

**INTELIGÊNCIA ARTIFICIAL**

**2024/2025**

**Professor**

**Paulo Oliveira**

**Eduardo Pires**

**RELATÓRIO**

aspiradores em netlogo

trabalho pratico 1 -> agentes

**79881 - David Fidalgo**

**78800 - Tiago Carvalho**

**2024/2025**

Conteúdo

[1.Introdução 3](#_Toc180237827)

[2. Robot1 3](#_Toc180237828)

[2.1. Adversidades ultrapassadas durante o projeto 5](#_Toc180237829)

[Robot 2 7](#_Toc180237830)

[Conclusão 8](#_Toc180237831)

[Bibliografia 8](#_Toc180237832)

# Introdução

A automação de tarefas de carácter diário tem evoluído significativamente com o uso de máquinas e robôs, especialmente em áreas que exigem contante manutenção, como a limpeza de espaços. Neste contexto, a simulação de sistemas multiagente surge como uma ferramenta eficaz para estudar e entender a interação entre diferentes agentes que atuam em um mesmo ambiente.

Este trabalho prático tem como objetivo desenvolver um modelo de simulação, utilizando o NetLogo, que reproduza o comportamento de agentes responsáveis pela limpeza e poluição de um espaço.

A proposta fornecida pelos docentes da cadeira de Inteligência Artificial baseia-se em um cenário simplificado onde um agente de limpeza interage com três agentes poluidores. A simulação visa representar o processo de degradação do espaço causado pelos Polluters e a restauração do espaço realizado pelo Cleaner.

A simulação, não só auxilia na análise das iterações entre agentes, mas também possibilita a análise e experimentação de várias estratégias para melhorar a eficiência do processo de limpeza em um ambiente poluído.

# Robot1

## Interface

Uma imagem com texto, software, Ícone de computador, Software de multimédia

Descrição gerada automaticamente

Figura 1-> Robot: Interface NETLOGO

* **Setup:** Reset ao mundo;
* **Go\_Once:** Os agentes efetuam um movimento;
* **Go\_N:** Os agentes efetuam N movimentos, sendo N adaptável na entrada;
* **Go:** Os agentes realizam de uma forma continua os seus movimentos;
* **Cleaner\_max\_battery:** Bateria máxima que o cleaner contém;
* **Cleaner\_max\_capacity:** Capacidade máxima que o cleaner contém;
* **Cleaner\_tempo\_carregamento:** Ticks necessários para carregar o cleaner;
* **Polluter\_1\_prob\_sujar, Polluter\_2\_prob\_sujar, Polluter\_3\_prob\_sujar:** A probabilidade de cada polluter de poluir;
* **Cleaner – Bateria Restante:** Bateria atual do cleaner;
* **Cleaner – Capacidade:** Capacidade atual do cleaner;
* **Num\_depositos:** Numero de depósitos existente no mundo;
* **Contaminação Vs Limpeza:** Grafico que mostra o evoluir a contaminação/limpeza do mundo. Exemplo:

Uma imagem com texto, diagrama, Gráfico, file

Descrição gerada automaticamente

Figura 2-> Robot 1: Gráfico Contaminação Vs Limpeza

## Adversidades ultrapassadas durante o projeto

1. Localização do Robot (Cleaner)

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente

Figura 3-> Robot 1: Algoritmo de movimentação cleaner

Para detetar a sua localização o robô recorre ao comando ***patch-here***, que devolve ao nosso agente *cleaner* (robô aspirador) o *patch* onde este se encontra.

Visto que esta abordagem obedece à regra de que o *cleaner* só pode ter perceção de apenas um *patch* à sua volta (**comando *neighbors***), pareceu-nos a mais adequada.

O nosso agente *cleaner* usa os seguintes comandos chave para determinar onde se encontra e quem são os seus vizinhos:

**Comandos chave:**

* patch-here
* patch-ahead
* patch-at-heading-and-distance
* neighbors

1. Localização dos Contentores:

* Numa fase inicial, o cleaner vai ter o modo mapeamento ativado, que serve para localizar todos os contentores existentes, que são todos os *patches-here* que se encontrem com o *pcolor = blue*;

**Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente**

Figura 4-> Robot 1: Função Modo\_Map

* Quando chega ao limite superior, ele começa o processo de limpeza;
* Quando o cleaner precisar de descarregar os detritos, ele calcula, na lista **depósitos**, procura as coordenadas mais próximas à localização atual do cleaner.

**Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente**

Figura 5-> Robot 1: Algoritmo de procura o deposito mais próximo

* Caso o utilizador queira mudar o número de contentores, o cleaner continua a mapear caso encontre um *patch* de cor *blue.* Caso o cleaner se desloque a um sítio que deveria conter um deposito e esse *patch* não tiver com a cor devida, essa coordenada é removida da lista de depósitos.

**Uma imagem com texto, captura de ecrã, Tipo de letra, file

Descrição gerada automaticamente**

Figura 6-> Robot 1: Remover deposito já não existente

1. Posto de Carregamento

* O posto de carregamento encontra se no canto inferior esquerdo;
* O Cleaner só se desloca para o posto de carregamento só quando faltar 50 movimentos;

**Uma imagem com texto, Tipo de letra, file, captura de ecrã

Descrição gerada automaticamente**

Figura 7-> Robot 1: Verificação e deslocação para posto de carregamento

* O Clenear, após o número de ticks de carregamento, volta para o local onde estava a limpar.

# Robot 2

## Interface

Uma imagem com texto, software, Software de multimédia, Ícone de computador

Descrição gerada automaticamente

Figura 8-> Robot 2: Interface

Nesta fase, mantivemos a base da fase transata, aplicando melhorias, aproximando o ambiente dos agentes o mais próximo da realidade.

* **Cleaner\_Modo:** O cleaner vai ter 3 modos de funcionamento: Eco Mode, Medium Mode e Full Mode;
* **Potencia Cleaner (em W):** Valor, em Watts, da Potencia do Cleaner;
* **Cleaner\_capacity\_battery:** Capacidade da bateria, em miliampere-hora, do Cleaner;
* **Cleaner\_tensao\_battery:** Tensão, em Volts, da bateria do Cleaner;
* **Consumo Cleaner:** Consumo, por tick, a cada movimento do Cleaner;
* **Cleaner\_max\_capacity (Alteração):** Capacidade máxima, em ml, do Cleaner;
* **Num\_obstaculos:** Numero de obstáculos do ambiente.

## Implementações para aproximação com a Realidade

1. **Inserção de modelo de bateria ao Cleaner**

**Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente**

Figura 9-> Robot 2: Função Config\_Battery

* Cada um dos modos vai ter diferentes valores de potência;
* Conforme o Modo do Cleaner, da capacidade da bateria e da tensão, é calculado o consumo da bateria por segundo (ou tick).

1. **Diferentes tons de poluição**

**Uma imagem com texto, software, Tipo de letra, captura de ecrã

Descrição gerada automaticamente**

Figura 10-> Robot 2: Comportamento Cleaner perante poluição

* Os *polluters* vão inserir ‘lixos’ de diferentes tons de uma determinada cor;
* O *Cleaner* vai ter diferentes comportamentos, dependendo do modo em que se encontrar:
  + **Full Mode:**
    - Neste modo, o *cleaner* vai apenas demorar 1 *tick* a limpar o *patch* que se encontra, com todo o tipo de lixo.
  + **Medium Mode:**
    - Os lixos considerados “leves” e “normais”, o *cleaner* só necessita de 1 *tick* para limpar o *patch* em questão;
    - Os lixos considerados “pesados”, o cleaner vai aumentar o valor do pcolor até o cleaner encontrar se em condições de limpar facilmente o lixo.
  + **Eco Mode:**
    - Os lixos considerados “leves”, o cleaner necessita de 1 tick para limpar o patch em questão;
    - Os livros considerados “normais” e “pesados”, o *cleaner* vai aumentar o valor do *pcolor* até o lixo que se encontrar no *patch* for considerado “leve”.
* Cada lixo tem diferente peso.

# Conclusão

# Bibliografia