PROJETO DE GRAFOS

Prof. Ivan Carlos Alcântara de Oliveira

**TRAFFIC SOLVER**

***Rota Alternativa***

**Autores:**

BRUNO CASTRO TOMAZ - 10389988

GUSTAVO SAAD MALUHY ANDRADE - 10332747 LAURA FONTE ABI DAUD - 10395586

**São Paulo - SP 2024**

**Relatório do Projeto**

# Introdução:

Imagine que você está planejando um trajeto tranquilo pela cidade de São Paulo. De repente, se depara com uma rua interditada, um cruzamento bloqueado ou até mesmo um acidente ou evento que impede a passagem. Nesse tipo de situação, muito comum na capital paulista, é essencial que veículos sejam capazes de realizar um contorno com o menor custo possível e pelo melhor caminho viável.



“Cratera aberta por vazamento de água interdita faixas da rua da Consolação, sentido Praça Roosevelt, na região central de São Paulo” [Disponível em

https://www1.folha.uol.com.br/cotidiano/2024/02/rua-da-consolacao-e-parcialmente-interditada-apos-pista-ceder.shtml Acesso em: 19/03/2024]

Pensando neste problema, nosso grupo resolveu desenvolver uma aplicação que mapeia algumas regiões da capital paulista como um grafo direcionado; com ruas sendo representadas por arestas e cruzamentos por vértices. Esta nossa aplicação abrange um dos 17 Objetivos de Desenvolvimento Sustentável da ONU: a *ODS 11 (Cidades e Comunidades Sustentáveis),* haja vista que busca facilitar o trânsito de São Paulo, estando em conformidade com a meta “11.2 - Até 2030, proporcionar o acesso a sistemas de transporte seguros, acessíveis, sustentáveis e a preço acessível para todos, melhorando a segurança rodoviária por meio da expansão dos transportes públicos, com especial atenção para as necessidades das pessoas em situação de vulnerabilidade, mulheres, crianças, pessoas com deficiência e idosos” [Disponível em: https://odsbrasil.gov.br/objetivo/objetivo?n=11; Acesso em 19/03/2024].

Nosso código deve ser capaz de sugerir a melhor rota alternativa em situações excepcionais, onde o trajeto habitual é bloqueado por razões distintas. Para isso podemos executar a remoção de vértices (cruzamentos) e/ou arestas (ruas) como uma simulação de um bloqueio no trajeto. Nessa fase inicial da aplicação, que será elaborada neste semestre, construíremos um protótipo simples que irá analisar as possibilidades em uma área no entorno da Universidade Presbiteriana Mackenzie (no campus Higienópolis). Vale considerar que estamos trabalhando com um cenário genérico, desconsiderando fatores dinâmicos como tempo de deslocamento, faróis, tipo de transporte (a pé, de carro ou ônibus, etc).

# Formação do Grafo:

Após pesquisar sobre o assunto, optamos por trabalhar em cima das seguintes ruas:



Tipo do grafo: 6- grafo orientado com peso na aresta

Essas ruas são listadas no arquivo excel, cujo link se apresenta abaixo. Foram feitas duas planilhas, uma com identificações dos vértices (cruzamentos) por número e outra com os nomes das ruas e o peso das arestas (a distância real do caminho entre dois cruzamentos, segundo Software Google Maps). As distâncias entre as ruas foram calculadas em metros.

Link: <https://docs.google.com/spreadsheets/d/16xTkdox8w65-dDSXWEc97QTZDLF5rMwHoS0m3t8MSJc/edit#gid=0>

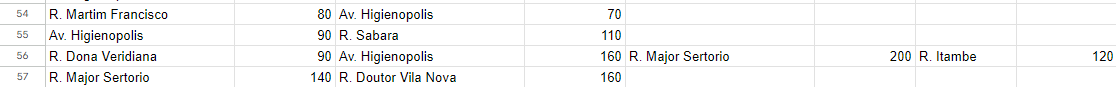
Exemplo parcial dos arquivos referenciados acima:

**Planilha 1: Definição de vértices e suas conexões:**

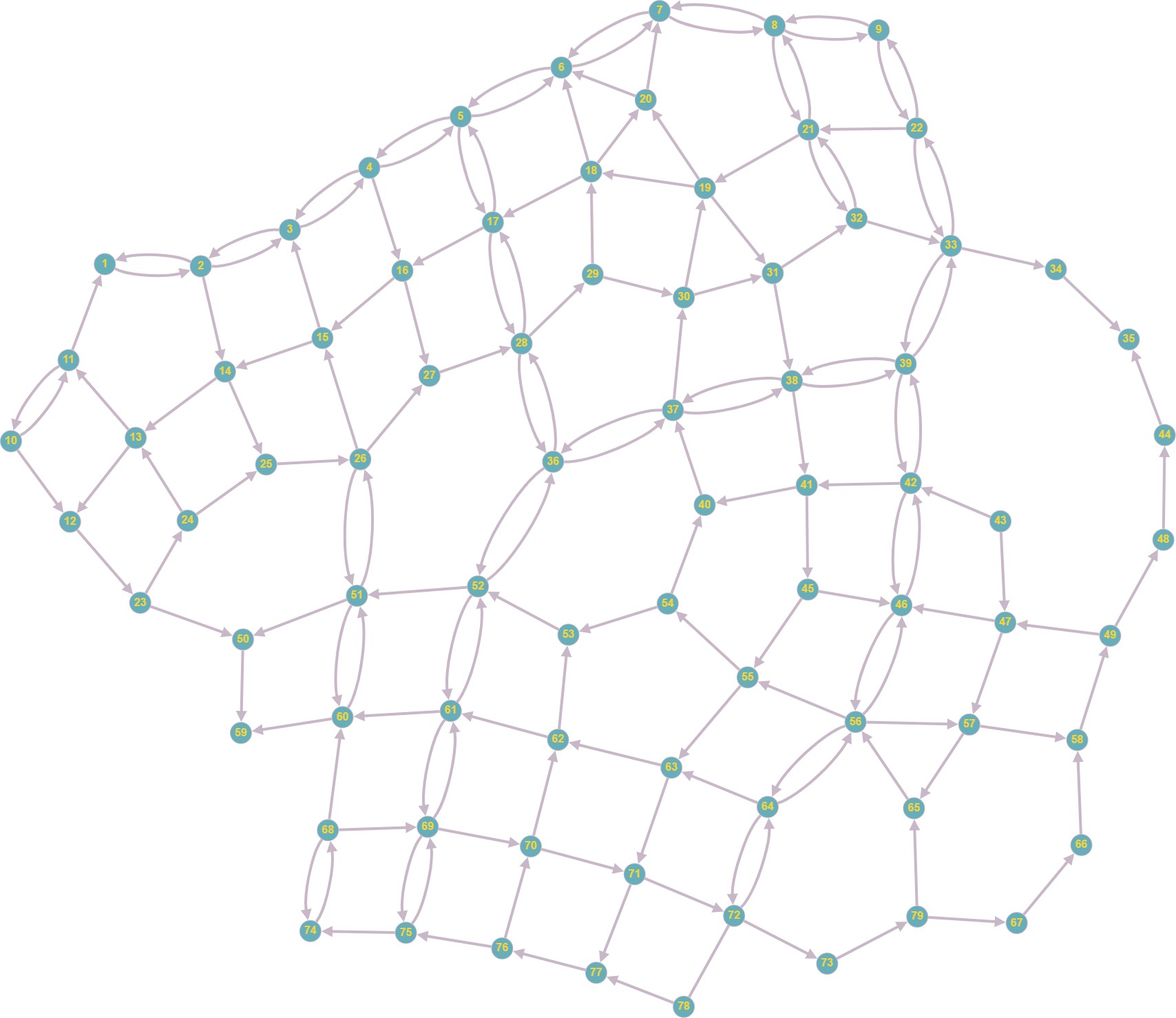
Tabela

Descrição gerada automaticamente

**Planilha 2: Definição dos nomes das arestas e seus respectivos pesos:**



Abaixo segue nossa modelagem do Grafo, através da ferramenta “graph.online”:



# Implementação:

Com esse grafo e nosso objetivo em mente, criamos um algoritmo com as seguintes funcionalidades:

* + Ler os dados de uma matriz de adjacência, e os dados desejados para transformar em um grafo;
  + Apresentar o nível de conexidade do grafo;
  + Adicionar ou remover vértices e arestas do grafo;
  + Permitir que o usuário insira sua origem e destino, para que o algoritmo calcule o caminho mínimo entre os dois vértices (feito pelo algoritmo de Dijkstra);
  + Permitir que o usuário insira um vértice (cruzamento) e descubra a quantidade de arestas (ruas) que entram e saem dele;
  + Verificar se um grafo é Euleriano ou não;
  + Verificar se um grafo é Hamiltoniano ou não

A partir da funcionalidade de remoção e inserção de vértices e arestas, o usuário pode remover ruas ou cruzamentos do algoritmo, para que ele possa calcular o caminho mínimo, mesmo que haja alguma interdição entre os dois pontos. O ponto interditado é removido do grafo e logo o algoritmo gera um novo caminho mínimo a partir das modificações do usuário.

Abaixo apresenta-se o menu do programa:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

**Resultados da Execução do menu:**

**Opção 1: Ler dados do arquivo**

**Interface gráfica do usuário

Descrição gerada automaticamente**

**Opção 2: Gravar dados no arquivo**

**Texto

Descrição gerada automaticamente com confiança média**

**Opção 3: Inserir vértice**

Teste 1:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Teste 2:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

**Opção 4: Inserir Aresta**

Teste 1:

Texto

Descrição gerada automaticamente

Teste 2:

Texto

Descrição gerada automaticamente

Teste 3:

Calendário

Descrição gerada automaticamente

**Opção 5: Remover Vértice**

Teste 1:

Texto

Descrição gerada automaticamente

Teste 2:

Tabela

Descrição gerada automaticamente com confiança média

**Opção 6: Remover Aresta:**

Teste 1:

Texto

Descrição gerada automaticamente

Teste 2:

Texto

Descrição gerada automaticamente

**Opção 7: Mostrar conteúdo do arquivo**

Texto

Descrição gerada automaticamente

....

Texto

Descrição gerada automaticamente

**Opção 8: Mostrar Grafo**

**Texto

Descrição gerada automaticamente com confiança média**

**....**

**Opção 9: Apresentar Componentes Conexas e Grafo Reduzido**

Teste 1:

Texto

Descrição gerada automaticamente

**Opção 10: Definir rota (GPS)**

Teste 1:

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Teste 2:

Texto

Descrição gerada automaticamente

Teste 3: antes e depois de remover arestas 46-> 56

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

DEPOIS DE REMOVER ARESTAS:

Texto

Descrição gerada automaticamente

**Opção 11: Verificar Informações de um cruzamento**

Texto

Descrição gerada automaticamente

Teste 2:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

**Opção 12: Verificar Grafo Euleriano**

Texto

Descrição gerada automaticamente

**Opção 13: Verificar Grafo Hamiltoniano**

Texto

Descrição gerada automaticamente

**Link github:** [**https://github.com/fabidaud/PROJETO-GRAFOS**](https://github.com/fabidaud/PROJETO-GRAFOS)