



Relatório do projeto

Nome do(a) aluno(a)	RA
Enzo Guarnieri	10410074
Erika Borges Piaui	10403716
Júlia Campolim de Oste	10408802

MackMap

- **Definição do projeto**

O objetivo deste projeto é criar um website, com foco na adaptabilidade para dispositivos móveis, que disponibiliza um mapa com a visão aérea do campus Higienópolis da Universidade Presbiteriana Mackenzie. Este mapa deve conter os principais pontos de referência da Instituição, como os prédios, lanchonetes e quadras.

Além da visão aérea, a aplicação também disponibilizará uma ferramenta de navegação para ajudar as pessoas a se locomover dentro do campus; nesta ferramenta, os usuários serão capazes de selecionar dois locais dentro da Universidade e encontrar o melhor caminho entre eles. Sendo que, o melhor caminho pode ser tanto o caminho mais curto quanto o caminho mais acessível (que evita escadas, por exemplo).

Cada local do mapa terá algumas informações básicas sobre ele, como seu nome e suas principais funcionalidades (salas de aula, laboratórios, alimentação, esportes, bibliotecas, etc.). Em adição, cada local terá uma sessão especial para informar se ele possui algum ponto de coleta de material reciclável, como a coleta de eletrônicos localizada no prédio 31 da Faculdade de Computação e Informática.

Dessa forma, a aplicação deverá facilitar a locomoção de pessoas pelo ambiente universitário, principalmente pessoas que ainda não possuem familiaridade com o campus, e irá promover o descarte sustentável de materiais recicláveis, que muitas vezes não são descartados corretamente.

Para mapear os locais do campus, o grupo optou pelo uso de um grafo não direcionado e ponderado nas arestas, onde os vértices representam os prédios dentro da Universidade e as arestas o caminho entre eles. Para uma modelagem inicial do problema, foi utilizado a ferramenta Graph Online, onde os vértices foram posicionados de modo semelhante ao seu posicionamento na vida real.

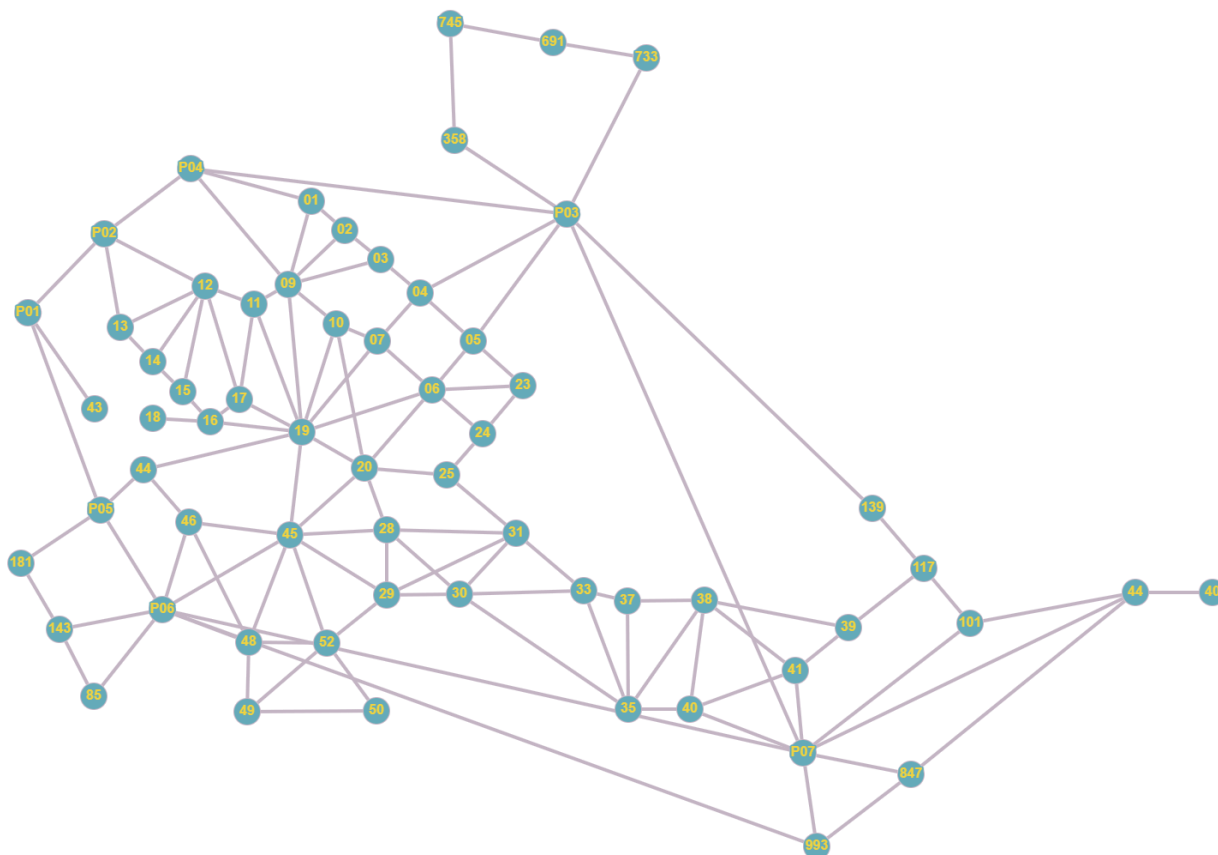


Figura 1: Protótipo do grafo. Link: <http://graphonline.top/?graph=MbVhwpCLhYabzWjk>.

Este grafo foi usado como base para a coleta de distâncias entre os locais do campus.

Nos vértices do grafo, o grupo optou por armazenar apenas o rótulo de cada vértice (número de cada prédio) e nas arestas está armazenado se o caminho é acessível ou não. Outras informações foram coletadas durante o desenvolvimento do projeto, como nome dos prédios, pontos de coleta de materiais recicláveis presentes e endereço. Porém, pelo caráter multidisciplinar do projeto, essas informações estão sendo armazenadas em um banco de dados que não possui relação com a parte de grafos do MackMap.

Dessa forma, o grupo optou por armazenar no grafo apenas as informações relevantes para o algoritmo implementado.

Com base no problema descrito, o grupo analisou como os algoritmos estudados em sala poderiam contribuir para a construção do MackMap. A conclusão foi:

- O algoritmo de **Dijkstra** pode ser usado para calcular o menor percurso entre dois locais do mapa.



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira

Disciplina: Teoria dos Grafos

- O algoritmo de **Prim** pode ser usado para encontrar uma árvore de custo mínimo do campus, útil se o objetivo é ligar todos os prédios gastando o mínimo possível (por exemplo, passar um cabo de rede por todos os prédios).
- O algoritmo de **coloração sequencial** pode ser usado para representar o número mínimo de partições desconexas. No MackMap isso significaria, por exemplo, a representação dos pontos de coleta de materiais recicláveis.

Link do Github: <https://github.com/jcampolim/grafos>.

Link da apresentação no Youtube: <https://youtu.be/qcjlIn6ZOw>.

● **Objetivos da ODS**

Um dos objetivos do projeto MackMap é disponibilizar informações de pontos de coleta de materiais recicláveis dentro do campus Higienópolis da Universidade Presbiteriana Mackenzie, como o ponto de coleta de eletrônicos localizado dentro do prédio 31 ou o ponto de coleta de esmaltes (extremamente poluentes se descartados da forma errada) na portaria da Piauí. Portanto, para o projeto, podemos associar a ODS 12 - Assegurar padrões de produção e consumo sustentáveis.

● **Etapas de desenvolvimento**

O desenvolvimento do projeto ocorreu em duas etapas principais: coleta de dados e desenvolvimento dos algoritmos.

A coleta dos dados ocorreu durante todo o processo de desenvolvimento do projeto e envolveu a coleta de: prédios pertencentes ao campus Higienópolis, seu nome, localização, se possui ponto de coleta de recicláveis e se é acessível. Para as arestas, coletamos a distância entre cada vértice de forma manual utilizando o Google Maps.

O desenvolvimento dos algoritmos ocorreu de forma simultânea com a apresentação deles em sala de aula.



- **Grafo gerado**

O grafo abaixo foi gerado automaticamente pelo programa a partir do arquivo com as informações dos vértices e arestas.

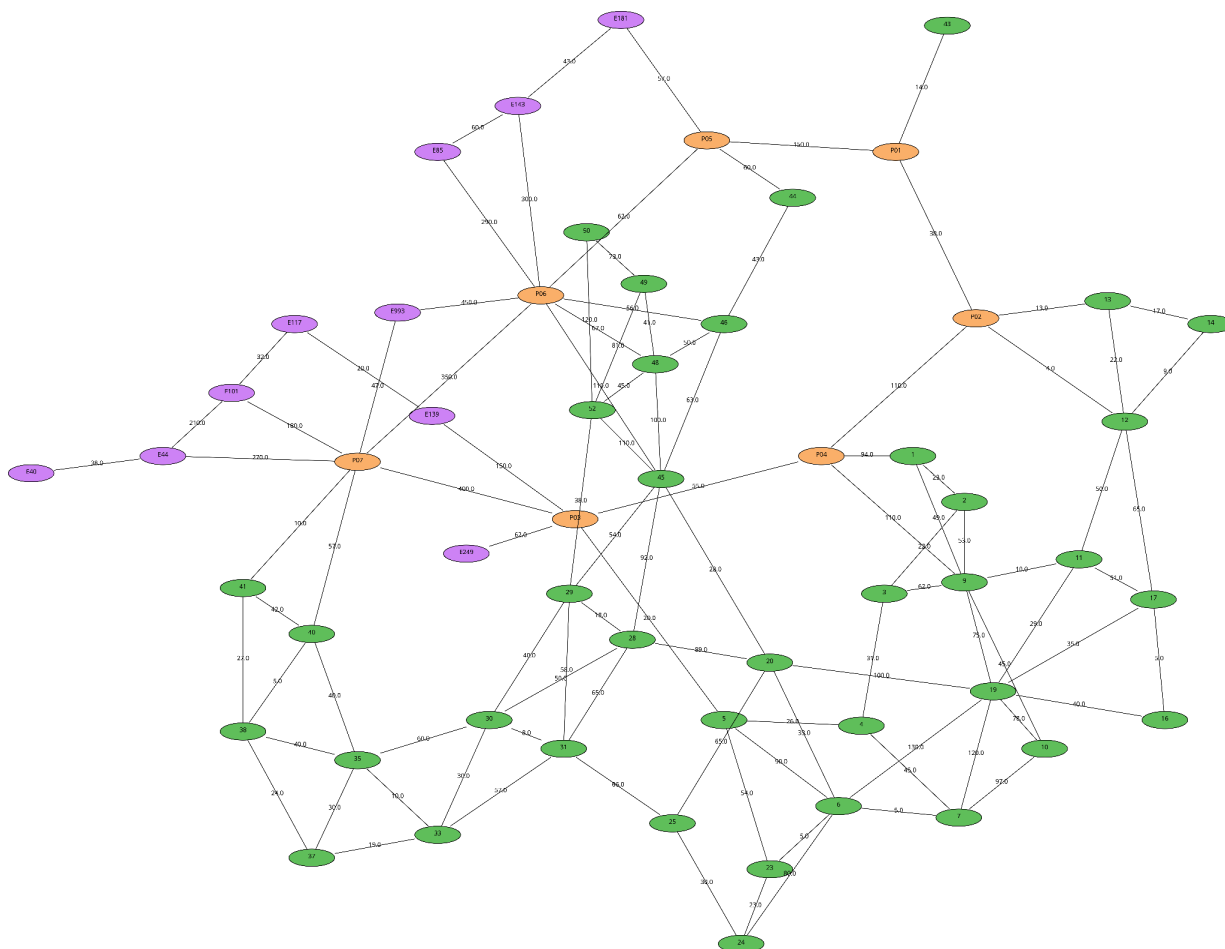


Figura 2: Grafo gerado

Nele, as cores representam:

- Verde: edifícios internos;
- Laranja: portarias;
- Roxo: edifícios internos.

- **Mapa final**

Abaixo está a visualização do mapa para o projeto final da aplicação desenvolvida com a utilização da biblioteca JGraphX.



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos



Figura 3: Mapa final

Nele, as cores representam:

- **Laranja:** Educação Básica;
- **Amarelo:** História, Tradição e Cultura;
- **Azul:** Pioneirismo e Inovação;
- **Vermelho:** Institucional;
- **Verde:** Tecnologia e Saúde.

Já os rótulos representam:

- **LXX:** edifícios internos;
- **EXXX:** edifícios externos.

Por fim, os ícones representam:

- **Rosa:** sanitários masculinos e femininos;
- **Amarelo:** estação de metrô Higienópolis-Mackenzie.



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

Além disso, para cada edifício selecionado pelo mapa, uma tela com informações complementares aparecerá como a seguir:

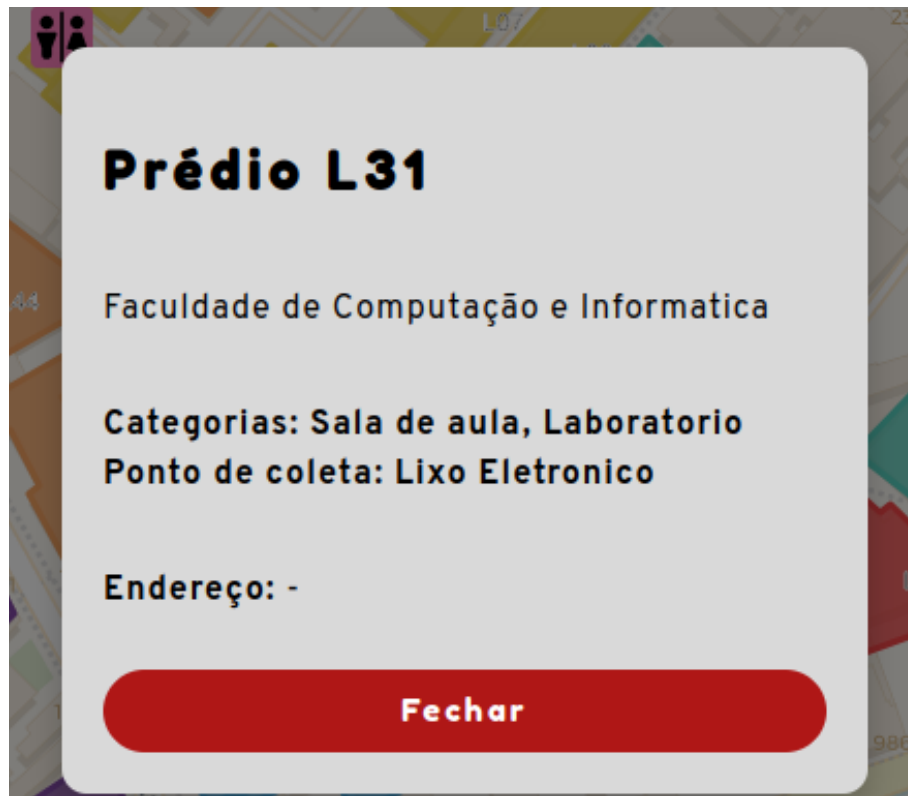


Figura 4: Expandir informações de locais

- **Execução (Parte I)**

Para facilitar na visualização dos vértices e arestas, o grupo optou por implementar os métodos `exibirVertices` (opção “*h”) e `exibirArestas` (opção “**h”) no menu.

- Ler dados do arquivo **grafo.txt**.



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
===== MENU DE OPÇÕES =====  
a. Ler dados do arquivo 'grafo.txt'  
b. Gravar dados no arquivo 'grafo.txt'  
c. Inserir vértice;  
d. Inserir aresta;  
e. Remover vértice;  
f. Remover aresta;  
g. Mostrar conteúdo do arquivo;  
h. Mostrar grafo;  
*h. Mostrar apenas vértices do grafo;  
**h. Mostrar apenas arestas do grafo;  
i. Apresentar conexidade;  
j. Encerrar a aplicação.  
=====
```

> Selecione uma opção: **a**

> Grafo lido com sucesso!

- Inserir vértice.



```
===== MENU DE OPÇÕES =====  
a. Ler dados do arquivo 'grafo.txt'  
b. Gravar dados no arquivo 'grafo.txt'  
c. Inserir vértice;  
d. Inserir aresta;  
e. Remover vértice;  
f. Remover aresta;  
g. Mostrar conteúdo do arquivo;  
h. Mostrar grafo;  
*h. Mostrar apenas vértices do grafo;  
**h. Mostrar apenas arestas do grafo;  
i. Apresentar conexidade;  
j. Encerrar a aplicação.  
=====
```

> Selecione uma opção: **c**

> Insira o rótulo do vértice: **34**

> Vértice inserido com sucesso!

```
===== MENU DE OPÇÕES =====  
a. Ler dados do arquivo 'grafo.txt'  
b. Gravar dados no arquivo 'grafo.txt'  
c. Inserir vértice;  
d. Inserir aresta;  
e. Remover vértice;  
f. Remover aresta;  
g. Mostrar conteúdo do arquivo;  
h. Mostrar grafo;  
*h. Mostrar apenas vértices do grafo;  
**h. Mostrar apenas arestas do grafo;  
i. Apresentar conexidade;  
j. Encerrar a aplicação.  
=====
```




UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
> Selecione uma opção: *h

> Início da impressão dos vértices (n = 61):
1
2
3
4
5
6
7
9
10
11
12
13
14
16
17
19
20
23
24
25
28
29
30
31
33
35
37
38
39
40
41
43
44
45
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
46
48
49
50
52
E101
E117
E139
E143
E181
E358
E40
E44
E691
E733
E745
E847
E85
E993
P01
P02
P03
P04
P05
P06
P07
34
> Fim da impressão dos vértices.
```



```
===== MENU DE OPÇÕES =====  
a. Ler dados do arquivo 'grafo.txt'  
b. Gravar dados no arquivo 'grafo.txt'  
c. Inserir vértice;  
d. Inserir aresta;  
e. Remover vértice;  
f. Remover aresta;  
g. Mostrar conteúdo do arquivo;  
h. Mostrar grafo;  
*h. Mostrar apenas vértices do grafo;  
**h. Mostrar apenas arestas do grafo;  
i. Apresentar conexidade;  
j. Encerrar a aplicação.  
=====
```

> Selecione uma opção: **c**

> Insira o rótulo do vértice: **51**

> Vértice inserido com sucesso!

```
===== MENU DE OPÇÕES =====  
a. Ler dados do arquivo 'grafo.txt'  
b. Gravar dados no arquivo 'grafo.txt'  
c. Inserir vértice;  
d. Inserir aresta;  
e. Remover vértice;  
f. Remover aresta;  
g. Mostrar conteúdo do arquivo;  
h. Mostrar grafo;  
*h. Mostrar apenas vértices do grafo;  
**h. Mostrar apenas arestas do grafo;  
i. Apresentar conexidade;  
j. Encerrar a aplicação.  
=====
```

> Selecione uma opção: ***h**



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
> Início da impressão dos vértices (n = 62):
```

```
1  
2  
3  
4  
5  
6  
7  
9  
10  
11  
12  
13  
14  
16  
17  
19  
20  
23  
24  
25  
28  
29  
30  
31  
33  
35  
37  
38  
39  
40  
41  
43  
44  
45  
46  
48  
49
```



```
49
50
52
E101
E117
E139
E143
E181
E358
E40
E44
E691
E733
E745
E847
E85
E993
P01
P02
P03
P04
P05
P06
P07
34
51
> Fim da impressão dos vértices.
```

- Inserir aresta.



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
===== MENU DE OPÇÕES =====
a. Ler dados do arquivo 'grafo.txt'
b. Gravar dados no arquivo 'grafo.txt'
c. Inserir vértice;
d. Inserir aresta;
e. Remover vértice;
f. Remover aresta;
g. Mostrar conteúdo do arquivo;
h. Mostrar grafo;
*h. Mostrar apenas vértices do grafo;
**h. Mostrar apenas arestas do grafo;
i. Apresentar conexidade;
j. Encerrar a aplicação.
=====

> Selecione uma opção: d

> Insira o primeiro vértice: 34
> Insira o segundo vértice: 35
> Insira o peso da aresta: 12
> Aresta adicionada com sucesso!

===== MENU DE OPÇÕES =====
a. Ler dados do arquivo 'grafo.txt'
b. Gravar dados no arquivo 'grafo.txt'
c. Inserir vértice;
d. Inserir aresta;
e. Remover vértice;
f. Remover aresta;
g. Mostrar conteúdo do arquivo;
h. Mostrar grafo;
*h. Mostrar apenas vértices do grafo;
**h. Mostrar apenas arestas do grafo;
i. Apresentar conexidade;
j. Encerrar a aplicação.
=====

> Selecione uma opção: **h
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
> Início da impressão das arestas (m = 114):
```

```
2 -- 1 = 23.0  
3 -- 2 = 22.0  
4 -- 3 = 31.0  
5 -- 4 = 26.0  
6 -- 5 = 90.0  
7 -- 4 = 45.0  
7 -- 6 = 5.0  
9 -- 1 = 49.0  
9 -- 2 = 53.0  
9 -- 3 = 62.0  
10 -- 7 = 97.0  
10 -- 9 = 45.0  
11 -- 9 = 10.0  
12 -- 11 = 50.0  
13 -- 12 = 22.0  
14 -- 12 = 9.0  
14 -- 13 = 17.0  
17 -- 11 = 51.0  
17 -- 12 = 65.0  
17 -- 16 = 5.0  
19 -- 6 = 130.0  
19 -- 7 = 120.0  
19 -- 9 = 75.0  
19 -- 10 = 78.0  
19 -- 11 = 29.0  
19 -- 16 = 40.0  
19 -- 17 = 35.0  
20 -- 6 = 33.0  
20 -- 19 = 100.0  
23 -- 5 = 54.0  
23 -- 6 = 5.0  
24 -- 6 = 80.0  
24 -- 23 = 23.0  
25 -- 20 = 65.0  
25 -- 24 = 30.0  
28 -- 20 = 89.0
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
28 -- 20 = 89.0
29 -- 28 = 18.0
30 -- 28 = 50.0
30 -- 29 = 40.0
31 -- 25 = 66.0
31 -- 28 = 65.0
31 -- 29 = 58.0
31 -- 30 = 8.0
33 -- 30 = 30.0
33 -- 31 = 57.0
35 -- 30 = 60.0
35 -- 33 = 10.0
37 -- 33 = 19.0
37 -- 35 = 30.0
38 -- 35 = 40.0
38 -- 37 = 24.0
39 -- 38 = 43.0
40 -- 35 = 40.0
40 -- 38 = 5.0
41 -- 38 = 27.0
41 -- 39 = 10.0
41 -- 40 = 42.0
45 -- 20 = 28.0
45 -- 28 = 92.0
45 -- 29 = 54.0
46 -- 44 = 43.0
46 -- 45 = 63.0
48 -- 45 = 100.0
48 -- 46 = 50.0
49 -- 48 = 41.0
50 -- 49 = 73.0
52 -- 29 = 38.0
52 -- 45 = 110.0
52 -- 48 = 45.0
52 -- 49 = 81.0
52 -- 50 = 120.0
E117 -- 39 = 210.0
```




UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
E117 -- 39 = 210.0
E117 -- E101 = 32.0
E139 -- E117 = 20.0
E181 -- E143 = 43.0
E44 -- E101 = 210.0
E44 -- E40 = 38.0
E733 -- E691 = 46.0
E745 -- E358 = 160.0
E847 -- E44 = 200.0
E85 -- E143 = 60.0
E993 -- E847 = 83.0
P01 -- 43 = 14.0
P02 -- 12 = 4.0
P02 -- 13 = 13.0
P02 -- P01 = 38.0
P03 -- 4 = 55.0
P03 -- 5 = 20.0
P03 -- E139 = 150.0
P03 -- E358 = 66.0
P03 -- E733 = 240.0
P04 -- 1 = 94.0
P04 -- 9 = 110.0
P04 -- P02 = 110.0
P04 -- P03 = 55.0
P05 -- 44 = 60.0
P05 -- E181 = 57.0
P05 -- P01 = 150.0
P06 -- 45 = 110.0
P06 -- 46 = 56.0
P06 -- 48 = 67.0
P06 -- E143 = 300.0
P06 -- E85 = 290.0
P06 -- E993 = 450.0
P06 -- P05 = 62.0
P07 -- 40 = 57.0
P07 -- 41 = 10.0
P07 -- E101 = 180.0
P07 -- E44 = 270.0
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
P07 -- E44 = 270.0
P07 -- E847 = 120.0
P07 -- E993 = 47.0
P07 -- P03 = 400.0
P07 -- P06 = 350.0
34 -- 35 = 12.0
> Fim da impressão das arestas.
```

```
===== MENU DE OPÇÕES =====
a. Ler dados do arquivo 'grafo.txt'
b. Gravar dados no arquivo 'grafo.txt'
c. Inserir vértice;
d. Inserir aresta;
e. Remover vértice;
f. Remover aresta;
g. Mostrar conteúdo do arquivo;
h. Mostrar grafo;
*h. Mostrar apenas vértices do grafo;
**h. Mostrar apenas arestas do grafo;
i. Apresentar conexidade;
j. Encerrar a aplicação.
=====

> Selecione uma opção: d

> Insira o primeiro vértice: 51
> Insira o segundo vértice: 52
> Insira o peso da aresta: 234
> Aresta adicionada com sucesso!

===== MENU DE OPÇÕES =====
a. Ler dados do arquivo 'grafo.txt'
b. Gravar dados no arquivo 'grafo.txt'
c. Inserir vértice;
d. Inserir aresta;
e. Remover vértice;
f. Remover aresta;
g. Mostrar conteúdo do arquivo;
h. Mostrar grafo;
*h. Mostrar apenas vértices do grafo;
**h. Mostrar apenas arestas do grafo;
i. Apresentar conexidade;
j. Encerrar a aplicação.
=====

> Selecione uma opção: **h
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
> Início da impressão das arestas (m = 115):
```

```
2 -- 1 = 23.0  
3 -- 2 = 22.0  
4 -- 3 = 31.0  
5 -- 4 = 26.0  
6 -- 5 = 90.0  
7 -- 4 = 45.0  
7 -- 6 = 5.0  
9 -- 1 = 49.0  
9 -- 2 = 53.0  
9 -- 3 = 62.0  
10 -- 7 = 97.0  
10 -- 9 = 45.0  
11 -- 9 = 10.0  
12 -- 11 = 50.0  
13 -- 12 = 22.0  
14 -- 12 = 9.0  
14 -- 13 = 17.0  
17 -- 11 = 51.0  
17 -- 12 = 65.0  
17 -- 16 = 5.0  
19 -- 6 = 130.0  
19 -- 7 = 120.0  
19 -- 9 = 75.0  
19 -- 10 = 78.0  
19 -- 11 = 29.0  
19 -- 16 = 40.0  
19 -- 17 = 35.0  
20 -- 6 = 33.0  
20 -- 19 = 100.0  
23 -- 5 = 54.0  
23 -- 6 = 5.0  
24 -- 6 = 80.0  
24 -- 23 = 23.0  
25 -- 20 = 65.0  
25 -- 24 = 30.0  
28 -- 20 = 89.0  
29 -- 28 = 18.0
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
29 -- 28 = 18.0
30 -- 28 = 50.0
30 -- 29 = 40.0
31 -- 25 = 66.0
31 -- 28 = 65.0
31 -- 29 = 58.0
31 -- 30 = 8.0
33 -- 30 = 30.0
33 -- 31 = 57.0
35 -- 30 = 60.0
35 -- 33 = 10.0
37 -- 33 = 19.0
37 -- 35 = 30.0
38 -- 35 = 40.0
38 -- 37 = 24.0
39 -- 38 = 43.0
40 -- 35 = 40.0
40 -- 38 = 5.0
41 -- 38 = 27.0
41 -- 39 = 10.0
41 -- 40 = 42.0
45 -- 20 = 28.0
45 -- 28 = 92.0
45 -- 29 = 54.0
46 -- 44 = 43.0
46 -- 45 = 63.0
48 -- 45 = 100.0
48 -- 46 = 50.0
49 -- 48 = 41.0
50 -- 49 = 73.0
52 -- 29 = 38.0
52 -- 45 = 110.0
52 -- 48 = 45.0
52 -- 49 = 81.0
52 -- 50 = 120.0
E117 -- 39 = 210.0
E117 -- E101 = 32.0
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
E117 -- 39 = 210.0
E117 -- E101 = 32.0
E139 -- E117 = 20.0
E181 -- E143 = 43.0
E44 -- E101 = 210.0
E44 -- E40 = 38.0
E733 -- E691 = 46.0
E745 -- E358 = 160.0
E847 -- E44 = 200.0
E85 -- E143 = 60.0
E993 -- E847 = 83.0
P01 -- 43 = 14.0
P02 -- 12 = 4.0
P02 -- 13 = 13.0
P02 -- P01 = 38.0
P03 -- 4 = 55.0
P03 -- 5 = 20.0
P03 -- E139 = 150.0
P03 -- E358 = 66.0
P03 -- E733 = 240.0
P04 -- 1 = 94.0
P04 -- 9 = 110.0
P04 -- P02 = 110.0
P04 -- P03 = 55.0
P05 -- 44 = 60.0
P05 -- E181 = 57.0
P05 -- P01 = 150.0
P06 -- 45 = 110.0
P06 -- 46 = 56.0
P06 -- 48 = 67.0
P06 -- E143 = 300.0
P06 -- E85 = 290.0
P06 -- E993 = 450.0
P06 -- P05 = 62.0
P07 -- 40 = 57.0
P07 -- 41 = 10.0
P07 -- E101 = 180.0
P07 -- E44 = 270.0
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
P07 -- E44 = 270.0  
P07 -- E847 = 120.0  
P07 -- E993 = 47.0  
P07 -- P03 = 400.0  
P07 -- P06 = 350.0  
34 -- 35 = 12.0  
51 -- 52 = 234.0  
> Fim da impressão das arestas.
```

- Remover vértice.



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
===== MENU DE OPÇÕES =====  
a. Ler dados do arquivo 'grafo.txt'  
b. Gravar dados no arquivo 'grafo.txt'  
c. Inserir vértice;  
d. Inserir aresta;  
e. Remover vértice;  
f. Remover aresta;  
g. Mostrar conteúdo do arquivo;  
h. Mostrar grafo;  
*h. Mostrar apenas vértices do grafo;  
**h. Mostrar apenas arestas do grafo;  
i. Apresentar conexidade;  
j. Encerrar a aplicação.  
=====
```

> Selecione uma opção: *e*

> Insira o vértice: *1*

> Vértice removido com sucesso!

```
===== MENU DE OPÇÕES =====  
a. Ler dados do arquivo 'grafo.txt'  
b. Gravar dados no arquivo 'grafo.txt'  
c. Inserir vértice;  
d. Inserir aresta;  
e. Remover vértice;  
f. Remover aresta;  
g. Mostrar conteúdo do arquivo;  
h. Mostrar grafo;  
*h. Mostrar apenas vértices do grafo;  
**h. Mostrar apenas arestas do grafo;  
i. Apresentar conexidade;  
j. Encerrar a aplicação.  
=====
```

> Selecione uma opção: **h*



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
> Início da impressão dos vértices (n = 61):
```

```
2  
3  
4  
5  
6  
7  
9  
10  
11  
12  
13  
14  
16  
17  
19  
20  
23  
24  
25  
28  
29  
30  
31  
33  
35  
37  
38  
39  
40  
41  
43  
44  
45  
46  
48
```




UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
48
49
50
52
E101
E117
E139
E143
E181
E358
E40
E44
E691
E733
E745
E847
E85
E993
P01
P02
P03
P04
P05
P06
P07
34
51
> Fim da impressão dos vértices.
```



```
===== MENU DE OPÇÕES =====  
a. Ler dados do arquivo 'grafo.txt'  
b. Gravar dados no arquivo 'grafo.txt'  
c. Inserir vértice;  
d. Inserir aresta;  
e. Remover vértice;  
f. Remover aresta;  
g. Mostrar conteúdo do arquivo;  
h. Mostrar grafo;  
*h. Mostrar apenas vértices do grafo;  
**h. Mostrar apenas arestas do grafo;  
i. Apresentar conexidade;  
j. Encerrar a aplicação.  
=====
```

> Selecione uma opção: *e*

> Insira o vértice: *E40*

> Vértice removido com sucesso!

```
===== MENU DE OPÇÕES =====  
a. Ler dados do arquivo 'grafo.txt'  
b. Gravar dados no arquivo 'grafo.txt'  
c. Inserir vértice;  
d. Inserir aresta;  
e. Remover vértice;  
f. Remover aresta;  
g. Mostrar conteúdo do arquivo;  
h. Mostrar grafo;  
*h. Mostrar apenas vértices do grafo;  
**h. Mostrar apenas arestas do grafo;  
i. Apresentar conexidade;  
j. Encerrar a aplicação.  
=====
```

> Selecione uma opção: **h*



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
> Início da impressão dos vértices (n = 60):
```

```
2  
3  
4  
5  
6  
7  
9  
10  
11  
12  
13  
14  
16  
17  
19  
20  
23  
24  
25  
28  
29  
30  
31  
33  
35  
37  
38  
39  
40  
41  
43  
44  
45  
46  
48  
49
```



```
49
50
52
E101
E117
E139
E143
E181
E358
E44
E691
E733
E745
E847
E85
E993
P01
P02
P03
P04
P05
P06
P07
34
51
> Fim da impressão dos vértices.
```

- Remover aresta.



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
===== MENU DE OPÇÕES =====  
a. Ler dados do arquivo 'grafo.txt'  
b. Gravar dados no arquivo 'grafo.txt'  
c. Inserir vértice;  
d. Inserir aresta;  
e. Remover vértice;  
f. Remover aresta;  
g. Mostrar conteúdo do arquivo;  
h. Mostrar grafo;  
*h. Mostrar apenas vértices do grafo;  
**h. Mostrar apenas arestas do grafo;  
i. Apresentar conexidade;  
j. Encerrar a aplicação.  
=====
```

> Selecione uma opção: *f*

> Insira o primeiro vértice: *P04*
> Insira o segundo vértice: *P03*
> Aresta removida com sucesso!

```
===== MENU DE OPÇÕES =====  
a. Ler dados do arquivo 'grafo.txt'  
b. Gravar dados no arquivo 'grafo.txt'  
c. Inserir vértice;  
d. Inserir aresta;  
e. Remover vértice;  
f. Remover aresta;  
g. Mostrar conteúdo do arquivo;  
h. Mostrar grafo;  
*h. Mostrar apenas vértices do grafo;  
**h. Mostrar apenas arestas do grafo;  
i. Apresentar conexidade;  
j. Encerrar a aplicação.  
=====
```

> Selecione uma opção: ***h*



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
> Início da impressão das arestas (m = 110):
```

```
3 -- 2 = 22.0  
4 -- 3 = 31.0  
5 -- 4 = 26.0  
6 -- 5 = 90.0  
7 -- 4 = 45.0  
7 -- 6 = 5.0  
9 -- 2 = 53.0  
9 -- 3 = 62.0  
10 -- 7 = 97.0  
10 -- 9 = 45.0  
11 -- 9 = 10.0  
12 -- 11 = 50.0  
13 -- 12 = 22.0  
14 -- 12 = 9.0  
14 -- 13 = 17.0  
17 -- 11 = 51.0  
17 -- 12 = 65.0  
17 -- 16 = 5.0  
19 -- 6 = 130.0  
19 -- 7 = 120.0  
19 -- 9 = 75.0  
19 -- 10 = 78.0  
19 -- 11 = 29.0  
19 -- 16 = 40.0  
19 -- 17 = 35.0  
20 -- 6 = 33.0  
20 -- 19 = 100.0  
23 -- 5 = 54.0  
23 -- 6 = 5.0  
24 -- 6 = 80.0  
24 -- 23 = 23.0  
25 -- 20 = 65.0  
25 -- 24 = 30.0  
28 -- 20 = 89.0  
29 -- 28 = 18.0  
30 -- 28 = 50.0  
30 -- 29 = 40.0
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
30 -- 29 = 40.0
31 -- 25 = 66.0
31 -- 28 = 65.0
31 -- 29 = 58.0
31 -- 30 = 8.0
33 -- 30 = 30.0
33 -- 31 = 57.0
35 -- 30 = 60.0
35 -- 33 = 10.0
37 -- 33 = 19.0
37 -- 35 = 30.0
38 -- 35 = 40.0
38 -- 37 = 24.0
39 -- 38 = 43.0
40 -- 35 = 40.0
40 -- 38 = 5.0
41 -- 38 = 27.0
41 -- 39 = 10.0
41 -- 40 = 42.0
45 -- 20 = 28.0
45 -- 28 = 92.0
45 -- 29 = 54.0
46 -- 44 = 43.0
46 -- 45 = 63.0
48 -- 45 = 100.0
48 -- 46 = 50.0
49 -- 48 = 41.0
50 -- 49 = 73.0
52 -- 29 = 38.0
52 -- 45 = 110.0
52 -- 48 = 45.0
52 -- 49 = 81.0
52 -- 50 = 120.0
E117 -- 39 = 210.0
E117 -- E101 = 32.0
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
E117 -- E101 = 32.0
E139 -- E117 = 20.0
E181 -- E143 = 43.0
E44 -- E101 = 210.0
E733 -- E691 = 46.0
E745 -- E358 = 160.0
E847 -- E44 = 200.0
E85 -- E143 = 60.0
E993 -- E847 = 83.0
P01 -- 43 = 14.0
P02 -- 12 = 4.0
P02 -- 13 = 13.0
P02 -- P01 = 38.0
P03 -- 4 = 55.0
P03 -- 5 = 20.0
P03 -- E139 = 150.0
P03 -- E358 = 66.0
P03 -- E733 = 240.0
P04 -- 9 = 110.0
P04 -- P02 = 110.0
P05 -- 44 = 60.0
P05 -- E181 = 57.0
P05 -- P01 = 150.0
P06 -- 45 = 110.0
P06 -- 46 = 56.0
P06 -- 48 = 67.0
P06 -- E143 = 300.0
P06 -- E85 = 290.0
P06 -- E993 = 450.0
P06 -- P05 = 62.0
P07 -- 40 = 57.0
P07 -- 41 = 10.0
P07 -- E101 = 180.0
P07 -- E44 = 270.0
P07 -- E847 = 120.0
P07 -- E993 = 47.0
P07 -- P03 = 400.0
```

```
P07 -- P03 = 400.0
P07 -- P06 = 350.0
34 -- 35 = 12.0
51 -- 52 = 234.0
> Fim da impressão das arestas.
```




UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
===== MENU DE OPÇÕES =====
a. Ler dados do arquivo 'grafo.txt'
b. Gravar dados no arquivo 'grafo.txt'
c. Inserir vértice;
d. Inserir aresta;
e. Remover vértice;
f. Remover aresta;
g. Mostrar conteúdo do arquivo;
h. Mostrar grafo;
*h. Mostrar apenas vértices do grafo;
**h. Mostrar apenas arestas do grafo;
i. Apresentar conexidade;
j. Encerrar a aplicação.
=====

> Selecione uma opção: f

> Insira o primeiro vértice: 37
> Insira o segundo vértice: 35
> Aresta removida com sucesso!

===== MENU DE OPÇÕES =====
a. Ler dados do arquivo 'grafo.txt'
b. Gravar dados no arquivo 'grafo.txt'
c. Inserir vértice;
d. Inserir aresta;
e. Remover vértice;
f. Remover aresta;
g. Mostrar conteúdo do arquivo;
h. Mostrar grafo;
*h. Mostrar apenas vértices do grafo;
**h. Mostrar apenas arestas do grafo;
i. Apresentar conexidade;
j. Encerrar a aplicação.
=====

> Selecione uma opção: **h
```



```
> Início da impressão das arestas (m = 109):
```

```
3 -- 2 = 22.0  
4 -- 3 = 31.0  
5 -- 4 = 26.0  
6 -- 5 = 90.0  
7 -- 4 = 45.0  
7 -- 6 = 5.0  
9 -- 2 = 53.0  
9 -- 3 = 62.0  
10 -- 7 = 97.0  
10 -- 9 = 45.0  
11 -- 9 = 10.0  
12 -- 11 = 50.0  
13 -- 12 = 22.0  
14 -- 12 = 9.0  
14 -- 13 = 17.0  
17 -- 11 = 51.0  
17 -- 12 = 65.0  
17 -- 16 = 5.0  
19 -- 6 = 130.0  
19 -- 7 = 120.0  
19 -- 9 = 75.0  
19 -- 10 = 78.0  
19 -- 11 = 29.0  
19 -- 16 = 40.0  
19 -- 17 = 35.0  
20 -- 6 = 33.0  
20 -- 19 = 100.0  
23 -- 5 = 54.0  
23 -- 6 = 5.0  
24 -- 6 = 80.0  
24 -- 23 = 23.0  
25 -- 20 = 65.0  
25 -- 24 = 30.0  
28 -- 20 = 89.0  
29 -- 28 = 18.0  
30 -- 28 = 50.0  
30 -- 29 = 40.0
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
30 -- 29 = 40.0
31 -- 25 = 66.0
31 -- 28 = 65.0
31 -- 29 = 58.0
31 -- 30 = 8.0
33 -- 30 = 30.0
33 -- 31 = 57.0
35 -- 30 = 60.0
35 -- 33 = 10.0
37 -- 33 = 19.0
38 -- 35 = 40.0
38 -- 37 = 24.0
39 -- 38 = 43.0
40 -- 35 = 40.0
40 -- 38 = 5.0
41 -- 38 = 27.0
41 -- 39 = 10.0
41 -- 40 = 42.0
45 -- 20 = 28.0
45 -- 28 = 92.0
45 -- 29 = 54.0
46 -- 44 = 43.0
46 -- 45 = 63.0
48 -- 45 = 100.0
48 -- 46 = 50.0
49 -- 48 = 41.0
50 -- 49 = 73.0
52 -- 29 = 38.0
52 -- 45 = 110.0
52 -- 48 = 45.0
52 -- 49 = 81.0
52 -- 50 = 120.0
E117 -- 39 = 210.0
E117 -- E101 = 32.0
E139 -- E117 = 20.0
E181 -- E143 = 43.0
E44 -- E101 = 210.0
```



```
E44 -- E101 = 210.0
E733 -- E691 = 46.0
E745 -- E358 = 160.0
E847 -- E44 = 200.0
E85 -- E143 = 60.0
E993 -- E847 = 83.0
P01 -- 43 = 14.0
P02 -- 12 = 4.0
P02 -- 13 = 13.0
P02 -- P01 = 38.0
P03 -- 4 = 55.0
P03 -- 5 = 20.0
P03 -- E139 = 150.0
P03 -- E358 = 66.0
P03 -- E733 = 240.0
P04 -- 9 = 110.0
P04 -- P02 = 110.0
P05 -- 44 = 60.0
P05 -- E181 = 57.0
P05 -- P01 = 150.0
P06 -- 45 = 110.0
P06 -- 46 = 56.0
P06 -- 48 = 67.0
P06 -- E143 = 300.0
P06 -- E85 = 290.0
P06 -- E993 = 450.0
P06 -- P05 = 62.0
P07 -- 40 = 57.0
P07 -- 41 = 10.0
P07 -- E101 = 180.0
P07 -- E44 = 270.0
P07 -- E847 = 120.0
P07 -- E993 = 47.0
P07 -- P03 = 400.0
P07 -- P06 = 350.0
34 -- 35 = 12.0
51 -- 52 = 234.0
> Fim da impressão das arestas.
```

- Mostrar conteúdo do arquivo.



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
===== MENU DE OPÇÕES =====  
a. Ler dados do arquivo 'grafo.txt'  
b. Gravar dados no arquivo 'grafo.txt'  
c. Inserir vértice;  
d. Inserir aresta;  
e. Remover vértice;  
f. Remover aresta;  
g. Mostrar conteúdo do arquivo;  
h. Mostrar grafo;  
*h. Mostrar apenas vértices do grafo;  
**h. Mostrar apenas arestas do grafo;  
i. Apresentar conexidade;  
j. Encerrar a aplicação.  
=====
```

> Selecione uma opção: *g*

> Imagem do grafo gerada com sucesso!

> O grafo ficará disponível para visualização após a execução do programa.

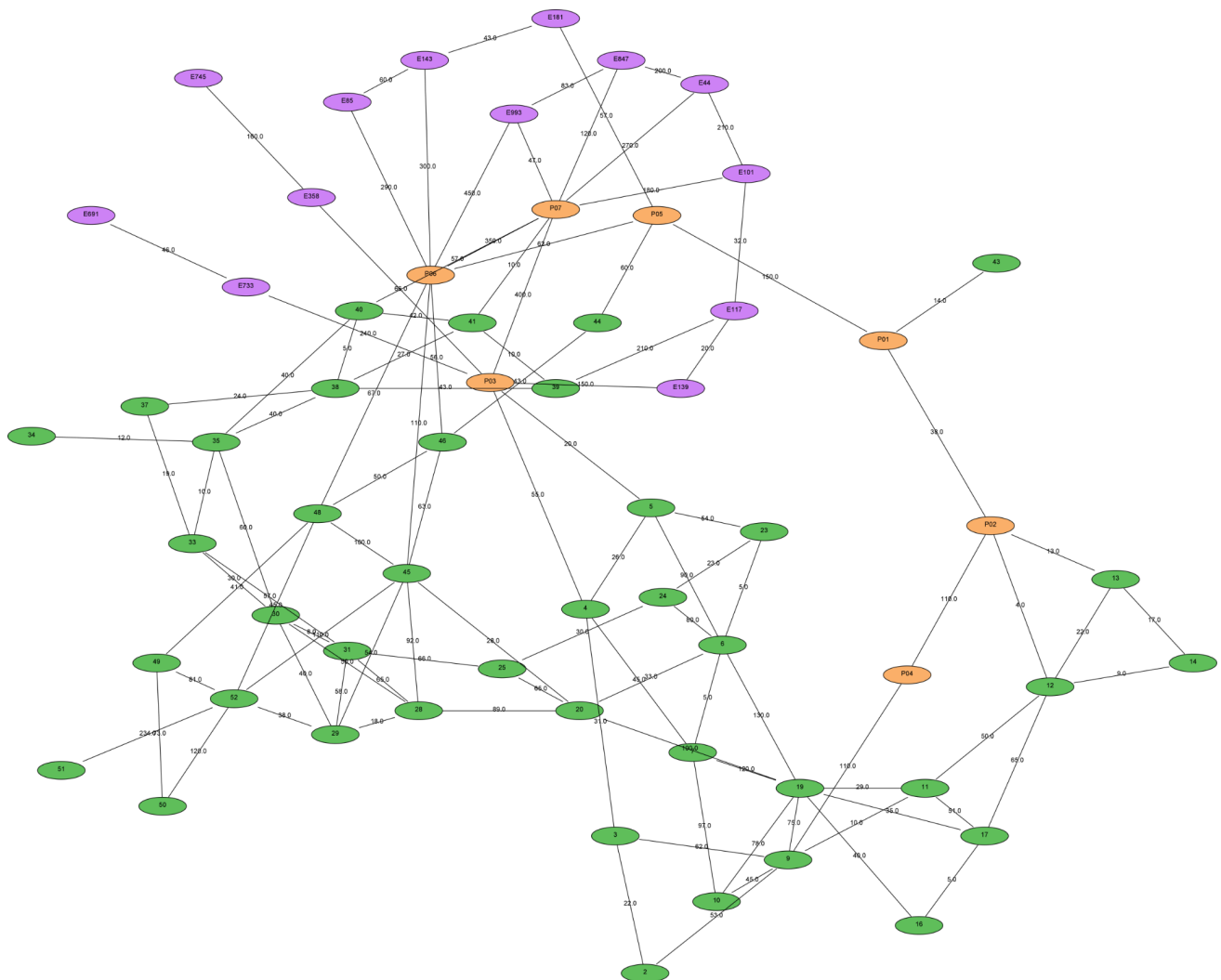


UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos



- Mostrar grafo.



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
===== MENU DE OPÇÕES =====
a. Ler dados do arquivo 'grafo.txt'
b. Gravar dados no arquivo 'grafo.txt'
c. Inserir vértice;
d. Inserir aresta;
e. Remover vértice;
f. Remover aresta;
g. Mostrar conteúdo do arquivo;
h. Mostrar grafo;
*h. Mostrar apenas vértices do grafo;
**h. Mostrar apenas arestas do grafo;
i. Apresentar conexidade;
j. Encerrar a aplicação.
=====

> Selecione uma opção: h

> Início da impressão do grafo (n = 60 e m = 109):

Adj[2,2] = INF Adj[2,3] = 22.0 Adj[2,4] = INF Adj[2,5] = INF Adj[2,6] = INF Adj[2,7] = INF Adj[2,8] = INF
Adj[2,12] = INF Adj[2,13] = INF Adj[2,14] = INF Adj[2,16] = INF Adj[2,17] = INF Adj[2,19] = INF
INF Adj[2,25] = INF Adj[2,28] = INF Adj[2,29] = INF Adj[2,30] = INF Adj[2,31] = INF Adj[2,33] =
Adj[2,38] = INF Adj[2,39] = INF Adj[2,40] = INF Adj[2,41] = INF Adj[2,43] = INF Adj[2,44] = INF
INF Adj[2,49] = INF Adj[2,50] = INF Adj[2,52] = INF Adj[2,E101] = INF Adj[2,E117] = INF Adj[2,E133] =
INF Adj[2,E358] = INF Adj[2,E44] = INF Adj[2,E691] = INF Adj[2,E733] = INF Adj[2,E745] = INF Adj[2,E751] =
= INF Adj[2,P01] = INF Adj[2,P02] = INF Adj[2,P03] = INF Adj[2,P04] = INF Adj[2,P05] = INF Adj[2,P06] =
Adj[2,51] = INF
Adj[3,2] = 22.0 Adj[3,3] = 22.0 Adj[3,4] = 31.0 Adj[3,5] = INF Adj[3,6] = INF Adj[3,7] = INF Adj[3,8] = INF
Adj[3,12] = INF Adj[3,13] = INF Adj[3,14] = INF Adj[3,16] = INF Adj[3,17] = INF Adj[3,19] = INF
INF Adj[3,25] = INF Adj[3,28] = INF Adj[3,29] = INF Adj[3,30] = INF Adj[3,31] = INF Adj[3,33] =
Adj[3,38] = INF Adj[3,39] = INF Adj[3,40] = INF Adj[3,41] = INF Adj[3,43] = INF Adj[3,44] = INF
INF Adj[3,49] = INF Adj[3,50] = INF Adj[3,52] = INF Adj[3,E101] = INF Adj[3,E117] = INF Adj[3,E133] =
INF Adj[3,E358] = INF Adj[3,E44] = INF Adj[3,E691] = INF Adj[3,E733] = INF Adj[3,E745] = INF Adj[3,E751] =
= INF Adj[3,P01] = INF Adj[3,P02] = INF Adj[3,P03] = INF Adj[3,P04] = INF Adj[3,P05] = INF Adj[3,P06] =
Adj[3,51] = INF
Adj[4,2] = INF Adj[4,3] = 31.0 Adj[4,4] = 31.0 Adj[4,5] = 26.0 Adj[4,6] = INF Adj[4,7] = 45.0 Adj[4,8] = INF
```

- Apresentar conexidade do grafo.



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
===== MENU DE OPÇÕES =====  
a. Ler dados do arquivo 'grafo.txt'  
b. Gravar dados no arquivo 'grafo.txt'  
c. Inserir vértice;  
d. Inserir aresta;  
e. Remover vértice;  
f. Remover aresta;  
g. Mostrar conteúdo do arquivo;  
h. Mostrar grafo;  
*h. Mostrar apenas vértices do grafo;  
**h. Mostrar apenas arestas do grafo;  
i. Apresentar conexidade;  
j. Encerrar a aplicação.  
=====
```

> Selecione uma opção: **i**

> Conexidade do grafo: conexo.

- Gravar dados no arquivo **grafo.txt**.

```
===== MENU DE OPÇÕES =====  
a. Ler dados do arquivo 'grafo.txt'  
b. Gravar dados no arquivo 'grafo.txt'  
c. Inserir vértice;  
d. Inserir aresta;  
e. Remover vértice;  
f. Remover aresta;  
g. Mostrar conteúdo do arquivo;  
h. Mostrar grafo;  
*h. Mostrar apenas vértices do grafo;  
**h. Mostrar apenas arestas do grafo;  
i. Apresentar conexidade;  
j. Encerrar a aplicação.  
=====
```

> Selecione uma opção: **b**

> Dados gravados com sucesso!

- Encerrar aplicação.



```
===== MENU DE OPÇÕES =====  
a. Ler dados do arquivo 'grafo.txt'  
b. Gravar dados no arquivo 'grafo.txt'  
c. Inserir vértice;  
d. Inserir aresta;  
e. Remover vértice;  
f. Remover aresta;  
g. Mostrar conteúdo do arquivo;  
h. Mostrar grafo;  
*h. Mostrar apenas vértices do grafo;  
**h. Mostrar apenas arestas do grafo;  
i. Apresentar conexidade;  
j. Encerrar a aplicação.  
=====
```

> Selecione uma opção: j

> Encerrando o programa...

- Execução (Parte II)

- Simulação Dijkstra.

```
===== MENU DE OPÇÕES =====  
a. Ler dados do arquivo 'grafo.txt'  
b. Gravar dados no arquivo 'grafo.txt'  
c. Inserir vértice;  
d. Inserir aresta;  
e. Remover vértice;  
f. Remover aresta;  
g. Mostrar conteúdo do arquivo;  
h. Mostrar grafo;  
*h. Mostrar apenas vértices do grafo;  
**h. Mostrar apenas arestas do grafo;  
i. Apresentar conexidade;  
j. Encontrar o menor caminho;  
k. Encontrar a árvore de custo mínimo;  
l. Encontrar a coloração de vértices;  
m. Encerrar a aplicação.  
=====
```

> Selecione uma opção: j

> Insira o primeiro vértice: 31
> Insira o segundo vértice: 1
> Caminho mínimo: 31 25 24 23 6 7 4 3 2 1



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

- Simulação Prim.

```
> Selecione uma opção: k  
  
> Exibindo arestas da árvore de custo mínimo:  
> Início da impressão das arestas (m = 54):  
2 -- 1 = 23.0  
3 -- 2 = 22.0  
4 -- 3 = 31.0  
5 -- 4 = 26.0  
7 -- 4 = 45.0  
7 -- 6 = 5.0  
9 -- 1 = 49.0  
10 -- 9 = 45.0  
11 -- 9 = 10.0  
12 -- 11 = 50.0  
14 -- 12 = 9.0  
17 -- 16 = 5.0  
19 -- 11 = 29.0  
19 -- 17 = 35.0  
20 -- 6 = 33.0  
23 -- 6 = 5.0  
24 -- 23 = 23.0  
25 -- 24 = 30.0  
29 -- 28 = 18.0  
30 -- 29 = 40.0  
31 -- 30 = 8.0  
33 -- 30 = 30.0  
35 -- 33 = 10.0  
37 -- 33 = 19.0  
38 -- 37 = 24.0  
40 -- 38 = 5.0  
41 -- 38 = 27.0  
45 -- 20 = 28.0  
45 -- 29 = 54.0  
46 -- 44 = 43.0  
48 -- 46 = 50.0  
49 -- 48 = 41.0  
50 -- 49 = 73.0  
52 -- 29 = 38.0  
52 -- 48 = 45.0  
E117 -- E101 = 32.0  
E181 -- E143 = 43.0  
E44 -- E101 = 210.0  
E44 -- E40 = 38.0  
E139 -- E117 = 20.0  
E85 -- E143 = 60.0  
P01 -- 43 = 14.0  
P02 -- 12 = 4.0  
P02 -- 13 = 13.0  
P02 -- P01 = 38.0  
P03 -- 5 = 20.0  
P03 -- E249 = 62.0  
P03 -- E139 = 150.0  
P04 -- P03 = 55.0  
P05 -- 44 = 60.0  
P05 -- E181 = 57.0  
P06 -- 46 = 56.0  
P07 -- 41 = 10.0  
P07 -- E993 = 47.0  
> Fim da impressão das arestas.  
> Custo total: 2017.0
```



- Simulação coloração sequencial.

```
> Selecione uma opção: 1  
> Exibindo cores: [[0, 2, 4, 6, 9, 11, 13, 16, 18, 21, 24, 27, 30, 31, 34, 36, 38, 40, 42, 43, 45, 47], [1, 3, 5, 8, 10, 19, 20, 25, 29, 33, 35, 39, 41, 44, 46, 48, 50], [7, 12, 14, 17, 22, 26, 28, 32, 49, 52], [15, 23, 37, 51, 53], [54]]
```

- **Apêndice**

Link do projeto no GitHub: <https://github.com/icampolim/grafos>

OBS: Esse é o nosso link da disciplina, o projeto encontra-se no diretório **grafos/src/projeto**

- **Código Fonte (Projeto)**

1. MainProjeto.java

```
// Enzo Guarnieri, 10410074
```

```
// Erika Borges Piaui, 10403716
```

```
// Júlia Campolim de Oste, 10408802
```

```
// Essa é a classe Main
```

```
// Responsável por imprimir o menu de opções para o usuário e, de acordo com a entrada dele,  
chamar o método da Grafo associado
```

```
package projeto;
```

```
import java.io.IOException;
```

```
import java.util.List;
```

```
import java.util.Scanner;
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
public class MainProjeto {  
  
    public static void menu() throws IOException {  
  
        Scanner scan = new Scanner(System.in);  
  
        String op, v, w;  
  
        boolean existeGrafo = false;  
  
  
        Grafo grafo = new Grafo(0);  
  
  
        do {  
  
            System.out.println("===== MENU DE OPÇÕES =====");  
  
            System.out.println(" a. Ler dados do arquivo 'grafo.txt'");  
  
            System.out.println(" b. Gravar dados no arquivo 'grafo.txt'");  
  
            System.out.println(" c. Inserir vértice;");  
  
            System.out.println(" d. Inserir aresta;");  
  
            System.out.println(" e. Remover vértice;");  
  
            System.out.println(" f. Remover aresta;");  
  
            System.out.println(" g. Mostrar conteúdo do arquivo;");  
  
            System.out.println(" h. Mostrar grafo;");  
  
            System.out.println(" *h. Mostrar apenas vértices do grafo;");  
  
            System.out.println(" **h. Mostrar apenas arestas do grafo;");  
  
            System.out.println(" i. Apresentar conexidade;");  
  
            System.out.println(" j. Encontrar o menor caminho;");  
  
            System.out.println(" k. Encontrar a árvore de custo mínimo;");  
  
        } while (true);  
    }  
}
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
System.out.println(" l. Encontrar a coloração de vértices;");
```

```
System.out.println(" m. Encerrar a aplicação.");
```

```
System.out.println("=====");
```

```
System.out.print("\n> Selecione uma opção: ");
```

```
op = scan.next();
```

```
while(!op.equals("a") && !op.equals("b") && !op.equals("c") && !op.equals("d") &&  
!op.equals("e"))
```

```
&& !op.equals("f") && !op.equals("g") && !op.equals("h") && !op.equals("**h") &&  
!op.equals("**h"))
```

```
&& !op.equals("i") && !op.equals("j") && !op.equals("k") && !op.equals("l") &&  
!op.equals("m")) {
```

```
System.out.print("\n> Opção inválida. Tente novamente: ");
```

```
op = scan.next();
```

```
}
```

```
System.out.println();
```

```
if(!existeGrafo && !op.equals("a") && !op.equals("j")) {
```

```
System.out.println("> É preciso ler um grafo antes de realizar esta operação!\n");
```

```
continue;
```

```
}
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
switch (op) {  
  
    case "a":  
  
        if(grafo.lerGTGrafo("grafo.txt") == 1) {  
  
            System.out.println("> Grafo lido com sucesso!");  
  
            existeGrafo = true;  
  
        }  
  
        break;  
  
    case "b":  
  
        grafo.escreverArquivo();  
  
        break;  
  
    case "c":  
  
        System.out.print("> Insira o rótulo do vértice: ");  
  
        v = scan.next();  
  
  
        if(!grafo.verificaVertice(v)) {  
  
            grafo.insereVertice(v);  
  
            System.out.println("> Vértice inserido com sucesso!");  
  
        } else {  
  
            System.out.println("> Vértice " + v + " já existe no grafo");  
  
        }  
  
  
        break;  
  
    case "d":  
  
        System.out.print("> Insira o primeiro vértice: ");
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
v = scan.next();
```

```
System.out.print("> Insira o segundo vértice: ");
```

```
w = scan.next();
```

```
System.out.print("> Insira o peso da aresta: ");
```

```
float valor = scan.nextFloat();
```

```
if(grafo.verificaVertice(v) || grafo.verificaVertice(w)) {
```

```
    grafo.insereAresta(v, w, valor);
```

```
    System.out.println("> Aresta adicionada com sucesso!");
```

```
} else {
```

```
    System.out.println("> Vértice " + (grafo.verificaVertice(v) ? w : v) + " inválido.");
```

```
}
```

```
break;
```

```
case "e":
```

```
System.out.print("> Insira o vértice: ");
```

```
v = scan.next();
```

```
if(grafo.verificaVertice(v)) {
```

```
    grafo.removeVertice(v);
```

```
    System.out.println("> Vértice removido com sucesso!");
```

```
} else {
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
        System.out.println("> Vértice " + v + " inválido");
    }

    break;

case "f":

    System.out.print("> Insira o primeiro vértice: ");

    v = scan.next();

    System.out.print("> Insira o segundo vértice: ");

    w = scan.next();

    if(grafo.verificaVertice(v) || grafo.verificaVertice(w)) {

        grafo.removeAresta(v, w);

        System.out.println("> Aresta removida com sucesso!");

    } else {

        System.out.println("> Vértice " + (grafo.verificaVertice(v) ? w : v) + " inválido.");

    }

    break;

case "g":

    grafo.mostrarGrafo();

    System.out.println("> O grafo ficará disponível para visualização após a execução do
programa.");

    break;
```




UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
case "h":  
    grafo.show();  
  
    break;  
  
case "*h":  
    grafo.exibirVertices();  
  
    break;  
  
case "***h":  
    grafo.exibirArestas();  
  
    break;  
  
case "i":  
    System.out.println("> Conexidade do grafo: " + (grafo.verificaConexidade() == 1 ?  
"desconexo." : "conexo."));  
  
    break;  
  
case "j":  
    System.out.print("> Insira o primeiro vértice: ");  
  
    v = scan.next();  
  
  
    System.out.print("> Insira o segundo vértice: ");  
  
    w = scan.next();  
  
  
    List<String> caminhoMinimo = grafo.caminhoMinimo(v, w);  
  
    if(caminhoMinimo.isEmpty()) {  
  
        System.out.print("> Não foi possível encontrar um caminho entre os vértices " + v + " e  
" + w + "."));  
  
    } else {
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
        System.out.print("> Caminho mínimo: " + String.join(" ", caminhoMinimo));  
    }  
  
    break;  
  
    case "k":  
  
        if(grafo.verificaConexidade() == 1) {  
  
            System.out.println("> Não é possível encontrar uma árvore de custo mínimo para um  
grafo não conexo.");  
  
        } else {  
  
            Grafo arvore = grafo.getArvoreCustoMinimo();  
  
            System.out.println("> Exibindo arestas da árvore de custo mínimo: ");  
  
            arvore.exibirArestas();  
  
            System.out.println("> Custo total: " + arvore.getTotalArestas());  
  
        }  
  
        break;  
  
    case "l":  
  
        System.out.println("> Exibindo cores: " + grafo.coloracaoSequencial().toString());  
  
        break;  
  
    case "m":  
  
        System.out.println("> Encerrando o programa...");  
  
        break;  
  
    }  
  
  
    System.out.println();  
  
} while(!op.equals("m"));
```



```
scan.close();  
  
}  
  
}
```

2. TGrafoND.java

```
// Enzo Guarnieri, 10410074  
  
// Erika Borges Piaui, 10403716  
  
// Júlia Campolim de Oste, 10408802  
  
package projeto;  
  
  
import com.mxgraph.layout.mxCircleLayout;  
  
import com.mxgraph.layout.mxFastOrganicLayout;  
  
import com.mxgraph.util.mxCellRenderer;  
  
import com.mxgraph.view.mxGraph;  
  
  
import javax.imageio.ImageIO;  
  
import java.awt.*;  
  
import java.awt.image.BufferedImage;  
  
import java.io.File;  
  
import java.io.FileNotFoundException;  
  
import java.io.FileWriter;  
  
import java.io.IOException;
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;


public class TGrafoND {


    private    int n;        // quantidade de vértices

    private    int m;        // quantidade de arestas

    private float[][] adj;   //matriz de adjacência

    private List<String> rotulos;        // vetor que armazena os valores dos vértices


    private final float INF = Float.MAX_VALUE;    // define o valor do infinito para grafos ponderados


    public TGrafoND(int n) {

        this.n = n;

        this.m = 0;        // inicialmente não há arestas

        this.adj = new float [n][n];


        this.rotulos = new ArrayList<>();


        // inicia a matriz com zeros

        for(int i = 0; i < n; i++) {

            for(int j = 0; j < n; j++) {

                this.adj[i][j] = INF;
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
    }  
}  
}  
  
// insere uma aresta no TGrafo tal que v é adjacente a w e w é adjacente a v  
public void insereAresta(String rotuloV, String rotuloW, float valor) {  
    int v = rotulos.indexOf(rotuloV);  
    int w = rotulos.indexOf(rotuloW);  
  
    if(adj[v][w] == INF) {    // verifica se não temos aresta  
        adj[v][w] = valor;  
        adj[w][v] = valor;  
        this.m++;  
    }  
}  
  
// remove uma aresta v->w/w->v do TGrafo  
public void removeAresta(String rotuloV, String rotuloW) {  
    int v = rotulos.indexOf(rotuloV);  
    int w = rotulos.indexOf(rotuloW);  
  
    if(adj[v][w] != INF) {    // verifica se temos a aresta  
        adj[v][w] = INF;  
        adj[w][v] = INF;  
    }  
}
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
this.m--;  
  
}  
  
}  
  
// insere vértice no grafo não direcionado  
public void insereVertice(String rotulo) {  
  
    rotulos.add(rotulo);  
  
    n++;  
  
    float[][] atualizarAdj = new float[n][n];  
  
    for (int i = 0; i < n - 1; i++) {  
        for (int j = 0; j < n - 1; j++) {  
            atualizarAdj[i][j] = adj[i][j];  
        }  
    }  
  
    for (int i = 0; i < n; i++) {  
        atualizarAdj[i][n - 1] = INF;  
        atualizarAdj[n - 1][i] = INF;  
    }  
  
    adj = atualizarAdj;
```



```
}
```

```
// remove vértice do grafo não direcionado
```

```
public void removeVertice(String vertice) {
```

```
    int v = rotulos.indexOf(vertice);
```

```
    rotulos.remove(v);
```

```
// remove as arestas do vértice v
```

```
for (int i = 0; i < n; i++) {
```

```
    if (adj[v][i] != INF) {
```

```
        adj[v][i] = INF;
```

```
        adj[i][v] = INF;
```

```
        this.m--;
```

```
    }
```

```
}
```

```
// remove a linha e a coluna do vértice na matriz de adjacência
```

```
for (int i = v; i < n - 1; i++) {
```

```
    for (int j = 0; j < n; j++) {
```

```
        adj[i][j] = adj[i + 1][j];
```

```
        adj[j][i] = adj[j][i + 1];
```

```
    }
```

```
}
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
n--;  
}
```

```
// retorna o tipo de conexidade
```

```
public int verificaConexidade() {
```

```
    // verifica se existe um caminho para todo par de vértice
```

```
    for(int i = 0; i < n; i++) {
```

```
        for(int j = 0; j < n; j++) {
```

```
            if (!existeCaminho(this, i, j)) {
```

```
                return 1;
```

```
            }
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

```
private boolean existeCaminho(TGrafoND g, int inicio, int fim) {
```

```
    boolean[] visitado = new boolean[g.n];
```

```
    return buscaCaminho(g, inicio, fim, visitado);
```

```
}
```

```
// faz uma busca de profundidade no grafo para verificar se é possível chegar de um vértice a  
outro
```




UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
private boolean buscaCaminho(TGrafoND grafo, int atual, int fim, boolean[] visitado) {  
    if (atual == fim) return true;  
  
    visitado[atual] = true;  
  
    for (int i = 0; i < grafo.n; i++) {  
        if(grafo.adj[atual][i] != INF && !visitado[i]) {  
            if(buscaCaminho(grafo, i, fim, visitado)) return true;  
        }  
    }  
  
    return false;  
}
```

```
public int lerGTGrafo(String file){  
    try {  
        Scanner scanner = new Scanner(new File(file));  
  
        int tipoDoGrafo = scanner.nextInt();  
  
        this.n = scanner.nextInt();    // número de vértices  
  
        for(int i = 0; i < n; i++) {  
            String aux = scanner.next();
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
this.rotulos.add(aux);

}

int arestasLidas = scanner.nextInt(); // número de arestas

this.adj = new float [n][n];

this.m = 0;

// inicializa a matriz com INF
for(int i = 0; i < n; i++) {
    for(int j = 0; j < n; j++) {
        this.adj[i][j] = INF;
        this.adj[j][i] = INF;
    }
}

while(scanner.hasNext()) {
    String origem = scanner.next();
    String destino = scanner.next();

    float valor = scanner.nextFloat();

    if(!verificaVertice(origem) || !verificaVertice(destino)) {
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
        System.out.println("> Vértice " + (verificaVertice(origem) ? destino : origem) + " não  
encontrado.");  
  
        return 0;  
  
    }  
  
    this.insereAresta(origem, destino, valor);  
  
}  
  
scanner.close();  
  
// verifica erro: número de arestas n e quantidade de linhas no arquivo diferente  
if(arestasLidas == m) return 1;  
  
System.out.println("> Arquivo em formato inválido.");  
  
return 0;  
  
} catch(FileNotFoundException e) {  
    System.out.println("> Arquivo não encontrado: " + file);  
    return 0;  
}  
catch(Exception e) {  
    System.out.println("> Arquivo em formato inválido.");  
    return 0;  
}  
}
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
public void escreverArquivo() {  
  
    try {  
  
        FileWriter fileWriter = new FileWriter("grafo.txt");  
  
        fileWriter.write("2" + "\n"); // tipo do grafo  
  
        fileWriter.write(this.n + "\n"); // quantidade de vértices  
  
  
        for(int i = 0; i < n; i++) {  
  
            fileWriter.write(rotulos.get(i) + "\n");  
  
        }  
  
  
        fileWriter.write(this.m + "\n"); // quantidade de arestas  
  
  
        for(int i = 0; i < n; i++) {  
  
            for(int j = 0; j < i; j++) {  
  
                if(adj[i][j] != INF) {  
  
                    fileWriter.write(rotulos.get(i) + " " + rotulos.get(j) + " " + adj[i][j] + "\n");  
  
                }  
  
            }  
  
        }  
  
  
        fileWriter.close();  
  
        System.out.println("> Dados gravados com sucesso!");  
    }  
}
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
} catch (IOException e) {  
    System.out.println("> Não foi possível gravar dados no arquivo.");  
}  
}
```

```
public boolean verificaVertice(String rotulo) {  
    return this.rotulos.contains(rotulo);  
}
```

```
public void mostrarGrafo() throws IOException {  
    mxGraph grafo = new mxGraph();  
    Object parent = grafo.getDefaultParent();  
  
    // iniciando a edição do grafo  
    grafo.getModel().beginUpdate();
```

```
try {  
    // adicionando os vértices no mxGraph  
    Object[] vertices = new Object[this.n];  
    for (int i = 0; i < this.n; i++) {  
        if(rotulos.get(i).startsWith("E")) {  
            vertices[i] = grafo.insertVertex(parent, null, rotulos.get(i), 0, 0, 80, 30,  
"shape=ellipse;fillColor=#CD82F5;strokeColor=#000000;rounded=1;whiteSpace=wrap;html=1;fontC  
olor=#000000;");  
        } else if(rotulos.get(i).startsWith("P")) {
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira

Disciplina: Teoria dos Grafos

```
vertices[i] = grafo.insertVertex(parent, null, rotulos.get(i), 0, 0, 80, 30,
"shape=ellipse;fillColor=#FAB06B;strokeColor=#000000;rounded=1;whiteSpace=wrap;html=1;fontC
olor=#000000;");

    } else {

        vertices[i] = grafo.insertVertex(parent, null, rotulos.get(i), 0, 0, 80, 30,
"shape=ellipse;fillColor=#5FBF5C;strokeColor=#000000;rounded=1;whiteSpace=wrap;html=1;fontC
olor=#000000;");

    }

}

// adicionando as arestas no mxGraph com base na matriz de adjacência

for (int i = 0; i < this.n; i++) {

    for (int j = i + 1; j < this.n; j++) {

        if (adj[i][j] != INF) {

            grafo.insertEdge(parent, null, adj[i][j], vertices[i], vertices[j],
"edgeStyle=orthogonalEdge;rounded=0;orthogonalLoop=1;exitDx=0;exitDy=0;endArrow=none;strok
eColor=#000000;fontColor=#000000;");

        }

    }

}

// opções para melhorar o layout do grafo

mxFastOrganicLayout layout = new mxFastOrganicLayout(grafo);

layout.setForceConstant(200);

layout.execute(parent);
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
} finally {
```

```
    // finaliza a edição do grafo
```

```
    grafo.getModel().endUpdate();
```

```
}
```

```
    BufferedImage image = mxCellRenderer.createBufferedImage(grafo, null, 5, Color.WHITE, true,  
null);
```

```
File file = new File("grafo.png");
```

```
ImageIO.write(image, "PNG", file);
```

```
System.out.println("> Imagem do grafo gerada com sucesso!");
```

```
}
```

```
public void exibirVertices() {
```

```
    System.out.println("> Início da impressão dos vértices (n = " + n + "): ");
```

```
    for(String v : this.rotulos) {
```

```
        System.out.println(v);
```

```
    }
```

```
    System.out.println("> Fim da impressão dos vértices.");
```

```
}
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
public void exibirArestas() {

    System.out.println("> Início da impressão das arestas (m = " + m + "): ");

    final int n = this.n;

    for(int i = 0; i < n; i++) {

        for(int j = 0; j < i; j++) {

            if(adj[i][j] != INF) {

                System.out.println(this.rotulos.get(i) + " -- " + this.rotulos.get(j) + " = " + this.adj[i][j]);

            }

        }

    }

    System.out.println("> Fim da impressão das arestas.");

}

public void show() {

    System.out.println("> Início da impressão do grafo (n = " + n + " e m = " + m + "): ");

    for(int i = 0; i < n; i++){

        System.out.println();

        for(int w = 0; w < n; w++) {

            if(adj[i][w] == INF) System.out.print("Adj[" + rotulos.get(i) + "," + rotulos.get(w) + "] = INF");

            else System.out.print("Adj[" + rotulos.get(i) + "," + rotulos.get(w) + "] = " + adj[i][w] + " ");

        }

    }

}
```




```
}
```

```
}
```

```
System.out.println("\n\n> Fim da impressao do grafo." );
```

```
}
```

```
}
```

3. Grafo.java

```
// Enzo Guarnieri, 10410074
```

```
// Erika Borges Piaui, 10403716
```

```
// Júlia Campolim de Oste, 10408802
```

```
// Essa classe é responsável pela construção do grafo bem como a criação de todos os métodos  
// associados a operações do/no grafo
```

```
// Principais métodos implementados: inserir aresta e vértice, remover aresta e vértice, verificar  
// conexidade,
```

```
// Verificar se existe caminho entre vértices, ler o grafo, escrever no arquivo, mostrar o grafo, exibir  
// vértices e exibir arestas
```

```
package projeto;
```

```
import com.mxgraph.layout.mxFastOrganicLayout;
```

```
import com.mxgraph.util.mxCellRenderer;
```

```
import com.mxgraph.view.mxGraph;
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
import javax.imageio.ImageIO;

import java.awt.*;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

import java.util.Scanner;

import java.util.stream.Collectors;

import java.util.stream.IntStream;


public class Grafo {


    private    int n;        // quantidade de vértices

    private    int m;        // quantidade de arestas

    private float[][] adj;   //matriz de adjacência

    private List<String> rotulos;        // vetor que armazena os valores dos vértices


    private final float INF = Float.MAX_VALUE;    // define o valor do infinito para grafos ponderados


    public Grafo(int n) {
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
this.n = n;

this.m = 0;    // inicialmente não há arestas

this.adj = new float [n][n];

this.rotulos = new ArrayList<>();

// inicia a matriz com zeros
for(int i = 0; i < n; i++) {
    for(int j = 0; j < n; j++) {
        this.adj[i][j] = INF;
    }
}

public void setRotulos(List<String> rotulos) {
    this.rotulos = rotulos;
}

// insere uma aresta no TGrafo tal que v é adjacente a w e w é adjacente a v
public void insereAresta(String rotuloV, String rotuloW, float valor) {

    int v = rotulos.indexOf(rotuloV);
    int w = rotulos.indexOf(rotuloW);

    if(adj[v][w] == INF) {    // verifica se não temos aresta
```



```
adj[v][w] = valor;

adj[w][v] = valor;

this.m++;

}

}

public void insereA(int v, int w, float valor) {

    if(adj[v][w] == INF) {        // verifica se não temos aresta

        adj[v][w] = valor;

        adj[w][v] = valor;

        this.m++;

    }

}

// remove uma aresta v->w/w->v do TGrafo

public void removeAresta(String rotuloV, String rotuloW) {

    int v = rotulos.indexOf(rotuloV);

    int w = rotulos.indexOf(rotuloW);

    if(adj[v][w] != INF) {        // verifica se temos a aresta

        adj[v][w] = INF;

        adj[w][v] = INF;

        this.m--;

    }

}
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
}
```

```
// insere vértice no grafo não direcionado
```

```
public void insereVertice(String rotulo) {
```

```
    rotulos.add(rotulo);
```

```
    n++;
```

```
    float[][] atualizarAdj = new float[n][n];
```

```
    for (int i = 0; i < n - 1; i++) {
```

```
        for (int j = 0; j < n - 1; j++) {
```

```
            atualizarAdj[i][j] = adj[i][j];
```

```
        }
```

```
    }
```

```
    for (int i = 0; i < n; i++) {
```

```
        atualizarAdj[i][n - 1] = INF;
```

```
        atualizarAdj[n - 1][i] = INF;
```

```
    }
```

```
    adj = atualizarAdj;
```

```
}
```



```
// remove vértice do grafo não direcionado

public void removeVertice(String vertice) {

    int v = rotulos.indexOf(vertice);

    rotulos.remove(v);

    // remove as arestas do vértice v

    for (int i = 0; i < n; i++) {

        if (adj[v][i] != INF) {

            adj[v][i] = INF;

            adj[i][v] = INF;

            this.m--;

        }

    }

    // remove a linha e a coluna do vértice na matriz de adjacência

    for (int i = v; i < n - 1; i++) {

        for (int j = 0; j < n; j++) {

            adj[i][j] = adj[i + 1][j];

            adj[j][i] = adj[j][i + 1];

        }

    }

    n--;

}
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
// retorna o tipo de conexidade
```

```
public int verificaConexidade() {
```

```
    // verifica se existe um caminho para todo par de vértice
```

```
    for(int i = 0; i < n; i++) {
```

```
        for(int j = 0; j < n; j++) {
```

```
            if (!existeCaminho(this, i, j)) {
```

```
                return 1;
```

```
            }
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

```
private boolean existeCaminho(Grafo g, int inicio, int fim) {
```

```
    boolean[] visitado = new boolean[g.n];
```

```
    return buscaCaminho(g, inicio, fim, visitado);
```

```
}
```

// faz uma busca de profundidade no grafo para verificar se é possível chegar de um vértice a outro

```
private boolean buscaCaminho(Grafo grafo, int atual, int fim, boolean[] visitado) {
```

```
    if (atual == fim) return true;
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
visitado[atual] = true;

for (int i = 0; i < grafo.n; i++) {
    if(grafo.adj[atual][i] != INF && !visitado[i]) {
        if(buscaCaminho(grafo, i, fim, visitado)) return true;
    }
}

return false;
}

public int lerGTGrafo(String file){
    try {
        Scanner scanner = new Scanner(new File(file));

        int tipoDoGrafo = scanner.nextInt();

        this.n = scanner.nextInt();    // número de vértices

        for(int i = 0; i < n; i++) {
            String aux = scanner.next();
            this.rotulos.add(aux);
        }
    }
}
```




UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
int arestasLidas = scanner.nextInt(); // número de arestas
```

```
this.adj = new float [n][n];
```

```
this.m = 0;
```

```
// inicializa a matriz com INF
```

```
for(int i = 0; i < n; i++) {
```

```
    for(int j = 0; j < n; j++) {
```

```
        this.adj[i][j] = INF;
```

```
        this.adj[j][i] = INF;
```

```
    }
```

```
}
```

```
while(scanner.hasNext()) {
```

```
    String origem = scanner.next();
```

```
    String destino = scanner.next();
```

```
    float valor = scanner.nextFloat();
```

```
    if(!verificaVertice(origem) || !verificaVertice(destino)) {
```

```
        System.out.println("> Vértice " + (verificaVertice(origem) ? destino : origem) + " não  
encontrado.");
```

```
        return 0;
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
}

    this.insereAresta(origem, destino, valor);

}

scanner.close();

// verifica erro: número de arestas n e quantidade de linhas no arquivo diferente
if(arestasLidas == m) return 1;

System.out.println("> Arquivo em formato inválido.");

return 0;

} catch(FileNotFoundException e) {

    System.out.println("> Arquivo não encontrado: " + file);

    return 0;

} catch(Exception e) {

    System.out.println("> Arquivo em formato inválido.");

    return 0;

}

}

public void escreverArquivo() {

    try {
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
FileWriter fileWriter = new FileWriter("grafo.txt");

fileWriter.write("2" + "\n"); // tipo do grafo

fileWriter.write(this.n + "\n"); // quantidade de vértices

for(int i = 0; i < n; i++) {

    fileWriter.write(rotulos.get(i) + "\n");

}

fileWriter.write(this.m + "\n"); // quantidade de arestas

for(int i = 0; i < n; i++) {

    for(int j = 0; j < i; j++) {

        if(adj[i][j] != INF) {

            fileWriter.write(rotulos.get(i) + " " + rotulos.get(j) + " " + adj[i][j] + "\n");

        }

    }

}

fileWriter.close();

System.out.println("> Dados gravados com sucesso!");

} catch (IOException e) {

    System.out.println("> Não foi possível gravar dados no arquivo.");

}
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
}
```

```
public boolean verificaVertice(String rotulo) {  
    return this.rotulos.contains(rotulo);  
}
```

```
public void mostrarGrafo() throws IOException {  
    mxGraph grafo = new mxGraph();  
    Object parent = grafo.getDefaultParent();  
  
    // iniciando a edição do grafo  
    grafo.getModel().beginUpdate();
```

```
try {
```

```
    // adicionando os vértices no mxGraph
```

```
    Object[] vertices = new Object[this.n];
```

```
    for (int i = 0; i < this.n; i++) {
```

```
        if(rotulos.get(i).startsWith("E")) {
```

```
            vertices[i] = grafo.insertVertex(parent, null, rotulos.get(i), 0, 0, 80, 30,  
            "shape=ellipse;fillColor=#CD82F5;strokeColor=#000000;rounded=1;whiteSpace=wrap;html=1;fontC  
            olor=#000000;");
```

```
        } else if(rotulos.get(i).startsWith("P")) {
```

```
            vertices[i] = grafo.insertVertex(parent, null, rotulos.get(i), 0, 0, 80, 30,  
            "shape=ellipse;fillColor=#FAB06B;strokeColor=#000000;rounded=1;whiteSpace=wrap;html=1;fontC  
            olor=#000000;");
```

```
        } else {
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
        vertices[i] = grafo.insertVertex(parent, null, rotulos.get(i), 0, 0, 80, 30,
"shape=ellipse;fillColor=#5FBF5C;strokeColor=#000000;rounded=1;whiteSpace=wrap;html=1;fontC
olor=#000000;");

    }

}

// adicionando as arestas no mxGraph com base na matriz de adjacência

for (int i = 0; i < this.n; i++) {

    for (int j = i + 1; j < this.n; j++) {

        if (adj[i][j] != INF) {

            grafo.insertEdge(parent, null, adj[i][j], vertices[i], vertices[j],
"edgeStyle=orthogonalEdge;rounded=0;orthogonalLoop=1;exitDx=0;exitDy=0;endArrow=none;strok
eColor=#000000;fontColor=#000000;");

        }

    }

}

// opções para melhorar o layout do grafo

mxFastOrganicLayout layout = new mxFastOrganicLayout(grafo);

layout.setForceConstant(200);

layout.execute(parent);

} finally {

    // finaliza a edição do grafo

    grafo.getModel().endUpdate();

}
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
BufferedImage image = mxCellRenderer.createBufferedImage(grafo, null, 5, Color.WHITE, true, null);
```

```
File file = new File("grafo.png");
```

```
ImageIO.write(image, "PNG", file);
```

```
System.out.println("> Imagem do grafo gerada com sucesso!");
```

```
}
```

```
public void exibirVertices() {
```

```
    System.out.println("> Início da impressão dos vértices (n = " + n + "): ");
```

```
    for(String v : this.rotulos) {
```

```
        System.out.println(v);
```

```
    }
```

```
    System.out.println("> Fim da impressão dos vértices.");
```

```
}
```

```
// algoritmo de Dijkstra
```

```
private int[] dijkstra(int no) {
```

```
    float[] dist = new float[this.n];           // vetor de distâncias (custo mínimo para alcançar cada vértice a partir do nó "no")
```

```
    boolean[] visitado = new boolean[this.n]; // vetor para marcar os vértices já visitados
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

`int[] pred = new int[this.n]; // vetor de predecessores: para cada vértice, guarda o nó anterior no caminho mínimo`

```
for (int i = 0; i < this.n; i++) {  
    dist[i] = INF;  
    visitado[i] = false;  
    pred[i] = -1;  
}
```

```
dist[no] = 0;
```

```
for (int i = 0; i < this.n; i++) {  
    int u = -1;  
    float menorDist = INF;  
    for (int j = 0; j < this.n; j++) {  
        if (!visitado[j] && dist[j] < menorDist) {  
            menorDist = dist[j];  
            u = j;  
        }  
    }  
}
```

```
if (u == -1) {  
    break;  
}
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
visitado[u] = true;

for (int v = 0; v < this.n; v++) {

    if (!visitado[v] && this.adj[u][v] != INF) {

        if (dist[u] + this.adj[u][v] < dist[v]) {

            dist[v] = dist[u] + this.adj[u][v];

            pred[v] = u;

        }

    }

}

return pred;

}

public List<String> caminhoMinimo(String origem, String destino) {

    if(!verificaVertice(origem) || !verificaVertice(destino)) return new ArrayList<>();

    int origemIndex = rotulos.indexOf(origem);

    int destinoIndex = rotulos.indexOf(destino);

    int[] caminhos = dijkstra(origemIndex);

    List<String> caminhoFinal = new ArrayList<>();

    if (caminhos[destinoIndex] == -1) {
```




Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```

        return caminhoFinal;
    }

    int atual = destinoIndex;

    while (atual != -1) {
        caminhoFinal.add(rotulos.get(atual));
        atual = caminhos[atual];
    }

    Collections.reverse(caminhoFinal);

    return caminhoFinal;
}

public Grafo getArvoreCustoMinimo() {
    // inicialização das variáveis

    float custo = 0;

    Grafo arvore = new Grafo(this.n);

    arvore.setRotulos(this.rotulos);

    List<Integer> verticesArvore = new ArrayList<>(); // vetor com os vértices já
    verticesArvore.add(0); // adicionados a árvore
}

```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
List<Integer> verticesRestantes = IntStream.rangeClosed(1, this.n - 1) // vetor com os vértices
que precisam

        .boxed() // ser adicionados à árvore

        .collect(Collectors.toList());

prim(arvore, verticesArvore, verticesRestantes, custo);

return arvore;
}

public void prim(Grafo arvore, List<Integer> verticesArvore, List<Integer> verticesRestantes, float
custo) {

    float valor = INF; // armazena o valor da menor aresta

    int vint = 0, vext = 0; // armazena o valor do vértice interno e do vértice externo

    for(int k : verticesArvore) {

        for(int i : verticesRestantes) {

            if(adj[k][i] < valor) {

                valor = adj[k][i];

                vint = k;

                vext = i;

            }

        }

    }

}
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
custo = custo + valor;
```

```
arvore.insereA(vint, vext, valor);    // adiciona a nova aresta na árvore
```

```
// insere novo vértice no vetor de vértice da árvore e remove do vetor de vértices que precisam  
ser adicionados
```

```
verticesArvore.add(vext);
```

```
verticesRestantes.remove(Integer.valueOf(vext));
```

```
if(verticesArvore.size() != this.n) {
```

```
    prim(arvore, verticesArvore, verticesRestantes, custo);
```

```
}
```

```
}
```

```
public float getTotalArestas() {
```

```
    float total = 0;
```

```
    for(int i = 0; i < this.n; i++) {
```

```
        for(int j = 0; j < i; j++) {
```

```
            if(adj[i][j] != INF) {
```

```
                total += adj[i][j];
```

```
            }
```

```
        }
```

```
    }
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
    return total;
}

// realiza a coloração utilizando coloração sequencial
public List<List<Integer>> coloracaoSequencial() {
    List<List<Integer>> cores = new ArrayList<>();
    boolean[] colorido = new boolean[n];

    for (int i = 0; i < n; i++) {
        if (!colorido[i]) {
            boolean in = false;

            for (int k = 0; k < cores.size(); k++) {
                boolean notIn = true;

                for (int v : cores.get(k)) {
                    if (adj[i][v] != INF) {
                        notIn = false;
                        break;
                    }
                }
            }

            if (notIn) {
                cores.get(k).add(i);
            }
        }
    }
}
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
        colorido[i] = true;

        in = true;

        break;
    }
}

if (!in) {
    List<Integer> novaCor = new ArrayList<>();
    novaCor.add(i);
    cores.add(novaCor);
    colorido[i] = true;
}
}
}

return cores;
}

public void exibirArestas() {
    System.out.println("> Início da impressão das arestas (m = " + m + "): ");

    final int n = this.n;

    for(int i = 0; i < n; i++) {
        for(int j = 0; j < i; j++) {
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

```
        if(adj[i][j] != INF) {  
            System.out.println(this.rotulos.get(i) + " -- " + this.rotulos.get(j) + " = " + this.adj[i][j]);  
        }  
    }  
}  
  
System.out.println("> Fim da impressão das arestas.");  
  
}  
  
public void show() {  
    System.out.println("> Início da impressão do grafo (n = " + n + " e m = " + m + "): ");  
    for(int i = 0; i < n; i++){  
        System.out.println();  
  
        for(int w = 0; w < n; w++) {  
            if(adj[i][w] == INF) System.out.print("Adj[" + rotulos.get(i) + "," + rotulos.get(w) + "] = INF  
");  
            else System.out.print("Adj[" + rotulos.get(i) + "," + rotulos.get(w) + "] = " + adj[i][w] + " ");  
        }  
    }  
  
    System.out.println("\n\n> Fim da impressao do grafo. " );  
}  
}
```



4. grafo.txt

2

55

1

2

3

4

5

6

7

9

10

11

12

13

14

16

17

19

20

23

24

25



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

28

29

30

31

33

35

37

38

40

41

43

44

45

46

48

49

50

52

E101

E117

E143

E181

E249

E40



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

E44

E139

E85

E993

P01

P02

P03

P04

P05

P06

P07

103

23	24	23
----	----	----

24	25	30
----	----	----

25	31	66
----	----	----

31	33	57
----	----	----

29	30	40
----	----	----

30	31	8
----	----	---

5	6	90
---	---	----

6	24	80
---	----	----

5	23	54
---	----	----

6	20	33
---	----	----

20	25	65
----	----	----

20	28	89
----	----	----



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

33	37	19
35	37	30
37	38	24
38	41	27
40	41	42
38	40	5
40	P07	57
41	P07	10
33	35	10
E117	E139	20
E101	P07	180
E993	P07	47
E40	E44	38
E44	P07	270
E44	E101	210
28	45	92
29	52	38
45	52	110
20	45	28
48	49	41
48	52	45
49	52	81
49	50	73
50	52	120



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

45	P06	110
48	P06	67
46	P06	56
44	46	43
16	17	5
19	20	100
13	14	17
E143	P06	300
E85	E143	60
E143	E181	43
E85	P06	290
44	P05	60
P05	P06	62
17	19	35
1	2	23
2	3	22
3	4	31
4	5	26
5	P03	20
7	10	97
6	7	5
4	7	45
7	19	120
10	19	78



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

9	11	10
11	12	50
9	10	45
12	14	9
12	17	65
43	P01	14
13	P02	13
12	P02	4
1	P04	94
9	P04	110
P02	P04	110
P01	P02	38
P01	P05	150
P03	P04	55
P03	P07	400
P06	P07	350
12	13	22
46	48	50
1	9	49
2	9	53
3	9	62
6	19	130
9	19	75
11	19	29



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática



Prof. Dr. Ivan Carlos Alcântara de Oliveira
Disciplina: Teoria dos Grafos

11	17	51
16	19	40
6	23	5
28	31	65
28	30	50
28	29	18
29	45	54
30	33	30
30	35	60
29	31	58
35	40	40
35	38	40
45	46	63
45	48	100
E181	P05	57
E993	P06	450
E101	E117	32
E139	P03	150
E249	P03	62