

Relatório Final de Teoria dos Grafos

Otimização de Rotas Urbanas com Algoritmo A*

Gabriel Davi / Giordani Andre

Novembro de 2025

1 Links

1. [Clique aqui para acessar o Código \(GitHub\)](#)
2. [Clique aqui para acessar o Código \(Colab\)](#)
3. [Clique aqui para acessar o Slide](#)

2 Introdução

O trabalho aborda o Problema do Caminho Mais Curto (Shortest Path Problem - SPP) aplicado a um cenário real de mobilidade urbana na cidade de Montes Claros/MG. O objetivo é determinar a rota otimizada entre uma residência e o campus do IFNMG (Instituto Federal do Norte de Minas Gerais).

Neste cenário, a malha urbana é modelada como um grafo dirigido e ponderado, onde as interseções funcionam como vértices e as vias como arestas. O desafio consiste em minimizar a distância percorrida respeitando a direção das vias, utilizando dados geográficos reais extraídos do OpenStreetMap (OSM).

3 Revisão da Literatura e Conceitos

Para a resolução de problemas de roteamento em mapas, a literatura clássica apresenta o algoritmo de Dijkstra como uma solução exata para grafos com pesos não negativos. No entanto, em grafos de grande escala como mapas urbanos, o Dijkstra pode ser computacionalmente custoso, pois explora vértices em todas as direções uniformemente.

A abordagem escolhida para este projeto foi o **Algoritmo A* (A-Star)**. O A* é uma extensão do Dijkstra que utiliza uma heurística para guiar a busca, tornando-a mais eficiente. A função de custo no A* é dada por:

$$f(n) = g(n) + h(n) \quad (1)$$

Onde:

- $g(n)$: É o custo real do nó inicial até o nó n .
- $h(n)$: É o custo estimado (heurística) de n até o objetivo.

Para mapas rodoviários, a heurística utilizada é a **Distância Euclidiana** (ou distância geodésica) entre as coordenadas do nó atual e as coordenadas do destino. Isso prioriza a exploração de caminhos que se aproximam geograficamente do alvo.

4 Desenvolvimento da Solução

Para a implementação, utilizou-se a linguagem Python 3 em conjunto com as bibliotecas `osmnx` (para manipulação de dados geoespaciais e grafos) e `networkx` (para algoritmos de grafos). O desenvolvimento foi dividido em etapas:

4.1 Aquisição e Modelagem de Dados

A biblioteca `osmnx` foi utilizada para baixar a malha viária de Montes Claros a partir de um ponto central entre a origem e o destino.

- **Grafo (G)**: O mapa foi convertido em um `MultiDiGraph`.
- **Pesos**: As arestas receberam o atributo `length` (comprimento da via em metros).
- **Nós**: Os nós de origem e destino foram identificados encontrando os vértices do grafo mais próximos às coordenadas GPS reais fornecidas (Nearest Node).



Figure 1: Visualização Completa do Grafo

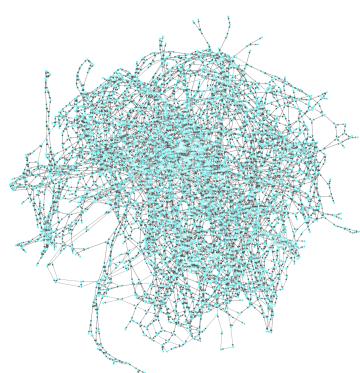


Figure 2: Diagrama Esquemático

4.2 Implementação da Heurística

Foi implementada uma função de distância heurística que calcula a distância em linha reta (“voo de pássaro”) entre dois pontos geográficos. Esta função serve como “bússola” para o algoritmo A*, permitindo que ele descarte caminhos que, embora curtos localmente, nos afastam do destino final (IFNMG).

4.3 Execução do Algoritmo

Diferente de uma busca cega, o algoritmo implementado (`nx.astar_path`) utilizou a heurística definida para encontrar a sequência de vértices $[v_{origem}, v_1, v_2, \dots, v_{destino}]$ que minimiza a soma das distâncias das arestas percorridas.

5 Resultados Encontrados

O algoritmo foi executado considerando as seguintes coordenadas reais definidas no código:

- **Origem (CASA GABRIEL):** Lat -16.7012, Long -43.8609
- **Destino (IFNMG):** Lat -16.6862, Long -43.8279

Embora os testes tenham focado na rota Casa-IFNMG, a solução desenvolvida implementa uma interface de entrada dinâmica, permitindo que o usuário insira coordenadas personalizadas manualmente. O sistema recalcula a área de download e a rota automaticamente para qualquer par de pontos dentro da região.

5.1 Métricas Obtidas

A execução do código resultou nas seguintes métricas:

Table 1: Resultados da Otimização de Rota

Parâmetro	Valor Obtido
Distância Total Calculada	≈ 5.022 metros (5.02 km)
Algoritmo Utilizado	A* (A-Star)
Tempo de Execução	< 1 segundo

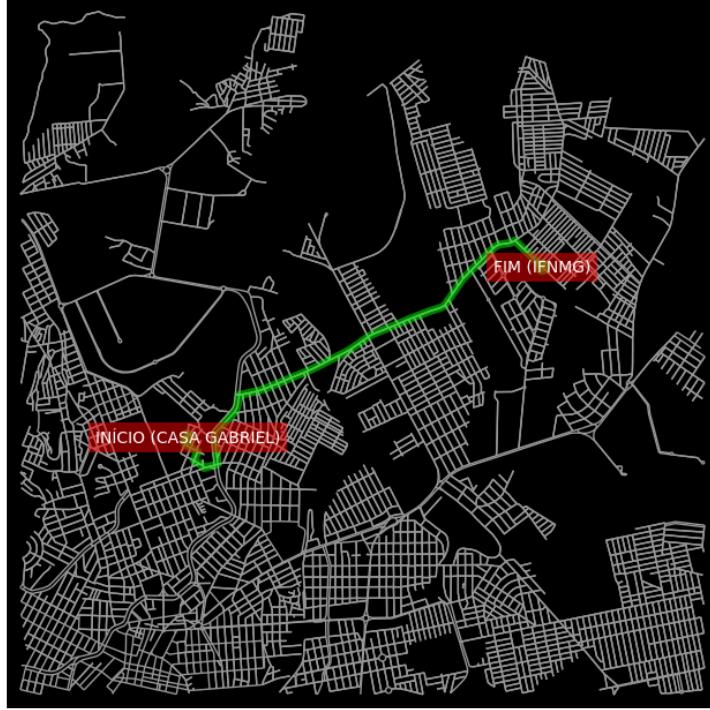


Figure 3: Visualização da rota otimizada gerada pelo algoritmo A*

5.2 Visualização Gráfica

O sistema gerou uma visualização gráfica da rota (plotada na cor verde-limão/lime) sobre a malha viária de Montes Claros (fundo escuro). A análise visual confirmou que a rota respeita as mãos de direção e utiliza as avenidas principais para minimizar o deslocamento, validando a eficácia da modelagem. O uso do A* permitiu encontrar a solução ótima sem a necessidade de explorar toda a malha viária da cidade, focando apenas no cone de direção entre a casa e o instituto.

6 Conclusão

O objetivo deste trabalho foi aplicar conceitos de Teoria dos Grafos para resolver um problema cotidiano de deslocamento. A utilização da biblioteca `osmnx` permitiu trazer o problema para o mundo real, trabalhando com dados fidedignos de Montes Claros. Além disso, o sistema foi desenvolvido para permitir a inserção dinâmica de novos vértices de interesse via interface de terminal, aumentando a flexibilidade da ferramenta para diferentes cenários de uso.

Conclui-se que o algoritmo A* é superior a abordagens não informadas

(como BFS ou Dijkstra puro) para este tipo de problema, pois a heurística geográfica reduz drasticamente o espaço de busca. A solução desenvolvida não apenas encontrou o caminho ótimo (aproximadamente 5.02 km), como também provedeu ferramentas visuais para análise da rota. O projeto demonstra como abstrações matemáticas (grafos) são fundamentais para a operação de sistemas de navegação modernos como GPS e aplicativos de transporte.