Algoritmo de Dijkstra Aplicado ao Problema de Locomoção Dos Usuários que Utilizam o Transporte Público em Eventos

DAVID XAVIER BRITO, JOÃO BATISTA BORGES JUNIOR

Ciência da Computação – Universidade Federal do Tocantins (UFT) – Palmas– To– Brasil

{juniorbatistaborges, david.151xavier}@gmail.com,

Abstract. In this paper we present an application of the Dijkstra algorithm in the solution of a classic problem of calculating the shortest path between Pali bus shelters and the consequences of a restrictive or erroneous survey of the variables involved in the problem. To view the results of the calculations, an open source javascript software has been developed that performs the calculation of the route and shows the best route.

Resumo. *Neste artigo é apresentada uma aplicação do algoritmo de Dijkstra na solução de um problema clássico de cálculo do trajeto mais curto entre abrigos de onibûs de Palmas as consequências de um levantamento restritivo ou errôneo das variáveis envolvidas no problema. Para visualizar os resultados dos cálculos foi desenvolvido um software de código-aberto em javascript que executa o cálculo da rota e exibe o melhor trajeto.*

1. Introdução

Atualmente o sistemas de distribuição torna-se algo evidente gerando grande impactos nos respectivos custo é na qualidade do serviço oferecido pelas empresas. existe uma variedade de problemas nessa área, em diferentes níveis estratégicos e táticos. Empresas buscam algoritmos que são utilizados na solução de problemas complexos que normalmente é diferente da oferecida pela

programação convencional. segundo DOWSLAND (1982). As últimas décadas vêm sendo marcadas por um crescente interesse na aplicação de técnicas de pesquisa operacional com vistas à resolução de problemas da cadeia logística.

Muitas aplicações do mundo real requerem o auxílio destes algoritmos completos. Neves (2007) cita vários exemplos de aplicações que utilizam estes algoritmos como, por exemplo, o roteamento de veículos em um sistema de transporte, que se integram a sistemas de GPS (*Global Positioning System*), computadores portáteis e celulares. Estes sistemas pertencem ao modelo dos Sistemas de informações Geográficas (SIG). Outra aplicação deste tipo de sistema e que utiliza o algoritmo de Dijkstra pode ser obtida em Derekenaris et al. (2000) onde os autores propõem um sistema de gerenciamento de ambulâncias e emergências baseado em SIG, GPS e GSM (*Global System for Mobile Communication)*. Outros exemplos clássicos são a transmissão de pacotes em redes de computadores e projetos de circuitos integrados em que os componentes precisam ser ligados por um menor caminho possível fazendo assim o custo da empresa ser maior, diminuindo seus gastos, consequentemente gerando maior lucro para uma empresa.

Tendo em vista que a cidade de Palmas oferece uma grande quantidade de eventos para seus moradores, muitos deles utilizam o transporte público para locomover-se, Um exemplo seria o festival gastronômico de Taquaruçu ele ocorre todo ano contendo várias atrações, atraindo a presença de milhares de pessoas tornando a locomoção difícil. para solucionar o problema seria definir pontos fixos com uma rota (expressa) aumentando consideravelmente a velocidade de chegada ao destino, atendendo com melhor qualidade os seus usuários. .

Partindo desta motivação, este trabalho visa ilustrar a aplicação do algoritmo de Dijkstra na solução de um problema clássico de cálculo do trajeto mais curto entre abrigos, bem como as consequências de uma definição errônea das variáveis envolvidas no problema. Como metodologia, foi desenvolvido um software que executa o cálculo das rotas e exibe o trajeto em um mapa. Este software é de código-aberto, está disponível gratuitamente para download. A sua utilização não fica restrita apenas a este trabalho e pode ser aplicada em outros fins.

2. Revisão bibliográfica

2.1. Algoritmo de Dijkstra

Existem vários algoritmos que podem ser usados para resolver o problema de caminhos mínimos. Um dos mais antigos e fundamentais é conhecido como algoritmo de Dijkstra desenvolvido Edsger Wybe Dijkstra e publicado em 1959. Sua utilização se deve a não necessidade de uma busca exaustiva em todos os caminhos, característica que permite um aumento da eficiência da busca.

2.2. Descrição do Algoritmo

Segundo Dijkstra (1959), para encontrar o menor caminho entre dois nós P e Q, usa-se o fato de que, se R é um nó no caminho mínimo entre P e Q, conhecendo o último, consequentemente conhece-se o caminho de P a R. Os caminhos mínimos de P até outros nós são construídos de forma incremental até que se atinja Q.

Dijkstra dividiu os nós em três conjuntos: A, B e C. No primeiro, estão os nós para os quais já se conhece a menor distância entre eles e o nó de origem (P). No segundo, os nós de onde será retirado o próximo nó a ser adicionado a A (nós que estão conectados a algum nó do conjunto A). No conjunto C estão os nós remanescentes (não conectados a nenhum nó do conjunto A).

As arestas também foram divididas em três grupos: I, II e III. No conjunto I, estão as arestas que fazem parte do menor caminho do nó P aos nós do conjunto A. O conjunto II, possui as próximas arestas que serão selecionadas para o conjunto I, ou seja, uma e somente uma aresta deste conjunto leva a cada nó do conjunto B. O terceiro conjunto possui as arestas remanescentes (rejeitadas ou ainda não consideradas).

O algoritmo inicia com todos os nós e arestas nos conjuntos C e III, respectivamente e adiciona-se o nó P ao conjunto A. Considera-se então o último nó adicionado ao conjunto A e realizam-se repetidamente dois passos a partir dele.

No primeiro passo, consideram-se todas as arestas r que conectam o nó P com nós R dos conjuntos B ou C. Se um nó R pertence ao conjunto B, é investigado se o uso da aresta r correspondente proporciona uma menor distância de P até R do que as arestas já conhecidas do conjunto II. Em caso negativo, a aresta r é rejeitada. Caso positivo, r substitui a aresta correspondente no conjunto II. Por outro lado, se um nó R pertence ao conjunto C, ele é adicionado ao conjunto B e a aresta r ao conjunto II.

No segundo passo, cada nó do conjunto B pode ser conectado ao nó P de somente uma maneira, se forem consideradas apenas as arestas dos conjuntos I e II. Assim, cada nó do conjunto B, possui uma distância ao nó P. O nó com menor distância é então transferido de B para A e a aresta correspondente é transferida de II para I. Retorna-se então ao primeiro passo e repete-se o processo até que o nó Q seja transferido para o conjunto A. Dessa forma, a solução é então obtida.

3. Materiais e Métodos

O problema abordado neste trabalho consiste em analisar o menor caminho de um ponto x e y de abrigos de palmas. No total foram 25 abrigos sendo eles: SANTA_BARBARA, 1306_SUL, 203_N, AURENY_III, 305_NORTE, 1202_SUL, 106_SUL, 409_NORT, 411_NORTE, 305_SUL, BERTAVILLE, TAQUARALTO, TAQUARUCU, BURITIRANA, 206_SUL, 306_SUL, 104_NORTE, 104_SUL, ESTACAO_APINAJE, ESTACAO_XAMBIOA, ESTACAO_KRAHO, ESTACAO_XERENTE, ESTACAO_KARAJA, ESTACAO_JAVAE e TAQUARI. Como Apresentada na Figura 3.1.

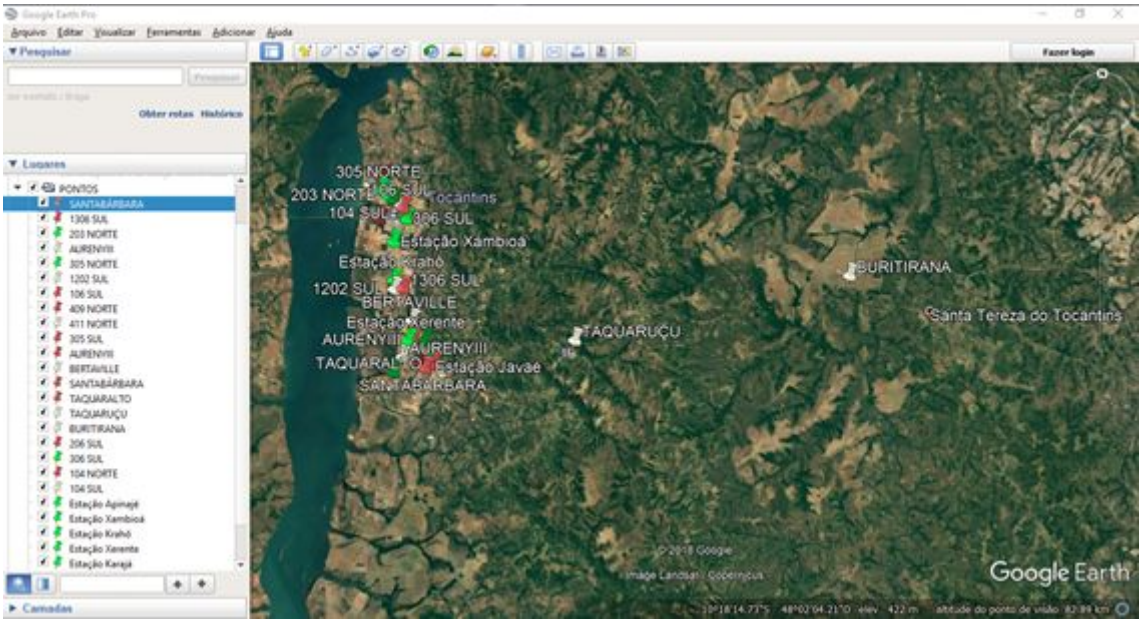


Figura 3.1 – Localização dos Abrigos de Onibûs

Esses abrigos foram coletado a localização real já implantada na capital de Palmas-TO e as rotas foram definidas utilizando o programa Google Earth. A distância foi definida como prioridade o menor caminho do abrigo X ao Y. Como Apresentada na Figura 3.2.

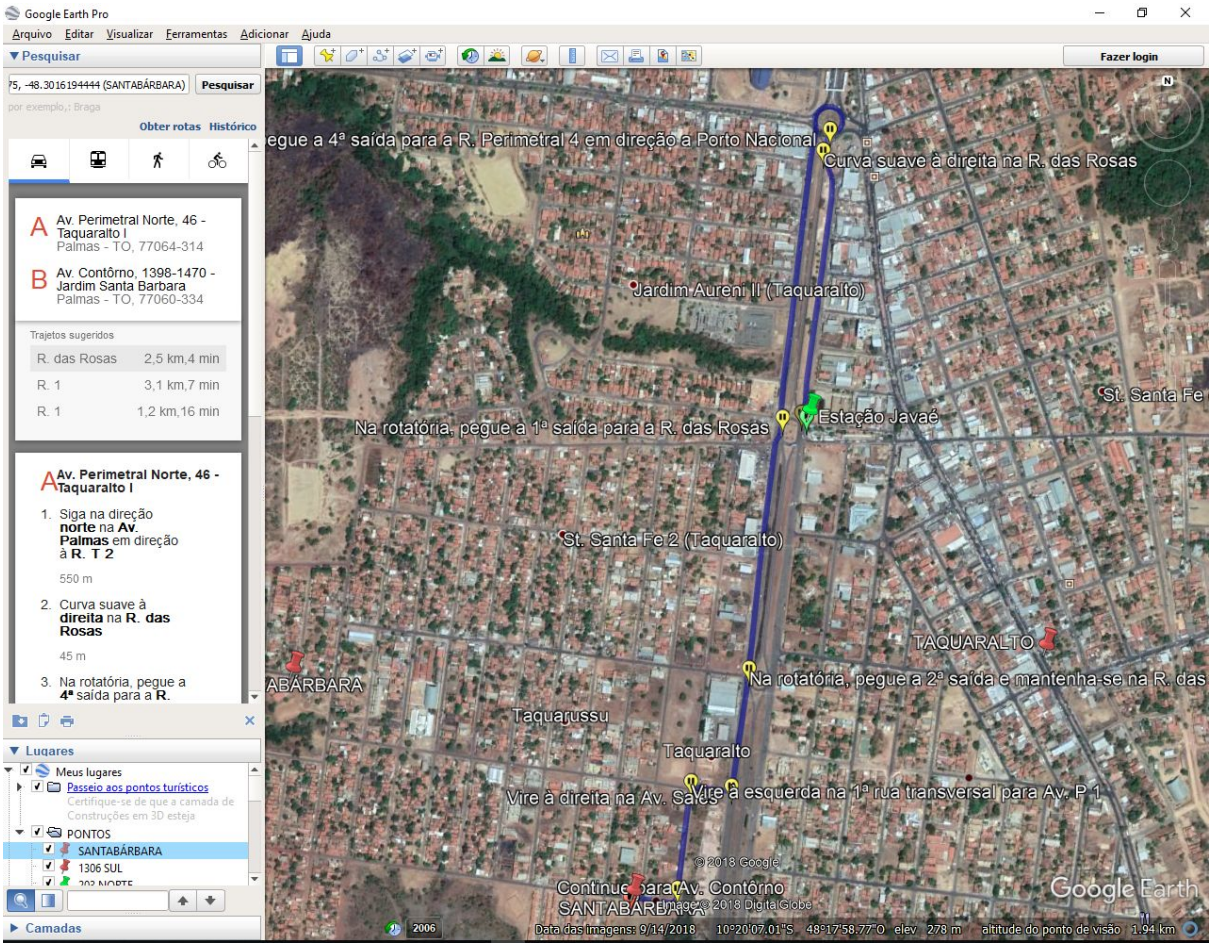
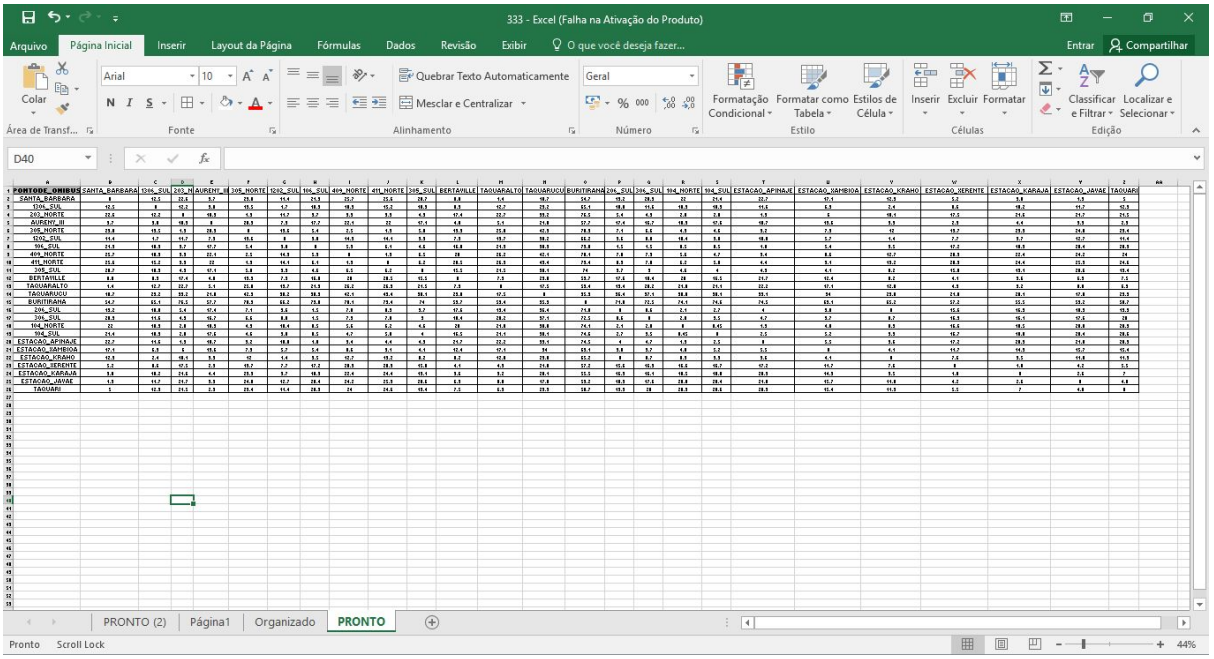


Figura 3.2 – Demonstração da Captura da Distância entre Abrigos

Logo após definir os pontos foi feito uma matriz ligando todos os abrigos contendo suas respectivas distância, com auxílio do programa excel para organizar todas as informações capturadas.Como Apresentada na Figura 3.3



.**Figura 3.3 – Ligações dos Abrigos**

Com a matriz montada foi convertido esses dados para um array aplicando ao software previamente criado contendo campos: Abrigo de Origem e Abrigo de Destino.Como Apresentada na Figura 3.4.

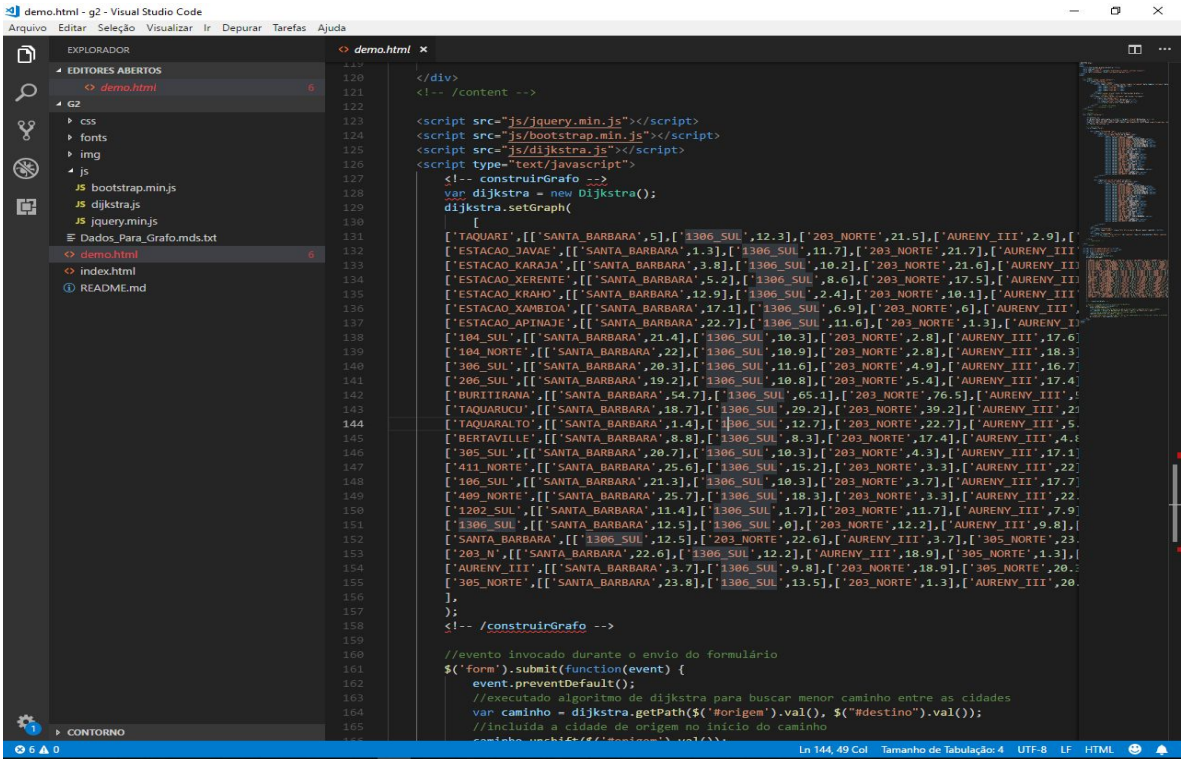


Figura 3.4 – Informações dos Abrigos

O layout e aplicação em se foram desenvolvida utilizando tais tecnologia (HTML, CSS, JavaScript, JQuery, BootStrap e XML), contendo um layout composto por uma interface que permite ao usuário configurar parâmetros para execução. Estes parâmetros são Abrigo de Origem, Abrigo Destino, contendo um botão de Busca menor caminho e um campo que apresentar o resultado da busca. Na Figura 3.5, é apresentado a interface do programa.

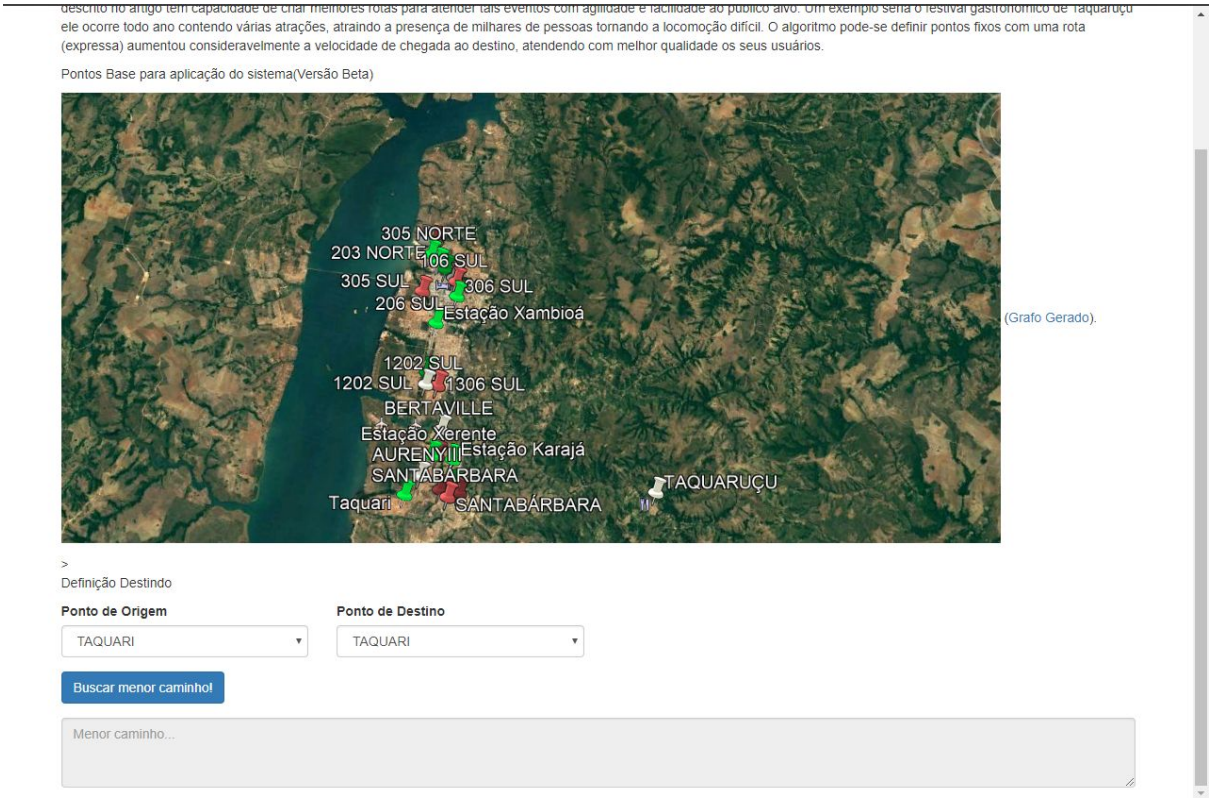


Figura 3.5 – layout do Usuário

Na implementação do algoritmo de Dijkstra, foi considerada apenas as variáveis Distância e abrigos. O software calcula o trajeto mais curto. Fornece as instruções ao usuário. No campo que apresentar o resultado mostra quais abrigos terá que passar até chegar no objetivo final.

Uma versão compilada do software e código fonte do projeto desenvolvido no ambiente integrado de desenvolvimento (IDE) Visual Code está disponível na plataforma do github (https://github.com/David15117/trabalho_grafos).

Para a construção do Grafo utilizamos uma aplicação ja criada (http://viz-js.com/) e reutilizamos a matrix criado no excel após converter as informações na forma que o software solicita foi obtida o grafo contendo todas ligações com as respectivas distâncias localizada nas arestas. Apresentada na Figura 3.6.

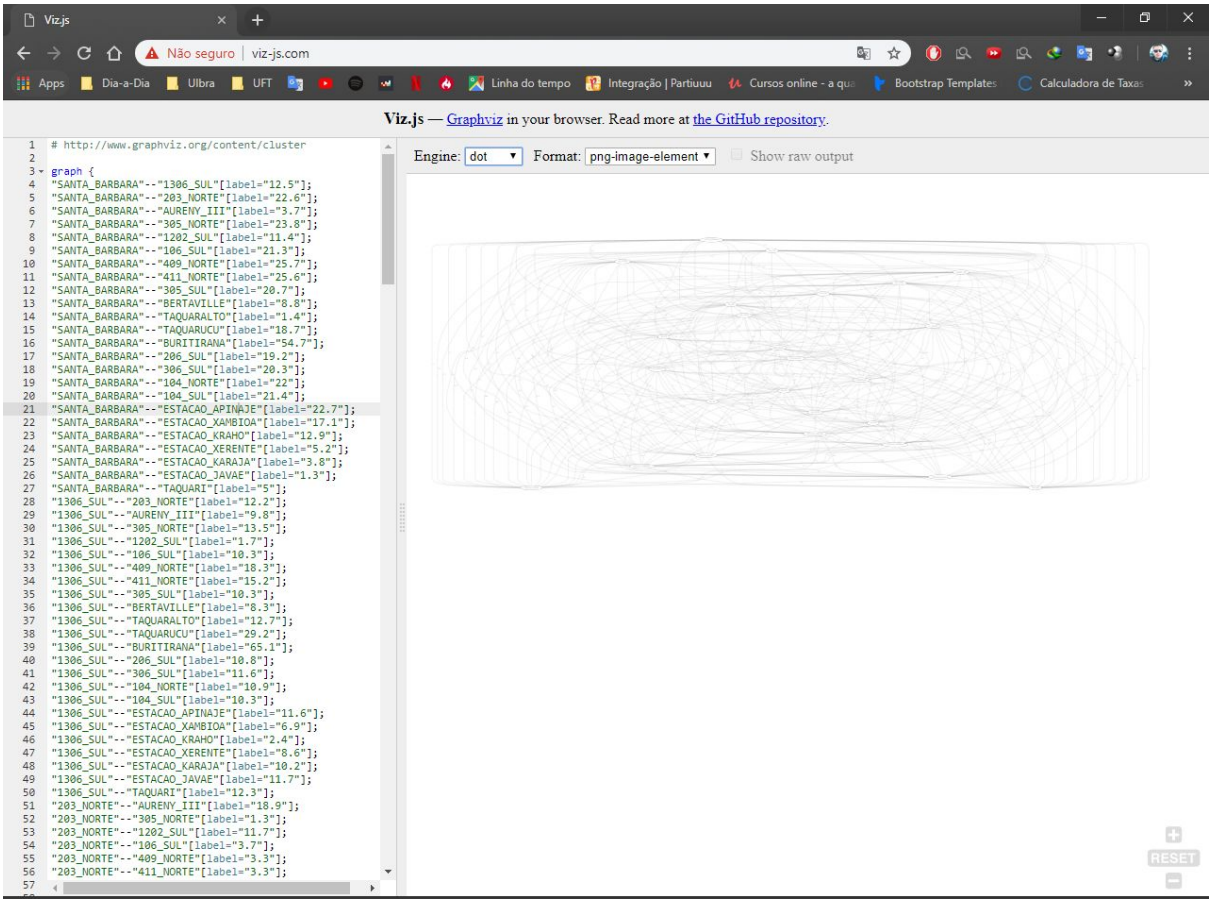


Figura 3.6 – Grafo dos Abrigos

4. Resultados e Discussões

fato de considerar apenas a distância como variável no algoritmo de Dijkstra, o cálculo do trajeto pode não apresentar o resultado do percurso mais rápido e sim do mais curto. Por exemplo, considere o ponto de partida o abrigo taquari com objetivo de chegada no abrigo taquaruçu conforme mostrada na figura 4.1.

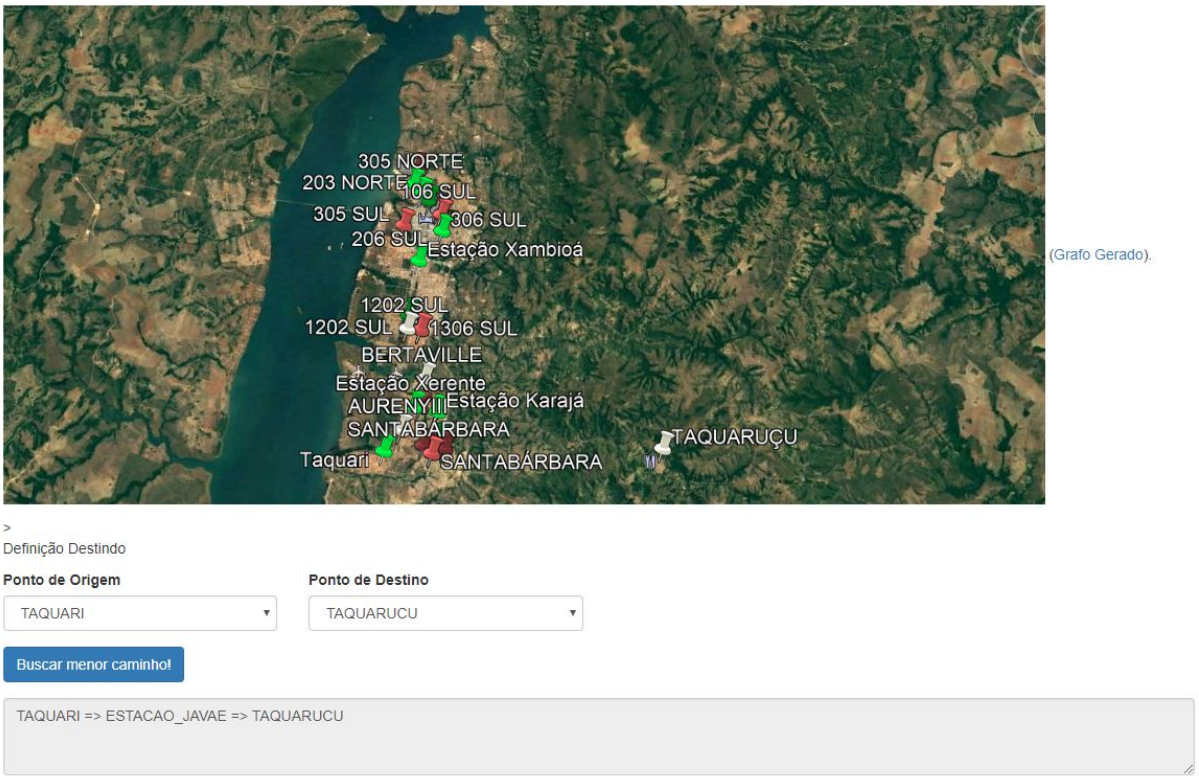


Figura 4.1 – Exemplo de um percurso

Considerando a Figura 4.1, percebe-se que o algoritmo conclui que o menor caminho e percorrer do abrigo taquari para o abrigo estão javaé para que depois seguir para o abrigo taquaruçu.

Assim, percebe-se que o levantamento das variáveis relevantes ao sistema foi realizado de forma muito restritiva. Como conseqüência, os resultados nem sempre oferecem a melhor alternativa ao passageiro. Do ponto de vista pedagógico, a ferramenta de software desenvolvida mostra-se muito útil, pois, permite ao usuário a visualização do funcionamento prático do algoritmo de Dijkstra, onde o usuário pode selecionar parâmetros ou modificar a codificação de forma a experimentar novos comportamentos no cálculo dos trajetos. A união entre os conceitos teóricos e práticos apresenta-se como uma alternativa essencial na evolução do processo ensino-aprendizagem.

6. Conclusão

O problema de encontrar caminhos mínimos em grafos se revela presente e de extrema importante em diversas áreas de conhecimento. Em contrapartida, estudar o seu funcionamento e características através da teoria pode, em muitos casos, não ser uma tarefa trivial.

Unir teoria e prática, em geral, proporciona uma evolução mais rápida e menos onerosa do processo ensino-aprendizagem. Dessa forma, este trabalho proporcionou uma aplicação prática do algoritmo de Dijkstra, bem como, uma análise da importância do levantamento adequado das variáveis relevantes ao problema em questão. Como legado, é deixado um software código-aberto, que permite ao usuário a manipulação do algoritmo de Dijkstra e pode ser utilizado para fins didáticos como, por exemplo, o ensino de algoritmos de busca.

7. Referências

Edsger W. Dijkstra. (1959) A note on two problems in connexion with graphs.

Numerische Mathematik, 1:269-271.

Neves, Patricia Takaki. (2007) Variações e Aplicações do Algoritmo de Dijkstra.

Campinas, [S.P. : s.n]. Dissertação de mestrado.

Ravindra K. Ahuja, Kurt Mehlhorn, James Orlin, and Robert E. Tarjan. (1990). Faster algorithms for the shortest path problem. J. ACM, 37(2):213-223.

Russel, Stuart Jonathan.; Norvig, Peter. (2004) Inteligência artificial. 2. ed.; Rio de Janeiro: Elsevier.

Tenenbaum, Aaron M.; Langsam, Yedidyah; Augenstein, Moshe J. (1995) Estruturas de dados usando C. São Paulo: Pearson Makron Books.

Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. (1999) Introduction to Algorithms. MIT Press Publ, Cambridge Mass, 22 edition

.