SERVIÇO PÚBLICO FEDERAL · MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE VIÇOSA · UFV
CAMPUS FLORESTAL

Relatório Trabalho 1 - Grafos

UTILIZAÇÃO

PACOTE NETWORKX

MARIA EDUARDA DE PINHO BRAGA - 5099
JOÃO GABRIEL ANGELO BRADACHI - 5078
ARTHUR ATAIDE DE MELO SARAIVA - 5070
GUILHERME BROEDEL ZORZAL - 5064

Florestal - MG

2023

Sumário

Introdução

O seguinte trabalho tem por objetivo obter certas métricas e realizar cálculos em grafos utilizando o pacote NetworkX, que permite manipulá-los com mais facilidade, além de possuir algumas funções prontas.

Para hospedagem do programa, optamos por utilizar a plataforma de versionamento GitHub.

Metodologia

Para um maior entendimento do nosso código também optamos por dividi-lo em diretórios, sendo eles a pasta raiz com o arquivo principal do projeto (main.py) e a pasta entradas para os arquivos de entradas (veja figura 1). A divisão de tarefas a fim de otimizar o tempo foi a divisão da implementação de 4 funcionalidades para cada membro do grupo, mas salientamos que a comunicação para a construção do projeto foi contínua, gerando uma grande integração.

Figura 1 - Arquivos

Para compilar o projeto criamos um arquivo makefile. Portanto, basta

executar o comando make no terminal para rodar a aplicação.

Ademais, foi desenvolvido um menu com o propósito de tornar a utilização mais amigável para o usuário, permitindo que ele selecione o algoritmo ou a métrica desejada. Além disso, um menu inicial foi implementado para aprimorar as verificações no grafo. É altamente recomendado que o usuário ative este menu para uma experiência aprimorada, conforme demonstrado nas Figuras 2 e 3.

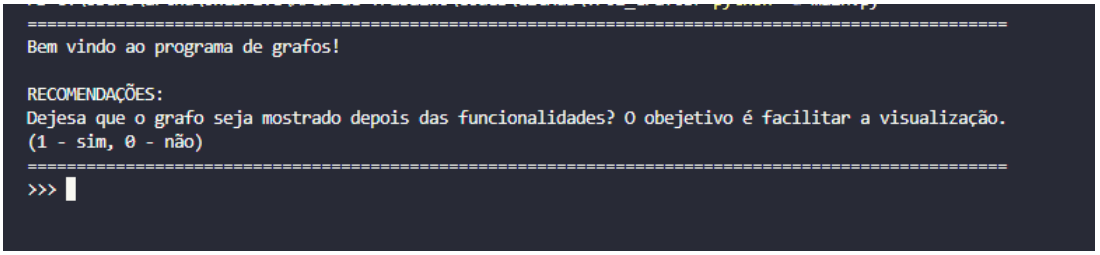

Figura 2 - recomendação de amostragem

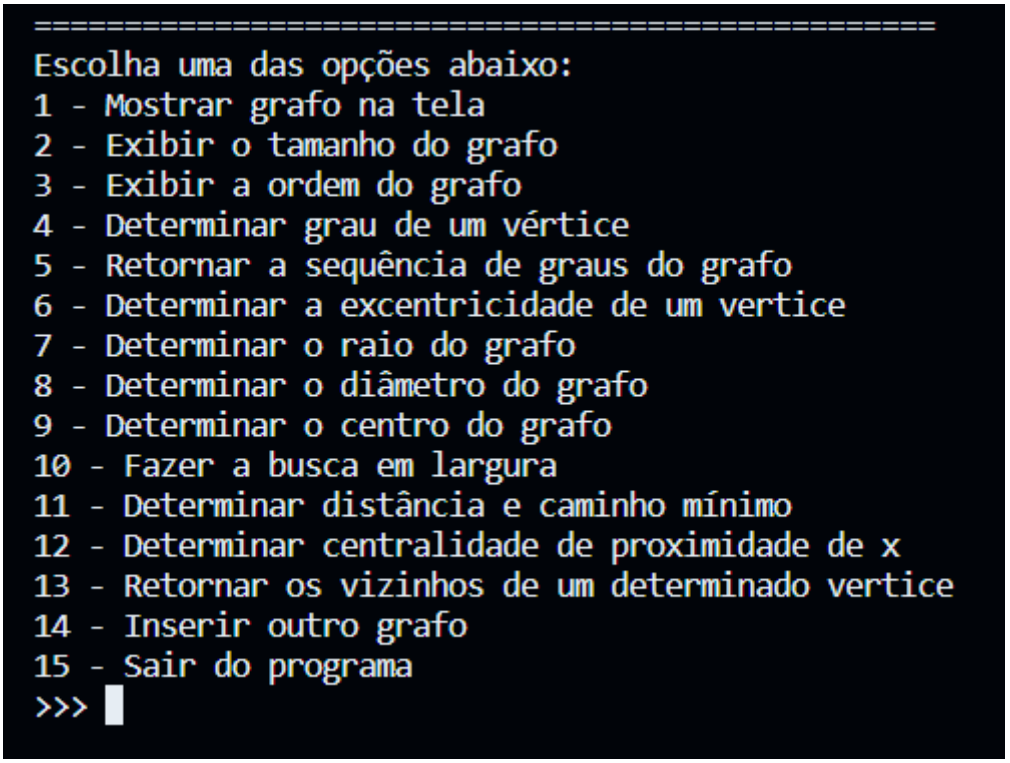

Figura 3 - Menu.

Outro ponto importante é que, a fim de tornar o código mais compatível, fizemos adaptações para que o mesmo consiga operar nos sistemas operacionais Linux e Windows.

Desenvolvimento

Como a biblioteca network trás funções prontas, a implementação do trabalho não teve grandes desafios, sendo o entendimento das funções a parte mais laboriosa.

Dentre alguns dos problemas encontrados, pode-se destacar a leitura de grafos ponderados. A leitura dos grafos ponderados se mostrou conflituosa, não sendo possível fazer a mesma somente com o pacote NetworkX. Dessa forma, utilizamos a biblioteca The ElementTree XML API para fazer essa leitura. A Figura 4 mostra como foi feita essa leitura.

```
# lendo o arquivo hml
limpar_tela()
nomeDoArquivo = str(input("Insira o nome do arquivo (sem a extensão .graphml) >>> "))
nomeDoArquivo =  "./entradas/"+nomeDoArquivo+".graphml"

# Ler o arquivo GraphML
tree = ET.parse(nomeDoArquivo)
root = tree.getroot()

# Crie um objeto Graph (grafo não direcionado) com o NetworkX
G = nx.Graph()

# Itere sobre as arestas no arquivo XML e adicione-as ao grafo NetworkX com seus pesos
for edge in root.findall(".//edge"):
    source = edge.get('source')
    target = edge.get('target')
    weight = float(edge.get('weight'))
    G.add_edge(source, target, weight=weight)
```

Figura 4 - Leitura dos Grafos.

Uma observação importante acerca do funcionamento da biblioteca networkx é o fato dela não ler vértices sozinhos durante a execução da leitura. Um exemplo disso pode ser visto abaixo (veja a diferença entre o arquivo de entrada na figura 5 e o exibido na tela na figura 6):

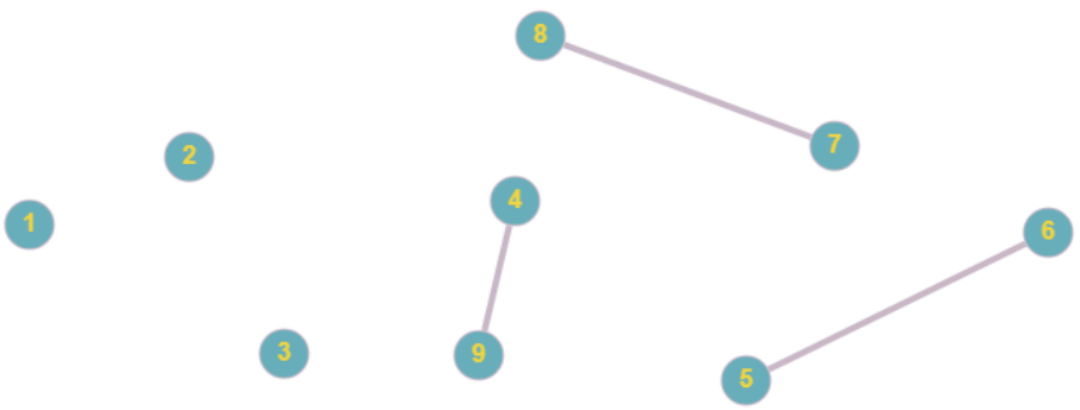

Figura 5 - Grafo de entrada passado para leitura

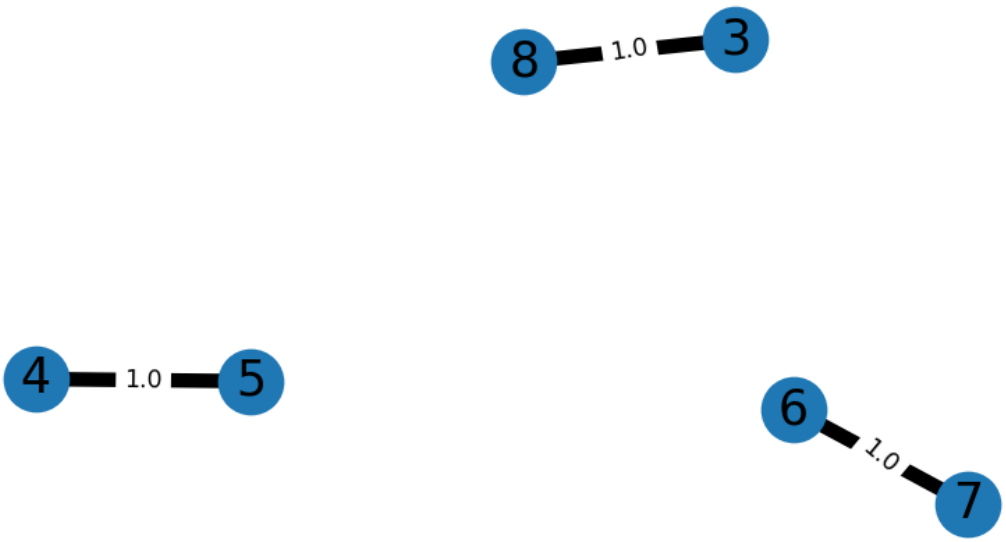

Figura 6 - Grafo exibido na saída

Outra observação interessante é o fato do arquivo graphml retornado pela biblioteca não seguir o mesmo padrão de xml utilizado, por exemplo, pelo site disponibilizado para construção dos grafos (https://graphonline.ru/pt/) e pela biblioteca de utilizada para leitura de grafos ponderados, a The ElementTree XML API. Após exportar o arquivo da busca em largura e tentar abri-lo com as ferramentas descritas, ou o grafo não abria ou ocorria perda de informações durante sua leitura. Veja o exemplo abaixo (Figura 7):

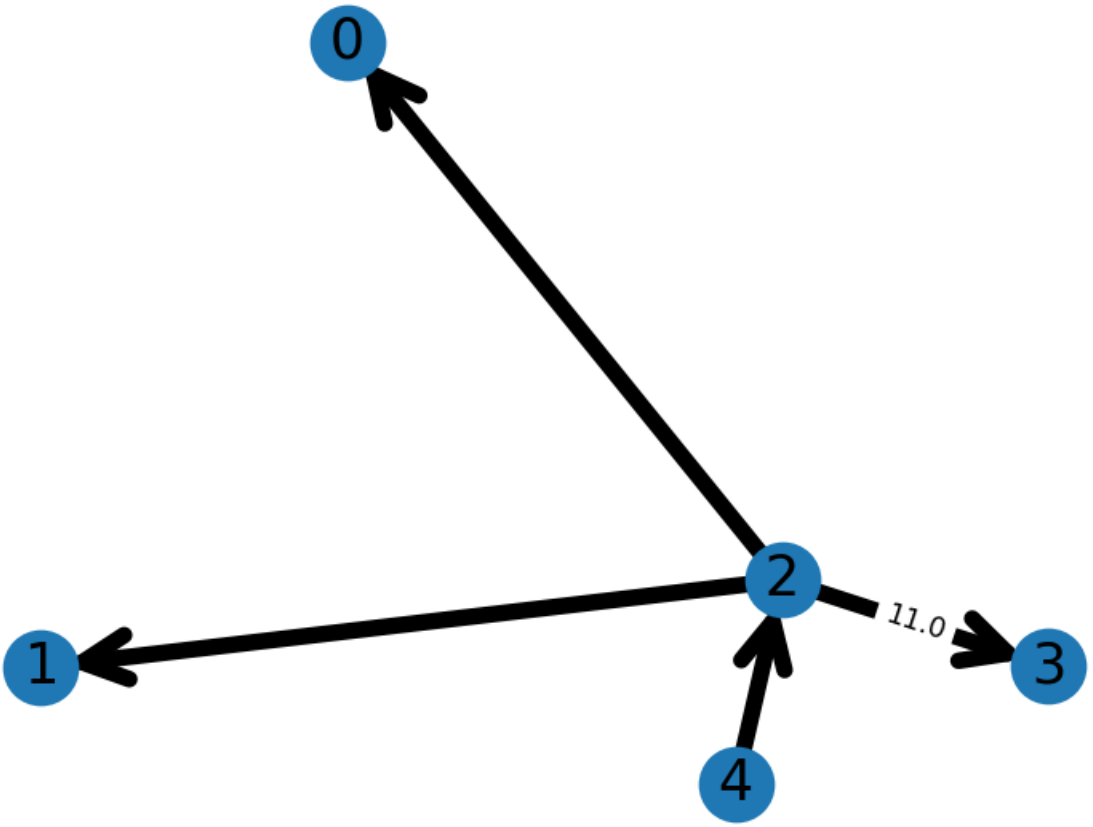

Figura 7 - Grafo gerado pela busca em largura e que é exibido pelo nosso programa

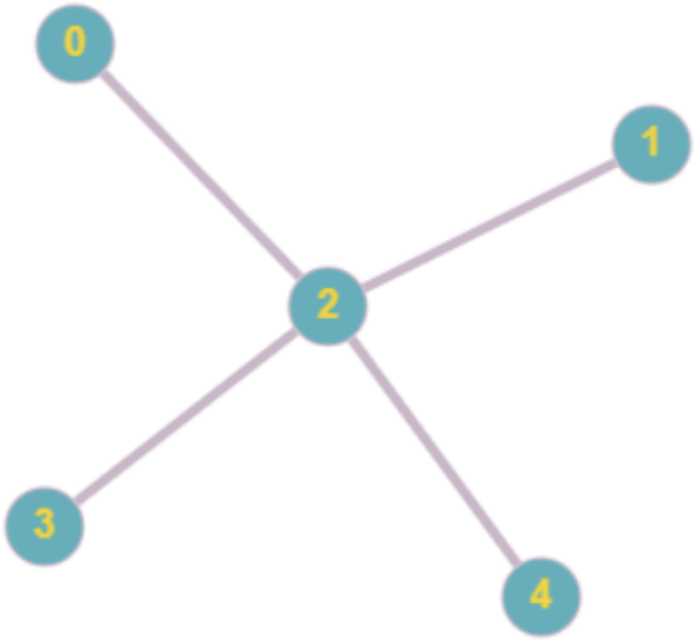

Figura 8 - graphml do grafo da figura 7 aberto no site (note que os arcos viraram arestas e ele também deixou de ser ponderado)

De modo geral, os demais resultados são gerados pela própria biblioteca

utilizada.

Resultados

Apesar dos problemas enfrentados e das dificuldades encontradas, a implementação do trabalho prático foi um sucesso. Para demonstrar, segue abaixo um exemplo de execução (para minimizar a quantidade de imagens e seu comprimento, o menu não será mostrado em todas elas):

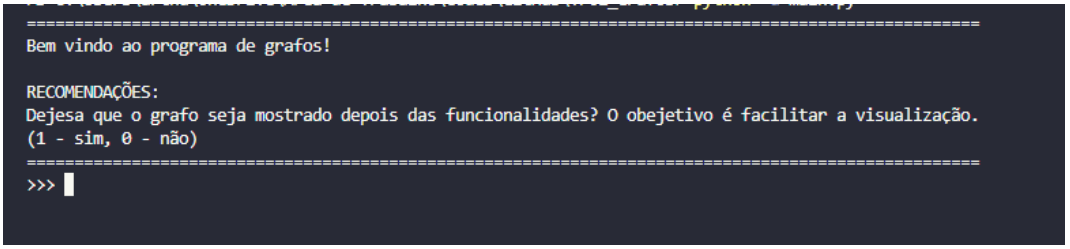

Figura 9 - Tela inicial.

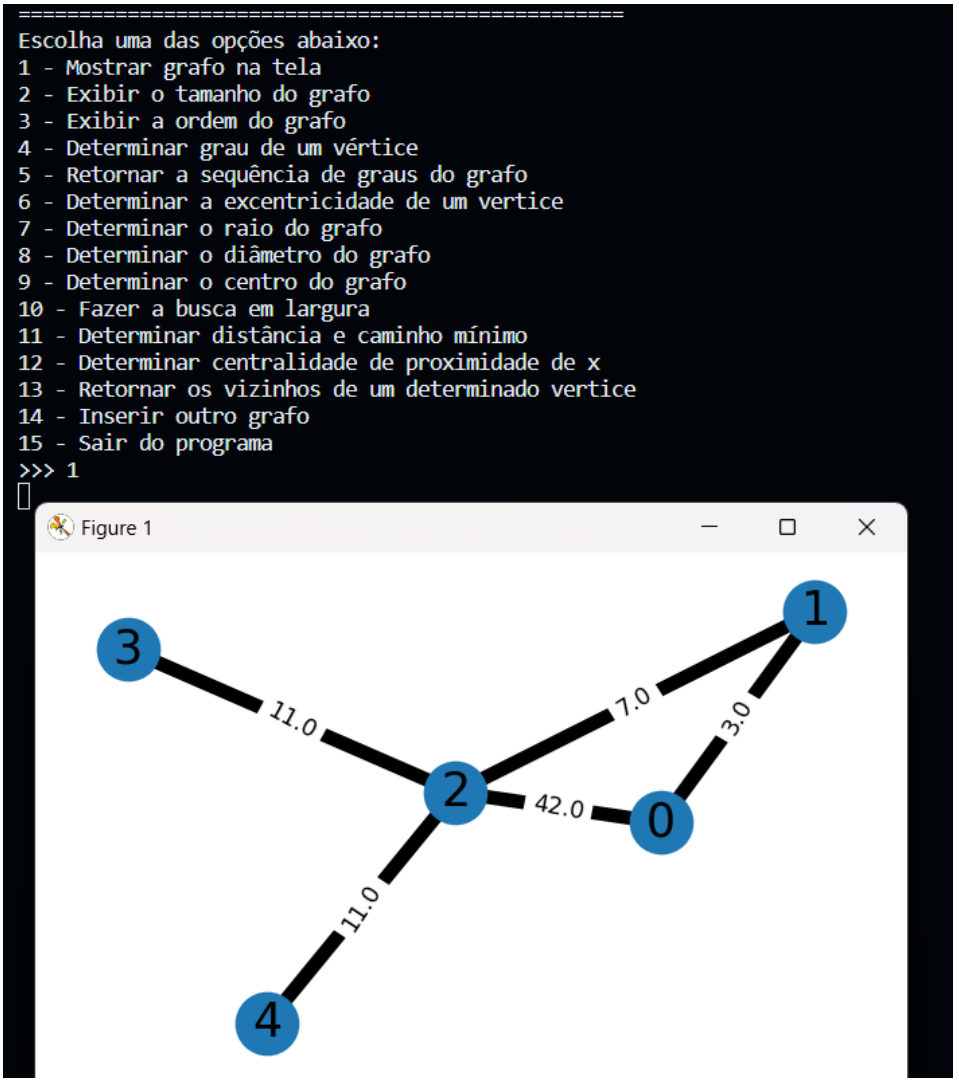

Figura 10 - Menu e opção de mostrar grafo na tela.

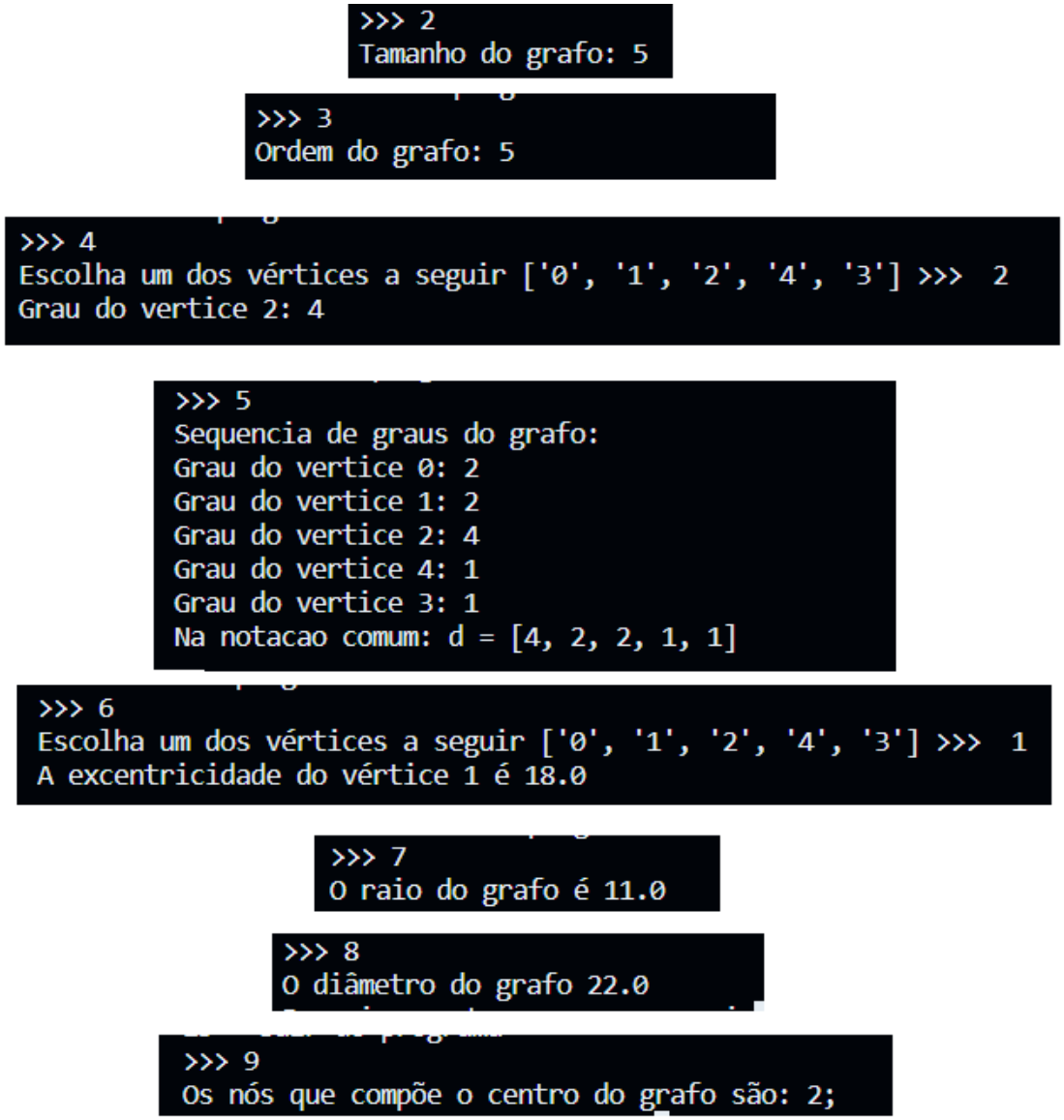

Figura 11.1 - Outras opções disponíveis no programa.

```
>>> 10
Escolha um dos vértices a seguir ['0', '1', '2', '4', '3'] >>>  4
Arestas que nao fazem parte da arvore:  [0, 1];
```

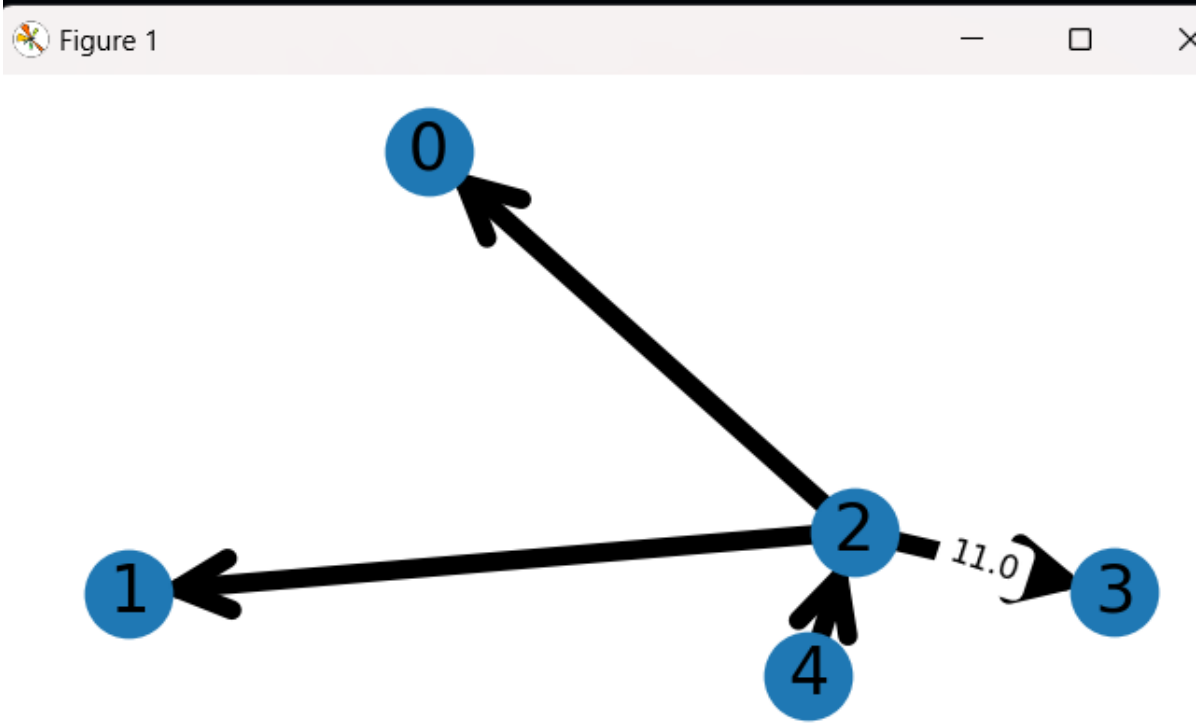

Figura 11.2 - Outras opções disponíveis no programa.

```
>>> 11
Escolha um dos vértices a seguir ['0', '1', '2', '4', '3'] >>>  0

Caminho mínimo do nó 0 até o nó 0: 0
Distância mínima do nó 0 até o nó 0: 0

Caminho mínimo do nó 0 até o nó 1: 0 -> 1
Distância mínima do nó 0 até o nó 1: 3.0

Caminho mínimo do nó 0 até o nó 2: 0 -> 1 -> 2
Distância mínima do nó 0 até o nó 2: 10.0

Caminho mínimo do nó 0 até o nó 4: 0 -> 1 -> 2 -> 4
Distância mínima do nó 0 até o nó 4: 21.0

Caminho mínimo do nó 0 até o nó 3: 0 -> 1 -> 2 -> 3
Distância mínima do nó 0 até o nó 3: 21.0
```

```
>>> 12
Escolha um dos vértices a seguir ['0', '1', '2', '4', '3'] >>>  0
Centralidade do vértice 0 = 0.07272727272727272
```

```
>>> 13
Escolha um dos vértices a seguir ['0', '1', '2', '4', '3'] >>>  0
Vizinhos de 0: 1; 2;
```

Figura 11.3 - Outras opções disponíveis no programa.

Conclusão

Portanto, podemos concluir que os objetivos propostos foram alcançados. Além disso, a exposição a novas ferramentas e sistemas auxiliou em muito no desenvolvimento individual e coletivo de todos os envolvidos. Também pode-se destacar que ocorreu um grande aprendizado e absorção do funcionamento e utilização da biblioteca networkx.

Referências

[1] Github. Disponível em: <https://github.com/JBradachi/TP01_Grafos> Último acesso em: 17 de outubro de 2023.

[2] Graph Online (Site para geração de grafos). Disponível em: <https://graphonline.ru/pt/> Último acesso em 26 de outubro de 2023.