Relatório de Laboratório 1

Guilherme Philippi

10 de junho de 2019

Sumário

1 Introdução

Este relatório tem como objetivo apresentar soluções de exercícios envolvendo *grafos*, que foi um tema estudado por diversos matemáticos proeminentes. Pode-se dizer que em 1736 é que a teoria teve início, com base no artigo publicado por Leonhard Euler, sobre as 7 pontes de Königsberg [1]. Existe uma íntima relação entre Grafos e algoritmos [2]. Podemos, por exemplo, representar um algoritmo por um grafo. Nos apegaremos a este conceito. Assim, segue a definição:

Definição 1.1 *Um grafo G é um par ordenado da forma $(V(G), E(G))$, composto por um conjunto de vértices $V(G)$, de arestas $E(G)$ e uma função de incidência ψ_g que, por sua vez, associa a cada aresta de $V(G)$ um par não ordenado de vértices (nem sempre distintos) de $E(G)$. É dito que as arestas ligam os vértices, bem como denomina-se também tais vértices como extremidades desta aresta.*

Utilizando esta definição, pode-se denominar outras características das estruturas de um grafo. Assim, por exemplo, estudamos: *ordem, tamanho, incidência, adjacência, vizinhança, laço* e *elo*. Assim, os separamos por tipos, como grafo: *finito, nulo, simples, pseudografo, completo, conexo* e *bipartido*. Dentre estas, frisamos algumas definições a seguir.

1.1 Classificações

Definição 1.2 *Um grafo que não possui laços (arestas que possuem o mesmo vértice como extremidades),bem como não possui arestas paralelas (arestas cujas extremidades são os mesmo vértices) e onde quaisquer dois vértices são ligados por uma aresta é denominado grafo completo.*

Definição 1.3 *Grafo conexo é o nome dado para o tipo de grafo onde é possível estabelecer um caminho de qualquer vértice para qualquer outro vértice dele. Caso contrário, temos um grafo desconexo.*

Definição 1.4 *Chamamos de grafo bipartido o grafo onde pode-se particionar o conjunto de vértices em dois, de modo que cada aresta possua extremidades em ambos os conjuntos.*

Em especial, também podemos discutir *caminhos* (tanto eulerianos quanto hamiltonianos), *cliques, ciclos, árvores, dígrafos* e, por fim, *grafos ponderados.* Aqui, é importante destacar a definição de *grafo ponderado*, pois podemos representar os esquemas de distâncias (muito utilizado em modelagem na engenharia) com base neles.

Definição 1.5 *Chamamos de grafo ponderado o grafo que possui valores numéricos atribuídos as suas arestas.*

Após isto, partimos para o estudo de *matrizes de incidência e adjacência.*

1.2 Grafos como matrizes

Seguem algumas definições importantes.

Definição 1.6 *As extremidades de uma aresta são denominadas incidentes à aresta, bem como esta aresta é incidente às extremidades.*

Definição 1.7 *Vértices distintos e incidentes à uma mesma aresta são denominados adjacentes. Nesse sentido, arestas que possuem um vértice em comum são adjacentes também.*

Definição 1.8 *Dois vértices distintos e adjacentes são denominados vizinhos. O conjuntos de vértices vizinhos geralmente é denotado por $N_G(v)$.*

Assim, a partir de matrizes, é possível definir outras maneiras de representar um grafo, donde

Matriz de Incidência: Cada coluna representa os vértices incidentes a determinada aresta. Assim, cada linha representa as arestas incidentes a cada vértice. Contamos 1 para aresta incidente ao vértice (ou vice-versa) e 2 se tivermos um laço. Observe a matriz de incidência de G_1 à seguir:

$$\begin{array}{cc} \begin{matrix} f & g & h & i & j & k \end{matrix} & \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 2 & 0 \end{pmatrix} & \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} \end{array}$$

Matriz de Adjacência: É uma tabela onde observa-se somente a relação de adjacência entre vértices. Caso um vértice seja adjacente a outro, bota-se número 1. No caso de um laço, bota-se 2. Observe a matriz de adjacência de G_1:

$$\begin{array}{cc} \begin{array}{ccccc} A & B & C & D & E \end{array} & \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix} & \begin{array}{c} A \\ B \\ C \\ D \\ E \end{array} \end{array}$$

Por último, temos alguns resultados sobre *grau de um vértice*. Assim, seguem outras definições, com a prova de 2 resultados na sequência.

Definição 1.9 *O grau de um vértice é o número de arestas incidentes a ele. Cada laço conta como duas arestas. O vértice com grau zero é chamado de vértice isolado.*

- $d_G(v)$: É o grau do vértice v. Se G é um grafo simples (sem laços ou arestas paralelas), então $d_G(v)$ denota o número de vizinhos de v em G.

Teorema 1.1 *Para todo grafo $G(V(G), E(G))$, com m arestas, vale:*

$$\sum_{v \in V} d_G(v) = 2m.$$

Considere a matriz de incidência M do grafo G. Para saber o grau de um determinado vértice basta olharmos para sua linha correspondente e somarmos os valores contidos na linha. Note que somar os valores presentes em todas as linhas é o mesmo que somar os valores presentes nas colunas. Como temos m colunas e cada aresta possui 2 vértices incidentes, segue que $\sum_{v \in V} d_G(v) = 2m$.

Teorema 1.2 *Em qualquer grafo, o número de vértices, cujo grau é ímpar, é par.*

Suponha que o número de vértices com grau ímpar seja ímpar. Note que a soma do grau de todos os vértices resulta em um número ímpar(resultado aritmético). Somando este resultado com o grau dos demais vértices, cujo grau é par, obtemos um número ímpar. Absurdo, pois $\sum_{v \in V} d(v) = 2m$. Portanto, o número de vértices, cujo grau é ímpar, é par.

Iremos apresentar alguns problemas-exemplos da utilização de grafos, implementando algorítimos bem conhecidos na literatura. Utilizou-se o software GRAFOS [3] para modelar e simular as soluções propostas, assim como segue na próxima seção.

2 Desenvolvimento

2.1 Distribuição de Sensores

Em uma casa que busca ser automatizada (vide Figura 1), realizou-se uma proposta do cabeamento utilizando a menor quantidade possível de matéria prima

(Cabos) levando em consideração que todas as tomadas de ar mostradas na planta (pontos em vermelho) deveriam ser visitadas. Considerou-se que todos os vértices do grafo que define o sistema podem ter ligações entre si de maneira linear.

Figura 1: Distribuição dos sensores

Para solucionar este problema, com o auxilio do software GRAFOS, utilizou-se a Figura 1 como parâmetro, uma vez que a mesma está em perfeita escala, possibilitando-se que se calculasse as ponderações das arestas como as distâncias euclidianas dos pontos na imagem. Com isso, gerou-se o grafo completo mostrado na Figura 2.

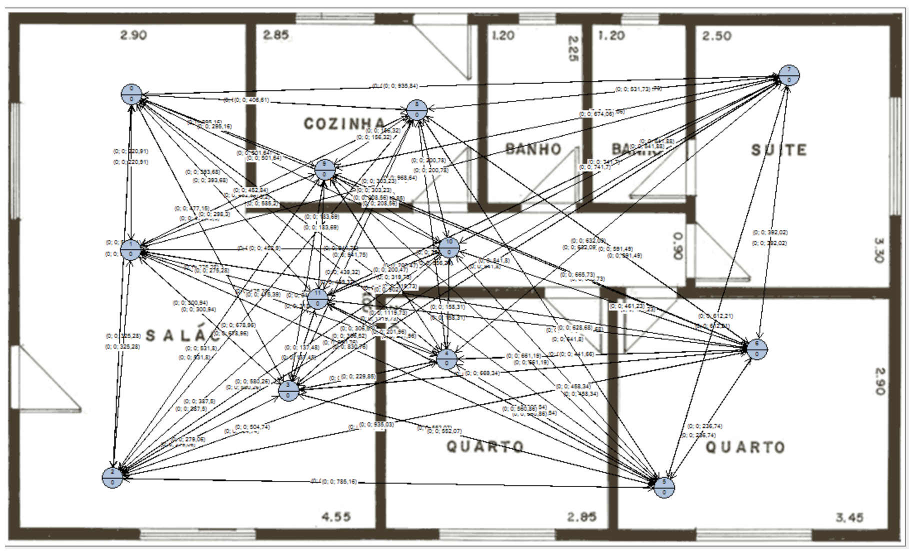

Figura 2: Grafo completo dos sensores

Com isso, pode-se utilizar o algorítimo de Kruskal, que busca uma árvore geradora mínima para um grafo conexo com pesos [4], ou seja, ele encontra um subconjunto das arestas que formam uma árvore incluindo todos os vértices onde o peso total desses, dado pela soma dos pesos das arestas da árvore, é mínimo. Observe o resultado obtido na Figura 3

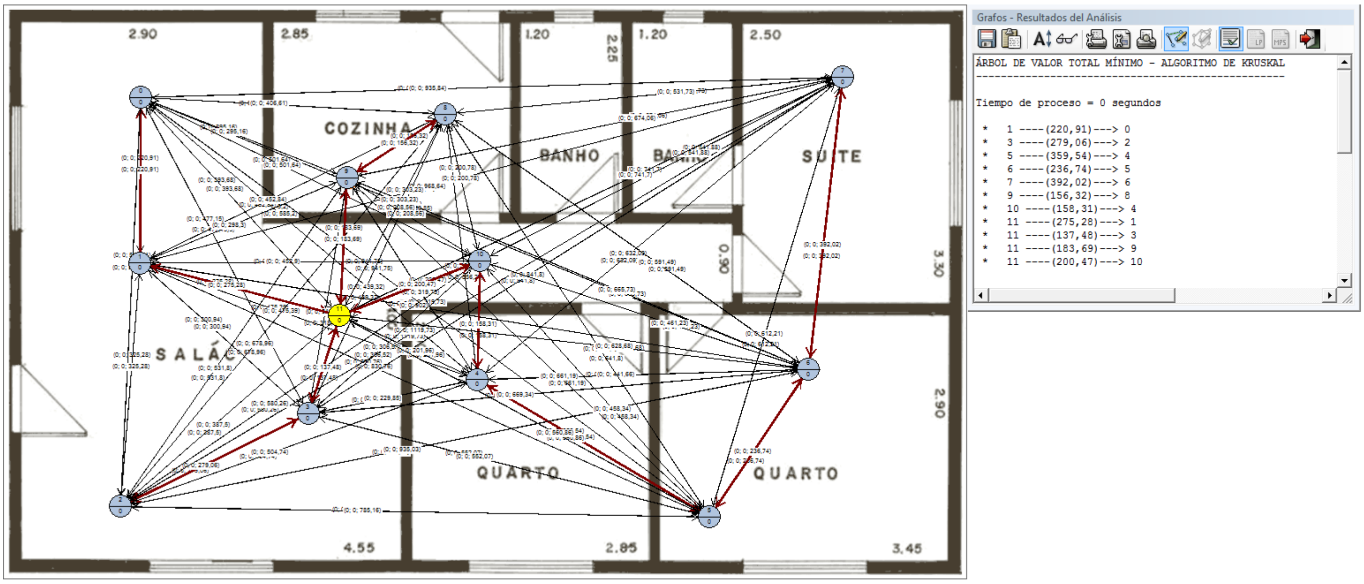

Figura 3: Aplicação do algorítimo de Kruskal no grafo completo

2.2 Sistema de Posicionamento Global

Outro excelente problema-exemplo envolvendo grafos está em obter o menor caminho entre dois pontos obedecendo um sistema de rotas. Aqui teremos como objetivo construir um GPS primitivo que permita calcular as menores rotas entre o ponto 2 e os demais pontos marcados no mapa da Figura 4.

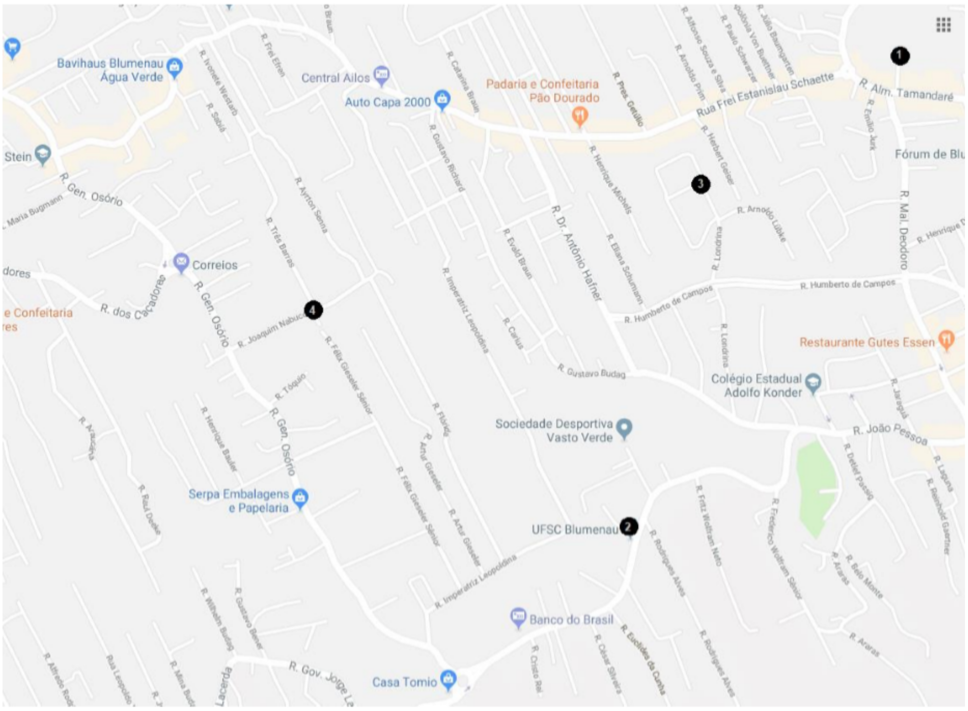

Figura 4: Mapa de rotas possíveis entre os pontos

Caminho do ponto 2 até 1

Semelhante a Figura 1, o mapa ilustrado na Figura 4 também está em perfeita escala, nos permitindo utilizá-lo como parâmetro para o software GRAFOS. Assim foi feito, obtendo os possíveis caminhos representados pelo grafo ponderado da Figura 5 (note que somente os caminhos validos estão presentes, ou seja, não fora levado em consideração os caminhos demasiadamente longos).

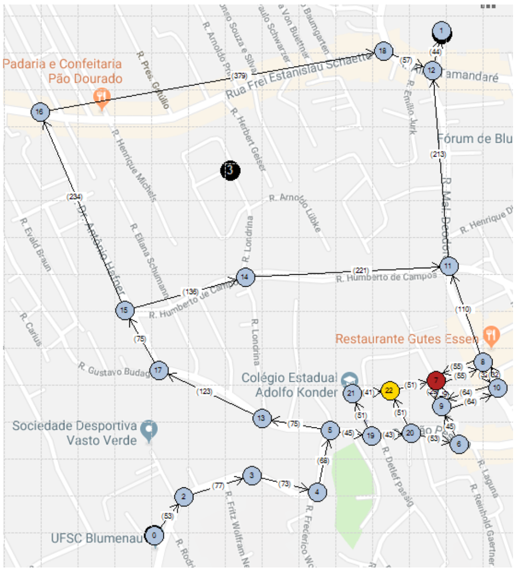

Figura 5: Mapa de rotas possíveis entre os pontos 2 e 1

Com este grafo montado, podemos utilizar o algorítimo de Dijkstra, concebido pelo cientista da computação holandês Edsger Dijkstra em 1956 e publicado em 1959 [5], que soluciona o problema do caminho mais curto num grafo dirigido (ou não) com arestas ponderadas não negativas, em tempo computacional $O(m+n\log n)$ onde m é o número de arestas e n é o número de vértices. Assim sendo, obtivemos o caminho representado pela Figura 6

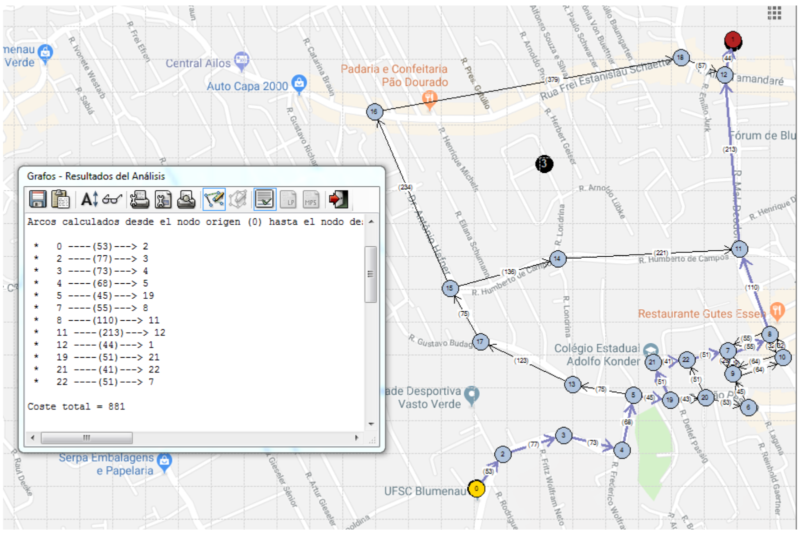

Figura 6: Mapa de rota mais curta entre os pontos 2 e 1

Caminho do ponto 2 até 3

Da mesma forma do destino anterior, construiu-se o grafo que melhor representa as possíveis rotas que ligam os pontos 2 e 3, como pode-se ver na Figura 7.

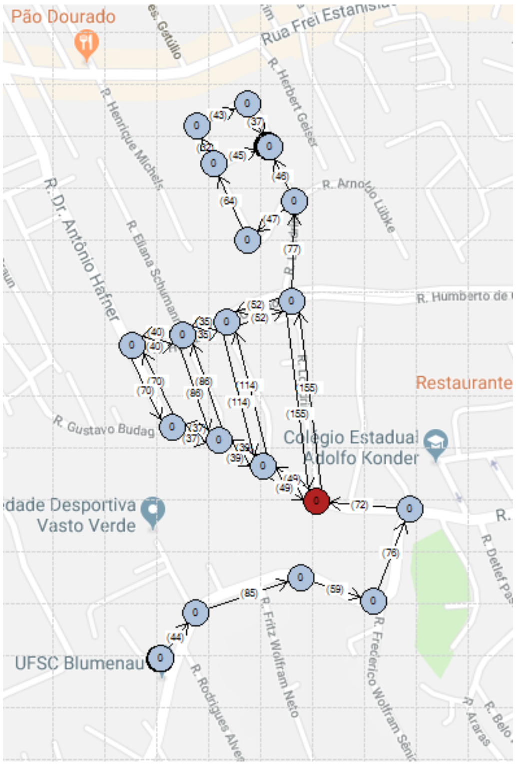

Figura 7: Mapa de rotas possíveis entre os pontos 2 e 3

Segue, na Figura 8, a representação do caminho mais curto obtido através da aplicação do algorítimo de Dijkstra.

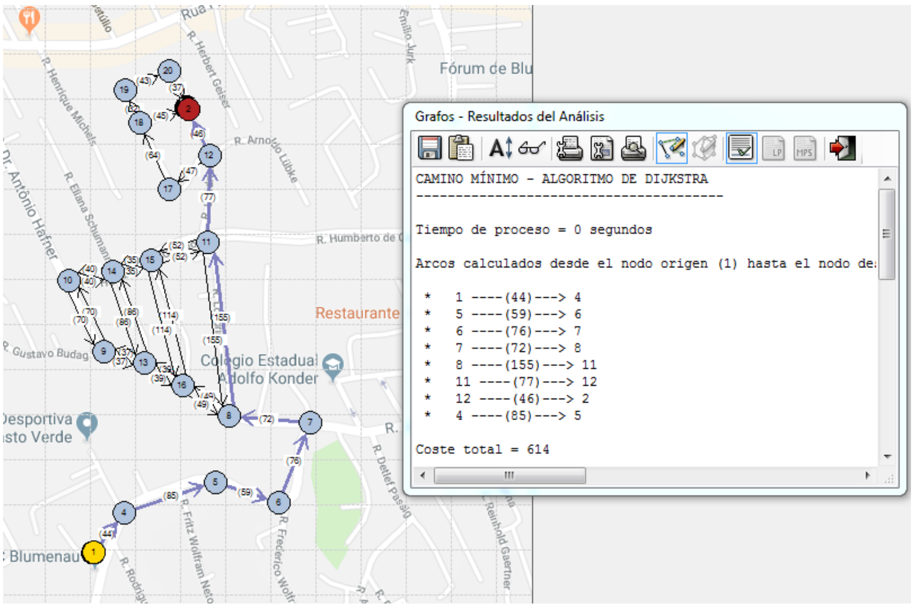

Figura 8: Mapa de rota mais curta entre os pontos 2 e 3

Caminho do ponto 2 até 4

Igualmente como os caminhos anteriores, construiu-se o grafo que melhor representa as possíveis rotas que ligam os pontos 2 e 4, como pode-se comprovar na Figura 9.

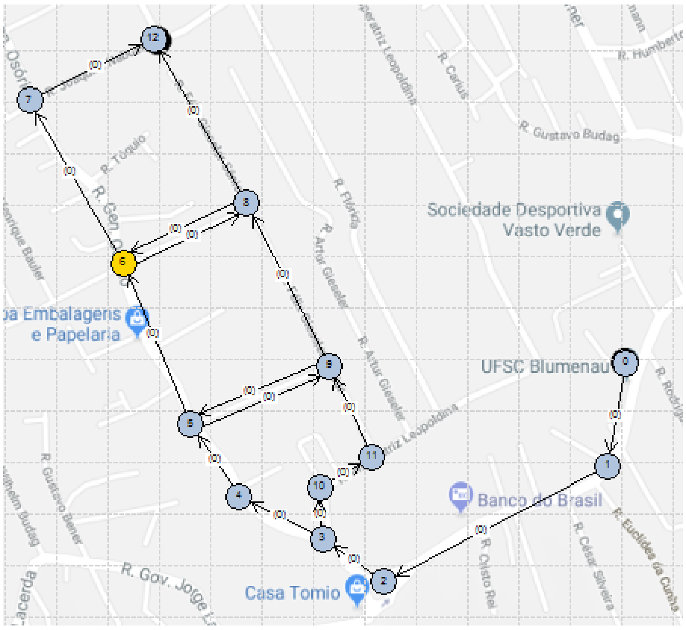

Figura 9: Mapa de rotas possíveis entre os pontos 2 e 4

Segue, na Figura 10, a representação do caminho mais curto obtido através da aplicação do algorítimo de Dijkstra entre os pontos 2 e 4.

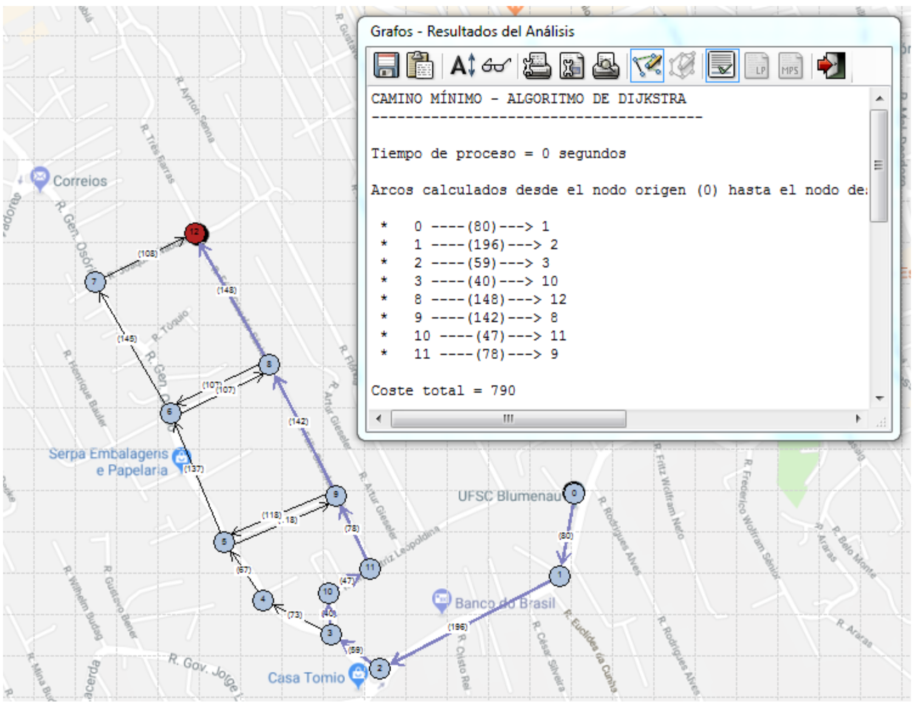

Figura 10: Mapa de rota mais curta entre os pontos 2 e 4

3 Considerações Finais

Perceba que no problema-exemplo de posicionamento global alguns caminhos foram propositalmente omitidos, por serem considerados desprezáveis (trivialmente longos), donde que um resultado mais robusto contaria com a presença de um grafo completo, envolvendo todos os possíveis caminhos. Ressalta-se também que um estudo sobre a complexidade computacional dos algorítimos apresentados seria proveitoso para algum possível trabalho futuro.

Espera-se ter deixado claro para o leitor quais os princípios básicos envolvendo grafos, principais resultados e esboçado possíveis aplicações com o desenvolvimento proposto. Vale a ressalva de que a bibliografia contém uma grande quantidade de algorítimos diferentes envolvendo grafos e uma diversa gama de aplicações.

Pede-se uma pequena pausa para a licença poética: Acreditamos que uma das mais belas características da representação dos grafos é que o conjunto de vértices V aceita qualquer elemento, ou seja, pode ser tanto um sensor, quanto uma esquina. Daí vem o grande poder de abstração da representação de grafos.

Referências

[1] J. A. Bondy and U. S. R. Murty. *Graph Theory With Applications.* Elsevier Science Publishing, New York, 5 edition, 1982.

[2] Paulo Oswaldo Boaventura Netto. *Grafos.* Blucher, São Paulo, 5 edition, 2012.

[3] Grafos - software para la construcción, edición y análisis de grafos. Disponivel em: http://arodrigu.webs.upv.es/grafos/doku.php?id=inicio.

[4] Clyde P Kruskal, Larry Rudolph, and Marc Snir. A complexity theory of efficient parallel algorithms. *Theoretical Computer Science*, 71(1):95–132, 1990.

[5] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.