

# RELATÓRIO DE AVALIAÇÃO DE PERFORMANCE DOS ALGORITMOS DE BUSCA

---

## 1. Visão Geral

Foram avaliados A\*, Dijkstra, BFS, DFS e Greedy Best-First Search em matrizes 4x4, 16x16, 32x32 e 64x64. Métricas analisadas: custo do caminho, nós expandidos e tempo de execução.

---

## 2. Heurísticas – Impacto Direto nos Resultados

### A\*

- A heurística **reduziu significativamente a expansão de nós**.
- Manhattan consistentemente expandiu menos nós que Euclidiana em todos os tamanhos de grafo.
- Sem heurística, o comportamento converge para Dijkstra, aumentando custo computacional.

**Conclusão operacional:** heurística foi determinante para eficiência.

### Greedy

- Heurísticas garantiram caminhos rápidos em tempo, mas muitas vezes **não ótimos em custo**, especialmente em matrizes maiores.

**Conclusão:** heurística melhora tempo, mas sacrifica qualidade da solução.

### 3. Comparativo de Performance

#### 31 Performance Geral (Síntese)

Algoritmo	Qualidade do Caminho	Custo Computacional
A*	Ótimo	Excelente com heurística
Dijkstra	Ótimo	Alto custo em grafos grandes
BFS	Ótimo em matrizes simples	Escalabilidade fraca
DFS	Péssima performance global	Insustentável em matrizes grandes
Greedy	Bom tempo, custo ruim	Ideal quando "tempo > precisão"

#### 32 Melhor Performance Observada

- **Menor esforço computacional:**  
**Greedy Manhattan** → menor número de nós expandidos.
- **Melhor equilíbrio entre custo e esforço:**  
**A\*** com Manhattan.
- **Pior desempenho absoluto:**  
**DFS** → caminhos gigantes, expansão massiva, uso impraticável.

### 4. Impacto do Tamanho do Grafo

#### Comportamento observado

- Aumento de matriz → crescimento exponencial de nós expandidos em BFS, Dijkstra e DFS.
- A\* mantém crescimento controlado devido à heurística.
- Greedy cresce pouco em expansão, mas gera soluções progressivamente piores.

#### Tabela comparativa (Nós Expandidos)

Tamanho	A* (Manhattan)	Dijkstra	BFS	DFS
4x4	16	40	25	61
16x16	264	512	481	4081
32x32	949	2082	1985	32737
64x64	(incompleto, mas tendência confirma escalonamento)			

### Conclusão executiva:

A\* escala melhor. BFS, Dijkstra e especialmente DFS tornam-se inviáveis em matrizes grandes.

## 5. Gráficos (Representação Conceitual)

### Evolução dos Nós Expandidos



### Tempo de Execução (tendência)

Pequenas matrizes → tempo irrelevante para todos  
 Matrizes grandes \* DFS > Dijkstra > BFS > A\* > Greedy

## 6. Padrões Identificados

- Heurística forte = redução direta na carga computacional.
- Manhattan supera Euclidiana na maioria dos casos.
- Greedy entrega SLA de tempo, sacrificando governança de qualidade.
- DFS é operacionalmente inviável para aplicações de produção.
- Dijkstra só faz sentido quando heurística não pode ser aplicada.

---

## 7. Conclusão Corporativa

- **A\*** é o algoritmo mais eficiente para ambientes produtivos em qualquer escala.
- **Greedy** ideal para pipelines onde latência tem prioridade sobre precisão.
- **Dijkstra, BFS e DFS** apresentam escalabilidade ruim, recomendados apenas para uso acadêmico, simulações controladas ou validação.