

Fundamentos da Programação

MAR

1.03

11 de abril de 2025



Projecto



Front page image - www

Disclaimer

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License". This document is provided as-is with no warranty, implied or otherwise. There have been attempts to eliminate errors from this document, but there is no doubt that errors remain. As a result, the author does not assume any responsibility for errors and omissions, or damages resulting from the use of the information provided.

Texto composto em L^AT_EX por myself

MAR2025

Parte I

Projecto 2025

Parte 1

Introdução

O projecto de *Fundamentos de Programação* destina-se a implementar uma versão simplificada do sistema de navegação de um empregado de restaurante (*Restaurant waiter*). Este desloca-se, de forma omnidirecional (ver anexo 10), numa sala definida, a sala do restaurante. A programação será incremental, começando com um projeto simples, indo incrementando de complexidade e realismo. Apresentam-se exemplos (mas não modelos) de sistemas semelhantes.

Embora o exercício proposto seja de programação, apresentam-se em anexos os exemplos reais da implementação proposta, que, obviamente não fazem parte do trabalho.

Correções / clarificações ao presente texto serão afixadas na página da disciplina.



Parte 2

Requisitos do programa

O programa deverá ser escrito, programado, e codificado **pelos alunos dos grupo, e utilizar** a biblioteca¹ **graphics** disponibilizada² pelos autores do livro: Python Programming: An Introduction to Computer Science, 3rd Ed., John M. Zelle, Franklin, Beedle & Associates, 2017). Toda a interação deverá ser efectuada por via de interfaces gráficas. Deverá ser criada um ponto de partida, onde será dada a ordem de execução. Em cada um dos níveis o programa deverá sempre sair de modo correcto e controlado, (ie, sem recorrer a consola, Ctrl-C ou outro comando de interrupção forçada).

O programa deve ter um dropdown contendo nome da disciplina, autores e ano lectivo.

2.1 Partida - Tier 0

A planta do restaurante é fornecida em anexo.

São indicadas as dimensões relativas. A planta é fixa, pelo que devrá ser

¹A utilização de outra biblioteca será penalizada

²<https://mcsp.wartburg.edu/zelle/python/graphics.py>, e <https://mcsp.wartburg.edu/zelle/python/graphics/graphics.pdf>

efectuada de inicio, *a partir de um ficheiro de texto (salaxx.txt)*. Pode ser alterada em termos visuais, mas não em termos de disposição.

A implementação deverá conter uma classe mesas, chamada *Table*, que poderá utilizar uma das classes *Retangulo*³. A área de navegação deverá ser retangular e deverá ter uma *docking station* encostada ao bordo e, adjacente, a recolha de pratos. Existe um ponto de entrega ao robot de pratos, num local fixo da sala. Neste é indicado o local de entrega (referência da mesa)

A implementação é independente do tamanho da janela no ecran, e efectuada em coordenadas relativas.⁴

A sala é apresentada na planta em anexo, escalável, consistindo de duas $(2n)$ filas de $3(m)+3(m)$ mesas separadas por uma divisoria inamovível (ver secção 2.5). Considera-se $n=m=1$. A posição das mesas é indicada por um ficheiro com formatação conhecida, e deve ser lido pelo programa. Embora a implementação e teste seja efectuada com o exemplo apresentado, o programa deve ser efectuado de forma que possa ser facilmente adaptado.

O programa deve:

- Criar a(s) janela(s) para a sala
- Desenhar os elementos, objectos, conforme requerido para as implementações seguintes.

2.2 Implementação básica (tier 1, nível básico)

- Navegação

O nível básico corresponde ao movimento do robot.

³ou não são por enquanto consideradas outras formas, e.g. Círculo ou eventualmente outra, nesta fase do trabalho

⁴Ver por exemplo:<https://dawn2021.orylab.com/>

A classe a criar para representar o robot deverá ter o nome ***Waiter***. Esta classe deverá ter todos os métodos que se considerarem necessários e deverá utilizar todas as classes que se considerarem necessárias. Note-se que o robot choca com os obstáculos, ou seja, não pode sobrepor-se a esses objetos. A área de navegação, o robot e os obstáculos deverão ter tamanhos representativos do cenário proposto.

O robot **deve**:

- O robot deve-se deslocar até cada uma das mesas.
- parar
- Retornar.

O robot **pode**:

- Ter indicação visual (ponto colorido) da tarefa a ser executada.
- Outras indicações visuais.

2.3 Implementação intermédia (tier 2, Segundo nível-entrega)

O robot destina-se primariamente a entregar o pedido nas mesas. O utilizador poderá clicar em qualquer das mesas onde o pedido será solicitado. O robot deve sair da sua docking station, ir e voltar a mesa, e após o tempo de confecção (tempo de confecção: e.g. 2 seg) dirigir-se a esse ponto, encostar-se a mesa, entregá-lo na mesa(para 2 seg), após o que regressará ao ponto de inicio.

Quando existir um obstáculo no caminho, o robot tem que parar para o evitar até ser removido. A remoção pode ser automática (e.g 3 seg após ser detectado),⁵. Os obstáculos deverão ser indicados por meio de um click de rato no local respectivo, modelando uma pessoa ou objecto que seja inadvertidamente colocado e removido.

O robot **deve, após o nível 2:**

- Receber uma indicação da mesa,
- Dirigir-se a mesa respectiva, verificando a não existência de obstáculos,
- Voltar ao ponto de partida
- Fazer a entrega na mesa
- Voltar ao ponto de partida

2.3.1 Implementação - notas

Robot

O robot deve ser implementado numa classe muito semelhante à classe Projectil dada nas aulas. Um click do rato numa mesa representa um pedido e deverá ser capaz de se dirigir em qualquer direcção. Os trajectos podem ser predefinidos, verificando se o espaço a frente está disponível.

Bateria

O robot pode ter uma indicação da carga da bateria. Esta deve ser carregada após percorrer duas vezes a área de navegação, ou quando estiver em espera. Poderá de ser efetuada a gestão da bateria de forma a fazer uma pausa

⁵eventualmente, por exemplo, contornar não é considerado nesta fase do trabalho, mas pode ser considerada

no serviço para voltar à docking station e carregar a bateria (a bateria é de carregamento super-rápido, demorando apenas 2 segundos). O robot deve ter uma luz indicadora da carga atual que muda de cor quando está a carregar. Poderá ainda mudar de cor quando a bateria se estiver a esgotar. A posição do carregamento do robot é a docking station.

2.4 Implementação nível superior (tier 3)

No terceiro nível, podem ser efectuadas várias entregas em simultâneo, minimizando o trajecto e tempo efectuado, isto é, por ordem lógica de entrega. **Não são recebidos novos pedidos enquanto o robot se desloca.** Do mesmo modo a entrega e a recolha são items diferentes.

Neste nível poder-se-á ter em conta o nível da bateria.

2.5 Sala - formato da descrição

O ficheiro deverá ter o formato (o tamanho da janela e as localizações são apenas exemplificativos):

Tamanho da janela de simulacao exemplificativa: 800 600, correspondente a dimensões de 1600,1200. Localizacao dos objectos (exemplificativo e não contratual):

```
# Sala 0.1
# Grupo xx
Dock1 Rectangle(Point(0,0), Point(80,60))
Dock2 Rectangle(Point(1520,1140), Point(1600,1200))
Table1 Oval(Point(320,240), Point(120,840))
Table2 Rectangle(Point(1440,1080),Point(1600,600)) ...
```

Nota: Os valores podem ser dados pelo ponto central e dimensões, uniformizando a notação

Parte 3

Indicações gerais

O trabalho deverá ser feito em grupos de **dois** alunos, **original e projectado/codificado pelos próprios**. O recurso a ferramentas que efectuem a geração de programas (e.g. ChatGPt ou afim) é fortemente penalizada podendo levar a anulação do projecto. O trabalho deve ser efetuado em Python e executado em ambiente Spyder (3.xx) (Instalação de defeito do Anaconda em 2025). ¹

3.1 Entrega

O projeto deverá ser entregue até ao dia (Ver página da disciplina / 1^a página), 23h59m (hora do sistema Fénix), constando de:

- Todos os ficheiros do programa, (incluindo graphics.py) ² necessários em bom estilo de programação
 - comentários iniciais indicando o numero do grupo, nº e nome dos elementos.

¹A correção é efectuada com base no ambiente referido.

²A falta é penalizada / Ver malus

- data e hora da versão apresentada
 - comentários esclarecedores do código,
 - paragrafação que facilite a leitura e sugira a estruturação do programa
 - nomes adequados para as variáveis, classes, métodos e funções;
 - exemplos dos resultados da execução, mostrando claramente como o seu programa aborda todas as situações possíveis;
 - listagem dos ficheiros com os dados requeridos;
- Documentação, sucinta e completa

3.2 Documentação do trabalho (ver ponto 4)

- Todos os ficheiros do programa, (incluindo graphics.py) necessários em bom estilo de programação

Estes documentos deverão ser submetidos via Fénix num único ficheiro .zip ou .rar., com o nome

FP25-grxxx.rar

e onde xxx representa o numero do grupo. • todos os documentos bem como o código fonte deverão ser identificados com os nomes e números dos alunos que constituem o grupo.

No caso dos documentos em formato .pdf a identificação é feita na capa, e no footer das páginas. No caso do código fonte essa identificação será feita em comentários internos em todos os ficheiros.

3.2.1 Tolerância

Haverá uma tolerância de 2 dias nas entregas com respetiva PENALIZAÇÃO.

- Os trabalhos entregues entre as 0h00min e as 23h59min do dia seguinte ao prazo estipulado **SOFRERÃO UMA PENALIZAÇÃO DE 2,5 VALORES.**
- Os trabalhos entregues entre as 0h00min e as 23h59min do 2º dia seguinte ao prazo estipulado **SOFRERÃO UMA PENALIZAÇÃO DE 5,0 VALORES.**

Não serão aceites quaisquer trabalhos após o 2º dia.

Parte 4

Documentação

Todo o código fonte deverá incluir comentários apropriados, incluindo a identificação dos autores.

A documentação do trabalho será dividida em dois tipos: a documentação destinada aos utilizadores do programa e a documentação técnica, destinada aos programadores que irão fazer a manutenção e possíveis adaptações ao programa.

4.1 Generalidades

:

A documentação deverá ser formatada como se segue

- Capa, com a identificação do trabalho e dos elementos do grupo que realizaram o trabalho, data.
- Todas as folhas devem conter:
 - Header – com o nome do trabalho

- Footer, numero de paginas/ total de páginas no exterior, e numero de grupo com elementos no interior (Gr-aaaaa-bbbb-bccc).
- Texto em tamanho 12 justificado
- Índice remissivo.

4.2 Sumário executivo

(max 1 pagina) Indicando explicitamente:

- As alíneas resolvidas
- As alineas não funcionais

Indicações erradas ou induzindo em erro serão fortemente penalizadas.

4.3 Documentação para utilizadores.

Esta deverá conter:

1. a descrição do que o programa faz, incluindo a área geral de aplicação do programa e uma descrição do seu comportamento;
2. a descrição do processo de utilização do programa (é boa ideia incluir um ou mais exemplos); a descrição da informação necessária para o bom funcionamento do programa (explicando por exemplo qual o formato correto de entrada de dados);
3. a descrição da informação produzida pelo programa;
4. a descrição, em termos não técnicos, *das limitações do programa* (deverão ser dados exemplos de situações em que o programa poderá ter interrupções inesperadas).

4.4 Documentação técnica

Esta deverá conter:

- a descrição da estrutura do programa incluindo as classes e os principais subprogramas (funções) e interligação entre eles, e contendo a descrição do algoritmo do programa principal e de cada subprograma, utilizando o aspetto gráfico do Capítulo 9 do livro;
- a descrição das estruturas de informação (de dados e de classes), justificando a sua escolha, e a descrição dos principais objetos, incluindo variáveis e instâncias das classes. Repare que a documentação técnica não deve ser apenas uma listagem do programa acompanhada de comentários.

Parte 5

Critérios de avaliação

A nota do projeto será baseada nos seguintes aspectos:

- Execução correta 60% ($25\% + 25\% + 10\%$)
- Facilidade de leitura 10%
- Estilo de programação 10%
- Documentação de utilização 10%
- Documentação técnica 10%

No item Facilidade de leitura serão avaliados items como os nomes das variáveis, classes, comentários, paragrafação e tamanho das classes.

O Estilo de programação avalia a utilização de boas práticas de programação na elaboração do código fonte. São avaliados tópicos como:

- a existência de documentação interna ao código (cabeçalho, comentários explicativos ao longo do código, nomes de variáveis com significado, indentação do código para facilitar a leitura, etc.)

- modularidade (organização do código em classes de forma a facilitar a sua reutilização noutras programas, funções com argumentos de entrada e de saída de dados onde tal for adequado, etc.)

Funções ou métodos com um numero excessivo de linhas, dificultando a compreensão do programa, serão penalizados.



Parte 6

Bonus/Malus

PenalizaçõesMalus

Além das penalizações referidas serão atribuídas penalizações (max 10%, 2 val.) por inconformidade relativamente às normas referidas, em particular,

- Adaptações para a execução do programa ((-½) cada, max 2 tentativas/adaptações)
- Nomes dos ficheiros requeridos (-½)
- Pobreza dos gráficos (-½)
- Formatação inadequada(-½)
- Não identificação da páginas dos relatórios(-½)
- Indicações erradas no sumário executivo (2 val)

poderão adicionar penalizações .

O corpo docente é o único juiz da sua definição como tal.

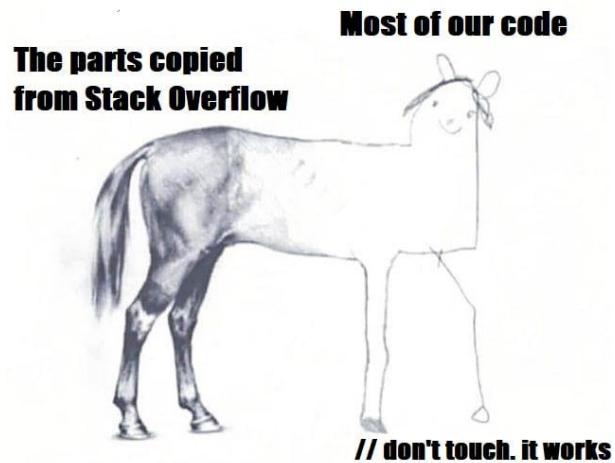
A correção é efectuada com base no compilador referido. A não conformidade ou adaptação necessária implica penalização correspondente da nota.

Bonificações / Bonus

:

Refinamentos do desempenho, dos gráficos ou do algoritmo usado poderão adicionar bonificações (até 10%, max 2.0 val no total, max 20 val no proj.).

O corpo docente é o único juiz da sua aceitação como tal.



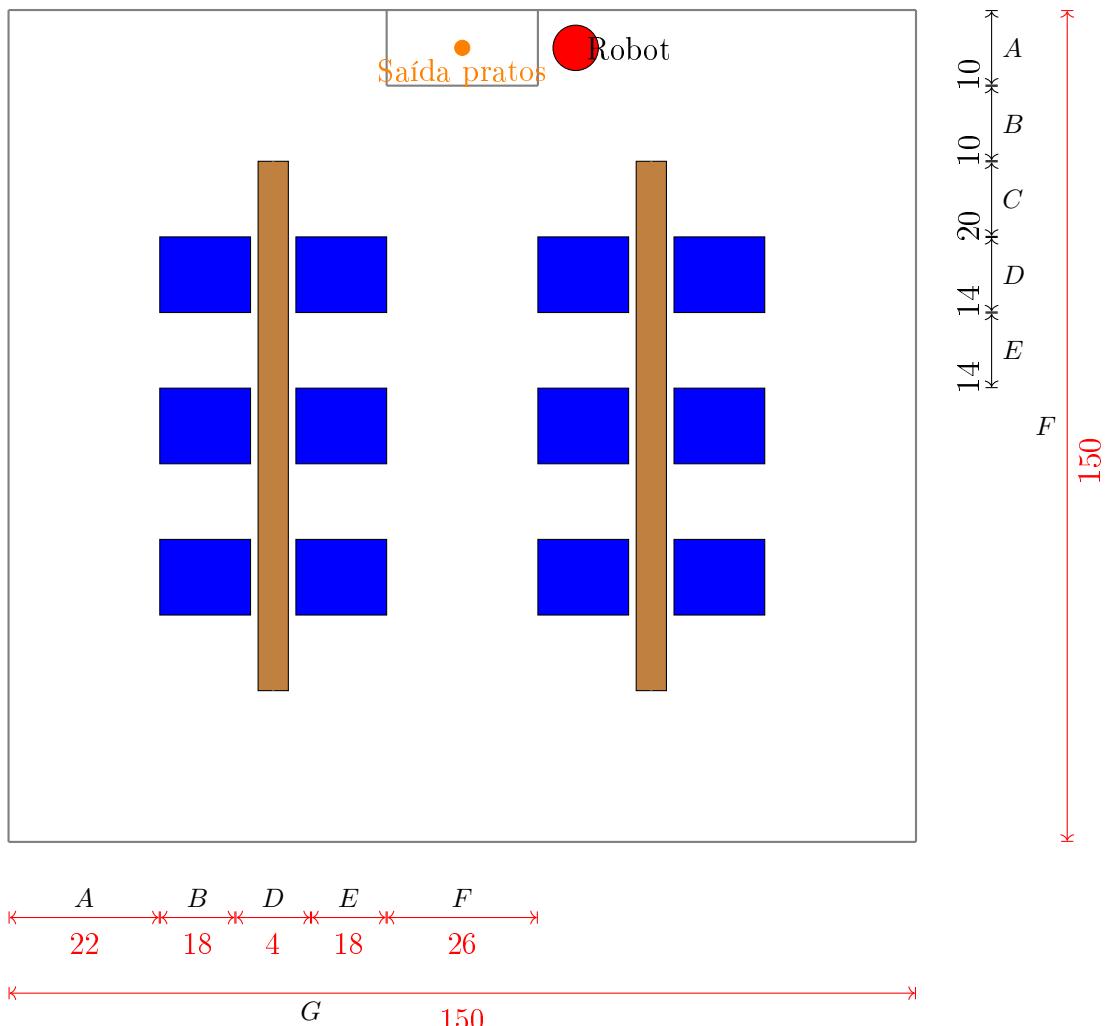
PODERÁ SER UTILIZADO UM SOFTWARE DE DETEÇÃO DE CÓPIAS. TODOS OS ALUNOS QUE SE PROVE TEREM COPIADO NOS PROJETO TERÃO A NOTA FINAL DE 0 (ZERO) VALORES, SENDO O FACTO PARTICIPADO AO CONSELHO PEDAGÓGICO DO IST.

BOM TRABALHO!

Parte II

Plantas

Planta da sala
R1.1



Parte III

Anexos justificativos - Não fazem
parte do trabalho

Anexo 7

Exemplos de robot's

Embora a simulação solicitada seja uma simplificação do problema, existem proposta comerciais de robot's com este fim (ver por exemplo as figuras 7.1 ou 7.2c), podendo-se encontrar nos arredores de Lisboa exemplos.

Os robôs de entrega de alimentos podem melhorar a eficiência do serviço, reduzir custos operacionais e criar características de serviço para a indústria de catering. Além de servir pratos em restaurantes, também pode ser utilizado para servir chá e água em escritórios, enviar e receber entregas expressas, entregar comida para viagem e prestar serviços de distribuição de mercadorias para a indústria hoteleira. ⁽¹⁾



Figura 7.1 – Robot waiter,from <https://www.pudurobotics.com/>

¹<https://www.keenonrobot.com/EN/index/Lists/index/catid/11.html>



(a) A



(b) B

(c) Robot waiter, ^a

(d) V

^afrom www.keenonrobot.com

Figura 7.2 – Examples of robotic waiters (from Web)

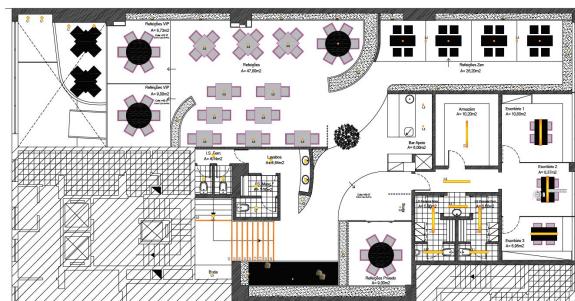
Anexo 8

Plantas

O ambiente de funcionamento é limitado, e consequentemente a planta da sala é conhecida.



(a) 1a



(b) 1b

Figura 8.1 – Exemplos de plantas de restaurantes

Anexo 9

Distâncias

De entre os métodos para medir as distâncias, são comuns o Sonar e Lidar:

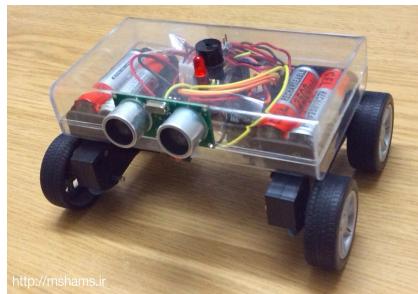
Sonar (do inglês **SOund NAVigation and Ranging** ou “Navegação e Determinação da Distância pelo Som”) é um instrumento inicialmente usado em época de guerra para a localização de submarinos, mas que hoje em dia passou a ter muita utilização na navegação, na pesca, no estudo e pesquisa dos oceanos, estudos atmosféricos.¹

Lidar Lidar (estilizado como LIDAR ou LiDAR; do acrônimo inglês "light detection and ranging"; alternativamente LADAR) é uma tecnologia óptica de deteção remota que mede propriedades da luz reflectida de modo a obter a distância e/ou outra informação a respeito um determinado objecto distante.²

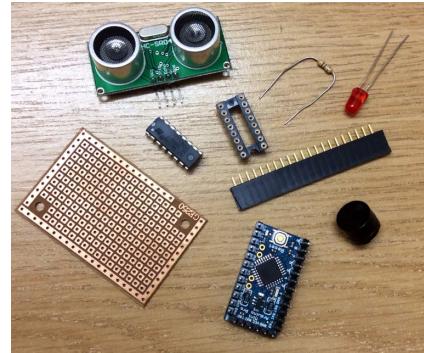
Estes sistemas são simulados directamente

¹wiki - <https://pt.wikipedia.org/wiki/Sonar>

²wiki - <https://pt.wikipedia.org/wiki/Lidar>

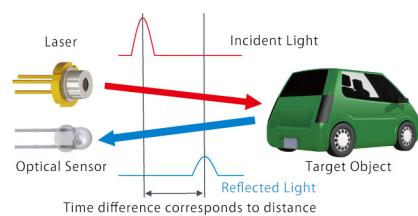


(a) agv com sonar



(b) sensor

Figura 9.1 – Sonar



(a) lidar



(b) sensor

Figura 9.2 – Lidar

Anexo 10

Omnidirectional

Um robô omnidirecional(¹) que se pode mover em qualquer direção, para frente/trás, mas também lateralmente para a esquerda/direita, e girar no mesmo lugar, graças às suas rodas, dando ao robô capacidades omnidirecionais para movimento no eixo horizontal em um trem de força, bem como para frente e para trás. movimento para trás. Além disso, os robôs omnidirecionais precisam de menos espaço de trabalho devido à sua tecnologia que economiza espaço. Os robôs omnidirecionais são altamente adaptáveis, além de serem feitos para instalação e remoção rápidas.

O principal benefício dos robôs omnidirecionais inclui capacidade de manobra, instalação rápida e tecnologia que economiza espaço. Um robô omnidirecional pode se mover em qualquer direção, para frente/trás, mas também lateralmente para a esquerda/direita, e girar no mesmo lugar, graças às suas rodas, dando ao robô capacidades omnidirecionais para movimento no eixo horizontal em um trem de força, bem como para frente e para trás. movimento para trás. Além disso, os robôs omnidirecionais precisam de menos

¹omni- elemento de composição Exprime a noção de tudo, todos, totalmente (ex.: omnidisciplinar). in Dicionário Priberam da Língua Portuguesa [em linha], 2008-2024, <https://dicionario.priberam.org/omni->

espaço de trabalho devido à sua tecnologia que economiza espaço. Os robôs omnidirecionais são altamente adaptáveis, além de serem feitos para instalação e remoção rápidas.

Omnidrive vs. robôs de acionamento diferencial

Os robôs Omnidrive apresentam muitas vantagens em termos de mobilidade, incluindo a capacidade de se movimentar em espaços confinados, situações lotadas e locais apertados, bem como a capacidade de virar de lado, virar no mesmo lugar e seguir trajetórias complexas, graças ao seu movimento de 360º.

Além disso, esses veículos são capazes de realizar tarefas com facilidade em ambientes com obstáculos estáticos e dinâmicos e espaços estreitos.

Um robô com acionamento diferencial é um robô móvel com movimento baseado em duas rodas acionadas separadamente, situadas em cada lado do corpo do robô, o que permite que o robô se move para frente e para trás, mas não para os lados. Isto significa que alcançar todos os locais e orientações num espaço bidimensional requer mais manobras e um planeamento de caminhos potencialmente mais complexo.

A escolha entre robôs de acionamento omnidrive e diferencial depende de cada aplicação. Por um lado, os robôs omnidrive parecem ser a melhor opção em situações que exigem extrema manobrabilidade, graças à sua versatilidade de movimento de 360º. O sistema omnidrive é particularmente útil ao manobrar em ambientes apertados, como transportar cargas longas lateralmente através de portas de tamanho padrão ou corredores estreitos ou estacionar em paralelo, enquanto um robô de acionamento diferencial faria mais manobras do que um robô de acionamento omnidirecional.

Os robôs de acionamento diferencial, por outro lado, são amplamente



Figura 10.1 – Omnidirectional robot, ACCAI, IST[?]

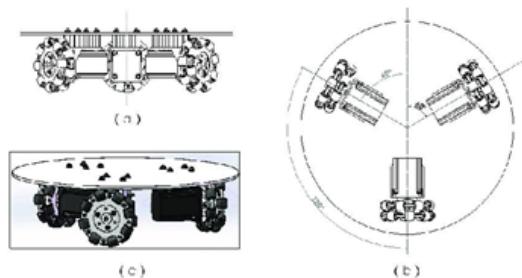


Figura 10.2 – Omnidirectional robot, web[?]

utilizados na robótica porque seu movimento é simples de programar e facilmente controlado. Muitos robôs de consumo no mercado hoje utilizam direção diferencial, devido à sua simplicidade. Além disso, os robôs de acionamento diferencial são uma escolha acertada quando se trata de impulsionar a potência. As rodas omnidirecionais também requerem motores individuais com velocidade controlada para cada roda.

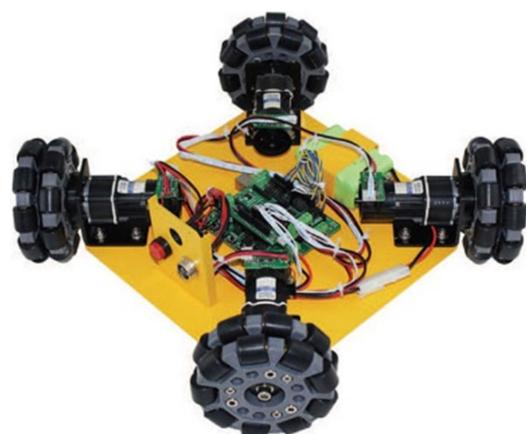


Figura 10.3 – Omnidirectional robot, web[?]

Table of Contents

I Projecto 2025	3
1 Introdução	5
2 Requisitos do programa	7
2.1 Partida - Tier 0	7
2.2 Implementação básica (tier 1, nível básico) - Navegação	8
2.3 Implementação intermédia (tier 2, Segundo nível-entrega	9
2.3.1 Implementação - notas	10
2.4 Implementação nível superior (tier 3)	11
2.5 Sala - formato da descrição	11
3 Indicações gerais	13
3.1 Entrega	13
3.2 Documentação do trabalho (ver ponto 4)	14
3.2.1 Tolerância	15
4 Documentação	17
4.1 Generalidades	17
4.2 Sumário executivo	18
4.3 Documentação para utilizadores.	18
4.4 Documentação técnica	19
5 Critérios de avaliação	21
6 Bonus/Malus	23
II Plantas	27
III Anexos justificativos - Não fazem parte do tra-	

Projecto	CONTENTS
lho	1
7 Exemplos de robot's	3
8 Plantas	5
9 Distâncias	7
10 Omnidirectional	9
Table of Contents	14