

UNIVERSIDADE ESTADUAL DE GOIÁS  
CÂMPUS ANÁPOLIS DE CIÊNCIAS EXATAS E TECNOLÓGICAS HENRIQUE  
SANTILLO  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

JORGE BRAYER GAZAL JUNIOR

Estudo do desenvolvimento de um veículo autônomo com viabilidade econômica e  
inteligência artificial

Anápolis  
Dezembro, 2019

UNIVERSIDADE ESTADUAL DE GOIÁS  
CÂMPUS ANÁPOLIS DE CIÊNCIAS EXATAS E TECNOLÓGICAS HENRIQUE  
SANTILLO  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

JORGE BRAYER GAZAL JUNIOR

Estudo do desenvolvimento de um veículo autônomo com viabilidade econômica e  
inteligência artificial

Trabalho de conclusão de curso apresentado ao Departamento de Sistemas de Informação  
Câmpus Anápolis de Ciências Exatas e Tecnológicas Henrique Santillo da Universidade  
Estadual de Goiás, como requisito parcial para obtenção do grau de Bacharel em Sistemas de  
Informação.

Orientador: Prof. Dr. Francisco Ramos de Melo

Anápolis  
Dezembro, 2019



## ATA DE DEFESA DE TRABALHO DE CURSO

**Título do Trabalho:** Estudo do desenvolvimento de um veículo autônomo com viabilidade econômica e inteligência artificial

**Autor:** Jorge Brayer Gazal Junior

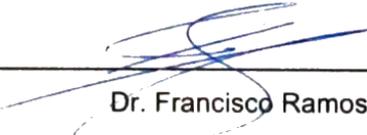
**Orientador:** Dr. Francisco Ramos de Melo

O autor apresentou em sessão pública o Trabalho de Curso, às 18h15min, de 19 de novembro de 2019, terça-feira, no curso de Sistemas de Informação do Câmpus Anápolis de Ciências Exatas e Tecnológicas Henrique Santillo da Universidade Estadual de Goiás. Em seguida, cada membro da Banca Examinadora fez suas considerações e arguições. A Banca examinadora, em reunião fechada, realizou a avaliação do trabalho e da apresentação, proclamando o seguinte resultado:

- aprovado
- aprovado com correções
- reprovado.

A presidência encerrou a sessão pública de defesa do Trabalho de Curso.

Anápolis, 19 de novembro de 2019.

  
\_\_\_\_\_  
Dr. Francisco Ramos de Melo  
Orientador / Presidente da Banca Examinadora

  
\_\_\_\_\_  
M.ª Juliana Martins de Bessa Ferreira  
Avaliadora

## **FICHA CATALOGRÁFICA**

**GAZAL JUNIOR, Jorge Brayer. Estudo do desenvolvimento de um veículo autônomo com viabilidade econômica e inteligência artificial.** Anápolis, 2019. (UEG / CCET, Bacharelado em Sistemas de Informação, 2019).

Monografia. Universidade Estadual de Goiás, Unidade Universitária de Ciências Exatas e Tecnológicas. Departamento de Sistemas de Informação.

1. Veículos guiados automaticamente. 2. Redes neurais artificiais. 3. Aprendizado por Reforço

## **REFERÊNCIA BIBLIOGRÁFICA**

**GAZAL JUNIOR, Jorge Brayer. Estudo do desenvolvimento de um veículo autônomo com viabilidade econômica e inteligência artificial.** Anápolis, 2019. 98 p. Monografia - Curso de Sistemas de Informação, CCET, Universidade Estadual de Goiás.

## **CESSÃO DE DIREITOS**

NOME DO AUTOR: Jorge Brayer Gazal Junior

TÍTULO DO TRABALHO: Estudo do desenvolvimento de um veículo autônomo com viabilidade econômica e inteligência artificial

GRAU/ANO: Graduação / 2019

É concedida à Universidade Estadual de Goiás permissão para reproduzir cópias deste trabalho, emprestar ou vender tais cópias para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste trabalho pode ser reproduzida sem a autorização por escrito do autor.

---

Jorge Brayer Gazal Junior

Anápolis, 02 de dezembro de 2019

Dedico este trabalho aos meus pais, a meus amigos, e a todos que estiveram comigo e contribuíram em meu desenvolvimento acadêmico.

## **AGRADECIMENTOS**

Ao meu orientador Prof. Dr. Francisco Ramos de Melo pelo apoio, sugestões, indicações, incentivo no desenvolvimento deste trabalho.

Aos meus pais, amigos e todos que contribuíram, direta ou indiretamente, no meu desenvolvimento acadêmico e neste trabalho.

## RESUMO

Veículos autônomos tem se mostrado como o futuro da locomoção. Poder ir de um ponto A para um ponto B deixando a navegação do veículo nas mãos da tecnologia, permitindo que ela tome as decisões por si mesma. Para se chegar a isso foram necessários vários avanços na eletrônica, informática, automação e robótica. Um tipo de veículo que surgiu muito antes da aplicação de tecnologias para tornar carros autônomos foram os AGVs (veículos guiados automaticamente). Esse tipo de veículo é utilizado em diversos tipos de indústria auxiliando no sistema produtivo e de armazenamento. Este trabalho tem como objetivo o estudo de aspectos construtivos desse tipo de veículo e de plataformas de prototipagem eletrônica e sensores que podem ser aplicados para se desenvolver um veículo de baixo custo aliado com a implementação de inteligência artificial por meio de redes neurais artificiais e aprendizado por reforço. A pesquisa mostra que é viável a construção de um AGV que atenda os requisitos do objetivo. Foi realizada a simulação de um veículo virtual em três diferentes cenários com diferentes obstáculos e diferentes objetivos para a realização de testes da rede neural artificial a fim de analisar o comportamento do veículo antes de uma implementação real em um futuro protótipo. Os resultados da simulação mostram que a complexidade do ambiente onde o agente está inserido, ou seja, a disposição dos obstáculos, influencia na quantidade de ciclos de aprendizado. Com o aprendizado consolidado, as próximas etapas para futuras implementações são a construção de um protótipo em pequena escala e posteriormente um AGV.

Palavras-chave: Veículos autoguiados, veículos autônomos, inteligência artificial.

## **ABSTRACT**

Autonomous vehicles have proving to be the future of locomotion. Being able to go from point A to point B leaving the navigation of the vehicle in the hands of technology, allowing it to make the decisions for itself. Achieving this required several advances in electronics, computing, automation and robotics. One type of vehicle that emerged long before the application of technologies to make cars autonomous were AGVs (Automated Guided Vehicles). This type of vehicle is used in various types of industry assisting in the production and storage system. This work aims to study the constructive aspects of this type of vehicle and electronic prototyping platforms and sensors that can be applied to develop a low cost vehicle coupled with the implementation of artificial intelligence through artificial neural networks and reinforcement learning. Research shows that building an AGV that meets the objective requirements is feasible. A virtual vehicle was simulated in three different scenarios with different obstacles and different objectives for testing the artificial neural network to analyze the vehicle's behavior before a real implementation in a future prototype. The simulation results show that the complexity of the environment where the agent is inserted, ie the arrangement of obstacles, influences the amount of learning cycles. With consolidated learning, the next steps for future implementations are building a small-scale prototype and later an AGV.

Keywords: Self-driving vehicles, Autonomous vehicles, artificial intelligence.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Um modelo de AGV que se locomove por meio de linhas guia no chão .....	21
Figura 2 - AGVs em uma linha de produção transportando a carroceria de um veículo .....	22
Figura 3- Diagrama de blocos de um AGV .....	25
Figura 4 - Sensor ultrassônico Sick UC12.....	26
Figura 5 - Sensor indutivo Götting .....	27
Figura 6 - Sensor óptico Götting .....	27
Figura 7 - AGV com sensor de navegação laser.....	28
Figura 8 - Scanner laser Sick Nav 300 .....	28
Figura 9 - Mapeamento de um ambiente feito por um scanner laser.....	28
Figura 10 - GPS Götting .....	29
Figura 11 - Sensores de segurança.....	29
Figura 12 - Encoder absoluto Sick A3M60 .....	30
Figura 13 - CLP Siemens ET 200S com seus módulos .....	30
Figura 14 - Esquema genérico de aplicação de um CLP .....	31
Figura 15 - Linguagem de programação Ladder .....	31
Figura 16 - Linguagem de diagrama de blocos (FBD).....	32
Figura 17 - Linguagem de lista de instruções .....	32
Figura 18 - Diferença entre microprocessador e microcontrolador .....	35
Figura 19 - Arduino UNO.....	36
Figura 20 - Raspberry Pi 3 modelo B+ .....	36
Figura 21 - AGV construído pela Peugeot Citroën que usa o Raspberry Pi.....	37
Figura 22 - Diagrama de funcionamento do sensor sonar .....	39
Figura 23 - Sensor sonar HC-SR04 .....	39
Figura 24 - Sensor sonar MB1414 USB-ProxSonar-EZ1.....	40
Figura 25 - Diagrama de funcionamento do sensor infravermelho .....	40
Figura 26 - Sensor Infravermelho Sharp GP2Y0A02YK.....	40
Figura 27 - Diagrama de blocos do sistema nervoso biológico .....	42
Figura 28 - Neurônio biológico .....	43
Figura 29 - Modelo de neurônio artificial MCP .....	45
Figura 30 - Gráfico da função de ativação ReLU .....	46
Figura 31 - Gráfico da função Softmax .....	47

Figura 32 - Exemplo de uma rede neural de três camadas de neurônios.....	49
Figura 33 - Rede Perceptron de camada única .....	50
Figura 34 - Rede Perceptron multicamadas .....	51
Figura 35 - Robô em um labirinto .....	52
Figura 36 - Diagrama em blocos da aprendizagem por reforço .....	54
Figura 37 - Estados de transição .....	56
Figura 38 - Matrizes Q e R .....	57
Figura 39 - Diagrama de um Agente .....	59
Figura 40 - AGV do tipo empilhadeira transportando caixas .....	59
Figura 41 - Exemplo de grafo .....	62
Figura 42 - Diagrama da montagem do protótipo de um AGV .....	69
Figura 43 - Diagrama esquemático da montagem dos componentes do protótipo de um AGV .....	69
Figura 44 - Aspecto provável do protótipo do AGV .....	70
Figura 45 - Ações do protótipo .....	70
Figura 46 - Fluxograma de ações do veículo ao detectar um obstáculo .....	71
Figura 47 - Exemplo do desvio de obstáculos realizado por um robô.....	73
Figura 48 - Gráfico com valores aleatórios para os três sensores.....	73
Figura 49 - Percepção dos sensores e ação tomada .....	74
Figura 50 - Modelo da rede neural .....	76
Figura 51 - Simulação do veículo autônomo .....	77
Figura 52 - Simulação do veículo autônomo no Cenário 1 .....	78
Figura 53 - Gráfico de recompensas do veículo autônomo no Cenário 1 .....	78
Figura 54 - Simulação do veículo autônomo no Cenário 2 .....	79
Figura 55 - Gráfico de recompensas do veículo autônomo no Cenário 2 .....	79
Figura 56 - Dificuldade encontrada pelo veículo virtual no Cenário 2 .....	80
Figura 57 - Simulação do veículo autônomo no Cenário 3 .....	80
Figura 58 - Gráfico de recompensas do veículo autônomo no Cenário 3 .....	81

## LISTA DE QUADROS

Quadro 1 - Mercado de AGVs no Brasil .....	17
Quadro 2 - Formas de redução de custos com o uso de AGVs .....	23
Quadro 3 - Especificação das plataformas Arduino e Raspberry Pi.....	37
Quadro 4 - Sensores comerciais usados em plataformas de prototipagem eletrônica .....	41
Quadro 5 - Lista de componentes para a montagem de um protótipo .....	68
Quadro 6 - Tabela verdade do funcionamento dos sensores e atuadores .....	72
Quadro 7 - Ação do robô de acordo com os valores de medição dos sensores .....	74

## LISTA DE ABREVIATURAS E SIGLAS

<b>Siglas</b>	<b>Descrição</b>
AGV	<i>Automated Guided Vehicle</i> - Veículo Guiado Automaticamente
CI	Circuito Integrado
CLP	Controlador Lógico Programável
DC	<i>Direct Current</i> - Corrente Direta
DQL	<i>Deep Q-Learning</i>
DQN	<i>Deep Q-Network</i>
E/S	Sigla de Entrada/Saída de portas digitais de componentes eletrônicas
IA	Inteligência Artificial
IDE	<i>Integrated Development Environment</i> - Ambiente de Desenvolvimento Integrado
IR	Infravermelho
MDP	<i>Markov Decision Process</i> - Processo de Decisão de Markov
MLP	<i>Multilayer Perceptron</i> - Perceptron de Múltiplas Camadas
PWM	<i>Pulse Width Modulation</i> - Modulação por Largura de Pulso
RFID	<i>Radio-Frequency Identification</i> - Identificação por Radiofrequência
RNA	Rede Neural Artificial
TD	<i>Temporal Difference</i> - Diferença Temporal
USB	<i>Universal Serial Bus</i> - Porta serial universal

## SUMÁRIO

<b>RESUMO.....</b>	<b>7</b>
<b>ABSTRACT.....</b>	<b>8</b>
<b>LISTA DE ILUSTRAÇÕES.....</b>	<b>9</b>
<b>LISTA DE QUADROS.....</b>	<b>11</b>
<b>LISTA DE ABREVIATURAS E SIGLAS.....</b>	<b>12</b>
<b>INTRODUÇÃO.....</b>	<b>16</b>
<b>CAPÍTULO 1 - VEÍCULOS GUIADOS AUTOMATICAMENTE.....</b>	<b>21</b>
1.1 O que é um AGV?.....	21
1.1.1 Benefícios e vantagens dos AGVs.....	22
1.1.2 Avanços tecnológicos que permitiram a evolução dos AGVs.....	24
1.2 Aspectos construtivos de AGVs.....	25
1.2.1 Sensores usados em AGVs.....	26
1.2.2 CLP.....	30
<b>CAPÍTULO 2 - PLATAFORMAS DE PROTOTIPAGEM ELETRÔNICA.....</b>	<b>33</b>
2.1 Construção de AGVs utilizando plataformas eletrônicas de prototipagem.....	33
2.2 Microprocessador e Microcontrolador.....	34
2.3 Plataformas para projetos eletrônicas.....	35
2.3.1 Arduino.....	35
2.3.2 Raspberry Pi.....	36
2.3.3 Comparativo entre Arduino e Raspberry Pi.....	37
2.4 Sensores usados em plataformas de prototipagem eletrônica.....	39
<b>CAPÍTULO 3 - INTELIGÊNCIA ARTIFICIAL.....</b>	<b>42</b>
3.1 Diferença entre AGV convencional e inteligente.....	42
3.2 Sistema nervoso biológico.....	42
3.2.1 Neurônio biológico.....	43
3.3 Neurônio artificial.....	44
3.4 Função de ativação.....	46
3.4.1 Função ReLU.....	46
3.4.2 Função Softmax.....	47
3.5 Redes neurais artificiais.....	47
3.5.1 Perceptron de camada única.....	49
3.5.2 Perceptron de múltiplas camadas.....	50

3.5.3 Aprendizado por reforço .....	52
3.6 Agentes inteligentes .....	58
3.7 Autonomia em robótica.....	59
3.8 Rotas.....	61
3.9 Estudos do uso de inteligência artificial em AGVs .....	62
<b>CAPÍTULO 4 - METODOLOGIA .....</b>	<b>64</b>
4.1 Tipos da pesquisa .....	64
4.1.1 Quanto aos fins .....	64
4.1.2 Quanto aos meios .....	64
4.2 Materiais e métodos .....	65
4.3 Instrumentos e procedimentos de coleta de dados .....	66
4.4 Tratamento dos dados .....	66
4.5 Validação da simulação .....	66
<b>CAPÍTULO 5 - PROTÓTIPO.....</b>	<b>67</b>
5.1 Escolha dos componentes .....	67
5.2 Diagrama esquemático .....	68
5.3 Funcionamento.....	70
5.3.1 Sensores .....	73
<b>CAPÍTULO 6 - SIMULAÇÃO DO VEÍCULO VIRTUAL.....</b>	<b>75</b>
6.1 Modelagem da rede neural .....	75
6.2 Rede neural .....	76
6.3 Simulação.....	77
6.3.1 Cenário 1 .....	78
6.3.2 Cenário 2 .....	79
6.3.2 Cenário 3 .....	80
<b>CAPÍTULO 7 - RESULTADOS DA PESQUISA.....</b>	<b>82</b>
<b>CONCLUSÃO.....</b>	<b>84</b>
<b>REFERÊNCIAS.....</b>	<b>86</b>
<b>GLOSSÁRIO.....</b>	<b>92</b>
<b>APÊNDICES.....</b>	<b>93</b>
Apêndice A – Cronograma previsto e realizado .....	93
Apêndice B - Pseudocódigo do algoritmo criado a partir da tabela-verdade dos sensores e atuadores .....	94
Apêndice C - Banner apresentado no X Simpósio de Tecnologia de Informação.....	95

<b>ANEXOS.....</b>	<b>96</b>
Anexo A - Pinagem do Raspberry Pi 3 B+ .....	96
Anexo B - Pinagem do sensor ultrassônico HC-SR04.....	97
Anexo C - Pinagem da Ponte H L293D .....	98

## INTRODUÇÃO

Entre os anos de 2016 e 2018 ocorreram consideráveis comentários e discussões sobre veículos autônomos, principalmente após o uso de sistemas nos carros elétricos da Tesla<sup>1</sup> que permitem a condução por direção autônoma usando câmeras, sensores e radares (TESLA..., 2016). Entretanto, foi entre os anos de 2008 e 2010 que houve o surgimento do interesse em veículos autônomos, quando a tecnologia para esses veículos já estava mais desenvolvida e começou a receber investimentos. Uma das principais empresas a se interessar foi a Google, em 2010, fazendo com que outras empresas passassem a se interessar pela pesquisa desse tipo de veículo (ELAINA, 2019).

Para que automóveis pudessem usar sistemas autônomos foi necessária a evolução da automação e robótica, e o que permitiu isso foram os avanços na computação, mais precisamente após o surgimento do microprocessador (ROSÁRIO, 2005).

Segundo Rosário (2005, p. 142): “Uma das maiores fantasias do homem é a construção de uma máquina com inteligência artificial, capaz de agir e pensar como ele”. Já Ullrich (2015, p. 1, tradução nossa) afirma que: “[...] robôs autônomos faziam parte do sonho que a humanidade tinha de deixar as máquinas realizarem nosso trabalho. O rápido desenvolvimento da tecnologia sensorial e regulatória, assim como os primeiros desenvolvimentos em microeletrônica, abriram o caminho para os Veículos Guiados Automaticamente.”.

Não estamos tão longe de algo assim, os veículos da Tesla são um exemplo disso. O conjunto de sensores, processadores, rede de comunicação e software permitem o aprendizado e a tomada de decisão na condução do veículo.

Dentro da robótica existe um grupo de robôs utilizado em muitos ramos da indústria e áreas de produção que auxilia no transporte de peças, componentes e equipamentos, trabalhando de forma autônoma e podendo funcionar 24 horas por dia conhecidos como AGV (do inglês *Automated Guided Vehicle* - Veículo Guiado Automaticamente). É sobre os AGVs que o estudo irá se concentrar.

AGVs podem se apresentar como uma solução interessante na realização de tarefas repetitivas, perigosas e outras situações nos diferentes processos produtivos e de armazenamento. Atualmente existem diferentes versões desses veículos, de acordo com o tipo de aplicação e ambiente de trabalho.

---

<sup>1</sup> Tesla Motors, empresa norte-americana que desenvolve e fabrica automóveis elétricos.

De acordo com o relatório do mercado de AGVs publicado em 2018 pela *Grand View Research* o tamanho desse mercado em 2018 foi avaliado em mais de US\$ 2 bilhões, com previsão de aumento de 15,8% entre 2019 e 2025 (AUTOMATED..., 2018). Já a *Globe Newswire* publicou um artigo sobre o relatório da *Zion Market Research* informando sobre a expectativa de mercado de AGVs até o ano de 2027. O artigo afirma que entre 2019 e 2027 o aumento desse mercado será de 12,8% e que os fabricantes de AGVs estão se concentrando na “integração de sua frota com inteligência artificial (IA) para reduzir o número de acidentes com veículos autônomos”. O crescimento da adoção da automação no manuseio de materiais para aumentar a eficiência operacional e reduzir o tempo de inatividade é um dos fatores que impulsionam o crescimento do mercado de AGVs (GLOBAL..., 2019).

Com relação a América Latina e, principalmente o Brasil, a *Globe Newswire* afirma que a “América Latina está contribuindo para o aumento do mercado global AGVs devido ao crescente setor ferroviário da região.”. Em 2017 o transporte ferroviário no Brasil cobria cerca de 21% do total de cargas e era cerca de 30% mais econômico em comparação ao transporte rodoviário. O governo brasileiro anunciou investimentos no valor de mais de US\$ 7,5 bilhões para o setor ferroviário (GLOBAL..., 2019).

De acordo com o relatório do mercado de AGVs, com relação ao Brasil, o Quadro 1 apresenta as principais tecnologias de navegação, aplicações e indústrias.

**Quadro 1-** Mercado de AGVs no Brasil

<b>Tecnologia de navegação</b>	<b>Aplicação</b>	<b>Indústria</b>
<ul style="list-style-type: none"> <li>● Indutivo</li> <li>● Laser</li> <li>● Magnético</li> <li>● Natural</li> <li>● Visão</li> <li>● Outros</li> </ul>	<ul style="list-style-type: none"> <li>● Carregamento e descarregamento de reboques</li> <li>● Embalagem</li> <li>● Logística e armazenamento               <ul style="list-style-type: none"> <li>○ Armazém de frios</li> <li>○ Atacado e distribuição</li> <li>○ <i>Cross-docking</i><sup>2</sup></li> <li>○ Distribuição</li> <li>○ Transporte</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>● Setor de atacado e distribuição               <ul style="list-style-type: none"> <li>○ <i>E-commerce</i></li> <li>○ Hotéis e restaurantes</li> <li>○ Mercarias</li> <li>○ Redes de varejo</li> </ul> </li> <li>● Setor de manufatura               <ul style="list-style-type: none"> <li>○ Aeroespacial</li> <li>○ Automotivo</li> <li>○ Defesa</li> <li>○ Eletrônicos</li> <li>○ Farmacêutico</li> </ul> </li> </ul>

(CONTINUA)

<sup>2</sup> Método de logística em que o empresário não precisa necessariamente armazenar os produtos em seu centro de distribuição. No momento em que um pedido é realizado, ele envia uma solicitação de compra para o fornecedor, que por sua vez envia para ele as mercadorias (CROSS..., 2017).

(CONTINUAÇÃO)

Tecnologia de navegação	Aplicação	Indústria
-	<ul style="list-style-type: none"> <li>● Manipulação de matéria prima</li> <li>● Montagem</li> <li>● Outros</li> </ul>	<ul style="list-style-type: none"> <li>○ FMCG<sup>5</sup></li> <li>○ Química</li> <li>○ Papel</li> <li>○ Plásticos</li> <li>○ Outros</li> </ul>

Fonte: Automated..., 2018

A escolha do tema se deu pela experiência do autor do projeto com esse tipo de robô, tendo trabalhado brevemente com AGVs, mais especificamente os orientados por indução e a laser, entendendo seu projeto, funcionamento e programação. Como contribuição para o curso de Sistemas de Informação, um AGV fornece estudo e aprendizado interdisciplinar na área pelo fato do mesmo ser um sistema automatizado, uma *hardware* e *software*, programável, com possibilidade de ajudar o usuário/empresa em uma melhor organização de seu sistema logístico, de armazenamento e de produção.

Analisando o cenário industrial da cidade de Anápolis, no Estado de Goiás, existe um distrito industrial, o DAIA (Distrito Agroindustrial de Anápolis), que concentra uma grande quantidade de indústrias, principalmente do setor farmacêutico. Além desse setor, existem também indústrias de alimentos e do setor automobilístico. Todas essas indústrias podem se beneficiar no uso de AGVs em seu processo produtivo e logístico, e encontrar uma solução confiável e de baixo custo é muito vantajoso, sendo um diferencial para um salto na produção e competitividade.

AGVs não são algo novo, surgiram a quase 70 anos e as tecnologias de desenvolvimento dos mesmos evoluíram muito de forma a torná-los mais autônomos, inteligentes e seguros. Este trabalho abordará sobre o que é um AGV, seus benefícios, métodos de construção, autonomia e inteligência artificial aplicada a robôs móveis. Também é descrito sobre as principais plataformas de prototipagem eletrônica existentes atualmente e a escolha de componentes para a construção de um protótipo de baixo custo.

Este trabalho teve como problema de pesquisa responder a questão: É possível o desenvolvimento de um veículo autônomo inteligente com abrangência de viabilidade econômica?

<sup>3</sup> *Fast-Moving Consumer Goods* ou *Consumer Packaged Goods* são produtos que são vendidos rapidamente e a um custo relativamente baixo. Exemplos incluem bens não duráveis, como alimentos embalados, bebidas, produtos de higiene pessoal, medicamentos sem receita e outros consumíveis (FAST-MOVING..., 2018).

A partir do problema foram levantadas as seguintes questões a serem respondidas pela pesquisa: O que é um AGV? Quais os sistemas inteligentes usados em AGVs? Como treinar um AGV para desviar de obstáculos? Como pode ser o processo de construção de um AGV usando componentes de baixo custo? É possível construir um protótipo em maior escala, para aplicação industrial, usando plataformas eletrônicas de baixo custo? Quais as tecnologias existentes para a construção de um AGV? Quais as diferenças entre um AGV convencional e um inteligente?

Para responder o problema de pesquisa foi levantada a seguinte hipótese: O emprego de plataformas eletrônicas de baixo custo aliado ao emprego de técnicas de inteligência artificial pode possibilitar o desenvolvimento de AGVs inteligentes e economicamente acessíveis.

A pesquisa teve como objetivo geral estudar tecnologias de desenvolvimento de veículos autônomos, plataformas eletrônicas e analisar a viabilidade do desenvolvimento de um veículo inteligente de baixo custo.

Os objetivos específicos da pesquisa foram:

1. Estudar trabalhos de autores que desenvolveram veículos autônomos.
2. Estudar sensores, microcontroladores e atuadores usados no desenvolvimento de veículos autônomos.
3. Estudar a evolução de tecnologias desenvolvidas para AGVs e o que existe atualmente.
4. Estudar plataformas como Arduino e Raspberry Pi.
5. Simulação e testes de um AGV virtual com inteligência artificial.

O trabalho está dividido em sete capítulos, sendo os capítulos 1 a 3 Referencial Teórico, estando cada capítulo dividido da seguinte forma:

- Capítulo 1 - Veículos Guiados Automaticamente: Uma abordagem geral sobre veículos guiados automaticamente, sua construção e componentes.
- Capítulo 2 - Plataformas de Prototipagem Eletrônica: Aborda as principais plataformas atualmente existentes para a construção de protótipos eletrônicos, o Arduino e Raspberry Pi, além de uma abordagem sobre alguns sensores que podem ser utilizados com essas plataformas.
- Capítulo 3 - Inteligência Artificial: Trata a diferença entre AGVs convencionais e inteligentes, e traz uma apresentação sobre inteligência artificial e redes neurais utilizadas no projeto.
- Capítulo 4 - Metodologia: Apresenta a metodologia utilizada na pesquisa.

- Capítulo 5 - Protótipo: Neste capítulo é descrito sobre a escolha dos componentes, processo de criação de um protótipo e seu funcionamento.
- Capítulo 6 - Simulação do veículo virtual: Apresenta e explica a rede neural usada para simulação do veículo autônomo.
- Capítulo 7 - Resultados da pesquisa: Trata da análise e estudo dos resultados observados com a simulação.
- Ao final se tem a conclusão, abordando o que foi analisado e as respostas obtidas para as questões levantadas durante o desenvolvimento da pesquisa.

## CAPÍTULO 1 - VEÍCULOS GUIADOS AUTOMATICAMENTE

Os veículos guiados automaticamente (AGV do inglês *Automated Guided Vehicle*) foram criados nos anos 1950 nos Estados Unidos tendo como objetivo inicial a substituição do motorista de um reboque de trator para o transporte de materiais e eram guiados por trilhos. Posteriormente o sistema de trilhos foi substituído por um condutor elétrico fixado no chão, sistema conhecido como pista guiada indutiva (ULLRICH, 2015).

### 1.1 O que é um AGV?

Um AGV é um robô móvel, autônomo, programável, que se move orientado por diferentes tecnologias de navegação, como linhas guias no chão (fitas magnéticas ou RFID<sup>4</sup>), sensores laser, câmeras, ondas de rádio ou GPS. É usado em diferentes tipos de indústria para o transporte de mercadorias em fábricas e depósitos (ULLRICH, 2015).

Uma definição mais simples pode ser a de veículo autodirigido usado para transportar materiais de um local para outro sem qualquer operador acompanhando (PATIL; G, 2018).

Praticamente qualquer carga pode ser transportada por AGVs. Qualquer operação em que paletes, containers, fardos, pacotes ou qualquer carga semelhante que precisa ser transportada pode em geral fazer uso dos AGVs. Desde meados da década de 1990 muitas indústrias, de diferentes ramos, passaram a utilizar os AGVs (ULLRICH, 2015). As Figuras 1 e 2 apresentam dois diferentes modelos de AGVs.

**Figura 1-** Um modelo de AGV que se locomove por meio de linhas guia no chão



Fonte: Website da Turck<sup>5</sup>

<sup>4</sup> Identificação por radiofrequência.

<sup>5</sup> Disponível em: <https://www.turck.us/en/rfid-in-an-automated-guided-vehicle-6870.php>. Acesso em 5 abr. 2019.

**Figura 2** - AGVs em uma linha de produção transportando a carroceria de um veículo



Fonte: Website da Selettra<sup>6</sup>

### 1.1.1 Benefícios e vantagens dos AGVs

Entre os benefícios dos AGVs estão o aumento na eficiência das operações de armazenamento e dos processos de gerenciamento logístico. O seu uso gera economia com mão-de-obra em tarefas repetitivas de manuseio de materiais, além de garantir operações mais seguras no manuseio de mercadorias muito pesadas e volumosas, ou a exposição a ambientes e mercadorias perigosas. Entre os pontos vantajosos no uso de AGVs, Ullrich (2015, p. 33) cita outros, entre eles:

- Melhorias na produtividade devido à melhor organização de materiais e no fluxo de informações;
- Redução do estoque de reserva e dos tempos de espera de produção;
- Redução de danos no transporte e de entregas perdidas;
- Alta disponibilidade e confiabilidade;
- Ambiente de trabalho mais seguro, limpo e silencioso;
- Investimento mínimo em infraestrutura;
- Uso de rotas existentes;
- Pode ser usado em ambientes internos e externos.

Um dos principais benefícios na adoção de AGVs em processos de fabricação é a redução de custos. O Quadro 2 apresenta as sete principais maneiras de reduzir custos na adoção de AGVs de acordo com Benevides (2016).

<sup>6</sup> Disponível em: <http://www.selettra.com.br/ver-novidades/3/automated-guided-vehicles-agv-selettrack>. Acesso em 5 abr. 2019.

**Quadro 2** - Formas de redução de custos com o uso de AGVs

<b>Benefício</b>	<b>Descrição</b>
<b>Redução de mão-de-obra</b>	Substituição de um funcionário e renúnciação de uma nova contratação. Custos eliminados com um funcionário humano: Salário; Planos de Saúde; Benefícios de aposentadoria; Férias; Horas extras; etc.
<b>Remoção de salário e aumento de benefícios</b>	Os custos trabalhistas tendem a subir, pois funcionários esperam que sua remuneração cresça com o passar do tempo e experiência, aumentando o custo de mão-de-obra. Ao substituir o funcionário por um AGV, após recuperar o investimento inicial, a cada ano existe um aumento no lucro. Os únicos gastos são com a manutenção, reparo e consumo de energia, mas ainda assim são menores do que os custos com um humano.
<b>Eliminação de danos a estruturas e produtos</b>	Humanos cometem erros, ficam cansados e distraídos, provocando acidentes, e acidentes geram prejuízos para empresas. Os acidentes podem afetar produtos, estruturas e pessoas.
<b>Aumento na segurança do trabalho e redução de custos</b>	AGVs removem o elemento humano, raiz de muitos acidentes. AGVs podem realizar trabalhos considerados perigosos para humanos como a elevação de cargas muito pesadas, manuseio de substâncias perigosas, trabalhos em temperaturas extremas, entre outros. Os AGVs se deslocam com velocidade controlada, acelerando e reduzindo a velocidade nos pontos determinados, diferente de um humano, que pode não seguir a velocidade permitida. O aumento na segurança reduz custos como seguro acidente e tempo com reparos estruturais.
<b>Mais baratos que sistemas de automação fixa</b>	Um sistema de automação fixo é mais caro e, durante a construção, afetam o fluxo de trabalho e levam muito tempo para serem instalados. Ao ser implementado, é difícil e caro alterar o layout caso seja necessário. Os AGVs precisam apenas ser reprogramados para seguir novos caminhos.
<b>Redução dos custos de serviço</b>	Um AGV pode trabalhar 24 horas, 7 dias por semana, aumentando dessa forma a produtividade e o resultado final. Podem operar em ambientes com baixa iluminação, e com altas/baixas temperaturas. Isso reduz custos em eletricidade e com aquecimento ou ventilação.
<b>Aumentam a eficiência e a precisão do inventário</b>	Quando os AGVs possuem um sistema de controle de estoque, o rastreamento de inventário é rápido e fácil, eliminando a necessidade de funcionários para realizar essa tarefa. Isso evita perda de tempo e permite saber com exatidão a quantidade de material disponível, quando solicitar a compra de novos materiais, e a redução de desperdício.

Fonte: Adaptado de Benevides (2016).

### 1.1.2 Avanços tecnológicos que permitiram a evolução dos AGVs

Segundo Ullrich (2015, p. 6-7 e 10), nos anos 1970, com o rápido avanço da eletrônica e da tecnologia sensorial, os AGVs puderam evoluir de modo a atender a grande demanda de fábricas, principalmente do setor automobilístico, que estavam se modernizando e automatizando. Entre esses avanços estão:

- Componentes eletrônicos e microprocessadores mais rápidos e de alto desempenho. Uso do CLP<sup>7</sup> (Controlador Lógico Programável). Sensores com maior precisão e mais acessíveis;
- Baterias com maior capacidade de carga e introdução do carregamento de baterias;
- Introdução da navegação por pista indutiva ativa, permitindo o controle do motor de direção;
- Maior amplitude de movimento dos veículos, permitindo se mover por áreas predeterminadas e em ambientes externos.
- Veículos guiados automatizados foram totalmente integrados nos processos de produção, tornando possível usar os veículos como bancadas móveis (linhas de montagem).
- Uso de sinais infravermelhos e de rádio para a transferência de dados.

Posteriormente, entre os anos 1990 e 2010 surgiram novas tecnologias, sendo os AGVs controlados por computadores convencionais e orientados por sensores magnéticos ou laser. É também estabelecido um novo meio para a transferência de dados, as redes sem fio. Os avanços tecnológicos surgidos nessa era possibilitaram:

- Velocidades mais altas para veículos durante a condução, manobras, manuseio de cargas, graças a melhor tecnologia sensorial;
- Veículos de baixo custo com o uso de soluções mais simples;
- Conceitos de energia alternativa com transmissão de energia indutiva;
- Novos processos de navegação (ponto magnético, laser, transponder, navegação predial);
- O sucesso dos computadores aplicados nos veículos para o controle de equipamentos e em tecnologia sensorial;

---

<sup>7</sup> Desenvolvidos nos anos 60, nos Estados Unidos, o CLP é um computador especializado, baseado em um microprocessador, projetado para comandar e monitorar máquinas ou processos industriais, desempenhando funções de controle por meio de softwares. É amplamente utilizado na indústria para o controle de diversos tipos e níveis de complexidade. Deve possuir um processador com software de controle e hardware que suporte operação em ambientes industriais (CONTROLADOR..., 2019).

- Transmissão de dados predominantemente via WLAN;
- Novas áreas de aplicação para os AGVs como armazéns a granel e hospitais.

As tecnologias no desenvolvimento dos AGVs continuam evoluindo com o surgimento de novos e melhores sensores, estudos da implementação de inteligência artificial e estudos de plataformas eletrônicas de baixo custo em substituição ao CLP convencional para o controle e processamento das informações.

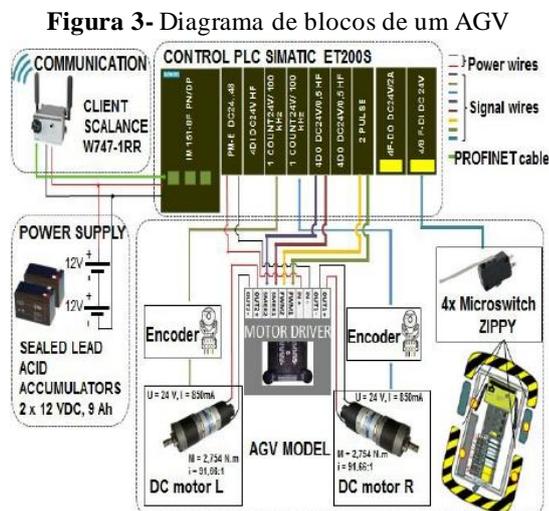
## 1.2 Aspectos construtivos de AGVs

O estudo da construção de um AGV está ligado ao tipo de indústria, tipo de carga, peso da carga, ambiente (interno ou externo), percurso que o mesmo deve se locomover, entre outros fatores (ULLRICH, 2015).

Protótipos de AGVs usados para testes e simulação são geralmente construídos tendo como base veículos comerciais elétricos, pois não é necessário projetar todo o sistema, apenas acrescentar os sensores e dispositivos necessários para que o veículo se transforme em um AGV, economizando dessa forma tempo e recursos (GALDAMES, 2011).

É possível ainda criar AGVs em pequena escala usando plataformas eletrônicas como o Arduino e o Raspberry Pi. A maioria dos exemplos de projetos utilizando essas plataformas é a de seguidores de linha, que seria o princípio de funcionamento de um AGV orientado por fita magnética. Algumas literaturas se aprofundam mais na construção dos projetos e utilizam técnicas de inteligência artificial para tornar os robôs mais autônomos.

A Figura 3 apresenta o diagrama de blocos de um AGV. Os principais componentes serão discutidos ao longo deste capítulo, sendo dado destaque ao CLP (Controlador Lógico Programável), representado na figura pelo Simatic ET 200s, e os sensores.



Fonte: Kajan et al. (2013, p. 345)

### 1.2.1 Sensores usados em AGVs

Um AGV é programado para seguir um trajeto e, em pontos pré-determinados, parar para a realização de uma ação, como por exemplo, a carga ou descarga de materiais. Ele possui sensores que servem para seguir o trajeto e sensores de segurança. Os sensores para seguir o trajeto podem ser indutivos e/ou laser, entre outros. Já os sensores de segurança servem para parar o veículo a fim de evitar colisões e acidentes.

Do ponto de vista da robótica, podemos afirmar que um robô que não possui um sensor é simplesmente uma máquina. Isto é, o robô precisa ter sensores para deixar de ser uma máquina qualquer e passar a ser um dispositivo que percebe o mundo a sua volta e reage às mudanças de situações (Martin, 1992) - ou seja, os sensores auxiliam um robô na execução de tarefas, ajudando-o a determinar os parâmetros do ambiente que o cerca e do objeto a ser manipulado. (ROSÁRIO, 2005, p. 56-57).

Conforme o que é dito por Rosário (2005) percebe-se que os sensores são elementos fundamentais para AGVs, é por meio deles que o ambiente é percebido e sinais são enviados para serem processados. Em um AGV os principais tipos de sensores são os usados como elementos de visão e orientação.

#### 1.2.1.1 Sensor ultrassônico

Os sensores ultrassônicos são geralmente usados como sensores de proximidade. Ele funciona a partir da emissão de um som de alta frequência (entre 40 KHz e 2 MHz) que é refletido ao encontrar um obstáculo. É feita então a medida de tempo entre a emissão e retorno do som transmitido (ROSÁRIO, 2005). AGVs usam esse tipo de sensor para evitar a colisão com obstáculos ou usar os valores obtidos do sensor para realizar cálculos e desviar de objetos. A Figura 4 apresenta um modelo de sensor ultrassônico industrial.

**Figura 4** - Sensor ultrassônico Sick UC12

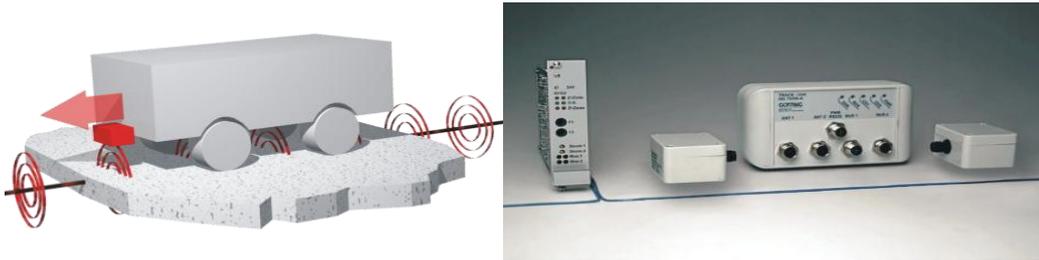


Fonte: <https://www.sick.com/ag/en/distance-sensors/ultrasonic-sensors/uc12/c/g185683>

### 1.2.1.2 Sensor indutivo

Uma fita metálica é colada no chão da fábrica desenhando o percurso que o AGV deve seguir. O sensor indutivo vai detectar a fita metálica e fazer com que o veículo percorra esse trajeto (ULLRICH, 2015). A Figura 5 apresenta o esquema de funcionamento e um equipamento de aplicação industrial.

**Figura 5 - Sensor indutivo Götting**

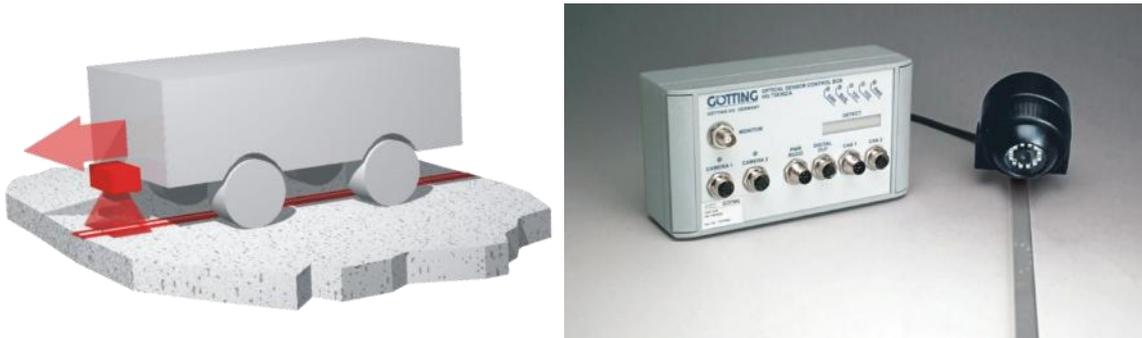


Fonte: <https://www.goetting-agv.com/solutions>

### 1.2.1.3 Sensor óptico

Os sensores ópticos aplicados em AGVs usam câmeras para identificar faixas no chão, sendo as imagens reconhecidas por meio de técnicas de processamento de imagem. É possível ainda reconhecer faixas codificadas ou marcadores ópticos (INTRODUCTION..., [201-]). A Figura 6 apresenta o esquema de funcionamento e um sensor óptico que pode ser usado em AGVs.

**Figura 6 - Sensor óptico Götting**



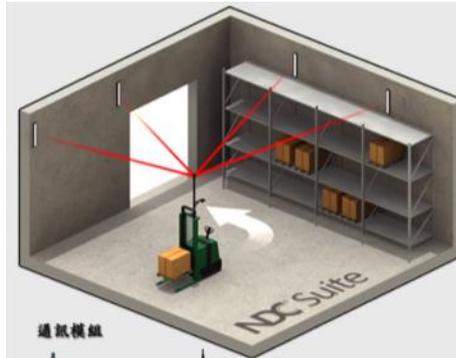
Fonte: <https://www.goetting-agv.com/solutions>

### 1.2.1.4 Sensor laser

Um scanner laser rotativo é colocado no topo do AGV para que o mesmo se oriente pelo espaço por meio da medição de distância e cálculo de posicionamento. Ao longo do percurso existem folhas refletivas colocadas em paredes e pilares, e é por meio da reflexão que o sensor laser se orienta e realiza os cálculos e medições de posicionamento (ULLRICH,

2015). Como o sensor gira 360° é possível gerar um mapeamento do ambiente (BEN-ARI; MONDADA, 2018). A Figura 7 apresenta o esquema de funcionamento desse sensor e a Figura 8 apresenta um modelo comumente usado em AGVs comerciais. A Figura 9 apresenta um exemplo do mapeamento de um ambiente feito por um scanner laser.

**Figura 7 - AGV com sensor de navegação laser**



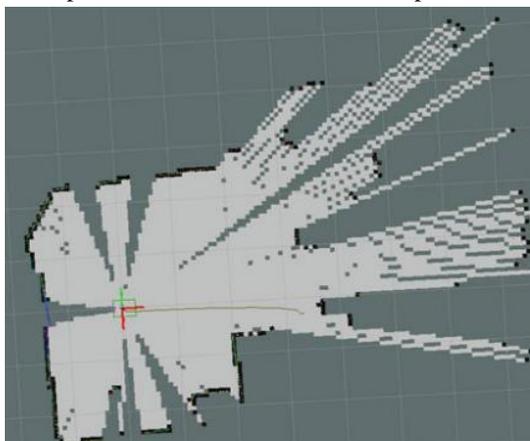
Fonte: <https://image.made-in-china.com/44f3j00hTkaEVACaduG/Laser-Guidance-Forklift-Agv-Laser-Guided-Vehicles.jpg>

**Figura 8 - Scanner laser Sick Nav 300**



Fonte: Ullrich (2015, p. 106)

**Figura 9 - Mapeamento de um ambiente feito por um scanner laser**



Fonte: Ben-Ari e Mondada (2018, p. 28)

### 1.2.1.5 GPS

Usado geralmente em AGVs que operam em ambientes externos, fornecendo a posição do veículo por meio de satélites (ULLRICH, 2015).

O GPS usualmente informa apenas a localização, em que a orientação é estimada, ou pela variação da posição do robô, ou através da integração desse tipo de dispositivo ao uso de um magnetômetro digital (bússola). O GPS é usado em ambientes externos, pois determina a localização através da triangulação de sinais de satélites, e a recepção desse tipo de sinais se torna inviável em ambientes fechados, ou mesmo dentro de túneis ou em áreas com cobertura vegetal muito densa (ROMERO et al, 2017, p. 22).

A Figura 10 apresenta o esquema de funcionamento do GPS e um modelo de GPS industrial.

**Figura 10 - GPS Götting**



Fonte: <https://www.goetting-agv.com/solutions>

### 1.2.1.6 Sensores de segurança

Os sensores mais usados para a segurança são o *bumper* e o scanner de segurança laser, sendo usados para parar o AGV. O *bumper* já não é mais tão utilizado, ele funciona de forma semelhante a um para-choque, ao encostar em uma pessoa ou estrutura o AGV para. O scanner de segurança laser varre a área ao seu redor e quando a barreira de luz é interrompida o AGV para (ULLRICH, 2015). A Figura 11 apresenta à esquerda um AGV equipado com um scanner de segurança laser e à direita um *bumper*.

**Figura 11 - Sensores de segurança**



Fonte: Ullrich (2015, p. 119)

### 1.2.1.7 Encoder

Um encoder converte movimento em uma sequência de pulsos digitais que podem ser convertidos em medidas de posição relativa ou absoluta (ALCIATORE; HISTAND, 2014). São usados pelos AGVs para calcular distâncias, saber posicionamento e controle de velocidade. A Figura 12 apresenta um modelo de encoder industrial.

**Figura 12** - Encoder absoluto Sick A3M60



Fonte: <https://www.sick.com/br/pt/encoders/encoder-absoluto/a3m60-profibus/c/g244409>

### 1.2.2 CLP

O CLP (Controlador Lógico Programável) é o cérebro do AGV. É ele que interpreta e processa os sinais recebidos pelos sensores e envia comandos aos motores, *drivers* e demais equipamentos ligados a sua saída. Um CLP é um componente eletrônico programável que realiza o processamento por meio de microprocessadores. A Figura 13 apresenta um modelo de CLP usado em AGVs comerciais.

**Figura 13** - CLP Siemens ET 200S com seus módulos

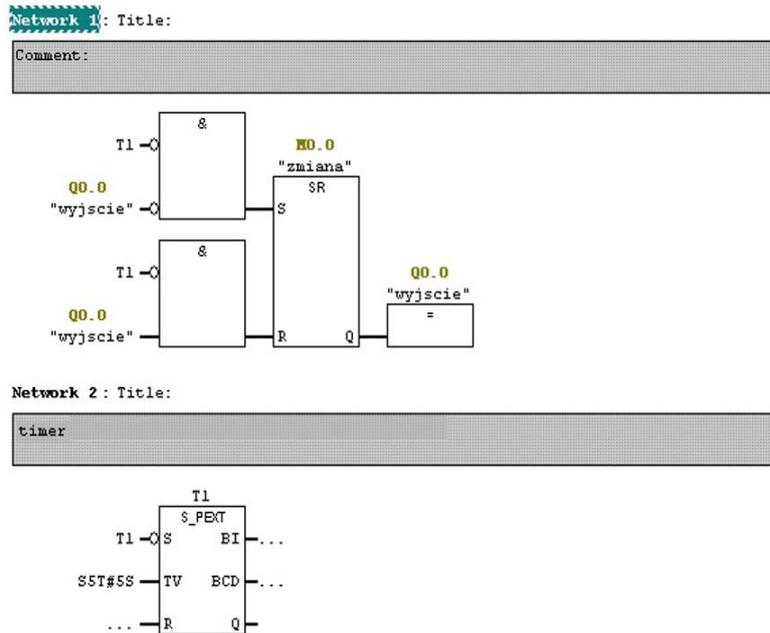


Fonte: <http://www.compel.in/Industrial%20Automation.html>

O CLP se conecta a outros equipamentos por meio de módulos específicos, como para a entrada e saída de sinais. Os sinais de entrada são recebidos pelo módulo de entrada, sendo então interpretados e processados pelo CLP e, com base em uma lógica desenvolvida pelo usuário, determina se serão enviados sinais para o módulo de saída (ROSÁRIO, 2015). A Figura 14 apresenta um esquema deste processo



**Figura 16 - Linguagem de diagrama de blocos (FBD)**



Fonte: Evghenii (2017)

**Figura 17 - Linguagem de lista de instruções**

OBI : "Main Program Sweep (Cycle)"

Comment:

Network 1: Title:

```

A   M   1.1
=   Q   4.1           //open valve 2
A   M   1.2
=   Q   4.2           //open valve 3

AN  M   1.0
BEC

L   MW  10           //load setpoint
ITD                               //Integer -> Double
T   MD  20
L   W#16#114         //load 276dez bzw. 114h
/D                               //MD 20 / 114h
T   MD  28
DTR                               //Double -> Real
T   MD  32

BE
    
```

Fonte: Evghenii (2017)

## CAPÍTULO 2 - PLATAFORMAS DE PROTOTIPAGEM ELETRÔNICA

Uma plataforma de prototipagem eletrônica consiste em uma placa com diversos componentes eletrônicos, projetada para a criação de diversos tipos de protótipos com os mais diferentes fins. Entre as aplicações estão projetos de automação residencial, comercial, industrial, robótica, segurança, brinquedos, entre outros (LOMBA, 2019).

### 2.1 Construção de AGVs utilizando plataformas eletrônicas de prototipagem

Em uma rápida pesquisa no buscador Google sobre o preço<sup>9</sup> de AGVs, é possível encontrar veículos na faixa de \$1.500,00 a mais de \$100.000,00. No geral, o custo destes veículos é alto por diversos fatores, mas principalmente por causa do valor elevado dos componentes utilizados.

O Arduino e o Raspberry Pi, entre outros, são plataformas eletrônicas que permitem a criação de projetos de baixo custo. Essas plataformas custam aproximadamente R\$60,00 e R\$200,00 (Arduino e Raspberry Pi, respectivamente) e permitem criar um grande número de projetos. Existem diversas literaturas que exploram a criação de AGVs em pequena escala utilizando essas plataformas.

McRoberts (2011) propõe a criação de um AGV que segue uma linha. É usado o Arduino e sensores LDR<sup>10</sup> (*Light Dependent Resistor* - Resistor Dependente de Luz, também conhecido como Fotorresistor) para percorrer o trajeto desenhado.

Grimmett (2014) possui literaturas onde constrói o mesmo projeto utilizando o Arduino e o Raspberry Pi. É criado um robô com rodas e em capítulo posterior é descrito sobre sensores infravermelho e sonar, e a configuração e programação para que o mesmo possa desviar de obstáculos. Em outro capítulo explora o uso do GPS para saber a localização do robô e também enviá-lo para um local específico.

Norris (2017) constrói um robô autônomo com o Raspberry Pi equipado com sensores sonar para detectar obstáculos e ao longo dos capítulos o aperfeiçoa utilizando *Machine Learning*<sup>11</sup> por meio de redes neurais artificiais e de *Deep Learning*<sup>12</sup>.

<sup>9</sup> Disponível em: <https://www.alibaba.com/showroom/agv-robot-price.html>. Acesso em 26 abr. 2019.

<sup>10</sup> Sensor cuja resistência varia de acordo com a quantidade de luz. Quanto mais luz, menor a resistência.

<sup>11</sup> Aprendizado de máquina. Sistema de aprendizado treinado ao invés de programado explicitamente. São apresentados exemplos para uma tarefa e então geradas estruturas estatísticas, conjuntos de dados grandes e complexos, que automatizam tarefas (CHOLLET, 2018).

<sup>12</sup> Aprendizado profundo. Faz parte do aprendizado de máquina. Redes neurais com um grande número de parâmetros e camadas (PATTERSON; GIBSON, 2017).

Os trabalhos de Grimmett e Norris são interessantes para a construção de um protótipo de AGV inteligente e autônomo. A junção do conhecimento das duas literaturas permite a construção de um veículo mais sofisticado e completo.

## 2.2 Microprocessador e Microcontrolador

Antes de iniciar a descrição das plataformas eletrônicas que podem ser utilizadas para a criação de um protótipo, será dada uma breve descrição sobre os elementos que as controlam, microprocessadores e microcontroladores, apresentando as diferenças e principais aplicações.

Um microprocessador é um componente eletrônico que executa tarefas específicas gravadas em sua memória. Pode se comunicar com o mundo real para coletar informações e atuar nele (NICOLOSI, 2013).

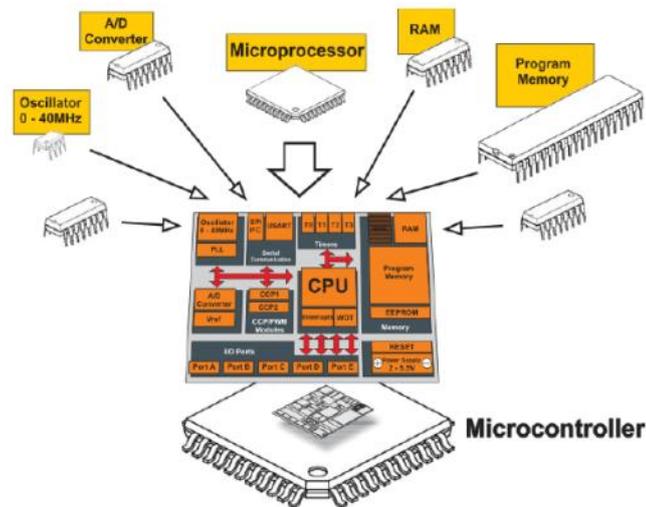
Um microcontrolador é um CI (Circuito Integrado) que integra as partes de um microcomputador (microprocessador, memórias e portas de E/S) em um único dispositivo, usado para o controle de tarefas simples e armazenar e manipular pequenas quantidades de dados (GIMENEZ, 2010).

A diferença entre microprocessador e microcontrolador, segundo Nicolosi (2013, p. 65) é: “Primeiramente o hardware interno do chip do microcontrolador (Mc) é diferente, isto é, tem mais funções que o do microprocessador (Mp).”.

[...] na maioria das aplicações utilizando microprocessadores necessita-se, além do chip do microprocessador, da ROM, do Latch, da RAM e de outros chips auxiliares, como Timers (que contam o tempo) e Serial (que faz interface com o barramento paralelo do microprocessador e gera uma linha serial de comunicação). O microcontrolador tipicamente já tem esses chips dentro da mesma pastilha do microprocessador. O microcontrolador corresponde a um microprocessador e seus periféricos típicos, todos juntos num só chip (NICOLOSI, 2013, p. 65).

A Figura 18 apresenta um esquema da diferença entre um microprocessador e um microcontrolador. O microcontrolador possui o microprocessador e outros componentes dentro do mesmo chip.

**Figura 18** - Diferença entre microprocessador e microcontrolador



Fonte: <http://microduvida.blogspot.com/p/microprocessadores-vs.html>

## 2.3 Plataformas para projetos eletrônicas

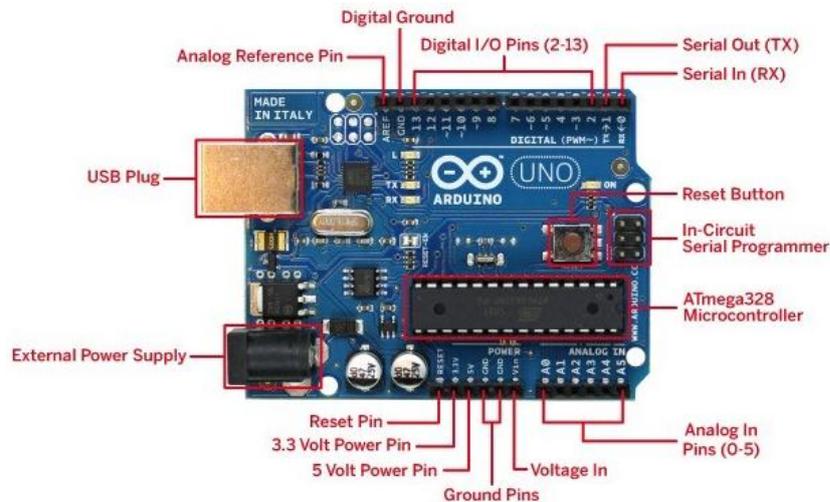
As duas plataformas mais populares atualmente existentes para a criação de projetos eletrônicos são o Arduino e o Raspberry Pi. Muitas literaturas as utilizam para a criação de projetos de AGVs em pequena escala, desde aplicações simples de robôs que seguem uma linha até os mais complexos, que utilizam técnicas de inteligência artificial.

### 2.3.1 Arduino

O Arduino surgiu na Itália, no *Ivrea Interaction Design Institute*. É uma plataforma eletrônica de código aberto que une hardware e software, sendo pensada inicialmente para ser usada por estudantes para a criação de projetos de forma rápida, mesmo sem conhecimentos aprofundados em eletrônica e programação (WHAT..., 201-). A Figura 19 apresenta a placa de um Arduino UNO com a identificação de seus componentes.

O Arduino é uma pequena placa de microcontrolador contendo um plugue de conexão USB (*Universal Serial Bus*), que permite a ligação com um computador, além de contar com um conjunto de pinos de conexão, que torna possível a ligação de dispositivos eletrônicos externos, como motores, relés, sensores luminosos, diodos a laser, alto-falantes, microfones e outros. Os Arduinos podem ser energizados pelo computador através de um cabo USB, por uma bateria de 9V ou por uma fonte de alimentação. Eles podem ser controlados diretamente pelo computador, ou então podem ser programados primeiro pelo computador e em seguida desconectados, permitindo assim que trabalhem independentemente do computador. O projeto da placa é aberto (*open source*). Isso significa que qualquer um tem permissão para construir placas compatíveis com o Arduino. Essa competição resultou em placas de baixo custo e em todo tipo de variantes a partir das placas “padrões”. As placas básicas são complementadas por placas acessórias denominadas *shields*, que podem ser encaixadas em cima da placa de Arduino (MONK, 2017, p. xv).

**Figura 19 - Arduino UNO**

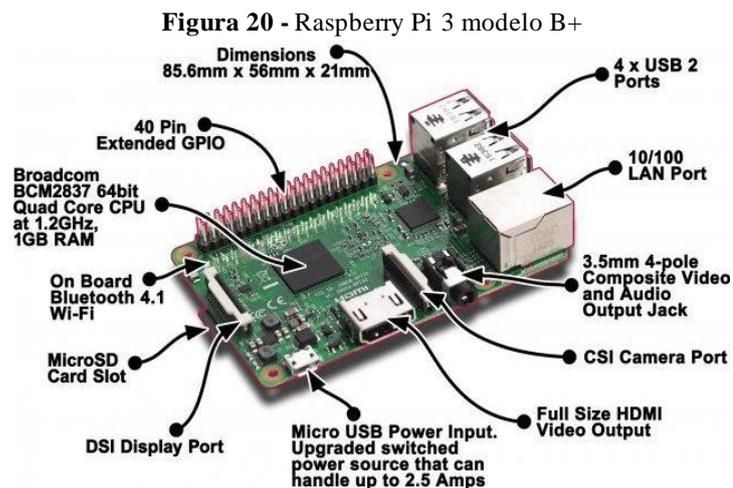


Fonte: <https://www.controlvoltage.net/arduino-arduino-uno-r3-development-board.html>

### 2.3.2 Raspberry Pi

O Raspberry Pi foi criado no Reino Unido e seu objetivo inicial era o ensino de programação para estudantes de Ciência da Computação (UPTON; HALFACREE, 2012). Diferente do Arduino, que é um microcontrolador, o Raspberry Pi é um microcomputador completo de baixo custo.

A Figura 20 apresenta a placa do Raspberry Pi modelo 3 B+ com a identificação de seus componentes.



Fonte: [https://www.researchgate.net/figure/Raspberry-Pi-3-model-B-v22\\_fig1\\_326090932](https://www.researchgate.net/figure/Raspberry-Pi-3-model-B-v22_fig1_326090932)

A fabricante de carros Peugeot Citroën, na sua fábrica em Portugal, criou um AGV utilizando o Raspberry Pi, sendo usado para o transporte de carros dentro da fábrica. O AGV é composto de um Raspberry Pi, um *Display Touch*, um Servo Controlador USB de 12 canais Pololu Maestro e um controlador de motor RoboClaw ST 2x45A, que conduzem os dois

motores DC. Um sensor de banda magnética Roboteq MGS1600 é usado para guiar o AGV ao longo da faixa magnética que percorre a fábrica. O robô usa um scanner Robopeak RPLidar 360° para segurança (PEUGEOT..., 2017). A Figura 21 apresenta o AGV construído pela Peugeot Citroën.

**Figura 21** - AGV construído pela Peugeot Citroën que usa o Raspberry Pi



Fonte: <https://www.raspberrypi.org/magpi/peugeot-citroen-robots/>

### 2.3.3 Comparativo entre Arduino e Raspberry Pi

O Arduino é um microcontrolador e o Raspberry Pi um microcomputador. O Quadro 3 apresenta as especificações técnicas dessas plataformas. Foram escolhidos os modelos Arduino UNO e Raspberry Pi 3 B+ por serem modelos mais populares e terem sido utilizados em diversas literaturas que abordam o projeto de robôs móveis.

**Quadro 3** - Especificação das plataformas Arduino e Raspberry Pi

Especificações	Arduino (modelo UNO)	Raspberry Pi (modelo 3 B+)
<b>Microcontrolador</b>	ATmega328P - 8 bit	-
<b>Processador</b>	-	Broadcom BCM2837B0
<b>CPU</b>	-	4× ARM Cortex-A53 - 64-bit
<b>Clock</b>	16 Mhz	1,4 Ghz
<b>Memória</b>	Flash 32KB, SRAM 2KB, EEPROM 1KB	1GB LPDDR2 SDRAM
<b>Tensão de operação</b>	5 V	5 V

(CONTINUA)

(CONTINUAÇÃO)

<b>Especificações</b>	<b>Arduino (modelo UNO)</b>	<b>Raspberry Pi (modelo 3 B+)</b>
<b>Tensão de entrada (recomendada)</b>	7-12 V	5 V
<b>Tensão de entrada (limite)</b>	6-20 V	-
<b>E/S digitais</b>	14	40
<b>E/S PWM<sup>13</sup></b>	6	-
<b>Entradas analógicas</b>	6	Não possui (Pode ser adicionado um conversor analógico)
<b>UBS</b>	1 porta USB 2.0	4 × USB 2.0 ports
<b>Conectividade</b>	Não possui (Podem ser adicionados módulos, conhecidos como <i>shield</i> )	2.4 GHz e 5 GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE  Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
<b>Linguagem de programação</b>	Baseada na linguagem C	Java, C, C++, C#, Python, etc.
<b>Preço médio</b>	R\$ 60,00	R\$ 200,00

Fonte: Adaptado da página de especificação<sup>14</sup> das plataformas eletrônicas

O Arduino, por ser um microcontrolador, pode executar apenas uma tarefa por vez, geralmente repetitiva, já o Raspberry Pi é um computador de uso geral, possui um sistema operacional e é capaz de executar várias tarefas ao mesmo tempo, além de poder realizar cálculos complexos (DI JUSTO, 2015).

<sup>13</sup> Técnica utilizada na eletrônica que altera a largura de uma onda quadrada permitindo dessa forma o controle de potência ou velocidade.

<sup>14</sup> Disponível em: [http://eventos.ifc.edu.br/micti/wp-content/uploads/sites/5/2014/08/ARDUINO%20AE-E-RASPBERRY-PI%20AE-UMA-COM PAR A%20C3%87%20C3%83O-DE-ESPECIFICA%20C3%87%20C3%95ES-E-APLICA%20C3%87%20C3%95ES-DE-MINICOMPUTADORES.pdf](http://eventos.ifc.edu.br/micti/wp-content/uploads/sites/5/2014/08/ARDUINO%20AE-E-RASPBERRY-PI%20AE-UMA-COM%20PAR%20A%20C3%87%20C3%83O-DE-ESPECIFICA%20C3%87%20C3%95ES-E-APLICA%20C3%87%20C3%95ES-DE-MINICOMPUTADORES.pdf). Acesso em 30 mai. 2019.  
Disponível em: <https://www.arduino.cc/en/products/compare>. Acesso em 30 mai. 2019.

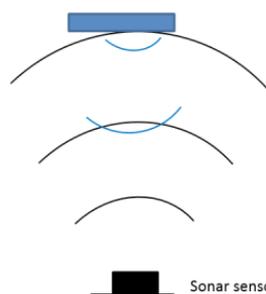
## 2.4 Sensores usados em plataformas de prototipagem eletrônica

Os sensores geralmente usados na construção de protótipos utilizando plataformas como o Arduino ou o Raspberry Pi são o sensor sonar (também conhecido como ultrassônico) e o sensor infravermelho. Esses sensores são usados para detectar objetos e, com base nos sinais recebidos e no processamento, o AGV realiza a ação, que pode ser a de desvio ou a de parada.

Norris (2017, p. 152, tradução nossa) afirma que “Tecnicamente, o robô não está realmente evitando obstáculos, porque isso implicaria um caminho predeterminado. Está evitando todas as superfícies de contenção, na verdade, quaisquer paredes e portas próximas.”. O robô desenvolvido por Norris usa um sensor ultrassônico para detectar os objetos à frente. As ações que o robô pode executar são: Seguir em frente; Virar à direita; Virar à esquerda.

Grimmett (2014, p. 112-113, tradução nossa) descreve o uso de dois tipos de sensores para que o robô desvie de obstáculos, são sensores que medem a distância até um objeto. Os sensores usados são o sensor sonar e sensor infravermelho (IR). O sensor sonar “[...] emite um som ultrassônico para calcular a distância até um objeto”. Se um objeto aparece no caminho da onda emitida pelo sonar, ela é refletida de volta ao sensor e é então calculado o tempo de envio e retorno da onda, sendo medida então a distância até o objeto. A Figura 22 apresenta um diagrama do funcionamento do sensor ao emitir a onda e encontrar um objeto e as Figuras 23 e 24 modelos comerciais usados em plataformas eletrônicas.

**Figura 22** - Diagrama de funcionamento do sensor sonar



Fonte: Grimmett (2014, p. 112)

**Figura 23** - Sensor sonar HC-SR04



Fonte: <https://www.filipeflop.com/produto/sensor-de-distancia-ultrassonico-hc-sr04/>

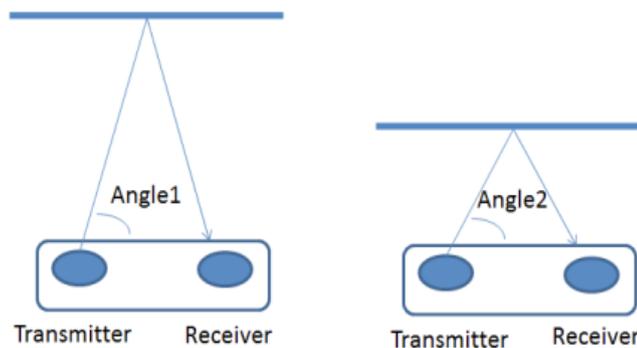
**Figura 24** - Sensor sonar MB1414 USB-ProxSonar-EZ1



Fonte: [https://www.maxbotix.com/Ultrasonic\\_Sensors/MB1414.htm](https://www.maxbotix.com/Ultrasonic_Sensors/MB1414.htm)

O sensor infravermelho emite uma luz infravermelha para detectar a distância. Existe um transmissor e um receptor, a diferença no ângulo da luz emitida e recebida indica a distância até o objeto (GRIMMETT, 2014). A Figura 25 apresenta o diagrama da transmissão e recepção do sensor infravermelho ao atingir um objeto e a Figura 26 um modelo comercial.

**Figura 25** - Diagrama de funcionamento do sensor infravermelho



Fonte: Grimmett (2014, p.113)

**Figura 26** - Sensor Infravermelho Sharp GP2Y0A02YK



Fonte: [https://www.sgbotic.com/index.php?dispatch=products.view&product\\_id=22](https://www.sgbotic.com/index.php?dispatch=products.view&product_id=22)

O Quadro 4 possui a especificação de alguns sensores sonar e infravermelho comerciais usados na construção de protótipos eletrônicos.

**Quadro 4** - Sensores comerciais usados em plataformas de prototipagem eletrônica

Especificações	Sensor		
	Sonar HC-SR04	Sonar MB1414 USB- ProxSonar-EZ1	Infravermelho Sharp GP2Y0A02YK
<b>Distância de medição</b>	2 cm a 4 m	1 mm a 150 cm	20 cm a 150 cm
<b>Tensão de operação</b>	5V	-	4,5 V a 5,5 V
<b>Tipo de saída</b>	-	-	Tensão analógica
<b>Preço</b>	R\$ 11,90	US\$ 49,95	R\$ 109,00

Fonte: Folha de especificação técnica<sup>15</sup> dos sensores

<sup>15</sup> Sonar HC-SR04 . Disponível em: <https://www.mouser.com/ds/2/813/HCSR04-1022824.pdf>. Acesso em 30 mai. 2019.

Sonar MB1414 USB-ProxSonar-EZ1. Disponível em: [https://www.maxbotix.com/documents/USB-ProxSonar-EZ\\_Datasheet.pdf](https://www.maxbotix.com/documents/USB-ProxSonar-EZ_Datasheet.pdf). Acesso em 30 mai. 2019.

Infravermelho Sharp GP2Y0A02YK. Disponível em: <https://multilogica-shop.com/sensor-de-dist%C3%A2ncia-sharp-gp2y0a02yk-20-150cm> . Acesso em 30 mai. 2019.

## CAPÍTULO 3 - INTELIGÊNCIA ARTIFICIAL

### 3.1 Diferença entre AGV convencional e inteligente

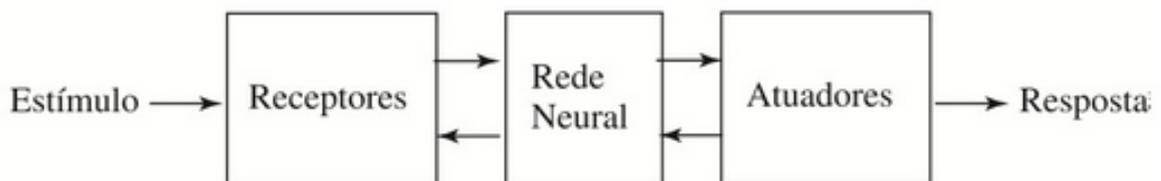
Em geral um AGV não possui inteligência, ele apenas segue rotas e ações programadas. Para realizar o aprendizado e tomar decisões é necessário o uso de inteligência artificial (IA). Com o uso de IA o AGV pode realizar o mapeamento do local e o desvio de obstáculos da melhor forma possível, além de evitar a colisão com objetos e pessoas. Para entrar nesse assunto é interessante abordar o funcionamento do sistema nervoso e redes neurais biológicas e posteriormente os sistemas artificiais.

### 3.2 Sistema nervoso biológico

O sistema nervoso é responsável por estabelecer a comunicação entre o meio em que o indivíduo vive, o mundo ao seu redor e as partes internas do organismo. O cérebro, parte mais importante do sistema nervoso, é quem toma “consciência das informações que chegam pelos órgãos dos sentidos” e processa as informações. Essas informações são então comparadas com “nossas vivências e expectativas” e então são emitidas respostas para que se atue sobre o ambiente (COSENZA; GUERRA, 2011, p. 11).

O sistema nervoso é composto de três estágios: Receptores; Rede Neural; Atuadores, como pode ser verificado na Figura 27. Os receptores “convertem estímulos do corpo humano ou do ambiente externo em impulsos elétricos que transmitem informação para a rede neural (cérebro)”. A rede neural, que está localizada no centro, é quem recebe constantemente as informações e toma as decisões necessárias com base em sua percepção. Por fim se tem os atuadores, que “convertem impulsos elétricos gerados pela rede neural em respostas discerníveis como saídas do sistema.” (HAYKIN, 2007, p. 32).

**Figura 27** - Diagrama de blocos do sistema nervoso biológico

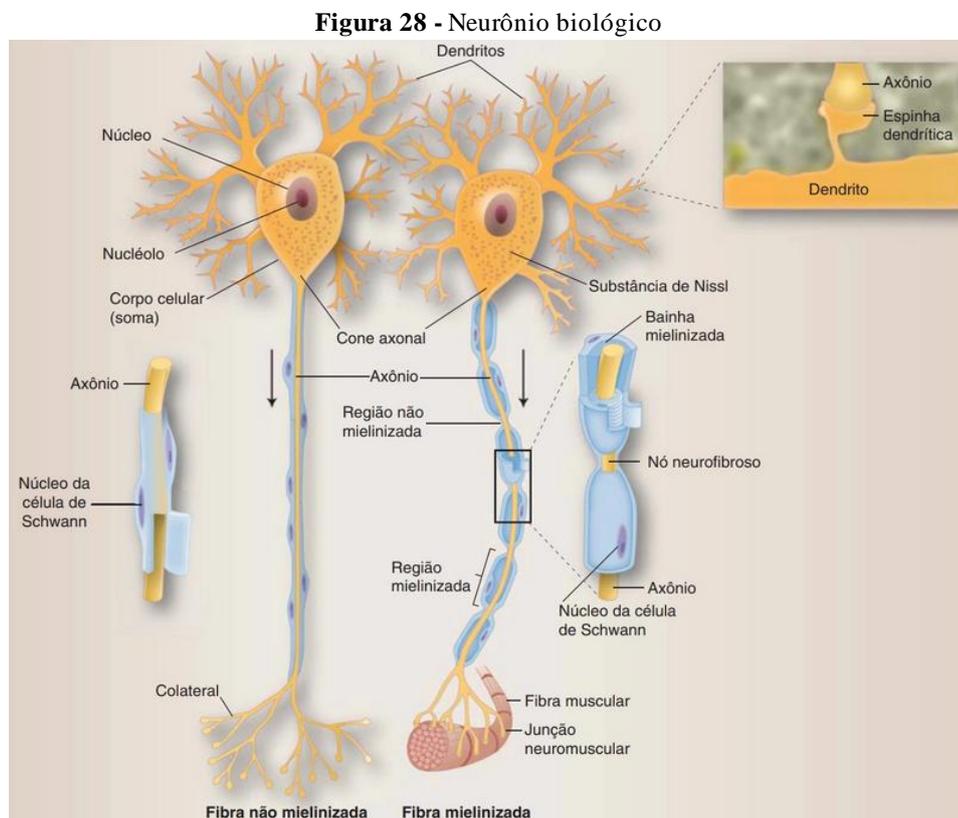


Fonte: Haykin (2007, p. 32)

### 3.2.1 Neurônio biológico

Kandel et al. (2014, p. 1378) afirma que: “Um único neurônio, por si só, não é inteligente. Contudo, uma vasta rede de neurônios é capaz de pensar, sentir, lembrar, perceber e produzir muitos fenômenos notáveis, conhecidos coletivamente como “a mente”.”.

Os neurônios fazem parte do sistema nervoso, sendo células excitáveis que geram, transmitem e recebem sinais elétricos. Eles enviam sinais que se propagam por meio de “potenciais de ação ou impulsos elétricos”. Os neurônios se comunicam entre si por meio de sinapses, que é o contato entre dois neurônios por meio da “terminação axônica de um neurônio e o dendrito de outro” sendo capazes de controlar o fluxo de informações, formando uma estrutura em rede que processa e armazena as informações. Os dendritos são os responsáveis por “receber as informações, ou impulsos nervosos” dos outros neurônios e enviar para o corpo celular (também conhecido como soma) onde a informação será processada e com isso novos impulsos serão gerados, sendo transmitidos a outros neurônios por meio dos axônios até os dendritos dos neurônios seguintes (KREBS; WEINBERG; AKESSON, 2013, p. 13, 16; BRAGA; DE CARVALHO; LUDERMIR, 2014, p. 6; AWAD; KHANNA, 2015, p. 129). A Figura 28 apresenta a estrutura de um neurônio biológico.



Fonte: Krebs, Weinberg e Akesson (2013, p. 15)

A comunicação entre os neurônios é feita pela liberação de uma substância química, um neurotransmissor, que é liberado na região das sinapses. Com isso podem ocorrer dois efeitos: excitar a membrana da outra célula fazendo com que impulsos nervosos sejam disparados ou dificultar o disparo de novos impulsos nervosos, ou seja, o efeito pode ser excitatório ou inibitório (COSENZA; GUERRA, 2011).

Um neurônio combina os sinais recebidos pelos neurônios anteriores e, caso esses sinais “estejam acima do limiar de excitação do neurônio, um impulso elétrico é produzido e propagado por meio do axônio para os neurônios seguintes.”. O disparo de um neurônio é estabelecido pelo grande número de entradas excitatórias e inibitórias “[...] medido pelo corpo da célula em um pequeno intervalo de tempo.”. Após o impulso, ou disparo, o neurônio fica um período em repouso se preparando para a geração de um novo impulso (BRAGA; DE CARVALHO; LUDERMIR, 2014, p. 7-8).

Em resumo, um neurônio biológico pode ser visualizado do ponto de vista funcional da seguinte forma: as suas múltiplas entradas recebem ativações excitatórias ou inibitórias dos neurônios anteriores, e caso essa soma de excitações e inibições ultrapasse um determinado limite, o neurônio emite um impulso nervoso (BRAGA; DE CARVALHO; LUDERMIR, 2014, p. 8).

Com base no funcionamento do neurônio biológico surgiram os estudos para o desenvolvimento de neurônios artificiais. O primeiro modelo de neurônio artificial surgiu dos estudos do psicólogo e neurofisiologista Warren McCulloch e do matemático Walter Pitts (BRAGA; DE CARVALHO; LUDERMIR, 2014). O neurônio artificial é apresentado no próximo subtítulo.

### **3.3 Neurônio artificial**

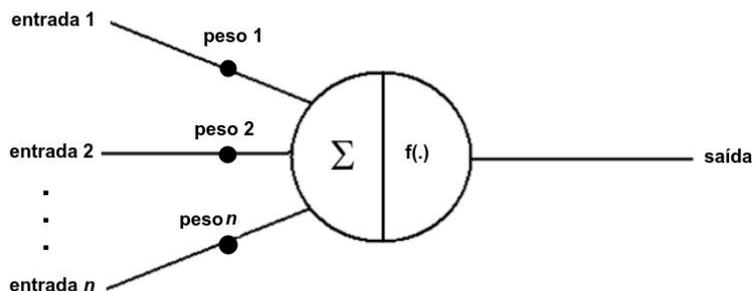
O primeiro modelo de neurônio artificial foi criado por Warren McCulloch e Walter Pitts em 1943 com base na fisiologia e função dos neurônios biológicos. O trabalho de McCulloch e Pitts descrevia o modelo matemático do neurônio artificial e as suas capacidades computacionais (BRAGA; DE CARVALHO; LUDERMIR, 2014).

O modelo de neurônio artificial de McCulloch e Pitts (que passou a ser conhecido como modelo MCP devido a **McCulloch** e **Pitts**) possuía uma descrição matemática onde o modelo tinha  $n$  terminais de entrada (representando os dendritos do neurônio biológico e as ativações dos neurônios anteriores) e um terminal de saída (que representa o axônio do neurônio biológico). Os terminais de entrada possuem pesos conectados que representam o comportamento das sinapses, esses pesos podem ter valores positivos ou negativos que corresponderão às sinapses serem inibitórias ou excitatórias. Esses pesos que determinam

quando sinais de disparo devem ocorrer naquela conexão (BRAGA; DE CARVALHO; LUDERMIR, 2014).

Dois neurônios de McCulloch-Pitts podem ser conectados de modo que a saída de um neurônio seja o sinal de entrada do outro. Isso corresponde ao fato biológico de que sinapses excitatórias convergindo sobre um neurônio são ativadas pela descarga dos neurônios pré-sinápticos. Fazendo-se muitas dessas conexões, é possível construir um modelo de rede neural (KENDAL et al., 2014, p. 1380).

**Figura 29 - Modelo de neurônio artificial MCP**



Fonte: Adaptado de Norris (2017, p. 172)

A Figura 29 apresenta um modelo do neurônio MCP e funciona da seguinte forma segundo Braga, De Carvalho e Ludermir (2014, p. 8-9), Haykin (2007, p. 36-37) e Kandel et al. (2014, p. 1380):

1. O neurônio recebe múltiplos sinais de entrada e produz um único sinal de saída.
2. Todos os sinais de entrada e o sinal de saída são variáveis binárias, 0 ou 1.
3. Cada variável de entrada representa a ativação de uma sinapse excitatória sobre o neurônio. O sinal de entrada é igual a 1 quando a sinapse excitatória está ativada.
4. O neurônio é caracterizado por um único parâmetro  $\theta$ , seu limiar. O parâmetro  $\theta$  é usado para representar o limiar de um neurônio biológico e é igual ao número mínimo de sinapses excitatórias que devem ser ativadas simultaneamente para produzir um potencial de ação. Se um subconjunto de  $\theta$  ou mais sinais de entrada for igual a 1, então o sinal de saída do neurônio é 1; de outra forma, o sinal de saída é 0.
5. Quando a soma ( $\Sigma$ ) dos impulsos recebidos na entrada ultrapassa um limiar de excitação (*threshold*) ocorre o disparo. Essa soma é feita por entrada  $i$  peso  $i$ , sendo feita uma soma ponderada que determina se o neurônio deve ou não disparar (saída 1 ou 0) comparando com a soma do limiar ou *threshold* do neurônio;
6. A ativação do neurônio é obtida por meio da aplicação de uma função de ativação  $f(.)$  que ativa ou não a saída, dependendo do valor da soma ponderada das suas entradas. A função gera a saída a partir dos vetores de peso e de entrada.

### 3.4 Função de ativação

A função de ativação é a responsável por gerar a saída do neurônio por meio dos valores dos vetores de peso e de entrada (BRAGA; DE CARVALHO; LUDERMIR, 2014). Ela que determina qual valor é alto ou baixo o suficiente para a decisão na rede neural, para um neurônio ou grupo de neurônios específico (MUELLER; MASSARON, 2019). A seguir são apresentadas as funções utilizadas no projeto.

#### 3.4.1 Função ReLU

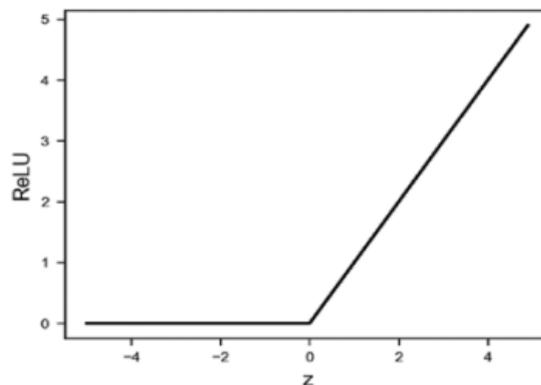
A função *ReLU* (Unidade Linear Retificada do inglês *Rectified Linear Unit*) é dada pela Equação 1 (MICHELUCCI, 2018, p. 42):

$$f(z) = \max(0, z) \quad (1)$$

A saída da função *ReLU* é uma matriz de zeros com alguns valores positivos. Um dos principais benefícios dessa função é que ela produz uma matriz esparsa como saída. É muitas vezes usada como função de ativação da camada oculta da rede neural (BEYSOLOW II, 2019).

Essa função tem sido muito usada porque tenta resolver o problema de fuga do gradiente, um problema que ocorre quando  $z > 0$  e, à medida que  $z$  aumenta, o gradiente dos sigmóides vai se tornando muito pequeno até quase desaparecer. A função *ReLU* impede que isso aconteça, além de ajudar em uma convergência mais rápida da descida do gradiente (SARKAR; BALI; SHARMA, 2018). A Figura 30 apresenta o gráfico dessa função. Quando  $z < 0$  o valor é zero, quando  $z > 0$  o valor é  $z$ .

**Figura 30** - Gráfico da função de ativação ReLU



Fonte: Michelucci (2018, p. 43)

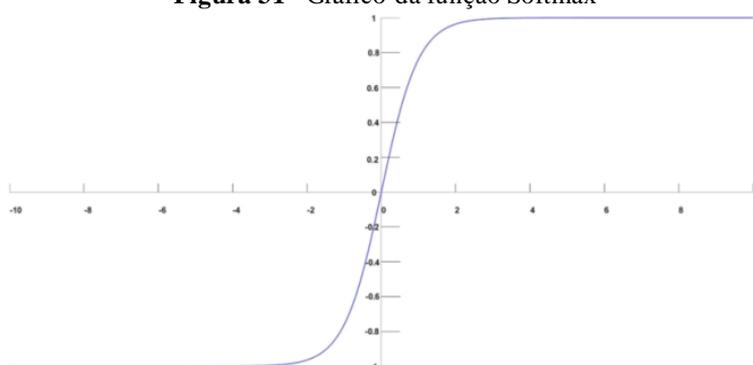
Como vantagem, a função *ReLU* requer menos poder de processamento, porque menos neurônios disparam. A desvantagem é que após um tempo pode ocorrer de os pesos da rede neural não terem mais efeito, parando de aprender, e os neurônios afetados não respondem mais a nenhuma entrada (MUELLER; MASSARON, 2019).

### 3.4.2 Função Softmax

A função *Softmax* é usada em redes neurais de classificação e pode ser usada para representar uma distribuição de probabilidade. A função pode ajustar a probabilidade de cada classe com base nos valores recebidos de uma camada anterior (SARKAR; BALI; SHARMA, 2018; MUELLER; MASSARON, 2019).

Essa função converte as saídas de cada uma das classes em um número entre 0 e 1 e, em seguida, divide o valor das saídas para que a soma de todas as saídas sejam iguais a 1 (GOVERS, 2018). A Figura 31 apresenta o gráfico da função *Softmax*.

**Figura 31** - Gráfico da função Softmax



Fonte: [https://www.researchgate.net/figure/Softmax-function-image\\_fig1\\_325856086](https://www.researchgate.net/figure/Softmax-function-image_fig1_325856086)

A função *Softmax* é dada pela Equação 2:

$$\varphi_i = \frac{e^{z_i}}{\sum_{j \in \text{grupo}} e^{z_j}} \quad (2)$$

Nessa equação,  $i$  representa o índice do neurônio de saída ( $o$ ) sendo calculado e  $j$  representa os índices de todos os neurônios de um nível. A variável  $z$  designa o vetor de neurônios de saída. Vale notar que a função de ativação softmax é calculada de forma diferente das demais apresentadas, uma vez que a saída de um neurônio depende dos outros neurônios de saída (REIS, 2016).

Segundo Neilsen (2019, tradução nossa): “A função softmax é idealmente usada na camada de saída do classificador, onde realmente estamos tentando gerar as probabilidades para definir a classe de cada entrada.”

## 3.5 Redes neurais artificiais

Segundo Kandel et al. (2014, p. 1378) os modelos computacionais do cérebro, cujos estudos surgiram por volta dos anos 1940, é baseado em dois princípios:

1. A experiência imediata resulta de padrões contínuos de potenciais de ação em células cerebrais.
2. A capacidade de aprender e lembrar experiências passadas resulta, pelo menos em parte, de modificações duradouras em conexões sinápticas.

Uma das formas de inteligência artificial que podem ser aplicadas em AGVs são as redes neurais. Uma rede neural pode ser definida como:

[...] um rede neural é uma máquina que é projetada para modelar a maneira como o cérebro realiza uma tarefa particular ou função de interesse; a rede é normalmente implementada utilizando-se componentes eletrônicos ou é simulada por programação em um computador digital (HAYKIN, 2007, p. 28).

Ainda segundo Haykin (2007, p. 75), a principal característica de uma rede neural é “[...] a sua habilidade de aprender a partir de seu ambiente e de melhorar seu desempenho por meio da aprendizagem. A melhoria do desempenho ocorre com o tempo de acordo com alguma medida preestabelecida.”. A aprendizagem ocorre na seguinte sequência:

1. A rede neural é estimulada por um ambiente.
2. A rede neural sofre modificações nos seus parâmetros livres como resultado desta estimulação.
3. A rede neural responde de uma maneira nova ao ambiente, devido às modificações ocorridas na sua estrutura interna.

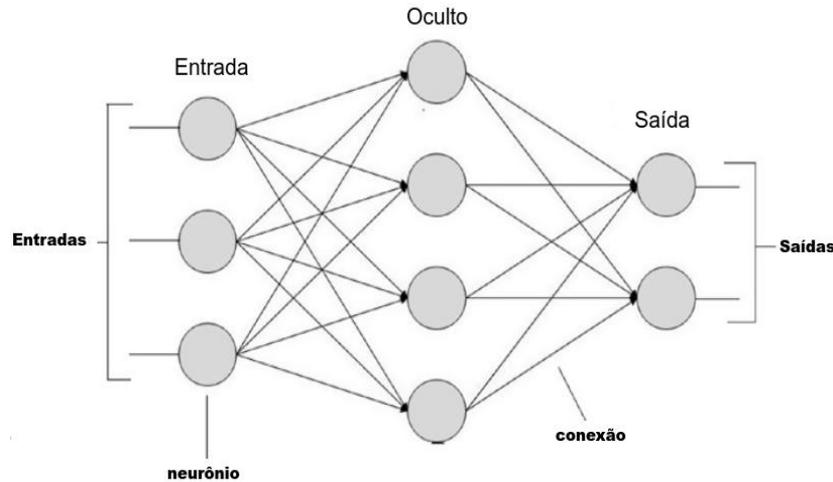
Braga, De Carvalho e Ludermir (2014, p. 3) afirmam que a solução de um problema por meio de uma rede neural passa por um processo de aprendizagem, a capacidade de aprender por meio de exemplos, onde “um conjunto de exemplos é apresentado para a rede, a qual extrai as características necessárias para representar a informação fornecida. Essas características são utilizadas posteriormente para gerar respostas para o problema.”.

Kandel et al. (2014, p. 1378) afirma que uma rede neural geralmente começa com a matriz de pesos inicializada com valores aleatórios. A rede é então, tal como expressado por Braga, De Carvalho e Ludermir, apresentada a uma série de estímulos, onde cada um desses estímulos faz a “matriz de pesos se modificar conforme determinado pela regra do aprendizado”.

Em um computador, ao escrever um programa, é dito passo-a-passo o que ele deve fazer. Previamente se sabe o que ele vai realizar, ele apenas segue as instruções. Já em um programa para simular uma rede neural não é preciso saber exatamente como resolver um problema. A rede é treinada por meio de várias entradas e saídas corretas, onde no início

haverá muitos erros e incongruências, mas com ajustes a rede aprende a resolver o problema de forma correta (MCKEON, 2018). A Figura 32 apresenta um exemplo de rede neural.

**Figura 32** - Exemplo de uma rede neural de três camadas de neurônios



Fonte: Adaptado de Norris (2017, p. 45)

### 3.5.1 Perceptron de camada única

Segundo Russell e Norvig (2013, p. 729) “Uma rede com todas as entradas conectadas diretamente com as saídas é chamada de rede neural de camada única ou rede perceptron.”.

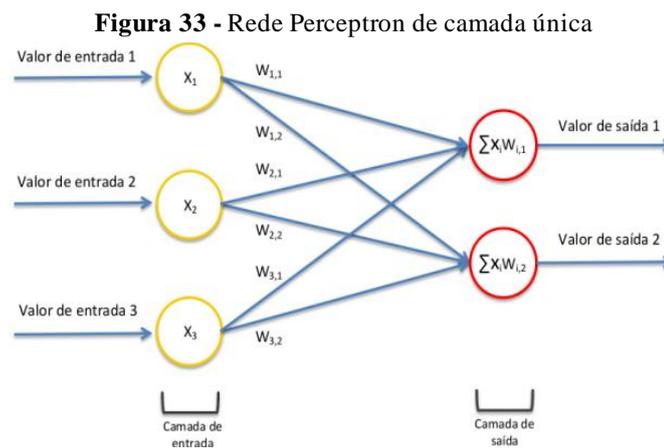
As redes Perceptron utilizam o modelo de neurônio artificial de McCulloch e Pitts, sendo uma rede de camada única, a primeira rede neural artificial implementada e a que introduziu o processo de treinamento de RNAs, por meio do trabalho de Frank Rosenblatt em 1958 (FACELI et al, 2011; BRAGA; DE CARVALHO; LUDERMIR, 2014).

Segundo Coppin (2013, p. 258) e Lugen (2013, p. 379) o aprendizado de uma rede Perceptron é do tipo supervisionado e funciona da seguinte forma:

1. Pesos aleatórios são atribuídos às entradas;
2. Um item de dados de treinamento é apresentado ao perceptron e a sua classificação de saída é observada;
3. Após tentar resolver uma ocorrência do problema, é apresentado o resultado correto;
4. Se os valores de saída, desejado e real, forem iguais, os pesos não são alterados;
5. Se a saída estiver incorreta os pesos são ajustados, incrementando ou decrementando os pesos para tentar classificar mais acertadamente a entrada;

6. Uma vez que esta modificação dos pesos tenha ocorrido, o próximo fragmento de dados de treinamento será empregado do mesmo modo.
7. Uma vez que todos os dados de treinamento tenham sido aplicados, o processo é iniciado novamente, até que todos os pesos estejam corretos e todos os erros sejam zerados. Cada iteração deste processo é conhecida como uma época.

A Figura 33 apresenta uma rede Perceptron simples com três entradas e duas saídas. Cada entrada está conectada aos dois neurônios de saída, e cada uma dessas conexões possui um peso, que é um valor que será ajustado durante o treinamento e utilizado na tomada de decisão. Os neurônios são ativados pelo somatório dos valores de entrada multiplicado pelos respectivos pesos ( $\sum x_i w_{i,1}$  e  $\sum x_i w_{i,2}$ ) mais uma função de ativação. Caso o somatório atinja o valor de limiar de ativação do neurônio, o mesmo dispara e existe um sinal na saída que faz com que uma ação seja realizada pelo agente.



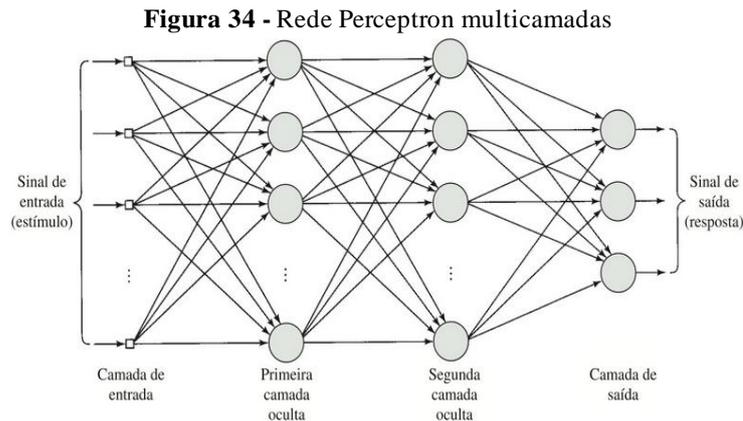
### 3.5.2 Perceptron de múltiplas camadas

As redes Perceptron de múltiplas camadas, conhecidas como MLP (do inglês *Multilayer Perceptron*), possuem um poder computacional maior do que o de redes de uma única camada. As múltiplas camadas da rede MLP tem o papel de “transformar, sucessivamente, o problema descrito pelo conjunto de dados no espaço de entrada em uma representação tratável para a camada de saída da rede.” (BRAGA; DE CARVALHO; LUDERMIR, 2014, p. 67-68).

Nesse tipo de rede existe uma ou mais camadas ocultas entre as camadas de entrada e saída. Essas camadas ocultas permitem à rede neural aprender mais, se adaptando de forma

mais eficaz ao problema que está tentando ser resolvido (NELLI, 2018). A Figura 34 apresenta uma rede Perceptron multicamadas com duas camadas ocultas.

As camadas intermediárias de uma MLP geram uma codificação interna para os padrões de entrada, que é então utilizada pela camada de saída da rede. Dado um número suficientemente grande de unidades intermediárias, é possível formar representações internas para qualquer conjunto de padrões de entrada. [...] uma camada intermediária é suficiente para aproximar qualquer função contínua, e duas camadas são suficientes para aproximar qualquer função (BRAGA; DE CARVALHO; LUDERMIR, 2014, p. 70).



Fonte: Haykin (2007, p. 186)

Braga, De Carvalho e Ludermir (2014, p. 71) afirmam que é necessário determinar a função de ativação de cada camada e a quantidade de neurônios em cada uma, sendo que “pelo menos uma das camadas intermediárias deverá ter funções de ativação não-lineares.”. Nas camadas intermediárias são utilizadas funções sigmoidais e na saída utilizadas funções sigmoidais ou lineares. “Normalmente, as funções lineares na saída serão utilizadas em problemas de aproximação de funções, e as funções sigmoidais em problemas de classificação, apesar de esta não ser uma regra geral.”.

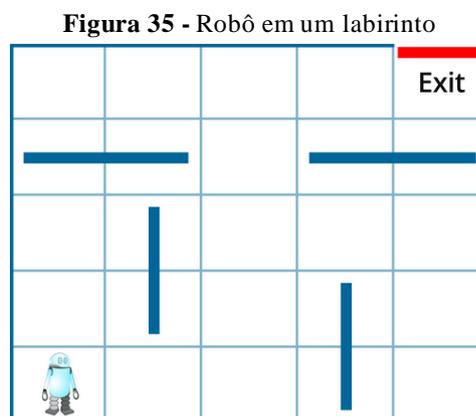
Ainda segundo Braga, De Carvalho e Ludermir (2014, p. 71-72) “O número de neurônios determina a capacidade da rede em resolver problemas de determinada complexidade. Quanto maior o número de neurônios, maior a complexidade da rede e maior a sua abrangência em termos de soluções possíveis.”. Apesar de ser necessário determinar a quantidade de neurônios de cada camada, não há uma regra que determine “qual deve ser o número de neurônios de uma rede neural para a resolução de um determinado problema.”. O que existe são abordagens para estimar o tamanho da rede ou analisar a sua resposta. O que se pode fazer é criar a estrutura da rede com um “tamanho mínimo” e ir adicionando neurônios gradualmente até que se atinja o limite de tolerância de erro.

### 3.5.3 Aprendizado por reforço

O aprendizado por reforço é um método muitas vezes considerado como um “caso particular de aprendizado supervisionado”, ou seja, existe um “supervisor” ou “professor” externo “responsável por estimular as entradas da rede por meio de padrões de entrada e observar a saída calculada pela mesma, comparando-a com a saída desejada.”. Os pesos da rede são ajustados aproximando os valores das saídas com os valores desejados. Acontece que existe uma diferença na atualização dos pesos entre o aprendizado supervisionado e o por reforço. No supervisionado a atualização dos pesos é baseada em “um critério de desempenho como o erro, por exemplo”, já no por reforço se “procura maximizar o reforço das ações boas executadas pela rede” (BRAGA; DE CARVALHO; LUDERMIR, 2014, p. 13, 15).

Um exemplo de aprendizado por reforço é apresentado na citação a seguir e no exemplo da Figura 35:

Imagine que um robô (também conhecido como agente) está tentando pegar uma caneta e ele falha. Tenta novamente, falha. Depois de repetir esse processo 1000 vezes ele finalmente consegue. O agente agora aprendeu a pegar uma caneta. Em poucas palavras, isso é um aprendizado de reforço, é muito parecido com o modo como as criaturas vivas aprendem (QU, 2018a, tradução nossa).



Fonte: Qu (2018a)

A Figura 35 apresenta um robô em um labirinto na posição (1,1). O robô pode realizar um conjunto de ações como se mover para cima ou se mover para a direita. Ao executar uma ação o robô recebe uma recompensa, que é definida pelo programador e pode ser:

- Ir para um quadro vazio: +1
- Bater na parede: -1
- Chegar na saída: +100

Ao executar uma ação (a) no estado (s) recebe uma recompensa (r). Ele atualiza constantemente a sua memória de recompensas ao tomar certas ações nos estados. No início

do treinamento as recompensas são definidas como 0 (zero), como ainda não tem experiência o robô realizará ações aleatórias e, ao longo de várias iterações, começa a aprender lentamente as recompensas para cada ação tomada. O robô pode então procurar o valor mais alto e executar aquela ação (QU, 2018a).

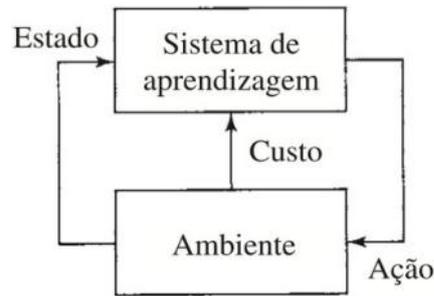
Ainda segundo Qu (2018a) existem três termos chave na aprendizagem por reforço, que são:

1. Estado: descreve a situação atual. (por exemplo: posição em um labirinto).
2. Ação: O que o agente pode fazer em uma situação (por exemplo, Mover para a direita).
3. Recompensa: Feedback para saber se uma ação específica em um determinado estado foi boa ou ruim.

Nesse tipo de aprendizado “um mapeamento entrada-saída é realizado por meio da interação contínua com o ambiente”, sendo baseado em comportamento onde “o sistema procura alcançar um objetivo específico apesar da presença de incertezas.”. O aprendizado do agente acontece por meio do que é chamado de recompensa, que dependerá da forma como o agente irá operar no ambiente para atingir o seu objetivo. Se operar corretamente então receberá uma recompensa positiva, caso contrário a recompensa será negativa (HAYKIN, 2007, p. 89, 651; COPPIN, 2013, p. 250).

Um dos maiores desafios no aprendizado por reforço é projetar recompensas que induzem os agentes a aprender os comportamentos desejados. [...] Quanto uma perda deve ser punida e quanto uma vitória deve ser recompensada? Ainda não existem boas respostas. Para comportamentos mais complexos isso pode ser extremamente desafiador. Vários estudos mostraram que recompensas simples podem resultar em agentes aprendendo inesperadamente e até comportamentos potencialmente prejudiciais. Esses sistemas estimulam o medo de futuros agentes com maior autonomia causando estragos quando inseridos no mundo real depois de terem sido treinados para otimizar as funções de recompensa ruim (RAMSUNDAR; ZADEH, 2018, p. 175, tradução nossa).

Segundo Coppin (2013, p. 250): “A informação que é fornecida ao sistema de aprendizado, quando ele realiza sua tarefa corretamente, não diz porque ou como operou corretamente, apenas diz que realizou a tarefa.”. A Figura 36 apresenta um diagrama em blocos da aprendizagem por reforço.

**Figura 36** - Diagrama em blocos da aprendizagem por reforço

Fonte: Haykin (2007, p. 653)

Nesse tipo de aprendizado o agente toma decisões no ambiente de forma a atingir uma meta, apesar da incerteza sobre o ambiente. As ações tomadas pelo agente podem afetar o estado futuro do ambiente, podendo alterar as opções e oportunidades para o agente futuramente. Como os efeitos das ações não são totalmente previstos, o agente deve monitorar frequentemente o ambiente para reagir a ele da melhor forma (SHUKLA, 2017; SUTTON; BARTO, 2018).

Segundo Sutton e Barto (1998, p. 4, tradução nossa) os algoritmos de aprendizagem por reforço requerem uma combinação de pesquisa e memória. “A pesquisa é necessária para encontrar boas ações, e a memória é necessária para lembrar quais ações funcionaram bem em situações no passado.”

### 3.5.3.1 Equação de Bellman

A equação de Bellman expressa a relação entre o valor de um estado e o valor dos seus estados sucessores. Ela calcula a média de todas as possibilidades, ponderando cada uma por sua probabilidade de ocorrência. O valor do estado inicial deve ser igual ao valor (descontado) do próximo estado esperado, mais a recompensa esperada ao longo do caminho (SUTTON; BARTO, 2018). A Equação 3 apresenta a definição da Equação de Bellman:

$$V(s) = \max_a (R(s, a) + \gamma V(s')) \quad (3)$$

Onde:

s = Estado

a = Ação

R = Recompensa

$\gamma$  = Desconto

Para este projeto a Equação de Bellman é usada para contabilizar as recompensas futuras.

### 3.5.3.2 Processo de decisão de Markov

O processo de decisão de Markov (do inglês *Markov Decision Process* - MDP), segundo Russell e Norvig (2013, p. 10) surgiu do trabalho de Richard Bellman, em 1957, que “formalizou uma classe de problemas de decisão sequencial chamados processos de decisão de Markov”.

Segundo Nandy e Biswas (2018, p. 47, tradução nossa) o MDP “é uma estrutura que envolve a criação de fórmulas e modelos matemáticos para a tomada de decisões, onde parte é aleatória e parte permanece nas mãos do tomador de decisão.” Para Ramsundar e Zadeh (2018, p. 173) o “coração ” do MDP é o par *ambiente* e *agente*, onde um ambiente codifica um “mundo” no qual o agente age.

O agente possui um conjunto de ações que pode executar no ambiente, e essas ações realizadas no ambiente podem ter um efeito sobre o mesmo, sendo muitas vezes probabilístico, e por fim se tem as recompensas (RAMSUNDAR; ZADEH, 2018).

O MDP consiste de um conjunto de Estados (com estado inicial  $s_0$ ); um conjunto de Ações ( $s$ ) de ações aplicáveis em cada estado; um modelo de transição  $P(s' | s, a)$  e uma função de recompensa  $R(s)$  (RUSSELL; NORVIG, 2013).

Beysolow II (2019, p. 57) define a MDP como  $(S, A, Pa, Ra)$  onde:

$S$  = Conjunto representando os estados;

$A$  = Conjunto representando as ações permitidas;

$Pa$  = Probabilidade de que a ação  $a$  no estado  $s$  no momento  $t$  resulte no estado  $s'$  no tempo  $t + 1$ ;

$Ra$  = Recompensa imediata recebida após a transição do estado  $s$  declarar  $s'$  devido à ação  $a$ .

### 3.5.3.3 Política

Segundo Shukla (2017, p. 149-150, tradução nossa) “A política sugere qual ação deve ser tomada em relação a um estado.”. Geralmente o agente escolhe a ação com maior recompensa, mas podem existir políticas onde a ação é escolhida de forma arbitrária.

Nancy e Biswas (2018, p. 54, tradução nossa) afirmam que “Uma política é uma função ou um processo de tomada de decisão que permite transições de um estado para outro.”.

Uma política define a maneira que o agente de aprendizagem se comportar em um determinado momento. Grosso modo, uma política é um mapeamento dos estados percebidos do ambiente para as ações a serem tomadas quando está nesses estados.

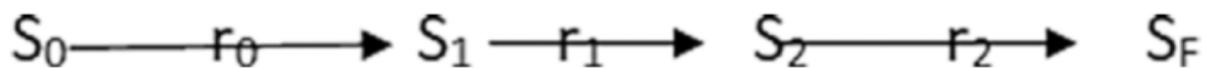
Corresponde ao que em psicologia seria chamado de conjunto de regras ou associações estímulo-resposta (SUTTON; BARTO, 2018, p. 6, tradução nossa).

### 3.5.3.4 Diferença temporal

Segundo Beysolow II (2019, p. 58, tradução nossa) a diferença temporal (do inglês *Temporal Difference* - TD) é um método para “prever uma quantidade que depende dos valores futuros de um sinal específico.”. Está relacionada a previsões em diferentes intervalos de tempo, onde a previsão atual é atualizada para que a próxima previsão esteja correta.

Nandy e Biswas (2018, p. 54 – 55, tradução nossa) afirmam que a TD é um tipo de aprendizado “baseado em sua própria vizinhança ou em seu próprio alcance. Geralmente o aprendizado da diferença temporal é aplicado quando se está em um estado e se quer saber o que está acontecendo em estados sucessivos.”. A Figura 37 apresenta um estado de transições.

Figura 37 - Estados de transição



Fonte: Nandy e Biswas (2018, p. 55)

Na Figura 37 os estados são representados por  $S$  e as recompensas por  $r$ . O objetivo é prever o melhor caminho ao longo de um período de tempo. O que se está tentando realizar é ir do estado  $S_0$  ao estado  $S_F$  e a cada estado são recebidas recompensas, sendo que está tentando se prever o desconto ou soma de recompensas a cada mudança de estado (NANDY; BISWAS, 2018).

### 3.5.3.5 Replay de experiência

O replay de experiência é um método que armazena a experiência do agente em cada etapa da memória e é acessada para realizar a atualização dos pesos. Ela permite que os agentes de aprendizado por reforço lembrem e reutilizem experiências do passado (SUTTON; BARTO, 2018; SHAUL et al., 2016).

### 3.5.3.6 Otimizador Adam

O Adam (estimação adaptativa do momento, do inglês *adaptive moment estimation*) é um algoritmo que atualiza a taxa de aprendizado com o que se aprende, é um otimizador estocástico baseado em gradiente. Requer apenas gradientes de primeira ordem e pouca

memória, e funciona bem para grandes conjuntos de dados (BEYSOLOW II, 2018; SWAMYNATHAN, 2017; SARKAR; BALI; SHARMA, 2018).

### 3.5.3.7 Q-Learning

Q-Learning é um algoritmo que aprende uma política examinando todas as ações possíveis e avaliando cada uma delas. É caracterizado pelo fato de existir uma política que informa ao agente as ações a serem tomadas em diferentes cenários (BEYSOLOW II, 2019).

Neste algoritmo existem duas matrizes, que são geralmente chamadas de matriz Q e matriz R. A matriz Q contém o conhecimento acumulado sobre o ambiente em que será implementada uma política, sendo todas as entradas inicializadas com zero e tem como objetivo maximizar as recompensas, sendo que a cada etapa essa matriz é atualizada. A matriz R representa o ambiente, onde cada linha representa o estado e as colunas representam as recompensas ao mudar de estado. Pode-se pensar na matriz Q como um mapa e a matriz R como o mundo (BEYSOLOW II, 2019). A Figura 38 apresenta uma representação dessas matrizes.

Figura 38 - Matrizes Q e R

		Action					
		0	1	2	3	4	5
Q=	0	0	0	0	0	0	0
	1	0	0	0	0	0	100
	2	0	0	0	0	0	0
	3	0	80	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

		Action					
State							
	0	1	2	3	4	5	
0	-1	-1	-1	-1	0	-1	
1	-1	-1	-1	0	-1	100	
2	-1	-1	-1	0	-1	-1	
3	-1	0	0	-1	0	-1	
4	0	-1	-1	0	-1	100	
5	-1	0	-1	-1	0	100	

Fonte: Beysolow II (2019, p. 56)

### 3.5.3.8 Deep Q-Learning

O Deep Q-Learning (DQL), é um método de aprendizado criado com a proposta de descobrir resultados em termos de recompensa de um determinado estado do ambiente. A recompensa é determinada aproximadamente após a execução de uma determinada ação, dado o estado atual (MUELLER; MASSARON, 2019).

O DQL vem do Q-Learning, sendo que a principal diferença é que o DQL aproxima os valores da matriz Q ao invés de preenchê-la manualmente (BEYSOLOW II, 2019).

A DQL, segundo Mueller e Massaron (2019, p. 300, tradução nossa): “não considera diretamente as ações passadas e o estado associado ou a sequência completa de ações que um

agente deve executar, mas funciona de maneira eficaz para muitos problemas, apontando a melhor ação a ser executada entre as alternativas.”.

Esse método de aprendizado não entende como o ambiente funciona, estando o entendimento do ambiente restrito ao conhecimento do estado de ações tomadas. A DQL associa o estado e as ações com as recompensas esperadas usando uma função matemática.

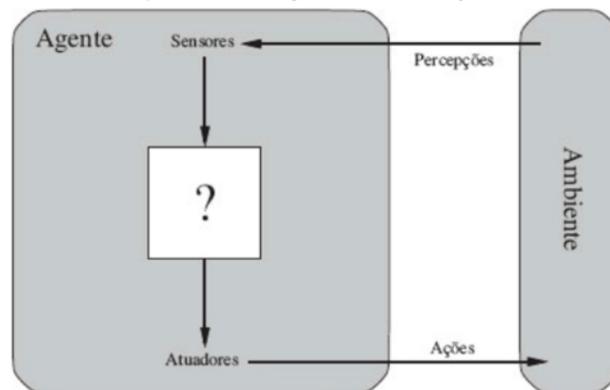
Essa rede neural alimenta o algoritmo DQN, juntamente com uma política (uma política é uma função que escolhe uma sequência de ações) e uma memória de ações e estados anteriores. A memória é necessária para permitir que o exemplo treine uma rede neural. Ela registra experiências anteriores do agente com o ambiente, e o código pode fazer uma amostra para extrair uma série de ações, dependendo de um estado. A rede neural usa a memória para aprender a estimar a provável recompensa de uma ação realizada em um estado (MUELLER; MASSARON, 2019, p. 300, tradução nossa).

### 3.6 Agentes inteligentes

Segundo Russell e Norvig (2013, p. 34) “Um agente é tudo o que pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores.”. Por essa definição um AGV pode ser considerado um agente, já que possui uma grande variedade de sensores para percepção do ambiente e atua sobre ele por meio de sua locomoção e realização de suas ações, como a carga e descarga de materiais. Romero et al (2017) faz uma definição sobre o que são robôs inteligentes, e que pode ser aplicado a AGVs pela forma como os mesmos usam os sensores e como utilizam os atuadores.

Os robôs inteligentes devem, além de poder perceber o ambiente, atuar sobre esse ambiente, ou seja, devem poder se deslocar, manipular e interagir de algum modo com os elementos presentes no ambiente em que estão inseridos. Para que isso seja possível, os robôs devem ser dotados de mecanismos de atuação (motores) capazes de produzir as ações requeridas pelo robô a fim de executar uma determinada tarefa (ROMERO et al, 2017, p. 22).

A Figura 39 apresenta o diagrama de um Agente. Nele o agente é composto de sensores e atuadores. Os sensores percebem os sinais do ambiente e enviam para uma unidade central que vai processar e interpretar as informações recebidas. Conforme o resultado do processamento o agente interage no ambiente por meio dos atuadores.

**Figura 39 - Diagrama de um Agente**

Fonte: Russell e Norvig (2013, p. 35)

No caso de um AGV, os seus sensores identificam objetos, localização, posição, obstáculos e locais de atuação. Os atuadores do AGV podem ser os motores, que permitem ao mesmo fazer com que as rodas girem e assim possam se deslocar pela rota para a realização do seu trabalho. Como existem diferentes modelos de AGVs para diferentes tipos de trabalho o AGV pode estar dotado de outros atuadores. Se o AGV for do tipo empilhadeira haverá um atuador que fará o garfo se mover para cima ou para baixo, um exemplo desse tipo de AGV é apresentado na Figura 40.

**Figura 40 - AGV do tipo empilhadeira transportando caixas**

Fonte: Website da Conveyco<sup>16</sup>.

### 3.7 Autonomia em robótica

Em robótica, a autonomia é a capacidade de perceber o ambiente e tomar decisões sobre ações que ajudam a realizar uma tarefa. Um robô totalmente autônomo não requer intervenção humana para realizar qualquer tarefa. Para conseguir autonomia o seu sistema deve estar bem estruturado para interpretar as informações de entrada (sensores) com a ação a ser realizada, tomando a melhor decisão (KANNIAH; ERCAN; CALDERON, 2014).

<sup>16</sup> Disponível em: <<https://www.conveyco.com/advantages-disadvantages-automated-guided-vehicles-agvs/>>. Acesso em 7 abr. 2019.

Um grande desafio enfrentado pelos cientistas que realizam pesquisas em Robótica (os roboticistas) é como tornar um sistema robótico autônomo. Robôs autônomos devem apresentar reações às mudanças do ambiente, comportamento inteligente, integração entre vários sensores, resolução de múltiplas tarefas, robustez, capacidade de operar sem falha, programabilidade, modularidade, flexibilidade, expansibilidade, adaptabilidade e raciocínio global. Ainda no contexto de autonomia, destaca-se o problema da navegação, que é comumente dividido em subproblemas que podem ser abstraídos em cinco níveis hierarquizados de autonomia: Mapeamento do Ambiente, Localização, Planejamento de Caminho, Geração de Trajetória e Execução de Trajetória (ROMERO et al, 2017, p. 161, tradução nossa).

Conforme escrito por Romero et al (2017) a autonomia em robótica é algo complexo e que envolve vários fatores, mas é a autonomia que faz um AGV ser interessante para uma empresa, um robô que realiza tarefas sem auxílio humano e que pode trabalhar de forma ininterrupta, tendo dessa forma redução de custos.

Ao final da citação de Romero et al (2017) é dito sobre o problema da navegação, item fundamental para um veículo autônomo e que deve ser bem estudado e planejado. Existem diferentes tecnologias que um AGV pode usar para navegação devendo percorrer um trajeto, quase sempre fixo, para a realização de tarefas. Com o uso de sistemas inteligentes um AGV pode tomar decisões sobre melhores rotas, calculando o tempo de percurso e dessa forma realizar tarefas em um tempo menor.

Veículos autônomos podem ser acoplados fisicamente ou através da tarefa de controle para realizar a tarefa específica. A informação é geralmente compartilhada através de uma rede para atingir a missão, e os veículos têm acesso apenas a informações parciais ao tomar decisões. Assim, novos desafios surgem no problema do controle. Por exemplo, a comunicação é realmente fraca em alguns cenários, como para veículos subaquáticos, onde atrasos, confiabilidade e restrições de taxa de dados são muito exigentes (LOSADA; RUBIO; BENCOMO, 2015, p. 12, tradução nossa).

Conforme citado por Losada, Rubio e Bencomo (2015) os robôs podem se comunicar entre si criando uma rede de troca de informações. Em um ambiente com diversos AGVs essa rede permite que avisos de diferentes tipos sejam gerados e compartilhados, como por exemplo, informar que determinada via está obstruída ou foi modificada, fazendo com que sejam decididas outras rotas.

As tecnologias usadas por robôs autônomos têm evoluído por meio de melhorias dos sensores, permitindo perceber detalhes e situações mais complexas. Um exemplo é o sistema de visão, que tem uma pesquisa muito ativa devido ao custo das câmeras ser baixo e as “informações que podem adquirir serem muito ricas”. Outro campo que tem evoluído é o do controle mais inteligente do robô, que deve ser “muito flexível e adaptável” de modo a aprender com um ser humano ou “se adaptar a novas situações” (BEN-ARI; MONDADA, 2018, p. 6).

Mesmo com diversos dispositivos de segurança implementados em robôs autônomos, como medida de segurança adicional robôs podem ter um controle remoto operado por um humano caso o mesmo não se comporte adequadamente ou apresente erros, mas o objetivo do robô permanece o mesmo, agir de forma autônoma (JOSEPH, 2015).

### 3.8 Rotas

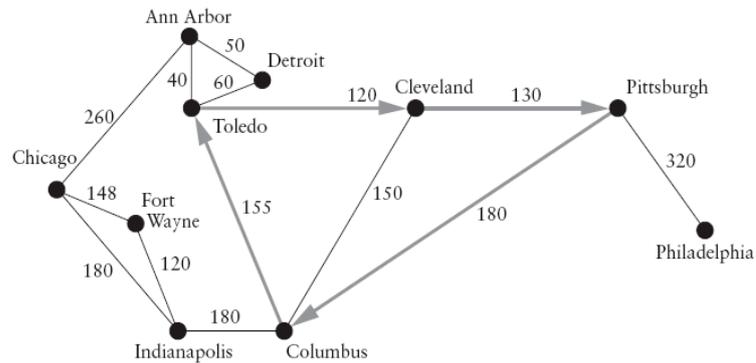
Um AGV deve se deslocar no ambiente para a realização de sua tarefa, e a rota que o mesmo deve fazer já é conhecida, programada em seu sistema, ou ele apenas segue a rota sem fazer nenhuma alteração nela, ou ainda ele pode aprender e buscar as melhores rotas para ir de um ponto A ao ponto B.

Em uma rota existem fatores que vão alterar o comportamento do robô, fazendo com que ele pare ou desvie de obstáculos e realize ações como a carga ou descarga de materiais em pontos determinados.

Romero et al. (2017) divide os métodos de planejamento de rotas em três categorias de algoritmos: mapa de rotas, decomposição de células, e funções potenciais.

Os métodos que utilizam mapa de rotas visam construir um mapa topológico que representa a conectividade de espaço de configuração livre. Geralmente utiliza-se um grafo para armazenar as informações desse mapa topológico, e os nós armazenam informações sobre o estado do robô e as arestas indicam a possibilidade de transição entre os estados. Os métodos de decomposição celular decompõem o espaço de configuração livre em uma coleção de regiões, chamadas células, que não se sobrepõem. Então, um grafo que armazena as informações de adjacência entre as células é utilizado para resolver o problema de planejamento de rotas. Por último, os algoritmos que utilizam funções potenciais representam o robô por uma partícula no espaço de configurações sujeita a forças que a guiam para o destino. Assim, os obstáculos provocam uma força de repulsão, enquanto a configuração de destino induz uma força de atração na partícula (ROMERO et al, 2017).

Os grafos são importantes no planejamento de rotas. Os grafos são um conjunto de vértices (também conhecidos como nós) e arestas (conhecidas como relações) que podem estar interligados representando caminhos ou conexões. A Figura 41 apresenta o exemplo de um grafo ligando cidades. Para um AGV, os vértices representam os pontos para onde o mesmo deve se deslocar e, dependendo do trajeto escolhido, decidir qual trajeto seguir de acordo com algum critério de escolha, se será o caminho mais curto, o mais rápido, entre outros.

**Figura 41 - Exemplo de grafo**

Fonte: Koffman e Wolfgang (2008)

### 3.9 Estudos do uso de inteligência artificial em AGVs

Na literatura acadêmica existem estudos do uso da inteligência artificial para tornar um AGV inteligente, mais autônomo. Bandhu e Panda (2011) em seu relatório de projeto descrevem o desenvolvimento de um veículo usando sistema de inferência Fuzzy<sup>17</sup> e rede neural artificial para o deslocamento autônomo de forma a desviar de obstáculos e evitar colisões.

O veículo usa três matrizes de sensores ultrassônicos para medir a distância de obstáculos. Para localizar o alvo, acompanhar sua posição e local de destino, são usados encoders, GPS e bússola. É usado um conjunto de regras de lógica Fuzzy, de acordo com informações recebidas pelos sensores, e então calculada a velocidade e ângulo. Para a rede neural são usadas as distâncias dos obstáculos e ângulo de posição como parâmetros de entrada, tendo como alvo a velocidade do veículo e o ângulo de direção. Esses dados são então colocados em uma matriz e é usado um algoritmo para treinar e simular a rede.

Martinez-Barbera e Herrero-Perez (2010) em seu artigo apresentam o desenvolvimento de um protótipo de AGV com inteligência artificial para aumentar a autonomia e sua flexibilidade em uma planta industrial que é frequentemente modificada devido à produção sazonal. O protótipo foi testado em uma fábrica real, fazendo o transporte de paletes entre a produção e linhas de armazenamento. O protótipo desenvolvido foi construído tendo como base uma empilhadeira elétrica comercial, onde foi instalado um sistema de navegação a laser para detectar o ambiente e um scanner de segurança laser para segurança.

<sup>17</sup> O sistema de inferência Fuzzy permite resolver problemas cujas variáveis apresentam imprecisões (VILLANUEVA, 2016).

Mezgár, Egresits e Monostori (1995) descrevem em seu artigo sobre o uso de redes neurais artificiais para determinar a velocidade de um AGV, sendo usados padrões de treinamento para o aprendizado. Com isso foi possível perceber uma redução no tempo de reação do sistema, possibilitando o seu uso em tempo real.

A rede neural do robô desenvolvido por Norris (2017) foi criada com base na Rede de Hopfield. As redes de Hopfield foram descritas por John Hopfield em 1982 e utilizam um modelo de memória associativa semelhante às funções da memória humana. Por meio desta rede algo é reconhecido pois já está gravado na memória, sendo realizada uma associação entre o que está armazenado e o que é real (NORRIS, 2017). A rede implementada neste projeto de Norris é usada para ajudar o robô a lembrar de ações passadas a fim de promover a seleção de ações melhores e comportamentos ao longo do ambiente.

Harvey (2016c) utiliza aprendizado por reforço para simular um veículo com o intuito de posteriormente criar um veículo autônomo em pequena escala utilizando o Raspberry Pi. Para a implementação no veículo real Harvey indica as mudanças que devem ser realizadas para ajustar a IA com a ligação dos componentes físicos.

## CAPÍTULO 4 - METODOLOGIA

### 4.1 Tipos da pesquisa

A pesquisa usou a metodologia proposta por Vergara (2013) que divide os tipos de pesquisa em duas categorias, quanto aos fins e quanto aos meios. Este trabalho tem como tipos de pesquisa quanto aos fins explicativa e aplicada, e quanto aos meios bibliográfica e experimental.

#### 4.1.1 Quanto aos fins

Segundo Vergara (2013, p. 45): “A investigação explicativa tem como principal objetivo tornar algo inteligível, justificar-lhe os motivos. Visa, portanto, esclarecer quais fatores contribuem, de alguma forma para a ocorrência de determinado fenômeno.”. A pesquisa, portanto, pode ser classificada como pesquisa explicativa, pois se pretendeu explicar sobre o que é um AGV, sua construção, e seu funcionamento.

Com relação a pesquisa aplicada, para Vergara (2013, p. 45): “A pesquisa aplicada é fundamentalmente motivada pela necessidade de ver problemas concretos; mais imediatos, ou não. Tem, portanto, finalidade prática, ao contrário da pesquisa pura, motivada basicamente pela curiosidade intelectual do pesquisador e situada sobretudo no nível da especulação.”. Com base no que Vergara propõe, a pesquisa é também do tipo aplicada, pois foram realizadas simulações e testes em um AGV virtual.

#### 4.1.2 Quanto aos meios

Quanto aos meios a pesquisa é bibliográfica e pode ser descrita como:

Pesquisa bibliográfica é o estudo sistematizado desenvolvido com base em material publicado em livros, revistas, jornais, redes eletrônicas, isto é, material acessível ao público em geral. Fornece instrumental analítico para qualquer outro tipo de pesquisa, mas também pode esgotar-se em si mesma (VERGARA, 2013, p. 46).

Com base nesta citação a pesquisa classifica-se como bibliográfica pois a maior parte do tema e assuntos relacionados foi publicado em livros, trabalhos acadêmicos, artigos e websites.

A pesquisa é também experimental, sendo definida por Vergara (2013) como:

Pesquisa experimental é a investigação empírica na qual o pesquisador manipula e controla variáveis independentes e observa as variações que tal manipulação e controle produzem em variáveis dependentes. Variável é um valor que pode ser dado por quantidade, qualidade, característica, magnitude, variando em cada caso

individual. [...] A pesquisa experimental permite observar e analisar um fenômeno, sob condições determinadas.

Portanto a pesquisa quanto aos meios é também classificada como pesquisa experimental pois foi realizada a simulação de um AGV virtual a fim de estudar o seu comportamento na realização do desvio de obstáculos utilizando inteligência artificial.

## **4.2 Materiais e métodos**

Havia inicialmente o objetivo de implementar um protótipo de AGV utilizando uma plataforma de prototipagem eletrônica, mas devido ao tempo disponível e recursos financeiros não foi possível realizar essa implementação. Foram realizados os estudos necessários para a construção do protótipo, como os dispositivos, componentes e materiais que poderiam ser usados, e a definição da plataforma de prototipagem mais adequada.

A escolha dos componentes para a possível construção do protótipo e o desenvolvimento da simulação do veículo virtual foi baseada nas escolhas realizadas por autores que desenvolveram robôs móveis, sendo realizadas as adaptações e mudanças necessárias.

Não se tem o objetivo de desenvolver um AGV seguidor de linha, princípio usado por AGVs mais simples que se orientam por fitas metálicas no chão e para isso usam sensores indutivos. Como o objetivo principal do AGV virtual é fazer com que em um primeiro momento ele realize o desvio de obstáculos, então devem ser usados sensores que percebam que existem obstáculos em uma ou mais direções próximos ao veículo e, com base em uma lógica, possa realizar a ação de desvio.

Além de desviar de obstáculos, como uma implementação futura, pretende-se que o AGV possa mapear e aprender o ambiente, de forma que possa saber sua localização, pontos de carga e descarga de material, e saber a melhor rota a ser tomada.

Foram realizadas simulações com o AGV virtual de modo a treinar uma rede neural artificial (RNA) e analisar o comportamento do veículo ao ir de um ponto de origem a um ponto de destino realizando o desvio de obstáculos que podem surgir sem um conhecimento prévio por parte do simulador.

Ao final dessa etapa, com base no comportamento do veículo simulado, são realizados ajustes e testes que forem necessários na rede neural artificial, servindo para uma futura implementação em um protótipo de veículo real.

### **4.3 Instrumentos e procedimentos de coleta de dados**

Os dados foram coletados em livros, artigos, trabalhos acadêmicos e sites. Com base no conhecimento adquirido foi realizado o desenvolvimento de um AGV virtual para a simulação e coleta de dados referentes ao comportamento do mesmo ao realizar o desvio de obstáculos e o processo de aprendizado da rede neural artificial.

### **4.4 Tratamento dos dados**

A análise e tratamento dos dados coletados foi realizada por meio da simulação do AGV virtual, sendo analisado o comportamento do AGV ao percorrer diferentes ambientes e os dados dos gráficos gerados.

### **4.5 Validação da simulação**

Para a simulação do veículo virtual foram testados três ambientes de diferentes tamanhos, diferentes objetivos e diferentes obstáculos.

A simulação foi dividida em três partes:

1. Treino - O veículo explora o ambiente e recebe recompensas de diferentes valores ao se aproximar do objetivo, encostar em obstáculos ou percorrer o ambiente.
2. Teste - Após o veículo estar familiarizado com o ambiente foi analisado se o veículo virtual começa a percorrer corretamente o trajeto até o objetivo.
3. Resultados - Ao começar a atingir o objetivo com frequência foram gerados os gráficos de recompensa para verificar a partir de quantas etapas o veículo começou a receber maiores recompensas, ou seja, atingir o objetivo.

Ao final foi realizada a comparação entre os resultados nos diferentes cenários para se avaliar em qual o veículo teve um melhor desempenho.

## **CAPÍTULO 5 - PROTÓTIPO**

No Capítulo 2 foi apresentado o caso do AGV construído pela Peugeot Citroën que emprega o microcomputador Raspberry Pi. Este caso é indício da viabilidade de construção de um protótipo de AGV empregando uma plataforma eletrônica de baixo custo.

O AGV desenvolvido pela Peugeot Citroën utiliza o princípio de um seguidor de linha, utilizando um sensor indutivo para se guiar por meio de uma faixa magnética que desenha o trajeto que o AGV deve percorrer.

O protótipo não foi implementado devido ao tempo disponível e recursos financeiros, mas o estudo de sua construção serviram de base para o desenvolvimento de um AGV virtual em um ambiente de simulação, apresentado no Capítulo 6.

### **5.1 Escolha dos componentes**

No Capítulo 2 foram apresentadas as plataformas Arduino e Raspberry Pi, além de sensores comerciais que podem ser usados por essas plataformas. Para uma futura construção de um protótipo, o Raspberry Pi é a escolha mais adequada devido ao seu maior poder de processamento e, caso se deseje implementar inteligência artificial, esse critério de escolha é um diferencial.

Existem vários estudos da implementação de inteligência artificial com o Arduino, mas devido a seu armazenamento limitado e baixo poder de processamento o processo pode ser muito lento, podendo fazer o protótipo ficar muito tempo “pensando” até tomar uma decisão, além da limitação de entradas e saídas disponíveis que tornam o aprendizado de uma rede neural limitado a poucos neurônios. Para resolver esse problema seria necessário usar algum poder computacional externo para realizar a tarefa, o que particularmente não parece interessante, sendo preferível usar o Raspberry Pi que é um computador completo.

Com relação aos sensores, foi escolhido o sonar HC-SR04 por ser de baixo custo, ter boa disponibilidade e aceitação, ser de fácil aquisição e ter uma instalação e configuração simples.

O Quadro 5 apresenta a lista de componentes para a montagem de um protótipo com a quantidade de cada componente e os valores vigentes na época do estudo do projeto. A cotação foi realizada em loja virtual e neste valor ainda será acrescentado um eventual valor de frete.

**Quadro 5** - Lista de componentes para a montagem de um protótipo

Descrição	Quantidade	Valor Unitário	Valor Total
Raspberry Pi 3 B+ (Kit Raspberry Pi Start)	1	R\$ 399,90	R\$ 399,90
Sensor ultrassônico HC-SR04	3	R\$ 10,90	R\$ 32,70
Motor Shield L293D Driver Ponte H	1	R\$ 24,90	R\$ 24,90
Kit Chassi 2WD Robô (Com os motores DC)	1	R\$ 69,90	R\$ 69,90
Fonte DC Chaveada 9V 1A Plug P4	1	R\$ 14,90	R\$ 14,90
Protoboard 400 Pontos	1	R\$ 13,90	R\$ 13,90
Kit Jumpers Macho-Macho x65 Unidades	1	R\$ 12,90	R\$ 12,90
Barra de Pinos 1x40 Fêmea 180 Graus	1	R\$ 2,40	R\$ 2,40
Suporte Bateria 9V Plug P4	1	R\$ 3,40	R\$ 3,40
Abraçadeira de Nylon Branca x100 Unidades - 140x2,5mm	1	R\$ 7,90	R\$ 7,90
Fio Wire Wrap 120m 30AWG 8 Cores	1	R\$ 69,90	R\$ 69,90
Resistor 1K $\Omega$ 1/4W x20 Unidades	1	R\$ 2,40	R\$ 2,40
Resistor 2K $\Omega$ 1/4W x20 Unidades	1	R\$ 2,40	R\$ 2,40
<b>Total</b>			R\$ 657,50

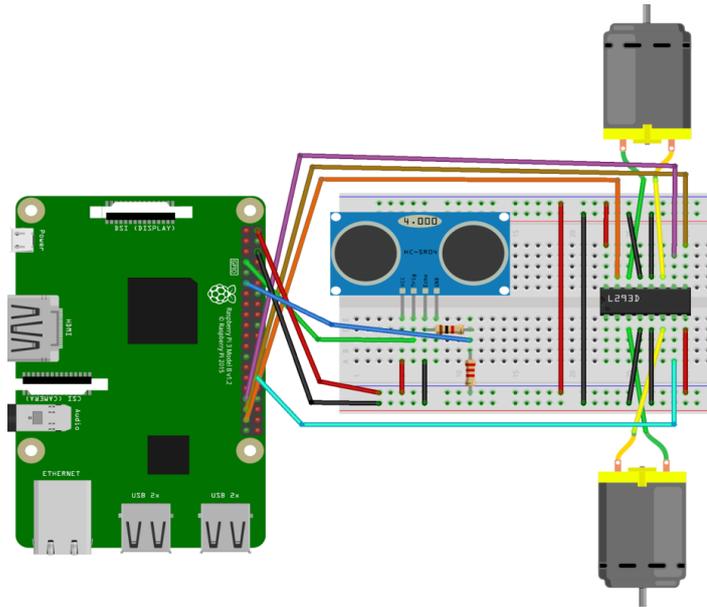
Fonte: O autor (2019)

## 5.2 Diagrama esquemático

A Figura 42 apresenta o diagrama de montagem dos componentes e a Figura 43 o diagrama esquemático. Na figura está sendo mostrado apenas um sensor ultrassônico HC-SR04, mas foi pensado o uso de três sensores para cobrir uma maior área de visão, e é o que foi desenvolvido no AGV virtual. Os mesmos não foram colocados no circuito pois a quantidade de representação de fios dificulta a visualização e organização do modelo.

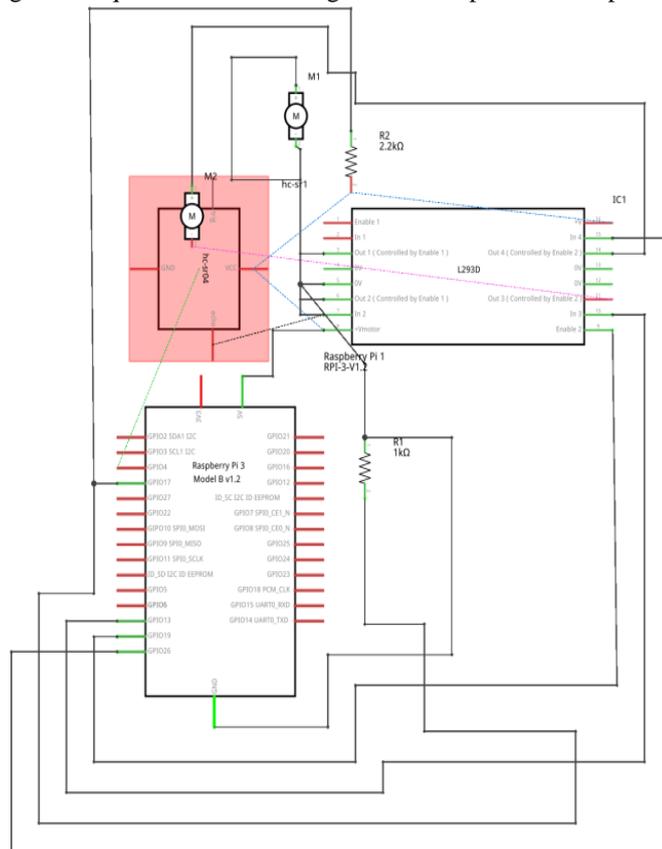
Os Anexos A, B e C apresentam a pinagem do Raspberry Pi 3 B+, sensor ultrassônico HC-SR04 e Ponte H L293D, respectivamente.

**Figura 42** - Diagrama da montagem do protótipo de um AGV



Fonte: O autor (2019)

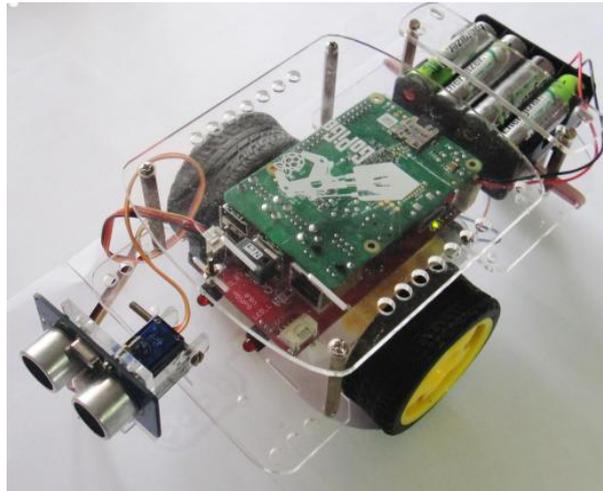
**Figura 43** - Diagrama esquemático da montagem dos componentes do protótipo de um AGV



Fonte: O autor (2019)

A Figura 44 apresenta uma representação do aspecto provável do protótipo do AGV caso o mesmo tivesse sido implementado, com a diferença que, ao invés de apenas um sensor ultrassônico serão três, cada um em uma direção (frente, esquerda e direita).

**Figura 44** - Aspecto provável do protótipo do AGV

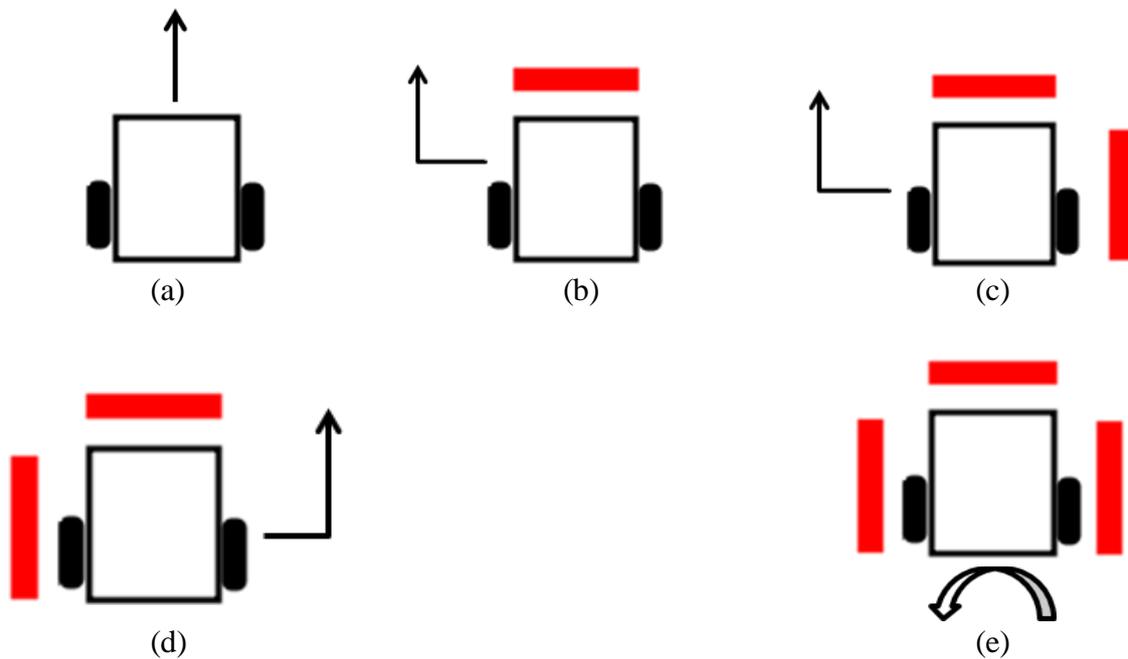


Fonte: <https://www.dexterindustries.com/GoPiGo/projects/python-examples-for-the-raspberry-pi/gopigo-robot-raspberry-pi-ultrasonic-sensor-example/>

### 5.3 Funcionamento

Os sensores localizados em cada lado do veículo detectam os obstáculos e, com base no processamento, é realizado o desvio. A Figura 45 apresenta as possibilidades de ação do veículo.

Figura 45 - Ações do protótipo

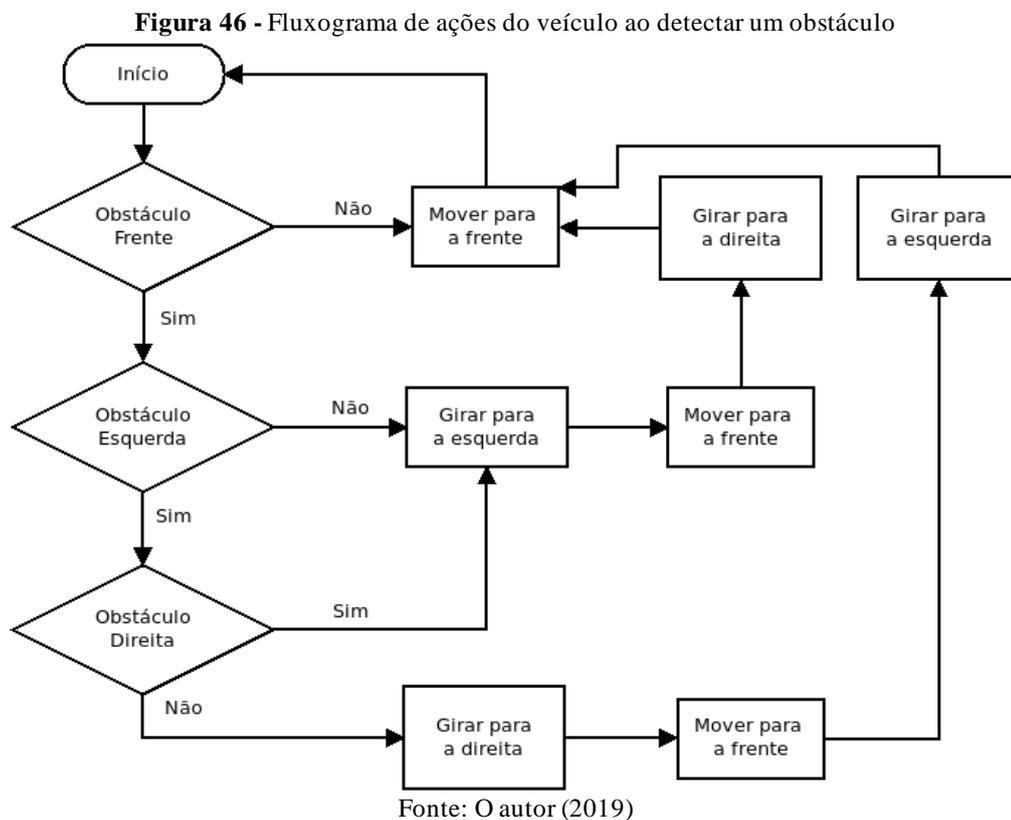


Fonte: autor (2019)

Em (a), quando não existe obstáculo à frente, o robô se desloca no sentido para a frente. Na imagem (b), ao detectar um obstáculo (representado pelo retângulo vermelho), o robô desvia para a esquerda. Esse padrão de desvio à esquerda foi definido pelo autor do

projeto. Na imagem (c) existe um obstáculo à frente e à direita, logo o robô realiza o desvio para a esquerda. A imagem (d) mostra um obstáculo à frente e na esquerda, assim o robô desvia para a direita. No caso (e) existe obstáculo à frente e nas duas laterais, esquerda e direita, mas não existe obstáculo atrás, logo o robô irá girar e, ao perceber que possui a liberdade de deslocamento, segue em frente. No caso de haver obstáculo nas quatro direções o robô irá girar e, após determinado tempo ao perceber que não tem liberdade de movimento, deve permanecer parado até que um dos sensores perceba que um obstáculo foi removido.

Foram pensadas as diversas possibilidades de ação para que o veículo realizasse o desvio de obstáculos, sendo então gerado o fluxograma apresentado na Figura 46 que mostra as decisões de ação a serem tomadas pelo veículo ao detectar obstáculos por meio dos sensores.



O Quadro 6 apresenta a tabela dos sinais de entrada (sensores) e das saídas (motores). Para os sensores, sinal igual a 0 (zero) significa que não houve a detecção de obstáculo, sinal 1 (um) a presença de obstáculo. Nos motores, sinal 0 (zero) significa que não está acionado e sinal 1 (um) que está acionado.

**Quadro 6** - Tabela verdade do funcionamento dos sensores e atuadores

Entradas			Saídas	
sensor_frente	sensor_direito	sensor_esquerdo	motor_direito	motor_esquerdo
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	0	0

Fonte: O autor (2019)

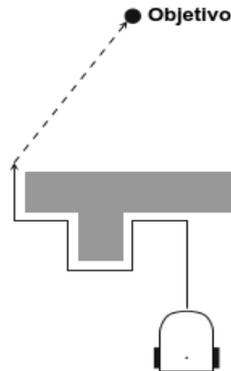
Com base na tabela verdade foi gerado o pseudocódigo do algoritmo. Quando não houver obstáculos a frente ( $\text{sensor\_frente} == 0$ ) o robô se move para a frente, independente do estado dos outros sensores. Se houver um obstáculo na frente e não houver sinal de obstáculo nos demais sensores, o padrão é o robô virar para a esquerda e realizar o desvio.

Havendo obstáculo a frente e detecção de obstáculos nas laterais ( $\text{sensor\_direito}$  ou  $\text{sensor\_esquerdo}$ ) dependendo do sensor lateral que detectou o obstáculo o veículo deve realizar o desvio para o lado oposto. Por exemplo, caso haja obstáculo à frente e na direita, o robô deve se mover para a esquerda.

No caso de haver obstáculo a frente e nas duas laterais, o robô gira e depois realiza o desvio para a esquerda. Se houver obstáculo nas quatro direções, como não existe a possibilidade de desvio, pois isso indica que o robô está obstruído, então o robô deve permanecer parado até que um dos obstáculos seja removido. O Apêndice B apresenta o pseudocódigo do algoritmo criado a partir da tabela-verdade do Quadro 6.

A Figura 47 demonstra como seria realizado o desvio de um obstáculo. Ao detectar um elemento a frente o robô gira para a esquerda e avança. Ao perceber um obstáculo à frente e na direita, o mesmo gira para a esquerda, avança e continua o trajeto realizando os desvios de acordo com a percepção dos sensores.

**Figura 47** - Exemplo do desvio de obstáculos realizado por um robô

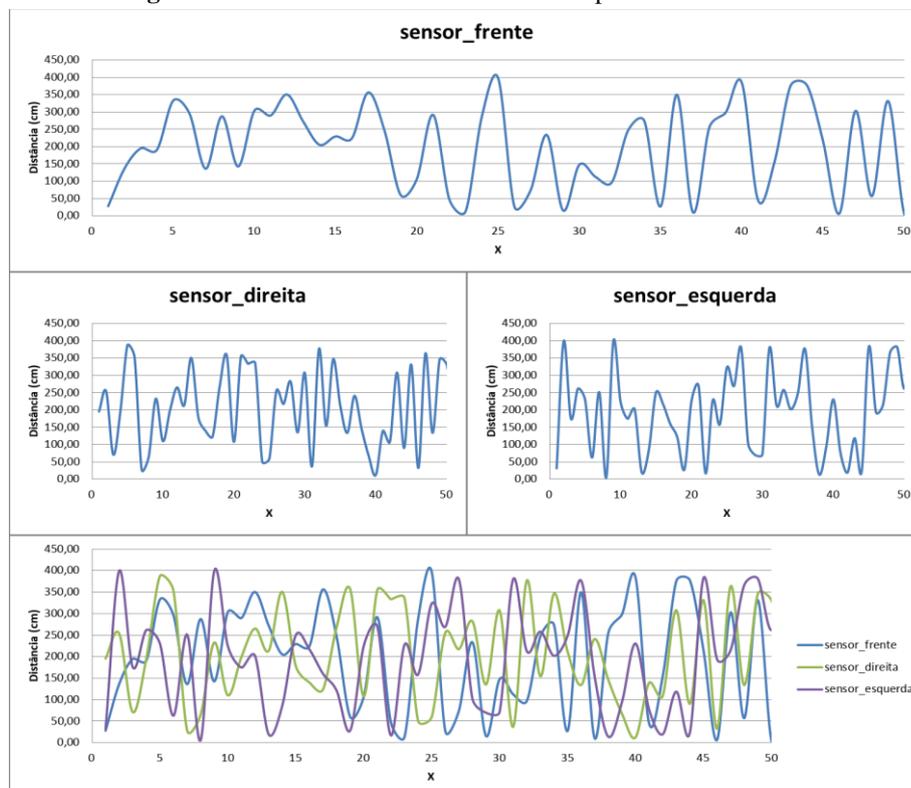


Fonte: Adaptado de Ben-Ari e Mondada (2018, p. 113)

### 5.3.1 Sensores

O sensor ultrassônico HC-SR04 trabalha com uma faixa de medição de 2 cm a 400 cm. Com base nessa informação, e utilizando uma planilha eletrônica, foram gerados 200 valores de distância aleatórios para os três sensores e montados os gráficos apresentado na Figura 48.

**Figura 48** - Gráfico com valores aleatórios para os três sensores



Fonte: O autor (2019)

Determinada faixa de valores não influenciará na tomada de ação do robô, já que uma distância muito grande de um obstáculo fará com que o veículo continue em frente. Somente determinada faixa de valores, que estiverem mais próximos do obstáculo, que serão

processados para que se realize uma ação. Por decisão do autor esses valores serão escolhidos na faixa entre 10 cm e 60 cm, por ser uma distância onde é possível realizar o desvio de forma segura para o veículo.

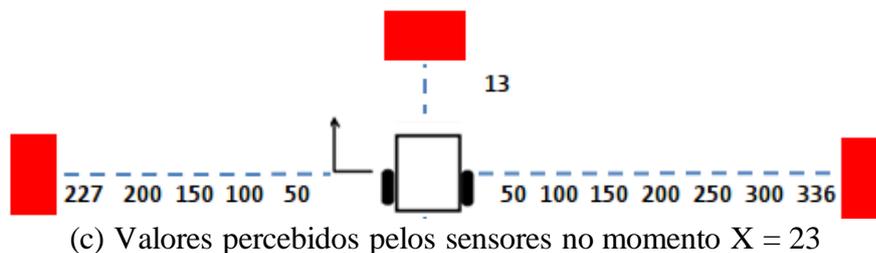
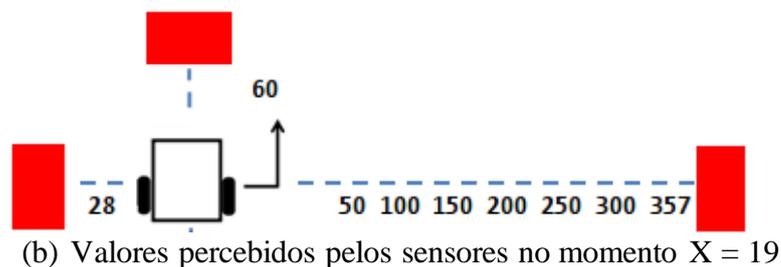
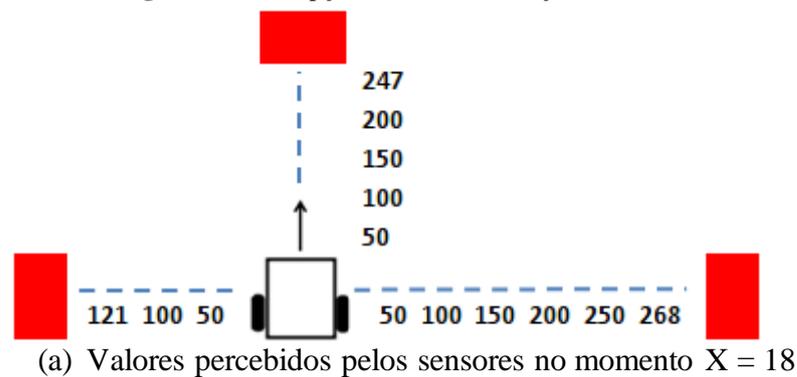
O Quadro 7 apresenta valores aleatórios para os três sensores em três momentos diferentes e a Figura 49 representa, em forma de imagem, os valores percebidos pelos sensores e a ação tomada, apresentados no Quadro 7. Os retângulos vermelhos são os obstáculos.

**Quadro 7** - Ação do robô de acordo com os valores de medição dos sensores

X	sensor_frente	sensor_direita	sensor_esquerda	Ação
18	247,16	268,15	121,47	Frente
19	60,05	357,42	28,02	Direita
23	13,19	336,1	227,68	Esquerda

Fonte: O autor (2019)

**Figura 49** - Percepção dos sensores e ação tomada



Fonte: O autor (2019)

## CAPÍTULO 6 - SIMULAÇÃO DO VEÍCULO VIRTUAL

Para a simulação do veículo virtual e testes da rede neural artificial foi usada a linguagem de programação Python versão 3.6, junto com a biblioteca PyTorch versão 0.3.1. Além da biblioteca PyTorch foram usadas as bibliotecas Kivy e Matplotlib. Foi utilizado o Anaconda Python versão 2019.07 e o algoritmo foi desenvolvido com a IDE Spyder versão 3.3.6.

Python é uma linguagem *open source*, colaborativa, de alto nível e uso geral que possui diversas bibliotecas para computação científica. Essa série de fatores fez com que muitos cientistas e engenheiros adotassem o Python para explorar dados, identificar padrões, criar e implantar modelos para ambientes de produção (SWAMYNATHAN, 2017). Python é a linguagem de programação oficial utilizada pelo Raspberry Pi e aparece em primeiro lugar no ranking da pesquisa de 2019 do IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos) das linguagens mais utilizadas em sistemas embarcados<sup>18</sup> (THE..., 2019).

O Anaconda Python é uma plataforma *open source* de gerenciamento de pacotes Python e R para desenvolvimento, teste e treinamento em Data Science e Machine Learning (ANACONDA, 2019).

A IDE Spyder é um ambiente de desenvolvimento científico *open source* para a linguagem Python que se integra a diversos pacotes usados em ciência dos dados e é extensível por meio de *plugins* (SPYDER, 2018).

Com relação às bibliotecas utilizadas, PyTorch é uma biblioteca *open source* para aprendizado de máquina (PYTORCH, 2019). Kivy é uma biblioteca *open source* para desenvolvimento de interfaces de usuário de aplicações (KIVY, 2019) e a biblioteca Matplotlib produz imagens 2D para gerar diversos tipos de gráficos (MATPLOTLIB, 2019).

A inteligência artificial foi criada com base no que foi desenvolvido e proposto por Harvey (2016) e Qu (2018).

### 6.1 Modelagem da rede neural

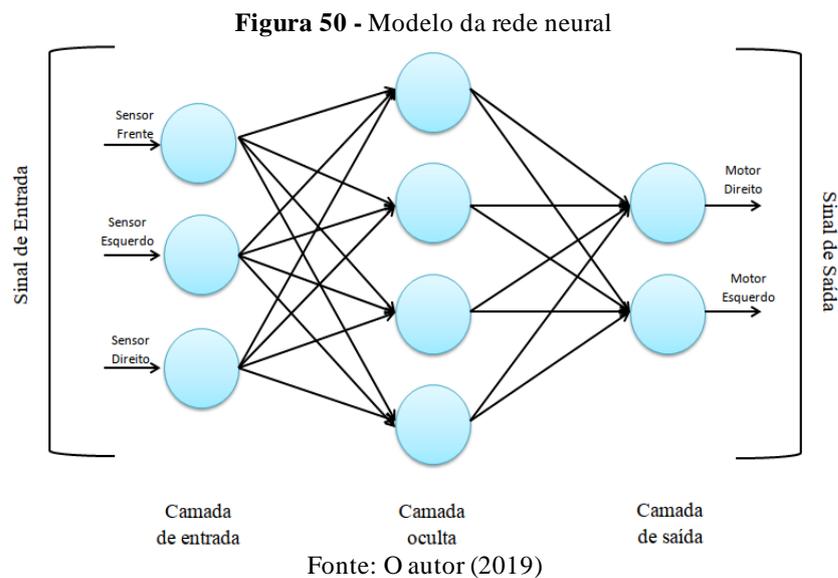
Com a definição dos componentes de entrada e saída, as ações a serem realizadas pelo veículo e a tabela verdade apresentada no Quadro 6, foi realizada a modelagem da rede

---

<sup>18</sup> Sistema microprocessado dedicado ao dispositivo ou sistema que ele controla que realiza um conjunto de tarefas predefinidas, geralmente com requisitos específicos (SISTEMA EMBARCADO, 2019).

neural, apresentada na Figura 50, e a sua implementação via código para realizar a simulação do veículo de modo a analisar o seu comportamento.

A camada de entrada possui três neurônios, que representam os sensores do veículo, logo após se tem a camada oculta, e na camada de saída existem dois neurônios. Os dois neurônios na saída servem para o acionamento dos dois motores de corrente contínua (DC), fazendo com que os mesmos se movam para frente, esquerda (o motor esquerdo permanece parado enquanto o motor direito é acionado), e direita (funcionamento contrário ao do movimento para esquerda).



## 6.2 Rede neural

Para a implementação da rede neural foram utilizadas técnicas de Aprendizado por Reforço e Deep Learning, mais especificamente Deep Q-Learning.

O veículo consiste de cinco estados:

- O estado de cada sensor (são três sensores)
- As orientações (orientação+ e orientação-) que fazem o veículo girar

As ações que o veículo pode executar são:

- Girar 10° no sentido horário
- Girar 10° no sentido anti-horário
- Não girar

As recompensas são:

- Se aproximar do objetivo: 0.1
- Se afastar muito do objetivo: -0.2
- Bater em um obstáculo: -1

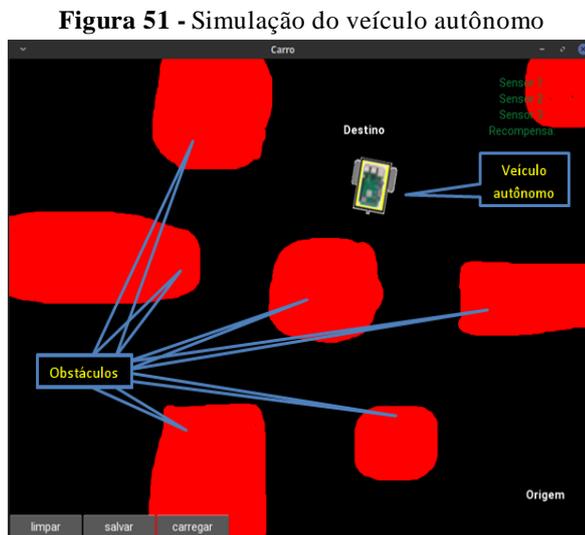
A arquitetura da rede neural é apresentada a seguir:

- Camada de entrada: 5 neurônios (um para cada entrada de estado)
- Camada oculta: 30 neurônios (se chegou a esse valor por experimentação)
- Camada de saída: 3 neurônios (um para cada ação: Não girar, girar no sentido horário, girar no sentido anti-horário)
- Função de ativação: Softmax (usada para a seleção das ações. O algoritmo determina a próxima ação de forma probabilística)
- Função de ativação da camada oculta: ReLU
- Otimizador: ADAM

### 6.3 Simulação

A Figura 51 apresenta uma imagem da simulação do veículo autônomo. O veículo se move pelo ambiente e os seus sensores detectam os obstáculos, representados em vermelho.

No início da simulação o veículo começa a percorrer o ambiente e em diversos momentos bate nos obstáculos. No decorrer do tempo, à medida que começa a atingir com maior frequência o objetivo e desviar dos obstáculos o veículo passa a colidir menos com os obstáculos e fazer o percurso de uma forma mais direta, sem percorrer tanto o ambiente.



Fonte: O autor (2019)

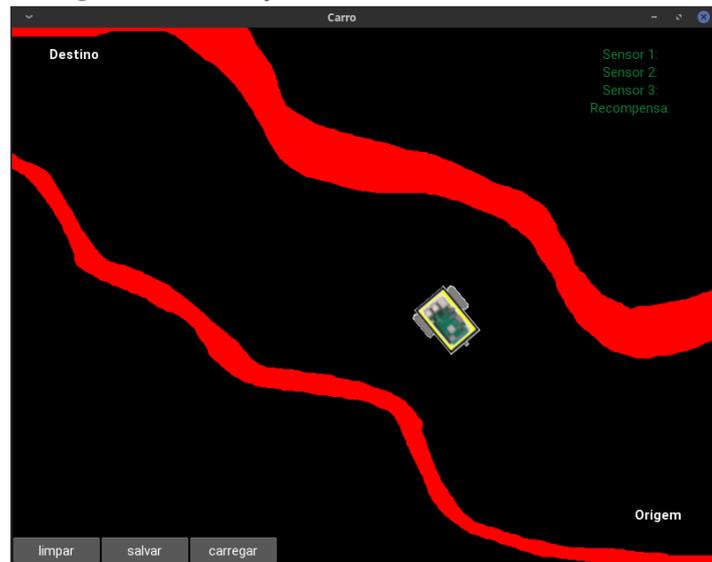
Foram realizadas simulações em três diferentes cenários. Cada cenário possui tamanhos diferentes, diferentes obstáculos e diferentes pontos de destino, sendo comum a todos eles somente o ponto de origem.

### 6.3.1 Cenário 1

O Cenário 1 pode ser considerado o mais simples já que simula uma estrada entre a origem e o destino, conforme apresentado na Figura 52. As especificações do Cenário são:

- Cenário: Altura = 600px, Largura = 600px
- Origem:  $x = 680\text{px}$ ,  $y = 5\text{px}$
- Destino:  $x = 20\text{px}$ ,  $y = 520\text{px}$

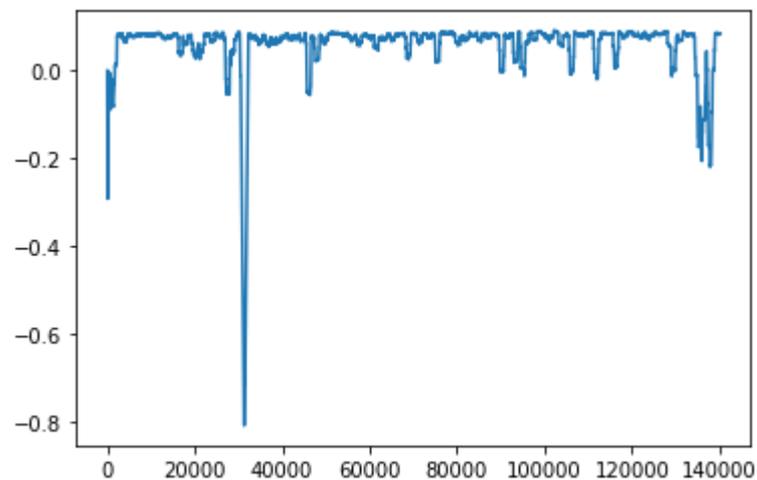
**Figura 52** - Simulação do veículo autônomo no Cenário 1



Fonte: O autor (2019)

Os testes mostram, conforme a Figura 53, que no início o veículo já começou a receber recompensas positivas, por volta de 1.000 ciclos de treinamento, se mantendo praticamente constante.

**Figura 53** - Gráfico de recompensas do veículo autônomo no Cenário 1



Fonte: O autor (2019)

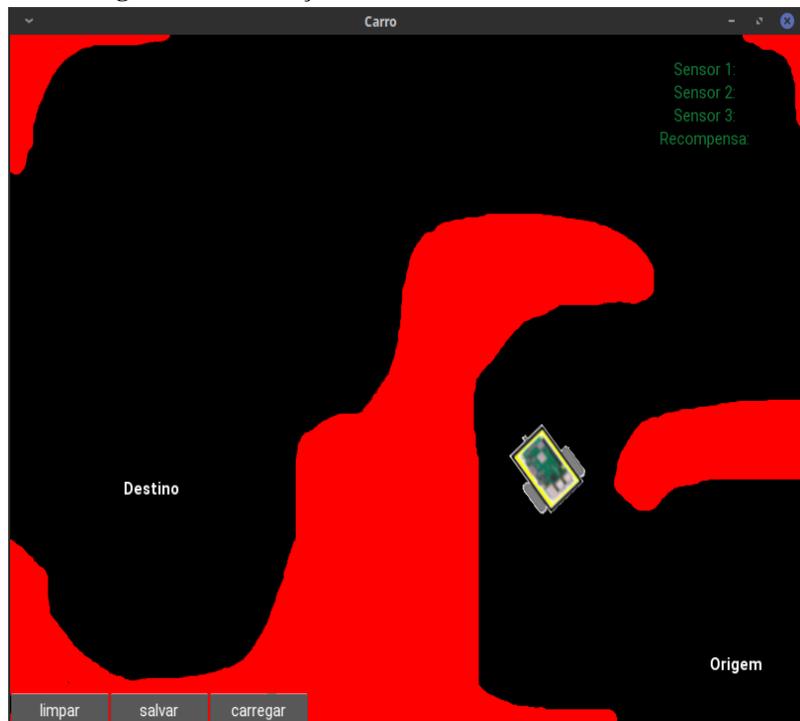
### 6.3.2 Cenário 2

O Cenário 2, apresentado na Figura 54, foi o que precisou de um maior número de ciclos para o aprendizado, levando aproximadamente 200.000 ciclos para começar a receber recompensas positivas por certo período, conforme o gráfico da Figura 55.

As especificações do Cenário são:

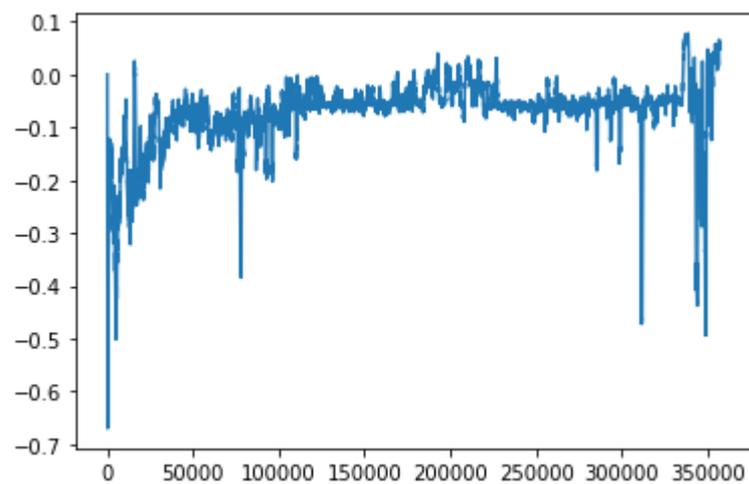
- Cenário: Altura = 800px, Largura = 800px
- Origem:  $x = 680\text{px}$ ,  $y = 5\text{px}$
- Destino:  $x = 92\text{px}$ ,  $y = 157\text{px}$

**Figura 54** - Simulação do veículo autônomo no Cenário 2



Fonte: O autor (2019)

**Figura 55** - Gráfico de recompensas do veículo autônomo no Cenário 2

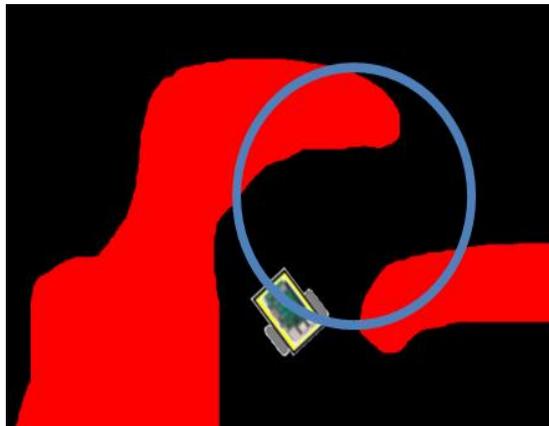


Fonte: O autor (2019)

A maior dificuldade encontrada pelo veículo foi ao passar entre os dois obstáculos circulares em azul na Figura 56. Ao chegar nesse ponto o veículo começava a girar e retornar para o ponto de origem. Foram necessários vários ciclos até conseguir com que o veículo passasse pelos obstáculos e fosse para o ponto de destino.

A dificuldade encontrada pode estar relacionada com a curva ser muito fechada e a proximidade entre os dois obstáculos, fazendo com que os sensores percebam a todo momento obstáculos. Uma solução encontrada foi reduzir a velocidade do veículo, fazendo os sensores terem uma melhor percepção.

**Figura 56** - Dificuldade encontrada pelo veículo virtual no Cenário 2



Fonte: O autor (2019)

### 6.3.2 Cenário 3

O Cenário 3 se assemelha a um labirinto e os pontos de origem e destino estão relativamente próximos, conforme apresenta a Figura 57. As especificações desse cenário são:

- Cenário: Altura = 1000px, Largura = 1000px
- Origem:  $x = 680\text{px}$ ,  $y = 5\text{px}$
- Destino:  $x = 433\text{px}$ ,  $y = 461\text{px}$

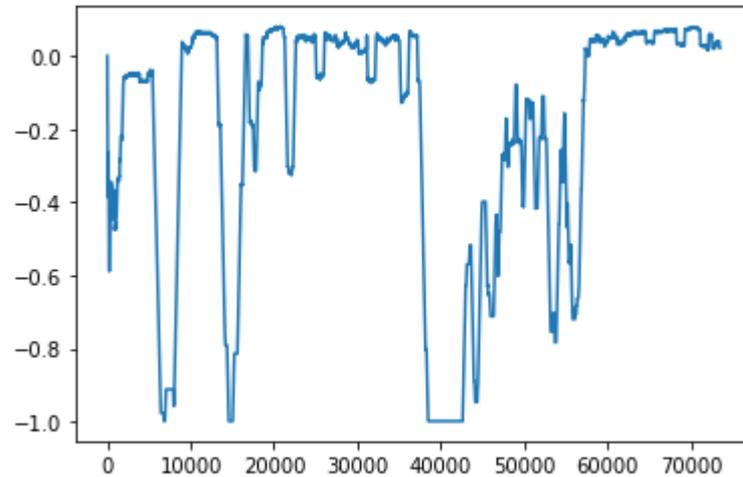
**Figura 57** - Simulação do veículo autônomo no Cenário 3



Fonte: O autor (2019)

Fazendo um comparativo com os outros cenários, o gráfico do Cenário 3 parece se mostrar muito irregular, conforme apresentado na Figura 58. A explicação está na quantidade de obstáculos encontrados pelo veículo fazendo com que receba muitas recompensas negativas ao se aproximar deles.

**Figura 58** - Gráfico de recompensas do veículo autônomo no Cenário 3



Fonte: O autor (2019)

Com cerca de 10.000 ciclos de treinamento o veículo começou a receber recompensas positivas, se mantendo quase constante até chegar a 40.000 ciclos, quando houve uma grande queda nas recompensas recebidas e permaneceu assim até aproximadamente 60.000 ciclos, quando voltou a receber recompensas positivas.

Essa queda nas recompensas por volta das 40.000 ciclos ocorreu devido ao veículo tentar explorar diferentes caminhos para ir até o destino ou chegar na origem, recebendo muitas recompensas negativas nesse processo devido aos obstáculos encontrados e os mesmo serem muito próximos.

## CAPÍTULO 7 - RESULTADOS DA PESQUISA

Durante o estudo das plataformas de prototipagem eletrônica mais utilizadas atualmente, o Arduino e o Raspberry Pi, foi possível perceber que o Raspberry Pi é a escolha mais adequada para a criação de um protótipo de AGV em pequena escala, caso se deseje implementar o projeto futuramente, devido ao seu maior poder de processamento e armazenamento. Esses fatores são decisivos para a sua escolha se houver o desejo de implementar inteligência artificial. O protótipo físico do AGV não foi implementado devido ao tempo disponível e recursos financeiros.

O CLP, comumente usado em AGVs comerciais, pode ser substituído pelo Raspberry Pi em aplicações reais. Com isso existe a redução no custo de produção de um AGV e a possibilidade de implementar novos recursos, como o uso da inteligência artificial.

A simulação do AGV virtual se mostrou uma etapa importante do processo pois permitiu que fossem realizados ajustes e correções na RNA para melhorar o desempenho do veículo. Em uma posterior implementação da IA em um veículo real seriam necessários poucos ajustes para o seu funcionamento.

A simulação também se mostrou como uma ferramenta de redução de custos já que, caso não haja recursos disponíveis no momento para a construção de um protótipo, é possível realizar testes e analisar o comportamento do veículo. Uma outra forma de redução nos custos é que, caso a IA fosse implementada diretamente no protótipo sem uma simulação prévia, provavelmente o protótipo sofreria danos que acarretariam novos custos para a reposição de peças e componentes danificados.

Durante a etapa de simulação do veículo virtual o mesmo se mostrou inicialmente desorientado, percorrendo aleatoriamente o ambiente e se chocando com obstáculos. Esse processo faz parte do aprendizado por reforço, onde o agente explora o ambiente e após diversas tentativas aprende. Após aproximadamente 1.000 ciclos de treinamento em um cenário simples o mesmo conseguiu aprender e realizar o deslocamento de um ponto de origem até um ponto de destino e retornar, e conseguiu realizar isso desviando de obstáculos.

Os testes e simulações foram realizados em cerca de um mês. Durante os testes foram realizados diversos ajustes e alterações em parâmetros e no algoritmo para se chegar a um melhor desempenho da simulação.

Com os testes em diferentes cenários foi possível perceber que o maior problema foi com relação a como os obstáculos estavam dispostos no ambiente, influenciando no tempo

para realizar o aprendizado. Fazendo um comparativo entre os Cenários 2 e 3 é possível perceber que o Cenário 3 possui uma quantidade de obstáculos maior do que o Cenário 2, mas o aprendizado foi mais rápido levando menos ciclos para receber recompensas positivas com maior frequência. O Cenário 1, por ser o mais simples entre os demais, foi o que apresentou um aprendizado mais rápido, ganhando recompensas positivas em pouco tempo.

Os resultados mostram que, mantendo o mesmo ambiente, mas mudando a disposição de obstáculos, o veículo virtual se adapta a esses novos obstáculos. Como já aprendeu anteriormente o caminho que deve percorrer, apenas realiza o aprendizado da disposição dos novos obstáculos, realizando o desvio dos mesmos.

## CONCLUSÃO

Os veículos autônomos apresentam um potencial para ser uma realidade em um futuro breve e muitos estudos e pesquisas tem sido realizadas nessa área para corrigir falhas e melhorar processos. O que existe atualmente ainda não pode ser considerado totalmente confiável e livre de falhas. Muitos avanços tecnológicos têm surgido nos ramos da eletrônica, automação, robótica e inteligência artificial especificamente para esse nicho.

Veículos guiados automaticamente, os AGVs, surgiram muito antes dos carros de passeio incorporarem tecnologias autônomas e, nos carros, um veículo totalmente autônomo ainda não existe, mesmo com as tecnologias que tem surgido muitos estudos estão em desenvolvimento.

A diferença entre um carro e um AGV é que os AGVs possuem aplicações em sua maioria restritas a processos industriais. Estudar as tecnologias de construção de AGVs permitem que sejam criados veículos para outras aplicações, inclusive veículos de passeio.

No presente trabalho foram realizados estudos sobre métodos de construção de AGVs e tecnologias existentes, plataformas de prototipagem que pudessem substituir componentes utilizados em AGVs comerciais, um comparativo dos custos e técnicas de inteligência artificial que pudessem ser implementadas em um veículo real.

Os estudos mostram que o Raspberry Pi, por ter um maior poder de processamento e armazenamento comparado com o Arduino é a escolha mais adequada para se trabalhar em projetos no qual se deseja utilizar inteligência artificial. O Raspberry Pi pode ser uma solução mais econômica e com maior possibilidade de recursos comparado com o que é comumente utilizado em AGVs comerciais, que é o CLP. A junção de uma plataforma de baixo custo aliada com a implementação de inteligência artificial possibilitam a construção de um AGV mais econômico e inteligente.

Empresas e indústrias estão a todo o momento pesquisando formas de redução de custos e aumento de produtividade. A construção de um veículo empregando componentes de baixo custo, mas de qualidade, e inteligente podem ser uma vantagem competitiva e econômica.

A construção de um ambiente virtual para a simulação da inteligência artificial permitiu analisar o comportamento do veículo, realizar ajustes na rede neural e testes, servindo como um guia para a implementação da RNA em um protótipo de veículo físico.

Infelizmente o curso, e a universidade, não dispõem de recursos como laboratórios de automação e robótica, e nem placas de prototipagem eletrônica. Caso houvesse no curso

disciplinas voltadas para as áreas de automação e robótica e laboratórios para essas disciplinas com os recursos necessários, a abrangência e o leque de possibilidades para os alunos do curso poderia ser maior. Esse fator também teria permitido que fossem usados recursos da universidade para a construção de um protótipo.

Como recomendação para futuros trabalhos se tem:

- Melhorias e ajustes na RNA.
- Estudo de outros sensores.
- Estudo de outras plataformas de prototipagem eletrônica.
- Construção de um protótipo em pequena escala.
- Construção de um AGV para aplicação real.

Como um questionamento para uma proposta de futuro trabalho sugere-se o desenvolvimento de um veículo que responda a questão: Um veículo, seja ele virtual ou físico, já treinado, se colocado em um ambiente diferente é capaz de navegar nesse novo ambiente ou ele apenas decora o ambiente de treinamento?

Os estudos na área de inteligência artificial, tecnologias sensoriais e na eletrônica aplicada em veículos autônomos tem aumentado e a tendência é continuar assim até que se chegue a um nível de autonomia no qual não seja mais necessária nenhuma forma de intervenção humana, sendo o ser humano passivo da tecnologia.

## REFERÊNCIAS

ALCIATORE, David G.; HISTAND, Michael B. **Introdução à mecatrônica e aos sistemas de medições**. Porto Alegre: AMGH, 2014.

ANACONDA (python distribution). In: Wikipédia: a enciclopédia livre. Disponível em: [https://en.wikipedia.org/wiki/Anaconda\\_\(Python\\_distribution\)](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)). Acesso em: 5 set. 2019.

ANACONDA distribution. **Anaconda documentation**. 2019. Disponível em: <https://www.anaconda.com/distribution/>. Acesso em: 5 set. 2019

AUTOMATED Guided Vehicles (AGV) Market Analysis Report By Type (Tow Vehicle, Unit Load Carrier), By Navigation Technology, By Application (Logistics & Warehousing), By End-use Industry, And Segment Forecasts, 2018 - 2025. 2018. Disponível em: <https://www.grandviewresearch.com/industry-analysis/automated-guided-vehicle-agv-market>. Acesso em: 9 mar. 2019.

AWAD, Mariette; KHANNA, Rahul. **Efficient learning machines: theories, concepts and applications for engineers and system designers**. New York: Apress Open, 2015.

BANDHU, Arpan; PANDA, Atula Kumar. **Navigation and control of automated guided vehicle using fuzzy inference system and neural network technique**. Orientador: Dr. Dayal R Parhi. 2011. Projeto (Bachelor of Technology in Mechanical Engineering) - National Institute of Technology Rourkela, Rourkela, 2011. Disponível em: [http://ethesis.nitrkl.ac.in/2478/1/thesis\\_abandhu\\_akpanda.pdf](http://ethesis.nitrkl.ac.in/2478/1/thesis_abandhu_akpanda.pdf). Acesso em: 16 mar. 2019.

BEN-ARI, Mordechai; MONDADA, Francesco. **Elements of robotics**. Cham: Springer Open, 2018.

BENEVIDES, Chris Benevides. **7 Cost-saving benefits of automated guided vehicles (AGVs)**. 14 nov. 2016. Disponível em: <https://www.conveyco.com/7-cost-saving-benefits-automated-guided-vehicles-agvs/>. Acesso em: 19 maio 2019.

BEYSOLOW II, Taweh. **Applied reinforcement learning with python: with openai gym, tensorflow, and keras**. New York: Apress, 2019.

BRAGA, Antônio de Pádua; DE CARVALHO, André Ponce de Leon F.; LUDERMIR, Teresa Bernarda. **Redes neurais artificiais: teoria e aplicações**. 2. ed. Rio de Janeiro: LTC, 2014.

CHOLLET, François. **Deep learning with python**. Shelter Island: Manning, 2018.

CONTROLADOR lógico programável. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2019. Disponível em: [https://pt.wikipedia.org/w/index.php?title=Controlador\\_l%C3%B3gico\\_program%C3%A1vel&oldid=55572552](https://pt.wikipedia.org/w/index.php?title=Controlador_l%C3%B3gico_program%C3%A1vel&oldid=55572552). Acesso em: 6 jul. 2019.

COOK, David. **Robot building for beginners**. 3. ed. New York: Apress, 2015.

COPPIN, Ben. **Inteligência artificial**. Rio de Janeiro: LTC, 2013.

COSENZA, Ramon M.; GUERRA, Leonor B. **Neurociência e educação**: como o cérebro aprende. Porto Alegre: Artmed, 2011.

CROSS Docking em logística: o que é e como implementar de forma eficiente. 20 jun. 2017. Disponível em: <http://www.patrus.com.br/blogpatrus/?p=540>. Acesso em: 21 mar. 2019.

DI JUSTO, Patrick. **Raspberry pi or arduino uno? one simple rule to choose the right board**. San Francisco, 4 dez. 2015. Disponível em: <https://makezine.com/2015/12/04/admittedly-simplistic-guide-raspberry-pi-vs-arduino/>. Acesso em: 9 jul. 2019.

EATON, John W.; BATEMAN, David; HAUBERG, Søren; WEHBRING, Rik. **GNU Octave**: A high-level interactive language for numerical computations. 5. ed. Boston: Free Software Foundation, 2019. Disponível em: <https://octave.org/octave.pdf>. Acesso em: 18 ago. 2019.

ELAINA, Jeniffer. **O cenário dos carros autônomos no Brasil e no mundo**. 28 jun. 2019. Disponível em: <https://www.smartia.com.br/blog/carros-autonomos-no-brasil-e-no-mundo/>. Acesso em: 4 jul. 2019.

EVGHENII. **The basics of Siemens PLC's and programming in Simatic Step7**. San Francisco, 16 jan. 2017. Disponível em: [http://isd-soft.com/tech\\_blog/basics-siemens-plcs-programming-simatic-step7/](http://isd-soft.com/tech_blog/basics-siemens-plcs-programming-simatic-step7/). Acesso em: 8 jul. 2019.

FACELI, Katti; LORENA, Ana Carolina; GAMA, João; DE CARVALHO, André C. P. L. F. **Inteligência artificial**: uma abordagem de aprendizado de máquina. Rio de Janeiro: LTC, 2011.

FAST-MOVING consumer goods. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2018. Disponível em: [https://en.wikipedia.org/wiki/Fast-moving\\_consumer\\_goods](https://en.wikipedia.org/wiki/Fast-moving_consumer_goods). Acesso em: 21 mar. 2019.

GALDAMES, Jorge Pablo Moraga. **Sistema de localização para AGVs em ambientes semelhantes a armazéns inteligentes**. 2011. Dissertação (Mestre em Engenharia Mecânica) - Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos, 2011. Disponível em: <http://www.teses.usp.br/teses/disponiveis/18/18149/tde-29092014-104604/pt-br.php>. Acesso em: 23 mar. 2019.

GIMENEZ, Salvador Pinillos. **Microcontroladores 8051**: Teoria e prática. São Paulo: Érica, 2010.

GLOBAL Automated Guided Vehicle (AGV) Fleet Management Platform Market Will Reach USD 676,351 Thousand By 2027: Zion Market Research. **GLOBE NEWSWIRE**, New York. 17 abr. 2019. Disponível em: <https://www.globenewswire.com/news-release/2019/04/17/1805195/0/en/Global-Automated-Guided-Vehicle-AGV-Fleet-Management-Platform-Market-Will-Reach-USD-676-351-Thousand-By-2027-Zion-Market-Research.html>. Acesso em: 4 jul. 2019.

GOVERS, Francis X. **Artificial intelligence for robotics: build intelligent robots that perform human tasks using ai techniques**. Birmingham: Packt, 2018.

GRIMMETT, Richard. **Arduino robotic projects: Build awesome and complex robots with the power of Arduino**. Birmingham: Packt Publishing, 2014.

\_\_\_\_\_. **Raspberry pi robotic projects**. Birmingham: Packt Publishing, 2015.

HARVEY, Matt. **Using reinforcement learning in Python to teach a virtual car to avoid obstacles**. 6 fev. 2016. Disponível em: <https://blog.coast.ai/using-reinforcement-learning-in-python-to-teach-a-virtual-car-to-avoid-obstacles-6e782cc7d4c6>. Acesso em: 7 set. 2019.

\_\_\_\_\_. **Reinforcement learning in Python to teach a virtual car to avoid obstacles — part 2**. 25 fev. 2016. Disponível em: <https://blog.coast.ai/reinforcement-learning-in-python-to-teach-a-virtual-car-to-avoid-obstacles-part-2-93e614fcd238>. Acesso em: 7 set. 2019.

\_\_\_\_\_. **Reinforcement learning in Python to teach an RC car to drive itself — Part 3**. 8 mar. 2016. Disponível em: <https://blog.coast.ai/reinforcement-learning-in-python-to-teach-an-rc-car-to-avoid-obstacles-part-3-a1d063ac962f>. Acesso em: 7 set. 2019.

HAYKIN, Simon. **Redes neurais: princípios e prática**. 2. ed. Porto Alegre: Bookman, 2007.

INTRODUCTION Optical Track Guidance. Lehrte, [201-]. Disponível em: <https://www.goetting-agv.com/components/optical/introduction>. Acesso em: 7 jul. 2019.

JOSEPH, Lentin. **Learning robotics using python: Design, simulate, program, and prototype an interactive autonomous mobile robot from scratch with the help of Python, ROS, and Open-CV!**. Birmingham: Packt Publishing, 2015.

KAJAN, Martin *et al.* Control of Automated Guided Vehicle with PLC SIMATIC ET200S CPU. **American journal of mechanical engineering**, New York, ano 2013, v. 1, n. 7, p. 343-348, 4 out. 2013. Disponível em: <http://pubs.sciepub.com/ajme/1/7/38/>. Acesso em: 8 jul. 2019.

KANDEL, Eric R.; SCHWARTZ, James H.; JESSELL, Thomas M.; SIEGELBAUM, Steven A.; HUDSPETH, A. J. **Princípios de neurociências**. 5. ed. Porto Alegre: AMGH, 2014.

KANNIAH, Jagannathan; ERCAN, M. Fikret; CALDERON, Carlos A. Acosta. **Practical robot design: Game playing robots**. Boca Raton: CRC Press, 2014. 406 p.

KIVY. **Kivy**. 2019. Disponível em: <https://kivy.org/#home>. Acesso em: 5 set. 2019.

KOFFMAN, Elliot B.; WOLFGANG, Paul A. T. **Objetos, abstração, estrutura de dados e projeto usando C++**. Rio de Janeiro: LTC, 2008.

KREBS, Claudia; WEINBERG, Joanne; AKESSON, Elizabeth. **Neurociência ilustrada**. Porto Alegre: Artmed, 2013.

LOSADA, María Guinaldo; RUBIO, Francisco Rodríguez; BENCOMO, Sebastián Dormido. **Asynchronous control for networked systems**. New York: Springer, 2015.

LOMBA, Matheus. **Prototipagem eletrônica**: descubra como o Arduino e o Raspberry Pi podem facilitar sua vida. 28 mar. 2019. Disponível em: <https://fluxoconsultoria.poli.ufrj.br/blog/tecnologia-informacao/prototipagem-eletronica-arduino-e-raspberry-pi/>. Acesso em: 3 ago. 2019.

LUGEN, George F. **Inteligência artificial**. 6. ed. São Paulo: Pearson, 2013.

MAGRIN, Matheus Braun. **Expansão do módulo de redes neurais do GNU Octave**. Orientador: Luiz Angelo Daros de Luca. 2013. Monografia (Bacharelado em Ciência da Computação) - Universidade Federal de Santa Catarina, Florianópolis, 2013. Disponível em: [https://projetos.inf.ufsc.br/arquivos\\_projetos/projeto\\_1429/relatorio\\_final.pdf](https://projetos.inf.ufsc.br/arquivos_projetos/projeto_1429/relatorio_final.pdf). Acesso em: 18 ago. 2019.

MARTINEZ-BARBERA, Humberto; HERRERO-PEREZ, David. Development of a flexible AGV for flexible manufacturing systems. **Industrial robot**: an international journal, Murcia, v. 37, p. 459-468, 2010.

MATPLOTLIB. **Matplotlib**. 2019. Disponível em: <https://matplotlib.org/>. Acesso em: 5 set. 2019.

MCKEON, Richard. **Neural networks for electronics hobbyists**: A Non-Technical Project-Based Introduction. Prescott: Apress, 2018.

MCROBERTS, Michael. **Arduino básico**. São Paulo: Novatec, 2011.

MEZGÁR, I.; EGRESITS, Cs.; MONOSTORI, L. **Design and real-time reconfiguration of robust manufacturing systems by using design of experiments and artificial neural networks**. 1995. 10 f. Article (Computer Science) - Hungarian Academy of Science, Budapest, 1996.

MICHELUCCI, Umberto. **Applied deep learning**: a case-based approach to understanding deep neural networks. New York: Apress, 2018.

MONK, Simon. **Programação com arduino**: começando com sketches. 2. ed. Porto Alegre: Bookman, 2017.

MUELLER, John Paul; MASSARON, Luca. **Deep learning**: for dummies. Hoboken: Wiley, 2019.

NANDY, Abhishek; BISWAS, Manisha. **Reinforcement learning**: with open ai, tensorflow and keras using python. New York: Apress, 2018.

NELLI, Fabio. **Python data analytics**: with pandas, numpy, and matplotlib. 2. ed. New York: Apress, 2018.

NICOLOSI, Denys Emilio Campion. **Microcontrolador 8051 detalhado**. São Paulo: Érica, 2013.

NIELSEN, Michael. **Improving the way neural networks learn**. 2019. Disponível em: <http://neuralnetworksanddeeplearning.com/chap3.html#softmax>. Acesso em: 7 set. 2019.

NORRIS, Donald J. **Beginning artificial intelligence with the raspberry pi**. New York: Apress, 2017.

PATTERSON, Josh; GIBSON, Adam. **Deep learning: A practitioner's approach**. Sebastopol: O'Reilly, 2017.

PATIL, Virendra; G, Bhatwadekar S. Automated guided vehicle system. **International research journal of engineering and technology (IRJET)**, India, ano 2018, v. 05, ed. 4, p. 4038-4043, Apr 2018. Disponível em: <https://www.irjet.net/archives/V5/i4/IRJET-V5I4901.pdf>. Acesso em: 23 mar. 2019.

PEUGEOT Citroen: Raspberry Pi factory robot. 2007. Disponível em: <https://www.raspberrypi.org/magpi/peugeot-citroen-robots/>. Acesso em: 11 maio 2019.

PYTORCH. **Pytorch**. 2019. Disponível em: <https://pytorch.org/>. Acesso em: 5 set. 2019.

QU, Jerry. **Training self driving cars using reinforcement learning**. 22 out. 2018. Disponível em: <https://towardsdatascience.com/reinforcement-learning-towards-general-ai-1bd68256c72d>. Acesso em: 7 set. 2019.

\_\_\_\_\_. **I created a virtual self driving car with deep q-networks**. 1 nov. 2018. Disponível em: <https://towardsdatascience.com/i-created-a-virtual-self-driving-car-with-deep-q-networks-7f87c0aad7c8>. Acesso em: 7 set. 2019.

RAMSUNDAR, Bharath; ZADEH, Reza Bosagh. **Tensorflow for deep learning: from linear regression to reinforcement learning**. Sebastopol: O'Reilly, 2018.

REIS, Bernardo. **Redes neurais – funções de ativação**. Ouro Preto, 11 jul. 2016. Disponível em: <http://www.decom.ufop.br/imobilis/redes-neurais-funcoes-de-ativacao/>. Acesso em: 7 set. 2019.

ROMERO, Roseli Aparecida F. *et al.* **Robótica móvel**. Rio de Janeiro: LTC, 2017.

ROSÁRIO, João Maurício. **Princípios de mecatrônica**. São Paulo: Pearson, 2005.

RUSSELL, Stuart; NORVIG, Peter. **Inteligência artificial**. Rio de Janeiro: Elsevier, 2013.

SARKAR, Dipanjan; BALI, Raghav; SHARMA, Tushar. **Practical machine learning with python: a problem-solver's guide to building real-world intelligent systems**. New York: Apress, 2018.

SCHAUL, Tom; QUAN, John; ANTONOGLOU, Ioannis; SILVER, David. **Prioritized experience replay**. In: INTERNATIONAL CONFERENCE ON LEARNING REPRESENTATIONS, 2016, San Juan. Disponível em: <https://arxiv.org/pdf/1511.05952.pdf>. Acesso em: 7 set. 2019.

SHUKLA, Nishant. **Machine learning with tensorflow**. New York: Manning, 2017.

SISTEMA EMBARCADO. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2019. Disponível em: [https://pt.wikipedia.org/w/index.php?title=Sistema\\_embarcado&oldid=56259715](https://pt.wikipedia.org/w/index.php?title=Sistema_embarcado&oldid=56259715). Acesso em: 30 set. 2019.

SOARES, Fábio M.; SOUZA, Alan M. F. **Neural network programming with java**: unleash the power of neural networks by implementing professional Java code. Birmingham: Packt Publishing, 2016.

SPYDER the scientific python development environment. **Spyder**. 2018. Disponível em: <https://www.spyder-ide.org/>. Acesso em: 5 set. 2019.

SUPERDATASCIENCE & IA EXPERT. **Augmented random search (ARS)**. 2019. 55 slides.

SUTTON, Richard S.; BARTO, Andrew G. **Reinforcement learning**: an introduction. 2. ed. Cambridge: MIT Press, 2018.

\_\_\_\_\_. **Reinforcement learning I**: introduction. 1998. Disponível em: <http://citeseer.ist.psu.edu/viewdoc/download?doi=10.1.1.32.7692&rep=rep1&type=pdf>. Acesso em: 7 set. 2019.

SWAMYNATHAN, Manohar. **Mastering machine learning with python in six steps**: a practical implementation guide to predictive data analytics using python. New York: Apress, 2017.

TESLA anuncia sistema 100% autônomo para todos os seus carros. **G1**, São Paulo, 20 out. 2016. Disponível em: <http://g1.globo.com/carros/noticia/2016/10/tesla-anuncia-sistema-100-autonomo-para-todos-os-seus-carros.html>. Acesso em: 2 mar. 2019.

THE top programming languages. 2019. Disponível em: <https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2019>. Acesso em: 30 set. 2019.

ULLRICH, Günter. **Automated guided vehicle systems**: a primer with practical applications. 2. ed. Voerde: Springer, 2015.

UPTON, Eben; HALFACREE, Gareth. **Raspberry pi user guide**. United Kingdom: Wiley, 2012.

VERGARA, Sylvia Constant. **Projetos e relatórios de pesquisa em administração**. 14. ed. São Paulo: Atlas, 2013.

VILLANUEVA, Juan Mauricio. **Sistema de inferência fuzzy**. 2016. 33 slides. Disponível em: <http://www.cear.ufpb.br/juan/wp-content/uploads/2016/08/Aula-1-L%C3%B3gica-Fuzzy.pdf>. Acesso em: 24 ago. 2019.

WHAT is Arduino?. [201-]. Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Acesso em: 11 maio 2019.

## GLOSSÁRIO

**AGV** - *Automated Guided Vehicle* (Veículo guiado automaticamente). Robô móvel usado geralmente em aplicações industriais no transporte de materiais.

**Automação** - Sistema de controle automático.

**CLP** - Controlador Lógico Programável. Computador de aplicação industrial usado em automação e robótica.

**Inteligência Artificial** - Inteligência similar a do cérebro humano, mas em uma máquina ou software.

**Rede neural** - Modelo computacional baseado no sistema nervoso central biológico capaz de realizar aprendizado.

**Robótica** - Ciência e técnica da concepção, construção e utilização de robôs.

**Robô** - Dispositivo eletromecânico que executa uma determinada tarefa dependendo das instruções dadas, controladas por circuitos elétricos e/ou eletrônicos.



## Apêndice B - Pseudocódigo do algoritmo criado a partir da tabela-verdade dos sensores e atuadores

```

// Ler o estado dos sensores
leia sensor_frente
leia sensor_direito
leia sensor_esquerdo
// Se não houver nenhum obstáculo detectado por sensor_frente, o robô vai se mover para
// a frente.
se sensor_frente == 0 && sensor_direito == 0 && sensor_esquerdo == 0
    ou sensor_frente == 0 && sensor_direito == 0 && sensor_esquerdo == 1
    ou sensor_frente == 0 && sensor_direito == 1 && sensor_esquerdo == 0
    ou sensor_frente == 0 && sensor_direito == 1 && sensor_esquerdo == 1
        mova_frente // motor_direito = 1; motor_esquerdo = 1
    senão se
        sensor_frente == 1 && sensor_direito == 0 && sensor_esquerdo == 0
            mova_esquerda // motor_direito = 0; motor_esquerdo = 1
    senão se
        sensor_frente == 1 && sensor_direito == 0 && sensor_esquerdo == 1
            mova_direita // motor_direito = 0; motor_esquerdo = 1
    senão se
        sensor_frente == 1 && sensor_direito == 1 && sensor_esquerdo == 0
            mova_esquerda // motor_direito = 1; motor_esquerdo = 0
    senão
        sensor_frente == 1 && sensor_direito == 1 && sensor_esquerdo == 1
            ficar_parado // motor_direito = 0; motor_esquerdo = 0

```

## Apêndice C - Banner apresentado no X Simpósio de Tecnologia de Informação

Câmpus  
Anápolis de Ciências  
Exatas e Tecnológicas  
Henrique Santillo

Universidade  
Estadual de Goiás

BACHARELADO EM SISTEMAS DE  
INFORMAÇÃO

X Simpósio de Tecnologia de Informação  
XI Semana de Iniciação Científica do Curso de Sistemas de Informação  
IV Colóquio de Estágio

Estudo do Desenvolvimento de um Veículo Autônomo

Autor: Jorge Brayer Gazal Junior  
Orientador: Prof. Dr. Francisco Ramos de Melo

jbrayer@gmail.com  
francisco.melo@ueg.br

### INTRODUÇÃO

Veículos Guiados Automaticamente podem se apresentar como uma solução interessante na realização de tarefas repetitivas, perigosas e outras situações nos diferentes processos produtivos e de armazenamento.

Atualmente existem diferentes versões desses veículos, de acordo com o tipo de aplicação e ambiente de trabalho.

No geral, o custo destes veículos é alto por diversos fatores, mas principalmente por causa do valor elevado dos componentes utilizados.

Este trabalho busca investigar a possibilidade de implementação de um AGV com viabilidade econômica, fundamentada em controladores de baixo custo.

### OBJETIVOS

A pesquisa tem como objetivo geral estudar tecnologias de desenvolvimento de veículos autônomos e plataformas eletrônicas e analisar a viabilidade do desenvolvimento de um veículo inteligente de baixo custo.

### REFERENCIAL TEÓRICO

Dentro da robótica existe um grupo de robôs utilizado em muitos ramos da indústria e áreas de produção que realiza o transporte de peças, componentes e equipamentos, trabalhando de forma autônoma e podendo funcionar 24 horas por dia conhecidos como AGV (do inglês *Automated Guided Vehicle* - Veículo Guiado Automaticamente).

Um AGV é um robô móvel, autônomo, programável, que se move orientado por diferentes tecnologias de navegação. AGVs são usados em diferentes indústrias para o transporte de mercadorias em fábricas e depósitos.



As duas plataformas mais populares atualmente existentes para a criação de projetos eletrônicos são o Arduino e o Raspberry Pi. Muitas literaturas apresentam a utilização destas plataformas no desenvolvimento de projetos de AGVs em pequena escala, desde aplicações simples de

robôs que seguem uma linha até os mais complexos, que utilizam técnicas de inteligência artificial.



O Arduino é um controlador enquanto o Raspberry Pi é um computador completo. A diferença está nos elementos que os controlam: Microcontroladores e Microprocessadores. Um microprocessador é um componente eletrônico que executa tarefas específicas gravadas em sua memória, enquanto o microcontrolador integra as partes de um microcomputador em um único dispositivo, usado para o controle de tarefas simples e para armazenar e manipular pequenas quantidades de dados.

A fabricante de carros Peugeot Citroën, na sua fábrica em Portugal, criou um AGV utilizando o Raspberry Pi. Este AGV é usado no transporte de carros dentro da fábrica. Um sensor magnético é usado para guiar o AGV ao longo da faixa magnética que percorre a fábrica.



### METODOLOGIA

A pesquisa se baseou nos trabalhos publicados em livros, artigos acadêmicos e websites a fim de explicar o que é um AGV, sua construção e seu funcionamento.

Com base nos robôs desenvolvidos pelos diferentes pesquisadores e nos estudos de sistemas inteligentes buscou-se implementar um AGV em pequena escala que a princípio realize o desvio de obstáculos e, caso não seja possível essa implementação, será realizada a simulação do funcionamento de uma rede neural que tenha esse objetivo.

### RESULTADOS

Como resultados preliminares foi possível identificar plataformas de prototipagem eletrônica de baixo custo, o Arduino e o Raspberry Pi, sendo o Raspberry Pi o que

melhor atende ao objetivo de criar um AGV inteligente que tenha aplicação industrial devido ao seu maior poder de processamento e armazenamento, sendo possível substituir o CLP (Controlador Lógico Programável) que é o controlador geralmente utilizado por fabricantes de AGV comerciais.

O CLP tem um custo elevado comparado com o custo de plataformas de prototipagem eletrônica, e oferece menor possibilidade de processamento e armazenamento, o que dificulta a implementação de uma rede neural que torne o mesmo mais autônomo e inteligente.

O custo de um CLP, sem os módulos de entrada, saída e outros que possam ser necessários é de aproximadamente R\$2.000,00. Com a adição dos módulos necessários esse valor pode ultrapassar os R\$4.000,00. O Raspberry Pi já possui as entradas e saídas disponíveis, e o seu valor chega a ser dez vezes menor que o de um CLP, custando na faixa de R\$200,00.

### CONSIDERAÇÕES FINAIS

Os estudos sobre o tema indicam que é possível construir um protótipo de AGV de baixo custo que use uma plataforma de prototipagem eletrônica para controlar o veículo, como é o caso do AGV desenvolvido pela Peugeot Citroën. O AGV da Peugeot Citroën se guia por faixa magnética, mas seria possível implementar inteligência artificial de maneira a torná-lo inteligente. Um exemplo do uso de inteligência seria poder fazer o AGV mapear o ambiente e saber as melhores rotas a serem tomadas.

Existem diversos estudos e protótipos desenvolvidos com plataformas eletrônicas onde foi implantada inteligência artificial, utilizando principalmente redes neurais.

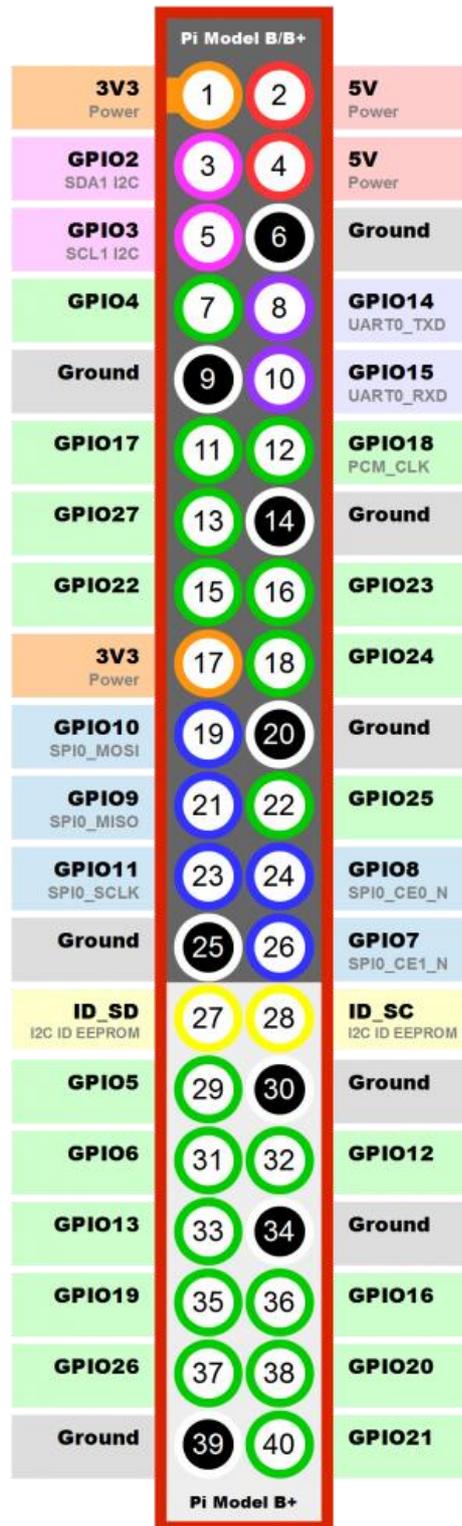
Diante do exposto, a possibilidade da construção de um AGV com aplicação industrial e com custo mais acessível se mostra uma alternativa viável partindo da inserção de plataformas de prototipagem eletrônica como o Arduino e o Raspberry.

### REFERÊNCIAS

- GIMENEZ, Salvador Pinillos. *Microcontroladores 8051: Teoria e prática*. São Paulo: Érica, 2010.
- NICOLOSI, Denys Emilio Campion. *Microcontrolador 8051 Detalhado*. São Paulo: Érica, 2013.
- ULLRICH, Günter. *Automated Guided Vehicle Systems: A Primer with Practical Applications*. 2. ed. Voerde: Springer, 2015.
- PEUGEOT Citroën: *Raspberry Pi factory robot*. 2007. Disponível em: <<https://www.raspberrypi.org/magpi/peugeot-citroen-robots/>>. Acesso em: 11 maio 2019.

## ANEXOS

### Anexo A - Pinagem do Raspberry Pi 3 B+



[www.raspberrypi-spy.co.uk](http://www.raspberrypi-spy.co.uk)

Fonte: <https://www.raspberrypi-spy.co.uk/2014/07/raspberrypi-b-gpio-header-details-and-pinout/>

## Anexo B - Pinagem do sensor ultrassônico HC-SR04



VCC = +5VDC

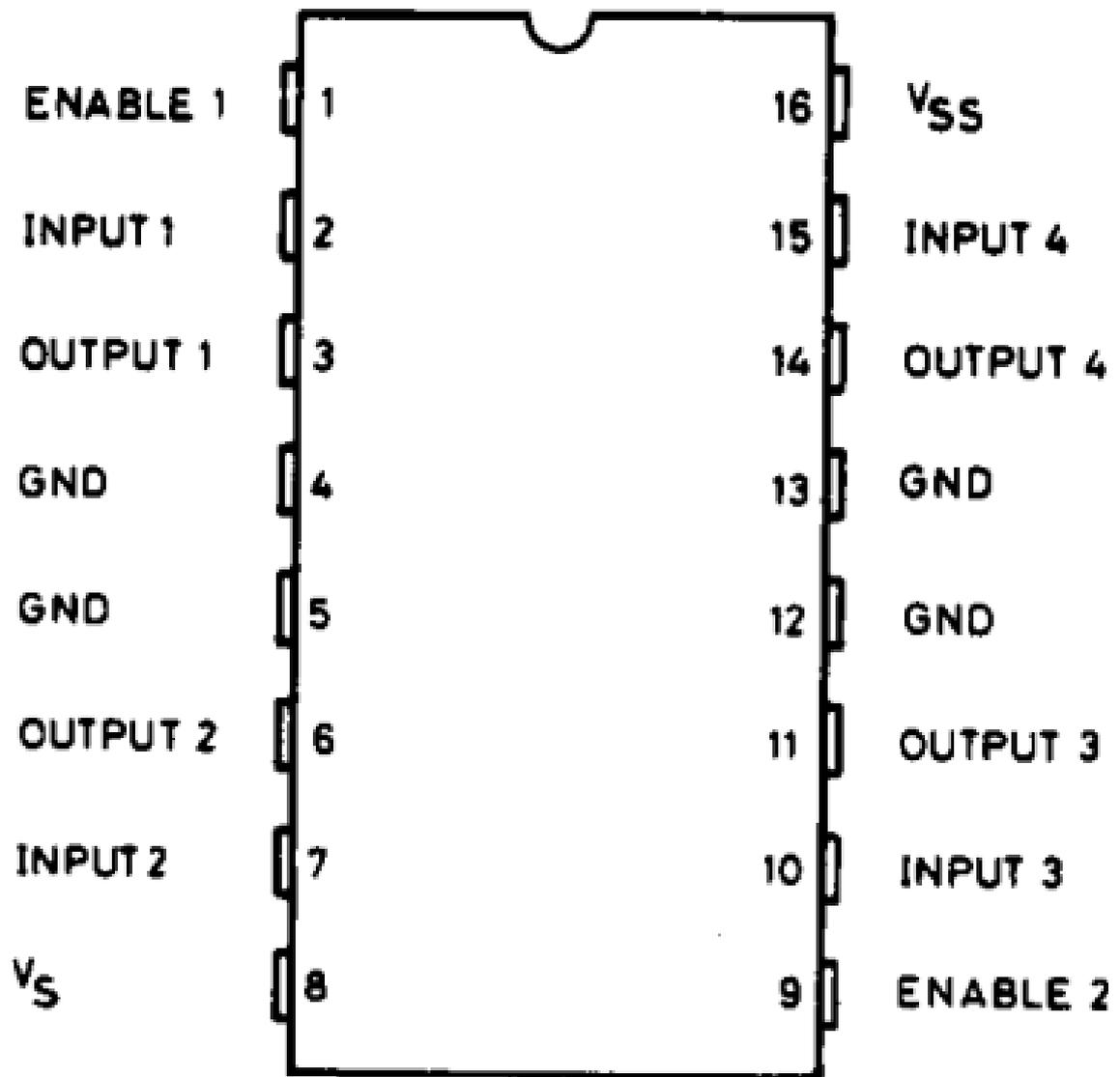
Trig = Trigger input of Sensor

Echo = Echo output of Sensor

GND = GND

Fonte: [raspoid.com/download/datasheet/HCSR04](http://raspoid.com/download/datasheet/HCSR04)

## Anexo C - Pinagem da Ponte H L293D



S-6574

Fonte: [https://www.arduino.cc/documents/datasheets/H-bridge\\_motor\\_driver.PDF](https://www.arduino.cc/documents/datasheets/H-bridge_motor_driver.PDF)