



Universidade de Évora - Escola de Ciências e Tecnologia

Mestrado em Engenharia Mecatrónica

Trabalho de Projeto

Desenvolvimento de um Veículo Autoguiado Baseado num Raspberry Pi

Ricardo André Zambujo Peixe

Orientador(es) | Fernando Manuel Janeiro
Mouhaydine Tlemcani

Évora 2022



Universidade de Évora - Escola de Ciências e Tecnologia

Mestrado em Engenharia Mecatrónica

Trabalho de Projeto

Desenvolvimento de um Veículo Autoguiado Baseado num Raspberry Pi

Ricardo André Zambujo Peixe

Orientador(es) | Fernando Manuel Janeiro
Mouhaydine Tlemcani

Évora 2022



O trabalho de projeto foi objeto de apreciação e discussão pública pelo seguinte júri nomeado pelo Diretor da Escola de Ciências e Tecnologia:

Presidente | João Manuel Figueiredo (Universidade de Évora)

Vogais | Fernando Manuel Janeiro (Universidade de Évora) (Orientador)
Frederico José Lapa Grilo (Instituto Politécnico de Setúbal) (Arguente)

Agradecimentos

Sem descrever com detalhe e de igual importância, estas ressalvas destinam-se a todos aqueles que contribuíram de algum modo para a execução deste trabalho de projeto, fazendo parte desta concessão a minha família, namorada sem-par, amigos dentro e fora do ambiente acadêmico e respectivos orientadores, que dentro de elogios e críticas procuram o melhor de mim, para promover um projeto à altura das minhas capacidades, contribuindo diretamente para a minha evolução pessoal. Compartilho com privilégio estes conhecimentos e agradeço o apoio incondicional prestando nesta etapa, agora finalizada. Muito obrigado a todos por tudo.

Em especial destaque, agradeço ao meu avô materno, sendo a quem dedico este trabalho, embora já não me conheça como neto, mas como desconhecido, não esqueço a memória que tinha por mim.

Resumo

Desenvolvimento de um Veículo Autoguiado Baseado num Raspberry Pi

Face à rápida expansão tecnológica vivenciada no presente século, existe uma procura contínua por enquadrar novas ferramentas, a fim de otimizar o rendimento dos processos fabris, através da redução de tempos de produção, custos associados, riscos para os trabalhadores, etc. Como resposta aos tradicionais mecanismos de transportes, que atualmente ainda dominam este setor, apresenta-se uma tecnologia com tendências crescentes, fornecendo a flexibilidade necessária e o rigor exigido, através de processos independentes de ações humanas. Este produto é o *Automated Guided Vehicle*, comumente designado por AGV. Conforme esta motivação, este trabalho de projeto visa demonstrar o desenvolvimento de um AGV de construção original baseado num Raspberry Pi 4B, segundo um modelo POC (*Proof of Concept*), com o intuito de seguir um sistema de guias, que perfaz um percurso operacional. Este sistema é um sistema unidirecional, com entroncamentos, cruzamentos e mudanças de direção, que serão realizadas de forma automatizada através da leitura dos sinais, integrados no piso, interpretados por visão artificial. O método de navegação do AGV assenta num sistema fechado, uma vez que existe um layout fixo no piso. Em termos de orientação, o AGV apresenta um sistema de deslocação diferencial e integra na sua estrutura uma câmara digital, que permite realizar a aquisição de imagem sendo o único método de aquisição do ambiente ao seu redor. Toda a informação obtida é processada por um sistema dedicado, que através do OpenCV (*Open Source Computer Vision Library*) e um algoritmo computacional desenvolvido em Python, realiza o processamento das imagens e realiza as decisões necessárias para uma navegação satisfatória. O processo de seleção de guias é automatizado pelo AGV, mas o Utilizador pré-define as rotas que este percorrerá, de modo que ao fim de um número de percursos programados, o AGV repete novamente os trajetos planeados. A portabilidade do AGV é assegurada por uma bateria, que é gerida por uma BMS (*Battery Management System*), acoplada numa estrutura física modelada e construída por um processo de tecnologia de fabricação aditiva – deposição de material fundido (FDM), de forma a conseguir-se um design económico, único e compacto.

Palavras-chave: AGV; Raspberry Pi; Visão Artificial; OpenCV; BMS

Abstract

Development of an Automated Guided Vehicle Based on a Raspberry Pi

In view of the rapid technological expansion experienced in the present century, there is a continuous demand for developing new tools in order to optimize the performance of manufacturing processes, through the reduction of production times, associated costs, risks for workers, etc. As a response to the traditional transport mechanisms, which currently still dominate this sector, a technology with growing trends is presented, providing the necessary flexibility and the required rigor, through processes independent of human actions. This product is the *Automated Guided Vehicle*, commonly referred as AGV. According to this motivation, this project aims to demonstrate the development of an AGV a original construction based on a Raspberry Pi 4B, according to a POC model (*Proof of Concept*), with the aim to follow a system of guides, which makes up an operational route. This system is an unidirectional system, with junctions, intersections and changes of direction, which will be carried out in an automated way by reading the signs, integrated into the floor, interpreted by artificial vision. The navigation method of the AGV is based on a closed system, since there is a fixed layout on the floor. In terms of orientation, the AGV features a differential displacement system and integrates a digital camera in its structure, which allows for image acquisition, being the only method of acquiring the environment around it. All the information obtained is processed by a dedicated system, which, through OpenCV (*Open Source Computer Vision Library*) and a computational algorithm developed in Python, performs the processing of the images and makes the necessary decisions for a satisfactory navigation. The guide selection process is automated by the AGV, but the User pre-defines the routes it will follow, so that after a number of programmed routes, the AGV repeats the planned routes again. The portability of the AGV is ensured by a battery, which is managed by a BMS (*Battery Management System*), coupled to a physical structure modeled and built using an additive manufacturing technology process – fused deposition modeling (FDM), in order to achieve an economical, unique and compact design.

Keywords: AGV; Raspberry Pi; Artificial Vision; OpenCV; BMS

Conteúdo

Resumo	ii
Abstract	iii
Lista de Figuras	vi
Lista de Tabelas	ix
Acrónimos	x
1 Introdução	1
1.1 Objetivos	1
1.2 Estrutura	2
1.3 Contribuições Originais	2
2 AGV's	3
2.1 Breve História	3
2.2 Benefícios dos AGV's	4
2.3 Layouts de Orientação	4
2.4 Sistemas de Navegação	6
2.4.1 Navegação Aberta	6
2.4.2 Navegação Fechada	7
2.5 Configurações de Locomoção	8
2.6 Sistemas de Alimentação	10
3 Visão Artificial	12
3.1 Processamento de Imagem	12
3.1.1 Aquisição de Imagem	13
3.1.2 Pré-processamento	15
3.1.3 Segmentação	17
3.1.4 Extração de Características	17
3.1.5 Detecção	18
3.1.6 Processamento de Alto Nível	19
3.2 OpenCV	20
4 Desenvolvimento do AGV	21
4.1 Hardware Mecânico	21
4.1.1 Modelação CAD	21
4.1.2 Técnica Aditiva de Fabrico	24
4.1.3 Dimensionamento dos Motores	26
4.2 Hardware Elétrico	32
4.2.1 Bateria Elétrica	32
4.2.2 BMS	35

4.2.3	Conversor Buck	44
4.2.3.1	LM2596	45
4.2.3.2	XL4015	45
4.2.4	Raspberry Pi	46
4.2.5	Câmara Digital	47
4.2.6	Sistema de Iluminação	48
4.2.7	Ponte H	49
4.3	Firmware	52
4.3.1	Sistema Operativo	52
4.3.2	Comunicação AGV-Utilizador	54
4.3.3	Arquitetura do Algoritmo Computacional	59
4.3.4	Captura de Imagem	61
4.3.5	Binarização	63
4.3.6	Transformações Morfológicas	66
4.3.7	Contornos	69
4.3.8	Contadores	77
4.3.9	Comportamentos Direcionais	78
4.3.9.1	Seguir em Frente	80
4.3.9.2	Mudar de Direção	83
4.3.10	Tomada de Decisão	86
4.3.11	Definição de Rotas	88
4.3.12	Controlo Motriz	91
5	Testes e Resultados	93
5.1	Percurso Operacional	94
5.2	Gestão Energética	95
5.2.1	Corrente de Carga	95
5.2.2	Tensão de Carga	96
5.2.3	Rendimento Energético	98
5.2.4	Autonomia	101
5.3	Performance Computacional	102
5.3.1	Tempo por Ciclo	102
5.3.2	Qualidade do Processamento de Imagem	104
5.4	Performance Motriz	106
6	Conclusão e desenvolvimentos futuros	110
	Anexo A Desenho Técnico	113
	Anexo B Esquema Elétrico	131
	Anexo C Algoritmo Computacional	133
	Referências	141

Lista de Figuras

2.1	Layout Convencional adaptado de [7]	5
2.2	Layout em Loop Único adaptado de [7]	5
2.3	Layout Tandem adaptado de [7]	5
2.4	Navegação por Triangulação Laser [8]	6
2.5	Orientação via Sonar [8]	6
2.6	Tipos de Orientação Indutiva [8]	7
2.7	Orientação ótica [8]	7
2.8	Orientação por Grelha Magnética [8]	8
2.9	Orientação Inercial [8]	8
2.10	Locomoção Diferencial [8]	9
2.11	Locomoção tipo Triciclo [8]	9
2.12	Locomoção <i>Ackerman</i> [8]	9
2.13	Sistema elétrico, híbrido e diesel, respetivamente [14].	10
2.14	Evolução do preço das células de Li-ion adaptado de [15]	11
3.1	Metodologia para o processamento de imagem	12
3.2	Processo de aquisição segundo uma câmara analógica [19]	13
3.3	Processo de aquisição de uma imagem digital [19]	14
3.4	Seção transversal de um sensor sob o arranjo do Filtro de Bayer [19]	14
3.5	Exemplo de conversão para o domínio das frequências [22]	16
3.6	Exemplo de remoção de ruído com um filtro passa-baixo [22]	16
3.7	Exemplo de segmentação [22]	17
3.8	Exemplo de deteção de arestas [22]	18
3.9	Exemplo de deteção da face humana [22]	19
3.10	Exemplo de identificação do rosto de um ser humano [22]	19
4.1	Benefícios das ferramentas CAD	22
4.2	Estrutura física do AGV modelado via Fusion 360	22
4.3	Corte transversal da estrutura do AGV a destacar as principais ligações aparafusadas	23
4.4	Modelo CAD do AGV	23
4.5	Modelação por deposição através de termoplásticos	24
4.6	Conversão do modelo CAD para o formato .STL	25
4.7	Conversão do formato .STL para o formato .GCODE	26
4.8	Componente obtido pela modelação por deposição de PLA	26
4.9	Diagrama do corpo livre do AGV	27
4.10	Conjunto motor-roda motriz	29
4.11	Conjunto parte traseira-roda giratória	30
4.12	Motor elétrico selecionado	31
4.13	Distribuição energética do AGV	32
4.14	Célula de lítio ICR18650-22E	33
4.15	Layout 3S2P	33

4.16	Localização da bateria elétrica	34
4.17	Diferenças na presença de BMS no processo de carregamento	35
4.18	Regulador linear LM317T	36
4.19	Esquema elétrico para obtenção de uma tensão constante	37
4.20	Esquema elétrico para obtenção de uma tensão constante e corrente limitada	38
4.21	Dissipador passivo + LM317T	39
4.22	Circuito responsável pelo balanceamento das células	40
4.23	Fluxo de corrente elétrica no estágio inicial do carregamento da bateria	41
4.24	Balanceamento da bateria elétrica	42
4.25	Processo de concepção da PCB	43
4.26	Modelo CAD da PCB	43
4.27	Modo de operação do conversor buck	44
4.28	Conversor LM2596	45
4.29	Conversor XL4015	45
4.30	Arquitetura do sistema de aquisição de imagem	48
4.31	L298N	49
4.32	Alteração da polaridade aplicada no motor elétrico	50
4.33	Variação da tensão média (U_m) com diferentes ciclos de trabalho	51
4.34	Cartão microSD utilizado	52
4.35	Raspberry Pi OS	53
4.36	Rede envolvente para a conexão AGV-Utilizador	55
4.37	Criptografia simétrica entre o Utilizador e o AGV	56
4.38	Decomposição de um endereço IPv4	57
4.39	Configuração da túnel criptográfico a partir do PuTTY + ligação VNC	58
4.40	Relação entre <i>scripts</i>	59
4.41	Fluxograma do algoritmo do AGV	60
4.42	Algoritmo da captura de imagem	61
4.43	Reestruturação da imagem adquirida na formação do ROI	62
4.44	Algoritmo de binarização	64
4.45	Segmentação através da binarização monocromática	65
4.46	Algoritmo basilar das transformações morfológicas	66
4.47	Atuação do elemento estruturante ao longo de uma imagem adaptado de [19]	67
4.48	Atuação do fechamento na amenização do ruído	68
4.49	Algoritmo basilar para a extração de características	69
4.50	Definição de formas e lacunas segundo [98]	70
4.51	Etapa 1.(a)	71
4.52	Etapa 2.(a)	71
4.53	Etapa 2.(b)	72
4.54	Etapa 2.(c)	72
4.55	Etapa 2.(d)	72
4.56	Relação hierárquica entre entidades relacionadas à formação dos contornos	73
4.57	Deteção de contornos	73
4.58	Algoritmo de Douglas – Peucker	75
4.59	Elementos de interesse não detetados devido à ausência de guia	76
4.60	Elementos de interesse detetados devido à presença de guia	76
4.61	Algoritmo basilar para a contabilização de sinais	77
4.62	Situações com múltiplos troços	80
4.63	Algoritmo para seguir em frente	81
4.64	Mudança de Direção: Seguir em frente	82
4.65	Interseção: Seguir em frente	83
4.66	Algoritmo para mudar de direção	84

4.67	Mudança de Direção: Mudar de direção	85
4.68	Algoritmo basilar para a tomada de decisão	87
4.69	Exemplo de um percurso operacional	88
4.70	Trajetos exequíveis dentro do percurso definido	89
4.71	Localização do AGV face à guia a seguir	91
4.72	Comportamentos motrizes	92
5.1	AGV desenvolvido em funcionamento	93
5.2	Representação esquemática do percurso operacional	94
5.3	Corrente de carga ao longo de 3 ciclos de carregamento	96
5.4	Tensão de carga ao longo de 1 ciclo de carregamento	97
5.5	Variação da temperatura no regulador de tensão linear LM317T	98
5.6	Comparação entre a potência entregue ao sistema em comparação com a potência útil	99
5.7	Rendimento energético da BMS durante o balanceamento das células	100
5.8	Consumo energético do AGV	101
5.9	Tempo entre cada ciclo concluído	103
5.10	Imagem em análise capturada pelo AGV	104
5.11	Distribuição de cores captados num modelo RGB com diferentes iluminações	104
5.12	Distribuição de cores captados num modelo HSV com diferentes iluminações	105
5.13	Variação do centroide da guia	107
5.14	Variação do erro relativo em cada amostra adquirida	108
5.15	Variação do centroide da guia	109
5.16	Variação do erro relativo em cada amostra adquirida	109

Lista de Tabelas

2.1	Caraterísticas das baterias de Chumbo-Ácido, Níquel-Cádmio e Lítio adaptado de [15]	11
4.1	Propriedades mecânicas do PLA adaptado de [40]	25
4.2	Dimensionamento das caraterísticas mecânicas do AGV	31
4.3	Especificações do Raspberry Pi 4 modelo B [60]	46
4.4	Especificações do Raspberry Pi câmara V2 [62]	47
4.5	Comandos alocados no diretório <code>/etc/dhcpd.conf</code>	58
4.6	Exemplo de definição de trajetos no ficheiro <code>percursos.py</code>	90
4.7	Modo de funcionamento dos motores elétricos na formação dos comportamentos motrizes	92
5.1	Resumo das propriedades estatísticas do gráfico 5.13	107
5.2	Resumo das propriedades estatísticas do gráfico 5.15	109

Acrónimos

AGV	Automated Guided Vehicle
ROI	Region of Interest
POC	Proof of Concept
BMS	Battery Management System
Li-ion	Lithium-Ion
CAD	Computer Aided Design
PCB	Printed Circuit Board
PWM	Pulse Width Modulation
LED	Light Emitter Diode
VCC	Common Collector Voltage
GND	Ground
OpenCV	Open Source Computer Vision Library
IDE	Integrated Development Environment
IP	Internet Protocol address
SSH	Secure Socket Shell
VNC	Virtual Network Computing
RoHS	Restriction of Certain Hazardous Substances
HASL	Hot Air Solder Leveling
FDM	Fused Deposition Modeling
PLA	Polylactic acid
GPIO	General Purpose Input/Output
CPU	Central Processing Unit
GPU	Graphics Processing Unit

Capítulo 1

Introdução

O desejo por automatizar tarefas realizadas pelo ser humano é antigo, ousando desde sempre pela procura de mecanismos que proporcionem um trabalho equivalente a este, ou mesmo por idealizar sistemas com desempenhos acima das suas capacidades físicas, de modo a melhorar a produtividade e a qualidade dos processos. Esta demanda e intenção no desenvolvimento destes mecanismos resultou no que hoje é a robótica, uma área científica interdisciplinar que reúne um vasto leque de ferramentas e modelos científicos, que promovem o desenvolvimento de sistemas autónomos. Em particular, motivado pelo desenvolvimento tecnológico crescente e limitações em termos de logística e problemas de gestão/organização de grandes inventários, surgiu a robótica móvel num contexto industrial, como resposta a esta necessidade crescente, a fim de melhorar o rendimento e rigor destes procedimentos. Neste domínio surgem os AGV's, como uma entidade que embebe dualmente os conceitos orquestrados pela Indústria 4.0, através do desenvolvimento de sistemas ciber-físicos descentralizados, tal como se transformam numa resposta primária face aos maiores volumes de produção, ou justamente pela possibilidade de empregabilidade de serviços de cognição, promovendo a concessão de sistemas inteligentes com poder de processamento, conseguido tomar decisões lógicas e criteriosas sob matrizes definidas, cumprindo em todas as ocasiões sem exceções, a ética deontológica introduzida em sistemas desta natureza. Em particular, no presente trabalho de projeto, demonstrar-se-á o desenvolvimento de um AGV tutelado por sistemas de visão artificial, evidenciando a sua potencialidade e abrangência num ambiente real, destacando igualmente as suas virtudes e defeitos, de modo a tramitar um conteúdo franco, redigido sob os moldes da engenharia atual.

1.1 Objetivos

O objetivo primordial do presente trabalho de projeto é a promoção direta do conhecimento científico nesta vertente, acrescentando-se valor pelo desenvolvimento de um veículo autoguiado, segundo um formato *Proof of Concept*, privilegiando-se a conceção de um sistema que utiliza serviços *Open-Source*, de modo a garantir a livre exposição dos vários produtos utilizados, a fim de possibilitar um desenvolvimento futuro livre de restrições, ou quaisquer conflitos em relação aos termos de serviço. Assim, preservando a autenticidade deste sistema, o AGV desenvolvido prioriza como objetivos a cumprir, a garantia de um seguimento autónomo e inteligente, apenas com base no processamento de imagem adquirido em tempo real, a partir de ferramentas de visão artificial, juntamente com um algoritmo desenvolvido em Python. Visto a cognição que se planeia integrar neste sistema, o AGV torna-se num sistema inteligente e independente de ações humanas, no requisito das ações a concretizar, nomeadamente no seguimento autónomo dos trajetos planeados, diferenciando-se de outros sistemas deste género, principalmente pela capacidade de mudança de direção integrada em situações duais (situações com múltiplas escolhas), promovendo tomadas de decisão, de acordo com o planeamento efetuado pelo Utilizador. Para não restringir a sua manobrabilidade e garantir a sua real autonomia motriz, ressalva-se também como objetivo, a integração não só de uma bateria elétrica dedicada, como um sistema de balanceamento (BMS), a fim de proteger as células de carregamentos elétricos danosos.

O último grande objetivo estipulado para o presente trabalho de projeto, passa pela elaboração de toda uma estrutura física que agregue todo o hardware do AGV numa única carcaça, com o objetivo de salvaguardar a integridade física dos vários elementos e promover um design baseado numa arquitetura, que otimize o alcance bem-sucedido dos objetivos estipulados. Toda esta estrutura é modelada através de um software CAD, onde todas as partes funcionais à exceção das ligações aparafusadas asseguradas por ferragens em metal, serão produzidas por um processo de deposição de material fundido (FDM), a partir de um termoplástico, a fim de garantir-se uma estrutura customizada e leve, diminuindo o esforço mecânico nos motores elétricos responsáveis pela locomoção do AGV. Todos estes elementos devem contribuir para a produção de um sistema preciso e exato, no seguimento dos trajetos definidos pelo utilizador num contexto real.

1.2 Estrutura

Estruturalmente, o presente trabalho de projeto é dividido em seis capítulos, estando dividido de forma a diferenciar diferentes tópicos colocados de ordem lógica, de modo a formar-se um fio condutor para o entendimento do tema exposto.

- **Capítulo 1.** No presente Capítulo, ocorre uma introdução ao tema a desenvolver, tal como uma descrição singela dos componentes mais relevantes que se irão enfatizar ao longo dos próximos capítulos, que motivaram a sua execução.
- **Capítulo 2.** No segundo Capítulo, ocorre uma revisão literária, sobre o estado de arte dos AGV's, destacando-se as principais características ligadas ao presente trabalho de projeto, como os principais sistemas de navegação, layouts de orientação e estilos de deslocação motriz.
- **Capítulo 3.** No terceiro Capítulo, à semelhança do Capítulo anterior, ocorre também uma revisão literária, mas neste caso sobre os fundamentos teóricos do processamento de imagem, construindo-se um raciocínio completo, que parte deste das ações mais elementares a realizar neste campo, até aos processos de mais alto nível, onde se empregam processos de cognição associados à inteligência artificial. Ainda nele se apresenta o que é o OpenCV.
- **Capítulo 4.** No quarto Capítulo e principal, apresenta-se em pormenor todo o desenvolvido técnico que permitiu desenvolver o AGV, inicialmente explicitando toda a construção mecânica e dimensionamento dos motores elétricos, seguindo-se para o dimensionado elétrico da bateria e respetiva BMS desenvolvida de raiz, destacando-se ainda os vários subsistemas elétricos deste sistema. Por último, completa-se com exposição do firmware, destacando-se todo o processo de comunicação remota AGV-Utilizador e algoritmo computacional que promove o seguimento de trajetos a partir do processamento de imagem.
- **Capítulo 5.** No quinto Capítulo, ocorre a apresentação dos resultados oriundos dos testes realizados às características mais importantes do AGV, que neste caso são a performance da BMS, o desempenho computacional do sistema desenvolvido e o rigor no seguimento dos trajetos planeados.
- **Capítulo 6.** No sexto Capítulo, é apresentado as várias conclusões retiradas do presente trabalho de projeto, onde se efetua uma retrospectiva crítica do trabalho desenvolvido. Ainda é mencionado alguns possíveis desenvolvimentos futuros, como forma de dar continuidade ao tema exposto.

1.3 Contribuições Originais

Do presente trabalho de projeto, resultou uma publicação científica no comité VII Jornadas do ICT 2022, em conferência nacional, intitulada por “Desenvolvimento de Veículo Autoguiado em Raspberry Pi” resultado uma exposição oral com explicação do trabalho desenvolvido [1].

Capítulo 2

AGV's

A introdução da robótica móvel no cotidiano e em especial em ambientes industriais, tornou os AGVs em componentes cruciais na logística requerida aos tempos atuais. O padrão tecnológico e o nível atual de experiência com esta tecnologia de automação levaram à dispersão do AGV em quase todos os ramos da indústria e áreas de produção, tornando-o numa solução imprescindível e competitiva face a outras tecnologias já bem estabelecidas no mercado laboral.

2.1 Breve História

A história dos AGVs começou há quase setenta anos, em 1953 nos Estados Unidos, quando após a segunda guerra mundial, as indústrias retornaram à sua atividade levando a um crescimento rápido da economia global. Graças a este período de prosperidade económica e aos avanços tecnológicos marcados por esta época, fazia agora parte dos planos do Homem criar máquinas para realizar trabalhos executados exclusivamente por este. O rápido desenvolvimento e aprimoramento de tecnologias sensoriais (à base de sensores), bem como os desenvolvimentos da microeletrónica pavimentou o caminho para o desenvolvimento dos AGVs. Em 1954 surge o primeiro AGV comercial utilizado num armazém de mercearia nos Estados Unidos. Nessa instância os AGVs eram guiados por uma guia no chão ou por um sensor ótico [2].

O AGV trazia muitas vantagens face às tecnologias maioritariamente implementadas na época como o aumento da eficiência em comparação com mecanismos semiautomáticos já vastamente integrados, redução genérica de custos de produção e redução de tempos entre etapas, entre outras características. Ao efetuar-se uma comparação entre a performance do AGV face às mesmas tarefas efetuados pelo Homem, conclui-se que este seria uma tecnologia revolucionária nesta área, e por isso deu-se início a uma grande demanda que impulsionou o desenvolvimento de tecnologias mais avançadas para os puder aprimorar. No final dos anos 80, houve uma recessão rigorosa que atingiu a maioria das indústrias. Isto levou a que o desenvolvimento do AGV fosse interrompido, porque estes eram muitos caros e era necessário economizar dinheiro devido ao período que se vivia. Quase em simultâneo, a indústria de fabricação de automóveis japonesa introduziu o conceito *Lean Production*, que tem como ideias reduzir custos de operação e aumentar a sua qualidade. Este conceito encaixa na essência e no porquê da utilização dos AGVs e por isso já no final da década de 90, graças aos constantes melhoramentos nas tecnologias implementadas nos AGVs, estes já possuíam outras formas de navegação como a navegação a laser, que deu início a uma nova era em que os AGVs eram mais flexíveis e confiáveis [2][3]. Esta era continua até hoje, e embora os primeiros AGVs fossem caros, difíceis de instalar e configurar, inflexíveis e não confiáveis, os AGVs modernos melhoraram até se tornarem um símbolo de confiabilidade e eficiência, sendo usados em uma grande variedade de indústrias (farmacêutica, química, automotiva, tipográfica, alimentar, etc) [4]. No entanto, um sistema AGV representa um investimento significativo para uma empresa, portanto, este sistema é quase exclusivamente para grandes empresas.

2.2 Benefícios dos AGV's

Como foi descrito na seção anterior, os AGVs apresentam um elevado nível de integração na sociedade atual, diversificando-se por várias áreas. O uso cada vez mais recorrente dos AGVs deve-se essencialmente aos benefícios que estes trazem, comparativamente a processos com o mesmo objetivo, mas com outros formatos de realização. A escolha dos AGVs ao invés de outras tecnologias é influenciada por alguns pontos fortes que se seguem.

1. **Maior Flexibilidade.** Os AGVs modernos facilmente são instalados e propensos a modificações que os permite integrar com outros equipamentos.
2. **Melhor segurança.** A segurança dos trabalhadores é garantida pela presença de luzes e sirenes, que alertam sobre a sua aproximação. Se ainda assim a aproximação do AGV for uma surpresa, este é equipado com sensores que permitem a paragem imediata, quando existe o risco de colisão com algum material em movimento, ou com algum trabalhador. Para além disso, o AGV viaja em caminhos pré-definidos arquitetados para evitarem bloqueios ou colisões e possuírem o mínimo de troços comuns em relação à área operacional dos trabalhadores.
3. **Redução do custo operacional.** Estes podem funcionar 24/7 a uma velocidade constante e consistente. Não precisam de férias, pausas para o almoço ou licença médica. Embora necessitem de serem carregados e realizarem revisões conforme o estilo de manutenção implementado.
4. **Redução do custo de trabalho.** Reduz-se a mão de obra, pois existe menos dinheiro gastos em salários, treinos e formações, baixas médicas, etc.
5. **Total controlo de inventários.** A posição e o status dos AGVs são constantemente rastreados e controlados por um sistema central, que armazena toda a informação do inventário.
6. **Diminuição de danos.** Os sensores incluídos impedem o choque entre veículos, paredes e cargas. Para além disso, o transporte de cargas é mais seguro, diminuindo o risco de danos na carga durante o transporte [5].

Em resumo, os AGVs conferem às empresas uma vantagem competitiva aumentando a sua produtividade e eficiência do tempo, além de reduzir custos associados. Para além disso, a longo prazo a integração desta tecnologia revela-se muito vantajosa devido à facilidade em os equipar com tecnologias atuais não os tornando obsoletos nem restringido a sua capacidade de atuação.

2.3 Layouts de Orientação

A definição do layout de navegação é uma parte importante no projeto de todo o sistema, uma vez que influencia diretamente no espaço necessário para realizar as operações em segurança, no número de veículos, no número de estações de carregamento, na complexidade do sistema, etc. De acordo com Tuan Le-Anh [6] o layout pode ser classificado como: Convencional, Loop Único e Tandem. Um layout convencional conecta todas as estações de trabalho, onde os AGVs intervêm, através de entroncamentos, cruzamentos, atalhos, etc. Estes tipos de layouts podem ser: unidirecionais ou bidirecionais. O mais popular é um layout unidirecional, onde os veículos navegam apenas em uma direção. Pelo contrário, num layout bidirecional, os veículos podem deslocar-se nos dois sentidos. Aparentemente é uma vantagem, pois consegue-se diminuir a área de operação, contudo facilmente estes layouts ficam bastante complexos devido ao risco de bloqueios ou colisões entre AGVs. Este problema pode ser resolvido ao utilizarem-se várias faixas de circulação, de forma paralela, na medida em que o sentido de deslocamento de um AGV numa das faixas é independente do sentido que existe nas restantes faixas. Por conta desta solução é necessário impor uma distância ampla entre cada faixa, de modo a assegurar-se deslocamentos seguros. Esta solução leva a uma maior ocupação de área para os AGVs poderem operar e por isso torna este layout mais dispendioso.

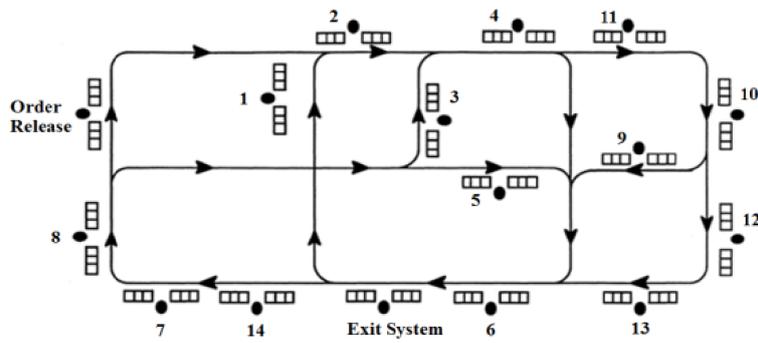


Figura 2.1: Layout Convencional adaptado de [7]

O layout em loop único apenas difere do layout convencional na medida em que os AGVs viajam sem nenhum atalho ou rotas alternativas. A viagem bidirecional também é menos utilizada, porque como não existem rotas alternativas, a interferência entre veículo é mais propensa a acontecer. Para se conseguir o mesmo desempenho que um layout convencional é necessário uma maior frota de AGVs, tal como uma melhor gestão de todo o processo.

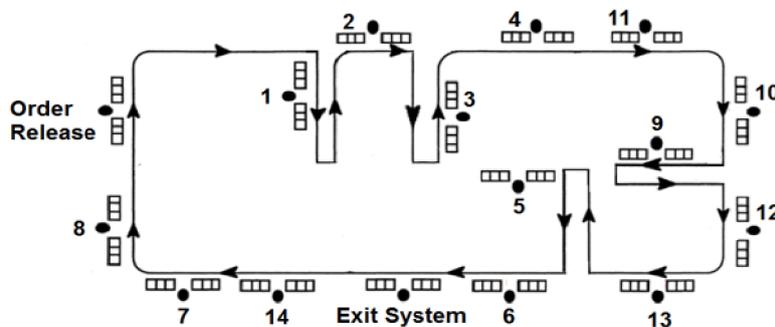


Figura 2.2: Layout em Loop Único adaptado de [7]

Por último, o layout Tandem é um layout dividido por zonas ou seções. A sua metodologia de funcionamento consiste apenas na circulação de um AGV por zona, e por isso os problemas de bloqueio ou interferência não existem, uma vez que existe somente um AGV por cada seção formada. Esta configuração permite múltiplas possibilidades de transporte, uma vez que se a carga precisar de ser movida de um local para outro, então através de áreas de transferência efetua-se esta deslocação para outra seção. O único local de interação entre loops é somente estas áreas de transferência. É um layout que facilita em termos de gestão de rotas, porém em percursos de menor dimensão é uma solução bastante dispendiosa devido ao alto número de veículos integrados para promover um sistema de deslocação funcional [7].

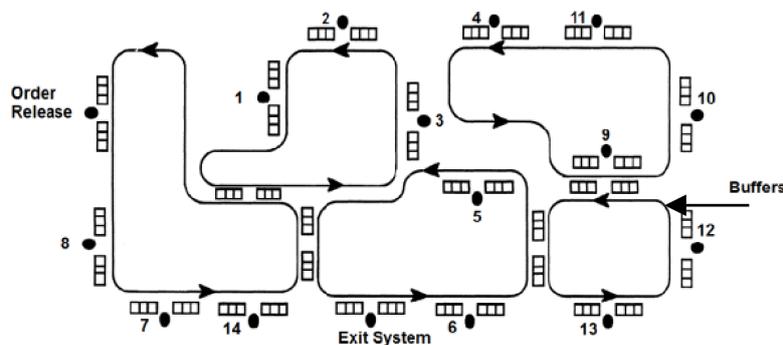


Figura 2.3: Layout Tandem adaptado de [7]

2.4 Sistemas de Navegação

Os AGVs podem ser classificados conforme vários aspectos tecnológicos e características físicas, porém, o tipo de sistema de navegação utilizado é a característica mais importante, de modo que existem somente duas classificações principais para o tipo de sistema de navegação incorporado nos AGVs: **Navegação Aberta** e **Navegação Fechada**. A seleção do tipo de sistema de navegação é realizada com base na aplicação a que se destina a sua integração, custos, área operacional, volume de percursos, complexidade de trajetos, condições de operação, entre outras características relevantes [2].

2.4.1 Navegação Aberta

Estes sistemas não possuem caminho físicos, embora os caminhos sejam praticamente pré-definidos na unidade de controle. Ao não se utilizar caminhos físicos, este sistema é utilizado para encobrir problemas relacionados à flexibilidade, visto ter a capacidade de alterar de rota, sem alterar fisicamente qualquer layout que lhe esteja associado. Dentro desta categoria existem dois tipos de sistema de navegação: Triangulação via Laser e Orientação via Sonar. Nestes sistemas, a localização do AGV é controlada por uma unidade supervisora, que emprega uma elevada precisão no trajeto que cada AGV realiza, o que o torna num sistema economicamente mais dispendioso comparado aos sistemas de navegação fechados [5].

Triangulação via Laser. Atualmente, este é o sistema de navegação mais popular, que funciona a partir de alvos reflexivos estabelecidos em posições bem definidas e conhecidas. O AGV utiliza um scanner a laser, como forma de encontrar estes alvos e, em seguida, com os algoritmos de controle que possui calcula a posição exata deste em função destes pontos de controle.

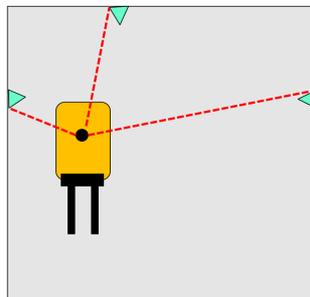


Figura 2.4: Navegação por Triangulação Laser [8]

Orientação via Sonar. É uma tecnologia alternativa à utilização de sistemas por triangulação via laser, uma vez que utiliza um sonar incorporado, para a produção de ultrassons como mecanismo de mapeamento. Este sistema tem menores custos financeiros, relativamente ao sistema acima, porém, apresenta uma precisão inferior, além de ser suscetível a perturbações induzidas por fatores externos ambientais.

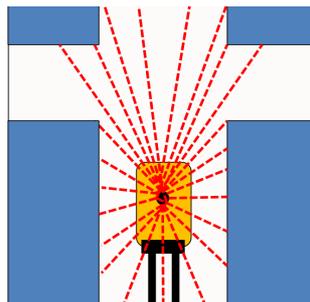


Figura 2.5: Orientação via Sonar [8]

2.4.2 Navegação Fechada

Estes sistemas caracterizam-se por necessitarem de meios de orientação, tal como pontos de referência ou guias fixas no solo, de forma a se poderem orientar por meio de um sensor acoplado (câmara, sensor infravermelho, antena, etc). Existem cinco sistemas de navegação que enquadram nesta categoria, sendo que dois baseiam-se em pontos de referência: Orientação Inercial e Grelha Magnética, enquanto os restantes sistemas de navegação percorrem os seus trajetos com base em guias fixas no solo: Orientação Ótica, Orientação Indutiva Passiva e Ativa. Note-se, que como a guia e os pontos de referência são fixos, a modificação do layout não é muito frequente, tornando este tipo de sistemas pouco flexíveis. No entanto, este tipo de sistemas são baratos e de fácil implementação [5]. Ainda se podem utilizar tecnologias mais avançadas, como etiquetas de identificação por radiofrequência (RFID), ou marcas visuais, como códigos de barras. Estas marcações funcionam como pontos de apoio, uma vez que podem fornecer informações ao AGV, de como interagir no próximo cruzamento ou tomada de decisão [3].

Orientação Indutiva. A navegação por indução funciona através da criação de um campo magnético ao longo da guia. Esta orientação pode ser ativa ou passiva, correspondendo a uma guia com um campo magnético ativo ou não. A navegação indutiva ativa, consiste na existência de um condutor de corrente elétrica colocado sob o piso, de modo a criar um campo magnético. O AGV deteta este campo graças às bobinas integradas nele, que permitem calcular desvios face à guia. Pelo contrário, na navegação indutiva passiva é utilizado uma fita magnética fixa no solo, de forma que esta é detetada por sensores magnéticos integrados no AGV. As duas técnicas apresentam diferentes vantagens, uma vez que na técnica de indução ativa o condutor elétrico colocado sob o piso fica imune contra possíveis acidentes causados pelo tráfego de AGVs e problemas como a deterioração das guias, porém, a técnica de indução passiva é mais flexível, permitindo a alteração do layout mais facilmente, com menores custos associados.

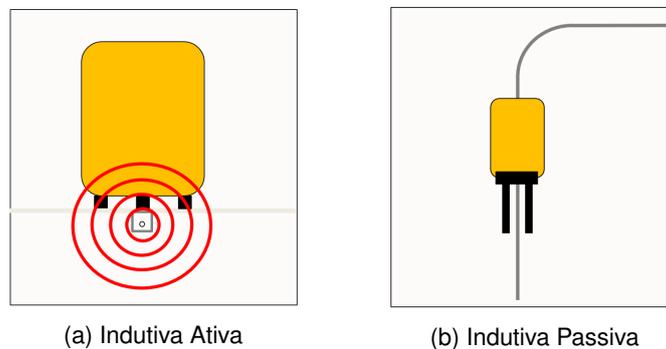


Figura 2.6: Tipos de Orientação Indutiva [8]

Orientação Ótica. É utilizado uma câmara de vídeo apontada para o solo, de forma que AGV siga a guia, a partir de ferramentas de processamento de imagem. A guia deverá ser marcada por uma fita colorida, ou uma linha pintada, de forma que a cor desta contraste com a cor do piso, como forma de facilitar a deteção das suas arestas.

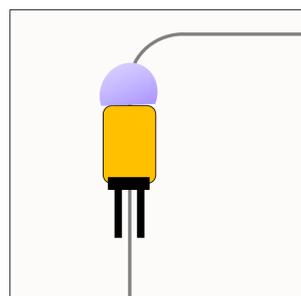


Figura 2.7: Orientação ótica [8]

Orientação por Grelha Magnética. O sistema de navegação por grelha magnética difere apenas no tipo de ímãs utilizado em sistemas de orientação inercial. Em alternativa, neste sistema os ímãs passivos são trocados por transponders quase-ativos, os quais são ativados por indução magnética, através de um sensor instalado no AGV. Esta configuração permite uma maior liberdade de movimentação, devido a uma maior precisão que capacita o AGV a obter informações adicionais sobre o Layout de navegação.

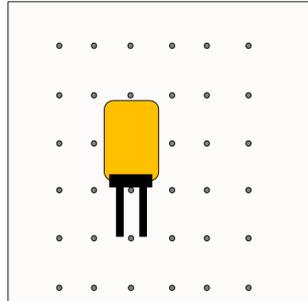


Figura 2.8: Orientação por Grelha Magnética [8]

Orientação Inercial. O sistema de navegação inercial utiliza feedback de três dispositivos diferentes (giroscópio, encoder e pontos de referência externos), como forma de determinar a sua posição. O giroscópio é responsável por medir a direção do veículo; o encoder calcula a distância percorrida; e os pontos de referência, que normalmente são ímãs passivos, são integrados no solo, em posições específicas, como forma de serem detetados posteriormente por um sensor acoplado no AGV. Este é um método altamente dependente da qualidade da odometria.

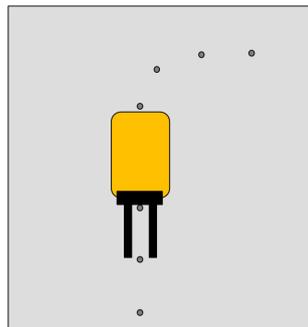


Figura 2.9: Orientação Inercial [8]

2.5 Configurações de Locomoção

A flexibilidade dos AGVs leva-os a uma fácil integração em qualquer ambiente *indoor*. Esta fácil integração é em grande parte conseguida pelas configurações motoras que influenciam a motricidade e a direção destes. Aspectos, como maneabilidade, complexidade, número de rodas, contexto de aplicação e custos económicos são fatores que influenciam na escolha de um sistema de locomoção. Existem quatro tipos principais de configurações motrizes: locomoção diferencial, locomoção tipo triciclo, locomoção *Ackerman* e locomoção omnidirecional. A locomoção diferencial consiste na integração de duas rodas motrizes independentes, de modo que a estabilidade do veículo é conseguida por uma terceira roda de apoio ou por um par de rodas de apoio. Esta configuração permite deslocamentos bastantes manobráveis, uma vez que o veículo pode girar sobre si próprio realizando raios de curvatura muito reduzidos comparado com os outros modelos. Em comparação com outras configurações motrizes, devido à sua alta mobilidade esta é a configuração que apresenta mais dificuldades em seguir uma rota com exatidão [9].

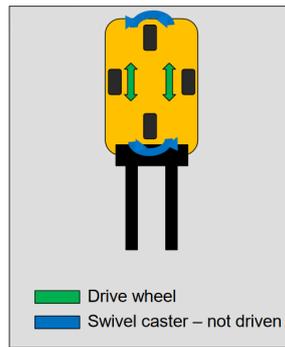


Figura 2.10: Locomoção Diferencial [8]

A locomoção tipo triciclo apresenta apenas uma roda motriz, sendo esta roda igualmente responsável por controlar a direção do AGV. As duas rodas empregues na traseira desta configuração são livres. Esta é a configuração mais comum para os AGVs, uma vez que consegue um rastreamento preciso da rota e em termos de manobrabilidade é aceitável para a maioria das aplicações não sendo tão manobrável, como o sistema de locomoção diferencial [5].

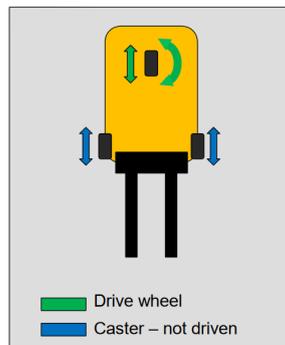


Figura 2.11: Locomoção tipo Triciclo [8]

A locomoção *Ackerman* apresenta duas rodas dianteiras responsáveis pela direção do veículo. Existe a opção destas rodas serem motrizes ou não. Independentemente desta escolha, as rodas dianteiras são rodas motrizes. Esta configuração consegue oferecer uma maior potência mecânica, o que é o ideal para o transporte de cargas mais pesadas, contudo, a sua potência e ergonomia mecânica tornam numa opção com maior gastos energéticos, o que leva a uma menor autonomia. Comparativamente aos modelos mencionados, apresenta um maior raio de curvatura, o que leva a ter dificuldades quando se depara com rotas, com mudanças de direção constantes, ou quando as rotas apresentam curvas com pequenos raios de curvatura [10].

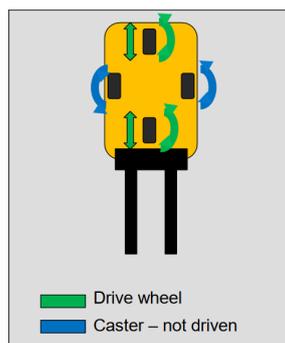


Figura 2.12: Locomoção *Ackerman* [8]

Menos usual, a locomoção omnidirecional possui três ou quatro rodas de tração, que permitem movimentar o AGV em qualquer direção. A diferença para as restantes configurações deve-se ao design da roda utilizada. A roda que integra este sistema é designada por roda *mecanum* e essencialmente é uma roda convencional com um número de rolos presos à sua circunferência. Esses rolos têm um eixo de rotação inclinado, estando fixos com juntas rotativas, de modo a girarem livremente em torno dos seus eixos. A obtenção do movimento omnidirecional é conseguido pelo controlo das velocidades angulares e do sentido de rotação, sem alterar a direção das rodas [11].

2.6 Sistemas de Alimentação

Embora alguns AGVs funcionem com motores a diesel, a maioria funciona com energia elétrica, fornecida por baterias recarregáveis. No caso dos motores a diesel, o diesel é um óleo derivado da destilação do petróleo bruto constituído essencialmente por hidrocarbonetos. Embora seja um produto pouco inflamável, pouco volátil, límpido e isento de matérias em suspensão, é um combustível fóssil e por isso limitado [12]. Para além deste fator, e face ao movimento atual de descarbonização, os combustíveis fósseis têm um grande impacto ambiental e motivar o desenvolvimento de novas tecnologias ainda com esta forma de gerar energia torna o processo de descarbonização mais demorado e ineficiente [13]. Nos últimos anos, a indústria automóvel deixou de construir exclusivamente motores de combustão interna de natureza puramente mecânica, passando a construir novos modelos híbridos. Porém, como uma confluência de forças, a indústria automóvel caminha num sentido progressivo para a inclusão de motores elétricos, e por isso impulsionam o grande desenvolvimento e aposta em veículos elétricos. A grande vantagem deste tipo de veículos deve-se à incorporação de baterias recarregáveis, uma vez que a fonte de alimentação destas pode ser obtida por energias renováveis, auxiliando na demanda atual de descarbonização [14].

As baterias eletroquímicas, mais comumente designadas por baterias, são dispositivos eletroquímicos que convertem energia elétrica em energia química durante o processo de carregamento. Quando descarregam, o processo é o inverso, convertendo energia química em energia elétrica. Uma bateria é composta por várias células empilhadas em série, de forma que uma célula é uma unidade independente e completa que possui todas as propriedades eletroquímicas. Uma célula de uma bateria é constituída por três elementos primários: dois elétrodos (um positivo e um negativo) e um eletrólito, substância onde estão imersos os elétrodos. A principal característica das baterias é a sua capacidade colométrica, ou seja, é o número de amperes-hora [Ah] ganho ao descarregar a bateria até um estado totalmente descarregado, onde a tensão aos terminais corresponda à tensão de corte. A capacidade das baterias diminui ao longo das sucessivas utilizações o que leva a possuírem um tempo de vida útil. Quando se alcança o fim de vida útil de uma bateria, a sua capacidade de armazenar energia é muito reduzida, tornando-a disfuncional. Existem inúmeros tipos de baterias no mercado com diferentes características e propriedades e por isso, é necessário conhecer algumas com interesse na área dos AGVs [15].

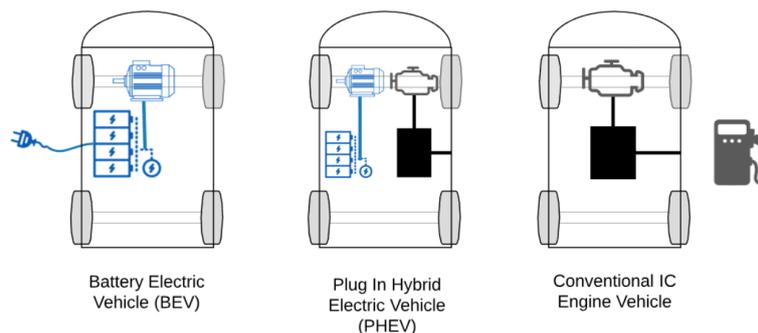


Figura 2.13: Sistema elétrico, híbrido e diesel, respetivamente [14].

As baterias de chumbo-ácido (*Lead-acid*) são as baterias mais utilizadas na indústria automotiva até à poucos anos, incluindo em sistemas como o AGV. Esta ampla utilização deve-se ao facto de serem baterias recarregáveis mais baratas, com 140 anos de desenvolvimento, o que a torna numa tecnologia confiável. A desvantagem é que esta bateria é muito pesada e tem um forte impacto ambiental. Já as baterias de níquel-cádmio (*Nickel-cadmium*) foram as principais concorrentes das baterias de chumbo-ácido, em termos da utilização em veículos elétricos. Elas possuem um ciclo de vida mais longo e quase o dobro da energia específica das baterias de chumbo-ácido. Porém, custam sensivelmente três vezes mais e são igualmente prejudiciais para o meio ambiente. As baterias de lítio (*Li-ion*) são as baterias mais leves e tornaram-se recentemente nas baterias mais comuns em veículos elétricos. Estas baterias geralmente apresentam uma alta densidade, que proporciona um grande armazenamento de energia. A única desvantagem é o preço, porém, este está a regredir a cada ano, uma vez que com o aumento do número de stock destas baterias, face à oferta existente, os preços vêm a diminuir também devido à maior implementação delas em vários setores. De 2010 a 2018 existiu uma queda de 85% no preço destas [5] [15].

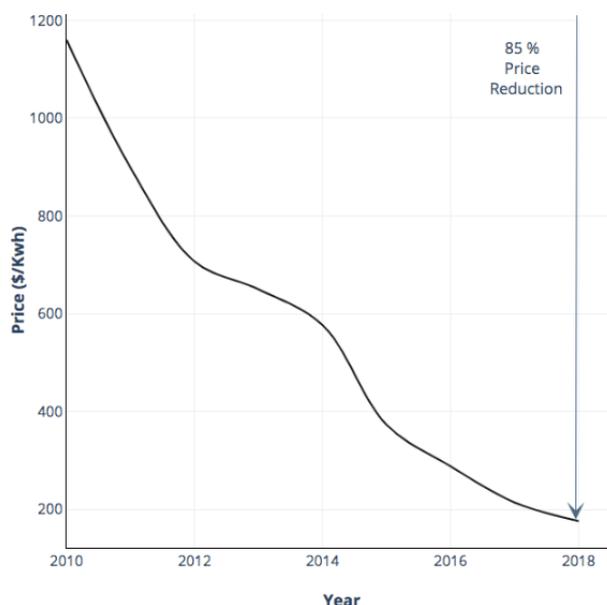


Figura 2.14: Evolução do preço das células de Li-ion adaptado de [15]

Fora as características mencionadas, na tabela 2.1 encontram-se as células mencionadas, descritas em outros campos técnicos relevantes à área deste trabalho.

Baterias	Lead-acid	Ni-cd	Li-ion
Energia específica (Wh/kg)	30-50	45-80	100-265
Densidade energética (Wh/L)	60-110	70-90	250-290
Potência específica (W/kg)	285	220	até 1350
Tensão nominal (V)	2.1	1.2	3.6
Capacidade (mAh)	4000-7000	600	2000-6000
Nº de carregamentos (80% do inicial)	700	1500	500-2000
Taxa de descarga (por mês)	5-20	10-20	2-8
Eficiência (%)	>80	75	>95
Custo (US\$/kWh)	100-200	300-600	300-1000
Toxicidade	Elevada	Elevada	Baixa
Reciclagem	Obrigatória	Obrigatória	Recomendável
Efeito memória	Não	Sim	Não
Manutenção	Não	Sim	Não

Tabela 2.1: Características das baterias de Chumbo-Ácido, Níquel-Cádmio e Lítio adaptado de [15]

Capítulo 3

Visão Artificial

A visão é o sentido mais avançado do Homem, como mecanismo de aquisição sensorial. Diferente dos outros sentidos, a visão tem capacidade de obter grandes quantidades de informação, de forma quase instantânea, apesar de estar limitada a uma banda do espectro eletromagnético [16]. Da tentativa de compreender o complexo da visão natural como principal mecanismo de aquisição, surge através da modelação e simulação um campo científico denominado por visão artificial. Assim, a visão artificial é um campo da inteligência artificial que permite a sistemas computacionais processarem imagens e vídeos digitais, com o objetivo destes sistemas tomarem ações ou realizarem recomendações de forma automatizada. Se a inteligência artificial permite que os computadores pensem, a visão artificial permite que eles vejam, observem e interpretem resultados, tudo isto graças ao processamento digital de imagens construído sobre uma base de formulações matemáticas e probabilísticas. Embora a visão artificial possa ultrapassar as capacidades do Homem, nomeadamente em processos com altas cadências de produção, a análise e intuíto deste é fundamental na realização de julgamentos visuais subjetivos [17].

3.1 Processamento de Imagem

Um sistema de visão artificial apresenta várias formas de realizar processamento de imagem, porém, genericamente integra-se um ordem lógica com o pretexto de otimizar e beneficiar-se de melhores resultados. Esta ordem forma um método que subdivide três níveis: baixo, médio e alto nível. Segundo Roy Devies [18] os processos de baixo nível envolvem operações primitivas, como o pré-processamento de imagem, com funções como a redução do ruído e melhoria do contraste e nitidez. Neste nível os *inputs* e *outputs* são apenas imagens, visto a natureza dos processos envolvidos. No nível médio o processamento envolve tarefas como a segmentação (partição da imagem em regiões) ou classificação (reconhecimento dos objetos na imagem). Neste nível os *inputs* geralmente são imagens, porém os *outputs* são atributos extraídos dessas imagens, como arestas, contornos, áreas, etc. Os processos de alto-nível estão relacionados com tarefas de cognição associadas com a visão humana.

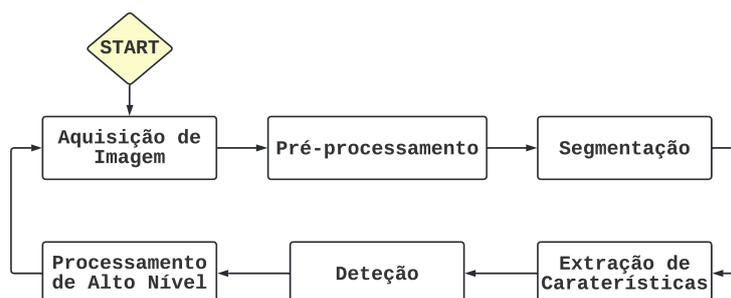


Figura 3.1: Metodologia para o processamento de imagem

3.1.1 Aquisição de Imagem

A aquisição de imagem é efetuada por dispositivos responsáveis pela captação e pela digitalização desta. Segundo Rafael Gonzalez [19] uma imagem pode ser definida como uma função bidimensional $f(x, y)$, onde x e y são coordenadas espaciais (planas), e a amplitude de f em qualquer par de coordenadas (x, y) é chamada de intensidade ou escala de cinzentos. Quando x, y e os valores de intensidade de f são todos quantitativamente finitos e discretos, chama-se de imagem digital. Os valores da intensidade de f são proporcionais à energia irradiada, a partir de ondas eletromagnéticas, que por sua vez dão origem a valores contínuos e analógicos que devem ser convertidos numa forma digital. O processo de conversão de uma cena tridimensional em uma imagem digital envolve uma redução da dimensionalidade, uma vez que o sensor que realiza a aquisição de imagem necessita de converter uma cena 3D em uma representação 2D. A aquisição pode ser realizada ou por uma câmara analógica, ou uma câmara digital. No caso do processo de aquisição de uma câmara analógica, esta opera através da discretização numa sequência de tensões elétricas correspondentes aos pixels em que a imagem foi discretizada. Esta informação, embora já sob a forma de um sinal elétrico, não é diretamente processável por uma unidade de processamento digital, uma vez que estas câmaras não possuem um processo de digitalização integrado, capaz de fornecer diretamente uma imagem digitalizada, sem a necessidade de uma interface auxiliar, responsável pela conversão do sinal analógico em digital. Logo, quando uma imagem é capturada precisa de ser digitalizada e guardada na memória antes de se efetuar uma nova aquisição de imagem. Comparativamente com a câmara digital, o sinal analógico é muito mais suscetível aos ruídos durante a transmissão, levando à adição de ruído no processo de conversão.

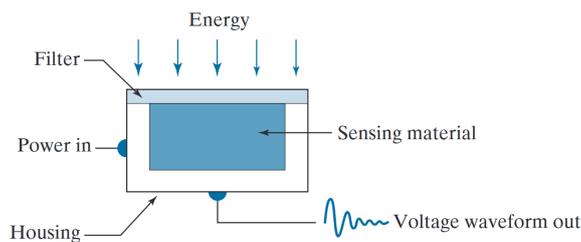


Figura 3.2: Processo de aquisição segundo uma câmara analógica [19]

Face a esta inconveniência, atualmente, o dispositivo mais utilizado é a câmara digital, onde tipicamente o sensor integrado apresenta uma matriz CCD (*Charge Coupled Device*) sendo constituída por uma matriz de semicondutores fotossensíveis, que trabalham como condensadores, fazendo um armazenamento da carga elétrica proporcional à energia luminosa incidente. Portanto, para criar uma imagem digital é necessário converter dados contínuos em dados com formato digital. Existem duas etapas para realizar esta conversão: Amostragem e Quantificação. A amostragem determina a resolução espacial da imagem digitalizada, convertendo-a numa única matriz (imagem monocromática) de dimensões $M \times N$, onde cada ponto de imagem é designado por pixel, de modo que

$$\begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \dots & f(1, N - 1) \\ \vdots & \vdots & & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & \dots & f(M - 1, N - 1) \end{bmatrix}. \quad (3.1)$$

Quanto maior a dimensão da matriz, maior será a resolução da imagem processada. A quantificação atribui a cada pixel um valor inteiro entre 0 e $2^n - 1$, onde n simboliza o número de bits utilizado. Assim, a digitalização, processo que se encarrega de formar imagens digitais, é uma conversão analógica-digital onde o número de amostras do sinal contínuo por unidade tempo indica a amostragem, enquanto que o número de bits (n) indica os níveis de cinzento, ou a intensidade de f de cada pixel como é destacado na figura 3.3.

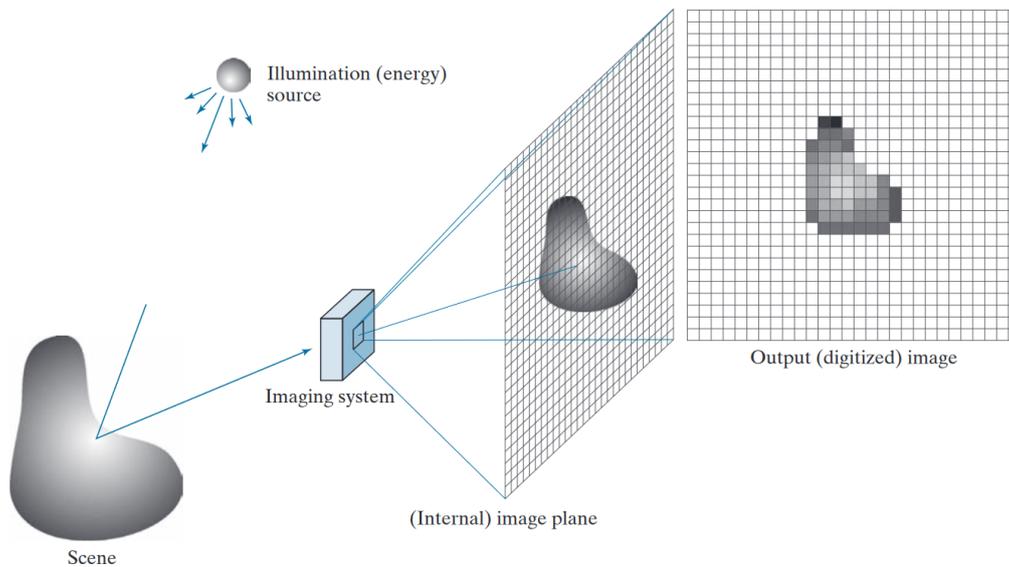


Figura 3.3: Processo de aquisição de uma imagem digital [19]

Quando se pretende capturar uma cena colorida é necessário não só utilizar uma função $f(x, y)$ para cada banda de cores primárias, como posteriormente realizar-se a decomposição da imagem através de prismas e filtros, uma vez que os sensores fotossensíveis não captam o comprimento de onda da luz incidente. Nestes termos introduz-se o sistema RGB, que é um modelo de cores representado por uma sigla anglo-saxónica para as cores primárias definidas **vermelho**, **verde**, e **azul**, respetivamente, onde pela mistura de intensidades entre as três cores primárias permite formar as restantes cores (cores secundárias), formando um vasto espectro cromático, visto este ser um sistema aditivo.

Digitalmente, a aquisição de uma cena colorida passa pela utilização de um filtro padrão, como o filtro de Bayer. Segundo este modelo, cria-se uma matriz quadrada, onde a cor verde tem o dobro da frequência, da cor vermelha e azul, formando um sistema designado por RGGB. Estas proporções estão relacionadas com a anatomia do olho humano, uma vez que este é mais sensível à cor verde, numa fase diurna e, portanto, como forma de aproximar com maior rigor a cena capturada face à imagem resultante, então justifica-se estas proporções. Deste modo, o resultado da aplicação do filtro de Bayer é uma imagem, onde cada pixel formado é representado apenas em uma das suas três possíveis componentes, ou seja, cada pixel especifica somente a cor vermelha, verde ou azul. Este processo devolve uma imagem em bruto da cena capturada, de modo que para replicar os gradientes de cores originais, é necessário interpolar através da substituição desta paleta de cores pela utilização de cores secundárias limitadas pelo número de bits utilizado (resolução), através da aplicação de um algoritmo de *demosaicing*, para se puder interpolar entre os possíveis valores das cores primárias em cada pixel da imagem.

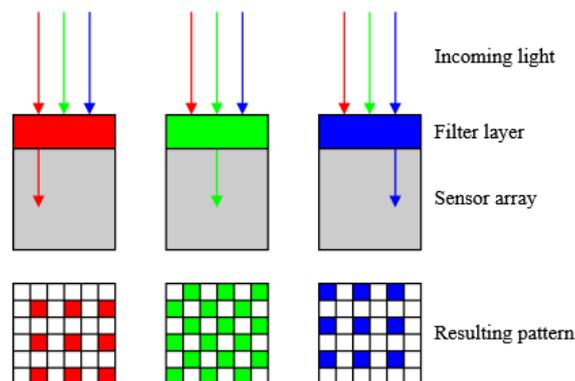


Figura 3.4: Seção transversal de um sensor sob o arranjo do Filtro de Bayer [19]

Dependendo dos algoritmos de *demosaicing* aplicados, diferentes imagens digitais são produzidas e com isso diferentes qualidades de imagem são obtidas, mesmo em imagem adquiridas com o mesmo número de bits, ou seja, com a mesma resolução. Qualquer imagem gravada numa memória, depois de sofrer todos os processos relacionadas à aquisição desta é guardada genericamente num formato JPG, que é um formato de imagem digital utilizado para comprimir as imagens adquiridas, sendo este um formato de arquivo digital. Esta compressão torna-se vantajosa, pois permite escolher o nível de compressão que se planeia implementar em cada imagem adquirida e com isso determinar qual o espaço que a imagem adquirida irá ocupar numa dada memória. Porém, quanto mais se comprime, menos espaço se ocupa e com isso menor é a qualidade da imagem final, o que em determinados contextos é uma característica que não se pode imolar, visto a importância dos detalhes e a necessidade de preservação de determinadas características. Este formato também não é o mais adequado, quando a imagem original sofre vários processos de edição, o que na área de processamento de imagem é uma constante levando à perda sucessiva da qualidade de imagem ao longo dos vários processos. Evita-se esta situação pela conversão desta imagem num formato digital PNG (*Portable Network Graphics*) que é um formato de dados empregue em imagens digitais que preserva a qualidade da imagem original independentemente dos processos aplicados nela. Contudo, a manutenção desta qualidade leva a serem imagem digitais computacionalmente mais pesadas do que um formato JPG, e por isso apenas é utilizado em situações onde realmente é necessário a sua aplicação.

3.1.2 Pré-processamento

Escolhido o formato para processar as imagens adquiridas, inicia-se o pré-processamento de imagens com o objetivo melhorar a qualidade de imagem, de modo a assegurar melhores condições gráficas, para a extração futura de características e tomadas de decisões mais corretas, por parte das funções de mais alto nível. Tipicamente promove-se nesta etapa a aplicação de filtros para remover ruídos provenientes da aquisição de imagem e funções para aumentar o contraste, para assegurar que as informações relevantes são detetadas. As imagens ainda podem sofrer processos de redimensionamento, para assegurarem um determinado formato. Nesta fase, os ruídos podem ter diversas origens, como um defeito no sensor de aquisição, iluminação inadequada, condições climáticas adversas ao processo de aquisição, etc. Porém, este não tem apenas origem no ato de aquisição da imagem, mas também pode ser considerado ruído elementos na imagem que podem atrapalhar a interpretação de resultados. Para prevenir interpretações dúbias, os filtros são ferramentas básicas para a remoção de ruídos na imagem, porém, apenas o ruído proveniente do processo de aquisição. Já o ruído originado por elementos na imagem pode ser eliminado pela Segmentação, processo descrito na seção 3.1.3. Os filtros subdividem-se em duas categorias: os filtros de domínio espacial (filtros que atuam diretamente na imagem) e os filtros de domínio da frequência, onde a imagem inicialmente é transformada para o domínio de frequência utilizando as transformadas de Fourier (transformadas discretas) [20].

Seguindo o raciocínio, a filtragem no domínio da frequência não atua na própria imagem. Ao invés disso, converte-se a imagem para o domínio das frequências como se mencionou e utiliza-se as transformadas de Fourier, equação 3.2, mais em concreto as Transformadas Discretas de Fourier (DFT), equação 3.3, para analisar a imagem e proceder aos métodos de filtragem necessários. O motivo da utilização das DFTs, ao invés das expressões genéricas de Fourier fundamenta-se na eficiência e rapidez do processamento segundo esta metodologia, uma vez ser um processo naturalmente com menor esforço computacional, em comparação com o seu cálculo teórico.

$$\text{Contínuo: } F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-2\pi j(ux+vy)} \quad (3.2)$$

$$\text{Discreto: } F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-2\pi j\left(\frac{ux}{M} + \frac{vy}{N}\right)} \quad (3.3)$$

Deste modo, ao considerar-se um dado sinal sinusoidal $y(t) = A \sin(2\pi ft)$, onde f é a frequência do sinal e A a sua amplitude, se o sinal amostrado for discreto, o domínio da frequência repete-se de forma periódica. Logo se uma imagem é considerada um sinal que é amostrado em duas direções, então realizando as transformadas de Fourier nas direções x e y , obtém-se a representação em frequência da imagem. Se a amplitude varia rápido em pouco tempo, significa que é um sinal de alta frequência, e um exemplo disso é o ruído e os limites da imagem. Porém, se variar lentamente, então é um sinal de baixa frequência. Foi explicado porque se utiliza as DFTs em relação às transformadas de Fourier, porém, apesar de ser um algoritmo eficiente, comparativamente com os métodos de domínio espacial é um algoritmo lento, necessitando de um maior esforço computacional [21].



Figura 3.5: Exemplo de conversão para o domínio das frequências [22]

Já a filtragem no domínio espacial realiza a manipulação direta dos valores dos pixels da imagem, ao longo da sua vizinhança, a partir de uma máscara que realiza todas as operações de filtragem necessárias. É uma técnica computacionalmente mais eficiente que requer um menor esforço computacional, para uma normal execução. Difere-se ainda em relação ao domínio da ação, ou seja, se são filtros de domínio espacial pontuais (ponto-a-ponto) ou locais. Nas operações pontuais, cada pixel da imagem de saída depende apenas do mesmo correspondente na imagem de entrada. Portanto, qualquer operação pontual pode ser compreendida como um mapeamento de pixel na imagem de entrada, para a imagem de saída. Já as operações locais, transformam o pixel da imagem de saída, como função não só do pixel correspondente na imagem de entrada, mas também aos pixels vizinhos que formam a vizinhança desse pixel. Os pixels mais próximos possuem um maior contributo, enquanto os pixels mais afastados possuem uma menor influência na aplicação desta filtragem [19].



Figura 3.6: Exemplo de remoção de ruído com um filtro passa-baixo [22]

3.1.3 Segmentação

A segmentação faz parte do processo de redução de dimensionalidade, no qual um conjunto inicial de dados brutos ou dados provenientes de um pré-processamento são divididos e reduzidos a grupos mais gerenciáveis sem a perda de nenhuma informação revelante [23]. A característica mais importante dos dados iniciais é possuírem um grande número de variáveis, porém, muitas destas características são redundantes, o que leva às técnicas de processamento de imagem processarem a mesma informação diversas vezes. Para evitar esta baixa eficiência computacional, a segmentação promove a partição de uma imagem em regiões, ou objetos distintos, de modo que estas qualidades consigam descrever o conjunto de dados iniciais com a precisão do original, onde o esforço computacional é muito mais reduzido, permitindo uma melhor eficiência [24]. Dentro desta temática, os processos de segmentação podem ser classificados como: automáticos ou semiautomáticos, sendo diferenciados pela existência ou não de intervenção humana. Nos processos automáticos todas as etapas são realizadas sem a ação humana, de modo a ser um processo totalmente automatizado, aplicando-se em áreas como a robótica. Os processos semiautomáticos requerem a intervenção humana com o intuito de auxiliar na escolha das ferramentas e técnicas que melhor se adequam à necessidade do processo. Normalmente este processo é utilizado em contextos, onde as condições de aquisição são variáveis, mas também em processos que buscam aperfeiçoar técnicas que posteriormente se pretendem implementar de forma automatizada.

Além dos processos tradicionais de segmentação, existem também métodos de segmentação através de redes neurais, que permitem modelar conhecimentos humanos usados nos processos semiautomáticos sem que seja necessário um imenso banco de dados. Este tipo de segmentação provém do processamento de pequenas áreas da imagem adquirida, utilizando um conjunto de redes neurais, onde após este processamento, o mecanismo de tomada de decisão distingue as áreas relevantes de acordo com a categoria identificada pela rede neuronal. As PCNNs (*Pulse-Coupled Neural Networks*) são as redes neuronais mais genéricas utilizadas na segmentação de imagens na maioria das aplicações atuais, de modo que cada neurónio faz-se corresponder a uma pixel, recebendo as informações da sua tonalidade como um estímulo externo [25].



Figura 3.7: Exemplo de segmentação [22]

3.1.4 Extração de Características

Em processos de análise de imagens, faz-se necessária a extração de características e informações das imagens de entrada de forma automatizada. O processo da extração de características visa a extração de atributos das imagens como forma de determinar/calcular as características que serão decisivas, para a tomada de decisões das funções de alto nível subsequentes. Normalmente, o *input* destas funções são imagens segmentadas provenientes da Segmentação, que agora ao sofrerem processos de extração de características, transformam-se em atributos. Deixa de ser necessário lidar com a imagem de forma direta, passando-se a tratar a imagem inicial com base nas suas características (cor, área, tamanho, quantidade, etc) [26].



Figura 3.8: Exemplo de detecção de arestas [22]

3.1.5 Detecção

Reconhecer significa conhecer de novo, e isto implica que em qualquer processo exista conhecimento prévio e algum tipo de armazenamento sobre o objeto a ser reconhecido. Sem o conhecimento prévio, ou memória, nunca se reconhece e apenas se conhece a cada iteração. Ora, esta é a parte onde os sistemas de visão possuem uma similaridade com a área da inteligência artificial, uma vez que esta procura identificar de forma autónoma comportamentos e padrões de forma análoga ao ser humano. Para realizar o reconhecimento, um sistema de visão computacional necessita de uma base de conhecimentos dos objetos a serem reconhecidos. Esta base de conhecimento pode ser implementada diretamente no algoritmo, através de um sistema baseado em regras ou então, esta base de conhecimento pode ser aprendida a partir de um conjunto de amostras dos objetos a serem reconhecidos utilizando técnicas do tipo *deep-learning* [20], na qual a cada geração de dados avaliados, o próprio sistema aprende com os seus erros em reconhecer/identificar padrões. Quanto mais situações forem analisadas e considerado as normais variações do processo, estes sistemas transformam-se em sistemas muito robustos à prova de erros, melhorando a cada nova iteração, uma vez que ao existir uma aprendizagem do processo, o sistema fica dotado de uma capacidade autónoma em detetar as características relevantes sem a intervenção do ser humano. Portanto, o reconhecimento de objetos é uma das principais funções da área da visão computacional, estando diretamente relacionada com o reconhecimento de padrões. Repare-se que um objeto pode ser definido por mais de um padrão (cor, dimensões, textura, forma, etc), de forma que o reconhecimento individual de cada um desses padrões possa facilitar no reconhecimento do objeto como um todo, a partir somente de uma das suas características. As técnicas de reconhecimento de padrões podem ser divididas em dois grandes grupos: técnicas estruturais, onde os padrões são descritos de forma simbólica e a estrutura é a forma como estes padrões se relacionam e as técnicas de *pattern matching* que utilizam uma teoria de decisão. Nestes grupos os padrões são descritos por propriedades quantitativas e deve-se decidir se o objeto possui ou não estas propriedades [27].

As técnicas estruturais consideram somente a relação entre as várias características do objeto em análise, para a sua classificação. O reconhecimento completo de um modelo pode exigir muito tempo de cálculo e consequentemente um maior esforço computacional, o que leva a um elevado tempo de reposta. Em sistemas com maior necessidade de resposta, face à elevada cadência torna-se impraticável. Portanto, como solução, a implementação desta técnica consegue assegurar uma resposta mais rápida, uma vez que ao analisar-se um número pequeno de características, num determinado contexto, culmina-se num leque restrito de hipóteses [28]. Por exemplo, um AGV necessita de virar num cruzamento, e existe um sinal no solo que indica essa viragem. Tratando-se de um AGV com processamento de imagem, este só necessita de perceber que área, cor e em que lado se encontra o sinal, para perceber a situação em que se encontra e entender a direção para onde vai virar. Note-se que não se identifica o sinal, mas sim as suas características naquele contexto. A maioria dos sistemas de visão computacional na robótica utiliza esta abordagem para o reconhecimento de objetos bidimensionais, de modo que os algoritmos de reconhecimento tenham uma deteção responsiva.

Já as técnicas de *pattern matching* procedem à identificação de objetos com base em categorias predeterminadas. O problema desta técnica é comparar o objeto em avaliação com uma configuração de características armazenadas, definida num modelo padrão, obtido durante o procedimento de treino no qual o sistema de visão é programado para conhecer objetos somente com determinadas características inseridas na sua base de dados. Por isso, é uma técnica apenas aplicável se não existir uma elevada variedade de modelos com características físicas significativamente diferentes dos atributos definidos no modelo padrão [29]. O procedimento é baseado no uso de um número ótimo de características, para minimizar a frequência de erros no processo de classificação e agilizar o processo, de forma que as características analisadas são comparadas com os valores correspondentes do modelo padrão. Quando se encontra uma combinação com uma percentagem de validação acima do limite estipulado, permitindo certas variações estatísticas no processo de comparação, visto nunca existir uma réplica perfeita do modelo aferidor, então esse objeto é classificado como sendo um objeto de uma determinada categoria [28].



Figura 3.9: Exemplo de detecção da face humana [22]

3.1.6 Processamento de Alto Nível

Nesta etapa, a entrada de informação é tipicamente um pequeno conjunto de dados. Já não se analisa a imagem obtida, nem segmentos dela, mas sim apenas as qualidades de interesse selecionadas. Isto promove a construção e implementação de funções que realizam tarefas de cognição associados à visão humana, onde se aplicam as técnicas de controlo baseadas em algoritmos de inteligência artificial. Enquanto os algoritmos de baixo e médio nível abrangem um espectro razoavelmente bem definidos de atividades, o processamento de alto nível é considerado mais vago e especulativo [30].



Figura 3.10: Exemplo de identificação do rosto de um ser humano [22]

3.2 OpenCV

O OpenCV (*Open Source Computer Vision Library*) é uma biblioteca *open source* de software de visão computacional e *machine learning*. O OpenCV foi desenvolvido com o objetivo de formar uma infraestrutura comum para aplicações de visão computacional e acelerar a aplicação destes algoritmos nos produtos comerciais. Sendo um produto com uma licença BSD (*Berkeley Source Distribution*), torna-o num software gratuito e habilitado a diversas modificações pelos utilizadores [31].

É uma biblioteca que possui mais de 2500 algoritmos otimizados, que englobam um conjunto abrangente de algoritmos de visão artificial e algoritmos de *machine learning*. Destacam-se os módulos de processamento de imagem e vídeo I/O, estrutura de dados, álgebra linear e GUI (*Graphical User Interface*). Esta biblioteca é amplamente utilizada pela comunidade, mas também por empresas (Google, Microsoft, Sony, Intel, Toyota, etc), grupos de pesquisas e órgãos governamentais. Os usos do OpenCV no quotidiano abrangem desde a captura de imagens e vídeos em ruas, deteção de intrusos em sistemas de vigilância em Israel, monitoração de equipamentos militares na República Popular da China, auxílio na robótica e domótica, na deteção de acidentes de afogamento em piscinas europeias, verificação de detritos em pistas de corrida automobilística na Turquia, na inspeção de rótulos de produtos em todo o mundo, reconhecimento facial realizado no Japão, entre muitas mais [32]

A interface nativa é em C++, porém conta com interfaces em Python, Java, MATLAB, suportando sistemas operativos como: Windows, Linux, Android e Mac OS. Como exemplo, de seguida encontra-se o exemplo de um *script*, que inicia a câmara integrada de um *laptop*. [33].

```
1 import cv2 #importacao do OpenCV
2 camera = cv2.VideoCapture(0) # selecao da camara a utilizar
3
4 if __name__ == '__main__': # quando se corre somente este ficheiro ,
5     while True: # entra-se em loop infinito
6         ret, frame = camera.read() # leitura da camara digital
7         if ret is True: # se a leitura retornar TRUE, entao:
8             roi = frame[250:300, 0:640] # definicao de uma regio de interesse a partir da
                variavel "frame"
9
10            cv2.imshow('frame', frame) # visualizacao grafica da imagem capturada pela
                camara digital
11            if cv2.waitKey(1) & 0xFF == ord('s'): # pressionar 's' para sair (condicao de
                saida)
12                break
13            else: # quebra do ciclo
14                print('Camara nao detectada')
15                break
16
17 camera.release() # libertacao de recursos graficos
18 cv2.destroyAllWindows() # comando para fechar corretamente as interfaces do Raspberry Pi
```

Algoritmo 3.1: Exemplo da ligação de uma câmara digital via OpenCV

Capítulo 4

Desenvolvimento do AGV

Pretende-se desenvolver um AGV de raiz, com o intuito de seguir um sistema de guias, que perfaz um percurso operacional. Este sistema é um sistema unidirecional, com entroncamentos, cruzamentos e mudanças de direção, que serão realizados de forma automatizada através da leitura dos sinais integrados no piso interpretados por visão artificial. O método de navegação do AGV assenta num sistema fechado, uma vez que existe um layout fixo no piso. Em termos de orientação, o AGV apresenta um sistema de deslocação diferencial e integra na sua estrutura uma câmara digital, que permite realizar a aquisição de imagem sendo o único método de aquisição do ambiente ao seu redor. Deste modo, a informação obtida é processada por um sistema dedicado, que através do OpenCV realiza o processamento de imagem e realiza as decisões necessárias, para uma navegação satisfatória. O processo de seleção de guias é automatizado pelo AGV, mas previamente é pré-definido pelo utilizador as rotas que este percorrerá, de modo que ao fim de um número pré-definido de percursos programados, o AGV repeta novamente esses percursos.

4.1 Hardware Mecânico

O projeto mecânico visa responder às necessidades de concessão de uma estrutura física capaz de oferecer uma reposta necessária, face aos requisitos empregues pelo sistema integrado. Esta estrutura deve ter a capacidade de alojar todo o hardware (mecânico e elétrico), de forma acessível e modular, onde a localização dos elementos se encontre estrategicamente distribuída por grupos, de modo a facilitar o processo de manutenção e possíveis métodos de deteção e resolução de problemas. Além disto, o sistema de locomoção deverá ter uma reposta flexível e eficiente, perante o controlo imposto pela unidade controladora. Todo o hardware referente à estrutura física do AGV será modelado e construído por uma tecnologia de fabricação aditiva - Impressão 3D.

4.1.1 Modelação CAD

Atualmente, na engenharia moderna, não existe a realização de projetos sem a utilização de ferramentas CAD (*Computer Aided Design*). Independentemente da importância, ou dimensões do projeto, um modelo CAD oferece um leque de vantagens incomparáveis, face à não realização de nenhum modelo. Por isso, a habilidade de criar modelos e conseguir simular, transforma-se numa habilidade necessária aos níveis da engenharia atual. Portanto, CAD é o nome padrão para utilização de ferramentas computacionais (software) que auxiliam na criação, otimização e simulação de um projeto, com base nas regras de desenho técnico. Com a sua utilização, o projetista consegue facilmente gerar diversas perspetivas de forma a facilitar o entendimento do elemento modelado. Para além disso, é possível gerar um banco de dados, tanto com elementos *standard* (anilhas, parafusos, roscas, rolamentos, etc), como com elementos modelados, que inclusive possuem versões e hierarquias, o que permite um melhor controlo e gestão do projeto. Normalmente, a sistemas CAD são associados sistemas CAM (*Computer Aided Manufacturing*) [34]. Estas ferramentas são responsáveis pelo controlo do processo de fabricação de peças, via software. A fácil integração de sistemas CAD deve-se ao

cumprimento de standards e boas práticas regulamentadas por normas (ISO), que visam melhorar a qualidade dos produtos e processos modelados, através da aplicação de práticas consistentes, que mantenham um estilo comum. Com isto, existe um aumento na eficiência no processo de modelação, uma vez que se for necessário utilizar um componente numa biblioteca sobre estas normas, basta integrá-lo, não sendo necessário estar a modelá-lo de novo. A produtividade nestas ferramentas é evidente, mas também com a redução de custos diretos, no processo de fabricação, uma vez que ao conseguir-se simular o projeto via computacional, só quando este satisfizer as condições impostas é que se inicia o processo de construção [35].



Figura 4.1: Benefícios das ferramentas CAD

Neste trabalho utilizou-se o software CAD da Autodesk, o Fusion 360, que permite a integração de dados 2D e 3D em um único ambiente, criando uma representação visual do produto modelado, com a opção da utilização de ferramentas paramétricas de edição livre. A estratégia utilizada durante o processo de modelação foi a escolha inicial de uma base (plataforma) que suportasse todos os problemas em termos de esforços mecânicos que o AGV sofresse, de modo a conseguir-se um design compacto, sem prejudicar a acessibilidade dos componentes. Assim, como apresentado na figura 4.2 a base é dividida em duas partes: (1) e (2). A parte 1 é onde estão embutidas as unidades controladoras, neste caso a unidade de processamento de imagem e a interface que realiza a gestão dos motores elétricos. Já na parte 2 estão todos os sensores e atuadores deste sistema. Como único sensor de aquisição, a câmara digital encontra-se integrada em (6), juntamente com um sistema de iluminação em (5), de modo a proporcionar condições homogêneas de aquisição, bem como um bom destacamento da guia no piso, não dependendo assim da luminosidade do meio envolvente. Em relação aos atuadores, os motores elétricos, um encontra-se encastrado em (7), enquanto que o outro encontra-se nas mesmas condições mecânicas, só que no lado oposto, não sendo visível na figura 4.2, estando ambos acoplados por ligações aparafusadas (2A) à parte 2.

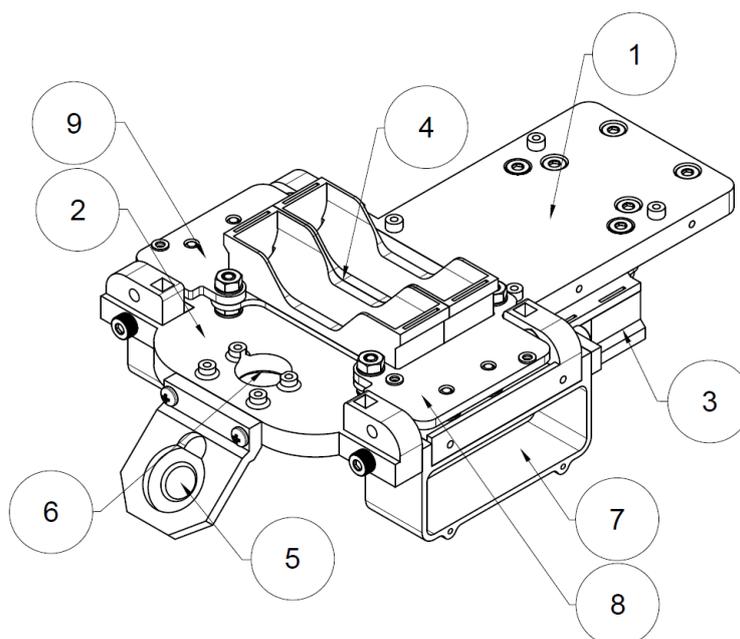


Figura 4.2: Estrutura física do AGV modelado via Fusion 360

Estas ligações aparafusadas (**2A**) realizam exclusivamente a fixação da carcaça (**7**) e do seu análogo que integram os motores elétricos, à parte (**2**), enquanto que as ligações aparafusadas (**1A**), que para além de partilharem esta mesma funcionalidade, também são responsáveis pela fixação mecânica entre as partes 1 e 2, resultando numa base estruturalmente e espacialmente apta, para responder às características desejadas. De forma auxiliar e complementar, como forma de ligação mecânica das partes 1 e 2, existe um sistema de alimentação elétrico, repartido por ambas as partes em (**3**) e (**4**), gerido por uma BMS. Por uma questão de design foi necessária esta partição, visto o sistema como um só não se enquadrar fisicamente na estrutura do AGV. Como solução realizou-se esta partição, de modo a não sacrificar as características elétricas detalhadas na seção 4.2.1. Como auxiliar a este sistema, existem dois elementos intermediários em (**8**) e (**9**), responsáveis por regular as características elétricas das unidades controladoras incorporadas na parte 1. Do ponto de vista estrutural, a arquitetura do sistema de alimentação permitiu integrar um segundo conjunto de ligações aparafusadas (**1B**) e (**2B**), que auxiliam estruturalmente na fixação da base do AGV resultando numa estrutura mais robusta. O resumo das principais ligações mecânicas encontra-se detalhado na figura 4.3.

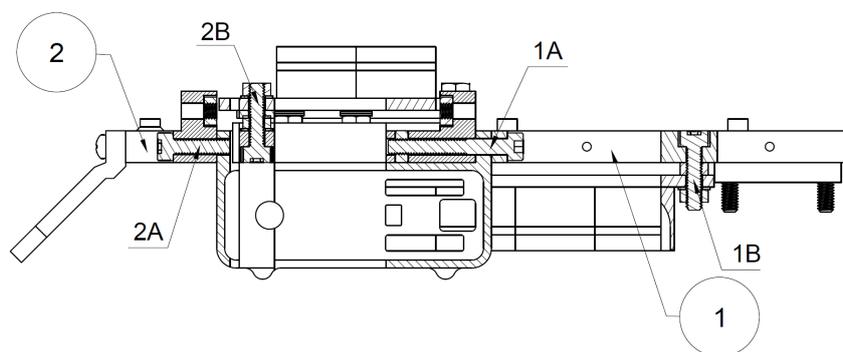


Figura 4.3: Corte transversal da estrutura do AGV a destacar as principais ligações aparafusadas

Esta constituição física resulta assim numa estrutura funcional e segura, para aguentar todos os esforços mecânicos a que está sujeita. Como forma de aproximar o modelo 3D, face à construção realística deste, foi necessário englobar os restantes elementos e modelar elementos acessórios, como rodas, motores, unidade processadora, etc, de forma a tornar o modelo, o mais fidedigno ao modelo real. Deste modo, o modelo completo do AGV, via a modelação computacional descreve-se na figura 4.4, estando no anexo A mais informações técnicas sobre este modelo.

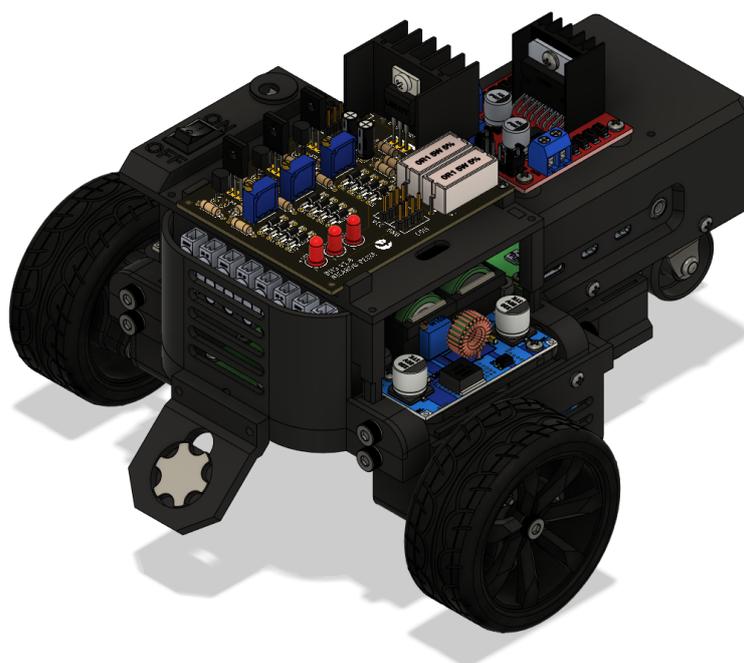


Figura 4.4: Modelo CAD do AGV

4.1.2 Técnica Aditiva de Fabrico

A conceção de um componente mecânico pode ser conseguido através de diversos processos de fabrico. A decisão de qual processo utilizar depende da aplicação a que se destina, das tolerâncias dimensionais e geométricas necessárias, custos, etc. Dentro do leque de opções de fabrico atuais, muitos destes métodos não permitem incorporar diretamente modelos CAD, para a construção de componentes, o que leva a que seja necessário aplicar aproximações ou adaptações, visto ser geometricamente difícil de alcançar o resultado concebido via CAD, em componentes de maior complexidade. Os métodos que conseguem incluir modelos CAD, muitos deles são financeiramente dispendiosos, o que é uma limitação a este nível. Porém, uma solução que vêm a ganhar popularidade nestes últimos anos são as técnicas aditivas de fabrico, vulgarmente conhecidas por impressão 3D [36]. Esta é uma tecnologia onde a partir de um modelo CAD se constrói um objeto tridimensional pela deposição de um material base.

Os componentes que integram o AGV foram impressos por esta tecnologia, mais em concreto pelo processo de modelação por deposição de material fundindo – FDM (*Fused deposition Modeling*). Neste método, um filamento de material termoplástico (1) é forçado a entrar numa cabeça de extrusão (2), previamente aquecida a uma temperatura que permita fundir este material, no bloco de aquecimento (3). Ao se encontrar fundido este é conduzido a um bico de extrusão (4), que o irá depositar numa mesa de extrusão (5), que se movimenta segundo as direções x e y , de forma a conseguir mapear a primeira camada. Quando esta camada se encontra completamente mapeada, existe um movimento segundo a direção z , que quando cessa, se inicia um novo ciclo de mapeamento segundo as direções x e y . Este processo repete-se até que o componente fique completo, obtendo-se um componente com um formato volumétrico (6) [37].

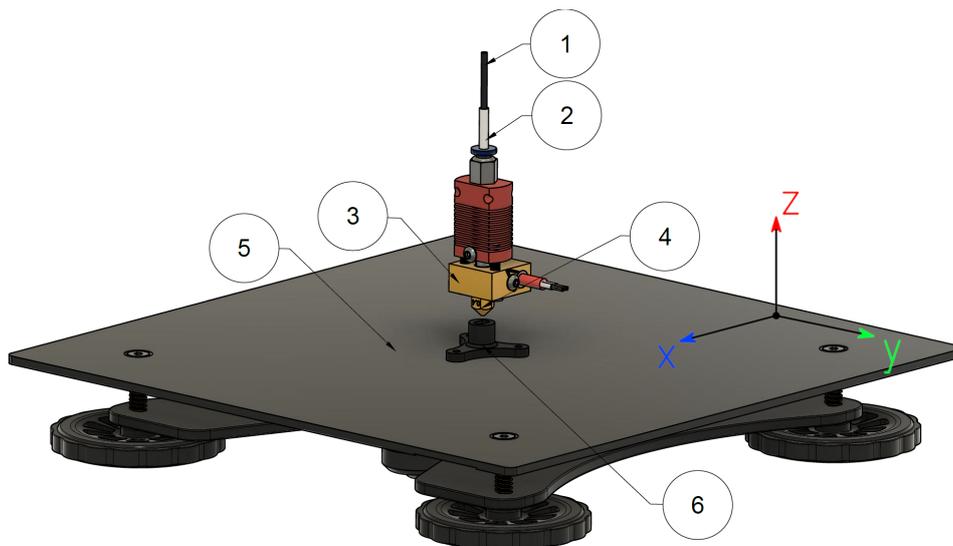


Figura 4.5: Modelação por deposição através de termoplásticos

Os termoplásticos são um polímero plástico, que se torna maleável ou moldável, a partir de uma temperatura crítica. Quando se atinge esta temperatura, as cadeias de polímeros interligadas por forças moleculares, enfraquecem resultando num líquido viscoso. Neste estado, os termoplásticos podem ser completamente remodelados e utilizados para a impressão de componentes, devido à alteração drástica nas suas propriedades físicas sem mudança de fase associada. Dentro da variedade de polímeros existentes utilizou-se como material de construção em todos os componentes do AGV ácido polilático (PLA) sobre a forma de filamento. Este é um polímero constituído por moléculas de ácido láctico (ácido com origem biológica) que advém de recursos renováveis, como amido de milho, raízes, canas-de-açúcar, etc [38]. A grande vantagem deste material, excluindo as suas características mecânicas como um elevado módulo de rigidez e módulo de elasticidade, é a sua fácil reciclagem, visto ser um material biodegradável, torando-o num material sustentável, sendo certificado em condições industriais, como um material que se decompõe segundo a norma ASTM D6400 [39].

O filamento utilizado apresenta as características mecânicas destacadas na tabela 4.1:

Filamento	Propriedades Mecânicas	PLA
	Densidade (ρ)	$1.25 \pm 0.05 \text{ g/cm}^3$
	Resistência à tração (σ)	$\geq 60 \text{ MPa}$
	Módulo de Young (E)	2 GPa
	Diâmetro (θ)	$1.75 \pm 0.02 \text{ mm}$
	Comprimento (L)	330 m
	Temperatura de Adesão (T_a)	50 - 65 °C
	Temperatura de Fusão (T_f)	190 - 220 °C

Tabela 4.1: Propriedades mecânicas do PLA adaptado de [40]

O processo de construção dos componentes inicia-se com a obtenção do modelo CAD, porém este não é diretamente utilizado pela máquina de impressão, visto a linguagem que codifica o modelo CAD não ser entendido por esta. Assim, é necessário converter o modelo CAD noutra linguagem, sendo que o mais comum e utilizado em processos aditivos é o formato STL (*stereolithography*). Os modelos segundo este formato descrevem a superfície geométrica de um objeto tridimensional, sem representação da sua textura, cor, escalas ou outros atributos associados durante o processo de modelação. Ao invés disso, realiza-se uma aproximação ao modelo sólido, através de conjuntos de triângulos não estruturados, formando uma rede (*mesh*). Este processo é designado por tecelagem, onde a partir de unidades básicas, como triângulos se forma uma superfície sem espaços, igual ao espaço particionado. Quanto mais pequenos os triângulos forem, melhor é a aproximação face ao modelo CAD original.

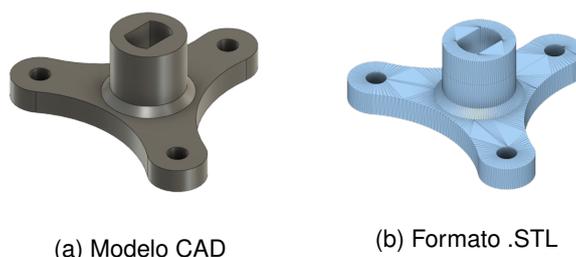


Figura 4.6: Conversão do modelo CAD para o formato .STL

Com a conversão do ficheiro CAD em STL, a partir do Fusion 360, agora é necessário passar este ficheiro por um software *slicer*, de modo a se configurar todas as propriedades mecânicas que se pretende impor no processo de fabrico. Neste caso, utilizou-se o Creality Slicer, um software CAM *open source* da Cura, que permite ajustar diversos parâmetros relacionados com o processo descrito. Para se impor os parâmetros relacionados a este processo é necessário converter o ficheiro STL, num outro formato que permita codificar esta informação, uma vez que o ficheiro STL apenas codifica o modelo CAD aproximado a uma *mesh* de triângulos. Assim, é necessário converter o ficheiro STL em um ficheiro G-code. O G-code é uma linguagem de programação de controlo numérico computacional, utilizado vastamente em softwares CAM, que permite a configuração das características das máquinas através dos seus códigos, codificados na mesma linguagem, que estas interpretam. Assim, um ficheiro G-code contém todas as informações que se pretende impor no processo de fabrico, segundo os parâmetros dimensionados pelo utilizador.

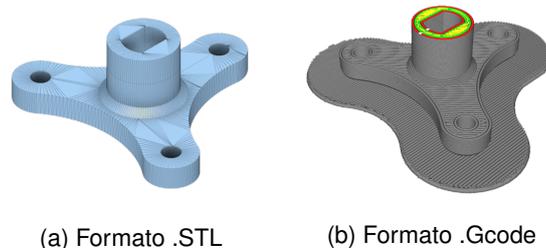


Figura 4.7: Conversão do formato .STL para o formato .GCODE

Assim, após se inserir o ficheiro STL no software *licer*, define-se todas as características relacionadas ao processo de fabrico, desde as propriedades mecânicas do PLA, até à orientação do componente, com a utilização de suportes ou não [37]. Após a definição destas características, o ficheiro STL é convertido num ficheiro G-code, onde a sua *mesh* é transformada num conjunto de camadas de secções transversais, através da interceção de sucessivos planos xy , em relação ao modelo integrado. Estas divisões em camadas transversais são diretamente proporcionais às características inseridas no software utilizado. Com isto, consegue-se um ficheiro totalmente caracterizado em linguagem máquina. Todos os componentes do AGV foram impressos através de uma Impressora Ender 5, seguindo toda esta lógica para a conceção dos vários componentes. Na figura abaixo encontra-se um exemplo deste processo.



Figura 4.8: Componente obtido pela modelação por deposição de PLA

4.1.3 Dimensionamento dos Motores

O AGV é um tipo de robô e, na robótica, o tipo de motores utilizados para os sistemas de deslocação são na grande maioria motores elétricos. O motor elétrico é uma máquina elétrica, que converte energia elétrica em energia mecânica, com uma grande eficiência, sem a emissão de poluentes. A grande vantagem deste tipo de motor é usufruir das vantagens da energia elétrica, ou seja, facilidade no processo de transporte e comando de energia, mas também possuir uma construção mecânica simples com um alto rendimento, com pouca necessidade de manutenção. Existe a possibilidade de se utilizar motores híbridos, ou motores de combustão interna, portanto com a utilização de diesel, como fonte de alimentação [41]. Porém, estas soluções não serão utilizadas, pois são mais dispendiosas, mais complexas de se incorporar e segundo a seção 2.6, não se irá promover a construção de novas tecnologias com base nestas soluções, uma vez que o futuro caminha para um processo de descarbonização e por isso, a sua utilização será cada vez mais diminuta.

Deste modo, antes de qualquer escolha em relação ao tipo de motor elétrico a utilizar no AGV, é necessário analisar a pior situação a que este estará sujeito, através de uma análise das condições de funcionamento e posteriormente uma análise mecânica, em relação aos esforços a que este está sujeito. Pretende-se desenvolver um AGV *indoor*, que na maioria dos casos circula em pisos planos com alguma rugosidade. Porém, podem surgir planos com alguma inclinação. Portanto, o pior cenário em termos de esforço mecânico para o AGV, é superar a inércia inerente a ele em planos inclinados. Mais em específico, em planos inclinados, mas na fase de subida, pois caso seja o inverso, então até se torna numa situação menos severa, em termos de esforços mecânicos, comparativamente a um plano horizontal. Estas situações são pontuais, mas acabarão por acontecer e ao se conseguir dimensionar para a pior situação, então nas restantes situações não haverá problemas. Assim, analisa-se o diagrama do corpo livre do AGV, no ato de subida de um plano inclinado:

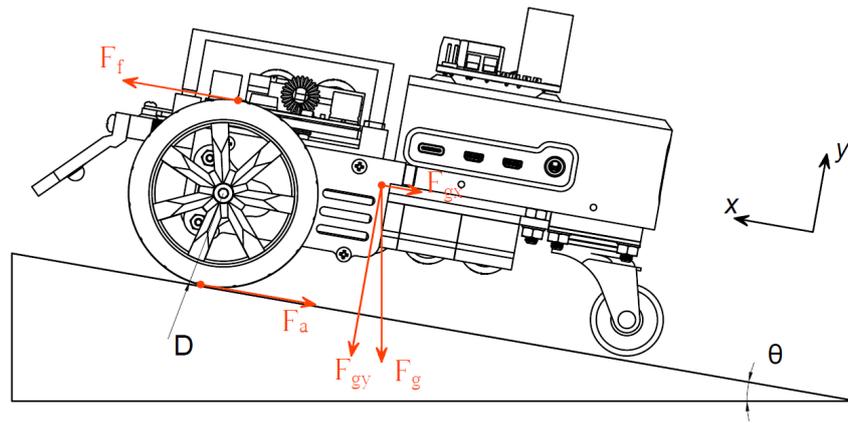


Figura 4.9: Diagrama do corpo livre do AGV

Nesta situação, o AGV apresenta essencialmente três forças que influenciam diretamente o seu movimento: F_f , F_a e F_g . A força exercida pelas rodas motrizes, ou seja, pelas rodas que proporcionam tração é representado pela força F_f . Esta força advém do trabalho exercido pelos motores elétricos, sendo que o sentido desta força é igual ao sentido do movimento do AGV, sendo a única força que promove o movimento deste. Em oposição, sempre que existe um movimento horizontal nestas situações, forma-se uma força de atrito, que é diretamente proporcional à rugosidade da superfície do piso, onde o AGV se desloca. Esta força representa-se fisicamente por F_a , sendo uma força sempre contrária ao movimento do AGV. Note-se que a roda traseira é uma roda giratória, que não possui qualquer tipo de mecanismo para a produção de trabalho, para promover a deslocação do AGV. Para além disso, o atrito aí gerado é muito inferior ao que existe nas rodas motrizes e portanto, a força de atrito que se forma nesta roda de apoio será desprezado nesta análise.

A última força a atuar no AGV é a força gravítica F_g . Porém, neste caso, o plano ao estar inclinado com um grau (θ), implica que esta força vertical se divida em duas componentes: componente normal F_{gy} e componente horizontal F_{gx} . A componente normal é responsável por exercer uma força no piso, que empurra o AGV contra este, de modo a contribuir diretamente, para a magnitude da força de atrito. Porém, ao não ser uma força segundo a direção (x), não poderá ser contabilizada nesta análise. A componente horizontal F_{gx} já apresenta a mesma direção das forças mencionadas, influenciado na movimentação do AGV, segundo a direção x . É uma força diretamente proporcional à inclinação do plano, mas neste caso, como se trata de uma subida, então esta será uma força contrária ao movimento do AGV.

Estabelecido os principais esforços mecânicos que atuam no AGV, incluindo os seus sentidos é necessário agora impor as condições iniciais deste dimensionamento. Pretende-se que o AGV inicialmente se encontre em repouso, e em seguida acelere até atingir uma velocidade constante. Assim, nestas condições é possível a aplicação da lei fundamental da dinâmica, a segunda lei de Newton. Segundo esta, a força provocada por um corpo é igual à taxa de variação temporal do momento linear em um sistema inercial. Como neste sistema, a massa do AGV (m) é constante ao longo do tempo, simplifica-se de modo que a resultante das forças aplicadas neste corpo, produz uma aceleração (a) diretamente proporcional [42]. Deste modo, o somatório das várias forças intervenientes na movimentação do AGV, segundo a direção x , manifestam-se de modo que

$$\sum F_x = ma \quad \Leftrightarrow \quad F_f - F_{gx} - F_a = ma. \quad (4.1)$$

Que ao se reorganizar consegue-se que a força (F_f) seja definida por

$$F_f = ma + F_a + F_{gx}. \quad (4.2)$$

A força mínima (F_{min}) que os motores elétricos têm de produzir, para superar a inércia é obtida pela expressão 4.2. Porém, esta característica física não é habitualmente fornecida pelos fabricantes de motores elétricos. Portanto, o seu cálculo não tem valor para o dimensionamento, uma vez que os motores elétricos são catalogados segundo outras características, como binário de arranque, potência e velocidade. Assim, ao invés do cálculo da força (F_f), troca-se pelo cálculo do binário de arranque (M). O binário é uma grandeza vetorial associado à força produzida, para rotacionar um corpo em torno de um eixo. A distância entre o ponto de aplicação da força e o eixo, neste caso é o raio da roda motriz (r), que genericamente é designado por braço, de modo que $M = rF$. Portanto, ao se efetuar esta substituição consegue-se que

$$\frac{M}{r} = ma + F_a + F_{gx}. \quad (4.3)$$

Onde, a força de atrito (F_a) é proporcional ao coeficiente de atrito (μ), que é diretamente influenciado pela rugosidade do piso e pelo tipo de material dos pneus das rodas motrizes, mas também em relação à componente normal da força gravítica F_{gy} , que o AGV exerce sobre o piso. Neste caso, quanto menor for a inclinação do piso, maior será a magnitude da força de atrito, visto a sua relação trigonométrica com o cenário ilustrado na figura 4.9. Assim, a força de atrito (F_a) é definida por

$$F_a = \mu F_{gy} \quad \Leftrightarrow \quad F_a = \mu mg \cos \theta. \quad (4.4)$$

Contrariamente, a magnitude da componente horizontal da força gravítica F_{gx} diminui com a diminuição do ângulo de inclinação (θ), uma vez ser exclusivamente depende do ângulo de inclinação, que leva a formação desta componente horizontal da força gravítica, resultando em

$$F_{gx} = mg \sin \theta. \quad (4.5)$$

Numa situação plana, esta componente não existe, visto $F_g = F_{gy}$. Com as expressões (4.4) e (4.5) substitui-se na expressão (4.3), de forma a se obter uma expressão válida, para o cálculo do binário de arranque (M), dada por

$$\frac{M}{r} = ma + \mu F_g \cos \theta + mg \sin \theta \quad \Leftrightarrow \quad M = rmg \left(\mu \cos \theta + \sin \theta + \frac{a}{g} \right). \quad (4.6)$$

Deste modo, o cálculo do binário de arranque (M) já é possível. Porém, o dimensionamento de um motor elétrico não acontece apenas com esta característica mecânica. Como suplemento, o cálculo da potência e da velocidade de saída são fundamentais, para determinar um dimensionamento completo, a fim de promover um futuro bom funcionamento. Portanto, a velocidade de saída que o AGV necessita de ter, para superar as condições inicialmente impostas depende de dois fatores: diâmetro da roda de tração (D) e da velocidade linear, que se pretende que o AGV adote, durante o seu estado de locomoção (v_n). O diâmetro da roda apenas depende do perfil de roda escolhido, de forma que rodas com menores diâmetros proporcionam momentos mais pequenos, visto o raio ser menor, para uma força constante. Porém, a velocidade que uma roda possui é inversamente proporcional ao momento que consegue gerar, ou seja, quanto maior for o raio, menor será a velocidade, assumindo a aplicação de uma forma constante. Pretende-se assim, escolher um perfil que permita tanto gerar binário suficiente, como uma velocidade superior à necessária, de modo a satisfazer as características impostas. Com isto, a velocidade em RPM, unidade comumente expressa, é obtida por

$$N_T = \frac{60v_n}{\pi D}. \quad (4.7)$$

Caso se pretenda calcular a velocidade angular (ω), que por exemplo é útil para o cálculo da potência, esta é definida por

$$\omega = \frac{2\pi N_T}{60}. \quad (4.8)$$

Com a expressão para o cálculo do binário de arranque (M) e a expressão para o cálculo da velocidade angular (ω) consegue-se o cálculo da potência de cada motor (P), pois esta resulta do produto entre ambos: $P = M\omega$. Contudo, esta é a potência teórica (P_t) assumindo que não existem perdas mecânicas, ao longo do redutor. O redutor é um sistema puramente mecânico de engrenagens, acoplado à saída do motor elétrico com o intuito de reduzir a velocidade inicial do motor, a fim de aumentar o binário de saída. Se a velocidade é decrementada n vezes, então o binário de saída é incrementado n vezes. Na prática, esta conversão não acontece com uma alta eficiência, por isso, por uma questão de boas práticas aplica-se um rendimento (η), para este sistema de engrenagens, de modo a cobrir todas as perdas mecânicas aqui existentes. Assim, a potência realista que os motores do AGV necessitam (P_r), advém de

$$P_r = \frac{P_t}{\eta}. \quad (4.9)$$

Como fator adicional, o tipo de configuração motora implementada influencia diretamente o dimensionamento dos motores elétricos. Neste sistema, implementou-se um sistema de deslocação diferencial, de modo a garantir-se uma elevada capacidade de mobilidade, através de dois motores na parte dianteira do AGV. Cada motor (1) está associado a uma roda com rasto em borracha sintética (3), de modo a se conseguir uma boa adesão em relação ao piso, onde irá atuar. Independentemente do perfil de roda escolhido e do tipo de motor implementado, existe uma interface entre ambos (2), que para além de tornar a ligação entre motor-roda, uma ligação mecanicamente solidária, permite um total ajuste aos perfis de ambos, de modo a que esta ligação seja mecanicamente estável. Este conjunto é evidenciado na figura 4.10.

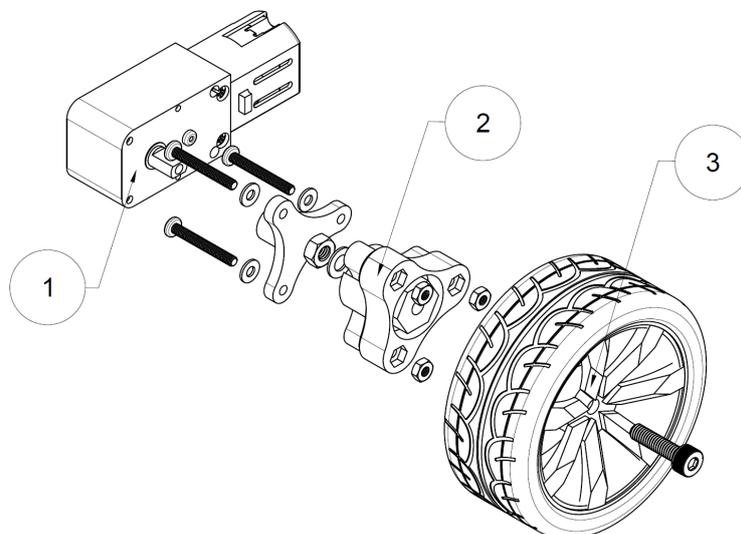


Figura 4.10: Conjunto motor-roda motriz

O facto de existirem dois motores elétricos significa que os esforços mecânicos serão divididos por ambos, de modo que cada motor elétrico apenas necessita de suportar metade dos esforços mecânicos totais, uma vez que são dois motores a exercer trabalho, com o mesmo princípio de funcionamento e sentido de rotação. Logo, tanto o binário, como a potência (caraterísticas mecânicas aqui destacadas) são metade do valor calculado, para cada motor e respetivo dimensionamento.

Para completar a arquitetura do sistema de deslocação é necessário garantir a estabilidade do sistema, visto só com duas rodas dianteiras esta estabilidade não é alcançável, uma vez que o centro de massa que o AGV possui não coincidir com a localização dos eixos dos motores elétricos, como é evidenciado na figura 4.9, segundo o plano cartesiano xy . Esta posição do centro de massa leva a que seja necessário incluir um apoio na parte traseira (4), a fim de atingir-se a estabilidade desejada. Este apoio mecânico é uma roda giratória (6). Esta roda, como já foi mencionado, não irá promover qualquer trabalho mecânico, para a deslocação do AGV, de facto, ao ser uma roda giratória, é formada por um rolamento axial de esferas simples (5), de modo a suportar o peso do AGV segundo a direção do seu eixo. Diferentes de outras rodas de apoio estáticas, que não possuem nenhum grau de liberdade, a não ser o movimento da roda sobre si, na roda giratória existe um grau de liberdade. Este grau de liberdade permite que a roda rotacione livremente em torno do seu eixo, através do rolamento mencionado. Isto permite, que em mudanças de direção, ou ajustes na rota do AGV, esta roda se ajuste à direção, que naquela instância promova uma menor resistência mecânica, ou seja, um menor atrito. Por este motivo, se desprezou o atrito aqui formado, visto a sua baixa significância na análise global, uma vez que o atrito formado permite otimizar a direção desta roda, que por sua vez ajuda no ato de mudança de direção, ao adquirir uma posição que mecanicamente é mais favorável ao movimento do AGV.

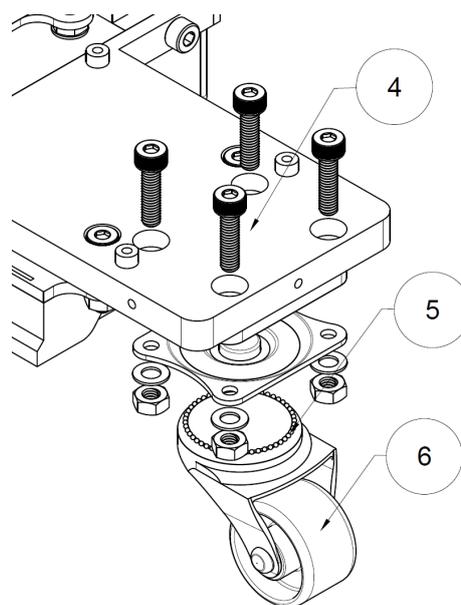


Figura 4.11: Conjunto parte traseira-roda giratória

Portanto, com a análise mecânica e definição da arquitetura do sistema de deslocação completas consegue-se nesta fase o cálculo das três propriedades mecânicas mencionadas, o que permite tirar ilações sobre o tipo de motor a implementar, bem como a magnitude das suas características. Por último, apenas é necessário escolher as propriedades físicas que o AGV será dotado. Neste caso, a massa do AGV com todo o hardware integrado (mecânico + elétrico), a velocidade linear que se pretende que ele se desloque, o ângulo que origina a pior situação num plano inclinado, o diâmetro das rodas de tração, o tempo que o AGV demora a atingir uma velocidade constante, partindo de uma posição de repouso, o rendimento do redutor, que advém do seu *datasheet* e por último o coeficiente de atrito entre o piso e as rodas motrizes. Com base nestas caraterísticas e nas expressões demonstradas até aqui, obtém-se a tabela 4.2 com o resumo de todas as informações processadas.

Caraterísticas	Motor Dimensionado	Motor Selecionado	Rácio
Massa (m) = 1.2 Kg	Binário (M) = 0,04579 Nm	Binário (M) = 0.104 Nm	2.3 ✓
Velocidade linear (v_n) = 0.2 m/s	Velocidade (N_T) = 59 RPM	Velocidade (N_T) = 110 RPM	1.9 ✓
Ângulo de inclinação (θ) = 10°	Potência (P_r) = 0,472 W	Potência (P) = 1.2 W	2.5 ✓
Diâmetro (D) = 6.5 cm			
Tempo de aceleração (t_a) = 0.4 s			
Rendimento (η) = 60%			
Coeficiente de atrito (μ) = 0.05			

Tabela 4.2: Dimensionamento das caraterísticas mecânicas do AGV

Como é evidenciado na tabela 4.2, o modelo selecionado supera os requisitos mecânicos dimensionados. Esta informação pode ser verificada quantitativamente através do rácio entre as propriedades do modelo selecionado, face às mesmas propriedades do modelo dimensionado. Caso exista um rácio inferior a um, então o motor selecionado não satisfaz as condições iniciais, logo este modelo é excluído. Caso o rácio seja superior a um, então o modelo selecionado supera as condições iniciais, porém não é desejável rácios próximos de um, pois significa que a característica mecânica comparada, é garantida com uma pequena margem. Por outro lado, rácios muito elevados também indicam um mau dimensionamento, visto se estar a utilizar um modelo muito acima, do que é necessário, não se tirando partido das capacidades deste modelo, podendo adicionar problemas, como um maior consumo energético e custo económico. Deste modo, deve-se escolher um modelo que possua rácios intermédios, de forma a otimizar todo o processo. Neste caso, o modelo selecionado, possui rácios superiores a dois em todos os parâmetros à exceção da velocidade (1.9), o que significa que consegue aproximadamente oferecer o dobro dos requisitos necessários. Em condições normais de funcionamento, os motores operam a metade da sua capacidade, o que proporciona um aumento da vida útil destes.

O modelo escolhido é um motor de íman permanente (1), vulgarmente conhecido por *Brushed DC Motor*. O seu funcionamento é igual a um motor de corrente contínua, porém, difere-se na forma de gerar o campo magnético do indutor. Neste caso, o motor *Brushed* gera o campo magnético através de ímãs permanentes, ao invés da corrente elétrica, como é o caso dos motores de corrente contínua [43]. É um motor normalmente utilizado em sistemas de baixa potência, de fácil construção e manutenção, o que os torna num motor com um baixo preço, comparativamente a outros tipos de motores. O seu controlo é realizado com interfaces externas, como será mais detalhado na seção 4.2.7. Este modelo ainda apresenta um redutor com um conjunto de engrenagens metálicas (2) com uma relação de 1:90, possibilitando a entrega de um binário maior, mas uma velocidade de rotação mais baixa, o que é uma prioridade neste sistema.

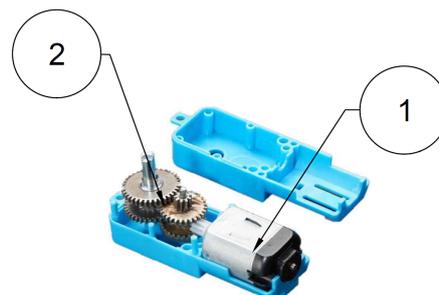


Figura 4.12: Motor elétrico selecionado

4.2 Hardware Eléctrico

O projeto eléctrico visa garantir todos os requisitos energéticos, através de uma eficiente distribuição energética a todas as interfaces do sistema, de modo a assegurar todas as capacidades de processamento de imagem e deslocação motora, sem limitar a sua performance. Esta distribuição é assegurada por um sistema de alimentação próprio que confere a portabilidade necessária ao AGV, tornando-o num sistema autónomo na fase de funcionamento. Na fase de carregamento, o AGV necessita de meios externos, para carregar o seu sistema de alimentação, juntamente com uma PCB integrada, que regula este processo de carregamento de forma automatizada.

4.2.1 Bateria Eléctrica

O sistema de alimentação do AGV necessita de garantir a portabilidade deste, sem limitações na sua mobilidade, mas também sem interferir negativamente na performance dos restantes elementos, por uma deficiente distribuição energética. Como foi mencionado na secção 2.6 o tipo de sistema de alimentação implementado é de natureza eléctrica, sendo o único sistema primário, como fonte de alimentação. Deste modo, o sistema a desenvolver deve ter um nível de independência energética, capaz de armazenar energia no seu interior, sem o auxílio de meios externos durante a fase de funcionamento. Porém, quando a bateria eléctrica já não possui uma capacidade energética suficiente para satisfazer os requisitos energéticos exigidos pelo sistema e também, como não existe nenhum meio de geração de energia eléctrica integrado no AGV a partir do meio envolvente, então deve-se passar à fase de carregamento, que é realizado com uma fonte de alimentação fixa. A elaboração deste sistema de alimentação eléctrico deve ser dimensionado, conforme as necessidades principais do sistema em causa, neste caso, conseguir uma condução responsiva e precisa, através de um contínuo fornecimento de energia, que não limite as capacidades motoras dimensionadas na secção 4.1.3, juntamente com um processamento de imagem otimizado, uma vez que a unidade processadora utilizada e apresentada da secção 4.2.4 possui elevados requisitos energéticos, comparativamente com os restantes elementos do hardware eléctrico.

Face a estes requisitos, existe uma grande variedade, em relação ao tipo de células que se podem utilizar, mas tendo em conta fatores, como características de descarga e carga eléctrica, rendimento energético, quantidade de energia acumulada (densidade), restrições mecânicas (dimensões e peso) e custos económicos, decidiu-se desenvolver uma bateria eléctrica a partir de células de lítio. Esta bateria eléctrica deve assim suportar todos os gastos e transições energéticas efetuadas pelo AGV, como é evidenciado na figura 4.13.

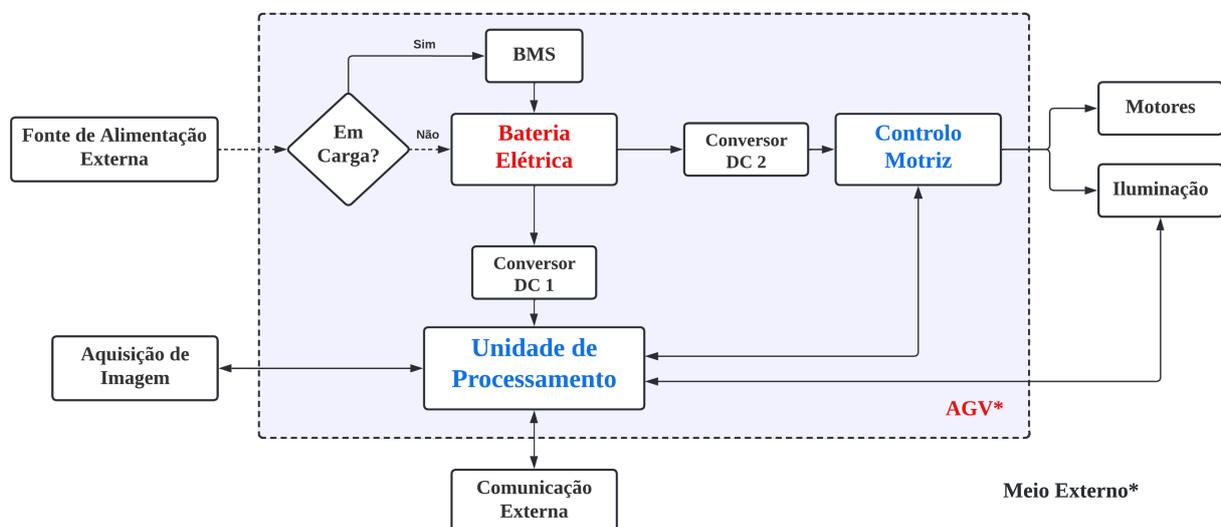


Figura 4.13: Distribuição energética do AGV

Esta escolha resulta do facto de que uma célula de lítio é um tipo de célula recarregável que se caracteriza por ter uma alta densidade energética, sem efeito memória com uma taxa de autodescarga baixa, comparativamente a outras soluções. Para além disso, face ao preço de cada célula e também às dimensões que estas são produzidas, a inclusão deste tipo de células torna-se na melhor solução. Porém, é um tipo de célula que possui eletrólitos inflamáveis, que quando danificados ou carregados de forma incorreta, podem causar a sua explosão, e por isso a necessidade de inclusão de uma BMS como se verá na seção 4.2.2, a fim de conseguir-se um carregamento seguro [44].

Deste modo, a bateria elétrica que se desenvolveu é, portanto, constituída por células do modelo da Samsung ICR18650-22E evidenciada na figura 4.14. Esta célula caracteriza-se por possuir uma tensão nominal de 3.7 V, com 2.2 Ah de capacidade energética, com uma tensão de corte em torno de 2.75 V. A tensão máxima que uma célula pode suportar é de 4.2 V, o que significa que quando uma célula atinge esta tensão elétrica, então esta encontra-se completamente carregada. Por outro lado, à medida que se gasta a energia contida na célula, a sua tensão elétrica vai diminuindo até atingir o valor da tensão de corte. Nomeadamente, a tensão de corte é a tensão elétrica que corresponde ao total descarregamento da célula. Por isso, a gama de funcionamento desta célula varia entre 2.75 e 4.2 V [45], uma vez que uma tensão abaixo, ou acima destes valores, respetivamente, resulta em condições favoráveis para a promoção de um envelhecimento prematuro desta. Por isso, o funcionamento seguro de cada célula fica restrito à gama mencionada.



Figura 4.14: Célula de lítio ICR18650-22E

Cada célula pode ser agrupada em conjuntos de modo a formar uma bateria elétrica com uma dada configuração, de modo que diferentes configurações com o mesmo número de células, resulta em baterias com diferentes características elétricas. Deste modo, uma célula pode agrupar-se em série, ou em paralelo, sendo que fora estas posições primárias, pode-se formar baterias onde coexistem ambas as configurações. No caso de se agrupar células em série, o resultado é o aumento da potência elétrica do conjunto, sendo que o fator que proporciona este aumento é a tensão elétrica, visto células em série aumentarem a tensão nominal, de modo proporcional ao número de células em série, mantendo a mesma capacidade energética. No caso de se agrupar as células em paralelo é o caso apostado, pois aumenta-se agora a capacidade energética, sendo esta proporcional ao número de células em paralelo, o que leva a que seja possível armazenar mais energia, porém, a potência é a mesma que uma única célula, visto a tensão nominal do conjunto não ser influenciado por esquemas em paralelo. No sistema desenvolvido, implementou-se uma bateria elétrica, conforme as necessidades energéticas de todo o sistema, de modo que a bateria projetada assenta num layout do tipo 3S2P, ou seja, existem dois *packs* em paralelo, onde cada *pack* é constituído por três células em série, totalizando numa bateria com seis células.



Figura 4.15: Layout 3S2P

A integração da bateria no AGV, como foi apresentado na seção anterior, leva à subdivisão desta em duas partições, com o propósito de hospedar em dois locais distintos, a fim de não sacrificar nenhuma das suas características elétricas, garantindo um fácil acesso aos elementos que diretamente se conectam a ela. O resultado desta divisão é a reestruturação do layout 3S2P convencional, num formato repartido como é destacado na figura 4.16. Portanto, com o dimensionamento desta bateria segundo o layout 3S2P, o AGV tem à sua disposição uma bateria com uma capacidade energética de 4.4 Ah, uma vez que devido ao esquema 2P integrado, a capacidade da bateria duplica face a capacidade de uma única célula que é de 2.2 Ah. Já o esquema 3S responsabiliza-se pelo aumento em três vezes da tensão nominal da bateria, passando de uma tensão nominal de 3.7 V, para uma tensão de 11.1 V ($3.7 \text{ V} \times 3$), o que significa que a potência elétrica do conjunto triplica, visto esta ser diretamente proporcional à tensão nominal, porquanto esta resultar do produto entre a tensão nominal e a corrente elétrica ($P = Vi$). Com o estabelecimento de uma nova tensão nominal, conseqüentemente existe um rearranjo na gama de funcionamento do conjunto. Esta nova gama de funcionamento, pelo mesmo motivo do aumento da tensão nominal, leva a que a tensão de corte, que antes era de 2.75 V triplique passando para 8.25 V ($2.75 \text{ V} \times 3$), e que a tensão máxima a que se expõe a bateria estabeleça-se em 12.6 V ($4.2 \text{ V} \times 3$). Com estas características elétricas, o AGV tem à disposição uma bateria com 48.84 Wh de energia que resulta do produto entre a tensão nominal do conjunto (11.1 V) e a capacidade energética da bateria (4.4 Ah).

A projeção de uma bateria com esta configuração fundamenta-se ao privilegiar antes o aumento da potência elétrica pelo aumento da tensão nominal, ao invés do aumento da capacidade energética. Esta escolha recai nesta ideologia, uma vez que a tensão elétrica é a característica mais difícil de regular, devido aos vários componentes elétricos que operam com diferentes tensões nominais. Além de mais, a tensão elétrica proveniente da bateria é dependente da quantidade de energia armazenada nas células, o que a torna variável ao longo do tempo, quando o AGV se encontra a funcionar. Como se verá na seção 4.2.3 serão integrados conversores de tensão, para que independentemente da tensão proveniente da bateria, a tensão à sua saída é constante e regulada para níveis aceites pelos componentes elétricos recetores, sendo eles a unidade processadora e a interface de controlo do movimento dos motores. Para além destes aspetos e auxiliando esta decisão, a regulação da corrente requerida pelas várias interfaces elétricas do AGV, não apresenta estas divergências em termos de compatibilidade energética, uma vez que cada dispositivo eletrónico apenas requerer a corrente que necessita, para o seu normal funcionamento. Garante-se este comportamento desde que a soma da corrente de consumo destes dispositivos seja inferior à corrente máxima que a bateria pode fornecer. Assim, como a bateria apresenta dois packs em paralelo, e como cada célula consegue fornecer de modo seguro 4.4 A, de acordo com as suas especificações técnicas, então esta bateria consegue dispor ao AGV um corrente máxima de 8.8 A devido o esquema 2P integrado, cumprindo os requisitos energéticos que este requer. Os detalhes do consumo do AGV, incluindo a sua autonomia são abordados na seção de testes.

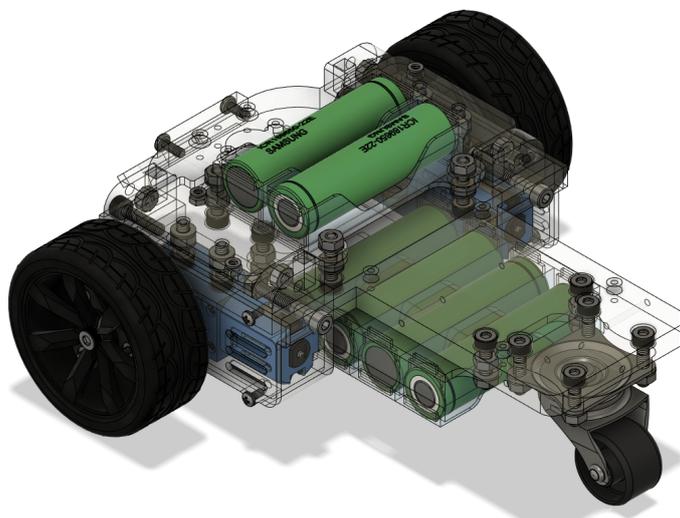


Figura 4.16: Localização da bateria elétrica

4.2.2 BMS

Não existem células idênticas mesmo sendo fabricadas nas mesmas condições e oriundas do mesmo lote de produção, uma vez que cada célula possui uma composição eletroquímica diferente de qualquer outra [46]. Dependendo da forma como são agrupadas, algumas células podem estar expostas a diferentes temperaturas, formando-se um gradiente térmico em torno da bateria. Esta desigualdade térmica acontece, ou por um design impróprio, ou por um sistema de refrigeração defeituoso, ou ainda por condições ambientais heterogêneas, o que leva a que operação de carregamento e outras características elétricas sejam amplamente diferenciadas no mesmo conjunto de células. Estas desigualdades levam ao aumento das diferenças eletroquímicas entre elas e portanto, quando se apresentam diferenças significativas a este nível, e em especial em células constituintes de um sistema móvel totalmente automatizado, cria-se um mecanismo prejudicial, que com o natural envelhecimento das células e com os sucessivos ciclos de carregamento sem a utilização de interfaces próprias, a fim de assegurar um carregamento seguro, promove-se uma diminuição da vida útil de cada célula, e obtenção de células com características elétricas inferiores às suas especificações [47].

Este problema é evidenciado em células em série, uma vez que células com esta configuração compartilham a mesma corrente elétrica, o que significa que a célula com uma capacidade de carregamento mais elevada, atinge a sua tensão máxima, antes que outras células atinjam a sua carga total. Visto de outra maneira, a célula com a taxa de carregamento mais baixa atinge a sua tensão mínima de descarga (tensão de corte), antes de qualquer célula esgote a sua capacidade energética. Estas diferenças, oriundas de composições eletroquímicas diferentes, culminam ou na sobrecarga elétrica da(s) célula(s) com maior taxa de carregamento, ou na descarga excessiva da(s) célula(s) com menor taxa de carregamento. Este comportamento leva a que algumas células se encontrem em condições fora dos limites de utilização, favorecendo-se uma deterioração precoce das suas propriedades elétricas. Para solucionar este problema, fora todo o design a fim de promover condições igualitárias, para o carregamento das células, é necessário estabelecer uma operação segura, respeitando as características das células, de modo a otimizar todo o processo. Este mecanismo compensa assim as naturais diferenças eletroquímicas entre as células, sendo designado por BMS. Portanto, uma BMS (*Battery Management System*) é um sistema eletrónico que gere o processo de carregamento de uma bateria elétrica, realizando o equilíbrio e protegendo-a ao evitar que esta opere fora da sua gama normal de funcionamento, evitando criar condições que promovam um aumento das diferenças eletroquímicas entre células [48].

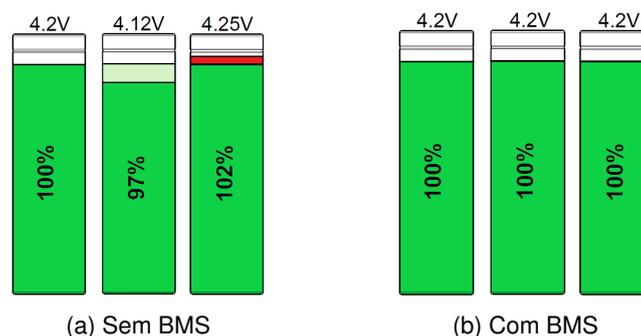


Figura 4.17: Diferenças na presença de BMS no processo de carregamento

Deste modo, a BMS desenvolvida terá a exclusiva tarefa de realizar o equilíbrio da bateria implementada, através de um sistema puramente elétrico. Não existe forma de monitorizar o estado das células, nem gerir o seu equilíbrio através de firmware, visto o objetivo desta BMS é realizar apenas um carregamento seguro sem a interferência de meios externos. Como se verá na última seção, existe a possibilidade de continuar a desenvolver esta BMS, de um modo a trocar de uma operação local, para um sistema acessível via algum protocolo de comunicação sem fios, o que acrescenta muitas vantagens em termos de monitorização e extração de dados. Esta possibilidade transforma-se assim num *upgrade* para trabalho futuro.

A BMS projetada, inicia-se com um circuito que regula, tanto a tensão, como a corrente elétrica a que se expõe a bateria. Estas características são bastante importantes, pois por um lado garante-se que existe controlo sobre a corrente máxima a que se submete as células, uma vez que ao não existir este controlo, não se garante que o processo de carregamento esteja a ser efetuado de um modo seguro, sem interferir negativamente nas propriedades eletroquímicas das células. Para além disso, ao se limitar a corrente máxima, consegue-se também proteger o restante circuito de possíveis curtos-circuitos, na eventualidade da corrente numa dada instância ser superior à corrente máxima tolerada por elementos integrantes da BMS. Por outro lado, ao ter-se a capacidade de regular a tensão de saída, independentemente do valor da tensão de entrada, consegue-se agora utilizar diversas fontes de alimentação e internamente regular a tensão de saída para o valor dimensionado. Assim, deixa de ser necessário utilizar uma fonte de alimentação específica, para se efetuar o carregamento da bateria, uma vez que sem esta possibilidade, a BMS estaria dependente de um número restrito de fontes de alimentação que fornecessem apenas a tensão requerida e naturalmente a portabilidade do AGV seria bastante limitada.

Neste sistema, a fim de assegurar estas características, utilizou-se um regulador linear ajustável de tensão positiva, o LM317T. Este possui três pinos: *Input*, *Output* e *Adjustment*, capaz de fornecer até 1.5 A de corrente de saída com a possibilidade de efetuar o processo de conversão entre 1.25 a 37 V (tensão de saída). Esta conversão apenas acontece caso a tensão de entrada seja superior à tensão de saída com uma diferença maior ou igual a 1.5 V. Porém, a diferença entre a tensão de entrada e a tensão de saída relacionada com a corrente conduzida pelo regulador, resulta em perdas manifestadas sob a forma de calor residual [49]. Quanto maior são estas perdas, menor é o rendimento de conversão, o que se torna numa desvantagem neste tipo de regulador, visto a potência elétrica dissipada sob a forma de calor poder ser superior à potência útil, para o processo de conversão. Existem alternativas a este tipo de regulador, que possuem melhores eficiências energéticas, porém são reguladores mais complexos de implementar de raiz, que necessitam de um maior número de elementos e conseqüentemente possuem um maior *footprint*, o que influencia o tamanho da PCB desenvolvida. Fora a questão da eficiência energética, o LM317T é de fácil implementação e caracterização, o que o torna num elemento bastante viável em diversas aplicações deste tipo, transformando-o num elemento-chave, para tornar a BMS desenvolvida numa placa mais genérica e adaptável, face ao uso de diferentes células e diferentes valores nominais, para o carregamento destas.



Figura 4.18: Regulador linear LM317T

O regulador LM317T pode fornecer um excesso de corrente à saída deste e, portanto, é conceitualmente considerado um amplificador operacional. Justamente por esta característica, o pino de *Adjustment* é uma entrada inversora do amplificador, mas também é o pino responsável por produzir um tensão de referência estável (V_{ref}) em torno de 1.25 V, entre este e o pino de *Output*. Logo, a tensão à saída deste pino pode ser regulada através de um divisor de tensão externo puramente resistivo, que através da relação entre as resistências utilizadas, resulta numa tensão constante à saída do regulador, entre o pino de *Output* e o *GND*. A manutenção desta continuidade deve-se ao facto da tensão de referência ser independente da tensão de alimentação (tensão de entrada), mas independente também da temperatura e da frequência de operação.

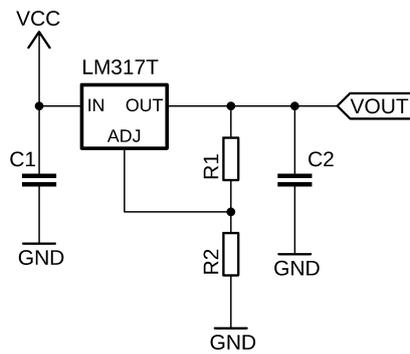


Figura 4.19: Esquema elétrico para obtenção de uma tensão constante

Deste modo, ao ligar-se o pino de *Adjustment* ao divisor de tensão resistivo evidenciado na figura 4.19, pelas resistências R_1 e R_2 , o valor da tensão de saída (V_{out}) pode ser obtido pela análise do circuito acima, de modo que

$$V_{out} = V_{ref} \left(1 + \frac{R_2}{R_1} \right) + I_{adj} R_2. \quad (4.10)$$

Porém, a corrente (I_{adj}) que circula entre o pino *Adjustment* e divisor de tensão é em torno de $50 \mu\text{A}$, o que perante as condições de funcionamento desta BMS, este valor pode ser desprezado, de modo a que expressão 4.10 simplifica-se ao ponto de

$$V_{out} \approx V_{ref} \left(1 + \frac{R_2}{R_1} \right). \quad (4.11)$$

Para auxiliar na regulação do sinal de saída, foram adicionados dois condensadores em paralelo, um à entrada do regulador e outro à saída deste, C_1 e C_2 , respetivamente. O condensador C_1 é um condensador eletrolítico responsável por filtrar os sinais de baixa frequência e, conseqüentemente, responder rapidamente a mudanças de carga, o que promove uma diminuição de ruído à entrada do regulador. Por si só, não consegue filtrar ruídos de alta frequência, uma vez que este tipo de condensador possui uma grande indutância e resistência interna. Como o domínio de frequências ao longo do processo de carregamento é predominantemente no espectro da baixa frequência, então a utilização exclusiva deste condensador justifica-se não sendo necessário adicionar um condensador cerâmico, visto um condensador deste tipo ao não ser polarizado possui uma baixa indutância, proporcionando uma excelente resposta à eliminação do ruído no espectro da alta frequência. O condensador C_2 também é um condensador eletrolítico, só que neste caso, este melhora a resposta transitória não sendo necessário, para a estabilidade do sistema [50]. É de notar, que a capacidade deste condensador não deve ter um valor significativo, para evitar sobrecarregar o LM317T causando-lhe danos físicos expressivos. Este cenário apenas acontece, caso no ato de carregamento ocorra a separação da parte que efetua o equilíbrio das células da parte onde está o regulador, simultaneamente que se deixa de alimentar o circuito através da fonte de alimentação. Deste modo, o circuito que integra o regulador fica em aberto à entrada e saída deste, levando ao condensador que armazena energia a descarregar no sentido do regulador devido à ausência de tensão que ocorre pela falta da fonte de alimentação. Como não existe nenhum mecanismo de proteção integrado, o LM317T sofre com esta descarga. Tendo em atenção estes fatores, o regulador LM317T foi projetado para fornecer 12.6 V à saída, porque como foi apresentado na seção anterior cada célula singular atinge a capacidade máxima aos 4.2 V e como existe um layout 3S implementado, então esta tensão triplica para o valor estipulado de 12.6 V, garantindo-se o carregamento completo de todas as células. Para se atingir esta tensão à saída (V_{out}) é necessário que a tensão de entrada (V_{in}) seja igual ou maior que 15 V (até 40 V), de modo a assegurar o processo de conversão.

Estas características proporcionam a transformação do LM317T num regulador de tensão estável, porém, ainda com o mesmo regulador é possível torná-lo também num elemento limitador da corrente máxima a que se submete a bateria elétrica. Para tal, basta integrar um resistência externa (R_3) ao regulador, estado esta em série entre o pino de *Output* e o início do divisor de tensão, como é evidenciado na figura 4.20. Salienta-se que a introdução desta resistência não provoca a alteração, nem oferece instabilidade em relação à tensão de saída previamente projetada.

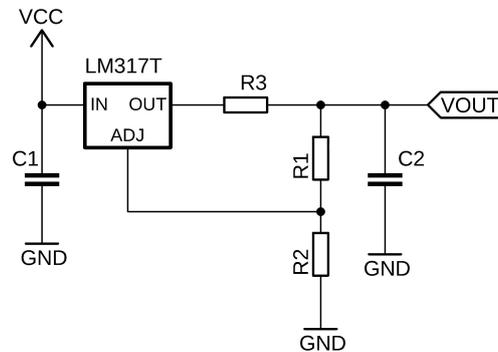


Figura 4.20: Esquema elétrico para obtenção de uma tensão constante e corrente limitada

De modo similar à análise efetuada para o regulador de tensão, a única alteração é a introdução da resistência (R_3), portanto, através da análise do circuito acima, a corrente limite (I_{limit}) é dada por

$$I_{limit} = \left(\frac{V_{ref}}{R_3} \right) + I_{adj}. \quad (4.12)$$

Mais uma vez, a corrente que circula entre o pino *Adjustment* e o divisor de tensão é em torno de $50 \mu A$, o que perante a ordem de grandeza a que a BMS opera, este valor pode ser desprezado apresentando um erro insignificante no final do dimensionado da corrente limite. Assim, a expressão 4.12 simplifica-se ao ponto de

$$I_{limit} \approx \left(\frac{V_{ref}}{R_3} \right). \quad (4.13)$$

Contudo, não se pode ignorar o facto de existir na prática uma resistência interna (R_i), entre o fim da resistência R_3 e o início da bateria elétrica. O facto de existir esta resistência interna em série com a resistência dimensionada R_3 , leva a que a resistência total a suprimir a corrente limite seja a soma entre as duas ($R_3 + R_i$). Idealmente esta resistência interna não existe, contudo, na prática o fio não é ideal. Como a corrente limite de forma simplificada depende apenas do valor da tensão de referência (V_{ref}) e este valor é um valor relativamente baixo, qualquer alteração mínima no denominador da expressão 4.13 leva a uma alteração expressiva na corrente máxima que se consegue conduzir pelo regulador. Como a resistência interna é sempre maior que zero, então a corrente limite real ($I_{limit,r}$) é sempre menor que a corrente limite projetada (I_{limit}). Para solucionar este problema, basta medir o valor desta resistência interna, que através de medições na PCB desenvolvida resultou numa resistência interna de 0.3 ohm. Portanto, das expressões formuladas, basta adicionar este valor ao denominador da expressão 4.13, resultando numa expressão onde a corrente máxima real é conseguida por

$$I_{limit,r} \approx \left(\frac{V_{ref}}{R_3 + R_i} \right). \quad (4.14)$$

Desta forma, ao ter-se em consideração esta resistência interna, dimensionou-se que a corrente máxima a que se submete a bateria fica em 1.4 A, sendo que o LM317T consegue conduzir até 1.5 A. Não se dimensiona para a sua capacidade máxima, uma vez que não é recomendado a qualquer componente elétrico operar continuamente nos seus limites, uma vez que ao estar exposto a estas condições, leva a uma diminuição da sua vida útil. O restante circuito elétrico encontra-se também protegido, na medida que todos os elementos que diretamente influenciam no processo de equilíbrio, suportam esta corrente calculada, não se correndo o risco de possíveis danos físicos devido à exposição a uma corrente acima das suas tolerâncias.

Esta corrente, juntamente com a tensão de saída regulada são resultados de um processo de conversão, na qual resulta sempre perdas energéticas sob a forma de calor. Perante estas perdas, o LM317T sozinho não detém uma capacidade física, capaz de dissipar o calor gerado pelo processo de conversão, a partir de uma certa potência sustentada por estas perdas. Quando se ultrapassa esta potência, gera-se um problema, uma vez que se está a produzir mais calor do que aquele pode ser dissipado. Esta situação leva à deterioração do regulador, devido à incapacidade de dissipar o calor gerado pelo meio envolvente, levando ao regulador a atingir temperaturas fora da sua gama de funcionamento. Para evitar esta situação, em qualquer fase do processo de carregamento, é necessário a introdução de um instrumento que promova uma maior dissipação do calor gerado, garantindo um melhor desempenho e confiabilidade do regulador em uso. Este componente é um dissipador de calor, como é identificável na figura 4.21, responsável por conceder um maior poder de refrigeração e consequentemente, uma maior capacidade de dissipação.

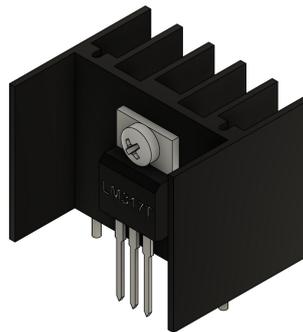


Figura 4.21: Dissipador passivo + LM317T

Este é um dissipador de calor passivo o qual é composto por alhetas de metal, mais em concreto de alumínio não possuindo mecanismos que proporcionam convecção forçada para auxiliar na dissipação do calor gerado. Ao ser um dissipador passivo, significa que a energia térmica proveniente do LM317T é transferida através de condução, provocando o aparecimento de um gradiente térmico entre eles, de modo que a direção da transferência de calor ocorre do regulador para o dissipador respetivamente, uma vez que a temperatura no regulador é superior à temperatura do dissipador. No dissipador, esta energia térmica continua a ser conduzida até às extremidades das alhetas, onde agora, por convecção natural a energia térmica passa a ser transferida de um meio sólido para um meio gasoso (atmosfera envolvente), realizando-se assim a dissipação da energia térmica [51].

O facto de o dissipador ser constituído por alumínio, deve-se à sua elevada condutividade térmica que é em torno de 235 W/mK. Quanto maior este valor for, mais energia térmica pode ser transferida através de condução de um meio para o outro, ou conduzir a mesma quantidade num menor intervalo de tempo. Como a condutividade térmica é proporcional à transferência de calor e à área do dissipador, a escolha de um dissipador de metal, com uma área superior à área do LM317T, proporciona uma capacidade de dissipação maior, além de mais, ganhar-se vantagens na obtenção de um sistema de refrigeração sem gastos energéticos que promove a inexistência de ruído, contrastando com um sistema de *cooling* e ao ruído provocado por este [52]. Porém, a principal desvantagem torna-se no seu tamanho que pode tornar-se complicado em certas arquiteturas. Na PCB desenvolvida, a introdução deste dissipador não apresentou qualquer problema deste género.

Com a regulação e dimensionamento dos atributos anteriores, a BMS realiza agora o que ela propõe, o equilíbrio das células. Esta parte do circuito evidenciada na figura 4.22 foi projetada em blocos, na medida em que como existe uma configuração 3S ao nível da bateria, então existem três blocos idênticos em série que perfazem o circuito responsável pelo equilíbrio, onde cada bloco é responsável por um par de células em paralelo, visto o esquema 2P agregado, outorgando no esquema 3S2P implementado. Cada bloco, funciona com o mesmo princípio de funcionamento, operando independentemente uns dos outros, de forma a se ajustarem ao respetivo par de células associadas, conforme a taxa de carregamento de cada uma, oriunda das suas composições eletroquímicas. Por isso, ao explicar-se o funcionamento de um bloco, explica-se para os restantes.

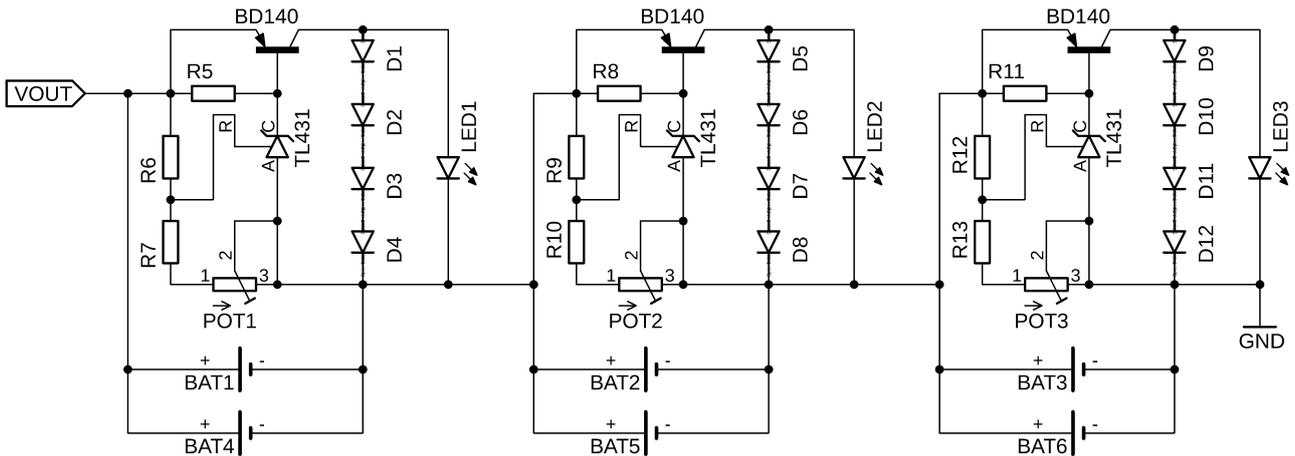


Figura 4.22: Circuito responsável pelo balanceamento das células

Para iniciar o carregamento da bateria elétrica são necessárias duas condições básicas: o AGV estar desligado, para tornar o processo de carregamento mais eficiente, não gastando uma fração desta energia para a manutenção do estado de operação do AGV, e assegurar que a bateria se encontre significativamente descarregada, pois se a bateria estiver próxima da sua capacidade máxima, o carregamento desta pouco influencia, levando a uma alteração mínima da sua capacidade. Garantindo-se estas condições triviais, o principal mecanismo do circuito fundamenta-se num diodo Zener, o TL431 composto por três terminais: a referência (R), o cátodo (C) e o ânodo (A). Ao nível lógico, o TL431 é um amplificador operacional de malha aberta, que possui uma tensão de referência, entre o terminal de referência e o ânodo, variável entre 2.5 a 36 V. Neste sistema, pretende-se regular esta tensão para 4.2 V, através de um divisor de tensão externo. Oferece-se a capacidade de se regular esta tensão manualmente, através de um potenciômetro (POT) em série com as resistências constituintes deste divisor, a fim de ser possível de o regular com precisão e ter a capacidade de escolher qual o valor da tensão de referência, uma vez que a escolha deste valor influencia até onde a BMS carrega a bateria elétrica. Uma vantagem de não efetuar um carregamento completo da bateria é por exemplo o aumento da vida útil das células, mas nas condições deste trabalho promove-se um carregamento total destas, e por isso, a escolha de 4.2 V por cada par de células em paralelo.

Esta tensão de referência compara-se sempre ao longo do processo de carregamento com a tensão proveniente das células do respetivo bloco. Sempre que a tensão da célula é inferior à tensão de referência, o TL431 funciona como um transistor bipolar, ou seja, sempre que a tensão da célula é inferior a 4.2 V, então o TL431 encontra-se aberto internamente, o que significa que não se consegue conduzir corrente do cátodo, para o ânodo. Como o cátodo se encontra ligado à base do transistor bipolar PNP BD140 e a base deste, possui uma resistência *pull-up*, então o efeito da ausência da corrente no TL431, provoca o mesmo efeito no transistor ao não existir corrente na sua base, o que o impede de conduzir corrente entre o emissor e o coletor, e conseqüentemente para o restante circuito. Este efeito, leva a que todo o circuito se encontre em aberto, à exceção do circuito que permite o carregamento das células, e que por isso resulta no carregamento destas.

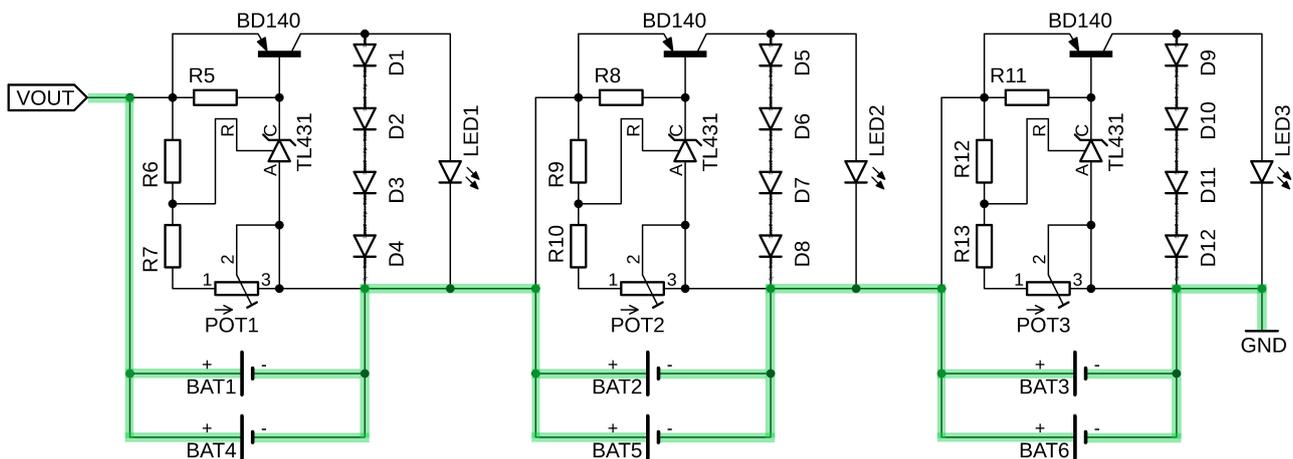


Figura 4.23: Fluxo de corrente elétrica no estágio inicial do carregamento da bateria

Quando todos os blocos apresentam esta característica, então todas as células constituintes da bateria elétrica encontram-se a carregar, como é evidenciado na figura 4.23 destacando-se visualmente o fluxo da corrente elétrica ao longo da BMS. Porém, as células possuem composições eletroquímicas diferentes, e por isso o tempo que levam a atingir o valor da tensão de referência varia, o que leva à seguinte situação: quando as células de um bloco atingem o mesmo valor que a tensão de referência, independentemente se as outras células ainda estão no processo de carregamento, internamente o componente TL431 ativa-se, de modo a permitir a condução de corrente entre o cátodo e ânodo do respetivo bloco. Ao existir esta corrente, e como o cátodo está ligado à base do transístor BD140, então promove-se a existência de corrente na base do transístor, o que segundo o seu modo de funcionamento, ao existir esta resistência *pull-up*, resulta no fluxo de corrente entre o emissor e o coletor. Graças a esta ação, forma-se um novo caminho, que provoca um desvio da corrente elétrica, deixando de atuar nestas células, passando agora a atuar na carga simulada pelos díodos de silício.

Este comportamento leva a duas particularidades, em primeiro a proteção das células contra a sobrecarga elétrica, pois sem este mecanismo estas continuariam a carregar de forma indefinida, passando a operar numa região fora da sua gama segura de funcionamento ($> 4.2\text{ V}$), trazendo riscos que comprometeriam a sua integridade física e características elétricas disponíveis. Deste modo, ao se adicionar quatro díodos em série, simula-se uma carga à saída do coletor do transístor BD140, protegendo-se as células que estão totalmente carregadas. Este engenho não é o mais eficiente energeticamente, uma vez que se continua a gastar energia elétrica nas células já carregadas, gastando-se de forma supérflua através do calor gerado pelos díodos, devido à queda de tensão (0.6 V) e resistência interna inerentes a eles. Independentemente do tempo que a bateria continua a ser carregada, mesmo com ela na sua capacidade máxima, esta encontra-se protegida por esta carga simulada, não havendo quaisquer riscos associados. Em segundo lugar, ao colocar-se um LED em paralelo a esta carga simulada, promove-se um indicador visual, que indica quando um conjunto de células de um bloco se encontra carregado. No início do carregamento, como não existe corrente nesta parte do circuito porque a tensão das células é inferior a 4.2 V , então o LED encontra-se desligado, o que torna este mecanismo de verificação visualmente intuitivo, como se observa na figura 4.24, onde existe um bloco com as células carregadas, enquanto os restantes ainda se encontram com as células a carregar. Por este raciocínio, quando existem três LEDs acessos, então é porque a bateria elétrica encontra-se totalmente carregada.

Fora a sua caracterização, durante o funcionamento do AGV a BMS não possui qualquer função útil, na gestão das células, a não ser de servir como interface intermediária entre a bateria elétrica e os conversores de tensão, uma vez que aproveita as suas ligações interna para realizar estas ligações através dos seus pinos comuns (COM), economizando-se ligações externas.

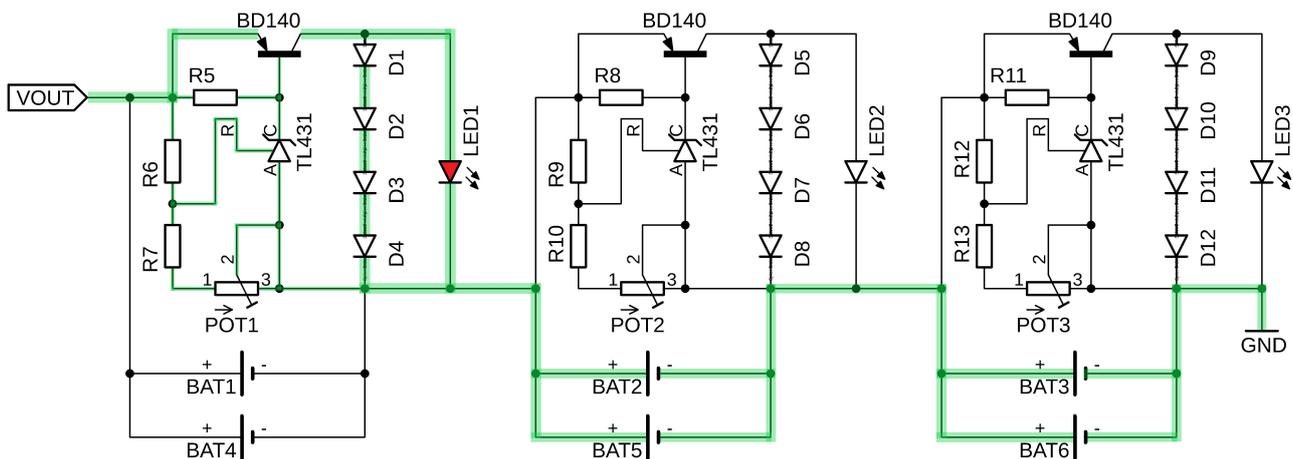


Figura 4.24: Balanceamento da bateria elétrica

A integração da BMS no AGV deve ser realizada de forma compacta e estratégica, de modo a ir ao encontro dos ideais defendidos na seção de hardware mecânico, através do desenvolvimento de um sistema portátil com fáceis acessos, com uma arquitetura projetada de modo a suplementar as suas necessidades e otimizar as funções a desempenhar. Ao seguir-se esta ideia conceptual, desenvolveu-se uma PCB com o circuito que forma a BMS desenvolvida, a fim de ir ao encontro a estas características. Uma PCB (*Printed Circuit Board*) é uma estrutura sanduíche constituída por uma placa isolante de epóxi de vidro FR-4 (elemento mais comum) ou outro compósito com propriedades isolantes intercalada por um determinado número de películas de cobre laminado conforme a sua complexidade, que por alterações químicas dão origem às trilhas condutoras, isoladas pelo material do substrato. A superfície de uma PCB apresenta um revestimento que protege o cobre contra a erosão designado por máscara de solda. As PCBs apresentam duas funções completares, onde uma é a fixação mecânica dos elementos elétricos nas camadas externas, através da brasagem destes nos respetivos locais indicados pelo seu símbolo presente na máscara de solda, e o segundo, é a conceção de um circuito compacto com ligações elétricas confiáveis, que evitam curtos-circuitos entre pontos de brasagem [53]. A brasagem é o processo térmico que permite realizar a junção de elementos metálicos por um metal de adição. Neste processo, existe a fusão do metal de adição através da sua fusão não ocorrendo a fusão do material base, o que torna este processo diferente de um processo de soldagem, visto num processo deste tipo existir fusão do material base [54].

A conceção da PCB é realizada num software da Autodesk, o EAGLE, que permite a automação de projetos elétricos através de ferramentas que permitem a criação do layout de uma PCB, a partir de um esquema elétrico. O EAGLE contém um editor esquemático para projetar diagramas elétricos, sendo estes constituídos por símbolos normalizados, que advém de bibliotecas padronizadas pelo próprio software. Ao se utilizar estes símbolos, constrói-se um diagrama elétrico evidenciando-se as ligações entre os vários elementos. Estas ligações na esquemática não correspondem necessariamente ao arranjo final destes na PCB. Portanto, quando o esquema elétrico está concluído, basta convertê-lo para o ambiente de edição respetivo, onde agora é modelado os trilhos dos vários elementos e as suas respetivas posições e ainda características relacionadas ao próprio processo de fabrico da PCB, como furos, largura de trilhas, número de camadas, etc. Com isto, a PCB desenvolvida é uma placa com 1.6 mm de espessura que possui duas camadas de cobre, isolada internamente por epóxi de vidro FR4 – Standard TG, livre de estanho com chumbo e outras substâncias perigosas usadas durante o seu processo de fabrico (Lead-Free HASL-RoHS) uma vez serem prejudicial ao ser humano e ao ambiente devido à sua toxicidade, certificando-se esta placa com norma de qualidade IPC (*Internacional Patent Classification*) classe 2 Standard rotulada como um produto elétrico de serviço especializado, para conceber um desempenho contínuo e um ciclo de vida longo garantindo uma boa qualidade de construção, segundo esta padronização [55].

A utilização de duas camadas permitiu criar dois conjuntos de trilhos, o *top* e o *bottom* evidenciados na figura 4.25 (a), ambos com 0.4 mm de espessura, de forma a garantir-se que a sua área transversal suporta a corrente máxima dimensionada, não se correndo o risco de ocorrerem curtos-circuitos. Cada trilho, em cada camada, não possui qualquer interseção entre si, a não ser que no esquema elétrico existam essas interseções. Por isto, foi necessário salvaguardar uma distância entre trilhos, de modo a evitar possíveis contactos entre eles. Para a máscara de solda criou-se uma camada *tNames*, onde se encontra a indicação simbólica de todos os elementos, e ainda a caracterização da versão da placa e elementos descritivos. Todo este grau de personalização é conservado durante o processo de fabrico, resultando na placa com as qualidades mencionadas na página anterior, apresentada por ordem cronológica na figura 4.25.

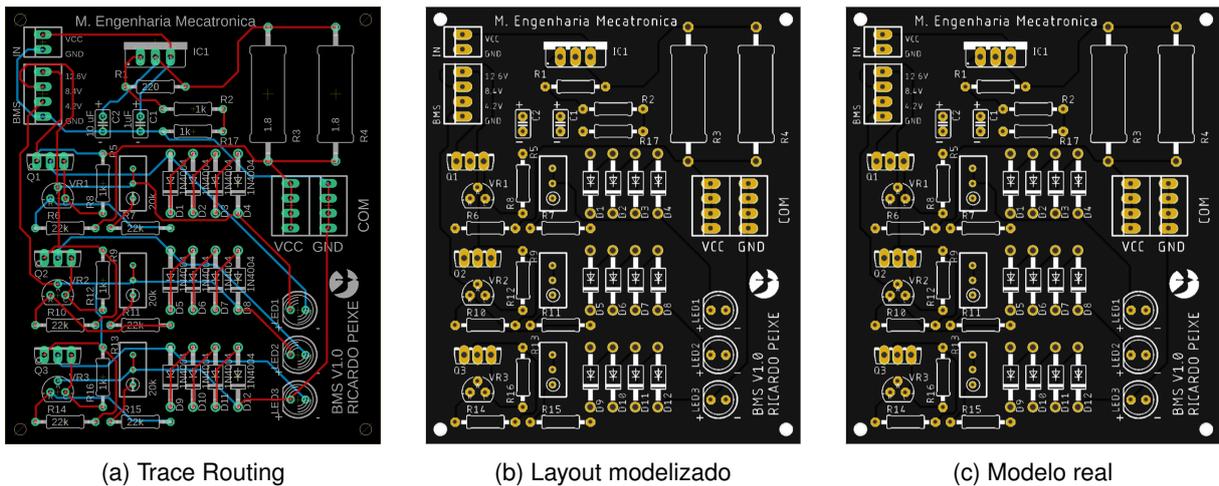


Figura 4.25: Processo de conceção da PCB

O software onde se realizou os desenhos técnicos mecânicos (Fusion 360) advém do mesmo grupo que o EAGLE, e como existe uma relação entre ambos, então é possível a modelação de uma modelo CAD da PCB desenvolvida, o que torna bastante benéfico, uma vez que permite a integração da PCB totalmente modelada com os seus componentes discretos no modelo global do AGV. Graças à disposição física de todos os elementos, ajuda-se na modelação de elementos estruturais, que intervêm diretamente e indiretamente nela, evitando-se criar componentes que fiquem dependentes de se verificar, se realmente as dimensões projetadas vão de encontro ou não às suas características geométricas através de um modelo de tentativa e erro. Porém, com esta modelação, evita-se esta metodologia e verifica-se todas as dimensões geométricas através de software. Assim, a PCB totalmente modelada encontra-se na figura 4.26.

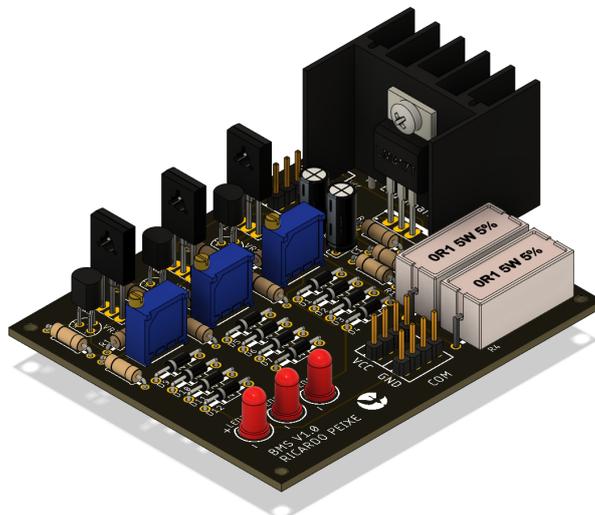


Figura 4.26: Modelo CAD da PCB

4.2.3 Conversor Buck

A tensão fornecida pela bateria é variável conforme a sua capacidade energética, apresentando uma tensão de 12.6 V à saída, quando esta se encontra totalmente carregada, porém perto da sua descarga total a tensão entregue é cerca de 8.25 V. Esta variação irregular impede a existência de um sinal constante à saída, impossibilitando a existência de condições que promovam um fornecimento energético constante, ao longo do funcionamento do AGV. Outro problema é o facto de a tensão provinda da bateria ser acima da tensão nominal das interfaces que directamente se conectam à bateria, causando um problema de compatibilidade energética. À semelhança do que se realizou na BMS, a inclusão de um regulador de tensão, neste caso um conversor buck, garante resolver estas características, e proporcionar outras bem-vindas relacionadas com a autonomia da bateria eléctrica. Portanto, um conversor buck é um conversor de energia que reduz a tensão de entrada fornecendo uma tensão de saída regulada. Este é um conversor de classe SMPS (*Switched Mode Power Supply*), o que significa que, ao contrário de um regulador linear, como o LM317T, este tipo de regulador possui um transístor de comutação que alterna continuamente entre estados de baixa dissipação (*Full-ON* e *Full-OFF*) com uma alta frequência, gastando muito pouco tempo nas transições de alta dissipação, o que minimiza as suas perdas energéticas, mantendo uma tensão à saída constante. Deste modo, a regulação da tensão de saída é conseguida pela variação da relação entre o tempo ligado e o tempo desligado do transístor, que realiza a comutação. Em contraste, num regulador linear, a tensão de saída é regulada dissipando continuamente a energia sob a forma de calor causando uma baixa eficiência no processo de conversão. Num conversor deste tipo, o processo possui uma grande eficiência e por isso a sua escolha, uma vez que se pretende maximizar a autonomia da bateria, que através de eficiências próximas de 96% permitem alcançar este objetivo, gerando perdas eléctricas quase insignificantes. Já no caso do carregamento da bateria eléctrica, a utilização de um regulador linear é válido, uma vez que a energia advém de uma fonte externa, e por isso não existe uma preocupação em relação à eficiência do processo de conversão, na ótica do AGV.

Um conversor buck é constituído por um circuito *Flywheel*, composto por um díodo (D1), um indutor (L1) e um condensador (C1), fora o transístor que realiza a comutação (Q1). Quando o transístor de comutação está ligado (estado *Full-ON*), este está a alimentar directamente a carga. Inicialmente, o fluxo para a carga é restrito, pois a energia advinda do transístor está a ser acumulada no indutor através de um campo magnético, e também armazenada no condensador. Durante este período, forma-se uma tensão positiva no cátodo do díodo e, por isso, o díodo encontra-se inversamente polarizado, não desempenhando qualquer função nesta etapa. Porém, quando o transístor se desliga (estado *Full-OFF*) a energia armazenada no campo magnético é libertada de volta para o circuito, de modo que a tensão aos terminais deste encontra-se inversamente polarizada. Neste período inicial, garante-se energia à carga através da energia proveniente do indutor até ao colapso do seu campo magnético. Com a polaridade inversa do indutor, então o díodo fica directamente polarizado começando a conduzir corrente fechando internamente o circuito. Quando a energia fornecida pelo indutor estiver perto de se esgotar e a tensão na carga começa a diminuir, o condensador em paralelo torna-se na principal fonte de corrente, alimentando a carga, até o próximo estado *Full-ON*. Este processo repete-se com uma alta frequência minimizando as perdas energéticas no processo de conversão [56].

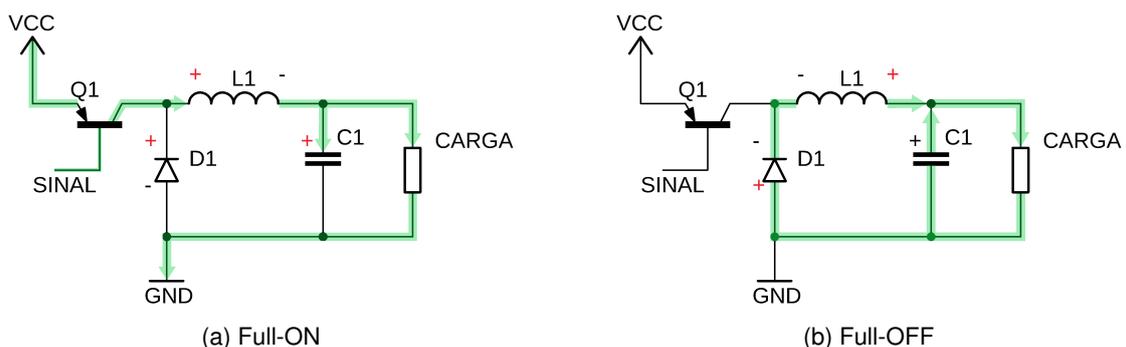


Figura 4.27: Modo de operação do conversor buck

Deste modo, foram integrados dois conversores buck comerciais, sendo que um é responsável pelo processo de conversão da Ponte H, a unidade responsável pelo controlo motriz do AGV, e o segundo responsável pela regulação da tensão de operação da unidade processadora, o Raspberry Pi 4B. Ambos possuem um potenciômetro às suas saídas, que permite regular a tensão desejada para cada interface, permitindo um ajuste preciso às tensões nominais de cada elemento. Estes conversores, respetivamente, são LM2596 e o XL4015.

4.2.3.1 LM2596

O conversor LM2596 presente na figura 4.28, é um circuito integrado com a capacidade de converter uma tensão de entrada até 40 V, em uma tensão de saída que varia de 1.2 a 37 V, conforme a regulação efetuada pelo potenciômetro. Este gama de valores à saída do conversor torna conformável o seu uso para operar perante a gama de funcionamento da bateria do AGV, pois esta varia entre 8.25 a 12.6 V. Esta conversão é conseguida com uma eficiência máxima de 94%, através do circuito de comutação que opera com uma frequência de 150 kHz, tendo a capacidade de conduzir uma corrente máxima de até 3 A, não limitando internamente a corrente necessária para a operação normal dos motores elétricos, geridos pela ponte H [57]. Ao configurar-se este conversor, para servir de fonte de alimentação à ponte H, a tensão de saída é regulada para 7 V, mantendo-se constante durante toda a fase de funcionamento do AGV, uma vez que a tensão mínima, para originar o sinal de saída ser de 8.2 V (7 V + 1.2 V).



Figura 4.28: Conversor LM2596

4.2.3.2 XL4015

O conversor XL4015 presente na figura 4.29, tal como o regulador anterior é um circuito integrado com a capacidade de converter uma tensão de entrada entre 8 e 36 V, através de um sistema de comutação *step-down*, resultando numa gama à sua saída abrangível dos 1.25 aos 32 V. Este processo de conversão é conseguido com uma eficiência máxima de 96%, graças à sua elevada velocidade de comutação fixa numa frequência de 180 kHz, através de um MOSFET controlado por um sinal PWM (*Pulse Width Modulation*). É um conversor com uma capacidade de condução de até 5 A, limitando-se a um dispositivo que consuma até 75 W de potência. Comparativamente com o conversor anterior é um conversor com uma maior potência e eficiência energética, com um número de características internas que o tornam mais seguro o que é bem-vindo, visto ser o elemento que irá alimentar a unidade processadora do AGV, e por isso a necessidade de se integrar um conversor com estas características [58]. Portanto, a tensão de saída foi regulada para 5.4 V, uma vez que devido às quedas de tensão que acontecem internamente no Raspberry Pi, a tensão a que se expõe o sistema cai para 5.1V, que é a tensão recomendada, para não existir perdas de performance no processador.



Figura 4.29: Conversor XL4015

4.2.4 Raspberry Pi

O AGV é um sistema robótico com total liberdade de atuação dentro do seu percurso e autonomia nas suas decisões. Estes atributos são proporcionados por todo um engenho global criado, interligando-se as diferentes funções, de modo a conseguir-se um sistema completo, que permite funcionar como um todo, segundo o ideal projetado. Para isto, é necessário um controlador interno, que tenha capacidade gráfica e a performance necessária, para suportar a análise de imagem em tempo real através de visão artificial, que tenha a conectividade para controlar a forma, como se gere o deslocamento mecânico de todo o sistema, e que tenha a capacidade de comunicação externa com o utilizador, através de diversas tecnologias de comunicação, garantindo-se não estar limitado a apenas um protocolo de comunicação. Por isso, perante estas necessidades, integrou-se como única unidade processadora um Raspberry Pi 4 modelo B. Portanto, o Raspberry Pi 4B é um computador de baixo preço lançado em 2019 com um alto desempenho, integrando-se numa placa de tamanho reduzido [59]. As principais especificações dele estão descritas na tabela 4.3.

Raspberry Pi 4 modelo B	Especificações
	<ul style="list-style-type: none">• CPU: Broadcom BCM2711, 1.5GHz quad-core 64-bit ARM Cortex-A72• RAM: 4GB LPDDR4• Portas USB: 2x portas USB 3.0 e 2x portas USB 2.0• Conectividade: Bluetooth 5.0, BLE, 2.4 GHz e 5.0 GHz IEEE 802.11b/g/n/ac wireless; LAN, Gigabit Ethernet;• GPIO: 40 pinos padrão• Vídeo e Som: 2x portas micro HDMI (até 4Kp60), 1x porta para câmara, 1x porta para display e Áudio estéreo• Cartão SD: Micro SD para sistema operativo (Raspbian)• Recursos Energéticos: 5V DC via USB-C/GPIO com 3A

Tabela 4.3: Especificações do Raspberry Pi 4 modelo B [60]

Existem diversas versões deste modelo, nomeadamente em relação à quantidade de memória RAM (*Random Access Memory*) utilizada, sendo que no AGV implementou-se um Raspberry com 4GB de RAM. Esta característica é bastante importante para o processamento de imagem, uma vez que possibilita um *multitasking* com melhor performance, o que é essencial, visto se estar a realizar em simultâneo o processamento de imagem e a comunicação externa com o utilizador, o que exige um maior uso da RAM. Deste modo, esta RAM extra serve como um serviço de cache de arquivos, acelerando os dados frequentemente acedidos que, de outra forma teriam que viajar através do cartão microSD, o que é relativamente mais lento. Por isso, com uma maior RAM consegue-se auxiliar o serviço da GPU, que é a unidade de processamento gráfico, altamente requerida para o processamento de imagem, que compartilha a sua memória com a CPU (*Central Processing Unit*), responsável pelo processamento de dados através de algoritmos aritméticos. Para garantir a sua melhor performance, deve-se garantir que o fornecimento energético é efetuado de forma correta, não expondo o Raspberry Pi 4B a um défice energético. Tal situação leva a uma redução da frequência de operação do processador e como consequência este teria uma performance mais baixa, que influencia diretamente no processamento de imagem. Por outro lado, como se alimenta o Raspberry Pi pelos GPIO2 e GPIO6, respetivamente VCC e GND, e como não existe nenhum circuito de proteção nestes pinos contra uma tensão acima das tolerâncias tabeladas, então a aplicação de uma tensão contínua acima de 5.25 V (tolerância máxima aceite), levaria a danos permanentes na placa e, conseqüente uma futura falha em todo o sistema.

4.2.5 Câmera Digital

O processamento de imagem em tempo real é uma das principais características deste AGV e, portanto, é necessário incluir um sistema de captação de imagem policromático que tenha capacidade de adquirir e transmitir imagens com uma qualidade razoável, ao nível de distinguir pequenos pormenores do meio envolvente, nomeadamente os sinais no piso e a própria guia, como se verá na seção 4.3. Também é essencial que seja um sistema que possua alta performance para transmitir as imagens a uma elevada taxa de quadrados por segundo (FPS), não limitando os quadros processados (*frames*) pelo firmware durante o processamento de imagem. Garantido estes requisitos, integrou-se um módulo de câmara digital dos mesmo fabricante da unidade processadora, e por isso incorporou-se a Raspberry Pi Camera V2 possuindo uma total compatibilidade com o Raspberry Pi 4B e com as suas versões anteriores, conectando-se por uma porta CSI (*Camera Serial Interface*) ao Raspberry Pi, que ergonomicamente é capaz de transmitir dados a uma elevada taxa, transportando exclusivamente dados de pixels diretamente para o processador, maximizando o número de quadros por segundos disponíveis para análise. É um módulo com a capacidade de captura de vídeo full-HD até 30 quadros por segundo, com uma boa conservação das cores e baixo ruído no ato de aquisição, integrada num design pequeno e leve (3 gramas) com uma baixa diafonia [61]. As características técnicas deste módulo de baixo custo encontram-se na tabela 4.4.

Raspberry Pi câmara V2	Especificações
	<ul style="list-style-type: none">• Resolução: 8 Megapixels nativos• Modos de Vídeo: 1080p30, 720p60 e 640 x 480p60/90• Drivers: V4L2 driver• Sensor: Sony IMX219 com 3280 x 2464 pixels• Área do sensor: 3.68 x 2.76 mm• Tamanho de cada pixel: 1.12 µm x 1.12 µm• Tamanho Ótico : 1/4"

Tabela 4.4: Especificações do Raspberry Pi câmara V2 [62]

Para além disso, é uma câmara digital com uma profundidade de campo ajustável, influenciada pela distância focal de 3.04 mm do sensor da SONY implementando, responsável por gerir o quão o sistema ótico converge ou diverge a luz, o que possibilita maiores ou menores aproximações na imagem adquirida, provocando uma perda ou ganho de nitidez, respetivamente, no ato de focar os elementos no piso. Conforme a distância focal e o tamanho do sensor, forma-se um campo de visão (FOV - *Field of View*), que é a área máxima de uma amostra que a câmara pode capturar, onde horizontalmente consegue ter um FOV de 62.2 graus, mas verticalmente é mais limitado possuindo apenas 48.8 graus, dando origem a uma área de captura retangular. A área de captura é proporcional à razão focal, que é um número adimensional que indica a relação entre a distância focal e o diâmetro da abertura da câmara. Esta câmara tem uma distância focal de 2.0, o que promove um maior campo de visão, mas um menor poder de ampliação [63]. É de notar, que o tamanho do sensor é determinado pelo número de pixels do sensor e pelo tamanho dos próprios pixels, sendo que neste caso pixels de 1.2 µm por 1.2 µm são pequenos, beneficiando-se de uma maior resolução espacial, mas com uma menor sensibilidade à luminosidade, o que se torna benéfico no ato de aquisição, visto tornar-se menos volátil a diferentes iluminações ao longo dos percursos.

4.2.6 Sistema de Iluminação

Para auxiliar na aquisição de imagem existe um sistema de iluminação direcionado para o piso, de modo a proporcionar condições homogêneas de aquisição, bem como um bom destacamento da guia e dos sinais integrados no piso, não estando assim depende da iluminação do meio envolvente, que à mercê desta iria proporcionar diferentes condições de aquisição que influenciariam no processamento das imagens captadas. Deste modo, garante-se uma constante iluminação através um único LED de 3 W, sendo que um LED é um díodo semiconductor de junção P-N, quando energizado emite radiação eletromagnética, que neste caso será no espectro da luz visível. A energização do LED leva à eletroluminescência, que é o fenômeno elétrico, na qual um material emite luz devido à passagem de uma corrente elétrica. O LED integrado promove a emissão de luz com a tonalidade branca, sendo que fisicamente, esta tonalidade é conseguida pela utilização de emissores de cor azul revestidos por uma camada de fósforo, utilizado comumente em lâmpadas fluorescentes, que através da absorção da tonalidade azul permite-se apenas a emissão da tonalidade branca. A escolha deste género de sistema de iluminação deve-se ao seu baixo preço, alto rendimento, durabilidade e diversidade em termos de design, o que permite integrar facilmente no AGV construindo-se um sistema que se adapte às suas necessidades.

Este sistema de iluminação (2) encontra-se integrado na parte dianteira do AGV, juntamente com a câmara (1) descrita da seção anterior. É de notar, que a câmara se encontra paralela ao piso, o que permite realizar a aquisição de imagens sem qualquer ângulo associado, não existindo a distorção dos objetos capturados, o que permite realizar um processamento de imagem mais preciso. Porém, o sistema de iluminação não preserva este design, integrando-se antes com um ângulo de 45 graus. A vantagem desta disposição é reflexão da luz incidente no piso, que ao refletir neste, reflete com o mesmo ângulo de incidência, o que impede de existirem reflexões em direção à câmara. Ao prevenir-se este comportamento, impede-se o aparecimento de manchas brancas na imagem capturada, sendo estas manchas apenas ruído prejudicial a todo o processo de aquisição. Ganha-se esta característica, porém, uma parte da luz incidente é dispersa e, por isso a luminosidade na zona de aquisição diminui de forma proporcional, quanto maior for o ângulo entre o sistema de iluminação e o piso. Energeticamente, a energização do LED advém da ponte H (4), através de uma saída de 5 V constante, o que promove mínimas oscilações na luminosidade sentida no processo de aquisição promovendo uma iluminação constante. Este sistema é independente do Raspberry Pi 4B (3), ligando-se apenas durante o estado de funcionamento do AGV, estando destacado na figura 4.30.

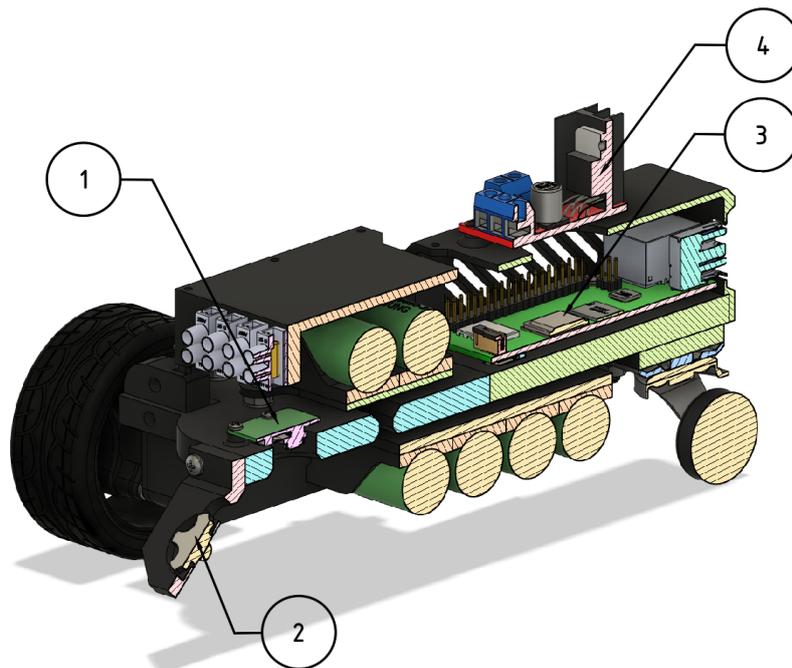


Figura 4.30: Arquitetura do sistema de aquisição de imagem

4.2.7 Ponte H

Uma ponte H é um circuito elétrico responsável por mudar a polaridade de uma tensão aplicada numa carga indutiva. Neste sistema, a carga indutiva são dois motores elétricos o que possibilita a capacidade de alterar o sentido de rotação destes, através de quatro elementos de comutação configurados, como ramos da letra “H”, onde cada motor é conectado de forma transversal a esta configuração viabilizando assim o seu controlo. Integrou-se uma solução comercial para o controlo motriz do AGV, o L298N que é um circuito integrado monolítico com um chip L298 (1) que integra duas pontes H, projetado para operar com níveis lógicos TTL (*Transistor – Transistor Logic*), ou seja, utiliza-se uma classe de transístores de junção bipolar com o propósito de funcionarem como portas lógicas e, como amplificadores, ao invés de componentes onde a comutação é realizada de forma mecânica. Deste modo, beneficiam-se de maiores velocidades de comutação (cerca de 200 ns) e melhores eficiências energéticas, visto serem acionados com correntes na base dos transístores em tornos dos miliamperes. Esta lógica possibilita a condução de até 2 A por cada ponte, limitando-se esta placa a uma potência máxima de até 25 W [64].

A alimentação do L298N acontece através dos pinos (3), onde estes são respetivamente o VCC, GND e o terminal de 5 V que poderá funcionar como input ou output. No caso de servir de output, a tensão de saída é garantida por um regulador de tensão interno, o 78M05 que independentemente da tensão de alimentação realiza esta conversão fixa conduzindo até 0.5 A. Este terminal apenas funciona deste modo, caso a tensão de alimentação da placa seja superior à tensão mínima de alimentação (5 V), mas abaixo de 12 V, porque caso seja acima deste valor, é necessário retirar o jumper integrado em (2), para viabilizar uma alimentação até 35 V, que é a tensão máxima suportada pela placa. Só que ao ganhar-se esta amplitude em termos de energização, perde-se a capacidade de gerar 5 V de saída e consequentemente desabilita-se a alimentação interna do chip L298, sendo necessário obrigatoriamente uma fonte de alimentação externa, que agora utilize o terminal de 5 V, como input (inviabilizando a sua utilização como uma saída) para a alimentação do circuito lógico e controlo da placa. Portanto, ao garantir uma tensão de alimentação fixa menor que 12 V promove dois benefícios para o conjunto total. O primeiro benefício é a utilização do terminal de 5 V como output, o que permite a sua utilização como fonte de alimentação do sistema de iluminação. Para além disso, poupa-se a integração de um outro regulador de tensão externo, para energizar o circuito interno da placa, através do terminal de 5 V, que serviria neste caso como input obrigatório para o funcionamento da placa. O segundo benefício é a exposição a uma tensão constante nos motores elétricos, através dos terminais (4) e (5), uma vez que a tensão sentida nos terminais de alimentação é a mesma sentida nestes, com uma diferença de 2 V devido às quedas de tensão internas. Deve-se assim manter uma tensão constante à entrada da placa, uma vez que, se esta varia conforme a capacidade energética da bateria, e como a tensão é diretamente proporcional à potência produzida pelos motores, então o comportamento motriz do AGV estaria dependente da capacidade energética da bateria, proporcionando diferentes características mecânicas advindas do AGV.

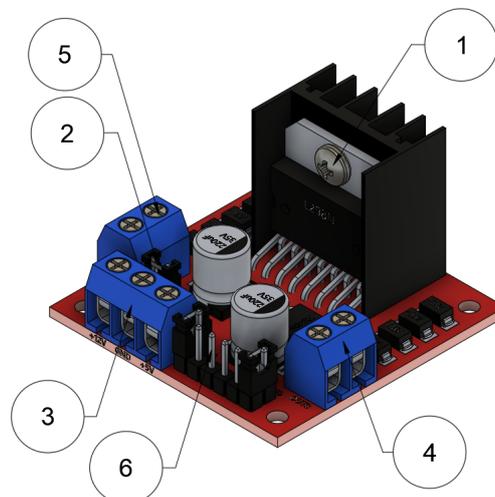


Figura 4.31: L298N

O controlo do L298N acontece a partir das portas programáveis do Raspberry Pi (GPIO) através do envio de dados processados por firmware, fazendo-os chegar ao longo dos terminais na parte frontal da placa, ao longo do porto de controlo indicado em (6). Neste porto existem dois tipos de terminais: os referentes ao controlo do sentido de rotação dos motores e os referentes à velocidade desta rotação. Os terminais responsáveis pelo sentido de rotação dos motores elétricos apenas possuem dois estados possíveis: HIGH ou LOW. A combinação destes estados binários resulta em quatro condições teóricas, uma vez que existem dois terminais para cada ponte, totalizando em quatro terminais no total (IN1, IN2, IN3 e IN4) e portanto, como a ordem dos valores binários dos terminais influencia no comportamento dos motores e como pode existir a repetição de estados, matematicamente geram-se arranjos completos que resultam em quatro possibilidades. Porém, existem duas combinações que promovem fisicamente o mesmo comportamento dos motores, independentemente da ordem lógica dos terminais e, por isso, ao invés de quatro estados possíveis, existem na prática apenas três por cada ponte H, nomeadamente os motores a rodar com o sentido horário, anti-horário, e a paragem destes devido aos dois arranjos distintos que igualmente promovem o mesmo comportamento.

Estes arranjos binários influenciam o modo de operação da ponte H aos quais estão associados, através da presença ou ausência de corrente na base dos transístores integrantes desta, resultando em combinações internas que determinam o modo, como os motores se comportam. É de notar, que cada ponte H é constituída por quatro transístores de junção bipolar, dois PNP (T1 e T3) e dois NPN (T2 e T4), de modo que, para que um transístor NPN conduza corrente é necessário que exista uma queda de tensão positiva entre a base e o emissor superior a 0.7 V, visto os transístores do chip L298N serem compostos de silício. Graças a esta queda de tensão positiva que polariza diretamente a junção base – emissor, leva a que a junção base – coletor se encontra inversamente polarizada, o que promove o movimento dos eletrões, no sentido coletor – emissor, resultado assim na condução de corrente. Porém, no caso dos transístores PNP, o funcionamento é o oposto, uma vez que para este tipo de transístor conduza corrente, é necessário que a queda de tensão entre a base e o emissor seja negativa (-0.7 V), de modo que a junção base – emissor se encontre inversamente polarizada.

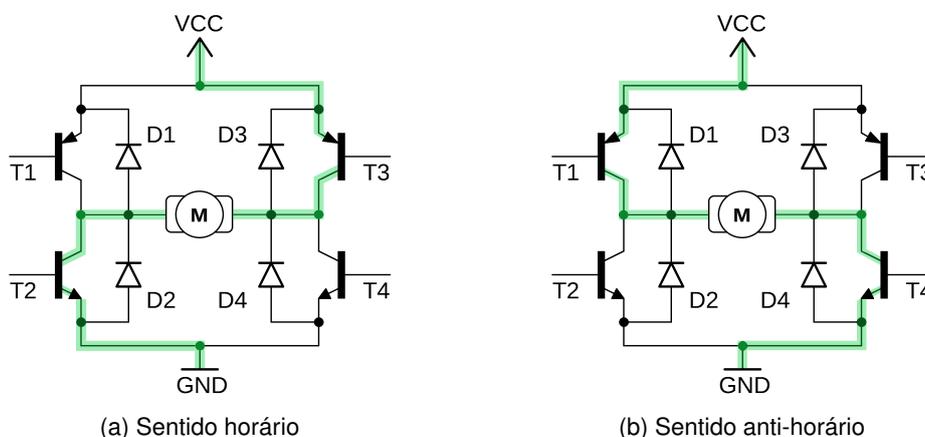


Figura 4.32: Alteração da polaridade aplicada no motor elétrico

Portanto, cada terminal de controlo IN é responsável por dois transístores de cada tipo, estando estes em série, o que significa que quando um terminal de controlo se encontra a HIGH, então o transístor NPN encontra-se ativo, enquanto o transístor PNP encontra-se inativo. Analogamente, quando o terminal de controlo se encontra a LOW, o comportamento é o oposto, estando o transístor NPN desativo e PNP ativo. O facto de não existir transístores do mesmo tipo em série previne a ocorrência de curtos-circuitos, uma vez que, se T1 e T2 ou T3 e T4 se encontrem simultaneamente a conduzir, então formar-se-ia um caminho de baixa resistência, que ignoraria o motor elétrico devido à sua resistência interna, completando o circuito com o GND sem qualquer carga, resultando em dados físicos para a ponte H. Evita-se estas combinações, através da arquitetura evidenciada na figura 4.32, garantindo-se que apenas um dos dois transístores de cada extremidade da ponte se encontre ativo durante o controlo, complementando o circuito através da passagem de corrente pelo motor e, consequentemente a ligação com o segundo transístor da extremidade oposta [65].

Na prática, este modo de operação leva aos motores rodarem no sentido horário ou anti-horário, através da alteração do sentido do fluxo da corrente elétrica, como é evidenciado na figura 4.32, que é controlado pela ordem e pelo valor lógico dos terminais de controlo (HIGH e LOW ou LOW e HIGH), respetivamente. O último estado dos motores é a paragem destes, que é conseguido pela partilha idêntica do mesmo valor lógico imposto nos terminais de controlo, ou seja, se ambos estiverem a HIGH ou a LOW, simultaneamente, impede-se que existe um fluxo de corrente ao longo da ponte H e assim, os motores ficam parados, perante o controlo imposto. Chama-se à atenção que o comportamento de uma ponte H é independente da outra ponte H, mesmo pertencentes ao mesmo chip, onde cada uma é controlada pelos respetivos terminais de controlo, proporcionando a possibilidade de existir distintos comportamentos motrizes em cada motor do AGV. Estes motores eletricamente comportam-se como um indutor, que ao formar o seu campo magnético produz uma indutância, com uma resistência interna, visto na prática nunca serem ideais. Como qualquer indutor, quando este para de ser eletricamente alimentado, a energia armazenada nele necessita de dissipar, visto não existir condições para que esta fique armazenada, e no caso da ponte H, se não existissem os díodos em paralelo aos transístores, gerar-se-ia um pico de tensão que gerava danos físicos nos transístores até à sua destruição. Estes díodos em paralelo (D1, D2, D3 e D4) são designados por díodos *Flyblack*, utilizados para proteger a ponte H de uma carga indutiva (motor), onde esta é controlada por elementos de comutação. Evita-se assim, que o indutor gere um pico de tensão repentino, causado por uma redução drástica ou total na sua corrente alimentação, que é direcionada para estes díodos, visto os transístores quando inativos não conseguirem conduzir corrente, e por isso são um obstáculo a esta energia que necessita dissipar. Deste modo, a tensão gerada pela carga indutiva é significativamente menor, não expondo os transístores a tensões acima das suas tolerâncias elétricas.

Para além disso, cada ponte H possui respetivamente um terminal responsável pelo controlo da velocidade de rotação dos motores, totalizando no conjunto em dois terminais (ENA e ENB) que através de PWM (*Pulse Width Modulation*) efetuam esta regulação. O PWM é um método que visa reduzir a potência média entregue, neste caso aos motores elétricos, por um sinal elétrico (sinal digital) através da comutação entre dois estados binários: estado de alimentação nominal (ON) e o estado que provoca a ausência total da energização (OFF). A alternância entre estes estados forma o *duty cycle* ou ciclo de trabalho, que descreve a relação a partir de uma onda quadrada, entre o tempo em que os motores estão ligados, e o tempo que os motores estão desligados. A relação entre estes tempos provoca a alteração da tensão média a que se expõe os motores e consequentemente, promove-se o controlo da sua velocidade, através da diminuição ou aumento da tensão média sentida nos motores [65]. De forma proporcional, quando maior for o tempo ON maior é a tensão média, que em oposição, quando maior é o tempo OFF, menor é a tensão média, como é evidenciado na figura 4.33. Assim, este método é particularmente adequado para o controlo da velocidade de rotação dos motores, uma vez que os motores ao serem uma carga inercial, do ponto de vista mecânico, leva-os a reagir lentamente a esta comutação binária devido à sua inércia. Para melhorar esta reação, a velocidade de comutação entre o estado ON e OFF necessita de ser elevada (frequências em torno dos kHz), para não afetar de forma expressiva a carga que se está a controlar, tornando as transições entre diferentes ciclos de trabalho o mais suave possível [66].

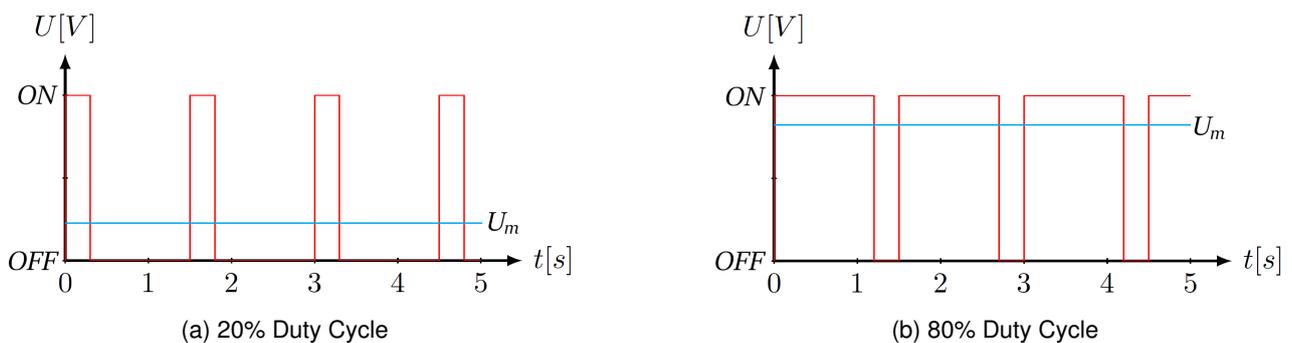


Figura 4.33: Variação da tensão média (U_m) com diferentes ciclos de trabalho

4.3 Firmware

Em computação, o firmware é um tipo de software dedicado ao controlo de hardware desde o baixo nível e portanto, no AGV, este controlo estende-se através do fornecimento de instruções de inicialização, levando ao arranque de um sistema operativo que serve de interface entre o Utilizador e o AGV, permitindo a assimilação do hardware mecânico e elétrico através do seu controlo computacional, resultando na formação de um sistema embebido com capacidades de tomada de decisão autónomas, sem intervenção externa no seu controlo. Neste sistema operativo, é integrado um conjunto de *scripts* que permitem gerir o controlo de toda a arquitetura mecânica, através dos vários subsistemas elétricos, em função do processamento de imagem em tempo real realizado pelo Raspberry Pi 4B, otimizando a utilização dos seus recursos computacionais direcionados às ferramentas de visão artificial utilizadas, módulos do OpenCV, juntamente com uma linguagem de programação de alto nível orientada a objetos (Python), construindo-se de raiz todo o software de controlo e tomada de decisão do AGV. Todo este controlo e processamento pode ser monitorizado através de um protocolo de comunicação sem fios com o utilizador, permitindo tornar o AGV num sistema com capacidades de conexão móvel, acedido através de um IP estático (*Internet Protocol Address*) definido, auxiliando-se de uma única chave de acesso, para garantir o estabelecimento de uma ligação segura.

4.3.1 Sistema Operativo

O Raspberry Pi 4B conta com um sistema operativo dedicado: Raspberry Pi OS que é um sistema baseado em Debian (Linux) otimizado para este, tornando-o num sistema rápido e estável. É um sistema completo *open-source* gratuito, com inúmeros módulos e funcionalidades necessárias ao funcionamento do AGV [67]. Existe a possibilidade de instalar outros sistemas operativos, porém neste caso, a integração do Raspberry Pi OS garante todas as valências necessárias ao bom funcionamento do AGV, oferecendo algumas características bastante aliciantes a um trabalho do tipo POC (*Proof of Concept*), contrastando com a instalação do sistema operativo mais comum (Windows) no Raspberry Pi, que para além de uma menor performance generalizada, é um sistema operativo pago e legalmente impossibilita a distribuição ou aplicação de alterações no código fonte sem uma permissão, visto ser um sistema operativo com um proprietário fixo [68].

A instalação do Raspberry Pi OS acontece fisicamente num cartão microSD externo, que é um cartão de memória não volátil, através por exemplo da aplicação *Pi Imager* com recurso a um computador externo. Esta aplicação permite não só instalar o OS no cartão microSD por meio da limpeza de todo o seu conteúdo, como permite escolher qual a versão e o tipo de sistema operativo que se pretende instalar dentro dos pacotes disponíveis. A instalação do sistema operativo num cartão de memória externo apenas acontece, porque o Raspberry Pi não possui memória interna, e por isso a necessidade de integrar um cartão de memória externo que opere como esta. Face a esta característica, o cartão microSD deve ter a capacidade de albergar o espaço necessário para incorporar o sistema operativo completo, mais um espaço auxiliar para incluir futuros programas instalados. Porém, existem limites em relação à capacidade máxima destes cartões, visto que o Raspberry Pi 4B apenas suporta cartões com uma capacidade máxima de 32 GB inclusive [69]. Esta limitação acontece, uma vez que estes cartões são formatados através de um sistema de arquivos exFAT ou FAT64 (*File Allocation Table*) utilizados na organização de dados em discos de memória *flash*. Esta limitação resulta do *bootloader* do Raspberry Pi 4B funcionar apenas com cartões formatados em FAT16 ou FAT32, onde o número em sufixo indica a número de bits utilizados. Contudo, é possível utilizar um cartão de maior capacidade, mas é necessário que este seja formatado segundo um dos formatos anteriores (FAT16 ou FAT32), levando a uma perda virtual da capacidade real do cartão, embora fisicamente exista mais memória disponível.



Figura 4.34: Cartão microSD utilizado

Fora a capacidade da memória interna do cartão microSD, também é necessário ter em atenção às velocidades de escrita e leitura do cartão, uma vez que a velocidade de um cartão microSD afeta diretamente a velocidade de execução do Raspberry Pi, da mesma forma que a velocidade de um disco rígido afeta a velocidade de execução de um computador convencional. Quanto mais rápido se conseguir ler os dados na memória interna do cartão microSD, mais rápido o Raspberry Pi inicializa os seus programas. De igual forma, a velocidade de escrita também afeta a execução de qualquer programa, devido ao volume de dados gerados, neste caso pelo processamento de imagem, e por isso é importante garantir-se a utilização de um cartão microSD que não limite as velocidades de execução praticadas pelo Raspberry Pi. Tendo em avaliação estas particularidades, integrou-se um cartãoSD com 32 GB de memória interna garantindo o espaço mínimo de 16 GB que este sistema operativo recomenda em relação à capacidade total do cartão utilizado. Como o Raspberry Pi 4B possui uma velocidade máxima de leitura de até 50 MB/s, perfazendo uma média aproximada de 40 MB/s ao longo da sua operação normal, de modo a não limitar esta velocidade de execução, o cartão integrado possui uma velocidade de leitura de até 90 MB/s e de escrita até 60 MB/s, possuindo uma performance categorizada na classe A1 garantindo no mínimo 1500 operações de leitura e 500 operações de escrita por segundo, suportando com folga as velocidades praticadas pelo Raspberry Pi 4B [70]. Ainda é um cartão com o padrão UHS com uma velocidade de execução classe 3 (U3), permitindo otimizar a execução de transmissões em tempo real e a captura de vídeos em Full HD beneficiando o processo de aquisição do AGV.

Ao contrário de um computador convencional, a inicialização do Raspberry Pi OS acontece de modo diferente, devido a algumas diferenças tanto ao nível do hardware, como do firmware. O sistema operativo é inicializado assim que o AGV se encontra ligado, levando nesse instante à inicialização do núcleo da GPU, enquanto o processador ainda se encontra desligado. Com a GPU a funcionar, começa-se a executar o *bootloader*, que é o firmware responsável pela ativação do sistema operativo, estando armazenado numa memória não volátil ROM, que persiste mesmo quando o sistema se encontra desligado. Ao carregar-se o *bootloader*, inicia-se a leitura do cartão microSD, que viabiliza a ativação da memória RAM, que por sua vez permite a interpretação do firmware de inicialização e assim, inicia-se o sistema operativo. Independentemente do tipo de sistema operativo instalado no cartão microSD, ao contrário de computadores convencionais, o Raspberry Pi 4B não possui BIOS (*Basic Input/Output System*), que é um firmware responsável pela inicialização do sistema operativo, possibilitando um conjunto de configurações prévias antes da inicialização do sistema operativo, como a configuração de novas interfaces e permitir a escolha da ordem de inicialização do sistema. Sem este firmware, o Raspberry Pi é obrigado a inicializar sempre a partir do cartão microSD, mesmo que se tenha à disposição um outro tipo de memória com o sistema operativo lá instalado.

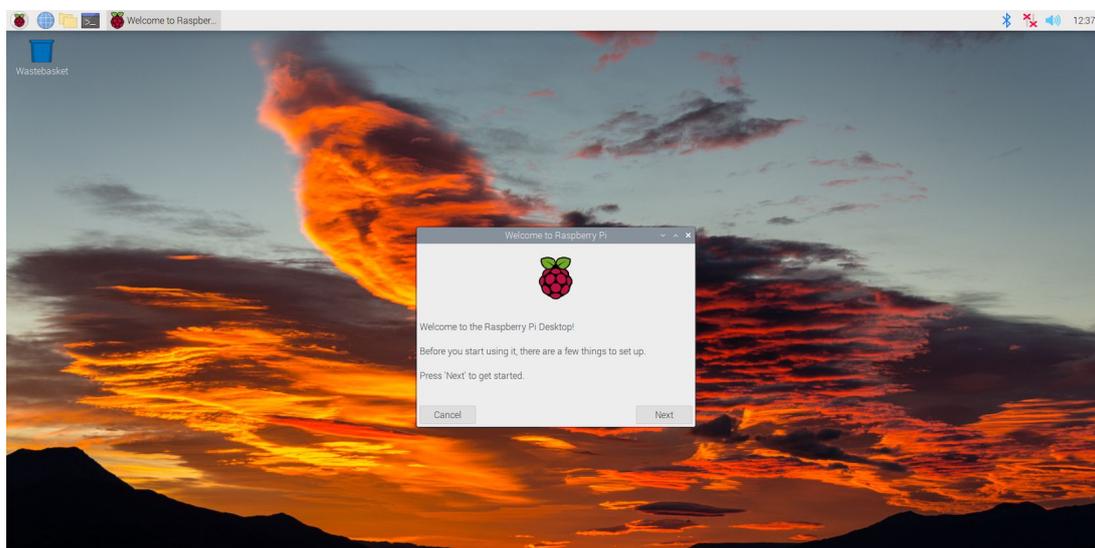


Figura 4.35: Raspberry Pi OS

A ausência de BIOS no Raspberry Pi apenas se deve, porque não existe um RTC físico (*Real-Time Clock*) integrado nele [68]. O RTC responsabiliza-se por gerir o tempo que passa dentro do sistema operativo, independentemente do estado de funcionamento do sistema, devido a uma bateria independente do sistema de alimentação principal. Em vez da integração física deste dispositivo, o Raspberry Pi assim que inicializa o seu sistema operativo, quando se conecta à Internet utiliza um servidor global NTP (*Network Time Protocol*), como RTC virtual realizando assim a gestão do tempo. Esta abordagem em conjunto com a ausência de outros elementos tanto ao nível do hardware, como do software justificam-se para manter um custo baixo em relação à placa desenvolvida, apresentado em muitos dos casos soluções alternativas que resolvem essas ausências. Face a estas características, instalou-se a versão 11 do Raspberry Pi OS de 32 bits de cognome “Bulleyes”, que através de um *autologin* (login automático) permite a inicialização do sistema operativo, sem a necessidade de se introduzir nomes de utilizador, nem chaves de acesso para aceder ao sistema, via local [71]. Contudo, como se verá na seção seguinte, para aceder via Internet é necessário estas credenciais de acesso, de forma a assegurar a proteção da ligação AGV-Utilizador.

4.3.2 Comunicação AGV-Utilizador

A indústria 4.0 molda a tendência atual em termos de ideologias a implementar em processos de fabrico e produtos oriundos desses processos, através da crescente automação das várias tecnologias e a adoção de sistemas de comunicação que proporcionam a partilha de dados ao longo de uma rede interna com acessos remotos possibilitando o ato de arquivar dados referentes ao sistema em qualquer instância garantindo a interoperabilidade através de protocolos de cibersegurança. Estes sistemas com estas capacidades formam sistemas ciber-físicos (*Cyber Physical System*), que na sua génese são sistemas com capacidades computacionais cooperativas com o propósito de controlar processos físicos [72]. O AGV embora um sistema embebido na sua essência, também se considera como um sistema ciber-físico em grande parte devido às múltiplas interfaces de comunicação que permitem o tráfego de dados segundo um protocolo de comunicação local ou sem fios. Esta extensa capacidade de comunicação permite dotar o AGV, como um sistema que vai ao encontro da demanda tecnológica vivida nos tempos atuais.

Por isso, a comunicação que se estabelece entre o AGV e o Utilizador oferece inúmeras vantagens, como a inicialização gráfica e acesso ao sistema operativo completo, controlo e monitorização de dados advindos do processamento de imagem em tempo real, recolha de informações sobre o estado global do AGV e modificação do código referente a todo o processo de gestão de rotas e algoritmos de decisão implementados. Dado a natureza mecânica do AGV, a portabilidade deste não pode ser limitada por um protocolo de comunicação local, como um cabo Ethernet e por isso, é necessário garantir um protocolo que estabeleça uma conexão sem fios que suporte o tráfego de dados necessário para o estabelecimento de uma boa conexão. Como forma de corresponder a este requisito estabeleceu-se um protocolo de comunicação Wi-Fi, através de uma rede local sem fios LAN (*Local Area Network*) definida pelo padrão de comunicação IEEE 802.11. Esta rede local forma uma área limitada pela capacidade de emissão do ponto de acesso, que através de ondas rádio estabelece um tráfego de dados com os dispositivos conectados a ele, que por meio de uma *gateway* permite estabelecer uma conexão entre estes e a Internet [73]. Quanto mais afastado do local de emissão o AGV estiver, ou no caso de existirem objetos sólidos que restringem a propagação do sinal, então existirá uma menor largura de banda para o tráfego de dados o que viabiliza maiores latências durante o processo de troca de dados. Latências mais elevadas significam que o tempo entre a emissão do pacote e a sua receção é mais elevado, o que em processos com controlo e monitorização em tempo real, como é o caso do AGV pode ser uma grande desvantagem principalmente na deteção e resolução de problemas técnicos. Uma forma de conservar a qualidade do sinal ao longo dos percursos do AGV e garantir uma latência mais baixa é expansão do sinal, através de uma rede de pontos de acesso estrategicamente colocados para otimizar a expansão da área útil coberta por estes dispositivos, oferecendo uma maior capacidade ao AGV de deslocar-se livremente pelos seus percursos com a garantia que a qualidade do seu sinal é conservada praticando um tráfego de dados com baixas latências.

Este tráfego de dados apenas acontece diretamente entre o AGV e o Utilizador, caso ambos estejam conectados na mesma rede local. Contudo, é possível por meio da modificação de hardware e principalmente do firmware do ponto de acesso, estabelecer uma conexão entre o AGV e um Utilizador de uma LAN diferente da qual o AGV está conectado. Neste sistema não se realizou essas modificações e por isso, a obrigatoriedade de ambos estarem conectados na mesma rede local, para o estabelecimento de uma ligação remota. Este acesso remoto por parte do Utilizador não é conseguido diretamente, uma vez que é necessário um conjunto de interfaces e modificações ao nível do software, mais o conhecimento das credenciais de acesso para efetuar esta ligação. Para além disso, para assegurar a ligação é necessário que o(s) ponto(s) de acesso tenham ligação à Internet, pois sem esta conexão os *routers*, dispositivos que funcionam como pontos de acesso, não conseguem realizar o tráfego de pacotes e consequentemente direcionar os IPs dos serviços para os respetivos clientes. Assim, de acordo com a figura 4.39 a ligação entre o AGV e o Utilizador segue as seguintes entidades computacionais até ao estabelecimento de uma ligação sem fios.

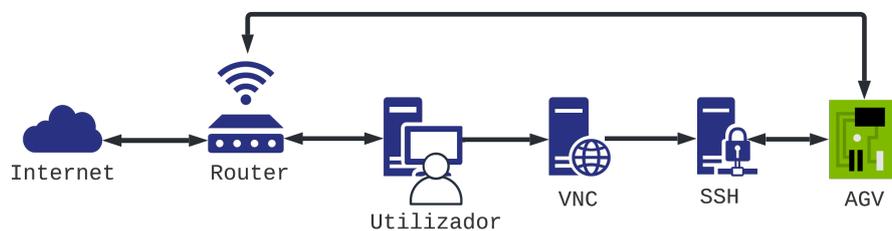


Figura 4.36: Rede envolvente para a conexão AGV-Utilizador

A ligação remota implementada é do tipo unidirecional, ou seja, é uma ligação que apenas se estabelece num sentido, onde o Utilizador consegue ter acesso e controlo sobre o AGV, mas o AGV já não consegue usufruir destas aptidões de comunicação e controlo. Seria incoerente que esta ligação fosse uma ligação bidirecional, uma vez que o AGV não deve controlar ou ter acesso sobre os dados arquivados no dispositivo que o Utilizador utiliza, visto não acrescentar nenhuma característica vantajosa ao processo, como tornar-se numa ligação mais suscetível a ataques informáticos, uma vez que o AGV teria a capacidade de controlar o dispositivo do Utilizador e ter acesso aos seus dados privados. Nesta situação hipotética, ataques informáticos ao AGV permitiriam ter um acesso facilitado ao dispositivo do Utilizador com consequências como a extração de informação sensível e ataques de *malware* que danificariam o sistema. Dificulta-se esta vulnerabilidade através de uma ligação unidirecional, tornando estes ataques mais difíceis, uma vez que o dispositivo do Utilizador não está preparado para interpretar e responder a comandos advindos do AGV, formando-se assim uma proteção passiva perante ataques desta natureza.

Face a estas características, o estabelecimento desta ligação inicia-se pela utilização de um sistema VNC (*Virtual Network Computing*) que é um sistema gráfico que permite a partilha da área de trabalho do AGV com acesso às suas funcionalidades, a partir de um protocolo de comunicação na qual um dispositivo externo com acesso às credenciais de acesso permite realizar qualquer ação remota, como se estivesse localmente no AGV. O sistema VNC opera nos dois dispositivos de forma distinta, uma vez que no AGV corre o servidor VNC que é o software responsável por permitir o acesso e o controlo às suas funcionalidades, enquanto no dispositivo do Utilizador corre o cliente VNC que é o programa responsável por representar graficamente os dados provindos do AGV e todo o tipo atualizações e ações que ocorrem neste [74]. Para além disso, é um serviço nativo que já se encontra integrado no Raspberry Pi OS e por isso, basta ativá-lo para que fique disponível para realizar este tipo de conexão, destacando-se também pela sua independência face à plataforma onde opera permitindo ainda ter múltiplas conexões, levando à possibilidade de existirem vários clientes (Utilizadores) para um único servidor (AGV). Repare-se que nesta situação apenas um dos Utilizadores tem acesso total ao controlo do AGV, enquanto os restantes apenas usufruem da visualização do que ocorre na área de trabalho deste.

Por defeito, esta ligação acontece segundo um protocolo RFB (*Remote Frame Buffer*) que é um protocolo que atua ao nível do *Frame Buffer* permitindo atuar em qualquer sistema operativo. O *Framebuffer* é uma porção de memória RAM que contém um *bitmap* responsável pela reprodução e exibição de vídeo [75]. A exibição deste conteúdo de forma remota acontece através de um protocolo de comunicação TCP, através da reserva de uma porta (5900 normalmente) como ponto de ligação, embora seja possível configurar outras portas com a mesma finalidade. Na prática não é um protocolo seguro, visto aplicar mecanismos de autenticação relativamente fracos aos dias atuais, não definindo mecanismos de criptografia especiais para salvaguardar a segurança das sessões entre o cliente e o servidor, e assim não garantindo o sigilo desta ligação expondo-a mais facilmente a ataques de espionagem e furto de informações [76]. Embora ao longo do tempo se venha a implementar melhorias e atualizações que aperfeiçoam este protocolo nativo em vários serviços VNC, este ainda é um protocolo de comunicação facilmente suscetível a ciberataques visto as suas limitações ao nível da segurança. Um problema grave a este nível segundo [77] é a definição de *passwords* até 8 caracteres, o que nos dias atuais é um número relativamente pequeno de bits tornando-o num sistema suscetível a ataques de "força bruta" que com a ajuda de dicionários pré-computados aumenta a eficácia destes ataques [78]. Mesmo que se estabeleça uma *password* superior a 8 caracteres, os caracteres em excesso são truncados.

Devido às fragilidades deste protocolo ao nível da cibersegurança que torna a ligação entre o AGV e o Utilizador instável do ponto de vista da manutenção da privacidade e segurança da conexão, substituiu-se este protocolo por um novo, o SSH (*Secure Shell Protocol*). Este é um protocolo de rede criptográfico que opera em redes LAN hospedando a operação de serviços conforme um modelo cliente-utilizador através da autenticação dos dois a partir da criptografia dos dados transmitidos ao longo da ligação. Segundo este protocolo, o Utilizador deve iniciar a conexão através de um "handshake TCP", ou seja, o reconhecimento de tanto o AGV, como do Utilizador que ambos estão prontos para iniciar uma ligação via o protocolo de comunicação TCP, garantindo uma ligação simétrica com a utilização dos mesmos padrões de criptografia verificando em seguida se a identidade exibida pelo AGV corresponde aos registos de sessões anteriores, caso existam. Simultaneamente, verifica-se as credenciais de acesso do Utilizador, para autenticar a conexão e estabelecer uma ligação remota através do VNC encapsulado por este protocolo de rede conferindo todos estes atributos de cibersegurança. O protocolo SSH possui dois tipos principais de tecnologias de criptografia: criptografia simétrica e assimétrica, embora a criptografia simétrica seja a responsável por criptografar toda a comunicação durante uma sessão SSH, uma vez que é uma forma de criptografia em que uma chave secreta é usada para criptografar e descriptografar qualquer pacote enviado do Utilizador para o AGV. Esta chave secreta é conhecida tanto pelo Utilizador, como pelo AGV, pois esta deriva do método de criptografia assimétrico acordado na ligação inicial. Esta chave secreta nunca é divulgada a terceiros e também nunca é enviada durante a ligação entre o Utilizador e o AGV, o que torna num mecanismo seguro, pois mesmo que um dispositivo externo capture dados partilhados nesta ligação, ele não poderá calcular a chave secreta, porque não conhece o padrão de criptografia estabelecido inicialmente.

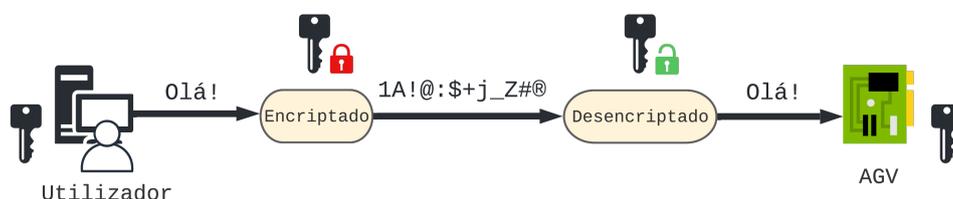


Figura 4.37: Criptografia simétrica entre o Utilizador e o AGV

A criptografia assimétrica não é utilizada para criptografar uma sessão SSH, pois esta atua somente durante o algoritmo de troca de chaves criptográficas simétricas entre o AGV e o Utilizador. Antes de se iniciar uma conexão segura, tanto o AGV, como o Utilizador geram chaves público-privadas temporárias com o intuito de as trocar entre si para formarem posteriormente uma chave secreta compartilhada [79]. É de notar que os mecanismos de criptografia são processos automáticos, na qual não é necessário interferir manualmente.

Para ativar o SSH juntamente com o VNC é necessário utilizar o PuTTY. Este é um emulador *open-source* que suporta o protocolo de rede SSH com capacidade de acesso remoto através da utilização do endereço IP do AGV, que por sua vez permite aceder aos serviços computacionais a partir da construção de túneis criptográficos mantendo a segurança da ligação entre este e o Utilizador. O estabelecimento desta ligação inicia-se pela procura do endereço IP do AGV, que localmente pode ser descoberto através do comando no terminal: `$ hostname -I`. Neste caso, o endereço IP é o `192.168.1.101` que caso seja utilizado em outra rede LAN para aceder via remota ao AGV não irá funcionar, porque este é um IP dinâmico privado atribuído numa rede local, não existindo compatibilidade nem a conservação deste endereço IP numa rede externa à qual este IP foi associado. O dinamismo deste IP é resultado da atribuição que o protocolo DHCP (*Dynamic Host Configuration Protocol*) realiza concebendo periodicamente e regularmente novos endereços IP dinâmicos segundo o padrão IPv4, porque aos dias atuais não existe um número suficiente de endereços IP estáticos disponíveis para utilizar em qualquer dispositivo conectado a uma rede e por isso a necessidade de utilizar endereços IP dinâmicos. O protocolo IPv4 forma endereços IP com 32 bits o que limita até a 2^{32} endereços estáticos disponíveis. Ao contrário destes, os endereços dinâmicos são ilimitados, pois podem ser reutilizados e por isso oferecer uma melhor camada de proteção, visto não estarem fixos a um único IP. Porém, a volatilidade dos endereços dinâmicos não é benéfica para o acesso remoto do AGV como um servidor, porque é necessário estar a procurar regularmente qual o IP dinâmico que este irá ter [80]. Nesta situação, a melhor solução é a definição de um IP estático privado para evitar estas mudanças de endereço, embora a configuração de um IP estático torne o AGV mais suscetível a ciberataques caso não se integre uma *firewall* que o proteja.

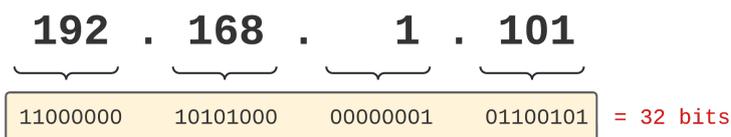


Figura 4.38: Decomposição de um endereço IPv4

A definição de um IP estático para o AGV é fundamental não só por uma questão de invariância deste, como ao ser um trabalho do tipo *proof of concept*, então na possibilidade futura de se formar frotas de AGVs com as características aqui descritas, é extremamente necessário atribuir endereços IP estáticos por uma questão de organização dos vários elementos da frota, eliminando flutuações que ocorrem ao nível dos endereços IP dinâmicos prescritos pelo protocolo DHCP. Portanto, para integrar um IP estático no AGV é necessário realizar duas operações ao nível deste. A primeira é descobrir qual o endereço IP da *gateway* que é o dispositivo intermediário que interliga redes. Para descobrir este endereço, tal como se realizou para o endereço IP dinâmico do AGV, basta aceder ao terminal e digitalizar o comando: `$ sudo route -n`. O resultado deste comando é a obtenção do endereço: `192.168.1.254` da *gateway* e a informação do tipo de conexão que o AGV estabelece com a rede, que neste caso é via Wi-Fi como é confirmado pela informação `wlan0`. A segunda operação é a configuração de comandos internos para permitir que o AGV seja identificado pela rede através de um endereço IP estático, a partir do comando no terminal: `$ sudo nano /etc/dhcpd.conf`. Neste diretório é necessário configurar o tipo de ligação que o AGV estabelece com a rede (sem fios ou local), o endereço IP estático que se pretende dotar o AGV, o endereço IP da *gateway* para que o AGV seja identificado pelo router e o endereço IP do sistema DNS (*Domain Name System*) que é um sistema responsável pela distribuição e gestão da nomenclatura de qualquer dispositivo ligado à rede. O tipo de ligação que o AGV estabelece com a rede, como já foi mencionado é do tipo Wi-Fi e por isso a inclusão do comando: `wlan0` para identificar esta característica. Na atribuição do endereço IP estático é necessário utilizar um endereço vazio que não esteja a ser utilizado por nenhum dispositivo nem pertencer à lista de endereços IP integrados pelo servidor DHCP. Neste caso, definiu-se: `192.168.1.150/24` como endereço IP estático do AGV, onde o sufixo `/24` é uma abreviatura da submáscara de rede `255.255.255.0`. A especificação do endereço IP da *gateway* e do sistema DNS normalmente é idêntica, uma vez que fisicamente são o mesmo dispositivo (*router*), de modo que o AGV recorre a este endereço IP, quando deseja enviar dados fora do domínio da sua LAN: `192.168.0`. O resumo destes comandos encontram-se na tabela 4.5.

```

interface wlan0
static ip_address = 192.168.1.150/24
static routers = 192.168.1.254
static domain_name_servers = 192.168.1.254

```

Tabela 4.5: Comandos alocados no diretório /etc/dhcpd.conf

Finalizado a configuração do endereço IP estático do AGV, este processo não é necessário repetir-se até que o Utilizador decida efetuar modificações ao nível do endereço IP. Deste modo, com a definição definitiva do endereço IP do AGV é necessário agora configurar o protocolo SSH para garantir o acesso remoto por parte do Utilizador. Para tal, inicia-se o PuTTY e insere-se o endereço IP estático já conhecido do AGV em: Host Name, onde simultaneamente se define a porta de acesso para este serviço, como a porta 22, que é a porta padrão para este protocolo, embora tal como no VNC exista a possibilidade de se configurar outras portas com a mesma finalidade. Com estas definições é necessário ainda criar um túnel criptográfico no qual o serviço VNC o utilize para estabelecer uma conexão segura através do protocolo SSH. Esta configuração é realizada no diretório: Connection > SSH > Tunnels definindo-se neste diretório a porta TCP que se irá utilizar, que neste caso é a mesma porta que se utiliza para a conexão do protocolo VNC (porta 5900). Estas atribuições devem ser guardadas localmente no dispositivo do Utilizador, para se evitar repetir a inserção de todos os parâmetros cada vez que se pretende ter acesso remoto ao AGV. Portanto, reunidos todos os fatores para efetuar a ligação, basta conectar-se via PuTTY e inserir as credenciais de acesso do AGV como se estivesse localmente neste, o nome do Utilizador e a sua *password*, respetivamente. Se o *login* for bem-sucedido, então o Utilizador criou um túnel criptográfico através do protocolo SSH entre este e o AGV com acesso completo ao terminal de comandos do sistema. Para se ter acesso gráfico ao Raspberry Pi OS e ao seu conteúdo é necessário agora estabelecer uma conexão através do VNC que comunicará com o AGV através do túnel criptográfico formatado no PuTTY. Como hospedeiro do serviço VNC utilizou-se o RealVNC que é um software comercial com opções gratuitas que permitem o acesso remoto sem limitações a este nível. Por isso, ao correr este software basta inserir o endereço IP estático do AGV, segundo o comando: localhost:0, e efetuar novamente a inserção das credenciais de acesso do AGV, com ocorreu no PuTTY. Com o *login* bem-sucedido o Utilizador tem acesso gráfico completo ao AGV via remota.

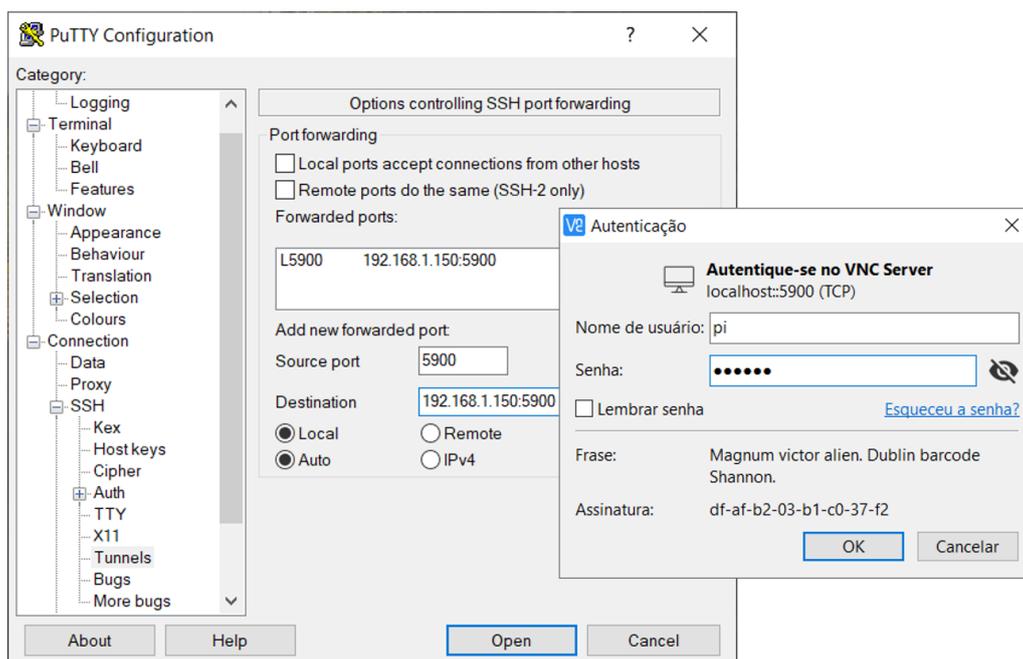


Figura 4.39: Configuração da túnel criptográfico a partir do PuTTY + ligação VNC

4.3.3 Arquitetura do Algoritmo Computacional

O desenvolvimento do algoritmo que corre no AGV é resultado de uma programação de alto nível através da linguagem de programação python, que é uma linguagem orientada a objetos concebida com o intuito de privilegiar a legibilidade do código ao invés da sua expressividade ou velocidade de computação [81]. A escolha desta linguagem deve-se principalmente a dois fatores: à sua total compatibilidade com o Raspberry Pi que utiliza o python como linguagem nativa e também, porque é uma das linguagens de programação mais utilizadas aos dias atuais [82] com um contínuo desenvolvimento comunitário sem fins lucrativos, o que permite manter um suporte atualizado devido aos *updates* regularmente lançados. Este suporte técnico proporciona a possibilidade de se continuar a desenvolver o AGV de forma duradoura, visto a relevância e populismo crescente desta linguagem de programação em aplicações industriais e aplicações que albergam as tecnologias propostas pela Indústria 4.0 [83]. Estas áreas de desenvolvimento são exatamente as áreas onde a inclusão de AGVs é mais promissora devido à diversidade de tarefas que se podem integrar com esta tecnologia em expansão. Fora estes fatores, é uma linguagem de programação também compatível com a utilização do OpenCV com acesso total às suas bibliotecas de visão artificial, embora este seja um módulo nativo em C++. Portanto, a inclusão de todas estas ferramentas tem como objetivo formar um algoritmo capaz de executar as ideias propostas ao longo deste trabalho, nomeadamente a vertente de análise de imagens em tempo real a partir da visão artificial, aliada às tomadas de decisão autónomas programadas em python que influenciam na escolha do trajeto percorrido pelo AGV, através do controlo imposto nos motores elétricos proporcionado uma capacidade de deslocação seletiva ao longo dos vários trajetos que formam o percurso operacional.

Deste modo, o desenvolvimento deste algoritmo foi realizado num editor de código *open-source* que é o Visual Studio Code que é uma plataforma com suporte para ferramentas de resolução e deteção de problemas, realce de sintaxe incluindo palavras reservadas, preenchimento inteligente de comandos inseridos, conectividade com plataformas nuvens, entre outras características que dotam esta interface, como um meio de desenvolvimento indicado para a elaboração deste programa, embora seja possível modificar o algoritmo em outra interface que suporte o python e o OpenCV [84]. O algoritmo desenvolvido subdivide-se em quatro *scripts* que se relacionam entre si de acordo com a figura 4.40. Singularmente não funcionam de forma isolada, visto estarem concebidos para operem em simbiose uns com os outros, de forma a criar-se um código modular e procedimental com o objetivo de se seguir um fluxo lógico de forma organizada que possibilite uma melhor interpretação do algoritmo, uma melhor deteção e resolução de problemas e permitir uma melhor eficiência do algoritmo global através da reutilização de funções modulares em diferentes localizações. A falta destes atributos no desenvolvimento do programa leva à criação de um algoritmo complexo que não segue regras nem estilos de programação, o que torna difícil a sua interpretação e por isso dificulta o seu contínuo desenvolvimento e aceitação por parte da comunidade ou entidades com interesse no seu desenvolvimento.

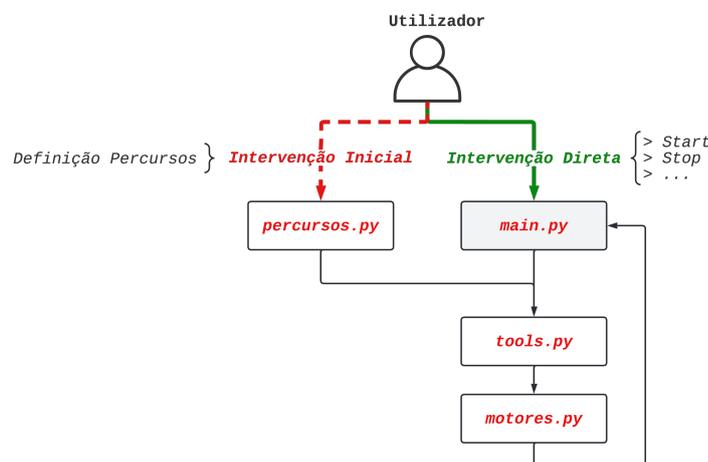


Figura 4.40: Relação entre *scripts*

Salvaguardando as boas práticas para a elaboração de um algoritmo modular e procedimental, o programa desenvolvido possui um ficheiro principal que é o `main.py`. Este é um ficheiro simples onde o Utilizador tem acesso em qualquer instância durante a fase de funcionamento do AGV, a partir da ligação remota existente entre eles, concedendo ao Utilizador acesso direto a informações sobre o percurso operacional e a possibilidade de intervenção direta no funcionamento do AGV, através de ações como a interrupção e o início de um ciclo de funcionamento, através dos botões virtuais de *stop* e *start*, respetivamente. Este é o único ficheiro com esta habilidade de intervenção em tempo real sobre estas condições, embora exista um segundo ficheiro que é `percursos.py` onde o Utilizador tem a obrigatoriedade de inserir os percursos caso não estejam definidos antes do funcionamento do AGV. Este é um ficheiro exclusivo para a definição dos percursos, de forma a tornar intuitivo a definição ou modificação destes, de modo que em regime fabril ou no contexto onde o AGV for aplicado, este ficheiro mais o `main.py` são os *scripts* que se modificam para realizar alterações de alto nível. Já as alterações de baixo nível são conseguidas quando se edita ou modifica as funções mais intrínsecas deste algoritmo, que possuem uma maior complexidade, embora sigam uma lógica modular. Estas funções apenas são acedidas, quando alterações de alto nível são ineficientes e por isso o pretexto para essa intervenção. Deste modo, a este nível existem dois ficheiros: o `tools.py` que contém toda a parte de processamento de imagem através de visão artificial e algoritmos de tomada de decisão, enquanto o ficheiro `motores.py` é responsável por integrar o algoritmo que permite controlar os motores elétricos. A ação conjunta de todos estes ficheiros leva ao funcionamento do AGV, de forma que na figura 4.41 existe um fluxograma que resume de forma lógica o algoritmo implementado que será detalhado nas próximas seções.

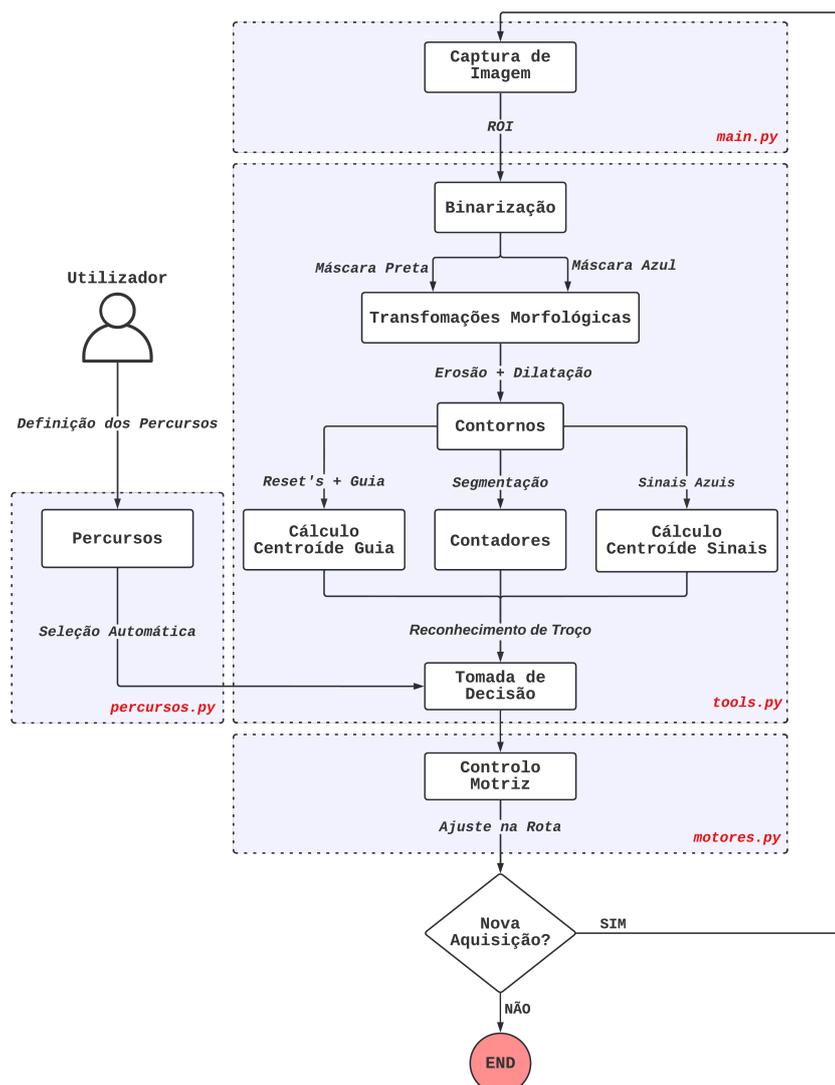


Figura 4.41: Fluxograma do algoritmo do AGV

4.3.4 Captura de Imagem

O algoritmo inicia os seus procedimentos através da aquisição de uma única imagem de forma cíclica através de um ciclo `while` com uma condição de saída controlada pelo Utilizador. A velocidade de processamento de cada imagem depende apenas da eficiência do algoritmo implementado e do conjunto de ações que é requisitado a este, como a visualização do percurso ou de outras informações relevantes ao Utilizador. No final de cada ciclo o algoritmo origina uma ação física que se traduz no movimento motriz do AGV, conforme a localização deste perante o trajeto que é analisado. Quanto mais rápido este processamento computacional for realizado, mais imagens do trajeto são adquiridas permitindo realizar um processamento mais responsivo com uma maior frequência amostral, o que possibilita dotar o AGV de uma maior resposta face ao trajeto em análise. Esta captura é realizada exclusivamente pela câmara digital policromática descrita anteriormente na seção 4.2.5, que se encontra posicionada de forma paralela ao piso fisicamente constrangida por ligações aparafusadas, a fim de diminuir-se oscilações ou perturbações no ato de aquisição que promovam diferentes condições de aquisição. Pelas mesmas razões enumeradas, para permitir uma aquisição uniforme ao longo do funcionamento do AGV, é integrado um sistema de iluminação que permite manter uma luminosidade constante, tornando o AGV independente da luminosidade do meio envolvente, na qual estaria sujeito a diferentes tipos de iluminação com diferentes intensidades e pontos de incidência. Esta volatilidade dependente da iluminação do meio envolvente influencia significativamente as funções de visão artificial utilizadas, o que comprometeria substancialmente todo o processo de processamento de imagem.

Perante estes fatores e tendo em consideração o poder de intervenção do Utilizador a este nível, antes de qualquer processamento de imagem é necessário garantir uma condição básica para desencadear todo o restante algoritmo de processamento, que se baseia na imagem aqui adquirida. Esta condição baseia-se na verificação se a câmara digital se encontra ligada ou com algum bloqueio imposto por software, que mesmo ligada, neste caso ao Raspberry Pi pode estar bloqueada por questões de segurança ligadas à privacidade impedindo a transmissão de imagem. Caso não exista nenhum constrangimento, então o processo continua de forma autónoma com a definição do ROI (*Region of Interest*), que é uma zona específica oriunda da imagem adquirida, limitada a um polígono de dimensões definidas pelo Utilizador com o objetivo de se analisar apenas essa área formada, ao invés de se analisar toda a imagem adquirida [85]. A principal vantagem de se definir um ROI é conseguir-se uma melhor eficiência computacional, porque de forma quantitativa analisa-se um menor número de pixels, o que em termos qualitativos permite eliminar zonas que não revelam informações relevantes ao processo, enquanto as zonas que contêm características importantes para o processo são incluídas nesta nova imagem formada.

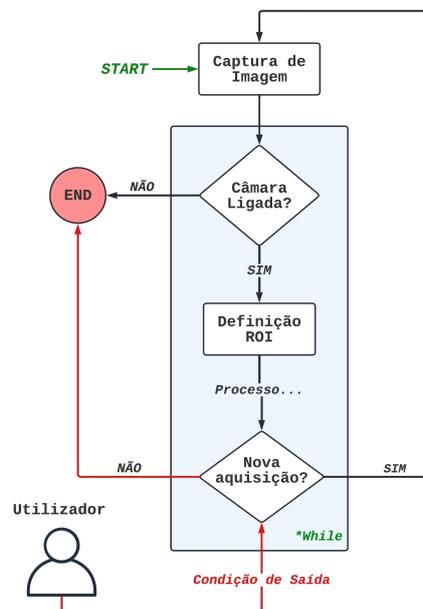


Figura 4.42: Algoritmo da captura de imagem

Porém, na eventualidade de o algoritmo não verificar esta condição, o processo é interrompido sem qualquer intervenção do Utilizador de forma autónoma com a aparição de um erro. Este erro sugere ao Utilizador que verifique qual a causa que leva à falha de verificação da condição anterior. Independentemente da origem desta causa, caso não se resolva esta situação, então o algoritmo não continua, pois não existe forma de adquirir imagens em tempo real, uma vez que a câmara digital não se encontra acessível. Não existem exceções a esta verificação e por isso é obrigatório garantir as condições de aquisição aqui mencionadas. Por isso, caso surja em problema a esta nível, o Utilizador deve resolvê-lo, de modo a garantir o funcionamento do AGV, sendo que as medidas que este tomou para resolver este problema de verificação são avaliadas novamente nessa condição a cada ciclo de funcionamento. Sempre que não se verifique esta condição, independentemente no número de ciclos processados, o algoritmo é interrompido. Na prática, verificar esta condição não é difícil e por isso é uma condição trivial de se verificar, porém torna-se necessário a sua integração a cada iteração, visto a sua importância no processo.

Por cada resultado positivo nesta condição e retornando o raciocínio da página 61, forma-se um novo ROI a cada ciclo de funcionamento. No AGV, este ROI é definido como um polígono de quatro lados, um retângulo de largura igual à imagem original, enquanto a altura deste é bastante reduzida comparativamente com o tamanho original. O motivo destas proporções deve-se essencialmente a dois fatores: à definição da melhor localização para realizar a aquisição de dados, o que mesmo com um sistema que providência condições uniformes de aquisição, existem locais da imagem original onde se destaca melhor as características a captar. O outro fator deve-se à natureza dos objetos que se pretendem adquirir sendo eles a guia e os sinais de reset e de mudança de direção. Ambos os objetos, por motivos distintos, necessitam ser adquiridos com rigor evitando a aquisição falsos positivos. A este nível, considera-se falsos positivos sempre que o algoritmo contabiliza um objeto mais que um determinado número de vezes, ou mais vulgarmente, quando se contabiliza um objeto em locais do trajeto onde este não existe. No caso da guia, é preferível um ROI com uma altura mais reduzida, uma vez que é com base na área do troço captado que se calculará o centroide desta, sendo este o principal mecanismo para o guiamento do AGV, como se verá nas próximas seções. Isto porque, quanto mais pequeno for a altura definida no ROI, o cálculo do centroide é mais rigoroso uma vez depender apenas da forma geométrica do troço em análise. Troços mais comprimidos ou ROI's com alturas superiores à definida, em especial em curvas são onde os erros são mais significativos devido à forma geométrica formada distanciando-se de um retângulo, que é a forma geométrica mais rigorosa para o cálculo do centroide nestas condições. Em relação aos sinais, como estes se encontram ao lado da guia prefere-se um ROI com uma largura mais significativa para os captar completamente, mas com uma altura também reduzida, uma vez serem elementos apenas com 1 cm de altura. Deste modo, com o ROI definido termina-se a aquisição de imagem e prossegue-se com o restante processamento, nomeadamente a definição das máscaras.

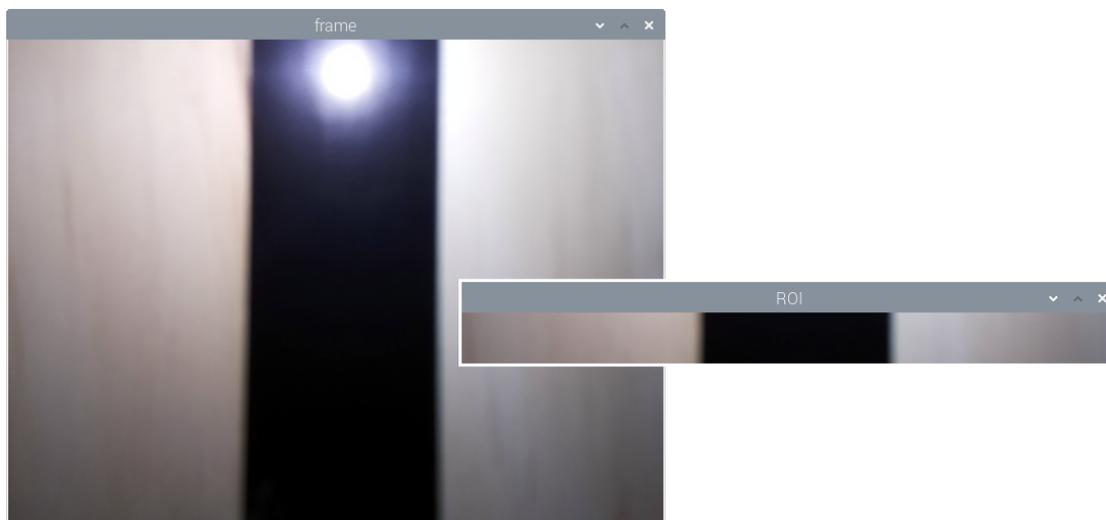


Figura 4.43: Reestruturação da imagem adquirida na formação do ROI

4.3.5 Binarização

A imagem que advém do processo de aquisição (redimensionada pela definição do ROI) é uma imagem policromática segundo um formato RGB. Retornado conceitos introduzidos na seção 3.1.1, o sistema RGB é um modelo de cores aditivo, o qual tem como cores primárias o vermelho (*Red*), o verde (*Green*) e o azul (*Blue*), sendo que outras tonalidades de cores podem ser obtidas a partir da adição destas cores primárias com diferentes intensidades formando uma vasto espectro cromático [86]. É um sistema popular em placas de vídeos de computadores e câmaras digitais, pois é um sistema com uma base teórica sólida que se baseia na detecção e representação de imagens em sistemas eletrônicos [87]. Porém, é um modelo com dois problemas para o processamento de imagem a este nível. O primeiro problema é ser um sistema onde a cor adquirida numa imagem depender do dispositivo que realiza a aquisição, uma vez que qualquer dispositivo deteta ou reproduz um valor RGB de forma diferente de qualquer outro, porque os elementos de cor que constituem o sensor da câmara (transdutores) possuem diferentes sensibilidades aos níveis de vermelho, verde e azul, o que mesmo em dispositivos oriundos do mesmo lote de produção ou do mesmo fabricante leva a diferenças na imagem adquirida. O segundo problema e o principal é a alta sensibilidade deste modelo a variações na iluminação, mesmo com condições de aquisição homogêneas e constantes, garantidas pelo sistema de iluminação implementado e pela fixação do local de aquisição. Mesmo com estes componentes a comportarem-se como uma constante, a iluminação do meio envolvente inevitavelmente influencia na imagem adquirida a cada iteração, influenciando substancialmente no processamento de cada imagem.

Como solução, ao invés de se utilizar diretamente o sistema RGB, converte-se este para um sistema HSV, que é um modelo de cor *Hue-Saturation-Value* desenvolvido pela necessidade de particularizar as propriedades das cores de forma quantitativa através das componentes de coloração, saturação e tom respetivamente. A coloração (H) verifica qual a cor dominante da área em análise abrangendo todas as cores do espectro, desde o vermelho ao violeta, a saturação (S) mede o quão colorido é essa área (intensidade) em relação ao contraste que existe com a sua vizinhança, onde um maior nível de saturação indica uma cor mais viva e o tom ou valor (V) é a componente responsável por quantificar o brilho nessa área ou a iluminância dessa cor, através da quantidade de luz refletida nesse local [88]. Este é um modelo de cor não linear proveniente do sistema RGB, pois não é possível aplicá-lo diretamente na câmara digital utilizada, o que leva a que seja necessário realizar um processo de conversão descrito pelas equações abaixo [89],

$$V = \max(R, G, B). \quad (4.15)$$

$$S = \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{se } V \neq 0. \\ 0, & \text{se } V = 0. \end{cases} \quad (4.16)$$

$$H = \begin{cases} \frac{60(G - B)}{V - \min(R, G, B)} & \text{se } V = R. \\ \frac{120 + 60(B - R)}{V - \min(R, G, B)} & \text{se } V = G. \\ \frac{240 + 60(R - G)}{V - \min(R, G, B)} & \text{se } V = B. \\ 0, & \text{se } R = G = B. \end{cases} \quad (4.17)$$

Que no caso do AGV, para uma imagem de 8 bits o H, S e V variam de 0 a 255 pixels.

Esta conversão justifica-se, porque o modelo RGB é tecnicamente desenvolvido/otimizado para exibir imagens em tela de exibição, enquanto o modelo HSV é contruído com o objetivo de imitar como o ser humano interpreta as cores, aproximando-se com maior rigor a um sistema de cores Munsell [90], que baseia o seu modelo em medições rigorosas a partir das respostas visuais do ser humano ao longo de um espectro cromático. Este sistema, ao contrário do RGB permite separar as cores puras da sua iluminância dando acesso a um conjunto de ações antes não possíveis, para se operar ao nível da coloração e da intensidade de cada cor [91]. Neste caso, no campo do processamento de imagem pretende-se que o AGV detenha uma inteligência visual semelhante à do ser humano, em relação à distinção de cores, e por isso, este está mais interessado em interpretar apenas a cor da imagem, do que interpretar qual o nível da sua iluminância, o que de facto, na maioria das aplicações incluído neste sistema, a luminosidade do meio envolvente é considerada como um ruído, pois afeta como o sistema interpreta uma cor, influenciando na aquisição de cada imagem conforme a sua localização física sem existir uma mudança real nas cores analisadas. Por isso, a implementação de um modelo HSV torna o algoritmo mais robusto e menos sensível à iluminância.

Este formato mune a imagem de uma melhor qualidade em termos de identificação de traços e características com um maior contraste visual, porém, ainda com estas condições, é uma imagem com vários elementos distintos e de diferentes tonalidades, que no conjunto global não facilitam na deteção ou reconhecimento de padrões a partir do processamento de imagem. Outro fator que dificulta o reconhecimento de formas ou padrões a este nível é a quantidade de ruído ainda presente, como por exemplo a cor do piso que para este sistema não revela diretamente nenhum interesse em ser computacionalmente processado. Com isto, para facilitar o processamento de imagem e evitar a extração de características com um baixo grau de confiança, é necessário não só diminuir a quantidade de ruído presente, como integrar um método de segmentação que permita refinar a imagem e dividi-la em conjuntos formados por grupos de pixels que compartilham as mesmas características. A binarização monocromática é um método de segmentação que responde a esta necessidade com o objetivo de facilitar a deteção de formas e a localização de contornos a partir da criação de imagens binárias [92]. De modo trivial, este método fundamenta-se na definição de um valor de referência (*threshold*), onde todos os pixels abaixo deste valor assumem o valor binário 0, que para uma imagem de 8 bits transformam-se em 0 pixels (preto), já os restantes pixels que possuem um valor acima ou igual a este valor de referência assumem o valor de binário 1, que para uma imagem com o mesmo formato corresponde a 255 pixels (branco). Este método é de fácil implementação, mas apenas consegue bons resultados globais, caso exista um baixo nível de ruído nas imagens e estas possuam grupos de pixels com fronteiras bem definidas. Na ausência destes atributos as imagens resultantes ou possuem elementos em defeito, ou elementos em excesso com formas anormais para o que se espera reconhecer via visão artificial. Por isso, face às vulnerabilidades deste método, no AGV, ao invés de definir-se um único valor de *threshold*, define-se antes dois valores designados por limite inferior e limite superior, como forma de criar-se uma gama de aquisição, onde tudo o que se encontra fora desta é eliminado destacando-se apenas a informação contida neste intervalo.

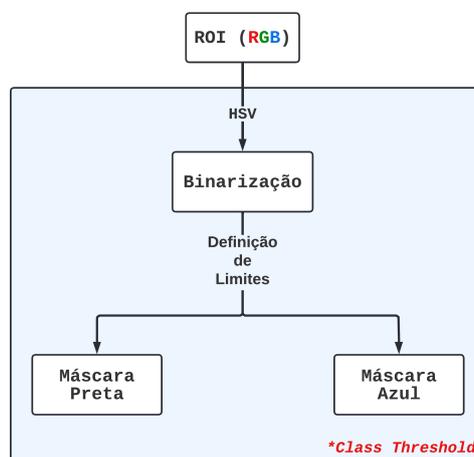


Figura 4.44: Algoritmo de binarização

É de notar, que não se deve aplicar apenas uma única binarização à imagem HSV que chega a esta função, porque no caso do AGV existem duas cores de interesse que se pretende identificar: o azul, que é exclusivamente oriundo dos sinais de mudança de direção, enquanto o preto advém da guia e dos sinais de reset. Portanto, o porquê de ser irrealizável a aplicação deste conceito, justifica-se pelo facto de ser impraticável definir um intervalo comum, que inclua ambas as tonalidades sem restringir significativamente o espectro cromático de cada cor. Esta ideia torna-se ainda mais frágil, porque ao definir-se um único intervalo comum, dá-se espaço a incluir outras cores entre o azul e o preto, que neste caso não revelam interesse em ser adquiridas. Então, ao implementar-se um mecanismo que capta estas cores de forma indireta, favorece-se a contaminação das imagens com mais ruído, levado a um efeito contrário do que este método pretende realizar. Para além disso, caso se pretenda adquirir mais alguma cor de forma consentida e não por coincidência no intervalo estipulado, então todo este conceito se torna mais impraticável, agravado negativamente todas estas defetividades apresentadas. Por isso, independentemente do número de cores a adquirir, introduz-se o conceito de máscara.

Em visão artificial, uma máscara é uma imagem binária resultante do processo de binarização, na qual se define apenas uma cor para aquisição por cada máscara formada, através da formação de um intervalo balizado por dois limites, levando à formação de um domínio fechado, que contém com eficiência uma grande parte do espectro cromático de cada cor, definida manualmente pelo Utilizador [23]. A vantagem desta metodologia é a formação de intervalos dedicados e não compartilhados com outras cores, o que reduz de imediato o risco de captação de cores indesejadas, que levariam à introdução de ruído nesta imagem. Fora isso, este é um conceito que permite ser implementado de forma modular, não possuindo limites para o número de máscaras que se pode implementar. Esta característica, caso implementada a várias dezenas de cores pode começar-se a notar em termos de performance computacional, visto o tempo dedicado a este processo. Evita-se esta “estagnação” através de boas práticas associada às cores que realmente são necessárias adquirir para o processo, que no caso do AGV, como se interessa apenas captar duas cores, então este esforço computacional extra é mínimo não sendo perceptível no processo global. Por isso, face a estas duas cores, existem duas máscaras que se formam: a máscara azul e a máscara preta. Independentemente da máscara a definir, é importante formar intervalos restritivos, mas flexíveis ao ponto de incluir variações da mesma cor, para se conseguir melhor resultados e uma melhor interpretação através de visão artificial, evitando situações onde com mínimas diferenças na iluminação a interpretação da cor por parte do algoritmo já falha. No AGV, como resultado de todo este algoritmo esquematizado na figura 4.44, nas imagens abaixo encontram-se os resultados práticos destes conceitos.

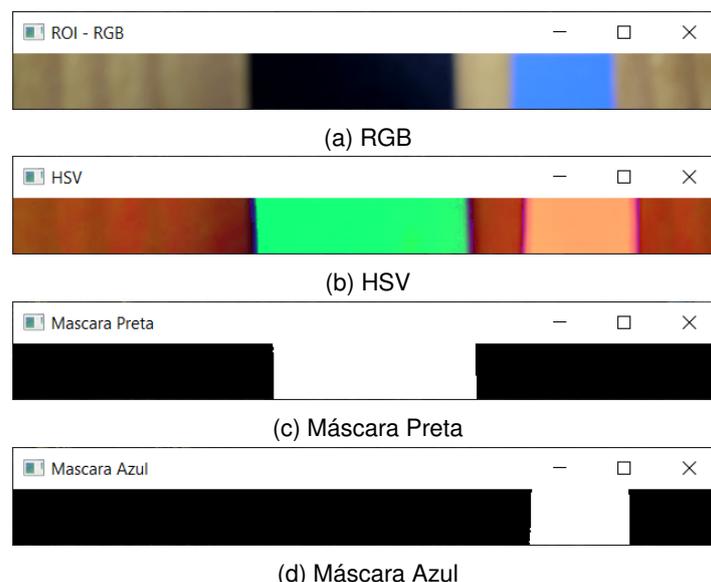


Figura 4.45: Segmentação através da binarização monocromática

4.3.6 Transformações Morfológicas

As máscaras obtidas na função anterior não são perfeitas, pois estas ainda possuem pequenos vestígios de ruído, especialmente ao longo das suas fronteiras, porque grande parte deste ruído é provocado por irregularidades no piso e devido à baixa nitidez sentida nestas zonas de fronteira, agravadas por diferentes iluminâncias provocadas pelo meio envolvente acabando por promover uma aquisição alterada nestas áreas. O resultado deste conjunto de ações leva ao aparecimento pontual de pequenos grupos de pixels ruidosos, que em grande parte, como foi mencionado, se encontram nas zonas fronteiriças entre os vários elementos. Já o ruído que surge no interior dos corpos, é gerado maioritariamente devido a defeitos físicos ou a sujidades presentes nos elementos integrados no piso, que com o natural uso dos vários troços, existe uma tendência para a aparição de anomalias e a aglomeração de sujidade ao longo dos trajetos. Ressalva-se que a sujidade, também é um fator comum na alteração física das dimensões das fronteiras, quando não existem quaisquer cuidados de limpeza, podendo-se gerar situações onde se formam pontes de sujidade que unem elementos e causam falhas ao nível do reconhecimento devido à forma anormal formada. Sem exceções, todos os fatores aqui abordados podem ser substancialmente amenizados ou quase resolvidos na sua totalidade, através da utilização de transformações morfológicas a partir do algoritmo descrito na figura 4.46.

Antes de se iniciar estas transformações, o ruído presente em cada máscara a este nível não é significativo nem em grande quantidade, mas ainda assim, as imagens formadas carecem de um toque de refinamento mais preciso face aos processos anteriores, o que na ausência destas técnicas de refinamento leva a aparição de áreas de maior vulnerabilidade que afetam como o algoritmo interpreta uma forma, ou como deteta um elemento. Por isso, justifica-se a aplicação destas transformações, como último processo de filtragem, uma vez que nos processos de maior complexidade, já não existe um trabalho ao nível da imagem reformulada, mas sim em relação às características que foram extraídas desta e por isso a necessidade de garantir-se a entrega de uma imagem de boa qualidade. Logo, ao atuar-se dentro do admissível para se reduzir a quantidade de ruído nas imagens adquiridas sem comprometer a performance computacional global, torna-se numa tarefa fulcral, uma vez que nestas etapas iniciais ao ter-se estes cuidados que diferenciam as imagens a este nível, promove-se diretamente um melhor processamento por parte das funções de mais alto nível. Para além disso, à semelhança do que acontece com a conversão da imagem RGB para o formato HSV, imagens a este nível bem transformadas e refinadas tornam o algoritmo mais robusto e menos suscetível a falhas, reduzindo o número de falsos positivos.

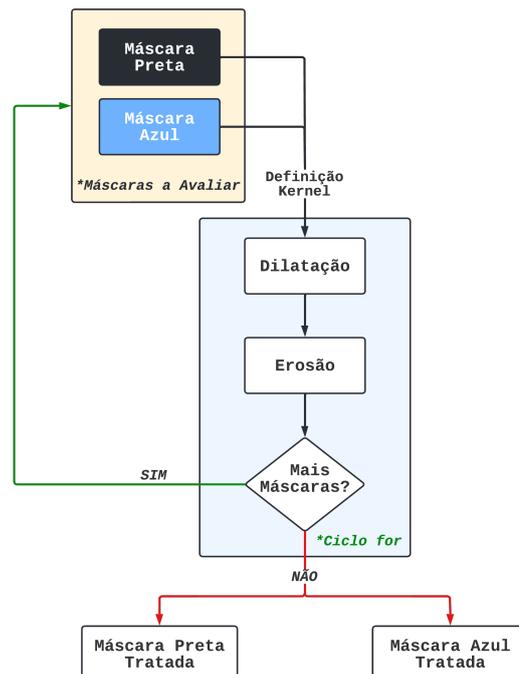


Figura 4.46: Algoritmo basilar das transformações morfológicas

Deste modo, as transformações morfológicas na área do processamento de imagem são técnicas que se aplicam na análise de imagens binárias baseadas no processamento de estruturas geométricas suportadas pela teoria de conjuntos através da utilização de conceitos topológicos como: tamanho, forma, distância, convexidade, etc [93]. Destes conceitos resulta uma operação morfológica binária limitada por um conjunto de pixels, que definem uma vizinhança de valor binário com um ponto central (x, y) , formando uma matriz de dimensões e forma arbitrária. Esta matriz é designada por elemento estruturante (*kernel*) e atua pixel a pixel funcionando como uma ferramenta de comparação com a vizinhança externa, fixando-se num determinado pixel da imagem (em cada atuação) e com isso aplica o processo morfológico em vigor para os pixels que formam esta matriz, dando origem no final do processo a uma imagem com a reestruturação do valor binário dos pixels que justificaram esta troca para o seu valor complementar [94]. A regra que dita esta troca depende do processo morfológico em vigor, que num nível básico de operações morfológicas dividem-se em dois operadores elementares: Erosão e Dilatação.

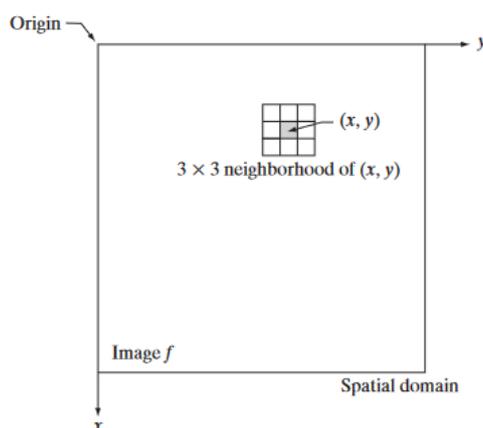


Figura 4.47: Atuação do elemento estruturante ao longo de uma imagem adaptado de [19]

A erosão é um operador elementar responsável por combinar dois conjuntos, através de métodos de subtração vetorial que provocam a remoção de corpos ruidosos na imagem. Quando um pixel integrante da vizinhança do elemento estruturante tiver uma intensidade binária 0, ou seja, for de cor preta, então caso existam pixels integrantes da vizinhança do elemento estruturante de cor branca, então estes são obrigatoriamente convertidos em pixels com uma tonalidade preta, existindo assim uma redução da área útil que se analisa posteriormente (área de pixels brancos). As vantagens desta conversão acabam por beneficiar a separação entre componentes conectados por ruído, favorecem a eliminação de componentes ruidosos no exterior dos corpos e promove-se uma redução do ruído ao longo das fronteiras dos elementos, que é o local onde tipicamente existem em maior quantidade. A dilatação é também um operador elementar que atua de forma oposta à erosão, uma vez que esta é responsável por combinar dois conjuntos a partir de métodos de adição vetorial levando ao preenchimento de espaços (*gap filling*) [95]. Quando um pixel integrante da vizinhança do elemento estruturante tiver uma intensidade binária 1, ou seja, for de cor branca, então caso existam pixels integrantes da vizinhança do elemento estruturante de cor preta, então estes são obrigatoriamente convertidos em pixels com uma tonalidade branca, existindo assim um aumento da área útil que se analisa posteriormente. As vantagens desta conversão acabam por facilitar na eliminação de componentes ruidosos no interior de corpos, auxiliam na conexão entre elementos separados por ruído e promovem o preenchimento de lacunas em objetos, como buracos causados por fronteiras pouco nítidas.

Não existe uma ordem específica em termos de aplicação das operações morfológicas, pois estas dependem de cada situação. Porém, a aplicação singular de apenas um dos operadores elementares leva à alteração da área dos vários objetos podendo trazer implicações que influenciam no processamento seguinte. Como solução, estes operadores permitem uma interatividade entre eles (permutação) sendo possível aplicar estas alterações nas imagens sem causar alterações expressivas na forma e tamanho dos elementos, o que significa que para se conseguir estes resultados é necessário aplicar ambos os operadores. Nestas situações, forma-se novas operações morfológicas designadas por Abertura e Fechamento.

Tanto a abertura, como o fechamento são operações morfológicas formadas pela dilatação e erosão diferenciando-se uma da outra a partir da ordem de aplicação. Portanto, a abertura é operação que se inicia com a erosão, imediatamente seguida por uma dilatação. Esta sequência morfológica, além de manter as dimensões dos elementos que constituem a imagem original com certo grau de semelhança, promove a suavização de contornos, permite a quebra de istmos e a eliminação de saliências finas. Contudo, como efeito deste processo morfológico obtém-se uma imagem à saída menos rica em detalhes e com os elementos mais isolado entre si. O fechamento é uma operação inversa à abertura, pois inicia-se em primeiro lugar com uma dilatação, imediatamente seguida por uma erosão. Esta sequência morfológica, além de manter as dimensões da imagem inicial também com um certo grau de semelhança, promove também a suavização de contornos, mas, ao contrário da abertura, geralmente permite quebras estreitas entre troços, elimina pequenos orifícios e preenche lacunas ao longo do contorno da imagem. No entanto, como efeito adverso deste processo morfológico obtém-se uma imagem à saída menos rica em detalhes e com os elementos mais juntos dando espaço a possíveis problemas por causa de conflitos de fronteiras [95].

No AGV, na maioria das situações como já foi mencionado, o ruído presente a este nível é pouco expressivo e por isso a sua baixa influência no processo. Porém, existem condições onde a luminosidade do meio envolvente é mais impactante no processamento de imagem e em especial, quando esta iluminação é causada por focos de luz direcionados, provocando a formação de ruídos essencialmente no interior dos corpos, ao invés nas suas fronteiras como acontece em condições normais. Este ruído advindo destas condições tem maior influência no interior da guia por efeito da reflexão dessa iluminação. É uma situação invulgar mesmo nestas condições, mas plausível de ocorrer dentro da gama de acontecimentos a que o AGV está sujeito. Por isso, mesmo nestas condições excepcionais, o AGV deve comportar-se de igual modo perante a quantidade substancial de ruído formado como se verifica num exemplo na imagem 4.48 (a). Como em grande parte se trata de ruído no interior dos corpos, e como é de interesse para o processo manter as dimensões dos vários elementos adquiridos, então aplica-se por defeito, em qualquer situação do trajeto independentemente da quantidade de ruído existente um fechamento, de modo a se preencher em primeiro lugar as lacunas no interior dos corpos e em seguida aplica-se uma erosão, para erodir essencialmente as fronteiras da guia, com o objetivo de melhorar os seus limites e restaurar as dimensões iniciais da guia.

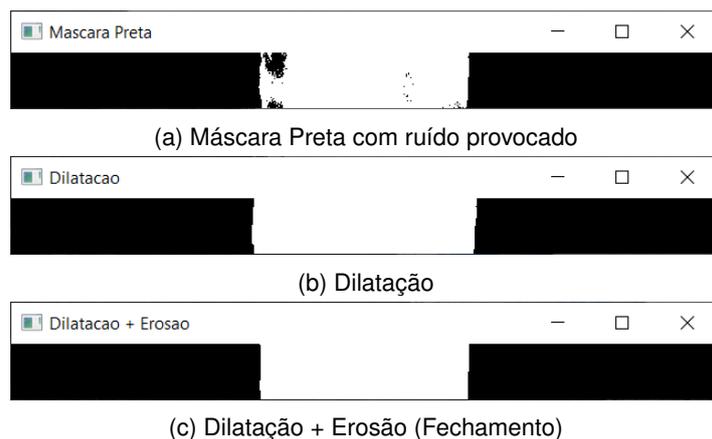


Figura 4.48: Atuação do fechamento na amenização do ruído

O resultado deste processo morfológico é a redução substancial deste ruído, que em condições normais não tem esta influência na imagem adquirida. É um algoritmo que consegue um melhor desempenho em situações extraordinárias, visto em situações normais causar uma menor contribuição para o processamento de imagem, porque nestas condições a grande parte do ruído é logo filtrado pelas máscaras binárias. Ainda assim, este mecanismo implementado melhora a força e inteligência deste algoritmo na segmentação das imagens e repartição dos elementos para a extração futuras das características de interesse.

4.3.7 Contornos

Os processos aplicados ao longo das seções anteriores tiveram como objetivo melhorar a qualidade da imagem captada a cada ciclo de operação. Desde a formação de uma região de interesse como forma de criar um ambiente de aquisição restrito, à aplicação de máscaras binárias, como forma de repartir a imagem original em grupos separados pela cor que se pretendeu isolar, aos processos morfológicos utilizados com o objetivo de reduzir o ruído em situações anormais, provocados essencialmente por sistemas de iluminação direcionados. Sempre se operou ao nível da imagem original com a intenção de melhorar a qualidade global dos vários elementos que a integram. Por isso, a este ponto consegue-se ter uma imagem segmentada quase “blindada” devido à imunidade que tem face a ruídos externos e outras perturbações, o que dispensa a aplicação de novos métodos que zelam pelos mesmos objetivos dos processos anteriores. Não se justifica o custo em termos de tempo e esforço computacional disposto, para se conseguir mínimas melhorias na imagem final. Portanto, como início de uma nova etapa neste algoritmo, inicia-se agora todos os processos de nível intermédio implementados, relacionados com a extração de características da imagem anteriormente tratada. Passa-se agora a se a trabalhar com estas características detetadas, ao invés da própria imagem, o que confere uma maior rapidez em termos de aplicação de métodos, porém, os algoritmos desta natureza tornam-se cada vez mais complexos de implementar. Face a estas condições, inicia-se esta nova etapa com a determinação dos contornos a partir do algoritmo presente na figura 4.49.

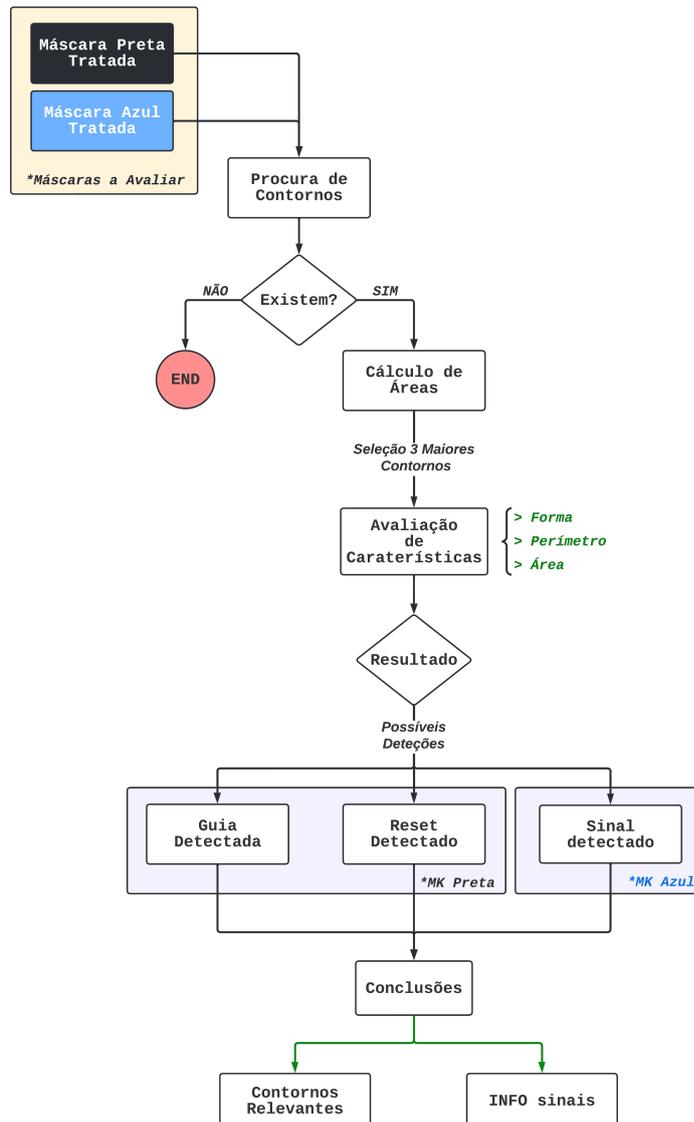


Figura 4.49: Algoritmo basilar para a extração de características

Os contornos são definidos como uma linha que liga todos os pontos contínuos com a mesma intensidade ou cor ao longo do limite de uma imagem [96]. A sua captação é de extrema utilidade para o processamento de imagem, pois é uma ferramenta que permite a análise de formas e a determinação indireta de propriedades aritméticas ligadas a agregados de pixels na imagem, como o cálculo de áreas, perímetros, arestas, etc. A determinação de um contorno em uma imagem digital em um plano bidimensional, nomeadamente numa máscara binária, acontece por uma mudança abrupta ao nível das intensidades dos pixels, ou seja, quando se passa de uma vizinhança com uma intensidade preta, para uma vizinhança de intensidade branca ou vive-versa, então os pixels que limitam ambos os agregados binários formam uma fronteira, sendo este o local ideal para a definição de um contorno [97]. Contudo, a definição de contornos é difícil, pois envolve na prática a presença de imperfeições na imagem em análise causada principalmente por ruído ao longo das fronteiras, devido à baixa nitidez nestas zonas. A atuação deste ruído leva à suavização dos pixels que se encontram nos limites das fronteiras entre os dois grupos, o que torna o processo mais complexo, visto não se conservar as dimensões originais dos elementos.

No AGV, a maior parte do ruído já foi eliminado pelos processos anteriores e portanto, a busca de contornos é baseada no algoritmo de Satoshi Suzuki e Keiichi Abe [98] integrado em módulos do OpenCV. A maior parte das técnicas utilizadas para a deteção de contornos tentam resolver este problema através da definição de um operador derivativo de primeira ou segunda ordem, associado a técnicas de suavização para reduzir o ruído presente ao longo das fronteiras. Neste caso, como a presença de ruído a este nível é insignificante e se procura um método computacionalmente mais eficaz e com uma melhor performance, então justifica-se a integração do método proposto. Este método fundamenta-se na integração de dois algoritmos similares que zelam pela busca e classificação de contornos, mas diferenciam-se um do outro, uma vez que o primeiro algoritmo distingue contornos de formas ¹ e lacunas ², enquanto o segundo algoritmo, apenas realiza a deteção das formas mais exteriores, ou seja, apenas realiza a identificação dos contornos das formas.

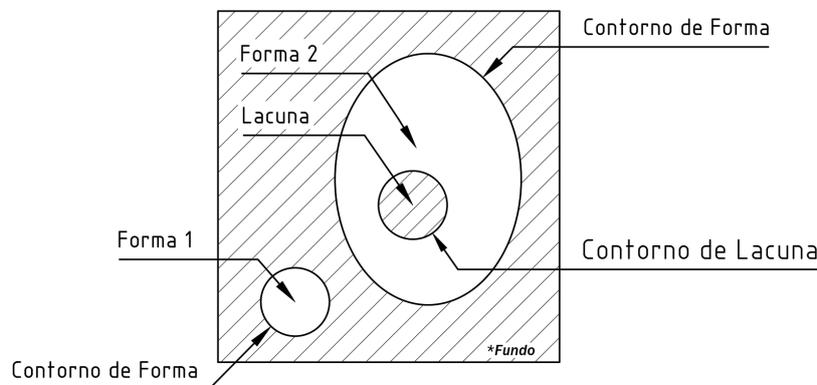


Figura 4.50: Definição de formas e lacunas segundo [98]

A determinação dos contornos à luz do primeiro algoritmo inicia-se pela procura de formas e lacunas nas máscaras binárias advindas dos processos anteriores. É analisado uma máscara de cada vez, através de uma análise que procura um pixel inicial que verifique se existem condições para se formar o início de um contorno e conseqüentemente uma forma. Este pixel inicial pode ser qualquer pixel pertencente ao contorno. Por isso, ao considerar-se um dado pixel como o ponto inicial do contorno, então a partir dele é iniciado um processo de seguimento e classificação, através da procura de todos os outros pixels que compartilham condições semelhantes a este ponto inicial. Este processo repete-se, sempre que se verifique a presença de outros contornos ainda não identificados e, portanto, ao considerar-se o limite exterior de uma imagem definida por $F = f_{xy}$, onde f_{xy} correspondem à intensidade de cada pixel de coordenadas (x, y) , então o processo de deteção de contornos aplica-se a qualquer pixel que verifique a seguinte condição: $f_{xy} \neq 0$.

¹ Grupos de pixels de intensidade binária 1.

² Grupos de pixels de intensidade binária 0 localizados no interior de formas.

1. Deste modo, inicia-se o processo de análise pela varredura da máscara binária em análise iniciando-se este processo deste o canto superior esquerdo até ao canto inferior direito. Repare-se que o algoritmo de forma recursiva irá analisar todas as linhas, percorrendo cada coluna a cada iteração, de forma a procurar pontos de interesse que integrem o(s) contorno(s) a identificar. Com base nesta metodologia, o algoritmo testa a cada iteração as seguintes condições:

- (a) Se $f_{xy} = 1$ e $f_{x,y-1} = 0$, então assume-se que o pixel de coordenada (x, y) é o ponto inicial e detém as condições necessárias para o seguimento de um contorno externo. Deste modo, o NCA (Número sequencial do Contorno Atual) é incrementado e a coordenada deste é modificada de modo que $(x, y - 1) \rightarrow (x_2, y_2)$.

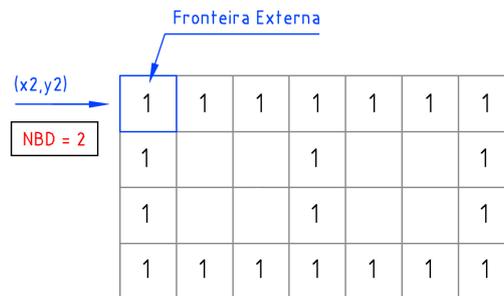


Figura 4.51: Etapa 1.(a)

- (b) Se $f_{xy} = 1$ e $f_{x,y+1} = 0$, então assume-se que o pixel de coordenada (x, y) é o ponto inicial e detém as condições necessárias para o seguimento de uma lacuna. Deste modo, o NCA é incrementado e a coordenada deste é modificada de modo que $(x, y + 1) \rightarrow (x_2, y_2)$, enquanto o NUCD (Número sequencial do último Contorno Detectado) corresponde à intensidade deste pixel.

- (c) Caso não se verifique nenhuma destas opções, então segue-se diretamente para o ponto 3.

2. Com a definição do ponto inicial de coordenada (x, y) , segue-se a deteção do contorno de modo que:

- (a) Ao iniciar-se no pixel de coordenadas (x_2, y_2) , começa-se por se analisar no sentido dos ponteiros do relógio os pixels que constituem a vizinhança de (x, y) com o objetivo de se encontrar um pixel diferente de zero. Quando se encontra um pixel nestas condições, este pixel passa agora a ter as coordenadas (x_1, y_1) , de forma que se não for encontrado nenhum pixel com este atributo, então torna-se negativo o valor do NCA e atribui-se este valor à sua intensidade (f_{xy}), passando diretamente para o ponto 4 neste caso espacial.

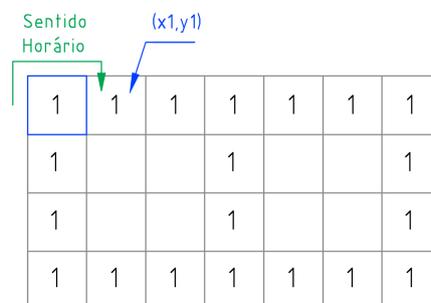


Figura 4.52: Etapa 2.(a)

- (b) Encontrado um pixel diferente de 0, converte-se as coordenadas deste e da vizinhança formada, de modo que: $(x_1, y_1) \rightarrow (x_2, y_2)$ e $(x, y) \rightarrow (x_3, y_3)$, respetivamente.

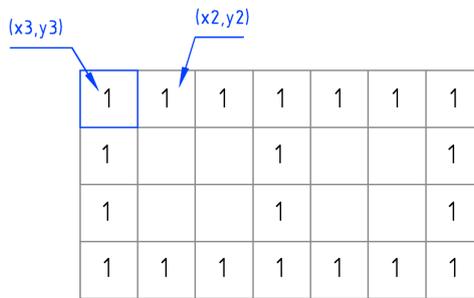


Figura 4.53: Etapa 2.(b)

(c) Ao chegar-se a este ponto, aplica-se o mesmo raciocínio da alínea a), só que agora analisa-se a vizinhança do pixel atual de coordenadas (x_3, y_3) , no sentido anti-horário dos ponteiros do relógio, iniciando-se este processo no pixel de coordenadas (x_2, y_2) na procura de um pixel diferente de zero, atribuindo-se a este pixel as coordenadas (x_4, y_4) quando descoberto. No fim, altera-se a intensidade da vizinhança (f_{x_3, y_3}) segundo uma das seguintes opções:



Figura 4.54: Etapa 2.(c)

- i. Se o pixel de coordenadas $(x_3, y_3 + 1)$ é um dos pixels iguais a zero já analisado na alínea c), então o valor negativo do Número de Contornos Atuais (- NCA) é atribuído a este pixel.
 - ii. Se o pixel de coordenadas $(x_3, y_3 + 1)$ não é um dos pixels iguais a zero mas, já analisado na alínea c), então o valor do Número de Contornos Atuais (NCA) é atribuído a este pixel.
 - iii. Se não se verificar nenhuma das opções anteriores, então o pixel de intensidade (f_{x_3, y_3}) não sofre alterações.
- (d) Se as coordenadas do pixel (x_4, y_4) coincidir com (x, y) e as coordenadas da vizinhança (x_3, y_3) coincidir com (x_1, y_1) , então regressa-se ao ponto inicial e depois segue-se para o ponto 4. Caso não seja verdade este conjunto condições, então modifica-se o valor do pixel atual (x_4, y_4) para (x_3, y_3) e o valor da vizinhança (x_3, y_3) para (x_2, y_2) e regressa-se à alínea c) desta indentação.



Figura 4.55: Etapa 2.(d)

3. Por último, se a intensidade do pixel atual ($f_{xy} = 0$), então o valor do NUCD (Número sequencial do último Contorno Detectado) recebe esta intensidade e a análise pixel a pixel é retomada ao próximo ponto de coordenadas $(x, y + 1)$.

Após a repetição destas etapas, obtém-se os respetivos contornos, de modo que o valor do NCA varia em função da natureza de cada entidade identificada, ou seja, de acordo com o fluxograma da figura 4.56, todas as entidades previamente definidas (formas, contornos de formas, lacunas e contornos de lacunas) encontram-se hierarquicamente representados.

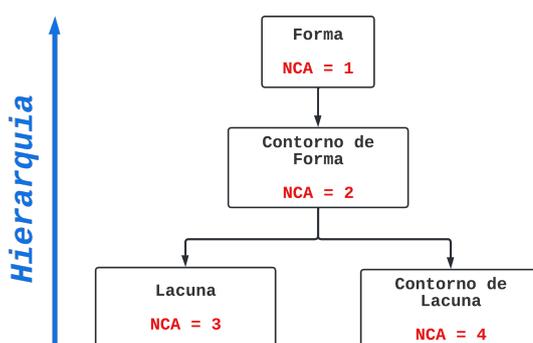


Figura 4.56: Relação hierárquica entre entidades relacionadas à formação dos contornos

Já o segundo algoritmo é referente à deteção exclusiva de contornos externos. Surge como uma modificação do primeiro algoritmo, uma vez que este restringe o processo de deteção a apenas elementos no exterior, ou seja, deteta exclusivamente formas desconsiderando as lacunas que existem nas imagens. As diferenças face ao algoritmo anterior são reduzidas, diferenciando-se principalmente nas condições de início, para o seguimento de contornos e na forma como classifica os mesmos [99]. Estas mínimas diferenças incorporadas neste algoritmo, permitem-no tornar computacionalmente mais eficiente, mas menos completo, o que neste trabalho é visto como uma vantagem, porque não existe nenhuma lacuna nos sinais de que se pretende adquirir, e por isso é o algoritmo mais vantajoso para se implementar na deteção dos contornos. O resultado da integração deste método é visível na figura 4.57 com a deteção de alguns dos elementos que se pretende adquirir.



Figura 4.57: Deteção de contornos

Com a deteção de contornos a este nível, qualquer contorno com diferentes atributos físicos como: forma, tamanho e cor é detetado. É necessário aplicar um conjunto de condições, de modo a se identificar a identidade de cada contorno, na medida que, os contornos captados que não possuem relevância para o processo são eliminados, enquanto os contornos de interesse devem ser identificados através da aplicação de um conjunto de condições que permitem a sua separação e correta identificação, deixando de se operar com segmentos de imagens, para se passar a utilizar apenas valores e características aritméticas destas. Uma destas características físicas é a tonalidade dos vários objetos, que já foram identificadas pela formação das máscaras binárias, contudo, destacam-se duas características ainda a processar: área e forma geométrica de cada contorno, como elementos decisivos para o reconhecimento de elementos. A área de um objeto numa imagem digital é definida como a quantidade bidimensional de pixels de intensidade diferente de 0, ou seja, de pixels brancos que se encontram no interior do objeto capturado limitado pelo seu contorno. O cálculo da área de um objeto depende apenas da sua forma geométrica e por isso face a esta dependência é necessário aplicar o Teorema de Green [100], visto o formato geométrico de cada forma ser irregular e não previsível, não sendo possível aplicar uma fórmula padrão que seria computacionalmente mais eficiente.

Deste modo, no cálculo vetorial, sejam G e F duas funções escalares, isto é, $G, F : \mathbb{R}^n \rightarrow \mathbb{R}$ continuamente diferenciáveis ao longo de um conjunto aberto no plano XoY , e ao considerar-se como C uma curva de Jordan (curva fechada simples) seccionalmente regular [101], onde a reunião da região plana D limitada por C com o seu interior ($D \subset C$), resulta na igualdade descrita pelo Teorema de Green de modo que

$$\iint_D \left(\frac{\partial G}{\partial x} - \frac{\partial F}{\partial y} \right) dx dy = \oint_C (f dx + g dy), \quad (4.18)$$

onde o caminho de integração ao longo de C acontece no sentido direto. Este teorema pode ser utilizado para calcular a área de qualquer polígono ou forma detetada através de um integral de linha, porém, segundo apenas as condições aqui descritas, o que leva a esta região planar D a respeitar obrigatoriamente a seguinte igualdade

$$\frac{\partial G}{\partial x} - \frac{\partial F}{\partial y} = 1, \quad (4.19)$$

uma vez apenas ser válida nestas condições, porque é uma exceção ao caso global, que resulta de uma particularização do Teorema de Stokes [102]. Portanto, como caso particular deste teorema, a área da região planar D limitada por C perante esta condição é dado por

$$A = \iint_D dx dy = \iint_D 1 dA. \quad (4.20)$$

É de notar, que existem várias funções escalares G e F que satisfazem as condições deste teorema como por exemplo $G = 0$ e $F = 0$, porém, ainda segundo [102] ao escolher-se $G(x, y) = x/2$ e $F(x, y) = -y/2$ é possível verificar a equação 4.19 não a transgredindo e com isso, acreditar que qualquer curva de Jordan C limitada por uma região D neste domínio, pode ser exposta ao Teorema de Green com uma solução real expressa do seguinte modo

$$A = \frac{1}{2} \oint_C (x dy - y dx). \quad (4.21)$$

Todavia, esta é uma expressão puramente matemática para o cálculo da área de um corpo sobre estas condições. No caso prático, é impossível aplicar uma expressão desta natureza com um domínio infinito ao longo de \mathbb{R} em termos computacionais, dado os recursos finitos que constituem este sistema. Face a esta limitação, é necessário realizar uma conversão, de modo a tornar-se prático o cálculo das áreas dos objetos em imagens digitais com domínios fechados e limitados no domínio dos números naturais (\mathbb{N}), de forma a não comprometer o esforço computacional no cálculo desta característica. Para se conseguir tal conversão, apenas é necessário que o algoritmo integrado no sistema se adapte ao conjunto que constitui a amostra em análise, que neste caso é uma amostra finita discreta igualmente espaçada formada por pixels [103]. Deste modo, a região planar bidimensional agora em análise é transformada numa série de pixels codificados numa sucessão de pontos de coordenadas $(x[i], y[i])$ limitados por um contorno externo. O resultado desta transformação é a troca do integral de linha por um somatório que contabiliza o número de pixels no contorno formado

$$A = \frac{1}{2} \sum_i x_{i-1} y_i - x_i y_{i-1}. \quad (4.22)$$

Com a exposição do método que permite o cálculo de áreas a partir de imagens digitais limitadas pelos seus contornos, falta demonstrar como se realiza a identificação de formas geométricas com base nestes mesmos contornos, visto ser uma característica vantajosa na identificação destes elementos. Deste modo, ao retroceder-se à fase inicial deste algoritmo, este caracteriza-se pela formação de um ROI com um formato retangular, onde a largura é muito menor que o comprimento. Estas proporções dimensionais promovem uma zona de aquisição, onde os elementos captados são identificados como polígonos de 4 arestas, mais em concreto como retângulos, mesmo que por exemplo os sinais de mudança de direção sejam quadrados. Ora, se assim se verifica esta particularidade, então a identificação da forma geométrica não ajuda numa primeira análise, visto todos os elementos de interesse terem uma forma geométrica semelhante, não permitindo ser uma condição de diferenciação entre estes. Contudo, a forma geométrica de cada elemento não é utilizada como uma característica de identificação principal, mas como uma condição auxiliar na verificação da identidade de cada elemento, na medida que permite diferenciar elementos de interesse de elementos externos irrelevantes ao processo, que por outras palavras são ruído de maiores dimensões, causados essencialmente por elementos externos que possuem uma tonalidade igual a uma das máscaras binárias, e por causa dessa característica aliada a uma área significativa não são eliminados nos processos de filtragem anteriores. Por isso, para evitar irregularidades desta natureza, é necessário verificar a forma de cada elemento, como um fator de autenticação.

A identificação de qualquer elemento neste algoritmo é fundamentada no método de Douglas – Peucker, sendo este um método iterativo que opera com ajustes realizados por pontos intermédios, que a partir de uma forma original forma uma forma aproximada (*polyline*) constituída por segmentos de reta com uma certa tolerância (ϵ) definida. A ideia principal deste método é substituir a curva original captada por uma curva simplificada modelada em termos computacionais, que se comporte aproximadamente como a forma original [104]. As vantagens desta computação é conseguir-se uma curva parametrizada com um menor esforço computacional, que pode gerar um conjunto de informações relevantes para o processo sem ser necessário dispor de recursos computacionais significativos, ao invés de se analisar a forma original. Portanto, em cada iteração (i) calcula-se a distância máxima entre a curva simplificada e a curva original pelo cálculo da distância Hausdorff, que é uma métrica responsável por calcular o quão dois conjuntos no espaço estão distantes, como se destaca na subfigura 4.58 (a) segundo o algoritmo desenvolvido por Dimitrie Pompeiu e Felix Hausdorff [104]. Quando determinada esta distância através do módulo `cv2.HausdorffDistanceExtractor` do OpenCV, é realizada uma nova aproximação que passa no ponto correspondente ao local onde se calculou a maior distância Hausdorff, resultando numa nova aproximação, onde agora a curva aproximada ganha um novo segmento de reta, que passa agora neste ponto como se evidencia na subfigura 4.58 (b). É de se notar que na primeira iteração, à exceção de todas as outras iterações, a forma aproximada resulta apenas da união do primeiro e do último ponto da forma original. No AGV, este método é utilizado com o objetivo de calcular o número de arestas que cada elemento possui, ou seja, como todos os elementos captados praticamente são polígonos com 4 arestas, então elementos com áreas significativas, mas com um número superior ou inferior de arestas são identificados como ruído e eliminados, uma vez serem elementos externos que não interessam ao processo, reduzindo-se o número de falsos positivos.

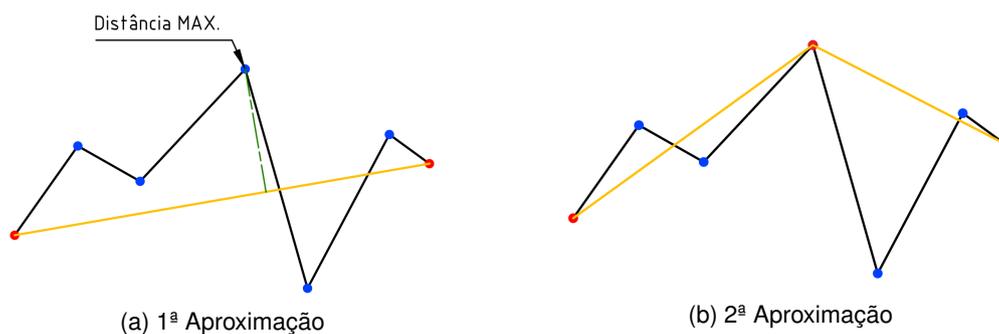


Figura 4.58: Algoritmo de Douglas – Peucker

Com estas condições, o sistema atual tem acesso a um conjunto de ferramentas que permitem isolar as cores do espectro cromático formando máscaras binárias personalizadas, a partir das quais se consegue formar contornos externos, o que possibilita o cálculo de áreas e oferece a inteligência necessária para reconhecer formas geométricas. Todos estes atributos juntos num único algoritmo, possibilita a constituição de um sistema rigoroso, onde a extração das características de interesse acontece de uma forma autónoma, convertendo-se imagens de 8 bits para um conjunto de dados que descrevem com precisão o conjunto inicial. Deste modo, o processo a este nível tem o objetivo de apenas analisar três elementos de interesse que podem surgir nas respetivas máscaras: guia, sinais de mudança de direção e o sinal de reset.

A identificação da guia é essencial para uma boa performance por parte do AGV. Sem a sua identificação, não é possível identificar mais nenhum sinal, pois sem o reconhecimento computacional da guia significa que o AGV não se encontra no trajeto e é ilógico detetar os restantes sinais, uma vez que estes não teriam influência alguma no estado de locomoção do AGV. Não existe nenhuma vantagem em detetar sinais fora do espaço físico da guia, uma vez que esses sinais não terão nenhum efeito colaborativo em qualquer processo do AGV. Ao contrário dos restantes sinais, que surgem de forma pontual ao longo dos trajetos, a guia é uma constante em todos os percursos, pois é devido à sua deteção que o AGV percorre de forma autónoma os vários trajetos programados. Por isso, sem a sua deteção, os restantes sinais perdem a sua identidade mesmo sendo detetados nas respetivas máscaras, transformando-se em elementos supérfluos para o processamento de imagem, como é visível na figura 4.59.

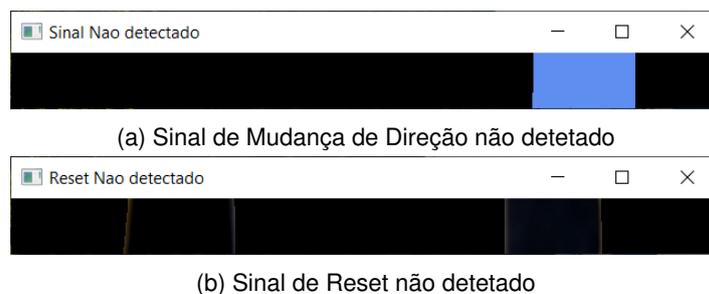


Figura 4.59: Elementos de interesse não detetados devido à ausência de guia

De forma prática, a deteção da guia é de fácil deteção. Isto porque possui uma forma retangular igual aos restantes elementos e em segundo, porque é o elemento a adquirir que possui o maior contorno, logo é o elemento que possui a maior área, visto não existirem lacunas nas formas a adquirir. A ordem de processamento destes elementos é efetuada com base na área de cada um, de forma decrescente, ou seja, do maior contorno para o menor, o que significa que o primeiro contorno em todas as iterações do algoritmo a analisar-se é o contorno da guia. Deste modo, se esta não for detetada, o algoritmo acaba por aqui não sendo possível navegar ao longo dos trajetos, porém, se detetada procura-se se os restantes contornos se identificam como um sinal de mudança de direção, ou como um sinal de reset, que diferente dos outros elementos é constituído por dois quadrados, que necessitam de ser captados em simultâneo para se contabilizar como um sinal de reset. Isto, porque poderia existir uma pequena diferença entre área deste sinal e a guia, o que aumentava as chances de más identificações, visto os contornos serem oriundos da mesma máscara binária.



Figura 4.60: Elementos de interesse detetados devido à presença de guia

4.3.8 Contadores

O reconhecimento dos sinais anteriores (sinal de mudança de direção e reset) necessitam de um mecanismo que contabilize o número de identificações que ocorrem, pois sem esta fiscalização não existe forma de controlar a quantidade de sinais adquiridos, tornando o AGV num sistema ineficaz no seguimento dos trajetos pré-definidos. Sem esta habilidade, o AGV perde-se nos próprios trajetos não conseguindo proporcionar um bom serviço, devido à incapacidade de se localizar ao longo do percurso operacional. A contabilização dos sinais funciona como um sistema de navegação indireto, pois permite ao AGV saber o número de sinais que já foram identificados e o número de sinais ainda por identificar. Note-se que na ausência da contabilização destes sinais, a presença física destes é inútil, tornando o mecanismo de seleção de trajetos da seção 4.3.10 igualmente supérfluo, uma vez que não existe forma de associar o sinal identificado a uma ação que o Utilizador definiu. Por estes motivos, a importância da integração de um sistema global de contadores dedicado a cada tipo de sinal, como forma de contabilizar todos os sinais mantendo o sistema sob controlo.

Existe um problema em relação à contabilização destes sinais, relativamente à presença de falsos-positivos provocando uma incorreta análise do trajeto. Destaca-se, que a contabilização correta de cada sinal é de extrema relevância a este nível, porque no caso de existir alguma má contagem, então o AGV pode acusar algum erro em termos de software, devido à incompatibilidade entre o que este identifica a partir da imagem captada, e a ação que este deveria realizar conforme o que o Utilizador programou. A aparição destes falsos-positivos ocorrem, quando o sistema contabiliza o mesmo sinal mais do que uma vez no mesmo local. Por exemplo, ao invés de se contabilizar um sinal de mudança de direção num troço, o AGV nesse mesmo troço contabiliza o mesmo sinal 8 vezes. Esta situação para além de provocar uma sobrecarga de dados no sistema, ainda torna possível a hipótese do último sinal contabilizado (que é o mesmo sinal da primeira aquisição), puder estar associado a uma ação diferente da qual o primeiro sinal aí identificado possui, ou seja, gera-se uma situação onde o AGV realiza uma ação não programada para esse troço, promovendo incoerências na seleção dos trajetos. Estas situações são recorrentes quando não se integra nenhum mecanismo de verificação, pois, como o tempo médio de cada ciclo de operação do sistema é na ordem dos milissegundos, e como a diferença espacial percorrida pelo AGV desde a primeira identificação do sinal até à próxima captura de imagem não é significativa, então esse sinal é contabilizado não uma mas várias vezes, identificando-o internamente como se tivesse a contabilizar diferentes sinais, dando origem a uma série de erros no algoritmo.

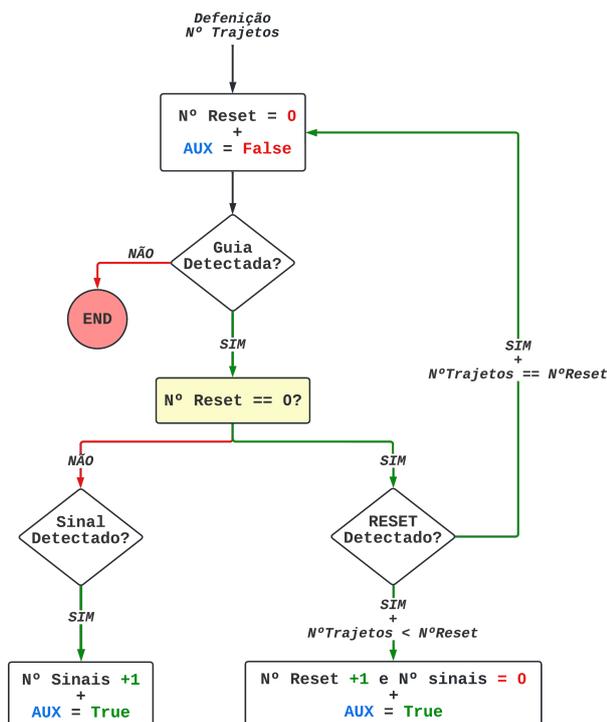


Figura 4.61: Algoritmo basilar para a contabilização de sinais

Evita-se problemas desta natureza, com a integração de um sistema de verificação como é evidenciado pelo fluxograma da figura 4.61 que passa pela criação de duas variáveis globais (N° Reset e N° Sinais) que auxiliam no processo de contagem de sinais. Basicamente, para evitar o efeito sequencial descrito anteriormente, um sinal só é realmente contabilizado, quando este é identificado um certo número de vezes de forma contínua, sendo que estas identificações são guardadas numa lista que assim que excede o limite estipulado, então o AGV contabiliza em +1 o número de sinais desse género contabilizado. Evita-se repetir o preenchimento desta lista, pela ativação de uma variável auxiliar ($aux = True$) que impede a contabilização de novas identificações mesmo que ainda se esteja a adquirir o sinal em questão. Só quando se deteta novamente apenas a guia, ou seja, quando se deixa de se identificar o sinal, é que esta variável volta ao seu estado padrão ($False$), que é o estado que permite a identificação livre de sinais. Repare-se que, embora a contagem dos dois tipos de sinais ocorra segundo o mesmo mecanismo, o sinal de reset possui influência sobre a contagem do sinal de mudança de direção. Ora, ao existir dois tipos de sinais, estes atuam de forma distinta na promoção da deslocação do AGV, sendo que o sinal de mudança de direção, como o nome indica, é responsável por identificar uma possível mudança de direção, uma vez que dependendo da programação realizada pelo Utilizador, este sinal indica ao AGV se muda de direção na próxima interseção, ou se continua em frente. Já o sinal de reset é o responsável por marcar o início e fim de um trajeto, o que perante esta condição, quando este sinal é detetado, a contagem de sinais de mudança de direção é reiniciada em 0, porque como se inicia num novo trajeto dentro dos vários definidos, então é necessário realizar uma “limpeza” ao contador que contabiliza estes sinais (N° Reset e N° Sinais = 0). Já o sinal de reset também sofre esta “limpeza”, mas é o próprio sinal que a realiza esta ação sobre si próprio, porque, quando se identifica o último trajeto a realizar (quando N° Trajetos == N° Reset) significa que na próxima passagem por este sinal, para além de se reiniciar o número de sinais de mudança de direção, também se reinicia o sinal de reset em 1 ao invés de 0, uma vez que o número de sinais de reset detetados equivale ao número de trajetos percorridos. Portanto, para manter esta similaridade é necessário reiniciar o reset sempre em 1, à exceção de quando se inicia o arranque no AGV e ainda não se identificou nenhum sinal de reset e por isso, somente nessa ocasião vale o N° Reset = 0.

4.3.9 Comportamentos Direcionais

Até este tópico, sempre se descreveu todos os algoritmos referentes a cada função integrante do software do AGV, de forma lógica e coesa com as respetivos fundamentos teóricos e inconveniências práticas presentes durante o seu desenvolvimento. Porém, ainda não se mencionou como o AGV através de visão artificial segue um dado trajeto integrado no piso, ou simplesmente, como segue uma guia. Face a esta questão, a resposta a esta incógnita é fundamentada no cálculo de uma característica geométrica designada por centroide. A este nível, o centroide é o ponto geométrico de uma figura plana bidimensional na qual é conseguido pela média aritmética de todos os pixels que a constituem [105][106]. No caso do cálculo do centroide da guia, o seu cálculo é realizado pela análise da máscara preta formada, e com base nela analisa-se tanto a área, como a forma geométrica desta, sendo estas as únicas características relevantes para a sua obtenção. Logo, como se analisam imagens bidimensionais, então o centroide é formado por duas coordenadas, dividindo-se em duas componentes (I_x , I_y) segundo as direções definidas, onde ambas são derivadas do cálculo dos momentos de imagem, que na área do processamento de imagem são definidos pelo valor médio das intensidades destes pixels segundo uma dada direção, facultando informações como a orientação, excentricidade e informações para o cálculo do centroide [107]. Não existem centroides em mais direções, porque, mais uma vez, as imagens adquiridas manifestam-se em planos 2D ao longo de duas direções já definidas. Portanto, segundo [19] em termos matemáticos, a definição de um momento de imagem é formalizado pela análise de uma função contínua bidimensional $f(x, y)$ com valores diferentes de zero, onde existem infinitos momentos de imagem (M_{pq}) de ordens p e q , de modo que um momento de imagem genericamente define-se como

$$M_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x, y) dx dy, \quad (4.23)$$

onde p e q são pertencentes ao conjunto dos números naturais (\mathbb{N}). Note-se que à semelhança da equação 4.22, a equação acima resulta de uma definição matemática genérica para um conjunto de pontos infinitos integrantes de \mathbb{R} , o que em termos práticos estas condições não se verificam, porque não se aplicam à natureza física deste problema. É impossível para este sistema, ou qualquer outro sistema computacional processar esta expressão com estas condições, independentemente do poder computacional a que lhe esteja associado, visto não terem a capacidade de analisar infinitos pontos e determinar uma solução exata. Ora, face a estas limitações é necessário tornar prático e eficiente o cálculo dos centroides, que tem como base o cálculo dos momentos de imagem, de forma a se conseguir analisar imagens digitais, evitando-se sobrecarregar o sistema com métodos com baixas performances. Portanto, é necessário adaptar a expressão acima para uma fórmula computacional e fisicamente correta, que respeite a física deste problema. Como se analisa imagens digitais bidimensionais de domínios limitados e fechados constituídos por sequências de pixels de diferentes intensidades, que digitalmente agem como pontos discretos com uma gama de intensidades também limitada, então face a estas particularidades os integrais de linha que antes integram funções definidas em domínios contínuos, agora são convertidos em domínios discretos [108], onde a soma de todos os pixels que formam uma imagem digital, resulta no cálculo de um momento imagem de qualquer ordem, de modo que

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y). \quad (4.24)$$

Neste caso, os momentos de imagem calculados advêm de planos bidimensionais, e, portanto, as duas componentes que formam a coordenada do centroide, resultam em momentos de imagem até à 1ª ordem, o que significa que p e q variam de $[0,1]$. Para além disso, o momento de imagem M_{00} , numa imagem binária equivale à área da forma em análise, visto a intensidade variar apenas entre dois valores e por isso essa coincidência, podendo-se utilizar a área já calculada anteriormente para otimizar o processo do cálculo do centroide. Recitando-se [108], o cálculo do centroide num plano 2D formula-se ao ponto de

$$(I_x, I_y) = \left\{ \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right\} = \left\{ \frac{M_{10}}{A}, \frac{M_{01}}{A} \right\}, \quad (4.25)$$

onde, M_{10} numa imagem binária corresponde à soma de todas as coordenadas diferentes de 0 na direção x , enquanto M_{01} corresponde à soma de todas as coordenadas diferentes de 0, só que na direção y . Com a definição destes conceitos e conseqüentemente com a definição do centroide, basta substituir cada momento de imagem pela respetiva expressão que resulta da equação 4.24, obtendo-se assim uma expressão apta para se integrar computacionalmente sem causar problemas ao nível da performance do AGV. Esta expressão é integrante nos módulos do OpenCV de modo que o centroide é dado por

$$(I_x, I_y) = \left\{ \frac{\sum_x \sum_y x I(x, y)}{\sum_x \sum_y I(x, y)}, \frac{\sum_x \sum_y y I(x, y)}{\sum_x \sum_y I(x, y)} \right\}. \quad (4.26)$$

Portanto, até este ponto demonstrou-se como deduzir o cálculo do centroide, contudo surge outras questões, do porquê da sua utilização e como se o utiliza, como mecanismo de orientação do AGV. Hipoteticamente num caso ideal, o AGV tem um desempenho perfeito quando segue uma guia sem nenhum desvio significativo. Este comportamento permite que o valor do centroide calculado em cada iteração, ou seja, em cada ciclo do algoritmo coincida com o valor do centro de imagem do ROI (C_x, C_y) , que é uma coordenada fixa semelhante ao centroide derivada das medidas definidas na região de interesse. Contudo, note-se que em qualquer aplicação real existem sempre erros associados ao processo, o que neste caso promove a não coincidência destes valores na grande maioria das situações.

Conseguir-se um seguimento bem-sucedido não é alcançável com uma metodologia que procure/obrigue tornar as duas propriedades coincidentes em cada iteração, porque mais uma vez essa tarefa é inexecutável. Ao invés desta abordagem, o objetivo realista no dimensionamento deste sistema, passa por se conseguir criar um intervalo com tolerâncias, de modo que o AGV tenha a capacidade de seguir um trajeto onde as diferenças face ao centro de imagem do ROI sejam as menores possíveis, e mais importante, conseguir que seja um processo consistente e estável ao longo do tempo, porque como em qualquer sistema dinâmico, quando não reunidas as condições para a estabilidade, o sistema torna-se instável e conseqüentemente incontrolável. Por isso, quanto maior for a diferença absoluta entre estas propriedades, então menos preciso é o seguimento por parte do AGV. Logo, a determinação aritmética desta diferença funciona como o principal mecanismo de avaliação, do quanto o AGV é preciso a seguir um dado trajeto. Para além disso, conforme o quanto expressiva é esta diferença, o AGV mecanicamente através do controlo motriz realiza os ajustes necessários através da modulação do sinal PWM nos motores elétricos, como forma de assegurar uma condução exemplar com a aplicação de ajustes singulares em cada motor, proporcionais face às diferenças existentes entre o centroide da guia e o centro de imagem do ROI.

4.3.9.1 Seguir em Frente

O cálculo do centroide da guia como mecanismo de orientação do AGV é na maior parte do tempo, o único sistema de orientação que o AGV utiliza para percorrer trajetos. Inevitavelmente, estes trajetos possuem interseções com outros troços do percurso, desde cruzamentos de guias (caso 1), trajetos que convergem na guia originando interseções (caso 2) e locais que possibilitam mudanças de direção (caso 3). O que existe em comum em todas estas situações é todas elas requererem uma decisão por parte do AGV, porque existem pelos menos sempre dois trajetos que se podem realizar. Nestes casos, calcular apenas o centroide para decidir por onde o AGV segue não é suficiente, pois o AGV entraria num estado de estagnação, uma vez não conseguiria decidir ou decidiria erradamente o trajeto a seguir. Face a estes problemas é necessário introduzir novos mecanismos que auxiliam na orientação do AGV, por isso nestas duas próximas seções descreve-se novas ferramentas complementares ao cálculo do centroide, com o intuito de auxiliar na correta tomada de decisões, de modo a tornar compatível com os trajetos planeados pelo Utilizador e evitar comportamentos não programados. Deste modo, o AGV ao atual momento possui apenas dois comportamentos programados face à capacidade de seleção de percursos de forma automatizada. Estes comportamentos, como foi mencionado na seção 4.3.8 são essenciais, quando por exemplo se adquire um sinal de mudança de direção, onde o comportamento físico do AGV é dependente da ação programada pelo Utilizador, que neste caso ou segue em frente, ou realiza uma mudança de direção. Em todas as restantes situações (caso 1 e caso 2), quando o AGV intercepta múltiplas interseções na guia onde segue, o comportamento padrão do AGV é seguir em frente, como se detalhará ao longo desta seção. Destaca-se que a capacidade de seleção do comportamento a conduzir nestas situações é exclusivamente dependente da inteligência computacional implementada e da capacidade de processamento de imagem do AGV, não existindo nenhuma intervenção do Utilizador neste procedimento, salvaguardando a definição dos trajetos.

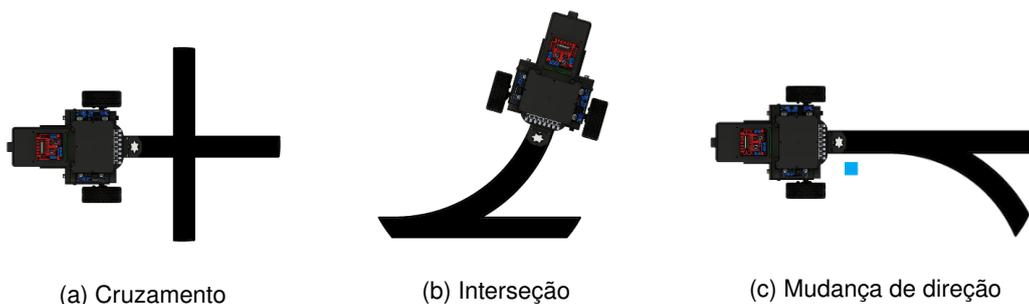


Figura 4.62: Situações com múltiplos troços

Em resumo, o AGV tem a capacidade de articular entre dois modos de atuação: mudar de direção e seguir em frente. Só existem possíveis mudanças de direção, quando se identifica no piso um sinal de mudança de direção, uma vez que em todas as restantes situações incluindo seguir uma única guia, o AGV percorre os trajetos no modo: seguir em frente. Portanto, este modo tem o objetivo de garantir ao AGV um comportamento distinto, que independentemente do número de interseções que se formam com a guia onde este circula, este tem a capacidade de ignorar os restantes troços e com isso eliminar as hipóteses de ocorrerem mudanças de direção involuntárias por um erro no algoritmo, nomeadamente devido ao cálculo incorreto do centroide de um troço que não é a guia onde este circula. Por isso, face aos requisitos que se desejam integrar neste algoritmo, este perfil computacional deve lidar com diferentes situações, onde o resultado final não deve ser influenciado por fatores externos nem por singularidades de cada situação. Estas situações dividem-se em três categorias, como se indicou na página anterior, estando destacadas no algoritmo da figura 4.63. A situação mais benevolente em termos de esforço computacional, fora seguir uma única guia que é situação mais simples de processar, é quando o AGV interceta um cruzamento. Nesta situação, o AGV passa de uma área comum da guia, para uma área muito superior de forma abrupta, pois o troço perpendicular do cruzamento rapidamente preenche todo o plano de aquisição, ou seja, toda a região de interesse. Isto leva, que o centroide da guia nesta situação passe a ser designado antes por centroide do cruzamento, que nesta situação coincide em quase toda a sua extensão ao centro de imagem do ROI, porque a máscara preta nestes instantes é toda preenchida por pixels brancos. Portanto, como esta é uma situação pontual, repentina e de breve duração, então apenas se espera que o AGV siga este “novo” centroide, embora o seu cálculo não seja muito preciso devido às grandes proporções geométricas do cruzamento.

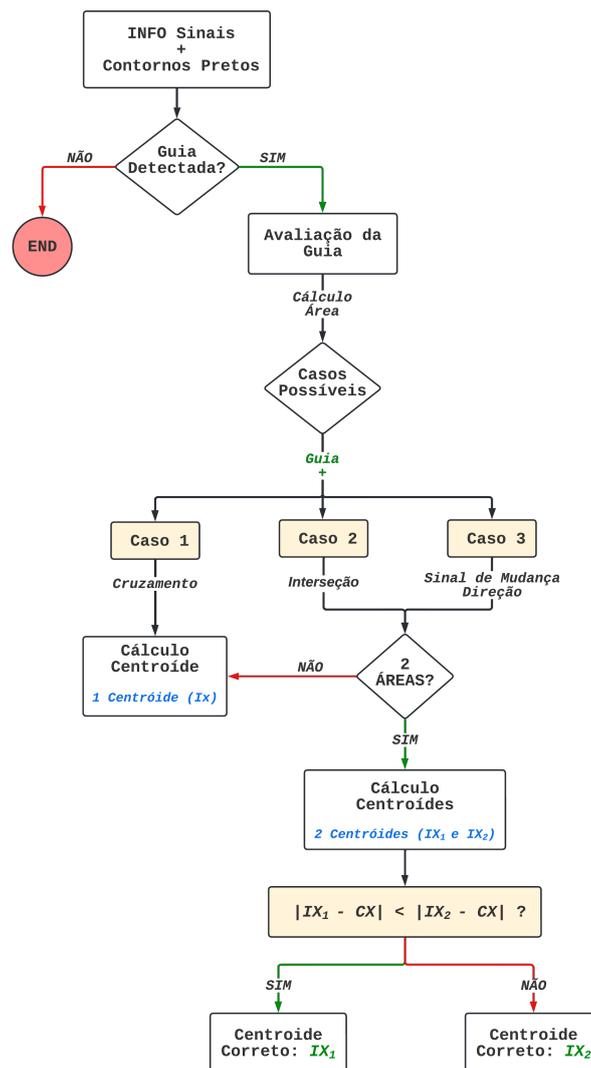


Figura 4.63: Algoritmo para seguir em frente

As restantes duas situações são mais complexas em termos computacionais, devido à quantidade de diferentes layouts integrados no piso, que proporcionam análises mais difíceis com mais estágios de avaliação, visto a dinâmica destas situações. Portanto, quando se pretende seguir em frente num sinal de mudança de direção existem três estágios de avaliação. A primeira fase após a identificação do sinal de mudança de direção é seguir a guia normal, pois ainda existe uma distância sem quaisquer troços até ao local onde se efetua a mudança de direção. Até aqui calcula-se o centroide de forma normal, sem nenhuma característica especial que influencie no seu cálculo como se evidencia na subfigura 4.64 (a). Contudo, ao aproximar-se do local que permite a mudança de direção, a guia começa a divergir formando uma bifurcação, onde a área desta porção de guia aumenta de forma proporcional conforme o AGV avança. Este é a segunda fase, pois agora forma-se uma área a cada iteração cada vez maior, o que leva ao deslocamento geométrico do **centroide** (**ponto verde**), uma vez que a área é agora formada pela guia de interesse mais a área do troço divergente levando ao centroide a deslocar-se no sentido deste novo troço. Como se pretende que o AGV nesta situação siga em frente, então esta singularidade não é benéfica para um seguimento correto da guia, porque nestas condições o centroide calculado leva ao AGV a ter desvios significativos. Impede-se este mau seguimento através do cálculo de um novo centroide, não da forma demonstrada neste trabalho, mas com base nos limites da área formada, ou seja, em vez de se calcular o centroide com base na área e na forma geométrica, assume-se que esta fase é uma condição excecional, e por isso é necessário calcular o centroide de um modo “manual”, de forma a conseguir-se manter um seguimento correto. Simplesmente, dependendo do local para onde o centroide tendencialmente se desvie, então, como resposta o algoritmo produz um centroide fictício na direção oposta com um valor de offset fixo em relação ao limite vertical da área formada, como é evidenciado na figura 4.64 (b). Neste caso, o centroide original é negligenciável, passando-se a seguir o centroide fictício produzido pelo algoritmo.

Chegará a um ponto onde o aumento desta área é interrompido dando origem a duas áreas de menores dimensões como se verifica na subfigura 4.64 (c), pois como o troço divergente afasta-se cada vez mais da guia ao longo de um raio de curvatura a cada iteração, então segundo o plano de visão do ROI, esta área formada até aqui tendencialmente crescente colapsa, levando à formação de duas áreas distintas que são identificadas como guias. Esta identificação por parte do algoritmo marca o início da terceira fase, desencadeando um problema, uma vez que ao existir duas guias identificadas, então o algoritmo calcula dois centroides o que leva ao AGV a entrar num estado de indecisão, porque nesta posição ele não tem capacidade de decisão para saber qual o centroide correto a seguir. Evita-se tais circunstâncias através da avaliação da diferença absoluta entre o valor de cada centroide e o **centro de imagem do ROI** (**ponto branco**), ou seja, o centroide que estiver mais perto deste valor teórico é o centroide correto a seguir, porque ao assumir-se que o AGV percorre os trajetos de forma precisa, então quando chega a esta fase já se encontra centrado com a guia que pretende seguir, enquanto o troço divergente encontra-se mais afastado, originando uma maior diferença absoluta o que leva a negligenciar este centroide. Quando já não se deteta o troço divergente, então é porque já se passou o local onde o AGV poderia mudar de direção, retornando-se ao processamento de uma guia.



Figura 4.64: Mudança de Direção: Seguir em frente

Já nas interseções, o único comportamento computacionalmente permitido é seguir em frente não existindo hipóteses de ocorrerem mudanças de direção nesses locais, uma vez que não existem aí sinais de mudança de direção. Porém, apesar da inexistência destes sinais nestes locais, o algoritmo ainda pode sofrer algum erro no processo e mudar de direção de forma inesperada e não programada, embora seja uma hipótese remota como se verificará pelos resultados no próximo capítulo. Caso aconteça este comportamento não é congruente nestas circunstâncias, porque existem locais dedicados para se realizarem estas mudanças de direção. Portanto, apesar das semelhanças com a situação anterior, diz-se que uma interseção é qualquer situação onde exista a convergência de duas guias unindo-as numa só, enquanto num local que permita a mudança de direção ocorre o processo contrário, nomeadamente uma divergência de guias a fim de promover novos trajetos. Por isso, face às parecenças com a situação anterior, exceto o domínio de comportamentos permitido nessa situação, do ponto de vista computacional analisar uma interseção, ou uma mudança de direção com o objetivo de seguir em frente é exatamente o mesmo processo, e por isso se utilizar o mesmo algoritmo sem características extras para se analisar. É de se destacar, que a ordem dos estágios de avaliação também pode ser idêntica, embora menos comum, uma vez que a situação mais genérica é analisar-se de modo inverso ao que acontece num sinal de mudança de direção, ou seja, a primeira fase é constituída pela identificação de duas guias, visto estas estarem a convergir sendo necessário escolher o centroide correto para permitir um seguimento de acordo com o esperado. Porém, diferente da situação anterior, como as guias convergem para um local comum, então à medida que o AGV avança a distância entre as suas áreas diminui até se formar uma grande área, que tendencialmente também diminui com o avançar do AGV. Nesta fase, é necessário calcular também um centroide fictício, uma vez que o centroide calculado está sob influência desta nova área que alberga a guia a seguir mais o troço convergente, levando à introdução de erros na marcação do centro geométrico (centroide) e por isso a sua exclusão, embora a cada iteração o centroide aproxima-se cada vez mais do valor correto, visto a convergência das duas guias. Esta substituição do centroide acontece até que a área formada seja menor que um valor estipulado que quando atingido, então é porque se alcançou a última fase. Ao alcançar-se esta etapa é porque se ultrapassou com sucesso a interseção e volta-se a processar novamente apenas uma guia a partir do seguimento do seu centroide.

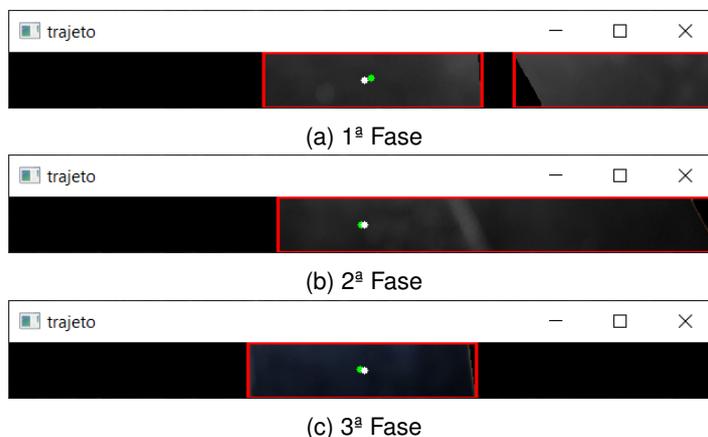


Figura 4.65: Interseção: Seguir em frente

4.3.9.2 Mudar de Direção

Oferecer a capacidade de mudar de direção em qualquer ponto que possibilite esta ação é das características mais importantes deste trabalho. Tipicamente, qualquer sistema deste género não usufrui desta habilidade, recorrendo antes a um único trajeto linear que passa em todos os locais de interesse por cada volta, mesmo que não seja necessário a sua passagem em todas as estações. Esta característica torna estes sistemas menos eficientes, menos flexíveis e mais complexos de se integrarem, pois quando se pretende adicionar algum novo ponto de interesse, esta adição pode não ser trivial devido à sobreposição com o percurso já marcado no piso. Neste trabalho, com um menor número de guias consegue-se criar um sistema mais flexível e com menos limitações desta natureza, mantendo um estilo minimalista.

O design do percurso operacional como se verá na seção 5.1 integra facilmente estes locais de mudança de direção, sendo intuitivo para qualquer Utilizador identificar fisicamente o local e a direção que o AGV pode assumir nestas ocasiões. Portanto, em termos computacionais, para dotar o AGV desta habilidade, é necessário à semelhança do modo anterior desenvolver um perfil computacional dedicado, agora com a sensibilidade necessária para favorecer o AGV a mudar de direção nos respetivos locais, de acordo com os trajetos pré-definidos pelo Utilizador. Teoricamente, este favorecimento deve ser uma obrigação, porque este perfil é destinado apenas a instâncias onde é realmente necessário ocorrer uma mudança de direção, ou seja, mesmo que o AGV identifique um local que possibilite a mudança de direção, caso não esteja programado para mudar de direção neste troço, então este perfil não é ativado, pois apenas se o ativa, quando se pretende efetivamente mudar de direção.

O motivo desta exclusividade recai na necessidade de se conceber um algoritmo rigoroso e computacionalmente preciso nas ações que concretiza, na medida que o comportamento que aqui se pretende dotar o AGV é de todas as situações até aqui mencionadas, o comportamento mais exigente computacionalmente e o que negativamente impacta mais, se a sua análise for mal processada, culminando numa incompatibilidade entre os trajetos planeados pelo Utilizador e os trajetos que o AGV na realidade executa. Falhas ao nível da mudança de direção são desastrosas para este sistema, porque ao ser hipoteticamente um mecanismo defeituoso, então a legibilidade deste sistema é posta em causa, uma vez que não se garante dentro do aceitável que este sistema cumpra os trajetos programados. Por isso, é indispensável garantir que esta função analise esta situação com o maior detalhe possível, dentro do tolerável sem comprometer as restantes funções até agora descritas, a fim de garantir uma elevada taxa de sucesso ao analisar corretamente estas situações que são o cerne deste sistema de navegação. Portanto, independentemente do trajeto a seguir, quando se identifica um sinal de mudança de direção e este indica que o AGV deve mudar de direção neste troço, então aplica-se o algoritmo da figura 4.66 detalhado ao longo da próxima página.

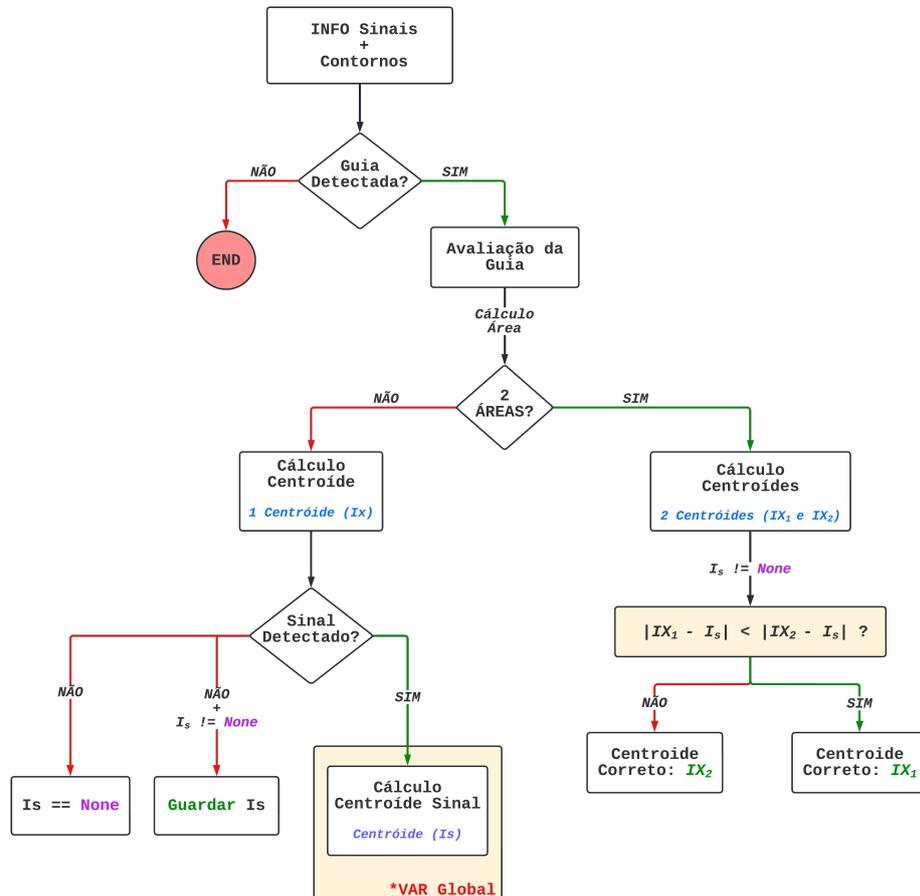


Figura 4.66: Algoritmo para mudar de direção

Para o AGV mudar de direção, do ponto de vista computacional este necessita de processar três estágios de avaliação à semelhança do modo: seguir em frente apresentado na última seção. Ambos os modos possuem parecenças, porém a primeira fase deste novo perfil computacional é onde existe as maiores diferenças devido à nova metodologia integrada. Portanto, a primeira fase de avaliação destacada na subfigura 4.67 (a) iniciando-se pela identificação do sinal de mudança de direção, que apenas ativa este perfil, caso esteja pré-definido pelo Utilizador mudar de direção no trecho onde se encontra este sinal. Caso contrário, este perfil nunca é ativo como já se mencionou. Ao confirmar-se a veracidade desta situação com o auxílio dos contadores, então para além do cálculo do centroide da guia, que neste caso não apresenta nenhuma anomalia ou irregularidades causadas por trechos paralelos à guia, então calcula-se também o centroide do sinal de mudança de direção (I_s) ficando armazenado numa variável global, sendo algo particular deste processo. Realiza-se o cálculo do centroide deste sinal, porque será com base na sua localização, que as próximas fases serão organizadas e processadas, a fim de promover o correto seguimento do centroide do trecho a seguir. Note-se que a localização de cada sinal de mudança de direção encontra-se sempre situado do lado do trecho divergente, ou seja, se existe a possibilidade de mudar de direção à esquerda, então este sinal encontra-se sempre localizado à esquerda da guia, contudo se pretender-se mudar de direção à direita, então o sinal de mudança de direção encontra-se sempre localizado à direita da guia. Não existem exceções a esta regra, pois esta distribuição fixa permite facilitar o processamento através de visão artificial pela redução do número de variáveis, mas também permite auxiliar o ser humano em termos visuais, a identificar intuitivamente os locais onde ocorrem mudanças de direção e as respetivas direções que se podem tomar nestes trechos.

Com estas características realiza-se a transição para a segunda fase destacada na subfigura 4.67 (b), quando a área da guia começa a aumentar de forma contínua a cada iteração, devido à bifurcação formada entre a guia e o trecho divergente. Conforme o AGV avança, a bifurcação gera uma área comum que inclui a área da guia mais a área do trecho divergente, impactando na forma como se calcula o centroide da guia, pois à medida que o AGV avança esta área aumenta e a localização do centroide é cada vez mais influenciada por este bloco formado. Nesta situação, esta tendência adquirida pelo centroide é inicialmente benéfica para auxiliar na mudança de direção, porém não é o suficientemente impactante para colaborar neste comportamento, uma vez que à semelhança da situação anterior, o centroide com esta localização não beneficia o AGV nem a seguir em frente, nem a mudar de direção ficando numa posição intermédia que não valoriza nenhum dos comportamentos programados. Portanto, este centroide nesta fase é negligenciável, e o algoritmo nestas condições produz um centroide fictício que se encontra localizado a uma distância de offset de um dos limites da área formada, de forma que esta distância é definida pelo Utilizador, como metade da largura da guia convertido em pixels, ou seja, 90 px. A escolha da localização do limite de referência recai nas coordenadas do centroide do sinal de mudança de direção detetado na primeira fase, ou seja, se $I_s > 320$, então escolhe-se o limite da direita como referência, visto a posição do sinal de mudança de direção estar à direita da guia, porém, se $I_s < 320$, então escolhe-se o limite da esquerda como referência, pelo motivo oposto. Note-se que 320 é o valor ao longo da direção x que corresponde a coordenada intermédia do centro de imagem do ROI.

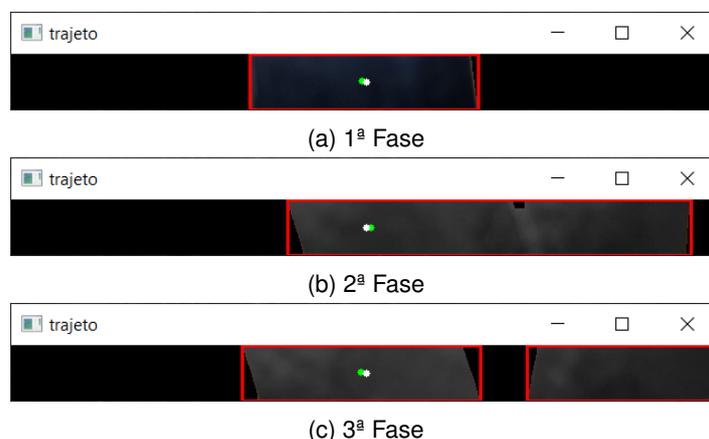


Figura 4.67: Mudança de Direção: Mudar de direção

Chega-se a um ponto onde o avanço do AGV é tal, que do ponto de vista do ROI esta área formada pela guia e pelo troço divergente colapsa, resultando em duas áreas de menores dimensões sendo ambas reconhecidas internamente como guias como é destacado na subfigura 4.67 (c). Esta identificação por parte do AGV está em parte incorreta, porque uma das áreas não é a guia, mas sim o troço divergente que nesta situação é o elemento a seguir a partir do cálculo do seu centroide. Independentemente destas atribuições, como o sistema identifica duas guias, então o algoritmo calcula dois centroides deixando o AGV num estado de indecisão, uma vez que este não tem os artifícios necessários para identificar qual o centroide correto a seguir. Resolve-se esta dualidade de centroides, através da exclusão de um deles a partir do valor armazenado do centroide do sinal de mudança de direção à semelhança do que se realizou na fase anterior. Portanto, efetua-se a diferença aritmética entre o centroide do sinal de mudança de direção e o centroide de cada elemento, de modo que o elemento que tiver a menor diferença numérica, então é o centroide correto a seguir, porque ao assumir-se que o AGV percorre os trajetos de forma precisa, então quando chega a esta fase já se encontra tendencialmente centrado com o troço divergente que pretende seguir, enquanto a guia encontra-se mais afastada, originando uma maior diferença o que leva a negligenciar o seu centroide.

4.3.10 Tomada de Decisão

Toda a informação relevante para o seguimento do AGV, do ponto de vista do processamento de imagem já foi processada ao longo das seções anteriores, não existindo mais nenhuma função dedica a rentabilizar novos dados, uma vez que todos os atributos proveitosos para este sistema já se encontram computadorizados e devidamente estruturados. Porém, ainda existe uma lacuna associada à cognição deste sistema em relação à gestão autónoma dos comportamentos direcionais. Assim, necessita-se de incluir no AGV uma inteligência artificial primária capaz de reconhecer diferentes situações, e com base nessas identificações comportar-se de modo a cumprir com os trajetos pré-definidos pelo Utilizador, efetuando uma gestão autónoma e ininterrupta do comportamento físico do AGV. É fundamental incluir esta inteligência artificial, que em parte deseja-se ter parecenças básicas com a inteligência do ser humano, apenas ao nível do reconhecimento de situações definidas, e conseqüentemente na tomada de decisões coesas em prologue das necessidades de cada instância. A este nível, apenas se realiza uma gestão dos dados, de forma a desencadear no final deste processo uma ação física por cada iteração computadorizada, com o objetivo de proporcionar um correto seguimento do AGV em função dos vários trajetos definidos. Esta é a função de mais alto nível de todo o sistema, integrada no ficheiro `tools.py` e face às suas características, esta é uma função exclusivamente desenvolvida em python sem a integração de módulos de visão artificial.

Portanto, esta função inicializa-se com a divisão dos vários trajetos pré-definidos pelo Utilizador, de modo a efetuar-se toda uma segmentação com o objetivo de organizar estes dados logo na primeira iteração, poupando recursos computacionais em todas as etapas subseqüentes. Feita esta reestruturação dos dados advindos do “dicionário” do ficheiro `percursos.py`, inicia-se o arranque do AGV, mas chama-se à atenção que até este detetar o primeiro sinal de reset, que é o sinal responsável por marcar o início/fim de um trajeto, o AGV ignora quaisquer sinais de mudança de direção, uma vez que o AGV neste modo não contabiliza nenhuma situação desta categoria, porque o único objetivo neste estágio é detetar somente o primeiro sinal de reset, para se dar início à realização dos trajetos programados. Por isso, nesta fase, o comportamento de mudança de direção encontra-se bloqueado permitindo-se somente ao AGV a seguir a guia principal, no modo seguir em frente sem a possibilidade de mudar de direção. Esta lógica justifica-se, na medida que é imprescindível iniciar todas as variáveis relacionadas à gestão dos trajetos somente quando é adquirido o primeiro sinal de reset, evitando hipóteses aleatórias de ocorrer uma identificação de um sinal de mudança de direção e com isso provocar algum erro no processo, ou algum incremento não programado no contador responsável por gerir estes sinais. Logo, até este ponto em termos computacionais, o AGV está num estado preliminar a preparar todos os seus mecanismos relacionados à gestão de trajetos, com o objetivo de os ativar, mas não os iniciar, porque a sua inicialização só tem fundamento, quando se deteta o primeiro sinal de reset, que é o sinal responsável por marcar o início oficial do seguimento do AGV.

Com a deteção do sinal de reset, o AGV já se encontra habilitado para executar decisões autónomas sem a influência direta do Utilizador, mas sempre sob tutela dos trajetos pré-definidos, de modo que o número de trajetos a realizar tal como as suas denominações encontram-se armazenados na variável `keys`. Por segurança, cada `key` apenas possui informação sobre o trajeto lhe atribuído, de modo que a cada novo sinal de reset identificado, o conteúdo das variáveis referentes à gestão do trajeto passado incluído a respetiva `key` é eliminada, uma vez que estas informações são irrelevantes para os próximos trajetos. Por isso, após a limpeza das variáveis a cada sinal detetado, estas variáveis são complementadas com a informação somente referente ao trajeto que se processa nessa instância, uma vez que esta importação de dados de forma seletiva, advém do conjunto de informação definido inicialmente neste algoritmo, permitindo tornar este processo computacionalmente mais eficiente. Da informação que se importa encontra-se o código de cada comportamento direcional que o AGV deve executar armazenado na variável `value`, quando identifica situações de mudança de direção, uma vez que nas restantes situações, por padrão este segue sempre em frente. Por isso, ao estarem programados somente dois comportamentos direcionais, então existem apenas dois códigos definidos como se detalhou na seção anterior. Caso o Utilizador defina um código fora deste domínio, então ocorre um erro, visto não existir uma correspondência computacional para esta situação. Note-se que independentemente do comportamento a adotar e das situações em análise, este é o algoritmo responsável por estabelecer a ponte contínua entre o que se processa computacionalmente e a ação física desencadeada com base no comando imposto pela ponte H nos motores elétricos.

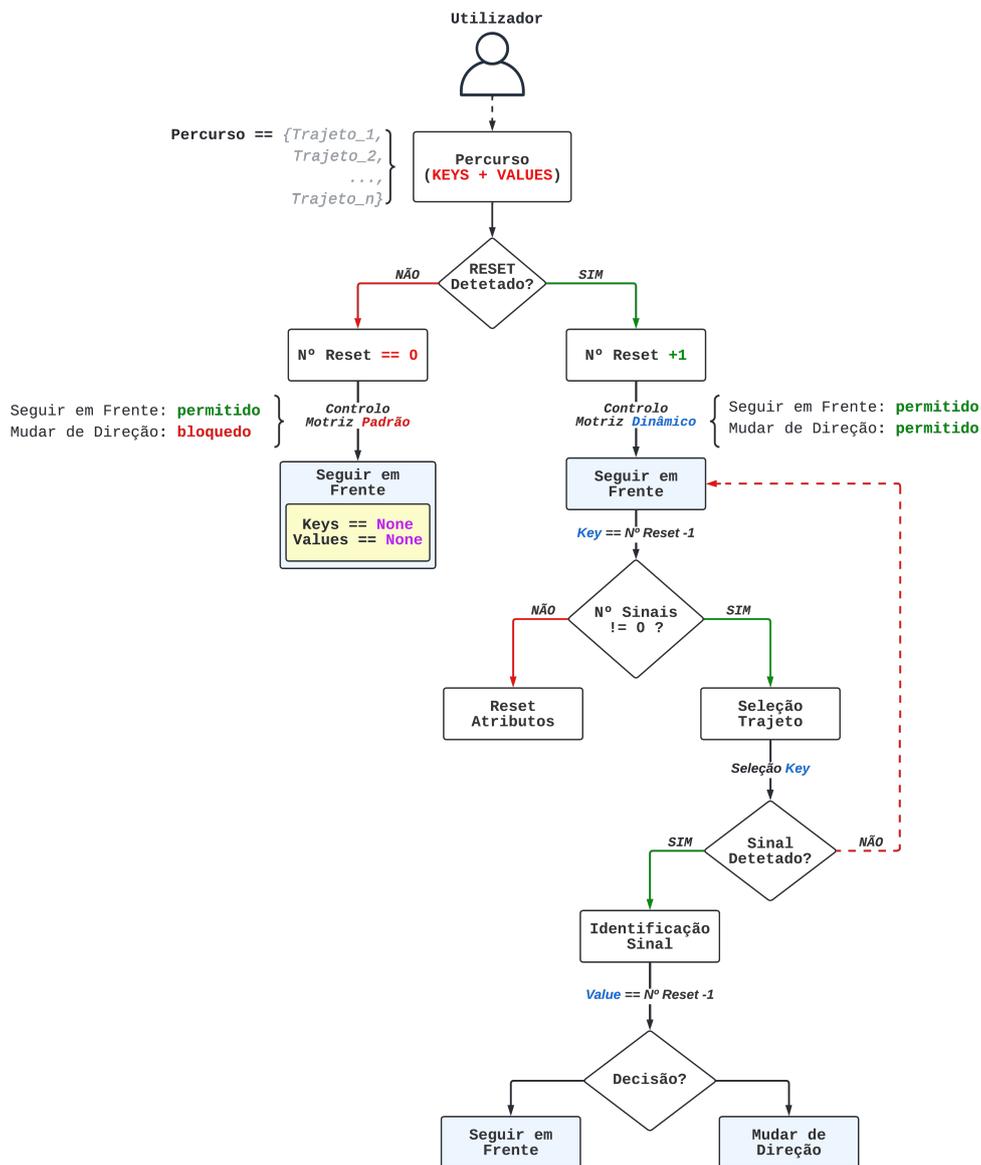


Figura 4.68: Algoritmo basilar para a tomada de decisão

4.3.11 Definição de Rotas

Todos os trajetos que o AGV realiza são fruto da sua aptidão para se deslocar autonomamente. Esta habilidade capacita o AGV a tornar-se num sistema com condições de operação contínuas e unicamente dependente do layout físico do percurso operacional para se movimentar. Este grau de independência é alcançado graças à programação efetuada pelo Utilizador ao nível do ficheiro `percursos.py`, na qual é uma programação intuitiva e de fácil aplicação. A planificação que o Utilizador realiza neste ficheiro é apenas referente a locais críticos que envolvem situações dúbias do ponto de vista computacional, em relação à escolha do comportamento a realizar nos locais onde existem múltiplas hipóteses de se executarem diferentes seguimentos. Um exemplo destas situações críticas são os locais que possibilitam mudanças de direção. Por isso, face a estas variâncias no percurso, define-se um trajeto como uma combinação sequencial de códigos referentes a estes locais críticos, e um percurso como um conjunto de todos os trajetos pré-definidos pelo Utilizador. Independentemente do percurso formado, qualquer trajeto não trivial possui locais críticos que necessitam do auxílio da função apresentada na seção anterior, para decidir que comportamento o AGV assume nestas situações. Recorda-se que apenas existem dois comportamentos disponíveis para se integrar nestes locais, apesar de um deles ser o comportamento padrão que é seguir em frente, enquanto o comportamento remanescente é visto como a única exceção à regra normal (mudança de direção), mas note-se que se existissem mais comportamentos mais códigos haveriam e mais combinações seriam possíveis de se integrar. Portanto, às condições atuais existem somente dois códigos computacionais: **0** para seguir em frente e **1** para mudar de direção. A sequência numérica formada por estes códigos dita como o AGV se comporta em locais de mudança de direção, que a este nível coincidem inteiramente com todos os locais críticos assinalados. É obrigatório definir um código para cada local sem exceções, uma vez que ao não se cumprir este preceito, o sistema falha devido a falta de informação previamente definida pelo Utilizador. Ao ter-se em atenção este fator, pode-se definir qualquer trajeto viável dentro do percurso formado, graças às diferentes combinações de códigos definidos de forma sequencial, permitindo obter trajetos mais eficientes e dinâmicos.

Esta logística é computacionalmente conseguida devido à definição dos trajetos num tipo de classe exclusivo desta linguagem de programação que são os dicionários. Estes são um tipo de dados de classe `<'dict'>` mutáveis entre si com propriedades associativas, que permitem conservar correspondências entre valores (*values*), não sendo elementos sequenciais, uma vez que os itens ai contidos não são organizados segundo uma ordenação explícita, como numa *string* ou *lista* onde as sequências de dados são indexadas através de números naturais com determinadas posições. Em oposição, nos dicionários estes elementos são indexados por chaves (*keys*), de modo que estas chaves podem ser qualquer tipo de dados à exceção de listas, como números, strings e tuplos, caso não incluam itens mutáveis, visto uma chave ter a capacidade de ser um item imutável. Portanto, um dicionário define-se como um conjunto não ordenado de coletividades segundo o formato `{key_1:value_1, key_2:value_2, ... key_n:value_n}`, existindo um único valor para cada chave. Escolheu-se os dicionários como elementos responsáveis pela gestão dos trajetos, pois são elementos nativos do python com uma elevada capacidade computacional, com características de organização mais robustas face a outros tipos de dados, para além da facilidade de manipulação de itens. O tamanho e características do dicionário constituído dependem do layout do percurso operacional formado e, portanto, como exemplo ilustrativo, ir-se-á analisar o seguinte percurso para detalhar todas estas singularidades aplicadas a este sistema.

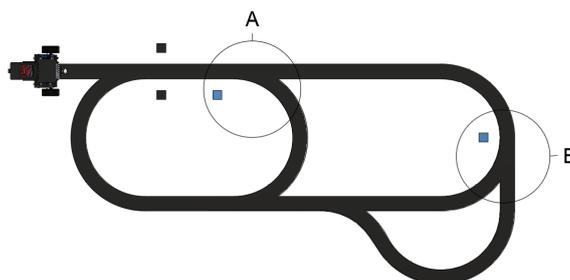
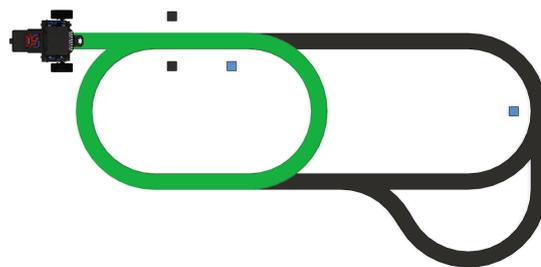
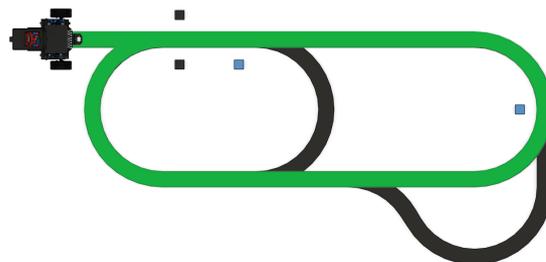


Figura 4.69: Exemplo de um percurso operacional

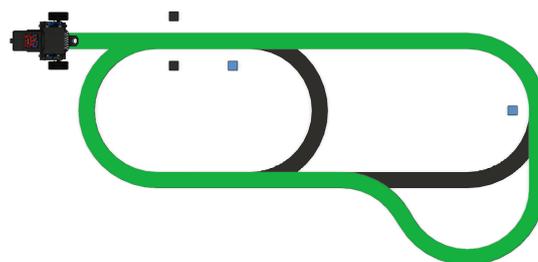
Neste percurso modelo existem duas situações de mudança de direção, que correspondem aos únicos locais críticos a assinalar (situação A e B). Das combinações possíveis resultam três trajetos distintos com o mesmo sentido de deslocamento, mas com diferentes rotas devido às diferentes sequências de códigos definidos ao nível do ficheiro `percursos.py`. O trajeto mais curto ('T1') apenas processa um sinal de mudança de direção (situação A), enquanto o trajeto mais longo ('T3') processa todos os sinais de mudança de direção deste percurso (situação A e B). Para efeitos elucidativos, o trajeto ('T2'), embora não seja o trajeto mais longo nem o mais curto, é uma rota que também analisa as mesmas duas situações do trajeto ('T3'), ainda que programado com uma sequência de códigos diferente. Independentemente do trajeto definido, não existe nenhuma obrigatoriedade na definição de trajetos com o mesmo número de locais críticos a analisar, porque cada trajeto é definido de forma singular e livre das escolhas implementadas para os restantes trajetos, não existindo nenhuma relação que force a definição de sequências de códigos com o mesmo tamanho ou número de sinais analisados. Logo, quando se define um trajeto apenas se tem em consideração os sinais que este irá diretamente processar, uma vez que todos os outros sinais que ficam fora do seu campo de visão, pertencentes a outras frações do percurso, não são incluídos nas sequências de código formadas. Cria-se assim um sistema intuitivo e lógico, na medida que apenas se define no dicionário formado os sinais que diretamente influenciam nas tomadas de decisão do AGV. Todos os sinais fora das rotas formadas não são contabilizados evitando-se criar um sistema ilógico e confuso. Assim, para este percurso exemplificativo, as rotas assinaladas pelo AGV encontram-se ilustradas na figura 4.70, com o objetivo de destacar os trajetos exequíveis.



(a) 1º Trajeto 'T1'



(b) 2º Trajeto 'T2'



(c) 3º Trajeto 'T3'

Figura 4.70: Trajetos exequíveis dentro do percurso definido

Todos os trajetos assinalados da figura 4.70 só são efetivamente praticáveis pelo AGV, caso o Utilizador os defina a todos no dicionário formado `'percursos'`, tendo a opção de escolher livremente quais os trajetos que realmente planeia integrar. Não existe nenhuma regra que obrigue a definição de todas as rotas possíveis oriundas de um dado layout do percurso, pois conforme as necessidades de cada sistema é que se define o número de trajetos primários para auxiliar nos requisitos das várias situações à qual este sistema presta assistência. Para efeitos explicativos, encontra-se na tabela 4.6 todas as sequências de códigos que originam fisicamente os trajetos anteriormente apresentados, com o objetivo de demonstrar o facilitismo que é a definição de rotas a partir de sequências de códigos, salientando algumas particularidades computacionais deste mecanismo.

Existe apenas um único dicionário formado para cada percurso. A sua síntese, à semelhança do que se apresentou na página 88, é este ser composto por um sistema de chaves e valores, na qual o conjunto de todos os dados forma os vários trajetos codificados por sequências de códigos imutáveis durante o funcionamento do AGV. Relembra-se que a alteração dos trajetos apenas ocorre, quando o AGV não está a funcionar nem a executar nenhuma função de seguimento. Neste sistema, as chaves são strings que intitulam os vários trajetos programados (`'T1'`, `'T2'` e `'T3'`), tirando-se vantagem deste tipo de dados, uma vez que é possível nomear trajetos com designações semelhantes às terminologias onde este sistema é aplicado. Para além disso, torna a identificação dos trajetos mais imediata e simples. Já os valores não beneficiam desta qualidade, pois para além de serem definidos por códigos (números inteiros), também se encontram guardados dentro de um outro tipo de dados que são as listas, embora ainda no interior do dicionário `'percursos'`. Existe esta gestão dos códigos referentes aos comportamentos direcionais, porque para cada chave formada existe somente um valor associado. Por isso, caso se defina diretamente estes códigos sem estarem integrados numa lista, os trajetos ficam limitados a apenas um código, porque nestas condições a lista que antes integrava um aglomerado de códigos vista como um único valor atribuído à respetiva chave, agora é vista uma série de valores individuais atribuídos a uma única chave. Esta situação não é tolerável por este tipo de estrutura computacional, levando a uma falha no sistema surgindo um erro de sintaxe que é um erro recorrente, quando se define incorretamente alguma estrutura de dados. Neste caso, a estrutura malformada é o dicionário. Evita-se esta situação através da integração destes códigos em listas imutáveis, que funcionam como um único valor, onde os vários códigos se encontram guardados num formato indexado do género $[0, 1, \dots, n]$. Escolheu-se este tipo de dados para hospedar estes códigos ao invés de tuplos, pois as listas são um tipo de dados mutáveis que permitem modificar os itens contidos nelas, ao contrário dos tuplos, o que desbloqueia um novo conjunto de métodos e mecanismos que os tuplos não tem acesso. Esta característica é essencial para este sistema, pois o algoritmo descrito na seção anterior utiliza frequentemente métodos que modificam a lista original formada, através do acesso direto a um elemento de posição i para aplicar uma dada alteração.

```
percursos = { 'T1' : [1],
              'T2' : [0, 1],
              'T3' : [0, 0]
            }
```

Tabela 4.6: Exemplo de definição de trajetos no ficheiro `percursos.py`

Em resumo, a definição de rotas ao atual nível de desenvolvimento caracteriza-se por ser uma metodologia simples, mas não trivial, que ameniza significativamente o nível de intervenção que o Utilizador realiza para programar trajetos. Um benefício deste sistema é a facilidade com que o Utilizador se familiariza com este sistema de programação, beneficiado de uma alta flexibilidade, tanto ao nível da nomeação dos trajetos, como ao nível da definição dos códigos dos comportamentos direcionais, embora como principal ponto a melhorar, quanto maior for a complexidade do percurso formado, mais difícil é a definição intuitiva destes trajetos, sem se recorrer um mapa com o layout do percurso para auxiliar na formação dos trajetos.

4.3.12 Controlo Motriz

Percorrer trajetos sem desvios é o principal objetivo do sistema motriz do AGV. Nestas condições hipotéticas, o AGV proporcionaria uma condução perfeita sem qualquer afastamento físico e por isso não existiram erros de *offset* perante a guia a seguir. Atingir este pico de performance é utópico, pois em qualquer sistema dinâmico como o AGV, existem sempre variáveis incontroláveis no processo, especialmente externas, que promovem o surgimento de erros que influenciam no desempenho de todo o sistema, tornando este estado impossível de se alcançar em qualquer situação realista, independentemente da otimização realizada ao nível do software e hardware. Por não ser alcançável este estado impraticável, não significa que o comportamento do AGV não seja de excelência em termos práticos, pois mesmo com desvios em relação à guia visível nas subfiguras 4.71 (a) e (c), estas variações desde que não ultrapassem as tolerâncias impostas não são consideradas como erros, mas sim como variações normais do processo, que não afetam de forma significativa a qualidade do seguimento. Nesta qualidade, como forma de preservar um seguimento contínuo e preciso à semelhança da subfigura 4.71 (b), o AGV conta com mecanismos integrados que amenizem as variações mais extremas, conduzindo a uma estabilização deste ao longo do tempo como em qualquer sistema estável. Todos os mecanismos que influenciam neste controlo dinâmico já foram detalhados, inclusive aprofundados em termos de hardware, como em termos do software, porém de toda a informação exposta ainda se requer informação computacional sobre o algoritmo preparatório integrado no ficheiro `motores.py`, que serve em exclusivo como base computacional para albergar todas as definições ao nível dos GPIO do Raspberry Pi, como na definição dos comportamentos motrizes praticados pelo AGV, que dependendo da situação em que este se encontra, diferentes funções motrizes são ativadas para promover o melhor seguimento possível e garantir a qualidade de condução do AGV.

Estas funções motrizes são definidas individualmente na classe (`'Motor'`) que é uma estrutura computacional que agrupa todos os dados relacionados ao controlo motriz do AGV numa única localização. A criação desta classe computacional permite formar novos objetos e com isso novas instâncias facilmente importadas em diferentes localizações do código sem a necessidade de se incorporar todas as definições antecedentes definidas do ficheiro original, salvaguardando a modularidade do algoritmo com a formação de instâncias ilimitadas. Cada objeto oriundo desta classe formada está associado a um número limitado de métodos, que conservam as características fundamentais de uma linguagem de programação orientada a objetos, como por exemplo os mecanismos de herança que permitem definir múltiplas classes e herdar os respetivos métodos de classes pré-definidas sobrescrevendo-os. Com este sistema de classes, evita-se a repetição de tarefas através da invocação de diferentes instâncias, facilitando a composição de um código modular e claro, onde as maiores vantagens em termos de desenvolvimento manifestam-se pela fácil manutenção/atualização do sistema, integração e leitura, ganhando-se globalmente uma melhor legibilidade do algoritmo reduzindo-se o risco de aparição de erros. De outro modo, repetir o ajuste das respetivas variáveis em cada instância formada seria uma prática redundante, levando a uma má otimização do algoritmo. Deste modo, a classe formada garante a união das várias funções definidas destinadas à gestão motriz do AGV em quatro métodos distintos que comandam o funcionamento dos motores elétricos promovendo os seguintes comportamentos físicos: virar para a direita, seguir em frente, virar para a esquerda e parar.

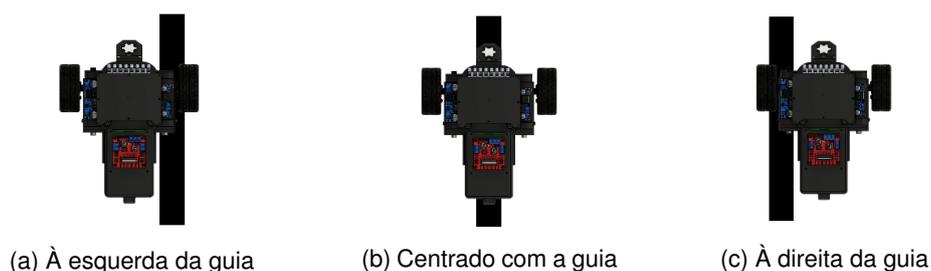


Figura 4.71: Localização do AGV face à guia a seguir

Em resumo, os comportamentos motrizes definidos como métodos na classe ('Motor') encontram-se na tabela 4.7 com as respectivas combinações ao nível dos motores elétricos, destacando-se o comportamento individual de cada motor.

Motor A	Motor B	Ação motriz
0 ³	0	Parar
-1 ⁴	1	Virar à esquerda
1 ⁵	-1	Virar à direita
1	1	Seguir em frente

Tabela 4.7: Modo de funcionamento dos motores elétricos na formação dos comportamentos motrizes

A combinação destas ações motrizes em conjunto com o controlo imposto pelo sinal PWM, permite criar um cardápio de comportamentos motrizes a partir dos diferentes ajustes entre o domínio de velocidades programáveis e os sentidos de rotação que alternam entre o sentido horário e o sentido anti-horário. Destes dois atributos, o Utilizador apenas tem influência direta na escolha da velocidade de rotação dos motores elétricos, configurando através do sinal PWM a potência de cada motor. Quanto maior for a velocidade selecionada, mais potente será o sistema, embora se perca precisão no seguimento da guia, uma vez que para o mesmo tempo de processamento, o AGV percorre uma maior distância, o que proporciona um maior espaço para existirem desvios mais significativos em relação à guia a seguir. Logo, cabe ao Utilizador gerir em função do percurso formado, a velocidade mais indicada para embutir neste sistema. De todo o modo, o algoritmo está desenvolvido para suportar velocidades superiores, ainda que com uma menor precisão. Este suporte contínuo deriva do mesmo mecanismo utilizado nos comportamentos direcionais – o cálculo do centroide da guia. Portanto, a combinação destas ações em simultâneo permite ativar somente um comportamento motriz por cada iteração processada, a partir da análise da diferença entre o valor do centroide da guia e o centro de imagem do ROI, na qual o resultado desta diferença é comparado posteriormente com três intervalos pré-estabelecidos. A criação destes intervalos numéricos leva ao AGV a priorizar o seguimento no intervalo central, que é o intervalo responsável por ordenar o AGV a seguir em frente. Nos casos em que a diferença entre estes atributos é superior ao limite estipulado (limite superior ou inferior), então o AGV deixa de seguir em frente e conforme a sua posição face à guia a seguir, então passa a favorecer a viragem deste num sentido que possibilita o AGV a retornar novamente ao centro da guia e a reentrar no intervalo central. Estes três comportamentos motrizes são as ações mais comuns, de modo que o AGV só realiza uma paragem sem a influência direta do Utilizador, quando este deixa de detetar a guia, mas note-se que numa condução exemplar este comportamento motriz nunca é ativo. Por isso, a sua integração é apenas justificada como um mecanismo de segurança, visto num serviço ótimo este modo raramente se ativar. Caso se ative, este comportamento tem uma prioridade superior aos restantes e independentemente da circunstância em que o AGV se encontre, prioriza a sua paragem.

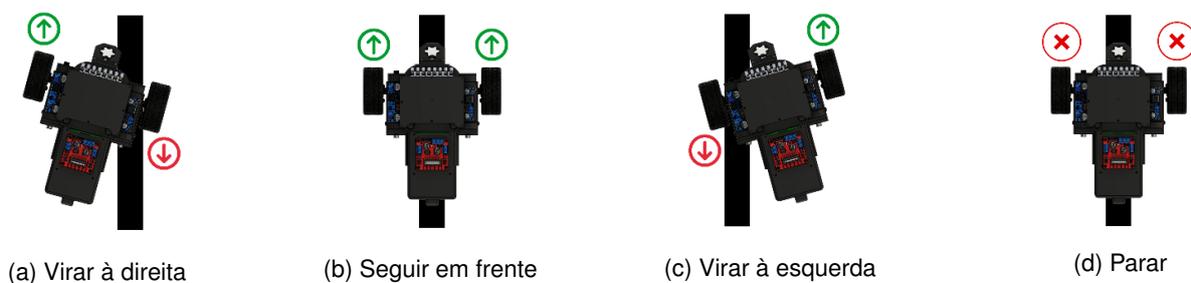


Figura 4.72: Comportamentos motrizes

³Motor desligado.

⁴Motor ligado no sentido anti-horário.

⁵Motor ligado no sentido horário.

Capítulo 5

Testes e Resultados

Com a conclusão do desenvolvimento técnico do AGV apenas falta aferir a performance realista deste sistema no seu ambiente de funcionamento, através de ensaios que permitem retirar resultados e respetivas conclusões, quanto ao modo de funcionamento dos vários subsistemas e da sintonia entre eles como um todo. Os testes aplicados dividem-se em duas categorias, os testes que visam descrever a performance do AGV através de um conjunto de ferramentas com um teor estatístico-probabilístico baseando-se em atributos matemáticos para classificar qualitativamente o comportamento dos dados obtidos, enquanto a segunda vaga de testes descreve quantitativamente os resultados alcançados baseando-se em conjuntos de amostras independentes, como forma de suportar as evidências destacadas, evitando possíveis incertezas quanto à singularidade dos comportamentos descritos, a partir da criação de populações de amostras com maiores dimensões com o objetivo de justificar as características patenteadas. Fora esta divisão de testes, como forma de diminuir as variáveis envolvidas na peritagem do AGV, nomeadamente nas provas relacionadas ao comportamento motriz deste, estas foram executadas num percurso operacional fixo, descrevendo os mesmos trajetos ao longo dos vários ensaios. Esta medida preventiva promove um maior controlo no processo de aquisição e reduz o risco de ocorrer misturas de dados adquiridos em diferentes condições. Dados com diferentes origens, neste caso, de diferentes trajetos não podem ser comparados entre si, pois não evidenciam padrões nem demonstrem características específicas dos comportamentos avaliados, impedindo uma correta análise das situações praticadas. Com isto, para qualquer teste realizado independentemente das características avaliadas, os dados provenientes dos vários ensaios executados revêm-se como um conjunto de informações íntegras e inalteráveis sem qualquer manipulação, como forma de avantajar virtudes e amenizar qualidades aquém das expectativas. Portanto, assumindo a veracidade desta premissa, todos os testes realizados descrevem o sistema desenvolvido na sua íntegra sem quaisquer modificações tendenciosas.

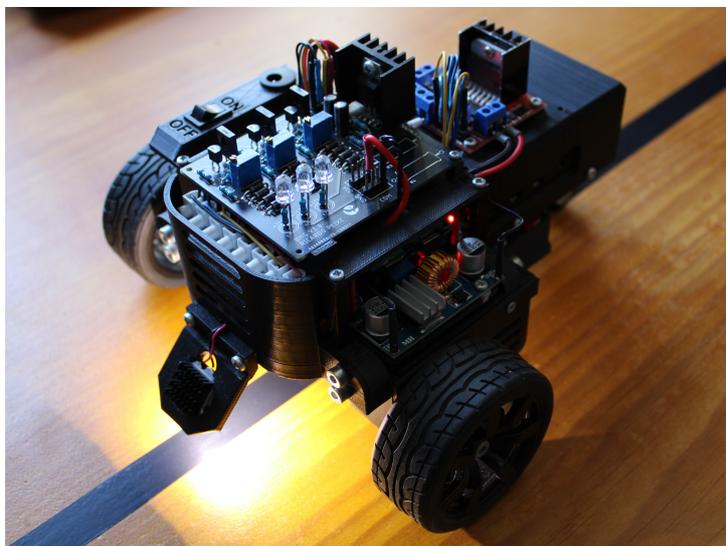


Figura 5.1: AGV desenvolvido em funcionamento

5.1 Percurso Operacional

O layout de orientação do AGV assenta num sistema convencional, com a integração de mudanças de direção, cruzamentos e atalhos, permitindo a existência de vários circuitos, tornando-o num sistema flexível, apesar de estar fisicamente limitado às guias integradas no piso. É um sistema unidirecional com uma navegação fechada, o que leva ao AGV a circular somente num sentido de deslocação evitando possíveis bloqueios, caso se integre mais do que um AGV no mesmo percurso operacional. Para além disso, o facto de se praticar apenas um único sentido de deslocação permite diminuir a complexidade do software, visto o AGV não necessitar de saber em que sentido se está a deslocar. A novidade implementada face a outros sistemas deste género em termos computacionais, fora todo o processamento de imagem realizado pelo Raspberry Pi através de visão artificial, é a flexibilidade existente na customização dos circuitos sem a alteração do layout físico, que em grande parte é conseguida devido aos vários algoritmos computacionais expostos no capítulo anterior. Deste modo, o percurso operacional que alberga os testes descritos ao longo das próximas seções é o percurso esquematizado na figura 5.2 sendo que se procurou produzir um design compacto, para fins demonstrativos com a integração de diversos elementos, de modo a se proporcionar diversas situações distintas a fim de se avaliar, como o sistema responde face à variedade de circunstâncias que este pode encontrar num ambiente real. Para mais detalhes acerca deste layout, encontra-se no anexo A o desenho técnico deste percurso com as respetivas cotas dimensionais.

A produção deste percurso realizou-se pela colagem de fita adesiva no piso, sem a utilização de ferramentas exclusivas à exceção de uma fita métrica, como forma de se garantir as dimensões estipuladas para este. Note-se que se escolheu uma tonalidade preta para as guias e o para o sinal de reset (elemento A), pois é uma cor que facilmente contrasta com a maioria dos pisos em ambientes industriais, uma vez estes serem maioritariamente em tons claros. Já os sinais de mudança de direção (elementos B, C e D), pelo contrário, apresentam uma tonalidade azul por dois motivos: evidenciar estes sinais dos restantes elementos do percurso como locais críticos onde existe a possibilidade de mudança de direção, sendo visualmente mais apelativo a tonalidade escolhida, e igualmente destacarem-se da tonalidade do piso, como um sinal com uma tonalidade associada genericamente a uma obrigatoriedade, ou seja, nesse local o AGV é obrigado a seleccionar uma guia, de modo coerente com os trajetos pré-definidos pelo Utilizador. Poder-se-ia ter utilizado tinta, ao invés de fita adesiva para formar os trajetos que no conjunto produzem o percurso operacional dimensionado, porém, visto este ser um sistema modelo de prova e apenas ser utilizado para efeitos demonstrativos aliado ao facto de terem existido constantes alterações no layout do percurso, então a utilização de fita adesiva torna-se na melhor solução, caracterizando-se como uma solução mais económica e de fácil aplicação. Contudo, esta opção necessita de um maior nível de manutenção por causa da facilidade em acumular resíduos que obstruem partes da guia, influenciando negativamente no processamento de imagem.

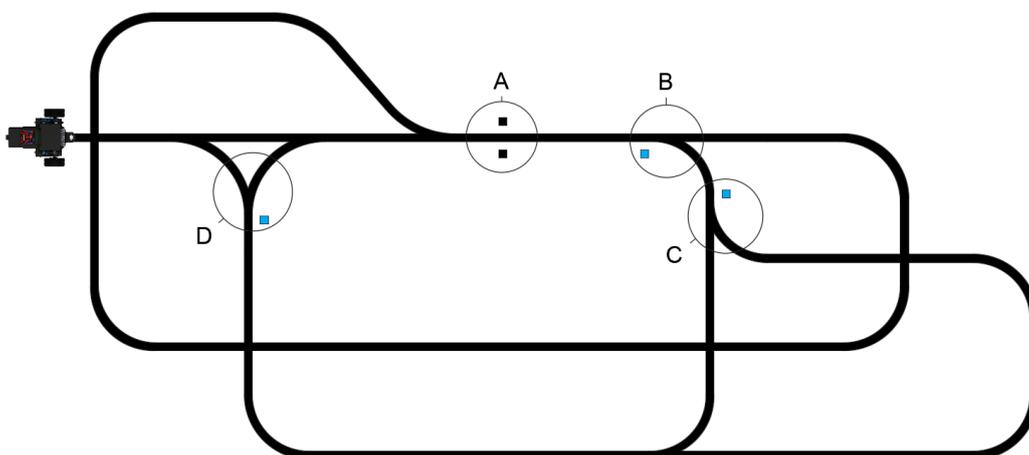


Figura 5.2: Representação esquemática do percurso operacional

5.2 Gestão Energética

Na demanda atual tecnológica em qualquer sistema elétrico, existe cada vez mais a procura por se conceber sistemas com melhores performances energéticas com a motivação de se reduzir as perdas elétricas que se manifestam maioritariamente sob a forma de calor residual. Quanto maior as perdas elétricas, maior são os prejuízos, tanto para o sistema a que se expõe esta energia térmica adicional e não recuperável, como monetariamente gerando maiores prejuízos económicos. No caso do AGV, todo o sistema elétrico que integra este sistema à exceção do carregamento da bateria elétrica está sob influência de conversores de tensão com eficiências acima de 92% de acordo com as características detalhadas pelos fornecedores e, portanto, aplicar otimizações a este nível não é praticamente exequível, uma vez que caso se aplicasse alguma alteração/-melhoria, muito provavelmente não teriam significância para justificar essa ação. Para além disso, ao serem placas comerciais estas já foram otimizadas para oferecer energeticamente os melhores resultados. Porém, do conjunto de sistemas elétricos integrados no AGV, a BMS desenvolvida é o único sistema modelado de raiz, tanto em termos de arquitetura eletrónica, tanto como em termos de fabricação não comercial que não apresenta quaisquer testes práticos que permitam analisar a sua performance num sistema real. Assim, como este é um dos sistemas mais importante do AGV e o mais importante em termos energéticos, esta seção é motivada a estudar a eficiência dos processos energéticos envolvidos nesta placa e dos comportamentos da BMS em diversos níveis, de modo a aferir as características realistas desta placa.

5.2.1 Corrente de Carga

A BMS desenvolvida encontra-se integrada numa PCB responsável por realizar o balanceamento das células de lítio de uma forma segura, não pondo em risco a segurança do Utilizador, nem a integridade física do AGV, através de uma correta gestão do carregamento destas células, que quando mal gerida pode culminar em problemas graves devido ao teor dos produtos que constituem este tipo de células. Tendo em conta estes fatores, selecionou-se um método de carregamento que assenta na utilização de uma tensão constante à saída do regulador de tensão LM317T, para provocar o balanceamento das células ao longo de cada ciclo de carregamento (*constant voltage method*). Empiricamente, este método é superior a outros métodos de carregamento no requisito de segurança, pois naturalmente conta com mecanismos que evitam mais facilmente a sobretensão das células e respetivos efeitos colaterais por exposições excessivas fora das suas gamas de funcionamento. A grande particularidade deste método é a manutenção de uma tensão constante ao longo de todo o estágio de carregamento, que por consequência leva à diminuição da corrente de carga ao longo do tempo, pois como a tensão à saída do regulador de tensão linear é fixa e à medida que as células carregam e caminham para um SoC (*State of Charge*) de 100%, então a diferença de potencial elétrico entre a tensão sentida nas células e a tensão imutável do regulador LM317T leva a uma diminuição gradual da corrente de carga ao longo do estágio de carregamento.

Este comportamento evidencia a principal desvantagem deste método, o elevado tempo de carregamento. Por um lado, garante-se um carregamento mais seguro, pois na fase terminal do carregamento a corrente de carga já é significativamente inferior àquela praticada no início deste estágio, e por isso mais dificilmente se incrementa a tensão das células para um mesmo período dificultando as células a atingirem um estado de sobretensão elétrica. Por outro lado, o principal desafio para um carregamento segundo esta metodologia é definir um valor de tensão que equilibre a velocidade de carregamento, de modo a evitar carregamentos muito demorados, mas também evitar os elevados fluxos de corrente que se formam no início do processo, podendo levar à formação de uma corrente de carga superior à corrente nominal suportada pelas células integrantes da bateria elétrica. É necessário salvaguardar a integridade das células, pois caso não se respeite as suas características de funcionamento promove-se um envelhecimento prematuro delas e uma redução substancial da sua performance. Deste modo, o comportamento descrito pela corrente de carga pode ser analisado no gráfico 5.3, onde se efetuou três ensaios independentes ([Ensaio 1](#), [Ensaio 2](#), e [Ensaio 3](#)) realizados com condições idênticas ($3.68 \pm 0.01V$), quando se iniciou o processo de carregamento.

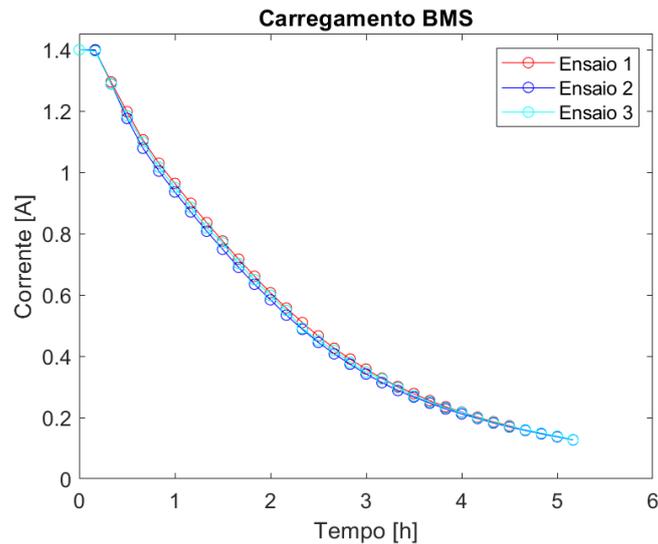


Figura 5.3: Corrente de carga ao longo de 3 ciclos de carregamento

O gráfico 5.3 formado pelos dados adquiridos suporta a premissa apresentada, verificando a diminuição da corrente de carga ao longo do tempo, em função do aumento do SoC. Esta diminuição não é constante durante o processo de balanceamento, sendo mais significativa nas primeiras fases, enquanto nas últimas etapas o comportamento da corrente de carga tende a estabilizar e variar menos ao longo do tempo. Note-se, que inicialmente existe um período onde a corrente é constante estando limitada somente pelo limitador de corrente a 1.4 A como se dimensionou. O facto de se integrar este limitador permite não só proteger a BMS e as células de lítio de uma corrente pejorativa a estas, acima das suas especificações, mas também leva à tensão de saída do LM317T a ser inferior à tensão estipulada, porque como se limita a corrente a um valor fixo, então a tensão de output a que se submete as células vai subindo gradualmente neste mesmo período até atingir o valor dimensionado. Neste período, o balanceamento das células é controlado pela corrente ao invés da tensão e, portanto, quando a tensão dimensionada para o regulador LM317T alcança o valor teórico, o sistema passa a ser limitado pela tensão ao invés da corrente, e por isso a corrente deixa de ser constante e começa a diminuir pelos motivos já mencionados. Apesar de se implementar um método que se fundamenta apenas na aplicação de uma tensão constante, devido à implementação de um limitador de corrente, nas primeiras fases existe uma etapa similar a um método que utiliza antes a corrente, como uma constante para realizar o balanceamento das células. Para além disso, e como se verá na seção 5.2.3 com mais detalhes, o efeito das perdas energéticas no LM317T sob a forma de calor residual influenciam a performance deste em conduzir corrente, e por isso existem maiores gradientes na corrente de carga nas primeiras etapas do balanceamento comparativamente com as últimas, onde as temperaturas são expressivamente mais baixas. Fora isso, consegue-se efetuar carregamentos seguros que duram entre 5 a 6 horas.

5.2.2 Tensão de Carga

Todas as células de lítio são quimicamente diferentes umas das outras. Não existem células iguais mesmo provindas do mesmo lote de produção, pois devido às próprias diferenças eletroquímicas de cada célula, formam-se diferentes velocidades de carregamento. A este nível, levante-se um problema, pois em baterias que possuem conjuntos de células, que é o caso do sistema de alimentação integrado, torna-se necessário integrar uma BMS para realizar o balanceamento de cada célula, de modo a equalizar as várias velocidades de carregamento. Como se evidenciou na seção 4.2.2, o principal objetivo da BMS é lidar com estas diferenças químicas e ajustar-se individualmente, de modo a proporcionar um carregamento ajustável a cada célula, evitando que estas operem fora da gama normal de funcionamento. Deste modo, o principal teste a que se pode submeter a BMS desenvolvida é conseguir-se classificar a sua performance, avaliando-se como esta realiza o balanceamento das células até atingirem a sua capacidade máxima.

Para avaliar a performance da BMS na área do balanceamento, efetuou-se um ensaio dedicado realizando-se a aquisição dos dados de uma forma contínua e igualmente segmentada em intervalos de 10 minutos, recolhendo-se informações sobre a tensão de carga que é a tensão aos terminais da bateria durante o processo de carregamento. Neste caso, foi medida a tensão de carga em cada par de células em paralelo, de modo a ocorrerem medições nos três packs (Pack 1, Pack 2, e Pack 3), onde cada pack é constituído por duas células em paralelo, visto estar-se a avaliar uma bateria com layout 3s2p, onde por cada célula em série existem duas células em paralelo. Deste modo, a aquisição destes dados proporcionou o gráfico 5.4, que marca desde o início até ao fim, a evolução temporal da tensão de carga em cada pack ao longo de um ciclo de carregamento.

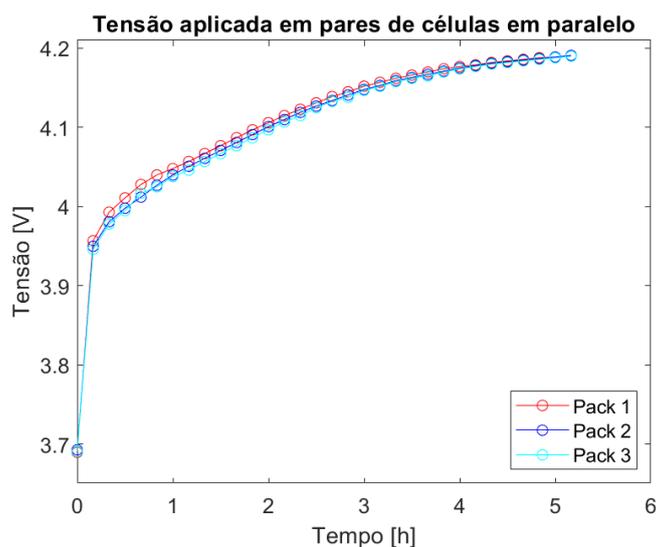


Figura 5.4: Tensão de carga ao longo de 1 ciclo de carregamento

Pela análise do gráfico 5.4 confirma-se o comportamento crescente da tensão de carga ao longo do tempo, de forma não constante à semelhança da corrente de carga com tendência a estabilizar em 4.2V. Existe somente esta estabilização, pois como se verifica um aumento gradual da capacidade energética das células de lítio ao longo do processo de carregamento, este comportamento leva a que a diferença entre a tensão de carga e a tensão fixa para o regulador de tensão LM317T seja cada vez menor, e por isso leva a existir uma desaceleração deste comportamento crescente, onde nas fases finais o ganho de tensão nas células é cada vez menor, contrastando com os altos ganhos destacados no início do processo. Note-se, que esta estabilização ocorre em torno de 4.2V, ao invés de 12.6V, pois está-se a analisar individualmente cada pack dos dos três packs integrantes desta bateria. Como o conjunto destes packs forma uma bateria elétrica com uma configuração do tipo 3s2p, isto significa que ao somar-se em cada período as três respetivas tensões, obtém-se diretamente a tensão de carga da bateria elétrica, e esta sim, tende a estabilizar em torno dos 12.6V, quando se completa o balanceamento da bateria. Deste modo, com os resultados apresentados comprova-se que as células não entram num estágio de sobretensão elétrica, pois cada pack não apresenta uma tensão acima de 4.2V, atingindo um SoC ligeiramente acima de 99%, que para todos os efeitos é um excelente indicador da gestão energética aplicada pela BMS. Para além disso, para melhorar estes resultados, verifica-se também a existência de uma convergência das tensões de carga de cada pack, onde no final as diferenças são tão insignificantes que apenas se observa um dado no mesmo espaço temporal ao invés de três. Isto significa que todas as células integrantes da bateria elétrica possuem uma tensão de carga praticamente igual no final do processo de balanceamento, solidificando assim os bons resultados apresentados pela BMS desenvolvida. De facto, este balanceamento homogéneo das células é somente destacado no final do processo, uma vez que durante todo o resto do procedimento não se verifica esta conduta, devido essencialmente à influência mais marcante das perdas energéticas que ocorrem no LM317T sob a forma de calor, criando-se um mecanismo prejudicial que limita o processo de carregamento originado estas maiores variações.

5.2.3 Rendimento Energético

Para tornar a BMS desenvolvida numa placa mais flexível e compatível com diferentes fontes de alimentação, esta conta com um regulador de tensão linear, o LM317T responsável por gerar uma tensão de saída fixa independentemente da tensão de entrada. Apenas se garante esta condição, quando a tensão de entrada é no mínimo superior a 1.5V em relação à tensão de saída definida, visto este ser um regulador do tipo *step-down* regulável. Neste caso, como se pretende definir uma tensão à saída de 12.6 V, então basta fornecer 14.1V (12.6V + 1.5V). No entanto, é incomum encontrar uma fonte de alimentação que forneça exatamente 14.1V ou mesmo 12.6V dispensando a utilização do regulador. Face a esta característica, justifica-se na fase de desenvolvimento a integração de um regulador de tensão, porque atualmente em qualquer área tecnológica, caminha-se num sentido convergente com o propósito de se reduzir as interfaces/padrões exclusivos de várias marcas comerciais, trocando-os antes por um conjunto limitado de padrões normalizados, a fim de universalizar e facilitar a sua integração, aumentado neste caso, a gama de fontes de alimentação possíveis de se integrar com esta BMS. Ao ganhar-se esta flexibilidade, perde-se por outro lado na eficiência energética do processo de balanceamento, pois como o LM317T é um regulador de tensão linear, então quanto maior for a diferença entre a tensão de entrada (U_i) em comparação com a tensão de saída (U_o), maior serão as perdas energética sentidas neste elemento. Estas perdas (P_{loss}) manifestam-se sob a forma de calor, sendo fisicamente manifestadas pelo aumento da temperatura no regulador, de modo que

$$P_{loss} = (U_i - U_o) \times i. \quad (5.1)$$

Estas perdas não são recuperáveis, nem utilizáveis para qualquer fim técnico e, portanto, quanto maior as perdas, maior a ineficiência do processo. O aumento da temperatura no regulador LM317T é um sinal direto da expressividade das perdas energéticas emitidas, porque para além da diferença de tensão entre a tensão de entrada e tensão de saída, o fluxo de corrente conduzido pelo regulador (i) é o principal responsável por formar estas perdas energéticas. Isto porque, como já foi verificado nas seções anteriores, grande parte do processo ocorre com uma tensão constante o que significa, que o único fator que varia ao longo do tempo é a corrente de carga, e note-se que as maiores temperaturas coincidem com os períodos onde existem os maiores fluxos de corrente. Existe uma única exceção a este princípio, que é o primeiro dado adquirido. Este período em específico, marca o início do processo de balanceamento, e por isso ainda apresenta uma temperatura igual à temperatura ambiente, contrastando logo em seguida em todos os ensaios representados no gráfico 5.5, como o período onde existe o maior pico de temperatura, sendo unânime em todos os ensaios realizados.

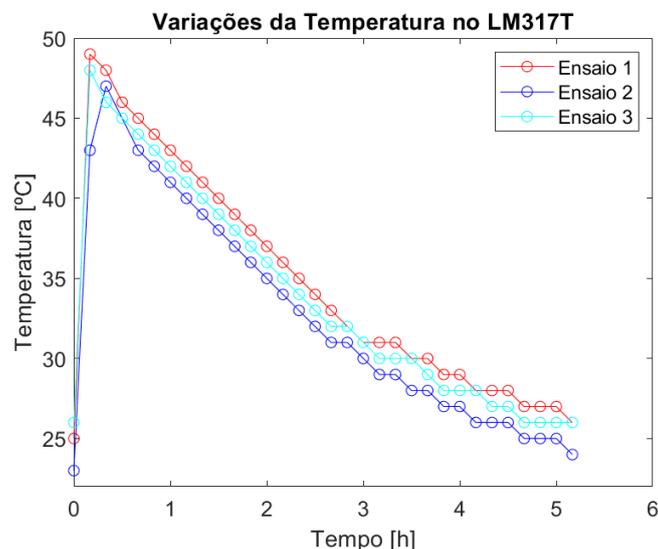


Figura 5.5: Variação da temperatura no regulador de tensão linear LM317T

Com a redução da corrente de carga conduzida pelo regulador ao longo do processo de balanceamento, a temperatura tendencialmente começa a baixar de forma proporcional, o que evidencia a influência direta da temperatura de operação do LM317T na performance da BMS, nomeadamente na condução de corrente. Praticar temperaturas de operação mais baixas é uma prioridade, pois caso se consiga efetuar esta manutenção, cria-se um mecanismo que protege o regulador de sofrer danos físicos, que comprometem diretamente as suas funcionalidades. Para além disso, repara-se que quanto maior for a quantidade de energia térmica produzida e emitida para o exterior, mais energia tem de ser entregue ao sistema, pois apenas uma parte desta energia é convertida em energia útil, ou seja, energia destinada somente ao processo de balanceamento. A restante energia, como já se mencionou é perdida sob a forma física de calor. Deste modo, para não limitar a capacidade de carregamento da BMS é necessário que a fonte de alimentação selecionada supra a potência máxima requerida pela BMS, mesmo que esta varia ao longo do tempo como se irá demonstrar. Assim, a potência total, que é a potência entregue inicialmente à BMS deriva da soma dos dois parâmetros energéticos apresentados, de modo que

$$P_{total} = P_{util} + P_{loss}. \quad (5.2)$$

A magnitude da potência entregue ao sistema apenas é influenciável pela intensidade das perdas energéticas, uma vez que num sistema ideal a potência entregue à bateria elétrica seria igual à potência inicialmente entregue na BMS. Qualquer sistema realista segundo estes moldes procura minimizar as perdas energéticas inerentes nos seus processos, de modo a maximizar a quantidade de energia útil utilizada nestes, tornando-os em sistemas mais eficientes e económicos. Deste modo, introduz-se o termo: rendimento energético, que é uma medida adimensional que indica a eficiência de um processo. Neste caso, o rendimento será expresso em relação à eficiência de conversão da energia à entrada do regulador de tensão, comparativamente com a energia entregue à bateria elétrica. No gráfico 5.6 é possível verificar a comparação entre a potência total e potência útil, onde a diferença entre ambas resulta nas perdas energéticas totais. Ambas as potências possuem um comportamento idêntico advindo dos vários ensaios realizados com uma tensão de entrada de 18VDC, de forma que existe sempre um pico de potência exatamente na altura onde há o transiente entre um balanceamento limitado por uma corrente constante, para um balanceamento limitado por uma tensão constante. Este pico de potência acontece, pois é o momento onde a diferença entre a tensão de entrada e a tensão de saída do regulador LM317T é máxima, que aliado de uma corrente previamente limitada, ou seja, no seu pico leva a requerer para o sistema uma maior quantidade de energia justificando este pico.

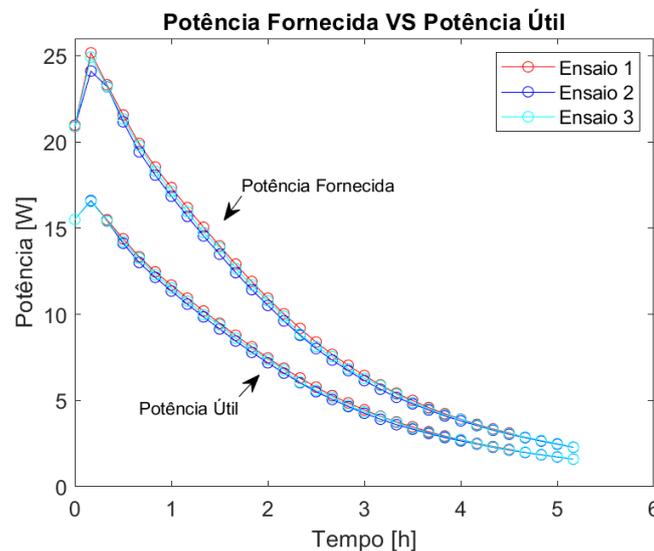


Figura 5.6: Comparação entre a potência entregue ao sistema em comparação com a potência útil

Por outro lado, também é nesta instância que a temperatura do LM317T é máxima, o que evidência o porquê do pico de potência entregue ao sistema ser significativamente superior ao pico de potência útil, pois como este é o período onde existem as maiores perdas energéticas, então o ganho de energia neste período é maioritariamente transformado em energia térmica, uma vez que o LM317T está a ser negativamente afetado pelas altas temperaturas praticadas, influenciando posteriormente na sua performance. Portanto, de forma análoga é claro a semelhança entre a variação da temperatura no gráfico 5.5 com o comportamento descrito por ambas as potências ao longo do tempo no gráfico 5.6, destacando esta relação de proporcionalidade, embora o ideal seria que área formada entre estas duas potências fosse o menor possível, visto era corresponder à magnitude das perdas energéticas ocorridas. Para evidenciar melhor esta influência da temperatura, produziu-se o gráfico 5.7 onde se destaca o rendimento energético do processo de conversão ao longo de um ciclo de carregamento para os três mesmos ensaios anteriormente mencionados.

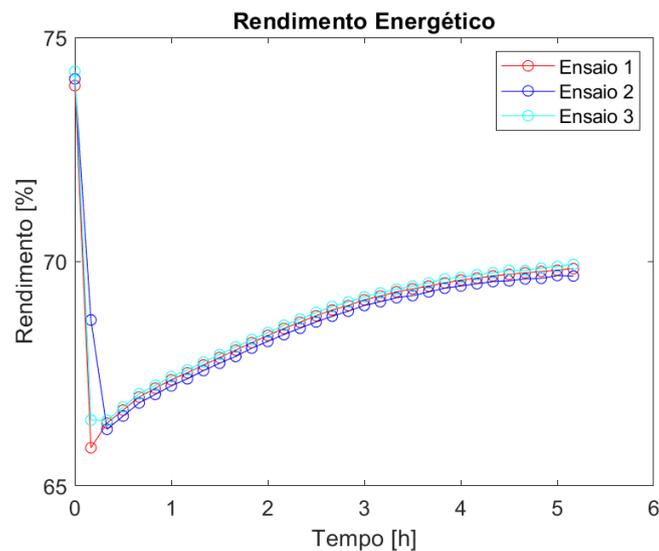


Figura 5.7: Rendimento energético da BMS durante o balanceamento das células

O pico de performance como seria expectável corresponde ao primeiro dado adquirido em todos os ensaios praticados, porque é a instância onde as temperaturas no regulador de tensão ainda se encontram baixas, não existindo ainda uma influência significativa que afete negativamente a performance do LM317T, e por isso alcança-se um rendimento energético médio de 74%. Porém, no período seguinte e novamente de forma unânime, verifica-se o impacto negativo das altas temperaturas praticas no regulador de tensão, uma vez que este mesmo período corresponde à instância onde se verifica o pior rendimento energético calculado de todo o processo de balanceamento, cerca de 66%, porque para além de coincidir com o pico de temperaturas praticas no regulador de tensão, também corresponde à fase onde existem as maiores perdas energéticas emitias sob a forma de calor. Como já se verificou e novamente se reforça com estas novas informações, a temperatura praticada pelo LM317T resultante do processo de conversão energética é o principal agente responsável por provocar estas variações no rendimento energético, de modo que o comportamento deste é inverso ao da temperatura, na medida que se a temperatura aumenta o rendimento diminui, porém se a temperatura diminui o rendimento aumenta. De todo o modo, consegue-se um processo com uma eficiência energética entre os 66% e os 74% nestas condições, o que comparado com outro tipo de reguladores de tensão, como os conversores buck utilizados em outras funções do AGV, conclui-se que este processo possui um rendimento inferior, mais ainda bastante razoável, como já se tinha previsto na seção 4.2.2 durante apresentação da arquitetura da BMS desenvolvida. No final destes ensaios, destaca-se o bom funcionamento conseguido suportado por todos estes dados adquiridos e processados. Note-se que a integração de um dissipador passivo no LM317T torna-se num elemento fundamental para promover um mecanismo que a partir de convecção natural permite realizar a transferência da energia térmica gerada, para o meio ambiente permitindo que ocorra uma melhor dissipação desta ao longo do processo de balanceamento.

5.2.4 Autonomia

Concluída a descrição dos ensaios relativos à BMS no ramo energético, falta ainda detalhar qual o consumo energético do AGV, de modo a determinar qual a sua autonomia. Esta característica é de extrema importância em qualquer sistema robótico, pois como são sistemas independentes, que não necessitam de qualquer fonte de alimentação fixa para operarem de forma ininterrupta, então torna-se necessário maximizar a eficiência energética de todo o sistema sem comprometer o rigor e a qualidade do processo, com o objetivo de otimizar o tempo útil de atividade. Conhecer a autonomia deste sistema é fulcral, pois é com base nestes ciclos de atividade contínuos que se define posteriormente o tempo de operação de cada AGV, evitando que estes fiquem inoperacionais no meio de um trajeto, devido à falta de energia nas suas baterias. Deste modo, quanto maior for a autonomia de um AGV maior é sua flexibilidade, podendo realizar trajetos de forma integral inúmeras vezes, visto existirem melhores rendimentos e otimizações que prolongam a capacidade da bateria elétrica. Neste caso, a bateria integrada no AGV desenvolvido possui uma capacidade energética nominal de 4.4 Ah, embora esta bateria seja constituída por células de lítio ICR18650 renovadas e como tal encontram-se com um número de ciclos intermédio, o que significa que a capacidade real da bateria é um pouco inferior à capacidade teórica desta, devido ao natural envelhecimento e número de ciclos de carregamento já efetuados.

Para efeitos demonstrativos, no gráfico 5.8 encontram-se discriminados cinco ensaios independentes adquiridos nos primeiros quinze minutos de operação do AGV em funções de seguimento de guias, de modo que a corrente média quando o AGV corre os algoritmos de visão artificial + serviço VNC + conexão Wi-Fi + sistema iluminação + alimentação dos motores elétricos, tudo isto embebido num sistema operativo, leva à corrente de consumo a alcançar uma média de 1.46 A. Tendo em consideração que o Raspberry Pi 4B é um computador de placa única e que existe um sistema de iluminação dedicado, tal com um controlo motriz alimentado pela mesma bateria elétrica, a magnitude da corrente consumida para promover estes serviços é proporcional, em comparação com a magnitude da corrente praticada pela BMS no ato de balanceamento, conseguindo-se equalizar ambos os processos energéticos. Note-se que, quanto mais serviços computacionais forem requeridos ao sistema, levando à inicialização de serviços multitarefas, mais energia o Raspberry Pi irá necessitar, verificando-se o mesmo comportamento quando os motores elétricos se encontram em situações em que necessitam de produzir um maior binário para superarem a inércia experimentada em dadas instâncias. Tendo em consideração estes fatores, e aproximando por arredondamento a corrente em média consumida a 1.5 A, então alcança-se uma autonomia de 2 horas e 55 minutos teoricamente. Na verdade, e como já foi mencionado, a performance das células decai ao longo do tempo e, portanto, a sua capacidade real para armazenar energia é cada vez inferior. Dos ensaios praticados e adquiridos registou-se autonomies práticas que sensivelmente rondam as 2h e 30 minutos, o que representa uma redução da capacidade nominal em 23%.

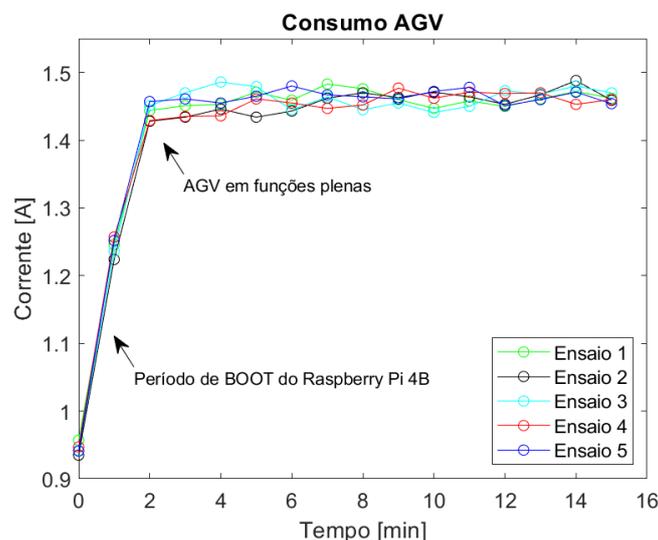


Figura 5.8: Consumo energético do AGV

5.3 Performance Computacional

A capacidade de computação do AGV é fundamental para proporcionar um grupo lato de capacidades que o tornam num elemento computacionalmente e funcionalmente bem conseguido. Lidar com algoritmos de visão artificial num computador de placa única é um desafio devido à complexidade e volume de dados processados em cada operação num elemento fisicamente limitado. A gestão de todos estes elementos, sem levar a falhas críticas no sistema ou a pobres performances é uma prioridade em qualquer fase de operação, de modo a preservar-se a cadeia de valores implementados: um sistema preciso e responsivo. Esta ideologia somente é conseguida, quando existe uma relação dinâmica entre todas as camadas desenvolvidas, especialmente ao nível do firmware, com um objetivo focalizado em manter a melhor performance possível sem afetar comandos básicos do sistema.

5.3.1 Tempo por Ciclo

Nestes termos, o tempo de processamento de qualquer processo do AGV é um indicador direto da sua performance computacional. Conseguir menores tempos de processamento sem afetar negativamente na performance geral do AGV é o principal objetivo quando se desenvolveu todo o ambiente que embebede estes procedimentos. O tempo entre a indicação de uma ação à execução desta é designado por período de latência, que em qualquer sistema realista é sempre uma característica inerente ao processo, à exceção de sistemas em tempo real, onde estes tempos são mínimos devido à própria arquitetura e funcionalidades a desempenhar. No caso do ambiente de desenvolvimento selecionado, é impossível conceber-se um processo onde não existem períodos de latências insignificantes, e por isso torna-se ainda mais importante diminuir estes períodos, de modo a otimizar todo o processo. No caso do AGV, como se trata de um sistema móvel independente, conseguir reduzir o tempo de processamento de qualquer ação é essencial para aumentar a frequência de amostragem, ou seja, possibilitar uma maior capacidade de aquisição num mesmo espaço temporal, ou ainda processar o mesmo número de situações num menor período temporal. Assim, destacado a importância deste parâmetro, torna-se imprescindível avaliar o tempo de execução do processo em si, nomeadamente o período entre a aquisição de duas imagens, onde neste espaço temporal são incluídos todos os aspetos relacionados a este algoritmo, desde a aquisição da imagem, aplicação de filtros, remoção de ruídos, aplicação de mecanismos de seleção, deteção e respetiva ação motriz em função do processamento gerido pelas funções de mais alto nível que tornam o AGV num sistema autónomo. Como forma de quantificar esta característica, recorreu-se à utilização da média aritmética (\bar{x}) que é um atributo estatístico definido como o quociente entre a soma de todos os valores de um conjunto pelo número total de elementos pertencentes a este conjunto. Por isso, se n simboliza o número total de elementos pertencentes a uma população finita e cada elemento x o respetivo valor, então a média aritmética é definida como

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} = \frac{1}{n} \sum_{i=1}^n x_i. \quad (5.3)$$

Para auxiliar nesta avaliação, na medida de conseguir-se obter um conjunto de informação mais rigoroso, calculou-se o desvio padrão (σ), de modo que para uma população finita de amostras o desvio padrão é uma medida de dispersão que indica o quão disperso os valores desta população estão face à média calculada. Portanto, consoante esta medida é possível verificar a exatidão dos dados adquiridos, de forma que na área da probabilidade o desvio padrão define-se como

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}. \quad (5.4)$$

Deste modo, para averiguar quais as velocidades de processamento do AGV procedeu-se à aquisição de 400 amostras de forma contínua, em relação ao tempo de processamento de cada ciclo computacional do AGV em plenas funções de seguimento, de modo a formar-se uma população de dados significativa com o objetivo de sustentar uma análise fundamentada. Todos estes dados encontram-se descritos no gráfico 5.9, de modo elucidativo.

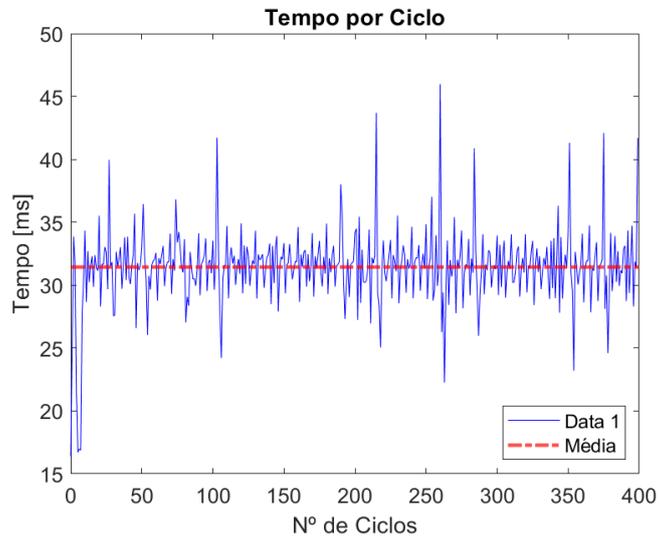


Figura 5.9: Tempo entre cada ciclo concluído

Como resultado, verifica-se um valor médio de 31.42 ms por cada ciclo de processamento, permitindo serem realizados por segundo 31 ciclos completos, o que significa que neste espaço temporal são realizados 31 ajustes de rotas por segundo. Este poder de computação é bastante positivo, pois verifica-se com estes dados o alto rendimento do algoritmo desenvolvido para um sistema de visão artificial embebido num computador de placa única. Aliás, se por um lado integrar este procedimento num sistema operativo é um fator fundamental para se ter acesso a vários serviços apenas disponíveis com a integração de um sistema operativo (Serviço VNC por exemplo) que beneficiam o baixo tempo de processamento do sistema com pouco esforço computacional associado, por outro lado, como o sistema operativo instalado (Raspbian) não é um OS de tempo real RTOS (*Real Time Operating System*), então como se verifica no próprio gráfico 5.9 existe uma maior dispersão de tempos de processamento, refletindo-se diretamente no valor do desvio padrão em 3.03 ms, cerca de 10% do valor da média. Este valor não é baixo, mas é próximo de um valor que garante a exatidão do processo, levando a concluir que o tempo de processamento de cada ciclo é estável, ainda que se verifiquem com mais frequência tempos mais afastados da média estabelecida, o que significa que a média calculada não é um valor extremamente exato devido à dispersão de resultados verificada.

Esta dispersão é explicada, pois o sistema operativo integrado quando em funcionamento realiza um serviço de multitarefas, executado em simultâneo várias funções diferentes no mesmo espaço temporal. Ora, nesta situação, como apenas existe um processador, então praticar múltiplos serviços sendo os principais o serviço VNC, que necessita obrigatoriamente de uma conexão WiFi e o próprio algoritmo de visão artificial desenvolvido, leva a criar diferentes condições de processamento que influenciam diretamente no tempo de processamento de cada ciclo. Isto porque, os dados provenientes dos vários serviços são armazenados temporariamente num buffer, visto o CPU ter uma alta frequência de operação, o que leva a ser necessário realizar este armazenamento. Como o Raspberry Pi 4B não possui um RTC como se referiu na seção 4.3.1, este não tem a capacidade básica de determinar o tempo entre cada iteração e, portanto, não existe forma de promover um mecanismo exato na receção e envio de dados do buffer para o CPU, deixando-se à mercê de cada instância conforme o fluxo de dados requeridos, levando a uma variação da taxa de dados recebidos/enviados. Mesmo como a diminuição de serviços terceiros a correrem em segundo plano, devido às características deste sistema operativo, não se consegue obter ciclos exatos e por isso a dispersão apresentada.

5.3.2 Qualidade do Processamento de Imagem

Na secção 4.3.5 defendeu-se a utilização de imagens num formato HSV para se analisar e aplicar os respectivos algoritmos de processamento de imagem, ao invés de um modelo RGB, devido aos benefícios que o modelo HSV traz face à utilização de imagens num formato RGB. Como prova dos benefícios desta conversão, analisa-se nesta secção o espaço de cor de uma imagem capturada pelo AGV durante o seu funcionamento. Procurou-se utilizar uma imagem que incluísse não só ambas as máscaras de cores definidas (máscara preta e máscara azul), de forma a tornar esta avaliação mais genérica, mas também com diferentes condições de iluminação do meio externo, para averiguar qual o grau de influência deste fator. Assim, encontra-se na figura 5.10 uma imagem modelo que inclui um elemento a ser detetado pela máscara preta (guia) mais um elemento a ser interpretado pela máscara azul (sinal de mudança de direção), sob um piso que ao nível do algoritmo desenvolvido não revela nenhuma informação relevante a não ser contrastar com os restantes elementos.

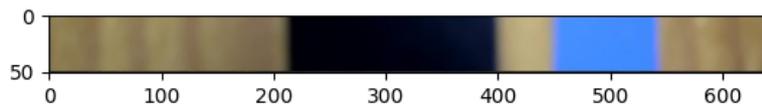


Figura 5.10: Imagem em análise capturada pelo AGV

O espaço de cores segundo um modelo RGB como é destacado na figura 5.11 apresenta um espectro cromático bastante vasto e coletivo, não existindo divisões claras entre as diferentes tonalidades. De facto, existe um único gradiente contínuo de tonalidades entre a cor azul até à cor preta, de modo que esta continuidade persiste mesmo sob a influência de diferentes iluminâncias por parte do meio externo, ainda com diferentes intensidades como é visível pelas escalas das cores primárias (vermelho, verde e azul) de ambos os espaços cromáticos RGB formados. Portanto, confirma-se a influência significativa da iluminância externa nas cores formadas, uma vez que como o sistema RGB é um modelo aditivo e como não é possível separar a iluminância da cor capturada, então a sua influência é bastante expressiva levando à formação desta paleta de cores altamente influenciável por este fator. Para além disso, a falta de diferenciação entre os vários espaços de cores é problemática, pois como existe um único elemento que agrupa todas as tonalidades capturadas, então torna-se difícil definir onde começa ou acaba o espaço de cor de um determinado elemento. Deste modo, esta característica promove a aquisição de maiores quantidades de ruído, quando se define espectros cromáticos mais abrangentes, correndo-se o risco de se incluir outras tonalidades sem relevância, manifestando-se neste caso sob a forma de ruído, ou promove-se por outro lado a formação de espectros cromáticos tão restritos que quando existem mínimas alterações na iluminação, estas alterações resultam na modificação da intensidade do espaço de cor formado, podendo levar à não identificação de uma determinada cor.

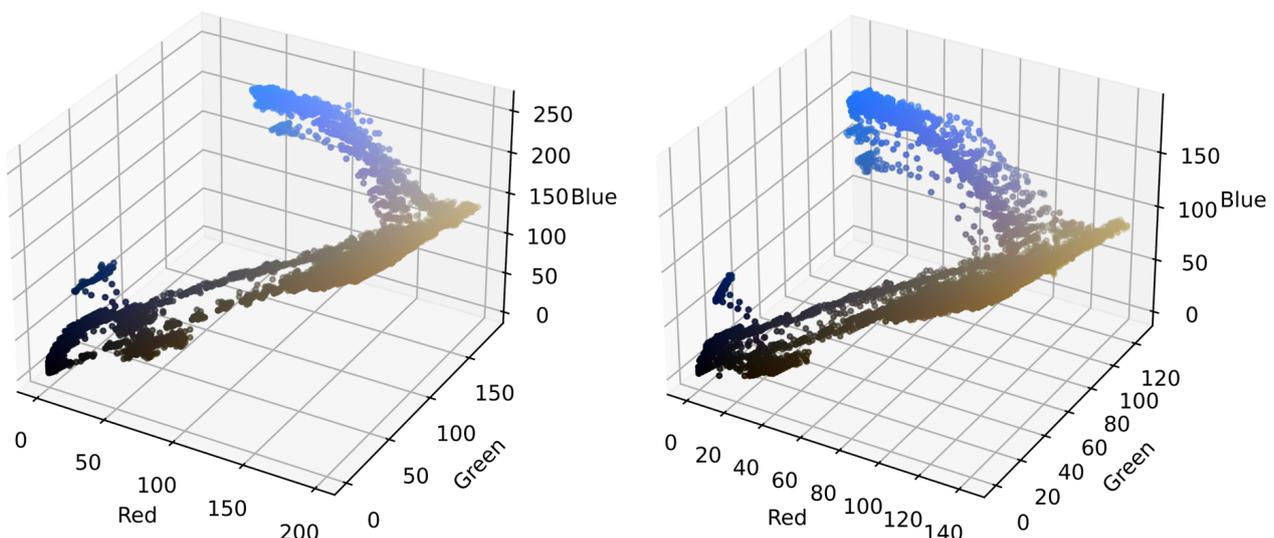


Figura 5.11: Distribuição de cores captados num modelo RGB com diferentes iluminações

No caso do espaço de cores segundo um modelo HSV como é destacado na figura 5.12, este apresenta um espectro cromático bastante segmentado e bem definido. Não existe um único gradiente contínuo como no modelo RGB, mas sim uma clara divisão com limites bem definidos em três zonas descontínuas, onde cada zona corresponde de modo intuitivo ao piso, sinal de mudança de direção e guia, respetivamente analisando-se da esquerda para direita. À semelhança do espaço de cor RGB, também existe aqui influência da iluminação externa, embora esta seja menos significativa não afetando a diferenciação existente entre os vários grupos de tonalidades definidos. Aliás, a iluminância do meio externo praticamente só tem influência no atributo *value*, justamente porque esta é a qualidade responsável por medir o efeito da iluminância. Repare-se que num espaço HSV as cores são mais localizadas e visualmente separáveis, uma vez que os níveis de saturação e de tonalidade se encontram maioritariamente distribuídas ao longo do parâmetro valor. Por isso, como se verificam apenas alterações significativas neste parâmetro, então criar espectros cromáticos para identificar uma cor é bastante mais facilitado do que no modelo anterior. Para além disso, neste modelo torna-se mais intuitivo perceber onde começa e acaba um espaço de cor, não se correndo o risco de incluir outras tonalidades que não se queiram adquirir, diminuindo-se passivamente a aquisição de ruído por parte de tonalidades não desejadas.

De todo o modo, a diferença entre um modelo RGB e HSV na formação de espaços de cores para o processamento de imagem é notável sendo claro com estes resultados o porquê da utilização de um modelo HSV ou invés de um modelo RGB. Em qualquer processo de aquisição de imagens, é importante que as características de interesse sejam identificadas nas melhores condições possíveis, o que numa numa imagem digital, as componentes R, G e B do modelo RGB estão diretamente correlacionados com a quantidade de luz que atinge os objetos de interesse e, portanto, devido a esta forte influência dificulta-se na discriminação dos objetos a identificar, nomeadamente na identificação das suas cores. Deste modo, enquanto num modelo RGB existem espectros cromáticos dificilmente divididos, onde a introdução de ruído pela definição de espectros cromáticos mais abrangentes é mais suscetível a acontecer do que num modelo HSV, tornando-o num modelo mais volátil e significativamente dependente das condições de iluminação do meio externo, num modelo HSV é possível isolar a cor capturada separando-a da sua saturação e efeitos de iluminância, tornando-a num modelo ideal para aplicações reais onde as condições de iluminação mesmo com sistemas de iluminação fixos variam devido a fatores externos. Portanto, para imagens digitais na área do processamento de imagem é preferível analisar imagens com base na sua saturação, coloração e iluminância.

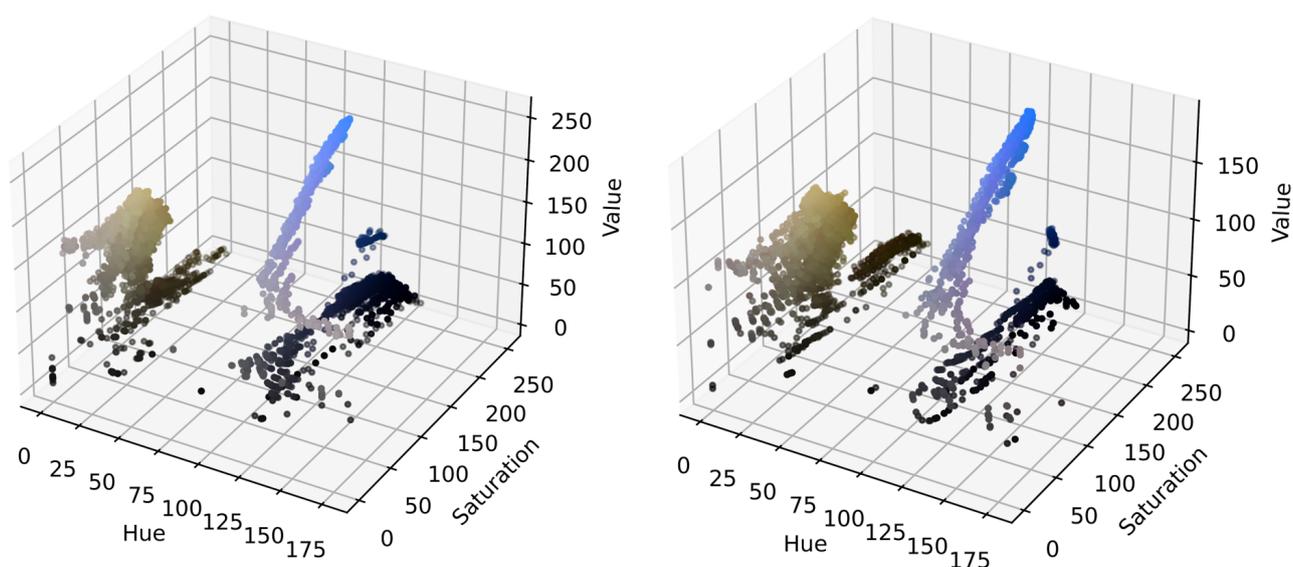


Figura 5.12: Distribuição de cores captados num modelo HSV com diferentes iluminações

5.4 Performance Motriz

A capacidade de deslocação do AGV é fundamental para garantir o bom funcionamento deste sistema como um veículo autónomo terrestre. Como foi mencionado na seção 2.5 e desenvolvido em termos técnicos na seção 4.1.3, o AGV apresenta uma configuração motora inspirada num sistema de deslocação diferencial com o propósito de oferecer uma maior flexibilidade de deslocação ao longo dos trajetos a realizar. Esta qualidade quando bem conseguida e mantida torna o AGV num sistema mais fluído e confiável, uma vez que fora todo o algoritmo computacional desenvolvido e hardware implementando, é justamente o modo como este se desloca que impacta diretamente com os intervenientes humanos presentes ao longo do percurso operacional, ainda que o sistema desenvolvido seja um modelo de prova. Fora este fator, manter rotas precisas e exatas é prioritário também por questões de segurança, uma vez que dada a tipologia deste sistema, este é normalmente integrado em ambientes comuns com outras tecnologias em circulação, fora o movimento pedestre no ser humano que não é programado, e por isso a necessidade de se efetuar uma manutenção de trajetos de forma rigorosa, tornando-se numa característica prioritária para preservar a segurança impingida no local de operação, visto o tipo de funcionalidades que os AGV's realizam. De todo o modo, o principal objetivo do sistema de deslocação desenvolvido é promover uma condução precisa, de modo a garantir a exatidão das rotas pré-estabelecidas pelo Utilizador. Assim, e justificada a importância deste subsistema do AGV, a avaliação da performance motriz torna-se no teste mais importante a realizar, e aquele onde podem surgir os maiores defeitos devido à quantidade de variáveis relacionadas a este.

Como forma de avaliar a exatidão do AGV face às rotas praticadas é necessário avaliar a sua capacidade de seguimento e, portanto, como varia o centroide da guia segundo a direção x , neste caso I_x em relação ao valor do centro de imagem do ROI (C_x), que é o valor teórico para um seguimento correto, avaliando-se a cada ciclo processado a diferença absoluta entre estes dois parâmetros. Quanto menor esta diferença, então melhor é seguimento por parte do AGV sendo esta uma medida direta de exatidão, uma vez que num cenário ideal o centroide da guia seria igual ao centro de imagem do ROI, ou seja, $I_x = C_x = 320$. Porém, devido a todos os fatores externos e variações naturais dos processos do AGV, existem sempre variações inerentes ao processo que originam estas diferenças. Num processo estável estas diferenças não impendem a estabilidade do sistema ao longo do tempo, pois a capacidade de regulação deste é maior do que as diferenças resultantes em cada instância, que de um modo proporcional, quanto mais expressivas estas diferenças forem, então menor é capacidade de controlo do sistema, caminhando-se no pior caso para a sua instabilidade, ou seja, o descontrolo total do AGV e falha de todas as funções de seguimento. Assim, para monitorizar a qualidade do seguimento do AGV, incluir-se-á os atributos estatísticos presentes na seção 5.3.1, mais o cálculo de erros com o objetivo de identificar qual a sua origem, a sua magnitude/influência e respetiva análise, de modo a minimizar a sua presença ou intesidade em eventos futuros através de uma manutenção preventiva. Deste modo, calculou-se o erro absoluto (E_a), sendo este o resultado absoluto entre a diferença entre o valor teórico (C_x) e o valor prático do centroide da guia (I_x), de modo que

$$E_a = |C_x - I_x|. \quad (5.5)$$

O erro absoluto apenas indica uma diferença quantitativa, o que o torna insensível não indicando de forma padronizada a seriedade do erro calculado. Portanto, independentemente na escala utilizada, calculou-se também o erro relativo (ε), sendo este o coeficiente entre o erro absoluto e o valor teórico (C_x), que multiplicado por 100 indica em termos percentuais, o quão o resultado obtido difere do valor real. A sua obtenção advém da seguinte formulação para este caso em específico

$$\varepsilon = \frac{E_a}{|C_x|} \times 100\%. \quad (5.6)$$

Assim, ao capturar-se qual a variação do centroide da guia (I_x) ao longo de 250 iterações em três ensaios distintos de forma interrupta, constrói-se o gráfico 5.13 com o propósito de avaliar qual a evolução temporal deste parâmetro ao longo dos trajetos praticados. Para auxiliar nesta análise, foi também simulado os limites físicos da guia, marcados como o Limite Inferior (230 pixels) e o Limite Superior (410 pixels), sendo que como fisicamente a guia possui 17.5 mm de largura, então torna-se necessário converter para a escala de pixels, de modo a formar as respectivas assíntotas horizontais assinaladas. Repara-se que a câmara digital se encontra encastrada na estrutura frontal da base do AGV, e por isso é possível conservar uma distância constante durante toda a fase de seguimentos, validando em qualquer altura a relação mm-px estabelecida, de modo que 1 px corresponde a $97.22 \times 10^{-3} \pm 0.05$ mm.

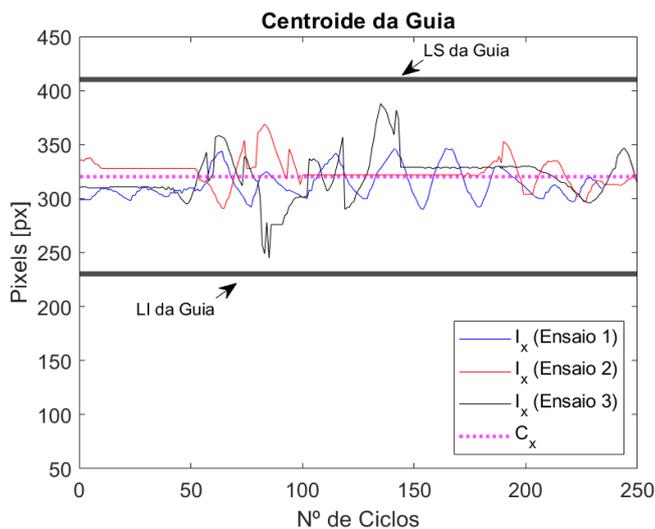


Figura 5.13: Variação do centroide da guia

Pela análise dos resultados obtidos e tendo em consideração o resumo das propriedades estatísticas presentes na tabela 5.1, em todos os ensaios realizados o AGV nunca transgrediu os limites físicos da largura da guia, ou seja, o centroide calculado nunca superou esses limites virtuais o que é um indicador da manutenção da estabilidade do processo, uma vez que ao longo das amostras adquiridas o comportamento do centroide tende a manter-se contido neste intervalo sem sinais de instabilidade. Somente existe instabilidade quando existe o aumento gradual e constante do centroide face ao centro de imagem do ROI (C_x) ao longo das várias iterações. Em termos práticos, esta instabilidade traduz-me no descontrolo gradual do AGV até à completa perda visual da guia incapacitando-o de a processar, e por isso este fica imóvel e inoperacional sendo necessário intervenção humana para o direcionar novamente. Nestes ensaios, este comportamento não se verificou e, portanto, consegue-se praticar trajetos de uma forma exata como o projetado. O seguimento não é perfeito, como seria expectável e é observável em testes práticos, existindo variações naturais presentes em cada ensaio que se traduzem por pequenas oscilações presenciadas ao longo dos testes práticos realizados. O comportamento mais característico deste processo é um movimento de subida e descida cíclico, que indica as sucessivas mudanças de direção repetitivas por parte dos comportamentos motrizes, nomeadamente quando de pretende que o AGV vire para a direita ou para a esquerda em ações de ajuste de rota de forma cíclica. Já as fases onde existem poucas flutuações resultam em linhas horizontais que correspondem aos períodos onde o AGV segue em frente, ainda que o centroide varie significativamente pouco.

Parâmetros	Ensaio 1 [px]	Ensaio 2 [px]	Ensaio 3 [px]
Média Aritmética (\bar{x})	318	323	317
Desvio Padrão (σ)	14	13	18

Tabela 5.1: Resumo das propriedades estatísticas do gráfico 5.13

Para a velocidade padrão definida na seção 4.1.3 e tendo em consideração as variações para os três ensaios realizados, a média aritmética é quase similar ao valor teórico o que revela a exatidão do processo, porém, neste caso a característica estatística mais importante é o desvio padrão, pois é o parâmetro responsável por indicar a dispersão das amostras adquiridas e, portanto, por demonstrar como o AGV oscila ao longo do seguimento dos trajetos. Repare-se que nos três ensaios efetuados este valor tende a ser baixo, o que torna possível concluir que o processo é realizado com precisão, conseguindo-se promover um seguimento com rigor. Naturalmente, dadas estas características o erro relativo médio ($\bar{\varepsilon}$) de cada ensaio também segue esta tendência e entre os três ensaios este varia em média entre os 3% e 5%. Contudo, é um valor médio e por vezes existem situações pontuais onde este erro é mais significativo devido a irregularidades ou no piso, ou devido a outro fator externo que o levam a gerar maiores desvios, sendo visivelmente claro quando acontecem estas situações em testes práticos. No gráfico 5.14 é possível constatar estas variações do erro relativo (ε) ao longo das várias iterações.

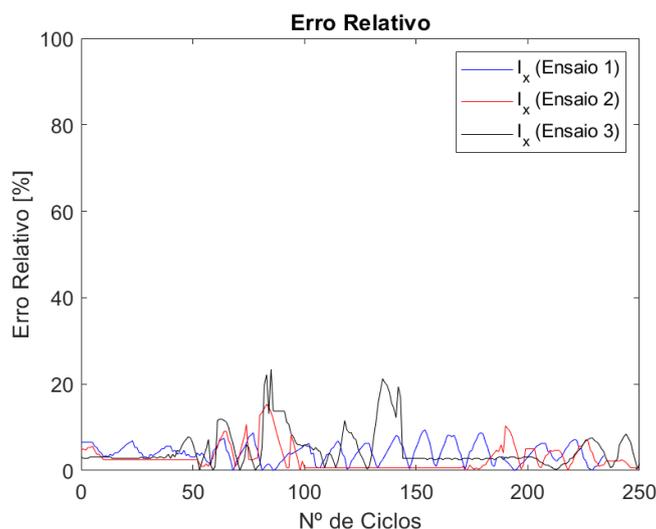


Figura 5.14: Variação do erro relativo em cada amostra adquirida

Estes comportamentos apenas são válidos quando a velocidade escolhida pelo Utilizador para os motores elétricos é igual à velocidade padrão definida, pois caso exista o aumento ou diminuição desta, os parâmetros aferidos alteram-se significativamente. No caso de se aumentar a velocidade pelo aumento da tensão média entregue os motores elétricos pela modificação do sinal PWM, então capacita-se o AGV a percorrer uma maior distância num mesmo período. Porém, se por um lado bonifica-se o sistema com este atributo, por outro lado prejudica-se nas suas funções de seguimento pela perda de forma proporcional da precisão com que este realiza o seguimento dos trajetos, pois como o AGV apresenta um tempo de processamento médio como se aferiu na seção 5.3.1 em torno de 31.42 ms e apenas é dependente de processos computacionais, ou seja, características mecânicas/elétricas não tem influência neste tempo, então as variações naturais que ocorrem no processo aliado a um maior espaço percorrido no mesmo espaço temporal, leva a promover maiores desvios, e por isso a maiores oscilações no movimento físico do AGV. Este comportamento promove maiores diferenças entre o centroide da guia (I_x) e o centro de imagem do ROI (C_x) diminuindo a precisão do seguimento, comprometendo as características que se pretendem integrar neste processo. Para demonstrar estes efeitos foram realizados novamente três ensaios sob os mesmos moldes, mas com uma velocidade duas vezes superior à velocidade padrão e, portanto, pela análise do gráfico 5.15 é visível como as oscilações do AGV tornam-se mais expressivas e impactantes no comportamento motriz do AGV, uma vez que apenas se observa um comportamento cíclico de mudanças de direção da esquerda e para a direita sucessivamente, não existindo em nenhum dos ensaios realizados qualquer troço onde o AGV circulou em frente, pois como a velocidade está num patamar relativamente elevado para este sistema, mesmo que o sistema precise de realizar um ajuste de rota singelo, este ajuste devido à maior velocidade entregue pelos motores leva à transformação de um pequeno ajuste teórico numa oscilação extrema ultrapassando na grande maioria das situações a largura da guia.

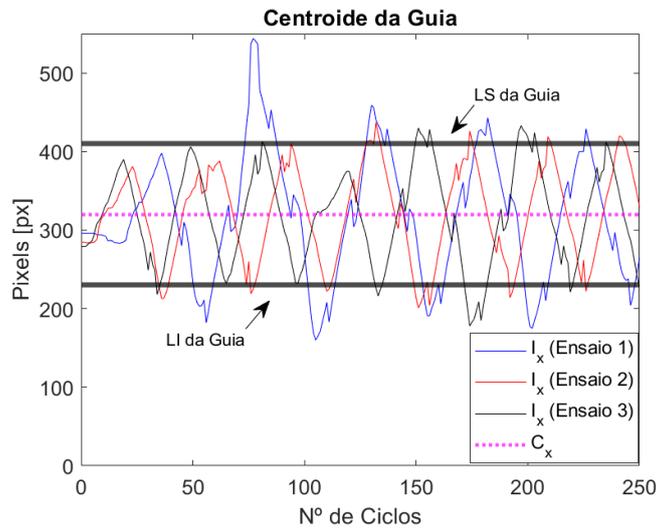


Figura 5.15: Variação do centroide da guia

Com base nos dados estatísticos da tabela 5.2, verifica-se uma maior dispersão do valor do centroide da guia fundamentando-se nos valores do desvio padrão significativamente mais elevados do que no ensaio anterior, cerca 20% a 25% da média de cada ensaio, embora a média aritmética dos três ensaios ser mais exata do que a média dos resultados anteriores. Num sistema ideal, o AGV seria um sistema preciso e exato, mas dado todos os fatores que intervêm nele, prefere-se que seja um sistema preciso, mas não exato, tendo um desvio padrão menos significativo ainda que a média seja menos exata (ensaio anterior), do que ser um sistema exato e não preciso (ensaio atual), onde as oscilações são tão extremas que o levam ao seu mau funcionamento com uma condução caracterizada por oscilações extremas com tendência a se descontrolar.

Parâmetros	Ensaio 1 [px]	Ensaio 2 [px]	Ensaio 3 [px]
Média Aritmética (\bar{x})	319	318	321
Desvio Padrão (σ)	80	56	67

Tabela 5.2: Resumo das propriedades estatísticas do gráfico 5.15

Como consequência, no gráfico 5.16 é visível o impacto do aumento da velocidade para valores não recomendados para o sistema desenvolvido, com erros que variam em média entre os 17% e 20%, comparativamente com 3% e 5% verificados no ensaio anterior.

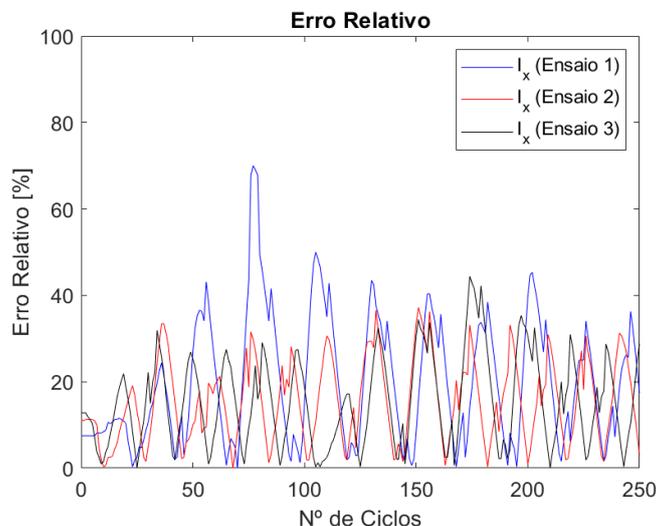


Figura 5.16: Variação do erro relativo em cada amostra adquirida

Capítulo 6

Conclusão e desenvolvimentos futuros

O principal objetivo do presente trabalho de projeto foi o desenvolvimento de raiz de um veículo autoguiado baseado num Raspberry Pi num formato *Proof of Concept*, que utiliza como único mecanismo de aquisição uma câmara digital, com o propósito de proporcionar um guiamento autónomo e inteligente, apenas com base no processamento de imagem efetuado a partir de ferramentas de visão artificial, com o objetivo de promover o desenvolvimento futuro de AGV's segundo as tecnologias aqui instauradas.

Para suportar esta finalidade, evitou-se utilizar qualquer dependência de tecnologias ou serviços *Closed-Source* em todas as etapas de desenvolvimento, visto a impossibilidade legal de realizar alterações/modificações, mas também devido a um conjunto de dependências que flutuam consoante a entidade prestadora do serviço. Por isso, para não se estar dependente de serviços terceiros, procurou-se em todas as etapas de desenvolvimento a utilização de serviços e estruturas *Open-Source*, de modo a assegurar toda uma capacidade de customização e compatibilidade entre todas as entidades utilizadas, sem existirem restrições em relação a contratos ou limitações técnicas empregues nos produtos utilizados. Quando não encontrada uma solução nesta área, ou quando soluções deste género não forneceram as condições necessárias ou os atributos para atender às necessidades técnicas do AGV, então efetuou-se o desenvolvimento original de novas estruturas, de modo a suprimir as necessidades que o sistema assim o exigia. Dentro destes elementos formados encontra-se toda a estrutura mecânica do AGV, produzida com o objetivo de alocar todo o hardware (mecânico + elétrico), segundo um design estruturalmente acessível, modular e principalmente dimensionado para suplementar todas as funcionalidades do sistema desenvolvido. Esta estrutura foi totalmente modelada através de um software CAD auxiliando na criação e simulação de todos os componentes mecânicos integrantes do AGV, respeitando o comprimento de normas ISO que visam melhorar a qualidade dos componentes produzidos, tal como garantir a sua compatibilidade com restantes elementos *standards* já modelados. Cada parte funcional modelada foi produzida através de um processo de deposição de material fundido (FDM), utilizando-se PLA como material base. Este processo de fabrico em tendências atuais crescentes revela-se como uma opção económica e versátil face a outros métodos convencionais, conseguido impingir nos elementos produzidos propriedades físicas de interesse advindas do material base, tornando-os resistentes aos esforços mecânicos resultantes de um sistema dinâmico como o AGV. Porém, como se verificou ao longo do processo de produção, para garantir certas tolerâncias dimensionais ou determinados formatos geométricos, o tempo de fabrico pode alcançar várias horas de produção.

Esta estrutura mecânica é completada pela integração de um sistema de deslocação terrestre num formato diferencial, onde o poder de tração é garantido por dois motores de íman permanente na parte frontal do AGV, com um redutor com uma relação de 1:90 com o objetivo de aumentar o binário de saída. Cada motor, respetivamente, está associado a uma roda com um rasto em borracha sintética com o objetivo de proporcionar uma melhor adesão a pisos *indoor* com pouco relevo. Para garantir a estabilidade do AGV, na traseira encontra-se uma roda giratória, que perfaz uma distribuição motora do tipo triciclo, tornando o AGV fisicamente mais manobrável, mas menos preciso a efetuar ações motrizes repetitivas.

Agregou-se a este hardware mecânico todo um sistema de alimentação, através de uma distribuição energética assegurada por uma bateria elétrica de 48.84 Wh de energia, formada por células de lítio ICR18650 com uma configuração elétrica 3S2P, fornecendo até 2h e 30 minutos de autonomia, face ao consumo médio de 1.46 A em funções plenas de seguimento. Dada a natureza química das células de lítio e os perigos associados a carregamentos mal efetuados, foi necessário desenvolver uma BMS com o objetivo de proporcionar balanceamentos seguros, garantindo a segurança interna do próprio sistema. Para além da função de balanceamento, a BMS originalmente criada também é dotada de um regulador de tensão linear, o que promove tensões à saída desta constantes independentemente da tensão de entrada, tornando-a num sistema altamente compatível com inúmeros transformadores de corrente contínua, quando se pretende efetuar o seu carregamento. Esta BMS é um sistema puramente elétrico com um sistema de supervisão visualmente apelativo, embora não seja possível monitorizar diretamente de forma precisa neste formato ou segundo algum protocolo de comunicação, o estado do balanceamento das células à exceção do estágio final de balanceamento. De forma análoga ao desenvolvimento da estrutura mecânica do AGV, desenvolveu-se uma PCB com o circuito elétrico que forma a BMS, sendo todo este processo modelado num software CAM, onde a construção física desta placa foi assegurada por um serviço externo, de modo a se assegurar a reprodutibilidade técnica deste mecanismo desenvolvido, tal como integrar um elemento com ligações elétricas confiáveis devido às suas ligações mecânicas cimentadas por um processo de brasagem. Tais funcionalidades embebidas nesta PCB convergem para um processo de balanceamento contínuo com uma duração entre 5 e 6 horas, estando dependente do nível de descarga das células, onde a corrente máxima permitida neste processo varia até 1.4 A, a fim de proteger a integridade física dos componentes integrantes da BMS, tal como respeitar a corrente recomendada para realizar o balanceamento destas células. Todo este processo ocorre com um rendimento energético médio em torno de 68%, pois como se utiliza um regulador de tensão linear, a diferença entre a tensão de entrada e a tensão de saída é eliminada através da formação de energia térmica, emitida para o ambiente externo através de um dissipador de calor passivo sem qualquer aproveitamento energético. De todo o modo, todas as restantes estruturas elétricas do AGV dependentes da energia contida na bateria elétrica possuem rendimentos energéticos acima de 92%, no pior dos casos, otimizando-se de modo bem-sucedido a autonomia da bateria através da diminuição das perdas energéticas inerentes aos processos de conversão energética.

Todo este hardware é controlado pelo firmware desenvolvido, tendo-se integrado um Raspberry Pi 4B que é um computador de placa única *Open-Source* com um sistema operativo Raspberry Pi OS alocado num cartão microSD externo, responsável por processar todos os processos computacionais do AGV, com acesso a inúmeras funcionalidades relevantes, como o serviço VNC permitindo o controlo e acesso a este, através de qualquer ponto de acesso via Wi-Fi na mesma rede LAN, desde que se conheça as credenciais de acesso. Estas entidades computacionais são exclusivamente a senha de acesso e o IP do AGV, tendo-se optado por se utilizar um IP estático ao invés de um IP dinâmico, visto a volatilidade que endereços dinâmicos possuem, garantindo-se nestas condições um endereço fixo e apenas mutável quando o dono do serviço (AGV) o desejar. Este serviço de controlo remoto é unidirecional, ou seja, estabelece-se apenas numa direção onde o Master/Administrador é o Utilizador, enquanto o AGV é Slave, embora seja um sistema totalmente autónomo, onde por defeito esta ligação utiliza um protocolo de comunicação criptograficamente vulnerável aos dias atuais e, portanto, visto esta vulnerabilidade utilizou-se antes o protocolo SSH, de modo a integrar uma nova camada de proteção, aumentando-se a segurança cibernética entre a ligação Utilizador-AGV. Todo este processo de comunicação culmina na execução via remota do algoritmo computacional desenvolvido em Python juntamente com módulos do OpenCV. Ao juntar-se estas duas entidades, desenvolveu-se um software composto de uma forma modular e expositiva, partindo desde a aquisição de imagens pela câmara digital integrada até à utilização de ferramentas de visão artificial, a fim de se garantir a formação de um sistema robótico independente de ações do ser humano, tendo-se suportado estes requisitos pela relação bidirecional entre todos os *scripts* desenvolvidos, de modo a conseguir-se criar um processo organizado, segundo um padrão que conserva as várias hierarquias introduzidas, como forma de facilitar futuros *updates* e correção de erros.

Com base nesta relação, o principal mecanismo que dota o AGV de uma condução autônoma é o cálculo do centroide da guia, onde em cada iteração processada leva a existir uma manifestação física, que se exterioriza por constantes ajustes de rotas realizados pelo sistema motriz integrado, de modo a garantir a precisão do seguimento. Todo este processo ocorre com um período de amostragem médio de 31.42 ms, embora devido à integração de um sistema operativo este tempo possui variações significativas relacionadas com o serviço multitarefas que o Raspberry Pi 4B executa em segundo plano. Fora todo o processamento de imagem, desde a captura de imagem, aplicação de filtros e modelos para melhorar a qualidade das imagens e facilitar processos de segmentação e identificação, o grande diferencial neste sistema face a outros AGV's baseados num guiamento por processamento de imagem pela análise de um sistema de guias no piso, é a capacidade de este efetuar mudanças de direção nos respetivos locais, tendo a opção de seguir em frente (manter a rota anterior) ou mudar de direção (seguir uma nova rota). Todo o sistema de decisão associado a esta tomada de decisão é autónomo, porém é baseado na definição prévia das rotas planeadas pelo Utilizador, tendo este a capacidade de modular/construir trajetos personalizados, a fim de suprimir as exigências do processo, através da elaboração de sequências de código baseados num conjunto de regulamentos definidos, onde cada código definido corresponde a um local de mudança de direção. Portanto, cada trajeto é independente de qualquer outro trajeto estabelecido, sendo que o tamanho do código definido equivale ao número de sinais de mudança de direção processados, não sendo necessário definir todos os sinais que formam o percurso operacional, tornando-o numa metodologia intuitiva e lógica.

Até este ponto, o AGV desenvolvido apresenta várias características bem conseguidas e exploradas, porém, em termos práticos, o AGV apresenta dificuldades em efetuar mudanças de direção com rigor, não por causa do algoritmo desenvolvido, mas devido à escolha dos motores elétricos utilizados. Foi efetuado um correto dimensionamento destes, contudo utiliza-se estes motores com uma velocidade modelada pelo sinal PWM em cerca de 30% da sua capacidade nominal, porque caso se utilize uma velocidade superior a esta, o AGV tende a desestabilizar, pois para um mesmo tempo de processamento computacional, como o AGV percorre uma maior distância, então leva-o a efetuar os trajetos com uma menor precisão. Para solucionar este comportamento, efetua-se a regulação do sinal PWM, porém, o problema desta regulação é que ao diminuir-se a velocidade de saída para um nível adequado para este sistema, também se diminui o binário de saída devido à relação mecânica do redutor à saída dos motores elétricos, introduzindo-se implicitamente um problema, pois com esta diminuição o AGV perde capacidade mecânica de gerar um momento que supere a sua inércia, causando paragens subitidas devido à incapacidade de formar um binário que supere a inércia experimentada nessas instâncias. A solução para este problema, mas não implementada no AGV no estado atual assenta no aumento da relação de redução do redutor, levando à diminuição da velocidade de saída e inversamente ao aumento do binário formado, resultando num sistema com uma menor velocidade à saída, mas com um maior binário mecânico, o que promove um maior rigor no seguimento dos trajetos e a eliminação de situações onde existe a paragem imprevista do AGV.

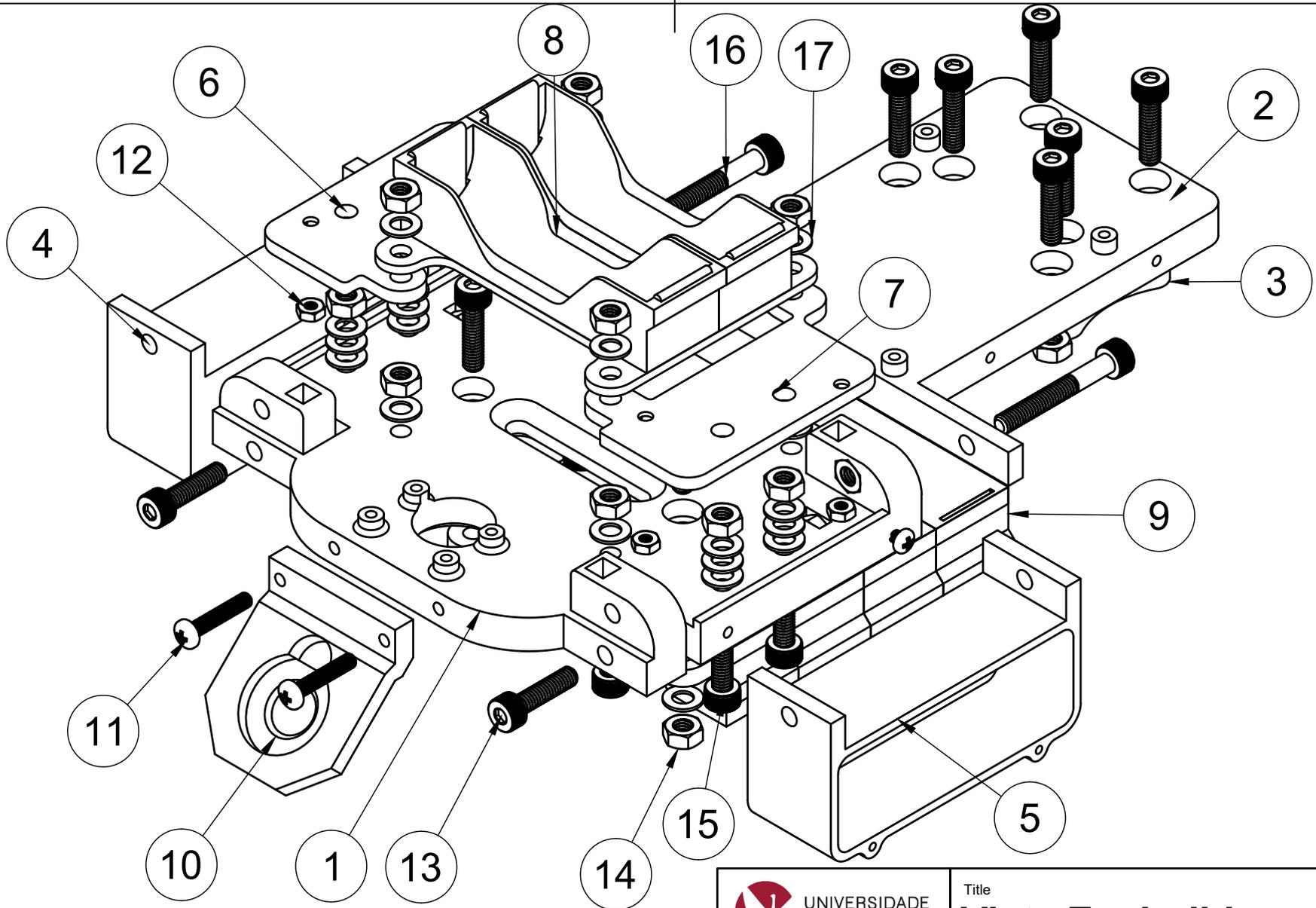
Futuramente, como melhorias a agregar a este sistema, seria útil do ponto de vista analítico integrar modelos do tipo *Deep Learning* fundamentados em rede neuronais, para promover um processamento de imagem mais preciso e principalmente com uma capacidade evolutiva conforme o número de situações analisadas, onde com a experiência acumulada destas situações promover-se-ia a formação de uma mecanismo auto-didata que aprenderia a distinguir variações dentro do próprio processo, o que é extremamente importante para se conseguir efetuar identificações mais precisas. Em termos de navegação, o sistema de deslocação diferencial seria substituído por um modelo chefiado por um controlador PID (*Proportional-Integral-Derivative controller*), responsável por criar um sistema de *feedback* contínuo que regula de forma precisa e responsiva, a posição do AGV face à guia a seguir, através das suas três componentes, conferindo um controlo motriz otimizado quando corretamente parametrizado. Visualmente, para promover uma melhor experiência de definição de trajetos, seria útil a construção de uma interface gráfica, onde ao invés de efetuar-se a definição de trajetos com base em códigos computacionais, seria interessante defini-los com base num mapa representativo do percurso operacional dimensionado, de forma a selecionar os locais por onde passaria o AGV e consequentemente as ações que este iria realizar.

Anexo A

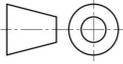
Desenho Técnico

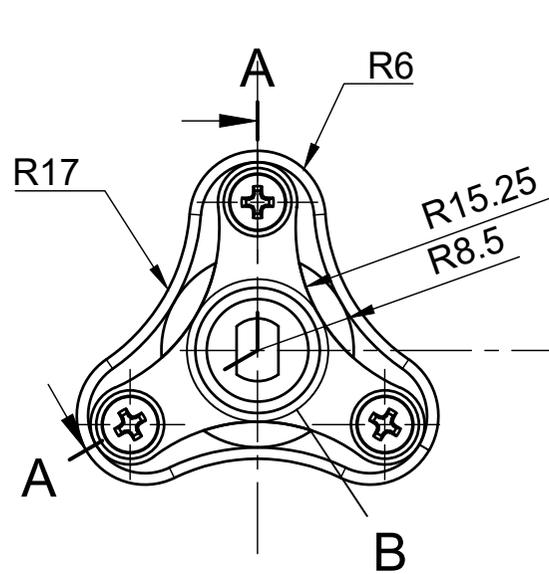
Partes referentes ao desenho técnico N°0 "*Estrutura Explodida AGV*":

1. Base Estrutural parte 1 (PLA)
2. Base Estrutural parte 2 (PLA)
3. Plataforma Roda Giratória (PLA)
4. Caixa Motor Elétrico DIR. (PLA)
5. Caixa Motor Elétrico ESQ. (PLA)
6. Base LM2596 (PLA)
7. Base XL4015 (PLA)
8. Bateria Superior (PLA)
9. Bateria Inferior (PLA)
10. Suporte Iluminação (PLA)
11. Parafuso M3 x 16 mm
12. Porca M3
13. Parafuso M4 x 12 mm
14. Porca M4
15. Parafuso M4 x 16 mm
16. Parafuso M4 x 20 mm
17. Anilha M4

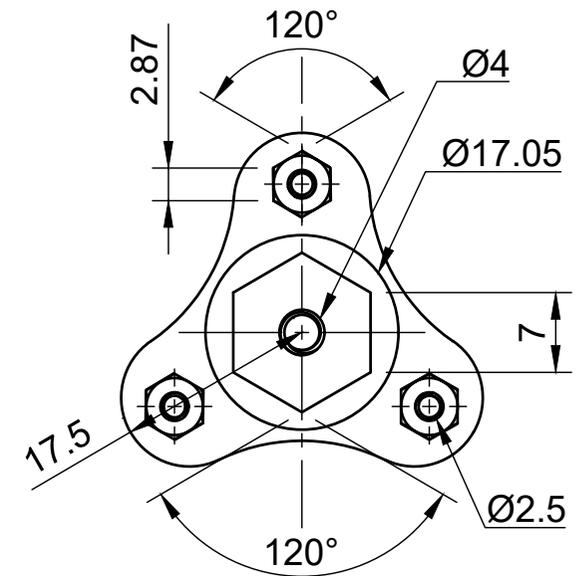
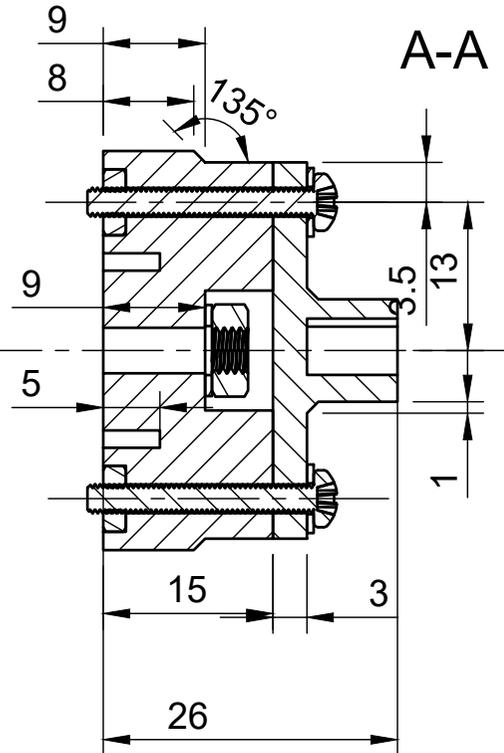
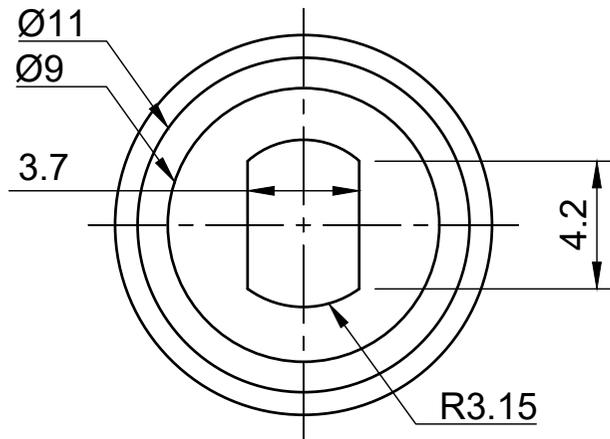


 UNIVERSIDADE DE ÉVORA	Title	
	Vista Explodida	

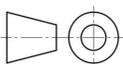
DRAW BY	Ricardo Peixe	11/10/2021	DWG No.	N°0 - Estrutura Explodida AGV 
CHECKED	Ricardo Peixe	05/01/2022	SCALE 1:1	
APPROVED	Ricardo Peixe	05/01/2022	STATUS IN USE	SHEET 1/1

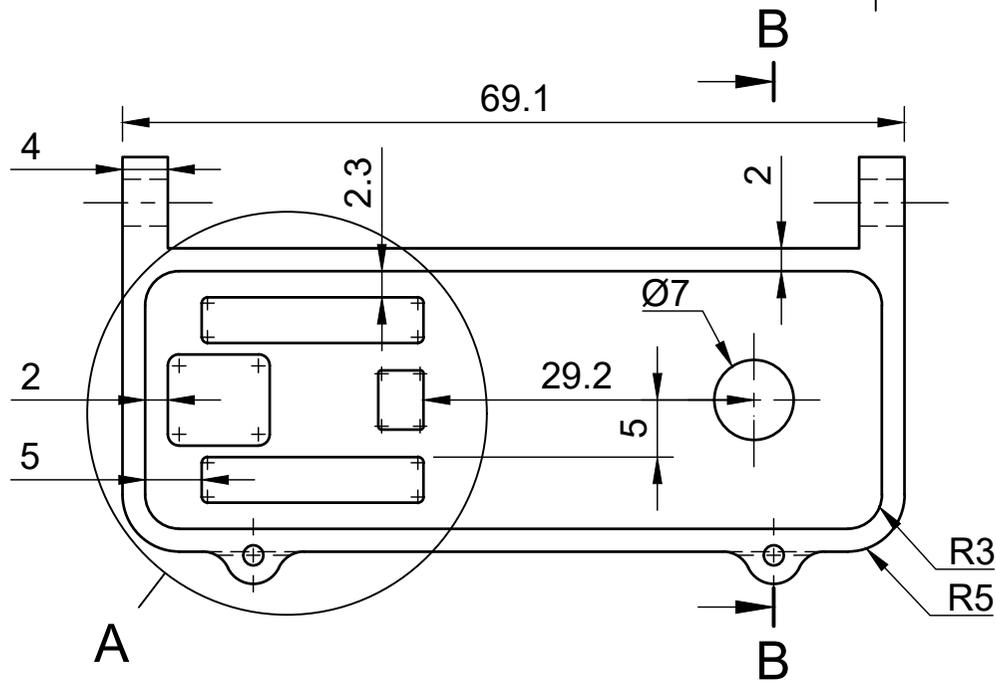


B (4:1)

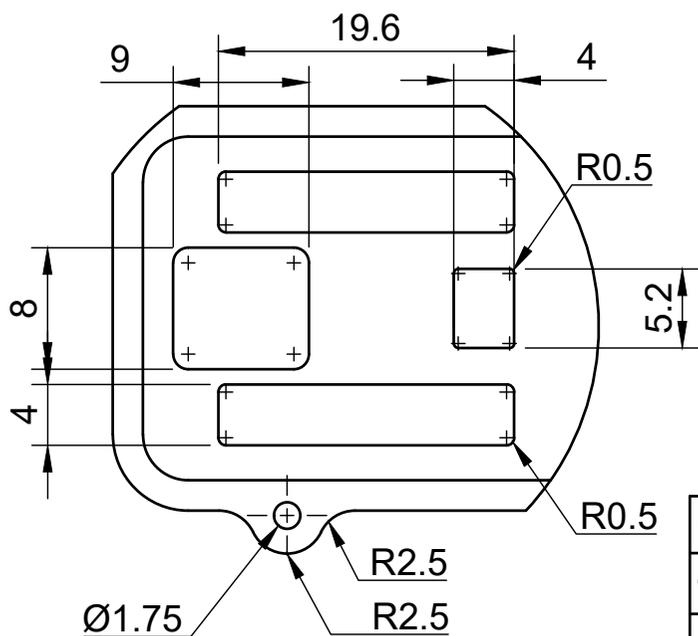
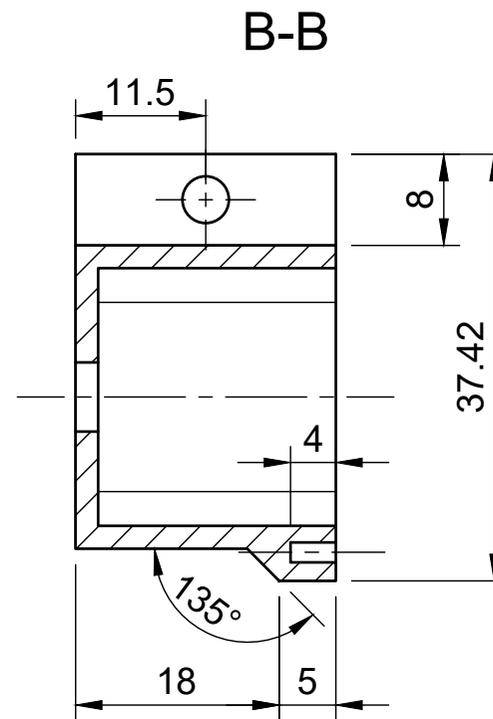


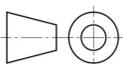
 UNIVERSIDADE DE ÉVORA	Title	
	Sistema Transmissão	

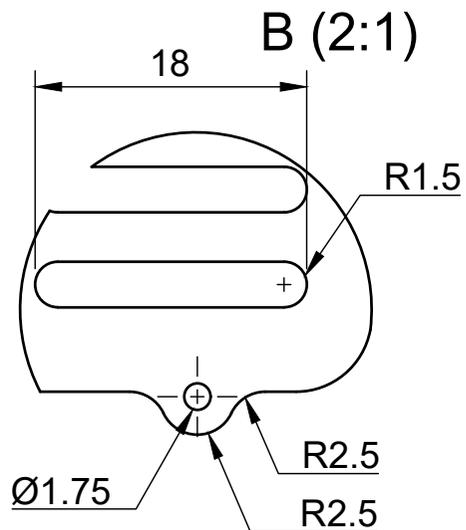
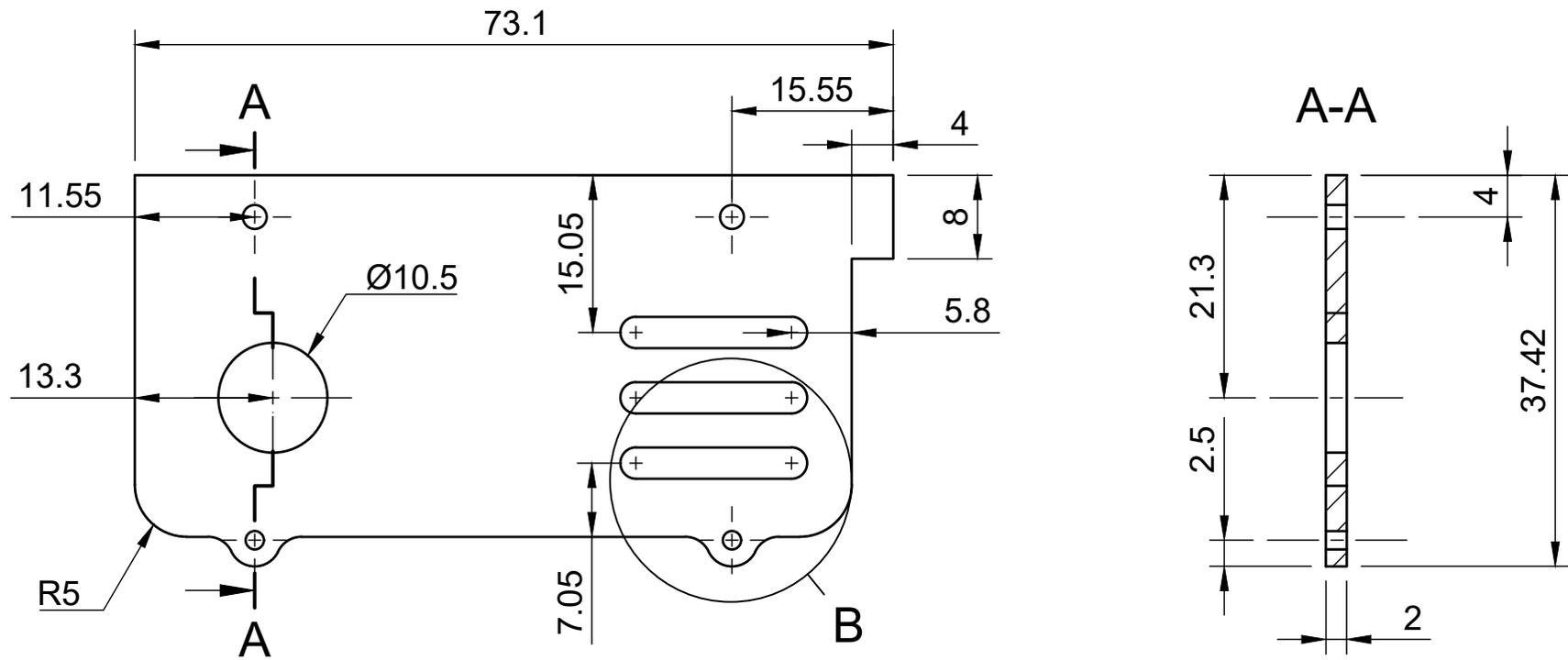
DRAW BY	Ricardo Peixe	11/10/2021	DWG No.	
CHECKED	Ricardo Peixe	05/01/2022	Nº1	
APPROVED	Ricardo Peixe	05/01/2022	SCALE 2:1	STATUS IN USE
				SHEET 1/1

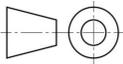


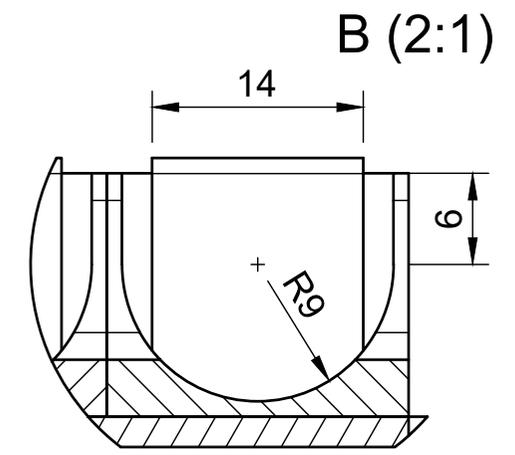
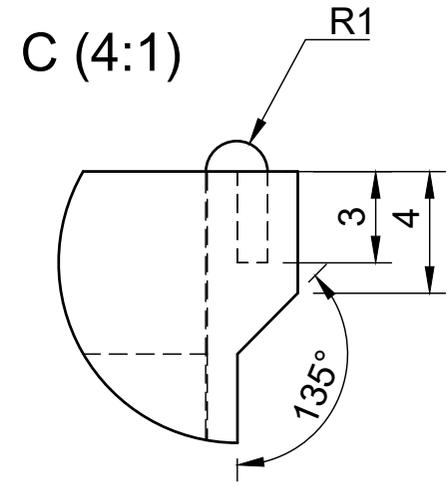
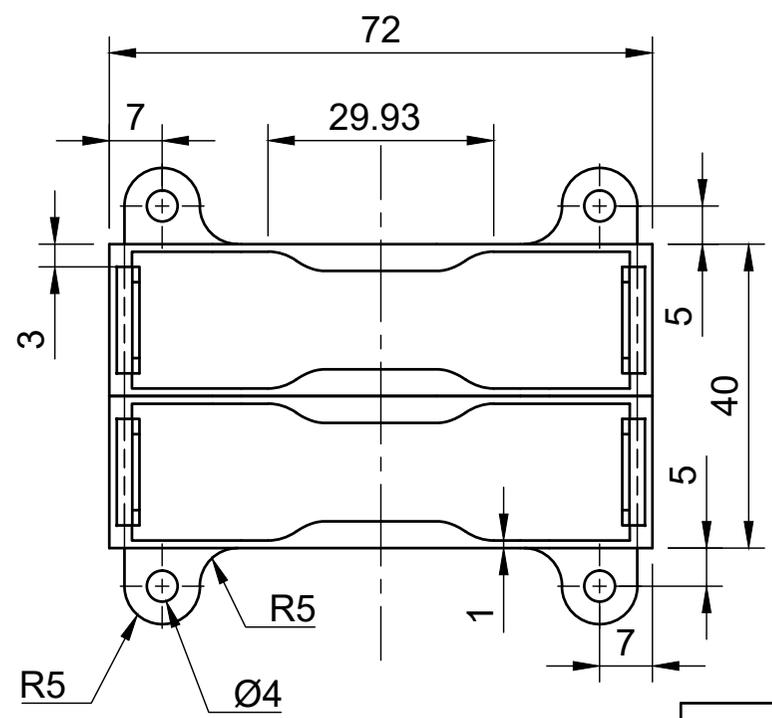
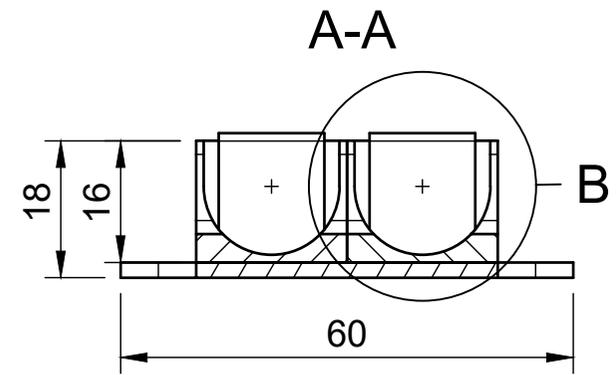
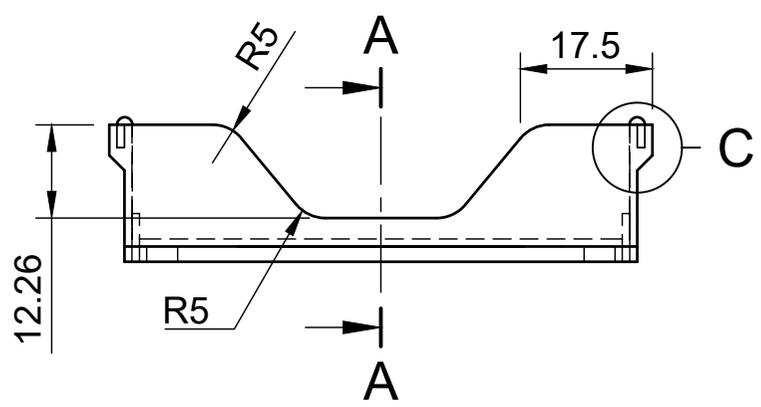
A (2:1)



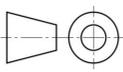
 UNIVERSIDADE DE ÉVORA		Title Caixa Motor Elétrico	
DRAW BY	Ricardo Peixe	11/10/2021	DWG No. Nº2
CHECKED	Ricardo Peixe	05/01/2022	
APPROVED	Ricardo Peixe	05/01/2022	SCALE 2:1 STATUS IN USE
			
			SHEET 1/1

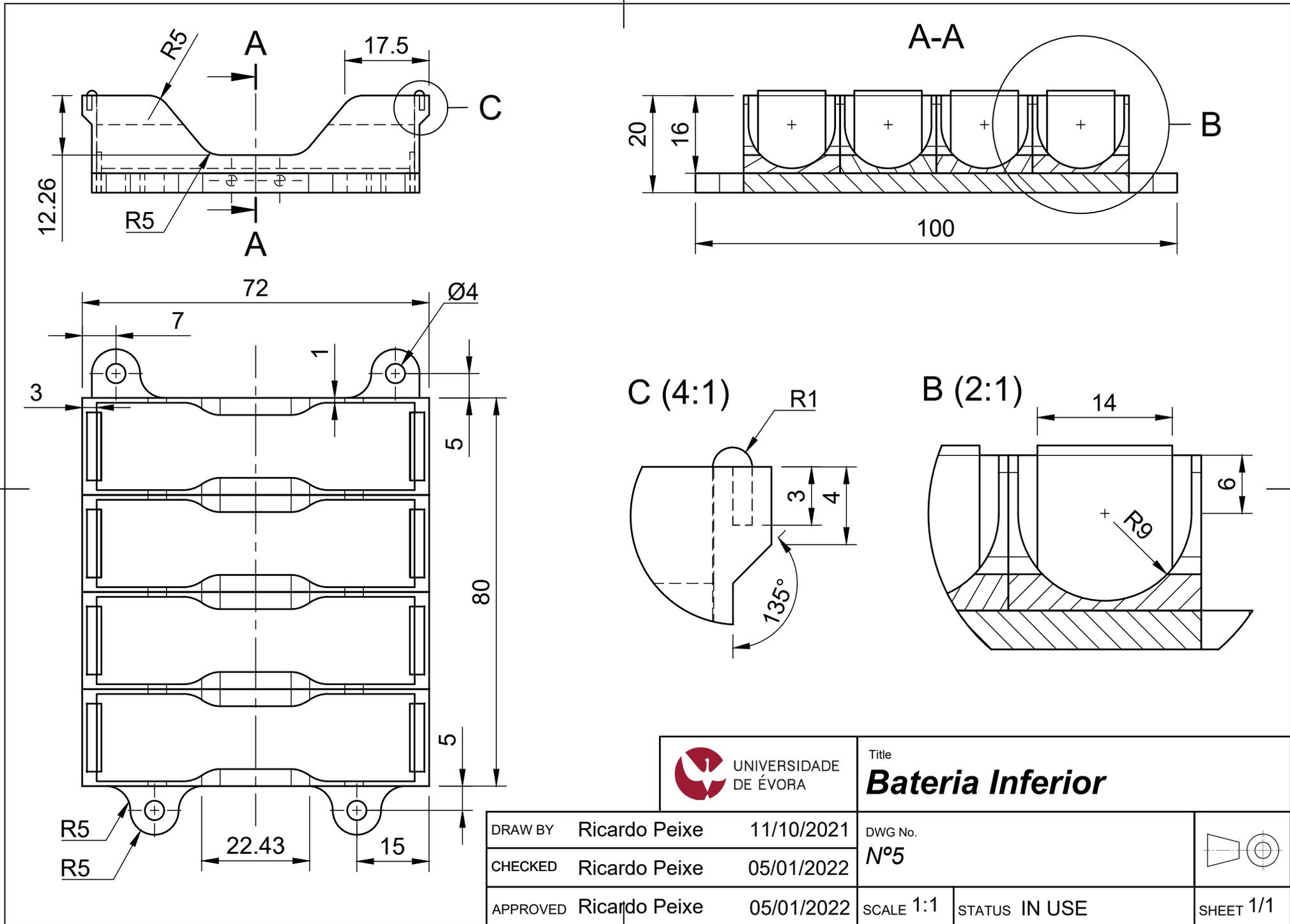


 UNIVERSIDADE DE ÉVORA		Title Tampa Motor Eléctrico	
DRAW BY	Ricardo Peixe	11/10/2021	DWG No. Nº3
CHECKED	Ricardo Peixe	05/01/2022	
APPROVED	Ricardo Peixe	05/01/2022	
			SHEET 1/1

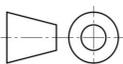


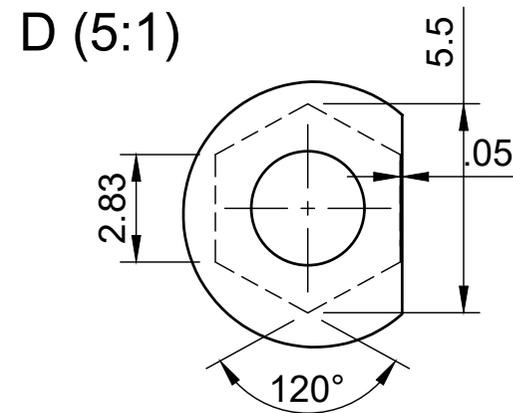
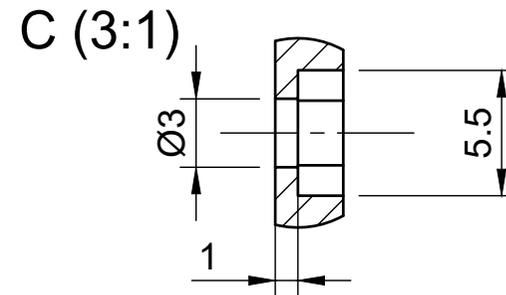
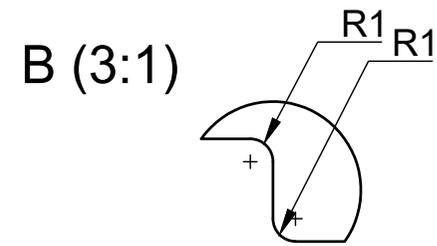
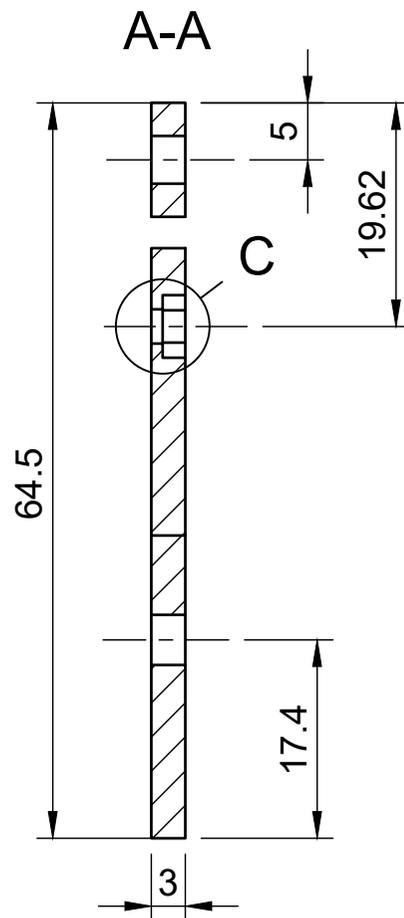
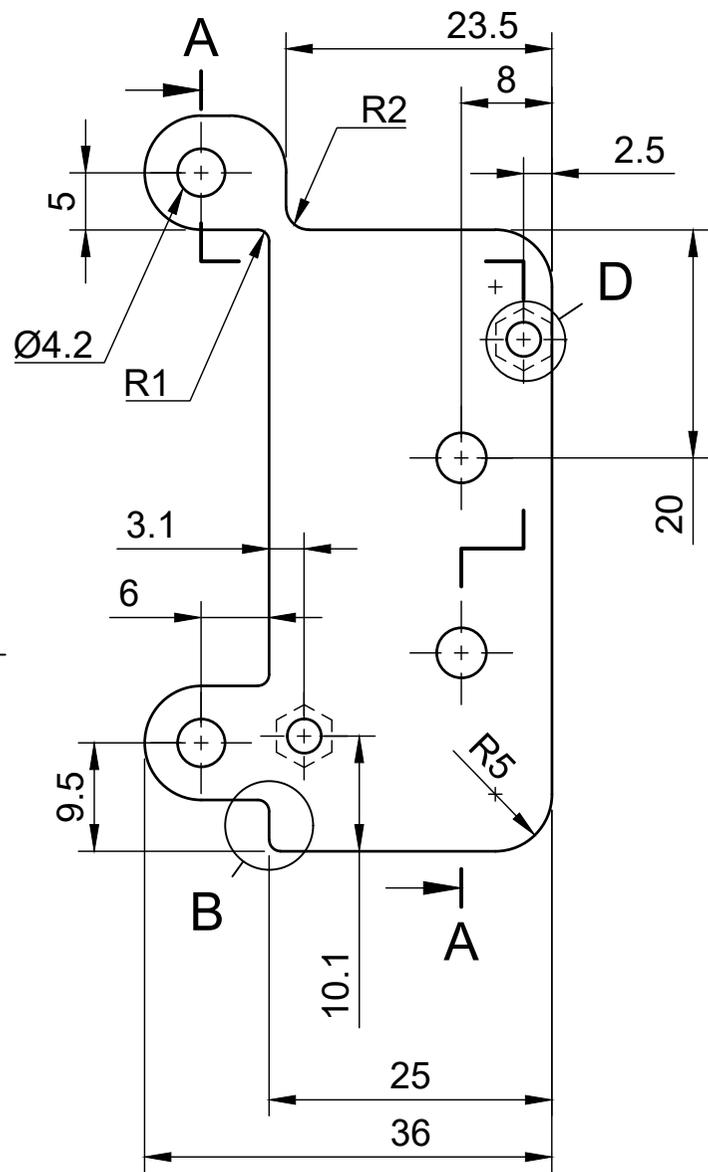
 UNIVERSIDADE DE ÉVORA	Title	
	Bateria Superior	

DRAW BY	Ricardo Peixe	11/10/2021	DWG No.	
CHECKED	Ricardo Peixe	05/01/2022	Nº4	
APPROVED	Ricardo Peixe	05/01/2022	SCALE 1:1	STATUS IN USE
				SHEET 1/1



 UNIVERSIDADE DE ÉVORA	Title	
	Bateria Inferior	

DRAW BY	Ricardo Peixe	11/10/2021	DWG No.	
CHECKED	Ricardo Peixe	05/01/2022	Nº5	
APPROVED	Ricardo Peixe	05/01/2022	SCALE 1:1	STATUS IN USE
				SHEET 1/1



UNIVERSIDADE DE ÉVORA

Title
Base XL4015

DRAW BY Ricardo Peixe 11/10/2021

CHECKED Ricardo Peixe 05/01/2022

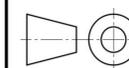
APPROVED Ricardo Peixe 05/01/2022

DWG No.

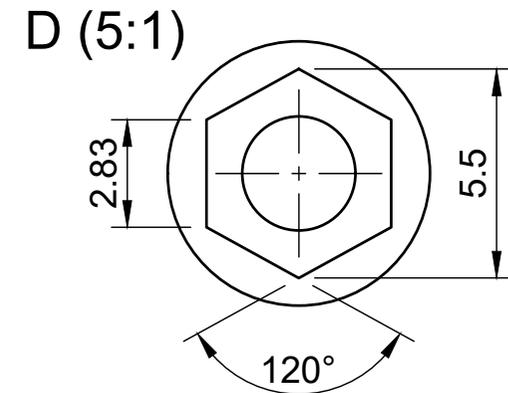
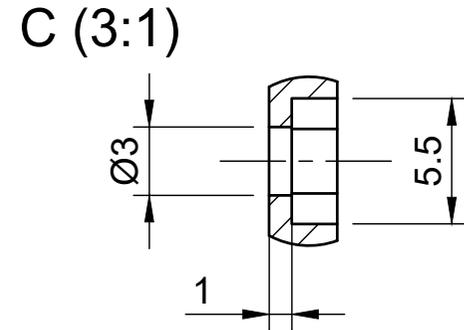
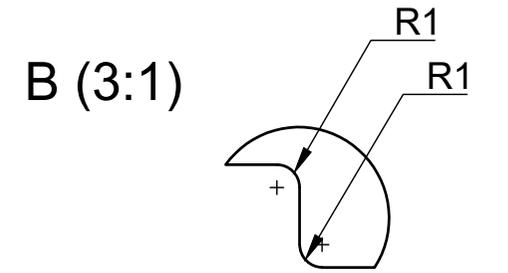
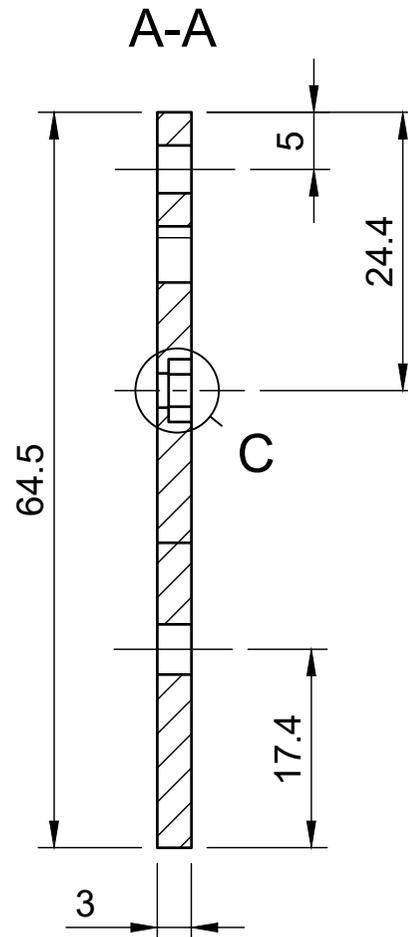
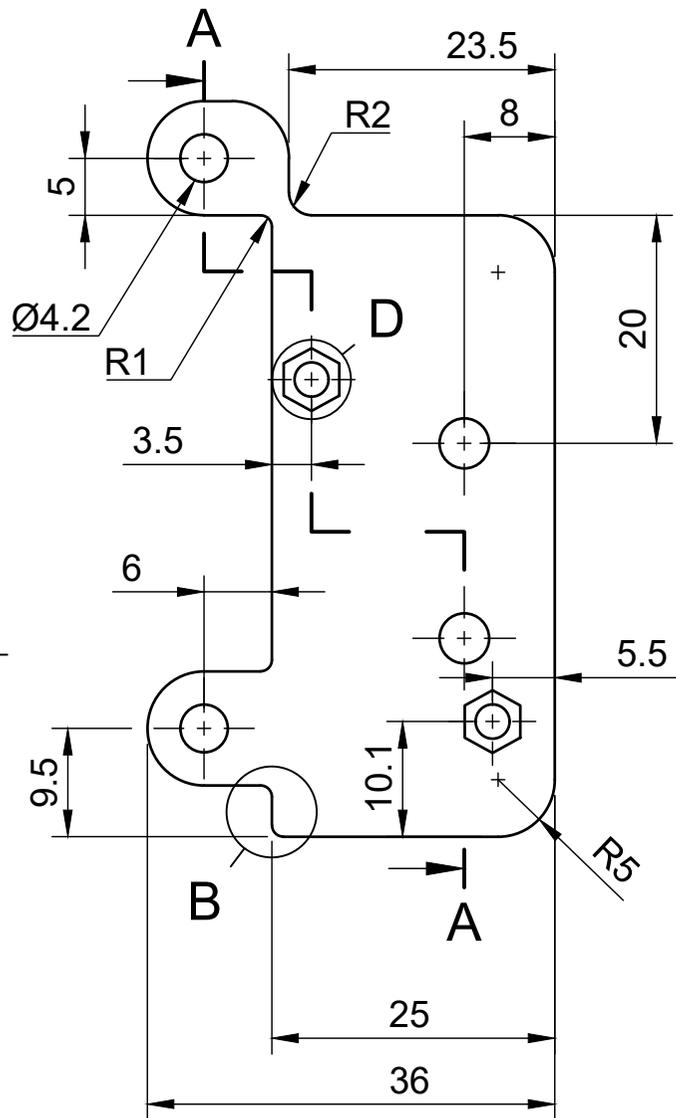
Nº6

SCALE 2:1

STATUS IN USE



SHEET 1/1



Title
Base LM2596

DRAW BY Ricardo Peixe 11/10/2021

CHECKED Ricardo Peixe 05/01/2022

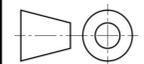
APPROVED Ricardo Peixe 05/01/2022

DWG No.

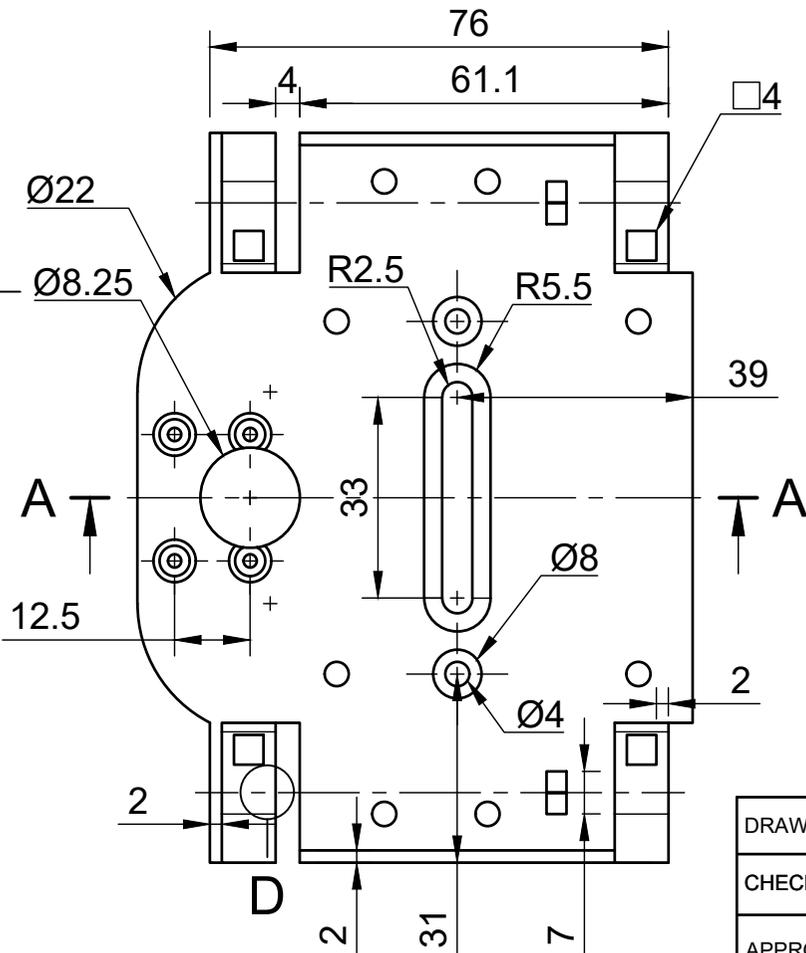
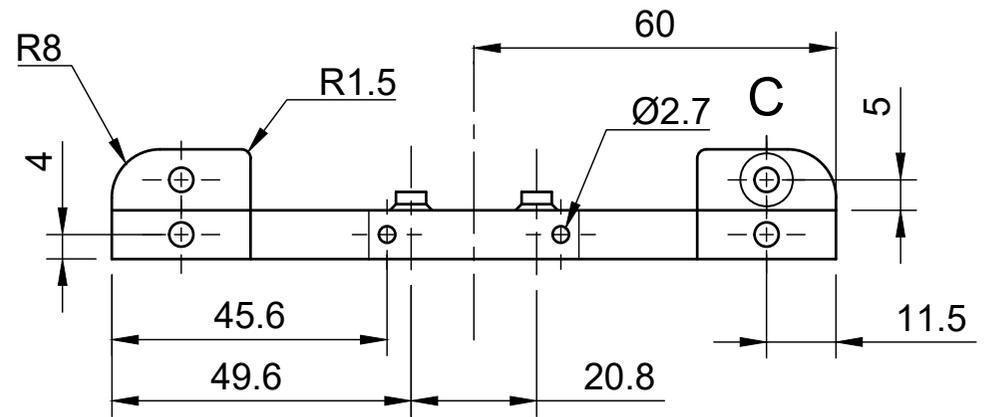
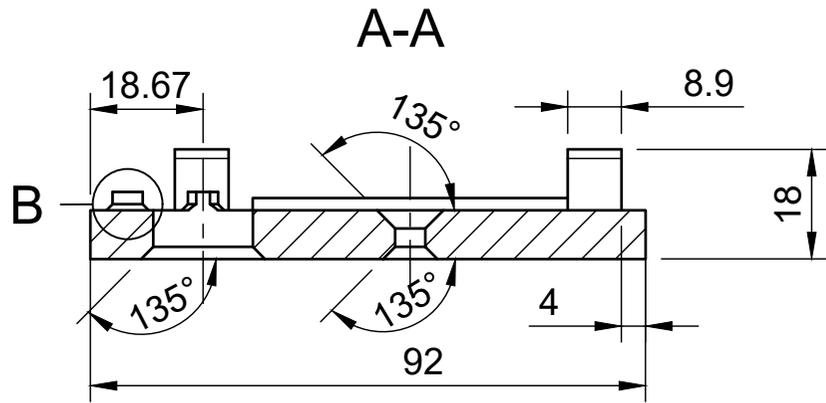
Nº7

SCALE 2:1

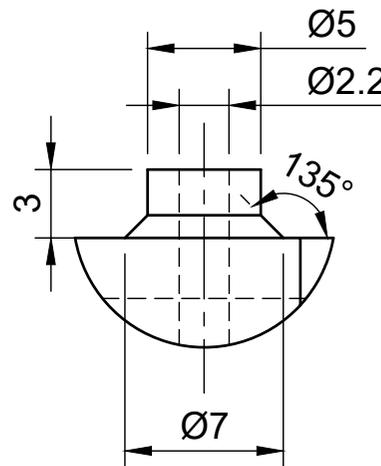
STATUS IN USE



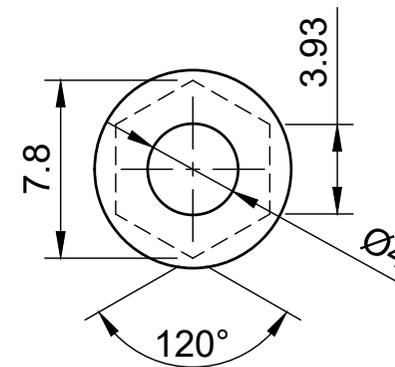
SHEET 1/1



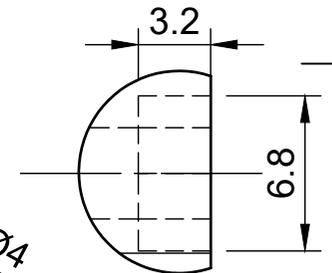
B (3:1)



C (3:1)



D (3:1)



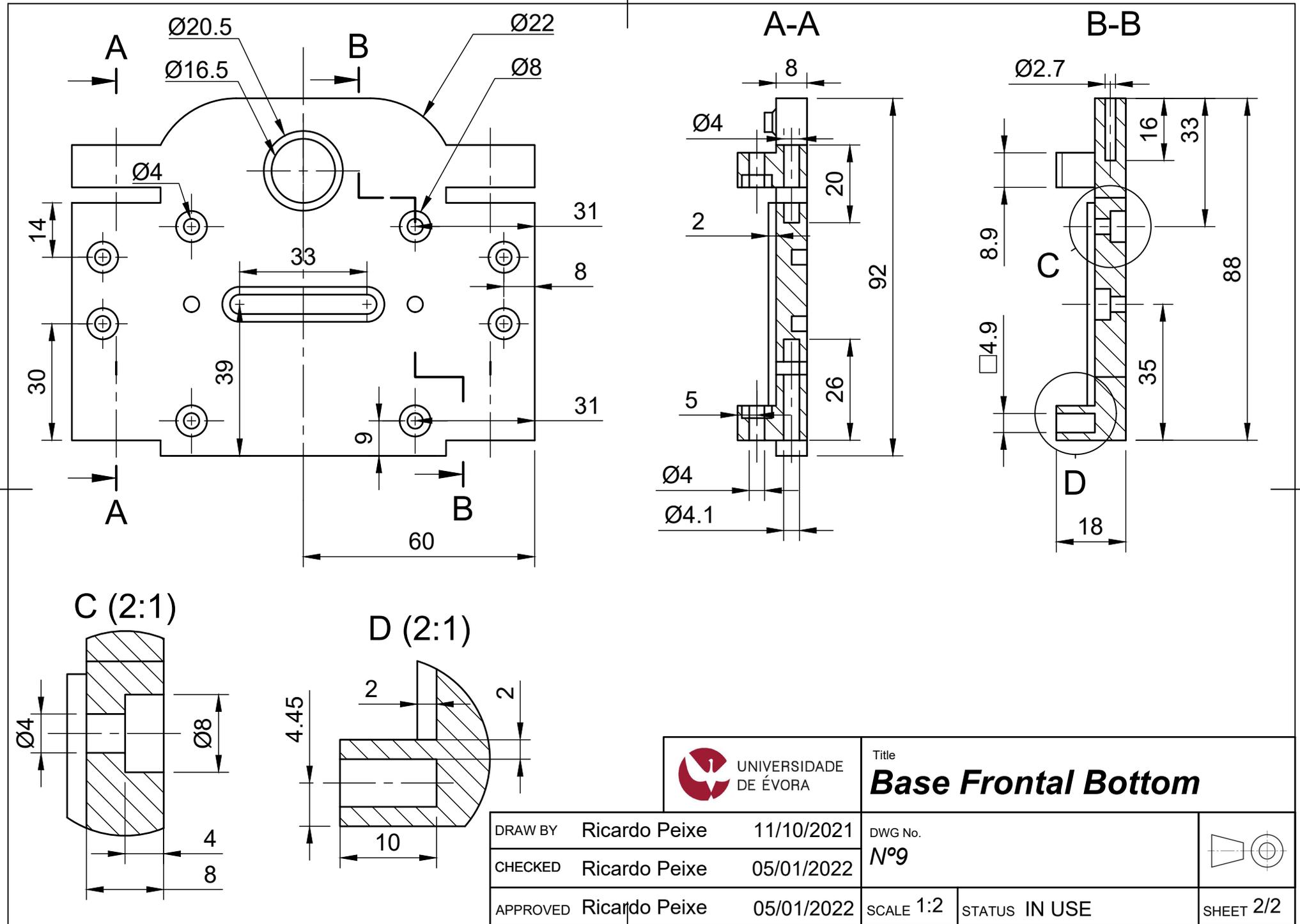
UNIVERSIDADE DE ÉVORA

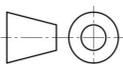
Title
Base Frontal Top

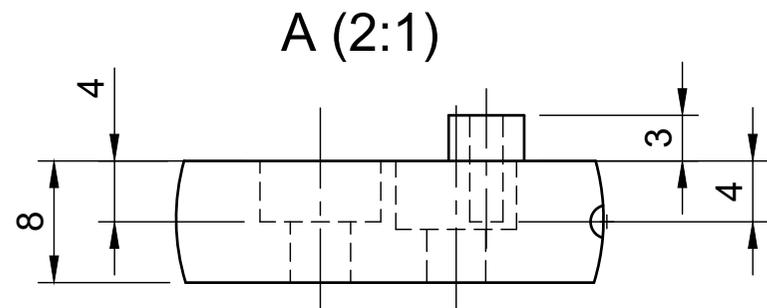
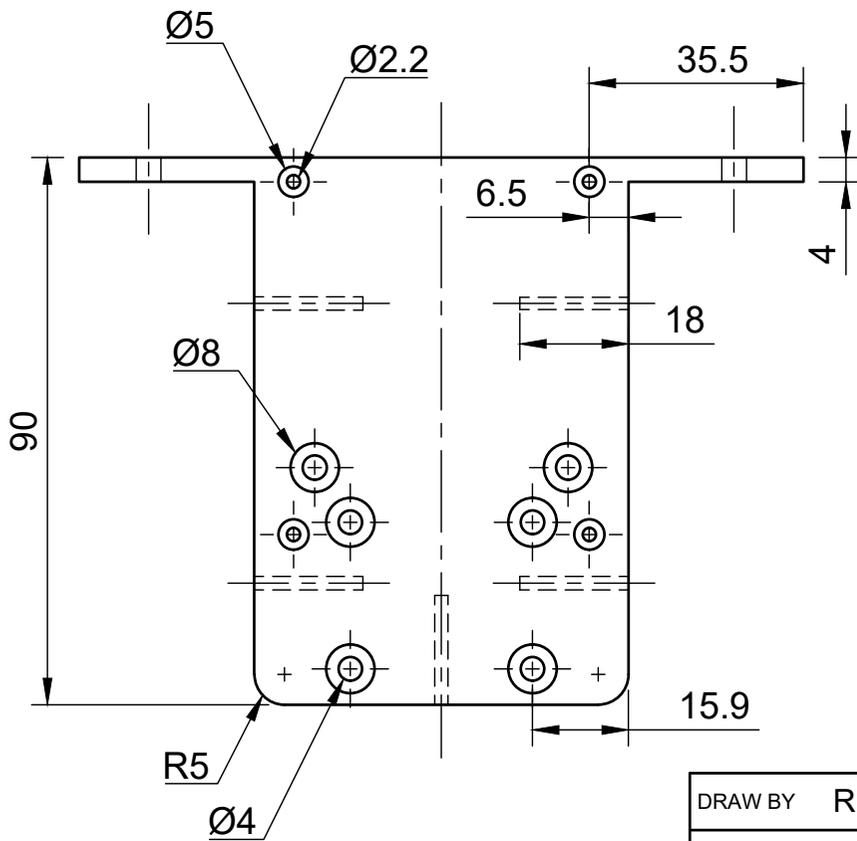
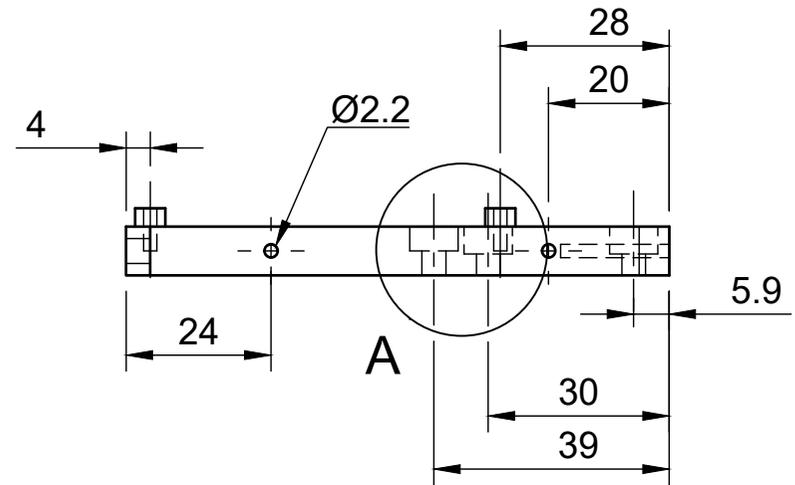
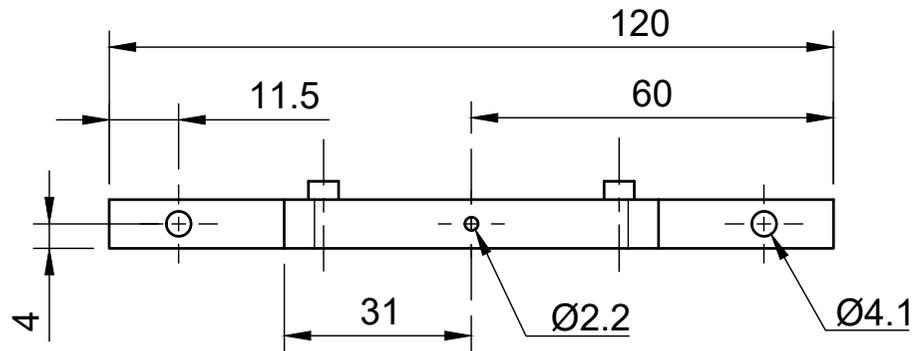
DRAW BY	Ricardo Peixe	11/10/2021
CHECKED	Ricardo Peixe	05/01/2022
APPROVED	Ricardo Peixe	05/01/2022

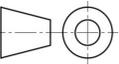
DWG No.	Nº8
SCALE	1:2
STATUS	IN USE

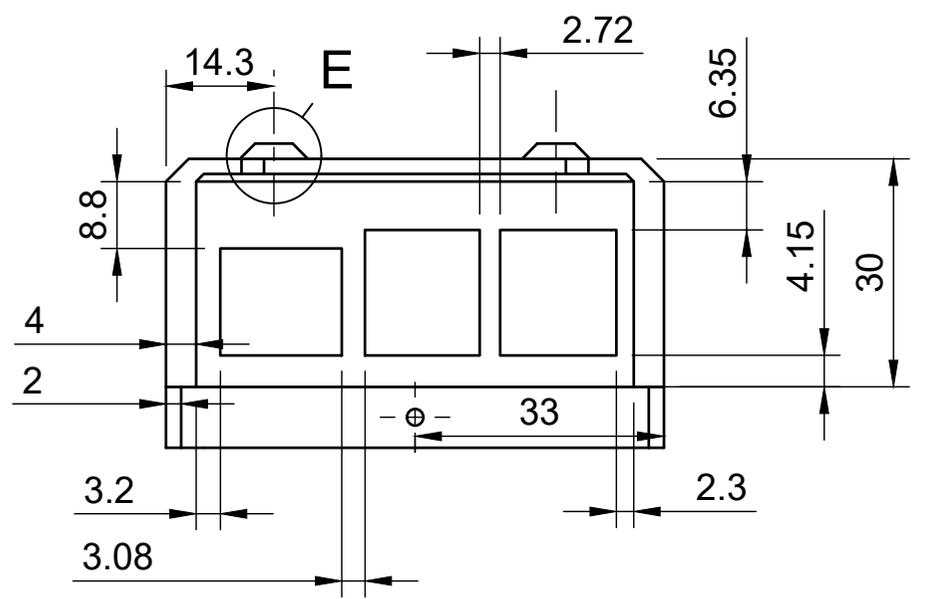
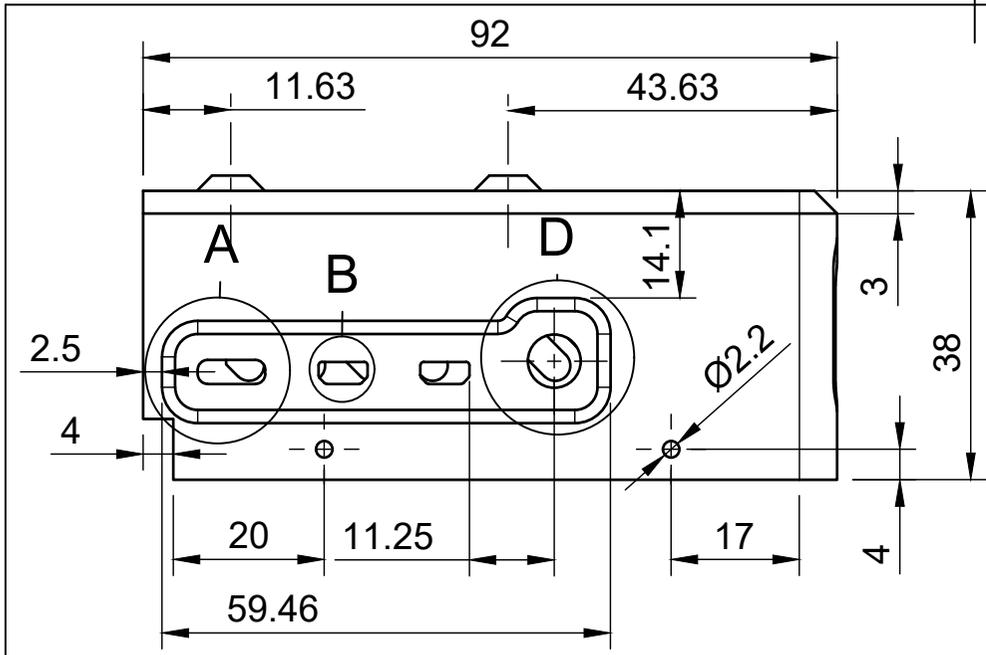
SHEET 1/2



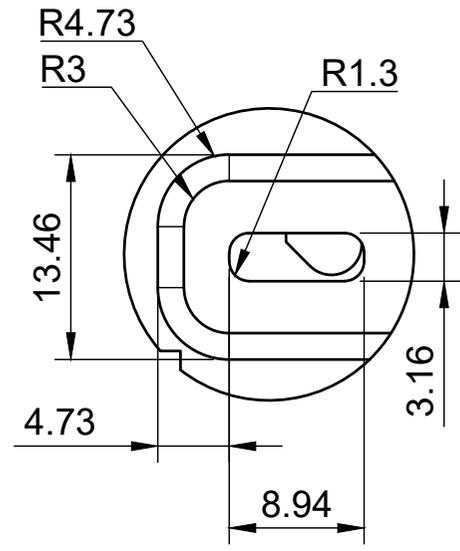
 UNIVERSIDADE DE ÉVORA		Title		
		Base Frontal Bottom		
DRAW BY	Ricardo Peixe	11/10/2021	DWG No.	
CHECKED	Ricardo Peixe	05/01/2022	Nº9	
APPROVED	Ricardo Peixe	05/01/2022	SCALE 1:2	STATUS IN USE
				SHEET 2/2



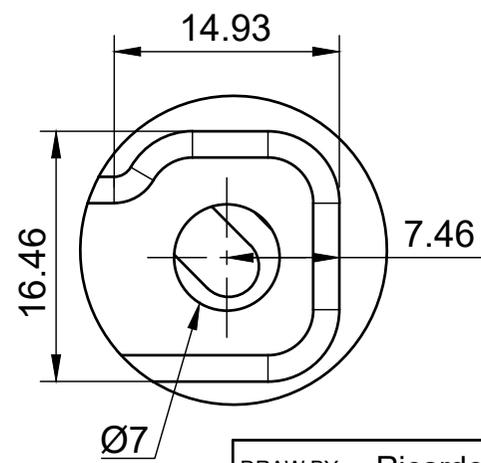
 UNIVERSIDADE DE ÉVORA		Title Base Traseira	
DRAW BY	Ricardo Peixe	11/10/2021	DWG No. Nº10
CHECKED	Ricardo Peixe	05/01/2022	
APPROVED	Ricardo Peixe	05/01/2022	SCALE 1:2 STATUS IN USE
			 SHEET 1/1



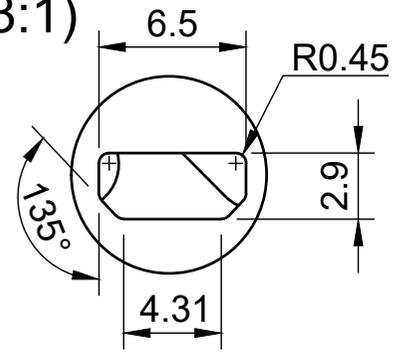
A (2:1)



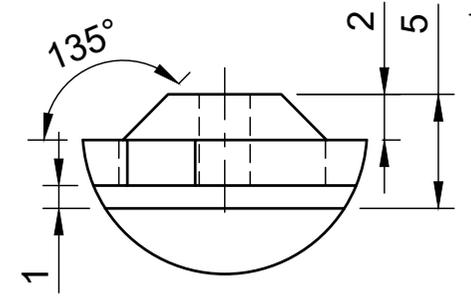
D (2:1)



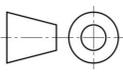
B (3:1)

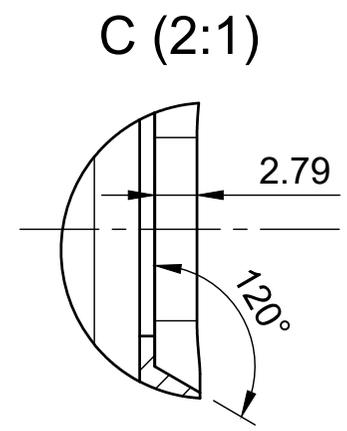
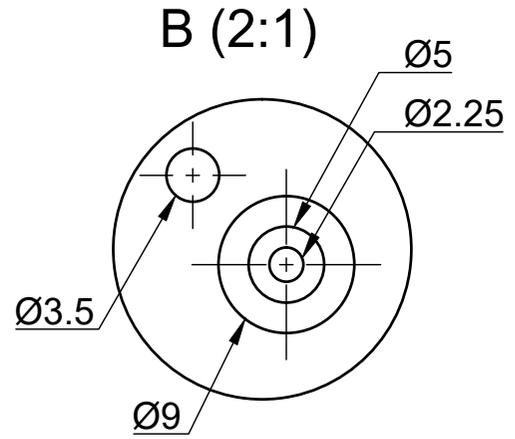
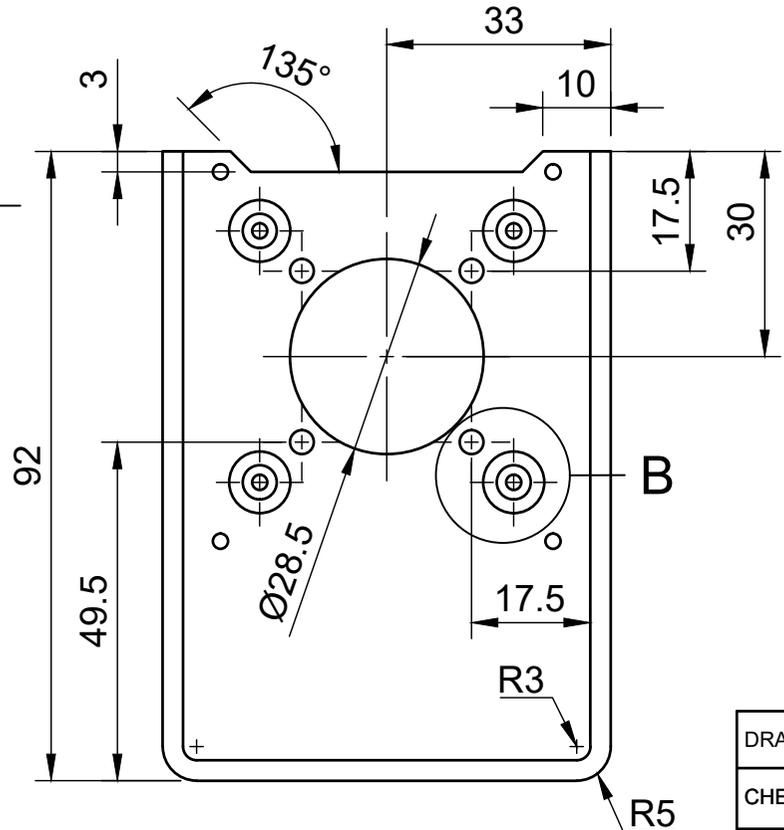
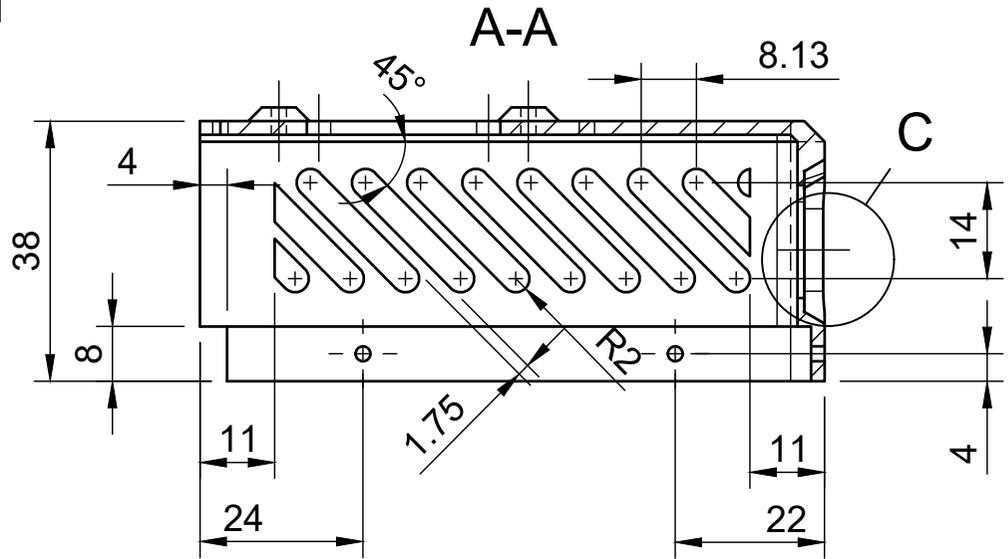
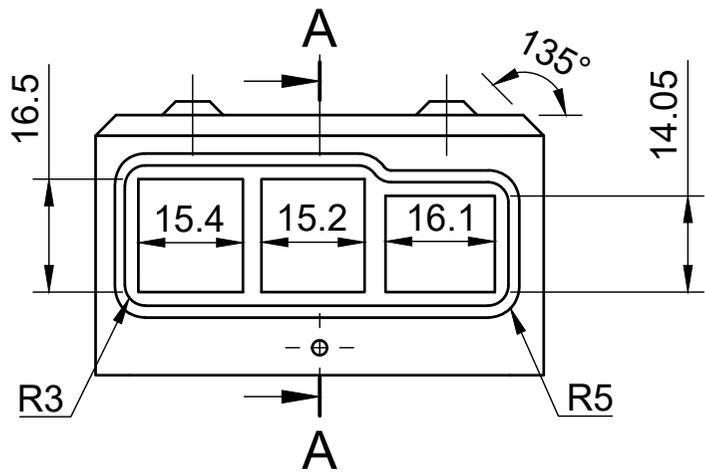


E (3:1)

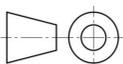


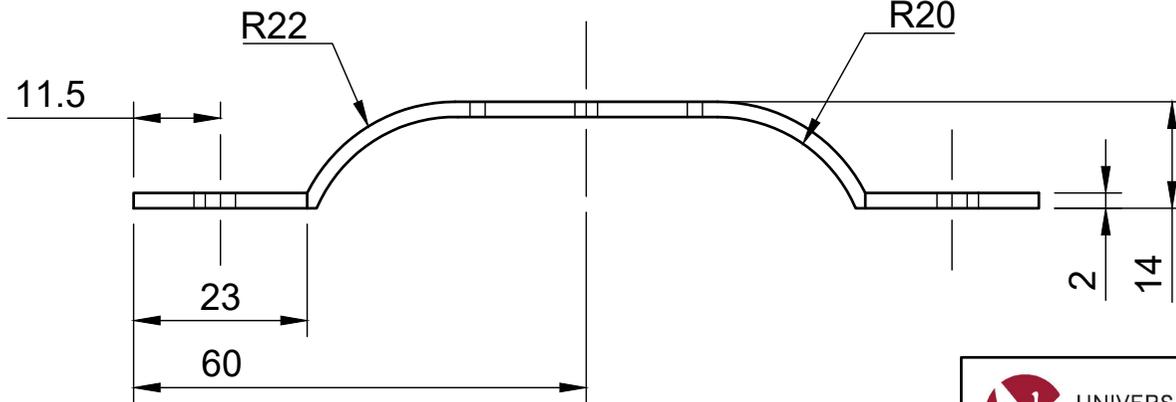
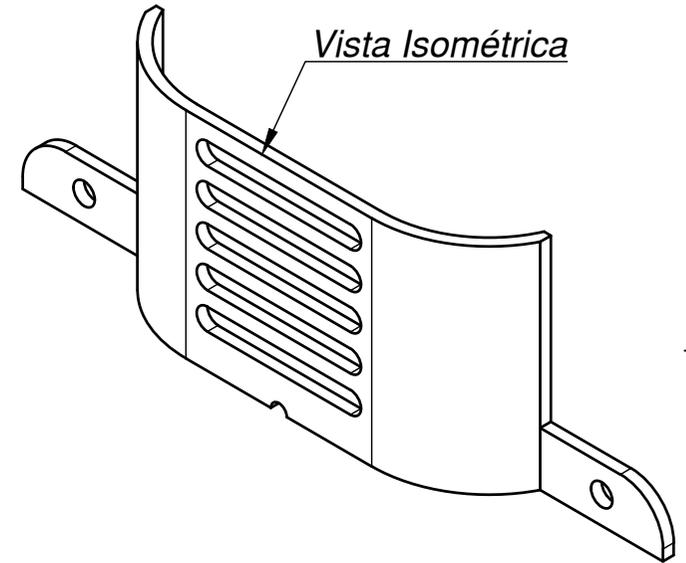
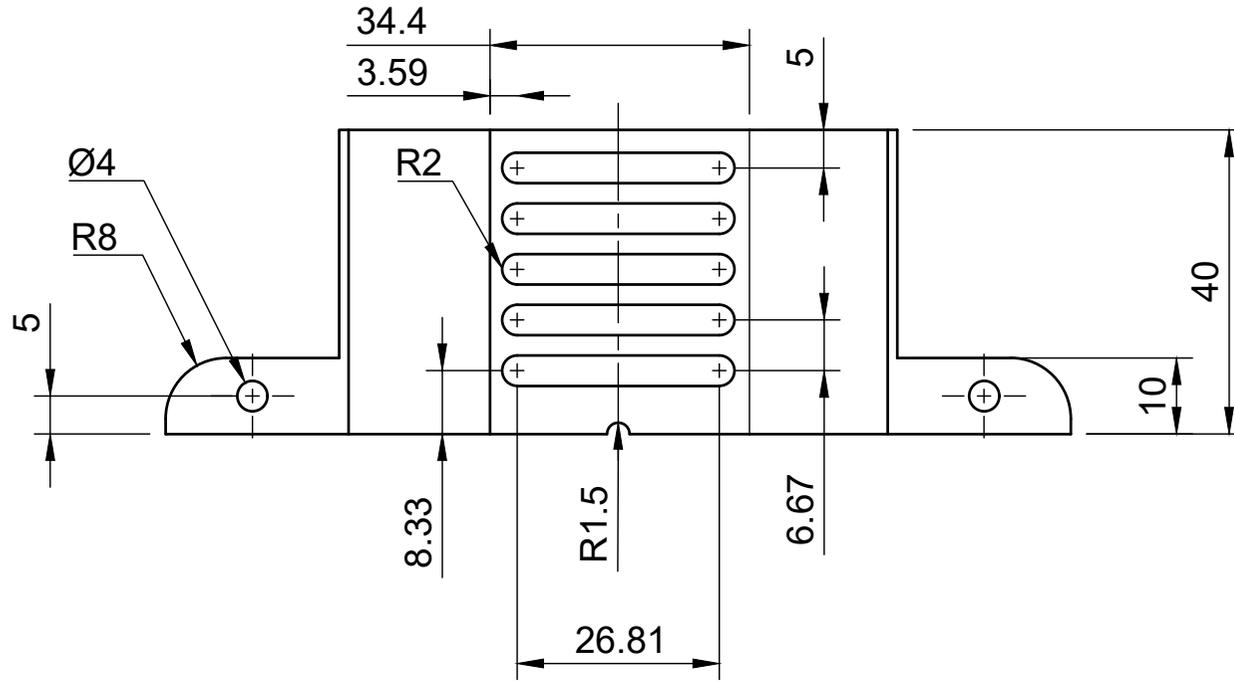
 UNIVERSIDADE DE ÉVORA	Title	
	<i>Carcaça Raspberry Pi</i>	

DRAW BY	Ricardo Peixe	11/10/2021	DWG No.	 N°11
CHECKED	Ricardo Peixe	05/01/2022		
APPROVED	Ricardo Peixe	05/01/2022	SCALE 1:1	STATUS IN USE
				SHEET 1/2



 UNIVERSIDADE DE ÉVORA	Title	
	<i>Carcaça Raspberry Pi</i>	

DRAW BY	Ricardo Peixe	11/10/2021	DWG No.	Nº12	
CHECKED	Ricardo Peixe	05/01/2022	SCALE 1:1		
APPROVED	Ricardo Peixe	05/01/2022	SCALE 1:1	STATUS IN USE	SHEET 2/2



UNIVERSIDADE
DE ÉVORA

Title

Grelha Frontal

DRAW BY Ricardo Peixe 11/10/2021

CHECKED Ricardo Peixe 05/01/2022

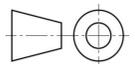
APPROVED Ricardo Peixe 05/01/2022

DWG No.

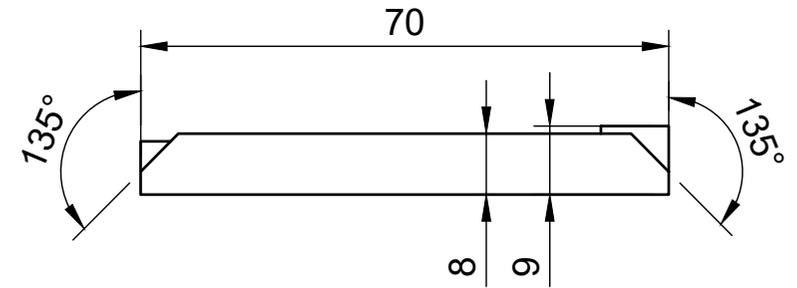
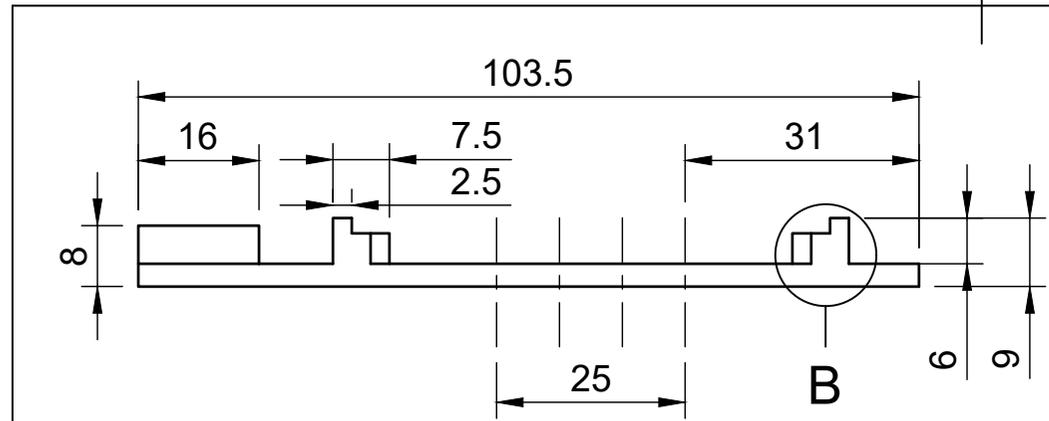
Nº13

SCALE 1:1

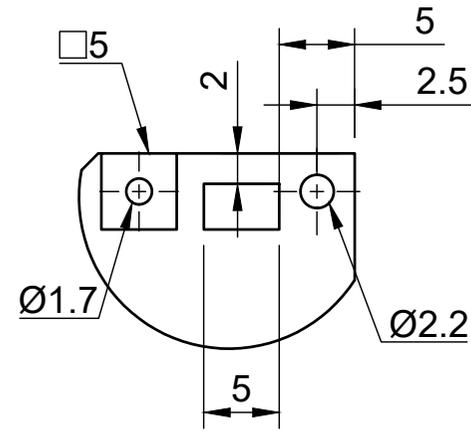
STATUS IN USE



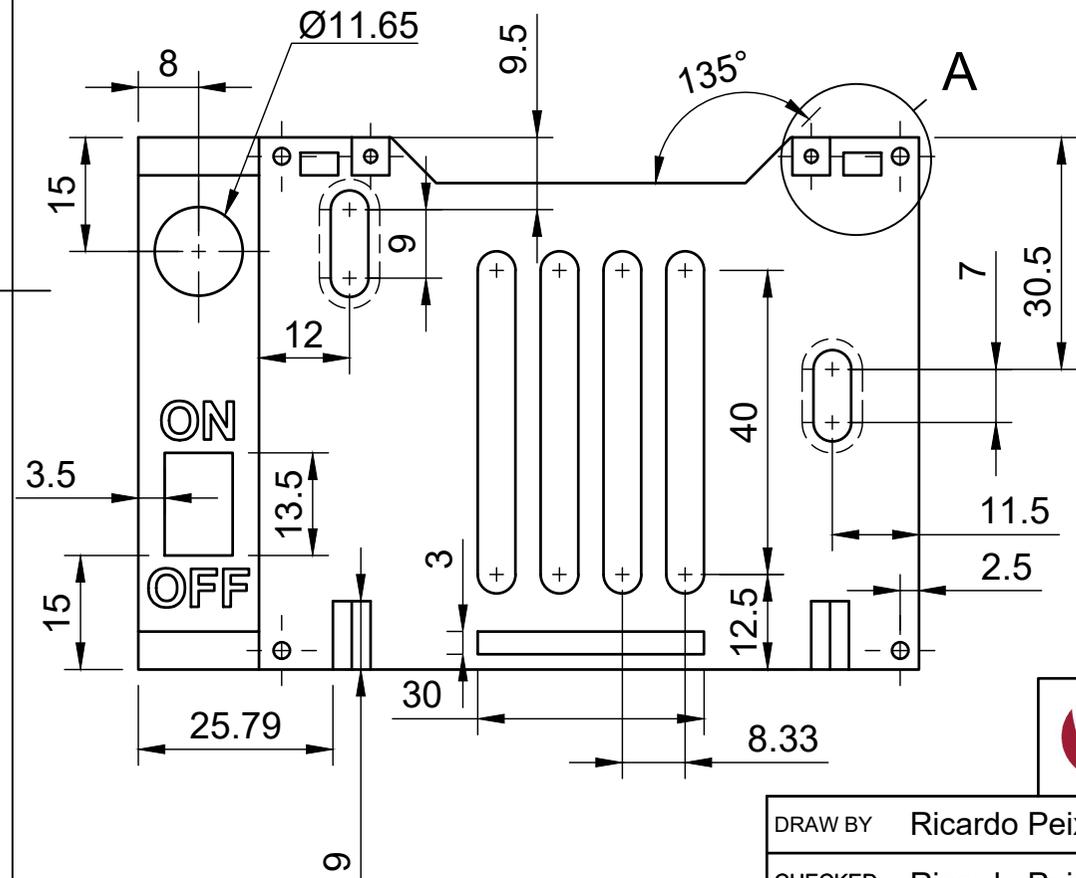
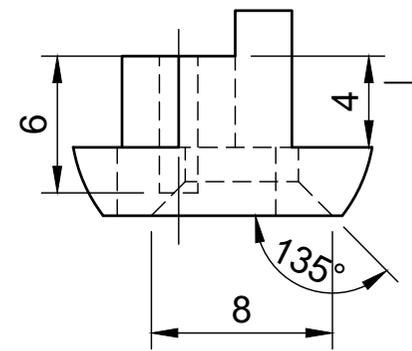
SHEET 1/1



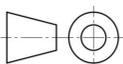
A (2:1)

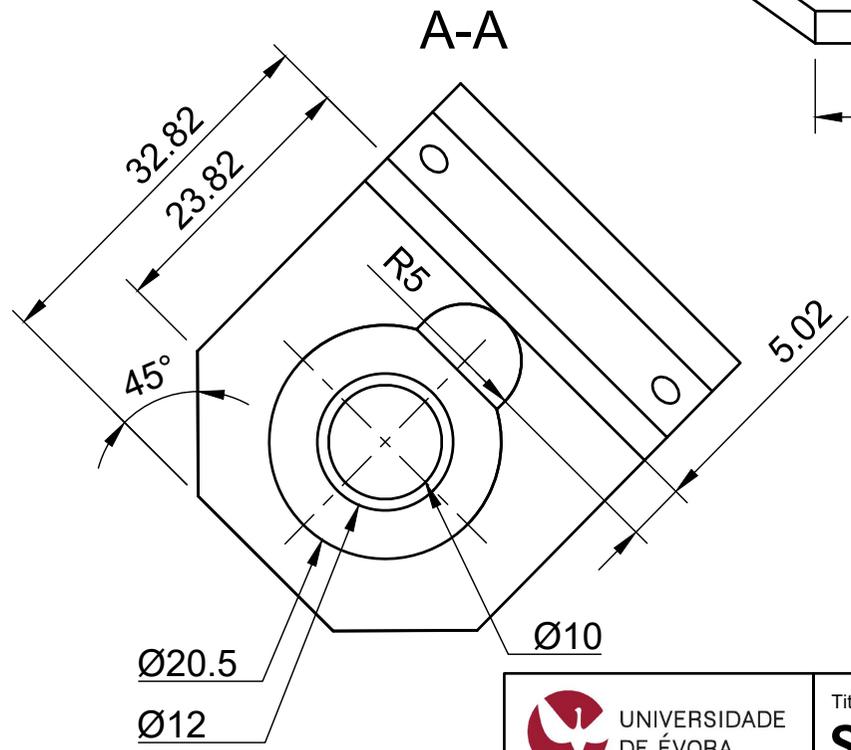
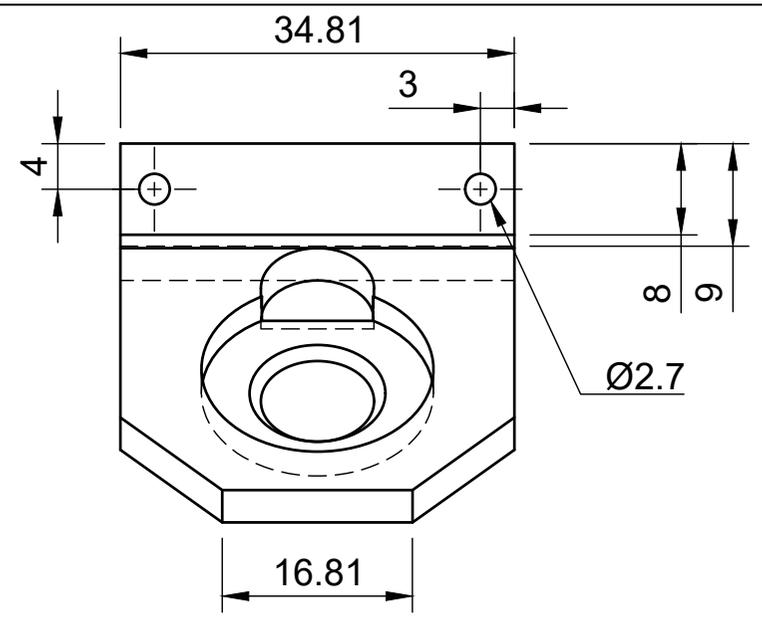
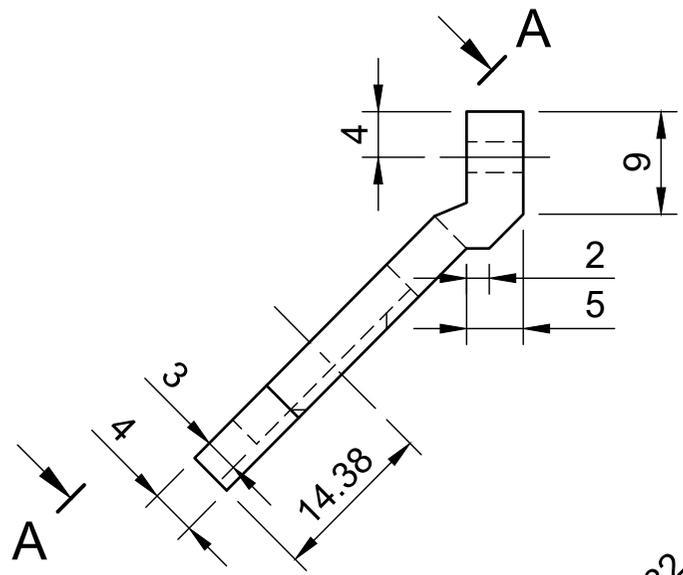


B (3:1)

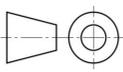


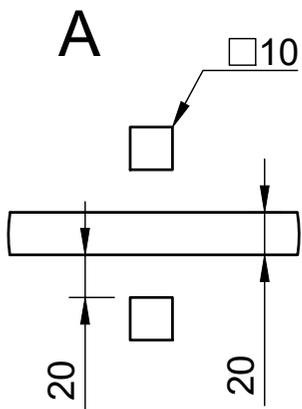
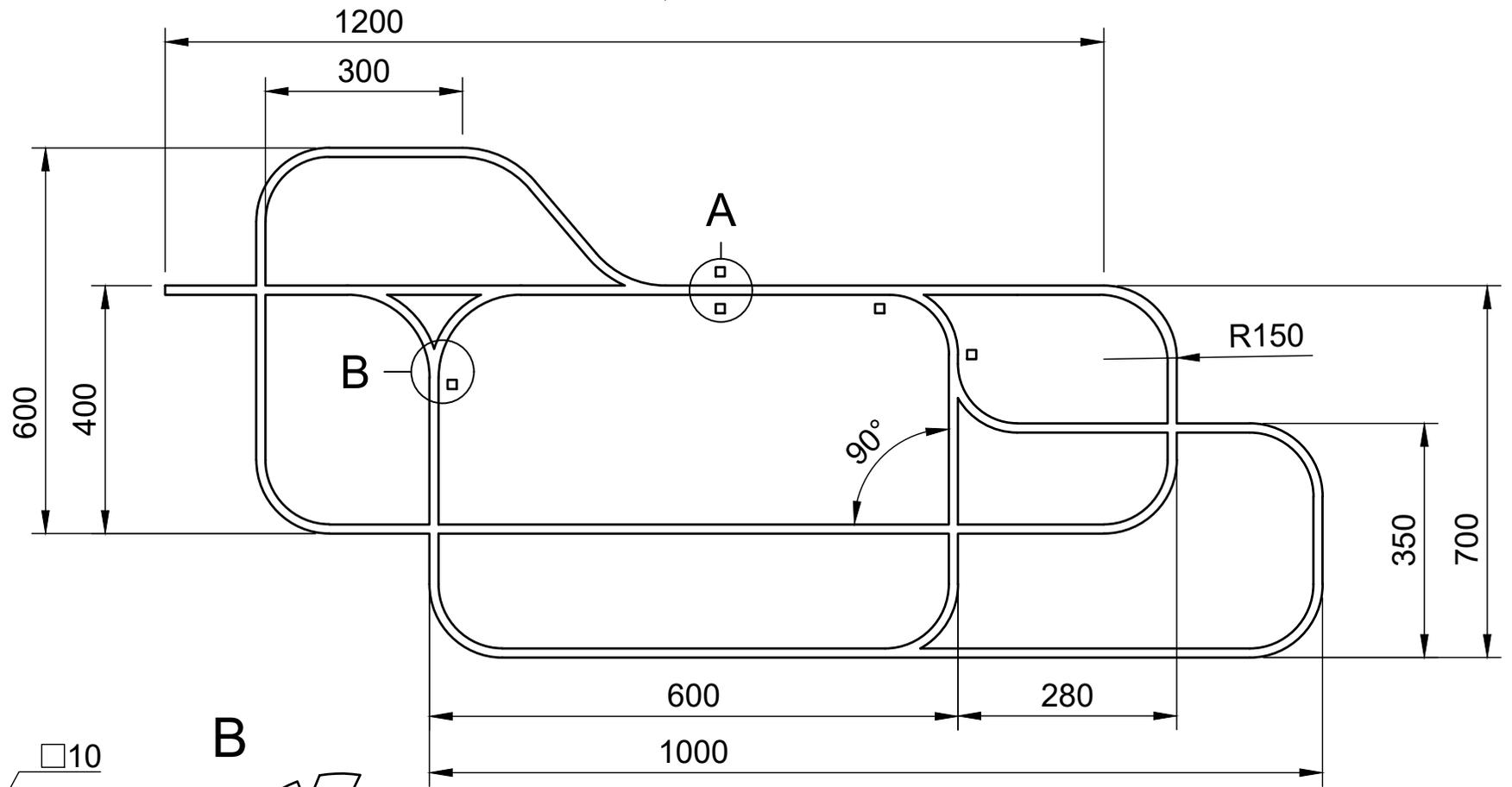
 UNIVERSIDADE DE ÉVORA	Title	
	Suporte BMS	

DRAW BY	Ricardo Peixe	11/10/2021	DWG No.	Nº14	
CHECKED	Ricardo Peixe	05/01/2022			
APPROVED	Ricardo Peixe	05/01/2022	SCALE 1:1	STATUS IN USE	

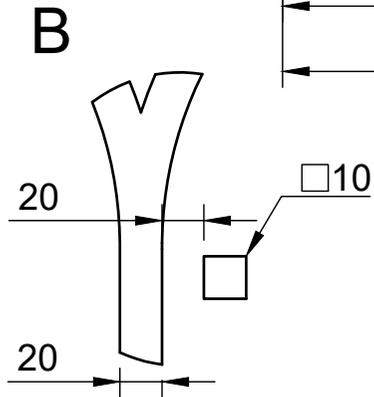


 UNIVERSIDADE DE ÉVORA	Title	
	Suporte Iluminação	

DRAW BY	Ricardo Peixe	11/10/2021	DWG No.	
CHECKED	Ricardo Peixe	05/01/2022	Nº15	
APPROVED	Ricardo Peixe	05/01/2022	SCALE 2:1	STATUS IN USE
				SHEET 1/1

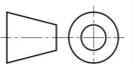


Sinal de Reset



Sinal de Mudança de Direção

 UNIVERSIDADE DE ÉVORA		Title Percurso Operacional	
DRAW BY	Ricardo Peixe	17/12/2021	DWG No. Nº16
CHECKED	Ricardo Peixe	06/01/2022	
APPROVED	Ricardo Peixe	06/01/2022	SCALE 1:10 STATUS IN USE
			SHEET 1/1

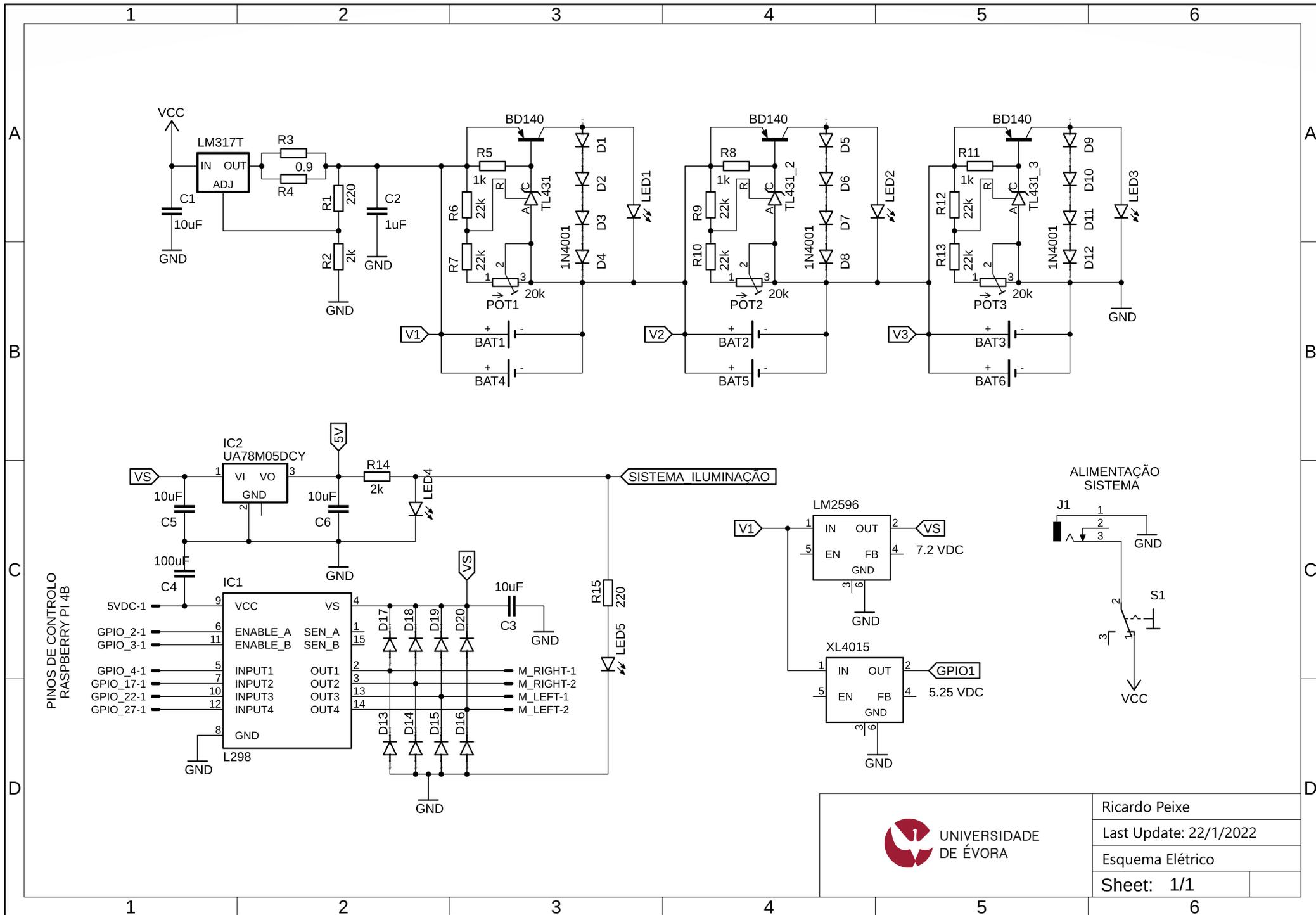


Anexo B

Esquema Elétrico

Lista de Componentes:

1. Resistência de 220 Ω (x4)
2. Resistência de 2.2 k Ω (x3)
3. Resistência de 1 k Ω (x3)
4. Resistência de 22 k Ω (x6)
5. Resistência de 1 Ω - 5 Watt (x2)
6. Condensador eletrolítico 10 μ F 25 V (x1)
7. Condensador eletrolítico 1 μ F 25 V (x1)
8. Regulador de Tensão LM317T + Alhetas (x1)
9. Díodo de silício 1N4001 - 25 V (x12)
10. Díodo Zener TL431 (x3)
11. Transistor Bipolar BD140 (x3)
12. Potenciômetro 20K \pm 5%TOL. (x3)
13. LED - 0.25 Watt (x3)
14. LED + alhetas (sistema de iluminação) - 1 Watt (x1)
15. Pinos macho para GPIO (4x1) (x3)
16. XL4015 (x1)
17. LM2596 (x1)
18. Ponte H I298N (x1)
19. Bateria ICR18650 (x6)
20. Interruptor de 2 estados ON/OFF (x1)
21. Raspberry Pi 4B (x1)



UNIVERSIDADE DE ÉVORA

Ricardo Peixe
Last Update: 22/1/2022
Esquema Elétrico
Sheet: 1/1

Anexo C

Algoritmo Computacional

Lista de *scripts*:

1. main.py
2. tools.py
3. motores.py
4. recorridos.py

```

1  ''' __ main.py __ '''
2
3  import cv2
4  from motores import Motor
5  from percursos import percursos
6  import tools, time, datetime
7
8  camera = cv2.VideoCapture(0)
9  FONTE = cv2.FONT_HERSHEY_SIMPLEX
10 agv = Motor(2,3,4,17,22,27)
11
12 if __name__ == '__main__':
13
14     seguir_em_frente = True
15     cnt_1, cnt_2 = 0, 0
16     keys = list(percursos.keys())
17     values = list(percursos.values())
18     while True:
19         start = time.time()
20         _, frame = camera.read()
21         roi = frame[250:300, 0:640]
22         N_sinais, N_reset, imagem_percurso = tools.contador(roi)
23         sinal_detected = tools.contornos(roi)[1]
24
25         if N_reset >= 1:
26             if seguir_em_frente == True:
27                 Ix = tools.direcao_frente(roi)[0]
28                 if N_sinais != 0:
29                     for i in range(N_sinais-1, len(values[N_reset-1])):
30                         if i == N_sinais-1:
31                             key = keys[N_reset-1]
32                             code = percursos.get(key)[N_sinais-1]
33                             if cnt_1 == N_sinais-1:
34                                 cnt_1 += 1
35                                 if code == 1: # Curva
36                                     if sinal_detected == True:
37                                         seguir_em_frente = False
38                                         subcode = 'Curva'
39                                 else: # Frente
40                                     code = 0
41                                     subcode = 'Frente'
42                             else:
43                                 code = 0
44                                 subcode = 'Frente'
45                         else:
46                             break
47                     elif N_sinais == 0:
48                         cnt_1 = 0
49                         key = keys[N_reset-1]
50                         code = 0
51                         subcode = 'Frente'
52                 elif seguir_em_frente is not True:
53                     cnt_2 += 1
54                     key = keys[N_reset - 1]
55                     Ix = tools.direcao_virar(roi)[0]
56                     code = 1
57                     subcode = 'Curva'
58                     if cnt_2 == 180:
59                         cnt_2 = 0
60                         seguir_em_frente = True
61
62                 elif N_reset == 0: # Ate nao encontrar o reset segue em frente
63
64                     imagem_percurso = tools.direcao_frente(roi)[2]
65                     Ix = tools.direcao_frente(roi)[0]
66                     key, code, subcode = None, None, None
67

```

```

68
69
70     Cx = 320
71     if Ix == None:
72         agv.stop()
73
74     else:
75         cv2.imshow('imagem', imagem_percurso)
76         if (Ix < Cx - 35):
77             agv.segue_esquerda(24,24)
78         elif (Ix > Cx - 35 and Ix < Cx + 35):
79             agv.segue_frente(23,23)
80
81         else:
82             agv.segue_direita(24,24)
83
84     if cv2.waitKey(1) & 0xFF == ord('s'):
85         break
86
87 camera.release()
88 cv2.destroyAllWindows()
89
90
91
92 ''' __ toos.py __ '''
93
94 import cv2
95 import numpy as np
96 from o_motores import Motor
97
98 class Threshold:
99     def black_mask (self, roi): # mascara preta
100
101         self.hsv = cv2.cvtColor(roi, cv2.COLOR_BGR2HSV)
102         self.lower_black = np.array([0, 0, 0]) # limite inferior
103         self.upper_black = np.array([70, 70, 70]) # limite superior
104         self.black_mask = cv2.inRange(roi, self.lower_black, self.upper_black)
105
106         return self.black_mask
107
108     def blue_mask (self, roi): # mascara azul
109         self.hsv = cv2.cvtColor(roi, cv2.COLOR_BGR2HSV)
110         self.lower_blue = np.array([100, 150, 150])
111         self.upper_blue = np.array([130, 255, 255])
112         self.blue_mask = cv2.inRange(self.hsv, self.lower_blue, self.upper_blue)
113
114         return self.blue_mask
115
116
117 def tratamento (roi):
118
119     mascara_azul, mascara_preta = Threshold().black_mask(roi),
120     Threshold().blue_mask(roi)
121     mascaras = [mascara_azul, mascara_preta]
122     mascaras_tratadas = []
123     kernel = np.ones((3,3),np.uint8) # elemento estrutural
124     for i in mascaras:
125         imagem_a_tratar = cv2.dilate (i, kernel, iterations = 4) # dilatacao
126         imagem_tratada = cv2.erode (imagem_a_tratar, kernel, iterations = 4) # erosao
127         mascaras_tratadas.append(imagem_tratada)
128
129     imagem_azul = cv2.bitwise_and (roi, roi, mask = mascaras_tratadas[0])
130     imagem_preta = cv2.bitwise_and (roi, roi, mask = mascaras_tratadas[1])
131     imagem = cv2.bitwise_or (imagem_azul, imagem_preta)
132
133     return mascaras_tratadas[0], mascaras_tratadas[1], imagem

```

```

134
135
136 def contornos (roi):
137
138     mascara_preta, mascara_azul, imagem = tratamento(roi)
139     contornos_a_avaluar = [mascara_preta, mascara_azul]
140     guia_detected, sinal_detected, reset = False, False, False
141     conjunto_contornos = []
142     for i in range(len(contornos_a_avaluar)):
143         contornos, _ = cv2.findContours(contornos_a_avaluar[i], cv2.RETR_EXTERNAL,
144                                         cv2.CHAIN_APPROX_NONE)
145         contornos = sorted(contornos, key = lambda x:cv2.contourArea(x),
146                             reverse=True)[:3] # os 3 maiores contornos pela area
147         if contornos != 0:
148             for j in contornos:
149                 area = cv2.contourArea(j)
150                 x,y,l,a = cv2.boundingRect(j)
151                 if i == 0: # mascara preta
152                     if area > 8000: #guia
153                         cv2.rectangle(imagem, (x,y), (x + l , y + a), (0, 0, 255),2)
154                         guia_detected = True
155
156                 elif 3800 < area < 5500: # reset
157                     if guia_detected == True:
158                         peri = cv2.arcLength(j, True) # calcula o perimetro
159                         aproximacao = cv2.approxPolyDP(j, 0.1*peri, True) #
160                         calcula a dimensao das arestas como vetores
161                         if len(contornos) == 3 and len(aproximacao) == 4: # se
162                             existirem 3 contornos + contornos com 4 lados
163                             cv2.rectangle(imagem, (x,y), (x + l , y + a), (0, 255,
164                                     255),2)
165                             reset = True
166                             del (contornos[3:])
167
168                 else: # mascara azul
169                     if guia_detected == True and area > 3800: # sinais
170                         peri = cv2.arcLength(j, True)
171                         aproximacao = cv2.approxPolyDP(j, 0.02*peri, True)
172                         if len(aproximacao) == 4:
173                             cv2.rectangle(imagem, (x,y), (x + l , y + a), (0, 255, 0),2)
174                             sinal_detected = True
175
176             else: # sem guia
177                 guia_detected, sinal_detected, reset = False, False, False
178                 conjunto_contornos.append(contornos) # contornos pretos + contornos azuis
179
180     return guia_detected, sinal_detected, reset, conjunto_contornos, imagem
181
182
183 N_sinais, N_reset = 0, 0
184 lista_sinais, lista_reset, percurso = [], [], []
185
186 def contador(roi):
187
188     global N_sinais, N_reset
189     global lista_sinais, lista_reset, percurso
190     VERIFICACAO = 15
191     guia_detected, sinal_detected, reset = contornos(roi)[0:3]
192     if guia_detected == True: # verificar a existencia de um sinal de reset
193         if reset == True:
194             lista_reset.append(reset)
195             if lista_reset.count(True) == 2: # N. de vezes que o reset e contado por
196                 sinal
197                 N_reset += 1 # Confirma-se que e um sinal de reset
198                 print('RESET n.: ' + str(N_reset))
199                 N_sinais = 0 # Entao, o sinais contados ate aqui sao resetados
200                 percurso.clear()

```

```

195     elif reset is not True and len(lista_reset) != 0: #
196         lista_reset.append(False)
197         if lista_reset.count(False) >= VERIFICACAO:
198             lista_reset.clear()
199
200     else: # verificar a existencia de um sinal azul
201         if sinal_detected == True:
202             lista_sinais.append(guia_detected)
203             #if lista_sinais.count(True) == 10: #tentar com 5
204             if lista_sinais.count(True) == 2: #tentar com 5
205                 N_sinais += 1
206                 print('Sinal detetado : ' + str(N_sinais))
207                 percurso.append(N_sinais)
208             elif sinal_detected is not True and len(lista_sinais) != 0:
209                 lista_sinais.append(False)
210                 if lista_sinais.count(False) >= VERIFICACAO:
211                     lista_sinais.clear()
212 if N_reset == 4:
213     N_reset = 1 # N. Inalteravel
214
215     return N_sinais, N_reset, percurso
216
217
218
219 def direcao_frente (roi):
220
221     guia_detected, sinal_detected, reset, aux, imagem = contornos(roi)
222     Cx, Cy = centro(imagem)
223     contornos_pretos = []
224     Ix, Iy = None, None
225     if guia_detected == True:
226         for j in range(len(aux[0])):
227             area = cv2.contourArea(aux[0][j])
228             if area < 7800: # antes 6000
229                 del j
230             else:
231                 contornos_pretos.append(aux[0][j])
232     lista_ix = []
233     for i in range(len(contornos_pretos)):
234         area = cv2.contourArea(contornos_pretos[i])
235         if 7800 < area < 10000: # guia normal / guia unica a expandir
236             (bifurcacao) # Antes 6000 a 8500
237             if len(contornos_pretos) == 1:
238                 Ix, Iy = centroide(contornos_pretos[0])
239             elif len(contornos_pretos) == 2:
240                 Ix, Iy = centroide(contornos_pretos[i])
241                 lista_ix.append(Ix)
242                 if i == 1 and len(lista_ix) == 2:
243                     if abs(lista_ix[0] - Cx) >= abs(lista_ix[1] - Cx):
244                         Ix, Iy = lista_ix[1], Iy
245                     else:
246                         Ix, Iy = lista_ix[0], Iy
247             else:
248                 del contornos_pretos[2:] # Nao existem mudancas de direcao
249                 simultaneas
250
251     elif 10000 < area < 20000: # ANTES 8500 A 18000
252         if len(contornos_pretos) == 1:
253             Ix, Iy = centroide(contornos_pretos[0])
254             diff = Cx - Ix
255             x,y,l,a = cv2.boundingRect(contornos_pretos[0])
256             if diff > 0: #print('curva da esquerda')
257                 Ix = x + l - 75
258                 Iy = (y + a)//2
259             else: #print('curva da direita')
260                 Ix = x + 75
261                 Iy = (y + a)//2

```

```

260         elif area > 20000:
261             Ix, Iy = centroide(contornos_pretos[0])
262
263         if Ix is not None and Iy is not None :
264             cv2.circle(imagem, (Ix, Iy), 3, (0, 255, 0), -1)
265             cv2.circle(imagem, (Cx, Cy), 3, (255, 255, 255), -1)
266     else:
267         Ix = None
268         Iy = None
269
270     return Ix, Iy, imagem
271
272
273
274 ix_global = 0
275
276 def direcao_virar (roi):
277
278     global ix_global
279
280     guia_detected, sinal_detected, reset, aux, imagem = contornos(roi)
281     Cx, Cy = centro(imagem)
282     contornos_pretos = []
283     Ix, Iy = None, None
284     if guia_detected == True:
285         for j in range(len(aux[0])):
286             area = cv2.contourArea(aux[0][j])
287             if area < 7800:
288                 del j
289             else:
290                 contornos_pretos.append(aux[0][j])
291     lista_ix = []
292     for i in range(len(contornos_pretos)):
293         area = cv2.contourArea(contornos_pretos[i])
294         if 7800 < area < 10000:
295             if len(contornos_pretos) == 1:
296                 Ix, Iy = centroide(contornos_pretos[0])
297                 if sinal_detected == True:
298                     contornos_azuis = aux[1]
299                     for k in contornos_azuis:
300                         area = cv2.contourArea(k)
301                         if area > 4000:
302                             Ix_sinal, Iy_sinal = centroide(k)
303                             ix_global = Ix_sinal
304                             print('Ix sinal = ' + str(ix_global))
305
306             elif len(contornos_pretos) == 2:
307                 Ix, Iy = centroide(contornos_pretos[i])
308                 lista_ix.append(Ix)
309                 if i == 1 and len(lista_ix) == 2:
310                     if abs(lista_ix[0] - Cx) >= abs(lista_ix[1] - Cx):
311                         Ix, Iy = lista_ix[1], Iy
312                 else:
313                     Ix, Iy = lista_ix[0], Iy
314             else:
315                 del contornos_pretos[2:]
316
317         elif 10000 < area < 20000:
318             if len(contornos_pretos) == 1:
319                 Ix, Iy = centroide(contornos_pretos[0])
320                 x,y,l,a = cv2.boundingRect(contornos_pretos[0])
321                 if ix_global > Ix:
322                     Ix = x + l - 75
323                     Iy = (y + a)//2
324                 else:
325                     Ix = x + 75
326                     Iy = (y + a)//2

```

```

327
328         elif area > 20000:
329             Ix, Iy = centroide(contornos_pretos[0])
330
331         if Ix is not None and Iy is not None:
332             cv2.circle(imagem, (Ix, Iy), 3, (0, 255, 0), -1)
333             cv2.circle(imagem, (Cx, Cy), 3, (255, 255, 255), -1)
334     else:
335         Ix = None
336         Iy = None
337
338     return Ix, Iy, imagem
339
340
341
342 def centro (roi):
343
344     altura, largura = roi.shape[:2]
345     Cx = int(largura/2)
346     Cy = int(altura/2)
347
348     return Cx, Cy
349
350
351 def centroide (contorno):
352
353     M = cv2.moments(contorno)
354     if M["m00"] == 0:
355         return 0
356
357     Ix = int(M["m10"] / M["m00"])
358     Iy = int(M["m01"] / M["m00"])
359
360     return Ix, Iy
361
362
363 def diferenca (a, b, erro):
364
365     if abs(a - b) < erro:
366         return True
367     else:
368         return False
369
370
371
372 ''' __ motores.py __ '''
373
374 import RPi.GPIO as GPIO
375 from time import sleep
376 GPIO.setmode (GPIO.BCM)
377 GPIO.setwarnings (False)
378
379
380 class Motor:
381
382     def __init__ (self, ENABLE_A, IN1_A, IN2_A, ENABLE_B, IN1_B, IN2_B):
383         self.ENABLE_A = ENABLE_A
384         self.IN1_A = IN1_A
385         self.IN2_A = IN2_A
386         self.ENABLE_B = ENABLE_B
387         self.IN1_B = IN1_B
388         self.IN2_B = IN2_B
389
390         GPIO.setup(self.ENABLE_A, GPIO.OUT)
391         GPIO.setup(self.IN1_A, GPIO.OUT)
392         GPIO.setup(self.IN2_A, GPIO.OUT)
393         GPIO.setup(self.ENABLE_B, GPIO.OUT)

```

```

394     GPIO.setup(self.IN1_B, GPIO.OUT)
395     GPIO.setup(self.IN2_B, GPIO.OUT)
396
397     self.PWM_A = GPIO.PWM(self.ENABLE_A, 100)
398     self.PWM_B = GPIO.PWM(self.ENABLE_B, 100)
399     self.PWM_A.start(0)
400     self.PWM_B.start(0)
401
402
403     def segue_frente (self, speed_A, speed_B):
404
405         self.PWM_A.ChangeDutyCycle (speed_A)
406         self.PWM_B.ChangeDutyCycle (speed_B)
407
408         GPIO.output(self.IN1_A, GPIO.HIGH)
409         GPIO.output(self.IN2_A, GPIO.LOW)
410         GPIO.output(self.IN1_B, GPIO.HIGH)
411         GPIO.output(self.IN2_B, GPIO.LOW)
412         #print('frente')
413
414
415
416     def segue_esquerda (self, speed_A, speed_B):
417
418         self.PWM_A.ChangeDutyCycle (speed_A)
419         self.PWM_B.ChangeDutyCycle (speed_B)
420
421         GPIO.output(self.IN1_A, GPIO.HIGH)
422         GPIO.output(self.IN2_A, GPIO.LOW)
423         GPIO.output(self.IN1_B, GPIO.LOW)
424         GPIO.output(self.IN2_B, GPIO.HIGH)
425         #print('vou virar para a ESQUERDA')
426
427
428
429     def segue_direita (self, speed_A, speed_B):
430
431         self.PWM_A.ChangeDutyCycle (speed_A)
432         self.PWM_B.ChangeDutyCycle (speed_B)
433
434         GPIO.output(self.IN1_A, GPIO.LOW)
435         GPIO.output(self.IN2_A, GPIO.HIGH)
436         GPIO.output(self.IN1_B, GPIO.HIGH)
437         GPIO.output(self.IN2_B, GPIO.LOW)
438         #print('vou virar para a DIREITA')
439
440
441
442     def stop(self):
443
444         self.PWM_A.ChangeDutyCycle (0)
445         self.PWM_B.ChangeDutyCycle (0)
446
447
448
449
450     ''' __ percursos.py __ '''
451
452
453     percursos = {
454         'P1' : [0, 0],
455         'P2' : [1, 0, 1],
456         'P3' : [1, 1, 1]
457
458         # 0 --> Segue em frente
459         # 1 --> Faz uma curva
460     }

```

Referências

- [1] J. do ICT page 100, “Desenvolvimento de veículo autoguiado em raspberry pi,” https://www.icterra.pt/wp-content/uploads/2022/02/Livro_Resumos_Jornadas_ICT_2022-1.pdf.
- [2] G. Ullrich *et al.*, “Automated guided vehicle systems,” *Springer-Verlag Berlin Heidelberg. doi*, vol. 10, pp. 978–3, 2015.
- [3] F. J. L. Grilo, “Modelo de processamento de imagem, com múltiplas fontes de aquisição, para manipulação aplicada à domótica,” 2019.
- [4] K. Bijanrostami, “Design and development of an automated guided vehicle for educational purposes,” Ph.D. dissertation, Eastern Mediterranean University (EMU), 2011.
- [5] E. Serrano Martínez, “Preliminary design of an automatic guided vehicle (agv) system,” Ph.D. dissertation, Universitat Politècnica de València, 2020.
- [6] T. Le-Anh and M. De Koster, “A review of design and control of automated guided vehicle systems,” *European Journal of Operational Research*, vol. 171, no. 1, pp. 1–23, 2006.
- [7] T. Al-Hawari, A.-Z. Ena’am S, and H. Alshraideh, “Studying the effect of facility size on the selection of automated guided vehicle flow configurations,” *Journal of Automation and Control Engineering Vol*, vol. 4, no. 2, 2016.
- [8] HMI, “Agv 101,” 2015. [Online]. Available: <https://www.mhi.org/mag/knowledge>
- [9] J. Borenstein, H. Everett, L. Feng *et al.*, “Where am i? sensors and methods for mobile robot positioning,” *University of Michigan*, vol. 119, no. 120, p. 27, 1996.
- [10] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.
- [11] J. Qian, B. Zi, D. Wang, Y. Ma, and D. Zhang, “The design and development of an omni-directional mobile robot oriented to an intelligent manufacturing system,” *Sensors*, vol. 17, no. 9, p. 2073, 2017.
- [12] “Diesel fuel,” Jan 2022. [Online]. Available: https://en.wikipedia.org/wiki/Diesel_fuel
- [13] R. Prasad and V. R. Bella, “A review on diesel soot emission, its effect and control,” *Bulletin of Chemical Reaction Engineering & Catalysis*, vol. 5, no. 2, p. 69, 2010.
- [14] A. SHARMA, P. ZANOTTI, and L. P. MUSUNUR, “Robotic automation for electric vehicle battery assembly: Digital factory design and simulation for the electric future of mobility.”
- [15] M. Ehsani, Y. Gao, S. Longo, and K. M. Ebrahimi, *Modern electric, hybrid electric, and fuel cell vehicles*. CRC press, 2018.
- [16] J. E. R. de Queiroz and H. M. Gomes, “Introdução ao processamento digital de imagens.” *Rita*, vol. 13, no. 2, pp. 11–42, 2006.
- [17] “Computer vision,” Jan 2022. [Online]. Available: https://en.wikipedia.org/wiki/Computer_vision

- [18] E. R. Davies, *Computer and machine vision: theory, algorithms, practicalities*. Academic Press, 2012.
- [19] R. C. Gonzalez, R. E. Woods, and B. R. Masters, "Digital image processing," 2009.
- [20] M. Marengoni and S. Stringhini, "Tutorial: Introdução à visão computacional usando opencv," *Revista de Informática Teórica e Aplicada*, vol. 16, no. 1, pp. 125–160, 2009.
- [21] A. Alamäki, "Implementation of opencv in a machine vision system," 2017.
- [22] "Opencv: Image processing," Jan 2022. [Online]. Available: https://docs.opencv.org/3.4.0/d2/df0/tutorial_js_table_of_contents_imgproc.html
- [23] L. Shapiro, "Computer vision/linda g," *Shapiro, George C. Stockman–Pearson*, 2001.
- [24] O. Marques Filho and H. V. Neto, *Processamento digital de imagens*. Brasport, 1999.
- [25] "Image segmentation," Jan 2022. [Online]. Available: https://en.wikipedia.org/wiki/Image_segmentation
- [26] G. Kumar and P. K. Bhatia, "A detailed review of feature extraction in image processing systems," in *2014 Fourth International Conference on Advanced Computing Communication Technologies*, 2014, pp. 5–12.
- [27] M. P. de Albuquerque and M. P. de Albuquerque, "Processamento de imagens: métodos e análises," *Rio de Janeiro, Brasil*, vol. 12, 2000.
- [28] D. J. R. Carvalho *et al.*, "Processamento de imagem num simulador de armazenamento automático," Ph.D. dissertation, Instituto Superior de Engenharia de Lisboa, 2016.
- [29] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 4, pp. 349–361, 2001.
- [30] P. Rio, "Processamento e análise digital de imagens," https://www.maxwell.vrac.puc-rio.br/21365/21365_6.PDF.
- [31] A. L. B. N. d. Cunha, "Sistema automático para obtenção de parâmetros do tráfego veicular a partir de imagens de vídeo usando opencv," Ph.D. dissertation, Universidade de São Paulo, 2013.
- [32] "About," Nov 2021. [Online]. Available: <https://opencv.org/about/>
- [33] I. Corporation, "Open source computer vision library," <http://www.cs.unc.edu/Research/stc/FAQs/OpenCV/OpenCVReferenceManual.pdf>.
- [34] M. J. A. Pereira, "As tecnologias no design de produto," Ph.D. dissertation, UNIVERSIDADE TÉCNICA DE LISBOA, 2010.
- [35] F. Robótico, "Arquitecturas de sistemas robóticos e localização em tempo real através de visão," Ph.D. dissertation, Universidade do Porto, 2003.
- [36] H. Lipson and M. Kurman, *Fabricated: The new world of 3D printing*. John Wiley & Sons, 2013.
- [37] B. Redwood, F. Schöffner, and B. Garret, *The 3D printing handbook: technologies, design and applications*. 3D Hubs, 2017.
- [38] O. Olabisi and K. Adewale, *Handbook of thermoplastics*. CRC press, 2016, vol. 41.
- [39] F. B.-T. C. PLA and J. L.-T. C. PLA, "End-of-life options for bioplastics," 2020.
- [40] S. Farah, D. G. Anderson, and R. Langer, "Physical and mechanical properties of pla, and their functions in widespread applications—a comprehensive review," *Advanced drug delivery reviews*, vol. 107, pp. 367–392, 2016.

- [41] A. Hughes and B. Drury, *Electric motors and drives: fundamentals, types and applications*. Newnes, 2019.
- [42] J. L. Jones, B. A. Seiger, and A. M. Flynn, *Mobile robots: Inspiration to implementation*. CRC Press, 1998.
- [43] A. Francisco, “Motores elétricos, 5ª edição,” *ETEP-Edições Técnicas e Profissionais*, 2013.
- [44] J. Wen, Y. Yu, and C. Chen, “A review on lithium-ion batteries safety issues: existing problems and possible solutions,” *Materials express*, vol. 2, no. 3, pp. 197–212, 2012.
- [45] Samsung, “Specification of product for lithium-ion rechargeable cell,” <https://datasheetspdf.com/pdf-file/825549/ETC/ICR18650-22E/1.pdf>.
- [46] J. M. B. Marques, “Battery management systems (bms) for li-ion batteries,” Ph.D. dissertation, Universidade de Coimbra, 2014.
- [47] L. Lu, X. Han, J. Li, J. Hua, and M. Ouyang, “A review on the key issues for lithium-ion battery management in electric vehicles,” *Journal of power sources*, vol. 226, pp. 272–288, 2013.
- [48] D. Andrea, *Battery management systems for large lithium-ion battery packs*. Artech house, 2010.
- [49] T. Instruments, “Lm317 3-terminal adjustable regulator,” <https://www.ti.com/lit/ds/symlink/lm317.pdf?ts=1655256541414>.
- [50] H. Wang and F. Blaabjerg, “Reliability of capacitors for dc-link applications in power electronic converters—an overview,” *IEEE Transactions on Industry Applications*, vol. 50, no. 5, pp. 3569–3578, 2014.
- [51] S. Lee, “Optimum design and selection of heat sinks,” *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part A*, vol. 18, no. 4, pp. 812–817, 1995.
- [52] H. Shabgard, M. J. Allen, N. Sharifi, S. P. Benn, A. Faghri, and T. L. Bergman, “Heat pipe heat exchangers and heat sinks: Opportunities, challenges, applications, analysis, and state of the art,” *International Journal of Heat and Mass Transfer*, vol. 89, pp. 138–158, 2015.
- [53] M. I. Montrose, *Printed circuit board design techniques for EMC compliance*. IEEE press Piscataway, NJ, 1996, vol. 1.
- [54] G. Humpston and D. M. Jacobson, *Principles of soldering*. ASM international, 2004.
- [55] L. Leydesdorff, D. Kushnir, and I. Rafols, “Interactive overlay maps for us patent (uspto) data based on international patent classification (ipc),” *Scientometrics*, vol. 98, no. 3, pp. 1583–1599, 2014.
- [56] J. Ejury, “Buck converter design,” *Infineon Technologies North America (TFNA) Core Design Note*, vol. 1, 2013.
- [57] T. Instruments, “Lm2596 simple switcher,” https://www.ti.com/lit/ds/symlink/lm2596.pdf?ts=1650296244931&ref_url=https%253A%252F%252Fwww.google.com%252F.pdf.
- [58] XLSEMI, “XL4015 pwm buck step-down,” https://datasheet.lcsc.com/lcsc/1811081616_XLSEMI-XL4015E1--_C51661.pdf.
- [59] “Raspberry pi 4B raspberry pi,” <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>.
- [60] R. Pi, “Raspberry pi 4 tech specs,” <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf>.
- [61] “Introducing the Raspberry Pi Cameras raspberry pi,” <https://www.raspberrypi.com/documentation/accessories/cameras/>.
- [62] Sparkfun, “Raspberry pi camera v2,” <https://cdn.sparkfun.com/datasheets/Dev/RaspberryPi/RPiCamMod2.pdf>.

- [63] M. A. Pagnutti, R. E. Ryan, M. J. Gold, R. Harlan, E. Leggett, J. F. Pagnutti *et al.*, "Laying the foundation to use raspberry pi 3 v2 camera module imagery for scientific and engineering purposes," *Journal of Electronic Imaging*, vol. 26, no. 1, p. 013014, 2017.
- [64] STMicroelectronics, "Dual full-bridge driver l298," <https://pdf1.alldatasheet.com/datasheet-pdf/view/22440/STMICROELECTRONICS/L298N.html.pdf>.
- [65] L. Petru and G. Mazen, "Pwm control of a dc motor used to drive a conveyor belt," *Procedia Engineering*, vol. 100, pp. 299–304, 2015.
- [66] M. Hagiwara, K. Nishimura, and H. Akagi, "A medium-voltage motor drive with a modular multilevel pwm inverter," *IEEE Transactions on Power Electronics*, vol. 25, no. 7, pp. 1786–1799, 2010.
- [67] A. Kurniawan and Kurniawan, *Raspbian OS Programming with the Raspberry Pi*. Springer, 2019.
- [68] M. Schmidt, *Raspberry Pi: a quick-start guide*. Pragmatic Bookshelf, 2014.
- [69] P. Francis-Mezger and V. M. Weaver, "A raspberry pi operating system for exploring advanced memory system concepts," in *Proceedings of the International Symposium on Memory Systems*, 2018, pp. 354–364.
- [70] "Raspberry Pi SD Card Speed raspberry pi," <https://www.raspberrypi.com/news/sd-card-speed-test/>, acedido: 2022-03-5.
- [71] "Operating system images raspberry pi," <https://www.raspberrypi.com/software/operating-systems/>, acedido: 2022-03-5.
- [72] Y. Lu, "Cyber physical system (cps)-based industry 4.0: A survey," *Journal of Industrial Integration and Management*, vol. 2, no. 03, p. 1750014, 2017.
- [73] J. Edney, W. A. Arbaugh, and W. Arbaugh, *Real 802.11 security: Wi-Fi protected access and 802.11 i*. Addison-Wesley Professional, 2004.
- [74] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper, "Virtual network computing," *IEEE Internet Computing*, vol. 2, no. 1, pp. 33–38, 1998.
- [75] T. Richardson and K. R. Wood, "The rfb protocol," *ORL, Cambridge, January*, 1998.
- [76] A. Stolerman, "Rfc 6143: The remote framebuffer (rfb) protocol analysis," 2013.
- [77] "Virtual Network Computing vnc," <http://www.123seminaronly.com/Seminar-Reports/024/49392709-VirtualNetworkComputing.pdf>, acedido: 2022-03-7.
- [78] R. X. P. d. Santos, "Password habits and cracking toolkit," Ph.D. dissertation, 2015.
- [79] D. J. Barrett, D. J. Barrett, R. E. Silverman, and R. Silverman, *SSH, the Secure Shell: the definitive guide*. "O'Reilly Media, Inc.", 2001.
- [80] G. Howser, *Computer Networks and the Internet*. Springer, 2020.
- [81] "Python," <https://www.python.org>, accessed: 2022-04-7.
- [82] K. Srinath, "Python—the fastest growing programming language," *International Research Journal of Engineering and Technology*, vol. 4, no. 12, pp. 354–357, 2017.
- [83] A. S. Saabith, M. Fareez, and T. Vinothraj, "Python current trend applications-an overview," *International Journal of Advance Engineering and Research Development*, vol. 6, no. 10, 2019.
- [84] "Visual Studio Code," <https://code.visualstudio.com/docs/setup/setup-overview>, accessed: 2022-04-7.

- [85] R. Brinkmann, *The art and science of digital compositing: Techniques for visual effects, animation and motion graphics*. Morgan Kaufmann, 2008.
- [86] R. W. G. Hunt, *The reproduction of colour*. Wiley Online Library, 1995, vol. 4.
- [87] C. Poynton, *Digital video and HD: Algorithms and Interfaces*. Elsevier, 2012.
- [88] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the hsv color space for image retrieval," in *Proceedings. International Conference on Image Processing*, vol. 2. IEEE, 2002, pp. II–II.
- [89] "Color conversions in OpenCV," https://docs.opencv.org/4.x/de/d25/imgproc_color_conversions.html#color_convert_rgb_luv, accessed: 2022-04-12.
- [90] S. Cochrane, "The munsell color system: A scientific compromise from the world of art," *Studies in History and Philosophy of Science Part A*, vol. 47, pp. 26–41, 2014.
- [91] A. Ford and A. Roberts, "Colour space conversions," *Westminster University, London*, vol. 1998, pp. 1–31, 1998.
- [92] Q. Chen, Q.-s. Sun, P. A. Heng, and D.-s. Xia, "A double-threshold image binarization method based on edge detector," *Pattern recognition*, vol. 41, no. 4, pp. 1254–1267, 2008.
- [93] L. Vincent, "Morphological transformations of binary images with arbitrary structuring elements," *Signal processing*, vol. 22, no. 1, pp. 3–23, 1991.
- [94] X. Zhuang and R. M. Haralick, "Morphological structuring element decomposition," *Computer vision, graphics, and image processing*, vol. 35, no. 3, pp. 370–382, 1986.
- [95] P. Soille *et al.*, *Morphological image analysis: principles and applications*. Springer, 1999, vol. 2, no. 3.
- [96] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2010.
- [97] "Contours in Image Processing," https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html, accessed: 2022-04-20.
- [98] S. Suzuki *et al.*, "Topological structural analysis of digitized binary images by border following," *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.
- [99] C. V. R. Coutinho, "Robótica móvel: sistema de condução autónoma," Ph.D. dissertation, Instituto Superior de Engenharia de Lisboa, 2014.
- [100] "Green's theorem," https://en.m.wikipedia.org/wiki/Green%27s_theorem, accessed: 2022-04-11.
- [101] F. Minhós, "Análise matemática iii," 2011.
- [102] J. Stewart, *Essential calculus: Early transcendentals*. Cengage Learning, 2012.
- [103] G. Y. Tang, "A discrete version of green's theorem," *IEEE transactions on pattern analysis and machine intelligence*, no. 3, pp. 242–249, 1982.
- [104] J. E. Hershberger and J. Snoeyink, "Speeding up the douglas-peucker line-simplification algorithm," 1992.
- [105] "Centroid," <https://en.wikipedia.org/wiki/Centroid>, accessed: 2022-04-27.
- [106] P. Bourke, "Calculating the area and centroid of a polygon," *Swinburne Univ. of Technology*, vol. 7, 1988.

- [107] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE transactions on information theory*, vol. 8, no. 2, pp. 179–187, 1962.
- [108] "Structural Analysis and Shape Descriptors," https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html, accessed: 2022-04-30.