

**UNIVERSIDADE DO SAGRADO CORAÇÃO**

**CARLOS EDUARDO DOS REIS**

**CÁLCULO DE TRAJETÓRIA DE UM ROBÔ DE  
TRANSPORTE DE CARGAS UTILIZANDO  
ALGORITMO DE BUSCA**

BAURU  
2016

**CARLOS EDUARDO DOS REIS**

**CÁLCULO DE TRAJETÓRIA DE UM ROBÔ DE  
TRANSPORTE DE CARGAS UTILIZANDO  
ALGORITMO DE BUSCA**

Trabalho de conclusão de curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade do Sagrado Coração, como parte dos requisitos para obtenção do título de bacharel em Ciência da Computação, sob a orientação do Prof.Me. Patrick Pedreira Silva.

BAURU  
2016

Reis, Carlos Eduardo dos

R3751c

Cálculo de trajetória de um robô de transporte de cargas utilizando algoritmo de busca/Carlos Eduardo dos Reis. -- 2016.

64f. : il.

Orientador: Prof. M.e Patrick Pedreira Silva.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade do Sagrado Coração – Bauru – SP.

1. Arduino. 2. Inteligência Artificial. 3. Robôs. 4. Automação. I. Silva, Patrick Pedreira. II. Título.

**CARLOS EDUARDO DOS REIS**

**CÁLCULO DE TRAJETÓRIA DE UM ROBÔ DE TRANSPORTE DE CARGAS UTILIZANDO ALGORITMO DE BUSCA**

Trabalho de Conclusão de Curso apresenta do ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade do Sagrado Coração como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, sob a orientação do Prof.Me. Patrick Pedreira Silva.

Banca examinadora:

---

Prof. Me. Patrick Pedreira Silva  
Universidade do Sagrado Coração

---

Prof. Dr. Elvio Gilberto da Silva  
Universidade do Sagrado Coração

---

Prof. Me. Henrique Pachioni Martins  
Universidade do Sagrado Coração

Bauru, 12 de dezembro de 2016.

## **AGRADECIMENTOS**

Primeiramente agradeço a Deus por poder concluir mais um objetivo em minha vida, me abençoando e podendo dar força, para cumprir cada etapa desse trabalho, podendo sempre estar ao meu lado em todas as horas. Agradeço também aos meus pais Antonio Carlos e Maria Lucia que sempre estiveram do meu lado podendo fazer de tudo para que eu pudesse cumprir meus objetivos na vida, me dando esperança e me ensinando como é a vida. Agradeço a eles também por toda a oportunidade que me deram com relação à educação e saúde. Dedico esse trabalho aos meus avôs tanto presente aqui ou não, mas sempre estarão em meu coração, me guiando para uma vida melhor e também os agradeço pelos maravilhosos pais que me deram, e agradeço aos meus irmãos por estarem sempre do meu lado.

Também agradeço e dedico esse trabalho ao meu Orientador Patrick Pedreira e meus professores que percorreram toda essa trajetória acadêmica, ao meu lado me apoiando e ensinando tudo que eu posso aprender de melhor, podendo nos mostrar como será a vida lá fora no mercado de trabalho e podendo concorrer a grandes vagas de entrega com todo meu aprendizado. Agradeço também a todos os amigos que pude fazer nessa jornada, podendo ajudá-los e ser ajudado.

Quero também agradecer à Universidade do Sagrado Coração com todos seus qualificados profissionais e pessoas acima de tudo.

## RESUMO

Com o grande avanço ocorrido na Robótica desde o século XXI, os robôs passaram a fazer parte da nossa vida. Na área industrial é ainda mais comum a atuação desses agentes. Dentre os tipos de robôs industriais, existem os chamados manipuladores que executam movimentação de objetos. Com passar do tempo surgiram conceitos de robôs com Inteligência Artificial. A Inteligência Artificial é um campo da Ciência da Computação e da Engenharia de Computação que procura soluções por meios computacionais, focando em técnicas que associem aos agentes várias capacidades de raciocinar, planejar, resolver problemas, comunicar-se através de uma linguagem, e ao mesmo tempo, que tenha a habilidade de obter sucesso no tratamento de novas situações. O problema da definição de rotas (trajetórias) é um exemplo de situação em que é possível encontrar boas soluções por meio dos algoritmos de busca. Um algoritmo de busca, em termos gerais é um algoritmo que toma um problema como entrada e retorna a solução para o problema, geralmente após resolver um número possível de soluções. Um exemplo típico da aplicação de algoritmo de busca é na resolução de problemas de trajetória. Tais algoritmos permitem, por exemplo, a um robô determinar sua trajetória a partir de busca em árvore ou grafos contendo estados válidos. Considerando esse contexto, o objetivo desse projeto é a automação de um cálculo da trajetória de um robô que empilha produtos em um depósito de mercadoria (ambiente simulado). O hardware do robô teve como base o Arduino, e sua inteligência é pautada na implementação de um algoritmo de busca, especificamente a busca de custo uniforme.

**Palavras-chave:** Arduino. Inteligência Artificial. Robôs. Automação.

## ABSTRACT

With the great progress that has taken place in Robotics since the 21st century, robots have become part of our lives. In the industrial area, these agents are even more common. Among the types of industrial robots, there are so-called manipulators that perform object movement. With the passing of time, concepts of robots with Artificial Intelligence emerged. Artificial Intelligence is a field of Computer Science and Computer Engineering that seeks solutions by computational means, focusing on techniques that associate agents with various abilities to reason, plan, solve problems, communicate through a language, and at the same time, who has the ability to succeed in dealing with new situations. The problem of defining routes (paths) is an example of a situation where you can find good solutions through search algorithms. A search algorithm in general terms is an algorithm that takes a problem as input and returns the solution to the problem, usually after solving a possible number of solutions. A typical example of the search algorithm application is in the resolution of trajectory problems. Such algorithms allow, for example, a robot to determine its trajectory from tree search or graphs containing valid states. Considering this context, the objective of this project is the automation of a calculation of the trajectory of a robot that stacks products in a warehouse of merchandise (simulated environment). The hardware of the robot was based on the Arduino, and its intelligence is based on the implementation of a search algorithm, specifically the uniform cost search.

**Keywords:** Arduino. Artificial Intelligence. Robots. Automation.

## LISTA DE ILUSTRAÇÕES

Figura 1- Áreas relativas com IA. ....	12
Figura 2 - Estrutura de um Grafo.....	14
Figura 3 - Estrutura de um Grafo ponderado.....	15
Figura 4 -Representação de um grafo de navegação. ....	16
Figura 5 - Exemplo de um mapa rodoviário fictício. ....	17
Figura 6 -Progresso da Busca em Profundidade.....	19
Figura 7 - Busca em largura em uma árvore binária simples. ....	20
Figura 8 - Amazon implementa robôs Kiva para .....	24
Figura 9 - Placa Arduino Uno .....	25
Figura 10 - Placa Arduino Mega.....	26
Figura 11 - Ambiente de Desenvolvimento .....	27
Figura 12 - Estrutura de programa C.....	28
Figura 13 - Imagem ilustrativa do Sensor RFID. ....	32
Figura 14- Imagem ilustrativa do Sensor.....	33
Figura 15- Imagem ilustrativa do Chassi .....	33
Figura 16– Modulo Segue Linha .....	34
Figura 17– Arquitetura do sistema .....	34
Figura 18– Configuração do Ambiente.....	35
Figura 19– Robô Móvel. ....	37
Figura 20– Chassi do Robô Móvel. ....	38
Figura 21 – Montagem do Chassi Rodas e Motores .....	39
Figura 22–Ligação dos Motores.....	39
Figura 23 - Ligações do Sensor RFID .....	40
Figura 24 - Ligação Sensor Ultrassônico com Arduino. ....	41
Figura 25 - Sensor Segue Linhas.....	42
Figura 26 - Arduino Uno no Robô Móvel. ....	43
Figura 27– IDE Arduino comunicando com Robô Móvel.....	43
Figura 28– Leitura e Captura de Dados do Produto .....	44
Figura 29– Trajetoria de Menor Custo.....	44
Figura 30 - Identificando o objetivo. ....	46
Figura 31 - Início da trajetória.....	46
Figura 32 - Meio da trajetória .....	487
Figura 33 - Chegada do Objetivo .....	487
Figura 34 - Obstáculo à frente do robô.....	48



## SUMÁRIO

<b>1.</b>	<b>INTRODUÇÃO</b>	8
<b>2.</b>	<b>OBJETIVOS</b>	10
2.1.	OBJETIVO GERAL	10
2.2.	OBJETIVOS ESPECÍFICOS	10
<b>3.</b>	<b>INTELIGÊNCIA ARTIFICIAL</b>	11
3.1.	HISTÓRIA DA INTELIGÊNCIA ARTIFICIAL	12
3.2.	ALGORITMOS DE BUSCA	13
3.3.	BUSCA CEGA	18
<b>3.3.1.</b>	<b>Busca em Profundidade</b>	19
<b>3.3.2.</b>	<b>Busca em Largura</b>	19
<b>3.3.3.</b>	<b>Busca de Custo Uniforme</b>	20
3.4.	BUSCA HEURÍSTICA	20
<b>4.</b>	<b>ROBÓTICA</b>	22
4.1.	PLATAFORMA ARDUINO	24
<b>4.1.1.</b>	<b>Hardware</b>	25
4.1.1.1.	<i>Arduino Uno</i>	25
4.1.1.2.	<i>Arduino Mega</i>	25
<b>4.1.2.</b>	<b>Sensores e Módulos</b>	26
<b>4.1.3.</b>	<b>Ambiente de Trabalho</b>	26
4.1.3.1.	Linguagem C	27
4.1.3.1.1.	Estrutura de Programa em C	28
<b>5</b>	<b>TRABALHOS CORRELATOS</b>	29
<b>6</b>	<b>METODOLOGIA</b>	31
6.1.	REVISÃO BIBLIOGRÁFICA	31
6.2.	CONSTRUÇÃO DO ROBÔ MÓVEL	31
6.3.	ARQUITETURA DO SISTEMA E DESCRIÇÃO DO AMBIENTE	34
6.4.	AVALIAÇÃO DO SISTEMA	36
<b>7.</b>	<b>RESULTADOS</b>	37
7.1.	ARQUITETURA DO ROBÔ	37
<b>7.1.1.</b>	<b>Elementos da Estrutura do Robô</b>	37
7.1.1.1.	Chassi do Robô (Base)	37
7.1.1.2.	Rodas e Motores	38

7.1.1.3. Arduino Uno e Sensores.....	40
<b>8. TESTE DO PROTÓTIPO.....</b>	<b>45</b>
<b>9. CONSIDERAÇÕES FINAIS.....</b>	<b>49</b>
<b>10. REFERÊNCIAS.....</b>	<b>51</b>
<b>11. ANEXOS.....</b>	<b>Erro! Indicador não definido.</b>

## 1 INTRODUÇÃO

Com o grande avanço ocorrido na Robótica desde o século XXI, os robôs passaram a fazer parte da nossa vida. Na área industrial é ainda mais comum a atuação desses agentes. Dentre os tipos de robôs industriais, existem os chamados manipuladores que executam movimentação de objetos. Esses robôs são como um veículo capaz de movimentar objetos em vários tipos de trajetórias como ponto a ponto, trajetórias de coordenadas ou isócrona e contínuas. Em uma trajetória ponto a ponto o comando de movimento de uma articulação que é independente do das demais tendo uma junta que pode alcançar seu destino em um menor tempo possível. Robôs móveis possuem uma capacidade de se movimentar ao redor do seu ambiente, não estando fixados em uma localização única. Eles são compostos por sensores, motores, que facilitam sua movimentação. Com passar do tempo surgiram conceitos de robôs com inteligência artificial.

A Inteligência Artificial se iniciou no ano de 1956 no famoso encontro de Dartmouth. Considerado um campo da Ciência da Computação e da Engenharia de Computação procura soluções por meios computacionais, focando em técnicas que associem aos agentes várias capacidades de raciocinar, planejar, resolver problemas, comunicar-se através de uma linguagem, e ao mesmo tempo, que tenha a habilidade de obter sucesso no tratamento de novas situações.

A abordagem de problemas nem sempre são fáceis, uma vez que muitos desses problemas são complexos para serem resolvidos através da capacidade de programação tradicional. O problema da definição de rotas (trajetórias) é um exemplo de situação em que é possível encontrar boas soluções por meio dos algoritmos de busca. Um algoritmo de busca, em termos gerais é um algoritmo que toma um problema como entrada e retorna a solução para o problema, geralmente após resolver um número possível de soluções. Um exemplo típico da aplicação de algoritmo de busca é na resolução de problemas de trajetória. Tais algoritmos permitem ao robô determinar sua trajetória a partir de busca em árvore ou grafos contendo estados válidos. Assim prove em robôs a capacidade de planejar rotas, viabilizando um deslocamento seguro na busca de um objetivo, que pode ser o simples deslocamento de um ponto a outro.

A possibilidade de expandir a capacidade de agentes robóticas move, sobretudo com técnicas de IA, tem estimulado o desenvolvimento de plataformas de prototipagem eletrônica que torna a robótica mais acessível a todos. Um exemplo dessa plataforma é o Arduino.

O primeiro Arduino surgiu no ano de 2005, sendo desenvolvido pelo professor Massimo Banzi que tentou criar alguma forma barata e fácil para que alunos pudessem trabalhar com programação eletrônica. As suas principais características incluem uma plataforma de código aberto que traz várias versões de Arduino e baseada em um microprocessador de 8 bits, tendo possibilidade de qualquer pessoa utilizar e recriar em cima da produção de outros, utilizando a linguagem C para desenvolvimento na programação da plataforma.

A linguagem C foi criada em 1972 no Bell Telephone Laboratories, tendo como objetivo escrita de um sistema operacional (O UNIX). As funcionalidades da linguagem permitem manipular bits, bytes e endereços, que são elementos básicos para funcionamento de um computador. A linguagem C tem cinco tipos de dados internos, pois não é uma linguagem muito rica em tipos de dados, sendo adequada à programação estruturada.

Considerando esse contexto, o objetivo desse projeto é a automação de um cálculo da trajetória de um robô que empilha produtos em um depósito de mercadoria (ambiente simulado). O hardware do robô terá como base o Arduino, e sua inteligência será pautada na implementação de um algoritmo de busca, especificamente a busca de custo uniforme.

## 2 OBJETIVOS

A seguir serão apresentados os objetivos deste trabalho.

### 2.1 OBJETIVO GERAL

Desenvolver um sistema robótico de transporte de carga utilizando algoritmos de busca para calcular sua trajetória.

### 2.2 OBJETIVOS ESPECÍFICOS

- a) Realizar uma investigação teórica sobre inteligência Artificial, Algoritmos de busca e plataforma Arduino;
- b) Estudar e definir uma linguagem de programação adequada para implementação de agente robótica na plataforma Arduino;
- c) Planejar um ambiente estático (com e sem obstáculos) para atuação de um robô;
- d) Escolher a melhor forma de representar trajetórias considerando e teoria dos algoritmos de busca;
- e) Propor diferentes configurações para os algoritmos utilizados;
- f) Realizar o cálculo de trajetórias utilizando algoritmos de busca executá-los no protótipo;
- g) Fazer a coleta de resultados para verificação do comportamento dos algoritmos de busca no ambiente proposto;

### 3 INTELIGÊNCIA ARTIFICIAL

A Inteligência Artificial (IA) teve seu início após a segunda guerra mundial, englobando uma enorme diversidade de sub campos, desde áreas de uso geral tais como aprendizado e percepção, quanto tarefas específicas como jogos de xadrez, demonstração de teoremas matemáticos e diagnóstico de doenças, estando também associada à sistematização e automatização de tarefas intelectuais, essencialmente voltadas para atividades intelectuais humanas. (RUSSEL; NORVIG, 2004).

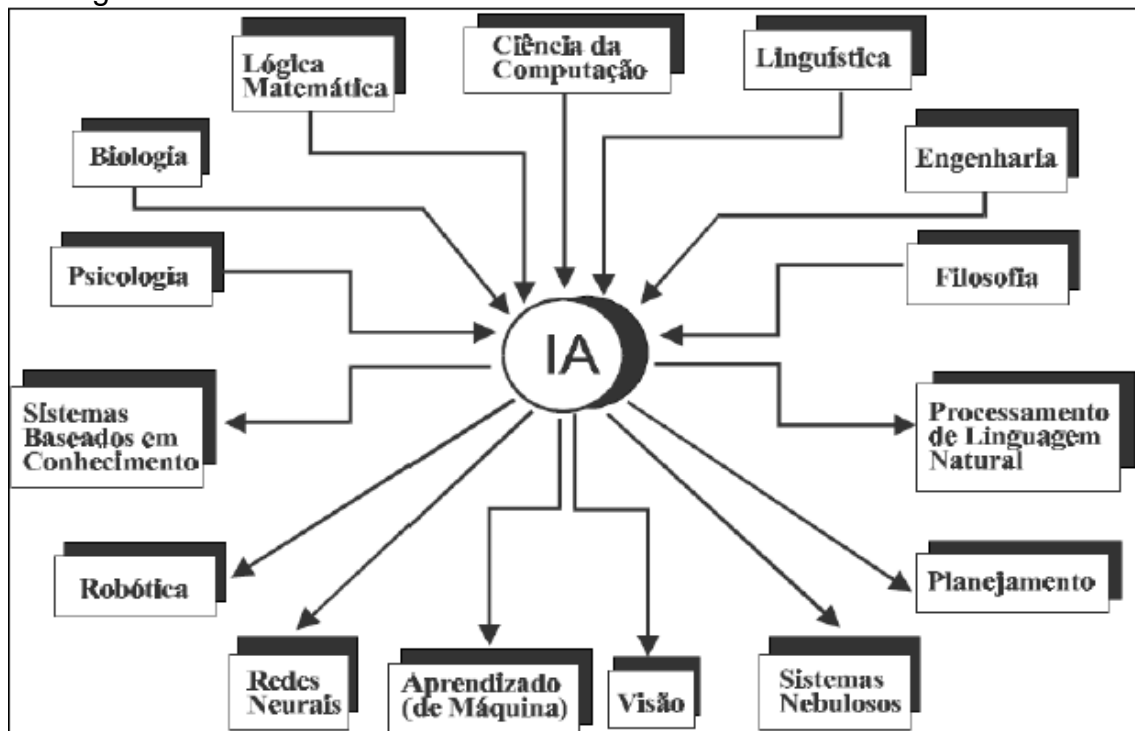
De acordo com Haugeland (1985) “O novo e interessante esforço para fazer os computadores pensarem... máquinas com mentes, no sentido total literal”, relacionasse com o processo de pensamentos e raciocínio.

Segundo POOLE (1998) “A Inteligência Computacional é o estudo do projeto de agentes inteligentes” cujo sucesso está atrelado a um conceito ideal de inteligência/racionalidade. Nesse sentido, um sistema é racional se faz tudo certo com os dados que tem. (RUSSELL; NORVIG, 2004).

Para Russel e Norvig (2004) os filósofos já muito antes dos computadores procuravam a resposta para entender o funcionamento da mente humana, tendo mesmo o objetivo da IA. Procurava-se verificar as máquinas podiam agir de forma inteligente que é chamado de hipótese de Inteligência Artificial fraca.

A Inteligência Artificial é considerada um ramo da Ciência da Computação cujo ganho é fazer com que o computador pense e se comporte de forma inteligente. A IA também pode ser considerado em áreas de psicologia, biologia, lógica matemática, lingüística, engenharia, filosofia e outras áreas científicas conforme mostra a Figura1.

Figura 1- Áreas relativas com IA.



Fonte: MONARD; BARANAUKAS (2000).

### 3.1 HISTÓRIA DA INTELIGÊNCIA ARTIFICIAL

De acordo com Russel e Norvig (2004) o primeiro trabalho relacionado com Inteligência Artificial foi desenvolvido por Warren McCulloch e Walter Pitts (1943) e estava focado no conhecimento da fisiologia básica e da função dos neurônios do cérebro. Outros trabalhos pioneiros na área de IA relacionavam-se com uma análise formal da lógica proposicional criada por Russel e Whitehead e com a teoria da computação de Turing. Contudo Alan Turing foi o primeiro a articular uma visão completa da Inteligência Artificial, sendo o responsável por apresentar o Teste Turing, que é baseado em impossibilidade de distinguir entre entidades inegavelmente inteligentes. O computador passará neste teste se um interrogador humano interagindo por meio de perguntas escritas, não conseguir decifrar se a resposta vem de uma pessoa ou de uma máquina.

Para Russel e Norvig (2004) o período de 1952 a 1969 foi marcado por grande entusiasmo e expectativas, John McCarthy, Hyman Minsky, Claude Shannon e Nathaniel Rochester foram os principais idealizadores da época, e responsáveis por organizar um seminário em Dartmouth no ano de 1965 onde havia mais participantes como Trenchard More (Princeton), Arthur Samuel (IBM), Allen Newell e

Herbet Simon (CMU), Ray Solomonoff e Oliver Selfridge (MIT). Neste evento os principais participantes foram Allen Newell e Herbet Simon, com o programa desenvolvido para raciocínio, o Logictheorist (LT).

No período de 1966 a 1979 o foco das pesquisas de IA foram os sistemas baseados em conhecimento. Na década de 1960, o pesquisador Simon fez uma previsão muito ousada. Segundo ele dentro de 10 dez anos um computador teria condições de poder jogar xadrez e teorema matemático seria amplamente provado por uma máquina.

No ano de 1969 a Universidade de Stanford desenvolveu programa DENTRAL que desenvolvia soluções capazes de encontrar as estruturas moleculares orgânicas. Graça ao modo automático de tomar decisões o programa teve sua importância como software inteligente representando o primeiro sistema bem-sucedido de conhecimento intensivo. (RUSSEL; NORVIG,2004).

Desde o período de 1980 até hoje, a Inteligência Artificial tornou-se uma indústria. O primeiro sistema especialista comercial bem-sucedido foi o R1. Tal programa desenvolvido pela Digital Equipment Corporation (DEC) contribuía para configurar pedidos de novos sistemas de computador. No ano de 1986, o programa já fazia a empresa faturar 40 milhões de dólares por ano, em 1988 o grupo da DEC já possuía 40 sistemas especialistas entregues. (CHARNIAK; MCDERMOTT, 1985).

De acordo com Russel e Norvig (2004), no período de 1981 os japoneses anunciaram um projeto (oFifthGeneration) com um plano de 10 anos, visando à montagem de computadores inteligentes por meio da utilização do Prolog. Em resposta, os Estados Unidos construíram a Micro electronics and Computer Technology Corporation (MCC) que foi projetada para assegurar a competitividade nacional. Em ambos os casos a Inteligência Artificial fazia parte de um amplo esforço, envolvendo tanto o projeto de chips quanto a pesquisa da interface humana. Atualmente é mais comum usar as teorias existentes como bases em vez de propor teorias inteiramente novas.

### 3.2 ALGORITMOS DE BUSCA

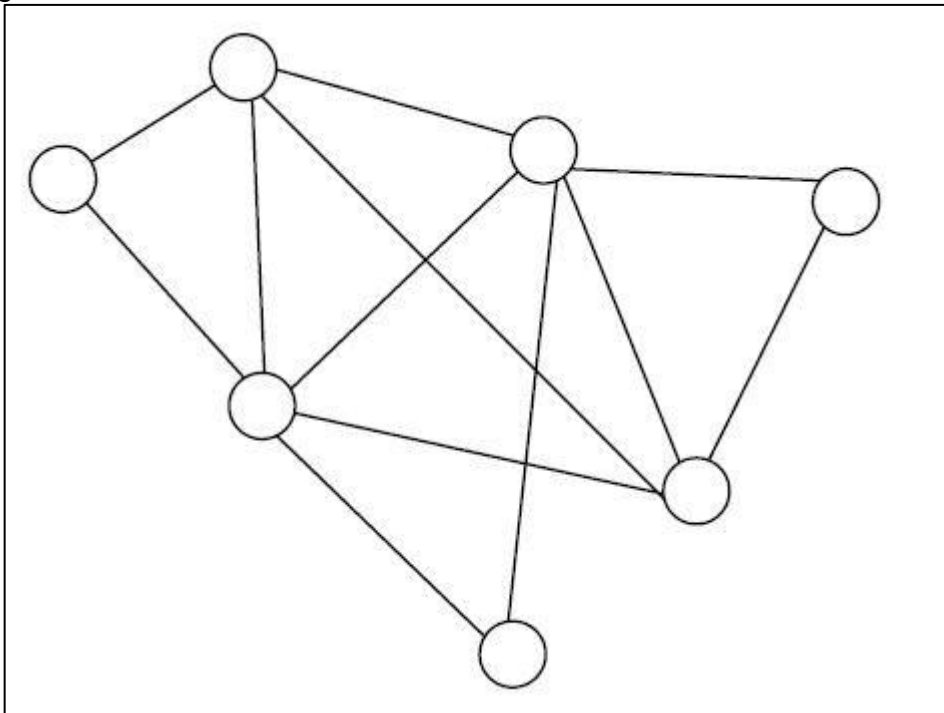
Os métodos de busca em ambientes computacionais são comumente implementados por meio de grafos.



Um grafo consiste em dois tipos diferentes de elementos: os nós que são freqüentemente desenhados como pontos ou círculos em um diagrama, enquanto as suas conexões são representadas através de linhas. (MILINGTON, 2006).

Formalmente, o gráfico é constituído por um conjunto de vértices (nós) e um conjunto de arestas (ligações), em que uma conexão é simplesmente um par ordenado de nós (os nós em cada extremidade da conexão). A Figura 2 mostra a estrutura de um grafo.

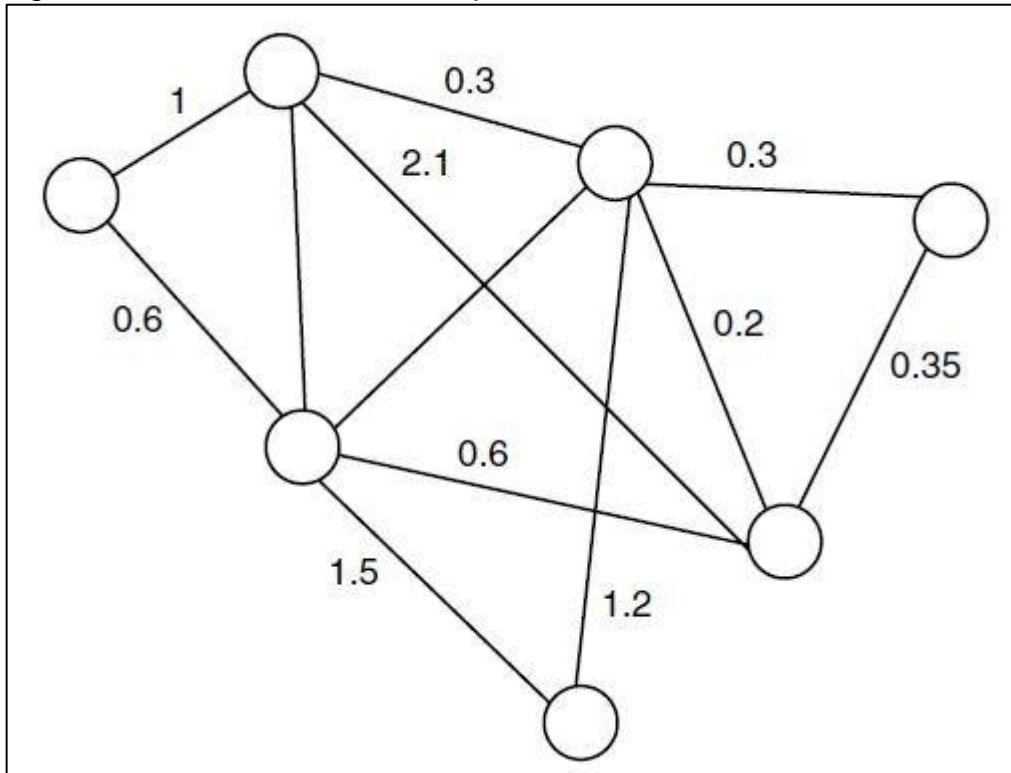
Figura 2 - Estrutura de um Grafo.



Fonte: Millington (2006).

Segundo Millington (2006), um grafo ponderado é composto de nós e conexões, assim como o grafo geral. Além de um par de nós em cada ligação, pode ser adicionado um valor numérico. Em teoria dos grafos isso é chamado de peso, ou mais comumente chamado de custo. A Figura 3 representa um grafo ponderado, onde cada conexão entre seus nós, é marcado com um valor de custo associado.

Figura 3 - Estrutura de um Grafo ponderado.

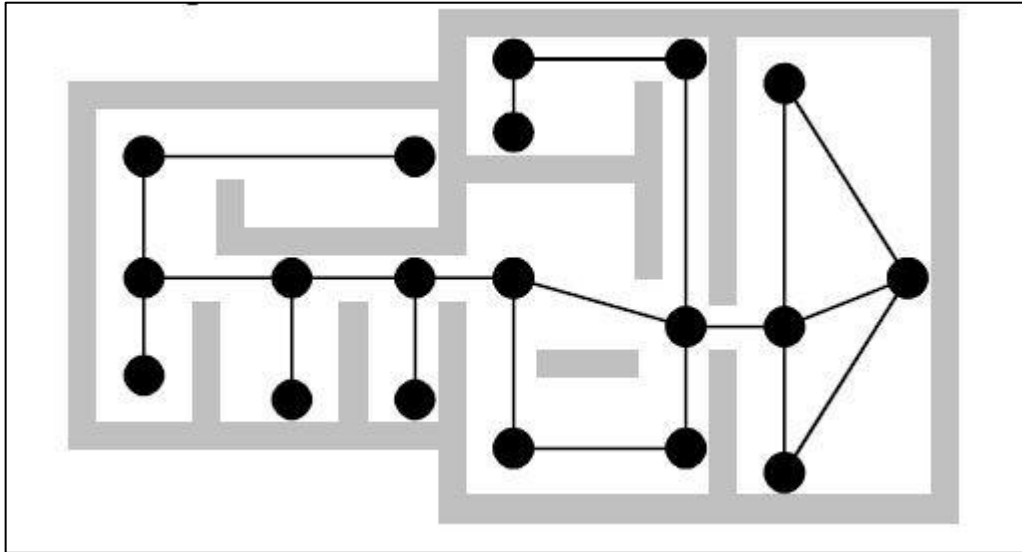


Fonte: Millington (2016).

O uso possível de grafos em robótica são os grafos de navegação. Eles representam uma abstração das posições onde os robôs podem se locomover no ambiente e a conexão (arestas) entre essas posições (pontos), essas conexões (arestas) podem ainda ter um peso associado, aonde representa algum custo entre os pontos que a definem, como por exemplo, a distância entre eles. Esse tipo de grafo é conhecido como grafo euclidiano.

A Figura 4 apresenta um exemplo de um grafo de navegação de um ambiente 2D delimitado por paredes.

Figura 4 - Representação de um grafo de navegação.



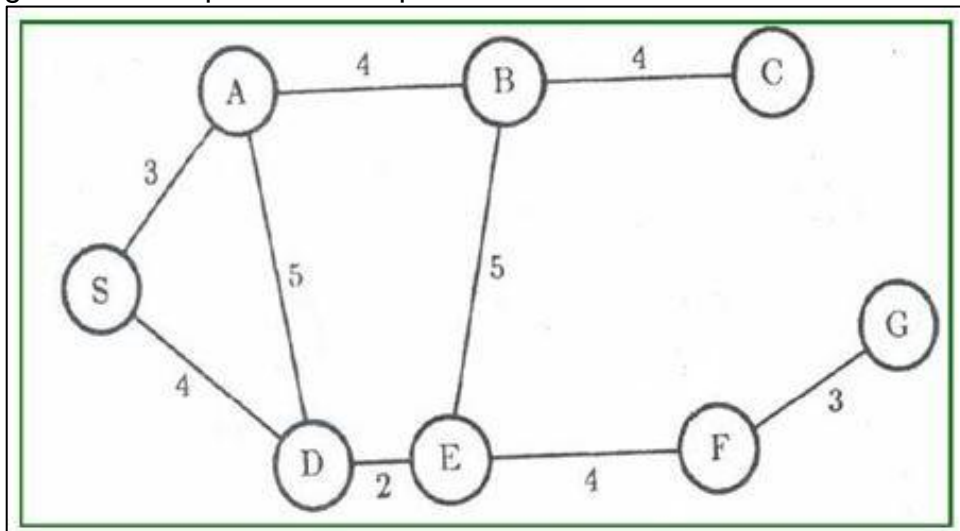
Fonte: Pozzer (2010).

Deve-se observar que as arestas não são os únicos caminhos que o robô pode percorrer. Eles representam caminhos que podem ser computados para encontrar o caminho de menor custo entre dois pontos quaisquer no cenário. Esse tipo de representação de grafo (árvore de busca) é muito utilizado para cálculos de trajetórias em ambientes. Geralmente faz-se uso de uma grade de células, onde cada célula representa regiões do espaço modelado. Assim os grafos de navegação são criados usando o ponto central de cada célula. Os pesos são dados pela distância e ou por algum outro critério. Essa estratégia geralmente produz grafos que podem ser muito grandes, dependendo do tamanho do ambiente. Essas informações são de vital importância para auxiliar o agente robótico no planejamento do melhor caminho entre dois pontos quaisquer (trajetória).

A árvore de busca de um problema é representada computacionalmente por meio de um grafo acíclico no qual cada vértice corresponde a um estado e as arestas indicam transições entre dois estados. A busca pelo objetivo de um problema compreende a navegação pelos vértices da árvore associada a fim de encontrar o(s) vértice(s) cujo(s) estado(s) corresponde(m) ao objetivo de um problema. A seqüência de estados percorridos para alcançar o objetivo de um problema é denominada de solução. Um mesmo problema pode possuir múltiplas soluções. (GOLDSCHMIDT, 2010).

O processo de busca em um problema consiste em encontrar um caminho a partir de um estado ou vértice inicial (S) até um vértice terminal (G) cujo estado corresponda o objetivo do problema em questão. A Figura 5 apresenta um exemplo de um mapa rodoviário fictício onde as cidades são representadas pelos vértices e as arestas indicam estradas entre as cidades. O número associado a cada aresta indica o custo de se percorrer a estrada correspondente. Por exemplo, na Figura 5, o custo da transição do vértice S para o vértice A é igual a três unidades de custo. O problema deste exemplo consiste em encontrar o caminho entre as cidades S e G com menor custo.

Figura 5 - Exemplo de um mapa rodoviário fictício.



Fonte: Goldschmidt (2010).

Russel e Norvig (2004) definem uma busca como um problema que pode ser formalmente definido por quatro componentes:

- Um estado inicial – O estado inicial de um problema corresponde ao ponto de partida para o processo de busca.
- Um conjunto de operadores – Um operador é uma função que recebe como argumento um estado e gera como saída um conjunto de pares ordenados do tipo (ação, sucessor). A ação corresponde a uma das ações válidas para o estado informado e o sucessor indica o próximo estado a ser obtido a partir da aplicação da referida ação.
- Um teste de objetivo – Função que indica se um estado de entrada atende ao objetivo do problema.

- d) Uma função de custo (heurísticas) – Função que atribui um custo numérico a cada caminho definido a partir do estado inicial até o estado corrente. Assim sendo, a formulação de um problema consiste em especificar os componentes indicados, o que, em geral, requer uma boa capacidade de abstração.

Considerando um problema de busca em que se deve achar um melhor caminho para solucioná-lo e esse caminho é desconhecido, deve-se definir uma função heurística, que pode ser considerada uma pesquisa de quantificação de proximidade a um determinado objetivo, tendo está definida de maneira correta, acredita-se encontrar a solução para o problema. Em oposição aos algoritmos que fazem uso de heurísticas e são, por vezes, muito custosos, tem-se os algoritmos de busca não. Os algoritmos de Busca Não Informada (sem informação) são normalmente chamados de algoritmos de Busca Cega. Essa classe de algoritmos não utiliza nenhuma informação sobre o domínio do problema e executa exaustivamente até encontrar uma solução ou retornar uma falha. (RUSSEL; NORVIG, 2004). As seções seguintes trazem os conceitos relacionados a esses tipos de busca.

### 3.3 BUSCA CEGA

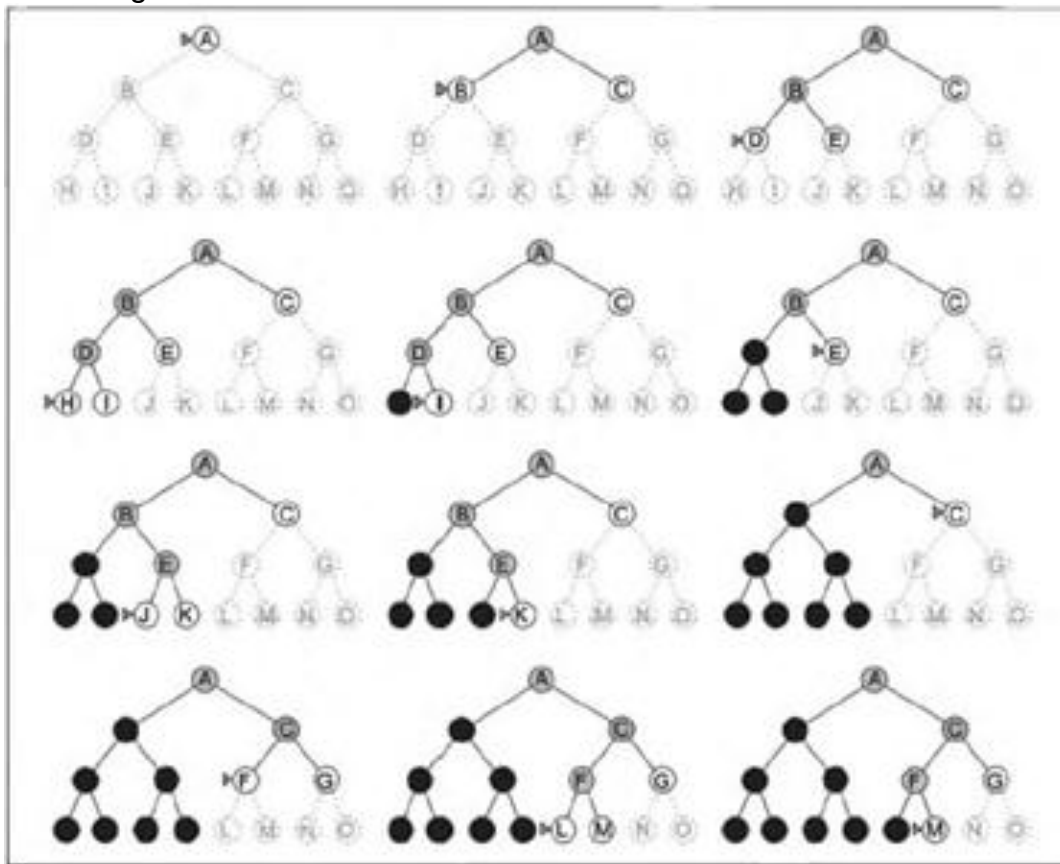
De acordo com Russel e Norvig (2004) estratégias de busca sem informação também conhecida como Busca Cega usam apenas a informação disponível na definição do problema. São estratégias de busca baseadas em tentativas de solução por força bruta onde o único conhecimento que pode ser aplicado ao problema para determinar o próximo passo em direção a uma solução é dado por uma função que estabelece de modo sistemático como os nós devem ser percorridos. Estas estratégias podem encontrar uma solução para o problema simplesmente gerando novos estados e testando-os para verificar se o objetivo foi atingido.

Dentre estas estratégias as mais importantes são conhecidas como busca em profundidade, busca em largura e busca de custo uniforme. As próximas seções descrevem cada tipo de busca não-informada, apresentando características que as diferenciam.

### 3.3.1 Busca em Profundidade

De acordo com Russel e Norvig (2004) a Busca em Profundidade sempre se expande o nó mais profundo na borda atual da árvore de busca, que prossegue imediatamente até o nível mais profundo da árvore de busca, onde os nós não têm sucessores. À medida que esses nós são expandidos eles são retirados da borda, e então sua busca retorna ao próximo nó mais raso que ainda tem sucessores inexplorados. A Figura 6 mostra o progresso da Busca em Profundidade em uma árvore binária.

Figura 6 - Progresso da Busca em Profundidade.



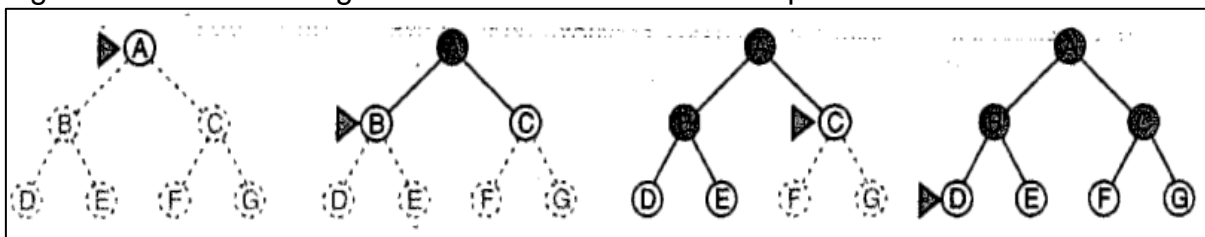
Fonte: MINISTÉRIO DA CIÊNCIA E TECNOLOGIA. Instituto Nacional de Pesquisas Espaciais - Desastres Naturais e Geotecnologias: Conceitos básicos. INPE Santa Maria, 2007.

### 3.3.2 Busca em Largura

De acordo com Russel e Norvig (2004) na busca em largura (também conhecida como busca em extensão), o nó raiz é o primeiro nó a ser expandido. Após gerar todos os nós sucessores possíveis, a partir do estado pertencente ao nó

raiz, esses são expandidos. Somente após a expansão de todos os nós pertencentes ao mesmo nível de profundidade na árvore de busca é que os nós do próximo nível serão expandidos. Um exemplo deste processo pode ser visto na Figura 7.

Figura 7 - Busca em largura em uma árvore binária simples.



Fonte: Russel, Norvig (2004).

### 3.3.3 Busca de Custo Uniforme

Para Russel e Norvig(2004), busca de custo uniforme expande o nó não-expandido que tenha o caminho de custo mais baixo, não se importando com número de passos que o caminho tem, mas apenas com seu custo total. Sua estratégia consiste, portanto, em expandir o nó com o caminho de custo mais baixo. Caso todos os custos sejam iguais, essa busca será idêntica à busca em largura. Por isso ela se paralisara em um laço de repetição infinito se expandir um nó que tenha uma ação de custo zero levando de volta ao mesmo estado.

Segundo Russel e Norvig (2004), Custo Uniforme é orientado por custo de caminhos em vez de profundidade assim sua complexidade não pode ser caracterizada com facilidade em termo de profundidade da arvore de busca e número de nós.

## 3.4 BUSCA HEURÍSTICA

Existem diversas estratégias para percorrer a árvore de busca de um problema. Em muitas delas, faz-se necessária a utilização de uma boa função heurística, sobretudo quando a árvore de busca é muito grande ou mesmo possui um número infinito de caminhos (RUSSEL; NORVIG, 2004).

Goldschmidt (2010) afirma que a palavra heurística está relacionada ao verbo grego original, “eu risco”, que significa “eu descobro”. É uma junção de intuição, lógica e conhecimento prévio sobre determinado assunto, formalizado em regras e/ou métodos, que podem ajudar na invenção, descoberta e solução de problemas. Em outras palavras, uma heurística é um conhecimento prévio sobre um determinado problema que auxilia na busca por soluções, simplificando a árvore de busca associada. Na procura por soluções em árvores de busca, heurísticas são formalizadas como regras que determinam a escolha dos ramos que possuem a maior probabilidade de levarem a uma solução aceitável para o problema.

De acordo com Russel e Norvig (2004) a Busca Heurística (ou Busca com Informação) são utiliza conhecimento específico sobre o problema de encontrar soluções de forma mais eficiente do que Busca Cega. Essas estratégias diferem das estratégias das buscas cegas por acrescentarem uma informação a mais na determinação da ordem de expansão dos nós durante o processo de busca. Esta informação é baseada numa função de avaliação e consiste em uma forma de mensurar a probabilidade de um nó convergir para uma solução considerando em seu estado corrente.

Nem sempre encontrar uma solução para um problema é suficiente. É muito comum a necessidade de busca pelo menor (melhor) caminho até o objetivo, onde passos adicionais podem levar a custos excessivos na identificação da solução. Uma heurística que permite encontrar o caminho de menor custo (ou o melhor caminho) até um objetivo (sempre que ele existir) é chamada de heurística admissível. Ou seja, uma heurística admissível não deve nunca superestimar o custo de se alcançar o objetivo.

Heurísticas admissíveis são otimistas por natureza, pois sempre imaginam que o custo para resolução de um problema seja menor do que ele é na realidade (RUSSELL; NORVIG 2004).



## 4 ROBÓTICA

Robôs são agentes inteligentes físicos que executam grandes tarefas manipulando o mundo físico. Sua arquitetura é composta por equipamentos como pernas, rodas articulações e garras, tendo um único propósito poder exercer força física sobre o ambiente. São equipados com uma diversidade de sensores que permitem perceber o ambiente, como: câmera, sensores de ultrassônico, giroscópios e aceleração. (RUSSEL; NORVIG, 2004).

De acordo com Stairs e Reynolds (2006), a maioria dos robôs se enquadra em três tipos de categorias: manipuladores, móveis ou híbridos.

- a) Robôs manipuladores são classificados como braços de robôs, que estão fisicamente fixos em seu local de trabalho como, por exemplo, robôs industriais. O movimento do manipulador permite que esses robôs coloquem seus efetadores em qualquer posição dentro do local de trabalho, os manipuladores são os mais encontrados no ramo industrial quando se trata de robôs.
- b) Os Robôs móveis são máquina que se deslocam pelo ambiente usando rodas, pernas ou algum mecanismo parecido. Foram projetados para entrega de alimentos em hospitais, movimentação de contêineres em docas de carga e tarefas semelhantes. Esses tipos de robôs são classificados em Veículos Terrestres, Veículos Aéreos ou Autônomos Subaquáticos. Os terrestres são veículos não tripulados que realizam sua navegação autônoma sem condutor em auto-estradas, os aéreos também são classificados como não tripulados, que são utilizados para vigilância, pulverização de lavouras, operações militares, já veículos autônomos aquáticos são utilizados para explorações no fundo do mar.
- c) Também existem os robôs híbridos que são robôs móveis equipados com manipuladores.

Segundo Stairs e Reynolds (2006) o campo da robótica não se limita apenas nesses três tipos de robôs citados, existem também os robôs protéticos que são

membros artificiais para os seres humanos, tanto como os ambientes inteligentes que são casa equipadas com sensores e efetadores conhecido com automação.

Para Stairs e Reynolds (2006) o sistema visual juntamente com hardware e software permite aos computadores a capturar, armazenar e manipular dados visuais que são utilizados juntamente com a robótica para dar a visão aos robôs. Tais recursos permitem aumentar sua capacidade facilitando a tomada de decisões com base na entrada visual que pode ser usada tanto para um sistema biométrico quanto para análise de peça defeituosa em linha de montagem.

A partir da Guerra do Golfo no ano de 1991, a força armada dos Estados Unidos, utilizou os robôs na área de logística e planejamento sendo capaz de automatizar com a programação a execução do transporte das tropas. Com as técnicas de planejamento de IA permitiram a geração em apenas algumas horas de um plano que exigiria semana com outros métodos. (STAIRS; REYNOLDS, 2006).

A empresa Kiva Systems (2014), vem aplicando um conceito de “Inteligência Distribuída” em armazéns ou pequenas empresas com grande circulação de produtos, com ajuda de dois especialistas no campo da engenharia e robótica, sendo professor Peter Wurman e Raffaello D’Andrea.

Objetivo da empresa é desenvolver um sistema de automação completo, melhorando simultaneamente a velocidade, precisão e flexibilidade, da tarefa de gerenciamento que envolve o enchimento ou esvaziamento de certos compartimentos.

Esse sistema foi implantado na empresa Amazon conforme a Figura 8, tendo uma equipe de robôs que de forma cooperativa permite o transporte de prateleiras a diferentes locais.

Figura 8 - Amazon implementa robôs Kiva para temporada de fim de ano.



Fonte: Exame (2014)

#### 4.1 PLATAFORMA ARDUINO

De acordo com Evans, Noble, Hochenbaum (2013) o Arduino teve seu início no Interaction Design Institute na Itália, no ano de 2005 pelo professor Massimo Banzi que procurava por um meio mais barato e prático para que os seus estudantes de design pudessem trabalhar com a tecnologia. Ele discutiu seu problema com David Cuartielles, um pesquisador visitante da Universidade de Malmö, na Suécia, que estava procurando uma solução semelhante e assim Arduino foi desenvolvido. (Evans, Noble e Hochenbaum, 2013).

Evans, Noble, Hochenbaum (2005) dizem que os produtos existentes no mercado eram de custo alto e difícil de trabalhar com eles, desse modo que Banzi e Cuartielles decidiram desenvolver seu próprio micro controlador facilitando seus estudos de arte e design com seus projetos, trazendo uma exigência de custo menor e mais fácil de trabalhar.

Segundo Evans, Noble, Hochenbaum (2013), David Cuartielles foi quem elaborou o desenho da placa e o aluno de Massimo, David Mellis foi quem desenvolveu a programação do software para a execução da placa. Para ajudar no desenvolvimento da placa Massimo contratou um engenheiro local, Gianluca Martino, que também trabalhou no Design Institut.

McRoberts (2011) diz que a fama não parava de aumentar e com isso cada vez mais as pessoas perceberam o potencial incrível do projeto de código livre para a criação de projetos importantes, como forma de aprendizagem.

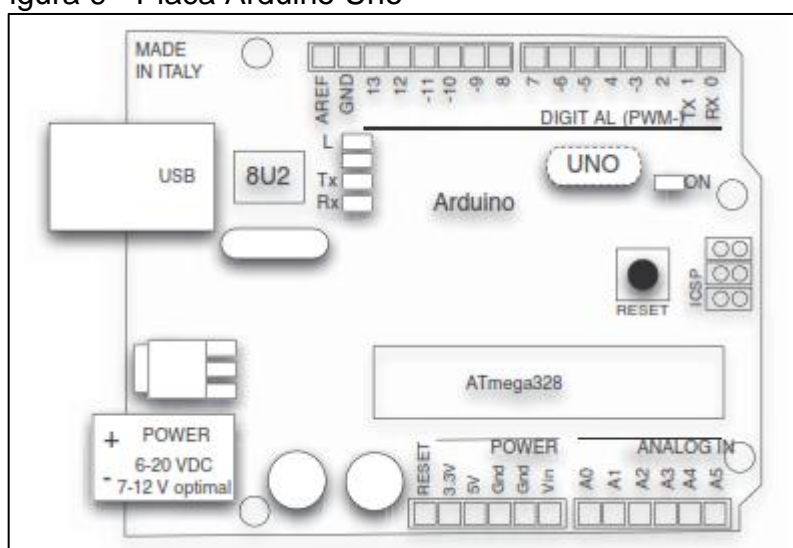
### 4.1.1 Hardware

De acordo com Evans (2005) existem no mercado diversos modelos de Arduino. Sua base trabalhar com um microprocessador com 8 bits da Atmel. A primeira placa Arduino rodava um Atmega8 tendo um clock de velocidade de 16mhz com sua memória flash de 8 Kb. (Evans, Noble, Hochenbaum, 2013).

#### 4.1.1.1 *Arduino Uno*

Segundo Evans, Noble e Hochenbaum (2013), a plataforma Arduino Uno (Figura 9) é a mais comumente utilizada. Ela possui uma compatibilidade de pinos com as suas versões anteriores, incluindo o Duemilanove e seu antecessor Dievimila. Essa plataforma é uma boa opção para multiuso sendo a melhor aposta para uma placa de partida, possuindo uma tensão integrada de 3.3v; ajudando na estabilidade e proteção.

Figura 9 - Placa Arduino Uno

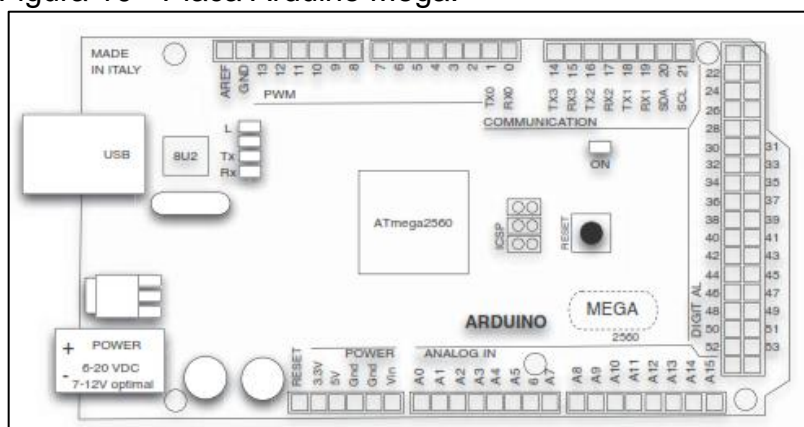


Fonte: Evans, Nobre, Hochenbaum (2013)

#### 4.1.1.2 *Arduino Mega*

De acordo com Evans, Noble e Hochenbaum (2013) a plataforma Mega (Figura 10) é a irmã mais velha da família do Arduino. A Mega usa um microprocessador com uma superfície mais elaborada com versão Atmega2560, possuindo uma memória flash de 256kb, superior aos 128kb da sua versão anterior.

Figura 10 - Placa Arduino Mega.



Fonte: Evans, Noble, Hochenbaum (2013)

#### 4.1.2 Sensores e módulos

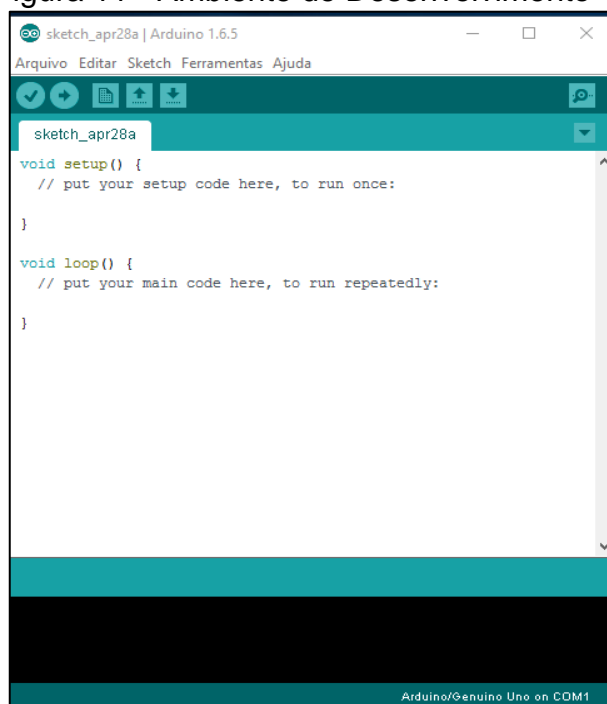
Alguns dos principais sensores e módulos usados em projetos de robótica são descritos a seguir:

- Sensor RFID: foi especialmente desenvolvido para aplicações de leitura com identificação de pessoas até mesmo produto tanto como por etiquetas quanto crachás, aplicado no produto, nas etiquetas pode servir para descrever ou identificar um produto. Cada etiqueta (tag) do leitor tem sua própria identificação (UID).
- Módulo segue linha: é composto por um emissor de luz infravermelho juntamente com um foto-transistor que, trabalhando junto, permite identificar linhas/objetos brancos, pretos e até mesmo cores intermediárias, dependendo do ajuste de sensibilidade.
- Módulo de Sensor Ultrassônico: tem uma capacidade de medir distâncias de 2 cm a 4 metros com uma ótima precisão, ele tem acoplado junto um circuito com seu ponto emissor e receptor.

#### 4.1.3 Ambiente de Trabalho

Segundo Evans, Noble e Hochenbaum (2013), o Arduino é uma placa com alguns componentes eletrônicos, cuja programação é realizada com uma IDE próprio (Figura 11), Assim como o hardware, o próprio software para o Arduino é um código aberto e pode ser baixado gratuitamente na internet.

Figura 11 - Ambiente de Desenvolvimento



Fonte: Elaborada pelo autor.

#### 4.1.3.1 Linguagem C

De acordo com Manzano (2010), a linguagem C foi projetada em 1972 no laboratório de uma empresa chamada Bell Telephonelab, por Dennis M. Ritchie juntamente com Brian W. Kernighan que a desenvolveram com objetivo de poder utilizá-la na codificação de um segundo sistema operacional UNIX. A linguagem C deriva da ALGOL 68 que trouxe evolução da BCPL. Com o tempo a linguagem C foi reconhecida como linguagem C ++, que seria linguagem C com suporte à Programação Orientada a Objeto.

De acordo com Manzano (2010), a linguagem C permite a integração direta entre um nível alto e baixo.

Ainda Manzano (2010) destaca que o passou por várias mudanças nos anos de 1983 e 1989, com o instituto de normas técnicas ANSI iniciando um trabalho visando padronizar a linguagem de forma que ela pudesse realmente ser portátil entre as diversas plataformas. A partir disso a linguagem C passou a ser utilizada em diversas plataformas de computadores e já no final da década dos anos 90, a linguagem C passou por nova revisão de acordo com as normas ISO/IEC

9899:1999. No ano de 2007 uma revisão nova foi iniciada, conhecida como C1X, mas, no entanto, esse trabalho não foi concluído.

#### 4.1.3.1.1. Estrutura de Programa em C

Segundo Manzano (2010) um programa em C consiste de várias funções, que permitem construir programas modulares e estruturados. A Figura 12 mostra a estrutura inicial de um programa em C;

Figura 12 - Estrutura de programa C.

```
1 int main ( void)
2 {
3     return 0;
4 }
```

Fonte: Elaborada pelo autor.

## 5 TRABALHOS CORRELATOS

O trabalho desenvolvido por Medeiros e Silva (2010) apresenta resultados da aplicação de um algoritmo Dijkstra ao planejamento de trajetórias para Veículos Aéreos Não Tripulados (VANTs). Os ambientes de navegação são representados por conjuntos de grafos de visibilidade construídos através de elevações de terreno destes ambientes. As trajetórias planejadas por este algoritmo são livres de colisão e dinamicamente viáveis. O algoritmo possui uma heurística de busca local para verificar situações de colisão que é baseada no modelo digital de elevações do ambiente de navegação.

Os resultados obtidos comprovam a eficiência do algoritmo de Dijkstra para planejar trajetórias que são dinamicamente viáveis e livres de situações de colisão com obstáculos do ambiente de navegação.

O trabalho de pesquisa feito por Souza et al. (2004) teve como objetivo construir um ambiente virtual de navegação de robôs, simulando o ambiente real, com a inserção de obstáculos que devem ser detectados e evitados pelo robô em sua trajetória e, além do mais, atingirem o ponto pré-estabelecido percorrendo o menor caminho. Implementou-se um software de forma a representar um ambiente real, formando, uma área quadriculada na forma de células, sobre uma das quais é inserido o “robô virtual” e indicado o seu destino. Objetos podem ser inseridos no percurso, caracterizados pelas mudanças das cores de fundos das células escolhidas. Para a consecução do objetivo deste trabalho fez-se uso da teoria dos grafos utilizando o

Algoritmo de Dijkstra que permite identificar todos os caminhos mais curtos, a partir de um único vértice de origem, bem como o desvio dos obstáculos interpostos no caminho. Os resultados das simulações podem ser considerados bons, visto que em todas elas o “robô virtual” percorreu a distância entre dois pontos pré-estabelecidos utilizando sempre o menor caminho, contornando os obstáculos interpostos em sua trajetória.

O trabalho desenvolvido por Campaneli e Mestria (2013) consistiu em encontrar os caminhos menores possíveis, a serem percorridos por uma cadeira de roda robótica, entre quaisquer dois pontos acessíveis do ambiente físico, utilizando um algoritmo de otimização. Por meio do uso do Algoritmo de Dijkstra foi possível traçar as trajetórias entre quaisquer dois pontos acessíveis escolhidos pelo usuário



de cadeira de rodas, validando o método utilizado. Após definir o caminho, o algoritmo informa as características do percurso. Estes caminhos permitem encontrar trajetórias para as cadeiras de rodas robótica, que serão acionados através da interface de usuário e comunicando sua interface com um sistema embarcado. Assim, o usuário de cadeira de roda poderá escolher dentre os pontos definidos pela interface de usuário e locomover-se neste ambiente, considerando a melhor rota encontrada.

## 6 METODOLOGIA

O objetivo desse trabalho foi desenvolver um Robô que possa calcular sua trajetória, simulando um ambiente de transporte de cargas, utilizando técnicas de inteligência Artificial, especificamente algoritmo de busca (no caso, o algoritmo de custo uniforme).

Para alcançar esse objetivo, a pesquisa foi dividida em etapas descritas nas seções seguintes.

### 6.1 REVISÃO BIBLIOGRÁFICA

Na primeira etapa foi elaborada a revisão bibliográfica que teve como meta buscar os embasamentos teóricos necessário em livros, artigos e sites.

Nos tópicos relacionados à Inteligência Artificial foram abordados aos conceitos pertinentes à área, a história e os primeiros trabalhos realizados em IA.

Também foram definidos os problemas de busca e principais algoritmos associados.

O capítulo sobre robótica abordou o histórico da área, hardwares associados ao desenvolvimento de agentes robóticas, de modo específico a plataforma Arduino (placas UNO e MEGA). Considerando que ambiente de programação da plataforma Arduino utiliza a linguagem de programação C, foi a linguagem base para programar o software de controle de agente robótico a ser desenvolvido nesta investigação.

### 6.2 CONSTRUÇÃO DO ROBÔ MÓVEL

O robô móvel desenvolvido na presente pesquisa contou com alguns periféricos, associados ao Arduino utilizados neste trabalho para locomoção do agente na rota definida. Ele possui, portanto, rodas acopladas a motores controlados pelo software.

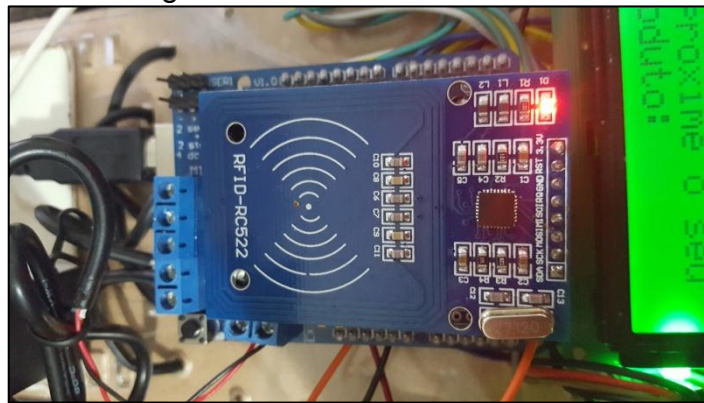
Embora seja possível que o robô disponha de um sistema de aquisição de imagens (via câmeras), o mesmo não foi utilizado neste trabalho, pois os métodos de visão computacional necessários para isso são relativamente complexos, fugindo ao escopo dessa pesquisa. Dessa forma o robô apenas calcula sua trajetória entre os pontos desejados, evitando os obstáculos, executando-a em seguida, num

ambiente previamente conhecido. O ambiente refere-se a uma simulação de um armazém e a tarefa do robô é transportar alguns objetos (produtos) neste local.

Para modelar o robô, o seu programa de controle foi implementado em um software de baixo nível, com a placa Arduino Uno. Para isso foi utilizada a IDE que é disponibilizada pela própria plataforma Arduino.

O robô utiliza diversos sensores da placa, como sensor RFID que identifica onde o “produto” será alocado no “armazém” conforme mostra a Figura 13.

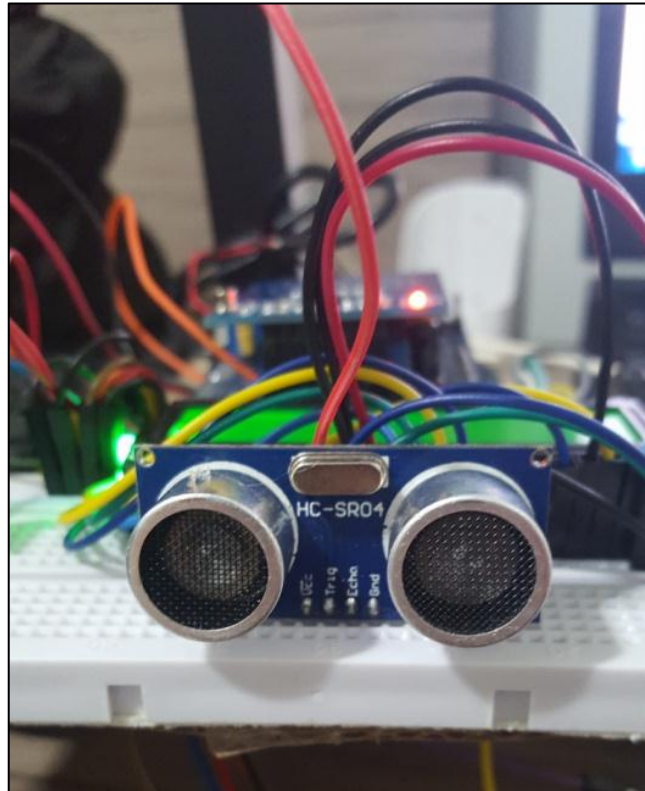
Figura 13 - Imagem ilustrativa do Sensor RFID.



Fonte: Elaborada pelo autor.

Também foi utilizado sensor ultrassônico conforme ilustra a Figura 14, assim o sensor emite uma onda sonora ao encontrar um obstáculo no caminho, permitindo que o agente se mova na direção correta.

Figura 14- Imagem ilustrativa do Sensor Ultrassônico



Fonte:Elaborada pelo autor.

Outro componente que foi utilizado no robô é o chassi, conforme mostra a Figura 15. Esse equipamento faz com que o robô possa se locomover no ambiente.

Figura 15- Imagem ilustrativa do Chassi

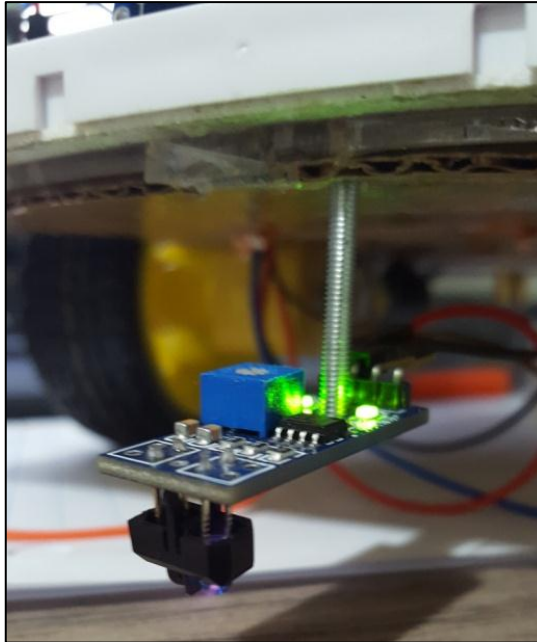


Fonte: Elaborada pelo Autor

Um módulo de sensor (modulo segue linha) é utilizado pelo robô, conforme Figura 16. Este módulo possui um emissor de luz infravermelho juntamente com um

foto-transistor que permite identificar as linhas e com esse comando que direciona o chassi do robô até o caminho desejado.

Figura 16– Modulo Segue Linha

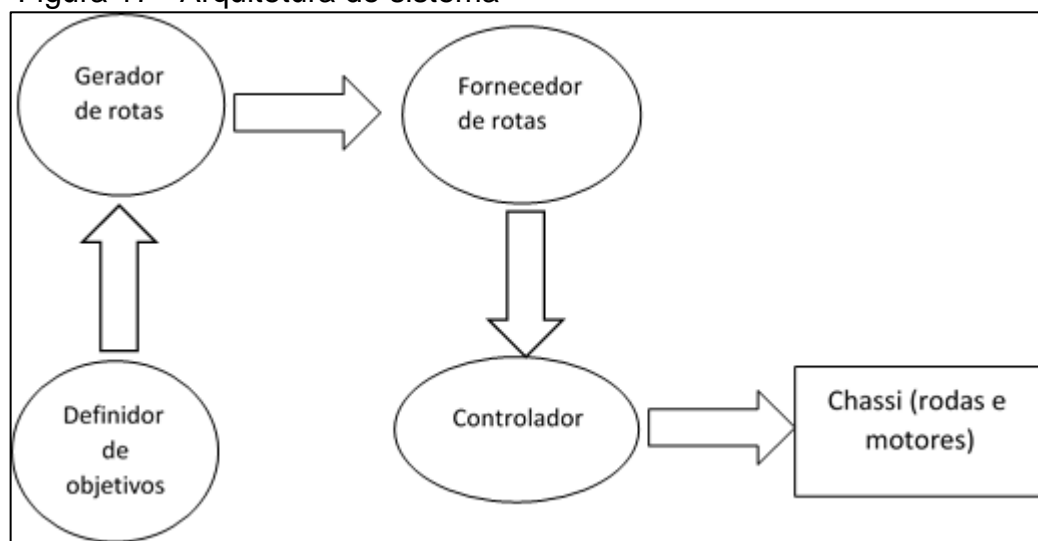


Fonte:Elaborada pelo autor.

### 6.3 ARQUITETURA DO SISTEMA E DESCRIÇÃO DO AMBIENTE

Um diagrama da arquitetura proposta pode ser visto na Figura 17. Os módulos implementados em software estão representados pelos círculos. Os retângulos ilustram os dispositivos físicos do robô que interagem com o ambiente.

Figura 17– Arquitetura do sistema



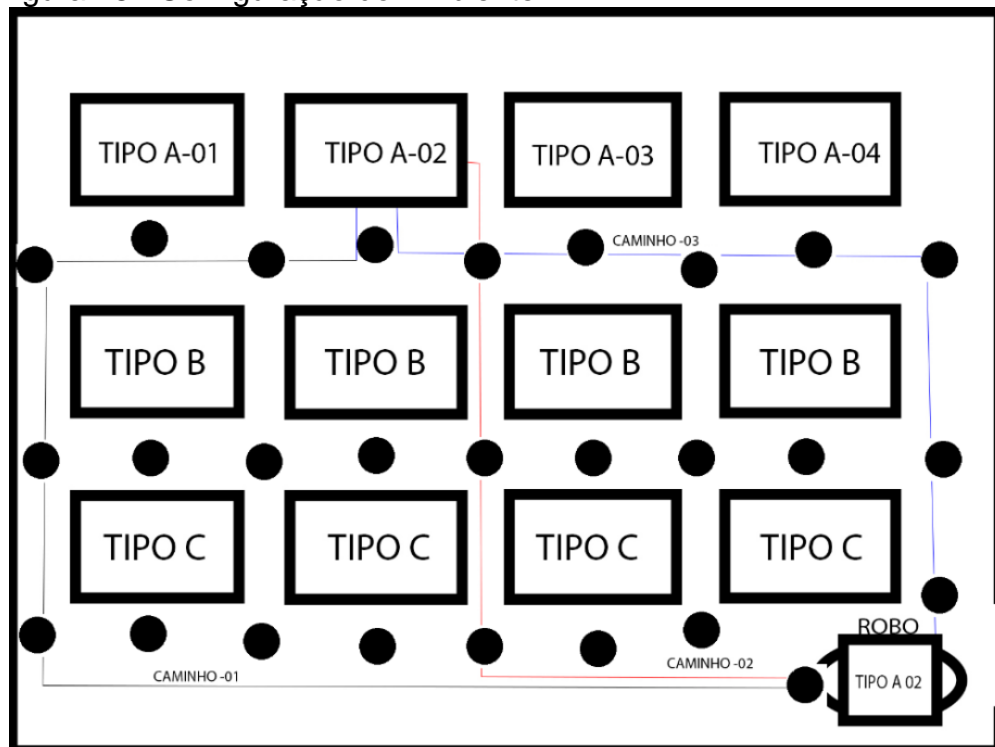
Fonte: Elaborada pelo autor.

A arquitetura proposta é formada pelos seguintes elementos:

- a) Definidor de objetivos: são responsáveis por definir quais são os pontos de origem e destino que deverão ser utilizados no processo de geração de rotas.
- b) Gerador de rotas: faz a geração da rota a ser seguida pelo robô. É importante destacar que embora a trajetória seja gerada completa, isto é, da origem até o destino (objetivo). O robô poderá utilizar o sensor ultrassônico para corrigir seus movimentos caso detecte algum obstáculo que o impeça de cumprir a rota.
- c) Fornecedor de Trajetória: gera a referência para o controlador, a partir da trajetória gerada, limitando os movimentos a trajetória definida pelo gerador de rotas.
- d) Controlador: faz o controle do robô propriamente dito do chassi garantido o cumprimento da trajetória, assim como permanência do robô em um ponto fixo ao final da trajetória;

Todo esse processo de execução de rotas é realizado num ambiente que simula um armazém. A Figura 18 mostra a configuração do ambiente

Figura 18– Configuração do Ambiente



Fonte: Elaborada pelo autor.

Na Figura 18 os pontos pretos indicam posições para as quais o robô pode se movimentar. A idéia é que o agente faça movimentos em linha reta entre os pontos, conforme trajetória calculada, evitando, assim, os obstáculos entre sua origem e destino. Os retângulos (tipo B, tipo C) representam obstáculo (como se fossem pilhas de produtos num armazém). Os demais retângulos (tipo A-01, tipo A-02, tipo A-03 e tipo A-04) representam os locais onde os produtos carregados pelo robô devem ser armazenados. Por exemplo, conforme mostra a Figura 16, se o robô estiver num determinado ponto (origem) com um produto do tipo A-02 ele deverá transportá-lo, até o local (objetivo) correspondente. Podem-se ver na imagem três exemplos de possíveis rotas a ser seguidas pelo robô (caminhos: 01, 02 e 03), entretanto, o caminho será definido pelo algoritmo de custo uniforme, traçando a rota entre os pontos (menor distância), de modo a garantir o melhor caminho (essa é a principal característica desse algoritmo e foi o motivo da sua escolha para implementação).

#### 6.4 AVALIAÇÃO DO SISTEMA

Para testar e validar o método proposto foram executados alguns testes envolvendo a definição de pontos de origem e destino e observados o funcionamento do robô. O intuito foi verificar se as rotas definidas para o robô coincidem, de fato, com as melhores rotas teóricas (previamente calculadas). Também foi avaliado o deslocamento físico, ou seja, se o mesmo consegue cumprir a rota definida evitando os obstáculos.

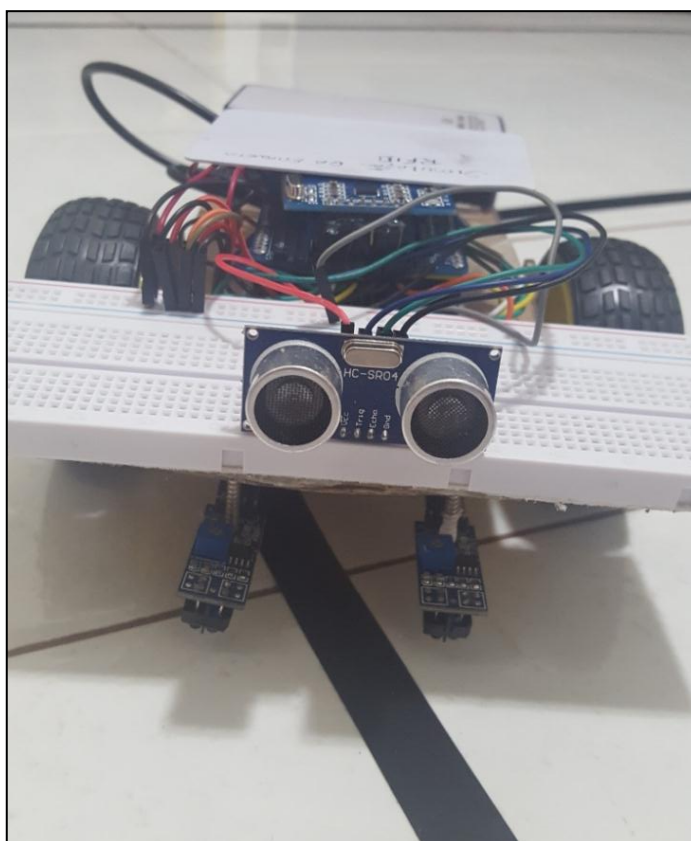
## 7 RESULTADOS

Esta seção descreve os resultados desta pesquisa que correspondem ao desenvolvimento do protótipo (robô).

### 7.1 ARQUITETURA DO ROBÔ

Nesta seção será apresentada a arquitetura do robô como mostra a Figura 19, conforme o objetivo principal deste trabalho.

Figura 19– Robô Móvel.



Fonte: Elaborada pelo autor.

#### 7.1.1 Elementos da Estrutura do Robô

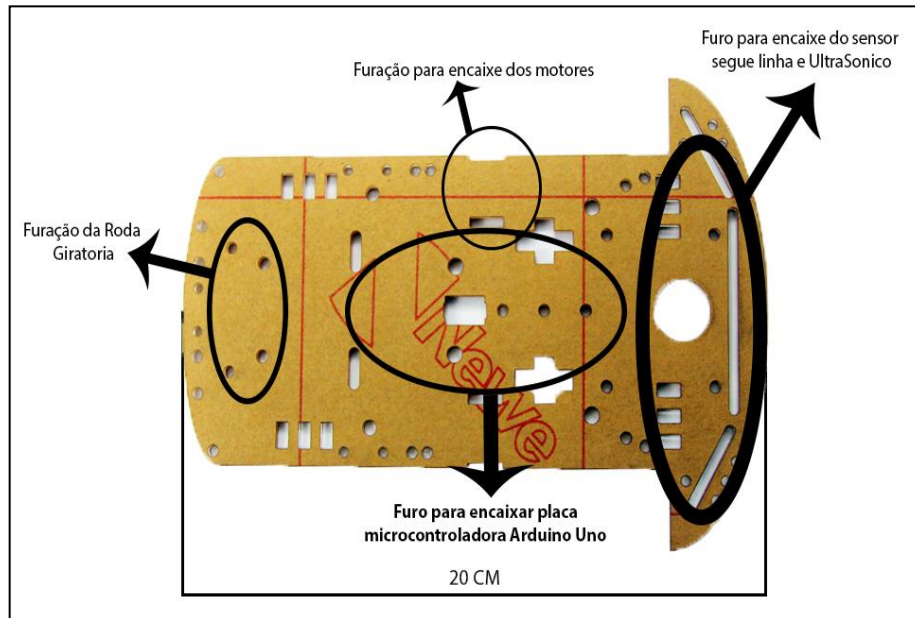
Serão descritos os componentes utilizados na estrutura do robô.

##### 7.1.1.1 Chassi do Robô (Base)



O chassi foi feito em acrílico conforme mostra a Figura 20. Ele é base para fixar eixos, motores, placas como Arduino e sensores que foram usados na montagem do robô.

Figura 20– Chassi do Robô Móvel.

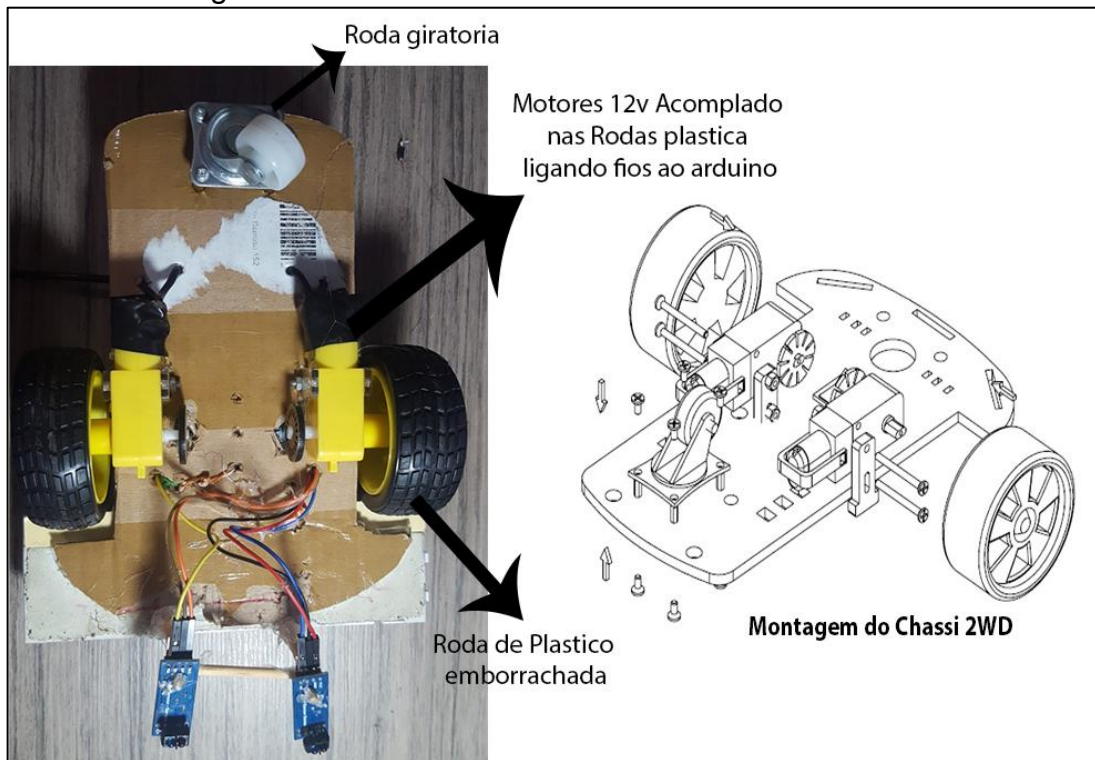


Fonte: Elaborada pelo autor.

#### 7.1.1.2 Rodas e Motores

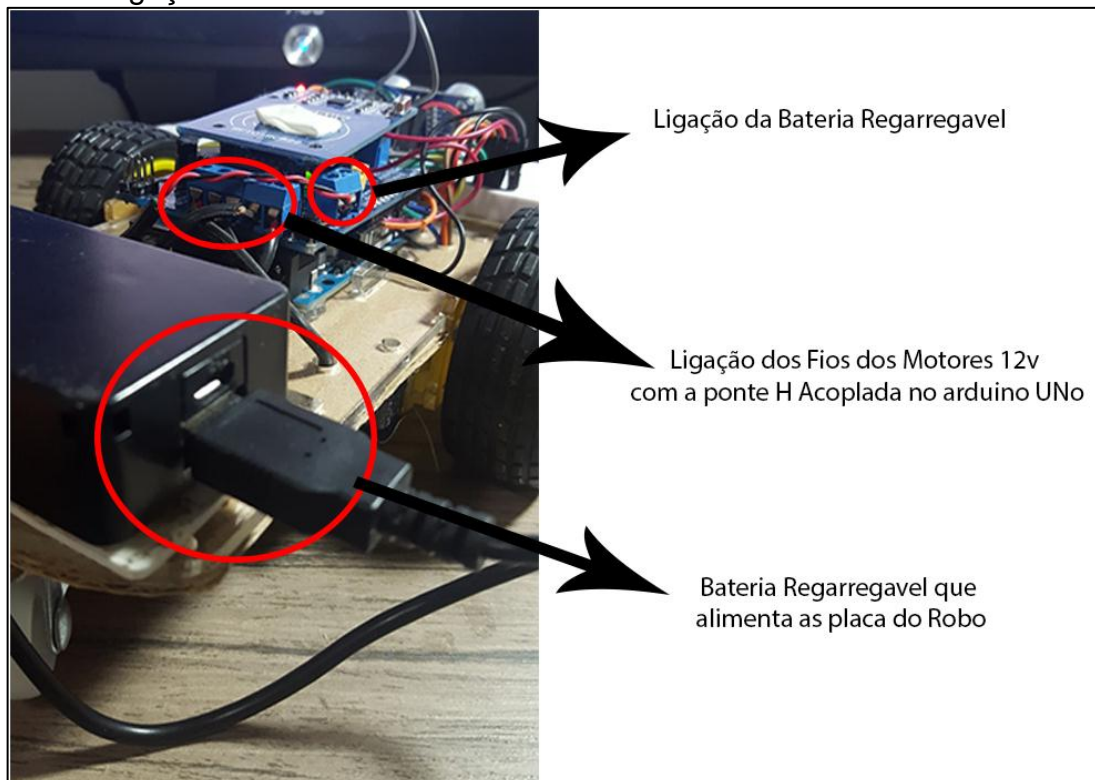
O robô é equipado com duas rodas de tração e uma roda rotativa fazendo que o robô vire para os lados com facilidade. É composto por dois motores de 12 volts que permitem a rotação das rodas do robô. A placa Arduino está ligada à ponte H com uma bateria recarregável colada no chassi conforme mostra nas Figuras 21 e 22.

Figura 21 – Montagem do Chassi Rodas e Motores



Fonte: Elaborada pelo autor.

Figura 22–Ligação dos Motores

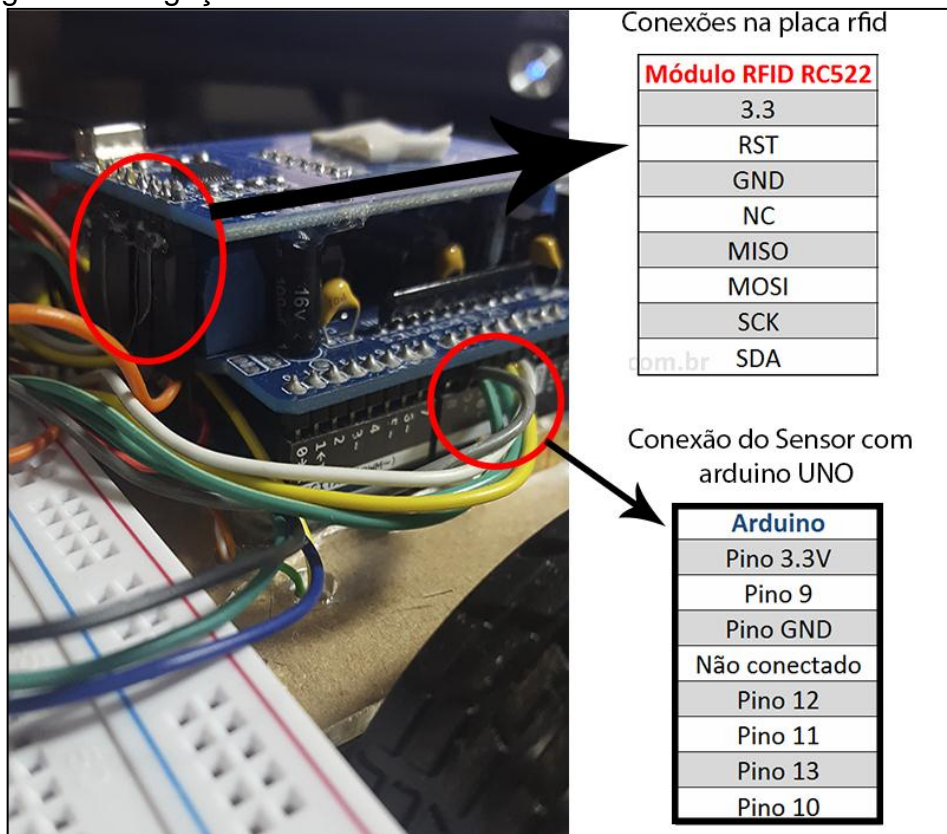


Fonte: Elaborada pelo autor.

### 7.1.1.3 Arduino Uno e Sensores

O Arduino Uno e sensores são os principais elementos do robô sendo responsáveis por seus movimentos. Os sensores, além de ajudar na movimentação, também podem identificar o objetivo. O sensor RFID permite identificar o objetivo, fornecendo dados para que o algoritmo de busca possa calcular o melhor caminho. Este sensor é alimentado por 3.3v que é ligado no Arduino. A ligação dos pinos, SDA, SCK, MOSI, MISO, GND e do pino RST nas portas digitais do Arduino é mostrada na Figura 23.

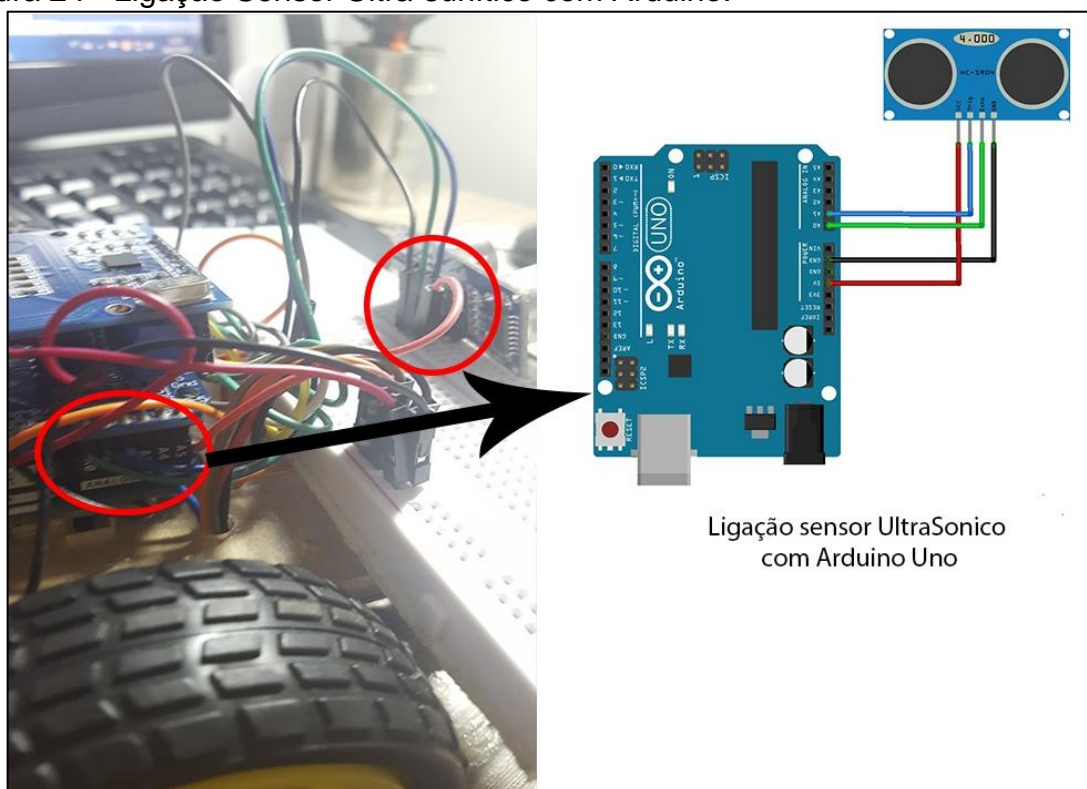
Figura 23 - Ligações do Sensor RFID



Fonte: Elaborada pelo autor.

O Robô também é composto por um Sensor Ultrassônico que emite uma onda sonora, verificando se tem um obstáculo no caminho. A Figura 24 mostra a ligação dos pinos VCC, GND, TRIG, ECHO na placa Arduino.

Figura 24 - Ligação Sensor Ultra-sônico com Arduino.

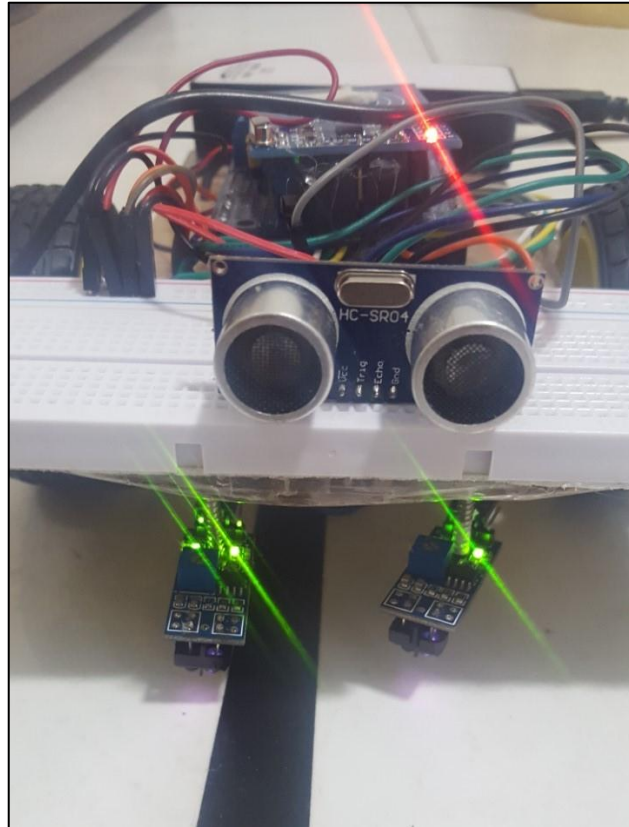


Fonte: Elaborada pelo autor.

Outros sensores que compõem na estrutura do Robô são os sensores segue linha, que tem ligação com Arduino Uno a partir dos pinos de alimentação VCC, GND, DO. O pino DO está ligado na porta analógica do Arduino, podendo transmitir os sinais a partir do seu emissor de luz infravermelho e com um foto-transistor que permite identificar as linhas da trajetória, direcionando-o ao objetivo, conforme demonstra a Figura 25.



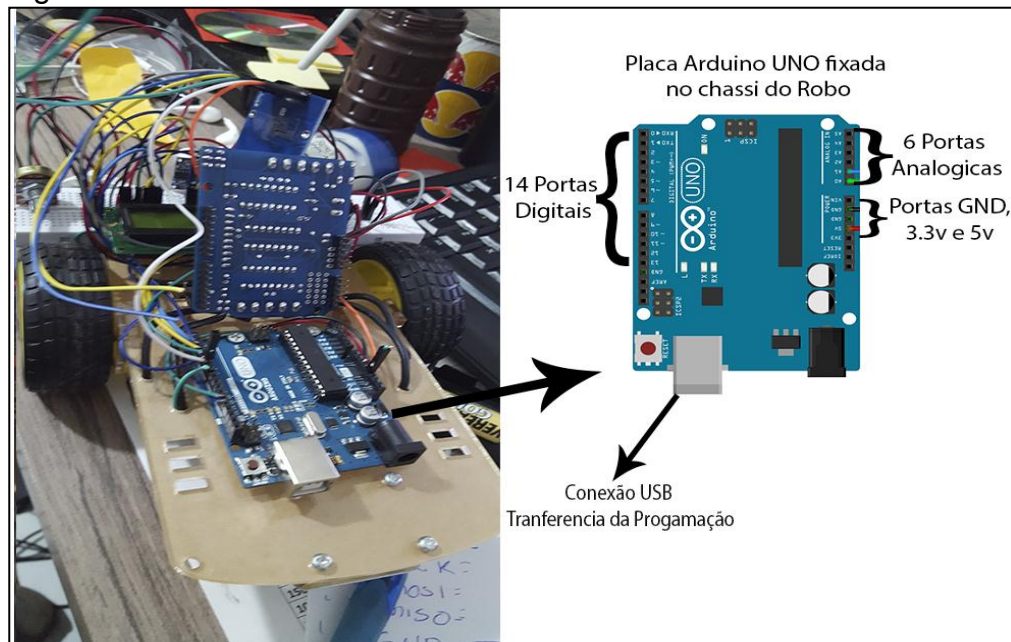
Figura 25 - Sensor Segue Linhas.



Fonte: Elaborada pelo autor.

Também faz parte da arquitetura um micro controlador (Arduino Uno), mostrado na Figura 25. Este componente hospeda toda programação, controlando o funcionamento por meio da linguagem de programação em C. Neste componente foi implementado o algoritmo de custo uniforme, desta forma, que micro controlador recebe os dados do produto (a partir da etiqueta RFID que estão coladas no produto). Estas etiquetas contêm informações de onde o produto será armazenado (ponto de destino). Ao colocar o produto no robô (Figura 26) o sensor RFID, conterà as informações da trajetória do robô, indicando um ponto de início e um ponto de destino.

Figura 26 - Arduino Uno no Robô Móvel.



Fonte: Elaborada pelo autor.

A Figura 26 ilustra o ponto de conexão USB. Esta conexão é usada para se ligar ao computador e estabelecer a comunicação com a IDE do Arduino, sendo possível, deste modo, compilar todo o código fonte para o micro controlador UNO (Figura 27).

Figura 27– IDE Arduino comunicando com Robô Móvel

```

oficial8.11 | Arduino 1.6.12
Arquivo Editar Sketch Ferramentas Ajuda

oficial8.11

#include <SPI.h>
#include <MFRC522.h>
#include <AFMotor.h>

#define SS_PIN 10
#define RST_PIN 9
AF_DCMotor M1(1);
AF_DCMotor M2(2);
MFRC522 mfrc522(SS_PIN, RST_PIN);

char st(20);

#define HC_SR04_TRIGGER A1 //Define o pino do Trigger do sensor ultrasonico no pino ANALÓGICO A2
#define HC_SR04_ECHO A0 //Define o pino do Echo do sensor ultrasonico no pino ANALÓGICO A3

int cont = 0;
int aux = 0;

void setup() {
  Serial.begin(9600); // Inicia a serial
  SPI.begin(); // Inicia SPI bus
  mfrc522.PCD_Init(); // Inicia MFRC522
  Serial.println("Aproxime o seu produtp...");
  Serial.println();
  pinMode(16, INPUT_PULLUP);
  pinMode(17, INPUT_PULLUP);
  Serial.begin(9600);
  pinMode(HC_SR04_TRIGGER, OUTPUT); //Define o trigger do sensor para enviar o sinal
  pinMode(HC_SR04_ECHO, INPUT); //Define o Echo do sensor para receber o sinal
}

void loop() {
  // ...
}

Compilação terminada

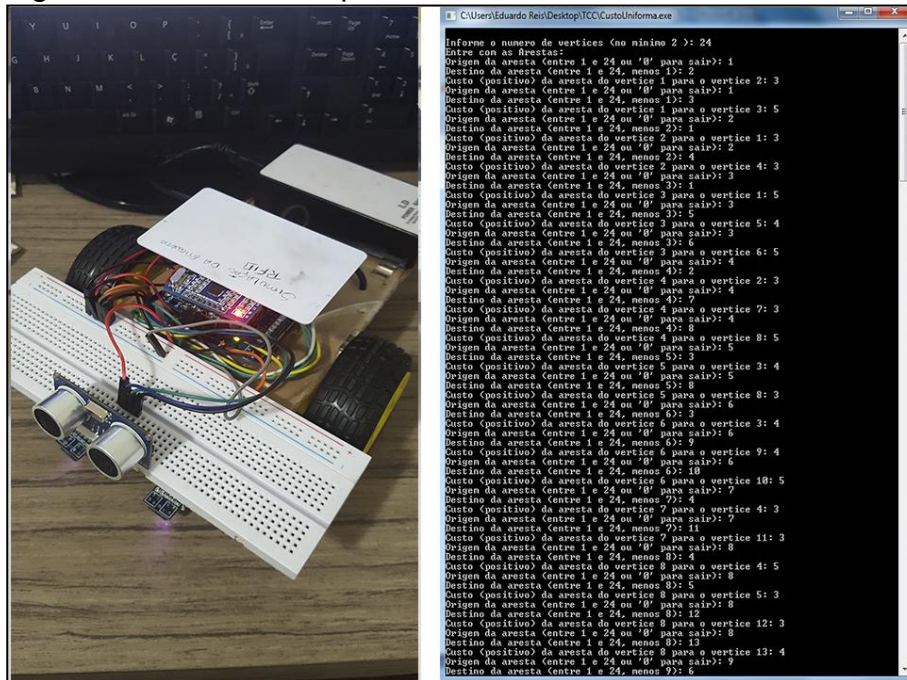
O sketch usa 6.668 bytes (20%) de espaço de armazenamento para programas. O máximo são 32.256 bytes.
Variáveis globais usam 317 bytes (15%) de memória dinâmica, deixando 1.751 bytes para variáveis locais. O máximo são 2.048 bytes.

```

Fonte:Elaborada pelo autor.

Após a leitura do produto, são repassadas as informações para o algoritmo de custo uniforme que pode, então, fazer o cálculo da melhor trajetória a ser realizada pelo robô (Figura 28).

Figura 28– Leitura e Captura de Dados do Produto



Fonte: Elaborada pelo autor.

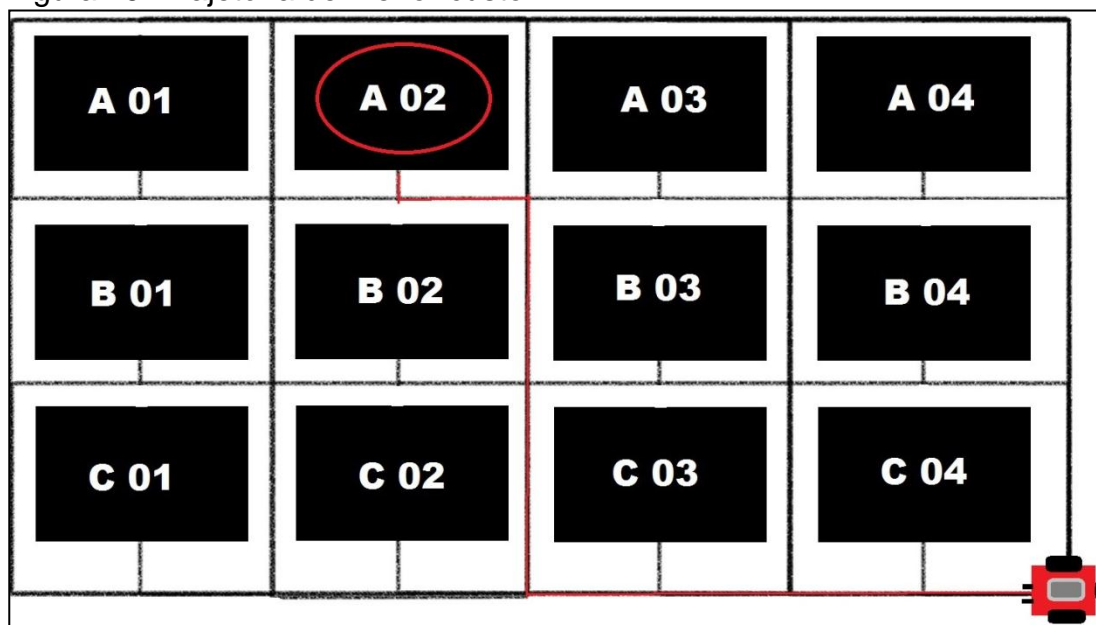
Conforme a Figura 28 o sensor lê a etiqueta RFID, indicando para o algoritmo o número do vértice que de origem e o destino. Logo depois ele calcula os valores de ponto a ponto, mostrando a melhor trajetória para o robô.

## 8 TESTE DO PROTÓTIPO

A fim de verificar o funcionamento do robô móvel, conforme descrito na seção anterior, foi elaborado um programa em linguagem C, que implemento algoritmo de custo uniforme. Tal algoritmo é, portanto, usado pelo robô para explorar o ambiente (simulando um armazém). Considerando os dados obtidos pelo sensor RFID, o robô define o seu objetivo dentro do ambiente e o algoritmo indica a trajetória de menor custo.

A Figura 29 mostra o ambiente que simula o armazém. O mesmo foi modelado fisicamente, utilizando fita isolante que indicam os caminhos por onde o robô poderia circular. Desta forma, o robô caminha sobre algumas dessas listras, dependendo do seu objetivo. Para testá-lo foi definido que o mesmo deveria sair do seu ponto inicial e chegar até o ponto A 02, conforme mostra a Figura 29.

Figura 29–Trajetória de menor custo



Fonte: Elaborada pelo autor.

A partir da identificação do objetivo do robô, o algoritmo de custo uniforme é executado pelo Arduino, fornecendo a trajetória de menor custo que é, então, executada pelo robô. As Figuras 30, 31 e 32 mostram a movimentação do robô até a chegada ao objetivo. Os anexos deste trabalho mostram o código-fonte do algoritmo implementado.



Figura 29 - Identificando o objetivo.



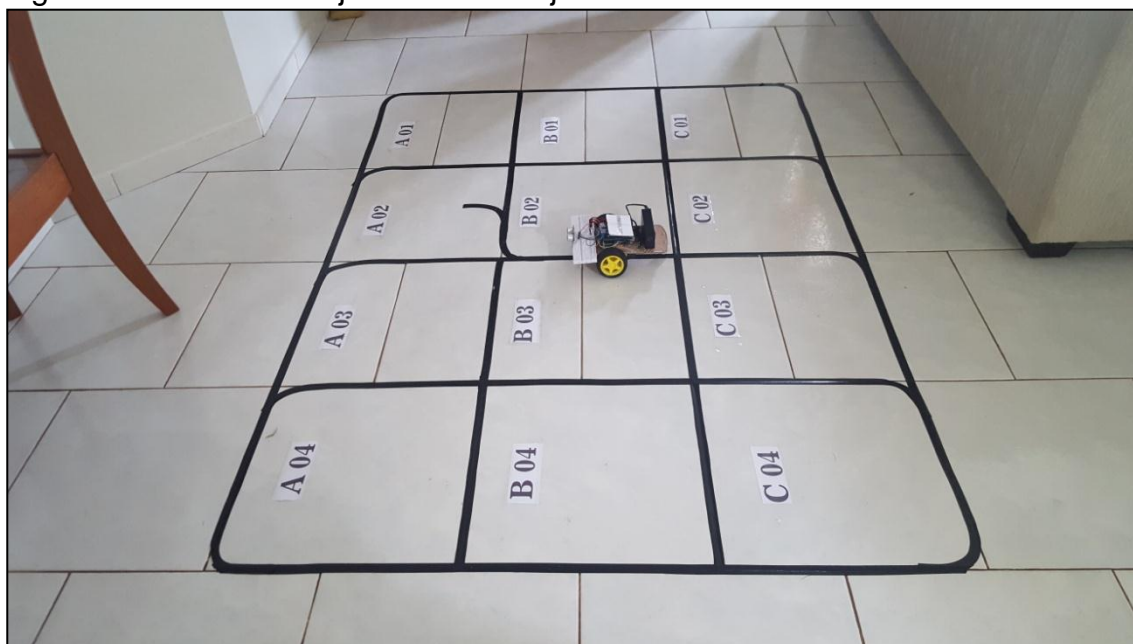
Fonte: Elaborada pelo autor.

Figura 30 - Início da trajetória.



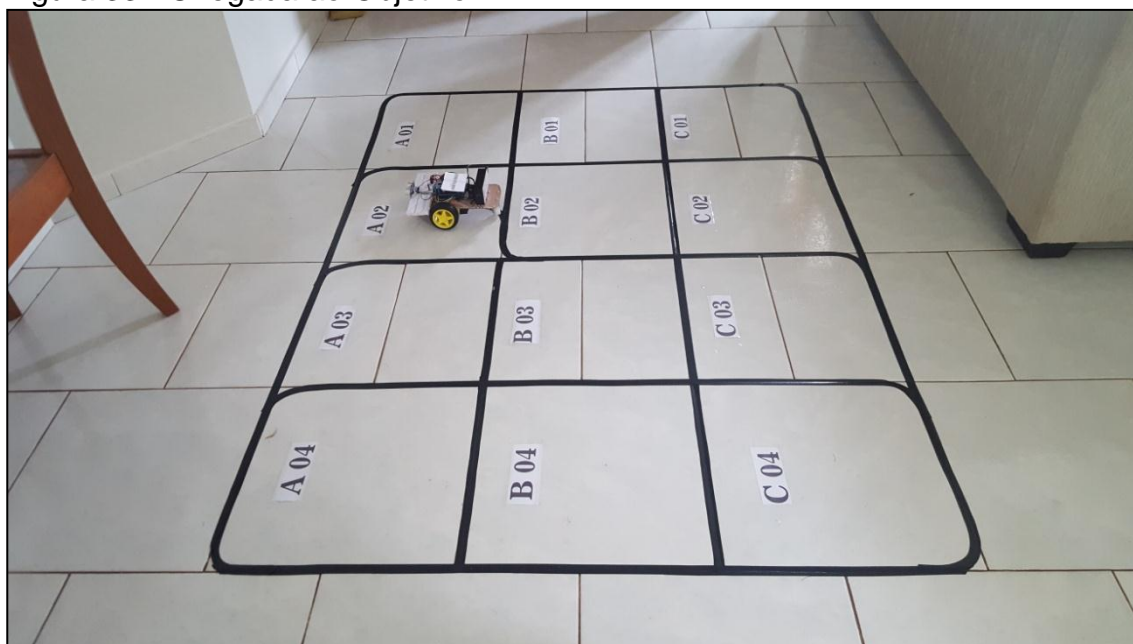
Fonte: Elaborada pelo autor.

Figura 32 - Meio da Trajetória até o objetivo.



Fonte: Elaborada pelo autor.

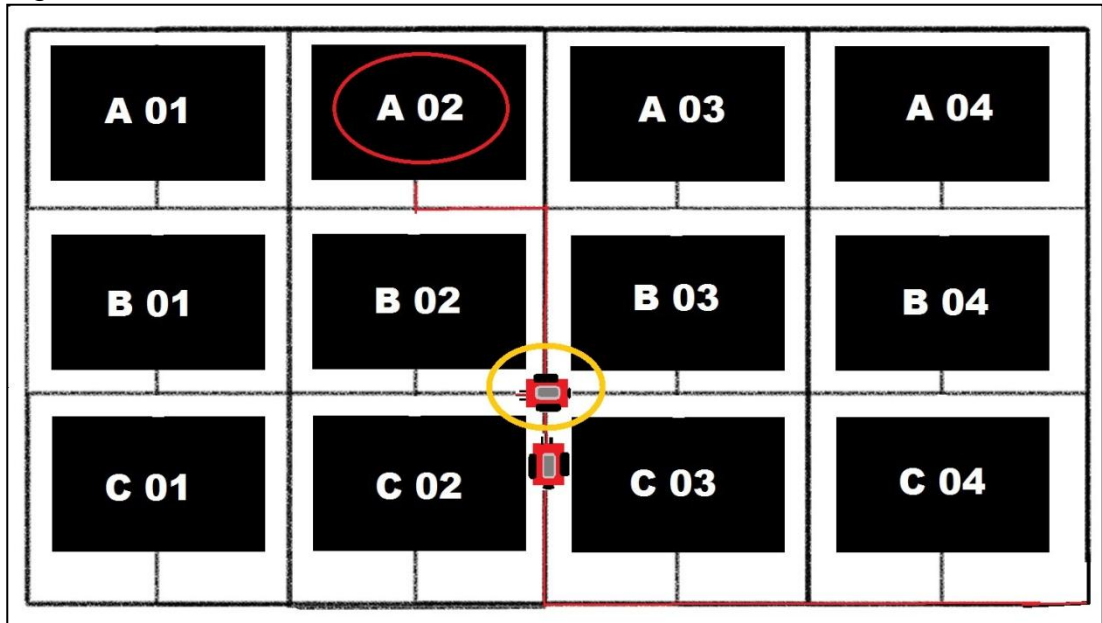
Figura 33 - Chegada ao Objetivo.



Fonte: Elaborada pelo autor.

Cabe destacar que o robô consegue identificar a presença de obstáculos à sua frente, usando o sensor sônico. Nestas situações, ele trava automaticamente as suas rodas, até que o objeto saia da sua frente. A Figura 34 ilustra uma situação que desencadearia a pausa do robô caso outro agente se encontrar à sua frente.

Figura 31 - Obstáculo à frente do robô



Fonte: Elaborada pelo autor.

## 9 CONSIDERAÇÕES FINAIS

Muitas são as áreas em que a robótica pode ser utilizada com sucesso, permitindo executar tarefas com os mais diversos propósitos, desde exploração de ambientes desconhecidos até o transporte de pessoas ou cargas. As aplicações envolvendo robôs móveis dão ao homem uma grande capacidade de desenvolver os mais diversos tipos de aplicações do mundo real, amplificando o potencial humano. Os robôs móveis são máquina que se deslocam pelo ambiente usando rodas, pernas ou algum mecanismo parecido. Foram projetados para entrega de alimentos em hospitais, movimentação de contêineres em docas de carga e tarefas semelhantes.

O projeto de um robô envolve geralmente grande complexidade, já que estão inseridos neste contexto projetos mecânicos e elétricos. Entretanto, com o avanço da tecnologia, atualmente, por meio de dispositivos relativamente baratos, a exemplo do micro controlador Arduino, é possível que qualquer pessoa desenvolva um protótipo de um robô. Além de um micro controlador, o robô pode ser incrementado com sensores que permitem a este agente autônomo extrair informações do ambiente levando-o a reagir às mudanças do mesmo de forma inteligente. A inteligência é uma das características desejadas nessas máquinas, pois permitem que, de fato, esse robô tenha autonomia para o processo de tomada de decisão. E este é, certamente, um dos principais desafios: projetar a estrutura lógica, ou seja, dar ao robô a “inteligência” necessária para que ele possa desempenhar suas tarefas de maneira eficiente.

Muitas são as possibilidades de se implementar programas que de em às máquinas algum tipo de capacidade especial ou inteligência. Neste sentido este projeto propôs a modelagem e o desenvolvimento de um robô móvel inteligente que pudesse realizar de forma automática o cálculo da melhor trajetória em ambiente simulado. Uma das principais contribuições desta pesquisa foi descrever o processo de modelagem e planejamento do agente robótico envolveu a descrição dos componentes utilizados no projeto. O hardware do robô teve como base o Arduino, e sua inteligência é pautada na implementação de um algoritmo de busca, especificamente a busca de custo uniforme. Como sugestão para trabalhos futuros recomenda-se a implementação de algoritmos que permitem a navegação em ambientes desconhecidos, usando algum tipo de exploração inteligente. Novos

sensores também poderão ser anexados ao robô, gerando novos dados que poderão ser explorados por outros algoritmos de Inteligência Artificial.

## REFERÊNCIAS

MEDEIROS. **Aplicação de Algoritmo Dijkstra ao Planejamento de Movimento de VANTS**. Disponível: <[http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18/2010/09.27.20.13/doc/REVISADO\\_Medeiros\\_Felipe\\_4.pdf](http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18/2010/09.27.20.13/doc/REVISADO_Medeiros_Felipe_4.pdf)> Acesso em 1 jul de 2016.

**Aplicando o Algoritmo de DIJKSTRA**. Disponível: <[http://www.sbmac.org.br/cnmacs/2004/cd\\_cnmac/files\\_pdf/10596a.pdf](http://www.sbmac.org.br/cnmacs/2004/cd_cnmac/files_pdf/10596a.pdf)> Acesso em 01/06/2016.

BENTO, CAMARGO. **Robô Bípe de Multifuncional**. Disponível em <[http://www.inicepg.univap.br/cd/INIC\\_2007/trabalhos/engenharias/inic/INICG004\\_01\\_01O.pdf](http://www.inicepg.univap.br/cd/INIC_2007/trabalhos/engenharias/inic/INICG004_01_01O.pdf)> Acesso em 10 abr de 2016.

CHARNIAK; MCDEMOTT. **Introdução à Inteligência Artificial**. Dunfermline, Reino Unido: Better World Books Ltd, 1985.

COPPIN. **Inteligência Artificial**. Rio de Janeiro: LTC, 2013

FACELI, Katti; LORENA, Ana Carolina; GAMA, João. **Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina**. Rio de Janeiro: LTC, 2011.

FERNANDES, Anita Maria da Rocha. **Inteligência Artificial: noções gerais**. Florianópolis: Visual Books, 2005.

GENESERETH, M. R. & Nilsson, N. J. **Logical Foundations of Artificial Intelligence**, Morgan Kaufmann Publishers, 1988.

MANZANO, José Augusto N. G. **Estudo Dirigido: Linguagem C**. São Paulo: Érica, 2010.

MCROBERT, Michel. **Arduino: Básico**. São Paulo: NOVATEC, 2011.

**Otimização de trajetórias através de caminhos mínimos para a locomoção de cadeira de rodas robótica**. Disponível: <[http://pse.ifes.edu.br/prppg/pesquisa/jornadas/jornada\\_2012\\_2013/anais/1692013044647.doc](http://pse.ifes.edu.br/prppg/pesquisa/jornadas/jornada_2012_2013/anais/1692013044647.doc)> Acesso em 1 jul de 2016.

PEREIRA, Prof.Dr. Silvio do Lago. **Introdução à Inteligência Artificial**. Disponível em <<http://www.ime.usp.br/~slago/IA-introducao.pdf>> Acesso em: 22 mar de 2016.

POOLE, D.; MACKWORTH, A. K.; GOEBEL, R. **Computational Intelligence: A Logical Approach**. Oxford: Oxford University, 1998.  
RUSSEL, Stuart; NORVIG, Peter. **Inteligência Artificial: Tradução da Segunda Edição**. Rio de Janeiro: Elsevier, 2004.

SOUZA, SCUCUGLIA, JUNIOR, MALDONADO, SILVA. **Ambiente Virtual para Simulação de Navegação de Robôs Autônomos Móveis Através de Grafos**

STAIR, M Ralph. REYNOLDS George W. **Princípios de Sistema de Informação**. 6ed. São Paulo: Pioneira, 2006.

## APÊNDICE

```
/*SOFTWARE – ROBÔ MÓVEL*/
```

```
#include <SPI.h>
```

```
#include <MFRC522.h> // Inclusão da Biblioteca do Sensor RFID
```

```
#include <AFMotor.h> // Inclusão da Biblioteca dos Motores
```

```
#define SS_PIN 10 //Definição dos Pino ligado do Sensor Rfid ao Arduino
```

```
#define RST_PIN 9 //Definição dos Pino ligado do Sensor Rfid ao Arduino
```

```
#define HC_SR04_TRIGGER A1 //Define o pino do Trigger do sensor ultra-sônico no pino ANALÓGICO A2
```

```
#define HC_SR04_ECHO A0 //Define o pino do Echo do sensor ultra-sônico no pino ANALOGICO A3
```

```
//configurando os pinos (executado apenas uma vez)
```

```
AF_DCMotor M1(1); //Definição dos Pino ligado do Sensor Rfid ao Arduino
```

```
AF_DCMotor M2(2); //Definição dos Pino ligado do Sensor Rfid ao Arduino
```