

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
ENGENHARIA DE COMPUTAÇÃO**

Alan Kunz Cechinel

**DESENVOLVIMENTO DE UM SISTEMA DE LOGÍSTICA  
PARA UM ROBÔ MÓVEL HOSPITALAR UTILIZANDO  
MAPAS DE GRADE**

Araranguá

2018



Alan Kunz Cechinel

**DESENVOLVIMENTO DE UM SISTEMA DE LOGÍSTICA  
PARA UM ROBÔ MÓVEL HOSPITALAR UTILIZANDO  
MAPAS DE GRADE**

Monografia de Conclusão de Curso submetida ao Curso de Engenharia de Computação para a obtenção do Grau de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Anderson Luiz Fernandes Perez

Araranguá

2018

Catalogação na fonte elaborada pela biblioteca da  
Universidade Federal de Santa Catarina

A ficha catalográfica é confeccionada pela Biblioteca Central.

Tamanho: 7cm x 12 cm

Fonte: Times New Roman 9,5

Maiores informações em:  
<http://www.bu.ufsc.br/design/Catalogacao.html>

Alan Kunz Cechinel

## **DESENVOLVIMENTO DE UM SISTEMA DE LOGÍSTICA PARA UM ROBÔ MÓVEL HOSPITALAR UTILIZANDO MAPAS DE GRADE**

Esta Monografia de Conclusão de Curso foi julgada aprovada para a obtenção do Título de “Bacharel em Engenharia de Computação”, e aprovada em sua forma final pelo Curso de Engenharia de Computação.

Araranguá, 5 de julho 2018.

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Eliane Pozzebon  
Coordenadora do Curso

---

Prof. Dr. Anderson Luiz Fernandes Perez  
Orientador

### **Banca Examinadora:**

---

Prof. Dr. Anderson Luiz Fernandes Perez  
Presidente

---

Prof. Dr. Gustavo Mello Machado



---

Prof. Dr. Fábio Rodrigues de la Rocha

---

Prof. Dr. Edson Roberto de Pieri



Para aqueles cujo as mais de vinte mil palavras contidas nesta Monografia de Conclusão de Curso não seriam suficientes em agradecimento, meus pais, Silvio Cechinel e Maria Salete Kunz Cechinel.



## **AGRADECIMENTOS**

Agradeço a Deus, pelas oportunidades que me foram concedidas durante a graduação.

A meus familiares que apoiaram minhas decisões e me auxiliaram a seguir o caminho correto nesta jornada. Especialmente a meus pais e padrinhos, sendo estes os que mais colaboraram para que o sonho de fazer a graduação na Universidade Federal de Santa Catarina se concretizasse.

Agradeço a meus amigos e colegas, que de alguma forma cooperaram para que fosse possível a conclusão deste curso de graduação. Em especial, ao meu conterrâneo, que tornou-se um grande amigo e companheiro de estudos, Thiago Raulino Dal Pont.

A todos amigos que fiz no decorrer destes anos no Laboratório de Automação e Robótica Móvel - LARM, pelo companheirismo, conhecimento e ambiente agradável compartilhado. Em destaque, Elder Dominghini Tramontin, por sua paciência ao me passar seus conhecimentos no início do curso.

A todos os professores por compartilharem seus conhecimentos. Especialmente ao meu orientador, o professor Anderson Luiz Fernandes Perez, que serviu como inspiração acadêmica desde o início do curso, pelo suporte prestado no desenvolvimento desta monografia. Também ao professor Ricardo Alexandre Reinaldo de Moraes, que me proporcionou a participação em projeto de iniciação científica, assim, tornando-se um exemplo de pesquisador para mim.

Por fim, agradeço ao professor Edson Roberto de Pieri e a professora Patrícia Della Mea Plentz pela confiança em mim depositada ao emprestarem o robô utilizado neste trabalho.



*The Three Laws of Robotics:*

*1: A robot may not injure a human being or, through inaction, allow a human being to come to harm;*

*2: A robot must obey the orders given it by human beings except where such orders would conflict with the First Law;*

*3: A robot must protect its own existence as long as such protection does not conflict with the First or Second Law;*

*The Zeroth Law: A robot may not harm humanity, or, by inaction, allow humanity to come to harm.*

Isaac Asimov



## RESUMO

Dispositivos robóticos estão inseridos no cotidiano industrial a décadas, atuando como manipuladores fixados em seu local de trabalho controlado, onde são responsáveis por realizarem tarefas repetitivas e por vezes insalubres. Robôs móveis, no entanto, devem lidar com o dinamismo de um ambiente (e.g. pessoas, animais e objetos) para se deslocarem, tornando a execução das atividades muito mais complexa do que em um ambiente controlado. No intuito de facilitar o processo de navegação de um robô móvel, mapas podem ser utilizados para auxiliar na localização e no planejamento de rota. A navegação de um robô móvel em um ambiente envolve a realização de tarefas que, muitas vezes, podem ter prioridades. Assim, o sistema que planeja a rota a ser percorrida pelo robô deve levar em consideração não apenas encontrar o menor caminho, mas também o que permite que a prioridade na execução da tarefa seja atendida. O ambiente hospitalar oferece um nível elevado de dinamismo, o que aumenta a complexidade de navegação, além de requisitar que suas tarefas sejam executadas de forma prioritária. Devido às características mencionadas tornarem as tarefas para robôs móveis desafiadoras, escolheu-se ambientes dinâmicos, como o hospitalar, para atuação do sistema desenvolvido, permitindo com que tarefas de logística sejam atendidas com base em suas prioridades em um ambiente previamente mapeado.

**Palavras-chave:** Robótica Móvel, Navegação, Prioridades, Escalonamento, Planejamento de Rotas.



## **ABSTRACT**

Robotic devices are inserted at industrial daily for decades, acting like manipulators fixed at its controlled workplace, where they are responsible by perform repetitive tasks and frequently unhealthy. Mobile robots, however, must deal with all environment dynamism (e.g. people, animals and objects) to move himself, making activities execution much more complex than in a controlled environment. In order to facilitate the navigation process of a mobile robot, maps can be utilized to assist the localization and path planning. The navigation of a mobile robot in an environment involves performing tasks that, many times, can have priorities. Thus, the system that plans the route to be traveled by the robot must take into account not just find the smallest path, but also what allows that the priority in the execution of the task is attended. The hospital environment offers a high dynamism level, what increases the navigation complexity, besides to request that its tasks be performed on a priority basis. Due the characteristics mentioned make tasks for mobile robots challenging, was chosen dynamic environments, like the hospitalar, for the developed system acting, allowing that logistic tasks be fulfilled basing in its priorities in an previously mapped environment.

**Keywords:** Robotics, Navigation, Priorities, Scheduling, Path Planning.



## LISTA DE FIGURAS

Figura 1	Diagrama da tarefa de navegação de um robô móvel.....	34
Figura 2	Exemplo de um mapa Métrico 200 x 200. Em vermelho áreas ocupadas, em verde áreas livres ou não mapeadas. ....	35
Figura 3	Exemplo de um mapa topológico contendo informações de qual espaço real está associado ao vértice. ....	36
Figura 4	Semântica do mapeamento.....	37
Figura 5	Exemplo de mapa topológico limitado e mapa de grade interno. ....	38
Figura 6	Anotação de cena.....	39
Figura 7	Diagrama de localização. ....	40
Figura 8	Robô manipulador móvel. ....	43
Figura 9	Imagen do Robô Curiosity em Marte. ....	43
Figura 10	Robô aspirador de pó. ....	44
Figura 11	Robô de transporte interno. ....	44
Figura 12	Robô VinBot para controle de qualidade de plantações de uva. ....	45
Figura 13	Robô militar TALON. ....	45
Figura 14	Posicionamento do robô. ....	47
Figura 15	Ilustração do funcionamento do algoritmo de localização de Markov.....	49
Figura 16	Ilustração do funcionamento do algoritmo de localização EKF. ....	53
Figura 17	Ilustração do funcionamento do algoritmo de localização de Monte Carlo.....	56
Figura 18	Ilustração de exemplo de modelo inverso sensorial. ....	59
Figura 19	Ilustração do mapeamento EKF.....	61
Figura 20	Ilustração do mapeamento de grade utilizando partículas.	64
Figura 21	Diagrama da arquitetura proposta. ....	67
Figura 22	Banco de dados. ....	70
Figura 23	Caminho convertido para pontos de passagem. ....	72
Figura 24	Diagrama do controle de baixo nível. ....	73
Figura 25	Modelo de tarefa. ....	75
Figura 26	Exemplo de coordenada para local. ....	77

Figura 27 Conjunto de tarefas de exemplo para o algoritmo.....	78
Figura 28 Fluxo de execução do controlador de alto nível.....	81
Figura 29 Mapa do ambiente. (a) o mapa original. (b) o mapa editado no editor GIMP.....	84
Figura 30 Ilustração da medida de distância.....	85
Figura 31 Esquemático do Experimento 1.....	87
Figura 32 Relação entre o tempo necessário para escalonar e o número de tarefas. Eixo tempo em escala logarítmica.....	89
Figura 33 Esquemático do Experimento 2.....	90
Figura 34 Esquemático do Experimento 3.....	92
Figura 35 Foto do robô Pioneer P3-DX com as modificações efetuadas para o experimento.....	95
Figura 36 Chegada do robô à Sala 306.....	96
Figura 37 Posição dos locais no mapa.....	120
Figura 38 <i>D* Lite</i> : Segunda Versão.....	177

## LISTA DE TABELAS

Tabela 1	Comparativo entre algoritmos de localização.....	57
Tabela 2	Comparativo entre algoritmos de mapeamento.....	66
Tabela 3	Dados de precisão com mapa reduzido em um meio....	86
Tabela 4	Dados de precisão com mapa reduzido em um terço....	86
Tabela 5	Sequência de execução das tarefas. As Tarefas são executadas da esquerda para a direita, sendo que o grupo de quinze e o de trinta tarefas possuem mais de uma parte.....	88
Tabela 6	Tamanho do grupo de tarefas.....	90
Tabela 7	Resultados do escalonamento para cada um dos grupos após a adição das tarefas.....	91
Tabela 8	Identificadores das tarefas utilizadas nas modificações..	93
Tabela 9	Grupo de tarefas em cada estágio.....	93
Tabela 10	Resultados do escalonamento para cada um dos grupos após a modificação.....	93
Tabela 11	Registro de tempo do experimento.....	94
Tabela 12	Tabela contendo a descrição e coordenadas dos locais utilizados nos experimentos.....	119
Tabela 13	Lista tarefas utilizadas nos experimentos. Id significa identificador e, L1, L2 e L3 são os locais a serem visitados. ....	121
Tabela 14	Medida de tempo para o escalonamento.....	122
Tabela 15	Escalonamento para o conjunto de cinco tarefas, com redução no mapa para um meio.....	122
Tabela 16	Escalonamento para o conjunto de cinco tarefas, com redução no mapa para um terço.....	123
Tabela 17	Escalonamento para o conjunto de dez tarefas.....	124
Tabela 18	Escalonamento para o conjunto de quinze tarefas. ....	125
Tabela 19	Escalonamento para o conjunto de trinta tarefas. ....	129
Tabela 20	Medida de tempo para o escalonamento.....	142
Tabela 21	Escalonamento para o conjunto de cinco até dez tarefas.	143
Tabela 22	Escalonamento para o conjunto de dez até quinze tarefas.	145
Tabela 23	Escalonamento para o conjunto de quinze até trinta tarefas.....	148
Tabela 24	Medida de tempo para o escalonamento.....	162

Tabela 25 Escalonamento para o conjunto de tarefas após a primeira modificação.....	162
Tabela 26 Escalonamento para o conjunto de tarefas após a segunda modificação.....	164
Tabela 27 Fluxo de execução das cinco tarefas em simulação.....	166
Tabela 28 Fluxo de execução das cinco tarefas em ambiente real ..	170

## LISTA DE ABREVIATURAS E SIGLAS

CEO	<i>Chief Executive Officer</i> .....	27
IRID	<i>International Research Institute for Nuclear Decommissioning</i> .....	29
OCDE	Organização para a Cooperação e Desenvolvimento Econômico .....	29
PID	Controlador Proporcional Integral Derivativo.....	41
AGV	<i>Autonomous Guided Vehicle</i> .....	44
EKF	<i>Extended Kalman Filter</i> .....	50
MCL	<i>Monte Carlo Localization</i> .....	53
MIS	Modelo Inverso Sensorial.....	58
SLAM	<i>Simultaneous Localization and Mapping</i> .....	60
SIR	<i>Sampling Importance Resampling</i> .....	64
AMCL	<i>Adaptive Monte Carlo Localization</i> .....	73



## **LISTA DE ALGORITMOS**

1	LOCALIZAÇÃO DE MARKOV .....	48
2	LOCALIZAÇÃO EKF .....	51
3	LOCALIZAÇÃO DE MONTE CARLO .....	54
4	MAPEAMENTO DE GRADE .....	59
5	EKF-SLAM .....	62
6	SAMPLING IMPORTANCE RESAMPLING .....	65
7	ORDENAÇÃO DE TAREFAS .....	80



## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	27
1.1 MOTIVAÇÃO E JUSTIFICATIVA.....	29
1.2 OBJETIVOS .....	30
1.2.1 Objetivos Gerais .....	30
1.2.2 Objetivos Específicos.....	30
1.3 METODOLOGIA.....	30
1.4 ORGANIZAÇÃO DO TRABALHO .....	31
<b>2 ROBÓTICA MÓVEL .....</b>	33
2.1 ROBÔ MÓVEL .....	33
2.2 MAPEAMENTO .....	34
2.3 LOCALIZAÇÃO.....	39
2.4 NAVEGAÇÃO .....	41
2.5 APLICAÇÕES DA ROBÓTICA MÓVEL .....	42
<b>3 LOCALIZAÇÃO E MAPEAMENTO EM ROBÓTICA MÓVEL .....</b>	47
3.1 CONSIDERAÇÕES A RESPEITO DO POSICIONAMENTO DO ROBÔ .....	47
3.2 LOCALIZAÇÃO .....	47
3.2.1 Markov .....	48
3.2.2 Filtro de Kalman Estendido .....	50
3.2.3 Monte Carlo .....	53
3.2.4 Comparativo .....	57
3.3 MAPEAMENTO .....	57
3.3.1 Grade .....	58
3.3.2 EKF-SLAM .....	60
3.3.3 Filtro de Partículas Rao-Blackwellized .....	63
3.3.4 Comparativo .....	65
<b>4 ARQUITETURA PROPOSTA PARA UM SISTEMA DE LOGÍSTICA HOSPITALAR .....</b>	67
4.1 ARQUITETURA GERAL .....	67
4.2 MAPEAMENTO .....	69
4.3 BANCO DE DADOS .....	70
4.4 PLANEJADOR DE ROTA GLOBAL .....	71
4.5 CONTROLE DE BAIXO NÍVEL .....	72
4.5.1 Localização .....	73
4.5.2 Planejador de Rota Local .....	74
4.5.3 Controle Robótico .....	74

4.6 ESCALONADOR DE TAREFAS .....	74
4.6.1 Modelo de Tarefa .....	75
4.6.2 Algoritmo de Ordenação .....	78
4.7 CONTROLADOR DE ALTO NÍVEL .....	81
<b>5 AVALIAÇÃO DO SISTEMA DE LOGÍSTICA HOSPITALAR.....</b>	<b>83</b>
5.1 MAPEAMENTO .....	83
5.1.1 Mapa obtido .....	83
5.1.2 Precisão durante a navegação .....	84
5.2 ESCALONADOR DE TAREFAS .....	86
5.2.1 Experimento 1 .....	87
5.2.2 Experimento 2 .....	90
5.2.3 Experimento 3 .....	92
5.3 AVALIAÇÃO EM AMBIENTE REAL .....	94
5.3.1 Modificações realizadas no robô .....	94
5.3.2 Resultados obtidos .....	95
<b>6 CONSIDERAÇÕES FINAIS E PROPOSTAS PARA TRABALHOS FUTUROS .....</b>	<b>99</b>
6.1 PROPOSTAS PARA TRABALHOS FUTUROS .....	100
6.2 INFORMAÇÕES ADICIONAIS .....	101
<b>REFERÊNCIAS .....</b>	<b>103</b>
<b>APÊNDICE A - Configurações de <i>drivers</i> do <i>player</i> .....</b>	<b>113</b>
<b>APÊNDICE B - Dados dos experimentos de validação ..</b>	<b>119</b>
<b>ANEXO A - Algoritmo <i>D* Lite</i> .....</b>	<b>177</b>

## 1 INTRODUÇÃO

A robótica conhecida hoje tem suas origens com George Devol, que projetou o primeiro robô programável, um braço robótico, tendo sua patente concedida em 1961. Através de uma parceria com Joe Engleberger, que era diretor na *Consolidated Controls Corp* e que foi responsável por convencer o *Chief Executive Officer* (CEO) da empresa a financiar a ideia de Devol, projetaram o primeiro robô industrial, o Unimate, a ser produzido em larga escala (RIA, 2015). A partir desse ponto da história, os robôs emergiram então dos laboratórios e se tornaram atrativos como produtos, atuando nas mais diversas áreas, como os casos de sucesso que são comumente conhecidos, envolvendo o transporte de materiais, limpeza, atuador em ambientes insalubres ou com atividades repetitivas, robôs com finalidade militar entre outros.

De acordo com Russell e Norvig (2002) a grande maioria dos robôs se enquadram em três categorias: manipuladores, móveis e humanoides. Manipuladores, ou braços robóticos, são ancorados em seu local de trabalho, onde devido às suas articulações têm a capacidade de se posicionar em qualquer ponto dentro do seu alcance. Móveis por sua vez podem se deslocar através do ambiente por meio de rodas, pernas ou outros mecanismos para locomoção. O terceiro tipo, um híbrido entre manipuladores e móveis são os humanoides, cuja estrutura física imita a dos seres humanos.

Para se deslocar no ambiente robôs móveis executam um processo chamado navegação, responsável por manter e determinar o caminho ou trajetória para um destino. Enquanto os seres humanos realizam a navegação utilizando dados proprioceptivos ou da navegação baseada em pontos de referência (LOOMIS; KLATZKY; GOLLEDGE, 2001), onde dependem de sua capacidade de percepção do ambiente juntamente com um mapa externo ou cognitivo, os robôs móveis realizam este processo através da leitura de diversos tipos de sensores e uso de ferramentas, como: bússola, quando deseja-se fazer a orientação do robô em relação ao norte magnético, mapa do terreno para se localizar e auxiliar no planejamento de rotas, encoder para medir a distância viajada e giroscópio para aumentar a acurácia deste tipo de estimativa, sensor de distância para que o robô possa detectar obstáculos, entre outros sensores que podem ser utilizados na assistência a navegação (BEKEY, 2005).

Diversos sistemas essenciais a navegação de robôs móveis vem sendo desenvolvidos, tal como mapeamento de ambientes, localização,

comunicação, planejamento de rota e algoritmos de aprendizado (TAKAHASHI et al., 2010). Através destas tecnologias a habilidade de navegação torna-se cada vez mais eficiente, possibilitando assim as atividades robóticas em ambientes dinâmicos até mesmo realizando a interação com seres humanos, como é o caso do robô MINERVA que foi projetado com a capacidade de fala e habilidade de acomodar pessoas em ambientes, sendo desenvolvido através da parceria entre *Carnegie Mellon University's Robot Learning Laboratory* e a *University of Bonn's Computer Science Department III* para ser utilizado no *National Museum of American History* (THRUN, 2000).

A navegação de um robô móvel em um ambiente qualquer exige a análise de dados provenientes de vários sensores. Esta análise determina qual ou quais tarefas o robô irá executar para alcançar o objetivo proposto. A tarefa mais básica que um robô móvel pode executar em um ambiente é a navegação livre de colisões. Entretanto, apesar da navegação livre de colisões ser considerada básica, geralmente, ela está associada ao mapeamento e a localização do robô no ambiente, o que demanda o uso de algoritmos mais elaborados, como por exemplo os de construção de mapas. Além de construir mapas um robô móvel pode realizar outras tarefas, por isso a fusão de sensores permite que o robô execute as tarefas para as quais foi programado com mais precisão. Embora o robô seja mais preciso ao navegar em um ambiente mapeado a partir dos dados provenientes de diversos sensores, não é possível garantir que a execução de uma tarefa ou função principal seja realizada em sua totalidade ou mesmo com precisão.

O ambiente hospitalar é repleto de dinamismo, devido ao alto fluxo de pessoas, equipamentos que podem ser realocados de acordo com necessidade, além de situações de emergência onde o espaço de trabalho deve se adaptar rapidamente. A entrega de medicamentos e materiais em hospitais faz parte da rotina de trabalho, portanto, consome recursos humanos que poderiam estar sendo aplicados em atividades onde o intelecto e habilidades humanas são, até o momento, mais eficientes, como o cuidado de pacientes, exames e preparo de alimentos.

Para realizar a tarefa de entregas no interior de um hospital o robô deve lidar com prioridades, pois um material ou medicamento necessário num dado momento em um centro cirúrgico é uma tarefa com maior urgência do que um medicamento que deve ser entregue em trinta minutos. Além disso, lidar com toda a dinâmica presente no ambiente, desviando de obstáculos e replanejando rota quando necessário.

O sistema proposto neste trabalho permitirá, através de um ambiente previamente mapeado, a realização de atividades relacionadas

a logística de entrega de materiais em um ambiente hospitalar de maneira eficiente. Desta forma, englobando o escalonamento de tarefas e navegação autônoma em ambientes dinâmicos internos.

## 1.1 MOTIVAÇÃO E JUSTIFICATIVA

Robôs móveis já são uma realidade para substituir seres humanos em tarefas insalubres em ambientes perigosos. Recentemente um robô controlado remotamente foi desenvolvido através de uma parceria entre a empresa Toshiba com a IRID (*International Research Institute for Nuclear Decommissioning*), para realizar a limpeza da câmara de contenção no desastre nuclear de Fukushima, infelizmente o alto índice de radiação danificou os dispositivos internos do robô e a missão não pode ser concluída.

Em uma visão futurista, robôs estarão inseridos de forma ubíqua na vida dos seres humanos, substituindo-os em variadas tarefas, como limpeza, assistência a idosos, recepcionista nos mais variados estabelecimentos, transporte de equipamentos e materiais em hospitais entre outros exemplos, de tal forma que as máquinas permitam que os seres humanos apliquem suas habilidades únicas em atividades cada vez mais complexas.

De acordo com Baalbaki e Xie (2009), devido ao progresso da medicina e à evolução da demografia, a maioria dos países da OCDE (Organização para a Cooperação e Desenvolvimento Econômico) estão sofrendo com o envelhecimento da sociedade, assim aumentando o percentual de idosos e por sua vez diminuindo a quantidade de mão-de-obra. Desta forma, um dos setores que mais sofre com a alta demanda e falta de colaboradores especializados é o de serviços de saúde.

Visto que as tarefas de logística dentro do hospital exigem tempo dos seres humanos que poderiam estar focados em atividades de cuidado da saúde, o desenvolvimento de sistemas capazes de atuarem em um ambiente dinâmico para realizarem tarefas de logística tornou-se então um área de pesquisa e desenvolvimento promissora e desafiadora. Para tanto, o sistema deve ser capaz de reconhecer o ambiente, planejar a navegação dando suporte ao desvio de possíveis obstáculos estáticos ou móveis, ou seja, se adaptar as mudanças, além de prover a possibilidade de atribuir missões programadas e estocásticas, bem como gerenciar seu próprio consumo para desempenhar todas as atividades previstas.

Este trabalho busca responder a hipótese de que dado um ambiente dinâmico, hospitalar, é possível desenvolver um sistema robótico

móvel capaz de realizar tarefas referentes a logística, compostas por diferentes locais a serem visitados, bem como com prioridades na execução, para que a carga de trabalho humana referente as atividades de transporte de materiais sejam reduzidas.

## 1.2 OBJETIVOS

Esta seção apresenta o objetivo geral e os objetivos específicos desta monografia de conclusão de curso.

### 1.2.1 Objetivos Gerais

Desenvolver um sistema para um ambiente hospitalar previamente mapeado que possibilite que um robô móvel possa navegar livre de colisões e cumprir metas de logística no transporte de materiais.

### 1.2.2 Objetivos Específicos

1. Estudar os principais algoritmos de localização e mapeamento em robótica móvel;
2. Realizar o mapeamento do ambiente utilizado para validação do sistema;
3. Desenvolver um algoritmo que com base em um ambiente mapeado realize o escalonamento de tarefas com base em suas prioridades;
4. Desenvolver um sistema de controle para um robô do tipo Pioneer P3-DX que permita a navegação em ambientes dinâmicos;
5. Avaliar os itens (2) e (3) em diferentes cenários visando verificar a eficiência dos mesmos.

## 1.3 METODOLOGIA

Este trabalho é uma pesquisa tecnológica destinada ao desenvolvimento de um sistema que através de um ambiente mapeado permite que um robô móvel se localize e navegue em ambientes dinâmicos atendendo a requisição de tarefas com prioridades.

O software de controle será desenvolvido em linguagem de programação C++ e fará uso do servidor robótico *Player* 3.0.2 responsável por prover uma interface entre o sistema desenvolvido e o sistema robótico onde será aplicado. Além disso, o *Stage* 4.1.1 é utilizado para realizar simulações de tempo real para se ter uma estimativa do comportamento do robô.

Para validar o sistema será utilizado o robô Pioneer P3-DX equipado com 8 sensores ultrassônicos, sensor laser de varredura SICK LMS200 e câmera Canon VC-C50i.

O sistema será avaliado em um notebook equipado com processador Intel i7-3610QM, memória RAM de 8GB do tipo DDR3 e frequência 1600MHz, placa de vídeo NVIDIA GT-650M e sistema operacional Ubuntu Gnome 16.04 LTS.

O ambiente utilizado durante o desenvolvimento consiste nos corredores da universidade, pois proporcionam um ambiente dinâmico semelhante ao de um hospital.

## 1.4 ORGANIZAÇÃO DO TRABALHO

Esta monografia está organizada em mais cinco capítulos que abordam os seguintes conteúdos:

O **Capítulo 2** explora o que entende-se por robô móvel mostrando os dispositivos que o compõem, explana o processo de mapeamento apresentando os possíveis tipos de mapa juntamente com a taxonomia dos componentes mais frequentes nas implementações. Aborda o conceito de localização em robótica móvel e divide seus desafios em dimensões, por fim traz exemplos de aplicações da robótica móvel.

O **Capítulo 3** descreve os algoritmos mais disseminados na comunidade de robótica móvel para as áreas de mapeamento e localização, trazendo ao final de cada seção um comparativo com o objetivo de expor pontos positivos e negativos de cada um dos algoritmos.

O **Capítulo 4** demonstra a arquitetura desenvolvida para um sistema de logística hospitalar inicialmente de forma geral, com o intuito de exibir o fluxo de execução do sistema através de seis elementos. Na sequência, cada um de seus componentes são aprofundados a fim de demonstrar como desempenham suas funções.

O **Capítulo 5** traz a análise e discussão dos experimentos criados para avaliação o sistema desenvolvido através de três fontes: o mapeamento, visando demonstrar a capacidade de navegação, o escalonador de tarefas, confirmando a competência na tomada de decisões

na execução de tarefas com base em prioridades e, por fim, o sistema aplicado ao ambiente real, comprovando a capacidade de atuação em ambientes dinâmicos.

Finalmente, no **Capítulo 6**, são apresentadas as considerações finais, propostas para trabalhos futuros e informações adicionais relacionados aos arquivos do sistema.

## 2 ROBÓTICA MÓVEL

Este capítulo explora o que entende-se por robô móvel abordando os dispositivos que o compõem. Também é descrito o processo de mapeamento apresentando os possíveis tipos de mapa juntamente com a taxonomia dos componentes mais frequentes nas implementações. Aborda o conceito de localização em robótica móvel e divide seus desafios em dimensões, por fim traz exemplos de aplicações da robótica móvel.

### 2.1 ROBÔ MÓVEL

Para Thrun, Burgard e Fox (2005), robótica é a ciência de perceber e manipular o mundo físico através de dispositivos controlados computacionalmente.

Os dispositivos percebem o ambiente por meio de sensores, porém, sua percepção é limitada pois existem limites físicos, como de alcance e resolução, presença de ruídos que podem perturbar as medições, além da possibilidade do sensor danificar-se.

No intuito de manipular o ambiente, o robô utiliza atuadores, normalmente motores, que podem tornarem-se imprevisíveis, devido ao ruído de controle ou até mesmo desgaste mecânico.

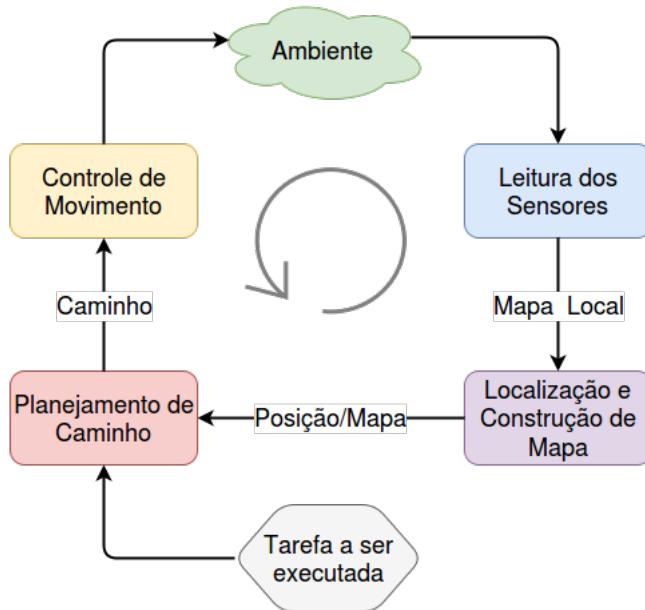
Um robô com a capacidade de percorrer o ambiente, é comumente chamado de robô móvel. De acordo com Bekey (2005) o que torna um robô, móvel, são seus atuadores para locomoção, como pernas, rodas ou hélices que exercem forças para o dispositivo se deslocar.

Quando se fala em robótica móvel, logo vem a cabeça máquinas bípedes, dotadas de rodas ou esteiras. Porém, submarinos e helicópteros controlados remotamente, e até mesmo *drones* podem ser considerados robôs móveis, afinal, percebem o ambiente através de seus sensores, interagem por meio de atuadores e ainda possuem certo grau de autonomia em seus controles.

Segundo LaValle (2016) dentre as tarefas mais comuns desempenhadas por um robô móvel está a navegação em um ambiente interno. Dentro desta atividade pode ser solicitado ao robô que ele mapeie o ambiente (LI et al., 2016), se localize de forma precisa através do mapa (JIANG et al., 2017) e que se locomova para algum local determinado, através do planejamento de rota (DUCHON et al., 2014). A navegação é ilustrada na Figura 1, onde pode-se perceber que o robô sente o am-

biente por meio de seus sensores, mas vale ressaltar que não apenas o ambiente, o robô pode possuir sensores para adquirir dados proprioceptivos (bateria, odometria, defeitos e etc.), além disso produz mapas, e através dos mesmos e seus dados sensoriais se localiza, por fim de acordo com a tarefa desejada planeja o caminho para se deslocar e desempenhar sua atividade.

Figura 1 – Diagrama da tarefa de navegação de um robô móvel.



Fonte: Adaptado de Raja e Pugazhenthi (2012)

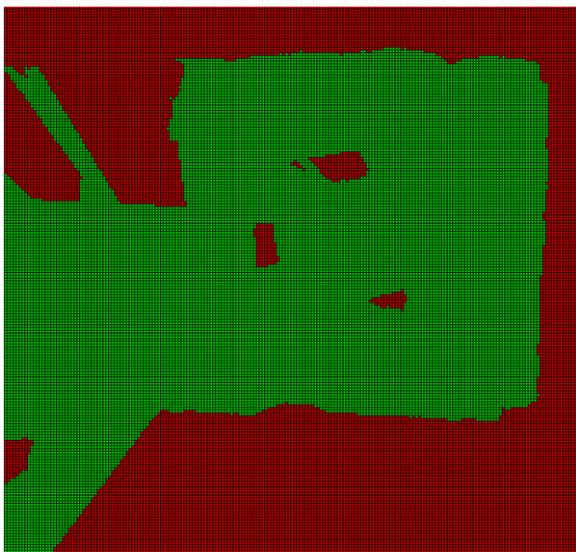
## 2.2 MAPEAMENTO

O mapeamento pode ser definido como o processo de estabelecer relações espaciais entre objetos estáticos (WANG et al., 2007). De acordo com Tapus e Siegwart (2005), robôs móveis normalmente fazem uso de mapas topológicos ou métricos para desempenhar sua navegação de forma confiável.

Os mapas métricos, ou de grade, são mapas que fazem uso de um conjunto de células com área pré determinada ao qual existe uma rela-

ção com o ambiente físico, ou seja, cada célula no mapa corresponde a uma unidade de área no meio ao qual o robô está inserido. Cada célula possui estados que podem ser livre, ocupado ou desconhecido (STACHNISS; BURGARD, 2005), ou apenas livre e desocupado (considerando não mapeado como desocupado) (KUNDU et al., 2016). Um exemplo deste tipo de mapa pode ser visualizado na Figura 2.

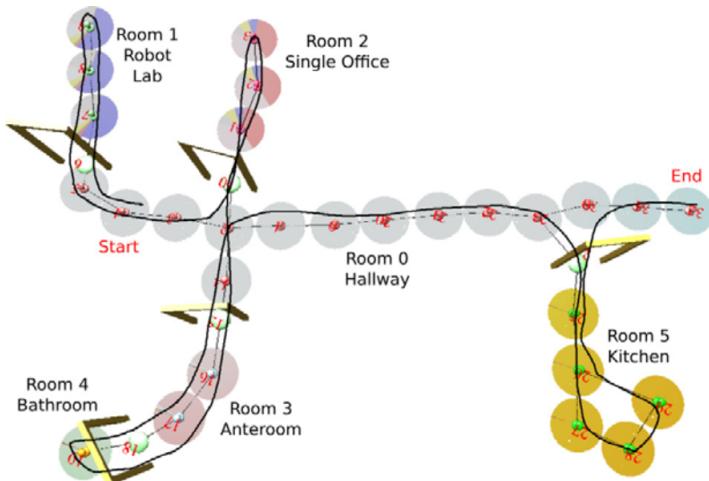
Figura 2 – Exemplo de um mapa Métrico 200 x 200. Em vermelho áreas ocupadas, em verde áreas livres ou não mapeadas.



Fonte: Kundu et al. (2016)

Os métodos topológicos, por sua vez, representam o ambiente através de grafos. Cada vértice é um local no ambiente (e.g. sala, escritório) (MARINHO et al., 2017), ou trecho de um possível caminho (RAMAITHITIMA et al., 2016), sendo conectados afim de estabelecerem uma representação total do local mapeado. Em cada vértice do mapa são armazenadas informações de características do local ao qual o vértice está posicionado (e.g. cantos de paredes, portas) e através destas características a localização do robô é inferida (KOSNAR et al., 2013). Um mapa topológico é apresentado na Figura 3.

Figura 3 – Exemplo de um mapa topológico contendo informações de qual espaço real está associado ao vértice.



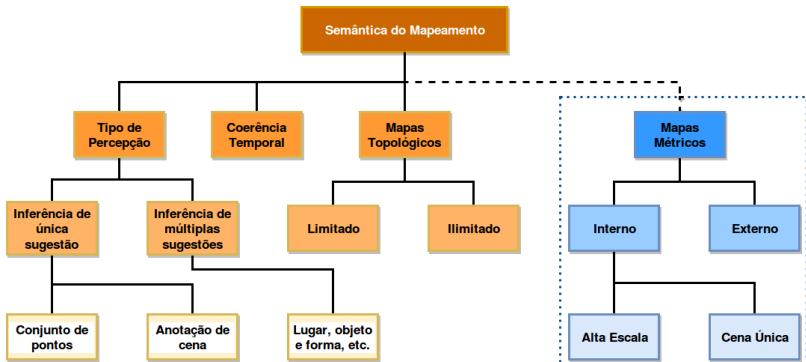
Fonte: Kostavelis e Gasteratos (2015)

No trabalho de Jia et al. (2010) é discutido que o método escolhido para o mapeamento causa impactos na precisão e no esforço para realizar os cálculos do mapa. Por exemplo, quando se utiliza da abordagem de grade, o mapa é facilmente feito e atualizado, além de garantir maior precisão métrica, porém, exige altíssimos níveis de processamento e memória para varrer e armazenar as células.

Mapas topológicos por sua vez, tem papel fundamental em problemas complexos de localização em robótica móvel, como o rapto do robô (PRONOBIS et al., 2010), localização global (BADINO; HUBER; KANADE, 2011), e na detecção de um mapa com circuito fechado (KOSNAR et al., 2013). Porém, a precisão métrica, já não é tão presente quanto na implementação de grade.

Kostavelis e Gasteratos (2015), apresentam uma ilustração da taxonomia dos componentes mais frequentes nas implementações de mapeamento robótico. Na Figura 4 os mapas métricos estão destacados na cor azul, pois normalmente são utilizados como componentes complementares, ou seja, dão suporte aos métodos semasiológicos (detecção de tipos de salas, objetos, de modo geral, ambientes habitados por humanos).

Figura 4 – Semântica do mapeamento.



Fonte: Adaptado de Kostavelis e Gasteratos (2015)

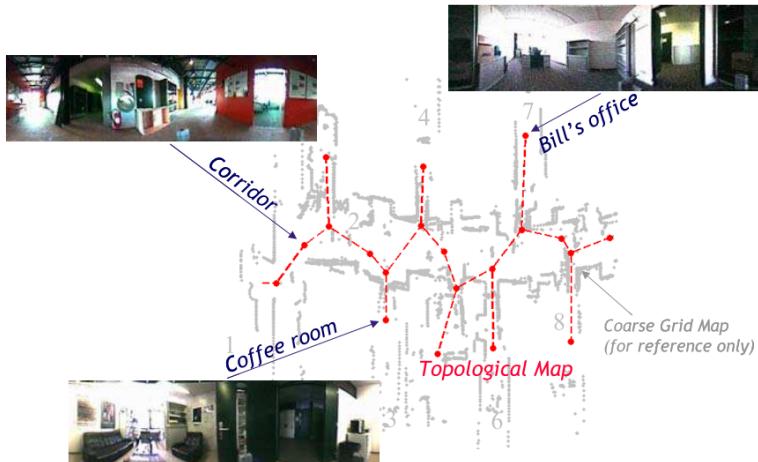
Além da diferenciação de mapas métricos e topológicos, é possível observar características de ambos, bem como outros componentes que são levados em consideração no mapeamento como a coerência temporal e o tipo de percepção.

A diferenciação nos mapas métricos internos e externos se dá principalmente pelo fato das implementações voltadas ao ambiente externo normalmente serem voltadas ao espaço 3D, como em Sengupta et al. (2013), detectando casas, árvores, carros. Na Figura 5 se observa um mapa interno, onde o ambiente é representado como, de certa forma, uma planta baixa. Quando se fala em alta escala, se diz respeito a basicamente mapear todo um ambiente, enquanto cena única é transformar o conteúdo de uma imagem, por exemplo, em um mapa métrico.

Mapas topológicos limitados, Figura 5, são aqueles que definem apenas alguns vértices principais (e.g. salas de uma universidade), já os não limitados são aqueles que possuem vértices por todo o caminho representando todos os pontos de referência detectados, como o da Figura 3.

A Figura 5 agrupa o mapa de grade em escala de cinza, e o topológico limitado em forma de pontos vermelhos.

Figura 5 – Exemplo de mapa topológico limitado e mapa de grade interno.

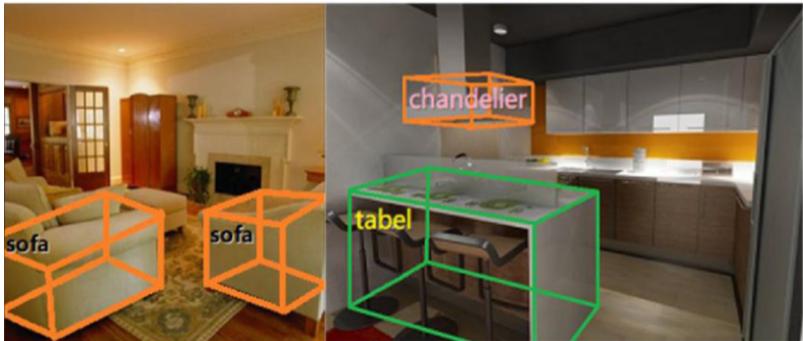


Fonte: Tapus e Siegwart (2005)

Kostavelis e Gasteratos (2015) apresentam através de outros trabalhos que a frequência de aquisição dos dados pode melhorar o mapeamento. Dessa informação surge o termo coerência temporal, ou seja, a leitura de um sensor deve ser tão rápida quanto a movimentação do robô, caso contrário ela pode se tornar inválida devido a perda de detalhes do ambiente.

A percepção do ambiente pode se dar através de uma única sugestão, ou seja, quando percebem apenas um elemento do ambiente (e.g. objetos ou geometria do ambiente), ou através de múltiplos elementos onde seria o caso de detectar, por exemplo, a geometria do local e os objetos inseridos nele. A inferência pode ainda ser subdividida em anotação de cena e conjunto de pontos. Quando se utiliza da anotação de cena se deseja encontrar características no ambiente (e.g. detectando objetos) como é o caso do trabalho de Ye et al. (2015). Por sua vez através do conjunto de pontos a geometria do ambiente é capturada, como no mapa da Figura 6.

Figura 6 – Anotação de cena.



Fonte: Ye et al. (2015)

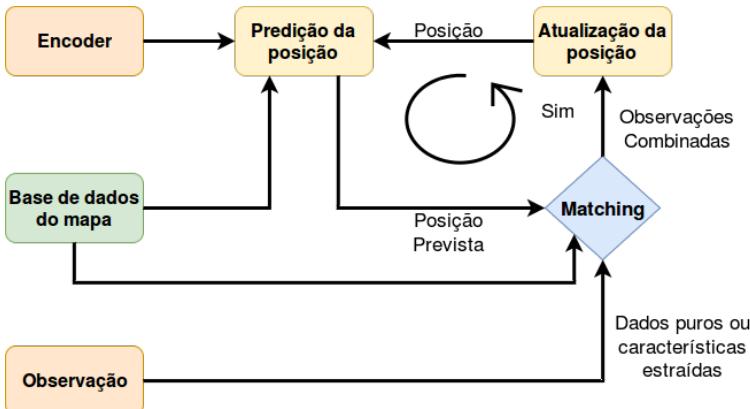
### 2.3 LOCALIZAÇÃO

Um robô móvel deve determinar o estado do seu mundo, ou seja, perceber a si mesmo e seus arredores para que seja possível navegar em um ambiente complexo. A habilidade de automaticamente determinar sua posição no espaço é chamada de localização (DULIMART; JAIN, 1997).

Localização pode ser definida como o problema de determinar a posição do robô em relação ao mapa que o mesmo dispõe do ambiente. No caso de robôs móveis utilizando um mapa 2D, a localização no instante de tempo  $t$ , é dada por  $x_t = [x, y, \theta]$  (THRUN; BURGARD; FOX, 2005).

Um exemplo de esquemático do sistema de localização que utiliza *encoder* nas rodas é ilustrado na Figura 7. É possível observar que para se localizar o robô utiliza um mapa do ambiente e dados adquiridos da leitura de seus sensores, desta forma ele faz uma correlação entre o ambiente físico e a base de dados do mapa e define a posição atual que é novamente utilizada no próximo ciclo para estimar a nova posição.

Figura 7 – Diagrama de localização.



Fonte: Adaptado de Siegwart, Nourbakhsh e Scaramuzza (2011)

Para compreender melhor a complexidade da localização em robótica móvel Thrun, Burgard e Fox (2005) dividem a taxonomia do problema em quatro dimensões: localização global versus local, ambientes estáticos e dinâmicos, implementações ativas e passivas, e por fim, localização singular versus multi robótica.

Quando a posição inicial do robô é conhecida, e a tarefa consiste simplesmente em localizar o robô a partir da posição inicial, pode-se dizer que é um problema de localização local desde que as incertezas estejam confinadas apenas na posição real do robô, e não no mapa, por exemplo. A localização global é ilustrada através do caso onde a posição inicial do robô é desconhecida e ele necessita determinar sua posição em um ambiente já mapeado. Uma situação ainda mais complexa pode ocorrer, conhecida como o rapto do robô, neste cenário o dispositivo robótico após um tempo de execução é colocado em outra posição do ambiente e necessita continuamente determinar sua posição, este caso, pode ser tratado como a recuperação de falha da localização global.

O ambiente ao qual o robô está inserido normalmente não mantém sua estrutura física estática, objetos podem mudar de lugar, portas podem estar abertas ou fechadas e pessoas podem transitar no local juntamente com o robô. Os ambientes estáticos são aqueles que não mudam, e possuem propriedades matemáticas ótimas, o que permite que sejam modelados de forma eficiente probabilisticamente. Porém, ambientes reais são dinâmicos, logo, os conceitos de dinâmica temporal ou persistente passam a ser importantes, pois algo que mudou

e persiste no tempo precisa ser incluso no mapa do ambiente, porém, uma pessoa movimentando-se no caminho ou portas, por exemplo, fazem parte da dinâmica temporária do ambiente e podem ser tratadas como ruído.

Na terceira dimensão, que diz respeito a implementação, no método passivo o sistema de localização apenas observa o robô funcionando e tenta estimar sua posição. Na sistema ativo, os algoritmos de controle são voltados para minimizar os erros e custos de deslocamento de tal forma que os resultados sejam melhores do que na prática passiva.

A localização pode ainda ser destinada para definir a posição de um único robô, ou grupos de robôs. Na localização singular o robô apenas define sua posição no ambiente, em contra partida na multi robótica cada dispositivo robótico do grupo define sua posição e em um momento oportuno compartilha com os demais.

Como visto, a localização é um grande desafio na robótica, conta com uma série de características que podem restringir ou trazer o problema para um alto nível de complexidade. Para resolver os desafios da localização podem ser utilizados algoritmos que fazem uso de filtros probabilísticos, que serão descritos em detalhes no Capítulo 3.

## 2.4 NAVEGAÇÃO

De acordo com Siegwart, Nourbakhsh e Scaramuzza (2011), navegação é uma das competências mais importantes em robôs móveis. Para que a navegação seja possível, se faz necessário que quatro atividades sejam desempenhadas com sucesso, sendo elas: percepção, localização, cognição e controle de movimento. Na percepção deve-se adquirir dados do ambiente, através dos dados o robô se localiza, por sua vez uma rota é definida por meio da cognição e finalmente com o uso do controle de movimento o robô locomove-se para o seu destino.

O controle de movimento é simplesmente o sistema que controla o hardware, fazendo com que os atuadores desempenhem a função desejada pelo sistema de navegação. Diferentes sistemas podem ser implementados como PID, Lógica Fuzzy, Redes Neurais Artificiais entre outros, ou ainda o caso onde ocorre a fusão das técnicas como apresentado por Ma, Yao e Wang (2016).

A cognição, também chamada de planejamento do caminho, consiste no problema de levar o robô de um ponto para outro no ambiente, definindo as tarefas que o dispositivo deve realizar para chegar ao des-

tino (DUCHON et al., 2014).

Com os avanços da robótica, a cognição agrega cada vez mais complicações como o tratamento da dinâmica do ambiente e incertezas. Segundo Fallah et al. (2013), sistemas de navegação humanos devem ser dinâmicos, refazendo rotas caso necessário. Essa afirmação também se aplica para a robótica móvel, pois em ambientes dinâmicos é comum realizar pequenas modificações na rota para desviar de obstáculos não existentes na representação do ambiente (e.g. pessoas, outros robôs).

O planejamento de rota pode ser dividido ainda em: global e local. No global são definidos o conjunto de coordenadas que devem ser visitadas para a chegada ao destino. Por sua vez, o planejador local tem foco nas informações recebidas do ambiente, possibilitando melhora na habilidade de desvio de obstáculos (CHEN et al., 2017). A combinação de ambos permite que sejam definidas as ações relacionadas ao controle robótico que devem ser realizadas para que o robô navegue entre as coordenadas fornecidas pelo planejador de rota global livre de colisões.

## 2.5 APLICAÇÕES DA ROBÓTICA MÓVEL

Segundo Siegwart, Nourbakhsh e Scaramuzza (2011), robôs são um caso de sucesso na indústria. Manipuladores, ou popularmente chamados braços robóticos, representam um setor no mercado de dois bilhões de dólares. Este tipo de dispositivo fixado em algum setor da linha de produção pode desenvolver as mais diversas tarefas com uma precisão inalcançável por seres humanos. Porém, possuem um alcance máximo onde seus movimentos podem ser úteis, que depende de sua fixação. Em contrapartida, um robô com a aptidão de navegar através do ambiente, permite que a mesma máquina desempenhe suas competências onde for necessário. Um exemplo deste tipo de robô pode ser observado na Figura 8.

Figura 8 – Robô manipulador móvel.



Fonte: Robotiq (2016)

Através da habilidade de navegar de forma autônoma, robôs móveis podem ser utilizados em ambientes inóspitos e insalubres para seres humanos, um exemplo dessa aplicação é o robô Curiosity desenvolvido para explorar Marte, com a capacidade de transpor obstáculos de até 75 centímetros, em média consegue percorrer 30 metros por hora, com base nos níveis de potência, deslizamento, inclinação do terreno, visibilidade e outras variáveis (NASA, 2015).

Figura 9 – Imagem do Robô Curiosity em Marte.



Fonte: NASA (2015)

Dentre as aplicações da robótica móvel, um exemplo que se popularizou nos últimos anos foi os robôs aspiradores de pó, ilustrado na Figura 10.

Figura 10 – Robô aspirador de pó.



Fonte: iRobot (2017)

Em ambientes fabris, hospitais ou centros de distribuição de pacotes, entre outros locais, robôs de carga podem ser utilizados. São exemplo desse tipo de aplicação os robôs do tipo AGV, do inglês *Autonomous Guided Vehicle*, que consistem em um mecanismo robótico seguidor de linha.

Figura 11 – Robô de transporte interno.



Fonte: Robotnik (2017a)

Na agricultura de precisão para auxiliar na extração de dados para maximizar o gerenciamento da produção, robôs móveis podem ser

utilizados de modo que não seja necessário um funcionário se deslocar e coletar os dados.

Figura 12 – Robô VinBot para controle de qualidade de plantações de uva.



Fonte: Robotnik (2017b)

O benefícios da robótica móvel se estendem também ao campo militar, robôs móveis manipuladores podem ser utilizados para desarmar bombas, ou como armamento operado remotamente, é o caso do robô TALON utilizado pelo exército dos Estados Unidos que pode ser utilizado em ambos os casos.

Figura 13 – Robô militar TALON.



Fonte: Army-Technology (2017)



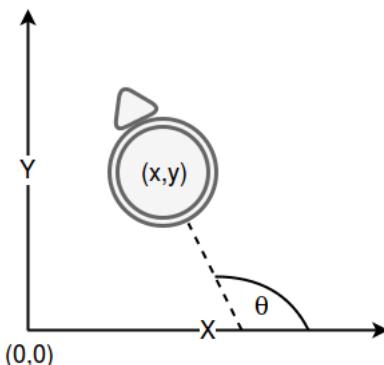
### 3 LOCALIZAÇÃO E MAPEAMENTO EM ROBÓTICA MÓVEL

Este capítulo descreve os algoritmos mais disseminados na comunidade de robótica móvel para as áreas de mapeamento e localização, trazendo ao final de cada seção um comparativo com o objetivo de expor pontos positivos e negativos de cada um dos algoritmos.

#### 3.1 CONSIDERAÇÕES A RESPEITO DO POSICIONAMENTO DO ROBÔ

Para critério de padronização, o posicionamento do robô em relação ao mapa deve ser considerado como apresentado na Figura 14 para todos os algoritmos apresentados na seções seguintes. Sendo a posição  $(x, y)$  representada pelo centro geométrico do robô e  $\theta$  o ângulo formado entre a frente do robô com o eixo  $x$  do mapa.

Figura 14 – Posicionamento do robô.



Fonte: Do Autor

#### 3.2 LOCALIZAÇÃO

Esta seção descreve as características de três algoritmos que são considerados base para as mais recentes implementações para localização baseada em mapas em robótica móvel.

### 3.2.1 Markov

Shoukry, Abdelfatah e Hammad (2009) afirmam que a localização de Markov endereça o problema de estimar o estado através de dados sensoriais. Siegwart, Nourbakhsh e Scaramuzza (2011) completam afirmando que o estado é dado através da posição  $(x, y, \theta)$ . É um caso específico de aplicação do filtro de Bayes para a localização robótica, que não mantém apenas uma probabilidade de qual célula o robô pode estar, mas sim uma distribuição probabilística para todas as possíveis posições, ou seja, estados. Vale ressaltar que para cada posição existe ainda um conjunto finito de possíveis valores para  $\theta$ . O algoritmo em sua versão discreta para mapas de grade, de acordo com Siegwart, Nourbakhsh e Scaramuzza (2011), é usualmente o mais utilizado. O Algoritmo 1 descreve a localização de Markov.

---

#### Algoritmo 1: LOCALIZAÇÃO DE MARKOV

---

```

Entrada:  $bel(x_{t-1}), u_t, z_t, m$ 
1 início
2   |   para todo  $x_t$  faz
3   |   |    $\bar{bel}(x_t) = \sum_{x_{t-1}} p(x_t|u_t, x_{t-1})bel(x_{t-1})$ 
4   |   |    $bel(x_t) = \eta p(z_t|x_t, m)\bar{bel}(x_t)$ 
5   |   fim
6 fim
7 retorna  $bel(x_t)$ 
```

---

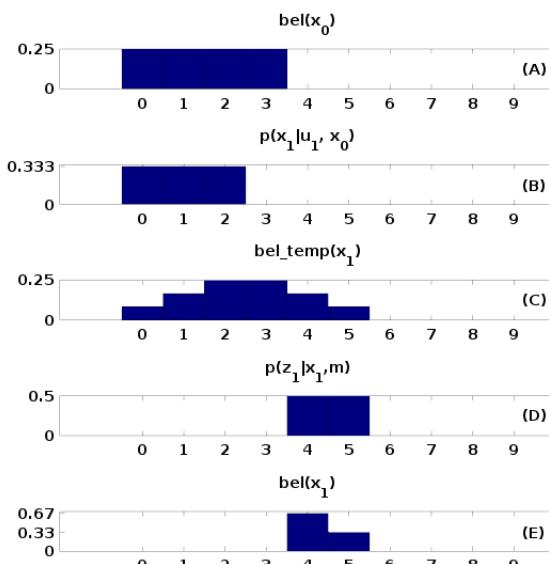
Nas linhas 3 e 4 do Algoritmo 1, o valor  $bel$ , do inglês *belief*, em português crença, é atualizado para cada possível posição chamada de  $x_t$ . O valor é chamado de crença pois não se pode extrair com exatidão a posição do robô como em um sistema de GPS, isso devido as incertezas geradas das leituras dos sensores. Consegue-se apenas estimar a melhor posição com base no valor da crença. Inicialmente o valor de crença deve ser uma distribuição uniforme para todos os estados.

Como entrada, o processo recebe o conjunto de crenças do passo anterior  $bel(x_{t-1})$ , o controle a qual o robô foi submetido  $u_t$ , as medidas realizadas  $z_t$  e o mapa de grade  $m$ .

O passo comumente chamado de atualização de predição é apresentado na linha 3. Este passo consiste em realizar a convolução entre  $p(x_t|u_t, x_{t-1})$  que representa a possível movimentação do robô e  $bel(x_{t-1})$ . Nesta etapa o estado atual é atualizado de acordo com os dados proprioceptivos (e.g. odometria e dados de controle) e a crença anterior.

A linha 4 compõem a atualização de percepção ou medida. O passo anterior resolveria apenas o problema do rastreamento de posição, este passo complementa o algoritmo para que possa executar a localização global. Através da regra de Bayes é computada a nova crença como função das medidas  $z_t$  e a crença temporária da linha anterior.

Figura 15 – Ilustração do funcionamento do algoritmo de localização de Markov.



Fonte: Do Autor

A Figura 15 exemplifica o algoritmo em uma dimensão. Em (A) está representada a crença atual da posição do robô, (B) está mostrando que o robô pode ter se movimentado de zero até três unidades para a direita, a atualização de predição é computada em (C) através da convolução entre (A) e (B), (D) ilustra a probabilidade do sensor ter realizado as medições que efetuou com base no estado calculado em (B) e no mapa (THRUN; BURGARD; FOX, 2005). Por fim, multiplica-se cada valor de (C) pelo seu respectivo em (D), para então normalizar os valores obtendo-se  $bel(x_1)$ .

### 3.2.2 Filtro de Kalman Estendido

A primeira proposta deste filtro, não estendida, foi apresentada por Kalman (1960) e por conta disso levou seu nome. Em sua versão básica, endereça a solução para o problema de estimar o estado de um processo discreto controlado por tempo, regido por uma equação de diferenças estocástica linear (WELCH; BISHOP, 1995).

Robôs móveis realizam também movimentos de translação, por conta disso, sua locomoção é regida por equações não lineares, isso implica que o filtro de Kalman puro não seria suficiente para efetuar a localização. Porém, no Filtro Estendido de Kalman, ou EKF do inglês *Extended Kalman Filter*, através de linearização de Taylor, é acrescentada a capacidade de endereçar a dinâmica não linear de movimento bem como das medições realizadas por seus sensores (KONG et al., 2006). A leitura dos sensores, o deslocamento do robô, bem como estimativa da nova posição é representada através de distribuições gaussianas. A proposta de Kalman (1960) não faz restrições quanto a representação do erro por gaussianas, porém, quando o erro é distribuído sobre gaussianas o filtro produz a probabilidade condicional exata, deste modo aumentando a exatidão da fase de correção.

Apesar de ser uma técnica relativamente antiga de filtragem, implementações atuais para a localização robótica fazem uso do filtro, como o caso de (SKOBELEVA; UGRINOVSKII; PETERSEN, 2016), (YUZHEN; QUANDE; BENFA, 2016) e (PAZOS-REVILLA; GUO; MACHIDA, 2016). O algoritmo será apresentado aqui em sua versão clássica para localização, utilizada como base para as implementações mais atuais, faz uso da fusão sensorial entre odometria e leitura de distâncias, meio pelo qual detecta-se pontos de referência no ambiente. A aplicação de Kalman em localização robótica oferece uma limitação, ou seja, ele é recomendado apenas para o caso do rastreamento de posição (THRUN; BURGARD; FOX, 2005). Vale ressaltar que este filtro devido a sua versatilidade pode ser utilizado também no mapeamento como será mostrado na Seção 3.3.2.

O algoritmo tem como entrada a média  $\mu_{t-1}$  e o desvio padrão  $\Sigma_{t-1}$  da gaussiana que presenta a posição atual, o vetor contendo o controle  $u_t$ , o conjunto de pontos de referência detectados  $z_t$  juntamente com a variável de correspondência  $c_t$  que representa o ponto de referência detectado em cada  $z_t^i$ , bem como o mapa topológico  $m$ .

---

**Algoritmo 2:** LOCALIZAÇÃO EKF
 

---

```

Entrada:  $\mu_{t-1}, \sum_{t-1}, u_t, z_t, c_t, m$ 
1 início
2    $\theta = \mu_{t-1, \theta}$ 
3    $G_t = \begin{pmatrix} 1 & 0 & -\frac{v_t}{\omega_t} \cos(\theta) + \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ 0 & 1 & -\frac{v_t}{\omega_t} \sin(\theta) + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\ 0 & 0 & 1 \end{pmatrix}$ 
4    $V_t = \begin{pmatrix} \frac{-\sin(\theta) + \sin(\theta + \omega_t \Delta t)}{\omega_t} & \frac{v_t(\sin(\theta) - \sin(\theta + \omega_t \Delta t))}{\omega_t^2} + \frac{v_t \cos(\theta + \omega_t \Delta t) \Delta t}{\omega_t} \\ \frac{\cos(\theta) - \cos(\theta + \omega_t \Delta t)}{\omega_t} & \frac{-v_t(\cos(\theta) - \cos(\theta + \omega_t \Delta t))}{\omega_t^2} + \frac{v_t \sin(\theta + \omega_t \Delta t) \Delta t}{\omega_t} \\ 0 & \Delta t \end{pmatrix}$ 
5    $M_t = \begin{pmatrix} \alpha_1 v_t^2 + \alpha_2 \omega_t^2 & 0 \\ 0 & \alpha_3 v_t^2 + \alpha_4 \omega_t^2 \end{pmatrix}$ 
6    $\bar{\mu}_t = \mu_{t-1} + \begin{pmatrix} -\frac{v_t}{\omega_t} \sin(\theta) + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos(\theta) - \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix}$ 
7    $\sum_t = G_t \sum_{t-1} G_t^T + V_t M_t V_t^T$ 
8    $Q_t = \begin{pmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & \sigma_s^2 \end{pmatrix}$ 
9   para pontos de referência  $z_t^i = (r_t^i, \phi_t^i, s_t^i)$  observados faça
10      $j = c_t^i$ 
11      $q = (m_{j,x} - \bar{\mu}_{t,x})^2 + (m_{j,y} - \bar{\mu}_{t,y})^2$ 
12      $\bar{z}_t^i = \left( \text{atan}2(m_{j,y} - \bar{\mu}_{t,y}, m_{j,x} - \bar{\mu}_{t,x}) - \bar{\mu}_{t,\theta} \right)$ 
13      $H_t^i = \begin{pmatrix} \sqrt{q} & & \\ -\frac{m_{j,x} - \bar{\mu}_{t,x}}{\sqrt{q}} & -\frac{m_{j,y} - \bar{\mu}_{t,y}}{\sqrt{q}} & 0 \\ \frac{m_{j,y} - \bar{\mu}_{t,y}}{q} & -\frac{m_{j,x} - \bar{\mu}_{t,x}}{q} & -1 \\ 0 & 0 & 0 \end{pmatrix}$ 
14      $S_t^i = H_t^i \sum_t [H_t^i]^T + Q_t$ 
15      $K_t^i = \sum_t [H_t^i]^T [S_t^i]^{-1}$ 
16      $\bar{\mu}_t = \bar{\mu}_t + K_t^i (z_t^i - \bar{z}_t^i)$ 
17      $\sum_t = (I - K_t^i H_t^i) \sum_t$ 
18   fim
19    $\mu_t = \bar{\mu}_t$ 
20    $\sum_t = \sum_t$ 
21    $p_{zt} = \prod_i \det(2\pi S_t^i)^{-\frac{1}{2}} \exp(-\frac{1}{2}(z_t^i - \bar{z}_t^i)^T [S_t^i]^{-1} (z_t^i - \bar{z}_t^i))$ 
22   fim
23 retorna  $\mu_t, \sum_t, p_{zt}$ 
  
```

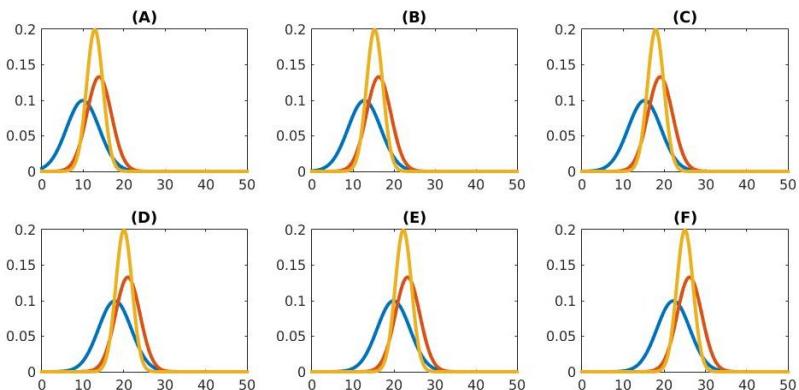
---

No algoritmo as linhas 3 e 4 calculam as Jacobianas necessárias para linearizar o modelo de movimentação. Através da linha 5 é determinada a covariância do ruído do controle. Nas linhas 6 e 7 é definida a média e o desvio padrão da gaussiana que define a posição do robô e

assim finaliza-se a etapa de predição do modelo de movimentação. Na linha 9 é atribuída a variância de cada um dos parâmetros de medição à variável  $Q_t$ . Da linha 10 até a 21 é realizada a predição do modelo de medição e a etapa de correção simultaneamente. Na linha 10 é atribuído a  $j$  o identificador do ponto de referência detectado na medição  $z_t^i$ . Entre 11 e 13 é atualizado o modelo de medição  $\bar{z}_t^i$  e a sua Jacobiana  $H_t^i$  para linearização, é importante ressaltar que a posição do ponto de referência está salva no mapa, por conta disso esta etapa é realizada com base na posição atual do robô e o mapa, para posteriormente ser utilizada em conjunto com a posição medida do ponto de referência. Então é computada a incerteza da medição  $S_t^i$  e posteriormente o ganho de Kalman  $K_t^i$ , o ganho é um fator chave, ele é responsável por definir se o filtro deve dar preferência as medidas ou as estimativas feitas na predição. Ao fim de cada ciclo é corrigida a média e o desvio padrão da posição do robô com base no ganho, na incerteza  $H_t^i$  nas medições efetuadas  $z_t^i$  e estimadas  $\bar{z}_t^i$ .

A Figura 16 apresenta a exemplificação do funcionamento do algorítimo em uma dimensão. Em todos os quadrantes a gaussiana menor representa a etapa de predição do modelo de movimentação com base em  $\mu_{t-1}$ ,  $\sum_{t-1}$  e  $u_t$ , já a predição do modelo de medição, gaussiana intermediária, é atualizada utilizando de  $z_t$ ,  $c_t$  e  $m$ , e por fim a etapa de correção (gaussiana maior) com base nas predições realizadas. É como se, com base na posição anterior, o modelo de movimentação gerasse uma nova gaussiana indicando a posição atual e com base nos pontos de referência detectados o modelo de medição produzisse uma gaussiana indicando a possível posição, então com base nas duas a posição real é estimada.

Figura 16 – Ilustração do funcionamento do algoritmo de localização EKF.



Fonte: Do Autor

### 3.2.3 Monte Carlo

De acordo com Aini, Jati e Sunarya (2016) o algoritmo de localização de Monte Carlo, ou MCL do inglês *Monte Carlo Localization*, é um variação de Markov baseada em filtros de partículas. Bukhori e Ismail (2016) afirmam que o MCL é caracterizado por convergência. Inicialmente distribui aleatoriamente as partículas ou também chamadas hipóteses por todo o espaço de solução, então gradativamente com a movimentação do robô e leituras sensoriais as partículas convergem para a região mais provável de posição do robô. Cada partícula representa uma posição, como o número de partículas é menor que o espaço de possíveis posições, além de que a cada hipótese é atribuído um peso correspondente a crença, o número de crenças a serem computadas neste algoritmo é menor do que no anterior.

---

**Algoritmo 3:** LOCALIZAÇÃO DE MONTE CARLO
 

---

```

Entrada:  $X_{t-1}, u_t, z_t, m$ 
1 início
2    $\overline{X}_t = X_t = \emptyset$ 
3   para  $1 \leq m \leq M$  faca
4      $x_t^{[m]} = f(u_t, x_{t-1}^{[m]})$ 
5      $\omega_t^{[m]} = g(z_t, x_t^{[m]}, m)$ 
6      $\overline{X}_t = \overline{X}_t + \langle x_t^{[m]}, \omega_t^{[m]} \rangle$ 
7   fim
8   para  $1 \leq i \leq M$  faca
9     normalize  $\omega_t^{[i]}$ 
10     $X_t = X_t + \langle x_t^{[i]}, \omega_t^{[i]} \rangle$ 
11  fim
12 fim
13 retorna  $X_t$ 
  
```

---

Os dados de entrada do processo são o conjunto de hipóteses do passo anterior  $X_{t-1}$ , o controle a qual o robô foi submetido  $u_t$ , as medidas realizadas  $z_t$  e o mapa de grade  $m$ .

Na linha 2 primeiramente se faz com que o conjunto temporário  $\overline{X}_t$  e o final  $X_t$  sejam vazios, sem apagar o conjunto de partículas do tempo anterior  $X_{t-1}$ .

Sendo  $M$  o número total de partículas, o primeiro laço itera para cada partícula contida no conjunto pertencente a  $X_{t-1}$ . Na linha 4 é gerado o estado hipotético  $x_t^{[m]}$  (partícula) atual, como base no anterior e no vetor de controle. Vale ressaltar que a função  $f(u_t, x_{t-1}^{[m]})$  calcula a nova posição de forma probabilística proporcional a função já vista anteriormente  $p(x_t|u_t, x_{t-1})$ . O conjunto de partículas obtido ao final deste laço representa então a crença temporária  $\overline{bel}(x_t)$  também vista anteriormente.

A linha 5 do algoritmo, de forma análoga a vista anteriormente, é calculada a probabilidade de cada partícula ter realizado a leitura  $z_t$ , com base na sua posição e no mapa. Cada valor resultante desta parte serve como um peso para determinar a representatividade da hipótese para a posição do robô. O conjunto dos pesos  $\omega_t^{[m]}$  representa uma aproximação do que foi visto anteriormente como o conjunto de crenças no tempo  $t$ , formalmente  $bel(x_t)$ , porém não normalizada.

Na linha 6, o par, partícula e seu respectivo peso, são adicionados ao conjunto de hipóteses, finalizando a iteração do laço.

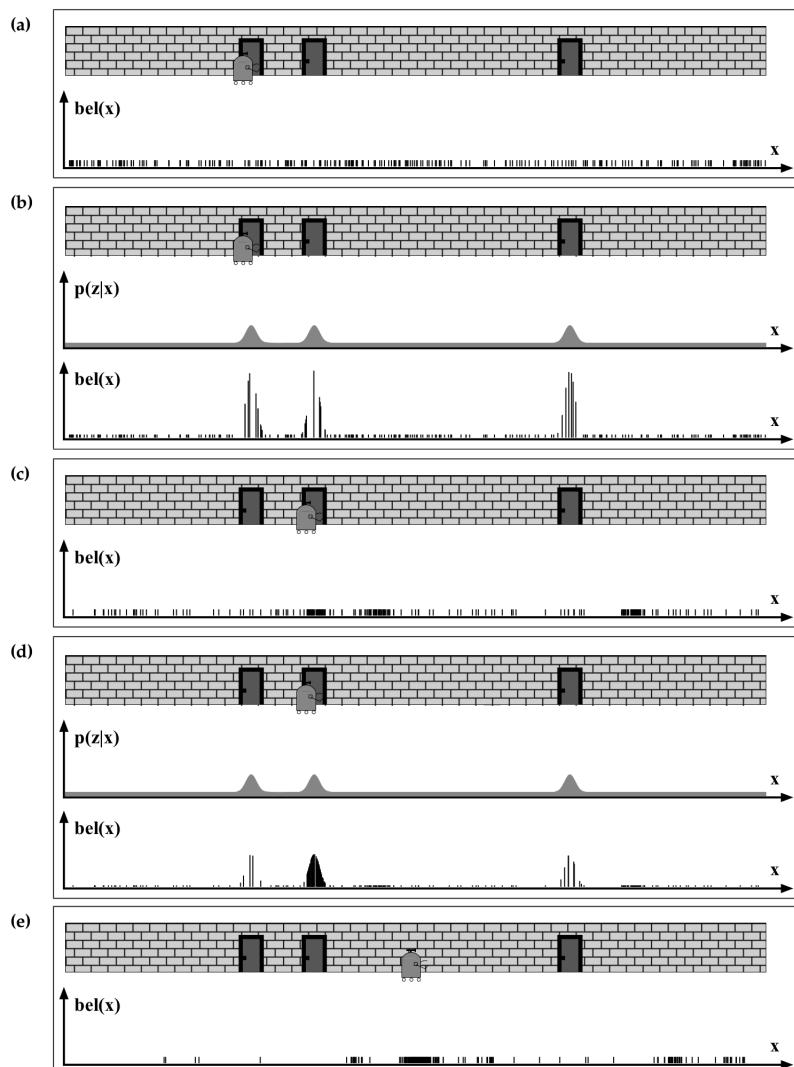
Entre as linhas 8 e 11 um laço de repetição executa para cada

partícula pertencente ao conjunto temporário  $\overline{X_t}$  o processo chamado de reamostragem. Basicamente é aplicada a normalização no peso de cada partícula, com base no peso normalizado um novo conjunto de partículas é criado, fazendo com que mais hipóteses sejam inseridas aos arredores das partículas com maior peso, podendo-se ter partículas duplicadas. As regiões com maior número de partículas são as regiões da possível localização real do robô.

A Figura 17 apresenta o funcionamento do algoritmo em uma dimensão. Considerando a função  $g(z_t, u_t, x_t^{[m]})$ , ou peso, como a probabilidade de estar em frente a uma porta, bem como, as partículas representadas por barras verticais onde quanto maior seu peso, maior é seu comprimento. Pode-se perceber que em (A) as partículas estão inicialmente dispostas aleatoriamente no mapa, com os pesos formando uma distribuição uniforme. Considerando que em (A) o robô tenha realizado um movimento e seus pesos já tivessem sido normalizados, em (B) é atribuído o peso de cada partícula de acordo com a leitura dos sensores, fazendo assim com que as partículas próximas as portas recebam pesos maiores. Em (C),  $bel(x)$  é o resultado da normalização após uma nova movimentação do robô, onde pode-se perceber que as partículas estão mais agrupadas onde anteriormente tinham maior peso. Em (D) ocorre novamente a atualização dos pesos de acordo com a leitura dos sensores. Finalmente em (E) pode-se observar o estágio de convergência onde a maioria das partículas após a normalização circundam a região de posição real do robô.

O MCL tem a capacidade de localização global como o algoritmo de Markov puro e além disso a resolução para o problema do rapto do robô. As partículas aleatórias afastadas da região de localização real podem ser utilizadas para detectar este caso, ou ainda mecanismos mais eficientes podem ser implementados, como o de Bukhori e Ismail (2016), para recuperar o sistema nesta situação.

Figura 17 – Ilustração do funcionamento do algoritmo de localização de Monte Carlo.



Fonte: Thrun, Burgard e Fox (2005)

### 3.2.4 Comparativo

Foram apresentados os algoritmos de Markov, EKF e MCL para localização robótica através de mapas. Por meio da análise do funcionamento e utilização dos recursos foi elaborado um comparativo baseando-se em Thrun, Burgard e Fox (2005), presente na Tabela 1.

Com relação a memória utilizada, considera-se o algoritmo de Markov como o que mais utiliza memória pois além de manter o mapa, necessita para cada uma das células manter a crença. Já no MCL a crença não será mantida para cada item da grade, apenas para a partícula ao qual que está alocada. Por sua vez, o EKF, por utilizar mapas topológicos, possui o menor consumo de memória.

Em eficiência de tempo, devido a quantidade de operações a serem realizadas o algoritmo de Markov dentre os três apresentará o menor desempenho. Como o EKF apenas realiza o processo de rastreamento da posição, o caso mais simples da localização, ele se torna mais rápido, porém, sendo dependente da velocidade de leitura dos pontos de referência. Como o MCL funciona de forma semelhante a Markov, porém, realiza cálculos apenas para as partículas, seu desempenho é intermediário.

Tabela 1 – Comparativo entre algoritmos de localização.

	<b>Markov</b>	<b>EKF</b>	<b>MCL</b>
<b>Tipo de Mapa</b>	Grade	Topológico	Grade
<b>Tipo de Percepção</b>	Distâncias	Pontos de Referência	Distâncias
<b>Consumo de Memória</b>	Alto	Baixo	Médio
<b>Eficiência em Tempo</b>	Baixa	Alta	Média
<b>Localização Global</b>	Sim	Não	Sim

### 3.3 MAPEAMENTO

Nesta seção três algoritmos de mapeamento em robótica móvel são explanados, sendo estes elementos base para as implementações que compõem o estado da arte.

### 3.3.1 Grade

De acordo com Hadji et al. (2015) o mapeamento de grade teve sua primeiras implementação propostas por Moravec e Elfes (1985) e nas últimas duas décadas foi a topologia de mapeamento dominante. O algoritmo aqui apresentado é aplicado aos casos em que o robô utiliza sensores medidores de distância (e.g. SICK LMS200, Ultrassom) bem como a estimativa de posição tem controle de ruído externo ao algoritmo, afinal ele não oferece tratamento de ruídos para o posicionamento do robô.

Cada célula varia seu valor entre 0 e 1, ficando a critério do desenvolvedor definir qual o valor probabilístico que considera cada item da grade como ocupado, livre ou estado desconhecido. Além disso um valor inicial deve ser dado a todas as células do mapa, de acordo com Thrun, Burgard e Fox (2005) este valor deve ser  $pLog_0$ , representando um estado desconhecido. Sendo  $p(m_i = 1)$  na Equação 3.1 a probabilidade definida para se considerar a célula ocupada e  $p(m_i = 0)$  o valor definido para considerar o estado como livre.

$$pLog_0 = \log \left( \frac{p(m_i = 1)}{p(m_i = 0)} \right) = \log \left( \frac{p(m_i = 1)}{1 - p(m_i = 1)} \right) \quad (3.1)$$

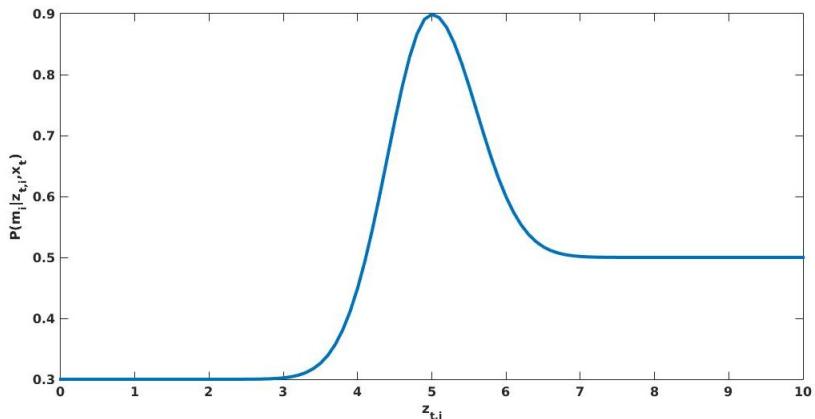
Cada célula da grade é atualizada através da função probabilística modelo inverso sensorial (MIS), responsável por capturar as incertezas do modelo espacial. Essa função calcula  $p(m_i|z_t, x_t)$ , ou seja, qual a possibilidade, de acordo com a posição atual e a leitura do sensor, da célula  $m_i$  estar ocupada. Por conta disso recebe o nome de modelo inverso sensorial, pois realiza a tarefa inversa ao do modelo sensorial que é definir a probabilidade de uma medida ocorrer de acordo com o mapa e a posição atual. Assim, incrementalmente é atualizado o valor do estado de cada célula quantas vezes forem necessárias durante o processo de mapeamento.

O MIS retorna um valor probabilístico obedecendo a Equação 3.2 para que os efeitos dos ruídos sensoriais sejam minimizados através da função logarítmica. Uma das possíveis implementações para a função  $p(m_i|z_t, x_t)$  é apresentada por Kaufman et al. (2016). A Figura 18 apresenta um exemplo onde foi realizada uma leitura de distância com valor de cinco metros, no entanto, o sensor tem alcance de até dez, desta forma para as células antes de cinco metros o valor de MIS é próximo a 0,3 (neste exemplo, valor aceitável para células livres), as células ao

entorno de cinco metros recebem maior valor de MIS representando como ocupadas, por fim, para as células com distâncias superiores a cinco metros é fornecido um valor intermediário que representa estado desconhecido.

$$pLog = \log \left( \frac{p(m_i|z_t, x_t)}{1 - p(m_i|z_t, x_t)} \right) \quad (3.2)$$

Figura 18 – Ilustração de exemplo de modelo inverso sensorial.



Fonte: Do Autor

---

#### Algoritmo 4: MAPEAMENTO DE GRADE

---

```

Entrada: { $pLog_{t-1,i}$ },  $x_t$ ,  $z_t$ 
1 início
2   | para todas células faz
3   |   | se célula no campo de percepção de  $z_t$  então
4   |   |   |  $pLog_{t,i} = pLog_{t-1,i} + MIS(m_i, x_t, z_t) - pLog_0$ 
5   |   |   | senão
6   |   |   |   |  $pLog_{t,i} = pLog_{t-1,i}$ 
7   |   |   | fim
8   |   | fim
9   | fim
10 retorna { $pLog_{t,i}$ }

```

---

### 3.3.2 EKF-SLAM

Os algoritmos que resolvem o problema de SLAM, do inglês *Simultaneous Localization and Mapping*, em português localização e mapeamento simultâneo, são caracterizados por usarem as informações do ambiente para eliminar ruídos aleatórios e o erro acumulativo do sistema, para assim produzirem mapas consistentes, sendo o EKF-SLAM o primeiro algoritmo voltado a resolução deste propósito (WANG et al., 2013).

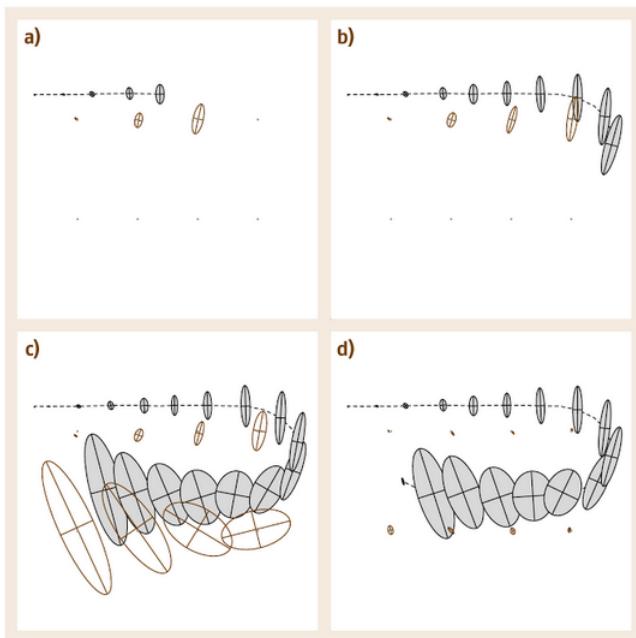
Existe uma diferença chave entre o EKF-SLAM e os algoritmos voltados apenas a localização. Para realizar o mapeamento o algoritmo deve também estimar a posição de todos os pontos de referência encontrados no caminho. É necessário adicionar a localização dos pontos de referência detectados ao vetor de estados (THRUN; BURGARD; FOX, 2005).

A versão apresentada irá considerar que existe um número fixo de pontos de referência a serem detectados, determinado pela constante  $N$ . Este seria um possível caso de ambiente onde os pontos de referência a serem detectados são virtuais, ou seja, colocados no ambiente especificamente para facilitar o mapeamento e localização, como visto na implementação de Jung et al. (2017).

A entrada do algoritmo é muito semelhante quando o mesmo é aplicado à localização. Sendo novamente  $\mu_{t-1}$  a média, ou seja, posição do centro da gaussiana, o desvio padrão  $\Sigma_{t-1}$  da posição, o vetor contendo o controle  $u_t$ , o conjunto de pontos de referência detectados  $z_t$  juntamente com a variável de correspondência  $c_t$ . Na primeira iteração do algoritmo deve-se fornecer uma previsão da posição inicial do robô no mapa, pois, assim como na localização o filtro de Kalman aqui empregado realiza rastreamento da posição durante o mapeamento.

A Figura 19 ilustra o funcionamento do EKF-SLAM, as elipses preenchidas representam a posição do robô, as não preenchidas a posição dos pontos de referência detectados. A medida que o robô locomove-se a incerteza quanto a sua posição aumenta, juntamente com a da posição dos pontos de referência, porém, em (d), quando o robô encontra novamente o primeiro ponto de referência a incerteza diminui, sendo essa uma das principais características deste algoritmo.

Figura 19 – Ilustração do mapeamento EKF.



Fonte: Siciliano e Khatib (2007)

---

**Algoritmo 5:** EKF-SLAM
 

---

**Entrada:**  $\mu_{t-1}, \sum_{t-1}, u_t, z_t, c_t$

1 **início**

$$2 \quad \theta = \mu_{t-1, \theta}$$

$$3 \quad F_x = \begin{pmatrix} 1 & 0 & 0 & zeros(3N) \\ 0 & 1 & 0 & zeros(3N) \\ 0 & 0 & 1 & zeros(3N) \end{pmatrix}$$

$$4 \quad \bar{\mu}_t = \mu_{t-1} + F_x^T \begin{pmatrix} -\frac{v_t}{\omega_t} \sin(\theta) + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos(\theta) - \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix}$$

$$5 \quad G_t = I + F_x^T \begin{pmatrix} 0 & 0 & -\frac{v_t}{\omega_t} \cos(\theta) + \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ 0 & 0 & \frac{v_t}{\omega_t} \sin(\theta) + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\ 0 & 0 & 0 \end{pmatrix} F_x$$

$$6 \quad \sum_t = G_t \sum_{t-1} G_t^T + F_x^T R_t F_x$$

$$7 \quad Q_t = \begin{pmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & \sigma_s^2 \end{pmatrix}$$

8 **para** pontos de referência  $z_t^i = (r_t^i, \phi_t^i, s_t^i)$  observados **faz**

$$9 \quad j = c_t^i$$

10 **se** ponto de referência  $j$  novo **então**

$$11 \quad \left( \begin{array}{c} \bar{\mu}_{j,x} \\ \bar{\mu}_{j,y} \\ \bar{\mu}_{j,s} \end{array} \right) + \left( \begin{array}{c} \bar{\mu}_{t,x} \\ \bar{\mu}_{t,y} \\ s_t^i \end{array} \right) + \left( \begin{array}{c} r_t^i \cos(\phi_t^i + \bar{\mu}_{t,\theta}) \\ r_t^i \sin(\phi_t^i + \bar{\mu}_{t,\theta}) \\ 0 \end{array} \right)$$

12 **fim**

$$13 \quad \delta = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix}$$

$$14 \quad q = \delta^T \delta$$

$$15 \quad \bar{z}_t^i = \begin{pmatrix} \sqrt{q} \\ atan2(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \\ \bar{\mu}_{j,s} \end{pmatrix}$$

16  $F_{x,j} =$

$$\begin{pmatrix} 1 & 0 & 0 & zeros(3j-3) & 0 & 0 & 0 & zeros(3N-3j) \\ 0 & 1 & 0 & zeros(3j-3) & 0 & 0 & 0 & zeros(3N-3j) \\ 0 & 0 & 1 & zeros(3j-3) & 0 & 0 & 0 & zeros(3N-3j) \\ 0 & 0 & 0 & zeros(3j-3) & 1 & 0 & 0 & zeros(3N-3j) \\ 0 & 0 & 0 & zeros(3j-3) & 0 & 1 & 0 & zeros(3N-3j) \\ 0 & 0 & 0 & zeros(3j-3) & 0 & 0 & 1 & zeros(3N-3j) \end{pmatrix}$$

$$17 \quad H_t^i = \frac{1}{q} \begin{pmatrix} -\sqrt{q} \delta_x & -\sqrt{q} \delta_y & 0 & \sqrt{q} \delta_x & \sqrt{q} \delta_y & 0 \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x & 0 \\ 0 & 0 & 0 & 0 & 0 & q \end{pmatrix} F_{x,j}$$

$$18 \quad K_t^i = \sum_t [H_t^i]^T (H_t^i \sum_t [H_t^i]^T + Q_t)^{-1}$$

$$19 \quad \bar{\mu}_t = \bar{\mu}_t + K_t^i (z_t^i - \bar{z}_t^i)$$

$$20 \quad \sum_t = (I - K_t^i H_t^i) \sum_t$$

21 **fim**

$$22 \quad \mu_t = \bar{\mu}_t$$

$$23 \quad \sum_t = \sum_t$$

24 **fim**

25 **retorna**  $\mu_t, \sum_t$

---

Em alguns pontos do algoritmo uma função chamada *zeros* está sendo chamada, deve-se considerar que ela retorna um vetor de zeros do tamanho especificado no parâmetro.

Na linha 3 é definida a matriz  $F_x$  responsável por garantir que apenas a posição do robô seja modificada no vetor de estados na etapa de atualização de movimento (linha 3 até linha 6). As linhas 4 e 5 calculam a média e a covariância da localização de acordo com o modelo de movimento. Então na linha 6 é calculada a variância da gaussiana que define a posição, finalizando a etapa da predição do modelo de movimento. Vale ressaltar que nesta linha, apesar da variável  $R_t$  não estar sendo calculada neste algoritmo, ela equivale ao termo  $V_t M_t V_t^T$  presente no Algoritmo 2 na linha 8. Entre 10 e 12 caso um novo ponto de referência seja detectado ele é adicionada a sua respectiva posição no ambiente em seu espaço reservado no vetor de estados juntamente com a assinatura que identifica o tipo de ponto de referência. Nas linhas de 13 a 15 é definida as leituras esperadas para distância entre o robô e o ponto de referência, ângulo em relação ao mapa, e a assinatura do ponto de referência. Nas linhas seguintes é definido então o ganho de Kalman  $K_t^i$ , para desta forma, atualizar o valor da média  $\bar{\mu}_t$  e do desvio padrão  $\sum_t$  que representa a gaussiana de posição do robô.

### 3.3.3 Filtro de Partículas Rao-Blackwellized

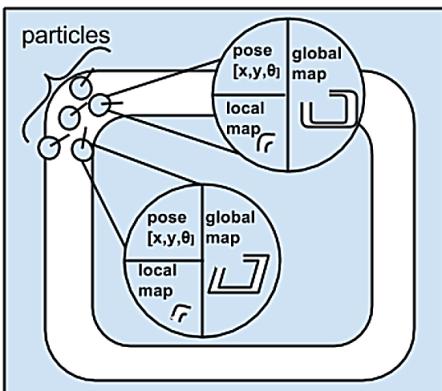
Este filtro foi introduzido pela primeira vez como solução para o SLAM por Murphy (1999) e Doucet et al. (2000) recebendo este nome pois é implementado a partir do teorema *Rao-Blackwell*, que segundo Galili e Meilijson (2016), oferece um procedimento para converter um estimador bruto imparcial de um parâmetro qualquer, em um estimador ótimo.

Este tipo de implementação tem um principal problema, sua complexidade no que diz respeito a quantidade de partículas necessárias para construir um mapa preciso, sendo que reduzir o número de partículas é um dos maiores desafios dessa família de algoritmos de acordo com Grisetti, Stachniss e Burgard (2007). Sendo que nas implementações mais comuns é mantido um mapa por partícula (JO et al., 2013), o que aumenta demasiadamente o uso de memória.

De acordo com Yatim e Buniyamin (2015), neste tipo de algoritmo de mapeamento, cada partícula contém uma hipótese quanto a posição do robô, e assume que está correta. Sendo que, cada partícula mantém um mapa com base em sua posição e as medições realizadas

pelo robô (e.g. pontos de referência ou distâncias), fazendo com que algumas partículas tenham acurácia (medida através do peso) no mapeamento melhor do que outras e ao final do processo a partícula com o maior peso possua o melhor mapa. A Figura 20 ilustra o funcionamento do algoritmo.

Figura 20 – Ilustração do mapeamento de grade utilizando partículas.



Fonte: Yatim e Buniyamin (2015)

Existem implementações que utilizam filtro de partículas apenas para determinar o caminho percorrido pelo robô, como é o caso do FastSLAM, proposto por Montemerlo et al. (2002), que possui ainda a característica de poder ser utilizado em mapas de grade (REINEKING; CLEMENS, 2013) e mapas topológicos (AMANDA; JATI; SUNARYA, 2016). Entretanto, o segundo utiliza EKF para estimar a posição dos pontos de referência. No DP-SLAM, apresentado por Eliazar e Parr (2003), a posição e o mapa passam a ser representados pelo filtro e utilizam uma estrutura de dados em árvore para otimizar o uso de memória. As implementações propostas para mapeamento utilizando de filtro de partículas derivam do algoritmo chamado *Sampling Importance Resampling* (SIR), ou em tradução livre Amostragem, Importância e Reamostragem, descrito no Algoritmo 6.

---

**Algoritmo 6:** SAMPLING IMPORTANCE RESAMPLING
 

---

```

Entrada:  $\{x_{t-1}\}, u_t, z_t, m$ 
1 início
2    $\{\bar{x}_t\} = amostragem(\{x_{t-1}\}, u_t)$ 
3   para todas as partículas  $i$  faz
4      $\omega_t^i = atualizar\_peso(z_t, \bar{x}_t^{[i]}, m_{t-1}^{[i]})$ 
5      $m_t^{[i]} = atualizar\_mapa(z_t, \bar{x}_t^{[i]}, m_{t-1}^{[i]})$ 
6   fim
7    $\{x_t\} = reamostragem(\{\bar{x}_t\}, \{\omega_t\})$ 
8 fim
9 retorna  $X_t$ 
  
```

---

Pode-se perceber que o algoritmo de localização por Monte Carlo segue a mesma linha de funcionamento desta implementação. Na linha 2 a nova posição das partículas é atualizada de acordo com o modelo de movimento probabilístico adotado pelo desenvolvedor. Entre as linhas 3 e 6 é atualizado o peso, ou seja a importância, e o mapa de cada partícula, um método para atualizar o peso é proposto por Schröter, Böhme e Gross (2007), a atualização do mapa, por sua vez, pode ser realizada para mapas de grade como no algoritmo já visto. Por fim, a etapa de reamostragem pode ser realizada da forma já descrita na localização por Monte Carlo.

### 3.3.4 Comparativo

O comparativo para algoritmos de mapeamento apresentado na Tabela 2 foi elaborado, de forma análoga ao de localização, baseando-se nas constatações de Thrun, Burgard e Fox (2005).

O consumo de memória é alto para o mapeamento através de filtros de partículas, pois, como visto, para cada partícula um mapa de grade é elaborado, e mesmo nas implementações utilizando mapas topológicos o consumo ainda seria maior, pois utiliza além das partículas (no procedimento de localização) o mapeamento por EKF para os pontos de referência. Considera-se o consumo baixo para EKF, pois utiliza de mapas topológicos, onde o número de pontos de referência são menores do que a quantidade células em mapas de grades. Sendo considerado, intermediário, o consumo de memória por meio do mapeamento de grade puro, devido ao fato de manter-se apenas um mapa de grade.

A filtragem de localização é um fator importante devido a im-

precisão contida nas medições de movimentação (e.g. odometria), o algoritmo de grade puro é o único com a desvantagem de não filtrar o posicionamento, ele considera que as posições não contém ruído, sendo assim se faz necessário um mecanismo externo para garantir a estimativa da localização com erro minimizado.

A utilização em ambientes dinâmicos é considerada limitada para os três algoritmos, devido ao fato de que nenhum leva em consideração a dinâmica do ambiente diretamente. Entretanto, pode-se implementar modelos de medição para qualquer um dos algoritmos que levem em consideração a detecção de leituras inválidas devido ao dinamismo do ambiente.

Tabela 2 – Comparativo entre algoritmos de mapeamento.

	<b>Grade</b>	<b>EKF</b>	<b>Partículas</b>
<b>Tipo de Mapa</b>	Grade	Topológico	Grade
<b>Tipo de Percepção</b>	Distâncias	Pontos de Referência	Distâncias
<b>Consumo de Memória</b>	Médio	Baixo	Alto
<b>Filtragem de Localização</b>	Não	Sim	Sim
<b>Ambientes Dinâmicos</b>	Limitado	Limitado	Limitado

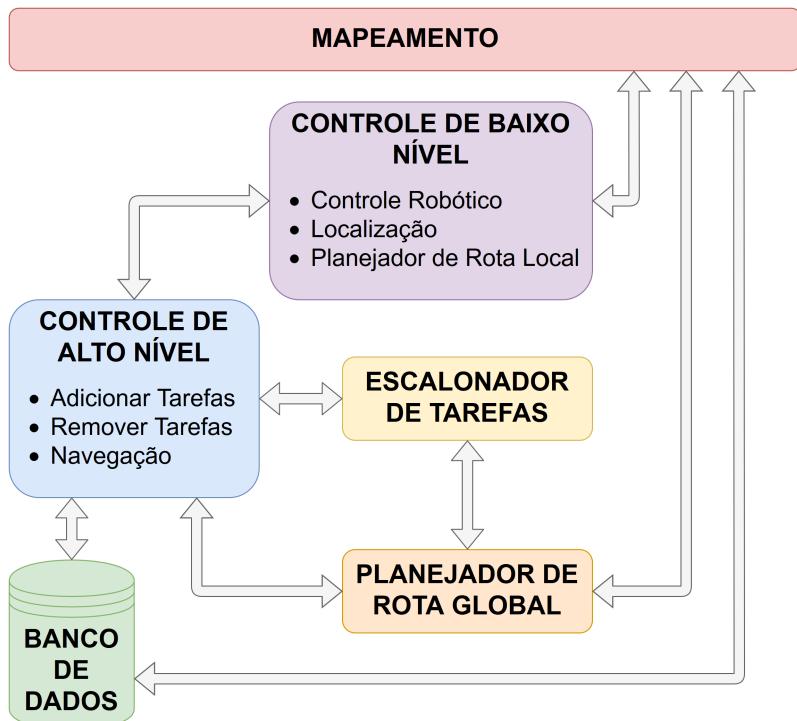
## 4 ARQUITETURA PROPOSTA PARA UM SISTEMA DE LOGÍSTICA HOSPITALAR

Neste capítulo a arquitetura desenvolvida para um sistema de logística hospitalar será apresentada inicialmente de forma geral, com o intuito de exibir o fluxo de execução do sistema através de seis elementos. Na sequência cada um de seus componentes serão aprofundados a fim de demonstrar como desempenham suas funções.

### 4.1 ARQUITETURA GERAL

A Figura 21 representa a visão macro da arquitetura proposta para um sistema de logística hospitalar dividida em seis componentes.

Figura 21 – Diagrama da arquitetura proposta.



Fonte: Do Autor

A primeira tarefa a ser executada para que o sistema possa funcionar no ambiente dinâmico e interno selecionado é o mapeamento. Neste trabalho, o mapa obtido deste processo é uma imagem semelhante a uma planta baixa, que posteriormente é utilizado como mapa de grade por outros elementos do sistema. O banco de dados armazena o mapa e os pontos na imagem que representam lugares que o robô pode visitar. Por sua vez, o planejador de rota global faz uso do mapa para definir o conjunto de coordenadas que deverão ser alcançadas para que o robô navegue de um ponto a outro no ambiente. O controle de baixo nível também faz uso do mapa utilizando-o em seu sistema de localização, onde, a partir de comparações entre o mesmo e as leituras obtidas do ambiente consegue refinar o posicionamento atual.

O banco de dados além de armazenar mapas e os lugares pertencentes a estes mapas, também é responsável por armazenar as tarefas. Através dele o usuário pode adicionar, remover e ser notificado quanto ao status atual das tarefas.

Um tarefa terá ao menos um lugar ao qual o robô deverá visitar, sendo assim é necessário um mecanismo que planeje uma rota eficiente entre a localização atual e o destino. O elemento encarregado de realizar esta atividade é o planejador de rota global.

Para que o robô possa navegar em um ambiente mapeado é necessário que ele possua um sistema de localização, que possa obter os dados provenientes dos sensores, bem como uma forma para controlar seus atuadores permitindo a navegação e, além disso, algum meio para interfacear o controle de alto nível com todas essas funcionalidades, preferencialmente de forma simples. Esta função é desempenhada pelo controle de baixo nível.

O escalonador de tarefas é o responsável por determinar com base em uma função custo, neste trabalho sendo composta pela distância a ser percorrida (calculada pelo planejador de rota global) e prioridade, qual tarefa deve ser executada.

A troca de informações entre os elementos do sistema é possibilitada pelo controlador de auto nível. Exemplificando, tarefas são inseridas e removidas do banco de dados, o controlador de auto nível recebe essa informação e notifica o usuário modificado o status das tarefas, em seguida as insere no escalonador de tarefas para que possa descobrir qual será o próximo local no ambiente que deverá visitar, e/ou tarefa, posteriormente com o lugar definido requisita ao planejador de rota global que compute a rota, para então enviar as coordenadas uma a uma para o controlador de baixo nível para que o planejador de rota local leve o robô até o destino. Em seguida, caso não receba mais in-

formações do banco de dados o sistema itera até que todas as tarefas tenham sido atendidas. Caso mais tarefas sejam inseridas ou removidas, quando o robô completar a rota até um local, o processo reinicia e uma nova sequência de execução é obtida. Cabe ressaltar que um trecho de tarefa talvez venha a ser executado posteriormente caso ela seja preemptada.

As seções a seguir abordam cada um dos seis elementos da Figura 21 detalhadamente.

## 4.2 MAPEAMENTO

Devido a forma simples, porém eficiente em se obter um mapa de grade e a disponibilidade de *drivers* para localização no servidor robótico *Player*, escolheu-se este tipo de mapeamento.

Como foi abordado no Capítulo 3, este tipo de mapa pode tornar-se impreciso devido a presença de ruído nos sensores medidores de distância, assim como na odometria. Então, o emprego de filtros se fez necessário para que o mapa possa ser construído com sucesso. Mapas de grade construídos utilizando filtros de partículas apresentam bons resultados em relação aos que não fazem uso de filtro algum. A implementação apresentada por Eliazar e Parr (2003) para mapeamento está disponível de forma livre, beneficiando-se deste fato, foi utilizada a aplicação desenvolvida pelos autores do artigo para construir o mapa que o robô viria a utilizar em todos os demais processos necessários para execução de tarefas.

Através do controle manual do Pionner P3-DX, utilizando *joystick*, o robô percorreu todo o andar do bloco a ser mapeado e os dados de distância provenientes do sensor SICK LMS200 e odometria foram aquisitados para que posteriormente o mapa fosse construído. Uma vez que o mapeamento é realizado através dos dados após a aquisição dos mesmos, o valor padrão da aplicação para densidade de pixels na imagem (35 pixels por metro) foi utilizado, sem a preocupação do impacto no desempenho. Obteve-se êxito na concepção do mapa, que precisou apenas que seu esquema de cores fosse reajustado para atender o padrão utilizado nos *drivers* do *Player*.

O mapa construído para este trabalho consiste em um mapa métrico, interno de alta escala. Além disso, sua percepção foi do tipo inferência de única sugestão utilizando conjunto de pontos. Ademais, para garantir a coerência temporal do mapa, durante o processo de aquisição dos dados referentes ao mapeamento, configurou-se a veloci-

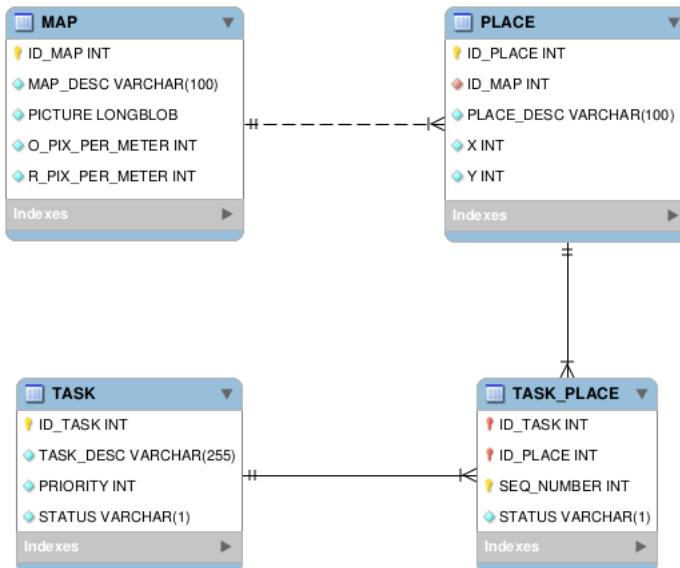
dade máxima de navegação para 0.5 m/s e foram efetuadas dez leituras por segundo.

#### 4.3 BANCO DE DADOS

O diagrama de entidade e relacionamento do banco de dados utilizado pelo sistema é apresentado na Figura 22. Como pode ser observado, a base de dados é muito simples, contando apenas com dados referentes ao mapa (tabela *MAP*), lugares pertencentes ao mapa (tabela *PLACE*), tarefas (tabela *TASK*) e lugares que devem ser visitados para completar a tarefa (tabela *TASK\_PLACE*).

A tabela *MAP* armazena um identificador, descrição (e.g. Bloco A, terceiro piso), a imagem do mapa, quantidade de pixels por metro original da imagem, e a que deve ser utilizada pelo planejador de rota global. Os dois últimos campos serão melhor explanados na Seção 4.7.

Figura 22 – Banco de dados.



Fonte: Do Autor

Para representar um lugar, a tabela *PLACE* faz uso de um identificador, o identificador do mapa ao qual pertence, descrição (e.g. Sala

303) e as coordenadas horizontal e vertical do local na imagem relativos a quantidade de pixels por metro original.

Uma tarefa, tabela *TASK*, possui identificador, descrição (e.g. levar desfibrilador), prioridade e status. Os campos utilizados por uma tarefa no banco de dados, e demais utilizados apenas em memória serão abordados na Subseção 4.6.1.

Por fim, a lista de lugares a serem visitados por uma tarefa é dada pela tabela *TASK\_PLACE*, onde a sequência de visitas é definida através do campo *SEQ\_NUMBER*.

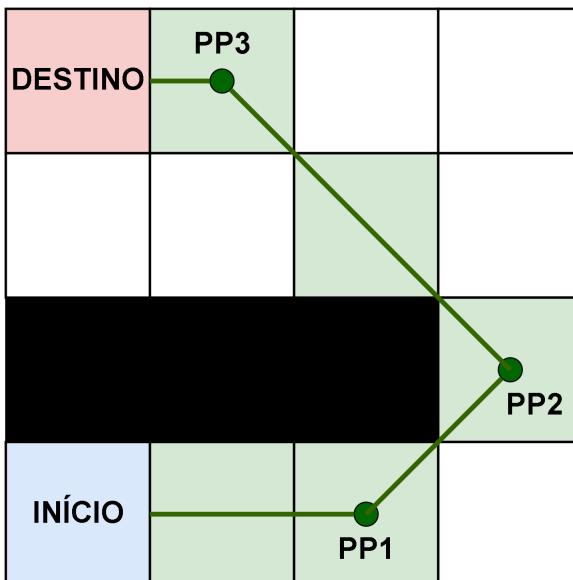
#### 4.4 PLANEJADOR DE ROTA GLOBAL

Para o planejamento de rota percebeu-se a necessidade de um algoritmo que permitisse a definição do caminho através de mapa de grade e além disso o recálculo do mesmo quando necessário, como por exemplo quando um corredor torna-se totalmente obstruído. O replanejamento da rota pode ser custoso em mapas com grandes dimensões, desta forma, reutilizar dados previamente calculados é uma alternativa para poupar processamento. Um algoritmo que implementa as funcionalidades citadas, e utilizado neste trabalho (em sua segunda versão), é o *D Star Lite* proposto por Koenig e Likhachev (2005). Este algoritmo é descrito na íntegra no Anexo A.

Para um mapa de grade ser utilizado neste algoritmo ele deve ser representado como um grafo, onde cada vértice (célula do mapa) é conectado aos seus oito vizinhos (quanto aplicável). Além disso, quando uma célula for ocupada, o custo para o vértice que a representa deve ser infinito. A distância entre vértices na horizontal ou vertical foi definida como 1,0 e para diagonal 1,41. Por fim, os vértices sucessores e predecessores foram determinados como iguais, uma vez que não existe restrição quanto o caminho a ser percorrido entre início e destino.

O algoritmo *D Star Lite*, por padrão, retorna todos os vértices necessários para realizar o melhor caminho, porém, como o mapa utilizado no sistema, mesmo com os redimensionamentos, possui uma quantidade de *pixels* por metro elevada. Caso fossem utilizadas todas as coordenadas, ocorreria um sobrecarregamento do planejador de rota local. Para solucionar este problema foram utilizados pontos de passagem, que consiste em coordenadas chave, onde o robô necessita modificar sua direção, um exemplo pode ser observado na Figura 23.

Figura 23 – Caminho convertido para pontos de passagem.



Fonte: Do Autor

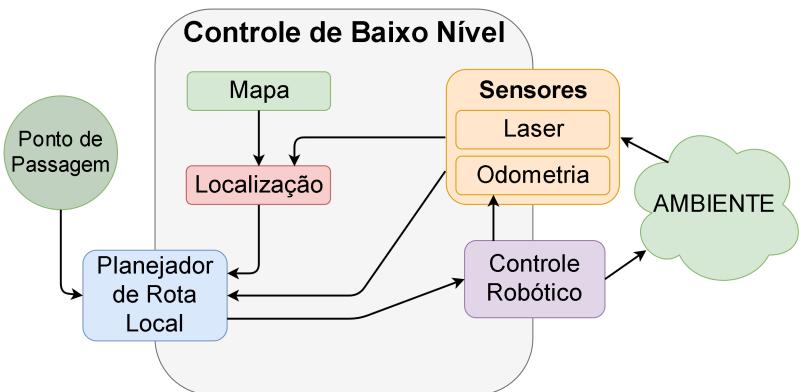
#### 4.5 CONTROLE DE BAIXO NÍVEL

O controle de baixo nível é composto por *drivers* do servidor robótico *player*, esses componentes permitem que o controle, localização e navegação sejam realizados de forma simples. Além disso, diferentes *drivers* realizam a mesma função, possibilitando que o desenvolvedor escolha o que mais se adequa as suas necessidades. A seguir o controle de baixo nível será explanado de forma abstrata para a compreensão de seu funcionamento. A configuração completa dos *drivers* utilizados está disponível no Apêndice A.

A Figura 24 ilustra o controle de baixo nível. Com a chegada de um ponto de passagem o planejador de rota inicia o envio de comandos ao controle robótico para que em linha reta o robô chegue à coordenada. Porém, o ambiente pode conter obstáculos não esperados, então, através da leitura dos sensores o planejador de rota local busca um percurso afim de evitar a colisão. O controle robótico além permitir ao robô percorrer o ambiente, fornece dados para odometria, que através da fusão sensorial com os dados obtidos do laser e comparações com o

mapa do ambiente, permite ao módulo de localização refinar a posição atual do robô.

Figura 24 – Diagrama do controle de baixo nível.



Fonte: Do Autor

Três módulos destacam-se por sua importância direta na navegação, são eles: localização, planejador de rota local e o controle robótico. Por conseguinte, estes componentes serão aprofundados nas seções subsequentes.

#### 4.5.1 Localização

Com a escolha do mapeamento baseado em grades se fez necessário um método de localização que fizesse uso desta metodologia, o *player* disponibiliza um *driver* para localização baseado em mapas de grade chamado AMCL, do inglês *Adaptive Monte Carlo Localization*, que foi implementado seguindo o algoritmo descrito por Fox et al. (1999). Esta aplicação permite a definição da posição atual do robô através da leitura de dados odométricos, distâncias e um mapa do ambiente, aplicando filtro de partículas.

Apesar de o *driver* permitir a localização global no ambiente, preferiu-se utilizar o mesmo apenas como um filtro para refinar a posição. O motivo desta escolha deu-se pelo fato de por vezes o algoritmo falhar na convergência e indicar uma localização errônea. Este erro aconteceu devido ao mapa do ambiente ser praticamente espelhado na vertical o que dificulta a convergência para este tipo de algoritmo uma vez que ele não utiliza pontos de referência. A solução para este pro-

blema foi a definição da posição inicial aproximada do robô, isso faz com que as partículas sejam dispersas em torno do ponto selecionado, e a convergência aconteça com poucos movimentos.

#### **4.5.2 Planejador de Rota Local**

O planejamento de rota global não leva em conta o desvio de obstáculos que podem aparecer aleatoriamente no ambiente. Desta forma, para contornar este problema se faz necessário um planejador de rota local. O *D Star Lite* fornece um conjunto de coordenadas ao qual o robô deve alcançar uma a uma até o seu destino, por sua vez, o planejamento de rota local está inserido para definir a rota entre cada uma destas coordenadas, desviando de obstáculos quando necessário.

Para computar o caminho local foi utilizado o *driver* SND, do inglês *Smooth Nearness Diagram*. Ele implementa o algoritmo desenvolvido por Durham e Bullo (2008). Para definir a rota que irá seguir recebe dados de posicionamento, que podem ser adquiridos diretamente da odometria ou após aplicação de filtros, bem como também faz uso de sensores de distância laser e/ou sonar. Este trabalho, utiliza dados odométricos filtrados pelo *driver* AMCL e, leituras de distâncias provenientes do sensor laser SICK LMS-200.

#### **4.5.3 Controle Robótico**

O robô escolhido para efetuar a validação do sistema foi o Pinonner P3-DX, que faz uso do sistema operacional P2OS. O servidor robótico fornece um *driver* também chamado p2os para controle do robô via comunicação serial RS232. Uma interface que permite que os mesmos comandos sejam utilizados também em simulação é fornecida pelo *Player*, sendo assim o desenvolvimento realizado para a simulação é o mesmo utilizado para o robô real.

### **4.6 ESCALONADOR DE TAREFAS**

Um sistema robótico para logística hospitalar necessita de um escalonador de tarefas para definir a melhor sequência de execução, uma vez que existirão tarefas mais prioritárias que outras, assim como, com distâncias variadas.

O robô necessita seguir seu caminho otimizando o custo de execu-

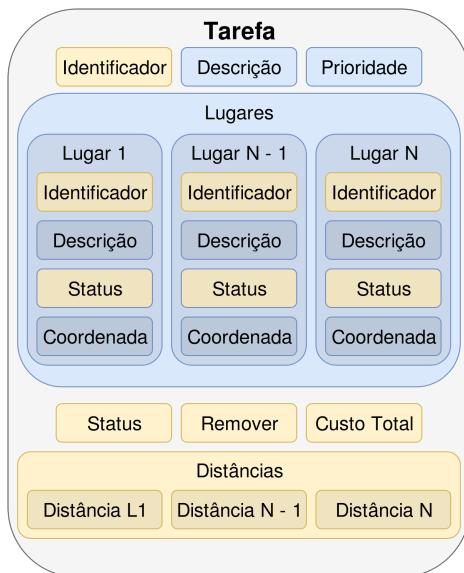
ção de suas tarefas, para isso o sistema proposto neste trabalho ordena as tarefas recebidas em uma fila de acordo com uma função custo.

A seguir serão descritos o conceito de tarefa, assim como o algoritmo de ordenação utilizado para definir a sequência em que as tarefas serão atendidas.

#### 4.6.1 Modelo de Tarefa

Os esforços expeditos neste trabalho estiveram focados no desenvolvimento de um sistema capaz de realizar tarefas pertinentes a logística em um hospital. Para tal, modelar como uma tarefa seria representada no sistema foi parte indispensável. A Figura 25 apresenta o modelo de tarefa, em azul estão os elementos que o usuário e o sistema fazem uso, em amarelo os componentes utilizados apenas internamente pelo sistema.

Figura 25 – Modelo de tarefa.



Fonte: Do Autor

Uma tarefa na visão do usuário consiste apenas em uma descrição do que o robô deve buscar, prioridade que é diretamente proporcional (número entre 1 e 5) e uma lista de locais que devem ser visitados

para cumprir esta tarefa e os status da tarefa e locais. No entanto, o status geral da tarefa, juntamente com o de cada lugar é utilizado para a notificação do estado atual da tarefa. Há um caso em que o usuário modifica o status de forma indireta, quando ele solicita ao robô que remova uma tarefa. O possíveis status gerais são representados através de letras:

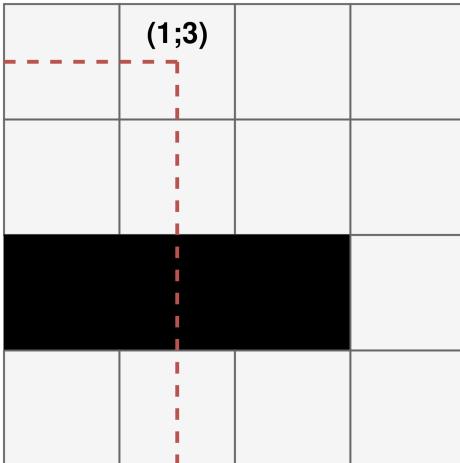
- R: Utilizado para sinalizar que a tarefa deve ser removida.
- W: Estado inicial, significa que o controle de alto nível ainda não efetuou a leitura da tarefa.
- O: Utilizado para notificar que a tarefa já foi lida e está na lista de espera.
- P: Sinaliza que a tarefa está sendo executada.
- D: Notifica o término da tarefa.
- F: Utilizado para informar que a não foi possível realizar a tarefa.

Um local possui código identificador, uma descrição (e.g. Sala 303), coordenada e status também representado através de uma letra:

- W: Estado inicial, significa que o controle de alto nível ainda não efetuou a leitura do local.
- O: Utilizado para notificar que o local já foi lido e está na lista de espera.
- P: Sinaliza que o robô está indo para o local.
- D: Notifica se o robô já passou pelo local.
- F: Utilizado para informar que a não foi possível realizar a tarefa, todos os locais são marcados com este status.

A Figura 26 apresenta um exemplo de coordenada para local, tendo sua origem centrada no canto inferior esquerdo.

Figura 26 – Exemplo de coordenada para local.



Fonte: Do Autor

O vetor de distâncias representa a distância entre o local atual do robô (no momento em que foi computado) até o primeiro lugar da lista juntamente com as distâncias entre os demais lugares pertencentes a tarefa. Logo, se existirem três locais de visita, o vetor distâncias terá tamanho três.

Um *flag*, chamado *remover*, é utilizado para sinalizar que a tarefa deve ser removida do escalonador. A tarefa não é removida no instante que é requisitado através da mudança de status no banco de dados e sim quando o robô alcança um local e por sua vez o algoritmo de ordenação é novamente chamado e antes de realizar o escalonamento remove as tarefas que foram marcadas para serem deletadas.

O custo total de uma tarefa é dado pela Equação 4.1. O sistema escolhe preferencialmente tarefas onde a distância é menor e a prioridade é maior. Na equação  $C_T$  representa o custo total,  $D_k$  a distância entre dois pontos, sendo  $D_0$  a distância do local atual do robô até o primeiro lugar na lista, os demais  $D_{ks}$  representam as distâncias entre os pontos a serem visitados,  $p$  a prioridade da tarefa e por fim  $N$  o tamanho do vetor distâncias.

$$C_T = \sum_{k=1}^N \frac{D_k}{p} \quad (4.1)$$

#### 4.6.2 Algoritmo de Ordenação

Dado um conjunto de cinco tarefas, listadas na Figura 27, em um primeiro momento é calculada a distância total que deve ser percorrida para cumprir cada tarefa, contando o lugar atual do robô, em seguida o custo é obtido dividindo a distância pela prioridade da tarefa. Como pode ser observado, a tarefa com identificador 3 é selecionada para ser a primeira a executar e então é removida da lista.

Considerando a localização atual do robô como o último lugar pertencente a tarefa 3, as distâncias são recalculadas para as tarefas restantes, e também o custo. Vale ressaltar, que neste momento apenas a distância do local atual até o primeiro local da lista é recalculado, uma vez que os valores de distância entre locais da tarefa já foram calculados anteriormente. Assim, o algoritmo itera até que todas as tarefas tenham sido selecionadas. No exemplo ilustrado na Figura 27 a cada iteração a tarefa selecionada está destacada em vermelho.

Figura 27 – Conjunto de tarefas de exemplo para o algoritmo.

<b>Id</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
1	4	855.88	213.97
2	1	659.27	659.27
<b>3</b>	<b>3</b>	<b>545.38</b>	<b>181.793</b>
4	1	2118.71	2118.71
5	5	3299.82	659.964
<hr/>			
<b>1</b>	<b>4</b>	<b>1111.02</b>	<b>277.755</b>
2	1	375.63	375.63
4	1	2321.93	2321.93
5	5	3503.04	700.608
<hr/>			
<b>2</b>	<b>1</b>	<b>366.61</b>	<b>366.61</b>
4	1	2312.91	2312.91
5	5	3494.02	698.804
<hr/>			
4	1	1965.78	1965.78
<b>5</b>	<b>5</b>	<b>3146.89</b>	<b>629.378</b>
<hr/>			
<b>4</b>	<b>1</b>	<b>1485.9</b>	<b>1485.9</b>

Fonte: Do Autor

Para o Algoritmo 7, se deve considerar  $\{T\}$  como o conjunto de tarefas a serem executadas, este conjunto pode conter tarefas que já passaram pelo processo de ordenação anteriormente além disso as tarefas marcadas para serem removidas não devem estar contidas,  $L_i$  como a localização inicial a ser considerada pelo algoritmo,  $\{T_o\}$  o conjunto de tarefas ordenadas e  $\{T_f\}$  o grupo de tarefas que falharam ao serem ordenadas, ou seja, custo infinito para execução.

Na linha 2, a variável  $l$  que representa o local atual é iniciada com a localização inicial. Na linha 3 inicia o *loop* de execução que termina apenas quando todas as tarefas forem removidas da lista  $\{T\}$ .

Para cada iteração o custo das tarefas é atualizado, linhas de 5 até 23. A condicional iniciada na linha 8 garante que a partir da segunda iteração do *loop* global apenas a distância do local atual para o primeiro da lista seja atualizada. Na implementação real a primeira atualização de distâncias faz uso de *threads* para atualizar todas as distâncias ao mesmo tempo.

A condicional iniciada na linha 19 garante que ao término das iterações do *loop* iniciado na linha 5 a tarefa com o menor custo seja selecionada e inserida ao final ou da lista de tarefas ordenada ou da lista de tarefas que falharam, como pode ser observado entre as linhas 24 e 30. Vale ressaltar que a localização atual é atualizada com o último lugar da tarefa selecionada (linha 27) apenas se a tarefa for adicionada a lista de tarefas ordenadas, que representa as tarefas que serão executadas.

---

**Algoritmo 7:** ORDENAÇÃO DE TAREFAS
 

---

**Entrada:**  $\{T\}, L_i, Mapa$

- 1 **início**
- 2     $l = L_i$
- 3    **enquanto** existirem tarefas em  $\{T\}$  **faça**
- 4      $c_{min} = \infty$
- 5     **para**  $i = 1$  até  $\{T\}.tamanho$  **faça**
- 6        $t = \{T\}[i]$
- 7        $l_{tmp} = l$
- 8       **se**  $t.distâncias.tamanho = 0$  **então**
- 9         **para**  $j = 1$  até  $t.lugares.tamanho$  **faça**
- 10            $t.distâncias[j] =$
- 11            $Distância(l_{tmp}, t.lugares[j], t.prioridade)$
- 12            $l_{tmp} = t.lugares[j]$
- 13         **fim**
- 14       **senão**
- 15          $t.distâncias[1] =$
- 16          $Distância(l_{tmp}, t.lugares[j], t.prioridade)$
- 17       **fim**
- 18        $t.CustoTotal = CustoTotal(t)$
- 19       **se**  $t.CustoTotal \leq c_{min}$  **então**
- 20          $c_{min} = t.CustoTotal$
- 21          $id_{min} = i$
- 22       **fim**
- 23     **fim**
- 24      $t = \{T\}[id_{min}]$
- 25     **se**  $t.custo \neq \infty$  **então**
- 26          $\{T_o\}.AdicionarAoFim(t)$
- 27          $l = t.lugares.último$
- 28     **senão**
- 29          $\{T_f\}.AdicionarAoFim(t)$
- 30     **fim**
- 31      $\{T\}.Remover(t)$
- 32     **fim**
- 33 **fim**
- 34 **retorna**  $\{T_o\}, \{T_f\}$

---

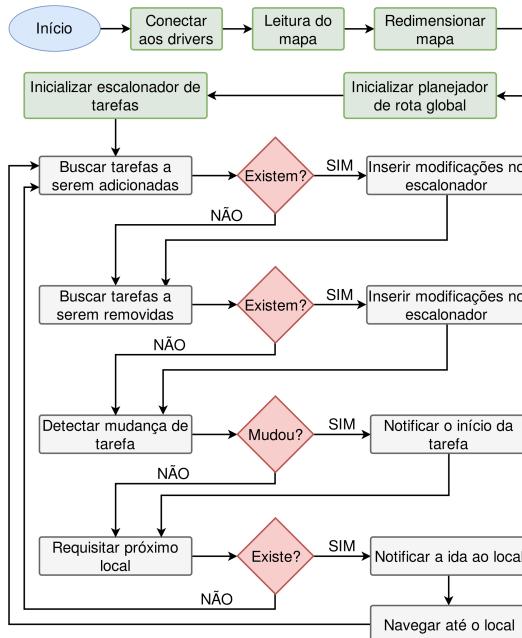
## 4.7 CONTROLADOR DE ALTO NÍVEL

A Figura 28 apresenta o fluxo de funcionamento do controlador de alto nível. Em verde estão os processos relacionados a inicialização, em cinza os pertencentes ao laço de repetição.

A fase de inicialização é composta por efetuar a conexão com os *drivers*, que já estão previamente estartados no servidor robótico. Em seguida é realizada a leitura e redimensionamento do mapa, então o mapa redimensionado é indicado para o planejador de rota global e o escalonador de tarefas.

O laço de repetição é iniciado com a busca por tarefas a serem inseridas e/ou removidas no escalonador, em seguida, caso houver o início da execução de uma tarefa o usuário é notificado, com o recebimento do próximo local o usuário também é notificado e inicia-se o processo de navegação onde é requisitado ao planejador de rota local um caminho até o destino que fornece as coordenadas de pontos de passagem a serem enviadas ao controlador de baixo nível.

Figura 28 – Fluxo de execução do controlador de alto nível.



Fonte: Do Autor



## 5 AVALIAÇÃO DO SISTEMA DE LOGÍSTICA HOSPITALAR

Este capítulo traz a análise e discussão dos experimentos criados para avaliação o sistema desenvolvido através de três fontes: o mapeamento, visando demonstrar a capacidade de navegação, o escalonador de tarefas, confirmando a competência na tomada de decisões na execução de tarefas com base em prioridades e, por fim, o sistema aplicado ao ambiente real, comprovando a capacidade de atuação em ambientes dinâmicos.

### 5.1 MAPEAMENTO

Nesta seção serão apresentados os resultados do mapeamento, assim como, da capacidade para navegação no ambiente mapeado em simulação.

#### 5.1.1 Mapa obtido

A Figura 30(a) é o mapa obtido através da navegação do robô controlada por *joystick* do terceiro andar de um dos blocos da universidade.

Porém, como dito anteriormente na Seção 4.2, o *driver* responsável por efetuar a leitura do mapa utiliza um esquema de cores diferente, sendo cinza para ambiente não mapeado, branco para ambiente mapeado livre e preto para representar os obstáculos. Por meio do software GIMP modificou-se o esquema de cores para atender a restrição do *driver*, o resultado está presente na Figura 30(b).

Figura 29 – Mapa do ambiente. (a) o mapa original. (b) o mapa editado no editor GIMP.

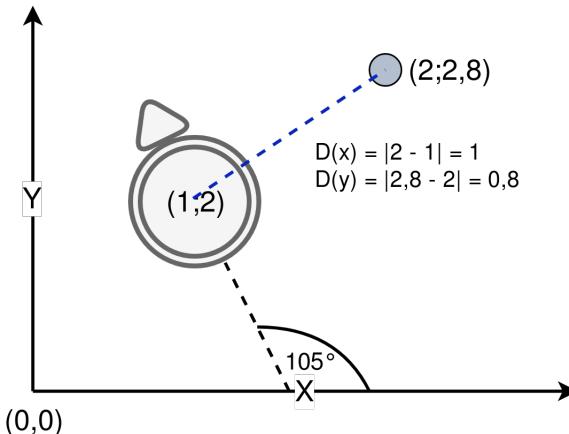


Fonte: Do Autor

### 5.1.2 Precisão durante a navegação

Após a conclusão do mapeamento definiu-se as coordenadas para a porta de cada uma das salas do ambiente. Como não se tinha conhecimento da precisão com que o robô poderia identificar a chegada a uma coordenada no mapa, escolheu-se a distância de um metro entre o local e o centro do robô (independente da posição angular do robô) como sendo o máximo aceitável na coordenada  $x$  e  $y$  para que o robô percebesse a chegada ao destino. Essa definição é ilustrada na Figura 30 onde pode-se observar que a distância em  $x$  e  $y$  é menor que um metro e desta forma é detectada a chegada ao local.

Figura 30 – Ilustração da medida de distância.



Fonte: Do Autor

Sabia-se de antemão que o mapa sofreria redimensionamento para um meio ou um terço do seu tamanho original para aumento de desempenho no planejador de tarefas. Desta forma, era necessário validar a navegação utilizando os mapas reduzidos para checar se a precisão de um metro seria alcançada. Para isso, elaborou-se um experimento simples, o robô teria que navegar até a porta de cinco das salas presentes no mapa, então a distância em que fosse detectada a chegada ao local seria registrada. Para realizar a avaliação, a distância medida foi retirada diretamente do posicionamento do robô no ambiente, fornecido pelo simulador, uma vez que o *driver AMCL* fornece apenas uma estimativa da posição. Para validar a consistência dos dados o experimento foi realizado cinco vezes para cada um dos mapas reduzidos.

A Tabela 3 apresenta os dados de precisão com mapa reduzido em um meio. Pode-se observar que a distância máxima foi respeitada em todas as leituras, uma vez que em linha reta o total aceitável é de 1,41 m. Além disso, o desvio padrão geral ficou em torno de 8 cm, um valor muito satisfatório, e compreensível, pois mesmo em simulação o comportamento do *driver AMCL* pode ser diferente em cada execução.

De forma análoga a anterior, a Tabela 4 apresenta os dados de precisão com mapa reduzido em um terço. Como o esperado, a distância em que ocorre a detecção aumentou um pouco, uma vez que a posição real do local se modifica mais quando a densidade de *pixels* do mapa é

menor. Novamente todas as leituras respeitaram o limite de distância, desta forma permitindo que tanto a redução de um meio quanto a de um terço no mapa possam ser utilizadas no sistema.

Tabela 3 – Dados de precisão com mapa reduzido em um meio.

<b>Local</b>	<b>Distância (m)</b>				
Estacionamento	1,036	0,953	0,786	1,001	0,945
Sala 304	0,981	0,931	1,009	0,975	0,981
Sala 312	0,916	0,921	0,848	0,864	0,949
Sala 316	0,855	0,806	0,808	0,788	0,788
Sala 324	0,946	0,970	0,954	0,967	0,967
<b>Média</b>	0,947	0,916	0,881	0,919	0,926
<b>Desvio Padrão</b>	0,068	0,065	0,096	0,090	0,078
<b>Média Geral</b>	0,918				
<b>Desvio Padrão Geral</b>	0,077				

Tabela 4 – Dados de precisão com mapa reduzido em um terço.

<b>Local</b>	<b>Distância (m)</b>				
Estacionamento	1,092	1,066	1,078	1,044	1,048
Sala 304	0,963	1,077	0,944	1,202	1,073
Sala 312	1,110	1,015	1,139	1,045	1,093
Sala 316	1,099	1,201	1,067	1,059	0,959
Sala 324	0,988	0,971	1,032	0,979	1,105
<b>Média</b>	1,050	1,066	1,052	1,066	1,056
<b>Desvio</b>	0,069	0,087	0,072	0,082	0,058
<b>Média Geral</b>	1,058				
<b>Desvio Padrão Geral</b>	0,068				

## 5.2 ESCALONADOR DE TAREFAS

Para validar o funcionamento do escalonador de tarefas, bem como definir o seu desempenho, três experimentos foram desenvolvidos, cada um executado cinco vezes e, compostos por tarefas, que podem ser vistas na íntegra no Apêndice B.1.2, contendo dois ou três locais, gerados aleatoriamente, a serem alcançados pelo robô.

Foi definido que 80% das tarefas criadas deveriam conter apenas dois lugares para serem visitados, por sua vez, os 20% restantes

deveriam ser preenchidos com tarefas compostas por três locais, essa definição foi escolhida empiricamente, uma vez que no cotidiano o mais comum em uma atividade de logística é transportar o material de um único local para outro.

Determinou-se também que os grupos de tarefas deveriam seguir a sequência dos identificadores, ou seja, se o conjunto possuísse cinco tarefas, dentro deste grupo estariam presentes as tarefas com identificador entre um e cinco.

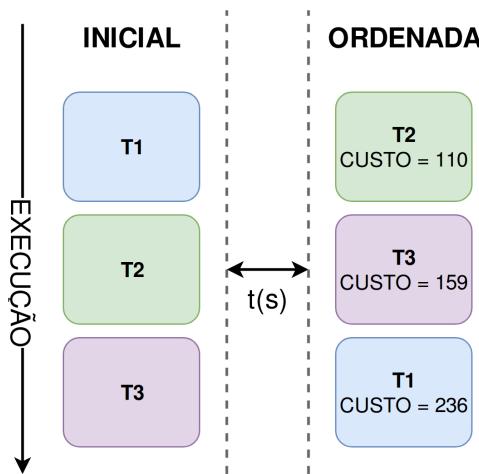
A seguir os três experimentos serão explanados de forma aprofundada.

### 5.2.1 Experimento 1

Este experimento visa validar a capacidade do escalonador de tarefas em ordená-las utilizando a função custo. Além disso, avaliar a complexidade do algoritmo no quesito tempo.

O experimento é exemplificado através da Figura 31, onde pode-se perceber que inicialmente tem-se um número fixo de tarefas. É registrado o tempo que o algoritmo necessitou para ordenar as tarefas bem como a sequência de execução.

Figura 31 – Esquemático do Experimento 1.



Fonte: Do Autor

Os testes foram realizados com cinco, dez, quinze e trinta tarefas. Em todos os casos o escalonador funcionou de acordo ao esperado, a sequência de execução obtida pode ser visualizada na Tabela 5.

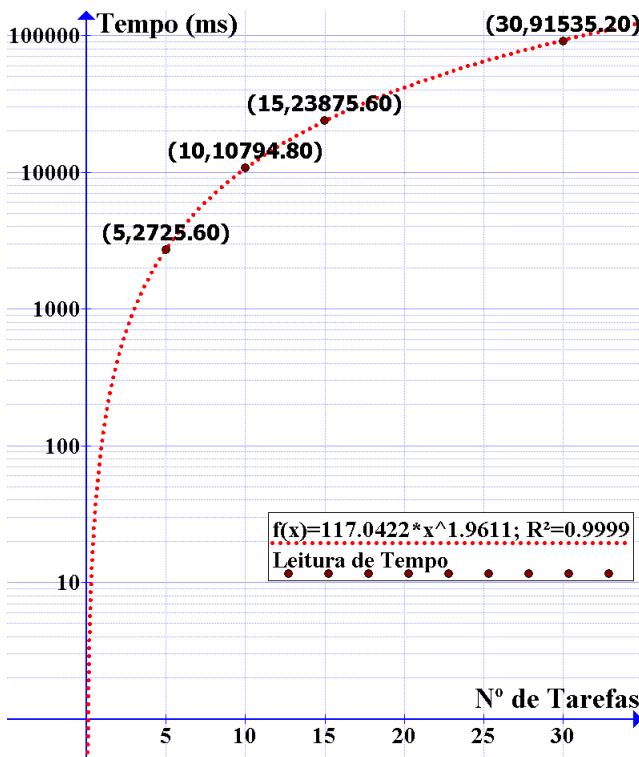
Tabela 5 – Sequência de execução das tarefas. As Tarefas são executadas da esquerda para a direita, sendo que o grupo de quinze e o de trinta tarefas possuem mais de uma parte.

Nº de Tarefas	Sequência de Execução									
5	3	1	2	5	4					
10	3	8	1	6	9	2	7	10	5	4
15 - 1	3	8	1	6	9	2	12	7	10	14
15 - 2	15	11	5	13	4					
30 - 1	25	21	8	1	3	23	30	6	19	9
30 - 2	2	12	16	17	10	7	18	22	20	24
30 - 3	5	14	15	11	13	28	27	26	4	29

Através dos dados relacionados ao tempo de escalonamento foi elaborado o gráfico presente na Figura 32. Pode-se perceber que o algoritmo possui características de complexidade quadrática, afinal, através do coeficiente de determinação ( $R^2$ ) pode-se afirmar que aproximadamente 99,99% da variável tempo, pode ser explicada através da Equação 5.1, onde  $t$  é o tempo em milissegundos e,  $n$  o número de tarefas obedecendo a relação de 80% contendo dois locais e 20% três.

$$t = 117,0422 \cdot n^{1,9611} \quad (5.1)$$

Figura 32 – Relação entre o tempo necessário para escalar e o número de tarefas. Eixo tempo em escala logarítmica.



Fonte: Do Autor

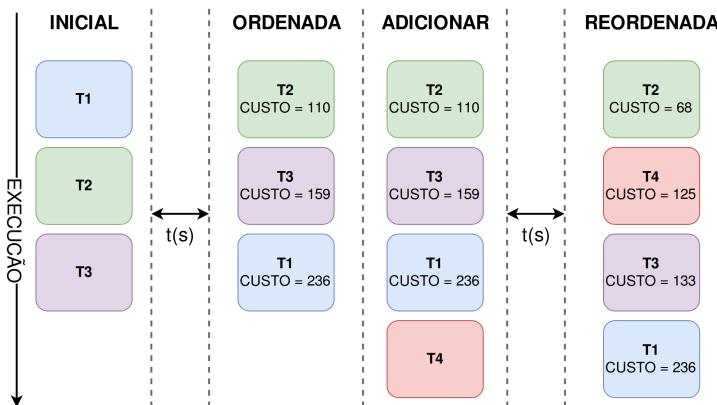
A primeira fase do experimento, utilizando 5 tarefas, foi realizada com o mapa reduzido pela metade e, em um terço, do tamanho original, foi perceptível que o desempenho seria muito reduzido utilizando o mapa com maiores dimensões. O tempo médio para escalar cinco tarefas foi de 6728,00 milissegundos com desvio padrão de 34,86 com o redimensionamento para um meio. Em contrapartida, o resultado para um terço melhorou significativamente para 2725,60 milissegundos e desvio padrão de 54,05. Além disso, a sequência de execução das tarefas foi igual em ambos os redimensionamentos do mapa, assim optou-se em manter a redução para um terço do tamanho original do mapa onde se obtém o melhor desempenho com relação ao tempo.

Os dados que aferem a veracidade das afirmações colocadas a respeito deste experimento estão presentes no Apêndice B.1.3.

### 5.2.2 Experimento 2

Este experimento agrega a avaliação da capacidade de receber novas tarefas durante a execução de um conjunto previamente escalonado. Por meio da Figura 33 pode-se obter o melhor entendimento do experimento. Dado um grupo inicial de tarefas, este é escalonado, então, registra-se o tempo para escalonar e a lista de tarefas ordenada. Durante a execução da primeira tarefa novas tarefas são inseridas e desta forma ficam em espera para serem escalonadas, ao completar o caminho até o primeiro local da lista de locais a serem visitados, o escalonador é chamado e as tarefas são reordenadas, em seguida, novamente o tempo e a sequência de execução são registrados.

Figura 33 – Esquemático do Experimento 2.



Fonte: Do Autor

A Tabela 6 apresenta os tamanhos de grupo de tarefas utilizados neste experimento. Estes valores foram escolhidos pois desta forma seria possível comparar os resultados do grupo inicial com os obtidos no Experimento 1 e assim comprovar a consistência do algoritmo.

Tabela 6 – Tamanho do grupo de tarefas.

Inicial	Adicionado	Total Final
5	5	10
10	5	15
15	15	30

Nenhuma das novas tarefas foi capaz de substituir a tarefa em execução, pois, assim como no exemplo, quando o robô chega ao primeiro local o custo da tarefa que está sendo executada diminui, uma vez que com a chegada este local é removido da lista. Porém, nada impede a preempção de uma tarefa no sistema.

Na primeira medida de tempo obteve-se valores muito próximos aos registrados no Experimento 1, o que exprime a consistência no tempo para escalonar as tarefas. Na segunda aquisição de tempo esperava-se uma redução no período necessário para o escalonamento do grupo total em relação aos registrados no primeiro experimento para conjuntos de mesmo tamanho, uma vez que o primeiro grupo de tarefas necessita apenas da atualização do custo para o primeiro local de sua lista. No entanto, o decréscimo não foi significativo, ficando em torno de aproximadamente 3%.

Quanto ao escalonamento, o experimento mostrou que o sistema possui a capacidade de se adaptar a inserção de novas tarefas sem problemas. A Tabela 7 apresenta os resultados de escalonamento para cada um dos grupos após a adição das tarefas. Como as mesmas tarefas que foram utilizadas no primeiro experimento foram também empregadas neste, para o conjunto contendo até quinze tarefas, os resultados na sequência de execução não se modificaram, uma vez que no experimento anterior, a tarefa com identificador três foi a primeira também para estes grupos.

Tabela 7 – Resultados do escalonamento para cada um dos grupos após a adição das tarefas

<b>Grupo</b>	<b>Sequência após a adição</b>									
	3	8	1	6	9	2	7	10	5	4
<b>5 até 10</b>	3	8	1	6	9	2	7	10	5	4
<b>10 até 15 - 1</b>	3	8	1	6	9	2	12	7	10	14
<b>10 até 15 - 2</b>	15	11	5	13	4					
<b>15 até 30 - 1</b>	3	8	25	21	23	1	30	6	19	9
<b>15 até 30 - 2</b>	2	12	16	17	10	7	18	22	20	24
<b>15 até 30 - 3</b>	5	14	15	11	13	28	27	26	4	29

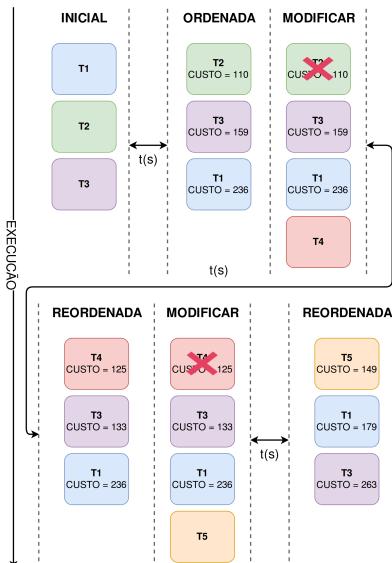
As afirmações realizadas quanto aos resultados deste experimento são comprovadas através dos dados inseridos no Apêndice B.1.4.

### 5.2.3 Experimento 3

Neste experimento foi analisado a capacidade de remoção de tarefas.

A Figura 34 ilustra o experimento de forma simplificada. Foram registradas três medidas de tempo, e três resultados do escalonamento, uma vez que a lista de tarefas é modificada duas vezes, em ambas adicionando e removendo tarefas. De forma análoga ao experimento da seção anterior, a primeira modificação é realizada a caminho do primeiro local a ser visitado. Ao chegar ao local as modificações são realizadas e o escalonamento efetuado, desta forma, gerando uma nova sequência de execução. Convém salientar que no grupo de tarefas removidas nas fases de modificação estava contida a tarefa em execução no momento em que foram recebidas as alterações e, por conta disso, a segunda modificação é realizada durante a execução de uma nova tarefa, visto que caso ela não fosse removida continuaria sendo executada.

Figura 34 – Esquemático do Experimento 3.



Fonte: Do Autor

Para realizar esta validação, como visto no exemplo a cima, dois grupos de modificações foram realizadas, a Tabela 8 apresenta os identificadores das tarefas adicionadas e removidas em cada um.

Tabela 8 – Identificadores das tarefas utilizadas nas modificações.

Modificação	Adicionar			Remover		
1 <sup>a</sup>	11	12	13	3	7	-
2 <sup>a</sup>	14	15	-	4	8	11

Após as modificações, os conjuntos resultantes podem ser observados na Tabela 9. Estas tarefas foram computadas pelo escalonador e ordenadas de tal forma que seriam executadas na sequência presente na Tabela 10.

Tabela 9 – Grupo de tarefas em cada estágio.

Grupo	Lista de tarefas										
	1	2	3	4	5	6	7	8	9	10	-
Inicial	1	2	4	5	6	8	9	10	11	12	13
1 <sup>a</sup> Modificação	1	2	4	5	6	8	9	10	11	12	13
2 <sup>a</sup> Modificação	1	2	4	5	6	8	10	11	12	13	-

Tabela 10 – Resultados do escalonamento para cada um dos grupos após a modificação.

Grupo	Sequência										
	3	8	1	6	9	2	7	10	5	4	-
Inicial	3	8	1	6	9	2	7	10	5	4	-
1 <sup>a</sup> Modificação	8	1	6	9	2	12	5	13	11	10	4
2 <sup>a</sup> Modificação	1	6	9	2	12	5	14	15	13	10	-

Através dos resultados obtidos referentes ao escalonamento, pode-se concluir que a funcionalidade de remover tarefas também é desempenhada com sucesso pelo escalonador, desta forma, estes dados mostram que o sistema para ordenar execução de tarefas exerce suas funções de forma eficaz. No entanto, sua eficiência no que diz respeito ao tempo ainda necessita de melhorias, pois assim como no experimento anterior, não houve melhoria significativa no tempo para o escalonamento, mesmo com o grupo resultante contendo tarefas com custos pré calculados.

A Tabela 11 apresenta os dados de registro de tempo. Para o grupo contendo dez tarefas a medida obtida foi muito semelhante ao do

Experimento 1 mostrando novamente a eficiência do algoritmo. Apesar de existirem tarefas com custos pré calculados, o tempo para o escalonamento modifica-se pouco, os resultados da primeira e segunda modificação se tornam justificáveis. Para a primeira modificação, apesar de o conjunto crescer em apenas uma unidade, ocorreu um aumento de aproximadamente 700 milissegundos no escalonamento, justificado pelo acréscimo de três tarefas que necessitaram ter seu custo totalmente computado. Já na segunda modificação o aumento, um pouco menor, é justificado pela adição de duas tarefas, porém uma contendo três locais para visitação.

Tabela 11 – Registro de tempo do experimento

	Inicial	1 <sup>a</sup> Modificação	2 <sup>a</sup> Modificação
Tempo (ms)	10739,60	11434,80	11279,00
Desvio Padrão	52,68	44,53	51,91

O Apêndice B.1.5 contém os dados referentes a este experimento.

### 5.3 AVALIAÇÃO EM AMBIENTE REAL

O experimento aqui realizado é o mesmo apresentado na Subseção 5.2.1 para cinco tarefas, porém em ambiente real. O objetivo é avaliar a capacidade do robô em realizar as mesmas tarefas em um ambiente dinâmico e ruidoso. Além disso, avaliar se as mesmas configurações utilizadas para os *drivers* em simulação seriam eficazes no ambiente real.

#### 5.3.1 Modificações realizadas no robô

Foram necessárias algumas modificações estruturais no robô para avaliar o sistema no ambiente real. Como o sistema desenvolvido utiliza apenas os dados provenientes do sensor laser, a câmera que estava alocada sobre o mesmo foi removida. Além disso, o robô conta com um computador interno, porém, devido ao seu baixo desempenho optou-se em utilizar um com melhor desempenho alocado externamente. Então, no lugar da câmera, posicionou-se um suporte acrílico onde o notebook utilizado para controlar o robô ficaria alocado. Dois cabos de comunicação RS-232 foram utilizados, um para a comunicação com o sistema operacional do robô e outro para o sensor laser. As modificações podem

ser visualizadas na Figura 35.

Figura 35 – Foto do robô Pioneer P3-DX com as modificações efetuadas para o experimento.



Fonte: Do Autor

Estas modificações se mostraram necessários pois o computador presente internamente no robô com as configurações: Processador Intel Pentium M de 1,8GHz e memória RAM de 512 MB e sistema operacional Xubuntu 14.04 LTS, teria um desempenho muito inferior no escalonamento em relação aos resultados obtidos em simulação. Além disso, o desempenho dos *drivers* poderiam ser afetados, gerando resultados muito diferentes aos obtidos em simulação.

### 5.3.2 Resultados obtidos

O sensor físico, utilizando o cabo de comunicação disponível, permitiu realizar até dez leituras por segundo, com alcance máximo de 8,192m e 1mm de precisão. Essa restrição impactou no tempo de convergência do *driver* de localização negativamente uma vez que o processamento depende da leitura para ser efetuado. Porém, reduzindo o número de partículas final (de 1000 para 500), através de tentativa e erro, obteve-se a convergência na posição atual do robô logo no início do experimento, após o robô girar 180°, assim como em simulação.

A velocidade máxima para navegação em simulação foi configurada para 0.5 m/s. Para avaliar se o robô seria capaz de atender esta configuração, decidiu-se iniciar com a velocidade máxima setada em 0.2 m/s e aumentar gradativamente (em 0.1 m/s) até que ocorressem colisões durante a execução das tarefas. Foi observado que velocidades superiores a 0.5 m/s causariam colisões, então, a velocidade máxima permaneceu igual a utilizada em simulação.

A navegação livre de colisões provida pelo *driver* SND foi alcançada, porém com limitações. Por exemplo, um obstáculos que bloqueia o corredor deixando apenas o espaço um pouco maior que o tamanho do robô, ou ainda, um obstáculos que ocupe a posição de um ponto de passagem, podeoccasionar o não cumprimento da tarefa.

Respeitando as limitações que foram percebidas, na experimentação em ambiente real, o sistema consegue executar as tarefas com sucesso assim como no ambiente simulado, inclusive, posicionando-se perfeitamente em frente as salas a serem visitadas, como pode ser observado na Figura 36. Contudo, os pontos de passagem (disponíveis no Apêndice B.2) em ambiente real e simulação foram diferentes com exceção ao último, que consiste na posição exata de cada local. Com a configuração de velocidade obtida o robô consegue completar as cinco tarefas (identificadores entre 1 e 5) em aproximadamente 20 minutos em simulação inclusive.

Figura 36 – Chegada do robô à Sala 306.



Fonte: Do Autor

Além de medir o tempo necessário para o robô executar as tarefas foi realizada a medição para uma pessoa realizando as mesmas atividades. Como era esperado, empiricamente, a pessoa realizou as funções em menos tempo, aproximadamente seis minutos, redução em torno de 70% no tempo para completar o conjunto de tarefas.

Um vídeo demonstrando a chegada do robô em um local, bem como, efetuado o desvio de obstáculos está disponível em:

[https://www.youtube.com/watch?v=NNk\\_tRQUOac](https://www.youtube.com/watch?v=NNk_tRQUOac)



## 6 CONSIDERAÇÕES FINAIS E PROPOSTAS PARA TRABALHOS FUTUROS

O uso de sistemas robóticos manipuladores é disseminado a anos para a realização de tarefas repetitivas, por vezes insalubres, nas mais diversas áreas da indústria. A popularidade na utilização deste tipo de automatização na produção ocorre pois a longo prazo é mais rentável manter um robô para executar funções triviais, por exemplo a soldagem em uma linha de produção, o que permite a contratação de colaboradores para a execução de tarefas que necessitam de maior atenção a detalhes, como a inspeção do acabamento de uma peça.

O uso de robôs móveis em empresas ainda é restrito, principalmente, ao uso de robôs do tipo AGV. No entanto, este tipo de robô exige o preparo prévio do ambiente, inserindo linhas no chão, onde serão executadas as funções relacionadas a logística. Para efetuar as mesmas tarefas de um AGV, porém sem a utilização de linhas, pode-se efetuar o mapeamento do ambiente e utilizar-se de métodos estatísticos para a localização, desta forma, permitindo a navegação através do planejamento de rotas utilizando mapas.

A utilização de pessoas para tarefas rotineiras de logística representam um desperdício de recursos humanos, uma vez que estas pessoas poderiam estar desempenhando atividades mais complexas. No entanto a utilização de robôs AGV na realização destas atividades implica na modificação do ambiente, e além disso, restrições ao dinamismo que o ambiente pode oferecer, uma vez que um pequeno obstáculo sobre a linha guia implicaria na parada do robô até que o mesmo fosse retirado.

Este trabalho teve os esforços direcionados ao desenvolvimento de um sistema capaz de se locomover em um ambiente mapeado, afim de realizar tarefas de logística com prioridades, lidando com o dinamismo proporcionado por pessoas e objetos não mapeados. Para avaliar seu funcionamento foram realizados experimentos onde a capacidade de navegação foi avaliada em simulação e ambiente real, bem como, foram analisados o desempenho e eficácia do escalonador de tarefas.

A capacidade de navegação do robô foi muito semelhante em ambos os ambientes, real e simulado. O que foi motivo de surpresa, afinal, esperava-se resultados inferiores quanto ao posicionamento do robô em relação ao destino, uma vez que o ambiente real é ruidoso. No entanto, em todas as tarefas o robô posicionou-se exatamente em frente a cada local de visitação.

Era esperado que obstáculos, onde o corredor não fosse total-

mente bloqueado, não gerasse problemas para navegação. No entanto, os obstáculos localizados aos arredores de um ponto de passagem fizeram com que o robô não fosse capaz de desviar, desta forma, impedindo o término da tarefa. Uma possível solução seria agregar ao sistema o módulo de atualização do mapa durante a navegação, o que permitiria o recálculo da rota considerando as alterações do ambiente e assim a finalização da tarefa.

A apesar de o mapa obtido do ambiente apresentar distorção em um dos corredores, o *driver* AMCL conseguiu realizar a localização sem problemas. Ao chegar no corredor a posição atual do robô é compensada pelo *driver*, fazendo com que a rota, que considera a distorção não existente no ambiente real, gerada pelo planejador de rotas não gere problemas na execução das tarefas. A distorção pode ser observada no corredor presente na parte superior da Figura 30(b).

O escalonador de tarefas se mostrou eficaz na execução de sua função, realizando suas atividades de acordo ao esperado. Contudo, seu desempenho em relação ao tempo mostrou-se baixo, apresentando complexidade quadrática, o que abre espaço para trabalhos futuros visando o aumento da performance temporal do algoritmo de escalonamento.

Os resultados dos experimentos demonstraram que o sistema é capaz de executar tarefas em uma ambiente dinâmico mapeado, de acordo com suas prioridades, dadas as restrições que apareceram durante a avaliação da aplicação. No estágio atual do sistema, ele poderia ser, inicialmente, utilizado em tarefas mais simples de logística no ambiente hospitalar, como a entrega de alimentos, transporte de roupas de cama ou ainda equipamentos, não envolvendo a logística em situações de emergência. Este trabalho abre espaço para diversos trabalhos futuros visando o aperfeiçoamento do sistema, sendo que os considerados como principais são apresentados na seção a seguir.

## 6.1 PROPOSTAS PARA TRABALHOS FUTUROS

Nesta seção são elencadas propostas para possíveis trabalhos futuros com o objetivo de aperfeiçoar o sistema desenvolvido.

1. Avaliação do sistema utilizando mapas topológicos, efetuando um comparativo do desempenho em relação aos mapas de grade.
2. Fazer com que a função custo considere a carga da bateria do robô.
3. Inclusão de um módulo no sistema de navegação que considere a

atualização do mapa em tempo real.

4. Criar uma ferramenta que possibilite a navegação em ambientes com múltiplos andares.
5. Desenvolver um mecanismo para estimativa do tempo de execução da tarefa antes de enviá-la ao robô.
6. Projetar uma interface gráfica que permita a descrição de tarefas ao sistema de navegação em tempo real.
7. Incluir plano de ações específicas a serem realizadas em situações de emergência no ambiente hospitalar.

## 6.2 INFORMAÇÕES ADICIONAIS

Este trabalho está disponibilizado na íntegra em um repositório no *GitHub*, onde estão contidos os arquivos de códigos fonte do sistema desenvolvido, banco de dados, configurações do *player* e *stage*, esta monografia e a planilha de dados relativos aos experimentos. A seguir o link para o repositório:

<https://github.com/alankc/TCC>



## REFERÊNCIAS

- AINI, F. R. Q.; JATI, A. N.; SUNARYA, U. A study of monte carlo localization on robot operating system. In: *2016 International Conference on Information Technology Systems and Innovation (ICITSI)*. [S.l.: s.n.], 2016. p. 1–6.
- AMANDA, M. P.; JATI, A. N.; SUNARYA, U. An evaluation of 2d indoor localization and mapping using fastslam. In: *2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*. [S.l.: s.n.], 2016. p. 147–151.
- ARMY-TECHNOLOGY. *TALON Tracked Military Robot, United States of America*. 2017. <<http://www.army-technology.com/projects/talon-tracked-military-robot/>>. Acessado em 15-03-2017.
- BAALBAKI, H.; XIE, X. *A decision framework for operation management of reconfigurable mobile service robots in hospitals*. [S.l.]: IFAC, 2009. 151–156 p. ISSN 14746670. ISBN 9783902661432.
- BADINO, H.; HUBER, D.; KANADE, T. Visual topometric localization. *IEEE Intelligent Vehicles Symposium, Proceedings*, n. Iv, p. 794–799, 2011. ISSN 1931-0587.
- BEKEY, G. A. *Autonomous Robots: From Biological Inspiration to Implementation and Control (Intelligent Robotics and Autonomous Agents series)*. [S.l.]: A Bradford Book, 2005. ISBN 0-262-02578-7.
- BUKHORI, I.; ISMAIL, Z. H. Detection strategy for kidnapped robot problem in monte carlo localization based on similarity measure of environment. In: *2016 IEEE International Conference on Underwater System Technology: Theory and Applications (USYS)*. [S.l.: s.n.], 2016. p. 55–60.
- CHEH, D. et al. A method for solving local minimum problem of local path planning based on particle swarm optimization. In: *2017 Chinese Automation Congress (CAC)*. [S.l.: s.n.], 2017. p. 4944–4949. ISBN 978-1-5386-3524-7.
- DOUCET, A. et al. Rao-blackwellised particle filtering for dynamic bayesian networks. In: *Proceedings of the 16th Conference on*

*Uncertainty in Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000. (UAI '00), p. 176–183. ISBN 1-55860-709-9.

DUCHON, F. et al. Path planning with modified A star algorithm for a mobile robot. *Procedia Engineering*, v. 96, p. 59–69, 2014. ISSN 18777058.

DULIMART, H. S.; JAIN, A. K. Mobile robot localization in indoor environment. *Pattern Recognition*, v. 30, n. 1, p. 99 – 111, 1997. ISSN 0031-3203.

DURHAM, J. W.; BULLO, F. Smooth nearness-diagram navigation. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, p. 690–695, 2008. ISSN 2153-0858.

ELIAZAR, A.; PARR, R. DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks. *IJCAI International Joint Conference on Artificial Intelligence*, p. 1135–1142, 2003. ISSN 10450823.

FALLAH, N. et al. Indoor human navigation systems: A survey. *Interacting with Computers*, v. 25, n. 1, p. 21–33, 2013. ISSN 09535438.

FOX, D. et al. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. *American Association for Artificial Intelligence*, n. Handschin 1970, p. 1–6, 1999.

GALILI, T.; MEILIJSON, I. An Example of an Improvable Rao–Blackwell Improvement, Inefficient Maximum Likelihood Estimator, and Unbiased Generalized Bayes Estimator. *The American Statistician*, v. 70, n. 1, p. 108–113, 2016.

GRISSETTI, G.; STACHNISS, C.; BURGARD, W. Improved Techniques for Grid Mapping. *Robotics, IEEE Transactions on*, v. 23, n. 1, p. 34–46, 2007.

HADJI, S. E. et al. 2D occupancy grid mapping with inverse range sensor model. *2015 10th Asian Control Conference: Emerging Control Techniques for a Sustainable World, ASCC 2015*, 2015.

IROBOT. *Romba*. 2017.

<[https://www.nasa.gov/mission\\_pages/msl/overview/index.html](https://www.nasa.gov/mission_pages/msl/overview/index.html)>. Acessado em 15-03-2017.

- JIA, S. et al. Mobile robot localization and mapping based on mixed model. In: *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*. [S.l.: s.n.], 2010. v. 5, p. V5–9–V5–14. ISSN 2154-7491.
- JIANG, R. et al. Geometric map-assisted localization for mobile robots based on uniform-gaussian distribution. *IEEE Robotics and Automation Letters*, v. 2, n. 2, p. 789–795, April 2017. ISSN 2377-3766.
- JO, H. et al. Grid mapping adaptive to various map sizes for sbot. In: *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*. [S.l.: s.n.], 2013. p. 1678–1680. ISSN 2093-7121.
- JUNG, J. et al. AUV SLAM using forward/downward looking cameras and artificial landmarks. In: *2017 IEEE Underwater Technology (UT)*. [S.l.]: IEEE, 2017. p. 1–3. ISBN 978-1-5090-5266-0.
- KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, v. 82, n. 1, p. 35, 1960. ISSN 00219223.
- KAUFMAN, E. et al. Bayesian occupancy grid mapping via an exact inverse sensor model. In: *2016 American Control Conference (ACC)*. [S.l.: s.n.], 2016. p. 5709–5715.
- KOENIG, S.; LIKHACHEV, M. Fast replanning for navigation in unknown terrain. *IEEE Transactions on Robotics*, v. 21, n. 3, p. 354–363, 2005. ISSN 15523098.
- KONG, F. et al. Mobile Robot Localization Based on Extended Kalman Filter. *2006 6th World Congress on Intelligent Control and Automation*, n. 60474021, p. 9242–9246, 2006.
- KOSNAR, K. et al. Comparison of shape matching techniques for place recognition. *2013 European Conference on Mobile Robots*, p. 107–112, 2013.
- KOSTAVELIS, I.; GASTERATOS, A. Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, Elsevier B.V., v. 66, p. 86–103, 2015. ISSN 09218890.
- KUNDU, A. S. et al. Occupancy Grid Map Generation using 360° Scanning Xtion Pro Live for Indoor Mobile Robot Navigation. *2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI)*, n. Cmi, p. 464–468, 2016.

LAVALLE, S. M. *Planning Algorithms*. [S.l.]: Cambridge University Press, 2016.

LI, X. et al. Mobile robot map building based on laser ranging and kinect. In: *2016 IEEE International Conference on Information and Automation (ICIA)*. [S.l.: s.n.], 2016. p. 819–824.

LOOMIS, J. M.; KLATZKY, R. L.; GOLLEDGE, R. G. Navigating without vision: basic and applied research. *Optometry and vision science : official publication of the American Academy of Optometry*, v. 78, n. 5, p. 282–289, 2001. ISSN 1040-5488.

MA, L.; YAO, Y.; WANG, M. The optimizing design of wheeled robot tracking system by pid control algorithm based on bp neural network. In: *2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*. [S.l.: s.n.], 2016. p. 34–39.

MARINHO, L. B. et al. A novel mobile robot localization approach based on topological maps using classification with reject option in omnidirectional images. *Expert Systems With Applications*, Elsevier Ltd, v. 72, p. 1–17, 2017. ISSN 0957-4174.

MONTEMERLO, M. et al. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In: *Proceedings of the AAAI National Conference on Artificial Intelligence*. Edmonton, Canada: AAAI, 2002.

MORAVEC, H.; ELFES, A. High resolution maps from wide angle sonar. In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation*. [S.l.: s.n.], 1985. v. 2, p. 116–121.

MURPHY, K. P. Bayesian map learning in dynamic environments. In: *Proceedings of the 12th International Conference on Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 1999. (NIPS'99), p. 1015–1021.

NASA. *Curiosity Overview*. 2015.  
[<https://www.nasa.gov/mission\\_pages/msl/overview/index.html>](https://www.nasa.gov/mission_pages/msl/overview/index.html).  
 Acessado em 15-03-2017.

PAZOS-REVILLA, M.; GUO, T.; MACHIDA, M. Extended kalman filter combined with fuzzy rules for localization using wireless transceivers. In: *2016 Annual Conference of the North American*

*Fuzzy Information Processing Society (NAFIPS)*. [S.l.: s.n.], 2016. p. 1–6.

PRONOBIOS, A. et al. A realistic benchmark for visual indoor place recognition. *Robotics and Autonomous Systems*, Elsevier B.V., v. 58, n. 1, p. 81–96, 2010. ISSN 09218890.

RAJA, P.; PUGAZHENTHI, S. Optimal path planning of mobile robots: A review. *International Journal of the Physical Sciences*, v. 7, n. 9, p. 1314–1320, 2012. ISSN 19921950.

RAMAITHITIMA, R. et al. Automated Creation of Topological Maps in Unknown Environments Using a Swarm of Resource-Constrained Robots. *IEEE Robotics and Automation Letters*, v. 1, n. 2, p. 746–753, 2016. ISSN 23773766.

REINEKING, T.; CLEMENS, J. Evidential fastslam for grid mapping. In: *Proceedings of the 16th International Conference on Information Fusion*. [S.l.: s.n.], 2013. p. 789–796.

RIA. *UNIMATE*: The first industrial robot. 2015.  
[<https://www.robotics.org/joseph-engelberger/unimate.cfm>](https://www.robotics.org/joseph-engelberger/unimate.cfm).  
 Acessado em 27-04-2017.

ROBOTIQ. *Mr. Roam: Mobile Platform for Industrial Robotic Applicationsa*. 2016. <<http://blog.robotiq.com/bid/65087/Mr-Roam-Mobile-Platform-for-Industrial-Robotic-Applications>>. Acessado em 15-03-2017.

ROBOTNIK. *Autonomous mobile robot AGVS*. 2017a.  
[<http://www.robotnik.eu/mobile-robots/autonomous-agvs/>](http://www.robotnik.eu/mobile-robots/autonomous-agvs).  
 Acessado em 15-03-2017.

ROBOTNIK. *Agriculture: Vinbot*. 2017b.  
[<http://www.robotnik.eu/services-robotic/mobile-robotics-applications/agriculture/>](http://www.robotnik.eu/services-robotic/mobile-robotics-applications/agriculture). Acessado em 15-03-2017.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach (2nd Edition)*. [S.l.]: Prentice Hall, 2002. ISBN 0137903952.

SCHRÖTER, C.; BÖHME, H.; GROSS, H. Memory-Efficient Gridmaps in Rao-Blackwellized Particle Filters for SLAM using Sonar Range Sensors. *Emcr*, n. January 2007, p. 1–6, 2007.

SENGUPTA, S. et al. Urban 3D semantic modelling using stereo vision. In: *2013 IEEE International Conference on Robotics and Automation*. [S.l.]: IEEE, 2013. p. 580–585. ISBN 978-1-4673-5643-5.

SHOUKRY, Y.; ABDELFATAH, W. F.; HAMMAD, S. A. Real-time markov localization for autonomous ugv. In: *2009 4th International Design and Test Workshop (IDT)*. [S.l.: s.n.], 2009. p. 1–6. ISSN 2162-0601.

SICILIANO, B.; KHATIB, O. *Springer Handbook of Robotics*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007. ISBN 354023957X.

SIEGWART, R.; NOURBAKHSH, I. R.; SCARAMUZZA, D. *Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents series)*. [S.l.]: The MIT Press, 2011. ISBN 978-0-262-01535-6.

SKOBELEVA, A.; UGRINOVSKII, V.; PETERSEN, I. Extended kalman filter for indoor and outdoor localization of a wheeled mobile robot. In: *2016 Australian Control Conference (AuCC)*. [S.l.: s.n.], 2016. p. 212–216.

STACHNISS, C.; BURGARD, W. Mobile Robot Mapping and Localization in Non-Static Environments. *Proceedings of the {AAAI} National Conference on Artificial Intelligence*, n. 2003, p. 1324–1329, 2005.

TAKAHASHI, M. et al. Developing a mobile robot for transport applications in the hospital domain. *Robotics and Autonomous Systems*, Elsevier B.V., v. 58, n. 7, p. 889–899, 2010. ISSN 09218890.

TAPUS, A.; SIEGWART, R. Incremental robot mapping with fingerprints of places. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, p. 172–177, 2005.

THRUN, S. Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva. *The International Journal of Robotics Research*, v. 19, n. 11, p. 972–999, 2000. ISSN 0278-3649.

THRUN, S.; BURGARD, W.; FOX, D. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series)*. [S.l.]: The MIT Press, 2005. ISBN 978-0-262-20162-9.

- WANG, C.-c. et al. Simultaneous Localization , Mapping and Moving Object Tracking Moving Object Tracking. *Robotics*, 2007. ISSN 0278-3649.
- WANG, D. et al. Lidar Scan matching EKF-SLAM using the differential model of vehicle motion. In: *2013 IEEE Intelligent Vehicles Symposium (IV)*. [S.l.]: IEEE, 2013. p. 908–912. ISBN 978-1-4673-2755-8.
- WELCH, G.; BISHOP, G. *An Introduction to the Kalman Filter*. Chapel Hill, NC, USA, 1995.
- YATIM, N. M.; BUNIYAMIN, N. Particle filter in simulataneous localisation and mapping (slam) using differential drive mobile robot. *Jurnal Teknologi (Sci Eng)*, v. 77, n. 20, p. 91–97, 2015.
- YE, Z. et al. Cognition inspired framework for indoor scene annotation. *Journal of Electronic Imaging*, v. 24, n. 5, p. 053013–1–053013–11, 2015.
- YUZHENG, P.; QUANDE, Y.; BENFA, Z. The application of adaptive extended kalman filter in mobile robot localization. In: *2016 Chinese Control and Decision Conference (CCDC)*. [S.l.: s.n.], 2016. p. 5337–5342.



## **APÊNDICE A - Configurações de *drivers* do *player***



Para a compreensão completa das definições aqui apresentadas, é recomendado o estudo do funcionamento do servidor robótico *Player 3.0.2* e o Simulador *Stage 4.1.1*.

## A.1 UTILIZADOS APENAS EM SIMULAÇÃO

O *driver stage* é utilizado para carregar a simulação, bem representar o robô na mesma juntamente com seus sensores.

```
driver
(
    name "stage"
    provides ["simulation:0"]
    plugin "stageplugin"
    worldfile "arquivo.world"
)

driver
(
    name "stage"
    provides ["position2d:0" "ranger:0"
              "ranger:1"]
    model "r0"
    alwayson 1
)
```

Como alguns *drivers* ainda não aceitam dados provenientes da nova representação de sensor medidor de distância (presente a partir da versão 4 do *stage*), o *ranger*, o *driver rangertolaser* foi utilizado para converter os dados do sensor *ranger* na simulação para o tipo que um sensor laser proveria.

```
driver
(
    name "rangertolaser"
    provides ["laser:0"]
    requires ["ranger:1"]
    alwayson 1
)
```

## A.2 UTILIZADOS APENAS EM AMBIENTE REAL

A fim de controlar o robô real através de cabo RS-232 foi utilizado o *driver p2os*.

```
driver
(
    name "p2os"
    provides ["odometry :::: position2d :0"
              "sonar :0"]
    port "/dev/ttyUSB1"
    max_xspeed 2
    pulse "2"
)
```

Para receber as leituras provenientes do sensor laser, SICK LMS-200, através de cabo RS-232 foi utilizado o *driver sicklms200*.

```
driver
(
    name "sicklms200"
    provides ["laser :0"]
    port "/dev/ttyUSB0"
    resolution 100
    serial_high_speed_mode 1
    serial_high_speed_baudremap 230400
    connect_rate [ 9600 500000 38400]
    transfer_rate 38400
    retry 10
    alwayson 1
)
```

## A.3 UTILIZADOS EM SIMULAÇÃO E AMBIENTE REAL

A leitura da imagem que representa o mapa de grande é realizada através do *driver mapfile*. Vale ressaltar que deve-se utilizar imagens com extensão PGM, o *driver* apresentou erro na leitura de outros tipos como JPEG e PNG.

```
driver
(
    name "mapfile"
```

```

    provides  ["map:1"]
    filename  "map/arquivo.pgm"
    resolution  0.028571429
    alwayson  1
)

```

Para que a rota criada pelo planejador de rota global possa considerar o robô como uma célula, é utilizado o *driver mapcspace*. Ele faz com que os obstáculos aumentem de tamanho, assim, mesmo que no caminho exista um ponto de passagem junto a parede, esse ponto, no ambiente real estaria distante ao menos o raio do robô. O mapa resultante deste processo é utilizado apenas pelo planejador de rota global.

```

driver
(
    name  "mapcspace"
    requires  ["map:1"]
    provides  ["map:0"]
    robot_shape  "circle"
    robot_radius  0.25
    alwayson  1
)

```

A seguir a configuração do *driver amcl*, responsável por realizar a localização baseada em filtros de partículas.

```

driver
(
    name  "amcl"
    provides  ["position2d:1"]
    requires  ["odometry:::position2d:0"
               "laser:0"  "laser:::map:1"]
    update_thresh  [0.01  0.012359877]
    init_pose  [-5.89  -21.2  -90.0]
    pf_max_samples  100000
    pf_min_samples  500
    alwayson  1
)

```

O *driver snd* provê o planejamento de rota local. Convém ressaltar que ele recebe a posição atual do robô provida pelo AMCL (input:::position2d:1) e também a interface onde deve enviar os comandos para locomoção (output:::position2d:0) proveniente do P2OS ou da si-

mulação.

```
driver
(
    name "snd"
    provides ["position2d:2"]
    requires ["input:::position2d:1"
              "output:::position2d:0" "laser:0"]
    robot_radius 0.25
    min_gap_width 0.3
    obstacle_avoid_dist 0.5
    max_speed 0.5
    alwayson 1
)
```

## **APÊNDICE B – Dados dos experimentos de validação**



## B.1 ESCALONADOR DE TAREFAS

Esta seção apresenta os dados referentes aos experimentos de validação do escalonador de tarefas.

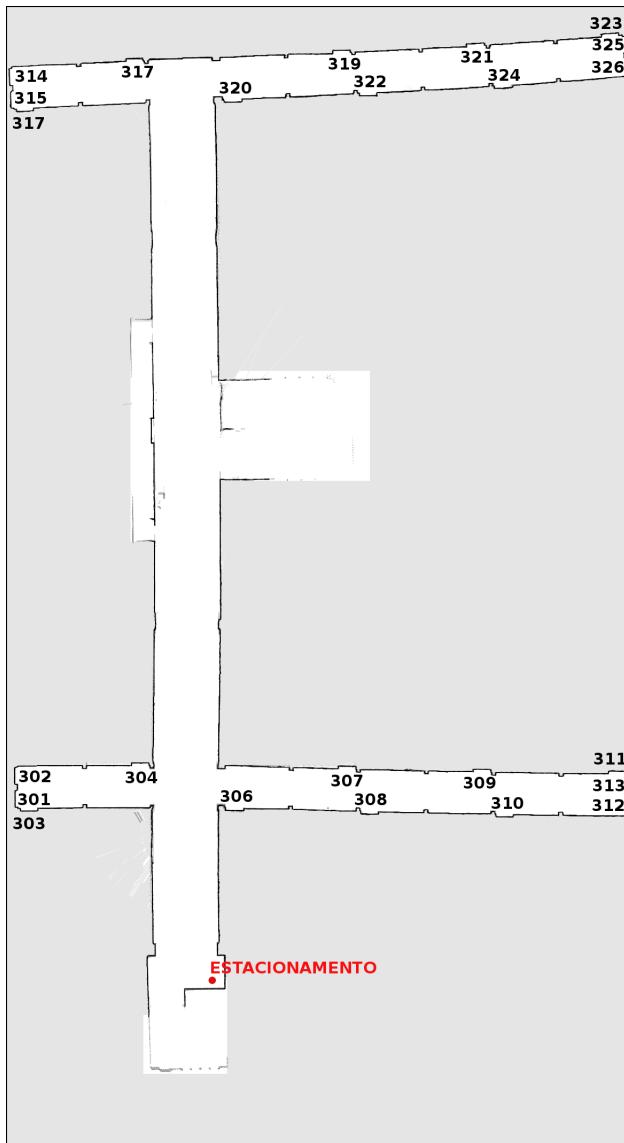
### B.1.1 Posição das salas

A Tabela 12 trás as coordenadas dos locais utilizados nos experimentos, e a Figura 37 ilustra a posição destes locais no mapa.

Tabela 12 – Tabela contendo a descrição e coordenadas dos locais utilizados nos experimentos.

Descrição do local	X	Y
Sala 301	40	652
Sala 302	40	696
Sala 303	44	650
Sala 304	250	698
Sala 306	430	651
Sala 307	637	692
Sala 308	681	646
Sala 309	890	686
Sala 310	932	639
Sala 311	1140	682
Sala 312	1147	636
Sala 313	1147	680
Sala 314	28	2003
Sala 315	28	1960
Sala 316	240	2016
Sala 317	34	1961
Sala 319	628	2031
Sala 320	425	1978
Sala 321	878	2043
Sala 322	678	1991
Sala 323	1130	2061
Sala 324	928	2004
Sala 325	1141	2064
Sala 326	1141	2022

Figura 37 – Posição dos locais no mapa.



Fonte: Do Autor

### B.1.2 Lista de tarefas

A Tabela 13 apresenta a lista de tarefas que foram utilizadas nos experimentos. A utilização das mesmas foi acumulativa ou seja, experimentos utilizaram as tarefas com identificador entre 1 e 5, 1 e 10 e assim sucessivamente.

Tabela 13 – Lista tarefas utilizadas nos experimentos. Id significa identificador e, L1, L2 e L3 são os locais a serem visitados.

<b>Id.</b>	<b>Descrição</b>	<b>Prioridade</b>	<b>L1</b>	<b>L2</b>	<b>L3</b>
1	Task 1	4	303	313	
2	Task 2	1	309	306	
3	Task 3	3	306	312	
4	Task 4	1	315	311	
5	Task 5	5	314	311	322
6	Task 6	5	314	323	
7	Task 7	4	317	310	
8	Task 8	5	312	303	
9	Task 9	4	322	308	
10	Task 10	3	313	316	322
11	Task 11	2	321	303	
12	Task 12	3	304	326	
13	Task 13	1	320	325	
14	Task 14	2	319	301	
15	Task 15	3	Estacionamento	307	323
16	Task 16	3	314	321	
17	Task 17	4	315	309	
18	Task 18	4	315	312	
19	Task 19	4	316	323	
20	Task 20	3	320	313	
21	Task 21	3	308	313	
22	Task 22	2	313	317	
23	Task 23	2	307	311	
24	Task 24	2	314	303	
25	Task 25	4	303	308	
26	Task 26	1	308	315	
27	Task 27	1	319	310	
28	Task 28	2	319	313	316
29	Task 29	1	316	311	317

<b>Continuação da Tabela 13</b>					
<b>Id.</b>	<b>Descrição</b>	<b>Prioridade</b>	<b>L1</b>	<b>L2</b>	<b>L3</b>
30	Task 30	4	313	309	317

### B.1.3 Experimento 1

A Tabela 14 traz os resultados referentes ao tempo necessário para escalonar cinco, dez, quinze e trinta tarefas respeitando a restrição de que 80% das tarefas devem conter dois locais e 20% três. A medida para cinco tarefas foi efetuada para o mapa redimensionado para a um meio e um terço do tamanho original.

Tabela 14 – Medida de tempo para o escalonamento.

<b>Número de Tarefas</b>	<b>5 (1/2)</b>	<b>5 (1/3)</b>	<b>10</b>	<b>15</b>	<b>30</b>
<b>Medição (ms)</b>	6749	2679	10703	23834	90491
	6691	2801	10745	23944	91851
	6772	2760	10785	23988	92529
	6695	2676	10917	23743	91592
	6733	2712	10824	23869	91213
<b>Média</b>	6728,00	2725,60	10794,80	23875,60	91535,20
<b>Desvio Padrão</b>	34,86	54,05	81,84	95,75	755,77

As Tabelas 15, 16, 17, 18 e 19 apresentam as etapas do escalonamento para os grupos de cinco com redução no mapa para um meio, cinco com redução no mapa para um terço, dez, quinze e trinta tarefas respectivamente.

Tabela 15 – Escalonamento para o conjunto de cinco tarefas, com redução no mapa para um meio.

<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
1	4	855,88	213,97
2	1	659,27	659,27
3	3	545,38	181,79
4	1	2.118,71	2.118,71
5	5	3.299,83	659,96

<b>Continuação da Tabela 15</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
1	4	1.111,02	277,76
2	1	375,63	375,63
4	1	2.321,93	2.321,93
5	5	3.503,04	700,61
2	1	366,61	366,61
4	1	2.312,91	2.312,91
5	5	3.494,02	698,80
4	1	1.965,78	1.965,78
5	5	3.146,89	629,38
4	1	1.485,90	1.485,90

Tabela 16 – Escalonamento para o conjunto de cinco tarefas, com redução no mapa para um terço.

<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
1	4	571,45	142,86
2	1	438,89	438,89
3	3	364,92	121,64
4	1	1.414,49	1.414,49
5	5	2.200,30	440,06
1	4	741,74	185,44
2	1	250,07	250,07
4	1	1.549,97	1.549,97
5	5	2.335,78	467,16
2	1	244,33	244,33
4	1	1.544,23	1.544,23
5	5	2.330,04	466,01
4	1	1.311,87	1.311,87
5	5	2.097,68	419,54
4	1	992,01	992,01

Tabela 17 – Escalonamento para o conjunto de dez tarefas.

<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
1	4	571,45	142,86
2	1	438,89	438,89
3	3	364,92	121,64
4	1	1.414,49	1.414,49
5	5	2.200,30	440,06
6	5	1.022,88	204,58
7	4	1.346,23	336,56
8	5	732,92	146,58
9	4	1.280,63	320,16
10	3	1.229,44	409,81
<hr/>			
1	4	741,74	185,44
2	1	250,07	250,07
4	1	1.549,97	1.549,97
5	5	2.335,78	467,16
6	5	1.158,36	231,67
7	4	1.481,71	370,43
8	5	369,64	73,93
9	4	1.415,29	353,82
10	3	876,88	292,29
<hr/>			
1	4	372,10	93,03
2	1	444,43	444,43
4	1	1.364,26	1.364,26
5	5	2.150,07	430,01
6	5	972,65	194,53
7	4	1.296,00	324,00
9	4	1.252,54	313,14
10	3	1.234,98	411,66
<hr/>			
2	1	244,33	244,33
4	1	1.544,23	1.544,23
5	5	2.330,04	466,01
6	5	1.152,62	230,52
7	4	1.475,97	368,99
9	4	1.409,55	352,39
10	3	862,88	287,63

<b>Continuação da Tabela 17</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
2	1	1.010,55	1.010,55
4	1	1.151,85	1.151,85
5	5	1.927,82	385,56
7	4	1.083,59	270,90
9	4	789,37	197,34
10	3	1.802,74	600,91
2	1	231,84	231,84
4	1	1.393,74	1.393,74
5	5	2.179,55	435,91
7	4	1.325,48	331,37
10	3	1.022,39	340,80
4	1	1.311,87	1.311,87
5	5	2.097,68	419,54
7	4	1.243,61	310,90
10	3	1.105,57	368,52
4	1	1.477,56	1.477,56
5	5	2.263,37	452,67
10	3	940,21	313,40
4	1	992,01	992,01
5	5	1.771,26	354,25
4	1	992,01	992,01

Tabela 18 – Escalonamento para o conjunto de quinze tarefas.

<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
1	4	571,45	142,86
2	1	438,89	438,89
3	3	364,92	121,64
4	1	1.414,49	1.414,49
5	5	2.200,30	440,06
6	5	1.022,88	204,58

<b>Continuação da Tabela 18</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
7	4	1.346,23	336,56
8	5	732,92	146,58
9	4	1.280,63	320,16
10	3	1.229,44	409,81
11	2	1.415,31	707,66
12	3	848,18	282,73
13	1	815,35	815,35
14	2	1.252,18	626,09
15	3	891,63	297,21
<hr/>			
1	4	741,74	185,44
2	1	250,07	250,07
4	1	1.549,97	1.549,97
5	5	2.335,78	467,16
6	5	1.158,36	231,67
7	4	1.481,71	370,43
8	5	369,64	73,93
9	4	1.415,29	353,82
10	3	876,88	292,29
11	2	1.549,97	774,99
12	3	1.013,16	337,72
13	1	950,01	950,01
14	2	1.386,84	693,42
15	3	1.252,91	417,64
<hr/>			
1	4	372,10	93,03
2	1	444,43	444,43
4	1	1.364,26	1.364,26
5	5	2.150,07	430,01
6	5	972,65	194,53
7	4	1.296,00	324,00
9	4	1.252,54	313,14
10	3	1.234,98	411,66
11	2	1.387,22	693,61
12	3	781,52	260,51
13	1	787,26	787,26
14	2	1.224,09	612,05
15	3	1.088,98	362,99

Continuação da Tabela 18			
Id.	Prioridade	Distância	Custo
2	1	244,33	244,33
4	1	1.544,23	1.544,23
5	5	2.330,04	466,01
6	5	1.152,62	230,52
7	4	1.475,97	368,99
9	4	1.409,55	352,39
10	3	862,88	287,63
11	2	1.544,23	772,12
12	3	1.007,42	335,81
13	1	944,27	944,27
14	2	1.381,10	690,55
15	3	1.256,19	418,73
<hr/>			
2	1	1.010,55	1.010,55
4	1	1.151,85	1.151,85
5	5	1.927,82	385,56
7	4	1.083,59	270,90
9	4	789,37	197,34
10	3	1.802,74	600,91
11	2	777,29	388,65
12	3	1.413,25	471,08
13	1	497,37	497,37
14	2	783,44	391,72
15	3	1.700,57	566,86
<hr/>			
2	1	231,84	231,84
4	1	1.393,74	1.393,74
5	5	2.179,55	435,91
7	4	1.325,48	331,37
10	3	1.022,39	340,80
11	2	1.393,74	696,87
12	3	856,93	285,64
13	1	793,78	793,78
14	2	1.230,61	615,31
15	3	1.096,68	365,56
<hr/>			
4	1	1.311,87	1.311,87

<b>Continuação da Tabela 18</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
5	5	2.097,68	419,54
7	4	1.243,61	310,90
10	3	1.105,57	368,52
11	2	1.311,87	655,94
12	3	772,11	257,37
13	1	711,91	711,91
14	2	1.148,74	574,37
15	3	1.013,50	337,83
<hr/>			
4	1	1.150,52	1.150,52
5	5	1.926,49	385,30
7	4	1.082,26	270,57
10	3	1.801,41	600,47
11	2	781,70	390,85
13	1	496,04	496,04
14	2	784,57	392,29
15	3	1.699,24	566,41
<hr/>			
4	1	1.477,56	1.477,56
5	5	2.263,37	452,67
10	3	940,21	313,40
11	2	1.477,56	738,78
13	1	877,60	877,60
14	2	1.314,43	657,22
15	3	1.180,50	393,50
<hr/>			
4	1	992,01	992,01
5	5	1.771,26	354,25
11	2	764,21	382,11
13	1	337,53	337,53
14	2	635,08	317,54
15	3	1.540,73	513,58
<hr/>			
4	1	1.364,85	1.364,85
5	5	2.150,66	430,13
11	2	1.387,81	693,91
13	1	787,85	787,85
15	3	1.090,39	363,46

<b>Continuação da Tabela 18</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
4	1	1.077,06	1.077,06
5	5	1.853,03	370,61
11	2	713,16	356,58
13	1	422,58	422,58
4	1	1.358,70	1.358,70
5	5	2.144,51	428,90
13	1	781,70	781,70
4	1	992,01	992,01
13	1	337,53	337,53
4	1	1.156,26	1.156,26

Tabela 19 – Escalonamento para o conjunto de trinta tarefas.

<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
1	4	571,45	142,86
2	1	438,89	438,89
3	3	364,92	121,64
4	1	1.414,49	1.414,49
5	5	2.200,30	440,06
6	5	1.022,88	204,58
7	4	1.346,23	336,56
8	5	732,92	146,58
9	4	1.280,63	320,16
10	3	1.229,44	409,81
11	2	1.415,31	707,66
12	3	848,18	282,73
13	1	815,35	815,35
14	2	1.252,18	626,09
15	3	891,63	297,21
16	3	936,42	312,14
17	4	1.330,08	332,52
18	4	1.422,64	355,66
19	4	885,60	221,40

<b>Continuação da Tabela 19</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
20	3	1.257,84	419,28
21	3	366,56	122,19
22	2	1.137,88	568,94
23	2	367,43	183,72
24	2	664,11	332,06
25	4	412,76	103,19
26	1	829,88	829,88
27	1	1.341,93	1.341,93
28	2	2.121,79	1.060,89
29	1	2.063,14	2.063,14
30	4	1.137,88	284,47
1	4	585,51	146,38
2	1	231,84	231,84
3	3	325,87	108,62
4	1	1.393,74	1.393,74
5	5	2.179,55	435,91
6	5	1.002,13	200,43
7	4	1.325,48	331,37
8	5	525,87	105,17
9	4	1.259,06	314,77
10	3	1.022,39	340,80
11	2	1.393,74	696,87
12	3	856,93	285,64
13	1	793,78	793,78
14	2	1.230,61	615,31
15	3	1.096,68	365,56
16	3	915,67	305,22
17	4	1.309,33	327,33
18	4	1.401,89	350,47
19	4	864,85	216,21
20	3	1.236,27	412,09
21	3	159,51	53,17
22	2	930,83	465,42
23	2	190,38	95,19
24	2	672,86	336,43
26	1	622,83	622,83
27	1	1.320,36	1.320,36

<b>Continuação da Tabela 19</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
28	2	2.100,22	1.050,11
29	1	2.042,39	2.042,39
30	4	930,83	232,71
1	4	744,20	186,05
2	1	244,33	244,33
3	3	483,74	161,25
4	1	1.544,23	1.544,23
5	5	2.330,04	466,01
6	5	1.152,62	230,52
7	4	1.475,97	368,99
8	5	383,64	76,73
9	4	1.409,55	352,39
10	3	862,88	287,63
11	2	1.544,23	772,12
12	3	1.007,42	335,81
13	1	944,27	944,27
14	2	1.381,10	690,55
15	3	1.256,19	418,73
16	3	1.066,16	355,39
17	4	1.459,82	364,96
18	4	1.552,38	388,10
19	4	1.015,34	253,84
20	3	1.386,76	462,25
22	2	771,32	385,66
23	2	340,87	170,44
24	2	823,35	411,68
26	1	782,34	782,34
27	1	1.470,85	1.470,85
28	2	2.250,71	1.125,35
29	1	2.192,88	2.192,88
30	4	771,32	192,83
1	4	372,10	93,03
2	1	444,43	444,43
3	3	370,46	123,49
4	1	1.364,26	1.364,26
5	5	2.150,07	430,01

<b>Continuação da Tabela 19</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
6	5	972,65	194,53
7	4	1.296,00	324,00
9	4	1.252,54	313,14
10	3	1.234,98	411,66
11	2	1.387,22	693,61
12	3	781,52	260,51
13	1	787,26	787,26
14	2	1.224,09	612,05
15	3	1.088,98	362,99
16	3	886,19	295,40
17	4	1.279,85	319,96
18	4	1.372,41	343,10
19	4	835,37	208,84
20	3	1.229,75	409,92
22	2	1.143,42	571,71
23	2	372,97	186,49
24	2	597,45	298,73
26	1	836,24	836,24
27	1	1.313,84	1.313,84
28	2	2.093,70	1.046,85
29	1	2.012,91	2.012,91
30	4	1.143,42	285,86
2	1	244,33	244,33
3	3	483,74	161,25
4	1	1.544,23	1.544,23
5	5	2.330,04	466,01
6	5	1.152,62	230,52
7	4	1.475,97	368,99
9	4	1.409,55	352,39
10	3	862,88	287,63
11	2	1.544,23	772,12
12	3	1.007,42	335,81
13	1	944,27	944,27
14	2	1.381,10	690,55
15	3	1.256,19	418,73
16	3	1.066,16	355,39
17	4	1.459,82	364,96

<b>Continuação da Tabela 19</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
18	4	1.552,38	388,10
19	4	1.015,34	253,84
20	3	1.386,76	462,25
22	2	771,32	385,66
23	2	340,87	170,44
24	2	823,35	411,68
26	1	782,34	782,34
27	1	1.470,85	1.470,85
28	2	2.250,71	1.125,35
29	1	2.192,88	2.192,88
30	4	771,32	192,83
2	1	250,07	250,07
4	1	1.549,97	1.549,97
5	5	2.335,78	467,16
6	5	1.158,36	231,67
7	4	1.481,71	370,43
9	4	1.415,29	353,82
10	3	876,88	292,29
11	2	1.549,97	774,99
12	3	1.013,16	337,72
13	1	950,01	950,01
14	2	1.386,84	693,42
15	3	1.252,91	417,64
16	3	1.071,90	357,30
17	4	1.465,56	366,39
18	4	1.558,12	389,53
19	4	1.021,08	255,27
20	3	1.392,50	464,17
22	2	785,32	392,66
23	2	346,61	173,31
24	2	829,09	414,55
26	1	779,06	779,06
27	1	1.476,59	1.476,59
28	2	2.256,45	1.128,22
29	1	2.198,62	2.198,62
30	4	785,32	196,33

<b>Continuação da Tabela 19</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
2	1	241,92	241,92
4	1	1.541,82	1.541,82
5	5	2.327,63	465,53
6	5	1.150,21	230,04
7	4	1.473,56	368,39
9	4	1.407,14	351,79
10	3	865,29	288,43
11	2	1.541,82	770,91
12	3	1.005,01	335,00
13	1	941,86	941,86
14	2	1.378,69	689,35
15	3	1.254,60	418,20
16	3	1.063,75	354,58
17	4	1.457,41	364,35
18	4	1.549,97	387,49
19	4	1.012,93	253,23
20	3	1.384,35	461,45
22	2	773,73	386,87
24	2	820,94	410,47
26	1	780,75	780,75
27	1	1.468,44	1.468,44
28	2	2.248,30	1.124,15
29	1	2.190,47	2.190,47
30	4	773,73	193,43
<hr/>			
2	1	842,01	842,01
4	1	772,91	772,91
5	5	1.567,44	313,49
6	5	390,02	78,00
7	4	704,65	176,16
9	4	848,63	212,16
10	3	1.634,20	544,73
11	2	983,31	491,66
12	3	1.221,75	407,25
13	1	383,35	383,35
14	2	820,18	410,09
15	3	1.531,21	510,40
16	3	303,56	101,19

Continuação da Tabela 19			
Id.	Prioridade	Distância	Custo
17	4	688,50	172,13
18	4	781,06	195,27
19	4	378,94	94,74
20	3	825,84	275,28
22	2	1.542,64	771,32
24	2	1.037,68	518,84
26	1	1.243,66	1.243,66
27	1	909,93	909,93
28	2	1.689,79	844,90
29	1	1.556,48	1.556,48
<hr/>			
2	1	1.010,55	1.010,55
4	1	1.151,85	1.151,85
5	5	1.927,82	385,56
7	4	1.083,59	270,90
9	4	789,37	197,34
10	3	1.802,74	600,91
11	2	777,29	388,65
12	3	1.413,25	471,08
13	1	497,37	497,37
14	2	783,44	391,72
15	3	1.700,57	566,86
16	3	663,94	221,31
17	4	1.067,44	266,86
18	4	1.160,00	290,00
19	4	604,30	151,08
20	3	939,86	313,29
22	2	1.711,18	855,59
24	2	1.229,18	614,59
26	1	1.412,20	1.412,20
27	1	873,19	873,19
28	2	1.653,05	826,53
29	1	1.781,84	1.781,84
<hr/>			
2	1	1.010,55	1.010,55
4	1	1.151,85	1.151,85
5	5	1.927,82	385,56
7	4	1.083,59	270,90

<b>Continuação da Tabela 19</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
9	4	789,37	197,34
10	3	1.802,74	600,91
11	2	777,29	388,65
12	3	1.413,25	471,08
13	1	497,37	497,37
14	2	783,44	391,72
15	3	1.700,57	566,86
16	3	663,94	221,31
17	4	1.067,44	266,86
18	4	1.160,00	290,00
20	3	939,86	313,29
22	2	1.711,18	855,59
24	2	1.229,18	614,59
26	1	1.412,20	1.412,20
27	1	873,19	873,19
28	2	1.653,05	826,53
29	1	1.781,84	1.781,84
<hr/>			
2	1	231,84	231,84
4	1	1.393,74	1.393,74
5	5	2.179,55	435,91
7	4	1.325,48	331,37
10	3	1.022,39	340,80
11	2	1.393,74	696,87
12	3	856,93	285,64
13	1	793,78	793,78
14	2	1.230,61	615,31
15	3	1.096,68	365,56
16	3	915,67	305,22
17	4	1.309,33	327,33
18	4	1.401,89	350,47
20	3	1.236,27	412,09
22	2	930,83	465,42
24	2	672,86	336,43
26	1	622,83	622,83
27	1	1.320,36	1.320,36
28	2	2.100,22	1.050,11
29	1	2.042,39	2.042,39

<b>Continuação da Tabela 19</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
4	1	1.311,87	1.311,87
5	5	2.097,68	419,54
7	4	1.243,61	310,90
10	3	1.105,57	368,52
11	2	1.311,87	655,94
12	3	772,11	257,37
13	1	711,91	711,91
14	2	1.148,74	574,37
15	3	1.013,50	337,83
16	3	833,80	277,93
17	4	1.227,46	306,87
18	4	1.320,02	330,01
20	3	1.154,40	384,80
22	2	1.014,01	507,01
24	2	588,04	294,02
26	1	707,65	707,65
27	1	1.238,49	1.238,49
28	2	2.018,35	1.009,18
29	1	1.960,52	1.960,52
<hr/>			
4	1	1.150,52	1.150,52
5	5	1.926,49	385,30
7	4	1.082,26	270,57
10	3	1.801,41	600,47
11	2	781,70	390,85
13	1	496,04	496,04
14	2	784,57	392,29
15	3	1.699,24	566,41
16	3	662,61	220,87
17	4	1.066,11	266,53
18	4	1.158,67	289,67
20	3	938,53	312,84
22	2	1.709,85	854,93
24	2	1.227,85	613,93
26	1	1.410,87	1.410,87
27	1	874,32	874,32
28	2	1.654,18	827,09

<b>Continuação da Tabela 19</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
29	1	1.780,51	1.780,51
4	1	1.065,39	1.065,39
5	5	1.841,36	368,27
7	4	997,13	249,28
10	3	1.716,28	572,09
11	2	690,83	345,42
13	1	410,91	410,91
14	2	696,98	348,49
15	3	1.614,11	538,04
17	4	980,98	245,25
18	4	1.073,54	268,39
20	3	853,40	284,47
22	2	1.624,72	812,36
24	2	1.142,72	571,36
26	1	1.325,74	1.325,74
27	1	786,73	786,73
28	2	1.566,59	783,30
29	1	1.695,38	1.695,38
4	1	1.457,41	1.457,41
5	5	2.243,22	448,64
7	4	1.389,15	347,29
10	3	949,70	316,57
11	2	1.457,41	728,71
13	1	857,45	857,45
14	2	1.294,28	647,14
15	3	1.171,01	390,34
18	4	1.465,56	366,39
20	3	1.299,94	433,31
22	2	858,14	429,07
24	2	736,53	368,27
26	1	697,16	697,16
27	1	1.384,03	1.384,03
28	2	2.163,89	1.081,94
29	1	2.106,06	2.106,06
4	1	992,01	992,01

<b>Continuação da Tabela 19</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
5	5	1.771,26	354,25
7	4	923,75	230,94
11	2	764,21	382,11
13	1	337,53	337,53
14	2	635,08	317,54
15	3	1.540,73	513,58
18	4	1.000,16	250,04
20	3	780,02	260,01
22	2	1.551,34	775,67
24	2	1.069,34	534,67
26	1	1.252,36	1.252,36
27	1	724,83	724,83
28	2	1.504,69	752,35
29	1	1.629,38	1.629,38
<hr/>			
4	1	1.477,56	1.477,56
5	5	2.263,37	452,67
11	2	1.477,56	738,78
13	1	877,60	877,60
14	2	1.314,43	657,22
15	3	1.180,50	393,50
18	4	1.485,71	371,43
20	3	1.320,09	440,03
22	2	848,65	424,33
24	2	756,68	378,34
26	1	706,65	706,65
27	1	1.404,18	1.404,18
28	2	2.184,04	1.092,02
29	1	2.126,21	2.126,21
<hr/>			
4	1	1.549,97	1.549,97
5	5	2.335,78	467,16
11	2	1.549,97	774,99
13	1	950,01	950,01
14	2	1.386,84	693,42
15	3	1.252,91	417,64
20	3	1.392,50	464,17
22	2	785,32	392,66

<b>Continuação da Tabela 19</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
24	2	829,09	414,55
26	1	779,06	779,06
27	1	1.476,59	1.476,59
28	2	2.256,45	1.128,22
29	1	2.198,62	2.198,62
4	1	772,91	772,91
5	5	1.567,44	313,49
11	2	983,31	491,66
13	1	383,35	383,35
14	2	820,18	410,09
15	3	1.531,21	510,40
20	3	825,84	275,28
24	2	1.037,68	518,84
26	1	1.243,66	1.243,66
27	1	909,93	909,93
28	2	1.689,79	844,90
29	1	1.556,48	1.556,48
4	1	1.544,23	1.544,23
5	5	2.330,04	466,01
11	2	1.544,23	772,12
13	1	944,27	944,27
14	2	1.381,10	690,55
15	3	1.256,19	418,73
24	2	823,35	411,68
26	1	782,34	782,34
27	1	1.470,85	1.470,85
28	2	2.250,71	1.125,35
29	1	2.192,88	2.192,88
4	1	784,91	784,91
5	5	1.552,62	310,52
11	2	979,57	489,79
13	1	386,17	386,17
14	2	816,44	408,22
15	3	1.537,31	512,44
26	1	1.249,76	1.249,76

<b>Continuação da Tabela 19</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
27	1	906,19	906,19
28	2	1.686,05	843,03
29	1	1.552,74	1.552,74
4	1	992,01	992,01
11	2	764,21	382,11
13	1	337,53	337,53
14	2	635,08	317,54
15	3	1.540,73	513,58
26	1	1.252,36	1.252,36
27	1	724,83	724,83
28	2	1.504,69	752,35
29	1	1.629,38	1.629,38
4	1	1.364,85	1.364,85
11	2	1.387,81	693,91
13	1	787,85	787,85
15	3	1.090,39	363,46
26	1	837,65	837,65
27	1	1.314,43	1.314,43
28	2	2.094,29	1.047,14
29	1	2.013,50	2.013,50
4	1	1.077,06	1.077,06
11	2	713,16	356,58
13	1	422,58	422,58
26	1	1.337,41	1.337,41
27	1	805,78	805,78
28	2	1.585,64	792,82
29	1	1.710,33	1.710,33
4	1	1.358,70	1.358,70
13	1	781,70	781,70
26	1	843,80	843,80
27	1	1.308,28	1.308,28
28	2	2.088,14	1.044,07
29	1	2.007,35	2.007,35

<b>Continuação da Tabela 19</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
4	1	1.156,26	1.156,26
26	1	1.416,61	1.416,61
27	1	877,60	877,60
28	2	1.657,46	828,73
29	1	1.786,25	1.786,25
<hr/>			
4	1	849,70	849,70
26	1	1.185,53	1.185,53
27	1	833,14	833,14
29	1	1.479,69	1.479,69
<hr/>			
4	1	1.477,56	1.477,56
26	1	706,65	706,65
29	1	2.126,21	2.126,21
<hr/>			
4	1	770,91	770,91
29	1	1.558,48	1.558,48
<hr/>			
29	1	2.190,47	2.190,47

#### B.1.4 Experimento 2

A Tabela 20 traz as medidas de tempo para escalonamento para o grupo inicial e após a adição das tarefas ao conjunto para os grupos cinco até dez tarefas, dez até quinze e quinze até trinta.

Tabela 20 – Medida de tempo para o escalonamento.

<b>Número de Tarefas</b>	<b>5 até 10</b>		<b>10 até 15</b>		<b>15 até 30</b>	
	<b>5</b>	<b>10</b>	<b>10</b>	<b>15</b>	<b>15</b>	<b>30</b>
<b>Medição (ms)</b>	2711	10477	10860	23319	24087	89115
	2785	10666	10905	22889	23826	89052
	2744	10540	10821	22866	23981	89813
	2802	10556	10714	22614	23924	89417
	2712	10431	10907	23141	23960	89109
<b>Média</b>	2750,80	10534,00	10841,40	22965,80	23955,60	89301,20
<b>Desvio Padrão</b>	41,61	89,17	79,59	271,60	94,51	319,79

O escalonamento do estado inicial dos grupos: cinco até dez tarefas, dez até quinze e quinze até trinta é respectivamente o mesmo presente nas Tabelas 16, 17 e 18.

As Tabelas 21, 22 e 23 apresentam as etapas do escalonamento após a adição das tarefas no conjunto inicial para os grupos cinco até dez tarefas, dez até quinze e quinze até trinta respectivamente.

Tabela 21 – Escalonamento para o conjunto de cinco até dez tarefas.

<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
1	4	501,51	125,38
2	1	315,02	315,02
3	3	241,05	80,35
4	1	1.311,87	1.311,87
5	5	2.097,68	419,54
6	5	920,26	184,05
7	4	1.243,61	310,90
8	5	610,69	122,14
9	4	1.177,19	294,30
10	3	1.105,57	368,52
<hr/>			
1	4	741,74	185,44
2	1	250,07	250,07
4	1	1.549,97	1.549,97
5	5	2.335,78	467,16
6	5	1.158,36	231,67
7	4	1.481,71	370,43
8	5	369,64	73,93
9	4	1.415,29	353,82
10	3	876,88	292,29
<hr/>			
1	4	372,10	93,03
2	1	444,43	444,43
4	1	1.364,26	1.364,26
5	5	2.150,07	430,01
6	5	972,65	194,53
7	4	1.296,00	324,00
9	4	1.252,54	313,14
10	3	1.234,98	411,66
<hr/>			

<b>Continuação da Tabela 21</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
2	1	244,33	244,33
4	1	1.544,23	1.544,23
5	5	2.330,04	466,01
6	5	1.152,62	230,52
7	4	1.475,97	368,99
9	4	1.409,55	352,39
10	3	862,88	287,63
<hr/>			
2	1	1.010,55	1.010,55
4	1	1.151,85	1.151,85
5	5	1.927,82	385,56
7	4	1.083,59	270,90
9	4	789,37	197,34
10	3	1.802,74	600,91
<hr/>			
2	1	231,84	231,84
4	1	1.393,74	1.393,74
5	5	2.179,55	435,91
7	4	1.325,48	331,37
10	3	1.022,39	340,80
<hr/>			
4	1	1.311,87	1.311,87
5	5	2.097,68	419,54
7	4	1.243,61	310,90
10	3	1.105,57	368,52
<hr/>			
4	1	1.477,56	1.477,56
5	5	2.263,37	452,67
10	3	940,21	313,40
<hr/>			
4	1	992,01	992,01
5	5	1.771,26	354,25
<hr/>			
4	1	992,01	992,01

Tabela 22 – Escalonamento para o conjunto de dez até quinze tarefas.

<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
1	4	501,51	125,38
2	1	315,02	315,02
3	3	241,05	80,35
4	1	1.311,87	1.311,87
5	5	2.097,68	419,54
6	5	920,26	184,05
7	4	1.243,61	310,90
8	5	610,69	122,14
9	4	1.177,19	294,30
10	3	1.105,57	368,52
11	2	1.311,87	655,94
12	3	772,11	257,37
13	1	711,91	711,91
14	2	1.148,74	574,37
15	3	1.013,50	337,83
<hr/>			
1	4	741,74	185,44
2	1	250,07	250,07
4	1	1.549,97	1.549,97
5	5	2.335,78	467,16
6	5	1.158,36	231,67
7	4	1.481,71	370,43
8	5	369,64	73,93
9	4	1.415,29	353,82
10	3	876,88	292,29
11	2	1.549,97	774,99
12	3	1.013,16	337,72
13	1	950,01	950,01
14	2	1.386,84	693,42
15	3	1.252,91	417,64
<hr/>			
1	4	372,10	93,03
2	1	444,43	444,43
4	1	1.364,26	1.364,26
5	5	2.150,07	430,01
6	5	972,65	194,53
7	4	1.296,00	324,00

<b>Continuação da Tabela 22</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
9	4	1.252,54	313,14
10	3	1.234,98	411,66
11	2	1.387,22	693,61
12	3	781,52	260,51
13	1	787,26	787,26
14	2	1.224,09	612,05
15	3	1.088,98	362,99
<hr/>			
2	1	244,33	244,33
4	1	1.544,23	1.544,23
5	5	2.330,04	466,01
6	5	1.152,62	230,52
7	4	1.475,97	368,99
9	4	1.409,55	352,39
10	3	862,88	287,63
11	2	1.544,23	772,12
12	3	1.007,42	335,81
13	1	944,27	944,27
14	2	1.381,10	690,55
15	3	1.256,19	418,73
<hr/>			
2	1	1.010,55	1.010,55
4	1	1.151,85	1.151,85
5	5	1.927,82	385,56
7	4	1.083,59	270,90
9	4	789,37	197,34
10	3	1.802,74	600,91
11	2	777,29	388,65
12	3	1.413,25	471,08
13	1	497,37	497,37
14	2	783,44	391,72
15	3	1.700,57	566,86
<hr/>			
2	1	231,84	231,84
4	1	1.393,74	1.393,74
5	5	2.179,55	435,91
7	4	1.325,48	331,37
10	3	1.022,39	340,80

<b>Continuação da Tabela 22</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
11	2	1.393,74	696,87
12	3	856,93	285,64
13	1	793,78	793,78
14	2	1.230,61	615,31
15	3	1.096,68	365,56
<hr/>			
4	1	1.311,87	1.311,87
5	5	2.097,68	419,54
7	4	1.243,61	310,90
10	3	1.105,57	368,52
11	2	1.311,87	655,94
12	3	772,11	257,37
13	1	711,91	711,91
14	2	1.148,74	574,37
15	3	1.013,50	337,83
<hr/>			
4	1	1.150,52	1.150,52
5	5	1.926,49	385,30
7	4	1.082,26	270,57
10	3	1.801,41	600,47
11	2	781,70	390,85
13	1	496,04	496,04
14	2	784,57	392,29
15	3	1.699,24	566,41
<hr/>			
4	1	1.477,56	1.477,56
5	5	2.263,37	452,67
10	3	940,21	313,40
11	2	1.477,56	738,78
13	1	877,60	877,60
14	2	1.314,43	657,22
15	3	1.180,50	393,50
<hr/>			
4	1	992,01	992,01
5	5	1.771,26	354,25
11	2	764,21	382,11
13	1	337,53	337,53
14	2	635,08	317,54

<b>Continuação da Tabela 22</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
15	3	1.540,73	513,58
4	1	1.364,85	1.364,85
5	5	2.150,66	430,13
11	2	1.387,81	693,91
13	1	787,85	787,85
15	3	1.090,39	363,46
4	1	1.077,06	1.077,06
5	5	1.853,03	370,61
11	2	713,16	356,58
13	1	422,58	422,58
4	1	1.358,70	1.358,70
5	5	2.144,51	428,90
13	1	781,70	781,70
4	1	992,01	992,01
13	1	337,53	337,53
4	1	1.156,26	1.156,26

Tabela 23 – Escalonamento para o conjunto de quinze até trinta tarefas.

<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
1	4	501,51	125,38
2	1	315,02	315,02
3	3	241,05	80,35
4	1	1.311,87	1.311,87
5	5	2.097,68	419,54
6	5	920,26	184,05
7	4	1.243,61	310,90
8	5	610,69	122,14
9	4	1.177,19	294,30
10	3	1.105,57	368,52
11	2	1.311,87	655,94
12	3	772,11	257,37

<b>Continuação da Tabela 23</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
13	1	711,91	711,91
14	2	1.148,74	574,37
15	3	1.013,50	337,83
16	3	833,80	277,93
17	4	1.227,46	306,87
18	4	1.320,02	330,01
19	4	782,98	195,75
20	3	1.154,40	384,80
21	3	244,33	814,43
22	2	1.014,01	507,01
23	2	243,56	121,78
24	2	588,04	294,02
25	4	342,82	85,71
26	1	707,65	707,65
27	1	1.238,49	1.238,49
28	2	2.018,35	1.009,18
29	1	1.960,52	1.960,52
30	4	1.014,01	253,50
1	4	741,74	185,44
2	1	250,07	250,07
4	1	1.549,97	1.549,97
5	5	2.335,78	467,16
6	5	1.158,36	231,67
7	4	1.481,71	370,43
8	5	369,64	73,93
9	4	1.415,29	353,82
10	3	876,88	292,29
11	2	1.549,97	774,99
12	3	1.013,16	337,72
13	1	950,01	950,01
14	2	1.386,84	693,42
15	3	1.252,91	417,64
16	3	1.071,90	357,30
17	4	1.465,56	366,39
18	4	1.558,12	389,53
19	4	1.021,08	255,27
20	3	1.392,50	464,17

<b>Continuação da Tabela 23</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
21	3	315,74	105,25
22	2	785,32	392,66
23	2	346,61	173,31
24	2	829,09	414,55
25	4	583,05	145,76
26	1	779,06	779,06
27	1	1.476,59	1.476,59
28	2	2.256,45	1.128,22
29	1	2.198,62	2.198,62
30	4	785,32	196,33
<hr/>			
1	4	372,10	93,03
2	1	444,43	444,43
4	1	1.364,26	1.364,26
5	5	2.150,07	430,01
6	5	972,65	194,53
7	4	1.296,00	324,00
9	4	1.252,54	313,14
10	3	1.234,98	411,66
11	2	1.387,22	693,61
12	3	781,52	260,51
13	1	787,26	787,26
14	2	1.224,09	612,05
15	3	1.088,98	362,99
16	3	886,19	295,40
17	4	1.279,85	319,96
18	4	1.372,41	343,10
19	4	835,37	208,84
20	3	1.229,75	409,92
21	3	372,92	124,31
22	2	1.143,42	571,71
23	2	372,97	186,49
24	2	597,45	298,73
25	4	213,41	533,53
26	1	836,24	836,24
27	1	1.313,84	1.313,84
28	2	2.093,70	1.046,85
29	1	2.012,91	2.012,91

<b>Continuação da Tabela 23</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
30	4	1.143,42	285,86
1	4	585,51	146,38
2	1	231,84	231,84
4	1	1.393,74	1.393,74
5	5	2.179,55	435,91
6	5	1.002,13	200,43
7	4	1.325,48	331,37
9	4	1.259,06	314,77
10	3	1.022,39	340,80
11	2	1.393,74	696,87
12	3	856,93	285,64
13	1	793,78	793,78
14	2	1.230,61	615,31
15	3	1.096,68	365,56
16	3	915,67	305,22
17	4	1.309,33	327,33
18	4	1.401,89	350,47
19	4	864,85	216,21
20	3	1.236,27	412,09
21	3	159,51	53,17
22	2	930,83	465,42
23	2	190,38	95,19
24	2	672,86	336,43
26	1	622,83	622,83
27	1	1.320,36	1.320,36
28	2	2.100,22	1.050,11
29	1	2.042,39	2.042,39
30	4	930,83	232,71
1	4	744,20	186,05
2	1	244,33	244,33
4	1	1.544,23	1.544,23
5	5	2.330,04	466,01
6	5	1.152,62	230,52
7	4	1.475,97	368,99
9	4	1.409,55	352,39
10	3	862,88	287,63

<b>Continuação da Tabela 23</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
11	2	1.544,23	772,12
12	3	1.007,42	335,81
13	1	944,27	944,27
14	2	1.381,10	690,55
15	3	1.256,19	418,73
16	3	1.066,16	355,39
17	4	1.459,82	364,96
18	4	1.552,38	388,10
19	4	1.015,34	253,84
20	3	1.386,76	462,25
22	2	771,32	385,66
23	2	340,87	170,44
24	2	823,35	411,68
26	1	782,34	782,34
27	1	1.470,85	1.470,85
28	2	2.250,71	1.125,35
29	1	2.192,88	2.192,88
30	4	771,32	192,83
<hr/>			
1	4	742,61	185,65
2	1	241,92	241,92
4	1	1.541,82	1.541,82
5	5	2.327,63	465,53
6	5	1.150,21	230,04
7	4	1.473,56	368,39
9	4	1.407,14	351,79
10	3	865,29	288,43
11	2	1.541,82	770,91
12	3	1.005,01	335,00
13	1	941,86	941,86
14	2	1.378,69	689,35
15	3	1.254,60	418,20
16	3	1.063,75	354,58
17	4	1.457,41	364,35
18	4	1.549,97	387,49
19	4	1.012,93	253,23
20	3	1.384,35	461,45
22	2	773,73	386,87

<b>Continuação da Tabela 23</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
24	2	820,94	410,47
26	1	780,75	780,75
27	1	1.468,44	1.468,44
28	2	2.248,30	1.124,15
29	1	2.190,47	2.190,47
30	4	773,73	193,43
2	1	244,33	244,33
4	1	1.544,23	1.544,23
5	5	2.330,04	466,01
6	5	1.152,62	230,52
7	4	1.475,97	368,99
9	4	1.409,55	352,39
10	3	862,88	287,63
11	2	1.544,23	772,12
12	3	1.007,42	335,81
13	1	944,27	944,27
14	2	1.381,10	690,55
15	3	1.256,19	418,73
16	3	1.066,16	355,39
17	4	1.459,82	364,96
18	4	1.552,38	388,10
19	4	1.015,34	253,84
20	3	1.386,76	462,25
22	2	771,32	385,66
24	2	823,35	411,68
26	1	782,34	782,34
27	1	1.470,85	1.470,85
28	2	2.250,71	1.125,35
29	1	2.192,88	2.192,88
30	4	771,32	192,83
2	1	842,01	842,01
4	1	772,91	772,91
5	5	1.567,44	313,49
6	5	390,02	78,00
7	4	704,65	176,16
9	4	848,63	212,16

<b>Continuação da Tabela 23</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
10	3	1.634,20	544,73
11	2	983,31	491,66
12	3	1.221,75	407,25
13	1	383,35	383,35
14	2	820,18	410,09
15	3	1.531,21	510,40
16	3	303,56	101,19
17	4	688,50	172,13
18	4	781,06	195,27
19	4	378,94	94,74
20	3	825,84	275,28
22	2	1.542,64	771,32
24	2	1.037,68	518,84
26	1	1.243,66	1.243,66
27	1	909,93	909,93
28	2	1.689,79	844,90
29	1	1.556,48	1.556,48
2	1	1.010,55	1.010,55
4	1	1.151,85	1.151,85
5	5	1.927,82	385,56
7	4	1.083,59	270,90
9	4	789,37	197,34
10	3	1.802,74	600,91
11	2	777,29	388,65
12	3	1.413,25	471,08
13	1	497,37	497,37
14	2	783,44	391,72
15	3	1.700,57	566,86
16	3	663,94	221,31
17	4	1.067,44	266,86
18	4	1.160,00	290,00
19	4	604,30	151,08
20	3	939,86	313,29
22	2	1.711,18	855,59
24	2	1.229,18	614,59
26	1	1.412,20	1.412,20
27	1	873,19	873,19

Continuação da Tabela 23			
Id.	Prioridade	Distância	Custo
28	2	1.653,05	826,53
29	1	1.781,84	1.781,84
<hr/>			
2	1	1.010,55	1.010,55
4	1	1.151,85	1.151,85
5	5	1.927,82	385,56
7	4	1.083,59	270,90
9	4	789,37	197,34
10	3	1.802,74	600,91
11	2	777,29	388,65
12	3	1.413,25	471,08
13	1	497,37	497,37
14	2	783,44	391,72
15	3	1.700,57	566,86
16	3	663,94	221,31
17	4	1.067,44	266,86
18	4	1.160,00	290,00
20	3	939,86	313,29
22	2	1.711,18	855,59
24	2	1.229,18	614,59
26	1	1.412,20	1.412,20
27	1	873,19	873,19
28	2	1.653,05	826,53
29	1	1.781,84	1.781,84
<hr/>			
2	1	231,84	231,84
4	1	1.393,74	1.393,74
5	5	2.179,55	435,91
7	4	1.325,48	331,37
10	3	1.022,39	340,80
11	2	1.393,74	696,87
12	3	856,93	285,64
13	1	793,78	793,78
14	2	1.230,61	615,31
15	3	1.096,68	365,56
16	3	915,67	305,22
17	4	1.309,33	327,33
18	4	1.401,89	350,47

<b>Continuação da Tabela 23</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
20	3	1.236,27	412,09
22	2	930,83	465,42
24	2	672,86	336,43
26	1	622,83	622,83
27	1	1.320,36	1.320,36
28	2	2.100,22	1.050,11
29	1	2.042,39	2.042,39
<hr/>			
4	1	1.311,87	1.311,87
5	5	2.097,68	419,54
7	4	1.243,61	310,90
10	3	1.105,57	368,52
11	2	1.311,87	655,94
12	3	772,11	257,37
13	1	711,91	711,91
14	2	1.148,74	574,37
15	3	1.013,50	337,83
16	3	833,80	277,93
17	4	1.227,46	306,87
18	4	1.320,02	330,01
20	3	1.154,40	384,80
22	2	1.014,01	507,01
24	2	588,04	294,02
26	1	707,65	707,65
27	1	1.238,49	1.238,49
28	2	2.018,35	1.009,18
29	1	1.960,52	1.960,52
<hr/>			
4	1	1.150,52	1.150,52
5	5	1.926,49	385,30
7	4	1.082,26	270,57
10	3	1.801,41	600,47
11	2	781,70	390,85
13	1	496,04	496,04
14	2	784,57	392,29
15	3	1.699,24	566,41
16	3	662,61	220,87
17	4	1.066,11	266,53

<b>Continuação da Tabela 23</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
18	4	1.158,67	289,67
20	3	938,53	312,84
22	2	1.709,85	854,93
24	2	1.227,85	613,93
26	1	1.410,87	1.410,87
27	1	874,32	874,32
28	2	1.654,18	827,09
29	1	1.780,51	1.780,51
4	1	1.065,39	1.065,39
5	5	1.841,36	368,27
7	4	997,13	249,28
10	3	1.716,28	572,09
11	2	690,83	345,42
13	1	410,91	410,91
14	2	696,98	348,49
15	3	1.614,11	538,04
17	4	980,98	245,25
18	4	1.073,54	268,39
20	3	853,40	284,47
22	2	1.624,72	812,36
24	2	1.142,72	571,36
26	1	1.325,74	1.325,74
27	1	786,73	786,73
28	2	1.566,59	783,30
29	1	1.695,38	1.695,38
4	1	1.457,41	1.457,41
5	5	2.243,22	448,64
7	4	1.389,15	347,29
10	3	949,70	316,57
11	2	1.457,41	728,71
13	1	857,45	857,45
14	2	1.294,28	647,14
15	3	1.171,01	390,34
18	4	1.465,56	366,39
20	3	1.299,94	433,31
22	2	858,14	429,07

<b>Continuação da Tabela 23</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
24	2	736,53	368,27
26	1	697,16	697,16
27	1	1.384,03	1.384,03
28	2	2.163,89	1.081,94
29	1	2.106,06	2.106,06
4	1	992,01	992,01
5	5	1.771,26	354,25
7	4	923,75	230,94
11	2	764,21	382,11
13	1	337,53	337,53
14	2	635,08	317,54
15	3	1.540,73	513,58
18	4	1.000,16	250,04
20	3	780,02	260,01
22	2	1.551,34	775,67
24	2	1.069,34	534,67
26	1	1.252,36	1.252,36
27	1	724,83	724,83
28	2	1.504,69	752,35
29	1	1.629,38	1.629,38
4	1	1.477,56	1.477,56
5	5	2.263,37	452,67
11	2	1.477,56	738,78
13	1	877,60	877,60
14	2	1.314,43	657,22
15	3	1.180,50	393,50
18	4	1.485,71	371,43
20	3	1.320,09	440,03
22	2	848,65	424,33
24	2	756,68	378,34
26	1	706,65	706,65
27	1	1.404,18	1.404,18
28	2	2.184,04	1.092,02
29	1	2.126,21	2.126,21
4	1	1.549,97	1.549,97

<b>Continuação da Tabela 23</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
5	5	2.335,78	467,16
11	2	1.549,97	774,99
13	1	950,01	950,01
14	2	1.386,84	693,42
15	3	1.252,91	417,64
20	3	1.392,50	464,17
22	2	785,32	392,66
24	2	829,09	414,55
26	1	779,06	779,06
27	1	1.476,59	1.476,59
28	2	2.256,45	1.128,22
29	1	2.198,62	2.198,62
<hr/>			
4	1	772,91	772,91
5	5	1.567,44	313,49
11	2	983,31	491,66
13	1	383,35	383,35
14	2	820,18	410,09
15	3	1.531,21	510,40
20	3	825,84	275,28
24	2	1.037,68	518,84
26	1	1.243,66	1.243,66
27	1	909,93	909,93
28	2	1.689,79	844,90
29	1	1.556,48	1.556,48
<hr/>			
4	1	1.544,23	1.544,23
5	5	2.330,04	466,01
11	2	1.544,23	772,12
13	1	944,27	944,27
14	2	1.381,10	690,55
15	3	1.256,19	418,73
24	2	823,35	411,68
26	1	782,34	782,34
27	1	1.470,85	1.470,85
28	2	2.250,71	1.125,35
29	1	2.192,88	2.192,88
<hr/>			

<b>Continuação da Tabela 23</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
4	1	784,91	784,91
5	5	1.552,62	310,52
11	2	979,57	489,79
13	1	386,17	386,17
14	2	816,44	408,22
15	3	1.537,31	512,44
26	1	1.249,76	1.249,76
27	1	906,19	906,19
28	2	1.686,05	843,03
29	1	1.552,74	1.552,74
<hr/>			
4	1	992,01	992,01
11	2	764,21	382,11
13	1	337,53	337,53
14	2	635,08	317,54
15	3	1.540,73	513,58
26	1	1.252,36	1.252,36
27	1	724,83	724,83
28	2	1.504,69	752,35
29	1	1.629,38	1.629,38
<hr/>			
4	1	1.364,85	1.364,85
11	2	1.387,81	693,91
13	1	787,85	787,85
15	3	1.090,39	363,46
26	1	837,65	837,65
27	1	1.314,43	1.314,43
28	2	2.094,29	1.047,14
29	1	2.013,50	2.013,50
<hr/>			
4	1	1.077,06	1.077,06
11	2	713,16	356,58
13	1	422,58	422,58
26	1	1.337,41	1.337,41
27	1	805,78	805,78
28	2	1.585,64	792,82
29	1	1.710,33	1.710,33
<hr/>			

<b>Continuação da Tabela 23</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
4	1	1.358,70	1.358,70
13	1	781,70	781,70
26	1	843,80	843,80
27	1	1.308,28	1.308,28
28	2	2.088,14	1.044,07
29	1	2.007,35	2.007,35
<hr/>			
4	1	1.156,26	1.156,26
26	1	1.416,61	1.416,61
27	1	877,60	877,60
28	2	1.657,46	828,73
29	1	1.786,25	1.786,25
<hr/>			
4	1	849,70	849,70
26	1	1.185,53	1.185,53
27	1	833,14	833,14
29	1	1.479,69	1.479,69
<hr/>			
4	1	1.477,56	1.477,56
26	1	706,65	706,65
29	1	2.126,21	2.126,21
<hr/>			
4	1	770,91	770,91
29	1	1.558,48	1.558,48
<hr/>			
29	1	2.190,47	2.190,47

### B.1.5 Experimento 3

A Tabela 24 apresenta os dados de aquisição de tempo para o escalonamento do grupo inicial, após a primeira e segunda modificação.

Tabela 24 – Medida de tempo para o escalonamento.

	Inicial	1 <sup>a</sup> Modificação	2 <sup>a</sup> Modificação
Número de Tarefas	10	11	10
Medição (ms)	10659 10749 10787 10783 10720	11391 11424 11439 11508 11412	11230 11340 11275 11323 11227
Média	10739,60	11434,80	11279,00
Desvio Padrão	52,68	44,53	51,91

As etapas do escalonamento após a primeira e segunda modificação no conjunto de tarefas são apresentadas nas Tabelas 25 e 26 respectivamente. O escalonamento do grupo inicial é o mesmo presente na Tabela 17.

Tabela 25 – Escalonamento para o conjunto de tarefas após a primeira modificação.

Id.	Prioridade	Distância	Custo
1	4	501,51	125,38
2	1	315,02	315,02
4	1	1.311,87	1.311,87
5	5	2.097,68	419,54
6	5	920,26	184,05
8	5	610,69	122,14
9	4	1.177,19	294,30
10	3	1.105,57	368,52
11	2	1.311,87	655,94
12	3	772,11	257,37
13	1	711,91	711,91
<hr/>			
1	4	372,10	93,03
2	1	444,43	444,43
4	1	1.364,26	1.364,26
5	5	2.150,07	430,01
6	5	972,65	194,53
9	4	1.252,54	313,14

<b>Continuação da Tabela 25</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
10	3	1.234,98	411,66
11	2	1.387,22	693,61
12	3	781,52	260,51
13	1	787,26	787,26
<hr/>			
2	1	244,33	244,33
4	1	1.544,23	1.544,23
5	5	2.330,04	466,01
6	5	1.152,62	230,52
9	4	1.409,55	352,39
10	3	862,88	287,63
11	2	1.544,23	772,12
12	3	1.007,42	335,81
13	1	944,27	944,27
<hr/>			
2	1	1.010,55	1.010,55
4	1	1.151,85	1.151,85
5	5	1.927,82	385,56
9	4	789,37	197,34
10	3	1.802,74	600,91
11	2	777,29	388,65
12	3	1.413,25	471,08
13	1	497,37	497,37
<hr/>			
2	1	231,84	231,84
4	1	1.393,74	1.393,74
5	5	2.179,55	435,91
10	3	1.022,39	340,80
11	2	1.393,74	696,87
12	3	856,93	285,64
13	1	793,78	793,78
<hr/>			
4	1	1.311,87	1.311,87
5	5	2.097,68	419,54
10	3	1.105,57	368,52
11	2	1.311,87	655,94
12	3	772,11	257,37
13	1	711,91	711,91

<b>Continuação da Tabela 25</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
4	1	1.150,52	1.150,52
5	5	1.926,49	385,30
10	3	1.801,41	600,47
11	2	781,70	390,85
13	1	496,04	496,04
4	1	992,01	992,01
10	3	1.642,90	547,63
11	2	764,21	382,11
13	1	337,53	337,53
4	1	1.156,26	1.156,26
10	3	1.807,15	602,38
11	2	781,70	390,85
4	1	1.358,70	1.358,70
10	3	1.234,34	411,45
4	1	992,01	992,01

Tabela 26 – Escalonamento para o conjunto de tarefas após a segunda modificação.

<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
1	4	741,74	185,44
2	1	250,07	250,07
5	5	2.335,78	467,16
6	5	1.158,36	231,67
9	4	1.415,29	353,82
10	3	876,88	292,29
12	3	1.013,16	337,72
13	1	950,01	950,01
14	2	1.386,84	693,42
15	3	1.252,91	417,64
2	1	244,33	244,33

<b>Continuação da Tabela 26</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
5	5	2.330,04	466,01
6	5	1.152,62	230,52
9	4	1.409,55	352,39
10	3	862,88	287,63
12	3	1.007,42	335,81
13	1	944,27	944,27
14	2	1.381,10	690,55
15	3	1.256,19	418,73
2	1	1.010,55	1.010,55
5	5	1.927,82	385,56
9	4	789,37	197,34
10	3	1.802,74	600,91
12	3	1.413,25	471,08
13	1	497,37	497,37
14	2	783,44	391,72
15	3	1.700,57	566,86
2	1	231,84	231,84
5	5	2.179,55	435,91
10	3	1.022,39	340,80
12	3	856,93	285,64
13	1	793,78	793,78
14	2	1.230,61	615,31
15	3	1.096,68	365,56
5	5	2.097,68	419,54
10	3	1.105,57	368,52
12	3	772,11	257,37
13	1	711,91	711,91
14	2	1.148,74	574,37
15	3	1.013,50	337,83
5	5	1.926,49	385,30
10	3	1.801,41	600,47
13	1	496,04	496,04
14	2	784,57	392,29
15	3	1.699,24	566,41

<b>Continuação da Tabela 26</b>			
<b>Id.</b>	<b>Prioridade</b>	<b>Distância</b>	<b>Custo</b>
10	3	1.642,90	547,63
13	1	337,53	337,53
14	2	635,08	317,54
15	3	1.540,73	513,58
<hr/>			
10	3	1.235,57	411,86
13	1	787,85	787,85
15	3	1.090,39	363,46
<hr/>			
10	3	1.727,95	575,98
13	1	422,58	422,58
<hr/>			
10	3	1.807,15	602,38

## B.2 AVALIAÇÃO EM AMBIENTE REAL

Como os pontos de passagem foram muito semelhantes entre as repetições do experimento, real e simulado, a seguir, serão colocados um exemplo dos dados obtidos em simulação (Tabela 27) e outro dos obtidos em ambiente real (Tabela 28). Os dados numéricos representam a posição  $x; y$  em metros no ambiente, considerando o centro da imagem como  $0, 0; 0, 0$ .

Tabela 27 – Fluxo de execução das cinco tarefas em simulação.

<b>Fluxo de execução</b>
<b>Task 3</b>
<b>Sala 306</b>
-5,90286;-12,1243
-5,56;-11,7814
-4,87429;-11,7814
-4,78857;-11,8671
-4,36;-11,8671
<b>Sala 312</b>
-4,36;-11,8671
-1,27429;-11,8671

<b>Continuação da Tabela 27</b>	
<b>Fluxo de execução</b>	
-1,18857;-11,9529	
2,15429;-11,9529	
2,24;-12,0386	
2,49714;-12,0386	
2,58286;-12,1243	
15,9543;-12,1243	
16,1257;-12,2957	
<b>Task 1</b>	
<b>Sala 303</b>	
6,18286;-12,21	
6,09714;-12,1243	
2,58286;-12,1243	
2,49714;-12,0386	
2,24;-12,0386	
2,15429;-11,9529	
-1,18857;-11,9529	
-1,27429;-11,8671	
-4,78857;-11,8671	
-4,87429;-11,7814	
-8,98857;-11,7814	
-9,07429;-11,8671	
-12,76;-11,8671	
-12,8457;-11,9529	
-15,4171;-11,9529	
<b>Sala 313</b>	
-9,07429;-11,8671	
-8,98857;-11,7814	
12,7829;-11,7814	
13,3829;-11,1814	
16,1257;-11,0957	
<b>Task 2</b>	
<b>Sala 309</b>	
11,4114;-11,1814	
11,3257;-11,0957	
8,92571;-11,0957	
8,75429;-10,9243	
<b>Sala 306</b>	
6,09714;-11,01	

<b>Continuação da Tabela 27</b>
<b>Fluxo de execução</b>
6,01143;-11,0957
3,35429;-11,0957
3,26857;-11,1814
-1,53143;-11,1814
-2,21714;-11,8671
-4,36;-11,8671
<b>Task 5</b>
<b>Sala 314</b>
-4,44571;-11,7814
-5,81714;-10,41
-5,81714;-10,3243
-6,50286;-9,63857
-6,50286;12,2186
-8,38857;14,1043
-8,38857;17,7043
-8,47429;17,79
-8,47429;22,3329
-8,56;22,4186
-8,56;25,59
-8,64571;25,6757
-8,64571;25,7614
-8,73143;25,8471
-8,81714;25,8471
-8,90286;25,9329
-12,4171;25,9329
-13,1029;26,6186
-15,8457;26,7043
<b>Sala 311</b>
-15,6743;26,5329
-10,36;26,5329
-10,2743;26,4471
-9,33143;26,4471
-8,64571;25,7614
-8,64571;25,6757
-8,56;25,59
-8,56;22,4186
-8,47429;22,3329
-8,47429;17,79

<b>Continuação da Tabela 27</b>	
<b>Fluxo de execução</b>	
-8,38857;17,7043	
-8,38857;13,5043	
-6,24571;11,3614	
-6,24571;-9,98143	
-5,21714;-11,01	
15,9543;-11,01	
<b>Sala 322</b>	
-5,13143;-11,0957	
-5,81714;-10,41	
-5,81714;-10,3243	
-6,07429;-10,0671	
-6,07429;25,6757	
-5,73143;26,0186	
-2,04571;26,0186	
-1,87429;26,19	
-0,0742857;26,19	
0,0114286;26,2757	
1,64;26,2757	
1,72571;26,3614	
2,75429;26,3614	
<b>Task 4</b>	
<b>Sala 315</b>	
2,58286;26,3614	
1,72571;26,3614	
1,64;26,2757	
-8,13143;26,2757	
-8,30286;26,1043	
-12,8457;26,1043	
-13,4457;25,5043	
-15,8457;25,5043	
<b>Sala 311</b>	
-13,1029;25,59	
-13,0171;25,6757	
-12,8457;25,6757	
-12,76;25,7614	
-9,41714;25,7614	
-9,33143;25,8471	
-9,16;25,8471	

<b>Continuação da Tabela 27</b>
<b>Fluxo de execução</b>
-9,07429;25,9329
-8,90286;25,9329
-8,81714;25,8471
-8,73143;25,8471
-8,64571;25,7614
-8,64571;25,6757
-8,56;25,59
-8,56;22,4186
-8,47429;22,3329
-8,47429;17,79
-8,38857;17,7043
-8,38857;13,5043
-6,24571;11,3614
-6,24571;-9,98143
-5,21714;-11,01
15,9543;-11,01

Tabela 28 – Fluxo de execução das cinco tarefas em ambiente real.

<b>Fluxo de execução</b>
<b>Task 3</b>
<b>Sala 306</b>
-6,33143;-13,9243
-5,81714;-13,41
-5,81714;-12,3814
-5,47429;-12,0386
-4,53143;-12,0386
-4,36;-11,8671
<b>Sala 312</b>
-4,10286;-12,1243
15,9543;-12,1243
16,1257;-12,2957
<b>Task 1</b>
<b>Sala 303</b>
-1,27429;-12,21
-1,36;-12,1243
-4,87429;-12,1243

<b>Continuação da Tabela 28</b>	
<b>Fluxo de execução</b>	
-4,96;-12,0386	
-15,4171;-11,9529	
<b>Sala 313</b>	
6,18286;-11,8671	
6,52571;-11,5243	
13,04;-11,5243	
13,3829;-11,1814	
16,1257;-11,0957	
<b>Task 2</b>	
<b>Sala 309</b>	
11,5829;-11,0957	
11,4971;-11,01	
8,75429;-10,9243	
<b>Sala 306</b>	
6,18286;-11,01	
6,09714;-11,0957	
3,44;-11,0957	
3,35429;-11,1814	
-1,53143;-11,1814	
-2,21714;-11,8671	
-4,36;-11,8671	
<b>Task 5</b>	
<b>Sala 314</b>	
-4,36;-11,6957	
-4,36;-11,4386	
-5,64571;-10,1529	
-5,64571;-9,72429	
-5,81714;-9,55286	
-5,81714;12,39	
-6,41714;12,99	
-6,41714;23,1043	
-8,98857;25,6757	
-12,16;25,6757	
-13,1029;26,6186	
-15,8457;26,7043	
<b>Sala 311</b>	
-15,6743;26,6186	
-9,93143;26,6186	

<b>Continuação da Tabela 28</b>
<b>Fluxo de execução</b>
-8,81714;25,5043
-8,81714;22,3329
-8,73143;22,2471
-8,73143;17,7043
-8,64571;17,6186
-8,64571;14,3614
-5,98857;11,7043
-5,98857;-9,81
-4,78857;-11,01
15,9543;-11,01
<b>Sala 322</b>
-4,70286;-11,0957
-5,47429;-10,3243
-5,47429;-10,2386
-5,81714;-9,89571
-5,81714;25,59
-5,64571;25,7614
-1,87429;25,7614
-1,70286;25,9329
-0,331429;25,9329
0,0114286;26,2757
2,75429;26,3614
<b>Task 4</b>
<b>Sala 315</b>
2,49714;26,2757
-8,13143;26,2757
-8,30286;26,1043
-12,8457;26,1043
-13,4457;25,5043
-15,8457;25,5043
<b>Sala 311</b>
-9,07429;25,59
-8,98857;25,6757
-8,81714;25,5043
-8,81714;22,3329
-8,73143;22,2471
-8,73143;17,7043
-8,64571;17,6186

<b>Continuação da Tabela 28</b>
<b>Fluxo de execução</b>
-8,64571;14,3614
-5,98857;11,7043
-5,98857;-9,81
-4,78857;-11,01
15,9543;-11,01



## **ANEXO A - Algoritmo *D\* Lite***



A baixo o algoritmo *D Star Lite* proposto por Koenig e Likhachev (2005) em sua segunda versão.

Figura 38 – *D\* Lite*: Segunda Versão.

```

procedure CalcKey( $s$ )
{01"} return [ $\min(g(s), rhs(s)) + h(s_{start}, s) + \mathbf{k_m}; \min(g(s), rhs(s))]$ ];

procedure Initialize()
{02"}  $U = \emptyset$ ;
{03"}  $\mathbf{k_m} = \mathbf{0}$ ;
{04"} for all  $s \in S$   $rhs(s) = g(s) = \infty$ ;
{05"}  $rhs(s_{goal}) = 0$ ;
{06"}  $U.Insert(s_{goal}, CalcKey(s_{goal}))$ ;

procedure UpdateVertex( $u$ )
{07"} if ( $u \neq s_{goal}$ )  $rhs(u) = \min_{s' \in \text{Succ}(u)}(c(u, s') + g(s'))$ ;
{08"} if ( $u \in U$ )  $U.Remove(u)$ ;
{09"} if ( $g(u) \neq rhs(u)$ )  $U.Insert(u, CalcKey(u))$ ;

procedure ComputeShortestPath()
{10"} while ( $U.TopKey() < \text{CalcKey}(s_{start})$  OR  $rhs(s_{start}) \neq g(s_{start})$ )
{11"}    $\mathbf{k_{old}} = U.TopKey()$ ;
{12"}    $u = U.Pop()$ ;
{13"}   if ( $\mathbf{k_{old}} < \text{CalcKey}(u)$ )
{14"}      $U.Insert(u, \text{CalcKey}(u))$ ;
{15"}   else if ( $g(u) > rhs(u)$ )
{16"}      $g(u) = rhs(u)$ ;
{17"}     for all  $s \in \text{Pred}(u)$   $\text{UpdateVertex}(s)$ ;
{18"}   else
{19"}      $g(u) = \infty$ ;
{20"}   for all  $s \in \text{Pred}(u) \cup \{u\}$   $\text{UpdateVertex}(s)$ ;

procedure Main()
{21"}  $s_{last} = s_{start}$ ;
{22"} Initialize();
{23"} ComputeShortestPath();
{24"} while ( $s_{start} \neq s_{goal}$ )
{25"}   /* if ( $g(s_{start}) = \infty$ ) then there is no known path */
{26"}    $s_{start} = \arg \min_{s' \in \text{Succ}(s_{start})}(c(s_{start}, s') + g(s'))$ ;
{27"}   Move to  $s_{start}$ ;
{28"}   Scan graph for changed edge costs;
{29"}   if any edge costs changed
{30"}      $\mathbf{k_m} = \mathbf{k_m} + h(s_{last}, s_{start})$ ;
{31"}      $s_{last} = s_{start}$ ;
{32"}     for all directed edges  $(u, v)$  with changed edge costs
{33"}       Update the edge cost  $c(u, v)$ ;
{34"}        $\text{UpdateVertex}(u)$ ;
{35"}     ComputeShortestPath();

```

Fonte: Koenig e Likhachev (2005)