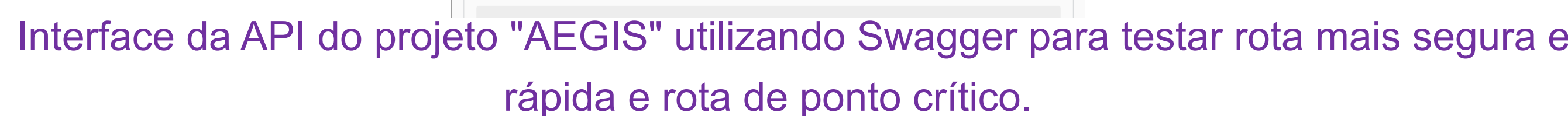


Amanda Duarte Meneghini do Carmo; Andy Hyong Tae Choi Youn; Renato Gonçalves Machado; Sedra Alhendi.

Ciência da Computação - Campus Mooca - bruno.mulina@animaeducacao.com.br

O transporte de valores requer planejamento rigoroso, já que envolve grandes quantias de dinheiro e bens de alto risco. Veículos Especiais de Transporte de Valores dependem de rotas que reduzam ameaças, encurtem o deslocamento e atendam às exigências operacionais das empresas. A análise manual desses trajetos tende a ser lenta e sujeita a falhas, o que motivou a criação de ferramentas computacionais capazes de apoiar esse processo.

O uso de um MinHeap otimizou o desempenho do Dijkstra, mantendo tempos de processamento baixos mesmo em grafos densos. O sistema apresentou maior precisão e menor subjetividade na avaliação de risco, confirmando a viabilidade de soluções computacionais para apoiar decisões operacionais no transporte de valores.



Este estudo teve como objetivo desenvolver um software capaz de analisar a malha urbana de forma estruturada, utilizando distância e nível de risco como parâmetros para selecionar trajetos adequados. A proposta buscou integrar estruturas de dados e algoritmos clássicos de análise de rotas, permitindo uma avaliação conjunta de eficiência e segurança. O trabalho se destaca por aplicar métodos consolidados da computação em um problema real, oferecendo uma solução de apoio à tomada de decisão em um setor sensível.

A pesquisa foi conduzida por meio da construção de um sistema baseado na modelagem da malha urbana como um grafo. Cada cruzamento foi tratado como um nó, e cada rua como uma aresta com peso. Esses pesos foram definidos com base em duas variáveis: distância física e nível de risco associado à região. Os dados foram previamente catalogados para permitir a simulação das rotas. Para armazenar as relações entre os nós, utilizou-se uma LinkedList, escolhida por sua eficiência na manipulação de elementos adjacentes. O programa também fez uso de pilhas para detectar a presença de ciclos, informação útil para a identificação de regiões com possíveis entradas e saídas repetidas. Essa característica ajudou a avaliar vulnerabilidades operacionais. O MinHeap foi utilizado na priorização de caminhos durante a execução do algoritmo de Dijkstra, garantindo menor custo computacional na busca pelos trajetos de menor peso. O algoritmo de Tarjan foi aplicado para reconhecer componentes fortemente conectados, o que permitiu localizar áreas críticas em termos de circulação e segurança. A implementação foi desenvolvida em Java e validada com dados simulados que representavam uma malha urbana típica de operações de transporte de valores. Os testes foram repetidos até a estabilidade dos resultados e a consistência da rota sugerida pelo sistema.

A aplicação demonstrou eficiência na análise de rotas seguras ao integrar estruturas de dados adequadas com algoritmos clássicos de grafos. O Dijkstra calculou trajetos que equilibram distância e risco, identificando alternativas sempre que o caminho mais curto envolvia áreas de risco. A aplicação do algoritmo de Tarjan permitiu localizar regiões urbanas com alto potencial estratégico, como áreas de retorno recorrente.

O desenvolvimento do sistema demonstrou que a modelagem por grafos, combinada ao uso de estruturas de dados e algoritmos apropriados, é eficaz na análise de rotas para veículos de transporte de valores. A abordagem adotada permitiu identificar trajetos que equilibram segurança e eficiência, além de revelar áreas críticas da malha urbana. O uso de LinkedList, pilhas e MinHeap mostrou-se adequado para organizar os dados e agilizar os cálculos. Os algoritmos de Dijkstra e Tarjan contribuíram para a identificação de caminhos seguros e para a compreensão estrutural do mapa analisado. Os resultados obtidos atendem aos objetivos da pesquisa e podem ser aplicados na rotina operacional das empresas do setor. Apesar disso, as conclusões se referem ao conjunto de dados utilizados nos testes, e estudos futuros podem ampliar a aplicação do sistema para malhas urbanas reais e ambientes dinâmicos.

W3SCHOOLS. **DSA Linked Lists**. Disponível em:
<https://www.w3schools.com/dsa/dsa_theory_linkedlists.php>. Acesso em: 17 nov. 2025.

W3SCHOOLS. **DSA Graphs**. Disponível em:
<https://www.w3schools.com/dsa/dsa_theory_graphs.php>. Acesso em: 17 nov. 2025.

W3SCHOOLS. **DSA Stacks**. Disponível em:
<https://www.w3schools.com/dsa/dsa_data_stacks.php>. Acesso em: 17 nov. 2025.

W3SCHOOLS. **DSA Dijkstra's Algorithm**. Disponível em:
<https://www.w3schools.com/dsa/dsa_algo_graphs_dijkstra.php>. Acesso em: 17 nov. 2025.

MENEGHINI, Amanda; Hyong, Andy;. **GitHub - AmandaMeneghini/aegis-core-logic: Base project for the “Aegis” system. This repository contains custom data structures (LinkedList, Stack, MinHeap) and core graph algorithms (Dijkstra, Tarjan’s) built from scratch in pure Java. This project serves as the foundational logic engine for a future Spring Boot API**. Disponível em: <<https://github.com/AmandaMeneghini/aegis-core-logic>>. Acesso em: 17 nov. 2025.

Agradecemos aos professores Bruno Henrique Oliveira Mulina e Otacilio José Pereira por toda orientação e contribuição ao longo do desenvolvimento deste trabalho.