



*Trabalho de Conclusão de Curso*

# Análise comparativa de algoritmos de vigilância persistente multi-agente para a limpeza de manchas de óleo na costa Alagoana.

de Livia de Maria Calado Machado Soares.

orientado por

Prof. Dr. Glauber Rodrigues Leite.

Universidade Federal de Alagoas  
Instituto de Computação  
Maceió, Alagoas  
23 de Abril de 2025

UNIVERSIDADE FEDERAL DE ALAGOAS  
Instituto de Computação

**ANÁLISE COMPARATIVA DE ALGORITMOS DE  
VIGILÂNCIA PERSISTENTE MULTI-AGENTE PARA A  
LIMPEZA DE MANCHAS DE ÓLEO NA COSTA  
ALAGOANA.**

Trabalho de Conclusão de Curso submetido  
ao Instituto de Computação da Universidade  
Federal de Alagoas como requisito parcial  
para a obtenção do grau de Engenheira de  
Computação.

Lívia de Maria Calado Machado Soares.

***Orientador: Prof. Dr. Glauber Rodrigues Leite.***

**Banca Avaliadora:**

Ícaro Bezerra Queiroz de Araújo. Prof. Dr., IC-UFAL  
Andressa Martins Oliveira. Msc.

Maceió, Alagoas  
23 de Abril de 2025

**Catálogo na Fonte**  
**Universidade Federal de Alagoas**  
**Biblioteca Central**  
**Divisão de Tratamento Técnico**

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

S676a Soares, Livia de Maria Calado Machado.

Análise comparativa de algoritmos de vigilância persistente multi-agente para a limpeza de manchas de óleo na costa alagoana / Livia de Maria Calado Machado Soares. – 2025.

54 f. : il.

Orientador: Glauber Rodrigues Leite.

Monografia (Trabalho de conclusão de curso em Engenharia de Computação) - Universidade Federal de Alagoas, Instituto de Computação. Maceió, 2025.

Bibliografia: f. 52-54.

1. Algoritmo A\*. 2. MATLAB (Programa de computador). 3. PyGnome. 4. Monte Carlo, Método de - Análise estatística de dados. 5. Vigilância persistente (Algoritmo). I. Título.

CDU: 004.421

UNIVERSIDADE FEDERAL DE ALAGOAS  
Instituto de Computação

**ANÁLISE COMPARATIVA DE ALGORITMOS DE  
VIGILÂNCIA PERSISTENTE MULTI-AGENTE PARA A  
LIMPEZA DE MANCHAS DE ÓLEO NA COSTA  
ALAGOANA.**

Trabalho de Conclusão de Curso submetido ao Instituto de Computação da Universidade Federal de Alagoas como requisito parcial para a obtenção do grau de Engenheiro de Computação.

Aprovado em 23 de Abril de 2025:

Documento assinado digitalmente  
 **GLAUBER RODRIGUES LEITE**  
Data: 25/04/2025 14:55:45-0300  
Verifique em <https://validar.iti.gov.br>

---

Glauber Rodrigues Leite.,  
Prof. Dr., Orientador

Documento assinado digitalmente  
 **ICARO BEZERRA QUEIROZ DE ARAUJO**  
Data: 27/04/2025 16:28:16-0300  
Verifique em <https://validar.iti.gov.br>

---

Ícaro Bezerra Queiroz de Araújo.,  
Prof. Dr., IC-UFAL

Documento assinado digitalmente  
 **ANDRESSA MARTINS OLIVEIRA**  
Data: 25/04/2025 15:10:31-0300  
Verifique em <https://validar.iti.gov.br>

---

Andressa Martins Oliveira.,  
Msc.

*Dedico este trabalho ao meu avô Pedro Soares da Silva,  
apesar de não estar mais aqui, a sua história ecoa  
pelo tempo e eu o sinto cada vez mais perto.*

# Agradecimentos

Em primeiro lugar, agradeço aos meus pais, Conceição de Maria Calado e Clenio Machado da Cunha Soares, pelo alicerce construído ao longo dos anos, pelos exemplos diários de força e perseverança, e por todo o suporte essencial à minha formação pessoal e acadêmica. Estendo meus agradecimentos aos meus irmãos, Pedro Calado Machado Soares e Matheus Calado Machado Soares, pelos valores compartilhados, que levarei comigo ao longo de toda a vida, especialmente no exercício da minha trajetória profissional.

Sou grata também aos meus avós paternos, Pedro Soares da Silva e Ceci Machado da Cunha Soares, e aos meus avós maternos, José Calado da Silva e Maria Neuza Calado, pelo legado de vida que me inspira e fortalece. Sem as raízes que construíram, meu caminho não seria o mesmo.

Minha gratidão se estende ao meu tio Clecio Machado da Cunha Soares, pelo incentivo contínuo e apoio durante minha trajetória acadêmica, bem como à minha tia Rosilda Maria Calado, cuja presença afetuosa e constante foi fundamental para meu crescimento pessoal.

Agradeço a Pedro Lucas Bernardo Costa, pelo incentivo constante, apoio diário e parceria ao longo desta jornada. Sua presença tem contribuído significativamente para meu crescimento pessoal e para o fortalecimento de uma trajetória compartilhada baseada em aprendizado e evolução mútua.

Agradeço também a todos os integrantes do grupo de pesquisa AI-SPARC, ao qual tive a honra de pertencer ao longo de toda a minha graduação. A vivência nesse ambiente colaborativo foi determinante para a minha formação acadêmica, tanto pelos exemplos inspiradores quanto pelo apoio mútuo que contribuiu para o crescimento coletivo do grupo ao longo dos anos. O incentivo constante à pesquisa e ao desenvolvimento científico deixou uma marca em minha trajetória, e levo comigo o compromisso de cultivar e disseminar esses valores na nova etapa que se inicia.

Registro aqui meus agradecimentos ao meu orientador, Professor Dr. Glauber Rodrigues Leite, não apenas pelos valiosos ensinamentos transmitidos ao longo dos anos nas áreas de robótica e pesquisa, mas também por ter contribuído significativamente para o meu amadurecimento pessoal. Seu exemplo e orientação me ensinaram a buscar equilíbrio entre vida pessoal, acadêmica e profissional, uma lição que continuo a cultivar e que levarei comigo ao longo de toda a minha trajetória.

Agradeço, de forma especial, ao Professor Dr. Ícaro Bezerra Queiroz de Araújo, meu primeiro orientador no curso, cujo voto de confiança no início da graduação foi fundamental para minha inserção no universo da pesquisa. Sua orientação e as oportunidades proporcionadas ao longo da trajetória acadêmica, tanto em projetos quanto em atividades científicas, tiveram papel decisivo na construção do meu percurso. Sou profundamente grata pelos conselhos e ensinamentos compartilhados ao longo dos anos, que contribuíram significativamente para a formação da profissional e da pessoa que sou hoje.

Gostaria de estender meus agradecimentos aos professores que contribuíram de forma significativa para a minha formação ao longo da graduação. Em especial, agradeço ao Professor Dr. Leandro Dias da Silva, à Professora Me. Andressa Martins Oliveira, ao Professor Dr. Tiago Alves de Almeida, ao Professor Dr. Thiago Damasceno Cordeiro, ao Professor Me. Jobson de Araújo Nascimento, ao Professor Dr. Erick de Andrade Barboza, ao Professor Dr. João Raphael Souza Martins, ao Professor Dr. Márcio de Medeiros Ribeiro, à Professora Dr<sup>a</sup>. Maria Cristina Tenório Cavalcante Escarpini, ao Professor Dr. Rian Gabriel Santos Pinheiro, ao Professor Dr. Aydano Pamponet Machado e ao Professor Dr. Marcelo Costa Oliveira. Suas aulas, orientações e exemplos profissionais deixaram marcas em minha trajetória acadêmica.

Agradeço, com especial consideração, a todo o corpo técnico-administrativo do Instituto, incluindo a equipe da secretaria e os profissionais responsáveis pela limpeza. Em particular, expresso minha gratidão a Lúcia Maria Batista de Souza e a servidora Maria de Fátima Guerra, cuja presença, histórias e sorrisos tornaram muitos dos meus dias mais leves e alegres. Registro também meus agradecimentos aos servidores Marcelo de Gusmão, Ana Luiza Ferreira Gomes e Laelson Batista Vilela, pelo apoio prestado ao longo da graduação, especialmente no que tange às questões administrativas e burocráticas.

Por fim, estendo meus agradecimentos a todos aqueles com quem construí laços durante a graduação. Tenho a alegria de mencionar Leony Oliveira Bernardo, Rebeca de Jesus Brandão, Luana Júlia Nunes Ferreira, Ester de Lima Pontes Andrade, Maria Fernanda Herculano Machado Gitaí, Ulyanne Julia Freire Patriota, Pedro Henrique Sátiro Correia, Miguel Mazzoleni, Rangel Gonçalves, Manoel Rocha dos Santos Neto, Otto Tenório Albuquerque Lins Bisneto e Rita de Kássia Lemos Pereira, Luiza Bezerra Bastos, Camila Maria Farias Silva, Hugo Tallys Martins Oliveira, Sofia Lopes de Castro, Thiago Fellype Marques Laurentino, Manoel Pedro de Aranda Terceiro, Matheus Moreira Rodrigues, Maria Fernanda Bernardes Pinto Costa, Pedro Henrique Silva de Almeida, entre tantos outros que, com palavras, gestos, apoio acadêmico e momentos partilhados, contribuíram para que essa caminhada fosse possível. A convivência com cada um de vocês foi essencial e será sempre lembrada com gratidão.

*”Cadê a flor que tava aqui? Poluição comeu  
O peixe que é do mar? Poluição comeu  
O verde onde é que está? Poluição comeu  
Nem o Chico Mendes sobreviveu”  
Luiz Gonzaga, Xote ecológico.*

# Resumo

Este trabalho apresenta uma análise comparativa de algoritmos de vigilância persistente aplicados à simulação da limpeza de manchas de óleo na costa Alagoana, com o objetivo de avaliar o desempenho de diferentes estratégias de controle multiagente em ambientes dinâmicos. A metodologia adotada integra o MATLAB® e a biblioteca PyGNOME para construção de cinco cenários, nos quais os fatores ambientais como vento e maré são considerados na modelagem da dispersão do óleo por meio de simulação lagrangiana. Três políticas de movimentação foram implementadas: aleatória, reativa e adaptada, as últimas duas utilizando o algoritmo A\* para tomada de decisão dos robôs, com suporte a mapas de densidade gerados por estimadores de núcleo (KDE). A análise estatística dos dados, com base nos testes de Shapiro-Wilk, Friedman e Wilcoxon, permitiu uma avaliação da eficiência de cada abordagem. Os resultados obtidos analisam como cada política se comportou de acordo com o cenário avaliado e o seu desempenho foi comparado. A pesquisa contribui para o desenvolvimento de estratégias de limpeza no contexto de resposta a desastres ambientais costeiros.

***Palavras-chave:*** A\*; MATLAB® ; PyGnome; Monte Carlo; Vigilância persistente.

# Abstract

*This paper presents a comparative analysis of persistent surveillance algorithms applied to the simulation of oil slick cleanup off the coast of Alagoas, with the aim of evaluating the performance of different multi-agent control strategies in dynamic environments. The methodology adopted integrates MATLAB® and the PyGNOME library to build five scenarios, in which environmental factors such as wind and tide are considered in the modeling of oil dispersion through Lagrangian simulation. Three movement policies were implemented: random, reactive and adapted, the latter two using the A\* algorithm for robot decision-making, with support for density maps generated by kernel estimators (KDE). The statistical analysis of the data, based on the Shapiro-Wilk, Friedman and Wilcoxon tests, allowed an evaluation of the efficiency of each approach. The results obtained analyze how each policy behaved according to the scenario evaluated and their performance was compared. The research contributes to the development of clean-up strategies in the context of response to coastal environmental disasters.*

**Keywords:** *A\* ; MATLAB® ; PyGnome; Monte Carlo; ; Persistent surveillance.*

# Lista de Figuras

|   |                                                                                                                                                                                                                                                                                                                                                                                                                      |    |
|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1 | Representação em 3D da função KDE gerada referente à concentração de óleo no mapa da simulação, o azul possui menos concentração e o amarelo maior concentração de petróleo. . . . .                                                                                                                                                                                                                                 | 23 |
| 2 | Ambiente de simulação com o cálculo probabilístico fornecido pelo KDE. A barra de cores ao lado refere-se à concentração de petróleo, azul escuro é referente à nenhuma concentração e vermelho a maior concentração(5). .                                                                                                                                                                                           | 24 |
| 3 | Diagrama referente ao ambiente de simulação, demonstrando como o <i>escript</i> principal “ <i>Manager</i> ” atua controlando a simulação. . . . .                                                                                                                                                                                                                                                                   | 33 |
| 4 | Distribuição dos robôs nos diferentes cenários de teste: (a)Cenário 1 - robôs no centro; (b)Cenário 2 - robôs embaixo à esquerda; (c)Cenário 3 - robôs em cima à esquerda; (d)Cenário 4 - robôs embaixo à direita; (e)Cenário 5 - robôs em cima à direita. A barra de cores ao lado refere-se à concentração de petróleo, azul escuro é referente à nenhuma concentração e vermelho a maior concentração(5). . . . . | 39 |
| 5 | Diagrama que representa o fluxo seguido pela simulação para coleta das amostras. . . . .                                                                                                                                                                                                                                                                                                                             | 41 |
| 6 | Diagrama que demonstra o processo da avaliação dos resultados. . . . .                                                                                                                                                                                                                                                                                                                                               | 42 |
| 7 | Resultado da simulação do cenário com início no centro. A imagem (a) é referente ao algoritmo aleatório, a (b) é referente ao algoritmo reativo e a (c) é referente ao algoritmo adaptado. A barra de cores ao lado refere-se à concentração de petróleo, azul escuro é referente à nenhuma concentração e vermelho a maior concentração(5). . . . .                                                                 | 43 |
| 8 | Resultado da simulação do cenário com início em cima e na direita. A imagem (a) é referente ao algoritmo aleatório, a (b) é referente ao algoritmo reativo e a (c) é referente ao algoritmo adaptado. A barra de cores ao lado refere-se à concentração de petróleo, azul escuro é referente à nenhuma concentração e vermelho a maior concentração(5). . . . .                                                      | 45 |

|    |                                                                                                                                                                                                                                                                                                                                                                  |    |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 9  | Resultado da simulação do cenário com início embaixo e na direita. A imagem (a) é referente ao algoritmo aleatório, a (b) é referente ao algoritmo reativo e a (c) é referente ao algoritmo adaptado. A barra de cores ao lado refere-se à concentração de petróleo, azul escuro é referente à nenhuma concentração e vermelho a maior concentração(5). . . . .  | 47 |
| 10 | Resultado da simulação do cenário com início em cima e na esquerda. A imagem (a) é referente ao algoritmo aleatório, a (b) é referente ao algoritmo reativo e a (c) é referente ao algoritmo adaptado. A barra de cores ao lado refere-se à concentração de petróleo, azul escuro é referente à nenhuma concentração e vermelho a maior concentração(5). . . . . | 48 |
| 11 | Resultado da simulação do cenário com início embaixo e na esquerda. A imagem (a) é referente ao algoritmo aleatório, a (b) é referente ao algoritmo reativo e a (c) é referente ao algoritmo adaptado. A barra de cores ao lado refere-se à concentração de petróleo, azul escuro é referente à nenhuma concentração e vermelho a maior concentração(5). . . . . | 49 |

# Lista de Tabelas

|    |                                                    |    |
|----|----------------------------------------------------|----|
| 1  | Pesos usados para a política reativa . . . . .     | 36 |
| 2  | Pesos usados para a política adaptada . . . . .    | 37 |
| 3  | Resultado inicial da simulação 01. . . . .         | 44 |
| 4  | Resultado final da simulação 01. . . . .           | 44 |
| 5  | Resultado inicial da simulação 02. . . . .         | 45 |
| 6  | Resultado final da simulação 02. . . . .           | 46 |
| 7  | Resultado inicial da simulação 03. . . . .         | 46 |
| 8  | Resultado final da simulação 03. . . . .           | 46 |
| 9  | Resultado inicial da simulação 04. . . . .         | 47 |
| 10 | Resultado inicial da simulação 05. . . . .         | 50 |
| 11 | Resultado final da simulação 05. . . . .           | 50 |
| 12 | Tabela com os resultados finais. . . . .           | 50 |
| 13 | Tabela com comparação final de resultados. . . . . | 51 |

# Lista de Símbolos

|               |                                                                                |
|---------------|--------------------------------------------------------------------------------|
| $\sum$        | Somatório.                                                                     |
| $\vec{V}$     | Vetor velocidade do fluido.                                                    |
| $C$           | Concentração de uma substância em um ponto do fluido.                          |
| $D$           | Coefficiente de difusão molecular.                                             |
| $\nabla C$    | Gradiente da concentração.                                                     |
| $\nabla^2 C$  | Laplaciano da concentração (difusão espacial).                                 |
| $f(n)$        | Função de avaliação do algoritmo A*, combinação de custo real e heurístico.    |
| $g(n)$        | Custo acumulado do estado inicial até o nó $n$ .                               |
| $h(n)$        | Estimativa heurística do custo do nó $n$ até o objetivo.                       |
| $v_{ij}$      | Valor resultante da política reativa/adaptada.                                 |
| $s_j$         | Valor de sensibilidade na política reativa.                                    |
| $c_i$         | Valor de concentração na política reativa/adaptada.                            |
| $w_{ci}$      | Peso atribuído a concentração na política reativa.                             |
| $w_{si}$      | Peso atribuído a sensibilidade na política reativa.                            |
| $w_{di}$      | Peso atribuído a distância na política reativa/adaptada.                       |
| $w_{ni}$      | Peso atribuído a vizinhança na política reativa/adaptada.                      |
| $\kappa_i$    | Ganho da equação na política reativa.                                          |
| $v_{ij}$      | Valor atribuído à célula $j$ pelo robô $i$ , segundo a política de vigilância. |
| $\delta_{ij}$ | Distância entre robô $i$ e célula $j$ .                                        |
| $\delta_{kj}$ | Distância entre robô vizinho $k$ e célula $j$ .                                |

$L_f$  Métrica acumulada de sujeira final no ambiente simulado.

$n$  Número de elementos ou simulações consideradas.

# Lista de Abreviaturas

**IC** Instituto de Computação.

**UFAL** Universidade Federal de Alagoas.

**NOAA** *National Oceanic and Atmospheric Administration.*

**GNOME** *General NOAA Operational Modeling Environment.*

**ORR** *Office of Response and Restoration.*

**KDE** *Kernel Density Estimator.*

**pdf** *Probability density function.*

**MATLAB®** *MATrix LABoratory.*

**UAV** *Unmanned Aerial Vehicle.*

**CSV** *Comma-Separated Values.*

# Sumário

|       |                                                             |    |
|-------|-------------------------------------------------------------|----|
| 1     | Introdução . . . . .                                        | 18 |
| 1.1   | Objetivos . . . . .                                         | 19 |
| 1.1.1 | Objetivo geral . . . . .                                    | 19 |
| 1.1.2 | Objetivos específicos . . . . .                             | 19 |
| 1.2   | Organização do texto . . . . .                              | 19 |
| 2     | Fundamentação Teórica . . . . .                             | 20 |
| 2.1   | Ambiente para a Simulação Dinâmica Usando PyGnome . . . . . | 20 |
| 2.1.1 | Simulação Lagrangiana . . . . .                             | 20 |
| 2.1.2 | Variáveis Climáticas de Interesse . . . . .                 | 21 |
| 2.1.3 | Tipos de derramamentos . . . . .                            | 21 |
| 2.2   | Estimador de Densidade Kernel . . . . .                     | 22 |
| 2.3   | Algoritmo A* . . . . .                                      | 24 |
| 2.4   | Método de Monte Carlo . . . . .                             | 25 |
| 2.5   | Técnicas Estatísticas . . . . .                             | 26 |
| 2.5.1 | Valor- $p$ . . . . .                                        | 26 |
| 2.5.2 | Teste de Normalidade (Shapiro-Wilk) . . . . .               | 27 |
| 2.5.3 | Teste de Friedman . . . . .                                 | 28 |
| 2.5.4 | Teste de Wilcoxon . . . . .                                 | 29 |
| 3     | Trabalhos relacionados . . . . .                            | 31 |
| 4     | Metodologia . . . . .                                       | 33 |
| 4.1   | Arquitetura de simulação . . . . .                          | 33 |
| 4.2   | Algoritmos de Vigilância . . . . .                          | 34 |
| 4.2.1 | Política aleatória . . . . .                                | 34 |
| 4.2.2 | Política reativa . . . . .                                  | 35 |
| 4.2.3 | Política adaptada . . . . .                                 | 37 |
| 4.3   | Proposta de métrica . . . . .                               | 37 |
| 4.4   | Cenários de simulação . . . . .                             | 38 |
| 4.4.1 | Implementação do método Monte Carlo . . . . .               | 40 |
| 4.4.2 | Análise dos dados colhidos . . . . .                        | 40 |
| 5     | Resultados e discussões . . . . .                           | 43 |
| 5.1   | Cenário 01 . . . . .                                        | 43 |

|                     |                          |           |
|---------------------|--------------------------|-----------|
| 5.2                 | Cenário 02 . . . . .     | 44        |
| 5.3                 | Cenário 03 . . . . .     | 46        |
| 5.4                 | Cenário 04 . . . . .     | 47        |
| 5.5                 | Cenário 05 . . . . .     | 49        |
| 5.6                 | Melhor cenário . . . . . | 50        |
| 6                   | Conclusões . . . . .     | 52        |
| <b>Bibliografia</b> |                          | <b>53</b> |

# 1 Introdução

A robótica é um campo interdisciplinar que integra conhecimentos da engenharia, ciência da computação, matemática e, mais recentemente, da inteligência artificial. Seu objetivo central é desenvolver sistemas capazes de perceber o ambiente, tomar decisões e realizar ações de forma autônoma ou semi-autônoma (Siciliano e Khatib, 2016). Ao longo das últimas décadas, a robótica tem desempenhado um papel cada vez mais relevante em setores como a indústria, saúde, agricultura e serviços, contribuindo para ganhos em eficiência, precisão e segurança.

Dentre os avanços recentes da área, destaca-se a robótica com múltiplos agentes, ou robótica multiagente. Essa abordagem envolve a coordenação de diversos robôs que interagem entre si para cumprir objetivos comuns ou complementares. Sistemas multiagentes têm-se mostrado eficazes em tarefas que exigem cobertura espacial, escalabilidade e tolerância a falhas, sendo aplicados em áreas como logística (Wurman et al., 2008), agricultura de precisão (Roldán et al., 2016), mapeamento ambiental (McCammon et al., 2021), operações de busca e resgate (Hauert et al., 2011) e vigilância distribuída (Rekleitis et al., 2004).

Dentro deste escopo, a robótica de vigilância é um ramo específico que se dedica ao monitoramento contínuo de áreas de interesse por meio de agentes autônomos ou semiautônomos. Segundo Elmaliach et al. (2009), a vigilância persistente busca maximizar a cobertura espacial e temporal de um ambiente, reduzindo zonas cegas e otimizando a rota dos agentes. Essa especialização tem ganhado destaque principalmente em contextos de segurança, controle de fronteiras, monitoramento ambiental e resposta a desastres.

A complexidade da vigilância se intensifica quando aplicada a ambientes dinâmicos, cenários que se transformam ao longo do tempo, exigindo que os agentes reajam e se adaptem em tempo real (Russell e Norvig, 2013). Um exemplo relevante dessa dinâmica é o derramamento de óleo na costa brasileira, especialmente o ocorrido em 2019, que afetou mais de 1.000 praias em 11 estados, incluindo todo o litoral de Alagoas. Esse tipo de desastre ambiental traz consequências severas para a fauna, flora, economia e saúde pública, além de representar um desafio logístico considerável para as equipes de contenção e limpeza.

Nesse contexto, a robótica surge como uma ferramenta para mitigação dos danos gerados por estes desastres ambientais. Por meio da simulação e implementação de estratégias de vigilância com robôs autônomos, é possível identificar, monitorar e atuar sobre áreas contaminadas de maneira mais rápida e segura. O uso de algoritmos inteligentes permite que esses sistemas tomem decisões com base em informações em tempo real, mesmo em ambientes sujeitos a alterações constantes, como o deslocamento de manchas de óleo em função das marés e ventos.

Este trabalho propõe a simulação de um cenário dinâmico de derramamento de óleo na costa brasileira, com foco na região de Alagoas. Serão analisados e comparados algoritmos de vigilância persistente com múltiplos agentes, visando identificar estratégias que otimizem a cobertura e o tempo de resposta na limpeza dessas manchas. A investigação busca contribuir para o desenvolvimento de soluções tecnológicas mais eficazes no enfrentamento de desastres ambientais.

## **1.1 Objetivos**

### **1.1.1 Objetivo geral**

O objetivo geral deste trabalho consiste em realizar a comparação entre as técnicas aqui apresentadas de patrulhamento em ambientes dinâmicos, a partir da métrica proposta, a fim de chegar a uma conclusão sobre qual é a melhor técnica para cada cenário apresentado e a técnica que apresentou o melhor desempenho geral.

### **1.1.2 Objetivos específicos**

- Introduzir conceitos necessários para o entendimento do trabalho;
- Trazer uma noção sobre trabalhos relacionados;
- Determinar os cenários de simulação e o método de geração de dados;
- Utilizar métricas estatísticas para obtenção de conhecimento sobre os dados;
- Analisar os resultados obtidos em cada cenário;
- Estabelecer a técnica com o melhor desempenho obtido;
- Propor os passos futuros do trabalho.

## **1.2 Organização do texto**

Com o intuito de facilitar a leitura, este trabalho está dividido em capítulos que abordam o conteúdo de maneira gradativa e lógica, iniciando com a Fundamentação teórica na seção 2, que trará uma base teórica para o entendimento do trabalho. Em seguida, uma base com trabalhos relacionados é estabelecida na seção 3, seguindo para a explicação da metodologia na seção 4. Por fim, na seção 5 serão trazidos os resultados obtidos em cada cenário e este trabalho será finalizado com uma conclusão clara e objetiva, na seção 6, que trará as possibilidades de trabalhos futuros.

## 2 Fundamentação Teórica

A presente seção traz conceitos importantes para o entendimento do trabalho, iniciando com a explicação sobre conceitos envolvendo o ambiente de simulação, nas subseções 2.1, 2.2, 2.3 e 2.4 por fim, as técnicas estatísticas usadas para analisar os resultados deste trabalho na subseção 2.5.

### 2.1 Ambiente para a Simulação Dinâmica Usando PyGnome

O ambiente para a simulação dinâmica utiliza a ferramenta MATrix LABoratory (MATLAB®), dada a sua eficácia em cálculos numéricos e visualizações. Integrado a ele, a linguagem Python tem permitido a ampliação das funcionalidades dos modelos, facilitando a customização e a integração com outras bibliotecas. Nesse contexto, o *General NOAA Operational Modeling Environment* (GNOME) se destaca como uma ferramenta para a simulação de derramamentos de petróleo e de outros materiais perigosos na costa marítima.

O GNOME<sup>1</sup> foi desenvolvido pela *National Oceanic and Atmospheric Administration* (NOAA), por meio do *Office of Response and Restoration* (ORR) e da *Emergency Response Division*, com o objetivo principal de fornecer um sistema de rastreamento de partículas capaz de representar a dispersão de substâncias no ambiente oceânico. Embora originalmente concebida para simulações específicas de derramamentos de óleo, apresenta uma arquitetura flexível que permite sua aplicação em diversos contextos de transporte oceanográfico, como o rastreamento de larvas de peixe e a dispersão de detritos marinhos.

Para facilitar a integração do GNOME com o ecossistema Python, foi desenvolvida a biblioteca PyGNOME. Essa biblioteca encapsula as funcionalidades do GNOME, possibilitando a execução e o controle das simulações de forma mais prática. Além disso, se destaca pela documentação completa, que inclui descrições detalhadas das funções e parâmetros disponíveis, permitindo a configuração de ambientes de simulação semelhantes ao real.

Dentro da abordagem de simulação dinâmica empregada, destacam-se três aspectos fundamentais: a simulação lagrangiana, a incorporação de variáveis climáticas de interesse e a modelagem dos diferentes tipos de *spills*.

#### 2.1.1 Simulação Lagrangiana

A simulação lagrangiana consiste na modelagem do movimento de partículas individuais, onde cada partícula é acompanhada ao longo do tempo. Essa abordagem permite a aplicação das equações de Navier–Stokes (Cengel e Cimbala, 2015) de forma discretizada, aplicando a dinâmica dos fluidos para determinar seu deslocamento. Um dos pilares

---

<sup>1</sup>Disponível em: <https://response.restoration.noaa.gov/gnome/> (Acesso em: 04/02/2025)

dessa simulação é a equação de advecção-difusão, que descreve a evolução temporal da concentração  $C$  de uma substância no meio fluido:

$$\frac{\partial C}{\partial t} + \vec{V} \cdot \nabla C = D \nabla^2 C, \quad (1)$$

onde  $\frac{\partial C}{\partial t}$  representa a variação temporal da concentração,  $\vec{V} \cdot \nabla C$  é o termo de advecção, que modela o transporte da substância pelo escoamento do fluido, e  $D \nabla^2 C$  é o termo de difusão, responsável pela dispersão da substância devido a efeitos moleculares. Essa equação é fundamental na mecânica dos fluidos, pois descreve como uma quantidade escalar se propaga e se dispersa em um fluido em movimento. O seu entendimento é necessário para a modelagem realista do transporte e dispersão de partículas em simulações lagrangianas.

### 2.1.2 Variáveis Climáticas de Interesse

No contexto das simulações ambientais, a incorporação de variáveis climáticas está ligada à obtenção de resultados que reflitam a dinâmica real do ambiente. Entre as variáveis relevantes estão o vento e as marés. O vento influencia diretamente a advecção das partículas, determinando a direção e a velocidade do seu deslocamento. Já as marés podem afetar tanto os processos de advecção quanto de difusão, modulando os padrões de dispersão das partículas.

Essas variáveis são tipicamente obtidas a partir de dados meteorológicos e oceanográficos, coletados por estações de monitoramento e bóias. A integração desses dados ao modelo permite que a simulação reproduza, de maneira mais fiel, as condições reais do ambiente, essencial para a validação dos resultados e para a aplicação prática dos modelos em cenários de resposta a emergências.

Os dados de vento e maré são dos dias 15 a 28 de setembro de 2020, época na qual o derramamento de óleo na costa do nordeste já havia acontecido e estava espalhando <sup>2</sup>. O atual trabalho foca na simulação deste desastre ambiental.

### 2.1.3 Tipos de derramamentos

O PyGNOME oferece a capacidade de simular diversos tipos de derramamentos (ou *spills*), cada um caracterizado por propriedades físicas e químicas específicas, tais como viscosidade, densidade e taxa de evaporação. Essas características influenciam diretamente os processos de advecção e difusão, determinando a forma como a substância interage com o meio ambiente. A escolha do tipo de *spill* a ser modelado deve levar em consideração as propriedades reais da substância em estudo, permitindo uma análise dos impactos ambientais e das estratégias de mitigação.

---

<sup>2</sup>Site G1 - Manchas de óleo no Nordeste. (Acesso em 09/02/2025)

Por exemplo, derramamentos de óleo pesado podem apresentar uma difusão mais lenta em comparação a derramamentos de óleo leve, devido às diferenças na viscosidade e densidade. Assim, a flexibilidade do PyGNOME em configurar diferentes tipos de spills possibilita a realização de simulações customizadas, que atendem a uma ampla gama de cenários e necessidades operacionais.

## 2.2 Estimador de Densidade Kernel

O Estimador de Densidade Kernel, do inglês *Kernel Density Estimator* (KDE) é um método não paramétrico que estima a função densidade de probabilidade, do inglês *probability density function* (pdf), de uma variável aleatória a partir de uma amostra de dados. A partir de um conjunto de dados observados, o KDE aproxima a densidade verdadeira distribuindo uma “função kernel” suave (como uma função Gaussiana), em torno de cada ponto dos dados e somando as contribuições dessas funções (Bishop e Nasrabadi, 2006).

O KDE é definido por uma função de suavização e um intervalo, que controla a suavidade da densidade resultante da curva. O Estimador de Densidade Kernel é a função de probabilidade estimada a partir de uma variável, como exposta na Equação 2:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \quad (2)$$

No qual  $x_i$  são amostras randômicas vindas de uma distribuição desconhecida,  $n$  é o tamanho da amostra,  $K$  é a função Kernel e  $h$  é a largura da amostra.

A função de suavização do Kernel define o formato da curva usada para gerar a pdf. O KDE pode ser comparado a um histograma, pois esse comporta a função de distribuição probabilística usando os dados amostrados. Porém, se diferencia de um histograma, pois, ao contrário desse, que aloca os valores em compartimentos discretos, o KDE soma as funções de suavização dos componentes para cada valor e gera uma curva de probabilidade contínua e suave.

Neste trabalho o KDE é usado para estimar e suavizar a posição do óleo espalhado no oceano. Por se tratar de uma simulação em uma região grande do oceano, as partículas de óleo estarão dispersas respeitando as leis de difusão e advecção. O cenário é dividido em quadrados e cada quadrado receberá um valor de densidade das partículas de óleo na determinada região. Tal cálculo é necessário para que cada robô, ao calcular para onde vai, possa receber um valor de peso exato e coloque isso em consideração. Caso contrário, seriam diversas partículas espalhadas no mapa, sem a densidade calculada, logo sem a informação de tomada de decisão para onde o robô deve ir.

A função é implementada pelo MATLAB®, utilizando a função nativa *ksdensity*. No

código ela é exposta da seguinte maneira, na Equação 3:

$$[f, \ ] = ksdensity(data, positions, 'Bandwidth', 0.02) \quad (3)$$

No qual *data* representa uma lista com as posições de latitude e longitude, limitando o cálculo da área geográfica, *positions* indica as posições das partículas de óleo dispersas no ambiente limitado, o '*Bandwidth*' aciona a modificação deste parâmetro na aplicação e o 0.02 é o tamanho da largura do '*Bandwidth*'. Esse parâmetro controla a suavidade da curva probabilística resultante.

Por fim, é possível visualizar a curva gerada a partir dos dados de óleo presentes no ambiente, na Figura 1. A partir dessa curva, o cenário de simulação, representado na Figura 2 é montado.

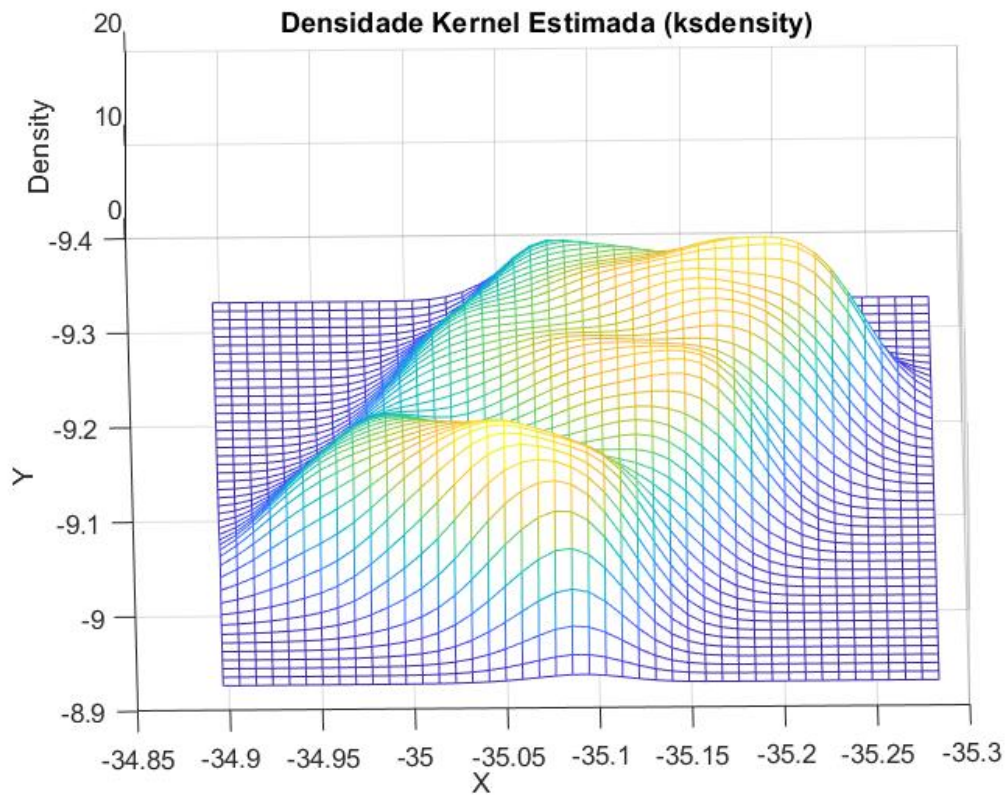


Figura 1: Representação em 3D da função KDE gerada referente à concentração de óleo no mapa da simulação, o azul possui menos concentração e o amarelo maior concentração de petróleo.

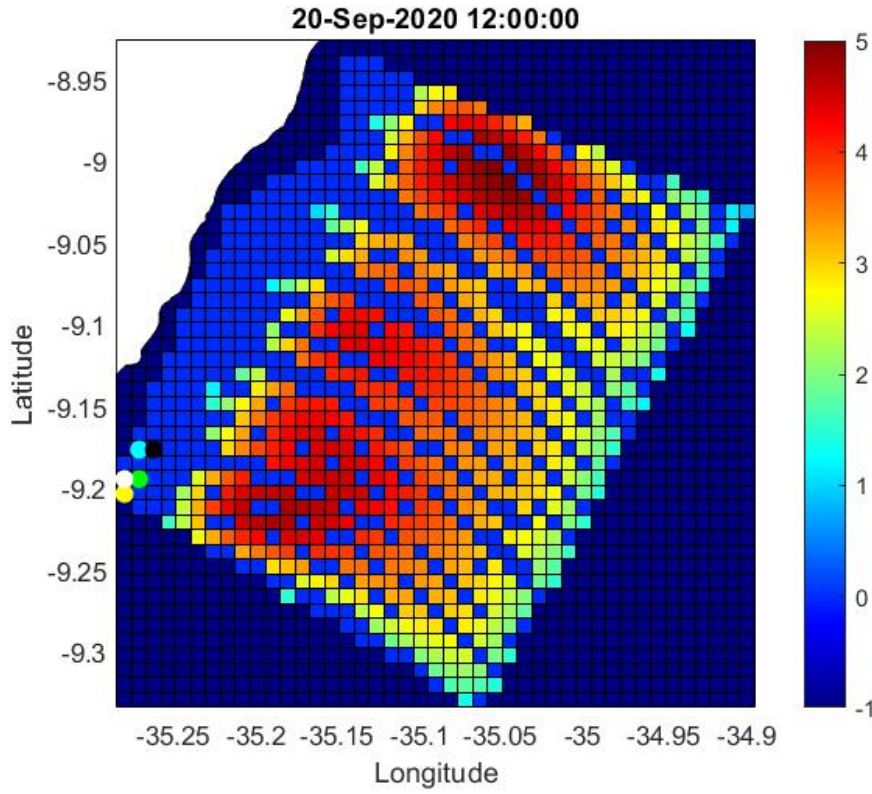


Figura 2: Ambiente de simulação com o cálculo probabilístico fornecido pelo KDE. A barra de cores ao lado refere-se à concentração de petróleo, azul escuro é referente à nenhuma concentração e vermelho a maior concentração(5).

### 2.3 Algoritmo A\*

O algoritmo A-estrela (A\*) é um dos métodos utilizados para busca de caminhos em grafos e planejamento de trajetórias, sendo amplamente aplicado em inteligência artificial e teoria da computação (Russell e Norvig, 2013). Ele pertence à classe dos algoritmos de busca heurística, combinando o custo real acumulado de um caminho com uma estimativa heurística do custo restante até o objetivo.

O algoritmo mantém uma lista de nós abertos, que representa os estados a serem explorados, e uma lista de nós fechados, que contém os estados já analisados. A cada iteração, o nó com o menor custo total estimado é selecionado para expansão, utilizando a função de avaliação, apresentada na Equação 4:

$$f(n) = g(n) + h(n), \quad (4)$$

onde  $f(n)$  representa o custo estimado total de um caminho que passa pelo nó  $n$ ,  $g(n)$  corresponde ao custo acumulado desde o estado inicial até  $n$  e  $h(n)$  é a estimativa heurística do custo restante até o objetivo. A escolha da heurística  $h(n)$  influencia diretamente o

desempenho do algoritmo, podendo torná-lo completo e ótimo Russell e Norvig (2013). O algoritmo usado nesse trabalho está disponível no *GitHub*<sup>3</sup>, se baseando no pseudocódigo que é exposto a seguir:

---

```

1: procedure  $F(\text{node})$ 
2:   return  $\text{node.pathCost} + \text{node.heuristic}$ 
3: end procedure
4: procedure  $\text{ASTARSEARCH}(\text{problem})$ 
5:    $\text{node} \leftarrow \text{Node}(\text{State} = \text{problem.initial}, \text{pathCost} = 0)$ 
6:    $\text{frontier} \leftarrow$  priority queue of “nodes” ordered by  $F(\text{node})$ 
7:    $\text{reached} \leftarrow$  empty set
8:   Insert  $\text{node}$  into  $\text{frontier}$ 
9:   Add  $\text{node.State}$  to  $\text{reached}$  with cost 0
10:  while  $\text{frontier}$  is not empty do
11:     $\text{node} \leftarrow \text{pop}(\text{frontier})$   $\triangleright$  Retrieves the node with minimal  $f$  and removes
    from  $\text{frontier}$ 
12:    if  $\text{problem.isGoal}(\text{node.State})$  then
13:      return  $\text{node}$ 
14:    end if
15:    for each  $\text{action}$  in  $\text{problem.Actions}(\text{node.State})$  do
16:       $\text{child} \leftarrow \text{ChildNode}(\text{problem}, \text{node}, \text{action})$ 
17:      if  $\text{child.State}$  is not in  $\text{reached}$  or  $\text{child.pathCost} < \text{reached}[\text{child.State}]$ 
    then
18:         $\text{reached}[\text{child.State}] \leftarrow \text{child.pathCost}$ 
19:        Insert  $\text{child}$  into  $\text{frontier}$ 
20:      end if
21:    end for
22:  end while
23:  return failure
24: end procedure

```

---

No contexto da simulação realizada neste trabalho, o algoritmo  $A^*$  é empregado para definir a trajetória dos robôs no ambiente simulado. Cada robô, conhecendo previamente o mapa, utiliza o  $A^*$  para planejar seu deslocamento até a próxima região a ser limpa. Esse processo busca minimizar o custo total de limpeza da área contaminada, otimizando a cobertura da operação (LaValle, 2006).

## 2.4 Método de Monte Carlo

O método de Monte Carlo é uma técnica estatística utilizada para resolver problemas por meio da geração de amostras aleatórias e da análise estatística dos resultados (Metropolis e Ulam, 1949; Kroese et al., 2014). Ele é útil em simulações onde modelos determinísticos são inviáveis ou apresentam alta complexidade computacional, permitindo

---

<sup>3</sup>Disponível em <https://github.com/EinarUeland/Astar-Algorithm> (Acesso em 13 de março de 2025)

a obtenção de estimativas estatísticas por meio da repetição de experimentos estocásticos.

Neste trabalho, o método de Monte Carlo é empregado para simular múltiplas instâncias de um cenário de dispersão de óleo na costa marítima do Nordeste, variando parâmetros como o instante inicial da simulação e a posição dos robôs. Essa abordagem possibilita uma avaliação probabilística das regiões de maior concentração de óleo, contribuindo para a análise dos algoritmos que definem as trajetórias dos robôs responsáveis pela limpeza.

A aplicação dessa técnica permite a geração de diversos cenários a partir da modificação de variáveis de entrada, fornecendo uma visão abrangente das possíveis dinâmicas de dispersão do óleo. O método de Monte Carlo tem sido amplamente utilizado em modelagens ambientais e na otimização de estratégias operacionais em sistemas dinâmicos (Rubinstein e Kroese, 2016; Liu e Li, 2020), devido à sua flexibilidade e capacidade de incorporar incertezas inerentes aos fenômenos estudados.

## 2.5 Técnicas Estatísticas

A análise estatística dos resultados obtidos com o método de Monte Carlo é essencial para garantir comparações adequadas entre os diferentes cenários simulados. Para isso, são utilizadas medidas de tendência central e testes estatísticos que verificam a normalidade dos dados, auxiliando na escolha de métodos apropriados para a comparação.

A média é uma medida de tendência central definida de acordo com a Equação 5:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad (5)$$

onde  $x_i$  representa cada observação da amostra e  $n$  é o número total de observações. A média é uma métrica adequada para comparação quando os dados seguem uma distribuição normal.

A mediana representa o valor central de um conjunto de dados ordenado. Para um número ímpar de elementos, corresponde ao valor central; para um número par, é a média dos dois valores centrais. A mediana é mais robusta a valores extremos e é preferível quando os dados não seguem uma distribuição normal (Wilcox, 2012).

### 2.5.1 Valor- $p$

O valor- $p$  ( $p$ -value) é um conceito da estatística utilizado para testar hipóteses e avaliar a significância de um resultado. Ele representa a probabilidade de obter um resultado tão ou mais extremo do que o observado nos dados, assumindo que a hipótese nula ( $H_0$ ) seja verdadeira (Wasserman, 2004).

Em testes estatísticos, a hipótese nula representa a ausência de efeito ou diferença entre grupos. O valor- $p$  é comparado a um nível de significância predefinido, geralmente

$\alpha = 0.05$  (Wasserman, 2004). Se o valor- $p$  for menor que  $\alpha$ , rejeita-se a hipótese nula, sugerindo que há evidências estatisticamente significativas contra ela. Caso contrário, não se pode rejeitá-la, indicando que os dados não fornecem suporte suficiente para uma diferença significativa (Casella e Berger, 2001).

O valor- $p$  é utilizado em testes estatísticos, como no teste de Shapiro-Wilk. Esse teste usa o valor- $p$  para avaliar se uma amostra segue uma distribuição normal. Se o valor- $p$  for menor que o nível de significância, rejeita-se a normalidade dos dados. O teste de Friedman, por sua vez, é um teste não paramétrico para comparar três ou mais grupos relacionados, e sua decisão também se baseia no valor- $p$ . De maneira similar, o teste de Wilcoxon é utilizado para comparar duas amostras dependentes, sendo uma alternativa ao teste  $t$  para dados não normalmente distribuídos. Em todos esses casos, o valor- $p$  orienta a decisão de aceitar ou rejeitar a hipótese nula, fornecendo uma medida da força da evidência contra ela.

### 2.5.2 Teste de Normalidade (Shapiro-Wilk)

O teste de Shapiro-Wilk (Shapiro e Wilk, 1965) verifica se um conjunto de dados segue uma distribuição normal. Ele calcula uma estatística baseada na correlação entre os valores ordenados da amostra e os valores esperados para uma distribuição normal. O valor- $p$  resultante do teste é interpretado da seguinte forma:

- Se  $p > 0.05$ , não há evidências estatísticas para rejeitar a hipótese nula, indicando que os dados podem ser considerados normalmente distribuídos.
- Se  $p \leq 0.05$ , a hipótese nula é rejeitada, sugerindo que os dados não seguem uma distribuição normal.

A Equação 6 expõe a forma que o teste é calculado:

$$W = \frac{\left(\sum_{i=1}^n a_i x_{(i)}\right)^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (6)$$

onde  $x_{(i)}$  representa os valores da amostra ordenados em ordem crescente,  $\bar{x}$  é a média da amostra, e  $a_i$  são coeficientes pré-determinados baseados nos quantis da distribuição normal. O numerador da equação corresponde ao quadrado de uma soma ponderada dos valores ordenados, enquanto o denominador representa a variabilidade total dos dados em torno da média amostral.

Quanto mais próximo de 1 for o valor de  $W$ , mais próxima a distribuição dos dados está de uma normal. O valor- $p$  associado ao teste indica a probabilidade de se obter um valor de  $W$  tão extremo ou menor, assumindo que os dados realmente sigam uma distribuição normal. Assim, um valor- $p$  pequeno sugere que a hipótese de normalidade deve ser rejeitada.

A aplicação do teste de Shapiro-Wilk no contexto deste trabalho é utilizado para determinar a escolha das técnicas estatísticas a serem utilizadas. Se os dados forem normalmente distribuídos, métodos paramétricos, como a comparação de médias, são apropriados. Caso contrário, testes não paramétricos devem ser empregados para garantir uma avaliação robusta das diferenças entre os cenários simulados.

### 2.5.3 Teste de Friedman

O teste de Friedman (Friedman, 1937) é um teste estatístico não paramétrico utilizado para comparar medições repetidas realizadas sobre o mesmo conjunto de indivíduos ou unidades experimentais em diferentes condições. Diferentemente de testes paramétricos, que assumem a normalidade dos dados, o teste de Friedman baseia-se na ordenação dos valores dentro de cada condição experimental. Ele avalia se há diferenças significativas entre as distribuições dos grupos analisados sem depender de pressupostos como normalidade ou homogeneidade das variâncias. Dessa forma, é especialmente indicado para experimentos com medidas repetidas cujos dados não seguem uma distribuição normal, possibilitando uma análise robusta em tais circunstâncias.

Dado um conjunto de  $k$  condições experimentais, cada uma avaliada em  $n$  instâncias (por exemplo, diferentes execuções do método de Monte Carlo), o teste de Friedman verifica se há diferenças estatisticamente significativas entre as distribuições dessas condições. O valor do teste estatístico  $Q$  é calculado conforme a Equação 7:

$$Q = \frac{12}{nk(k+1)} \sum_{j=1}^k R_j^2 - 3n(k+1), \quad (7)$$

onde  $R_j$  representa a soma dos postos (*rank*s) atribuídos à  $j$ -ésima condição experimental. O procedimento do teste consiste nas seguintes etapas:

- Classificar os valores observados dentro de cada instância atribuindo postos (*rank*s);
- Calcular a soma dos postos para cada condição experimental;
- Determinar se a distribuição dos postos difere significativamente entre as condições analisadas.

O valor de  $Q$  segue aproximadamente uma distribuição qui-quadrado ( $\chi^2$ ) com  $k - 1$  graus de liberdade, quando  $n$  é suficientemente grande. Para avaliar a significância do resultado, compara-se o valor de  $Q$  com valores críticos da distribuição  $\chi^2$  ou calcula-se o valor  $p$ , que representa a probabilidade de obter um resultado tão ou mais extremo sob a hipótese nula. Se  $p \leq 0.05$ , rejeita-se a hipótese nula, indicando que pelo menos uma das condições experimentais apresenta uma distribuição significativamente diferente das demais.

No contexto deste trabalho, o teste de Friedman é empregado para verificar se há diferenças estatísticas entre os resultados obtidos pelo método de Monte Carlo sob diferentes configurações de simulação, garantindo uma avaliação do impacto das variações nos parâmetros do experimento.

#### 2.5.4 Teste de Wilcoxon

O teste de Wilcoxon (Wilcoxon, 1945) é um teste não paramétrico utilizado para comparar duas amostras dependentes quando não se pode assumir que os dados seguem uma distribuição normal. Ele avalia se há diferenças estatisticamente significativas entre duas condições experimentais analisando os sinais e magnitudes das diferenças entre pares de observações. Em vez de utilizar médias e variâncias, como nos testes paramétricos, o teste de Wilcoxon ordena as diferenças absolutas entre os pares, atribui postos e examina se há um padrão sistemático nos sinais dessas diferenças. Sendo assim, uma ferramenta adequada para a análise de dados assimétricos ou com presença de valores extremos.

A estatística de teste do Wilcoxon, denotada por  $W$ , é definida conforme a Equação 8:

$$W = \sum_{i=1}^n \text{sgn}(x_i - y_i) R_i \quad (8)$$

onde:  $x_i$  e  $y_i$  representam as observações emparelhadas dos dois grupos comparados,  $\text{sgn}(x_i - y_i)$  é a função sinal, que assume os valores  $+1$ ,  $0$  ou  $-1$ , dependendo se  $x_i - y_i$  for positivo, nulo ou negativo, respectivamente,  $R_i$  é o posto atribuído ao valor absoluto de  $x_i - y_i$  na ordenação crescente.

O procedimento do teste de Wilcoxon consiste nas seguintes etapas:

- Calcular as diferenças  $x_i - y_i$  entre os pares de observações;
- Excluir os pares onde  $x_i = y_i$ , pois eles não contribuem para a estatística de teste;
- Ordenar os valores absolutos das diferenças e atribuir postos  $R_i$ ;
- Somar separadamente os postos dos valores positivos e negativos;
- Determinar a estatística de teste  $W$  e compará-la com os valores críticos da distribuição de Wilcoxon.

A significância estatística do teste é avaliada pelo valor-p ( $p$ ), que representa a probabilidade de se observar uma estatística de teste tão extrema quanto a obtida, sob a hipótese nula de que não há diferença entre os grupos. Se  $p \leq 0.05$ , rejeita-se a hipótese nula, indicando que há uma diferença estatisticamente significativa entre os dois conjuntos de dados. No contexto deste trabalho, o teste de Wilcoxon é empregado para comparar

os resultados de diferentes configurações da simulação de Monte Carlo, assegurando que as diferenças observadas não sejam aleatórias.

### 3 Trabalhos relacionados

Um problema relacionado às tarefas de busca e exploração é a vigilância persistente, na qual áreas de interesse devem ser revisitadas periodicamente em um ciclo contínuo. Esse problema pode ser modelado como um problema de patrulhamento, classificado como NP-difícil e amplamente estudado no contexto de tarefas multiagente (Machado et al., 2003; Chevaleyre, 2004; Pasqualetti et al., 2010). Além da detecção de derramamento de óleo, sistemas desenvolvidos para esse fim podem fornecer dados amostrais úteis para o planejamento e execução de operações navais, gestão da poluição e pesca, entre outras aplicações (Ferri et al., 2015).

A representação do mundo varia de acordo com o algoritmo de vigilância implementado. (Othmani-Guibourg et al., 2017) implementou um algoritmo multiagente dinâmico por meio da representação gráfica do ambiente. Por outro lado, os algoritmos baseados em grade são usados quando não há uma representação topológica clara do mundo ou quando a técnica depende do cálculo de funções baseadas em relações de células de grade. Um sistema de Veículos Aéreos Não Tripulados ou UAV (Unmanned Aerial Vehicle) que realiza vigilância de área e monitoramento de intrusos foi proposto em (Patrinopoulou et al., 2022), usando um mundo de grade. Esse trabalho usou a representação de grade, pois as implementações de linha de base já consideravam os valores das células e a área do mundo é esparsa.

Abordando diretamente o problema de persistência em missões de vigilância por UAVs, o trabalho de (Nigam et al., 2011) apresenta uma política de controle reativa. A abordagem combina métodos formais baseados em derivação e heurísticas escaláveis. O estudo analisa o acoplamento entre a política de controle e as restrições dinâmicas e de resistência, propondo modificações pouco exploradas na literatura. Testes de voo são realizados para validar a robustez da solução diante de limitações e falhas reais. A pesquisa considera um espaço-alvo bidimensional, discretizado em células com um atributo de idade, que representa o tempo desde a última observação. O objetivo é minimizar a idade máxima das células ao longo do tempo, garantindo que nenhuma região permaneça sem observação por períodos prolongados. A coordenação multi-UAV é modelada de acordo com a Equação 9.

$$v_j = \max \left\{ \left( c_j + w_{di} \delta_{ij} + w_{ni} \min_{k \neq i} (\delta_{kj}) \right), 0 \right\} \quad (9)$$

No qual cada  $i$ -ésimo agente calcula o valor de cada  $j$ -ésima célula no mapa,  $c_j$  é o valor que representa a idade da célula,  $w_{di}$   $w_{ni}$  representam pesos relacionados às distâncias em cima dos robôs,  $\delta_{ij}$  representa a distância entre o agente  $i$  e a célula  $j$  e  $\delta_{kj}$  a distância entre o vizinho  $k$  e a célula  $j$ . O trabalho (Nigam et al., 2011) possui a limitação do ambiente

de simulação, sendo somente aplicado em ambientes com uma geometria quadrada.

Já o trabalho (Leite et al., 2021) possui uma abordagem mais abrangente, é baseado no trabalho (Nigam et al., 2011), com a adaptação para ambientes com diferentes formatos, com a possibilidade de inserção de obstáculos e adaptação a cenários diversos. Nesse trabalho, o cenário usado é a orla alagoana, entretanto, é trazida uma estrutura modular para inserção de ambientes diferentes. O presente trabalho se baseia nessa estrutura fornecida.

Outra diferença significativa para ser destacada é a inserção do algoritmo  $A^*$ , como introduzido na subseção 2.3. Este algoritmo ajuda no processo de decisão do próximo passo de cada robô, sendo adaptado a diferentes formatos de mapas. Em conjunto com a tomada de decisão para o próximo passo do robô, este trabalho usa a seguinte política, que no  $A^*$  será  $h(n)$ , representada na equação 10.

$$v_{ij} = \max \left\{ \kappa_i + w_{ci}c_j + w_{si}s_j + w_{di}\delta_{ij} + w_{ni} \min_{k \neq i} \delta_{kj}, 0 \right\} \quad (10)$$

No qual  $w_{ci}$  e  $w_{si}$  são pesos referentes à concentração e sensibilidade, respectivamente,  $w_{di}$  e  $w_{ni}$  são os pesos em cima dos robôs relacionados às distâncias,  $\kappa_i$  uma constante de ganho,  $c_j$  o valor do KDE, que representa a sujeira da célula, ou ainda, a sua idade,  $s_j$  o valor de sensibilidade do ambiente,  $\delta_{ij}$  a distância entre o agente  $i$  e a célula  $j$  e  $\delta_{kj}$  a distância entre o vizinho  $k$  e a célula  $j$ .

A estratégia adotada busca distribuir os agentes de forma a priorizar células com alta probabilidade de conter óleo ou áreas consideradas criticamente sensíveis. O alvo de cada agente é a célula com maior valor de  $v_{ij}$ . Para navegar até seu destino, cada agente utiliza o algoritmo de busca  $A^*$ , garantindo a evasão de colisões e o respeito às zonas de exclusão aérea, além de possibilitar a cobertura de regiões de interesse com formatos não quadráticos. A cada período  $t_r$ , a política reativa retorna as posições atuais dos agentes.

Diante desses trabalhos analisados, observa-se uma evolução na modelagem de estratégias para vigilância persistente com múltiplos agentes. As abordagens mais recentes incorporam técnicas clássicas de busca, como o algoritmo  $A^*$ , e ampliam a representação do espaço, contemplando cenários com maior complexidade geométrica e presença de obstáculos. Nesse contexto, o presente trabalho propõe uma adaptação do trabalho visto em (Nigam et al., 2011), seguindo a sua política exposta pela Equação 9, adaptada ao contexto do trabalho visto em (Leite et al., 2021), para que ambas as técnicas possam ser comparadas no mesmo contexto.

## 4 Metodologia

Na seção atual, será abordada a dinâmica dos experimentos para geração dos resultados. Será abordada a Arquitetura de simulação, explicando como o sistema está funcionando para a geração dos dados, os três algoritmos estudados, a métrica utilizada para comparar os algoritmos e, por fim, os cenários de simulação implementados.

### 4.1 Arquitetura de simulação

Os experimentos desenvolvidos neste trabalho são estruturados em torno de duas tecnologias principais: a plataforma MATLAB® e a linguagem *Python*. O primeiro é responsável por compilar e rodar os *scripts* na linguagem MATLAB® e o segundo é usado como ferramenta acessória à aplicação central.

Como exibido na Figura 3, a aplicação central é comandada pelo *Manager*, que gerencia os *Grids* e os ciclos da simulação. O ambiente de simulação a cada ciclo irá ser atualizado, ou seja, o óleo será espalhado respeitando a sua dinâmica, como explicado na subseção 2.1. Para isso, as informações sobre a posição das partículas e o tempo atual da simulação serão passadas para o *script* feito em *Python*, que utiliza a biblioteca *PyGnome* para simular a próxima posição das partículas. Em seguida, essa informação é mandada novamente para o *script* principal *Manager*.

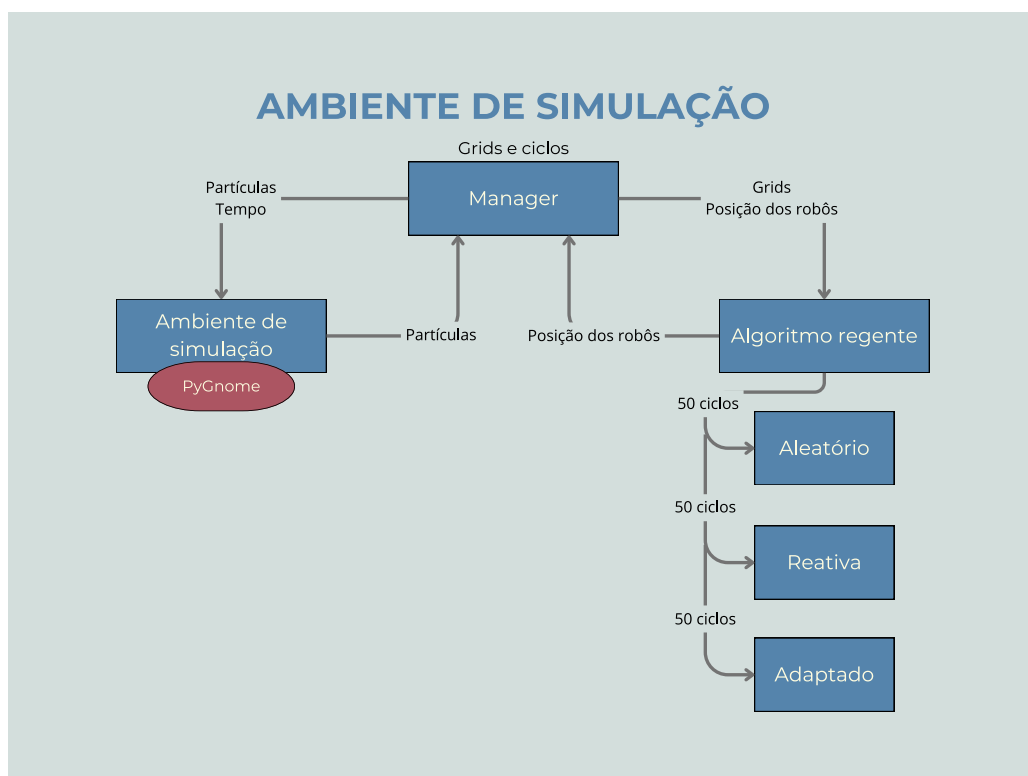


Figura 3: Diagrama referente ao ambiente de simulação, demonstrando como o *escript* principal “*Manager*” atua controlando a simulação.

O gerenciador da aplicação irá controlar qual algoritmo regente irá movimentar os robôs, para isso, é mandada a informação sobre como o *Grid* atual está, com as posições das manchas de óleo. O algoritmo regente atual da simulação será selecionado, podendo ser a política aleatória, reativa ou adaptada, explicados com mais detalhes na subseção 4.3. Após passar um ciclo da simulação, a posição dos robôs já atualizada é mandada para o gerenciador. Por fim, esse realiza a limpeza do ambiente de acordo com a posição atual dos robôs e atualiza o ciclo. Dessa forma, o ambiente de simulação é atualizado e os robôs realizam a limpeza do ambiente de acordo com a lógica de limpeza vigente.

## 4.2 Algoritmos de Vigilância

Neste trabalho são comparados 3 algoritmos: uma política aleatória, servindo de pior cenário para os robôs se comportarem; a política reativa; e a política adaptada, como introduzidas na seção 3. Nessa subseção, cada um será mais explorado, apresentando uma noção melhor de seu funcionamento.

### 4.2.1 Política aleatória

A política aleatória é usada neste trabalho para servir de base. É esperado aqui o pior comportamento, no qual os robôs irão navegar pelo ambiente de maneira aleatória, sem a pretensão de limpar o ambiente. Sendo assim, caso os resultados sejam próximos do caso aleatório, significa que os outros dois algoritmos não são significativamente melhores que o pior caso e, portanto, não valem a pena de serem usados por possuírem um custo bem maior que o pior caso.

O algoritmo funciona movimentando os robôs de maneira aleatória pelo mapa, respeitando os limites do mapa. Esse algoritmo é usado tanto no momento da simulação que utiliza a política aleatória como algoritmo regente, como é usado também no método Monte Carlo, como será visto à frente, para que os robôs sempre iniciem em posições aleatórias na região de início.

---

**Algorithm 1** Caminhada Aleatória de Robôs
 

---

```

1: Entrada: Grid, posição dos robôs, direções, máscara de obstáculos
2: Saída: Nova posição dos robôs e suas direções
3:  $n\_robots \leftarrow$  número de robôs
4:  $aux\_mask \leftarrow mask$ 
5: for cada robô  $r$  de 1 até  $n\_robots$  do
6:    $neighbors \leftarrow$  posições dos outros robôs
7:    $max\_attempts \leftarrow 8$ 
8:    $attempt \leftarrow 0$ 
9:    $valid\_move \leftarrow$  falso
10:  while  $attempt < max\_attempts$  e  $valid\_move$  é falso do
11:     $move \leftarrow$  posição aleatória em  $\{-1, 0, 1\}^2$ 
12:     $new\_position \leftarrow robots[r] + move$ 
13:    if  $new\_position$  está dentro dos limites e não é obstáculo then
14:       $robots[r] \leftarrow new\_position$ 
15:       $heading[r] \leftarrow atan2(move_y, move_x)$ 
16:       $valid\_move \leftarrow$  verdadeiro
17:    else
18:       $attempt \leftarrow attempt + 1$ 
19:    end if
20:  end while
21: end for

```

---

#### 4.2.2 Política reativa

A política reativa, como exposta na seção 3 é regida pela Equação 10, exposta no algoritmo abaixo. O algoritmo recebe o *grid* atual, preenchido com o óleo espalhado, as posições dos robôs, as direções que os robôs estão apontando, uma máscara que delimita o ambiente que o robô pode navegar, a matriz de distâncias e uma matriz com os pesos. O algoritmo irá calcular as novas posições dos robôs, retornando isso e para onde o robô está virado.

**Algorithm 2** Patrulha Reativa de Robôs

---

```

1: Entrada: Grid, posição dos robôs, direções, máscara de obstáculos, matriz de dis-
   tâncias, pesos
2: Saída: Nova posição dos robôs e suas direções
3:  $n\_robots \leftarrow$  número de robôs
4:  $aux\_mask \leftarrow mask$ 
5: for cada robô  $r$  de 1 até  $n\_robots$  do
6:    $neighbors \leftarrow$  posições dos outros robôs
7:    $target \leftarrow \text{COMPUTETARGETMULTI}()$ 
8:   if  $\|target - robots[r]\| > 0$  then ▷ Verifica se o robô deve se mover
9:      $GoalRegister \leftarrow$  matriz de zeros do tamanho de  $mask$ 
10:     $GoalRegister[target_y, target_x] \leftarrow 1$ 
11:    for cada vizinho  $n$  em  $neighbors$  do
12:       $aux\_mask[n_y, n_x] \leftarrow 1$  ▷ Marca posições ocupadas
13:    end for
14:     $result \leftarrow \text{ASTARPATH}(robots[r], aux\_mask, GoalRegister, 1)$ 
15:    if  $\text{size}(result, 1) > 1$  then ▷ Se há um caminho válido
16:       $move \leftarrow [result[end - 1, 2] - robots[r, 1], result[end - 1, 1] - robots[r, 2]]$ 
17:       $robots[r] \leftarrow robots[r] + move$ 
18:       $heading[r] \leftarrow \text{atan2}(move_y, move_x)$ 
19:    end if
20:  end if
21: end for

```

---

O elemento da matriz dos pesos é o que vai diferenciar a política reativa e a política adaptada. Na política reativa, os pesos podem ser expostos na Tabela 1:

| Robô | $W_{ci}$ | $W_{si}$ | $W_{di}$ | $W_{ni}$ | $\kappa_i$ |
|------|----------|----------|----------|----------|------------|
| 1    | 2.0      | 0.1      | 0.3      | 0.2      | 1          |
| 2    | 2.0      | 0.1      | 0.3      | 0.2      | 1          |
| 3    | 2.0      | 0.1      | 0.3      | 0.2      | 1          |
| 4    | 0.1      | 2.0      | 0.5      | 0.1      | 1          |
| 5    | 0.1      | 2.0      | 0.5      | 0.1      | 1          |
| 6    | 0.1      | 2.0      | 0.5      | 0.1      | 1          |
| 7    | 0.1      | 2.0      | 0.5      | 0.1      | 1          |

Tabela 1: Pesos usados para a política reativa

A referente Tabela possui os pesos usados na simulação, sendo possível perceber que existem dois grupos principais de robôs. O primeiro grupo é movido pelo peso  $W_{ci}$ , com

o foco na limpeza de células com alta concentração, e o grupo movido pelo peso  $W_{si}$ , com foco em atuar em regiões de maior sensibilidade. Tal configuração de pesos foi herdada do trabalho (Leite et al., 2021).  $k_i$ , por ser unitário em todos, não possui diferença significativa para a simulação. Porém, caso fosse necessário, ele agiria dando um peso maior para um grupo específico de robôs.

### 4.2.3 Política adaptada

A política adaptada é a política da Equação 9 adaptada ao trabalho (Leite et al., 2021). Ou seja, utiliza todos os recursos adicionados na política anterior, mas, assim como a política do trabalho (Nigam et al., 2011), não possui alguns pesos específicos para os robôs. Sendo assim, o algoritmo apresentado na subseção 4.2.2 também será usado para a abordagem atual, com a diferença da matriz dos pesos, exposta pela Tabela 2:

| Robô | $W_{ci}$ | $W_{si}$ | $W_{di}$ | $W_{ni}$ | $\kappa_i$ |
|------|----------|----------|----------|----------|------------|
| 1    | 1.0      | 0.0      | 0.3      | 0.2      | 0.0        |
| 2    | 1.0      | 0.0      | 0.3      | 0.2      | 0.0        |
| 3    | 1.0      | 0.0      | 0.3      | 0.2      | 0.0        |
| 4    | 1.0      | 0.0      | 0.5      | 0.1      | 0.0        |
| 5    | 1.0      | 0.0      | 0.5      | 0.1      | 0.0        |
| 6    | 1.0      | 0.0      | 0.5      | 0.1      | 0.0        |
| 7    | 1.0      | 0.0      | 0.5      | 0.1      | 0.0        |

Tabela 2: Pesos usados para a política adaptada

É possível observar que a Tabela 2 apresenta o peso  $W_{ci}$  de maneira homogênea, igual a 1, ou seja, o peso da concentração da célula não apresenta distinção, tendo influência sobre o algoritmo, porém não tendo o mesmo efeito de preocupação com a orla. Já o peso  $W_{si}$ , referente à sensibilidade da célula, é retirado da equação. Os outros pesos são mantidos iguais à simulação anterior, o foco da atual simulação é realizar um comportamento livre e sem preocupação com a concentração nem a sensibilidade das células. Por fim, importante ressaltar que  $k_i = 0$  pois, a Equação 9 não apresenta um valor de ganho para nenhum robô, não permitindo distinção ou privilégio a grupos específicos de robôs.

## 4.3 Proposta de métrica

A métrica elaborada para comparação entre os algoritmos está representada pela Equação 11.

$$L_f = \sum_{n=1}^{n_f} \sum_{i=0}^{i_f} \sum_{j=0}^{j_f} (X_{ij}^n) \quad (11)$$

no qual  $n_f$  é o número de simulações para cada dado,  $i_f$  é o número de linhas do ambiente,  $j_f$  o número de colunas do ambiente,  $L_f$  é a soma de células sujas no tempo de simulação e  $X_{ij}^n$  representa cada célula no instante  $n$  da simulação. A métrica é a medida do acúmulo da quantidade de células sujas. Tal métrica sempre será positiva, pois os valores sempre serão positivos, portanto não precisa de valor absoluto ou quadrático.

#### 4.4 Cenários de simulação

Os conjuntos de dados obtidos estão estruturados em cima de 5 cenários de simulação. Cada cenário difere no início da posição em que os 7 robôs irão realizar a limpeza do ambiente.

Como é possível destacar na Figura 4, o cenário 1 possui o início dos robôs no centro do cenário válido de simulação, o cenário 2 possui o início dos robôs embaixo à esquerda, o cenário 3 com o início dos robôs em cima à esquerda, o cenário 4 embaixo à direita e o cenário 5 em cima à direita.

Os 5 cenários estão implementados para testar e comparar os algoritmos de acordo com a posição inicial dos robôs no cenário. Por ser um ambiente dinâmico, a posição inicial dos robôs pode influenciar em como o ambiente é limpo, visto que a água e as manchas de óleo estão sob influência do vento e da corrente, como exposto na subseção 2.1.

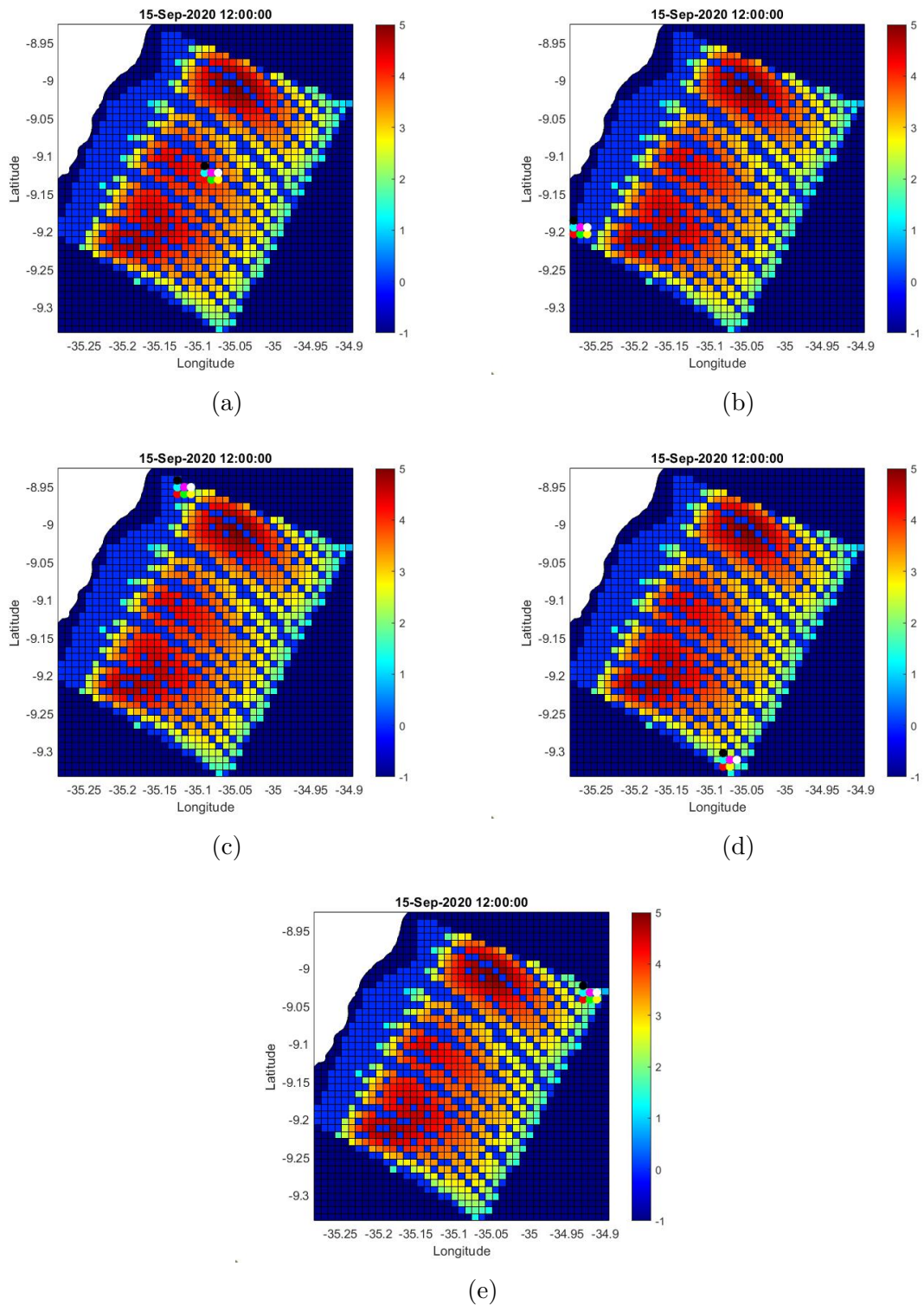


Figura 4: Distribuição dos robôs nos diferentes cenários de teste: (a)Cenário 1 - robôs no centro; (b)Cenário 2 - robôs embaixo à esquerda; (c)Cenário 3 - robôs em cima à esquerda; (d)Cenário 4 - robôs embaixo à direita; (e)Cenário 5 - robôs em cima à direita. A barra de cores ao lado refere-se à concentração de petróleo, azul escuro é referente à nenhuma concentração e vermelho a maior concentração(5).

#### 4.4.1 Implementação do método Monte Carlo

Na subseção 2.4, o método Monte Carlo é trazido. Muitos dados são gerados nas simulações e, para a aplicação do método, o início do tempo referente à simulação, mudança entre os tipos de cenário e a posição inicial dos robôs para a geração de dados sem o mesmo viés.

O primeiro ponto levantado é sobre o início do tempo referente na simulação. Essa variável muda precisamente o início temporal que os dados de vento e maré são iniciados na simulação. Por exemplo, a primeira amostra iniciará às 12h00 do dia 15 de setembro de 2020. A próxima amostra iniciará às 12h10 do mesmo dia. Tal técnica, que é um janelamento temporal, contribui para que as amostras tenham contato com diferentes dados de vento e maré ao longo do tempo.

O segundo ponto levantado é o tipo de cenário. Cada um dos 5 tipos de cenários é rodado em cada uma das 3 políticas, gerando ao todo 300 dados por cenário, divididos em 3 grupos de 100 dados. Para a geração de uma maior aleatoriedade nos dados, seguindo as regras do método dessa subseção, ao início de cada geração de dado, independente do cenário inicial dos robôs, a posição inicial de cada robô é alterada de maneira aleatória, utilizando o algoritmo trazido pela subseção 4.2.1.

Essa parte anterior é referente ao ponto 3 levantado previamente. Sendo assim, cada robô, iniciando em cada um dos cenários expostos, irá ter uma posição aleatória dentro de um raio de uma célula e dentro do cenário permitido da simulação.

#### 4.4.2 Análise dos dados colhidos

A geração de todos os dados está apresentada na Figura 5. Este diagrama representa o fluxo da simulação para a geração de todos os dados. Como por cenário, são gerados 300 dados, a simulação toda gera 1500 dados. A comparação das técnicas será feita por cenário, ou seja, cada cenário possuirá uma amostragem de 300 dados, 100 dados por algoritmo.

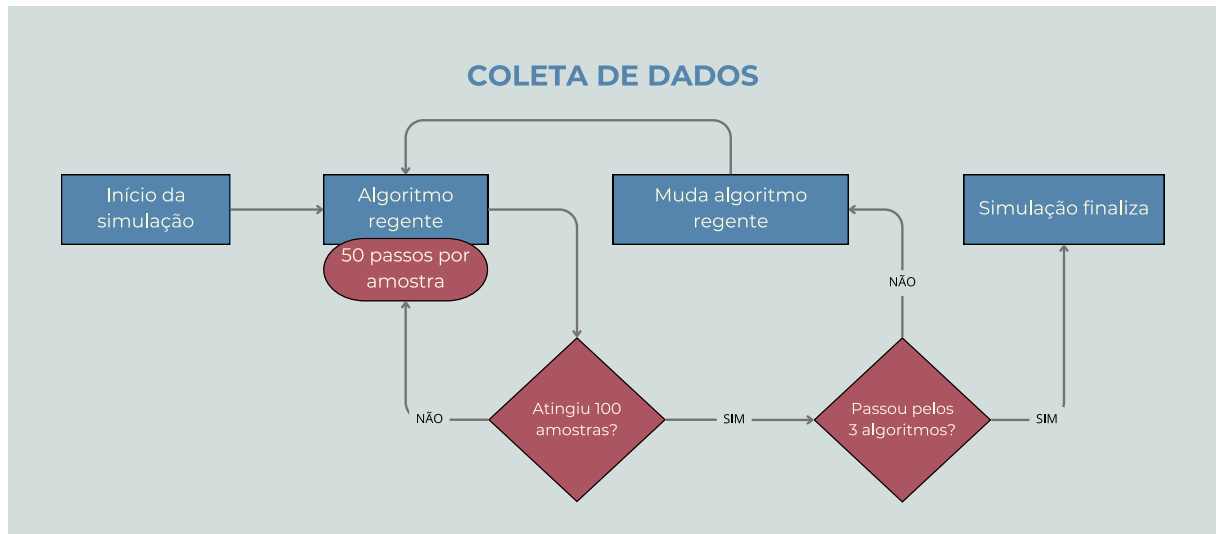


Figura 5: Diagrama que representa o fluxo seguido pela simulação para coleta das amostras.

Os testes estatísticos neste trabalho são usados para comparar os conjuntos de dados de maneira correta, seja para poder usar a média, seja para usar a mediana, seguindo a seguinte lógica:

- O teste de Shapiro-Wilk é aplicado para verificar a normalidade dos dados gerados pela simulação de Monte Carlo. É gerado um valor- $p$ , caso esse valor seja menor que 0.05, os dados são normalizados, caso sejam maiores, não.
- Se os dados forem normalmente distribuídos, a comparação é feita utilizando a média.
- Se os dados não forem normais, testes não paramétricos são utilizados:
  - O teste de Friedman é empregado para comparar múltiplas execuções da simulação. Um valor- $p$  é gerado e caso ele seja menor que 0.05, os dados são considerados inválidos, caso sejam maiores, o próximo teste é aplicado.
  - O teste de Wilcoxon é utilizado para comparações entre pares de condições experimentais. Um valor- $p$  é gerado e caso ele seja menor que 0.05, os dados são considerados inválidos, caso sejam maiores, a mediana é usada para comparação dos dados.
- Ao passar pelos testes paramétricos, caso o resultado mostre que as amostras não são relacionadas, a mediana é usada para comparar os resultados das simulações.

A aplicação dessas técnicas garante que a análise dos resultados da simulação seja conduzida de forma estatisticamente rigorosa, permitindo comparações adequadas entre diferentes cenários. Tal lógica é demonstrada na Figura 6, a seguir.



Figura 6: Diagrama que demonstra o processo da avaliação dos resultados.

## 5 Resultados e discussões

Nesta seção encontram-se expostos os resultados das simulações realizadas, assim como a aplicação das métricas estatísticas e a interpretação dos resultados. Por fim, é realizada uma análise final sobre o cenário mais eficiente da aplicação.

### 5.1 Cenário 01

O cenário 01, previamente estabelecido com o início dos robôs no centro, gerou os dados representados na Tabela 3 e as trajetórias de uma das simulações estão expostas na Figura 7.

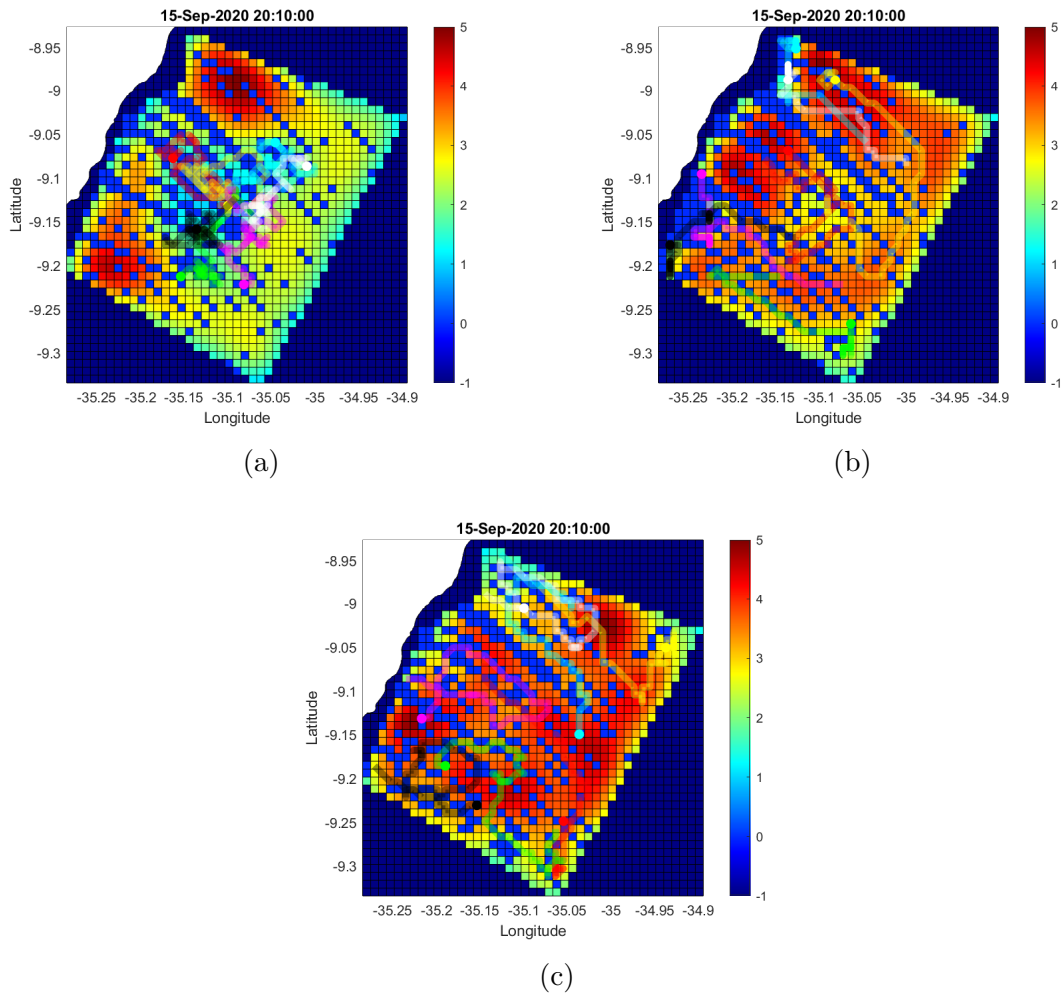


Figura 7: Resultado da simulação do cenário com início no centro. A imagem (a) é referente ao algoritmo aleatório, a (b) é referente ao algoritmo reativo e a (c) é referente ao algoritmo adaptado. A barra de cores ao lado refere-se à concentração de petróleo, azul escuro é referente à nenhuma concentração e vermelho a maior concentração(5).

| Política  | Média    | Mediana | Teste de normalidade |
|-----------|----------|---------|----------------------|
| Aleatória | 36297.43 | 36357.5 | Negativo             |
| Reativa   | 3588.94  | 35372.0 | Negativo             |
| Adaptada  | 35472.8  | 35545.0 | Negativo             |

Tabela 3: Resultado inicial da simulação 01.

Com os resultados expostos acima, é possível notar que os testes de normalidade retornaram “Negativos”, indicando que os dados não estão distribuídos de maneira normalizada e que o  $p < ou = 0.05$ . Portanto, faz-se necessário realizar os testes de Friedman 2.5.3 e Wilcoxon 2.5.4.

Em ambos os testes,  $p < ou = 0.05$ , portanto, os dados seguem válidos na avaliação, sendo possível usar a mediana como referência estatística de comparação entre os resultados. Dessa forma, os resultados a serem comparados estão expostos na Tabela 4.

| Política  | Resultado |
|-----------|-----------|
| Aleatória | 36357.5   |
| Reativa   | 35372.0   |
| Adaptada  | 35545.0   |

Tabela 4: Resultado final da simulação 01.

Como previsto, o resultado vindo da política “Aleatória” é o pior resultado, com uma diferença de 985.5 células sujas com relação à melhor política, que foi a “Reativa”. Nesse caso em específico, é coerente que tal algoritmo tenha desempenhado melhor resultado, pois possui uma preocupação com a orla, enquanto 3 robôs estão indo para limpar a orla, realizam a limpeza acompanhando a movimentação da maré, como é possível observar na Figura 7.

Já na política “Adaptada”, os robôs se espalham de maneira mais uniforme pelo ambiente, deixando de acompanhar o movimento dinâmico em direção à orla.

## 5.2 Cenário 02

O cenário 02, previamente estabelecido com o início dos robôs em cima e na direita, gerou os dados representados na Tabela 5 e as trajetórias de uma das simulações estão expostas na Figura 8.

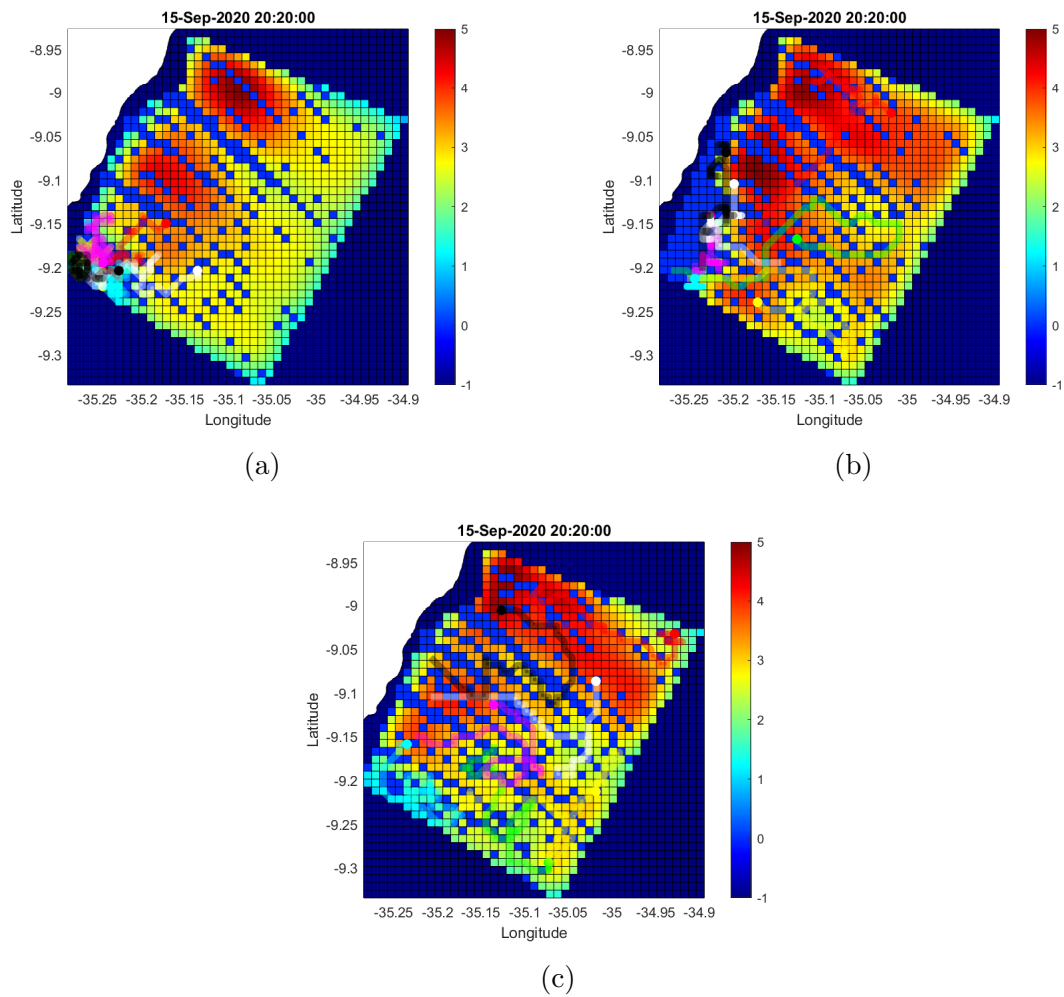


Figura 8: Resultado da simulação do cenário com início em cima e na direita. A imagem (a) é referente ao algoritmo aleatório, a (b) é referente ao algoritmo reativo e a (c) é referente ao algoritmo adaptado. A barra de cores ao lado refere-se à concentração de petróleo, azul escuro é referente à nenhuma concentração e vermelho a maior concentração(5).

| Política  | Média    | Mediana | Teste de normalidade |
|-----------|----------|---------|----------------------|
| Aleatória | 36087.53 | 36145.5 | Positivo             |
| Reativa   | 35737.72 | 35824.0 | Positivo             |
| Adaptada  | 35662.54 | 35735.0 | Positivo             |

Tabela 5: Resultado inicial da simulação 02.

Como é possível observar, todos os testes de normalidade vieram positivos, sendo assim, a média pode ser usada para comparar os resultados. É possível visualizar os resultados finais na Tabela 6.

| Política  | Resultado |
|-----------|-----------|
| Aleatória | 36087.53  |
| Reativa   | 35737.72  |
| Adaptada  | 35662.54  |

Tabela 6: Resultado final da simulação 02.

A política aleatória resultou no pior resultado, com 424.99 células sujas de diferença com o melhor resultado vindo da política Adaptada. Uma possível explicação para esse resultado é o início dos robôs, como começam embaixo e na esquerda, uma região que a maré e o vento estão indo, portanto, inicia limpando sem distinção. Entretanto, importante ressaltar que a diferença foi de menos de 100 células para a política reativa. Para um resultado melhor com uma diferença maior, é recomendada uma simulação com  $n > 50$ .

### 5.3 Cenário 03

O cenário 03, previamente estabelecido com o início dos robôs embaixo e na direita, gerou os dados representados na Tabela 7 e as trajetórias de uma das simulações estão expostas na Figura 9.

| Política  | Média    | Mediana | Teste de normalidade |
|-----------|----------|---------|----------------------|
| Aleatória | 36211.59 | 36241.5 | Positivo             |
| Reativa   | 35629.51 | 35627.0 | Positivo             |
| Adaptada  | 35803.68 | 35814.5 | Positivo             |

Tabela 7: Resultado inicial da simulação 03.

Como é possível observar, todos os testes de normalidade vieram positivos, sendo assim, a média pode ser usada para comparar os resultados. É possível visualizar os resultados finais na Tabela 8.

| Política  | Resultado |
|-----------|-----------|
| Aleatória | 36211.59  |
| Reativa   | 35629.51  |
| Adaptada  | 35803.68  |

Tabela 8: Resultado final da simulação 03.

A política aleatória resultou no pior resultado, com 582.08 células sujas de diferença com o melhor resultado vindo da política Adaptada. Com um resultado muito bom, a política reativa, como demonstrado na Figura 9, mostra a sua preocupação com a orla, limpando muito bem de onde inicia e impedindo que as manchas de óleo cheguem na orla, diferente das outras duas políticas.

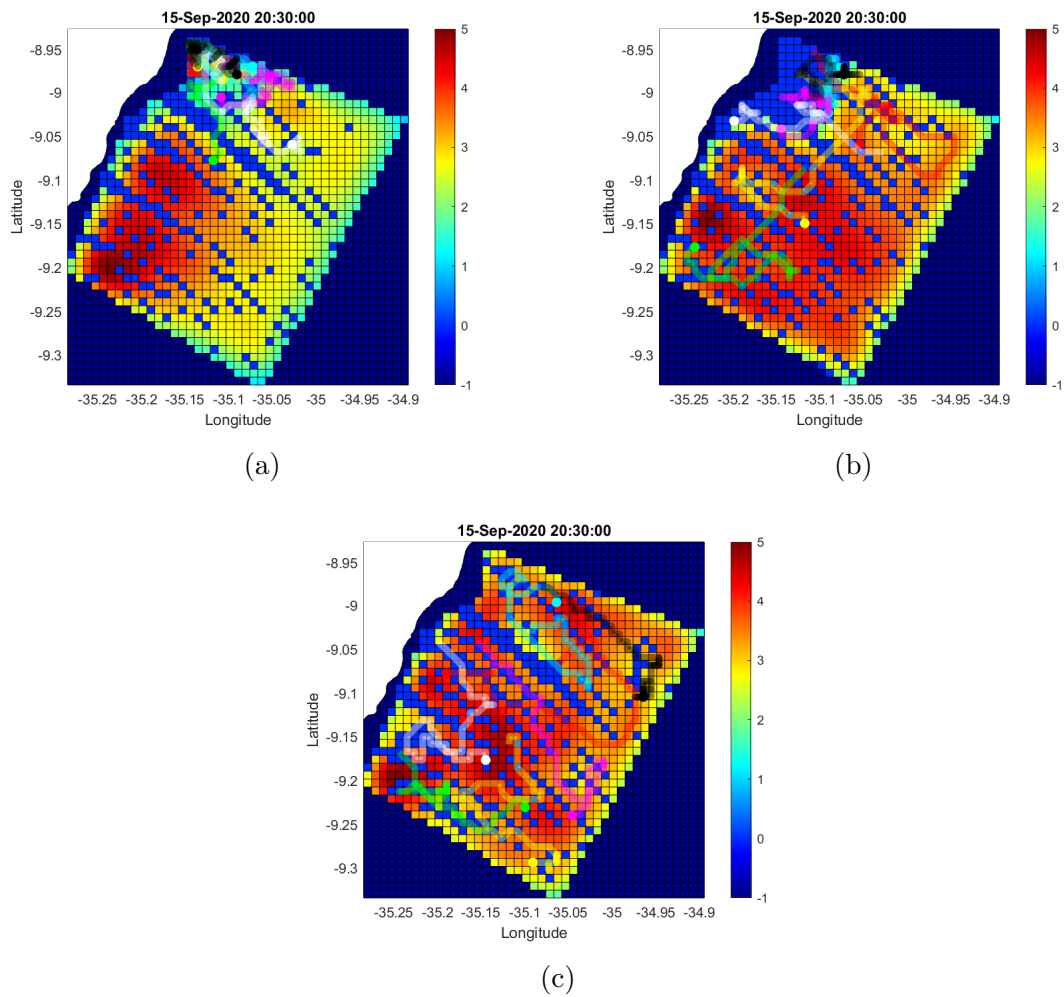


Figura 9: Resultado da simulação do cenário com início embaixo e na direita. A imagem (a) é referente ao algoritmo aleatório, a (b) é referente ao algoritmo reativo e a (c) é referente ao algoritmo adaptado. A barra de cores ao lado refere-se à concentração de petróleo, azul escuro é referente à nenhuma concentração e vermelho a maior concentração(5).

## 5.4 Cenário 04

O cenário 04, previamente estabelecido com o início dos robôs em cima e na esquerda, gerou os dados representados na Tabela 9 e as trajetórias de uma das simulações estão expostas na Figura 10.

| Política  | Média    | Mediana | Teste de normalidade |
|-----------|----------|---------|----------------------|
| Aleatória | 36686.77 | 36737.0 | Positivo             |
| Reativa   | 35543.7  | 35614.0 | Positivo             |
| Adaptada  | 35558.91 | 35675.0 | Negativo             |

Tabela 9: Resultado inicial da simulação 04.

É possível notar como o dado vindo das políticas Aleatório e Reativo retorna Positivo

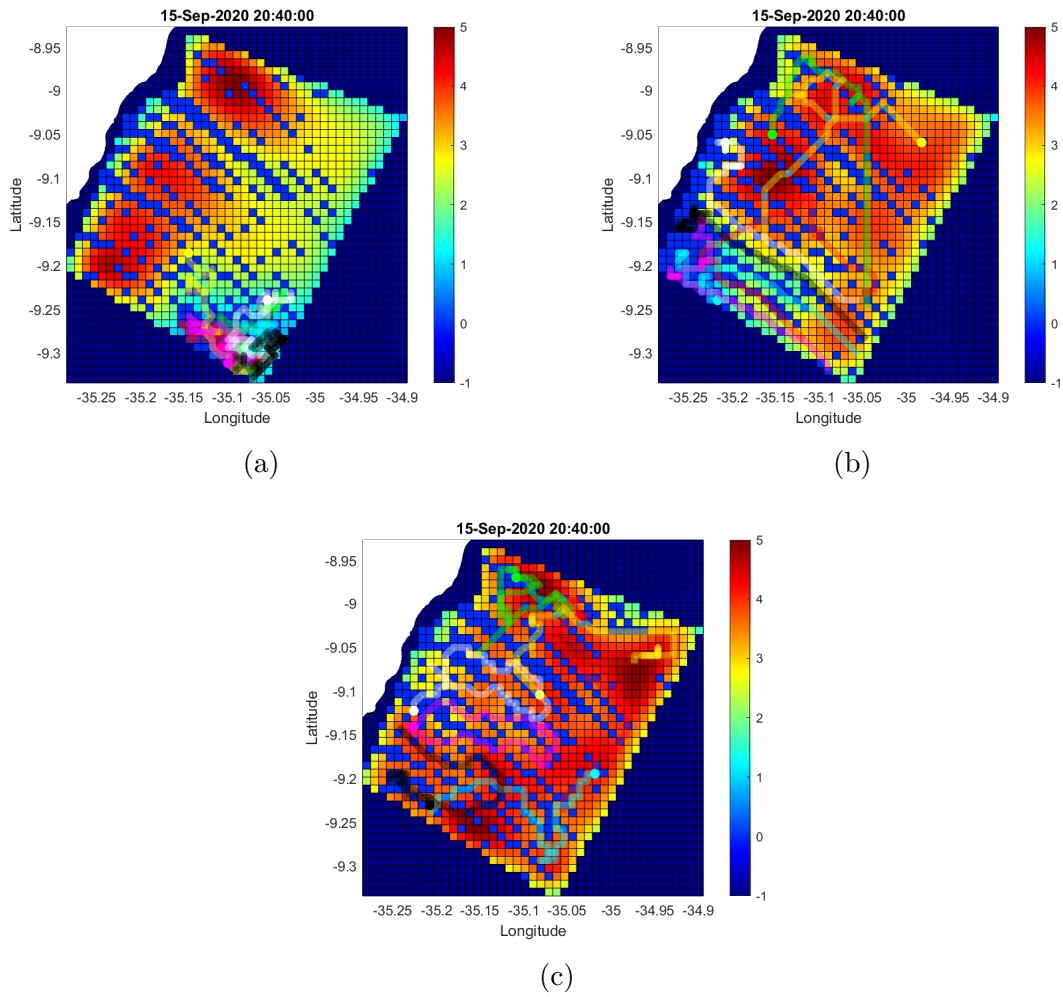


Figura 10: Resultado da simulação do cenário com início em cima e na esquerda. A imagem (a) é referente ao algoritmo aleatório, a (b) é referente ao algoritmo reativo e a (c) é referente ao algoritmo adaptado. A barra de cores ao lado refere-se à concentração de petróleo, azul escuro é referente à nenhuma concentração e vermelho a maior concentração(5).

para o teste de normalidade, porém o dado vindo da política adaptada retorna Negativo para este teste. Portanto, faz-se necessário realizar os testes de Friedman 2.5.3 e Wilcoxon 2.5.4.

Ao realizar o primeiro teste, a amostra retorna Positivo, constatando que as três amostras não possuem similaridades significativas. Porém, ao rodar o segundo teste, a resposta obtida da comparação de 2 em 2 retorna que a amostra vinda da política reativa contém similaridades com a amostra vinda da política Adaptada.

Tal resultado, como explicitado na Figura 6, invalida os dados oriundos dessas duas políticas. Por consequência, é possível elaborar duas justificativas para essa resposta, sabendo que os dados foram invalidados por possuírem um comportamento parecido nesse cenário ao iniciarem nesta determinada posição no mapa.

- Esse fato pode ocorrer pela quantidade de passos estabelecida neste experimento,

que pode ter sido insuficiente para que os robôs andassem de maneira diferente, já que 50 passos fazem os robôs andarem pelo mapa e não limpem toda a simulação.

- Por outro lado, tal cenário pode indicar que os dois algoritmos possuem comportamentos semelhantes ao iniciarem na referente posição do mapa. Esta hipótese conclui que ambos os algoritmos são adequados para esse cenário, não possuindo um melhor que o outro.

## 5.5 Cenário 05

O cenário 04, previamente estabelecido com o início dos robôs embaixo e na esquerda, gerou os dados representados na Tabela 10 e as trajetórias de uma das simulações estão expostas na Figura 11.

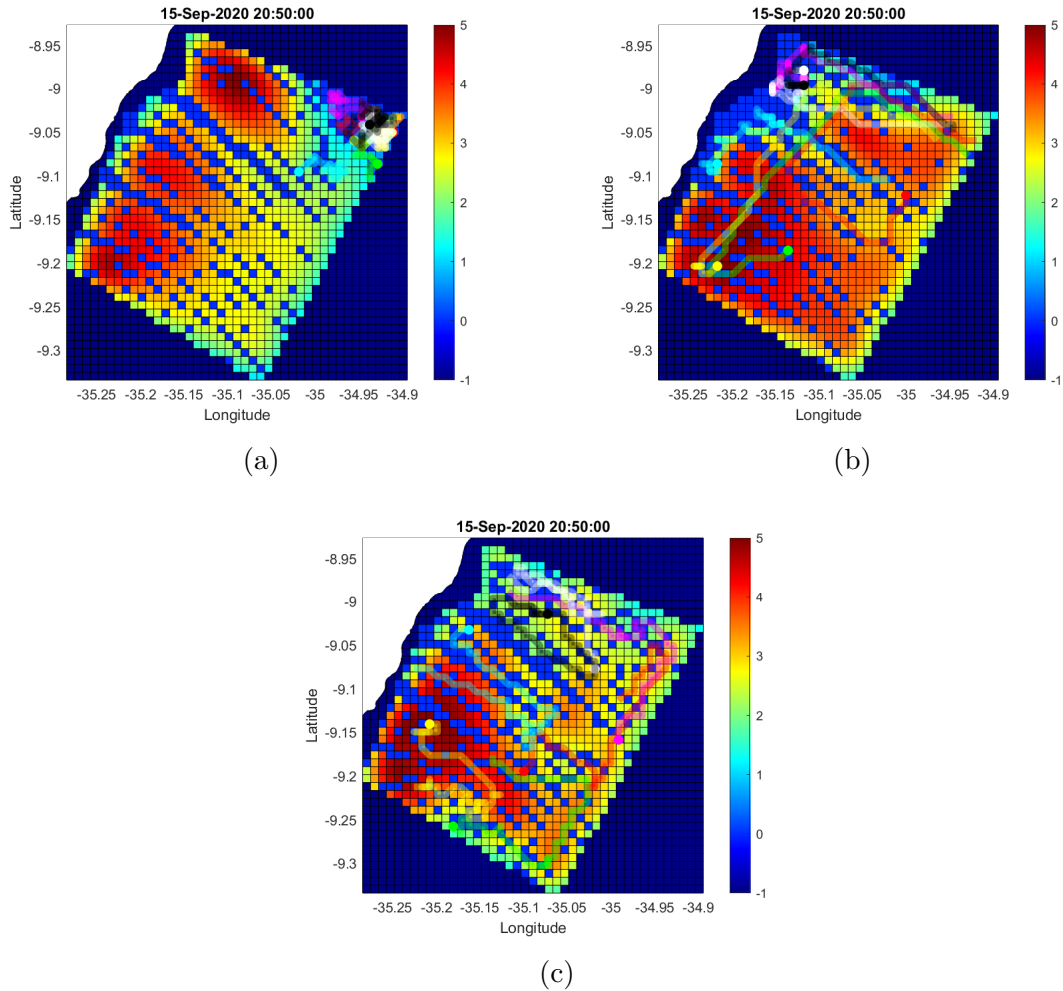


Figura 11: Resultado da simulação do cenário com início embaixo e na esquerda. A imagem (a) é referente ao algoritmo aleatório, a (b) é referente ao algoritmo reativo e a (c) é referente ao algoritmo adaptado. A barra de cores ao lado refere-se à concentração de petróleo, azul escuro é referente à nenhuma concentração e vermelho a maior concentração(5).

| Política  | Média    | Mediana | Teste de normalidade |
|-----------|----------|---------|----------------------|
| Aleatória | 37098.67 | 37158.0 | Positivo             |
| Reativa   | 35699.64 | 35682.5 | Positivo             |
| Adaptada  | 35617.25 | 35686.5 | Positivo             |

Tabela 10: Resultado inicial da simulação 05.

Como é possível observar, todos os testes de normalidade vieram positivos, sendo assim, a média pode ser usada para comparar os resultados. É possível visualizar os resultados finais na Tabela 11.

| Política  | Resultado       |
|-----------|-----------------|
| Aleatória | <b>37098.67</b> |
| Reativa   | 35699.64        |
| Adaptada  | <b>35617.25</b> |

Tabela 11: Resultado final da simulação 05.

A política aleatória resultou no pior resultado, com 1481.42 células sujas de diferença com o melhor resultado vindo da política adaptada. Assim como na simulação do cenário 3, a diferença entre a política adaptada e reativa foi pouca, podendo concluir que uma simulação maior poderá expor melhor que a adaptada é a melhor para esse cenário.

## 5.6 Melhor cenário

Após a exposição dos resultados de todos os 5 cenários, é possível realizar uma comparação dos resultados obtidos em cada cenário e apontar o melhor cenário para o início dos robôs, assim como a melhor política que apresentou o melhor desempenho. A Tabela 12 demonstra um resumo do melhor algoritmo em cada caso. Como resultado, a política adaptada apresentou melhor desempenho em 2 casos, a política reativa foi melhor em 2 e a política aleatória sempre foi o pior caso, como esperado.

| Cenário | Melhor Resultado |
|---------|------------------|
| 01      | Reativa          |
| 02      | Adaptada         |
| 03      | Reativa          |
| 04      | Inconclusivo     |
| 05      | Adaptada         |

Tabela 12: Tabela com os resultados finais.

A Tabela 13 sintetiza os melhores resultados conquistados em cada caso, sendo o Cenário 01 apresentando a menor soma, o melhor desempenho. Sendo assim, como nesse

caso, o algoritmo de melhor desempenho foi o reativo, vindo do trabalho (Leite et al., 2021), pode-se concluir que, para o ambiente de simulação deste trabalho, o ambiente mais adequado para o início dos robôs é no centro do mapa e o algoritmo mais adequado é o reativo.

| Cenário | Melhor Resultado |
|---------|------------------|
| 01      | 35372.0          |
| 02      | 35662.54         |
| 03      | 35629.51         |
| 04      | Inconclusivo     |
| 05      | 35617.25         |

Tabela 13: Tabela com comparação final de resultados.

No geral, pode-se concluir também que a política reativa apresentou um melhor comportamento, apesar de a adaptada ter sido melhor em 2 casos, mesma quantidade de casos que a reativa. A diferença foi pouca, em comparação com a diferença de quando a reativa foi melhor. Sendo assim, de maneira lógica, faz sentido que a reativa tenha apresentado um melhor desempenho, já que o óleo está em direção à orla e essa política possui a preocupação em não deixar a substância atingir essa posição do mapa. Essa política ajuda como uma barreira de contenção, diminuindo os impactos do óleo em corais da costa alagoana.

## 6 Conclusões

O trabalho se mostrou satisfatório em cobrir os objetivos expostos, pois realizou uma comparação entre as técnicas apresentadas de patrulhamento em ambientes dinâmicos. A comparação foi realizada a partir da métrica proposta, chegando, no final, a uma conclusão sobre qual política apresentou o melhor desempenho geral. Importante ressaltar que a presente comparação entre as técnicas não é suficiente para afirmar que uma é melhor que a outra, e sim que, no cenário exposto, a política reativa apresentou um desempenho melhor sobre a política adaptada.

A fim de precisar o desenvolvimento do atual trabalho, faz-se necessário ressaltar que os conceitos necessários para o entendimento do trabalho foram introduzidos, uma noção sobre os trabalhos relacionados foi estabelecida, os cenários de simulação e o método de geração de dados foram determinados, métricas estatísticas foram utilizadas para a obtenção de conhecimento sobre os dados, os resultados em cada cenário foram analisados e a melhor técnica foi definida.

Tal estudo traz uma aplicação modular que contribui para o desenvolvimento de soluções tecnológicas mais eficazes no enfrentamento de desastres ambientais. Por ser modular, permite a adição de novos algoritmos para teste. Como um dos passos futuros, pode-se destacar a comparação de diferentes técnicas às expostas, como uma aprendizagem por reforço ou otimização por enxame de partículas.

Entretanto, importante ressaltar que a atual aplicação possui uma limitação no tempo de simulação, devido à complexidade computacional vinda da simulação do movimento das partículas. Cada cenário durou uma semana para gerar os resultados, limitando o aumento do número de dados e o número de simulações por cenário. Uma possível solução é a implementação da aplicação no *Docker*. Tal ferramenta poderá modularizar ainda mais o sistema e deixar a simulação mais rápida, além de facilitar o uso por outros usuários.

Outro ponto a ser investido futuramente é a adição de mais passos por dados, diferente de 50, podendo comparar se os resultados obtidos possuem diferenças significativas com os resultados atuais, ou seja, o atual número de passos por amostra é suficiente. Nessa lógica, outra questão que pode ser explorada é a geração de mais dados por cenário e por política, mais uma vez podendo ser comparados com o trabalho atual.

Por fim, mais uma área desta aplicação pode ser modificada para produzir resultados diferentes, que é a tabela de pesos de cada política. Estes pesos possuem influência sobre a movimentação dos robôs, portanto, podem melhorar ou piorar o desempenho de cada política. Seguindo a lógica da implementação da política adaptada, que é baseada no trabalho (Nigam et al., 2011) e adaptada nos pesos do trabalho (Leite et al., 2021), a manipulação da tabela de pesos pode ir além da simplificação dos valores. Pode-se zerar ou tornar igual a 1 alguns campos da Tabela 1, para a investigação das possibilidades trazidas pela Equação 10.

# Bibliografia

- Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701.
- Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.
- Metropolis, N. e Ulam, S. (1949). The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341.
- Shapiro, S. S. e Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611.
- Casella, G. e Berger, R. L. (2001). *Statistical Inference*. Cengage Learning, 2 edition.
- Machado, A. et al. (2003). Multi-agent patrolling: An empirical analysis of alternative architectures. *International Journal of Autonomous Agents and Multi-Agent Systems*.
- Wasserman, L. (2004). *All of Statistics: A Concise Course in Statistical Inference*. Springer.
- Rekleitis, I., Lee-Shue, V., New, A., e Choset, H. (2004). Limited communication, multi-robot team based coverage. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3462–3468. IEEE.
- Chevaleyre, Y. (2004). Theoretical analysis of the multi-agent patrolling problem. *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*.
- Bishop, C. M. e Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.
- LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press, Cambridge, UK.
- Wurman, P. R., D’Andrea, R., e Mountz, M. (2008). Coordinating hundreds of cooperative, autonomous vehicles in warehousing: A case study. *AI magazine*, 29(1):9–9.

- Elmaliach, Y., Shiloni, A., e Kaminka, G. A. (2009). Multi-robot area patrol under frequency constraints. *Annals of Mathematics and Artificial Intelligence*, 57(3-4):293–320.
- Pasqualetti, F., Franchi, A., e Bullo, F. (2010). On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms. *IEEE Transactions on Robotics*, 26(5):880–890.
- Nigam, N., Bieniawski, S., Kroo, I., e Vian, J. (2011). Control of multiple uavs for persistent surveillance: Algorithm and flight test results. *IEEE Transactions on Control Systems Technology*, 20(5):1236–1251.
- Hauert, S., Leven, S., Varga, M., Ruini, F., Cangelosi, A., e Nembrini, J. (2011). Reynolds flocking in reality with fixed-wing robots: Communication range vs. maximum turning rate. *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5015–5020.
- Wilcox, R. R. (2012). *Introduction to robust estimation and hypothesis testing*. Academic Press, 3 edition.
- Russell, S. e Norvig, P. (2013). *Artificial Intelligence: A Modern Approach*. Pearson, Upper Saddle River, NJ, 3 edition.
- Kroese, D. P., Taimre, T., e Botev, Z. I. (2014). *Handbook of Monte Carlo Methods*. John Wiley & Sons.
- Cengel, Y. A. e Cimbala, J. M. (2015). *Mecânica dos fluidos-3*. Amgh Editora.
- Ferri, G. et al. (2015). Multi-robot systems for environmental monitoring. *Marine Technology Society Journal*.
- Roldán, J. J., Garcia-Aunon, P., Garzón, M., de León, J., del Cerro, J., e Barrientos, A. (2016). Heterogeneous multi-robot system for mapping environmental variables of greenhouses. *Sensors*, 16(7):1018.
- Rubinstein, R. Y. e Kroese, D. P. (2016). *Simulation and the Monte Carlo Method*. John Wiley & Sons.
- Siciliano, B. e Khatib, O. (2016). *Springer Handbook of Robotics*. Springer.
- Othmani-Guibourg, M., Fallah-Seghrouchni, A. E., Farges, J.-L., e Potop-Butucaru, M. (2017). Multi-agent patrolling in dynamic environments. In *2017 IEEE International Conference on Agents (ICA)*, pages 72–77.

- Liu, J. e Li, W. (2020). Monte carlo simulation for environmental risk assessment: A review. *Environmental Modelling & Software*, 131:104780.
- Leite, G. R., Savino, H. J., Araújo, Í. B., Cordeiro, T. D., Ferreira, L. J., Cavalcante, G. H., Lima, A. M., e Lacroix, S. (2021). Framework for multiagent ocean monitoring: dynamic environment and reactive policy. In *OCEANS 2021: San Diego-Porto*, pages 1–4. IEEE.
- Leite, G. R., Costa, D. H. S., Souza, M. F. B., Roque, L. J., Vidal, C. A., e Cavalcante-Neto, J. B. (2021). Multiple quadrotor-uav autonomous mission for persistent surveillance with coverage path planning. *Expert Systems with Applications*, 185:115611.
- McCammon, S., dos Santos, G. M., Frantz, M., Welch, T. P., Best, G., Shearman, R. K., Nash, J. D., Barth, J. A., Adams, J. A., e Hollinger, G. A. (2021). Ocean front detection and tracking using a team of heterogeneous marine vehicles. *Journal of Field Robotics*, 38(6):854–881.
- Patrinopoulou, N., Daramouskas, I., Meimetis, D., Lappas, V., e Kostopoulos, V. (2022). A multi-agent system using decentralized decision-making techniques for area surveillance and intruder monitoring. *Drones*, 6(11).