



UNIVERSIDADE FEDERAL DO MARANHÃO  
Curso de Engenharia da Computação

Amanda Mikely Abreu Macedo  
Gabriela Torres de Queiroz  
Matheus Silva Cunha

**Relatório Técnico: Atividade Unidade I – Busca**

São Luís  
2025

Amanda Mikely Abreu Macedo  
Gabriela Torres de Queiroz  
Matheus Silva Cunha

## **Relatório Técnico: Atividade Unidade I – Busca**

Relatório Técnico apresentado ao professor Alex Oliveira Barradas Filho do curso de Engenharia da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção de nota parcial na primeira avaliação da disciplina Inteligência Artificial.

São Luís  
2025

# Sumário

<b>1 INTRODUÇÃO.....</b>	<b>3</b>
<b>1.1 Objetivo Geral .....</b>	<b>3</b>
<b>1.2 Objetivos Específicos .....</b>	<b>3</b>
<b>2 METODOLOGIA .....</b>	<b>4</b>
<b>2.1 Organização do Processo.....</b>	<b>4</b>
<b>2.2 Estrutura do Projeto .....</b>	<b>4</b>
<b>2.3 Modelagem em Grafo.....</b>	<b>4</b>
<b>2.4 Implementação do A* .....</b>	<b>4</b>
<b>2.5 Reprodutibilidade .....</b>	<b>5</b>
<b>3 RESULTADOS.....</b>	<b>5</b>
<b>3.1 Código.....</b>	<b>5</b>
<b>3.2 Execução da busca .....</b>	<b>5</b>
<b>4 DISCUSSÃO.....</b>	<b>6</b>
<b>5 CONCLUSÃO.....</b>	<b>6</b>
<b>REFERÊNCIAS.....</b>	<b>7</b>

# 1 Introdução

Este trabalho apresenta a solução da Atividade da Unidade I da disciplina de Inteligência Artificial, cujo objetivo é modelar um problema de busca como grafo e implementar o algoritmo A\*.

Para isso, um problema de navegação em labirinto foi modelado como um grafo, no qual os vértices representam células livres e as arestas representam conexões possíveis entre posições adjacentes. A solução integra modelagem, visualização e execução animada do A\*, utilizando Python, NetworkX e Matplotlib.

## 1.1 Objetivo Geral

Implementar um sistema completo de busca usando A\*, incluindo modelagem do ambiente como grafo e visualização da execução.

## 1.2 Objetivos Específicos

Destaca-se como objetivos específicos deste trabalho:

- Modelar um labirinto como grafo;
- Implementar o algoritmo A\*;
- Demonstrar visualmente a evolução da busca;
- Garantir reproduzibilidade e clareza do código;
- Produzir código reproduzível, documentado e modular;
- Gravar vídeo demonstrando a execução.

## 2 Metodologia

### 2.1 Organização do Processo

O desenvolvimento foi dividido em etapas:

#### 1. Geração do labirinto

Utilizando o algoritmo DFS recursivo para gerar caminhos aleatórios.

#### 2. Estruturação como grafo

Cada célula livre virou um nó, com arestas para vizinhos acessíveis.

#### 3. Implementação do A\*

Utilizando:

- fila de prioridade
- $g(n)$ ,  $h(n)$  e  $f(n)$
- reconstrução do caminho ao encontrar o objetivo

#### 4. Visualização simultânea

Três janelas:

- Labirinto explorado
- Árvore de busca
- Painel de ações

### 2.2 Estrutura do Projeto

Arquivo	Função
main.py	Execução da simulação
mazeGen.py	Geração do labirinto via DFS
agente.py	Implementação do A*
grafos.py	Construção do grafo
acoes.py	Visualização das ações relativas do movimento
requirements.txt	Dependências

### 2.3 Modelagem em Grafo

Cada célula livre do labirinto é um nó e cada conexão entre vizinhos compõe uma aresta. É construído um grafo completo (full\_grid\_graph) e uma árvore comprimida (display\_tree) para visualização.

### 2.4 Implementação do A\*

O A\* utiliza:

- open\_set (PriorityQueue)

- $g\_score$  (custo acumulado)
- heurística Manhattan
- $f\_score = g + h$

A busca continua até o objetivo ser encontrado ou não restarem nós.

## 2.5 Reprodutibilidade

Garantida por:

- seed fixa ou gerada e exibida
- versão definida de bibliotecas
- código modular
- execução única a partir de main.py
- dependências listadas em requirements.txt.

# 3 Resultados

## 3.1 Código

O código completo está disponível no repositório do GitHub:

<https://github.com/MSJCunha/Python-ObjAgent-Graphs-Maze/blob/main/main.py>

## 3.2 Execução da Busca

A simulação exibe:

- Labirinto com nós explorados.
- Caminho final do A\*.
- Grafo da busca.
- Ações do agente (esq/dir/retorno).

## 4 Discussão

A metodologia empregada demonstrou:

- A eficácia do A\* em ambientes estruturados
- A importância da função heurística na redução do número de nós explorados
- A utilidade de visualização para compreensão didática

A heurística Manhattan mostrou-se eficiente para grids ortogonais. A visualização facilita o entendimento da busca, porém labirintos maiores podem aumentar o custo computacional.

## 5 Conclusão

Todos os objetivos foram alcançados. A implementação do A\* foi bem-sucedida e a visualização gráfica torna o processo pedagógico.

Este trabalho permitiu:

- Compreender na prática o funcionamento do algoritmo A\*
- Modelar um ambiente real como grafo e navegar por ele
- Visualizar o processo de busca e o caminho encontrado
- Estruturar um sistema reproduzível, modular e de fácil análise

# Referências

- Russell, S.; Norvig, P. *Artificial Intelligence – A Modern Approach*.
- Documentation – NetworkX
- Documentation – Matplotlib