

**UNIVERSIDADE PRESBITERIANA MACKENZIE
(CAMPUS ALPHAVILLE)**

**TEORIA DOS GRAFOS
CIÊNCIA DA COMPUTAÇÃO, 5º SEMESTRE**

**ARTHUR CHRYSTIAN DE MORAIS STELLA; RA: 10314030
GLENDAL MILÉO TRIGO; RA: 10418587**

**ATIVIDADE PROJETO 1 E 2
ANÁLISE DO CONSUMO DE ENERGIA ELÉTRICA NO BRASIL**

**ALPHAVILLE
2025**

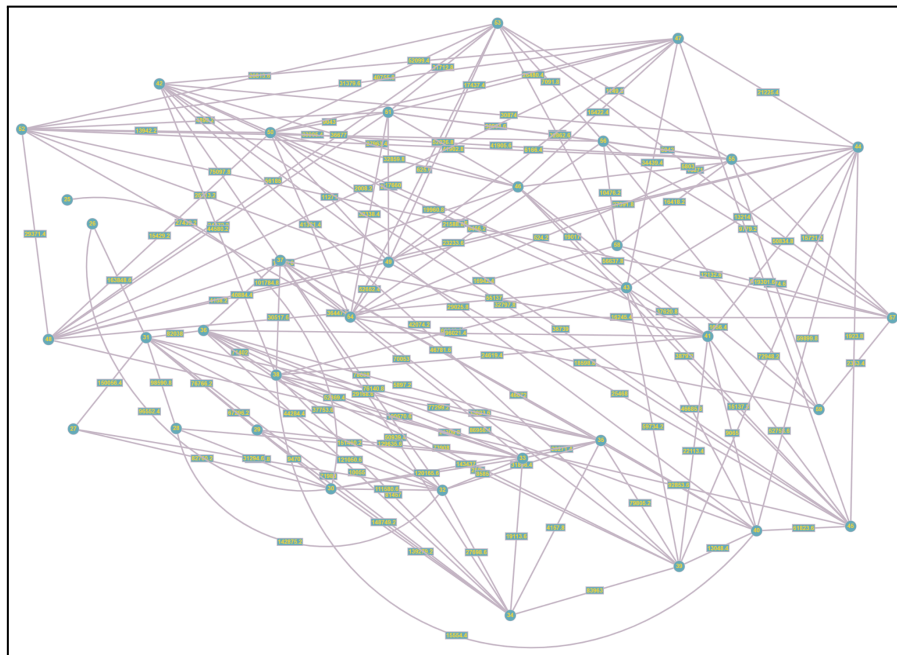
SUMÁRIO

DESCRIÇÃO TEXTUAL	3
Vértices (Localidades)	3
Arestas (Conexões entre Cidades)	4
ODS ESCOLHIDA	5
CÓDIGO - PYTHON	6
Atributos	6
Métodos	6
1.1 Carregar Grafo (carregar_grafo)	6
1.2 Salvar Grafo (salvar_grafo)	6
1.3 Mostrar Conteúdo do Arquivo (mostrar_conteudo_arquivo)	7
1.4 Mostrar Grafo (mostrar_grafo)	7
1.5 Inserir Vértice (inserir_vertice)	7
1.6 Inserir Aresta (inserir_aresta)	7
1.7 Remover Vértice (remover_vertice)	7
1.8 Remover Aresta (remover_aresta)	7
1.9 Verificar Conexidade (conexidade_grafo)	7
TESTES	9
PARTE 2 DO PROJETO - ATUALIZAÇÃO	18
Opção 10 Caminho Mínimo (Dijkstra)	18
Opção 11 Verificar se o Grafo é Euleriano	18
Opção 12 Caminhos Mínimos entre Todos os Pares (Floyd-Warshall)	18
Opção 13 Grau dos Vértices	19
Opção 14 Coloração de Vértices	19
TESTES (10 AO 14)	20
APÊNDICE A - GITHUB / FOTO GRAFO	32

DESCRIÇÃO TEXTUAL

É necessário refletir e analisar a difícil situação de interdependência da energia elétrica entre regiões e setores consumidores no Brasil; influenciada por fatores climáticos, econômicos e estruturais. Compreender essas relações é um caminho chave para alcançarmos a otimização da distribuição de energia, redução de desperdícios e a previsibilidade do consumo.

Posto isso, este projeto propõe a modelagem da rede de consumo elétrico por meio da Teoria dos Grafos, onde os vértices representam diferentes regiões ou setores e as arestas indicam relações baseadas em variações temporais e padrões de consumo. Essa abordagem permite visualizar conexões e identificar influências entre unidades consumidoras, auxiliando no desenvolvimento de estratégias mais eficientes para o setor brasileiro de energia.



[Link para melhor visualização da imagem do grafo no apêndice do arquivo.](#)

Vértices (Localidades)

1. Critério de Seleção:

- Selecionamos as 60 localidades com o maior consumo total médio de energia elétrica ao longo dos anos disponíveis no dataset.

- Isso garante que o grafo representa as cidades com maior demanda energética, tornando o modelo mais relevante.

2. Cálculo do Consumo Médio:

- O consumo médio de cada localidade foi calculado somando os valores anuais e dividindo pelo número de anos disponíveis.
- Isso ajuda nas variações anuais e prioriza cidades que, de forma consistente, apresentam alto consumo.

Arestas (Conexões entre Cidades)

1. Critério de Conexão:

- Conectamos cada localidade às mais próximas em consumo, ordenando os valores de consumo médio.
- O objetivo é garantir que cidades com padrões de consumo semelhantes estejam interligadas.

2. Quantidade Mínima de Conexões:

- Para garantir pelo menos 150 arestas, seguimos duas estratégias:
 - **Primeira fase:** Criamos conexões naturais entre cidades com consumo próximo.
 - **Segunda fase:** Se o número de arestas ainda for insuficiente, ampliamos o critério de proximidade, conectando cidades com diferenças maiores, até alcançar o mínimo necessário.

3. Controle de Conexões por Vértice:

- Para evitar que algumas cidades fiquem muito conectadas e outras pouco conectadas, cada localidade tem um limite inicial de até 5 conexões.

4. Peso das Arestas:

- O peso de cada aresta representa a diferença de consumo elétrico entre as duas cidades conectadas.
- Isso permite que análises sobre o grafo levem em conta a variação no consumo entre as localidades.

ODS ESCOLHIDA

ODS 7 – Energia Acessível e Limpa: Buscando compreender os padrões de consumo de energia elétrica no Brasil, alinhar o projeto com esse ODS permite uma análise das correlações entre os diferentes setores e regiões. Ao identificar essas relações por meio da Teoria dos Grafos, será possível auxiliar na formulação de políticas públicas mais eficazes, que promovam uma distribuição energética mais equitativa, eficiente e sustentável. Além disso, o projeto também pode contribuir para o melhor aproveitamento de fontes renováveis, garantindo maior acessibilidade e segurança para a população.

ODS 9 – Indústria, Inovação e Infraestrutura: A aplicação de modelagem matemática e análise de redes complexas no setor elétrico representa um avanço significativo na forma como a infraestrutura energética é compreendida e gerenciada. Portanto, se torna possível identificar padrões de consumo relevantes, prever variações na demanda e otimizar a distribuição de energia de maneira mais inteligente. Podendo trazer inovação e modernização na indústria e na infraestrutura que realiza o fornecimento de energia no território brasileiro.

CÓDIGO - PYTHON

O código implementa uma classe Grafo que utiliza a biblioteca networkx para manipular grafos direcionados. O programa permite carregar um grafo a partir de um arquivo, exibir sua representação, realizar operações como inserção e remoção de vértices e arestas, verificar conexidade e salvar as alterações no arquivo. Um menu interativo possibilita a interação do usuário com o grafo.

A **classe Grafo** contém os seguintes atributos e métodos:

Atributos

- grafo: Um objeto DiGraph da biblioteca networkx, que representa o grafo direcionado.
- nomes_vertices: Um dicionário que mapeia IDs dos vértices para nomes de cidades.
- arquivo: Uma string contendo o caminho do arquivo utilizado para carregar e salvar o grafo.

Métodos

1.1 Carregar Grafo (carregar_grafo)

Este método lê um arquivo de texto contendo a definição do grafo e carrega suas informações na estrutura de dados.

- Lê o arquivo linha por linha e extrai:
 - Tipo do grafo.
 - Número de vértices.
 - Número de arestas.
 - Vértices, incluindo ID, nome e peso.
 - Arestas, incluindo vértices de origem e destino, e peso.
- Utiliza networkx para adicionar os vértices e arestas ao grafo.

1.2 Salvar Grafo (salvar_grafo)

Este método escreve o grafo atualizado de volta para o arquivo de origem.

- Formata e escreve os vértices e arestas no arquivo.

1.3 Mostrar Conteúdo do Arquivo (mostrar_conteudo_arquivo)

Exibe o conteúdo do arquivo que define o grafo, permitindo ao usuário visualizar sua estrutura em texto.

1.4 Mostrar Grafo (mostrar_grafo)

Exibe o grafo em duas representações:

- **Lista de Adjacência:** Mostra cada vértice e seus vizinhos.
- **Matriz de Adjacência:** Representa a conexão entre vértices em uma matriz numérica.

1.5 Inserir Vértice (inserir_vertice)

Adiciona um novo vértice ao grafo com ID, nome da cidade e peso.

- Atualiza nomes_vertices para mapear o novo vértice ao seu nome.

1.6 Inserir Aresta (inserir_aresta)

Adiciona uma aresta entre dois vértices existentes, associando um peso à conexão.

1.7 Remover Vértice (remover_vertice)

Remove um vértice do grafo e todas as arestas conectadas a ele.

1.8 Remover Aresta (remover_aresta)

Remove uma aresta entre dois vértices.

1.9 Verificar Conexidade (conexidade_grafo)

Analisa a conexidade do grafo e classifica em:

- **Fortemente conexo (C3):** Existe um caminho entre todos os pares de vértices.
- **Simplesmente conexo (C2):** O grafo não é fortemente conexo, mas todas as componentes estão conectadas por caminhos indiretos.

- **Semi fortemente conexo (C1):** Existem componentes conexas, mas não para todos os vértices.
- **Desconexo (C0):** O grafo possui vértices isolados.

O **menu interativo** permite ao usuário escolher a operação desejada:

1. **Mostrar Conteúdo do Arquivo:** Chama `mostrar_conteudo_arquivo`.
2. **Mostrar Grafo:** Chama `mostrar_grafo`.
3. **Inserir Vértice:** Solicita ID, nome e peso, e adiciona ao grafo.
4. **Inserir Aresta:** Solicita vértices e peso, e cria uma aresta.
5. **Remover Vértice:** Remove um vértice informado.
6. **Remover Aresta:** Remove uma aresta informada.
7. **Verificar Conexidade:** Exibe o status de conexidade do grafo.
8. **Salvar e Sair:** Salva o grafo no arquivo e encerra o programa.
9. **Sair sem Salvar:** Fecha o programa sem salvar alterações.

O menu é executado em um loop `while`, garantindo que o usuário possa realizar múltiplas operações antes de sair.

TESTES

Testes realizados respeitando a regra de testar duas vezes as funções principais do código:

```
✓ Grafo carregado com sucesso!

✧ MENU
1. Mostrar Arquivo
2. Mostrar Grafo
3. Inserir Vértice
4. Inserir Aresta
5. Remover Vértice
6. Remover Aresta
7. Conexidade
8. Salvar e Sair
9. Sair
Escolha: 1

📁 Conteúdo do Arquivo:

Squeezed text (213 lines).

✧ MENU
1. Mostrar Arquivo
2. Mostrar Grafo
3. Inserir Vértice
4. Inserir Aresta
5. Remover Vértice
6. Remover Aresta
7. Conexidade
8. Salvar e Sair
9. Sair
Escolha: 2

📊 Representação do Grafo:

Lista de Adjacência:
0: []
1: []
2: []
3: []
4: []
```

```
5: []
6: []
7: []
8: []
9: []
10: []
11: []
12: []
13: []
14: []
15: []
16: []
17: []
18: []
19: []
20: []
21: []
22: []
23: []
24: []
25: []
26: []
27: []
28: []
29: []
30: [25, 26, 27, 28, 29]
31: [26, 27, 28, 29, 30]
32: [27, 28, 29, 30, 31]
33: [28, 29, 30, 31, 32]
34: [29, 30, 31, 32, 33]
35: [30, 31, 32, 33, 34]
36: [31, 32, 33, 34, 35]
37: [32, 33, 34, 35, 36]
38: [33, 34, 35, 36, 37]
39: [34, 35, 36, 37, 38]
40: [35, 36, 37, 38, 39]
41: [36, 37, 38, 39, 40]
42: [37, 38, 39, 40, 41]
43: [38, 39, 40, 41, 42]
44: [39, 40, 41, 42, 43]
45: [40, 41, 42, 43, 44]
```

45: [40, 41, 42, 43, 44]
46: [41, 42, 43, 44, 45]
47: [42, 43, 44, 45, 46]
48: [43, 44, 45, 46, 47]
49: [44, 45, 46, 47, 48]
50: [45, 46, 47, 48, 49]
51: [46, 47, 48, 49, 50]
52: [47, 48, 49, 50, 51]
53: [48, 49, 50, 51, 52]
54: [49, 50, 51, 52, 53]
55: [50, 51, 52, 53, 54]
56: [51, 52, 53, 54, 55]
57: [52, 53, 54, 55, 56]
58: [53, 54, 55, 56, 57]
59: [54, 55, 56, 57, 58]

Matriz de Adjacência:

[illegible][illegible]

Nome: CIDA

Peso: 500

☒ Vértice 60 (CIDA) inserido com sucesso!

✧ MENU

1. Mostrar Arquivo
2. Mostrar Grafo
3. Inserir Vértice
4. Inserir Aresta
5. Remover Vértice
6. Remover Aresta
7. Conexidade
8. Salvar e Sair
9. Sair

Escolha: 3

Vértice: 61

Nome: CIDB

Peso: 600

☒ Vértice 61 (CIDB) inserido com sucesso!

✧ MENU

1. Mostrar Arquivo
2. Mostrar Grafo
3. Inserir Vértice
4. Inserir Aresta
5. Remover Vértice
6. Remover Aresta
7. Conexidade
8. Salvar e Sair
9. Sair

Escolha: 4

Origem: 59

Destino: 60

Peso: 600

☒ Aresta de 59 para 60 inserida com sucesso!



MENU

1. Mostrar Arquivo
2. Mostrar Grafo
3. Inserir Vértice
4. Inserir Aresta
5. Remover Vértice
6. Remover Aresta
7. Conexidade
8. Salvar e Sair
9. Sair

Escolha: 4

Origem: 60

Destino: 61

Peso: 300



Aresta de 60 para 61 inserida com sucesso!



MENU

1. Mostrar Arquivo
2. Mostrar Grafo
3. Inserir Vértice
4. Inserir Aresta
5. Remover Vértice
6. Remover Aresta
7. Conexidade
8. Salvar e Sair
9. Sair

Escolha: 2



Representação do Grafo:

Lista de Adjacência:

0: []

1: []

2: []

3: []

4: []

5: []

6: []

7: []

```
8: []
9: []
10: []
11: []
12: []
13: []
14: []
15: []
16: []
17: []
18: []
19: []
20: []
21: []
22: []
23: []
24: []
25: []
26: []
27: []
28: []
29: []
30: [25, 26, 27, 28, 29]
31: [26, 27, 28, 29, 30]
32: [27, 28, 29, 30, 31]
33: [28, 29, 30, 31, 32]
34: [29, 30, 31, 32, 33]
35: [30, 31, 32, 33, 34]
36: [31, 32, 33, 34, 35]
37: [32, 33, 34, 35, 36]
38: [33, 34, 35, 36, 37]
39: [34, 35, 36, 37, 38]
40: [35, 36, 37, 38, 39]
41: [36, 37, 38, 39, 40]
42: [37, 38, 39, 40, 41]
43: [38, 39, 40, 41, 42]
44: [39, 40, 41, 42, 43]
45: [40, 41, 42, 43, 44]
46: [41, 42, 43, 44, 45]
47: [42, 43, 44, 45, 46]
48: [43, 44, 45, 46, 47]
```

[illegible][illegible]

PARTE 2 DO PROJETO - ATUALIZAÇÃO

Para dar continuidade à modelagem do problema proposto no projeto (ANÁLISE DO CONSUMO DE ENERGIA ELÉTRICA NO BRASIL) optamos por seguir a primeira abordagem sugerida na Atividade Projeto 2: Investigar uma solução para o problema baseado no conteúdo da disciplina.

Nosso objetivo foi utilizar técnicas estudadas ao longo da disciplina para aprofundar a análise do grafo e extrair informações relevantes sobre o comportamento da rede elétrica representada. As novas funcionalidades implementadas no menu (opções 10 a 14) foram selecionadas com base em responder a questões pertinentes ao problema.

Opção 10 Caminho Mínimo (Dijkstra)

- Objetivo: Identificar o trajeto mais eficiente (menor custo) para transmitir energia de uma cidade até outra.
- Justificativa da escolha: No contexto do consumo de energia, encontrar o caminho mais curto entre dois pontos pode representar a rota de menor perda, menor tempo ou menor custo operacional para transporte de energia. O algoritmo de Dijkstra, estudado na aula 09, é eficiente para esse tipo de análise em grafos ponderados com pesos positivos, exatamente como o nosso modelo. Essa escolha permite compreender e otimizar conexões importantes na rede elétrica simulada.

Opção 11 Verificar se o Grafo é Euleriano

- Objetivo: Verificar se é possível visitar todas as conexões (arestas) da rede sem repetir nenhuma.
- Justificativa da escolha: Grafos eulerianos têm aplicação direta em rotas de inspeção e manutenção da rede elétrica, pois indicam se é possível percorrer toda a rede utilizando todas as conexões exatamente uma vez. Aula 16 da disciplina é aplicada aqui para investigar a estrutura do sistema e sua capacidade de ser percorrido integralmente de forma eficiente.

Opção 12 Caminhos Mínimos entre Todos os Pares (Floyd-Warshall)

- Objetivo: Calcular o custo mínimo de transmissão de energia entre todas as combinações possíveis de cidades.
- Justificativa da escolha: Através do algoritmo de Floyd-Warshall, aula 10, é possível construir uma matriz completa com os menores caminhos entre todos os pares de vértices. Isso fornece uma visão abrangente sobre a eficiência da rede em diferentes cenários de abastecimento energético, otimizando decisões operacionais de distribuição e expansão da malha elétrica.

Opção 13 Grau dos Vértices

- Objetivo: Avaliar quantas conexões de entrada e saída cada cidade possui na rede.
- Justificativa da escolha: O grau de cada vértice representa diretamente o papel de cada cidade como centro distribuidor ou ponto consumidor. Esta análise ajuda a identificar hubs da rede, gargalos e regiões isoladas, sendo essencial para o planejamento energético. Abordado nas aulas 03, 05 e 13.

Opção 14 Coloração de Vértices

- Objetivo: Agrupar cidades em clusters que podem operar simultaneamente sem conflito (ex: sem sobrecarga na rede).
- Justificativa da escolha: A coloração de vértices, aula 15, foi usada para estimar o número mínimo de grupos que podem operar independentemente. Em termos de energia, isso pode representar clusters regionais ou níveis de carga balanceada. Essa análise é útil para evitar sobreposições de carga, planejamento de manutenção regionalizada e balanceamento de energia.

Com essas cinco novas funcionalidades, conseguimos atender aos dois pontos exigidos pela atividade:

- (1) Aplicamos técnicas aprendidas em sala que contribuem diretamente para a solução e entendimento do problema modelado;
- (2) Investigamos ao menos três características do problema, como caminhos mínimos, estrutura euleriana, grau dos vértices, coloração e conexidade.

TESTES (10 AO 14)



OBS.: Testamos apenas essas partes, pois já temos exemplos das outras opções do menu.

```
✦ MENU
1. Mostrar Arquivo
2. Mostrar Grafo
3. Inserir Vértice
4. Inserir Aresta
5. Remover Vértice
6. Remover Aresta
7. Conexidade
8. Salvar e Sair
9. Sair
10. Caminho Mínimo (Dijkstra)
11. Verificar se é Euleriano
12. Verificar Ciclo Hamiltoniano
13. Mostrar Grau dos Vértices
14. Coloração de Vértices
Escolha: 10
Origem: 1
Destino: 15
```

✕ Caminho não encontrado.

```
✦ MENU
1. Mostrar Arquivo
2. Mostrar Grafo
3. Inserir Vértice
4. Inserir Aresta
5. Remover Vértice
6. Remover Aresta
7. Conexidade
8. Salvar e Sair
9. Sair
10. Caminho Mínimo (Dijkstra)
11. Verificar se é Euleriano
12. Verificar Ciclo Hamiltoniano
13. Mostrar Grau dos Vértices
14. Coloração de Vértices
Escolha: 10
Origem: 59
Destino: 54
```

Destino: 54


 Caminho mínimo de 59 para 54 (São Paulo  São Bernardo do Campo): custo = 18598.79999999999

 MENU

1. Mostrar Arquivo
2. Mostrar Grafo
3. Inserir Vértice
4. Inserir Aresta
5. Remover Vértice
6. Remover Aresta
7. Conexidade
8. Salvar e Sair
9. Sair
10. Caminho Mínimo (Dijkstra)
11. Verificar se é Euleriano
12. Verificar Ciclo Hamiltoniano
13. Mostrar Grau dos Vértices
14. Coloração de Vértices


Escolha: 11

 O grafo NÃO é Euleriano.

 MENU

1. Mostrar Arquivo
2. Mostrar Grafo
3. Inserir Vértice
4. Inserir Aresta
5. Remover Vértice
6. Remover Aresta
7. Conexidade
8. Salvar e Sair
9. Sair
10. Caminho Mínimo (Dijkstra)
11. Verificar se é Euleriano
12. Caminhos Mínimos entre Todos os Pares (Floyd-Warshall)
13. Mostrar Grau dos Vértices
14. Coloração de Vértices

Escolha: 12

 Matriz de Caminhos Mínimos entre Todos os Pares (Floyd-Warshall):

Indaiatuba → Araraquara: custo = 101784.80
Indaiatuba → Mogi Guaçu: custo = 96552.40
Indaiatuba → Guarujá: custo = 82760.20
Indaiatuba → São Caetano do Sul: custo = 31294.60
Indaiatuba → Taubaté: custo = 9470.00
São Carlos → Araraquara: custo = 169081.00
São Carlos → Mogi Guaçu: custo = 163848.60
São Carlos → Guarujá: custo = 150056.40
São Carlos → São Caetano do Sul: custo = 98590.80
São Carlos → Taubaté: custo = 76766.20
São Carlos → Indaiatuba: custo = 67296.20
Bauru → Araraquara: custo = 213365.40
Bauru → Mogi Guaçu: custo = 208133.00
Bauru → Guarujá: custo = 194340.80
Bauru → São Caetano do Sul: custo = 142875.20
Bauru → Taubaté: custo = 121050.60
Bauru → Indaiatuba: custo = 111580.60
Bauru → São Carlos: custo = 44284.40
Rio Claro → Araraquara: custo = 221950.40

Rio Claro → Araraquara: custo = 221950.40
Rio Claro → Mogi Guaçu: custo = 216718.00
Rio Claro → Guarujá: custo = 202925.80
Rio Claro → São Caetano do Sul: custo = 151460.20
Rio Claro → Taubaté: custo = 129635.60
Rio Claro → Indaiatuba: custo = 120165.60
Rio Claro → São Carlos: custo = 52869.40
Rio Claro → Bauru: custo = 8585.00
Sumaré → Araraquara: custo = 241064.00
Sumaré → Mogi Guaçu: custo = 235831.60
Sumaré → Guarujá: custo = 222039.40
Sumaré → São Caetano do Sul: custo = 170573.80
Sumaré → Taubaté: custo = 148749.20
Sumaré → Indaiatuba: custo = 139279.20
Sumaré → São Carlos: custo = 71983.00
Sumaré → Bauru: custo = 27698.60
Sumaré → Rio Claro: custo = 19113.60
Diadema → Araraquara: custo = 245221.80
Diadema → Mogi Guaçu: custo = 239989.40
Diadema → Guarujá: custo = 226197.20
Diadema → São Caetano do Sul: custo = 174731.60
Diadema → Taubaté: custo = 152907.00
Diadema → Indaiatuba: custo = 143437.00
Diadema → São Carlos: custo = 76140.80
Diadema → Bauru: custo = 31856.40
Diadema → Rio Claro: custo = 23271.40
Diadema → Sumaré: custo = 4157.80
Paulínia → Araraquara: custo = 251119.00
Paulínia → Mogi Guaçu: custo = 245886.60
Paulínia → Guarujá: custo = 232094.40
Paulínia → São Caetano do Sul: custo = 180628.80
Paulínia → Taubaté: custo = 158804.20
Paulínia → Indaiatuba: custo = 149334.20
Paulínia → São Carlos: custo = 82038.00
Paulínia → Bauru: custo = 37753.60
Paulínia → Rio Claro: custo = 29168.60
Paulínia → Sumaré: custo = 10055.00
Paulínia → Diadema: custo = 5897.20
Jacareí → Araraquara: custo = 292003.40
Jacareí → Mogi Guaçu: custo = 286771.00
Jacareí → Guarujá: custo = 272978.80

Jacareí → Guarujá: custo = 272978.80
 Jacareí → São Caetano do Sul: custo = 221513.20
 Jacareí → Taubaté: custo = 199688.60
 Jacareí → Indaiatuba: custo = 190218.60
 Jacareí → São Carlos: custo = 122922.40
 Jacareí → Bauru: custo = 78638.00
 Jacareí → Rio Claro: custo = 70053.00
 Jacareí → Sumaré: custo = 50939.40
 Jacareí → Diadema: custo = 46781.60
 Jacareí → Paulínia: custo = 40884.40
 Americana → Araraquara: custo = 322521.00
 Americana → Mogi Guaçu: custo = 317288.60
 Americana → Guarujá: custo = 303496.40
 Americana → São Caetano do Sul: custo = 252030.80
 Americana → Taubaté: custo = 230206.20
 Americana → Indaiatuba: custo = 220736.20
 Americana → São Carlos: custo = 153440.00
 Americana → Bauru: custo = 109155.60
 Americana → Rio Claro: custo = 100570.60
 Americana → Sumaré: custo = 81457.00
 Americana → Diadema: custo = 77299.20
 Americana → Paulínia: custo = 71402.00
 Americana → Jacareí: custo = 30517.60
 São José do Rio Preto → Araraquara: custo = 325027.00
 São José do Rio Preto → Mogi Guaçu: custo = 319794.60
 São José do Rio Preto → Guarujá: custo = 306002.40
 São José do Rio Preto → São Caetano do Sul: custo = 254536.80
 São José do Rio Preto → Taubaté: custo = 232712.20
 São José do Rio Preto → Indaiatuba: custo = 223242.20
 São José do Rio Preto → São Carlos: custo = 155946.00
 São José do Rio Preto → Bauru: custo = 111661.60
 São José do Rio Preto → Rio Claro: custo = 103076.60
 São José do Rio Preto → Sumaré: custo = 83963.00
 São José do Rio Preto → Diadema: custo = 79805.20
 São José do Rio Preto → Paulínia: custo = 73908.00
 São José do Rio Preto → Jacareí: custo = 33023.60
 São José do Rio Preto → Americana: custo = 2506.00
 Pindamonhangaba → Araraquara: custo = 338075.40
 Pindamonhangaba → Mogi Guaçu: custo = 332843.00
 Pindamonhangaba → Guarujá: custo = 319050.80
 Pindamonhangaba → São Caetano do Sul: custo = 267585.20

Pindamonhangaba → São Caetano do Sul: custo = 267585.20
Pindamonhangaba → Taubaté: custo = 245760.60
Pindamonhangaba → Indaiatuba: custo = 236290.60
Pindamonhangaba → São Carlos: custo = 168994.40
Pindamonhangaba → Bauru: custo = 124710.00
Pindamonhangaba → Rio Claro: custo = 116125.00
Pindamonhangaba → Sumaré: custo = 97011.40
Pindamonhangaba → Diadema: custo = 92853.60
Pindamonhangaba → Paulínia: custo = 86956.40
Pindamonhangaba → Jacareí: custo = 46072.00
Pindamonhangaba → Americana: custo = 15554.40
Pindamonhangaba → São José do Rio Preto: custo = 13048.40
Mauá → Araraquara: custo = 347140.40
Mauá → Mogi Guaçu: custo = 341908.00
Mauá → Guarujá: custo = 328115.80
Mauá → São Caetano do Sul: custo = 276650.20
Mauá → Taubaté: custo = 254825.60
Mauá → Indaiatuba: custo = 245355.60
Mauá → São Carlos: custo = 178059.40
Mauá → Bauru: custo = 133775.00
Mauá → Rio Claro: custo = 125190.00
Mauá → Sumaré: custo = 106076.40
Mauá → Diadema: custo = 101918.60
Mauá → Paulínia: custo = 96021.40
Mauá → Jacareí: custo = 55137.00
Mauá → Americana: custo = 24619.40
Mauá → São José do Rio Preto: custo = 22113.40
Mauá → Pindamonhangaba: custo = 9065.00
Mogi das Cruzes → Araraquara: custo = 367101.20
Mogi das Cruzes → Mogi Guaçu: custo = 361868.80
Mogi das Cruzes → Guarujá: custo = 348076.60
Mogi das Cruzes → São Caetano do Sul: custo = 296611.00
Mogi das Cruzes → Taubaté: custo = 274786.40
Mogi das Cruzes → Indaiatuba: custo = 265316.40
Mogi das Cruzes → São Carlos: custo = 198020.20
Mogi das Cruzes → Bauru: custo = 153735.80
Mogi das Cruzes → Rio Claro: custo = 145150.80
Mogi das Cruzes → Sumaré: custo = 126037.20
Mogi das Cruzes → Diadema: custo = 121879.40
Mogi das Cruzes → Paulínia: custo = 115982.20
Mogi das Cruzes → Jacareí: custo = 75097.80

Mogi das Cruzes → Bauru: custo = 153735.80
Mogi das Cruzes → Rio Claro: custo = 145150.80
Mogi das Cruzes → Sumaré: custo = 126037.20
Mogi das Cruzes → Diadema: custo = 121879.40
Mogi das Cruzes → Paulínia: custo = 115982.20
Mogi das Cruzes → Jacareí: custo = 75097.80
Mogi das Cruzes → Americana: custo = 44580.20
Mogi das Cruzes → São José do Rio Preto: custo = 42074.20
Mogi das Cruzes → Pindamonhangaba: custo = 29025.80
Mogi das Cruzes → Mauá: custo = 19960.80
Barueri → Araraquara: custo = 384761.20
Barueri → Mogi Guaçu: custo = 379528.80
Barueri → Guarujá: custo = 365736.60
Barueri → São Caetano do Sul: custo = 314271.00
Barueri → Taubaté: custo = 292446.40
Barueri → Indaiatuba: custo = 282976.40
Barueri → São Carlos: custo = 215680.20
Barueri → Bauru: custo = 171395.80
Barueri → Rio Claro: custo = 162810.80
Barueri → Sumaré: custo = 143697.20
Barueri → Diadema: custo = 139539.40
Barueri → Paulínia: custo = 133642.20
Barueri → Jacareí: custo = 92757.80
Barueri → Americana: custo = 62240.20
Barueri → São José do Rio Preto: custo = 59734.20
Barueri → Pindamonhangaba: custo = 46685.80
Barueri → Mauá: custo = 37620.80
Barueri → Mogi das Cruzes: custo = 17660.00
Suzano → Araraquara: custo = 397975.20
Suzano → Mogi Guaçu: custo = 392742.80
Suzano → Guarujá: custo = 378950.60
Suzano → São Caetano do Sul: custo = 327485.00
Suzano → Taubaté: custo = 305660.40
Suzano → Indaiatuba: custo = 296190.40
Suzano → São Carlos: custo = 228894.20
Suzano → Bauru: custo = 184609.80
Suzano → Rio Claro: custo = 176024.80
Suzano → Sumaré: custo = 156911.20
Suzano → Diadema: custo = 152753.40
Suzano → Paulínia: custo = 146856.20
Suzano → Jacareí: custo = 105971.80

Suzano → Bauru: custo = 184609.80
Suzano → Rio Claro: custo = 176024.80
Suzano → Sumaré: custo = 156911.20
Suzano → Diadema: custo = 152753.40
Suzano → Paulínia: custo = 146856.20
Suzano → Jacareí: custo = 105971.80
Suzano → Americana: custo = 75454.20
Suzano → São José do Rio Preto: custo = 72948.20
Suzano → Pindamonhangaba: custo = 59899.80
Suzano → Mauá: custo = 50834.80
Suzano → Mogi das Cruzes: custo = 30874.00
Suzano → Barueri: custo = 13214.00
Limeira → Araraquara: custo = 399899.00
Limeira → Mogi Guaçu: custo = 394666.60
Limeira → Guarujá: custo = 380874.40
Limeira → São Caetano do Sul: custo = 329408.80
Limeira → Taubaté: custo = 307584.20
Limeira → Indaiatuba: custo = 298114.20
Limeira → São Carlos: custo = 230818.00
Limeira → Bauru: custo = 186533.60
Limeira → Rio Claro: custo = 177948.60
Limeira → Sumaré: custo = 158835.00
Limeira → Diadema: custo = 154677.20
Limeira → Paulínia: custo = 148780.00
Limeira → Jacareí: custo = 107895.60
Limeira → Americana: custo = 77378.00
Limeira → São José do Rio Preto: custo = 74872.00
Limeira → Pindamonhangaba: custo = 61823.60
Limeira → Mauá: custo = 52758.60
Limeira → Mogi das Cruzes: custo = 32797.80
Limeira → Barueri: custo = 15137.80
Limeira → Suzano: custo = 1923.80
Santos → Araraquara: custo = 403778.20
Santos → Mogi Guaçu: custo = 398545.80
Santos → Guarujá: custo = 384753.60
Santos → São Caetano do Sul: custo = 333288.00
Santos → Taubaté: custo = 311463.40
Santos → Indaiatuba: custo = 301993.40
Santos → São Carlos: custo = 234697.20
Santos → Bauru: custo = 190412.80
Santos → Rio Claro: custo = 181827.80

Alumínio → Santo André: custo = 16418.20
Alumínio → Guarulhos: custo = 10476.20
Alumínio → Campinas: custo = 697.00
Alumínio → cida: custo = 5000.00
São Paulo → Araraquara: custo = 496618.20
São Paulo → Mogi Guaçu: custo = 491385.80
São Paulo → Guarujá: custo = 477593.60
São Paulo → São Caetano do Sul: custo = 426128.00
São Paulo → Taubaté: custo = 404303.40
São Paulo → Indaiatuba: custo = 394833.40
São Paulo → São Carlos: custo = 327537.20
São Paulo → Bauru: custo = 283252.80
São Paulo → Rio Claro: custo = 274667.80
São Paulo → Sumaré: custo = 255554.20
São Paulo → Diadema: custo = 251396.40
São Paulo → Paulínia: custo = 245499.20
São Paulo → Jacareí: custo = 204614.80
São Paulo → Americana: custo = 174097.20
São Paulo → São José do Rio Preto: custo = 171591.20
São Paulo → Pindamonhangaba: custo = 158542.80
São Paulo → Mauá: custo = 149477.80
São Paulo → Mogi das Cruzes: custo = 129517.00
São Paulo → Barueri: custo = 111857.00
São Paulo → Suzano: custo = 98643.00
São Paulo → Limeira: custo = 96719.20
São Paulo → Santos: custo = 92840.00
São Paulo → Osasco: custo = 77417.60
São Paulo → São José dos Campos: custo = 75409.40
São Paulo → Ribeirão Preto: custo = 71251.20
São Paulo → Jundiaí: custo = 59980.20
São Paulo → Sorocaba: custo = 50937.20
São Paulo → Piracicaba: custo = 46038.00
São Paulo → Cubatão: custo = 19224.40
São Paulo → São Bernardo do Campo: custo = 18598.80
São Paulo → Santo André: custo = 18074.60
São Paulo → Guarulhos: custo = 12132.60
São Paulo → Campinas: custo = 2353.40
São Paulo → Alumínio: custo = 1656.40
São Paulo → cida: custo = 6656.40

☞ MENU

1. Mostrar Arquivo
 2. Mostrar Grafo
 3. Inserir Vértice
 4. Inserir Aresta
 5. Remover Vértice
 6. Remover Aresta
 7. Conexidade
 8. Salvar e Sair
 9. Sair
 10. Caminho Mínimo (Dijkstra)
 11. Verificar se é Euleriano
 12. Caminhos Mínimos entre Todos os Pares (Floyd-Warshall)
 13. Mostrar Grau dos Vértices
 14. Coloração de Vértices
- Escolha: 13

☒ Grau dos Vértices:

0 - In: 0, Out: 0
1 - In: 0, Out: 0
2 - In: 0, Out: 0
3 - In: 0, Out: 0
4 - In: 0, Out: 0
5 - In: 0, Out: 0
6 - In: 0, Out: 0
7 - In: 0, Out: 0
8 - In: 0, Out: 0
9 - In: 0, Out: 0
10 - In: 0, Out: 0
11 - In: 0, Out: 0
12 - In: 0, Out: 0
13 - In: 0, Out: 0
14 - In: 0, Out: 0
15 - In: 0, Out: 0
16 - In: 0, Out: 0
17 - In: 0, Out: 0
18 - In: 0, Out: 0
19 - In: 0, Out: 0
20 - In: 0, Out: 0
21 - In: 0, Out: 0
22 - In: 0, Out: 0

22	-	In:	0,	Out:	0
23	-	In:	0,	Out:	0
24	-	In:	0,	Out:	0
25	-	In:	1,	Out:	0
26	-	In:	2,	Out:	0
27	-	In:	3,	Out:	0
28	-	In:	4,	Out:	0
29	-	In:	5,	Out:	0
30	-	In:	5,	Out:	5
31	-	In:	5,	Out:	5
32	-	In:	5,	Out:	5
33	-	In:	5,	Out:	5
34	-	In:	5,	Out:	5
35	-	In:	5,	Out:	5
36	-	In:	5,	Out:	5
37	-	In:	5,	Out:	5
38	-	In:	5,	Out:	5
39	-	In:	5,	Out:	5
40	-	In:	5,	Out:	5
41	-	In:	5,	Out:	5
42	-	In:	5,	Out:	5
43	-	In:	5,	Out:	5
44	-	In:	5,	Out:	5
45	-	In:	5,	Out:	5
46	-	In:	5,	Out:	5
47	-	In:	5,	Out:	5
48	-	In:	5,	Out:	5
49	-	In:	5,	Out:	5
50	-	In:	5,	Out:	5
51	-	In:	5,	Out:	5
52	-	In:	5,	Out:	5
53	-	In:	5,	Out:	5
54	-	In:	5,	Out:	5
55	-	In:	4,	Out:	5
56	-	In:	3,	Out:	5
57	-	In:	2,	Out:	5
58	-	In:	1,	Out:	6
59	-	In:	0,	Out:	5
60	-	In:	1,	Out:	0

✧ MENU

1. Mostrar Arquivo
 2. Mostrar Grafo
 3. Inserir Vértice
 4. Inserir Aresta
 5. Remover Vértice
 6. Remover Aresta
 7. Conexidade
 8. Salvar e Sair
 9. Sair
 10. Caminho Mínimo (Dijkstra)
 11. Verificar se é Euleriano
 12. Caminhos Mínimos entre Todos os Pares (Floyd-Warshall)
 13. Mostrar Grau dos Vértices
 14. Coloração de Vértices
- Escolha: 14

☺ Coloração dos Vértices (Cluster Estimado):

Vértice 30 -> Cor 0
Vértice 31 -> Cor 1
Vértice 32 -> Cor 2
Vértice 33 -> Cor 3
Vértice 34 -> Cor 4
Vértice 35 -> Cor 5
Vértice 36 -> Cor 0
Vértice 37 -> Cor 1
Vértice 38 -> Cor 2
Vértice 39 -> Cor 3
Vértice 40 -> Cor 4
Vértice 41 -> Cor 5
Vértice 42 -> Cor 0
Vértice 43 -> Cor 1
Vértice 44 -> Cor 2
Vértice 45 -> Cor 3
Vértice 46 -> Cor 4
Vértice 47 -> Cor 5
Vértice 48 -> Cor 0
Vértice 49 -> Cor 1
Vértice 50 -> Cor 2
Vértice 51 -> Cor 3
Vértice 52 -> Cor 4

```
Vértice 52 -> Cor 4
Vértice 53 -> Cor 5
Vértice 54 -> Cor 0
Vértice 55 -> Cor 1
Vértice 56 -> Cor 2
Vértice 57 -> Cor 3
Vértice 58 -> Cor 4
Vértice 29 -> Cor 5
Vértice 59 -> Cor 5
Vértice 28 -> Cor 4
Vértice 27 -> Cor 3
Vértice 26 -> Cor 2
Vértice 25 -> Cor 1
Vértice 60 -> Cor 0
Vértice 0 -> Cor 0
Vértice 1 -> Cor 0
Vértice 2 -> Cor 0
Vértice 3 -> Cor 0
Vértice 4 -> Cor 0
Vértice 5 -> Cor 0
Vértice 6 -> Cor 0
Vértice 7 -> Cor 0
Vértice 8 -> Cor 0
Vértice 9 -> Cor 0
Vértice 10 -> Cor 0
Vértice 11 -> Cor 0
Vértice 12 -> Cor 0
Vértice 13 -> Cor 0
Vértice 14 -> Cor 0
Vértice 15 -> Cor 0
Vértice 16 -> Cor 0
Vértice 17 -> Cor 0
Vértice 18 -> Cor 0
Vértice 19 -> Cor 0
Vértice 20 -> Cor 0
Vértice 21 -> Cor 0
Vértice 22 -> Cor 0
Vértice 23 -> Cor 0
Vértice 24 -> Cor 0
```

APÊNDICE A - GITHUB / FOTO GRAFO

Link para o GitHub do projeto:

<https://github.com/GlendaTrigo/ProjetoGrafosEnergia>

Link para visualização do Grafo em melhor qualidade:

 [krwPUmgCXmjNIIHT.png](#)

Link para o vídeo do projeto:

<https://www.youtube.com/watch?v=W4yn-SeLMa0>