



Orlando Bento  
Pinheiro

Gestão inteligente de trajetórias para AGVs





**Orlando Bento  
Pinheiro**

## **Gestão inteligente de trajetórias para AGVs**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestrado em Engenharia Mecânica, realizada sob orientação científica de Abílio Manuel Ribeiro Borges, Assistente Convidado do Departamento de Engenharia Mecânica da Universidade de Aveiro e de Pedro Nicolau Faria da Fonseca, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Apoio financeiro dos projetos  
UID/EMS/00481/2013-FCT e CENTRO-  
01-0145-FEDER-022083



## **O júri / The jury**

Presidente / President

**Professor Doutor José Paulo Oliveira Santos**

Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro

Vogais / Committee

**Professor Doutor Artur José Carneiro Pereira**

Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

**Engenheiro Abílio Manuel Ribeiro Borges**

Assistente Convidado do Departamento de Engenharia Mecânica da Universidade de Aveiro (orientador)



## **Agradecimentos / Acknowledgements**

Gostaria de agradecer ao Engenheiro Abílio Borges pela orientação e apoio que sempre me deu, pelas sugestões e conhecimentos que me transmitiu ao longo desta dissertação.

Ao Professor Doutor Pedro Fonseca o meu sincero agradecimento pela co-orientação neste projeto, por todo o apoio e disponibilidade demonstrada.

Agradeço ao Professor Doutor Vítor Santos pela disponibilidade, simpatia, partilha de conhecimento e conselhos dados ao longo deste projeto.

Ao Eng. Sérgio Almeida da Renault CACIA por me ter possibilitado a realização do projeto na Renault e pelo auxílio, apoio e conhecimento transmitido no desenvolvimento da dissertação que foram fundamentais para que chegasse ao fim com um enorme sentimento de satisfação.

Gostaria de agradecer à minha namorada pela ajuda, paciência e apoio que me deu ao longo destes meses.

Aos meus colegas do Laboratório de Automação e Robótica que sempre me transmitiram confiança, força, motivação e que de uma maneira ou de outra me foram ajudando a ultrapassar os obstáculos.

Deixo ainda um agradecimento a todos os meus amigos com quem partilhei momentos nestes últimos anos. Obrigado pela amizade e pelo incentivo que me deram todos os dias.

Finalmente quero deixar um agradecimento especial à minha família, em particular aos meus pais, irmão, avós e tia pela dedicação, pelo apoio constante e por sempre terem acreditado em mim.





**Palavras-chave**

AGV; Gestão de AGV; Planeamento de Trajetórias; Sistema Inteligente; Interface de comunicação

**Resumo**

Esta dissertação propõe uma solução de gestão de AGV de forma autónoma e inteligente com base em problemas reais da Renault CACIA.

Inicialmente foi desenvolvida uma interface de comunicação com controladores industriais (PLC) permitindo transferir informação a linguagens de alto nível de modo a possibilitar a integração destes controladores no sistema de gestão de AGV.

De seguida foi desenvolvido um sistema de controlo dos AGV. Através da informação recebida pela interface anteriormente referida, atribui tarefas definindo trajetos específicos aos AGV. Este sistema baseia-se na informação recebida em tempo real pelos AGV.

Por fim foi desenvolvida uma interface web para permitir a monitorização e controlo do sistema. Esta permite efetuar a configuração do sistema bem como monitorizar todas as atividades a decorrer, nomeadamente ver as trajetórias planeadas e reais dos AGV, os pedidos efetuados bem como o seu estado entre outras funcionalidades.



**Keywords**

AGV; AGV Management; Path-Planning; Intelligent System; Communication Interface

**Abstract**

This thesis proposes an autonomous and intelligent solution for AGV management based on real problems of Renault CACIA.

Initially, it was developed a communication interface with industrial controllers (PLC) allowing the information transfer to high level application languages to allow the connection between the controllers and the AGV management system.

After, it was developed an AGV control system, based on the received information by the interface, assigns tasks and sets specific targets to the AGV. This system make decisions based on the information received in real time by the AGV.

Lastly, it was developed an interface web to enable the monitoring and control of the system. This interface web allows the configuration of the system as well as the monitoring of all the ongoing activities, namely to see the planned and real pathways of the AGV, the received requests as well as their status, in addition to other functionalities.



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Enquadramento . . . . .	1
1.2	Descrição do Problema . . . . .	2
1.3	Objetivos Principais . . . . .	2
1.4	Plataformas de Desenvolvimento . . . . .	3
1.4.1	<i>Software</i> . . . . .	3
1.4.2	<i>Hardware</i> . . . . .	3
1.5	Estrutura do Documento . . . . .	6
<b>2</b>	<b>Revisão do Estado da Arte</b>	<b>7</b>
2.1	Sistemas de Guiamento de AGV Convencionais . . . . .	7
2.1.1	Guiamento por Fio (Filoguiamento) . . . . .	7
2.1.2	Guiamento por Fita Magnética . . . . .	8
2.1.3	Guiamento Ótico . . . . .	8
2.1.4	Guiamento por Laser . . . . .	9
2.1.5	Guiamento por Triangulação . . . . .	9
2.2	Sistemas de Localização . . . . .	10
2.2.1	Sistemas de Localização Relativa . . . . .	10
2.2.2	Sistemas de Localização Absoluta . . . . .	10
2.3	Sistemas de Supervisão e OPC . . . . .	11
2.4	Sistemas de gestão de trajetórias e AGV . . . . .	12
2.4.1	Planeamento de Trajetórias . . . . .	12
2.4.2	Sistema Comercial de Gestão de Frotas AGV . . . . .	13
2.5	Algoritmos de Problemas de Procura de Caminhos . . . . .	15
2.5.1	Disjktra . . . . .	15
2.5.2	A* . . . . .	15
<b>3</b>	<b>Interface de Comunicação com Controladores Industriais</b>	<b>17</b>
3.1	Protocolos de Comunicação Industriais . . . . .	17
3.2	Interface de Comunicação . . . . .	19
<b>4</b>	<b>Planeamento de Trajetórias</b>	<b>23</b>
4.1	Mapeamento da Zona de Navegação . . . . .	24
4.2	<i>Feedback</i> do Sistema . . . . .	25
4.3	Algoritmo de Busca . . . . .	26

<b>5</b>	<b>Sistema de Gestão de Trajetórias Inteligente</b>	<b>27</b>
5.1	Arquitetura da Solução Proposta . . . . .	27
5.2	Base de Dados . . . . .	28
5.3	Programa de Gestão de Trajetórias . . . . .	29
5.3.1	Atribuição de Tarefas . . . . .	29
5.3.2	Pedidos . . . . .	30
5.3.3	Atualização . . . . .	31
5.4	Interface Gráfica . . . . .	32
5.4.1	<i>Overview</i> . . . . .	32
5.4.2	<i>Map</i> . . . . .	33
5.4.3	<i>Configurations</i> . . . . .	34
<b>6</b>	<b>Equipamentos para Testes e Resultados Experimentais</b>	<b>37</b>
6.1	Dispositivo de Monitorização da Posição dos AGV . . . . .	37
6.2	Configuração do AGV MiR100 . . . . .	39
6.3	Resultados Experimentais . . . . .	41
6.3.1	Funcionalidade da Interface de Comunicação com Controladores Industriais . . . . .	41
6.3.2	Funcionalidade do Sistema de Planeamento e Gestão de Trajetórias . . . . .	41
6.3.3	Planeamento de Trajetórias . . . . .	42
<b>7</b>	<b>Conclusões e Trabalho Futuro</b>	<b>45</b>
7.1	Conclusões . . . . .	45
7.2	Trabalho Futuro . . . . .	46

# Lista de Tabelas

3.1	Características dos diferentes tipos de PROFIBUS [31]. . . . .	18
5.1	Tabelas de parametrização do sistema. . . . .	28
5.2	Tabelas de registo e comunicação do sistema. . . . .	29





# Lista de Figuras

1.1	Representação da interação entre os diferentes equipamentos. . . . .	3
1.2	Placa Raspberry Pi 3, modelo B. [4] . . . . .	4
1.3	Leitor RFID-RC522. [6] . . . . .	4
1.4	Automato Siemens ET 200s. [8] . . . . .	5
1.5	Robô Móvel MiR100. [9] . . . . .	5
2.1	Ilustração do sistema filoguiado. [12] . . . . .	7
2.2	Sistema por guiamento magnético. [12] . . . . .	8
2.3	Sistema por guiamento ótico. [12] . . . . .	8
2.4	Sistema guiado por laser. [12] . . . . .	9
2.5	Sistema guiado por Triangulação. [12] . . . . .	10
2.6	Fluxo de informação de um sistema servidor e cliente OPC. [22] . . . . .	12
2.7	Arquitetura do sistema baseado no <i>SGV Manager Software</i> . [25] . . . . .	14
3.1	Esquema dos protocolos de comunicação presentes na linha de fabrico. . . . .	18
3.2	Programa da leitura dos PLC. . . . .	19
3.3	Página de configuração. . . . .	20
3.4	Interface gráfica do programa de leitura dos PLC. . . . .	21
4.1	Mapa com o sentido do fluxo dos componentes. . . . .	23
4.2	Exemplo da construção da matriz de adjacências com base num mapa. . . . .	24
4.3	Mapa com a numeração dos nós e dos caminhos. . . . .	25
5.1	Arquitetura da solução proposta. . . . .	28
5.2	Arquitetura do programa de gestão. . . . .	29
5.3	Processo de Atribuição de Tarefas do programa principal. . . . .	30
5.4	Processo de Pedidos do programa principal. . . . .	31
5.5	<i>Overview</i> - Pedidos das linhas. . . . .	32
5.6	<i>Overview</i> - Estado dos AGV. . . . .	33
5.7	<i>Overview</i> - Tempos médios e atrasos nos troços. . . . .	33
5.8	Separador <i>Map</i> da interface gráfica. . . . .	34
5.9	<i>Configurations</i> - Definição dos caminhos possíveis. . . . .	35
5.10	<i>Configurations</i> - Posicionamento das linhas no mapa. . . . .	35
5.11	<i>Configurations</i> - Definição e posicionamento das TAG RFID. . . . .	35
5.12	<i>Configurations</i> - Definição dos tempos médios . . . . .	36
5.13	<i>Configurations</i> - Propriedades dos nós. . . . .	36
6.1	Dispositivo de monitorização dos AGV. . . . .	37

6.2	Esquema de ligações entre o leitor de RFID e o Raspberry Pi [33]. . . . .	38
6.3	Montagem do sistema desenvolvido num <i>Automated Guided Vehicle</i> (AGV) Cmajor. . . . .	38
6.4	Mapa obtido pelo MiR100. . . . .	39
6.5	Lista dos pontos de interesse configurados no MiR100. . . . .	40
6.6	Configuração das missões no MiR100. . . . .	40
6.7	Exemplo da estrutura de dados recebida do MiR. . . . .	41
6.8	Planeamento sem paragem de AGV. . . . .	42
6.9	Planeamento com paragem de AGV. . . . .	43

# Lista de Acrónimos

**AGV** *Automated Guided Vehicle*. vi, 1, 2, 5–14, 17, 19, 20, 23–34, 37–43, 45, 46

**API** *Application Programming Interface*. 11, 30, 41, 46

**ARM** *Acorn RISC Machine*. 3

**CBA** *Component Based Automation*. 18

**CSS** *Cascading Style Sheets*. 3

**DP** *Decentralized Peripherals*. 18

**FMS** *Field Message Specification*. 17, 18

**GB** *Gigabyte*. 3

**GPIO** *General Purpose Input/Output*. 3

**HMI** *Human Machine Interface*. 17

**HTML** *HyperText Markup Language*. 3

**IEEE** *Institute of Electrical and Electronics Engineers*. 18

**IO** *Input Output*. 18

**IP** *Internet Protocol*. 18, 20

**MiR** *Mobile Industrial Robots*. 5

**MPI** *Multi-Point Interface*. 17, 18

**OPC** *Open Platform Communication*. 6, 11, 17

**PA** *Process Automation*. 18

**PHP** *Hypertext Preprocessor*. 3, 32

**PLC** *Programmable Logic Controller*. 2–4, 17, 19, 27, 41, 45

**PROFIBUS** *Process Field Bus*. 17, 18

**PROFINET** *Process Field Net.* 4, 17, 18

**RAM** *Random Access Memory.* 3

**REST** *Representational State Transfer.* 5, 6, 29, 30, 37, 41

**RFID** *Radio-Frequency IDentification.* 2, 4, 10, 34, 37, 38

**ROS** *Robotic Operating System.* 13

**SCADA** *Supervisory Control and Data Acquisition.* 11

**SPI** *Serial Peripheral Interface.* 4

**TCP** *Transmission Control Protocol.* 18, 20

**VPN** *Virtual Private Network.* 46

**VRP** *Vehicle Routing Problem.* 12

# Capítulo 1

## Introdução

*"Manufacturing is more than just putting parts together. It's coming up with ideas, testing principles, and perfecting the engineering, as well as final assembly." - James Dyson*

### 1.1 Enquadramento

A Robótica é uma área em constante desenvolvimento. Desde a década de 70 tem sofrido um crescimento exponencial e tem vindo a permitir às empresas aumentarem a sua produtividade e consequentemente a sua competitividade.

Numa época de constante preocupação com a flexibilidade dos sistemas fabris, surge o interesse e a procura do desenvolvimento de sistemas robotizados para o auxílio nos processos de fabrico. Com o aumento da produção, um dos fatores que afetam a produtividade é a movimentação de materiais dentro da instalação fabril, representando assim uma grande percentagem de tempo e custo de fabricação.

Outra motivação para as empresas optarem por estes sistemas é a questão ambiental. Com o passar dos anos houve um grande crescimento na utilização de equipamentos que recorrem a motores de combustão para a sua locomoção. Estes equipamentos apesar de terem características interessantes, nomeadamente a massa que conseguem movimentar poluem o ambiente, tornando assim o espaço de trabalho desagradável. Substituindo a utilização de equipamentos movidos através de motores de combustão por AGV, como estes são elétricos, existe uma redução significativa nas emissões de poluentes. Desta forma contorna-se os problemas ambientais.

A fábrica de CACIA - Renault desde a sua criação recorre a diversos equipamentos robóticos para desempenhar certas funções nas linhas de produção da fábrica, no entanto só muito recentemente recorreu a veículos com guiamento automático (AGV) para fazer parte dos transportes de produtos. A implementação destes equipamentos permitiu à Renault diminuir os custos inerentes aos transportes, ter um fluxo de materiais mais contínuo, uma fábrica mais limpa e um espaço de trabalho mais seguro. Em contrapartida, com o aumento do número de AGV a circular pela instalação fabril surge a necessidade de haver um controlo inteligente da frota dos AGV.

Só com a seleção acertada de equipamentos, comunicação entre eles e uma gestão de recursos adequada é possível chegar a elevados níveis de produção e obter o máximo de rentabilidade possível para a empresa. Neste sentido, surge o tema para esta dissertação que tem como finalidade integrar os AGV na rede das células de fabrico e planear as

trajetórias a percorrer otimizando assim o transporte de mercadorias bem como os custos associados.

## 1.2 Descrição do Problema

A fábrica de CACIA - Renault no departamento de fabrico de componentes mecânicos, onde está a ser desenvolvida esta dissertação, está equipada com cinco AGV não dedicados e um dedicado, isto é, cinco AGV efetuam transportes para diversas células de fabrico e um executa somente o transporte numa célula de fabrico . O guiamento dos mesmos é efetuado através de fitas magnéticas e listas pretas presentes no pavimento, cujo funcionamento se encontra descrito no ponto 2.1. Para efetuarem a monitorização dos AGV utilizam TAGs *Radio-Frequency IDentification* (RFID). As TAGs têm ainda como finalidade informar os AGV dos caminhos a escolher em entroncamentos, quando devem aguardar por uma autorização externa entre outras ordens.

Atualmente os AGV funcionam independentes das linhas de produção. O operador da linha, quando necessita de um transporte faz um pedido através de um sistema de botoneiras. Este desencadeia um pedido ao departamento de logística, local onde um operador deste departamento visualiza o pedido, carrega o AGV com os materiais solicitados e, noutro sistema independente ao sistema de pedidos, dá a instrução ao AGV para fazer a entrega da mercadoria.

Todo este processo requer a intervenção de dois operadores e dois sistemas independentes, resultando num sistema com baixa produtividade. Devido ao departamento sofrer alterações regulares, surge ainda a necessidade de haver um controlo dos AGV no sentido da flexibilização, recorrendo a outras técnicas que não as existentes.

De modo a reduzir estes problemas surge a necessidade da criação de um sistema que receba os pedidos automaticamente das unidades de fabrico e faça a atribuição das tarefas automaticamente aos AGV tendo em atenção a melhor trajetórias a efetuar.

## 1.3 Objetivos Principais

O desenvolvimento desta dissertação tem por objetivo a criação de um *software* de gestão inteligente de trajetórias e monitorizar as atividades dos AGV. Esta solução deverá facilitar e otimizar os percursos dos AGV, criando assim soluções alternativas para as suas trajetórias. Para isso a solução a desenvolver deve receber os pedidos de transportes diretamente das linhas de produção, atribuir a tarefa a um AGV especificamente escolhido bem como estabelecer o melhor trajeto para realizar a tarefa com base no *feedback* proveniente dos AGV.

Outro objetivo deste trabalho é o desenvolvimento de uma interface de comunicação com os controladores industriais existentes na fábrica, mais concretamente *Programmable Logic Controller* (PLC) Siemens SIMATIC de modo a tornar a comunicação entre o sistema de AGV e os controladores independente dos operadores.

Para a concretização deste projeto será ainda desenvolvida uma interface gráfica web de forma a facilitar a monitorização e controlo do sistema.

Na figura seguinte pode-se observar a interação entre os diferentes equipamentos.

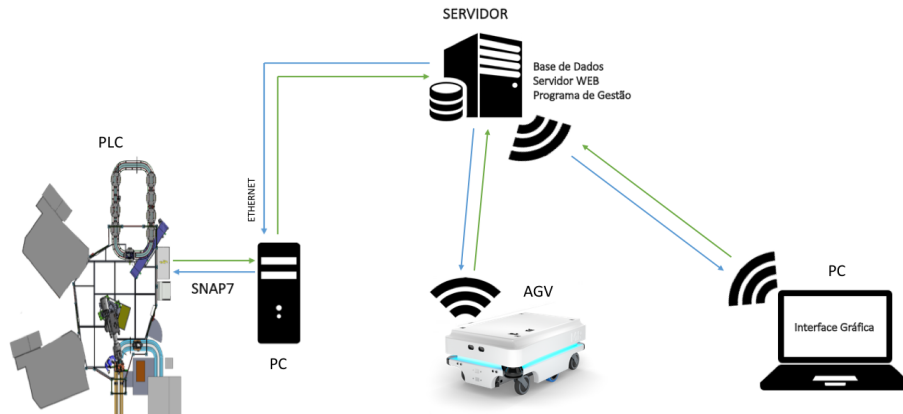


Figura 1.1: Representação da interação entre os diferentes equipamentos.

## 1.4 Plataformas de Desenvolvimento

### 1.4.1 *Software*

Para o desenvolvimento desta dissertação foram utilizados diversos equipamentos com o objetivo de realizarem tarefas específicas que serão explicadas no capítulo 5.

A nível de *software* foram utilizadas diversas linguagens de programação: Python, Visual Basic, *Hypertext Preprocessor* (PHP), *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS) e JavaScript sendo escolhidas por conveniência à situação em questão.

Para o desenvolvimento de *software* foi ainda utilizada a biblioteca SNAP7 que permite a comunicação via *Ethernet* com os PLC S7 da Siemens. Esta biblioteca é *open source*, 32 e 64 bits e multiplataforma. [1]

Para o armazenamento de dados foi utilizada uma base de dados MySQL. Esta base de dados é *open source*, bastante versátil, podendo ser facilmente acedida através de diversas plataformas. [2]

### 1.4.2 *Hardware*

#### Raspberry Pi

O Raspberry Pi é um microcomputador com o tamanho aproximadamente igual a um cartão de crédito, com a particularidade de poder interagir com o mundo real através de pinos, designados por *General Purpose Input/Output* (GPIO) que permitem ligar sensores e atuadores.

Nesta dissertação utilizou-se um Raspberry Pi 3, modelo B, que contém um processador *quad-core* com 1.2 GHz, arquitetura *Acorn RISC Machine* (ARM), 64 bit, 1 *Gigabyte* (GB) de memória *Random Access Memory* (RAM), *Wireless, Bluetooth*, entre outras características. [3]

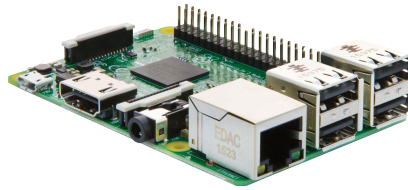


Figura 1.2: Placa Raspberry Pi 3, modelo B. [4]

### Leitor RFID MFRC522

O leitor RFID utilizado tem por base o chip MFRC522 da empresa NXP, permite a leitura de *TAGs* RFID a uma frequência de 13,56MHz que sigam o padrão MIFARE. O chip tem dimensões reduzidas, baixo consumo e baixo custo.

A tensão de funcionamento do leitor é 3.3V DC e consome 13-26 mA, o que permite ser alimentado diretamente pelo Raspberry. Comunica através do protocolo *Serial Peripheral Interface* (SPI). [5]



Figura 1.3: Leitor RFID-RC522. [6]

### PLC Siemens SIMATIC ET 200

O PLC Siemens SIMATIC ET 200 é um automato modular, robusto, que permite ser expandido facilmente acrescentando entradas e saídas digitais e analógicas. Este PLC permite ainda ser utilizado num sistema descentralizado ou seja comunicando com módulos externos de entradas e saídas que ficam distribuídas pelos equipamentos a automatizar.

O Siemens SIMATIC ET 200 vem equipado com *Process Field Net* (PROFINET) que facilita a comunicação com outros equipamentos, inclusive com computadores. [7]





Figura 1.4: Automato Siemens ET 200s. [8]

### MiR100

O MiR é um AGV de grande flexibilidade e de fácil programação. Este AGV permite transportar cargas até 100 Kg sobre o equipamento ou rebocar uma carga de 300 Kg.

O MiR100 contém um sistema de segurança da marca SICK, que utiliza um automato e dois sensores laser e um sistema de controlo desenvolvido pelo fabricante que permite programar e controlar o AGV via Wi-Fi através de uma página gerada pelo equipamento.

Existe ainda a possibilidade de configurar o software do AGV de forma a receber pedidos / chamadas *Representational State Transfer* (REST), permitindo assim ao AGV receber ordens por parte de outros sistemas exteriores ao nativo.

O AGV permite ainda adicionar dispositivos extra, nomeadamente entradas e saídas digitais, *Bluetooth* entre outros dispositivos disponibilizados pela *Mobile Industrial Robots* (MiR). [9]



Figura 1.5: Robô Móvel MiR100. [9]

## 1.5 Estrutura do Documento

Este documento é constituído por sete capítulos incluindo o presente capítulo.

### Capítulo 1:

Este capítulo faz um breve enquadramento bem como uma descrição do problema, apresenta os objetivos do trabalho e as plataformas de desenvolvimento, tanto o *software* como o *hardware*.

### Capítulo 2:

Este apresenta alguns métodos de guiamento utilizados em AGV e como podemos localizá-los. O capítulo descreve ainda alguns *software* de supervisão e *Open Platform Communication* (OPC) existentes, projetos e trabalhos relacionados com planeamento e otimização de trajetórias bem como os principais algoritmos utilizados para o efeito.

### Capítulo 3:

O capítulo 3 contém uma apresentação detalhada da interface desenvolvida para a comunicação entre controladores industriais e o sistema de gestão dos AGV.

### Capítulo 4:

No quarto capítulo é explicado o procedimento adotado para o planeamento automático de trajetórias.

### Capítulo 5:

Explica a arquitetura da solução proposta para a gestão automática dos AGV bem como a implementação do planeamento de trajetórias. No final é ainda apresentada a interface gráfica que permite monitorizar a atividade dos AGV.

### Capítulo 6:

Descreve o desenvolvimento do dispositivo de monitorização dos AGV presentes na fábrica de CACIA - Renault bem como toda a configuração do MiR100 para receber as instruções através dos pedidos REST.

O capítulo 6 apresenta ainda os resultados obtidos durante os testes executados. Estes incluem o resultado da interface de comunicação com os controladores industriais, a funcionalidade do sistema desenvolvido e em detalhe os resultados do procedimento adotado para o planeamento de trajetórias dos AGV.

### Capítulo 7:

Por fim são apresentadas as conclusões do trabalho desenvolvido bem como sugestões de trabalhos futuros.

## Capítulo 2

# Revisão do Estado da Arte

Para a criação de um sistema de gestão de trajetórias para AGV é fundamental o conhecimento prévio de tecnologias de localização e navegação atualmente existentes bem como técnicas e trabalhos relacionados. No presente capítulo é feita uma introdução aos sistemas de localização e guiamento, sistemas de supervisão e gestão de trajetórias e por fim são apresentados dois algoritmos de procura de caminhos que poderão ser usados para planeamento de trajetórias.

### 2.1 Sistemas de Guiamento de AGV Convencionais

Atualmente são utilizados diversos sistemas de localização aplicados em AGV, apresentando vantagens e desvantagens uns em relação aos outros das quais se destacam a precisão, a flexibilidade e o custo. Os sistemas mais utilizados em ambientes industriais são o guiamento por fio, fita magnética, laser, inercial e visão artificial. [10]

#### 2.1.1 Guiamento por Fio (Filoguiamento)

O guiamento por fio é um dos sistemas mais utilizados dada a sua fiabilidade. Este sistema é composto por um ou mais fios condutores colocados sob o solo, que quando fazemos por ele passar uma corrente elétrica alternada cria um campo magnético com uma dada frequência que será detetada pelo AGV. [11] Para uma melhor interpretação podemos observar a figura 2.1. Esta solução é bastante económica pois requer menos meios de programação relativamente a soluções alternativas, no entanto é um sistema pouco flexível, pois é um sistema difícil de alterar devido à sua natureza.

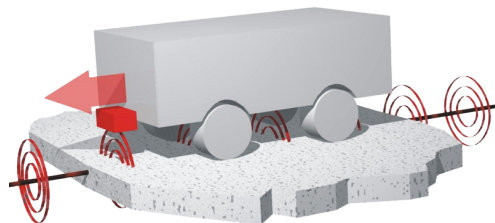


Figura 2.1: Ilustração do sistema filoguiado. [12]

### 2.1.2 Guiamento por Fita Magnética

O guiamento por fita magnética é bastante semelhante ao guiamento por fio como podemos ver na figura 2.2, consistindo na deteção de um campo magnético criado pelas fitas que são coladas diretamente na superfície do chão, não requerendo cortes no pavimento. Esta solução é mais flexível que a anterior devido à facilidade de alteração da configuração das fitas. No entanto é mais suscetível ao desgaste, pois como está à superfície, com a movimentação das pessoas e de veículos, estes acabam por danificar as fitas. [13]

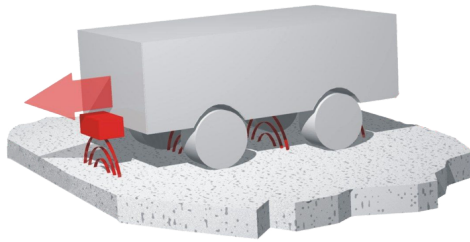


Figura 2.2: Sistema por guiamento magnético. [12]

### 2.1.3 Guiamento Ótico

O guiamento ótico é um método que consiste na leitura de uma marca contínua no solo, normalmente preta delimitada por duas linhas mais grossas brancas. O AGV é equipado com um ou mais fotossensores que permitem a deteção dessa marca, como ilustrado na figura 2.3. Este método é bastante flexível na medida que facilmente se pode alterar a configuração das marcas, económico, no entanto, em ambientes partilhados com outros meios de transporte, como por exemplo empilhadores, as marcas vão-se degradando e acabam por causar problemas de deteção. [14]

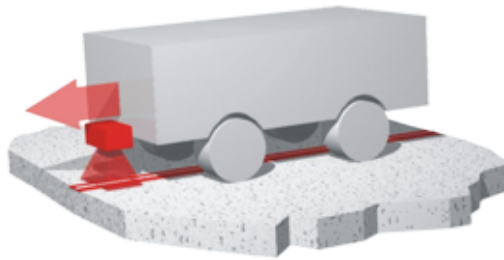


Figura 2.3: Sistema por guiamento ótico. [12]

### 2.1.4 Guiamento por Laser

O método de guiamento por laser é um método que se baseia em triangulação laser. Existem vários alvos dispostos no edifício, com um espaçamento de 6 a 15 metros entre si. A cada alvo são associadas coordenadas X e Y que são conhecidas na unidade de controlo do AGV. O AGV possui um emissor e recetor laser rotativo, que quando emitido um feixe com um dado ângulo conhecido, refletido pelo alvo e recebido novamente pelo AGV, permite determinar a posição em relação ao alvo. Repetindo este processo para vários alvos permite-nos saber com bastante precisão a posição do AGV. O controlador compara a posição calculada com a trajetória pré-definida e determina as instruções necessárias para cumprir o planeamento. Na figura 2.4 podemos observar um exemplo.

Nesta tecnologia é fácil alterar a posição dos alvos e bem como alterar as trajetórias planeada. Devido a estas características este método bastante flexível. No entanto, a utilização desta tecnologia é mais cara que as anteriores, é difícil de implementar em áreas amplas e está sujeita a erros. Alguns problemas desta tecnologia surgem devido à reflexão de sinais. Nestes casos, quando o recetor triangular os sinais vai ser induzido em erro. Outra grande desvantagem é a possibilidade de os alvos ficarem obstruídos não permitindo assim ao AGV saber a sua posição correta. [15]

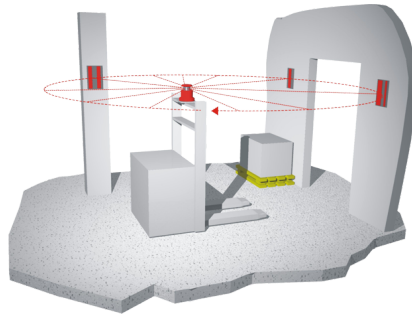


Figura 2.4: Sistema guiado por laser. [12]

### 2.1.5 Guiamento por Triangulação

Este método de guiamento consiste na triangulação de sinais provenientes de antenas ativas permitindo assim ao sistema saber com precisão a posição do AGV. [16]

O AGV desloca-se através de coordenadas para chegar ao ponto alvo da tarefa, usando sensores auxiliares para evitar colisões, nomeadamente por ultrassons, laser entre outros.

Este método é bastante flexível, no entanto pode haver problemas na receção dos sinais, pois dentro de ambientes fabris existe muitas zonas sombra. Este sistema apresenta ainda a desvantagem de ser um sistema que requer uma forte cobertura do sinal que esteja a usar, encarecendo assim o sistema.

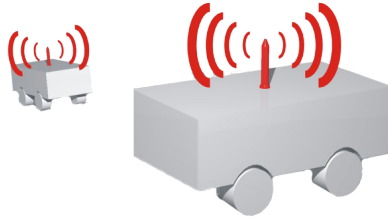


Figura 2.5: Sistema guiado por Triangulação. [12]

## 2.2 Sistemas de Localização

Enquanto a secção anterior refere as tecnologias associadas ao deslocamento do AGV, nesta serão apresentadas técnicas utilizadas para se saber a posição dos mesmos. Este estudo é importante porque algumas tecnologias de navegação não permitem saber a posição real do AGV e neste caso tem de se recorrer a tecnologias complementares.

Os sistemas de localização podem ser divididos em duas categorias, relativa e absoluta.

### 2.2.1 Sistemas de Localização Relativa

Um sistema odométrico é o mais comum no que diz respeito aos sistemas de localização relativa. Este sistema recorre a sensores inerciais como acelerómetros e giroscópios ou a *encoders* e com base numa posição anterior estima a sua posição atual. [17] Este método não é invasivo na instalação fabril, apresentando inúmeras vantagens, desde a facilidade de implementação à flexibilidade. No entanto, apesar destas vantagens, o método sendo um método que tem por base a posição anterior para saber a sua posição atual, acumula erros ao longo da navegação que torna pouco fiável, pois com o passar do tempo existem grandes erros associados. [15]

### 2.2.2 Sistemas de Localização Absoluta

Existem diversos sistemas de localização absoluta, os que recorrem a localizadores ativos, localizadores passivos e os que estabelecem correspondência com um modelo conhecido.

Os sistemas através de localizadores ativos recorrem a emissores que enviam um sinal e através de 3 ou mais sinais recebidos o AGV consegue triangular os sinais e saber a sua posição exata com uma elevada precisão. Este sistema é o mesmo que foi referido no ponto 2.1.5, para além de ser um sistema de localização permite também ao AGV navegar através da triangulação. Este sistema apresenta diversas vantagens tais como flexibilidade, não invasivo, entre outras. No entanto, é um sistema dispendioso, de difícil programação e com possibilidade de existirem erros, como por exemplo devido à reflexão de sinais induzindo assim erros na triangulação. [18]

Os sistemas através de localizadores passivos recorrem normalmente a marcas naturais, inerentes ao espaço onde navegam, ou artificiais, que são colocadas estrategicamente. Estes marcadores podem ser visuais e recorre-se a técnicas de perceção para os reconhecer ou RFID e neste caso recorre-se a antenas específicas para as ler. Sabendo a posição destes marcadores no ambiente, aquando a passagem do AGV por eles, pode-se estimar

a sua posição. Existe ainda a possibilidade de recorrer a 3 ou mais marcadores visuais e desta forma triangular a posição do AGV.

Os sistemas de correspondência com um modelo conhecido consiste na comparação da informação recebida pelos sensores do AGV com um modelo ou mapa previamente instalado no sistema. Este sistema é não invasivo, bastante flexível, no entanto de difícil programação e robustez, pois está dependente do meio de navegação, isto é, caso haja alterações do ambiente de navegação poderá ser necessária reconfiguração do mapa. [19] Esta reconfiguração, em muitas soluções comerciais requer a intervenção do vendedor para a efetuar.

### 2.3 Sistemas de Supervisão e OPC

Os sistemas *Supervisory Control and Data Acquisition* (SCADA) consistem em *software* de monitorização e supervisão de sistemas industriais, como por exemplo unidades fabris. Recorrem a programas de aquisição de dados utilizando diferentes protocolos de comunicação industriais.

Com a evolução destes sistemas houve um crescimento na utilização de *softwares* OPC de forma a permitir o acesso aos dados dos controladores industriais através de programas de alto nível.

Um servidor OPC consiste num módulo de *software* que comunica com os controladores industriais através de protocolos específicos do fabricante e que, posteriormente, permite aceder a essa informação através do protocolo OPC. Normalmente este *software* é disponibilizado pelos fabricantes dos controladores.

Para aceder às informações previamente recebidas pelo servidor OPC é necessário um outro *software* que se comporta como cliente OPC. Este envia comandos específicos ao servidor para que comunique com o controlador e retribua a informação pedida. [20] Na figura 2.6 é possível observar um esquema do fluxo de informação num sistema baseado em servidor e cliente OPC. Este *software* pode ser comercial ou desenvolvido através de uma *Application Programming Interface* (API). A API disponibilizada pela *OPC Foundation* permite o desenvolvimento de servidores e clientes OPC em ambiente Microsoft .NET. [21]



Figura 2.6: Fluxo de informação de um sistema servidor e cliente OPC. [22]

## 2.4 Sistemas de gestão de trajetórias e AGV

No mercado já existem diversos sistemas de gestão de AGV de forma a automatizar a atribuição de tarefas aos AGV, no entanto, são sistemas com um custo elevado. Estes sistemas, em geral, não fazem um planeamento em tempo real da melhor trajetória que o AGV deve efetuar.

No sentido de desenvolver uma solução completa foi analisado *software* comercial de gestão de AGV e foi feita uma pesquisa por projetos desenvolvidos sobre planeamento de trajetórias.

Nos pontos seguintes vão ser apresentados dois trabalhos de dissertação relacionado com o planeamento de trajetórias, um sistema comercial gestão de AGV e o sistema existente na fábrica de CACIA - Renault.

### 2.4.1 Planeamento de Trajetórias

Na literatura realizada destacam-se duas dissertações que se relacionam com o trabalho a ser desenvolvido.

A dissertação de Luiz Caccalano, intitulada "Roteirização de veículos para abastecimento de linhas de produção" apresentada à Escola Politécnica da Universidade de São Paulo é um exemplo de planeamento de trajetórias. Esta dissertação visa desenvolver uma heurística de forma a obter uma solução para um problema de *Vehicle Routing Problem* (VRP).

A dissertação propõe uma solução tendo por base uma matriz de distâncias mínimas entre cada interceção de caminhos. O mapa é dividido em caminhos e interceções e realiza-se uma matriz que define a distância mínima de cada caminho, ou seja entre duas interceções consecutivas.

A heurística adotada estima inicialmente o número de caminhos necessários para chegar ao seu destino através de um método sequencial. Posteriormente otimiza a solução recorrendo ao método de inserção paralelo combinado com um algoritmo para melhorar as soluções dos anteriores.



No método sequencial, o algoritmo constrói a trajetória caminho a caminho até esgotar todas as possibilidades para chegar ao seu destino enquanto os métodos paralelos assumem que o número de caminhos é conhecido e associam os destinos a trajetórias ainda em aberto com base num critério definido. [23]

A dissertação de Joana Santos, apresentada à Faculdade de Engenharia da Universidade do Porto intitulada de "*Multi-Robot Cooperation Systems for Assembly Automobile Industry*" tem como objetivo a criação de um sistema realista de simulação com múltiplos robôs, estudar algoritmos de coordenação e fazer a sua implementação.

Numa fase inicial foi desenvolvido um ambiente de simulação em V-Rep, ferramenta esta disponível através da arquitetura *Robotic Operating System* (ROS).

Posteriormente foi desenvolvido um algoritmo de planeamento de caminhos para sistemas multi-robôs. Este calcula os caminhos para os diversos robôs com base nas suas posições e nos caminhos que os outros vão efetuar. O Algoritmo tem por base o A\*.

Por fim, o algoritmo é integrado no ambiente de simulação desenvolvido. [24]

#### 2.4.2 Sistema Comercial de Gestão de Frotas AGV

No mercado existem diversos sistemas de gestão de frotas de AGV, por vezes facilitados pelos fabricantes dos equipamentos. Para a recolha de informação sobre estes sistemas foram estudados dois sistemas em detalhe: o *SGV Manager Software* da empresa JBT Corporation e o *vsystems* da empresa hedesis utilizado pela fábrica de CACIA - Renault.

Ambos os sistemas são muito semelhantes. Estes permitem o controlo e monitorização dos AGV em tempo real. Através das suas interfaces é possível configurar o sistema para agir em função de redes de *Inputs / Outputs* standardizados, ou seja ler sensores, leitores de códigos de barras, entre outros. Para além de reagir conforme os dados recebidos permite ainda atuar alguns dispositivos como por exemplo portas automáticas.

O *software* permite ainda configurar alarmes e notificações para informar o utilizador dos acontecimentos na instalação fabril e gerar relatórios da atividade dos AGV. [25]

Um entrave destes programas é não permitirem dar indicações da melhor trajetória a efetuar no momento da tarefa.

Na figura 2.7 é possível observar um esquema da arquitetura do sistema *SGV Manager Software*.

Analisando a figura, na parte superior pode-se ver todos os equipamentos que são permitidos comunicar no sistema através de comunicação em rede. Toda a comunicação entre equipamentos fixos é efetuada através de cabo. Na parte inferior da figura pode-se verificar que, através dos diferentes *Access Points* o sistema comunica com os AGV através de *Wireless*. Para a monitorização do sistema pode-se usar diversos equipamentos como explica a figura. Estes recorrem também a *Wireless* para comunicar.

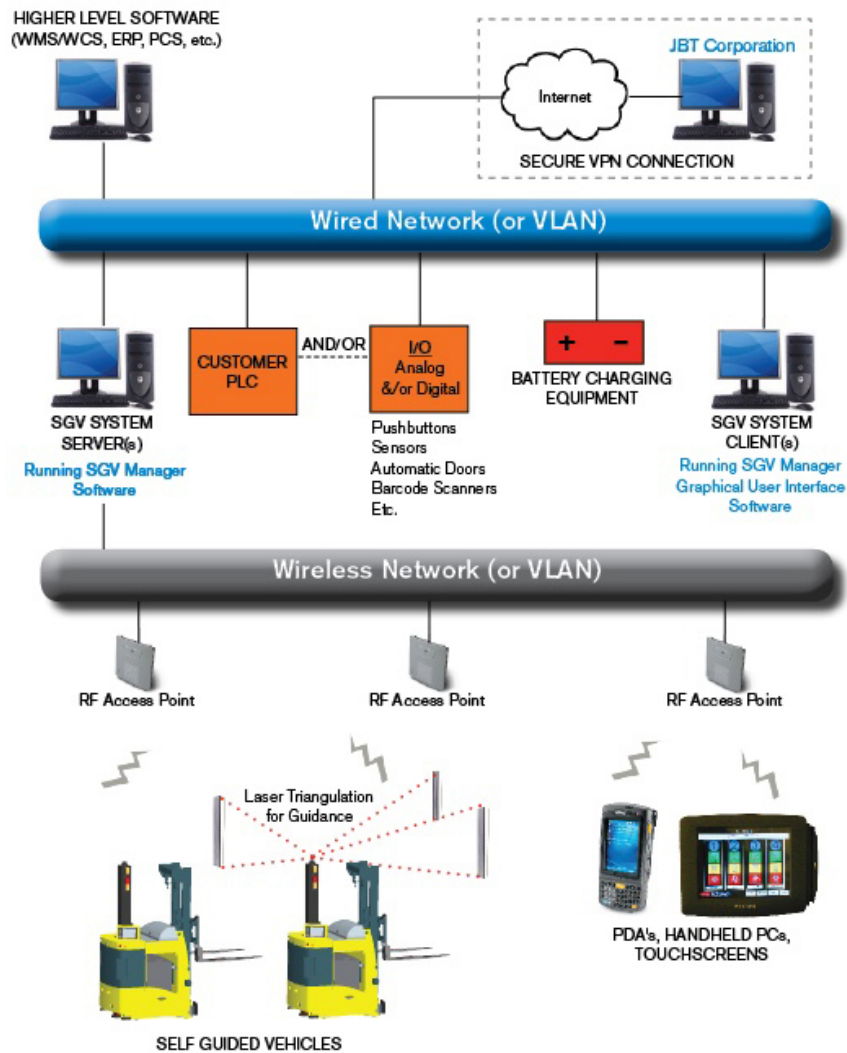


Figura 2.7: Arquitetura do sistema baseado no *SGV Manager Software*. [25]

Como referido, o *vsystems* é muito semelhante, no entanto recorre a AGV que navegam através de guias magnéticas e óticas e não por laser como os apresentados na figura.

## 2.5 Algoritmos de Problemas de Procura de Caminhos

Sendo o objetivo principal do trabalho a escolha otimizada e inteligente de trajetórias é necessário perceber algumas técnicas de otimização existentes. Este estudo tem como finalidade perceber qual é o melhor algoritmo a utilizar para auxiliar no desenvolvimento do software de planeamento. Existem diversos algoritmos de otimização, no entanto iremos analisar em particular o algoritmo Dijkstra e A\*, pois são os mais utilizados nestas aplicações.

### 2.5.1 Dijkstra

O algoritmo Dijkstra, desenvolvido pelo holandês Edsger Dijkstra em 1956 [26], é um algoritmo que procura o acesso mais curto entre dois vértices de um grafo com arestas de peso não negativo, isto é, a partir de um grafo que relaciona os seus vértices e através de valores correspondentes às arestas, o algoritmo procura o caminho com menor custo para chegar de um ponto a outro.

O algoritmo Dijkstra inicia com uma matriz de distâncias entre o nó inicial e todos os outros do gráfico, em que a distância a si próprio é zero e a todos os outros é infinito. Ao correr o algoritmo a distância entre a origem e cada nó é recalculado e termina quando a distância mínima for encontrada. No final do algoritmo todos os nós foram visitados, ou seja, quantos mais nós existirem mais nós tem que visitar, tornando assim o algoritmo bastante pesado computacionalmente. [27]

### 2.5.2 A\*

Por vezes o itinerário ótimo não é o mais curto, pois pode ser um caminho menos desejado em certas situações. Nestes casos o algoritmo A\* [28] apresenta vantagens comparativamente ao Dijkstra.

O Algoritmo A\* é uma versão otimizada do Dijkstra que permite a adição de uma estratégia que torna a escolha direcionada.

Este algoritmo utiliza a informação dos nós como o anterior mas também uma heurística, geralmente da distância ao nó final. Numa fase inicial o algoritmo ignora os pontos mais afastados do objetivo chegando assim mais rapidamente ao objetivo, no entanto caso não consiga chegar ao destino tem em conta os nós mais distantes.

A possibilidade de utilizar uma heurística permite ao algoritmo ser considerado um algoritmo de inteligência artificial contrariamente ao Dijkstra.

Dependendo dos resultados obtidos podemos considerar a heurística admissível (ou otimista) ou não admissível. Uma heurística é admissível se nunca superestimar o custo mínimo possível no nó atual.

Outra grande diferença entre o algoritmo Dijkstra e o A\* é o facto de o primeiro ser um algoritmo exaustivo e desta forma dar uma solução exata enquanto o A\*, como não analisa todas as possibilidades pode não chegar à solução ótima. [29]



## Capítulo 3

# Interface de Comunicação com Controladores Industriais

De modo a integrar o sistema de controlo dos AGV com as linhas de produção é necessário utilizar uma interface de comunicação que permita efetuar a ligação entre os dois sistemas.

As interfaces existentes, baseadas em servidor e cliente OPC são dispendiosas, pois geralmente o servidor OPC é do fabricante dos controladores industriais. De forma a contornar esse inconveniente foi estudada a possibilidade de criar uma interface de comunicação dedicada para o efeito.

Visto que este trabalho visa propor uma solução baseada nos sistemas existentes na fábrica de CACIA - Renault, numa primeira fase foram estudados os equipamentos que equipam as linhas de produção.

Neste capítulo é feita uma breve introdução aos protocolos de comunicação existentes na empresa e apresentada a solução proposta para a interface de comunicação com os controladores Siemens SIMATIC.

### 3.1 Protocolos de Comunicação Industriais

Os equipamentos da fábrica de CACIA - Renault estão equipados com controladores April, Schneider, Siemens, Fanuc, entre outros. No entanto, todos têm também instalado controladores Siemens. Os protocolos de comunicação disponíveis são: *Multi-Point Interface* (MPI), *Process Field Bus* (PROFIBUS) e PROFINET.

O protocolo MPI é propriedade da Siemens e permite comunicar entre equipamentos da família SIMATIC. Na fábrica de CACIA - Renault, este protocolo está presente essencialmente na comunicação entre os PLC e as consolas de diálogo, também conhecidas por *Human Machine Interface* (HMI) ou entre PLC. Esta tecnologia é inspirada no protocolo PROFIBUS e suporta velocidades de transmissão entre 187.5 kbps a 12 Mbps. O endereço do equipamento MPI deve ser único, compreendido entre 1 e 256 [30].

O PROFIBUS é um protocolo de comunicação industrial aberto. Este permite comunicar entre equipamentos industriais independentemente da marca e sem qualquer adaptação na interface, o que fez com que se tornasse um dos protocolos mais usados a nível industrial.

Este protocolo baseia-se no modelo *Master/Slave*. Atualmente existem três variantes, o PROFIBUS *Field Message Specification* (FMS) destinado à comunicação entre PLC, o

PROFIBUS *Decentralized Peripherals* (DP), que é uma simplificação do FMS otimizado para dispositivos do ambiente fabril, preparado para grandes volumes de informação e grande velocidade de transferência. Existe ainda o PROFIBUS *Process Automation* (PA) em que a grande diferença é o meio de transmissão, pois usa o cabo de alimentação DC dos dispositivos. A velocidade de transferência é inferior à anterior [31].

A informação mais detalhada destas variantes pode ser encontrada na tabela 3.1.

Tabela 3.1: Características dos diferentes tipos de PROFIBUS [31].

	PROFIBUS DP	PROFIBUS FMS	PROFIBUS PA
Aplicação	Chão de Fábrica	Células	Processos
Padrão	EN 50 170, Vol. 2	EN 50 170, Vol. 2	IEC 1158-2
Tempo de Resposta	1 a 5 ms	menos de 60 ms	menos de 60 ms
Distâncias	até 23 m	até 23 m	até 1.9 m
Velocidade de Transmissão	9.6Kbps a 13 Mbps	9.6Kbps a 1.5 Mbps	até 1.9 Km

Tanto o Protocolo MPI como o PROFIBUS usam maioritariamente o meio físico RS-485 para comunicarem.

Por último o PROFINET é um protocolo de comunicação aberto e utiliza o padrão *Ethernet Institute of Electrical and Electronics Engineers* (IEEE) 802.3.

Existem duas variantes de PROFINET, *Component Based Automation* (CBA) e *Input Output* (IO). PROFINET CBA é apropriado para comunicações baseadas em *Transmission Control Protocol* (TCP)/*Internet Protocol* (IP) em tempo real muito utilizada em controladores modulares, enquanto que o PROFINET IO foi desenvolvido para comunicações com os periféricos de sistemas descentralizados [31].

Na figura 3.1 estão representados os protocolos de comunicação presentes numa linha de produção da Renault que será utilizada para testes da solução proposta.

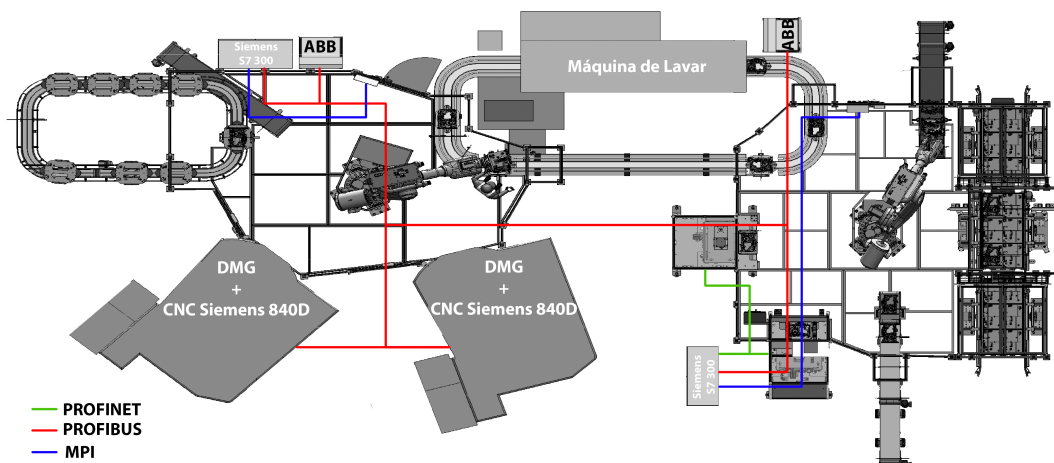


Figura 3.1: Esquema dos protocolos de comunicação presentes na linha de fabrico.

## 3.2 Interface de Comunicação

Após o levantamento do sistema foi desenvolvida a interface de comunicação em Visual Basic. Este programa tem como finalidade ler as variáveis dos PLC da linha que fazem a contagem das peças produzidas, peças com defeito e peças que entram em bruto para posteriormente armazenar um pedido de transporte na base de dados do sistema dos AGV como mostra a figura 3.2.

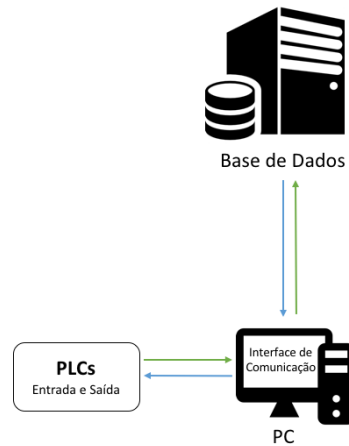


Figura 3.2: Programa da leitura dos PLC.

O software foi desenvolvido de forma a ser o mais flexível possível permitindo assim funcionar na maioria dos equipamentos que sejam geridos por um PLC Siemens. Como a maioria das linhas da Renault são geridas por um ou mais PLC, o programa foi desenvolvido para permitir ler dois PLC, um para entrada de peças e um de saída de peças. Dentro de cada PLC é possível ler os contadores de peças que entram em bruto na linha, peças que saem acabadas ou peças que saem com defeito. Dentro das peças com defeito, é possível ler dois contadores diferentes: um contador de peças que apesar de terem defeito são recuperáveis e outro de peças que são sucata.

No caso da linha ter tudo num único PLC, é possível configurar o programa para operar somente com um PLC.

Através da interface gráfica representada na figura 3.3 o utilizador pode configurar tudo o que foi referido anteriormente entre outros parâmetros.

The screenshot shows a configuration window titled 'Configurações'. It contains two main panels: 'Entrada' and 'Saída'. Each panel has a checked checkbox and fields for IP, Rack, and Slot. Below these are four rows of checkboxes for 'Brutos', 'Acabados', 'Sucata', and 'Recuperáveis', each with 'DB' and 'DBD' sub-fields. At the bottom, there is a 'Base de Dados' section with fields for 'Endereço IP' and 'Nome'. Two buttons, 'Cancelar' and 'Guardar', are located at the bottom right.

Figura 3.3: Página de configuração.

Sendo o objetivo associar uma interface de comunicação a cada linha existente é possível introduzir os parâmetros que dizem respeito à linha, tais como as quantidades de peças de referência para que seja efetuado um pedido de transporte, o número da linha, a quantidade de peças que cada contentor transporta e o tempo de ciclo médio da linha.

O programa após iniciado, cada vez que deve chamar um AGV envia uma mensagem para uma base de dados que será explicada no capítulo 5.

De modo a priorizar os pedidos que chegam à base de dados, juntamente com o pedido, o programa envia o tempo de rutura da linha. Este parâmetro é um descriptor utilizado pela Renault para priorizar os pedidos que existem de forma a satisfazer todos os pedidos sem penalizar outras linhas de produção. Este descriptor tem por base a seguinte fórmula de cálculo:

$$Trut = Tcy * NCont * NPcont \quad (3.1)$$

em que  $Tcy$  é o tempo de ciclo da linha,  $NCont$  é o número de contentores que a linha pode ter em espera e o  $NPcont$  é o número de peças que cada contentor transporta.

Para implementar o programa, são utilizadas duas bibliotecas principais, a SNAP7 que permite estabelecer uma conexão com os PLC Siemens e efetuar a leitura das variáveis e uma biblioteca que realiza a comunicação com uma base de dados MySQL.

Recorrendo à biblioteca SNAP7 o programa estabelece uma comunicação TCP/IP com os autómatos de entrada e de saída. Esta efetua-se através de uma *thread* dedicada para o efeito.

O envio das informações para a base de dados MySQL é estabelecido através da biblioteca MySQL *Connector* de modo a facilitar a comunicação. Esta é efetuada por uma *thread* independente da anterior.

A restante gestão da interface é efetuada no processo principal do *software*.

O programa permite ainda visualizar os contadores selecionados bem como o número de peças que falta para desencadear o pedido de transportes e ainda visualizar os pedidos que foram efetuados ao sistema mas que não estão finalizados, como é possível observar na figura 3.4.



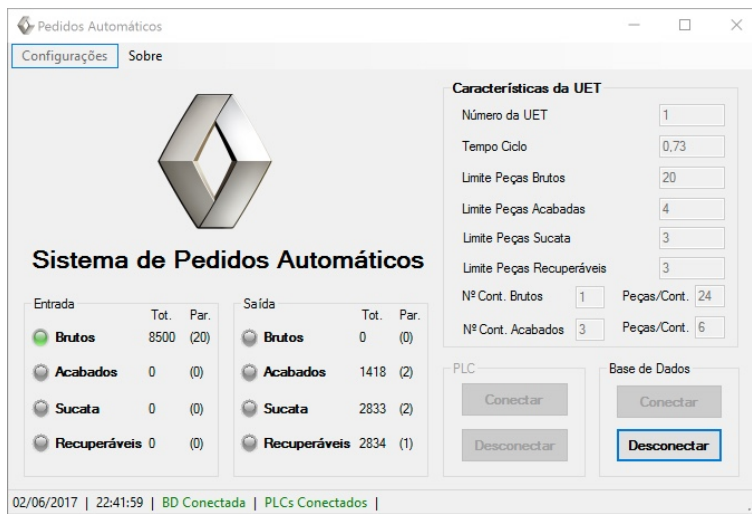


Figura 3.4: Interface gráfica do programa de leitura dos PLC.



## Capítulo 4

# Planeamento de Trajetórias

Neste capítulo é apresentada a metodologia adotada para a realização do planeamento das trajetórias. Baseada na realidade da fábrica de CACIA - Renault, foi definida a melhor abordagem a adotar com base nas restrições existentes.

Para formular uma solução foram tidas em conta as restrições iniciais por parte da empresa. Devido aos AGV existentes serem guiados por fitas magnéticas ou por guiamento ótico e como só podem andar num sentido é possível considerar a navegação dos mesmos a 1 dimensão.

Na fábrica de CACIA - Renault o fluxo de materiais efectua-se essencialmente no sentido dos ponteiros do relógio, havendo exceções em alguns casos. Na figura 4.1 podemos observar o sentido dos transportadores.

Representado a vermelho estão os troços que só têm um sentido, enquanto que a verde estão representados os que têm dois sentidos. A azul encontra-se um troço que só tem um sentido, no entanto, só deve ser utilizado em situações que não haja alternativa. Para que isso aconteça este foi considerado um troço secundário.

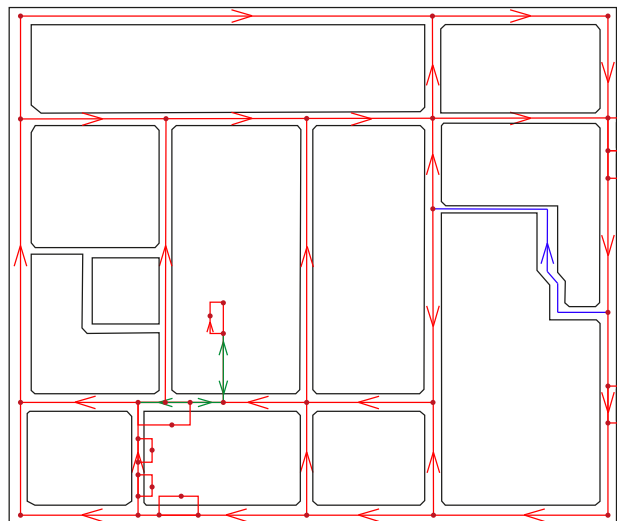


Figura 4.1: Mapa com o sentido do fluxo dos componentes.

A abordagem adotada para solucionar o problema pode ser dividida em três fases.

Numa fase inicial é necessário informar o sistema das restrições iniciais apresentadas anteriormente, isto é, as trajetórias possíveis, em que sentido as pode efetuar bem como se é um trajeto principal ou secundário. Posteriormente é necessário interpretar o *feedback* proveniente dos AGV. Finalmente, dependendo das condicionantes, encontrar o melhor percurso a efetuar.

## 4.1 Mapeamento da Zona de Navegação

Para a criação do sistema de planeamento adotou-se a matriz de adjacências para representar o mapa da fábrica de CACIA - Renault bem como todas as imposições iniciais.

A matriz de adjacências é uma matriz quadrada  $n \times n$  em que  $n$  é o número de nós presentes num grafo finito. Normalmente a relação entre os nós é a distância entre eles caso exista um itinerário que ligue um nó ao outro. Caso contrário, deve estabelecer-se o valor correspondente infinito para que nunca seja possível estabelecer essa ligação. Na posição da matriz que relaciona o ponto consigo próprio o valor é nulo, pois não existe nenhum custo para chegar até ele.

No exemplo presente na figura 4.2 está representado à esquerda um mapa bidirecional em que a distância entre cada nó é 1. À direita podemos observar a matriz de adjacências correspondente de acordo com a metodologia adotada.

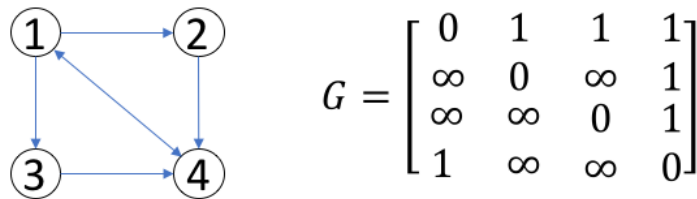


Figura 4.2: Exemplo da construção da matriz de adjacências com base num mapa.

Na matriz de adjacências o número da linha corresponde ao nó de partida e o número da coluna corresponde ao nó de chegada. Por exemplo, do nó 1 para o 3 existe um trajeto possível, logo o valor na linha 1, coluna 3 é 1. No sentido inverso já não tem um trajeto possível logo o valor na linha 3, coluna 1 é infinito.

Apesar de normalmente ser usada a distância como critério para criar a matriz de adjacências, pode ser usado outro valor de custo. Para incluir todas as restrições descritas optou-se pela seguinte fórmula de custo:

$$Custo = C1 * Dist. + C2 * Tipo + C3 * TaxaUtil. \quad (4.1)$$

As constantes  $C1$ ,  $C2$  e  $C3$  são pesos que cada variável tem no custo. Após alguns testes experimentais chegou-se aos valores  $C1 = 2$ ,  $C2 = 75$  e  $C3 = 15$ . As variáveis "Dist.", "Tipo" e "TaxaUtil." são a distância em metros entre dois nós consecutivos, o tipo de caminho e a taxa de utilização, respetivamente. O tipo de caminho pode tomar os valores 0 caso seja um caminho principal e 1 se for secundário. A taxa de utilização é um parâmetro que permite associar pesos diferentes dependendo da sua taxa de utilização, 0 caso tenha uma utilização normal ou 1 caso à partida já se saiba que é muito utilizado.

Para implementar a matriz de adjacências foi necessário dividir o mapa representado na figura 4.1 em nós e caminhos. O resultado dessa divisão está representado na figura 4.3, em que os números a vermelho representam os nós e os números rodeados representam os caminhos entre os nós.

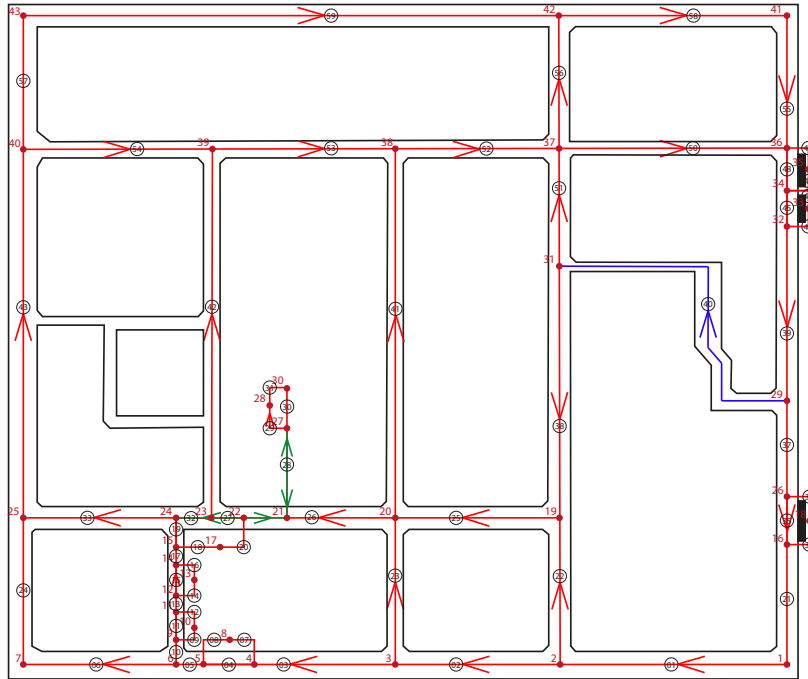


Figura 4.3: Mapa com a numeração dos nós e dos caminhos.

A construção da matriz resulta numa matriz quadrada 43 x 43 com a diagonal nula. Nas posições correspondentes aos caminhos possíveis o valor da matriz é o valor do custo e nas posições onde não é possível efetuar um caminho é infinito.

## 4.2 *Feedback* do Sistema

Tirando proveito das potencialidades da matriz de adjacências, esta solução propõe utilizar uma matriz dinâmica que se atualiza ao longo do tempo dependendo do *feedback* recebido pelos AGV. Esta solução permite que o sistema em tempo real escolha o melhor percurso para o AGV poder chegar ao seu destino com base no que está a acontecer nesse momento.

No arranque do *software*, este calcula o trajeto de menor custo com base nas imposições que foram anteriormente descritas, no entanto é a única altura que se descarta o que está a acontecer.

O *software* para além do controlo dos AGV monitoriza a sua atividade. Passado pouco tempo é possível ter acesso a variáveis como o tempo médio de cada troço, as paragens de AGV e os troços ocupados.

De modo a que o planeamento seja o mais completo possível, e com base no modo de funcionamento dos AGV da fábrica de CACIA - Renault , foi atualizada a fórmula do custo adicionando novos parâmetros: paragens, variação do tempo médio e o número de

AGV que estão planeados para esse troço de forma a evitando atrasos e evitar caminhos obstruídos.

$$Custo = C1*Dist.+C2*Tipo+C3*TaxaUtil.+C4*Paragens+C5*Atrasos+C6*NAGVs \quad (4.2)$$

As constantes C1, C2 e C3 mantêm-se 2, 75 e 15 respetivamente e as C4, C5 e C6 foram também obtidas experimentalmente chegando-se ao valor de 10, 5 e 30, respetivamente. A variável "Paragens" é incrementada a cada cinco segundos aumentando o custo do troço 120 ao fim de um minuto. Os "Atrasos" são resultados do *feedback* proveniente do último AGV que passou por esse caminho. Comparando o tempo que este demorou a efetuar um troço com o padrão que foi obtido num ambiente ótimo é possível chegar a atrasos existentes. Com este valor é possível detetar os troços mais problemáticos devido ao decréscimo da velocidade média. Um aumento de 10 segundos no troço representa um custo de 50. Por último, como o sistema efetua um planeamento prévio dos trajetos dos AGV podemos evitar sobrecargas nos troços, desta forma associou-se um custo de 30 por troço a cada AGV que está planeado passar por ele.

Após algum destes parâmetros voltar à normalidade, o respetivo custo que foi incrementado volta a zero.

### 4.3 Algoritmo de Busca

Após a criação da matriz de adjacências e sua atualização em tempo real é necessário calcular o trajeto com menor custo para ir de um dado ponto para outro.

De forma a tornar essa procura possível foi utilizado o algoritmo Dijkstra. Este algoritmo, como explicado na secção 2.5, é um algoritmo que procura o percurso com menor custo a partir de uma matriz de adjacências. O dijkstra é muito semelhante ao A\*, no entanto o primeiro apresentou vantagens nesta aplicação.

O Dijkstra é um algoritmo exato, que o torna computacionalmente pesado, contudo devolve uma solução que, neste caso é garantidamente o percurso de menor custo. O algoritmo A\* é mais rápido computacionalmente, pois não analisa todas as possibilidades.

Dado que a matriz que o algoritmo tem que analisar é 43 x 43, o tempo de computação é desprezável pois os dados são poucos de forma a ter um tempo computacional decisivo. Desprezando o tempo computacional, o Dijkstra adequa-se mais ao problema por dar uma solução exata.

Com base na matriz de adjacências, atualizando-a com o *feedback* dos AGV e recorrendo ao Dijkstra obtém-se assim um planeamento da melhor trajetória a efetuar. Em certas situações a melhor trajetória pode não ser a mais curta, mas certamente será a mais adequada no instante em que foi planeada.

## Capítulo 5

# Sistema de Gestão de Trajetórias Inteligente

De forma a otimizar as trajetórias dos AGV em tempo real foi necessário desenvolver um sistema que, a partir da informação proveniente da interface de comunicação descrita no capítulo 3, atribui-se o melhor percurso a cada AGV para satisfazer as necessidades da fábrica. Este sistema consiste num programa que corre num servidor, analisa os pedidos, processa-os e envia as instruções para o AGV estrategicamente escolhido. Foi desenvolvida ainda uma interface web que permite configurar e monitorizar o sistema.

Todo o procedimento encontra-se descrito no presente capítulo.

### 5.1 Arquitetura da Solução Proposta

A solução proposta consiste num sistema composto essencialmente dois programas, um de comunicação com os PLC, um de gestão e planeamento de trajetórias dos AGV e uma interface gráfica de monitorização e controlo. Numa fase inicial a interface de comunicação desenvolvida, e descrita no capítulo 3, faz a leitura das variáveis dos PLC da linha a que está conectado, processa esses dados comparando com a configuração que o utilizador introduziu no programa. Desta forma são enviados os pedidos para uma base de dados MySQL que está armazenada num servidor.

Em paralelo corre um programa desenvolvido em Python que faz a gestão de todo o sistema, explicado no ponto 5.3. Este programa lê os dados presentes na base de dados, efetua a gestão do sistema e procura o AGV mais perto da tarefa a realizar atribuindo o melhor caminho para chegar ao destino. Após a decisão, o sistema envia para o AGV selecionado as indicações estrategicamente escolhidas no servidor.

Para a monitorização das trajetórias dos AGV foi ainda desenvolvida uma interface gráfica para o efeito.

Para uma melhor perceção da arquitetura podemos observar o esquema na figura 5.1.

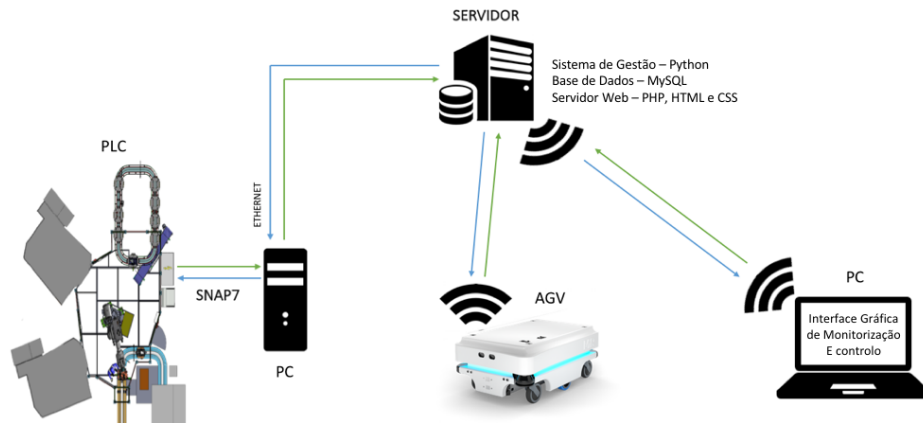


Figura 5.1: Arquitetura da solução proposta.

## 5.2 Base de Dados

Para o armazenamento dos dados do sistema foi criada uma base de dados MySQL. Esta base de dados permite ainda a comunicação entre os diversos sistemas armazenando assim todo o histórico das atividades do sistema. Para um melhor entendimento da sua organização é possível consultar as tabelas 5.1 e 5.2 .

Tabela 5.1: Tabelas de parametrização do sistema.

Correlações (Linha, Pedido 1, Pedido 2, Pedido 3, Pedido 4)  
Troço - Nós (Troço, Nó I, Nó F)  
TAG - Troço (N<sup>o</sup> TAG, Código, Troço)  
Coordenadas (Nó, Web X, Web Y, MiR X, MiR Y, ID Missão)  
Tempos (Troço, T Méd, Delay)  
MAdj (Matriz 43x43)

Na tabela 5.1 pode-se observar as tabelas da base de dados que são indispensáveis para o funcionamento do sistema. Estas tabelas com estas variáveis correspondem à parametrização do sistema. Nesta tabela é também possível obter o nó correspondente a cada pedido de cada linha, isto é, relaciona o tipo de pedido de transporte e a linha proveniente e saber o nó de destino do AGV. Na tabela apresentada como "Troço-Nós" guarda-se a relação entre os nós e os troços, como por exemplo o troço 1 tem nos seus extremos os nós 1 e 2. Na tabela "TAG-Troço" encontra-se a informação sobre as TAGs, o número que é apresentado na superfície do cartão, o código da TAG e a posição da mesma na fábrica. A tabela "Coordenadas" contém as coordenadas de cada nó tanto na interface web como no mapa do AGV. A tabela "Tempos" guarda os tempos médios de cada troço e os excessos de tempo que os AGV demoram comparativamente aos tempos médios. Por último a tabela designada por "MAdj" guarda a matriz de adjacências do sistema, que relaciona uns nós com os outros como explicado no ponto 5.3.

Na tabela 5.2 estão representadas as tabelas da base de dados que permitem a comunicação entre os diversos sistemas e que registam toda a atividade do sistema. A tabela



Tabela 5.2: Tabelas de registo e comunicação do sistema.

AGV (Nº AGV, Modo, Percurso, Coord. X, Coord. Y)  
 Localização (Data, Hora, Nº AGV, Posição, Nó)  
 Pedidos (Data, Hora, Nº Linha, PLC, Tipo Pedido, T. Rutura, Estado, AGV)

"AGV" serve para registar o estado e a posição dos AGV, ou seja, permite saber se estão a realizar alguma tarefa ou se estão livres, o percurso que estão a efetuar e as coordenadas da sua posição na instalação fabril. Na tabela "Localização" encontra-se todas as posições dos AGV bem como a data e hora que foi registada a posição. Na tabela de "Pedidos" encontram-se todos os pedidos solicitados pelas linhas de fabrico. Esta tabela armazena os pedidos das linhas, a data e hora a que foi feito, o tipo de pedido que a linha efetuou, o PLC que enviou o pedido, o tempo de rutura da linha, o estado do pedido e o AGV que irá realizar o transporte.

### 5.3 Programa de Gestão de Trajetórias

No servidor do sistema, para além da base de dados, corre um programa desenvolvido em Python, multiprocessos que permite tratar todos os dados e atribuir as tarefas aos AGV.

Este software usa três processos distintos para corresponder às necessidades do sistema, como esquematizado na figura 5.2.

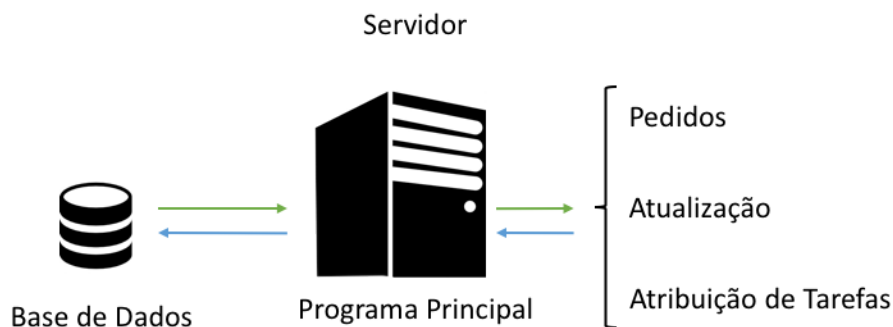


Figura 5.2: Arquitetura do programa de gestão.

#### 5.3.1 Atribuição de Tarefas

Este processo é responsável pela tomada de decisões. Numa fase inicial verifica a existência de pedidos de transportes na base de dados. No caso de existir, procura o AGV disponível mais próximo para realizar a tarefa. Após a seleção do AGV, o processo procura o melhor caminho para este efetuar nesse momento recorrendo ao algoritmo Dijkstra e à tabela de adjacências atualizada.

Após todos os cálculos, são enviadas as instruções para o AGV escolhido. No caso desta dissertação, como o AGV utilizado é o MiR100, utilizou-se a arquitetura REST para efetuar o envio.

É enviado um código que está associado a uma missão para o MiR100. Previamente foi configurado o AGV, como explicado no capítulo 6.2, de forma a associar os códigos a uma missão, que tem como objetivo chegar a um dado nó. Com o envio da sequência de códigos desejada o AGV efetua o caminho planejado. Após o envio das instruções envia-se uma autorização para realizar as tarefas.

A estrutura das mensagens enviadas através de REST encontram-se explicadas no manual da API REST do MiR [32].

A seguinte figura demonstra a ordem das atividades deste processo.

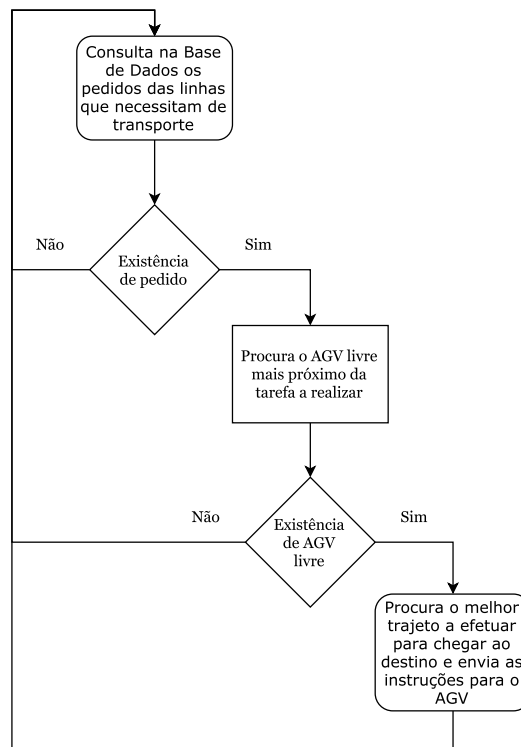


Figura 5.3: Processo de Atribuição de Tarefas do programa principal.

### 5.3.2 Pedidos

O processo "Pedidos" tem como objetivo fazer a monitorização do estado dos pedidos efetuados pelas linhas de fabrico. Este consulta a tabela "Pedidos" da base de dados e verifica que pedidos estão a ser processados. Caso exista pedidos em curso o processo verifica a posição do AGV que lhe foi atribuído previamente e confirma se já chegou ao seu destino. Caso se verifique que o AGV chegou ao destino dá-se por concluído o transporte. Neste caso o estado do pedido altera para finalizado e o estado do AGV é alterado para que ele fique disponível para novos transportes. Caso não se verifique o término do transporte, as variáveis mantêm-se inalteráveis. No diagrama da figura 5.4 podemos ver um fluxograma do algoritmo de verificação.

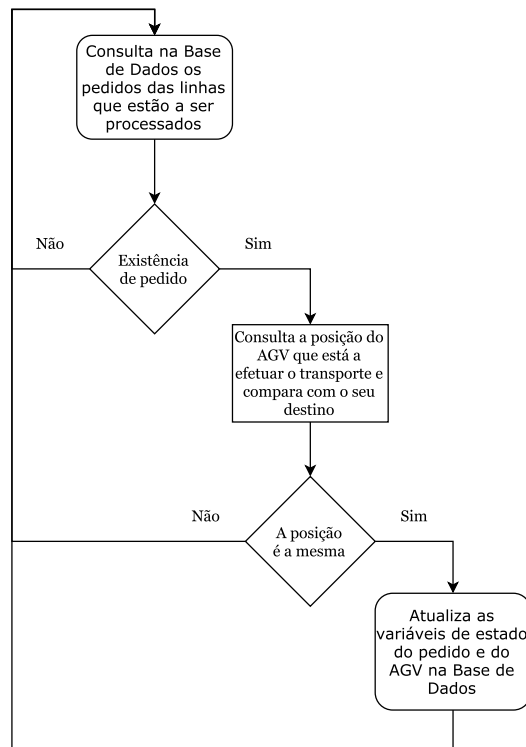


Figura 5.4: Processo de Pedidos do programa principal.

### 5.3.3 Atualização

Este processo tem quatro tarefas principais: gerir o sistema, analisar as paragens dos AGV, verificar perturbações nos troços com base no tempo médio e analisar possíveis sobrecargas nos troços com base no número de AGV planeados a passar em cada troço.

Na gestão do sistema este processo consulta as tabelas da base de dados e completa-as com as configurações previamente definidas nas tabelas de parametrização. Na tabela onde se encontra a localização dos AGV, o próprio hardware envia o código da TAG que acabou de ler. No entanto não tendo ainda a informação do troço em que se encontra, este processo compara o código com a informação da tabela "Tag Troço" e completa a tabela "Localização". Juntamente com esta atualização o programa altera ainda as coordenadas de cada AGV com base na sua localização e na tabela "Coordenadas". Estas etapas vêm simplificar os algoritmos de atribuição de tarefas e a monitorização do sistema.

O algoritmo de análise de paragens dos AGV analisa as duas últimas posições de cada AGV e, caso sejam iguais, considera que o AGV está parado. Se o AGV estiver parado o algoritmo incrementa a matriz de adjacências a cada 5 segundos na posição correspondente ao troço onde está. Este incremento consiste no aumento da variável "Paragens" da formula 4.2. No caso de este estar em andamento, altera o valor correspondente para o valor inicial calculado previamente.

Para além da análise das paragens dos AGV, este processo verifica também possíveis perturbações nos troços com base no tempo médio que os AGV demoram a efetuar o troço. O algoritmo calcula o tempo que o AGV demorou a executar o troço, compara-o com o tempo médio normal e caso seja superior incrementa o custo do troço com base

no tempo a mais que demorou a atravessar.

Por fim, o processo verifica os troços por onde cada AGV vai passar e adiciona 30 ao custo de cada troço por cada AGV que está previsto passar por este. Após a passagem estes incrementos voltam a zero.

Desta forma, ao fim de pouco tempo conseguimos evitar que o sistema envie os AGV para o troço que está sobrecarregado ou mesmo interrompido.

## 5.4 Interface Gráfica

De modo a facilitar a monitorização e o controlo do sistema de gestão foi desenvolvida uma interface que permite a visualização gráfica das posições dos AGV, as tabelas de monitorização do sistema e ainda a alteração das tabelas de parametrização do sistema.

Para a realização da interface foram utilizadas as linguagens HTML e CSS para o aspeto gráfico da interface e, JavaScript e PHP para efetuar a gestão dos elementos da interface e comunicação entre esta e a base de dados. O JavaScript corre no *browser* do utilizador e o PHP no servidor.

A interface encontra-se dividida em três paginas principais: *Overview* que permite monitorizar alguns parâmetros do sistema, *Map* onde se pode observar os percursos planeados para cada AGV bem como a posição real do mesmo e *Configurations* onde é possível parametrizar o sistema consoante a instalação fabril e os equipamentos disponíveis.

### 5.4.1 Overview

No primeiro separador da interface gráfica, presente na figura 5.5, é possível consultar os pedidos efetuados pelas linhas de fabrico. Nesta tabela da interface gráfica é possível ver a data e hora a que se desencadeou o pedido, o tipo de pedido que é, a prioridade do mesmo, o seu estado e ainda o AGV que efetuou ou está a efetuar o transporte.

Neste mesmo separador é possível ainda consultar o estado de cada AGV, figura 5.6, e no caso de estarem a efetuar uma trajetória podemos ainda ver em que nós este irá passar.

Por fim, existe uma outra tabela que permite visualizar os atrasos existentes em cada troço como se pode observar na figura 5.7.

Date	Time	Line	Request Type	Rupture Time	State	AGV
2017-06-07	18:39:26	1	Finished	13	Finished	111
2017-06-07	18:35:26	1	Finished	13	Finished	111
2017-06-07	18:34:46	1	Raw	18	Waiting	No AGV assigned
2017-06-07	18:31:26	1	Finished	13	Finished	111
2017-06-07	18:28:06	1	Raw	18	On Going	111

Figura 5.5: *Overview* - Pedidos das linhas.

Requests AGVs Delays

### AGVs

Show 10 entries Search:

AGV	State	Path
111	Free	
305	Free	
394	Free	
403	Free	
405	Free	
832	Free	

Showing 1 to 6 of 6 entries Previous 1 Next

Figura 5.6: *Overview* - Estado dos AGV.

Requests AGVs Delays

### Delays

Show 10 entries Search:

Edge	Delay
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0

Showing 1 to 10 of 20 entries (filtered from 10 total entries) Previous 1 2 Next

Figura 5.7: *Overview* - Tempos médios e atrasos nos troços.

### 5.4.2 Map

No separador *Map* é possível monitorizar em tempo real a posição dos AGV no mapa do edifício de fabrico dos componentes mecânicos da fábrica de CACIA - Renault. Esta página permite também verificar se os AGV se encontram ou não conectados ao sistema. Caso não estejam, estes aparecem no lado direito da janela fora do mapa.

Se os AGV estiverem a efetuar um percurso, estes aparecem também representados com uma linha por onde vão passar, um marcador verde de início e um vermelho de destino.

Na figura figura 5.8 pode-se visualizar um exemplo efetuado durante os testes. Nesta mesma figura é possível visualizar o aspeto gráfico geral da interface, à esquerda os separadores principais e na parte superior o logotipo da Renault com a designação interna do sistema dada pela fábrica de CACIA - Renault.

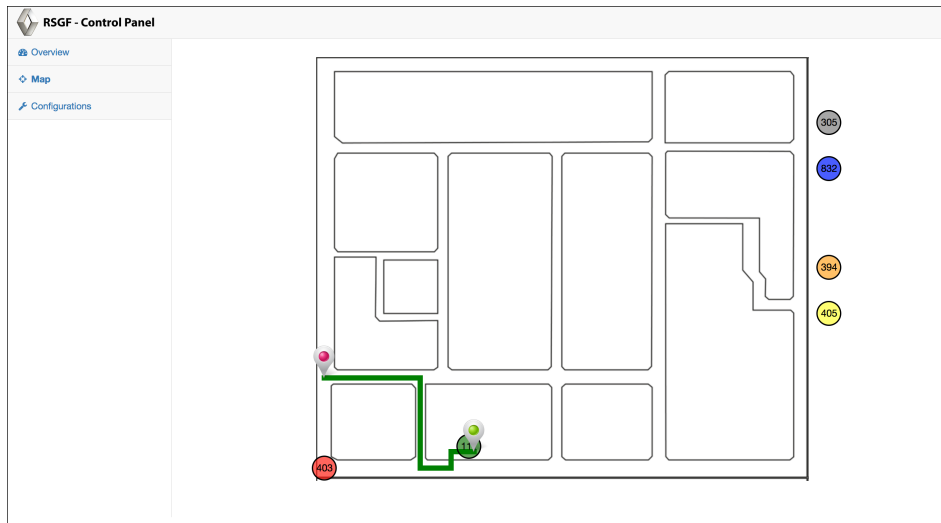


Figura 5.8: Separador *Map* da interface gráfica.

### 5.4.3 *Configurations*

Neste separador é possível consultar e alterar as tabelas de configuração do sistema. Através destas páginas pode-se alterar as tabelas da base de dados MySQL através de botões e caixas de texto, o que o torna acessível a qualquer pessoa.

Na sua abertura, figura 5.9 é permitido configurar os nós e os caminhos do mapa. Desta forma, com base no mapa apresentado na figura 4.3 pode-se configurar o sistema informando-o das relações entre os nós e os caminhos.

A tabela *Lines*, figura 5.10, permite informar o sistema da localização da linha no mapa. Através desta pode-se informar qual o nó a que o AGV se deve dirigir dependendo da linha e do tipo de pedido que a linha efetua.

Recorrendo à tabela *TAGs* presente na figura 5.11 é possível associar os códigos das TAG RFID existentes no solo da fábrica a um caminho presente no mapa da figura 4.3. Dado que essas TAG têm um número gravado para fácil identificação por parte do utilizador, esta tabela permite ainda adicionar esse mesmo número.

Na tabela *Times* é possível informar o tempo médio de cada troço e alterar os atrasos existentes, como por exemplo repor o valor a zero. Na figura 5.12 é possível consultar um exemplo desta tabela.

Por último, a tabela *Coordinates*, figura 5.13, permite configurar as coordenadas de cada nó, tanto no mapa do MiR bem como no mapa da interface gráfica. Apesar das coordenadas de cada nó, é possível ainda guardar o código de identificação da missão que permite ir para esse nó.

Paths Lines TAGs Times Coordinates

### Paths

Show 10 entries Search:

Path	Start Node	End Node	Edit	Delete
60	21	22	<a href="#">Update</a>	<a href="#">Delete</a>
59	43	42	<a href="#">Update</a>	<a href="#">Delete</a>
58	42	41	<a href="#">Update</a>	<a href="#">Delete</a>
57	40	43	<a href="#">Update</a>	<a href="#">Delete</a>
56	37	42	<a href="#">Update</a>	<a href="#">Delete</a>
55	41	36	<a href="#">Update</a>	<a href="#">Delete</a>
54	40	39	<a href="#">Update</a>	<a href="#">Delete</a>
53	39	38	<a href="#">Update</a>	<a href="#">Delete</a>
52	38	37	<a href="#">Update</a>	<a href="#">Delete</a>
51	31	37	<a href="#">Update</a>	<a href="#">Delete</a>

Showing 1 to 10 of 60 entries (filtered from 10 total entries) Previous 1 2 3 4 5 6 Next

Figura 5.9: *Configurations* - Definição dos caminhos possíveis.

Paths Lines TAGs Times Coordinates

### Lines

Show 10 entries Search:

Lines	Raw Parts	Finished Parts	Scrap Parts	Recovered Parts	Edit	Delete
1	8	17	13	10	<a href="#">Update</a>	<a href="#">Delete</a>
13	25	34	42	30	<a href="#">Update</a>	<a href="#">Delete</a>

Showing 1 to 2 of 2 entries Previous 1 Next

Figura 5.10: *Configurations* - Posicionamento das linhas no mapa.

Paths Lines TAGs Times Coordinates

### TAGs

Show 10 entries Search:

Number	TAG	Edge	Edit	Delete
0	[136, 4, 209, 231, 186]	2	<a href="#">Update</a>	<a href="#">Delete</a>
0	[94, 102, 46, 32, 54]	6	<a href="#">Update</a>	<a href="#">Delete</a>
1	[228, 130, 17, 224, 151]	32	<a href="#">Update</a>	<a href="#">Delete</a>
2	[180, 254, 243, 169, 16]	19	<a href="#">Update</a>	<a href="#">Delete</a>
3	[180, 214, 188, 223, 1]	18	<a href="#">Update</a>	<a href="#">Delete</a>
4	[100, 219, 44, 170, 57]	13	<a href="#">Update</a>	<a href="#">Delete</a>
5	[164, 26, 174, 223, 207]	13	<a href="#">Update</a>	<a href="#">Delete</a>
6	[28, 152, 172, 21, 61]	10	<a href="#">Update</a>	<a href="#">Delete</a>
7	[139, 91, 192, 3, 19]	8	<a href="#">Update</a>	<a href="#">Delete</a>
8	[123, 67, 192, 3, 251]	8	<a href="#">Update</a>	<a href="#">Delete</a>

Showing 1 to 10 of 47 entries (filtered from 10 total entries) Previous 1 2 3 4 5 Next

Figura 5.11: *Configurations* - Definição e posicionamento das TAG RFID.

Paths Lines TAGs Times Coordinates

### Times

Show 10 entries Search:

Edge	Average	Delay	Edit	Delete
60	10	0	<a href="#">Update</a>	<a href="#">Delete</a>
59	205	0	<a href="#">Update</a>	<a href="#">Delete</a>
58	72	0	<a href="#">Update</a>	<a href="#">Delete</a>
57	38	0	<a href="#">Update</a>	<a href="#">Delete</a>
56	57	0	<a href="#">Update</a>	<a href="#">Delete</a>
55	57	0	<a href="#">Update</a>	<a href="#">Delete</a>
54	60	0	<a href="#">Update</a>	<a href="#">Delete</a>
53	55	0	<a href="#">Update</a>	<a href="#">Delete</a>
52	74	0	<a href="#">Update</a>	<a href="#">Delete</a>
51	57	0	<a href="#">Update</a>	<a href="#">Delete</a>

Showing 1 to 10 of 60 entries (filtered from 10 total entries) Previous 1 2 3 4 5 6 Next

Figura 5.12: *Configurations* - Definição dos tempos médios

Paths Lines TAGs Times Coordinates

### Nodes Properties

Show 10 entries Search:

Node	WEB X	WEB Y	MIR X	MIR Y	Mission ID	Edit	Delete
1	738	625	0	0	361c72b9-5741-11e7-ad76-b8aeed719c1f	<a href="#">Update</a>	<a href="#">Delete</a>
2	522	625	0	0	48bf13c4-5741-11e7-ad76-b8aeed719c1f	<a href="#">Update</a>	<a href="#">Delete</a>
3	365	625	67	-1	055256ff-574b-11e7-9c0b-b8aeed719c1f	<a href="#">Update</a>	<a href="#">Delete</a>
4	280	625	43	-1	0b324943-574b-11e7-9c0b-b8aeed719c1f	<a href="#">Update</a>	<a href="#">Delete</a>
5	205	625	36	-1	0fb65ebd-574b-11e7-9c0b-b8aeed719c1f	<a href="#">Update</a>	<a href="#">Delete</a>
6	160	625	32	-1	1317204e-574b-11e7-9c0b-b8aeed719c1f	<a href="#">Update</a>	<a href="#">Delete</a>
7	12	625	5	0	19a090c2-574b-11e7-9c0b-b8aeed719c1f	<a href="#">Update</a>	<a href="#">Delete</a>
8	240	600	39	1	1cb83ba3-574b-11e7-9c0b-b8aeed719c1f	<a href="#">Update</a>	<a href="#">Delete</a>
9	158	600	0	0	25f3dbe5-574b-11e7-9c0b-b8aeed719c1f	<a href="#">Update</a>	<a href="#">Delete</a>
10	178	585	0	0	2968e82d-574b-11e7-9c0b-b8aeed719c1f	<a href="#">Update</a>	<a href="#">Delete</a>

Showing 1 to 10 of 32 entries (filtered from 10 total entries) Previous 1 2 3 4 Next

Figura 5.13: *Configurations* - Propriedades dos nós.



## Capítulo 6

# Equipamentos para Testes e Resultados Experimentais

Dadas as limitações do hardware existente bem como as limitações no acesso a este, pois o sistema dos AGV da Cmajor não tem a possibilidade de interação com sistemas externos, foi necessário desenvolver um equipamento que permitisse monitorizar a posição dos AGV como explicado no ponto 6.1. Para a realização dos testes e validar o conceito foi utilizado um AGV MiR100. Para a integração do AGV no sistema desenvolvido foi necessário configurá-lo para receber instruções através da arquitetura REST. Todo o processo encontra-se explicado no ponto 6.2.

Neste capítulo são ainda apresentados os resultados experimentais.

### 6.1 Dispositivo de Monitorização da Posição dos AGV

Como o sistema de localização dos AGV da fábrica de CACIA - Renault é através da leitura de cartões RFID foi criado um dispositivo que efetua a leitura dos cartões e envia o código lido para a base de dados.

Este sistema é constituído por uma caixa de plástico estanque o que serve de suporte para os componentes, um Raspberry Pi 3, um leitor de RFID MFRC522, conectores e uma *power bank* que alimenta o sistema. A disposição do hardware pode ser observada na figura 6.2.



Figura 6.1: Dispositivo de monitorização dos AGV.

O Leitor RFID encontra-se conectado ao Raspberry segundo o esquema seguinte:

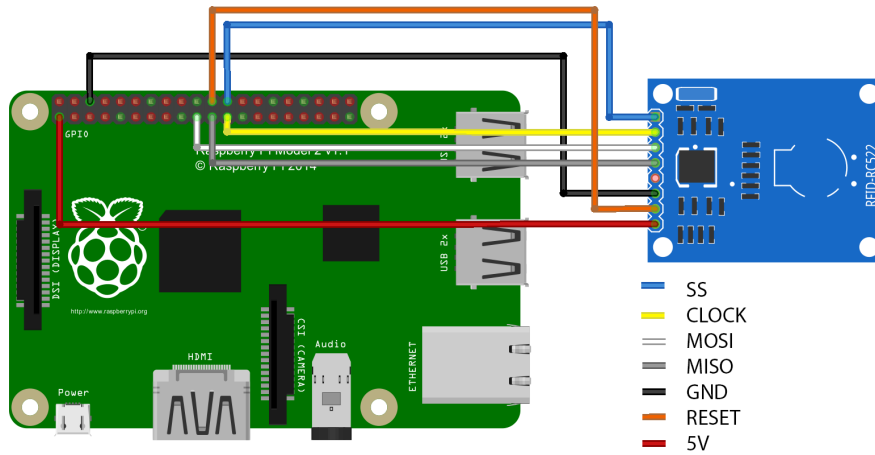


Figura 6.2: Esquema de ligações entre o leitor de RFID e o Raspberry Pi [33].

Para a programação do Raspberry foi previamente instalado o sistema operativo oficial Raspbian, muito similar ao Debian, mas otimizado para o Raspberry [34].

Para cumprir o objetivo deste sistema foi desenvolvido um programa em Python que efetua a leitura das TAG RFID, quando o AGV passa por elas e envia para a base de dados que está no servidor através do Wi-Fi integrado no Raspberry.

No caso do AGV ficar parado na TAG o software, a cada 5 segundos, envia novamente a sua posição para informar que está parado. Esta situação não é a ideal, pois existe a possibilidade do AGV parar fora de uma TAG e o sistema não receber essa informação. No entanto, sendo este um equipamento temporário e unicamente para a realização de testes, esta adversidade não é relevante.

Na figura 6.3 é possível observar a montagem do sistema nos AGV da fábrica de CACIA - Renault.



Figura 6.3: Montagem do sistema desenvolvido num AGV Cmajor.

## 6.2 Configuração do AGV MiR100

Para a utilização dos AGV MiR100 é necessário numa fase inicial efetuar o reconhecimento da área em que este vai navegar. Desta forma foi necessário criar um novo mapa no robô com as informações do departamento de fabricação de componentes mecânicos da Renault CACIA. Para isso e, através da interface gráfica do MiR, criou-se um novo mapa navegando com o AGV de modo a ele recolher a informação através dos seus lasers, sensores ultra-sons e da câmara 3D.

Para este processo o MiR 100 cruza os dados recebidos pelos lasers, ultra-sons e pela câmara com a parte do mapa já lido e vai construindo o mapa comparando os dados recolhidos com os conhecidos. Através dos dados conhecidos o AGV consegue localizar-se no seu mapa. Sabendo a sua posição completa o mapa com a informação que está a receber e ainda não guardou. Juntamente com este sistema de localização, o MiR 100 usa os *encoders* para ajudar na localização.

Para a realização de testes foi apenas mapeada uma zona mais pequena do departamento devido ao tempo que a tarefa demora a ser efetuada e ao facto de o tempo de acesso ao equipamento ser limitado.

Após terminar o reconhecimento e o próprio AGV processar os dados foi possível chegar a um resultado que está representado na figura 6.4.

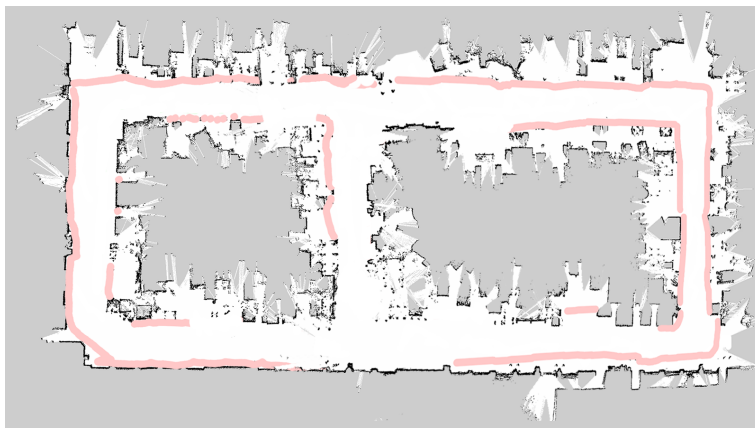
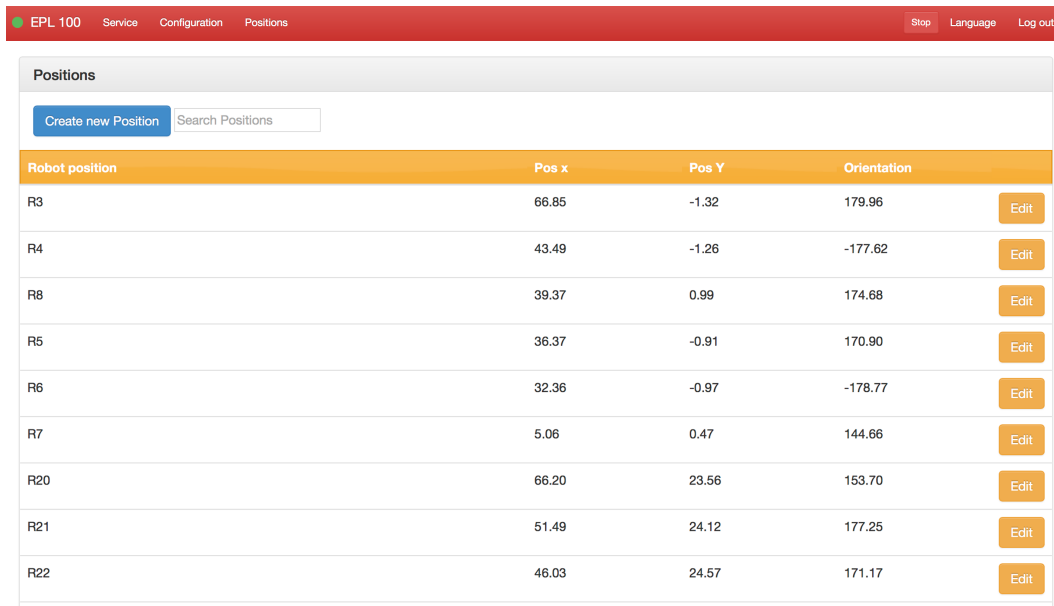


Figura 6.4: Mapa obtido pelo MiR100.

Após esta etapa concluída, foi necessário configurar os pontos de interesse no mapa. Novamente, através da interface gráfica, o AGV foi movido para o ponto de interesse. Foi guardada a sua posição e orientação e, associou-se o número do nó correspondente com base no mapa da figura 4.3. Este processo repetiu-se para todos os nós.

O resultado obtido foi uma lista de nós com as coordenadas que estes têm no mapa e a sua orientação como demonstra a figura 6.5

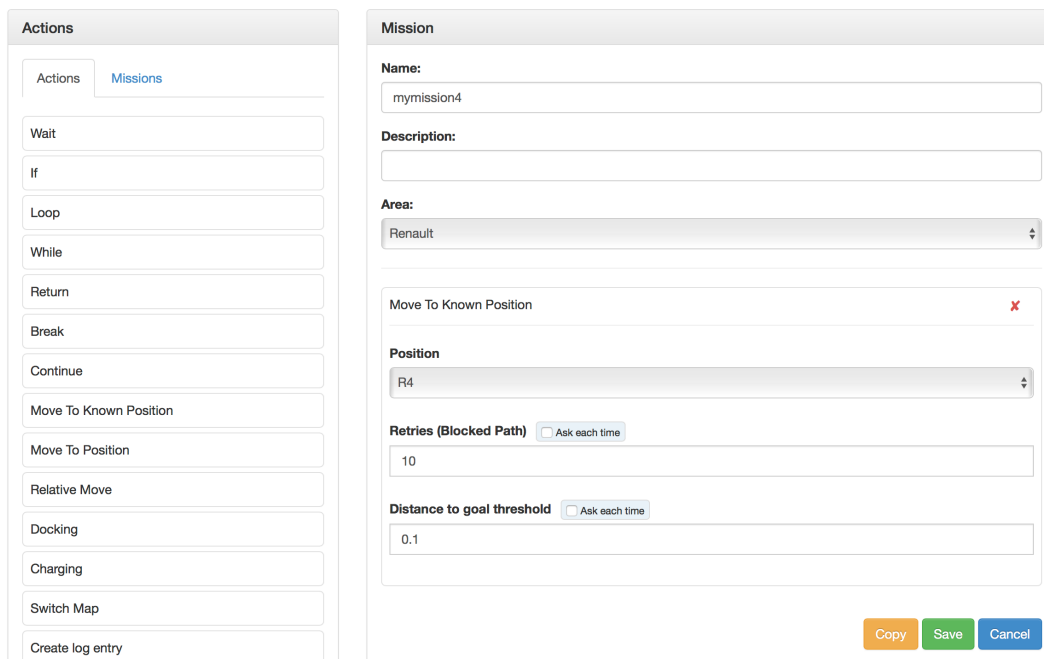


Robot position	Pos x	Pos Y	Orientation
R3	66.85	-1.32	179.96
R4	43.49	-1.26	-177.62
R8	39.37	0.99	174.68
R5	36.37	-0.91	170.90
R6	32.36	-0.97	-178.77
R7	5.06	0.47	144.66
R20	66.20	23.56	153.70
R21	51.49	24.12	177.25
R22	46.03	24.57	171.17

Figura 6.5: Lista dos pontos de interesse configurados no MiR100.

Após a criação dos pontos no AGV foi necessário criar missões em que o objetivo é chegar a esses mesmos pontos. Desta forma para cada ponto criou-se uma missão cujo o objetivo é chegar a esse ponto conhecido.

Novamente através da interface do AGV efetuou-se esta tarefa como demonstra a figura 6.6.



**Actions**

Actions Missions

Wait

If

Loop

While

Return

Break

Continue

Move To Known Position

Move To Position

Relative Move

Docking

Charging

Switch Map

Create log entry

**Mission**

**Name:** mymission4

**Description:**

**Area:** Renault

Move To Known Position

**Position:** R4

**Retries (Blocked Path)**  Ask each time: 10

**Distance to goal threshold**  Ask each time: 0.1

Copy Save Cancel

Figura 6.6: Configuração das missões no MiR100.

Por último, foi necessário aceder aos códigos de identificação de cada missão. Estes são códigos que o programa de atribuição de tarefas envia para o AGV para criar um percurso para este efetuar. Para aceder a esses códigos recorreu-se a um pedido REST, novamente com base na informação disponibilizada no manual da API REST do MiR [32]. Após o conhecimento dos códigos de identificação foram adicionados à base de dados através da interface gráfica desenvolvida.

Na figura seguinte, é possível observar um exemplo dos dados recebidos após o envio do pedido de códigos de identificação das missões.

```
{
  "guid": "361c72b9-5741-11e7-ad76-b8aeed719c1f",
  "name": "mymission1",
  "url": "/v1.0.0/missions/361c72b9-5741-11e7-ad76-b8aeed719c1f"
},
```

Figura 6.7: Exemplo da estrutura de dados recebida do MiR.

## 6.3 Resultados Experimentais

Na fase final da dissertação foram realizados diversos testes de funcionalidade e robustez do trabalho desenvolvido.

Para a realização de testes foi usado um AGV MiR100 e um AGV Cmajor com o dispositivo de monitorização desenvolvido.

### 6.3.1 Funcionalidade da Interface de Comunicação com Controladores Industriais

De forma a avaliar o desempenho da interface de comunicação com os PLC Siemens foram feitos diversos testes.

Numa fase inicial testou-se a funcionalidade do sistema recorrendo a todos os recursos do programa. Realizou-se testes usando apenas o PLC de entrada, o PLC de saída e com dois PLC em simultâneo.

Numa fase posterior foram realizados testes de robustez. Com este propósito o programa ficou diversas vezes em funcionamento até verificar a recorrência de falha. Depois de alguns testes concluiu-se que devido a problemas de cobertura de Wi-Fi o programa bloqueava. Esta falha estava relacionada com a perda de comunicação com a base de dados. Após a resolução deste problema o programa ficou em funcionamento mais de 48 horas seguidas sem ocorrer qualquer falha.

Por último, o sistema ficou em funcionamento para a realização de testes das outras ferramentas desenvolvidas o que comprovou o referido acima.

### 6.3.2 Funcionalidade do Sistema de Planeamento e Gestão de Trajetórias

À semelhança do programa anterior foram desenvolvidos testes de funcionalidade e robustez, no entanto estes foram realizados em simultâneo.

Durante os testes, o sistema não apresentou nenhuma falha. Como esperado respondia a todos os pedidos efetuados pela interface de comunicação com os controladores,

planeava a trajetória do AGV selecionado e efetuava todas as funcionalidades descritas anteriormente.

Como a base de dados se encontrava no mesmo equipamento que este programa, não existiram problemas de comunicação entre elas, no entanto, numa situação futura esta estará localizada num servidor. Para que o programa não bloqueie inesperadamente num caso de falha de comunicação foram adicionados alguns mecanismos de deteção de falha para atuar nesses casos.

### 6.3.3 Planeamento de Trajetórias

Como previsto no capítulo 4 esperava-se que o programa respondesse a diversas adversidades nos transportes, nomeadamente a AGV parados, atrasos nas linhas, AGV planeados nos caminhos entre os critérios base de planeamento como distância, taxa de utilização e se o caminho é principal ou não.

De forma a testar todas estas necessidades foram realizados testes específicos para cada questão.

Como referido anteriormente, os testes foram efetuados numa zona reduzida da fábrica, o que limitou estes testes, no entanto houve um esforço para comprovar o funcionamento de todas as respostas do sistema.

Nas figuras 6.8 e 6.9 é possível observar através da interface gráfica, a resposta do sistema de planeamento no caso de um AGV estar parado numa dada zona da instalação fabril. Neste caso o programa teve de encontrar uma alternativa que é mais distante todavia preferível no momento da realização da tarefa.

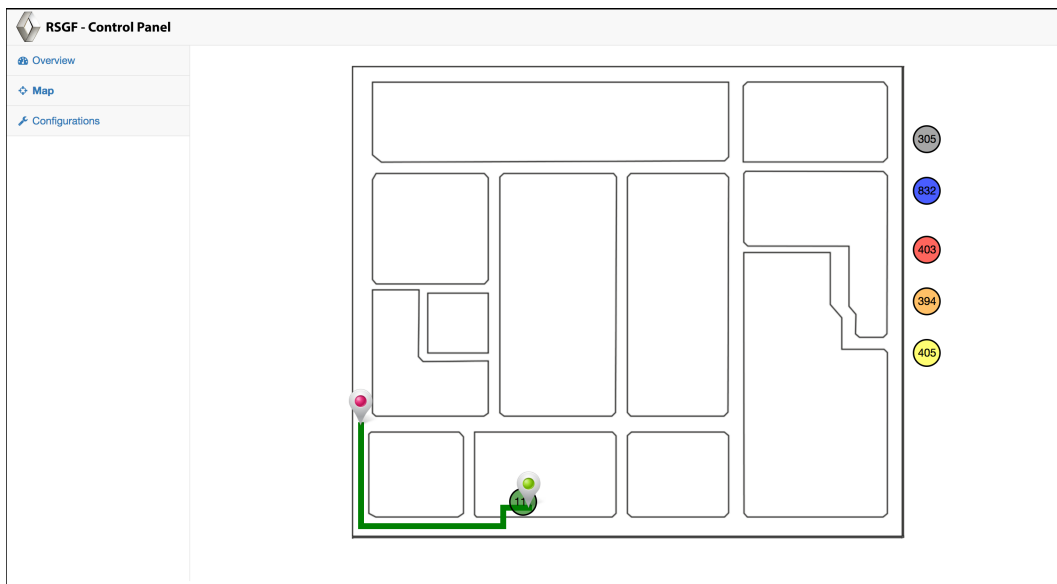


Figura 6.8: Planeamento sem paragem de AGV.

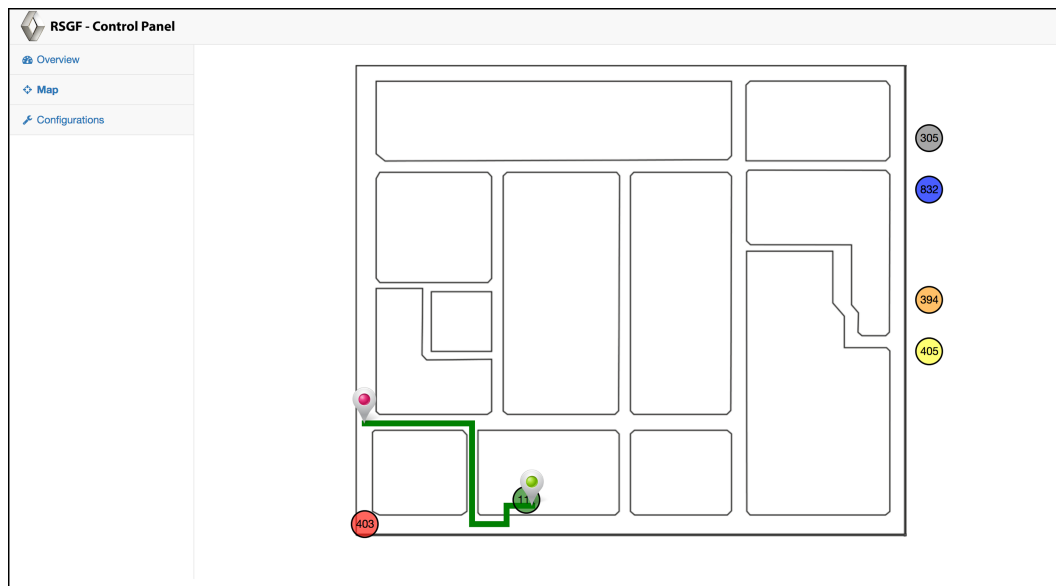


Figura 6.9: Planeamento com paragem de AGV.

Outros testes foram efetuados para comprovar o planeamento com base no tempo médio e no número de AGV planeados no caminho. No entanto para estes testes, dadas as limitações dos mesmos, tiveram que ser feitas simulações de paragens de AGV manualmente na base de dados.





## Capítulo 7

# Conclusões e Trabalho Futuro

Este capítulo tem por objetivo fazer uma apreciação global ao trabalho desenvolvido. São apresentadas as conclusões do trabalho e ainda sugestões de trabalhos futuros de modo a dar continuidade ao trabalho.

### 7.1 Conclusões

Os principais objetivos para esta dissertação eram o desenvolvimento de uma interface de comunicação com controladores industriais, especificamente Siemens SIMATIC e o desenvolvimento de um sistema de gestão e planeamento de trajetórias de AGV.

Para o desenvolvimento da interface de comunicação com os PLC Siemens foi utilizado o Visual Basic devido à sua fácil implementação e robustez. Dado que os computadores existentes nas linhas de produção tinham o sistema operativo Windows, a linguagem de programação foi a adequada. A utilização da biblioteca SNAP7 foi fundamental no desenvolvimento desta ferramenta, pois facilitou a comunicação com o hardware. Para a realização de testes foi utilizada uma rede Wi-Fi, no entanto, dado que este programa estará num computador fixo na linha é preferível estabelecer a comunicação via *Ethernet*. Através deste meio a comunicação é mais fiável diminuindo as falhas de comunicação com a base de dados. Os restantes objetivos desta interface foram todos alcançados, podendo-se considerar uma aplicação final pronta para entrar em funcionamento na fábrica.

Relativamente ao sistema de gestão e planeamento de trajetórias dos AGV, o conceito foi provado com sucesso. O sistema de gestão de AGV desenvolvido apresenta inúmeras vantagens comparativamente ao existente. Dado que este sistema não depende da intervenção humana são reduzidos diversos problemas tais como o esquecimento de efetuar o pedido ou mesmo esquecimento de anular o pedido após este ser efetuado. Outras vantagens surgem com este sistema dado que este se baseia em software e não requer hardware adicional ao existente, tornando o custo de implementação menor.

O planeamento de trajetórias com base nos critérios escolhidos satisfazem as necessidades da fábrica. Com este planeamento é possível efetuar os transportes com normalidade mesmo que haja diversos caminhos com perturbações, ou com AGV parados ou com atrasos na passagem pelos caminhos. Devido à previsão da ocupação dos caminhos é possível evitar a sobrecarga dos mesmos, o que torna o fluxo dos AGV mais distribuído e desta forma reduz o impacto nos transportes devido a possíveis perturbações.

No desenvolvimento deste projeto surge a necessidade de criar uma interface gráfica

de monitorização e controlo do sistema. Esta ferramenta veio facilitar a interação entre o utilizador e o sistema desenvolvido. Dado o crescimento de aplicações baseadas no *Internet of Things* e na Indústria 4.0 esta interface foi desenvolvida para ser acedida em qualquer dispositivo. Uma vez que a Renault possui *Virtual Private Network* (VPN) estando esta interface no servidor da fábrica de CACIA - Renault torna possível o acesso em qualquer parte do mundo.

Em suma, com a solução desenvolvida a Renault CACIA irá beneficiar de uma solução de baixo custo bem como melhorar o sistema de fluxo dos materiais. Esta solução irá permitir uma gestão automática e inteligente dos AGV pelo facto de usar o *feedback* dos equipamentos para tomar decisões. De salientar que, para beneficiar de todas as potencialidades do sistema é necessário a utilização de AGV adequados, é recomendado usar robôs que não necessitem de guias para se orientarem e que permitam estabelecer uma comunicação através de uma API ou algum sistema semelhante. Desta forma a Renault CACIA irá ter um planeamento favorável em tempo real.

## 7.2 Trabalho Futuro

Após o desenvolvimento deste projeto ficam ainda alguns desenvolvimentos pendentes. Desta forma sugere-se as seguintes tarefas para dar continuidade ao trabalho:

### 1 Fiabilização do sistema desenvolvido

Embora tenham sido realizados diversos testes ao sistema é necessário testar todo o sistema de gestão dos AGV para a totalidade da fábrica. O sistema foi testado controlando um AGV e monitorizando os Cmajor no entanto, quando o sistema entrar em funcionamento todos os AGV serão controlados e não monitorizados, pelo que o sistema deverá ser testado com todo o hardware final.

### 2 Desenvolvimento de uma interface de monitorização e controlo

Apesar de ter sido desenvolvida uma interface de monitorização e controlo, esta foi direcionada para os testes que foram realizados. No entanto, para tornar este sistema numa solução final e poder ser controlado por qualquer utilizador final é necessário tornar toda a configuração mais intuitiva, desenvolver um sistema de autenticação, atribuir níveis de acesso distintos, entre outras funcionalidades a discutir com a Renault CACIA.

### 3 Criação de diferentes níveis de urgências nos transportes

Por vezes existem transportes mais urgentes que outros devido a diversos fatores, como por exemplo rutura de componentes numa linha ou avaria de um equipamento. Para o sistema ter em atenção estes casos é necessária a adição da priorização dos transportes com base na urgência. Nestes casos o sistema pode atuar de várias formas, como por exemplo atribuir um AGV primeiro a este pedido ou ainda condicionar as rotas dos outros AGV para que este possa ir pelo trajeto mais rápido e não se cruzar com outro AGV.

#### **4 Implementação do sistema na fábrica de CACIA - Renault**

Dado que este projeto é em conjunto com a Renault CACIA é fundamental a implementação do sistema na instalação fabril. Após a realização das tarefas em falta é necessário implementar a solução.



# Bibliografia

- [1] *Snap7 Homepage*. URL: <http://snap7.sourceforge.net/> (acedido em 29/05/2017).
- [2] *MySQL | The Most Popular Open-Source Database*. URL: <https://www.oracle.com/mysql/index.html> (acedido em 29/05/2017).
- [3] *Raspberry Pi 3 Model B*. URL: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> (acedido em 29/05/2017).
- [4] *Raspberry Pi*. Fev. de 2015. URL: <https://www.raspberrypi.org/blog/raspberry-pi-2-on-sale/> (acedido em 29/05/2017).
- [5] NXP Semiconductors N.V. *MFRC522 Data Sheet*. URL: [https://www.nxp.com/documents/data\\_sheet/MFRC522.pdf](https://www.nxp.com/documents/data_sheet/MFRC522.pdf).
- [6] *Arduino RFID Reader / MFRC522 Tutorial: 3 Steps (with Pictures)*. URL: <http://www.instructables.com/id/Arduino-RFID-Reader-MFRC522-Tutorial/> (acedido em 29/05/2017).
- [7] Siemens AG. *brochure\_simatic-et200\_en.pdf*. en.
- [8] *Interface Module for ET 200S*. URL: <https://support.industry.siemens.com/cs/document/30851575> (acedido em 29/05/2017).
- [9] Mobile Industrial Robots. *MiR100 | Products | Mobile Industrial Robots*. URL: <http://www.mobile-industrial-robots.com/products/mir100> (acedido em 29/05/2017).
- [10] Z. Katz e G. Bright. "A guidance technique for an automated guided vehicle". en. Em: *The International Journal of Advanced Manufacturing Technology* 7.4 (), pp. 198–202. ISSN: 0268-3768, 1433-3015. DOI: 10.1007/BF02601624. URL: <https://link.springer.com/article/10.1007/BF02601624> (acedido em 29/05/2017).
- [11] B.D. Douglas e J.A. Bencel. *Navigation system for automatic guided vehicle*. Google Patents, abr. de 2000. URL: <https://www.google.com/patents/US6049745>.
- [12] *Solutions | Gotting KG*. URL: <http://www.goetting-agv.com/solutions> (acedido em 29/05/2017).
- [13] *AGV-AGC Vehicle Navigation*. URL: <http://www.agvsystems.com/wp-content/uploads/2013/02/Nav.pdf> (acedido em 13/04/2017).
- [14] R.D. Dyer et al. *Optical guidance system for industrial vehicles*. Google Patents, ago. de 1990. URL: <https://www.google.com/patents/US4947094>.
- [15] *AGVS Basic Information*. Ago. de 2012. URL: <http://www.agvsystems.com/agvs-basics/basics-agvs/> (acedido em 29/05/2017).

- [16] Sik Hong Youn. *An Indoor Location Tracking using Wireless Sensor*. URL: <http://www.indjst.org/index.php/indjst/article/viewFile/60758/47493> (acedido em 30/05/2017).
- [17] A Carvalho A Azenha. *Dynamic analysis of AGV control under dead-reckoning algorithm*. URL: [http://journals.cambridge.org/article\\_S0263574708004244](http://journals.cambridge.org/article_S0263574708004244) (acedido em 30/05/2017).
- [18] A.D. Tews.n F. Duvallet. *WiFi Position Estimation in Industrial Environments Using Gaussian Processes. Em Intelligent Robots and Systems, 2008*. (Acedido em 30/05/2017).
- [19] Dr. Sunita Chaunhan Azhar Dilshad. *Latest Advancements in Mobile Robot Localization in Manufacturing Environment 2010*. (Acedido em 30/05/2017).
- [20] *What is OPC*. URL: <https://opcfoundation.org/about/what-is-opc/> (acedido em 30/07/2017).
- [21] *Data Access API*. URL: <https://opcfoundation.org/developer-tools/developer-kits-classic> (acedido em 03/06/2017).
- [22] *Active OPC Server | OPC Server | Moxa*. URL: [http://www.moxa.com/product/automation\\_software\\_html](http://www.moxa.com/product/automation_software_html) (acedido em 03/06/2017).
- [23] *Roteirizacao de veiculos para o abastecimento de linhas de producao*. URL: <http://www.teses.usp.br/teses/disponiveis/3/3148/tde-10062013-145714/> (acedido em 03/06/2017).
- [24] *Multi-Robot Cooperation Systems for Assembly Automobile Industry*. URL: [https://sigarra.up.pt/feup/pt/pub\\_geral.pub\\_view?pi\\_pub\\_base\\_id=31820](https://sigarra.up.pt/feup/pt/pub_geral.pub_view?pi_pub_base_id=31820) (acedido em 18/06/2017).
- [25] *JBT AGVs - AGV System Software - SGV Manager*. URL: <http://www.jbtc-agv.com/en/Solutions/Products/AGV%20System%20Software> (acedido em 03/06/2017).
- [26] E. W. Dijkstra. *A note on two problems in connexion with graphs, Numerische Mathematik, vol. 1, no. 1, pp. 269-271, 1959*. (Acedido em 03/06/2017).
- [27] *Dijkstra's Shortest Path Algorithm | Brilliant Math & Science Wiki*. URL: <https://brilliant.org/wiki/dijkstras-short-path-finder/> (acedido em 30/05/2017).
- [28] N. Nilsson P. Hart e B. Raphael. *A Formal Basis for the Heuristic Determination of Minimum Cost Paths, IEEE Trans. Syst. Sci. Cyber., vol. 4, no. 2, pp. 100-107, 1968*. (Acedido em 03/06/2017).
- [29] *Path-Finding- A\**. URL: <http://www.leocck.com/blog/pt/03/05/2012/path-finding-3-a-star/> (acedido em 30/05/2017).
- [30] Siemens AG. *Communication with SIMATIC*. URL: [https://cache.industry.siemens.com/dl/files/686/1254686/att\\_46478/v1/](https://cache.industry.siemens.com/dl/files/686/1254686/att_46478/v1/) (acedido em 03/06/2017).
- [31] *Protocolos Industriais Profibus e Profinet*. URL: <http://www.feng.pucrs.br/professores/tergolina/> (acedido em 03/06/2017).
- [32] *MiR robot REST api*. URL: <https://goo.gl/MaV1Gq> (acedido em 18/06/2017).
- [33] *Raspberry Pi 2 + MFRC522-python, to read RFID tag*. URL: <http://www.instructables.com/id/Simple-and-intuitive> (acedido em 30/05/2017).

- 
- [34] *Raspbian for Raspberry Pi*. URL: <https://www.raspberrypi.org/downloads/raspbian/> (acedido em 31/05/2017).