

# Documentação Grafos

## Documentação do Sistema de Gerenciamento de Grafos

### Grupo: Os Teóricos dos Grafos

#### Integrantes:

- Théo Moura
- Israel Erlich
- Guilherme Alcoforado
- Rafael Menezes
- Victor Montarroyos
- Pedro Henrique

#### A) Tabela de Horas por Integrante e Atividade

Atividade	Théo	Israel	Guilherme	Rafael	Victor	Pedro	Total
Especificação	0	0.5	1	1	0.5	0.5	3.5
Codificação	2	1	0.5	0.5	1	1	6
Testes	0.5	0.5	0.5	0.5	1	0.5	3.5
<b>Total por Pessoa</b>	2.5	2	2	2	<b>2.5</b>	<b>2</b>	13

**Total Geral de Horas:** 8 horas

Comece instalando os requirements.txt!:

```
pip install requirements.txt
```

#### B) Manual de Funcionalidades do Sistema

##### Visão Geral

O sistema permite ao usuário criar e manipular grafos simples, direcionados ou não, valorados ou não. A interação ocorre via terminal, com opções claras e instruções para facilitar o uso.

##### 1. Iniciando o Sistema

Ao executar o programa, o usuário é recebido com a opção de inserir informações iniciais de um arquivo CSV ou começar com um grafo vazio.

##### Exemplo:

Se desejar inserir informações iniciais de um arquivo CSV, insira o nome do arquivo (ou pressione Enter para continuar):

##### 2. Definição do Tipo de Grafo

O sistema pergunta se o grafo será direcionado e/ou valorado.

##### Exemplo:

O grafo é direcionado? (S/N): S

O grafo é valorado? (S/N): N

##### 3. Menu de Opções

Após a definição, o usuário tem acesso ao menu principal:

Opções:

1. Adicionar vértice
2. Adicionar aresta

3. Inserir informações em lote de um arquivo CSV
4. Inserir itens em lote manualmente
5. Visualizar grafo
6. Obter ordem e tamanho do grafo
7. Obter lista de adjacentes de um vértice
8. Obter grau de um vértice
9. Verificar se dois vértices são adjacentes
10. Encontrar o caminho mais curto entre dois vértices
11. Verificar se o grafo é Euleriano
12. Sair

#### **4. Funcionalidades Detalhadas**

##### **4.1. Adicionar Vértice (Opção 1)**

Permite adicionar um vértice individualmente.

**Passos:**

- Selecionar a opção 1.
- Inserir o nome do vértice.

**Exemplo:**

Escolha uma opção: 1

Insira o nome do vértice: A

Vértice 'A' adicionado com sucesso!

##### **4.2. Adicionar Aresta (Opção 2)**

Adiciona uma aresta entre dois vértices existentes. Se o grafo for valorado, solicita o peso.

**Passos:**

- Selecionar a opção 2.
- Inserir os vértices de partida e chegada.
- Se aplicável, inserir o peso.

**Exemplo:**

Escolha uma opção: 2

Vértices disponíveis: {'A', 'B'}

Insira o nome do vértice de partida: A

Insira o nome do vértice de chegada: B

Aresta adicionada entre 'A' e 'B'!

##### **4.3. Inserir Informações em Lote de um Arquivo CSV (Opção 3)**

Carrega vértices e arestas de um arquivo CSV com colunas 'Source', 'Target' e opcionalmente 'Weight'.

**Passos:**

- Selecionar a opção 3.
- Inserir o nome do arquivo CSV.

**Exemplo:**

Escolha uma opção: 3

Insira o nome do arquivo CSV: grafo.csv

Informações em lote do arquivo inseridas com sucesso!

#### 4.4. Inserir Itens em Lote Manualmente (Opção 4)

Permite inserir múltiplos vértices e arestas de uma só vez.

##### Passos:

- Selecionar a opção 4.
- Inserir os vértices separados por espaço.
- Inserir as arestas no formato especificado.

##### Exemplo:

Escolha uma opção: 4

Digite os vértices separados por espaço: C D E

Digite as arestas no formato 'v1 v2 [peso]', separadas por vírgula:

A B, B C, C D

#### 4.5. Visualizar Grafo (Opção 5)

Mostra os vértices e arestas atuais e exibe uma representação gráfica.

##### Exemplo:

Escolha uma opção: 5

Visualizando grafo...

Vértices: ['A', 'B', 'C', 'D', 'E']

Arestas: [('A', 'B', {}), ('B', 'C', {}), ('C', 'D', {})]

*Uma janela gráfica será aberta exibindo o grafo.*

#### 4.6. Obter Ordem e Tamanho do Grafo (Opção 6)

Informa o número de vértices (ordem) e arestas (tamanho) do grafo.

##### Exemplo:

Escolha uma opção: 6

Ordem do grafo (número de vértices): 5

Tamanho do grafo (número de arestas): 3

#### 4.7. Obter Lista de Adjacentes de um Vértice (Opção 7)

Mostra os vértices adjacentes ao vértice informado.

##### Exemplo:

Escolha uma opção: 7

Insira o vértice para obter suas adjacências: B

Vértices adjacentes: ['A', 'C']

#### 4.8. Obter Grau de um Vértice (Opção 8)

Informa o grau do vértice. Em grafos direcionados, mostra graus de entrada e saída.

##### Exemplo:

Escolha uma opção: 8

Insira o vértice para obter seu grau: B

Grau: 2

#### 4.9. Verificar Adjacência entre Vértices (Opção 9)

Verifica se dois vértices são adjacentes.

**Exemplo:**

Escolha uma opção: 9

Insira o primeiro vértice: A

Insira o segundo vértice: C

Os vértices 'A' e 'C' não são adjacentes.

**4.10. Encontrar o Caminho Mais Curto (Opção 10)**

Calcula o menor caminho entre dois vértices, mostrando o custo e a sequência de vértices.

**Exemplo:**

Escolha uma opção: 10

Insira o vértice de origem: A

Insira o vértice de destino: D

Custo do menor caminho: 3

Menor caminho: A → B → C → D

**4.11. Verificar se o Grafo é Euleriano (Opção 11)**

Informa se o grafo é Euleriano.

**Exemplo:**

Escolha uma opção: 11

O grafo não é Euleriano.

**4.12. Sair (Opção 12)**

Encerra o programa.

**C) Descrição do Código-Fonte****Visão Geral**

O sistema foi desenvolvido em Python utilizando as bibliotecas networkx para manipulação de grafos e matplotlib para visualização gráfica. O código está estruturado em funções para facilitar a manutenção e a compreensão.

**Estrutura de Arquivos**

- main.py: Arquivo principal que contém toda a lógica do sistema.

**Principais Bibliotecas Utilizadas**

- networkx: Manipulação e análise de grafos.
- matplotlib.pyplot: Visualização gráfica dos grafos.
- csv: Leitura de arquivos CSV.

**Descrição das Funções****create\_graph()**

- **Descrição:** Interage com o usuário para definir se o grafo será direcionado e/ou valorado. Inicializa o objeto Graph ou DiGraph do networkx.
- **Parâmetros:** Nenhum.
- **Retorno:** Objeto do grafo e variável booleana indicando se é valorado.

**insert\_batch\_info(filename, graph, vertices, weighted)**

- **Descrição:** Lê um arquivo CSV e insere vértices e arestas no grafo.
- **Parâmetros:**
  - filename: Nome do arquivo CSV.
  - graph: Objeto do grafo.

- vertices: Conjunto de vértices existentes.
- weighted: Booleano indicando se o grafo é valorado.

#### **insert\_batch\_items(G, weighted)**

- **Descrição:** Permite inserir múltiplos vértices e arestas manualmente em lote.
- **Parâmetros:**
  - G: Objeto do grafo.
  - weighted: Booleano indicando se o grafo é valorado.

#### **create\_graph\_from\_file(filename)**

- **Descrição:** Função principal que controla o fluxo do programa e apresenta o menu de opções.
- **Parâmetros:**
  - filename: Nome do arquivo CSV inicial (pode ser vazio).

#### **Como Executar o Sistema**

##### **1. Pré-requisitos:**

- Python 3 instalado.
- Bibliotecas networkx, matplotlib e csv instaladas.
- Comando para instalar as bibliotecas:

pip install networkx matplotlib

##### **2. Execução:**

- Navegue até o diretório do arquivo main.py.
- Execute o comando:

python main.py

- Siga as instruções apresentadas no terminal.

#### **Explicação da Implementação**

- **Interação com o Usuário:**
  - Utiliza funções input() para receber dados.
  - Apresenta um menu para facilitar a navegação.
- **Manipulação do Grafo:**
  - Dependendo das escolhas iniciais, utiliza nx.Graph() para grafos não direcionados ou nx.DiGraph() para grafos direcionados.
  - As arestas e vértices são armazenados nos objetos do networkx, permitindo fácil manipulação e análise.
- **Visualização:**
  - A função nx.draw() é utilizada para desenhar o grafo.
  - matplotlib exibe o grafo em uma janela gráfica.
- **Cálculo de Menor Caminho:**
  - Para grafos valorados, utiliza o algoritmo de Dijkstra (nx.dijkstra\_path).
  - Para grafos não valorados, utiliza o algoritmo de busca em largura (nx.shortest\_path).
- **Verificação Euleriana:**
  - Utiliza a função nx.is\_eulerian() para determinar se o grafo é Euleriano.