

**Érika Mara de Moraes Machado**  
**Maria Luiza Bernardo Madeira**

**Matrícula: 202310058**  
**Matrícula: 202310981**

## RELATÓRIO TÉCNICO - BUSCA E JOGO

Foram desenvolvidos dois algoritmos de busca aplicados à resolução do jogo dos oito (8-puzzle): a Busca em Largura (Breadth-First Search – BFS) e o Algoritmo A\*. Ambos foram implementados em Python, com o objetivo de comparar o desempenho entre uma abordagem não informada, que explora o espaço de estados de forma sistemática, e uma informada, que utiliza heurísticas para orientar a busca de forma mais eficiente.

No algoritmo Busca em Largura, a lógica baseia-se em explorar todos os estados possíveis do tabuleiro de maneira uniforme, sem utilizar qualquer informação adicional sobre a proximidade do objetivo. Para isso, é utilizada uma fila (estrutura FIFO) que armazena os estados em ordem de descoberta. A cada iteração, o algoritmo retira o primeiro estado da fila, verifica se ele corresponde ao estado objetivo e, caso negativo, gera todos os vizinhos possíveis a partir da movimentação do espaço vazio. Esses novos estados são adicionados ao final da fila, garantindo que os estados mais antigos sejam explorados primeiro. Essa estratégia assegura que a primeira solução encontrada seja a de menor profundidade, ou seja, com o menor número de movimentos. Entretanto, como o BFS precisa armazenar uma grande quantidade de estados simultaneamente, seu custo de memória cresce exponencialmente, tornando-o pouco eficiente em instâncias mais complexas do problema.

Já o Algoritmo A\* foi desenvolvido como uma busca informada, incorporando uma função heurística para estimar a distância restante até o estado objetivo. O A\* combina o custo do caminho percorrido ( $g$ ) com o custo estimado até o objetivo ( $h$ ), formando a função  $f = g + h$ . Nesta implementação, foi utilizada a distância de Manhattan como heurística, que calcula a soma das distâncias horizontais e verticais de cada peça em relação à sua posição correta. Essa heurística é admissível, ou seja, nunca superestima o custo real, o que garante que o algoritmo sempre encontre a solução ótima. Diferentemente do BFS, o A\* utiliza uma fila de prioridade (heap), onde os estados com menor valor de  $f$  são explorados primeiro, priorizando os caminhos mais promissores.

Em termos de desempenho, a Busca em Largura e o A\* apresentam tempos e números de passos semelhantes em estados simples, como nos níveis fáceis do jogo, pois o espaço de busca é reduzido. Entretanto, conforme a complexidade do estado inicial aumenta, o A\* demonstra maior eficiência, resolvendo o problema com menos expansões de nós e menor tempo de execução. Isso ocorre porque a heurística de Manhattan direciona a busca, evitando explorar caminhos desnecessários que o BFS percorre obrigatoriamente.

Na prática, a Busca em Largura se destaca pela simplicidade e pela garantia de encontrar a solução mais curta, sendo útil para problemas menores ou como base de comparação. Por outro lado, o Algoritmo A\* apresenta-se como uma solução mais sofisticada e escalável, equilibrando custo e desempenho ao utilizar informação heurística para guiar a exploração. Assim, o estudo comparativo demonstra claramente a diferença entre abordagens cegas e informadas, evidenciando como o uso de heurísticas pode reduzir drasticamente o esforço computacional em problemas de busca e planejamento.