como sobreposições ou ultrapassagem dos limites da paleta e deve ainda ser possível verificar a aparência lateral da paleta bem como consultar dados como o peso total e a altura.

Por fim, como requisito extra, é ainda considerada a criação de uma paleta completamente personalizável em que cada camada contém um *layout* diferente e com um número de artigos igualmente variável. Apesar de não ser muito comum a utilização desta funcionalidade, esta é por vezes útil.

Deve ainda ser possível consultar que entradas já foram introduzidas tanto para paletes como para artigos, intercalares e mosaicos e deve também ser possível eliminar as entradas que forem pretendidas.

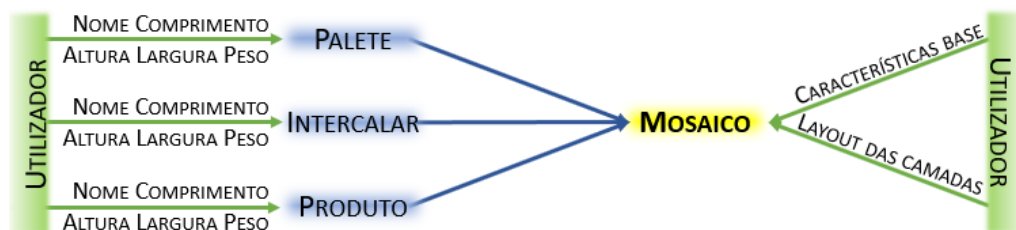


Figura 4.2: Origem dos dados para configuração de um mosaico.

Noutro foco desta interface está o controlo e monitorização da ilha de paletização. Neste campo de trabalho o *software* deve ser capaz de apresentar informações que indiquem o estado atual do processo de paletização, permitindo que sejam consultados parâmetros como o número atual e o número de itens ou camadas que têm de estar em cada secção da ilha. A interface deve também contemplar comandos de arranque da instalação, paragem e colocação da mesma em modo manual, bem como comandos de arranque e paragem do tapete de entrada. É igualmente útil a apresentação de um contador que indique a quantidade de paletes construídas na ilha desde a colocação do modelo atual em produção.

No que respeita à seleção de um novo tipo de palete para ser colocado em produção este deve ser feito de modo a que não hajam equívocos entre modelos diferentes, permitindo uma consulta prévia das suas características.

A interface deve ainda ser capaz de apresentar uma lista de alarmes e eventos referentes à ilha de paletização e deve ainda dispor de um sistema de controlo de acessos que permita escalonar diferentes níveis de utilizadores de forma a vedar o acesso a diversas funcionalidades aos níveis mais baixos de utilizadores.

Por fim, o desenvolvimento de uma interface homem-máquina pode ser feito em diversos *softwares* de diferentes fabricantes, cada um com as suas vantagens e desvantagens, no entanto, como as ilhas de paletização com as quais a interface se deve de interligar são baseadas num PLC da Siemens, é requisito que a mesma seja desenvolvida num *software* que a Siemens oferece para este mesmo fim.

É ainda importante perceber que, atualmente, já existem no mercado interfaces com estas finalidades, umas com melhor qualidade que outras. No entanto estas interfaces de melhor qualidade são propriedade de empresas concorrentes estrangeiras de nível internacional, amplamente trabalhadas ao longo de anos de desenvolvimento principalmente nos aspetos *user-friendly* que tornam a interface mais fácil de utilizar e mais apelativa para os clientes que adquirem uma ilha de paletização para a sua empresa. Assim sendo, é importante salientar que a interface deve ficar aberta a posteriores desenvolvimentos e melhorias que permitam aumentar o seu leque de funcionalidades e corrigir problemas que sejam posteriormente identificados.

### 4.3 Alternativas de Solução

Feita uma análise dos requisitos são retidos dois aspetos mais relevantes nesta interface, a capacidade de guardar grandes quantidades de informação consultando-a quando necessário e a capacidade de dinamização de operações relativamente complexas tais como a criação de um mosaico.

Como descrito na secção dos requisitos, a interface tem de ser desenvolvida num ambiente que permita a interligação da mesma com um PLC Siemens. Para

este fim a Siemens oferece uma vasta gama de *softwares* integrados no TIA Portal que permitem o desenvolvimento de interfaces homem-máquina bem como realizar toda a configuração necessária para a sua interligação com um PLC. No entanto, como a interface necessita de ter capacidades que permitam a introdução dos mosaicos de uma forma criativa são eliminados os *softwares* mais básicos que acompanham as interfaces mais simples. A capacidade de gestão de grandes volumes de dados também é um fator eliminatório que impossibilita a utilização das interfaces típicas baseadas numa consola que se liga ao PLC. Posto isto, apenas resta a hipótese da criação de uma interface com recurso ao WinCC Advanced ou ao WinCC Professional, *softwares* tratados como *add-ons* do Siemens TIA Portal.

Ao serem utilizados estes programas de desenvolvimento a interface resultante não se irá prender a um modelo específico de consola. Com estes é desenvolvido um ficheiro que irá correr em qualquer computador que corra um sistema operativo Windows com o *Runtime* instalado. O *Runtime* é a aplicação que executa o *software* desenvolvido no TIA Portal. Posto isto, ao desenvolver uma interface nestas condições, as limitações de capacidade de gestão de dados serão reduzidas, bem como as limitações referentes à capacidade de processamento. A maneira como a interface é apresentada ao utilizador também não fica dependente de qualquer configuração no ambiente de desenvolvimento, sendo permitido que a mesma interface seja apresentada de uma forma mais tradicional com um rato e um teclado, ou de uma forma mais moderna, recorrendo à utilização de um *touchscreen*. Deste modo, a interface será utilizada sobre qualquer PC que corresponda aos requisitos mínimos para a execução do *Runtime* deixando em aberto a escolha do *hardware*. Para uma utilização ao nível industrial normalmente a opção recai nos PC industriais que se distinguem dos PC mais comuns principalmente pela sua robustez física de modo a permitir que estes trabalhem em ambientes com condições adversas. Na Figura 4.3 é apresentado como exemplo o



Figura 4.3: PC industrial Nuvo-3000 [70].

PC industrial Nuvo-3000 que possui como características um processador Intel® Core™ i7, memória RAM de até 16 GB e ainda um sistema de refrigeração sem ventoinhas, permitindo o seu funcionamento em ambientes com poeiras e reduzindo potenciais focos de avaria. Para visualização a opção recai quase sempre para os painéis *touchscreen* pelo facto de facilitarem a interação com o utilizador. Nestes sistemas, os painéis têm, normalmente, entre 15 e 21 polegadas de modo a facilitar a interação com o utilizador. Muitas das vezes, para ser poupado algum espaço, é adquirido o PC já integrado no painel. Esta opção pode trazer algum transtorno caso o painel ou o PC tenham uma avaria, exigindo a substituição de todo o conjunto por um novo.

Como referido existem duas hipóteses pelas quais se pode optar para criar a interface homem-máquina, o WinCC Advanced e o WinCC Professional. Na secção 3 do capítulo 2, foram descritas as diferenças entre estes *softwares*, destacando-se como principal característica diferenciadora a capacidade de criação de sistemas multi-utilizador, isto é, com várias consolas a trabalhar em sintonia umas com as outras.

Para o desenvolvimento da programação da interface ambos os meios de desenvolvimento oferecem as opções de programação com recurso a *scripts* em C e *scripts* em Visual Basic. O WinCC permite que na mesma interface coexistam *scripts* dos dois tipos, pelo que o programador pode ir optando pela linguagem que melhor se adapta a cada funcionalidade em específico. A Siemens oferece manuais de apoio para ambas as linguagens de programação, no entanto no fórum de discussão entre utilizadores e engenheiros ao serviço do fabricante é mais abundante a informação e o *troubleshooting* referente aos *scripts* em Visual Basic, em grande parte por serem visualmente mais fáceis de estruturar que os *scripts* em C. Com estes *scripts* é possível fazer todo o tipo de operações de cálculo, fazer leitura e escrita de variáveis, dinamizar a interface, comunicar com bases de dados, abrir documentos em PDF e muitas outras ações. Posto isto, torna-se visível que uma das capacidades que distancia estes *softwares* dos mais comuns é exatamente esta possibilidade de programação oferecida a quem desenvolve a solução.

No que respeita à gestão dos dados referentes aos produtos, paletes, intercalares e mosaicos existem 4 principais focos: recolha, armazenamento, consulta e eliminação. É ainda importante referir que os conjuntos de dados a armazenar podem ser extensos de maneira que pode ser considerada uma forma de armazenamento exterior ao *software* da interface. Como primeira hipótese de solução pode ser considerado que o WinCC disponibiliza uma forma de armazenamento integrada, tratada pelo nome de “*recipes*”, que apesar de ser um pouco limitada, na versão Professional do WinCC se baseia numa base de dados SQL. A segunda opção para armazenamento dos dados do sistema passa por um *software* separado da interface com o qual a mesma comunicará, ou seja, recorrer a uma base de da-

dos SQL para armazenar os dados recolhidos pela interface acerca dos produtos, paletes, intercalares e mosaicos.

Por fim é importante referir ainda que a interface tem diversas informações a comunicar com a ilha através de um canal pré determinado. Noutros termos, o PC industrial tem de comunicar com o PLC através de um dos protocolos de comunicação disponíveis. Os protocolos de comunicação em questão baseiam-se na porta Ethernet que o PC industrial disponibiliza que será ligada à porta de comunicação Ethernet disponibilizada pelo PLC através de um cabo de rede RJ45. Os protocolos tipicamente utilizados nestas aplicações são o PROFIBUS e o PROFINET, no entanto, o PROFINET por ser uma versão mais recente do PROFIBUS tem ganho popularidade, já que reúne as vantagens do PROFIBUS e melhora os seus pontos fracos. A grande vantagem da utilização destes protocolos está na capacidade de comunicação entre equipamentos de diferentes fabricantes, com necessidade de poucos equipamentos e materiais específicos para este fim. Uma das grandes vantagens do PROFINET é a capacidade de trabalhar com redes sem fios e sem limite de dispositivos ligados na rede, ao contrário do PROFIBUS, que tem um limite máximo de 126 dispositivos. No que toca às velocidades de transferência de dados e capacidade de comunicação, o protocolo PROFINET supera o PROFIBUS com 100 Mbit/s *full-duplex* contra 12 Mbit/s *half-duplex* [71].

#### 4.4 Descrição da Arquitetura

Para o desenvolver da interface foram tomadas, inicialmente, um conjunto de opções que começaram por definir o rumo do trabalho a desenvolver. Uma das principais decisões foi o desenvolvimento da interface na versão Professional do WinCC, *software* que está integrado no Siemens TIA Portal. Para fundamentar esta decisão estão as vantagens deste em relação à sua versão mais baixa, o WinCC Advanced, que vão desde um leque de funcionalidades e objetos disponíveis mais abrangente até à criação de sistemas multi-utilizador. Apesar de não ser objetivo deste trabalho a criação de um sistema multi-utilizador, esta opção não se encontra descartada num desenvolvimento futuro pelo que a utilização da versão Advanced iria condicionar a evolução da interface neste sentido.

No que respeita ao armazenamento e gestão de dados existe a opção da utilização da funcionalidade “*recipes*” embutida no WinCC Professional ou a utilização de uma base de dados SQL. Após uma análise das vantagens e desvantagens de cada uma das opções concluiu-se que a utilização da funcionalidade “*recipes*” apesar de ter a vantagem de já estar incluída no *software* de desenvolvimento, o que simplifica o acesso aos dados, tem a desvantagem de oferecer uma flexibilidade no armazenamento muito limitada o que significa que a informação teria de ser organizada de forma complexa e muito resumida. Deste modo a opção mais viável

para se obter uma armazenamento de dados flexível e organizado é a utilização de uma base de dados SQL. Esta opção, apesar de trazer complexidade ao projeto, é a mais indicada tanto para uma boa organização da informação como para manter em aberto o armazenamento de mais dados que surjam em desenvolvimentos futuros.

Tomadas estas duas decisões torna-se necessária a existência de comunicação entre dois *softwares* no mesmo computador, a interface e a base de dados SQL. Para este fim o *software* de desenvolvimento WinCC Professional coloca ao dispor duas opções: a utilização de *scripts* em C ou *scripts* em Visual Basic. Apesar de num mesmo projeto desenvolvido no WinCC Professional poderem coexistir *scripts* dos dois tipos entendeu-se ser mais lógica a utilização prioritária de uma das linguagens de modo a manter coerência no desenvolver do trabalho, relegando uma das opções para segundo plano sendo apenas utilizada se for expressamente necessário. Feitos alguns testes e experiências com *scripts* das duas linguagens foi tomada a decisão de utilizar prioritariamente *scripts* em Visual Basic e deixar em segundo plano a utilização de *scripts* em C. Para apoiar o desenvolvimento do *software* existem manuais de programação para ambas as linguagens mas, para basear esta decisão em muito contribuiu a abundância de informação referente aos *scripts* em Visual Basic em fóruns *on-line* da Siemens bem como a maior facilidade de estruturação e análise da programação em Visual Basic.

Assim sendo, com base nestas decisões, a interface irá consistir num *software* desenvolvido em WinCC Professional que, através de *scripts* em Visual Basic comunica com uma base de dados SQL que é responsável por armazenar os dados sob estruturas organizadas. Para as comunicações com a ilha de paletização será utilizado o protocolo PROFINET (ver Figura 4.4) pois, como indicado anteriormente, é o que melhor se adapta a este sistema, sendo que o Siemens TIA Portal oferece ferramentas para uma rápida configuração e interligação entre estes siste-

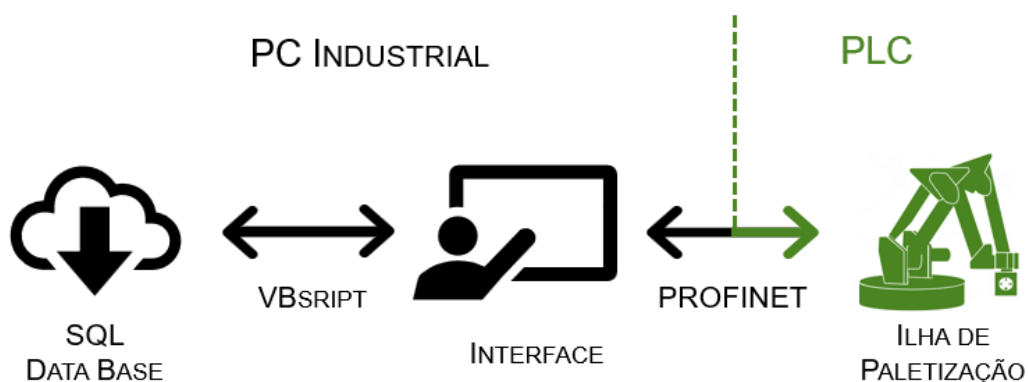


Figura 4.4: Esquema ilustrativo das comunicações entre a interface, a base de dados e o PLC.

mas. Após configurada a rede no TIA Portal, basta indicar na tabela de variáveis quais as que se pretende que sejam partilhadas com o PLC.

De modo a conseguir uma boa organização da interface, a mesma foi projetada com uma organização que permita separar as funcionalidades de acordo com o seu enquadramento. Assim sendo, foi pensado que a interface deveria de ter um ecrã que permite navegação entre os seus principais menus. Com base nesta ideia a interface possui 6 menus cada um deles com funcionalidades independentes uns dos outros (ver Figura 4.5). O menu "Home" disponibilizará funcionalidades mais gerais como o encerrar da aplicação, consulta de documentação e acesso à Internet. O menu de comando geral irá permitir enviar um modelo de mosaico/paleta para produção bem como atuar sobre o estado de outros componentes da ilha de um modo independente. O menu de informação da ilha apresenta uma visão geral da ilha e mostra informação acerca do número de *packs* presentes em cada secção da ilha. O menu de alarmes apresenta uma lista dos alarmes ativos e permite reconhecer os mesmos. O menu de gestão de utilizadores apresenta uma lista dos utilizadores do sistema de acordo com o grau de permissão que possua o atual utilizador com *login* ativo. O menu de gestão de produto disponibiliza quatro submenus ao longo dos quais é possível adicionar, eliminar e consultar as entradas correspondentes às paletes, artigos, intercalares e mosaicos, em que o submenu

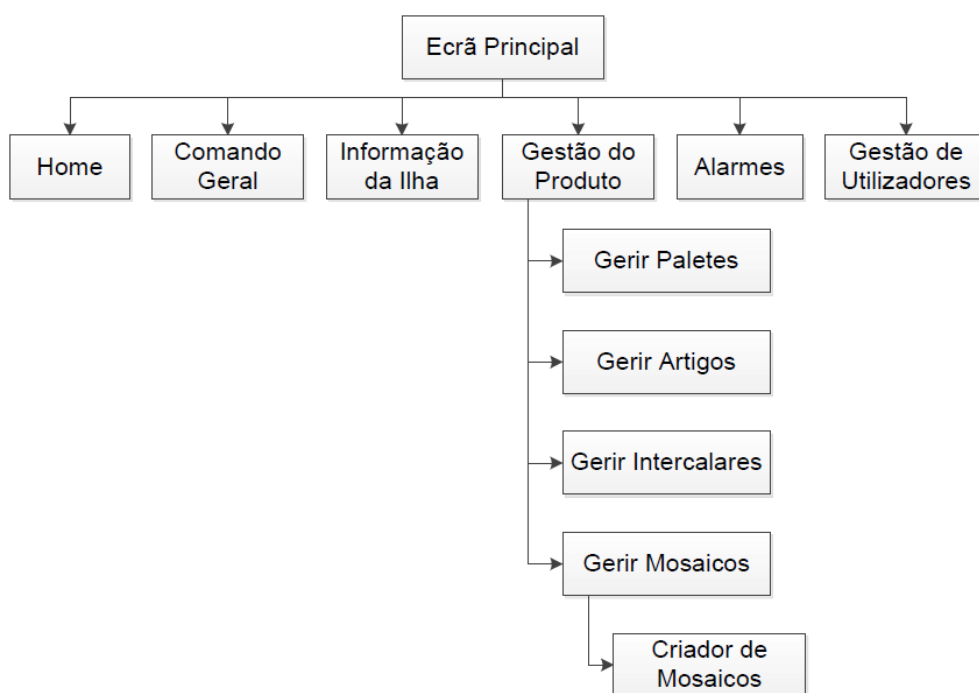


Figura 4.5: Estrutura dos menus e submenus da interface para ilhas de paletização.

de gestão de mosaicos dá ainda acesso ao criador de mosaicos onde é possível configurar a construção de uma palete para posterior envio para produção.

Para que os dados sejam armazenados de forma organizada é necessário que a base de dados seja bem estruturada e que seja capaz de relacionar a informação de vários itens de modo a que exista uma interdependência entre cada mosaico e os objetos que o compõe. Foi com este objetivo que se definiu que a estrutura da base de dados irá ser composta por cinco tabelas. Uma tabela irá armazenar as características dos produtos, a segunda irá armazenar as características dos intercalares e a terceira irá armazenar as características das paletes. As duas tabelas que restam ficam responsáveis por armazenar as características dos mosaicos. A quarta tabela armazena as características base do mosaico e a quinta tabela armazena as coordenadas de cada item de cada mosaico em relação à paleta (ver Figura 4.6).

## 4.5 Procedimentos de Teste

De modo a certificar que a interface está a funcionar de maneira correta, esta tem de responder a diversas situações de modo previsível e consistente, consoante as rotinas programadas. Para considerar que esta condição é satisfeita são apresentados alguns comportamentos esperados por parte da interface para as situações mais pertinentes.

Recorrendo ao Microsoft SQL Server Management Studio 17 é possível gerir a base de dados e consultar os dados atuais de cada tabela. Deste modo, ao utilizar a interface para gerir os produtos, paletes, intercalares ou mosaicos, deve ser verificada a alteração ordenada pelo utilizador na tabela correspondente. Em relação aos produtos, intercalares e paletes, quando é criado um novo item aparece

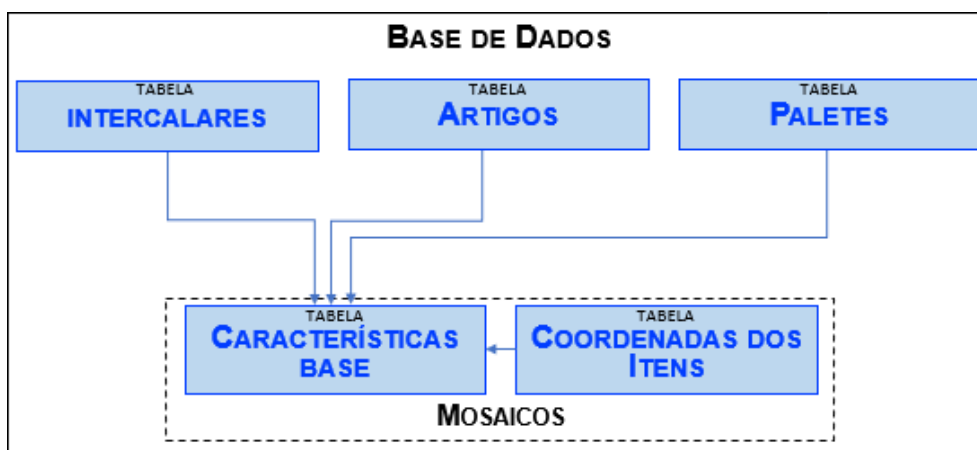


Figura 4.6: Esquema ilustrativo da estrutura da base de dados.

na tabela correspondente uma linha com as informações organizadas pelas devidas colunas da tabela (ver Figura 4.7), e quando é apagado um item a linha que lhe corresponde desaparece da base de dados. Existem duas situações especiais, caso na criação de um novo item o utilizador lhe tente atribuir um nome já existente na base de dados a interface chama a atenção a este facto e não permite a criação do item. A segunda situação está relacionada com a eliminação de itens que estão a ser utilizados em mosaicos sendo que, quando o utilizador tenta apagar um item usado num mosaico a interface reporta um erro e não permite concluir a ação.

De modo a testar o armazenamento de mosaicos na base de dados, quando é criado um mosaico deve aparecer uma linha na tabela correspondente da base de dados com as informações básicas do mosaico, as coordenadas dos produtos sobre a palete aparecem numa tabela dedicada exclusivamente a este fim. Ao apagar um mosaico não só deve desaparecer a linha correspondente da tabela de mosaicos como também devem desaparecer as linhas correspondentes aos produtos do mosaico da tabela de coordenadas.

Ao ser pedida uma listagem, quer seja dos artigos, das paletes, dos intercalares ou dos mosaicos existentes no sistema, deve ser apresentada uma lista atualizada com o ID e o nome de cada entrada correspondente à consulta requisitada. Deste modo, cada linha que seja apresentada na listagem corresponde a uma entrada na tabela que está a ser consultada, para o confirmar recorre-se mais uma vez ao Microsoft SQL Server Management Studio 17.

No criador de mosaicos é dada especial atenção à verificação do correto cálculo das posições dos artigos das camadas ímpares de acordo com a metodologia de sucessão de camadas adotada, sendo utilizado para este fim a ferramenta de-

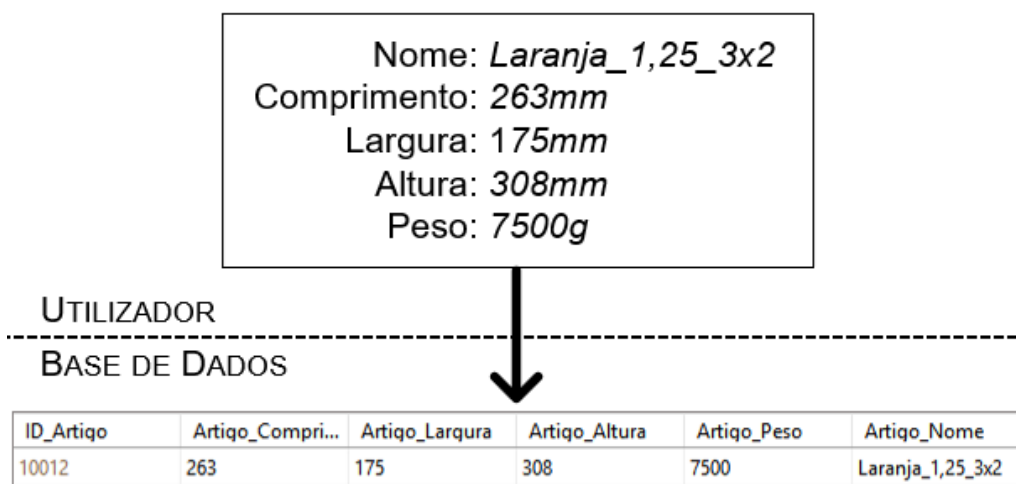


Figura 4.7: Informação de um produto que o utilizador introduz na interface e linha correspondente na base de dados.

envolvida na aplicação que apresenta graficamente a orientação dos produtos na camada ímpar.

Na criação de mosaicos existem também dados acerca do peso total da palete, do peso de cada camada e da altura total da palete. Para que estes sejam confirmados são feitos estes cálculos numa calculadora comum e são confrontados os resultados obtidos com os apresentados pela interface.

Na criação do *layout* do mosaico existem acontecimentos que a interface deve identificar e sinalizar corretamente, tais como sobreposição de produtos, ultrapassagem do limite das margens da palete e excesso de produtos na camada. De modo a testar estas capacidades, são feitos estes erros propositadamente e é verificado o modo como a interface responde a estes acontecimentos. É ainda testada a restrição à formação de grupos com um número de *packs* superior ao que o *gripper* é capaz de manipular em simultâneo.

Ainda na criação do *layout* do mosaico, é testado o menu de ação rápida que possibilita posicionar e rodar os artigos de uma forma mais intuitiva, bem como incrementar a sua posição em milímetros, sendo dada especial atenção ao correto posicionamento dos itens uns em relação aos outros, sendo utilizado este menu para posicionar itens com vários artigos e com diferentes rotações para certificar que não existem falhas nesta importante funcionalidade.

Por fim é testado se a ferramenta de "Vista Frontal/Lateral" funciona corretamente, devendo, para isso, mostrar uma representação correta da distribuição das paletes e dos intercalares e uma representação aproximada da sucessão de camadas consoante foram construídas, verificando que as camadas se encontram fora do limite da palete, caso seja uma característica do mosaico criado e verificando que a representação gráfica não excede os limites do ecrã, ajustando a sua escala de acordo com o número de camadas que está a ser representado.

No menu de comando geral é dada especial atenção à criação de uma nova ordem de produção e à consulta da atual ordem. Para testar esta funcionalidade é colocado em produção um novo mosaico e, com o monitor de variáveis do TIA Portal é consultado o valor atual das variáveis, que deve ter sido atualizado de acordo com os dados do mosaico em produção. Ao clicar sobre o resumo do mosaico em produção são apresentados os detalhes sobre o mesmo, estas informações devem de ser coerentes com o mosaico selecionado.

No que respeita às configurações de segurança é testado se o acesso a determinadas funcionalidades se encontra devidamente vedado a certos utilizadores de modo a reservar operações sensíveis a técnicos com maior formação.

## 4.6 Protótipo Funcional

### 4.6.1 Implementação Realizada

O protótipo da interface para ilhas de paletização desenvolvida pode subdividir-se em dois principais focos de trabalho: o desenvolvimento da base de dados e o desenvolvimento da interface. No desenvolvimento da base de dados estão englobados aspetos como o estudo da sua estrutura e a sua interligação com a interface. No desenvolvimento da interface são contemplados aspetos como o desenvolvimento gráfico dos menus e os *scripts* desenvolvidos em VBScript que possibilitam o seu funcionamento.

#### 4.6.1.1 Base de Dados

Neste sistema, a base de dados tem como função guardar os dados acerca das características dos elementos trabalhados na interface de modo a que seja possível a sua consulta em qualquer momento que seja necessário.

Inicialmente é feito um levantamento das informações que a base de dados deve receber. No que toca aos artigos a base de dados terá de receber o seu nome, e as suas características físicas, isto é, o comprimento, a largura, a altura e o seu peso. Deve ainda ser atribuído a cada artigo um ID que o identificará e que será incrementado a cada entrada que a tabela receba de modo a não haver possibilidade de existirem duplicados.

Para o armazenamento de paletes a lógica é semelhante ao armazenamento de artigos, devendo ser retidas informações como o nome da paleta, o comprimento, a largura, a altura, o peso e um campo de informação adicional que pode indicar a cor ou o material. Deve ser igualmente tido em conta um campo para atribuir um ID a cada paleta.

Para armazenar os intercalares é também retida informação acerca do nome de cada intercalar bem como as suas dimensões (comprimento, largura e altura) e o seu peso. Como informação adicional é ainda considerado que indica o material em que foi fabricado o intercalar. À semelhança dos artigos e das paletes é considerado ainda o ID que irá identificar cada intercalar.

Para armazenar as informações de cada mosaico serão consideradas todas as características que sejam essenciais para o utilizador e para a ilha de paletização. Deste modo é considerado o nome do mosaico, o modelo de intercalar utilizado, o modelo de paleta utilizado, o modelo de artigo a paletizar, o número de grupos por camada, o número de artigos por camada, o número de camadas, o tipo de sucessão de camadas, o peso de cada camada, o peso total da paleta, a distribuição de paletes, a distribuição de intercalares, o sentido de formação da paleta, as coordenadas de todos os grupos de artigos e a sua rotação, e por fim um campo onde podem ser introduzidas notas ou observações acerca do mosaico. É ainda

considerado um ID para cada mosaico de modo a uma identificação inequívoca de cada um.

Depois de um estudo detalhado das informações que a base de dados terá de ser capaz de guardar foi esboçado um modelo relacional da base de dados. Um modelo relacional é um conceito simples que se baseia num conceito matemático, o conceito de relação, isto é, a associação entre elementos de conjuntos não vazios. Este, é utilizado pela maioria dos sistemas de gestão de base de dados e é neste modelo que o SQL se baseia. Neste contexto uma relação é muitas vezes referida como tabela e, deste modo, uma base de dados traduz-se num conjunto de tabelas. Na base de dados, cada tabela tem um nome, e cada atributo contemplado em cada tabela tem também um nome. Já os atributos de cada tabela têm cada um o seu domínio, isto é, um formato de dados (inteiro, binário, caracteres, etc) [72].

Por fim cada tabela terá as suas chaves que são um atributo, ou conjunto de atributos, que identificam cada entrada da tabela, ou seja, não podem existir duas entradas com a mesma combinação de valores nestes atributos. Existem dois tipos de chaves: as chaves primárias (PK) e as chaves estrangeiras (FK). As chaves primárias são o atributo, ou conjunto de atributos, que identificam individualmente cada entrada da tabela. As chaves estrangeiras servem para identificar atributos de uma tabela cujas ocorrências são referências a uma chave primária de outra tabela [73].

Tendo em mente estes conceitos foi necessário pensar na organização e nas relações entre os grupos de informações. Deste modo a base de dados irá consistir em cinco tabelas, uma primeira tabela reúne as informações dos modelos de artigos, uma segunda tabela possui as informações dos modelos de paletes, noutra tabela são armazenadas as informações referentes aos modelos de intercalares, numa quarta tabela reunidas as informações base de cada mosaico e na última são armazenadas as coordenadas dos produtos e a sua rotação para cada mosaico. Para uma mais fácil análise da estrutura da base de dados foi criado o diagrama entidade-relação apresentado na Figura 4.8 onde são expostos os campos de cada tabela e apresentados os tipos de relação entre as tabelas.

Para que as informações das várias tabelas estejam devidamente relacionadas são definidas as chaves de cada tabela, como foi anteriormente descrito. Deste modo, nas tabelas dos artigos, das paletes e dos intercalares a chave primária é o respetivo campo do ID, o que significa que a chave não irá permitir que na mesma tabela existam entradas com igual valor no campo ID. Na tabela dos mosaicos a chave primária é composta pelo ID do mosaico, pelo ID do artigo a paletizar, pelo ID do intercalar escolhido e pelo ID da palete utilizada. Deste modo a base de dados certifica que não existem entradas na tabela de mosaicos que repitam a combinação destes quatro campos. Existem ainda, na tabela dos mosaicos três chaves estrangeiras, que correspondem ao ID do artigo, ao ID da palete, e ao

ID do intercalar. Desta forma é estabelecida uma relação entre esta tabela e as tabelas dos intercalares, dos artigos e das paletes.

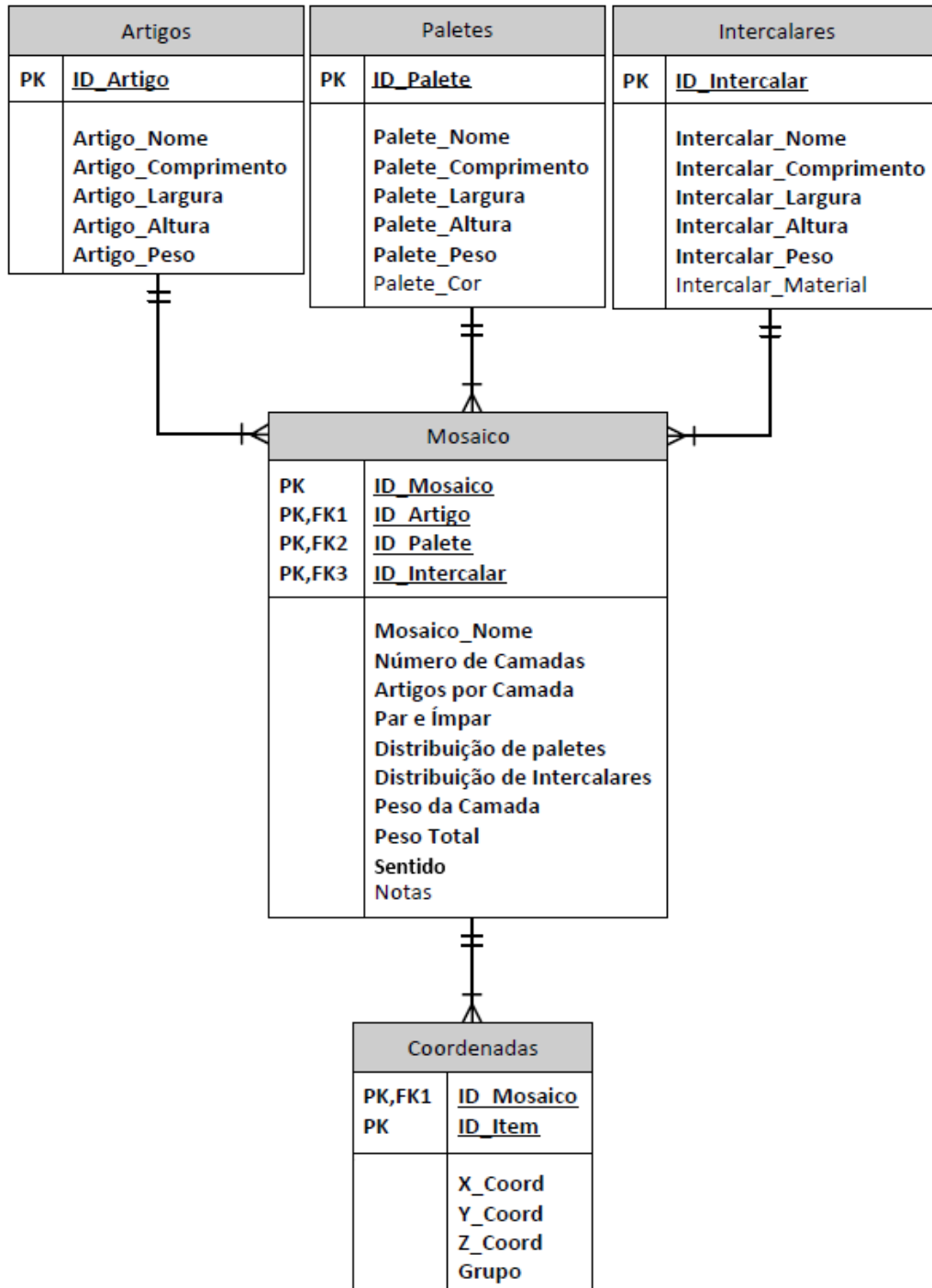


Figura 4.8: Diagrama entidade-relação da base de dados.

Na tabela das coordenadas a chave primária é composta pelos campos que indicam o ID do item e o ID do mosaico ao qual pertence o item em questão, deste modo a chave primária não permite que para um mosaico existam dois itens com o mesmo ID. Existe ainda uma chave estrangeira, o campo do ID do mosaico. Deste modo é criada uma relação entre esta tabela das coordenadas e a tabela dos mosaicos. Nesta tabela, todas as entradas com o mesmo ID do mosaico correspondem a artigos desse mesmo mosaico, sendo aqui armazenadas as informações do posicionamento na paleta, da sua rotação e quantos *packs* são agrupados no item em questão.

Ao observar o diagrama entidade-relação da base de dados podem ainda ser observadas as conexões entre as tabelas, estas identificam o tipo de relação entre as tabelas que se encontram nos extremos da ligação. Entre a tabela dos artigos e a tabela dos mosaicos pode ser observada um conector que indica uma relação de um para vários, isto é, a cada artigo pode corresponder mais que um mosaico. Por outras palavras, um modelo de artigo pode ser utilizado para construir vários mosaicos diferentes. O mesmo acontece entre a tabela das paletes e a dos mosaicos e entre a tabela dos intercalares e a dos mosaicos.

A relação entre a tabela dos mosaicos e a das coordenadas é igualmente do tipo um para vários. Isto deve-se ao facto de na tabela das coordenadas existir uma entrada para cada item de um mosaico, sendo que a um mosaico irão corresponder todos esses itens. Todas as quatro relações são do tipo um para vários, tal como representado no diagrama da Figura 4.8.

Para a criação e configuração da base de dados foi utilizado o Microsoft SQL Server Management Studio 17. No processo de criação das tabelas da base de dados foram definidos os domínios de cada atributo. Nas tabelas dos artigos (Figura 4.9), dos intercalares (Figura 4.10) e das paletes (Figura 4.11) o domínio do ID é *"bigint"*, um formato de dados que suporta números inteiros até um máximo de 8 bytes. Foi escolhido este formato pois como o ID incrementa a

Tabela_Artigos			
	Column Name	Data Type	Allow Nulls
🔑	ID_Artigo	bigint	<input type="checkbox"/>
	Artigo_Comprimento	int	<input type="checkbox"/>
	Artigo_Largura	int	<input type="checkbox"/>
	Artigo_Altura	int	<input type="checkbox"/>
	Artigo_Peso	int	<input type="checkbox"/>
	Artigo_Nome	nvarchar(MAX)	<input type="checkbox"/>

Figura 4.9: Características da tabela dos artigos.

Tabela_Intercal			
	Column Name	Data Type	Allow Nulls
🔑	ID_Intercalar	bigint	<input type="checkbox"/>
	Intercalar_Nome	nvarchar(MAX)	<input type="checkbox"/>
	Intercalar_Comprimento	int	<input type="checkbox"/>
	Intercalar_Largura	int	<input type="checkbox"/>
	Intercalar_Altura	int	<input type="checkbox"/>
	Intercalar_Peso	int	<input type="checkbox"/>
	Intercalar_Material	nvarchar(MAX)	<input checked="" type="checkbox"/>

Figura 4.10: Características da tabela dos intercalares.

Tabela_Paletes			
	Column Name	Data Type	Allow Nulls
🔑	ID_Palete	bigint	<input type="checkbox"/>
	Palete_Nome	nvarchar(MAX)	<input type="checkbox"/>
	Palete_Comprimento	int	<input type="checkbox"/>
	Palete_Largura	int	<input type="checkbox"/>
	Palete_Altura	int	<input type="checkbox"/>
	Palete_Peso	int	<input type="checkbox"/>
	Palete_Cor	nvarchar(MAX)	<input checked="" type="checkbox"/>

Figura 4.11: Características da tabela das paletes.

cada entrada que é inserida o valor neste campo pode tornar-se elevado mas, como não existem restrições de espaço de armazenamento, usar 8 bytes para armazenar estes dados não é um problema. Nos campos de caracteres o tipo de dados escolhido foi o "nvarchar(max)" já que se pretendeu não impôr um limite para o tamanho dos dados nestes campos. Nos campos referentes às dimensões e peso o tipo de dados escolhido foi o "int". Já que os dados são introduzidas em milímetros para as dimensões e em gramas para o peso prevê-se que as entradas em qualquer um destes campos não atinjam valores extremamente elevados, pelo que o espaço necessário para armazenar cada um destes campos não excederá os 4 bytes deste formato de dados.

Na tabela das coordenadas (Figura 4.12) os campos que possuem ID têm o formato de dados "bigint", à semelhança do que acontece nas três tabelas anteriores.

Para os campos das coordenadas X e Y, que localizam na palete o ponto em que se situa canto superior esquerdo do item em questão têm o formato "int", já que os valores que estes campos recebem são inteiros e não ultrapassam os valores das dimensões das paletes com a sua margem, sendo mais que suficiente

Tabela_Coorden			
	Column Name	Data Type	Allow Nulls
🔑	ID_Mosaico	bigint	<input type="checkbox"/>
🔑	ID_Item	bigint	<input type="checkbox"/>
	Xcoord	int	<input type="checkbox"/>
	Ycoord	int	<input type="checkbox"/>
	Zcoord	bit	<input type="checkbox"/>
	grupo	tinyint	<input type="checkbox"/>

Figura 4.12: Características da tabela das coordenadas.

os 4 bytes disponibilizados por este formato de dados. Para o campo que indica se o item se encontra rodado ou não (campo "Zcoord") optou-se por utilizar o tipo de dados "bit" já que este campo apenas receberá o valor 0 caso o item não esteja rodado e o valor 1 caso o item esteja rodado. Para o campo que indica quantos *packs* constituem o item foi escolhido o formato "tinyint", visto que neste campo os valores são sempre maiores que zero e os grupos nunca ultrapassam os dez *packs*, um byte é suficiente para armazenar este campo.

Na tabela dos mosaicos (Figura 4.13) os campos que recebem ID seguem a mesma ideia das tabelas anteriores ou seja, têm o formato de dados "bigint". Para os campos que recebem caracteres, como o campo do nome do mosaico e o campo de notas/observações, como não se pretendeu impôr um limite de caracteres o

Tabela_Mosaicos			
	Column Name	Data Type	Allow Nulls
🔑	ID_Mosaico	bigint	<input type="checkbox"/>
🔑	ID_Artigo	bigint	<input type="checkbox"/>
🔑	ID_Palete	bigint	<input type="checkbox"/>
🔑	ID_Intercalar	bigint	<input type="checkbox"/>
	Mosaico_Nome	nvarchar(MAX)	<input type="checkbox"/>
	Num_Camadas	tinyint	<input type="checkbox"/>
	Agrupamentos	tinyint	<input type="checkbox"/>
	Artigos_por_camada	bigint	<input type="checkbox"/>
	Par_Impar	tinyint	<input type="checkbox"/>
	Peso_Camada	int	<input type="checkbox"/>
	Peso_Total	int	<input type="checkbox"/>
	Distribuicao_Paletes	int	<input type="checkbox"/>
	Distribuicao_Intercalares	int	<input type="checkbox"/>
	sentido	tinyint	<input type="checkbox"/>
	notas	nvarchar(MAX)	<input checked="" type="checkbox"/>

Figura 4.13: Características da tabela dos mosaicos.

formato *"nvarchar(max)"* foi o mais indicado. No campo que recebe o número de camadas (*"Num\_camadas"*) os valores recebidos serão sempre menores que 20, pelo que o formato *"tinyint"* é capaz de armazenar estes dados.

Para o campo que recebe o número de itens que existe numa camada independentemente do número de *packs* que cada item agrupe (campo *"Agrupamentos"*) o formato escolhido foi o *"tinyint"* já que estes valores serão sempre maiores que zero e nunca maiores que vinte e cinco.

No campo que recebe o número de *packs* por camada (*"Artigos\_por\_camada"*) os valores a receber podem-se tornar algo complexos. O número de *packs* nunca ultrapassará os 100 para os mosaicos que não sejam feitos com sucessão de camadas personalizada. Para os mosaicos com sucessão de camadas personalizada este campo receberá um número em que cada par de algarismos consecutivos indica o número de *packs* numa camada (ver Figura 4.14). Deste modo, como a criação deste tipo de mosaicos é limitada a 10 camadas, o valor a receber neste campo nunca terá mais de 20 dígitos, ou seja, o formato *"bigint"* é capaz de o receber já que este é um formato de 8 bytes (o seu limite é  $2^{64}$ ).

No campo onde é indicada a sucessão de camadas (*"Par\_Ímpar"*) o tipo de dados é o *"tinyint"* já que este campo apenas recebe valores de 0 a 4, o 0 significa todas as camadas são iguais, o 1 significa que a sucessão de camadas é do tipo "par/ímpar" em que a camada ímpar é o espelhamento da camada par sob um eixo horizontal, o 2 significa que a sucessão de camadas é do tipo "par/ímpar" em que a camada ímpar é o espelhamento da camada par sob um eixo vertical, o 3 significa que a sucessão de camadas é do tipo "par/ímpar" em que a camada ímpar é a rotação da camada par em 180°, o 4 significa que a sucessão de camadas é do tipo personalizado, pelo que cada camada terá o seu próprio *layout*.

Nos pesos de cada camada (*"Peso\_camada"*) e da palete (*"Peso\_Total"*), como estes são definidos em gramas estes campos podem atingir valores de 6 dígitos, pelo que o mais apropriado para estes campos será o tipo de dados *"int"*.

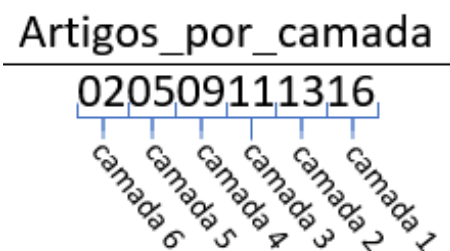


Figura 4.14: Exemplo de um número recebido no campo *"Artigos\_por\_camada"* para um mosaico com sucessão de camadas personalizada

Para armazenar a maneira como estão distribuídos os intercalares e as paletes recorre-se a uma variável em que cada bit corresponde ao intervalo entre camadas. Se o bit estiver a 0 então não há intercalar/paleta, se estiver a 1 há intercalar/paleta. Estas variáveis estão formatadas de acordo com a ordem *little-endian*. Como o máximo permitido são 20 camadas então o formato de dados na base de dados terá de suportar pelo menos 21 bits, o que indica que o formato *"int"*, de 4 bytes (32 bits), é indicado para estes dois campos "Distribuicao\_Paletes" e "Distribuicao\_Intercalares".

Por fim, o campo "sentido" indica a direção de formação do mosaico, este irá receber o número 1 caso a direção seja vertical e irá receber o número 2 caso a direção seja horizontal. Como tal o formato *"tinyint"* é suficiente para receber estes valores.

Para exemplificar como são recebidas as informações dos artigos, paletes, intercalares e mosaicos nas tabelas da base de dados é tomado um exemplo do caso de estudo. O mosaico *"pack Laranja 4x3 (1,25 L)"* apresentado no Apêndice A é um bom exemplo para demonstrar o pretendido. A paleta utilizada é uma paleta de formato "EURO" pelo que as suas dimensões são 1200x800x150 mm e o seu peso 20 kg (20.000 g). Assim estas informações são armazenadas na base de dados de acordo com o apresentado na Figura 4.15.

Os intercalares utilizados são intercalares de cartão liso, com um mm de altura, 1200 mm de comprimento e 800 mm de largura. Deste modo o seu armazenamento na base de dados terá o formato apresentado na Figura 4.16.

O artigo a paletizar é um *pack* de garrafas com comprimento de 350 mm, largura de 263 mm, altura de 308 mm e peso de 15 kg. Deste modo, este produto será recebido na base dados de acordo com o apresentado na Figura 4.17.

Tabela_Paletes ×							
ID_Paleta	Paleta_Nome	Paleta_Compri...	Paleta_Largura	Paleta_Altura	Paleta_Peso	Paleta_Cor	
10030	Paleta_LPR (euro)	1200	800	150	20000	natural	

Figura 4.15: Informação recebida na tabela das paletes da base de dados para o exemplo apresentado.

Tabela_Intercal ×							
ID_Intercalar	Intercalar_Nome	Intercalar_Co...	Intercalar_Larg...	Intercalar_Altura	Intercalar_Peso	Intercalar_Mat...	
9	Cartão_Liso	1200	800	1	200	cartao	

Figura 4.16: Informação recebida na tabela dos intercalares da base de dados para o exemplo apresentado.

Tabela_Artigos						
ID_Artigo	Artigo_Compri...	Artigo_Larqura	Artigo_Altura	Artigo_Peso	Artigo_Nome	
40012	350	263	308	15000	Laranja_1,25_4x3	

Figura 4.17: Informação recebida na tabela dos artigos da base de dados para o exemplo apresentado.

No mosaico em análise pretende-se que a palete seja construída com 4 camadas, com 10 artigos por camada, com uma sucessão de camadas par/ímpar com a camada ímpar espelhada sob um eixo horizontal, com intercalares em todas as camadas e no topo e apenas com palete na base. Com estas informações e por observação do mosaico determina-se que é mais viável a formação do mosaico no sentido vertical e com base em 7 grupos de *packs* que terão as suas coordenadas de acordo com a Figura 4.18

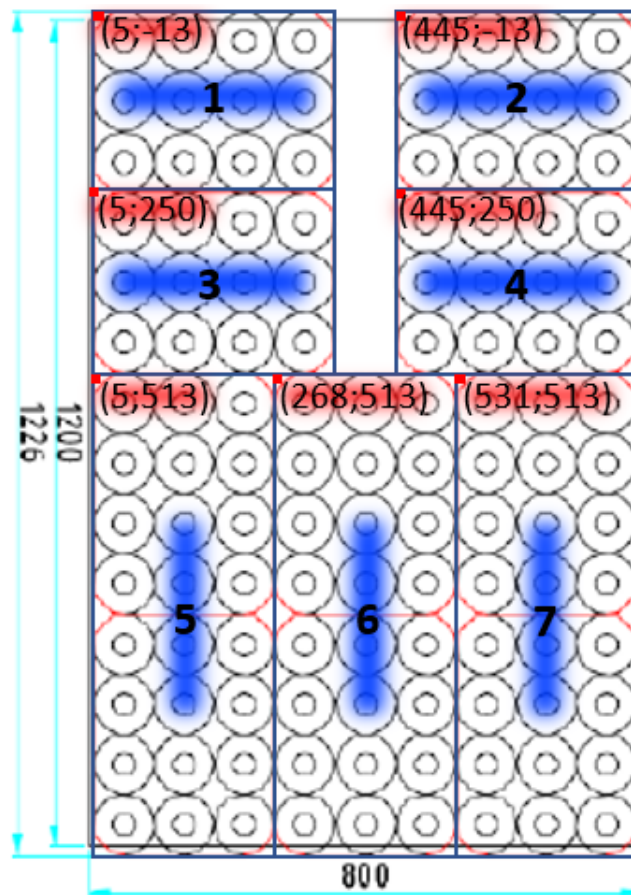


Figura 4.18: Coordenadas atribuídas e grupos formados na construção do mosaico.

O facto de se pretenderem intercalares em todas as camadas e no topo resulta num número binário "11111" que convertido para decimal é "31", é este valor que a base de dados receberá no campo de distribuição de intercalares. O mesmo acontece para a distribuição de paletes, como apenas se pretende paletes na base o número em binário é 1, tal como em decimal.

O peso de cada camada, previamente calculado pela interface, é de 150 kg, e o peso total da paletes construída é de 621 kg, pelo que estes campos irão receber os valores 150000 e 621000, respetivamente.

Como se pretende que a formação do mosaico seja feita na vertical então o campo sentido recebe o valor 1 e como se pretende que a camada ímpar seja o espelhamento da camada par sob um eixo horizontal o campo "Par\_Ímpar" recebe o valor 1.

Nos campos de ID do artigo, do intercalar e da paletes, os valores são os que são apresentados nas Figuras 4.17, 4.16 e 4.15 respetivamente.

Posto isto, a linha da tabela dos mosaicos que contém este mosaico tem o formato apresentado na Figura 4.19.

Por fim, a tabela das coordenadas irá conter várias linhas que se referem a artigos deste mesmo mosaico. Como já foi descrito, nos campos "Xcoord" e "Ycoord" serão recebidos valores das coordenadas de posicionamento na paletes, no campo "Zcoord" é retido se o item em questão é rodado ou não e no campo "grupo" fica a informação que indica quantos *packs* tem o item em questão. Para se armazenarem as informações de cada item é atribuído um ID a cada um de acordo com a ordem de formação (numeração apresentada na Figura 4.18), no entanto nos casos que existem camadas ímpares é necessário armazenar um novo conjunto de coordenadas e distinguir estes itens dos anteriores, para isso foi definido que ao ID dos itens da camada ímpar será adicionado 100 de modo a que haja uma distinção clara. Deste modo os itens do mosaico apresentado como exemplo são retidos na base de dados conforme apresentado na Figura 4.20.

ID_Mosaico	ID_Artigo	ID_Paletes	ID_Intercalar	Mosaico_Nome	Agrupamentos	Num_Camadas
50004	40012	10030	9	Laranja 4x3 (1,25L)	7	4

Artigos_por_ca...	Par_Ímpar	Peso_Camada	Peso_Total	Distribuicao_P...	Distribuicao_In...	notas	sentido
10	1	150000	621000	1	31	este é um mosa...	1

Figura 4.19: Informação recebida na tabela dos mosaicos da base de dados para o exemplo apresentado.

ID_Mosaico	ID_Item	Xcoord	Ycoord	Zcoord	grupo
50004	1	5	-13	1	1
50004	2	445	-13	1	1
50004	3	5	250	1	1
50004	4	445	250	1	1
50004	5	5	513	0	2
50004	6	268	513	0	2
50004	7	531	513	0	2
50004	101	5	950	1	1
50004	102	445	950	1	1
50004	103	5	687	1	1
50004	104	445	687	1	1
50004	105	5	-13	0	2
50004	106	268	-13	0	2
50004	107	531	-13	0	2

Figura 4.20: Informação da tabela das coordenadas para o exemplo apresentado.

A última ação relacionada com a preparação da base de dados é a configuração do nome da origem da base de dados (*DSN-Data Source Name*). Para isto é utilizada a ferramenta do Windows "Administrador da Origem de Dados ODBC", onde é definido que o DSN desta base de dados será o "WINCCSQL1".

#### 4.6.1.2 Interface Homem-Máquina

Depois de apresentada a estrutura e a lógica de funcionamento da base de dados é abordada a interface Homem-máquina que será o ponto de contacto entre o utilizador e a ilha de paletização. A base de dados, apesar de ser transparente ao utilizador é um ponto fulcral no funcionamento da interface já que grande parte das funcionalidades que são disponibilizadas incluem algum tipo de acesso à base de dados que tanto pode ser de escrita como de leitura.

No seguimento das ideias introduzidas acerca da arquitetura da interface, foi tomado como ponto de partida para o desenvolvimento da interface a criação de uma base gráfica sob a qual se possa trabalhar e evoluir. Esta base gráfica traduz-se na criação dos botões, menus e janelas onde as ações do utilizador ganham forma. Deste modo, seguindo o esquema apresentado anteriormente na Figura 4.5, foi criado o ecrã-base para a interface, capaz de alternar ente vários menus. Este é o ponto de partida para a utilização da interface e, este ecrã, funciona como base dos menus "Home", "Comando Geral", "Informação da Ilha", "Gestão de Produto", "Alarmes" e "Gestão de Utilizadores" que podem ser consultados através da barra de navegação (ponto 1 na Figura 4.21). Estes menus aparecem na zona do ecrã que se encontra vazia na Figura 4.21.

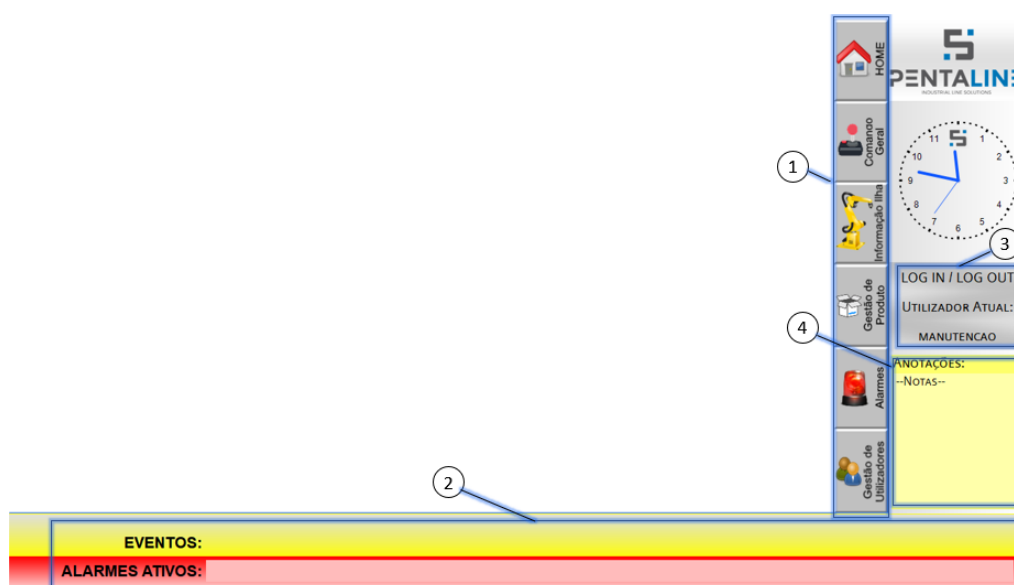


Figura 4.21: Base do ecrã da interface Homem-máquina desenvolvida.

Neste ecrã-base existe ainda uma zona reservada à notificação de alarmes ativos e eventos recentes (ponto 2 na Figura 4.21). Para inserir as credenciais de acesso a níveis superiores e para apresentar o nível de acesso atual está reservada a área marcada com o ponto 3. Por fim, existe ainda uma área reservada para anotação de recados ou observações marcada na Figura 4.21 com o ponto 4.

Com recurso às ferramentas disponibilizadas pelo WinCC são criados os elementos gráficos, no entanto para que estes desencadeiem grande parte das ações é necessário recorrer à criação de *scripts*, neste caso em VB. Assim, a interação com a barra de navegação neste ecrã traduz-se na atualização de uma variável "Screen\_ID" o que, segundo o *Script* B.1 do Apêndice B fará apresentar o menu pretendido na zona do ecrã destinada a esse fim.

Agora será feita uma análise das funcionalidade existentes em cada menu, deixando para última análise os menus de comando geral e de gestão de produto. O menu "Home" não contém muitas funcionalidades, este serve principalmente para quatro ações. Estas são a consulta de documentação acerca da ilha (esquemas elétricos, pneumáticos, *layout*, etc.), consulta de informação acerca da Pentaline (contactos, morada, etc.), utilização de um *browser* para acesso à Internet (apenas para níveis de acesso superiores) e neste menu é também possível encerrar a aplicação (reservado a níveis de acesso superiores).

O menu "Informação da Ilha" é o que terá o aspeto mais alterado de umas soluções para outras, isto é, como no plano principal é apresentada uma imagem da ilha de paletização esta estará diretamente dependente do tipo de ilha em que a interface é integrada (ver Figura 4.22). Neste menu são apresentadas informa-

ções da ilha que sejam pertinentes. Na interface desenvolvida são principalmente contempladas informações acerca do número de *packs* ou número de camadas processados em cada zona. Estas informações são recolhidas pelo PLC da ilha que posteriormente as disponibiliza em diversas variáveis criadas para este efeito. Deste modo a interface, pela consulta dessas variáveis, mantém a informação que é apresentada ao utilizador sempre atualizada.

O menu "Alarmes" tem como função apresentar todas as mensagens de alarme e eventos ativas bem como o histórico das mesmas. Deste modo torna-se possível construir uma linha do tempo dos acontecimentos na ilha em caso de existir algum problema que surja ao longo do seu funcionamento (ver Figura 4.23). A maioria dos alarmes e eventos que possam ser apresentados neste ecrã têm uma relação direta com o tipo de instalação em que a interface está inserida. Deste modo os alarmes e eventos que estejam relacionados com informações que vêm ilha são diferentes de solução para solução.

No entanto há um grupo de alarmes e eventos que são desencadeados apenas por ações realizadas na consola, assim estes encontram-se já configurados e são lançados assim que seja necessário. Os eventos já configurados referem-se a ordens que a consola está a enviar, tal como paragem da produção e o arranque do tapete de *packs*, nos alarmes já configurados é considerada a falta de dados acerca das dimensões do *gripper* do robô manipulador de *packs* (ver Figura 4.24). Esta configuração do *gripper* apenas precisa de ser feita na primeira utilização da ilha, depois ficará armazenada na memória do PLC.

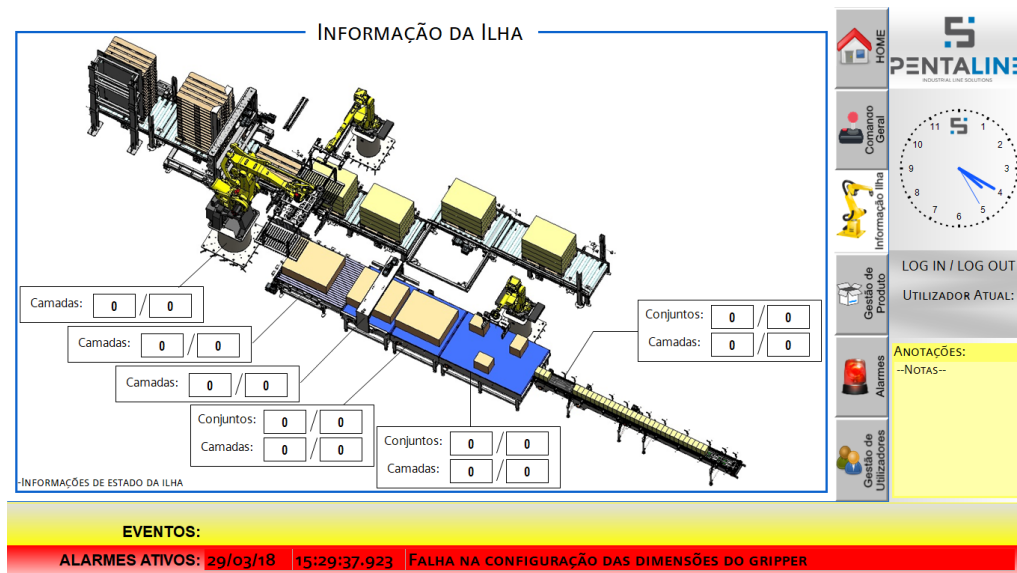


Figura 4.22: Ecrã apresentado ao utilizador no menu de informações da ilha.

ID	Data	Hora	Descrição
1	05/04/18	12:22:49.932	A consola emitiu uma ordem de arranque automático das instalações
2	05/04/18	12:17:06.226	A consola emitiu uma nova ordem de produção
3	05/04/18	12:16:57.929	A consola emitiu uma ordem de paragem do tapete de packs
4	05/04/18	12:16:54.874	A consola emitiu uma ordem de arranque do tapete de packs
5	05/04/18	12:16:40.026	Falha na configuração das dimensões do gripper
6	1000202	05/04/18	WC CRT:DESKTOP-M0S8NUK:DESKTOP-M0S8NUK has activated Runtime
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			

EVENTOS: 05/04/18 12:22:49.932 A CONSOLA EMITIU UMA ORDEM DE ARRANQUE AUTOMÁTICO DAS INSTALAÇÕES

ALARMES ATIVOS: 05/04/18 12:16:40.026 FALHA NA CONFIGURAÇÃO DAS DIMENSÕES DO GRIPPER

Figura 4.23: Ecrã apresentado ao utilizador no menu de alarmes.

ID	Name	Alarm text	Alarm class	Trigger tag	Trigge..	Acknowledg...	Ackn...
100	Configuração_Grip...	Falha na configuração das dimensõ	Errors	alarmes_grip...	0	alarmes_grip...	1
2	Ordem de arranque..	A consola emitiu uma ordem de arr	Warnings	eventos_hmi	0	<No tag>	0
3	Ordem de arranque..	A consola emitiu uma ordem de arr	Warnings	eventos_hmi	1	<No tag>	0
4	Ordem de arranque..	A consola emitiu uma ordem de arr	Warnings	eventos_hmi	2	<No tag>	0
5	Ordem de paragem..	A consola emitiu uma ordem de par	Warnings	eventos_hmi	3	<No tag>	0
6	Requisição de interr..	A consola emitiu uma ordem de inte	Warnings	eventos_hmi	4	<No tag>	0
7	Nova ordem	A consola emitiu uma nova ordem d	Warnings	eventos_hmi	5	<No tag>	0
8	Requisição de fim d...	A consola emitiu uma ordem de fim	Warnings	eventos_hmi	6	<No tag>	0
9	Requisição de reini...	A consola emitiu uma ordem de reir	Warnings	eventos_hmi	7	<No tag>	0
10	Requisição de reini...	A consola emitiu uma ordem de reir	Warnings	eventos_hmi	8	<No tag>	0

Figura 4.24: Configurações dos eventos e alarmes já configurados.

Os alarmes encontram-se configurados na categoria "Errors", categoria que faz com que as suas mensagens surjam sombreadas a vermelho e faz também com que a notificação mais recente surja na barra "Alarmes Ativos". Esta barra consiste numa janela de alarmes semelhante à apresentada no menu de alarmes mas tem a particularidade de ter configurado um filtro que apenas permite a visualização de notificações da categoria "Errors". Já os eventos encontram-se configurados na categoria "Warnings" que faz com que estes surjam sombreados a amarelo e também faz com que a notificação mais recente surja na barra "Eventos", na parte inferior do ecrã principal. O funcionamento desta barra de "Eventos" é semelhante ao descrito para a barra "Alarmes Ativos" com a diferença de ter um filtro que apenas permite a visualização de notificações da categoria "Warnings".

A configuração de novos alarmes que sejam comunicados pelo PLC é feita de forma semelhante aos que já se encontram configurados, a diferença encontra-se

na variável definida na coluna "Trigger tag" que em vez de ser uma variável apenas interna da HMI é uma variável da HMI que se encontra ligada a uma variável do PLC, deste modo, a variável do PLC que reporta alarmes estará sempre em sintonia com a variável da HMI que desencadeará a sinalização ao utilizador. Esta ligação entre as duas variáveis é feita na definição da variável da HMI, onde existe um campo onde é indicado o nome da variável do PLC com a qual se pretende fazer a ligação. Esta ligação de variáveis não serve apenas para os alarmes e eventos, a mesma técnica é utilizada para transmitir ao PLC várias informações, nomeadamente as ordens de arranque e paragem da ilha.

O menu de "Gestão de Utilizadores" é apenas útil aos utilizadores que tenham credenciais para efetuar *log in*. Para aceder a este menu não é necessário efetuar *log in*, no entanto, como só é apresentada informação referente ao utilizador atual e aos níveis de acesso inferiores, sem ter sessão iniciada não é apresentada qualquer informação. No seguimento desta ideia, este menu possibilita ao nível de acesso superior a criação de novos utilizadores e a alteração dos parâmetros dos utilizadores já existentes, como o tempo de *log off* e a palavra passe. Aos outros níveis de acesso apenas é permitida a visualização das informações do seu próprio utilizador e alteração das configurações de tempo de *log off* e de palavra passe. Na Figura 4.25 pode ser observado o aspeto do menu apresentado ao nível superior de acesso.

No menu de "Gestão de Produto" é onde grande parte do trabalho realizado com *scripts* se foca. Através deste menu torna-se possível adicionar dados acerca dos produtos que se pretendem paletizar, adicionar dados sobre as paletes e os

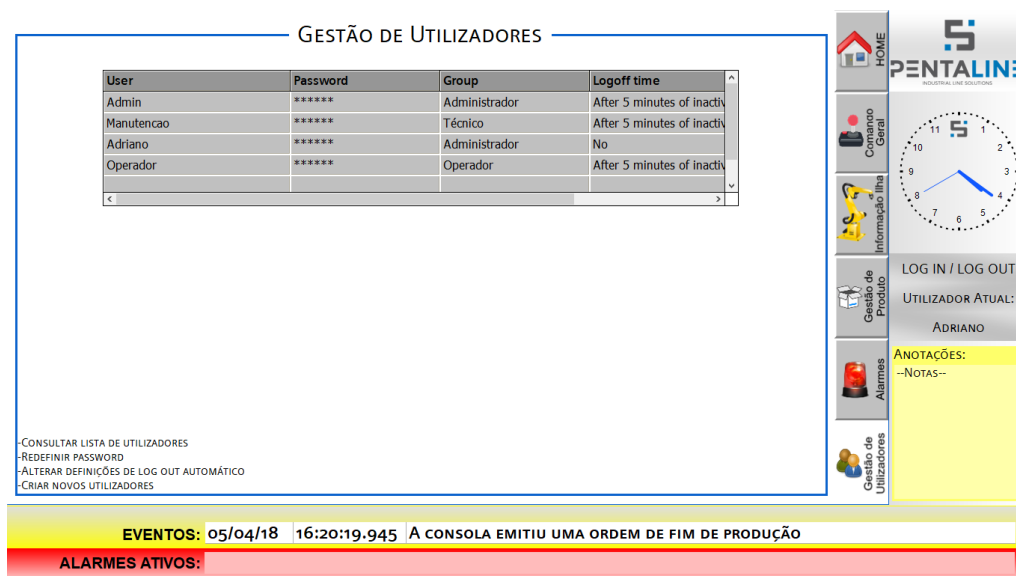


Figura 4.25: Ecrã apresentado ao utilizador no menu de alarmes.

intercalares que farão parte das paletes construídas e criar os modelos das paletes a construir, cada um com o seu mosaico. Quando o utilizador navega até a este menu é lhe apresentado um ecrã com cinco principais áreas de foco. Ao observar a Figura 4.26 é possível perceber que o ponto 1 aponta para a barra de navegação no topo deste menu. É através desta que é possível alternar entre a gestão de artigos, de paletes, de intercalares e de mosaicos. Para a gestão destes itens recorre-se a 3 janelas que vão alterando o seu conteúdo de acordo com a opção que o utilizador escolheu na barra de navegação deste menu (pontos 2, 3 e 4 da Figura 4.26) e a uma janela de impressão de caracteres para apresentação de mensagens relacionadas com a atividade do utilizador neste menu (ponto 5 da Figura 4.26).

A janela assinalada com o número 2 tem como principal função a criação de novas entradas. Nesta, são apresentados campos para introdução dos dados referentes às características dos itens a adicionar, a exceção é na gestão de mosaicos em que nesta janela é apresentado um botão que dá acesso ao "Criador de Mosaicos".

Na janela assinalada com o número 3 é apresentada uma janela que permite eliminar entradas da base de dados. É apresentado um campo onde o utilizador introduz o ID da entrada que pretende eliminar e um botão que irá desencadear a eliminação da entrada na tabela correspondente ao submenu que tenha sido indicado na barra de navegação no topo.

Na janela assinalada com o número 4 é apenas apresentado um botão. Dependendo do submenu selecionado o botão é alterado e este faz com que seja

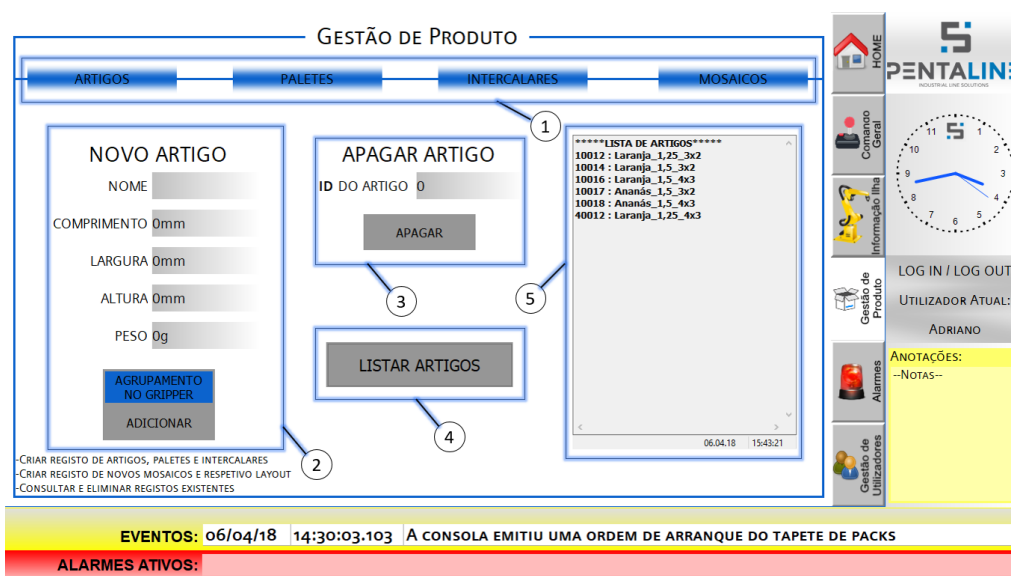


Figura 4.26: Menu que dá acesso à gestão dos itens da base de dados.

NOVO ARTIGO	NOVA PALETE	NOVO INTERCALAR
NOME <input type="text"/>	NOME <input type="text"/>	NOME <input type="text"/>
COMPRIMENTO <input type="text" value="0mm"/>	COMPRIMENTO <input type="text" value="0mm"/>	COMPRIMENTO <input type="text" value="0mm"/>
LARGURA <input type="text" value="0mm"/>	LARGURA <input type="text" value="0mm"/>	LARGURA <input type="text" value="0mm"/>
ALTURA <input type="text" value="0mm"/>	ALTURA <input type="text" value="0mm"/>	ALTURA <input type="text" value="0mm"/>
PESO <input type="text" value="0g"/>	PESO <input type="text" value="0g"/>	PESO <input type="text" value="0g"/>
<input type="button" value="AGRUPAMENTO NO GRIPPER"/>	<input type="text" value="COR"/>	MATERIAL <input type="text"/>
<input type="button" value="ADICIONAR"/>	<input type="button" value="ADICIONAR"/>	<input type="button" value="ADICIONAR"/>

Figura 4.27: Janelas de criação de artigos, paletes e intercalares (da esquerda para a direita).

apresentada uma listagem (ID e nome) das entradas ordenadas por ordem crescente do ID na janela assinalada com o número 5.

Sendo que a gestão de mosaicos tem diversas particularidades a sua análise será deixada para depois da gestão de artigos, paletes e intercalares.

Por trás do funcionamento deste ecrã de "Gestão do produto" estão alguns *scripts* que permitem realizar a escrita, consulta e eliminação de entradas a base de dados. Estes *scripts* mais importantes são disponibilizados para consulta no Apêndice B.

No que toca aos submenus de gestão de artigos, intercalares e paletes estes contam com a janela assinalada com o número 2 na Figura 4.26 para aceder à base de dados e escrever na mesma uma nova entrada com os dados recolhidos. As janelas que podem aparecer em cada um destes três casos são apresentadas na Figura 4.27. A introdução de dados, o tratamento desta informação e a sua escrita na base de dados é semelhante para cada uma destas três situações pelo que será feita uma análise mais detalhada aos *scripts* e modo de funcionamento do ecrã "Novo Artigo".

No ecrã "Novo Artigo" existem campos de introdução de dados, um botão para submissão dos dados e existe ainda um botão que não está presente no ecrã "Nova Palete" nem no ecrã "Novo Intercalar", este dá acesso a uma ferramenta de comparação de tamanho entre artigos ou grupos de artigos e o *grripper* do robô manipulador de *packs*.

Cada campo de introdução de dados tem atribuído um formato de dados, uma variável e, em certos casos a unidade dos dados a introduzir. No caso da criação de um novo artigo existem cinco campos, o primeiro é o nome que é atribuído ao novo produto, este campo não tem restrição de caracteres já que esta é tratada

como uma entrada de texto na variável "caixa\_nome" do tipo "WString" de modo a evitar restrições no tamanho do nome. No campo onde é inserido o comprimento do produto apenas são permitidos algarismos e é atribuída a unidade milímetros e a variável atualizada com os valores introduzidos neste campo é a "caixa\_c", uma variável do tipo "Int", de 16 bits, ou seja, capaz de armazenar um número até 32767. No campo para receber a largura, à semelhança do anterior, apenas são permitidos algarismos e a unidade é milímetros, a variável atualizada com o valor aqui introduzido é a "caixa\_l", também do tipo "Int". A altura é introduzida também apenas em números e em milímetros, a variável que recebe este valor é a "caixa\_h" do tipo "Int". Por fim, o campo que recebe o peso do *pack* em gramas, tem associada a variável "caixa\_p" do tipo "Int".

O passo seguinte é ligar à base de dados e escrever na tabela pretendida os valores recebidos nestas variáveis de forma que possam ser consultados posteriormente. Para isso recorre-se a um *script* em VB. Este *script* é corrido no evento "click" do botão "Adicionar". Este *script* pode ser resumido em 3 etapas: recolha e tratamento dos dados, acesso e escrita na base de dados e término da criação de um novo artigo. No início são definidas todas as variáveis utilizadas com o comando "Dim". Depois das declarações segue-se a primeira etapa, nesta os dados são recolhidos das variáveis da HMI e copiados para variáveis do *script*. Na criação de novos artigos são cinco as variáveis relacionadas com o artigo a trabalhar no script: "nome", "comp", "larg", "alt" e "peso" que correspondem, respetivamente, ao nome, comprimento, largura, altura e peso do novo artigo. A recolha dos dados das variáveis da HMI para as variáveis do *script* é feita da seguinte forma:

```
nome =HMIRuntime.Tags("caixa_nome").Read
comp =HMIRuntime.Tags("caixa_c").Read
larg =HMIRuntime.Tags("caixa_l").Read
alt =HMIRuntime.Tags("caixa_h").Read
peso =HMIRuntime.Tags("caixa_p").Read
```

A segunda etapa, acesso e escrita na base de dados, começa com uma condição de que só se todos os campos estiverem preenchidos é possível proceder à criação de um novo artigo, depois disso é definido o "SQL statement" que, como neste caso se pretende inserir uma entrada, é do tipo "INSERT INTO". Esta declaração inclui o nome da tabela onde se pretendem inserir dados, o nome das colunas nas quais se pretende inserir informação e as variáveis que contêm os valores a inserir na tabela. Deste modo esta declaração tem o seguinte formato:

```
"INSERT INTO table_name(column1, column2, column3, ...) VALUES (value1, value2, value3, ...);"
```

Depois desta declaração segue-se a "Connection String". Esta reúne os quatro principais parâmetros de acesso à base de dados: o "Provider" (fornecedor do servidor), o DSN (definido anteriormente pelo programador), o UID (nome de

utilizador para acesso à base de dados) e o PWD (password para acesso à base de dados). Deste modo a "Connection String" tem o seguinte formato:

```
"Provider=MSDASQL;DSN=dsnName;UID=MyUserID;PWD=MyPassword;"
```

Após estas declarações é aberta a ligação à base de dados e é, então, executado o "SQL statement". Depois disto é terminada a ligação à base de dados de modo a deixar o acesso livre para um novo acesso à mesma. A parte do código à qual se referem estes últimos três parágrafos pode ser consultada no *Script* B.2 do Apêndice B.

Por fim, o término da criação de um novo artigo consiste no envio de uma caixa de mensagem a confirmar ao utilizador a criação do artigo, na impressão de uma mensagem com a mesma informação na janela de relatórios e, por último, é feito o *reset* dos valores das variáveis da HMI. Assim sendo, o código em Visual Basic que permite realizar estas operações pode ser consultado no *Script* B.3 do Apêndice B.

Para terminar este *script* existe ainda uma condição que lança uma *message box* com uma descrição de erro caso exista algum durante as operações realizadas com a base de dados.

Nesta janela de criação de artigos existe ainda uma funcionalidade que previne a criação de entradas com o mesmo nome. Assim que é detetada uma alteração do texto do campo "Nome" é feita uma ligação à base de dados e é procurada uma entrada com o nome inserido, se for retornada uma resposta positiva então o texto do campo "Nome" é colocado a 0 e surge no ecrã uma mensagem a indicar o sucedido. O código que desencadeia estas ações pode ser consultado no *Script* B.4 do Apêndice B.

Por fim, é ainda importante referir uma utilidade desenvolvida em particular para auxiliar na criação de novos artigos. Para se ter acesso a esta é utilizado o botão "Agrupamento no Gripper" que se encontra por cima do botão "Adionar". Este botão dá acesso ao ecrã da Figura 4.28 onde pode ser criado um artigo da mesma maneira que nesta janela, no entanto este ecrã possibilita que durante a criação do artigo o utilizador tenha uma perceção da relação entre as dimensões do artigo e as dimensões do *grripper* do robô de formação de mosaico, possibilitando ainda testar até quantos artigos o robô consegue manipular em simultâneo.

Neste ecrã é ainda possível configurar as dimensões do *grripper* bem como a altura de pega em relação ao tapete, no entanto essas configurações encontram-se vedadas aos níveis inferiores de acesso, reservando-se apenas ao programador.

A próxima funcionalidade a analisar será a de apagar entradas da tabela a partir do seu ID. Esta funcionalidade é obtida de forma semelhante para os artigos, paletes e intercalares, havendo algumas particularidades para os mosaicos. As janelas que permitem estas ações são as apresentadas na Figura 4.29. Para se



Figura 4.28: Ecrã de agrupamento de novo artigo no *gripper*.

ter conhecimento do ID do item que se pretende apagar é necessário pedir a lista dos mesmos. Para isso foi desenvolvida uma funcionalidade que será analisada depois da funcionalidade "apagar".

Para a criação desta funcionalidade seguiu-se uma lógica semelhante à seguida nos *scripts* da criação de entradas nas tabelas, isto é, primeiro são declaradas as variáveis, depois o valor introduzido no campo "ID" que se encontra armazenado na variável da HMI "ID\_caixa\_del" (para o caso da gestão de artigos) é copiado para uma variável do *script* e depois disso são feitas as ligações à base de dados

<p><b>APAGAR ARTIGO</b></p> <p>ID DO ARTIGO 0</p> <p>APAGAR</p>	<p><b>APAGAR PALETE</b></p> <p>ID DA PALETE 0</p> <p>APAGAR</p>
<p><b>APAGAR INTERCALAR</b></p> <p>ID INTERCALAR 0</p> <p>APAGAR</p>	<p><b>APAGAR MOSAICO</b></p> <p>ID DO MOSAICO 0</p> <p>APAGAR</p>

Figura 4.29: Janelas de eliminação de artigos, paletes e intercalares e mosaicos.

e executadas as ações necessárias. Deste modo, um dos pontos mais importantes a referir para esta funcionalidade é o "SQL statement" que, como se pretendem eliminar entradas, é do tipo "DELETE FROM" (ou apenas "DELETE"). Esta declaração segue o seguinte formato:

```
"DELETE FROM table_name WHERE condition;"
```

em que "table\_name" refere-se ao nome da tabela da qual se pretende apagar entradas e "condition" refere-se à condição que se verifica como verdadeira nas linhas a eliminar, que neste caso é ter o ID igual ao indicado pelo utilizador. Deste modo, depois das declarações de variáveis, o código deste *script* apresenta as seguintes linhas:

```
ID_apagar =(HMIRuntime.Tags("ID_caixa_del").Read)
strSQL = "DELETE Tabela_Artigos WHERE ID_Artigo = '" & ID_apagar & "'"
```

Nestas linhas é possível perceber que o valor da variável que está associada ao campo de introdução é copiado para a variável "ID\_apagar", depois disso é feita a declaração que indica que se pretende apagar da "Tabela\_Artigos" a linha que contém no campo "ID\_Artigo" um valor igual ao indicado pelo utilizador. A estas linhas segue-se a ligação à base de dados, a execução do comando de apagar e o fecho da ligação, de forma semelhante ao que acontece na funcionalidade anterior. Por fim, caso a operação tenha sucesso, a variável que está associada ao campo de entrada é colocada a 0 e é enviada uma mensagem de confirmação ao utilizador. Caso não haja sucesso é mostrada uma mensagem com a descrição do erro de forma semelhante ao que é feito na funcionalidade "adicionar".

É importante referir que, quando o utilizador tenta apagar uma entrada que esteja referida noutra tabela surge uma mensagem de erro a indicar que não é possível eliminar o item pretendido já que colocaria em causa o funcionamento da base de dados. Um exemplo disto é o utilizador tentar apagar um artigo que é utilizado em um ou mais mosaicos, a eliminação deste iria gerar falhas quando fosse necessário utilizar algum dos mosaicos em questão, já que a HMI iria, a partir do ID armazenado na tabela dos mosaicos, consultar as características do artigo na tabela dos artigos. Esta gestão da coerência entre tabelas não resulta do código desenvolvido na HMI, resulta de um bom projeto da base de dados, das chaves primárias e estrangeiras de cada tabela e das relações estudadas e criadas entre as mesmas.

Para a funcionalidade "apagar" a particularidade que a gestão de mosaicos apresenta prende-se com o facto de existir uma tabela à qual o utilizador não acede diretamente, a tabela das coordenadas. Deste modo, para apagar um mosaico, é necessário apagar primeiro as entradas da tabela das coordenadas que se referem ao mosaico em questão a fim de não ser apagada uma entrada de uma tabela que está referida noutra. Por este motivo são feitos dois acessos à base de dados para concluir a operação desejada, primeiro são apagadas todas as entradas da

tabela das coordenadas que contenham no campo "ID\_Mosaico" um valor igual ao indicado pelo utilizador, depois é apagada a entrada da tabela dos mosaicos que contém no campo "ID\_Mosaico" o valor indicado pelo utilizador. O código que desencadeia estas ações é apresentado no *Script* B.5 do Apêndice B.

Para que o utilizador fique a par das entradas que existem nas tabelas foi criada a funcionalidade listar, que é semelhante para os quatro campos da gestão de produto. A janela de listagem possui apenas um botão que desencadeia a execução de um *script* que fará com que seja exibida na janela de relatórios uma listagem das entradas da tabela que o utilizador pretende consultar, esta listagem contém o ID e o nome de cada entrada e é apresentada por ordem crescente do ID.

Para o desenvolvimento deste *script* foram utilizadas ferramentas diferentes das utilizadas nos *scripts* das funcionalidades anteriores, no entanto a lógica de ligação à base de dados e a forma de reportar erros é semelhante. Assim o código difere no *SQL statement* e nos comandos executados na base de dados. Como apenas se pretende ler valores armazenados na base de dados o *SQL statement* é do tipo "SELECT FROM" e tem o seguinte formato:

```
"SELECT column1, column2, ... FROM table_name;"
```

O *script*, após as declarações iniciais, abre a ligação à base de dados seguindo-se a execução do *SQL statement*. Com esta execução ficam selecionadas todas as linhas e todas as colunas da base de dados, resultado da utilização do símbolo "\*" ", que significa "all", em vez de se especificar o nome de cada coluna. Assim, com as linhas selecionadas, é utilizado um objeto do tipo "ADODB.recordset". Um "Recordset" mantém os valores de uma linha da tabela e para trabalhar com este é utilizada a propriedade "EOF" que retorna o valor 1 caso se tenha chegado ao fim das linhas selecionadas e é também utilizado o procedimento "MoveNext" que faz o cursor avançar uma linha, isto é, os campos do "Recordset" tomam os valores da linha seguinte [74]. Deste modo, o *Script* B.6 do Apêndice B apresenta um resumo do código que torna possível o funcionamento desta funcionalidade.

Com o ciclo *while* é feita, sequencialmente, a impressão dos campos do ID e do nome (dos artigos, neste caso) até que seja atingido o fim da tabela, ou seja, até que a propriedade "EOF" retorne 1 (ou *True*). É ainda importante salientar a utilização do procedimento "MoveNext" no fim do ciclo *while*, que faz o cursor avançar para a linha seguinte.

Para terminar a análise do trabalho desenvolvido falta analisar a criação de novos mosaicos. Para o utilizador proceder à criação de um novo mosaico basta aceder ao criador de mosaicos (Figura 4.30) através do submenu de gestão de mosaicos do menu de gestão de produto. Ao aceder ao criador de mosaicos é apresentado ao utilizador o ecrã onde são introduzidas as informações base do mosaico. Ao carregar este ecrã é executado um *script* que coloca todas as

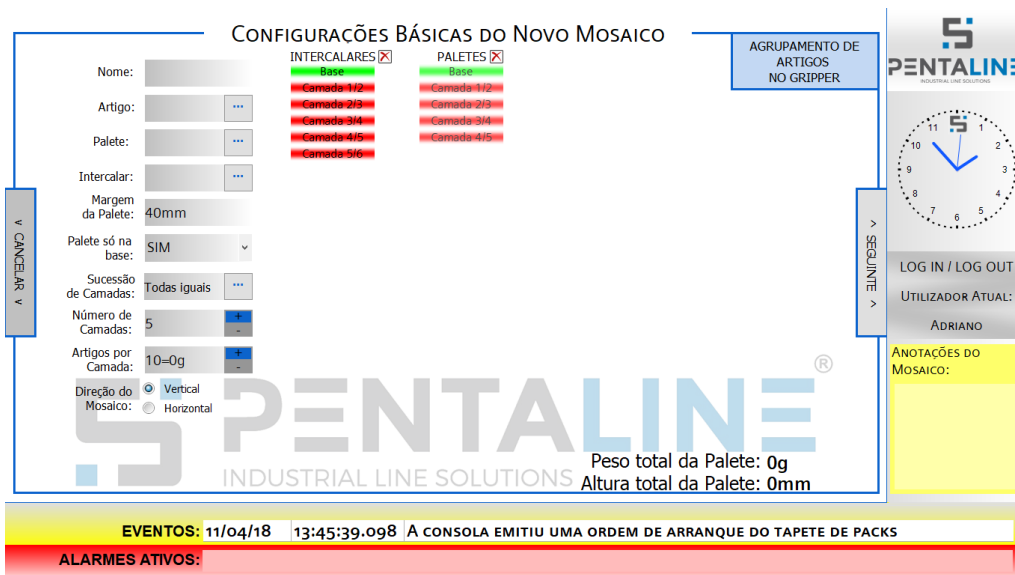


Figura 4.30: Ecrã inicial do criador de mosaicos.

definições nos valores *default* para certificar que tudo é apresentado inicialmente da mesma maneira ao utilizador. O primeiro campo de introdução é destinado ao nome do novo mosaico, este campo possui a função de verificação de duplicados na base de dados, tal como na criação de artigos, paletes e intercalares.

De seguida é escolhido com qual artigo se pretende formar o mosaico. Para isto existe um pequeno botão *"browse"* com o símbolo *"..."* que dá acesso a uma janela de escolha de artigo (ver Figura 4.31). Nesta janela existe uma *"listbox"* para a qual são carregados os nomes dos artigos na execução de um *script* aquando da abertura desta janela.

O *script* segue a lógica normal de uma ligação à base de dados e, depois disso,

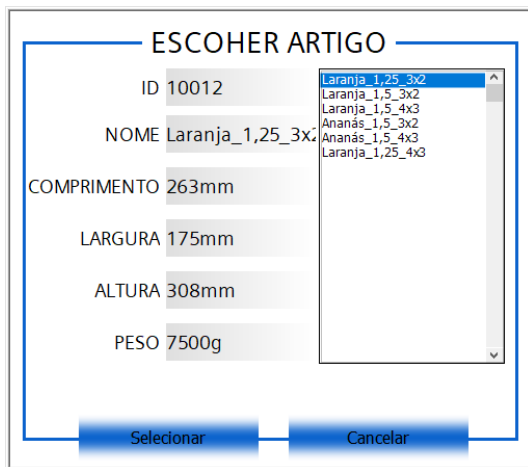


Figura 4.31: Ecrã apresentado para escolha do artigo.

procede à leitura da tabela pretendida e recolha dos campos com importância para este efeito, como pode ser visto no *Script* B.7 do Apêndice B.

Como se pode observar, os ID são armazenados num vetor. Este armazenamento vem facilitar a seleção dos itens da *"listbox"* pois existe uma propriedade da mesma que contém o valor da linha selecionada, posto isto é criado um *script* que é executado sempre que há um clique na *"listbox"*.

Tendo conhecimento do número da linha que foi selecionada, o *script* consulta o vetor `"array_ID[ ]"` e recolhe o valor do ID da linha selecionada. Sabendo este ID, é feito um acesso à tabela dos artigos em busca da entrada com o ID indicado e são recolhidas as suas características para as apresentar nos campos destinados a este efeito na janela de escolha do artigo. O *script* que é executado com o clique na *"listbox"* segue a lógica apresentada no resumo apresentado no *Script* B.8 do Apêndice B.

A escolha de paletes e de intercalares funciona do mesmo modo que a a escolha de artigo portanto, segue-se a "Margem da Pallet". Neste campo o utilizador indica a distância máxima que os artigos podem exceder a pallet. Depois deste surge o campo "Pallet só na base" ao qual se pode responde afirmativamente ou negativamente. Caso a resposta seja positiva a zona de definição de distribuição de paletes fica bloqueada apenas com pallet na base, caso seja negativa esta zona fica disponível para personalização do utilizador. Ao lado desta zona de distribuição de paletes encontra-se a distribuição de intercalares. Estas zonas de distribuição de paletes e intercalares (ver Figura 4.32) funcionam com recurso a diversos botões, cada um tem associado um bit da variável de distribuição que lhe corresponde e ao clicar sobre um botão o bit que lhe corresponde troca de estado. O estado do bit que corresponde a um botão reflete-se na sua cor, isto é, o botão assume a cor verde caso o seu bit esteja a 1 e assume a cor vermelha caso o seu bit esteja a 0. Cada botão representa o intervalo entre duas camadas, exceto o último botão da distribuição de intercalares, este representa o topo da pallet. É importante referir também que o número de botões varia de acordo com o número de camadas indicadas no campo "Número de camadas".

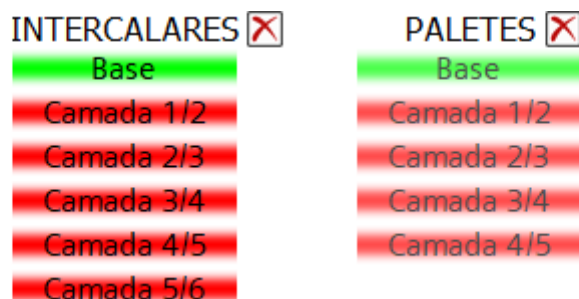


Figura 4.32: Botões apresentados para indicar distribuição de intercalares e paletes.

Existe também um campo onde é indicado quantos artigos por camada se pretendem colocar, ao inserir um número é automaticamente calculado o peso da camada, o que é muito útil se for conhecida a capacidade de carga do robô manipulador de camadas. Em situações singulares é mais indicada a formação do mosaico horizontalmente pelo que é dada opção ao utilizador de escolher a direção de formação horizontal ou vertical. Por fim existe o campo de "Sucessão de Camadas" onde é definido se todas as camadas são iguais, se há uma sucessão par/ímpar (espelhado na vertical, na horizontal ou rodado 180°) ou se são todas diferentes, isto é, o utilizador desenha cada camada individualmente. Caso o utilizador escolha personalizar cada camada surgem vários campos para indicar quantos artigos se colocarão em cada camada. Para escolher a sucessão de camadas é apresentada uma janela com algumas indicações gráficas para ajudar o utilizador a perceber que tipo de sucessão está a escolher (ver Figura 4.33).

Existe um *script* que calcula o peso total e altura da palete. Este vai sendo executado a cada alteração que vai sendo feita de modo a manter os dados apresentados sempre atualizados. Por fim, no canto superior direito existe um botão que dá acesso a um ecrã que semelhante ao "Agrupamento de Artigos no *Gripper*", na criação de artigos, de modo a que o utilizador perceba o enquadramento dos grupos de artigos no *gripper* do robô formador de mosaico, e também é apresentada uma indicação do máximo de artigos que podem ser agrupados no *gripper*.

Os *scripts* mais pertinentes no funcionamento deste ecrã de "Configurações base do novo mosaico" são apresentados no Apêndice B de modo a permitir uma análise mais aprofundada.

Ao carregar no botão avançar o utilizador passa para o ecrã "*Layout do Novo Mosaico*" (ver Figura 4.34). Neste ecrã é desenhado o mosaico pelo utilizador. À direita é mostrado um resumo das definições introduzidas até agora, no fundo

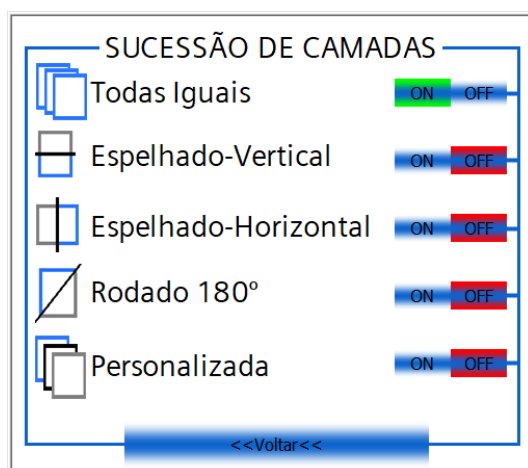


Figura 4.33: Janela apresentada para indicar sucessão de camadas.

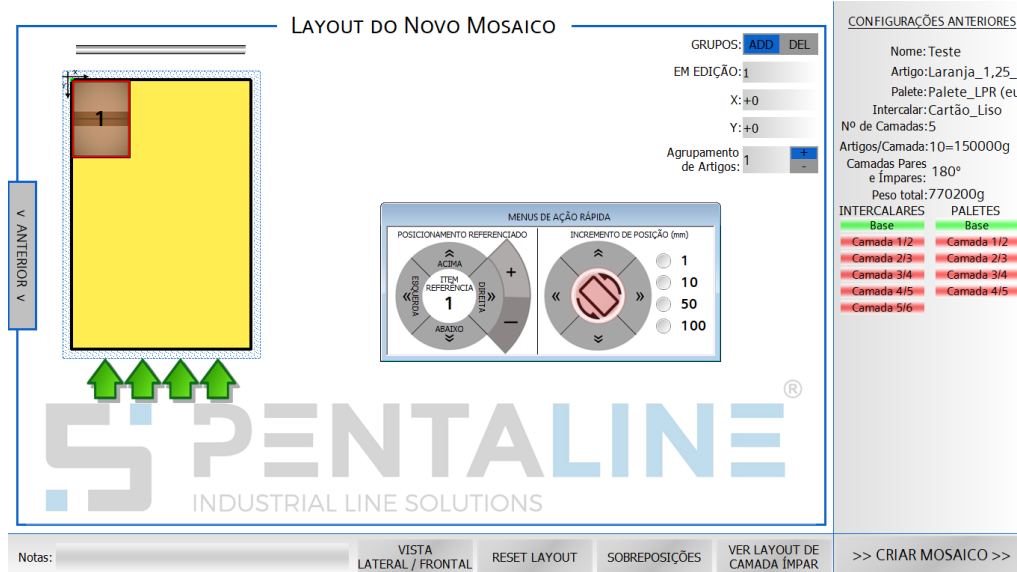


Figura 4.34: Janela apresentada para criar o *layout* do mosaico.

existem as ferramentas vista lateral e frontal, *reset* do *layout* desenhado, verificação de sobreposições de artigos e ver camada ímpar. Na zona de criação do mosaico é apresentada uma base amarela, que representa a palete, e uma sombra azul que representa a margem. Existe junto à palete um conjunto de setas e um objeto cinzento, que representa um batente, de modo a dar uma perceção da direção que o mosaico toma na linha de formação. Sob o canto superior esquerdo da palete pode ser observado um ponto que indica a origem do referencial bem como as setas indicativas do sentido de X e Y. Para criação e manipulação de itens existe um conjunto de campos e botões no canto superior direito que permite a adição de itens, formar grupos e posicionar os mesmos com base nas coordenadas. Existe também um menu de ação rápida, este menu de ação rápida permite uma manipulação dos artigos com referência uns aos outros e permite também um incremento de posição gradual.

Fazendo uma análise aos constituintes e modo de funcionamento deste ecrã, pode começar por dizer-se que, ao carregar este ecrã, são logo posicionados e dimensionados a palete e a sua margem de acordo com a escala escolhida (30%) e são também atribuídos valores aos campos de configurações anteriores. Após análise de várias soluções, os itens a colocar sobre a palete são representados por botões com ajuste dinâmico da sua imagem, isto é, um botão que represente um grupo de dois artigos toma automaticamente a imagem de duas caixas. Deste modo o utilizador tem uma melhor perceção dos *packs* de cada grupo. A vantagem da utilização de botões está na possibilidade do desencadear de várias ações a partir de um clique sobre o mesmo. Neste caso, ao clicar sobre um item (um

botão) a variável que define qual o item que está a ser editado é atualizada para o item no qual se clicou, tornando mais rápida e intuitiva uma reconfiguração do item. A alternativa a clicar no botão que corresponde ao item para o selecionar para edição é a alteração direta no campo "Em Edição" que tem associada esta variável.

Para o posicionamento e orientação dos itens existem dois tipos de coordenadas, as reais (em relação à palete) e as de ecrã. A cada botão de item foram associadas variáveis para o posicionamento no ecrã, estas são do tipo "x\_item\*\_ecr" e "y\_item\*\_ecr" em que o "\*" será o número do item. As variáveis para o posicionamento real são "x\_item\*" e "y\_item\*" as 4 têm uma relação direta. A sua atualização deve ser sempre feita com o cuidado de manter a sincronia. Para posicionar os itens pode recorrer-se aos campos "X" e "Y" para uma configuração mais analítica, trabalhando as coordenadas do canto superior esquerdo dos artigos em relação à palete. Nestes campos são trabalhados valores reais, como tal a sua atualização implica uma atualização dos valores das coordenadas reais e um cálculo das coordenadas de ecrã. Para isto, foi feito um *script* que é executado quando é feita uma alteração no campo, neste exemplo o "X":

```
Dim referencia
Dim act_item
Dim x_coord

'saber que item está em edição
act_item = HMIRuntime.Tags("em_edicao").Read
referencia = "ITEM_" & act_item

x_coord = HMIRuntime.Tags("introduzido_x").Read 'recolher o valor introduzido

referencia = "x_item" & act_item 'copiar valor do campo para a coordenada real
HMIRuntime.Tags(referencia).Write x_coord

x_coord=x_coord*0.3 'calcular a coordenada de ecrã-escala de 30%
x_coord=x_coord+85 'offset para o canto da palete

referencia = "x_item" & act_item & "_ecr" 'copiar p variavel associada ao botão
HMIRuntime.Tags(referencia).Write x_coord

Sobreposicoes() 'execução de outros scripts
margens()
```

De modo a tornar este posicionamento mais intuitivo e também mais rápido foi desenvolvido o "Menu de ação rápida" que tenta abstrair o utilizador das coordenadas. Este pequeno menu é constituído por duas vertentes, o posicionamento referenciado e o incremento de posição em milímetros. O posicionamento referenciado toma como referência um item já inserido e posiciona o item em edição numa posição adjacente a esse item. Utilizando os botões é dada a hipótese de posicionamento acima, abaixo, à direita e à esquerda. Para este posicionamento é executado um *script* que lê as coordenadas do item referência e calcula a coordenada do canto inferior direito. Com estes dois pontos é possível posicionar o novo artigo numa zona adjacente sem sobreposição. O *script* desenvolvido para cada

tipo de posicionamento é semelhante alterando-se apenas as linhas de cálculo das coordenadas do item em edição para cada caso. Inicialmente é feita a recolha de dados e, com estes dados é feito o cálculo do seu canto inferior direito. Depois, com os dados do canto superior esquerdo e os do canto inferior direito é possível posicionar o artigo em edição, tendo em conta que é necessário contemplar também se o item em edição tem vários *packs* ou se tem rotação.

A vertente de incremento de posição contempla um botão que faz alternar a rotação do item em edição e tem quatro botões que incrementam a posição do item de acordo com o número selecionado nas opções à direita deste menu. Assim sendo estes botões podem incrementar na direção pretendida a posição dos itens em 1, 10, 50 ou 100 milímetros. Os *scripts* executados por estes botões podem ser resumidos a 2 pontos principais leitura da opção pretendida e atualização da coordenada pretendida de acordo com a opção selecionada.

No topo direito existe um botão que apaga e outro que cria artigos. Estes botões atualizam uma variável que indica quantos itens o utilizador criou e, a partir desta variável, os botões que representam os itens tornam-se visíveis ou invisíveis. Ao clicar sobre estes botões são também atualizadas as posições dos itens para que estes, ao se tornarem visíveis, apareçam sempre no canto superior esquerdo da paleta. São também contados quantos *packs* já se encontram sobre a paleta de modo a que o número de artigos por camada indicado anteriormente não seja excedido, caso já tenha sido atingido o limite surge uma mensagem a indicar o sucedido. A criação de itens encontra-se limitada a 24, que é o número de botões criados e que têm a sua visibilidade definida pela variável descrita. É ainda importante referir que a limitação a 24 itens é bastante ampla, tendo em conta que cada item pode ser constituído por vários *packs*. O código desenvolvido para execução no clique deste botão segue a lógica apresentada na Figura 4.35.

Como configuração existe ainda o campo "Agrupamento de artigos" que trabalha sobre a variável do grupo de *packs* do item em edição. A criação de grupos está limitado ao máximo suportado pelo *gripper* e, à semelhança do que acontece na criação de itens, neste botão também são contados os artigos que já existem para que não seja ultrapassado o máximo indicado pelo utilizador.

Na barra inferior existem as "funcionalidades extra" que permitem uma supervisão da conceção do mosaico. A vista lateral e frontal faz com que seja mostrado um ecrã com um resumo visual da paleta, permitindo haver uma melhor perceção do que foi configurado. O botão "reset layout" simplesmente retorna o mosaico que foi desenhado ao seu estado inicial. O botão sobreposições começa a piscar assim que forem detetadas sobreposições e ao clicar sobre ele são mostrados em que parte dos artigos existe sobreposição. Para a deteção de sobreposições foi criado um *script* que é chamado no final de quase todas as ações feitas neste ecrã, à semelhança do *script* que deteta se existe ultrapassagem das margens. O botão

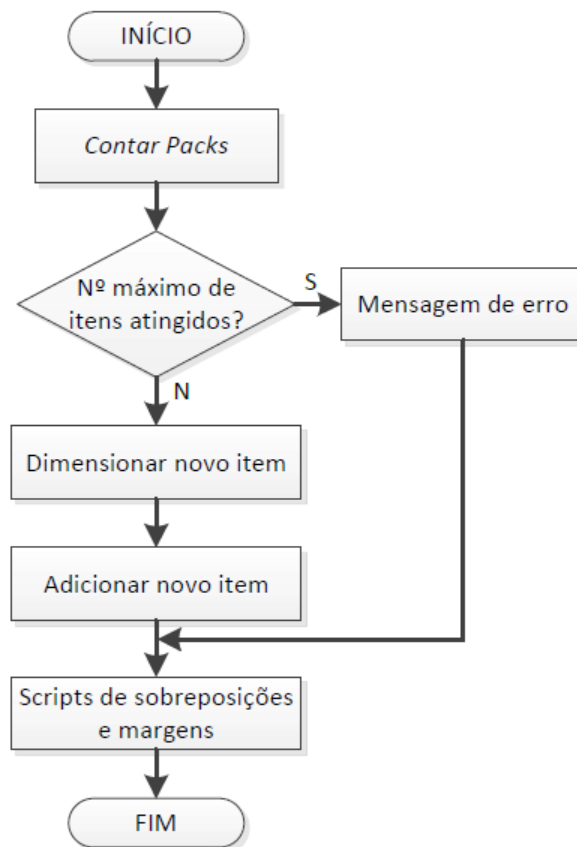


Figura 4.35: Fluxograma que representa o *script* de adicionar item.

à direita deste último serve para o utilizador visualizar o *layout* da camada ímpar. Para que os artigos assumam as novas posições é executado um dos três *scripts* possíveis, de acordo com o tipo de camada ímpar que tenha sido escolhida. Estes três *scripts* de sobreposições, de margens, e de camadas ímpares são apresentados nos *scripts* 3.10, 3.4 e 3.11 do Apêndice B, respetivamente.

Por fim, para terminar a criação do novo mosaico basta clicar no botão "Criar Mosaico". Com o clique neste botão é executado um *script* que verifica se todos os requisitos estão cumpridos e se existem falhas na construção do *layout* do mosaico e então guarda as características do mosaico na base de dados. Para melhor se acompanhar a descrição desta funcionalidade é apresentado o "Fluxograma de criação de mosaicos" no Apêndice B que resume este *script*, que pode ser consultado no *script* 3.12 do Apêndice B. A primeira coisa que é feita é a contagem e validação do número de artigos na paleta, surgindo uma mensagem de erro caso, tenham sido introduzidos menos artigos que o indicado para a camada. De seguida são verificados outros erros, como as sobreposições e a ultrapassagem de margens. Cada um dos *scripts* que faz estas verificações coloca uma variável da HMI a 1, pelo que se alguma delas estiver nesse estado nesta verificação é

apresentada uma mensagem de erro. Caso não existam estes erros então tudo indica que se pode proceder à escrita na base de dados.

Uma das particularidades na escrita das características na base de dados é que, como é preciso saber o ID atribuído pela base de dados ao mosaico na tabela dos mosaicos para se poder escrever esse ID na tabela das coordenadas é necessária uma ordem específica nos procedimentos de escrita. Deste modo, primeiro é feito o acesso à base de dados para escrita das características base do mosaico na tabela dos mosaicos, depois é feito um acesso para consulta do ID que foi atribuído ao mosaico com nome que o utilizador indicou para este. Tendo conhecimento do ID do mosaico estão reunidos também os dados todos para escrita na tabela das coordenadas. A escrita na tabela das coordenadas é feita linha a linha pelo que existe um ciclo "while" que faz um acesso para escrita na base de dados por cada item.

Por fim, no que toca à criação de mosaicos apenas falta referir a criação de mosaicos personalizados camada a camada. Depois de escolhido o tipo de sucessão "Personalizado" são indicados os artigos que cada camada deve levar. Depois disso o ecrã apresentado para criação do *layout* é semelhante ao apresentado para os outros casos sendo adicionado apenas um campo que indica que camada está em edição. No entanto em termos de gestão das informações existem várias diferenças nomeadamente na sequência e nas ocasiões em que são gravadas as informações. Neste caso, as informações base do mosaico são gravadas na base de dados logo que o utilizador avança para o desenho das camadas. Já as informações de cada camada são escritas ao fim da criação de cada uma. Deste modo é evitado o armazenamento na HMI de centenas de variáveis com dados temporários, enquanto são criadas as camadas seguintes o que tornaria o programa extremamente pesado, exigindo grandes capacidades de processamento. Assim, a cada camada que é concluída a mesma é gravada, é atualizada a variável de camada em edição e é feito o *reset* à disposição dos artigos, eliminando também a necessidade de um ecrã por camada. O único inconveniente é que, ao contrário dos outros casos de sucessão de camadas, não é possível editar um dos passos anteriores, isto é, depois de criadas voltar a editar a camada anterior ou as características base do mosaico.

De modo a permitir uma análise mais completa dos *scripts* que permitem o funcionamento deste ecrã de "Layout do novo mosaico", é apresentada no Apêndice B uma compilação dos códigos mais importantes.

Para terminar a análise da HMI apenas falta analisar o ecrã de comando geral, apresentado na Figura 4.36. Neste, encontram-se os comandos básicos de funcionamento automático ou funcionamento em modo manual, arranque e paragem do transportador de entrada de *packs* e também são apresentados comandos e contadores relacionados com o ciclo de paletização. Estes botões atualizam os

valores das variáveis da HMI que serão posteriormente associadas às variáveis do PLC que irão atuar no sistema.

O comando "Interrupção Imediata do Ciclo de Paletização" tem como objetivo que a linha de formação de paletes pare imediatamente, já com o comando "Terminar Produção" pretende-se que a ilha só pare após ser completa a paleta que atualmente está em produção. Com o comando reiniciar ciclo pretende-se que seja recomeçada a ordem de produção, já com o comando reiniciar paleta apenas se pretende reiniciar a paleta que estava a ser produzida. Estes comandos são, normalmente, utilizados em situações de erros no processo pelo que em ambas as situações é necessário retirar manualmente os produtos que tenham ficado na ilha.

Já o comando "Nova Ordem" faz com que surja uma janela semelhante às janelas que surgem para escolha dos artigos na criação do mosaico. Como tal é feito um acesso à base de dados para listagem dos mosaicos existentes e outro acesso para identificar o mosaico em que o utilizador clicar. Depois de selecionar as características do mosaico, dos artigos, da paleta e do intercalar são copiados para variáveis que serão associadas/partilhadas com o PLC e assim o PLC recebe as informações necessárias para construção das paletes. Depois disto, se o utilizador clicar nos campos da janela "Mosaico em Produção" é apresentado um resumo das características do campo selecionado.

Para uma melhor perceção do trabalho desenvolvido no que toca à criação da interface, é apresentado um vídeo no Anexo B onde é demonstrado o funcionamento da interface.

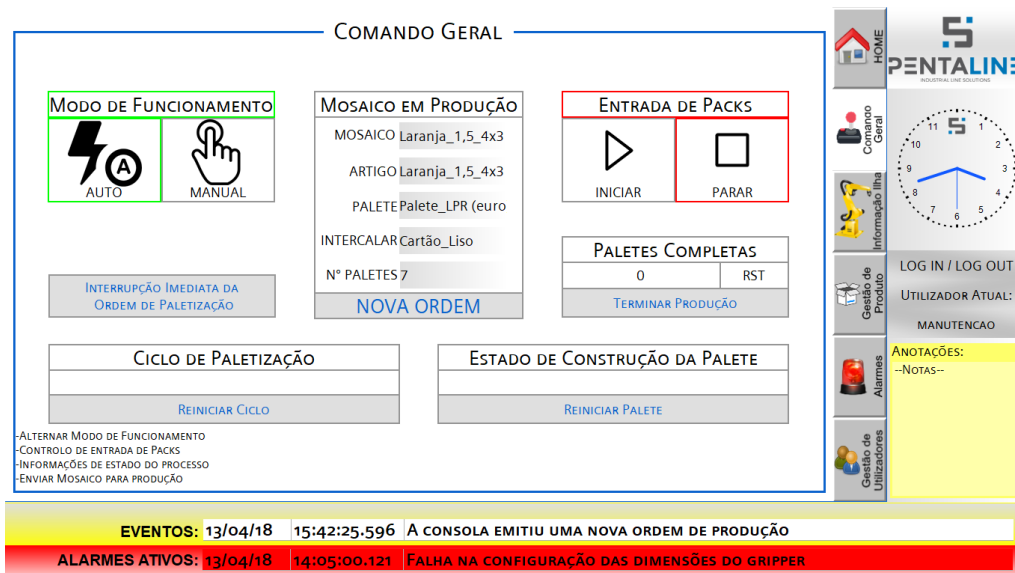


Figura 4.36: Janela apresentada para o menu de comando geral.

## 4.7 Testes

A filosofia de desenvolvimento desta interface permite que esta seja capaz de funcionar em qualquer computador, sendo necessário que este possua uma porta Ethernet para futuramente comunicar com o PLC e um sistema operativo Windows 7 ou superior. Para executar a aplicação desenvolvida basta ter no PC o *software* TIA Portal Runtime Professional. Este *runtime* é apenas um *software* capaz de iniciar a aplicação desenvolvida do WinCC, no entanto o seu licenciamento pode custar alguns milhares de euros.

Para a execução da aplicação num PC com as características descritas é necessário exportar a aplicação desenvolvida para um ficheiro do tipo "*Siemens SCADA RT project file (.mcx)*" e depois fazer uma cópia do mesmo para o PC no qual se pretende implementar a interface.

A capacidade funcional da aplicação nunca está em causa, isto é, não está dependente do tipo e modelo de PC em que a mesma esteja a ser executada, no entanto pode ser feito um reajuste de resolução de acordo com a utilizada por cada máquina de modo a que a interface tenha um melhor aspeto para o utilizador, ocupando o ecrã na sua totalidade. Este ajuste do aspeto e resolução pode ser feito rapidamente no *software* de desenvolvimento.

No que toca à portabilidade da aplicação da interface são estes os aspetos a ter em conta, no entanto existe ainda a base de dados. Para criação da base de dados noutra PC de forma mais rápida foi necessário proceder à cópia da base de dados desenvolvida. Para isto foram feitos procedimentos de criação de um *script* SQL que, ao ser executado em outro PC, cria a base de dados já com as definições de chaves estrangeiras e primárias pretendidas. No entanto com este procedimento feito com recurso ao Microsoft SQL Server Management Studio não são copiados os dados já inseridos nas tabelas. Para a transferência destes dados é usada outra ferramenta do Microsoft SQL Server Management Studio, o "SQL Import and Export Wizard", onde são indicadas as tabelas a exportar. Na exportação e importação é necessária especial atenção. Os campos de ID devem permanecer inalterados já que, por exemplo, o ID atribuído automaticamente a um artigo está guardado na tabela dos mosaicos numa determinada linha que referencia o mesmo. Deste modo, na importação dos dados é necessário desativar a atribuição automática de ID de forma temporária, de modo a não haverem obliterações dos dados destes campos.

O propósito principal do teste da interface será a sua implementação bem sucedida noutra PC, de modo a que sejam realizados todos estes passos descritos.

Na cópia dos dados surgiram alguns contratempos relacionados com as relações entre tabelas. Estes deveram-se ao facto de, ao ser copiada a informação da tabela dos mosaicos primeiro que a dos artigos, das paletes ou dos intercalares, seriam introduzidas referências a campos de ID de outras tabelas que ainda não

existiam. Deste modo a informação teve de ser importada com o especial cuidado de ser importada a informação das tabelas dos artigos, paletes e dos intercalares em primeiro lugar, depois destas é importada a informação dos mosaicos, e por último é importada a informação da tabela das coordenadas.

Feitas todas as configurações necessárias é então necessário testar a ligação entre a aplicação e a base de dados. Para isto basta realizar algumas ordens de escrita e consulta à base de dados. Para isto pode ser criado um artigo para teste. De seguida, para confirmar que os dados foram todos escritos na base de dados existem duas opções. Ou é aberta a base de dados no seu *software* de gestão e consulta-se a tabela correspondente ou, na aplicação desenvolvida, inicia-se a criação de um mosaico e faz-se uma escolha do artigo a incluir no mosaico. Na janela apresentada serão listados os artigos da base de dados bem como as suas características. Clicando sobre o artigo criado podem ser conferidas as suas características, depois basta cancelar a criação do novo mosaico.

Para testar a capacidade do PC pode ainda ser criado um mosaico com muitos itens, de modo a exigir grandes quantidades de cálculos e avaliar a rapidez com que são executadas operações como o "ver camada ímpar" e o "vista frontal/lateral".

Embora o PC no qual foi testada a interface não disponha de um ecrã *touchscreen* esta funcionalidade não estará posta em causa já que os teclados apresentados no ecrã para introdução de dados estão funcionais. Para provar que a utilização da interface num *touchscreen* não está comprometida foram feitos testes apenas utilizando o rato do PC. Deste modo, a utilização bem sucedida de todas as funcionalidades da interface sem recurso a um teclado físico mostra que é possível a sua implementação num ecrã tátil.

Disponibilizando a interface para utilização aos responsáveis pela supervisão do trabalho foi então validada a capacidade funcional da interface, tendo sido identificados alguns pontos de melhoria e novas funcionalidades para trabalho futuro.

## Capítulo 5

---

# Programação e Simulação de uma Ilha de Paletização

---

A programação *offline* da ilha de paletização é o segundo grande objetivo deste trabalho, e deste resultará um modelo tridimensional que representa a ilha de paletização pretendida. Para isto, além da criação dos *layout*, terá de ser feita a programação e configuração dos controladores e manipuladores de forma que, no futuro, seja rapidamente feita a implementação real desta ilha a partir do trabalho apresentado neste capítulo.

### 5.1 Objetivos

Como segunda parte do trabalho a desenvolver no âmbito desta tese de mestrado pretende-se que seja feita a modelação e programação de uma ilha de paletização à camada, de modo a que seja feita a programação *offline* de uma ilha e que possam ser previstos os resultados de variações de parâmetros, e de diferentes hipóteses ou ideias que surjam futuramente.

Um dos objetivos aos quais esta ilha deve corresponder é que esta seja capaz de se adaptar por completo de acordo com o mosaico e com as ordens que o PLC lhe transmite quer esses mosaicos possuam grupos de artigos de tamanho variável e com rotação diferente. É ainda importante sublinhar o objetivo de que o processamento das coordenadas seja feito pelo robô de maneira a que o PLC fique liberto do processamento destes dados, ficando este apenas encarregue de transmitir os mesmos tal como vêm da HMI.

Existe também grande interesse que, a partir da modelação tridimensional da ilha se vede o acesso às zonas em que o robô não deve entrar e que o seu funcionamento se confine apenas às zonas de trabalho.

Outro objetivo passa pela criação de referenciais que definam as zonas de trabalho, isto é, ao reposicionar o robô os pontos sob os quais este deve trabalhar devem permanecer inalterados. Outro objetivo relacionado com este é que a movimentação dos objetos de trabalho sem reposicionamento do robô seja rápida ao nível de programação, situação que será útil para uma rápida reprogramação que resulte do reposicionamento de um elemento da ilha.

Em suma, ao ter um modelo que, com pequenos ajustes, sirva de base a todas as soluções de ilhas de paletização à camada, é facilitado o estudo de novos desafios e novas ideias de solução bem como a validação de alcances, tempos de operações/ciclos e estudo de cadências.

## 5.2 Requisitos

Para ser útil, a criação de um modelo tridimensional de uma ilha deve aproximar-se o mais possível da realidade, não no aspeto, mas na dimensão das formas e no posicionamento da mesmas. Como tal, a partir de peças fornecidas deve ser construído o *layout* da ilha, incluindo as linhas de formação, *grippers*, etc.

Um dos pontos mais importantes é a formação do mosaico. Ao manipulador de formação de mosaico corresponderá um controlador, e a este robô será atribuída a tarefa de posicionar os artigos de acordo com o mosaico que lhe foi transmitido pelo PLC. Este robô deve ser capaz de receber as coordenadas do canto superior esquerdo de cada produto em relação à paleta e, a partir destes dados, posicionar os mesmos no tapete que os leva até aos pontos seguintes do processo. Este posicionamento dos artigos será feito acompanhando o movimento do tapete de formação de mosaico de modo a que sejam atingidas maiores taxas de produção. Este tapete, que é controlado pelo PLC, deve de ser parado sempre que o robô não esteja a ser capaz de acompanhar a velocidade imposta pelo mesmo. Para isto o PLC deve receber um sinal que indica a ordem de paragem ou de arranque por parte do robô.

Para sincronização de tarefas e para controlo do funcionamento da ilha, devem ser criados um conjunto de sinais que indiquem o estado de cada robô, e que permitam que lhes sejam enviadas diferentes ordens de acordo com as necessidades da ilha.

Para *"pick and place"* das camadas de produtos e de intercalares serão utilizados dois manipuladores diferentes, no entanto ambos devem ser trabalhados pelo mesmo controlador já que estas tarefas não serão extremamente exigentes em termos de processamento. A sincronização entre estes manipuladores será feita pelo PLC, no entanto deve ser prevenida a entrada de ambos no mesmo espaço de trabalho de modo e evitar possíveis colisões.

É ainda importante referir que é requisito impedir, por *software*, que os robôs se movimentem para certas zonas do espaço, limitando os seus movimentos a

um espaço, principalmente para serem evitadas colisões inadvertidas com outros equipamentos na ilha.

Pode ser ainda dito que os robôs devem trabalhar sempre com as características atualizadas do *gripper* correspondente, isto é, os dados de peso devem corresponder à realidade, de modo a que não seja causado desgaste prematuro nos motores e engrenagens. No caso do manipulador de camadas esta atualização deve contemplar também o momento de inércia do *gripper*, pois, como este é de grandes dimensões, os dados referidos sofrem grandes alterações dependendo de o *gripper* estar aberto ou fechado.

### 5.3 Descrição da Arquitetura

Visto que são apresentados robôs da ABB, são impostos um conjunto de condições tanto no *software* utilizado como na linguagem de programação que será utilizada. Como tal, o *software* a utilizar será o Robot Studio 6.07, com RobotWare na versão mais recente à data de início do trabalho, a versão 6.06.01. Para programar os robôs será utilizada a linguagem própria da ABB, a linguagem RAPID.

No que toca à organização dos programas a mesma foi definida de acordo com as necessidades da solução. Um programa é composto por instruções e dados, programados na linguagem de programação RAPID, que controlam não só o robô como também equipamentos periféricos, se tal for necessário. É ainda importante sublinhar que, de forma a manter o programa acessível para um acesso rápido, este é armazenado na "memória do programa". Para que se perceba a organização adotada para o programa da ilha desenvolvida é necessário perceber a organização da informação no robô (ver Figura 5.1). No robô, a informação é organizada de forma hierárquica. No primeiro nível existem as *tarefas*, que agrupam *módulos* de programa, que por sua vez agrupam *rotinas* de programa. Ao nível das *rotinas* encontram-se as instruções. Dentro de cada módulo existe ainda um conjunto de dados usados para definir, entre outros, posições, valores numéricos e sistemas de coordenadas. Esse grupo de dados é chamado de *Dados do Programa* [75].

É importante referir um conjunto de particularidades nestes níveis descritos. Por cada *tarefa* apenas pode existir um *módulo* com uma "rotina principal" ou "*main*" (rotina que inicia a execução do programa), todas as outras são consideradas "sub-rotinas", utilizadas principalmente para subdividir o programa em blocos para uma melhor análise. As tarefas são executadas em paralelo, isto é, em simultâneo, pelo que ações no robô em ambas as tarefas iriam gerar conflitos, deste modo apenas são permitidos movimentos no robô numa das tarefas, as outras podem ser utilizadas para gestão de periféricos ou entradas e saídas [75].

Postas estas observações pode ser descrita a organização adotada no desenvolver desta ilha. Por cada manipulador, existirá pelo menos uma tarefa que possui

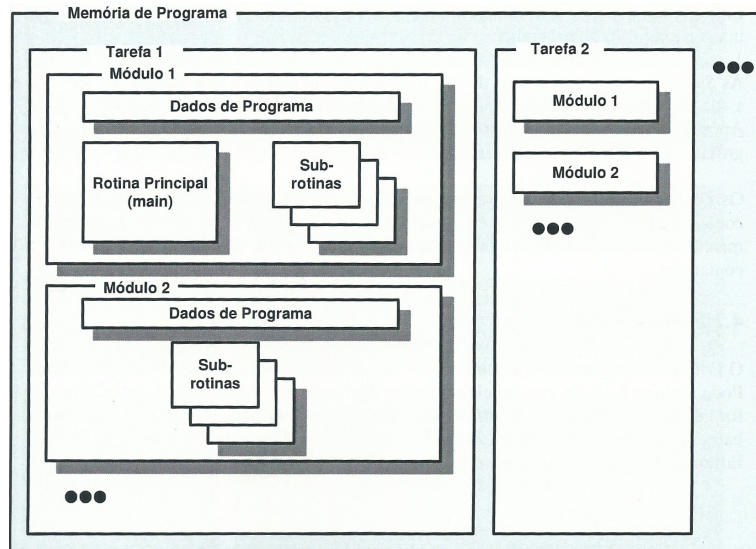


Figura 5.1: Organização da informações na memória do robô [75].

um módulo que ficará encarregue pelo movimento do manipulador e uma tarefa encarregue por gerir os dados que provêm do PLC, tal como ilustrado na Figura 5.2.

De modo a corresponder à necessidade de manipular os produtos num movimento que acompanhe o funcionamento do tapete da mesa de formação de camada é feita a utilização de uma ferramenta de *software* que apenas pode ser habilitada no controlador pelo fabricante. Com a compra deste "add-on" o robô torna-se capaz de realizar operações em referenciais em movimento. Esta opção de *software* chama-se *conveyor tracking* e será utilizada no desenvolvimento deste programa.

O *conveyor tracking* baseia-se em cinco pontos principais, como é ilustrado na Figura 5.3. O ponto A representa um interruptor de sincronização, que na ilha é, normalmente, uma fotocélula que envia um pulso quando passa um produto na

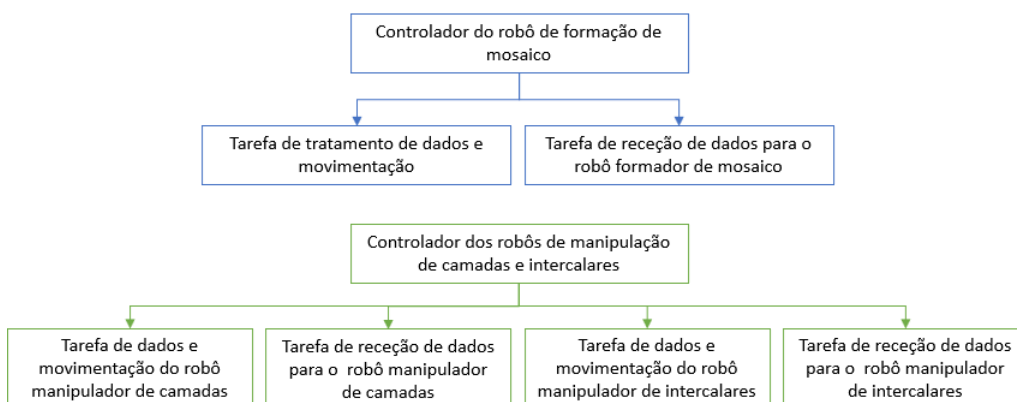


Figura 5.2: Organização das tarefas do programa da ilha de paletização.

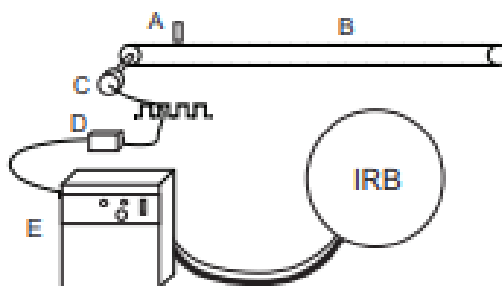


Figura 5.3: Principais constituintes do *conveyor tracking* [76].

sua frente; esta encontra-se sempre no início do tapete. O ponto B representa o tapete com o qual se pretende sincronizar o movimento do robô, a sua velocidade pode variar ao longo do tempo, sendo que o ponto C contribui para que o robô se mantenha a par destas variações [76].

O ponto C representa um *encoder* que, normalmente, está integrado no servomotor mas pode ser acoplado a um dos veios do transportador. Este *encoder* gera um sinal que irá variar de acordo com o modelo e fabricante (deve de gerar entre 1250 e 2000 pulsos digitais com uma tensão fixa entre 10 e 30 V de corrente contínua por metro de tapete movimentado) e transmite-o à interface do controlador, representado no ponto D, ao qual também se liga o interruptor de sincronização. Por fim a interface (que se encontra dentro do armário do controlador) liga-se ao controlador (Ponto E) responsável pelo robô que fará os movimentos sincronizados (representado com IRB) [76].

Com esta opção de *software* torna-se possível seguir mais que um objeto que esteja em cima do tapete, saber quantos objetos estão no tapete à espera de serem manipulados e é possível manter a fiabilidade destas funcionalidades mesmo com variações de velocidade e até paragens do transportador.

## 5.4 Procedimentos de Teste

De modo a verificar o bom funcionamento da ilha modelada e programada no Robot Studio poderia ser feita a programação de uma ilha real de modo a testar o seu funcionamento de uma forma mais palpável, no entanto esta realidade iria envolver uma grande quantidade de recursos pelo que o teste da ilha foca-se mais nos aspetos de lógica e programação que nos aspetos de afinação e ajuste.

Como é sabido, a ilha de paletização é gerida pelo PLC mestre. Este tem várias funções, e uma delas é fazer a ponte entre interface e os robôs, processando algumas das informações para que os robôs trabalhem de acordo com o pretendido (ver Figura 5.4).

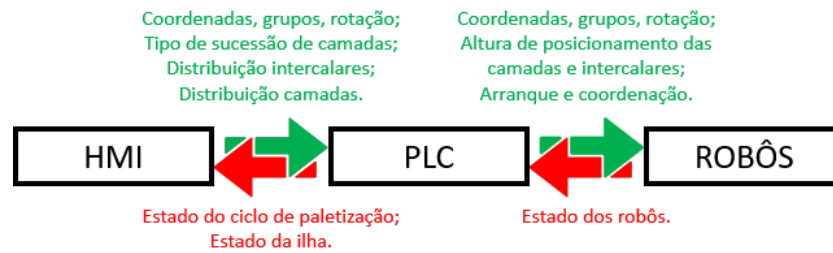


Figura 5.4: Relação do PLC com os principais constituintes da ilha.

Visto que o PLC não se encontra com o *software* completamente desenvolvido, não é possível realizar comunicações com o mesmo, no entanto estas serão simuladas pela atribuição dos dados às variáveis que receberiam os valores comunicados. Os valores em questão são as coordenadas dos produtos e sua rotação, a altura a que é colocada cada camada (e a sua rotação, se necessário) e a altura a que é colocado cada intercalar.

A estes valores que o PLC comunica acrescentam-se os sinais (saídas digitais) que o PLC atua para iniciar, parar e coordenar o funcionamento de cada robô. Para que estes sejam simulados o Robot Studio disponibiliza o "I/O Simulator" que possibilita atuar manualmente nos sinais de entrada da ilha e monitorizar o estado dos sinais de saída.

Utilizando estes métodos e ferramentas é selecionado um dos mosaicos apresentados no caso de estudo apresentado no Capítulo 3 e atribuem-se os valores das coordenadas, alturas e outros, de acordo com esse mosaico, do mesmo modo que o PLC o faria. Já os sinais digitais serão atuados e desatuados manualmente na sequência que o PLC o faria.

Com estes *inputs* a ilha deverá funcionar de acordo com o esperado, isto é, o robô de formação de mosaico deve organizar os produtos de forma correta e, deve adotar trajetórias previsíveis nas movimentações entre produtos. Este é um ponto de especial importância já que, como os produtos estão em movimento, é necessário realizar o cálculo do ponto de aproximação ao item seguinte de acordo com a quantidade de produtos que este agrupa. O facto de as coordenadas transmitidas serem do canto do artigo exige que o robô faça um cálculo das coordenadas do seu centro de modo a que os grupos sejam sempre manipulados de forma correta. Este é mais um aspeto a ter em atenção nos testes.

Nos robôs de intercalares e de camadas é necessário certificar que os robôs estão a cumprir com as indicações de altura de posicionamento. Um dos pontos que é também muito importante é a certificação de que os robôs nunca cruzam o espaço de construção da paleta em simultâneo. Mesmo que o PLC tenha dado ordens para que ambos os robôs entrem em funcionamento, o último a ser ordenado deve esperar que o espaço fique livre.

Por fim, é ainda importante referir que na simulação deve ser verificado se

existem colisões ou passagens próximas entre objetos de modo a que sejam eliminados esses riscos numa posterior passagem para a realidade.

## 5.5 Implementação realizada

No seguimento das ideias apresentadas acerca da importância da modelação da ilha de paletização é começada a implementação pela criação de uma representação tridimensional de uma ilha de paletização à camada.

Para a criação do modelo 3D da ilha foram fornecidas as maquetes dos pedestais, da linha de transportadores de paletes, da linha de formação de mosaico e do armazém de paletes. Para a criação dos atuadores finais foram fornecidas as maquetes de cada elemento individual que compõe o *gripper* de modo a possibilitar a criação de mecanismos.

Na posse destes elementos o passo seguinte é, logicamente a criação dos mecanismos, de modo a construir os atuadores finais que depois serão acoplados aos robôs. Analisando o desenvolvimento do *gripper* de camadas é possível perceber que primeiro foi necessário juntar todos os elementos nas respetivas posições. Na margem esquerda da Figura 5.5 é apresentada a listagem dos sete principais elementos que compõem este *gripper*, entre os quais se encontram os dois garfos, as quatro garras laterais e o corpo do *gripper*. Tendo estas partes em posição o passo seguinte é a criação e configuração de ligações entre estes para se criar um mecanismo.

Visto que este *gripper* é composto por 7 elementos, dos quais um é o corpo do *gripper*, será formado por 6 juntas, todas elas prismáticas. As juntas dos garfos têm ambas 560 milímetros de translação, as garras laterais têm ambas 350 mi-

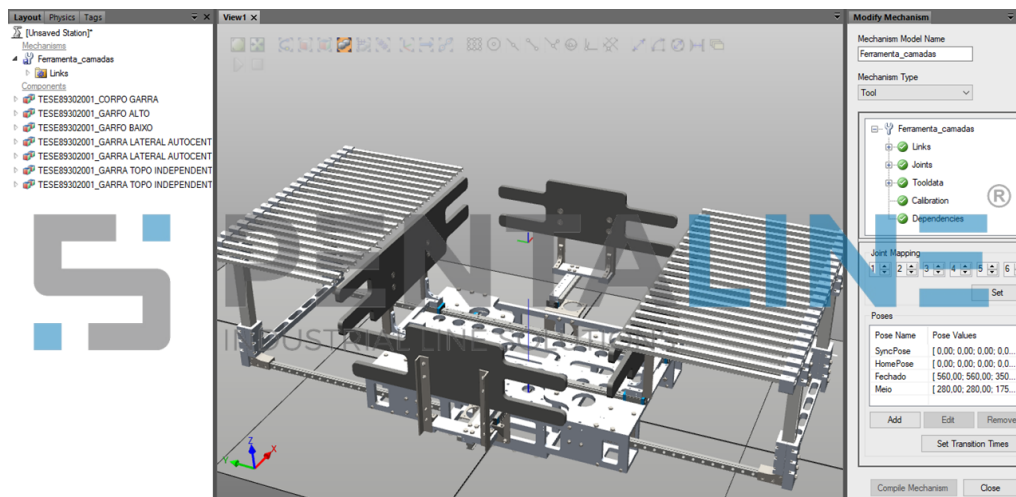


Figura 5.5: Criação do *gripper* do robô manipulador de camadas.

límetros de translação, e as garras dos topos têm 188 milímetros de translação. Definidos os sentidos de translação é terminada a configuração destes parâmetros. Após estas, as configurações que faltam estão relacionadas com os dados físicos da ferramenta, o "tooldata". Nestas configurações basta apenas definir o referencial da ferramenta já que o momento de inércia e a massa serão configurados na programação RAPID realizada a seguir. Por fim, apenas para efeitos de simulação, são ainda definidos *timings* aproximados para abertura e fecho do *gripper*.

Para terminar a criação do *gripper* basta gravar este mecanismo que foi criado como uma biblioteca e deste modo pode ser posteriormente importada no projeto da ilha.

Existem ainda mais dois *grippers* que, embora sejam tecnicamente mais simples, requerem um processo de criação semelhante ao anteriormente descrito (ver Figura 5.6). A única exceção é encontrada no manipulador de intercalares que, como não apresenta partes com movimento apenas necessita da configuração do *tooldata*.

Em seguida à criação dos *grippers* é então iniciada a modelação da ilha, são importadas as partes principais já descritas e dispostas no seu posicionamento típico na ilha. De seguida é importado o robô de manipulação de *packs* que, como possui um controlador individual é importado normalmente pela biblioteca da ABB e posicionado no pedestal correspondente. Este robô é um ABB IRB460 de 5 eixos, com 110 kg de capacidade máxima e com 2,40 m de alcance, capaz de alcançar velocidades elevadas em qualquer um dos eixos.

De seguida, como o manipulador de camadas e o de intercalares partilham um controlador, são apenas importados da biblioteca os braços robóticos. Só depois de importados e posicionados nos respetivos pedestais é que se procede à criação do controlador com recurso ao menu "Robot System" com a opção "From Layout". Deste modo é criado um controlador que será associado a manipuladores

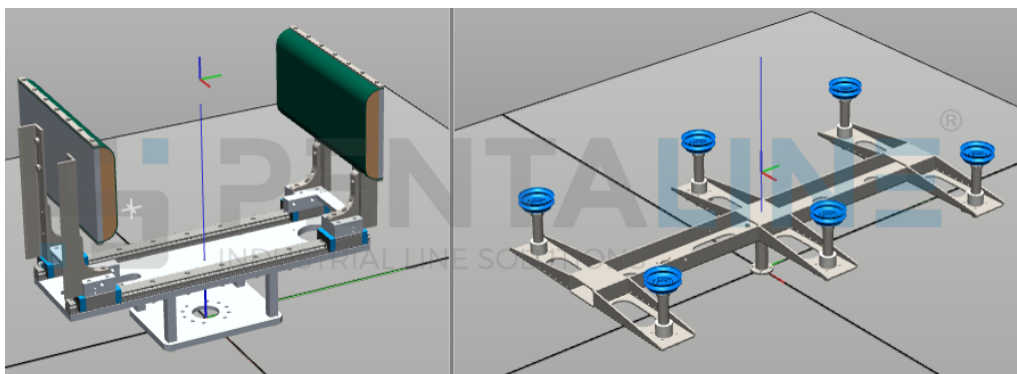


Figura 5.6: *Grippers* criados para manipulação de *packs* (à esquerda) e de intercalares (à direita).

do *layout* que não tenham um atribuído. Deste modo, como um controlador é capaz de suportar até quatro manipuladores, são selecionados os dois manipuladores pretendidos e assim este controlador fica associado a estes dois manipuladores. O manipulador de camadas, como terá de suportar os pesos elevados de uma camada e do *gripper* precisará de ser mais robusto que o anterior, mantendo a característica do número de eixos. Como tal, este é um ABB IRB760 capaz de manipular até 450 kg com um alcance de 3,18 m. Já o manipulador de intercalares terá de possuir seis eixos, exigência que se deve ao facto de este precisar de mergulhar no armazém de intercalares com flexibilidade de movimentos suficiente para contornar as laterais do mesmo, fator que se vai agravando à medida que vão existindo menos intercalares no armazém. Como tal o robô que foi atribuído a esta tarefa foi o ABB IRB4600 com capacidade para manipular até 20 kg com 2,51 m de alcance.

Depois de incluídos no *layout* todos estes elementos, ficam a faltar apenas os *grippers*. Como já foram criadas anteriormente bibliotecas com estes elementos basta importar as mesmas. Depois de ter os elementos importados é necessário acoplar os mesmos ao respetivo robô, tarefa realizada apenas com um clique já que a orientação que a ferramenta toma no punho já foi definida na criação da sua biblioteca.

Com todos os equipamentos no *layout* é feito um estudo do posicionamento que estes possam tomar de acordo com as áreas de trabalho de cada robô. Para isto é ativada a visualização do espaço de trabalho do robô já com o *gripper* acoplado. Com a área de trabalho do robô identificada, a linha de formação de mosaico é reposicionada de modo a que seja encontrada uma posição que favoreça o alcance da maior área possível sobre a mesa de formação (ver Figura 5.7).

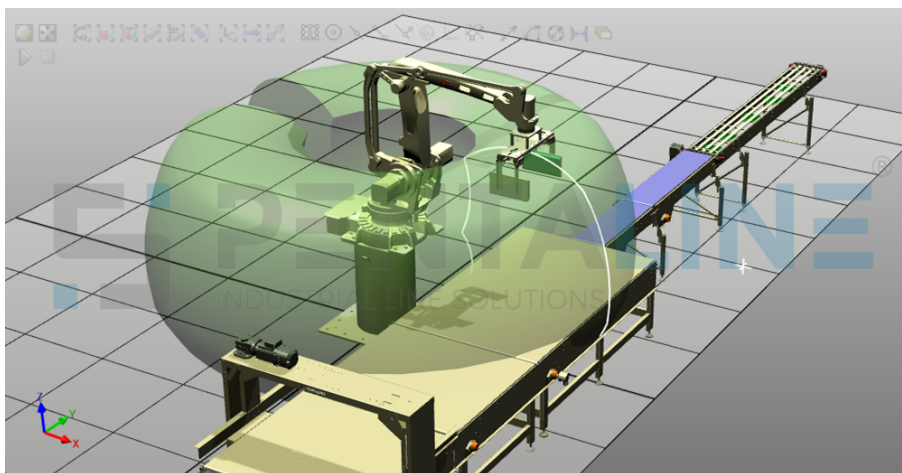


Figura 5.7: Enquadramento da linha de formação de mosaico com o robô de manipulação de *packs*.

A partir do posicionamento da linha de formação de mosaico é definido o posicionamento do robô que manipula as camadas. Posto isto, o robô de manipulação de camadas é posicionado de maneira a que a mesa de *picking* esteja numa posição da sua área de trabalho que não exija que o robô fique nem muito esticado nem muito encolhido ao atingir o centro da mesa de modo a favorecer os movimentos do robô com grandes cargas. Tendo isto em conta, são várias as maneiras de posicionar o robô, pelo que a sua posição será ainda escolhida com base no ponto de formação de paletes. Neste ponto o *gripper* é aberto para largar a camada que está a ser manipulada e, quando aberto, este atinge uma largura considerável. Assim sendo o ponto de *placing* das camadas tem de estar afastado o suficiente para que seja dado espaço de manobra para que o *gripper* não colida com a linha de formação de mosaico. Em suma, o robô manipulador de camadas é posicionado de forma a que haja um equilíbrio entre os pontos de *pick* e *place*, como é ilustrado na Figura 5.8.

Tendo estes equipamentos já nas posições ideais resta posicionar o robô manipulador de intercalares e o armazém de intercalares. O posicionamento deste robô e do armazém é mais flexível que os anteriores equipamentos já que se trabalham cargas pequenas e o robô de seis eixos tem maior flexibilidade. No entanto existem alguns aspetos a ter em consideração, nomeadamente o cuidado que se deve ter para o intercalar não bater no próprio manipulador e o cuidado no posicionamento do armazém de intercalares para que o manipulador não colida com as suas laterais quando existem poucos intercalares, como ilustrado à direita na Figura 5.9.

Com o *layout* completo o passo seguinte é a criação dos pontos que os robôs deverão atingir. Estes pontos, também referidos como *targets* serão agrupados

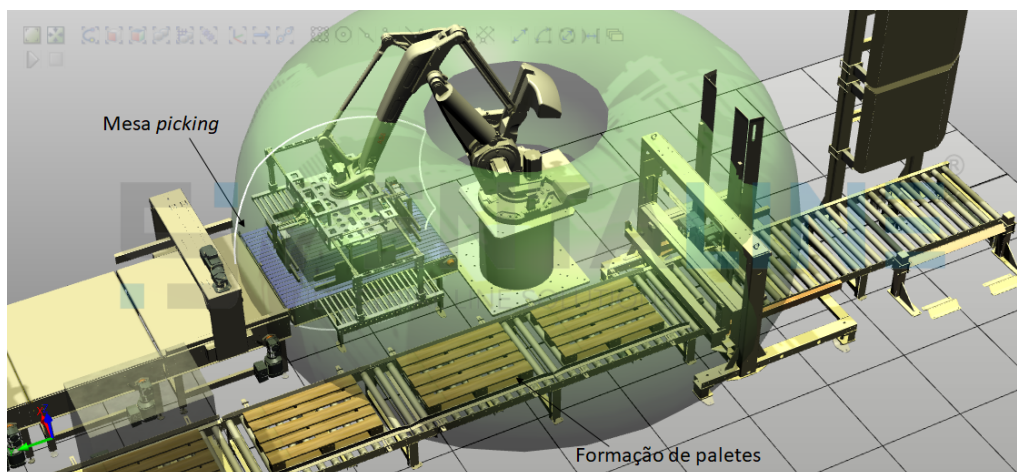


Figura 5.8: Enquadramento da linha de formação de mosaico e do *conveyor* de paletes com o robô de manipulação de camadas.

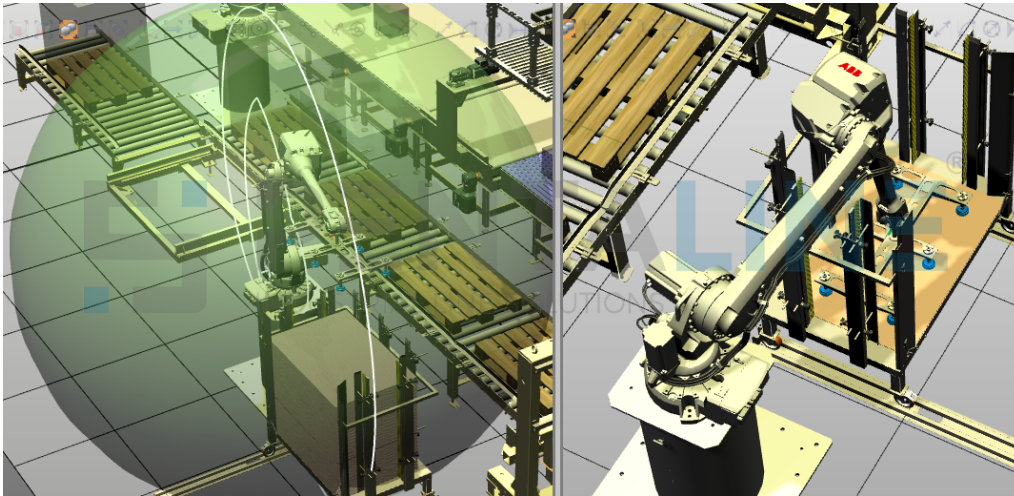


Figura 5.9: Enquadramento do robô de manipulação de intercalares com o *conveyor* de paletes e com o armazém de intercalares.

em *workobjects* que funcionam como referenciais que se podem reposicionar. Ao reposicionar um *workobject* todos os *targets* que estejam associados são arrastados consigo. Deste modo torna-se possível reposicionar um equipamento do *layout* sem ter consequências ao nível da programação. Basta reajustar o *workobject* ao equipamento em questão e os *targets* ficarão realinhados com o mesmo.

Por observação da ilha podem ser identificadas quatro principais áreas de trabalho, pelo que cada uma terá o seu *workobject*. Começando pelo armazém de intercalares, serão criados três *targets*. Um será no topo do armazém, para alinhamento do *gripper* com o centro do armazém, outro ponto será para aproximação ao intercalar até ao qual o robô se moverá a grande velocidade, o terceiro ponto será no intercalar que se pretende manipular, onde o robô irá parar para o apanhar. A altura dos dois últimos pontos será calculada de acordo com a altura da pilha de intercalares no armazém.

A área de trabalho seguinte é a zona de formação de palete, em que a lógica é semelhante à anterior. Existem os três *targets*, um para alinhamento, um para aproximação e um no ponto final. Estes dois últimos também têm a sua altura recalculada de acordo com a altura a que se pretende posicionar a camada ou o intercalar.

Outra área de trabalho é na mesa de *picking*, esta composta por quatro *targets*. O primeiro é o ponto em que o robô se mantém alinhado com a mesa e à espera de uma ordem de *pick and place*. O segundo é o ponto de pré aproximação, no qual o robô verifica se tem a zona partilhada com outro robô livre para, depois de pegar na camada, a utilizar. Caso a zona de construção de palete esteja ocupada o robô aguarda neste ponto de pré aproximação. O terceiro é o ponto de aproximação à camada que será agarrada depois de atingido o quarto ponto.

Por fim resta a área de trabalho de manipulação de *packs* e formação do mosaico. Nesta área existem dois *workobjects*, um que é estático e outro que se movimentará com o sistema de *conveyor tracking*. O estático apenas possui um *target* que é o ponto em que o robô espera por um novo *pack* para manipular. Para se utilizar o sistema de *conveyor tracking* é primeiro criado o *conveyor*. Na sua criação são definidas as dimensões da área onde este sistema funcionará, que é por cima da mesa de formação de mosaico (ver Figura 5.10), e são também definidas outras características como a distância de *reverse tracking* (quando o tapete anda para trás, não utilizada neste sistema) e a distância entre o *trigger* e o início do *tracking* (que, para este sistema, é aproximadamente zero já que a fotocélula estaria no início do tapete). Com estes parâmetros definidos ficam a faltar as definições relacionadas apenas com a simulação, como a velocidade de movimento do tapete, o corpo que irá passar no tapete e a distância entre os corpos que aparecem no tapete. Numa situação real estes três parâmetros não são controlados pelo robô, sendo que a velocidade é o PLC que define, e a distância entre corpos depende do tapete que traz os *packs* até à ilha.

Com a criação do *conveyor* surge um *workobject* no qual serão agora criados os *targets* que se movimentarão à velocidade do tapete assim que haja um *trigger*. Existem cinco *targets* neste objeto de trabalho, no entanto o seu posicionamento irá ser alterado de acordo com o tamanho dos *packs*, o tamanho dos grupos e a rotação pretendida para os mesmos. A passagem por estes pontos segue a lógica

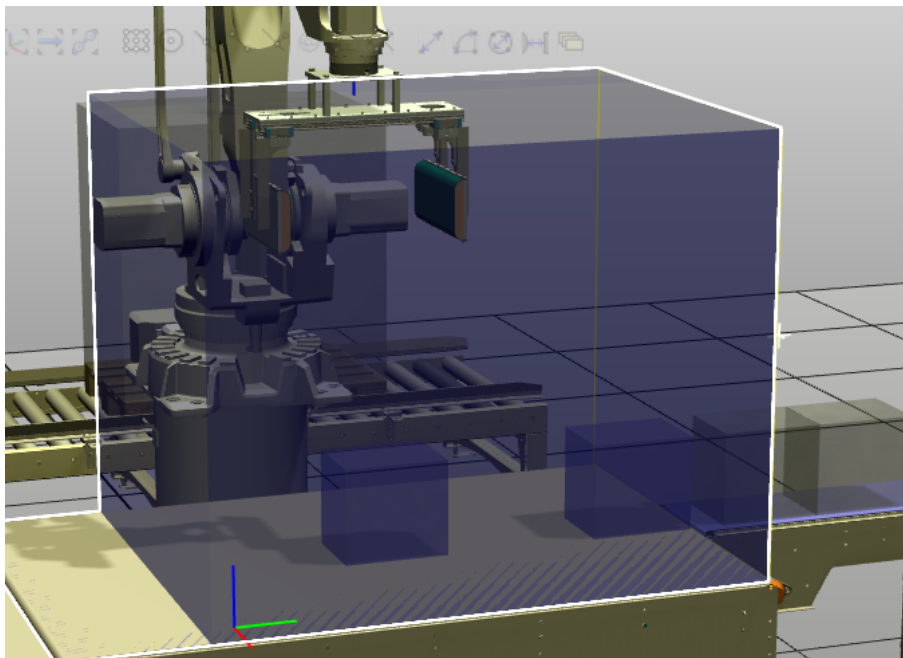


Figura 5.10: Área definida para o *conveyor tracking*.

típica de um *pick and place* comum, primeiro o robô está no ponto estático à espera de um *pack* e assim que este surge no tapete o robô inicia o seu *tracking* enquanto se aproxima do primeiro ponto de aproximação e alinhamento ao centro do produto. Depois de atingido esse ponto o robô vai até ao ponto em que apanha o produto e, depois de agarrar o produto, o robô movimenta-se até ao ponto de aproximação à zona de largar. Depois disto o robô atinge o ponto onde irá largar o produto e, assim que o larga, sobe até ao ponto de saída. Deste modo está concluída a trajetória com *conveyor tracking* e o robô volta a atingir o ponto de espera estático que já tem as suas coordenadas recalculadas para que este se encontre próximo do *pack* seguinte no tapete, caso exista. É importante referir ainda a existência de um *target* de juntas para o qual o robô se move no início do funcionamento da instalação. Ao atingir este ponto certifica-se que o robô tem todas as juntas a 0 graus.

Antes de se iniciar a programação são ainda criados os sinais de entrada e saída e os eventos necessários para que na simulação os robôs movam os objetos que representam as camadas e os intercalares e para os *grippers* terem a animação de abrir e fechar.

O controlador dos robôs de camadas e de intercalares tem 6 entradas digitais e 6 saídas digitais, conforme apresentado na Tabela 5.1, e o controlador do robô de formação de mosaico tem também 6 entradas digitais e 6 saídas digitais, conforme apresentado na Tabela 5.2.

Antes de começar a programação pode ser ainda referido mais um aspeto de configuração, neste caso do controlador. Para que seja possível utilizar comandos relacionados com o *conveyor tracking* ou com *world zones* e até mesmo a utilização de várias tarefas e de vários robôs, é necessário que o controlador venha com essas opções de *software* ativadas pelo fabricante e é também necessário ativar as mesmas nas opções de configuração do controlador. Para isto são ativadas as

Tabela 5.1: Entradas e saídas do controlador dos robôs de camada e intercalares.

Sinal - Função	
Entradas	<b>g_camadas_aberto</b> - Indicação que o gripper de camadas está aberto
	<b>g_camadas_fechado</b> - Indicação que o gripper de camadas está fechado
	<b>g_interc_ativo</b> - Indicação que o vácuo do gripper de interc. está ativo
	<b>g_interc_inativo</b> - Indicação que o vácuo do gripper de interc. está inativo
	<b>iniciar_ciclo_camadas</b> - Ordem do PLC para iniciar colocação de camada
	<b>iniciar_ciclo_interc</b> - Ordem do PLC para iniciar colocação de intercalar
Saídas	<b>em_ciclo_camadas</b> - Indicação de robô de camada na área partilhada
	<b>em_ciclo_interc</b> - Indicação de robô de intercalar na área partilhada
	<b>g_camadas_abrir</b> - Ordem abrir gripper de camadas
	<b>g_camadas_fechar</b> - Ordem fechar gripper de camadas
	<b>g_interc_apanhar</b> - Ordem ativar vácuo do gripper de intercalares
	<b>g_interc_largar</b> - Ordem desativar vácuo do gripper de intercalares

Tabela 5.2: Entradas e saídas do controlador do robô de formação de mosaico.

<b>Sinal - Função</b>	
<b>Entradas</b>	garra_aberta - Indicação que o gripper está aberto
	garra_fechada - Indicação que o gripper está fechado
	Iniciar_producao - Ordem para iniciar a manipular packs
	Parar_producao - Terminar a camada e depois parar de manipular packs
	Pausar_producao - Parar de manipular packs e esperar por ordem retomar
	Retomar_producao - Retomar a manipulação no ponto onde estava
<b>Saídas</b>	Abrir_garra - Ordem abrir gripper
	Fechar_garra - Ordem fechar gripper
	Arrancar_transp - Ordem para arrancar transportador
	Parar_transp - Ordem para parar transportador
	box_count - Pulso que indica que foi virado mais um pack
	em_producao - Sinal que indica que o robô está em produção

opções "Motion Coordination » » Conveyor Control Options » » 606-1 Conveyor Tracking" para o conveyor tracking, "Motion Events » » 608-1 World Zones" para a restrição de zonas, "Engeneering Tools » » 623-1 Multitasking" para utilização de várias tarefas e "Motion Coordination » » Multimove Options » » 604-1 MultiMove Coordinated" para a utilização de vários robôs no mesmo controlador.

Tendo reunidas as condições para iniciar a programação, são então criadas as tarefas na memória de programa do controlador (ver Figura 5.11). Como foi visto na descrição da arquitetura, o controlador dos robôs de camada e de intercalar ("Controlador\_cam\_int") terá quatro tarefas. A tarefa "T\_ROB1" estará encarregue do programa principal, com os movimentos do robô de camadas e a tarefa "T\_ROB1\_leitura" ficará encarregue de ler os valores auxiliares da "T\_ROB1" que viriam do PLC. Na "T\_ROB2" poderá ser encontrado o programa principal com os movimentos do robô de intercalares, já na "T\_ROB2\_leitura" serão lidos os valores auxiliares ao trabalho do robô de intercalares que viriam do PLC.

O controlador do robô de formação de mosaico ("IRB\_460\_110kg\_2.40m\_1") tem duas tarefas, a "T\_ROB1" fica com o programa principal com os comandos de movimentação do robô, a "Tarefa\_2" fica com a leitura dos valores que viriam do PLC e com a conversão das coordenadas, que são recebidas nos cantos e passam a ser nos centros e com um referencial diferente.

Estes *scripts* em RAPID são apresentados no Apêndice C para que possa ser mais facilmente acompanhada a descrição do funcionamento dos mesmos.

Começando pelo robô de camadas, os dados que se pretendem que sejam recebidos do PLC é a altura a que esta deve ser posicionada e se o posicionamento da camada é ou não é rodado 180° em relação à posição em que foi agarrada. Posto isto a "T\_ROB1\_leitura", como não é possível simular uma ligação a um PLC, simula que estes valores foram recebidos e vai alterando estes valores de acordo com as características do mosaico que se pretende simular. Neste caso

pretende-se simular o mosaico já visto no capítulo 4, o "Laranja\_1,25\_4x3" pelo que são colocadas 4 camadas por palete, rodadas a 180° umas em relação às outras.

Como pode ser observado no código desta tarefa de leitura, na execução é esperado que a tarefa principal dê a ordem para ser feita uma nova leitura e depois de esta estar concluída é sinalizado à tarefa principal para que esta prossiga a execução. Pode ser ainda observado que existem variáveis do tipo "Const", do tipo "VAR" e do tipo "PERS". A diferença é que as do tipo "PERS" são partilhadas com outra tarefa, enquanto que as do tipo "Const" e "Var" são internas à tarefa, sendo que as primeiras são inalteráveis ao longo da execução, ao contrário das do tipo "Var" que podem ter o seu valor alterado. Em suma, esta tarefa transmite à principal tarefa três valores: a altura de posicionamento da camada, o peso da camada, e se o robô a deve rodar 180° ou não.

Na tarefa "T\_ROB1" do controlador do robô de camadas são ordenados os movimentos de *pick and place*. Por análise do código pode ser observado que existe um ciclo de "first run" que faz com que o robô se posicione no seu ponto inicial. Este ciclo apenas é executado no arranque do robô, depois disto é esperado que o PLC dê ordem para iniciar um ciclo de posicionamento e, assim que é dada a ordem, é feita a leitura dos valores na tarefa de leitura e são atualizados os *targets*. Depois disto é realizado o processo de manipulação da camada, atualizando os parâmetros do *gripper* sempre que este é aberto ou fechado.

Foram ainda implementadas duas hipóteses de rotação de modo a haver uma perceção de qual a melhor a aplicar em diferentes soluções, provando a grande utilidade que tem este tipo de desenvolvimento com recurso a simulação da ilha. No primeiro método de rotação o *gripper* apenas roda durante os movimentos entre o *picking* e o *placing*, ou seja, num mosaico em que todas as camadas sejam

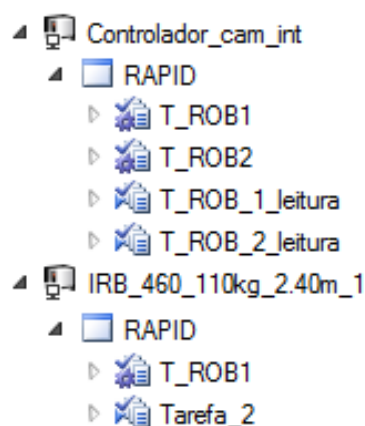


Figura 5.11: Tarefas que cada um dos controladores tem programadas.

rodadas entre si todos os movimentos até ao ponto de alinhamento com o ponto de formação de camadas têm rotação do punho, umas vezes no sentido horário, outras no sentido anti-horário. No segundo método, quando uma camada tem de ser rodada o punho roda 180° no movimento até ao ponto de alinhamento com a formação de paletes, mas depois de ter largado a paleta, quando sobe de novo até ao ponto de alinhamento, roda o punho para a sua posição original.

O robô de intercalares, para realizar um ciclo, terá de receber os dados da altura que tem a pilha de intercalares e terá de receber a altura a que tem de colocar o intercalar. Posto isto, a tarefa de leitura de valores terá de simular a chegada destas informações para o mosaico acima indicado como exemplo. Este *script* da "T\_ROB\_2\_leitura" terá, à semelhança do utilizado para ler os valores para o robô de camada, um mecanismo que lhe indica o pedido de leitura e que indica também que a leitura foi realizada.

Depois desta existe a tarefa principal dos movimentos do robô de intercalar, "T\_ROB2". Nesta tarefa existe também um ciclo que apenas é executado no arranque e que posiciona o robô no topo do armazém. Assim que o PLC dá ordem para colocar um intercalar é feita a leitura na tarefa auxiliar e são calculadas as alturas que os *targets* tomam. Depois disso o robô aproxima-se da pilha e verifica que a área de construção de paleta está livre. Se estiver ocupada, espera que fique livre, se já estiver livre o robô continua a avançar e realiza o ciclo de colocação de intercalar, enquanto sinaliza numa saída digital que se encontra em ciclo e a ocupar a zona de construção de paleta.

O robô que falta analisar é o robô de construção de mosaico, ou de manipulação de *packs* e está associado ao controlador "IRB\_460\_110kg\_2.40m\_1". Este robô necessita de diversas informações como o número de artigos por camada, as dimensões do artigo e o seu peso, a largura da paleta e a margem associada, e por fim as coordenadas dos artigos na paleta. Enquanto que a maioria dos valores não necessita de qualquer tipo de conversão ou processamento, as coordenadas precisam de uma conversão para que o robô possa posicionar os artigos como pretendido.

É importante salientar que este robô apenas precisa dos valores da coordenada "X", da rotação e do grupo. Para trabalhar estes valores a tarefa de leitura, "Tarefa\_2", recorre a três vetores para os valores lidos e três vetores para os valores convertidos. A conversão dos valores traduz-se em, a partir das coordenadas em "X" em relação à paleta, obter coordenadas em "X" em relação ao referencial da mesa de formação de mosaico, como mostrado na Figura 5.12.

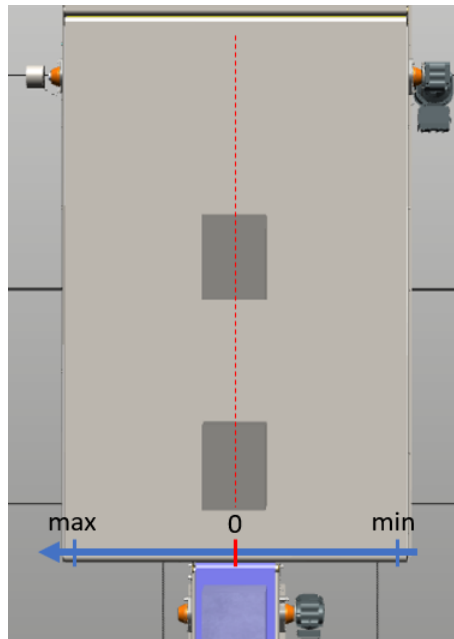


Figura 5.12: Referencial definido para posicionamento na mesa de formação de mosaico.

Para ser feita a conversão é usada a largura e o comprimento do artigo, bem como o grupo de *packs* do item em questão e a sua rotação. Caso o artigo não esteja rodado, será necessária a largura do item (que é igual à largura do *pack*) para a conversão, caso esteja rodado será necessário o comprimento do item (que é o comprimento do *pack* multiplicado pelo número de *packs* agrupados no item). A secção do código que faz esta conversão é a seguinte:

```
!São recebidos as coordenadas dos cantos de cada grupo na paleta
!aqui é feita a conversão para coordenadas em x para o centro de cada grupo em relação à paleta
!tendo em conta a sua rotação e número de packs, é tomado como eixo o centro da paleta/da mesa,
!assim os artigos com x<0 ficam mais próximos do robô e os x>0 ficam para lá do centro da mesa
FOR i FROM 1 TO num_artigos DO
  IF array_r_rec{i}=0 THEN
    array_x{i}:=-((array_x_rec{i}+(largura_artigo/2)-(largura_paleta/2))*escala);
  ELSE
    array_x{i}:=-((array_x_rec{i}+((comprimento_artigo*array_g_rec{i})/2)-(largura_paleta/2))*escala);
  ENDIF
  array_r{i}:=array_r_rec{i};
  array_g{i}:=array_g_rec{i};
ENDFOR
```

Nesta secção de código existe ainda o fator escala que permite ajustar o espaçamento entre os artigos na mesa de formação de mosaico de acordo com as dimensões da paleta. Este é obtido da seguinte maneira:

```
!cálculo automático da escala (espaçamento) com que é formado o mosaico
!no robô utilizando toda a largura da mesa
escala := largura_mesa/(largura_paleta+(margem_paleta*2));
```

Com os dados processados, estes ficam disponíveis para utilização pela tarefa principal que, com base nestes valores irá movimentar o robô de modo a pegar e posicionar os *packs* nas posições pretendidas para cada um. No arranque do robô, é executado na tarefa principal ("T\_ROB1") um movimento do robô que o coloca com todas as juntas a 0°. Depois disso é ordenado que o robô vá para a posição de espera, ajustando essa posição 300 mm para a frente de modo a ter já algum avanço para apanhar o produto que entrar no tapete. Assim que é dada uma ordem de início de produção são atualizados os *targets* com os valores de "X" convertidos e são também rodados, caso o item tenha essa característica. Para rodar os itens existem quatro hipóteses: rodar sempre no sentido horário, rodar sempre no sentido anti-horário, rodar no sentido horário quando o item tem "X>0" e no sentido anti-horário quando "X<0" e, por fim, rodar no sentido anti-horário quando o item tem "X>0" e no sentido horário quando "X<0". Estas são quatro hipóteses que podem ser facilmente testadas e que, apesar de não representarem ganhos em termos de cadência, representam ganhos em termos de estabilidade de acordo com as características físicas dos itens. Para utilizar cada uma destas opções basta alterar os sinais "<" e ">" das condições "IF" desta parte do código.

De modo a gerir o funcionamento do tapete recorre-se a uma variável que indica quantos itens por manipular estão no tapete, se excederem um número que implique que o tapete de formação esteja cheio, então é dada ordem de paragem ao PLC, assim que o robô volte a acompanhar é dada ordem para retomar o funcionamento do tapete.

Foi ainda desenvolvida uma otimização que faz com que o robô passe à frente itens que não precisam de ser rodados nem posicionados, o que acontece em itens que tenham uma posição alinhada com o meio da paleta.

É ainda importante salientar que por vezes é necessário ignorar as configurações atribuídas pelos *targets* ao robô já que estes tiveram as suas coordenadas recalculadas bem como a sua orientação. No entanto esta configuração automática não representa um perigo de falha já que, como o robô foi colocado com todas as juntas a 0° no arranque e todos os movimentos seguintes seguem as configurações, o ângulo de rotação do punho está sob controlo.

Por fim, caso existam produtos já no tapete, é calculado o ponto de aproximação a esse produto de modo a que o robô não tenha de ir a um "Home Position" para iniciar novo ciclo de virar um produto entre dois *packs*.

## 5.6 Testes

De modo a testar que a simulação se encontra em boas condições de funcionamento foi escolhido um mosaico dos que foram introduzidos na interface desenvolvida e, com base nos valores da base dados, foram introduzidos, nos *scripts* de leitura, as características desse mosaico. Tendo os valores necessários (coordenadas, pesos, dimensões, etc) no Robot Studio apenas resta simular as entradas digitais que iriam ser controladas pelo PLC.

Para simular a troca de estado dessas variáveis o *software* possui uma janela "I/O Simulator" e, através da mesma, foram geridas as entradas e as saídas na ordem lógica que o PLC seguiria.

Foram ainda testadas as várias soluções desenvolvidas para o robô de formação de mosaicos e para o robô de camadas que, com recurso à ferramenta de gravação da simulação disponível no Robot Studio, foram gravadas em vídeo para uma fácil análise do resultado. Todos os vídeos estão anexados a este relatório.

Foi ainda feita uma gravação do funcionamento geral da ilha para o mosaico "Laranja\_1.25\_4x3" através da qual se pode perceber que, para este mosaico, o robô formador de mosaico é capaz de formar uma camada (10 itens) em cerca de 15 segundos, o que significa que num minuto é capaz de processar cerca de 40 *packs* que são 4 camadas completas.

Por observação do caso de estudo é possível observar que para este mosaico era exigida uma cadência de 34 *packs*/min, pelo que se pode concluir que a ilha simulada obteve bons resultados no que toca às temporizações obtidas.

Foram ainda levados a cabo testes com outro dos mosaicos apresentados no caso de estudo, o mosaico de "Laranja 1.5L 3x2". Este mosaico de 19 *packs* foi introduzido na interface e os valores necessários ao Robot Studio foram introduzidos no *script* de leitura. O mosaico foi organizado em 9 itens consoante o apresentado na Figura 5.13 e a simulação foi gravada num ficheiro de vídeo anexo a este relatório. Na análise pode ser observada uma cadência de cerca de 64 *packs*/min. Por observação do caso de estudo pode ser retido que o pretendido para este mosaico seriam pelo menos 67 *packs*/min. Mesmo estudando outra forma de fazer este mosaico não foi possível obter esta cadência nesta ilha. Este facto deve-se apenas ao posicionamento do robô que na solução apresentada no caso de estudo se encontra numa posição mais favorável a este mosaico, no entanto não é possível colocar este robô da ABB numa posição semelhante.

Para resolver esta situação recorre-se normalmente a um virador que roda alguns *packs* antes de estes chegarem à ilha, deste modo é possível ultrapassar a cadência pretendida já que o robô não precisaria de rodar grande parte dos *packs*. Como neste momento uma camada deste mosaico ronda os 18 segundos, para se

atingir uma cadência de 80 *packs*/min apenas seriam precisos poupar uma média de 0,19 segundos por *pack*.

O facto de ter sido possível adaptar o funcionamento da ilha de paletização a três mosaicos completamente diferentes comprova que o seu funcionamento está conforme o pretendido, apenas precisando de otimizações consoante certas características de cada projeto em específico.

Os testes realizados para os mosaicos apresentados na Figura 5.13 foram gravados em vídeo para permitir uma observação mais prática dos mesmos. Estes vídeos são apresentados no Apêndice C.

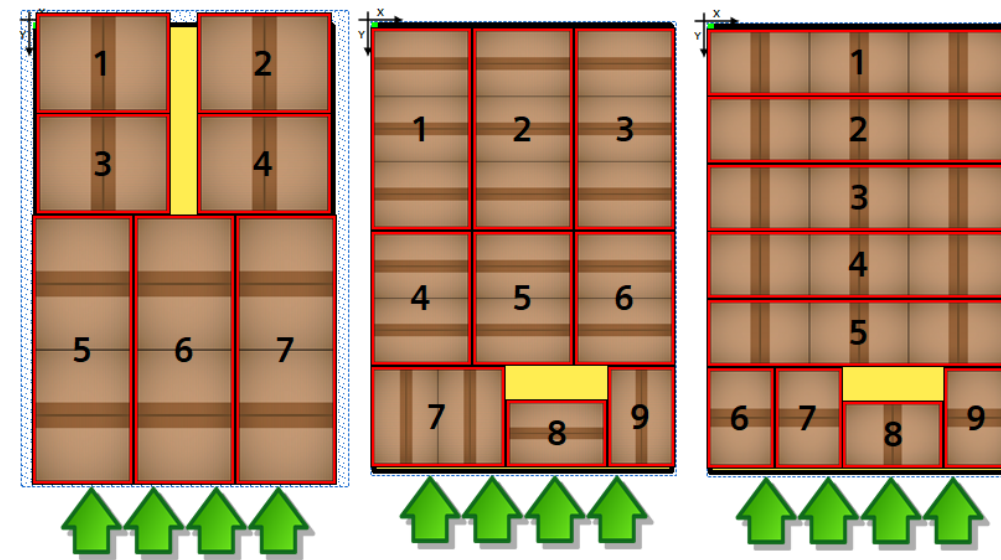


Figura 5.13: Mosaicos testados na ilha de paletização (Laranja 1.25 4x3 à esquerda, Laranja 1.5 3x2 (método 1) ao meio e Laranja 1.5 3x2 (método 2) à direita).

## Capítulo 6

---

# Conclusão

---

Ao longo deste trabalho foram abordados dois aspetos distintos, embora relacionados com o mesmo projeto. No que toca ao desenvolvimento de uma interface de elevado desempenho pode ser observado que foi desenvolvido um trabalho complexo e extenso, resultando numa interface que pode ser considerada simpática para o utilizador e, acima de tudo, funcional. Sendo que um dos principais objetivos para o desenvolvimento da interface se relacionava com estes aspetos, pode considerar-se este objetivo como atingido.

Com a utilização de uma base de dados baseada num servidor SQL pode ser afirmado que o espaço de armazenamento, como apenas depende do disco rígido do PC, é praticamente ilimitado e, deste modo, é atingido também o objetivo de possuir uma interface com grande capacidade de armazenamento.

Ao utilizar o WinCC Professional como *software* de desenvolvimento foi possível criar rotinas em Visual Basic que agilizam o processamento e a comunicação com a base de dados, possibilitando a ligação da interface a qualquer modelo de PLC com algumas configurações básicas, sendo que se o fabricante do mesmo for a Siemens estas configurações apenas se resumem às configurações de rede e ativação da partilha de variáveis com o dispositivo em questão. Assim, a interface torna-se capaz de se interligar a sistemas diferentes, sendo possível adotar a mesma em diferentes projetos bem como ter um funcionamento *standalone*, isto é, isolado da ilha de modo a possibilitar, por exemplo, a criação de modelos de paletes ou mosaicos mesmo quando a ilha está desligada. Deste modo são atingidos os objetivos que restavam em relação à interface.

Futuramente a interface desenvolvida será integrada numa ilha de paletização. Será utilizado um PC industrial equipado com os *softwares* necessários apenas dedicado à interface, um conjunto que terá um custo a rondar os 5.000 euros. Por fim, em relação à interface, foi discutido também como trabalho futuro a criação

de um *software* que possibilite a criação de mosaicos, e seus constituintes num PC de escritório e que essas entradas criadas possam ser transferidas para o PC industrial por uma *pendrive* ou por rede.

A segunda parte do trabalho focou-se na programação dos robôs de uma ilha *standard* de paletização à camada. Tendo como base a parceria que a Pentaline tem com a ABB, foram indicados quais os robôs desse fabricante a utilizar para cada função e, a partir daí foi desenvolvido o trabalho. Na programação foram sendo utilizadas as tecnologias sugeridas pelo fabricante para este tipo de soluções e deste modo foi possível obter um funcionamento mais fluído (utilização do *conveyor tracking*) e mais seguro (utilização das *World Zones*). Para ser atingido o objetivo de se ter uma simulação que se aproxime da realidade foram criados os mecanismos dos *grippers* de modo a que também fosse possível perceber a variação do estado dos mesmos. Ainda no que respeita à simulação, é possível também simular a entrada dos *packs* na mesa de formação de mosaico. No entanto o *software* não permite a simulação da manipulação dos mesmos por parte do robô pelo que fica visível apenas a movimentação e orientação do robô conforme o pretendido, todavia o efeito apresentado na simulação é suficiente para perceber se este processo se está a realizar conforme o desejado, pelo que o objetivo de se ter uma modelação tridimensional que reflita o funcionamento real da ilha é considerado atingido.

Na programação dos robôs foram definidos os sinais que os mesmos receberão do PLC para o sincronismo do seu funcionamento e, através do simulador e entradas e saídas foi possível dar aos robôs estas ordens, como por exemplo a ordem de arranque. Na programação do robô formador de mosaico foi possível desenvolver um algoritmo capaz de interpretar as coordenadas de uma camada tal como ela é introduzida na interface. Apesar de, inicialmente, ter sido planeado receber as coordenadas já trabalhadas no PLC para uma mais fácil interpretação do robô, ao longo da programação foi-se verificando que o tratamento destes valores podia ser feito pelo controlador do robô, libertando o PLC desta responsabilidade. Depois de serem programados todos os procedimentos previstos para cada robô foram implementadas alternativas de solução para os robôs de formação de mosaico e de manipulação de camada, tornando possível perceber que o estudo e implementação destas e de outras ideias que surjam futuramente é um processo que pode ser feito mesmo quando não se dispõe de uma ilha de paletização montada e pronta a testar, sendo este um objetivo também atingido.

No futuro está planeada a implementação do trabalho desenvolvido no RobotStudio na produção de várias ilhas de paletização fabricadas na Pentaline exigindo pequenas alterações relacionadas com cada projeto em específico. O trabalho desenvolvido foi testado virtualmente para vários mosaicos, como foi explicado no relatório, e foi validado pelos responsáveis da Pentaline experientes na área da paletização, não sendo possível, apesar de vários esforços, um teste

numa ilha real já que os projetos deste género são planeados com tempos superiores a um ano de antecedência com um custo total dos equipamentos a ultrapassar facilmente os 500 mil euros.

Deste modo pode ser considerado que os objetivos propostos a este trabalho foram atingidos com sucesso, prevendo a continuação do desenvolvimento do trabalho no futuro com vista à implementação das componentes desenvolvidas em conjunto com a programação do autómato mestre.



---

# Bibliografia

---

- [1] RobotWorx. “What is the Difference Between Soft Automation and Hard Automation?” Disponível em: <https://www.robots.com/faq/show/what-is-the-difference-between-soft-automation-and-hard-automation> [cited at p. 2]
- [2] Eletrical Technology. “What is Industrial Automation | Types of Industrial Automation” (2015). Disponível em: <https://www.electricaltechnology.org/2015/09/what-is-industrial-automation.html> [cited at p. 2]
- [3] Blog for PDF Eletric & Supply Company Inc. “Overview Of The Evolution Of Programmable Logic Controllers (PLCs)” Disponível em: <https://www.pdfsupply.com/blog/a-brief-history-of-the-programmable-logic-controller-plc/> [cited at p. 7]
- [4] Innovative Automation. “What is PLC Programming and Automation?” Disponível em: <http://www.innovativeautomation.com/plc-programming-automation/> [cited at p. xiii, 8]
- [5] Automation Library. “History of the PLC” Disponível em: <https://library.automationdirect.com/history-of-the-plc/> [cited at p. 8, 9]
- [6] Michael O’Hare. “Industry 4.0 and what we learnt from the previous 3 industrial revolutions?” (2017) Disponível em: <https://www.linkedin.com/pulse/industry-40-what-we-learnt-from-previous-3-industrial-michael-o-hare/> [cited at p. xiii, 9]
- [7] Mendelsons. “Modicon 184 Programmable Controller Mainframe” (2017) Disponível em: <http://meci.com/electrical/modicon/modicon-184-programmable-controller-mainframe.html> [cited at p. xiii, 9]
- [8] Antonio Garcia Netto. “De Leonardo a Da Vinci Uma Breve História da Robótica” (2017) Disponível em: [http://www.citi.pt/educacao\\_final/trab\\_final\\_inteligencia\\_artificial/historia\\_da\\_robotica.html](http://www.citi.pt/educacao_final/trab_final_inteligencia_artificial/historia_da_robotica.html) [cited at p. 9, 10]

- [9] Centro de Investigação para Tecnologias interativas. “Estudo Sobre Inteligência Artificial - História da Robótica” Disponível em: <https://pt.linkedin.com/pulse/de-leonardo-da-vinci-uma-breve-hist%C3%B3ria-rob%C3%B3tica-antonio-garcia-netto> [cited at p. 9]
- [10] All On Robots. “Leonardo da Vinci’s robots” Disponível em: <http://www.allonrobots.com/leonardo-da-vinci.html> [cited at p. xiii, 10]
- [11] Manuel Silva. “Introdução à Robótica” Disponível em: [https://moodle.isep.ipp.pt/pluginfile.php/119900/mod\\_resource/content/2/ROBIN\\_Cap\\_1\\_Introducao\\_Robotica\\_2017.pdf](https://moodle.isep.ipp.pt/pluginfile.php/119900/mod_resource/content/2/ROBIN_Cap_1_Introducao_Robotica_2017.pdf) [cited at p. 10]
- [12] Rui; Diretório de Artigos. “Surgimento, História e Evolução da Robótica” Disponível em: <http://diretoriodeartigos.net/surgimento-e-evolucao-da-robotica/> [cited at p. 11, 12]
- [13] Bright Hub Engineering. “Articulated Robots – SCARA and PUMA” Disponível em: <http://www.brighthubengineering.com/robotics/30146-articulated-robots-scara-and-puma/> [cited at p. xiii, 11]
- [14] Digi-Key Electronics. “Sensors Make Sense of Robots’ Worlds” Disponível em: <https://www.digikey.co.il/en/articles/techzone/2012/jan/sensors-make-sense-of-robots-worlds> [cited at p. xiii, 12]
- [15] Daniel Loureiro Carreira. “Sistema de Paletização Robotizado no setor de Pesagem/Etiquetagem do queijo.” Disponível em: <http://repositorio.ipv.pt/bitstream/10400.19/1612/1/Disserta%C3%A7%C3%A3o%20Daniel%20Loureiro%20Carreira.pdf> [cited at p. xiii, 13, 14]
- [16] International Federation of Robots. “1.7 million new robots to transform the world’s factories by 2020” Disponível em: <https://ifr.org/ifr-press-releases/news/ifr-forecast-1.7-million-new-robots-to-transform-the-worlds-factories-by-20> [cited at p. xiii, 14]
- [17] RobotWorx. “What are the main parts of an industrial robot?” Disponível em: <https://www.robots.com/faq/show/what-are-the-main-parts-of-an-industrial-robot> [cited at p. 15]
- [18] Yaskawa Electric. “What is “Robot”?” Disponível em: <https://www.yaskawa-global.com/product/robotics/about> [cited at p. xiii, 15]
- [19] Rodrigo Vertulo, Laboratório de Eletrónica. “Robôs de Classe” Disponível em: <http://labdeeletronica.com.br/robos-de-classe/> [cited at p. xiii, 16]
- [20] Siciliano; Bruno, Khatib, Oussama. “Springer Handbook of Robotics” Publicado em Maio de 2008 [cited at p. 17, 18, 21, 22, 23, 24, 25, 26]

- [21] Wikipedia. “Modelo de viga de Euler-Bernoulli” Disponível em: [https://pt.wikipedia.org/wiki/Modelo\\_de\\_viga\\_de\\_Euler-Bernoulli](https://pt.wikipedia.org/wiki/Modelo_de_viga_de_Euler-Bernoulli) [cited at p. xiii, 17]
- [22] Berti, Éderson (Revisão de Prof. Jean Carlos Strutz). “Robotica” Disponível em: <http://www.ebah.pt/content/ABAAAftFkAD/robotica> [cited at p. 18]
- [23] Wikipedia. “Backlash (engineering)” Disponível em: [https://en.wikipedia.org/wiki/Backlash\\_\(engineering\)](https://en.wikipedia.org/wiki/Backlash_(engineering)) [cited at p. xiii, 19]
- [24] Jyothi Engineering College. “Mechatronics 4” Disponível em: <https://www.slideshare.net/jineshkj/mechatronics-4-46301168> [cited at p. xiii, 20]
- [25] Motion Controls Robotics. “Unraveling Degrees of Freedom and Robot Axis” Disponível em: [www.motioncontrolsrobotics.com/unraveling-degrees-of-freedom-and-robot-axis-what-does-it-mean-to-have-a-multiple-axis-pick-and-place-or-multiple-axis-robot/](http://www.motioncontrolsrobotics.com/unraveling-degrees-of-freedom-and-robot-axis-what-does-it-mean-to-have-a-multiple-axis-pick-and-place-or-multiple-axis-robot/) [cited at p. xiii, 19, 21]
- [26] Manuel Silva. “Noções Básicas de Robótica” Disponível em: [https://moodle.isep.ipp.pt/pluginfile.php/119901/mod\\_resource/content/2/ROBIN\\_Cap\\_2\\_Nocoas\\_Basicas\\_Robotica\\_2017.pdf](https://moodle.isep.ipp.pt/pluginfile.php/119901/mod_resource/content/2/ROBIN_Cap_2_Nocoas_Basicas_Robotica_2017.pdf) [cited at p. xiii, xvii, 22, 23, 26, 27, 30, 31]
- [27] ABB. “Product specification - IRB 2400” Disponível em: <https://library.e.abb.com/public/a87a296caab448da904a35e09890cad6/3HAC042195-en.pdf> [cited at p. xiii, 23]
- [28] RAD. “Ultimatic Pneumatic Stainless Steel Collision Sensors” Disponível em: <http://www.rad-ra.com/Collision-Sensors-Pneumatic-Stainless-Steel.htm> [cited at p. xiii, 24]
- [29] ABB. “IRB 140 - Data sheet” Disponível em: [http://search.abb.com/library/Download.aspx?DocumentID=PR10031EN\\_R15&LanguageCode=en&DocumentPartId=&Action=Launch](http://search.abb.com/library/Download.aspx?DocumentID=PR10031EN_R15&LanguageCode=en&DocumentPartId=&Action=Launch) [cited at p. xiii, 25]
- [30] Luis A.M. Riascos. “Fundamentos de Robótica: manipuladores e robôs móveis” Disponível em: <https://graduacao.ufabc.edu.br/ear/conteudo/ensino/disciplinas/Robotica/FundamentosRobotica.html> [cited at p. xiii, 28]
- [31] João Maurício Rosário. “Robótica Industrial I Modelagem, Utilização e Programação, Volume 1” Disponível em: <https://graduacao.ufabc.edu.br/ear/conteudo/ensino/disciplinas/Robotica/FundamentosRobotica.html> [cited at p. 29, 30, 31]
- [32] Jinbo Wu, Zhouping Yin. “A Novel 4-DOF Parallel Manipulator H4” Disponível em: [https://www.intechopen.com/books/parallel\\_manipulators\\_towards\\_new\\_applications/a\\_novel\\_4-dof\\_parallel\\_manipulator\\_h4](https://www.intechopen.com/books/parallel_manipulators_towards_new_applications/a_novel_4-dof_parallel_manipulator_h4) [cited at p. xiii, 29]

- [33] Robotpark Academy. “CARTESIAN and GANTRY Robots” Disponível em: <http://www.robotpark.com/academy/all-types-of-robots/stationary-robots/cartesian-gantry-robots/> [cited at p. xiii, 29]
- [34] Machine Design. “What’s the Difference Between Industrial Robots?” Disponível em: <http://www.machinedesign.com/robotics/what-s-difference-between-industrial-robots> [cited at p. xiii, 30]
- [35] B. Siciliano “Robotics Foundations I - Robot Modelling” Disponível em: [http://www.federica.eu/1/introduction\\_to\\_robotics#20](http://www.federica.eu/1/introduction_to_robotics#20) [cited at p. xiii, 31]
- [36] Norhafiza Samion “Industrial Robot Technology” Disponível em: <https://www.slideshare.net/mkazree/dek3223-chapter-2-robotic> [cited at p. xiii, 31]
- [37] Francisco Restivo. “Robótica” Disponível em: [ftp://ftp.fe.up.pt/pub/Pessoal/Deec/fjr/ai/tsai\\_robotica.ppt](ftp://ftp.fe.up.pt/pub/Pessoal/Deec/fjr/ai/tsai_robotica.ppt) [cited at p. 32]
- [38] Manuel Silva. “Atuadores Finais para Robôs Industriais” Disponível em: [www.moodle.isep.ipp.pt/pluginfile.php/119902/mod\\_resource/content/2/ROBIN\\_Cap\\_3\\_Actuadores\\_para\\_Robos\\_2017.pdf](http://www.moodle.isep.ipp.pt/pluginfile.php/119902/mod_resource/content/2/ROBIN_Cap_3_Actuadores_para_Robos_2017.pdf) [cited at p. xiv, 32, 33]
- [39] Direct Industry “Pneumatic gripper / parallel / 2-jaw / for small loads” Disponível em: <http://www.directindustry.com/prod/schunk/product-12463-551794.html> [cited at p. xiii, 32]
- [40] Lift Rite “Steel Grippers & Handling by Magnetic Grippers” Disponível em: [http://www.liftright.ie/products/steel\\_lifting\\_by\\_mech\\_grippers.asp](http://www.liftright.ie/products/steel_lifting_by_mech_grippers.asp) [cited at p. - ]
- [41] Jenny Eagle “PWR Pack expands its production facilities as it moves into 3D robotic grippers” Disponível em: <https://www.bakeryandsnacks.com/Article/2016/02/01/PWR-Pack-moves-into-3D-robotic-grippers> [cited at p. -]
- [42] Siemens “SIMATIC Industrial Flat Panel” Disponível em: <http://w3.siemens.com/mcms/pc-based-automation/en/industrial-pc/industrial-monitors/simatic-ifp/pages/default.aspx> [cited at p. xiv, 34]
- [43] Eclipse Knowledgebase “Metodologia para Desenvolvimento de IHMs de Alta Performance Visual” Disponível em: <http://kb.elipse.com.br/pt-br/questions/5289/Metodologia+para+Desenvolvimento+de+IHMs+de+Alta+Performance+Visual>. [cited at p. xiv, 34, 35, 36, 37, 38, 39]
- [44] Simone Barbosa, Bruno Silva “Interação Homem Computador”, edição de 2010 Disponível em: <https://pt.scribd.com/document/341611719/Simone-Barbosa-and-Bruno-Silva-Interacao-Homem-Computador> [cited at p. 35, 36]

- [45] ISO "ISO 9241-11:2017(en) Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts"Disponível em: <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en> [cited at p. 35]
- [46] ISO "ISO/IEC 25010:2011 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models"Disponível em: <https://www.iso.org/standard/35733.html> [cited at p. 35]
- [47] U.S. Chemical Safety and Hazard Investigation Board "Investigation Report No. 2005-04-I-TX Refinery Explosion and Fire"Disponível em: <http://www.csb.gov/assets/1/19/CSBFinalReportBP.pdf> [cited at p. xiv, 36, 37]
- [48] Siemens "Operator devices"Disponível em: <http://w3.siemens.com/mcms/human-machine-interface/en/operator-devices/Pages/Default.aspx> [cited at p. 39]
- [49] Siemens "SIMATIC WinCC Runtime Advanced (TIA Portal)"Disponível em: <http://w3.siemens.com/mcms/automation-software/en/tia-portal-software/wincc-tia-portal/wincc-tia-portal-runtime/wincc-runtime-advanced/Pages/Default.aspx> [cited at p. 39]
- [50] Siemens "SIMATIC WinCC Runtime Professional (TIA Portal)"Disponível em: <http://w3.siemens.com/mcms/automation-software/en/tia-portal-software/wincc-tia-portal/wincc-tia-portal-runtime/wincc-rt-professional/Pages/default.aspx> [cited at p. 40]
- [51] Siemens "SIMATIC WinCC (TIA Portal) - Runtime Software"Disponível em: <http://w3.siemens.com/mcms/automation-software/en/tia-portal-software/wincc-tia-portal/wincc-tia-portal-runtime/Pages/default.aspx> [cited at p. xiv, 40]
- [52] 123RF "Icon pictogram apresentando estágios diferentes no processo"Disponível em: [https://pt.123rf.com/photo\\_44803706\\_icon-pictogram-apresentando-est%C3%A1gios-diferentes-no-processo-agr%C3%ADcola-e-de-jardinagem-arar-semear-regar-p.html](https://pt.123rf.com/photo_44803706_icon-pictogram-apresentando-est%C3%A1gios-diferentes-no-processo-agr%C3%ADcola-e-de-jardinagem-arar-semear-regar-p.html) [cited at p. xiv, 42]
- [53] RobotWorx "Speed vs Flexibility: 4-axis vs 6-axis robots"Disponível em: <https://www.robots.com/articles/viewing/speed-vs-flexibility-4-axis-vs-6-axis-robots> [cited at p. 45]
- [54] ABB "IRB 660 features and specification"Disponível em: <http://search.abb.com/library/Download.aspx?DocumentID=PR10284EN&LanguageCode=en&DocumentPartId=&Action=Launch> [cited at p. xvii, 46]

- [55] ABB "IRB 6640 features and specification"Disponível em: <http://search.abb.com/library/Download.aspx?DocumentID=ROB0001EN&LanguageCode=en&DocumentPartId=&Action=Launch> [cited at p. xvii, 46]
- [56] JMP Engineering "Robot Grippers"Disponível em: <http://www.jmpengineering.com.au/systems/palletising/robot-grippers/> [cited at p. xiv, 47]
- [57] ABB "Grippers for palletizing – Clamp, Claw, Vacuum"Disponível em: <https://search-ext.abb.com/library/Download.aspx?DocumentID=9AKK105408A0210&LanguageCode=en&DocumentPartId=&Action=Launch> [cited at p. xiv, 48, 49, 51]
- [58] Robot Palletizer "Bag Palletizing"Disponível em: <http://www.robotpalletizer.com/bag-palletizing.html> [cited at p. xiv, 49]
- [59] Alligator Automation "Bag Hand Gripper"Disponível em: <http://www.alligatorautomations.com/bags.html> [cited at p. xiv, 49]
- [60] Igus "Multiple bag gripper for robotic palletising"Disponível em: [https://www.igus.eu/wpck/2247/app\\_igli\\_bagripper](https://www.igus.eu/wpck/2247/app_igli_bagripper) [cited at p. xiv, 49]
- [61] Mason Nicholson "Vacuum handling for the masses "Disponível em: <http://www.hydraulicspneumatics.com/vacuum-technology/vacuum-handling-masses-0> [cited at p. xiv, 50]
- [62] Fluidotronica "Motivos para aposta em garras de vácuo FIPA"Disponível em: [http://www.fluidotronica.com/pt/noticias\\_fluidotronica.159/5\\_motivos\\_para\\_apostar\\_em\\_garras\\_de\\_vacu\\_fipa.a1496.html](http://www.fluidotronica.com/pt/noticias_fluidotronica.159/5_motivos_para_apostar_em_garras_de_vacu_fipa.a1496.html) [cited at p. 50]
- [63] Premier Tech "Paletizadora de caixa robótica"Disponível em: <https://www.ptchronos.com/pt-pt/produtos/paletizacao-robotica/paletizadoras-de-caixa-robotica/paletizadora-automatica-pacotes/> [cited at p. 52, 53]
- [64] ACE - Automated Cells & Equipment "Robotic Palletizing Solutions"Disponível em: <http://www.autocells.com/robotic-palletizing-solutions> [cited at p. xiv, 52]
- [65] Premier Tech "Paletizadora de caixa robótica (paletizadora de fila)"Disponível em: <https://www.ptchronos.com/pt-pt/produtos/paletizacao-robotica/paletizadoras-de-caixa-robotica/paletizadora-robotica-fileiras/> [cited at p. xiv, 53, 54]
- [66] Premier Tech "Paletizadora de caixa robótica (paletizadora de camada)"Disponível em: <https://www.ptchronos.com/pt-pt/produtos/paletizacao-robotica/paletizadoras-de-caixa-robotica/paletizadora-robotica-camada/?videoid=A6pXr0RUjaE> [cited at p. 54, 55]

- [67] Elettric 80 "Palletization Solutions"Disponível em: <http://www.elettric80.com/en/products/palletization-solutions/> [cited at p. xiv, 54, 56]
- [68] Elettric 80 "Robot Turbo Eagle"Disponível em: <http://www.elettric80.com/en/products/palletization-solutions/robot-turbo-eagle/> [cited at p. xiv, 59]
- [69] Elettric 80 "Paletizadoras de saco robóticas"Disponível em: <https://www.ptchronos.com/pt-pt/produtos/paletizacao-robotica/paletizadoras-de%20saco-rob%C3%B3ticas%20/> [cited at p. xiv, 60]
- [70] Industrial PC "Nuvo-3000"Disponível em: <https://industrialpc.com/products/embedded/nuvo-3000/> [cited at p. xv, 76]
- [71] Rahul Shekhawat "Profibus vs profinet"Disponível em: <https://pt.slideshare.net/RahulShekhawat/profibus-vs-profinet> [cited at p. 78]
- [72] Fernando Lobo "Modelo Relacional"Disponível em: <https://pt.slideshare.net/nitrocell/modelo-relacional-bases-de-dados> [cited at p. 85]
- [73] Jesualdo Fernandes, Ana Lucas, Chaves Magalhães, Pedro Neves "Modelo Relacional"Disponível em: <https://www.iseg.ulisboa.pt/aquila/getFile.do?method=getFile&fileId=75066> [cited at p. 85]
- [74] W3Schools "ADO Recordset Object"Disponível em: [https://www.w3schools.com/asp/ado\\_ref\\_recordset.asp](https://www.w3schools.com/asp/ado_ref_recordset.asp) [cited at p. 105]
- [75] Departamento de Robótica - ABB Portugal "Introdução ao IRC5" [cited at p. xvi, 119, 120]
- [76] ABB Robotics "Conveyor Tracking - Application Manual"Disponível em: <http://isa.uniovi.es/~jalvarez/abb/en/3HAC16587-en.pdf> [cited at p. xvi, 121]



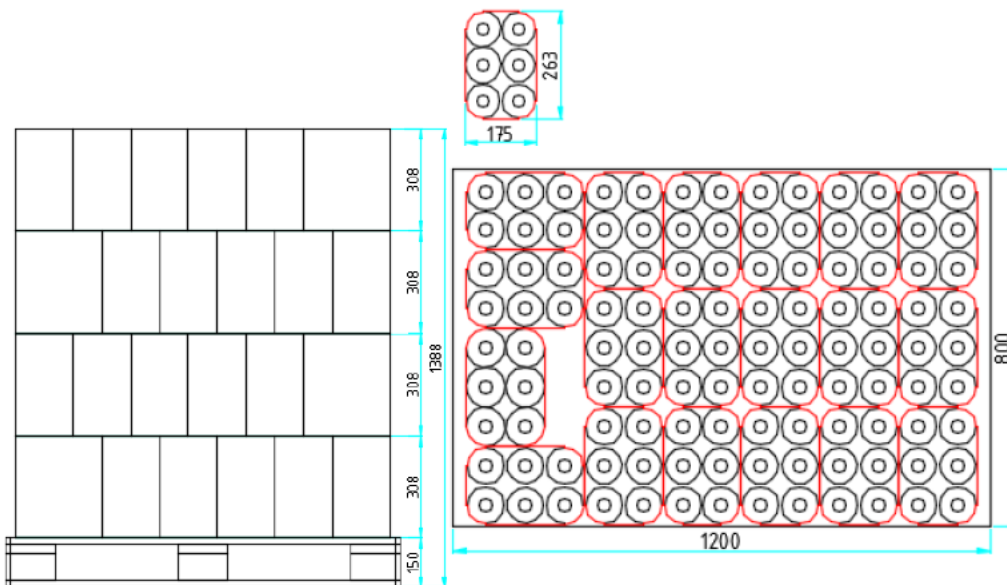
## Apêndice A

---

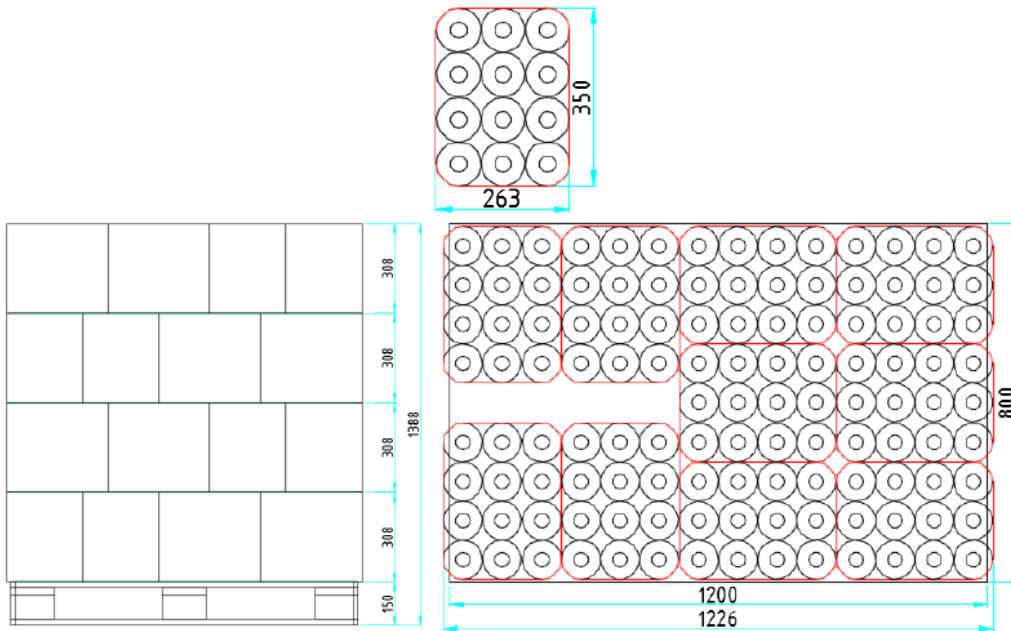
# Esquemas das Paletes a Formar na Ilha de Paletização

---

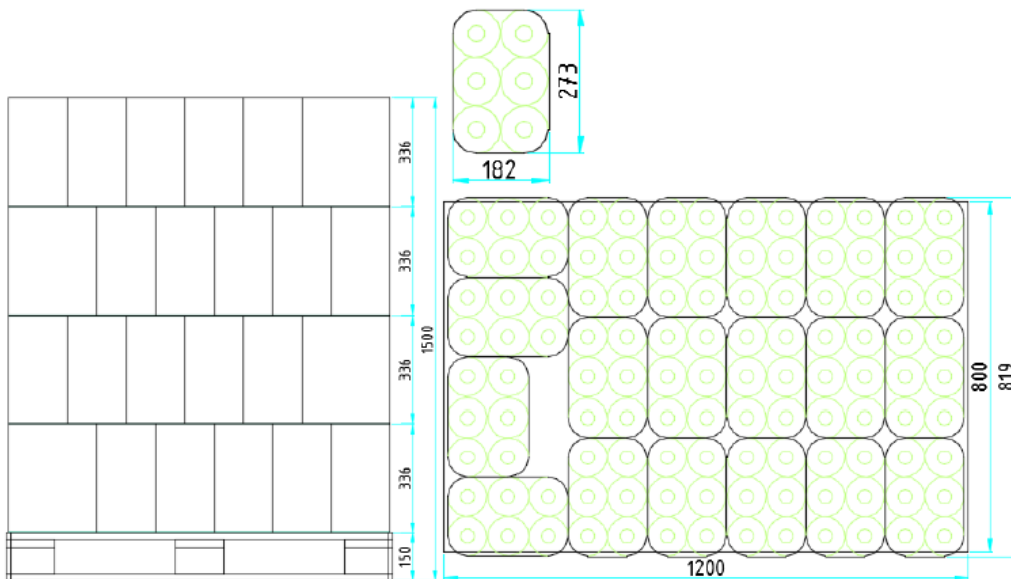
*Pack* Laranja 3x2 (1,25 l)



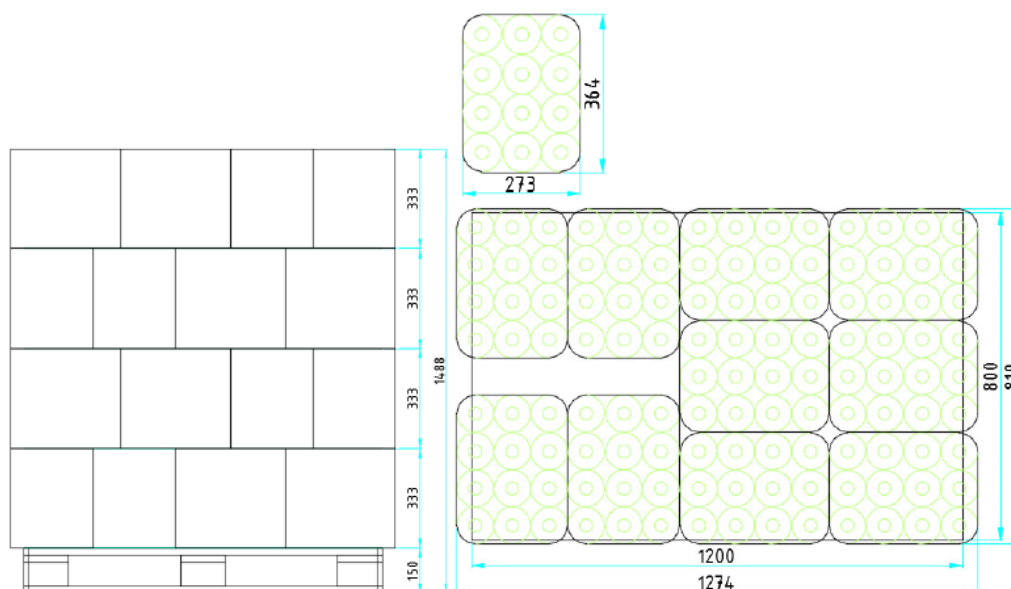
*Pack Laranja 4x3 (1,25 l)*



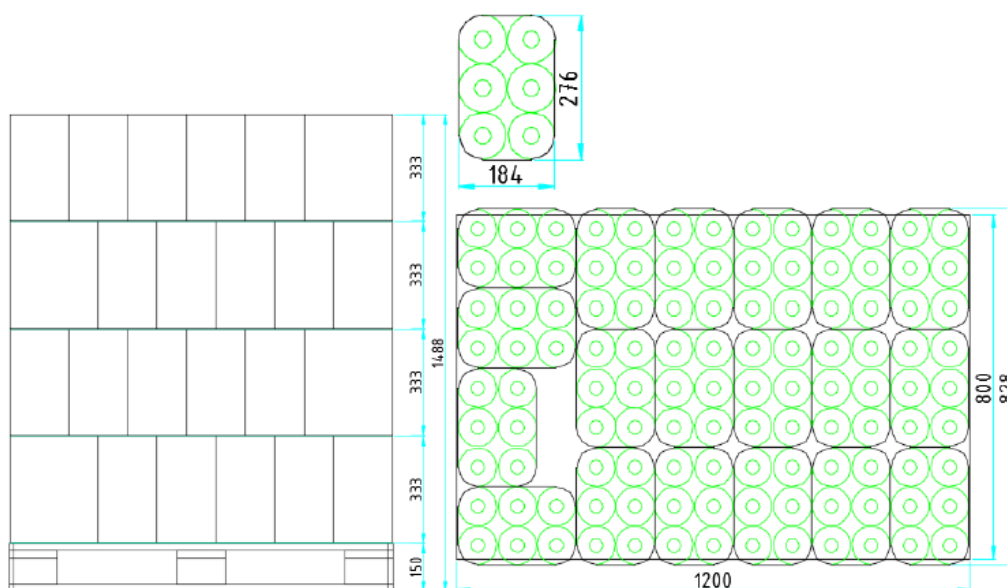
*Pack Laranja 3x2 (1,50 l)*



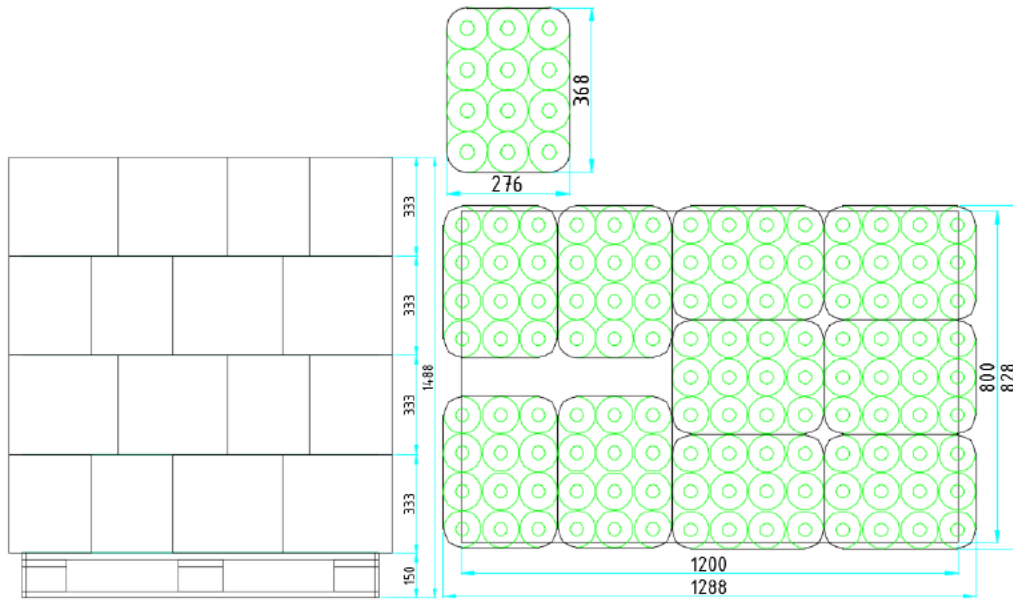
*Pack Laranja 4x3 (1,50 l)*



*Pack Ananás 3x2 (1,50 l)*



Pack Ananás 4x3 (1,50 l)



## **Apêndice B**

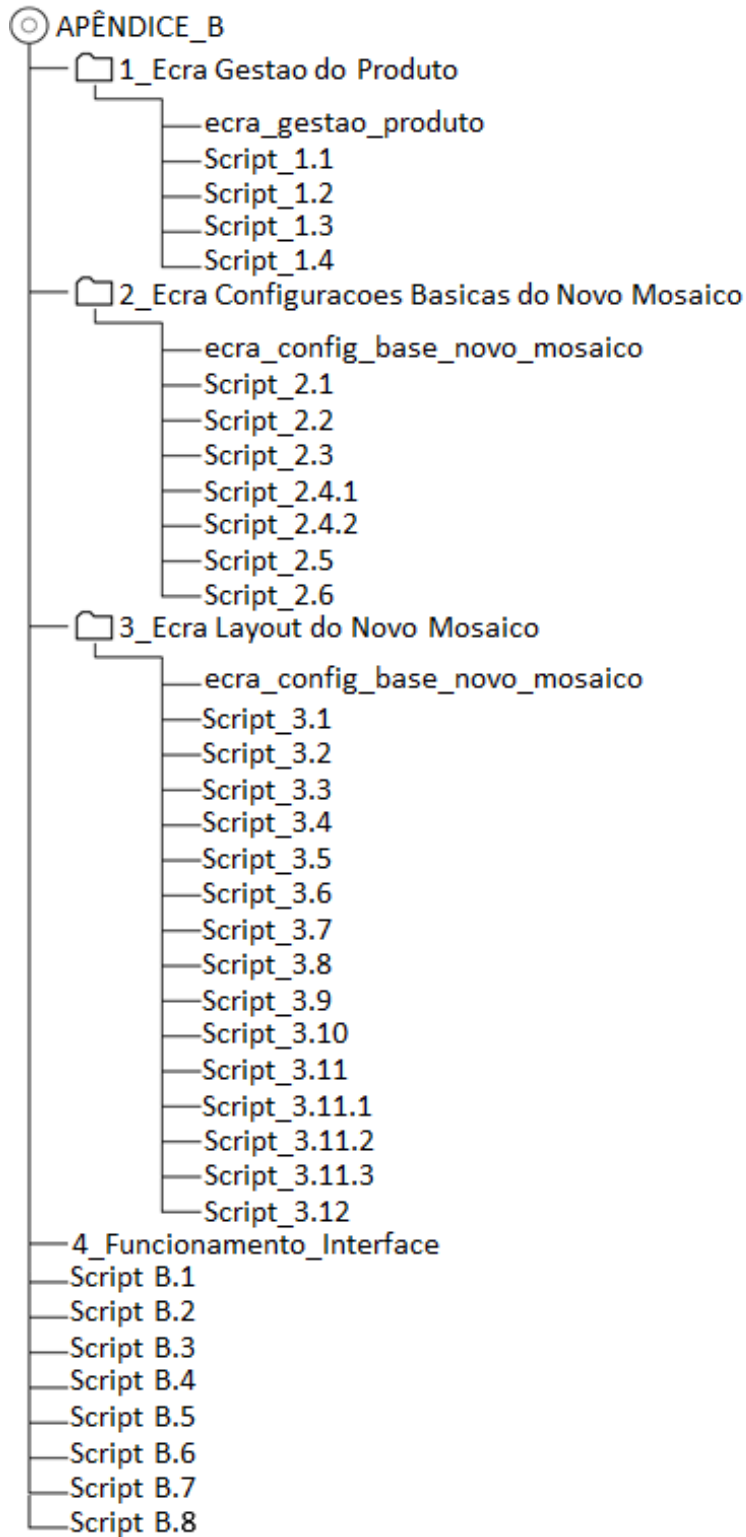
---

# **Programação dos procedimentos da interface em Visual Basic**

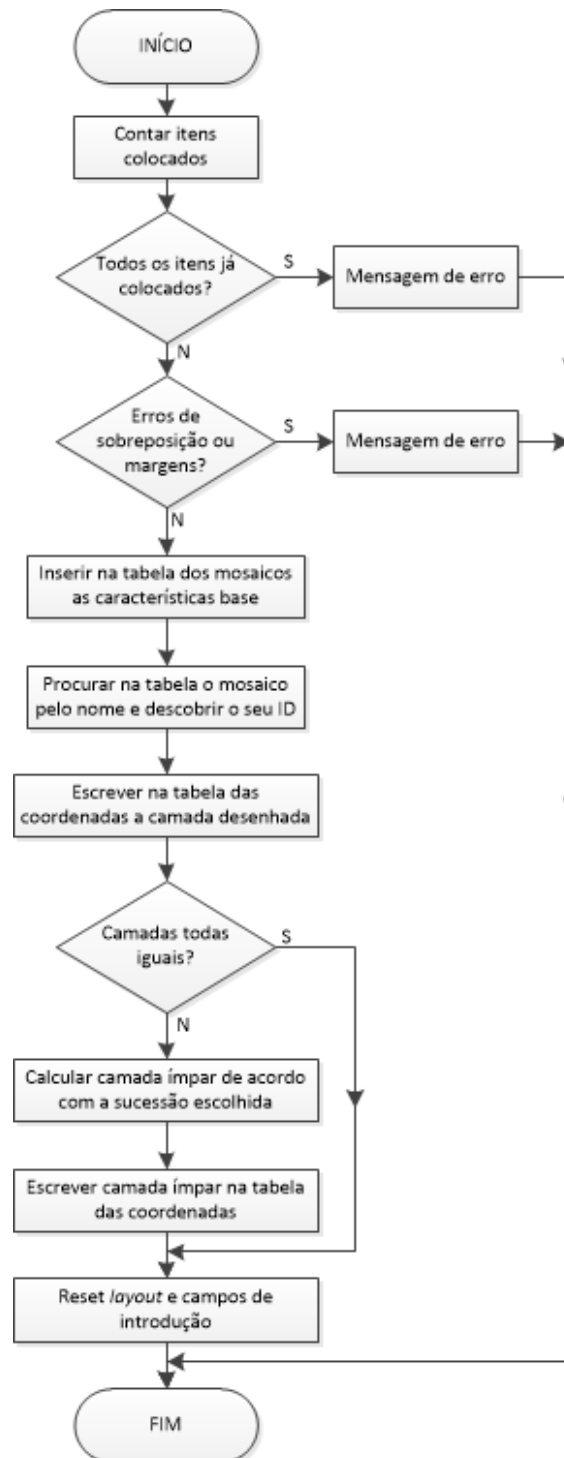
---

Os ficheiros a ser apresentados no Apêndice B, à exceção do "Fluxograma da criação de mosaicos", encontram-se no CD anexo a este relatório. Neste CD pode ser encontrada uma pasta "Apêndice B" que está organizada segundo o esquema da página seguinte.

Organização da pasta APENDICE B do CD anexo ao  
relatório



## Fluxograma da criação de mosaicos





## Apêndice C

---

# Implementação de Ilha de Paletização no RobotStudio

---

Os ficheiros a ser apresentados no Apêndice C encontram-se no CD anexo a este relatório. Neste CD pode ser encontrada uma pasta "Apêndice C" que está organizada segundo o esquema da página seguinte.

Organização da pasta APENDICE C do CD anexo ao relatório

