

# Aplicação do Algoritmo A\* para rotas do CEDETEG

Nomes: Iuri Pedroso e  
Arthur Carboni

## RESUMO

Este artigo é uma aplicação que calcula os melhores caminhos entre diferentes pontos baseado no mapa geral do CEDETEG. O código foi feito na linguagem de programação Java Script, a partir da aplicação do algoritmo A\*, com intuito de apresentação na matéria de Inteligência Artificial e Computacional I.

## 1. INTRODUÇÃO

O mapa do CEDETEG é composto por diferentes lugares, sendo eles: salas, blocos, laboratórios, campos, e até mesmo muitos lagos. Todos eles sendo ligados por caminhos diferentes, o que leva o estudante a refletir sobre qual percurso fazer para chegar de um ponto A a um B dentro desse mapa. Assim o Algoritmo apresentado neste documento, faz uma relação entre esses percursos, calculando as melhores rotas entre elas.

Nesse contexto, o trabalho propõe a aplicação do algoritmo A\* para solucionar problemas de pathfinding (encontrar o melhor trajeto entre dois pontos). O A\* é amplamente utilizado em sistemas de navegação, jogos e robótica devido à sua eficiência e garantia de encontrar o menor caminho quando certas condições são atendidas (RUSSELL; NORVIG, 2016).

## 2. FUNDAMENTAÇÃO TEÓRICA DO ALGORITMO A\*

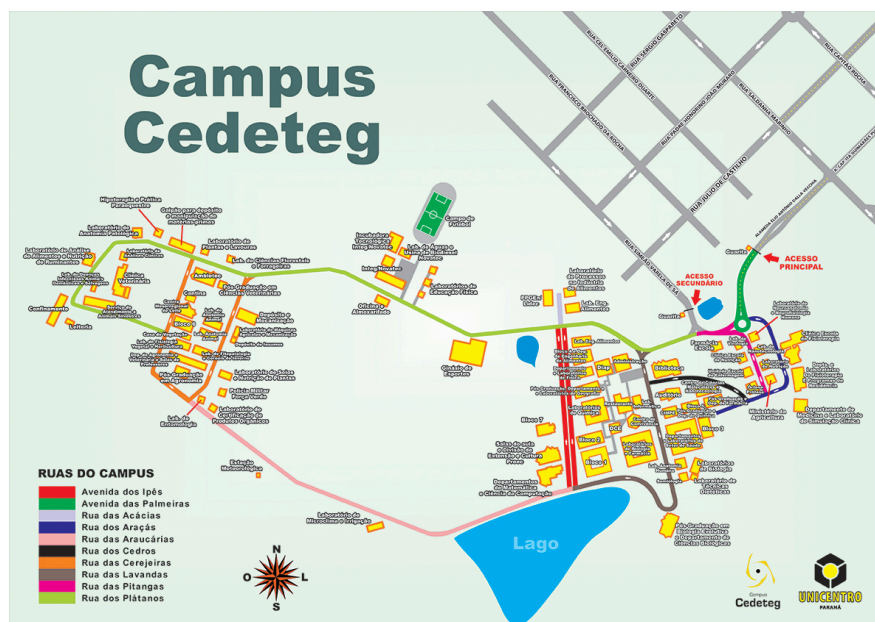
O algoritmo A\* é um método de busca heurística que combina o custo real do caminho percorrido ( $g(n)$ ) com uma estimativa do custo restante até o destino ( $h(n)$ ), conhecida como heurística (HART et al., 1968). Sua fórmula de avaliação é dada por:

$$f(n)=g(n)+h(n)$$

Onde:

- $g(n)$  é definido por: Custo acumulado do nó inicial até o nó atual.
- $h(n)$  é definido por: Heurística (estimativa do custo do nó atual até o objetivo).

Para garantir que o A\* encontre o caminho ótimo, a heurística deve ser admissível (nunca superestimar o custo real) e consistente (monotônica) (PEARL, 1984). No contexto do algoritmo produzido a heurística foi utilizada com base nas distâncias tiradas do mapa central do CEDETEG conforme mostra a figura 1:



Fonte: <https://www3.unicentro.br>

Apesar do algoritmo não ter todos os pontos representados, ele utiliza locais chaves do mapa para que o aluno siga na direção mais próxima de um desses pontos e se localize da melhor maneira possível.

Comparado a outros algoritmos como Dijkstra (que não usa heurística) ou Busca em Largura, o A\* é mais eficiente em cenários onde uma boa heurística pode direcionar a busca (LAVALLE, 2006).

### 3. IMPLEMENTAÇÃO NO CEDETEG

#### 3.1 Modelagem do grafo

Durante a produção do algoritmo, foi utilizado o mapa do CEDETEG como um grafo ponderado, onde os nós são pontos de interesses e as arestas são os caminhos entre os nós.

Exemplo de estrutura (em JavaScript):

```
const grafo = {  
  "acesso principal": [{ para: "farmacia escola", custo:  
63.98 }],  
  "biblioteca": [{ para: "auditorio", custo: 35.00 }],  
  // ... outros nós  
};
```

#### 3.2 Heurística adotada

A heurística ( $h(n)$ ) foi definida com base em distâncias estimadas até o destino principal ("decomp"). Por exemplo:

```
const heuristica = {  
  "acesso principal": 309.71,  
  "biblioteca": 250,  
  // ... outros valores  
};
```

#### 3.3 Funcionamento do algoritmo

Visando um algoritmo didático e com bom desempenho, os seguintes passos foram utilizados para a implementação do mesmo:

##### 3.3.1 Inicialização:

- Criação de duas listas, sendo uma aberta (open list), para armazenar os nós a serem explorados, e outra fechada (closed

list), para armazenar os nós que já foram visitados anteriormente.

- O nó inicial foi adicionado à lista aberta com um valor de  $f(n) = g(n) + h(n)$ , onde:
- $g(n)$  é o custo acumulado (iniciado em 0).
- $h(n)$  é a heurística estimada até o destino (utilizando a distância em linha reta entre os pontos como argumento).

### 3.3.2 Exploração:

- Enquanto a lista aberta não estiver vazia, o nó com menor valor  $f(n)$  é selecionado para expansão.
- Para cada nó vizinho, o custo acumulado  $g(n)$  é calculado tendo como base a soma do custo do caminho percorrido até o nó atual adicionado do custo da aresta até seu vizinho.
- Caso o vizinho não esteja na lista aberta ou fechada, ou se um caminho aprimorado for encontrado, o vizinho é adicionado à lista aberta com seu novo valor  $f(n)$ .

### 3.3.3 Critério de parada e finalização:

- O algoritmo acaba quando o nó de interesse é alcançado ou quando a lista aberta se esvazia (o que indica que não há caminho possível para o ponto de destino).
- O caminho de melhor otimização é reconstruído retroativamente a partir de um nó de interesse até o inicial.

## 4. RESULTADOS E DISCUSSÃO

Os testes realizados utilizando o algoritmo A\* no contexto do Campus Cedeteg demonstraram resultados significativos pela sua maestria em eficiência. Nesta pequena, porém significativa amostra de seu potencial, o algoritmo foi capaz de calcular a rota ótima entre o “Acesso Principal” e o “Decomp: Departamento de Matemática e Ciência da Computação”, proporcionando um caminho otimizado e curto em comparação à outras rotas para o mesmo destino.

A heurística adotada, inspirada nas distâncias do mapa, demonstrou-se eficaz, conforme previsto na literatura (Hart et al., 1968). Destaca-se que a precisão do algoritmo depende diretamente da qualidade e

precisão dos dados de entradas utilizados como base para o processamento, podendo influenciar nos resultados de forma crítica.

Por fim, a aplicação A\* no Campus Cedeteg comprovou-se viável e útil, especialmente para auxiliar principalmente visitantes e estudantes que estão iniciando sua formação. Em síntese, o estudo além de validar a eficácia do algoritmo em um contexto real, também destacou uma oportunidade e potencial de contribuição social desta tecnologia, com enfoque para usuários em fase de adaptação ao ambiente universitário.

## **5. CONCLUSÃO**

Este trabalho foi feito tanto para demonstração de viabilidade do algoritmo A\* quanto para a prática dos nossos ensinamentos durante a disciplina de Inteligência Artificial até agora. Assim, o nosso algoritmo teve como prioridade a otimização de rotas do CEDETEG e a redução da complexidade de navegação do campus. Por fim esta implementação não apenas comprova a eficácia do algoritmo A\*, mas também abre portas para novos projetos que envolvam o assunto mencionado.

## **REFERÊNCIAS**

- HART, P. E.; NILSSON, N. J.; RAPHAEL, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. \*IEEE Transactions on Systems Science and Cybernetics\*, 1968.
- PEARL, J. \*Heuristics: Intelligent Search Strategies for Computer Problem Solving\*. Addison-Wesley, 1984.
- RUSSELL, S.; NORVIG, P. \*Artificial Intelligence: A Modern Approach\*. 3ª ed. Pearson, 2016.
- LAVALLE, S. M. \*Planning Algorithms\*. Cambridge University Press, 2006.