

# Relatório — 8-Puzzle

Projeto de Implementação e Comparação de Algoritmos de Busca

Julio Cesar Thurow Buzzi   Arthur Henrique Tristão Pinto  
Luis Henrique Ferreira Costa   Filipi Pereira de Mesquita Faria

GitHub: [JulioBuzzi/8puzzle](https://github.com/JulioBuzzi/8puzzle)

## Introdução

Este relatório apresenta uma visão geral do problema do *8-puzzle* e descreve a implementação desenvolvida utilizando a linguagem Dart no framework Flutter, que serviu como base para a criação de uma interface interativa e responsiva para execução dos algoritmos. O objetivo do projeto foi comparar o desempenho de três algoritmos de busca — A\*, BFS e DFS — bem como analisar o impacto de três heurísticas distintas empregadas no algoritmo A\*. A partir desses experimentos, busca-se compreender como cada abordagem se comporta em termos de eficiência, custo computacional e qualidade das soluções produzidas.

## Algoritmos Utilizados

- A\* — Busca informada que combina custo real (g) e estimativa heurística (h). Retorna solução ótima quando a heurística é admissível.
- BFS (Busca em Largura) — Expande nós por nível; garante solução de menor número de movimentos, mas pode explorar muitos estados.
- DFS (Busca em Profundidade) — Explora caminhos até o fim antes de retroceder; pode não encontrar solução em limites práticos.

## Heurísticas (A\*)

As heurísticas usadas no A\* e a justificativa resumida:

- Distância de Manhattan — rápida, informativa e consistente para o 8-puzzle; é a escolha padrão por equilibrar custo e precisão.
- Misplaced Tiles (tiles fora do lugar) — muito simples; mostra como uma heurística fraca impacta negativamente a busca.
- Nilsson Sequence Score — adiciona penalidade por quebra de sequência; tende a ser mais seletiva em certos casos.

## Resultados Experimentais

Algoritmo / Heurística	Solução	Tempo (ms)	Nós Expandidos	Profundidade / Caminho (passos)
A* (Manhattan)	✓	125	696	21 passos
A* (Misplaced)	✓	871	4457	21 passos
A* (Nilsson)	✓	318	1599	23 passos
BFS	✓	4194	61075	21 passos
DFS	✓	297	56569	51 passos

## Análise Comparativa

Resumo das observações que apareceram nos experimentos:

- Tempo: em geral, A\* com Manhattan tende a ser o mais rápido entre as opções testadas.
- Nós expandidos: A\* (Manhattan) costuma visitar menos nós que A\* (Misplaced) e que as buscas cegas (BFS/DFS).
- Heurísticas: heurísticas mais informativas (Manhattan, Nilsson) reduzem significativamente o espaço de busca; Misplaced é a mais fraca.
- Caminho: todos os algoritmos que encontram solução normalmente apresentam caminhos de comprimentos semelhantes; BFS garante a menor profundidade quando usado sem limitações.

## Conclusão

De forma prática: para o 8-puzzle, A\* com a heurística de Manhattan oferece o melhor balanço entre velocidade e número de nós expandidos. Heurísticas fracas aumentam custo computacional. Entre as buscas cegas, BFS encontra soluções ótimas mas com custo muito maior.