

# Projeto 1 Teoria dos Grafos Relatório

## 1. Introdução

O presente relatório tem como objetivo documentar o desenvolvimento de um sistema de manipulação de grafos voltado para aplicações de localização urbana, denominado **Cajo Traffic System**. O sistema permite carregar, salvar e manipular grafos representados por matrizes de adjacência, simulando a estrutura de uma malha urbana. A aplicação foi dividida em três módulos principais:

- `GrafoMatriz.py` : implementação da estrutura de dados do grafo.
- `GrafoTeste.py` : operações e funções de entrada/saída para manipulação do grafo.
- `main.py` : interface de menu interativo com o usuário.

## 2. Objetivos

- Implementar uma estrutura de grafo orientado utilizando matriz de adjacência.
- Permitir operações como inserção e remoção de vértices e arestas.
- Permitir o carregamento e salvamento de dados em arquivos `.txt`.
- Determinar propriedades do grafo como grau, simetria e conexidade.

## 3. Estrutura dos Arquivos

### 3.1 `GrafoMatriz.py`

Contém a classe `Grafo`, que representa um grafo direcionado com as seguintes funcionalidades:

- Inserção e remoção de vértices e arestas.
- Impressão da matriz de adjacência.
- Cálculo de grau de entrada, grau de saída, e grau total.
- Verificação de simetria.
- Identificação de fontes e sorvedouros.
- Determina se o grafo é conexo (alcançável a partir do vértice 0).

Também inclui as funções:

- `carregar_grafo(arquivo)` : carrega um grafo de um arquivo `.txt`.
- `salvar_grafo(arquivo, grafo, nomes_vertices)` : salva o grafo em arquivo `.txt`.

### 3.2 GrafoTeste.py

Implementa as funcionalidades de interação com o grafo:

- Leitura e gravação de dados.
- Inserção/remoção de vértices e arestas.
- Impressão do conteúdo do arquivo.
- Impressão da matriz do grafo.
- Apresentação da conexidade.

### 3.3 main.py

Arquivo principal da aplicação com um menu interativo que permite ao usuário:

- Ler e gravar dados no arquivo `grafo.txt`.
- Inserir/remover vértices e arestas.
- Visualizar o grafo e seu estado no arquivo.
- Verificar se o grafo é conexo e apresentar a forma reduzida.
- Encerrar a aplicação.

## 4. Funcionamento Geral do Sistema

O sistema opera por meio de um menu textual com as seguintes opções:

- **a)** Ler dados de `grafo.txt`
- **b)** Gravar dados em `grafo.txt`
- **c)** Inserir vértice (com rótulo definido pelo usuário)
- **d)** Inserir aresta (entre dois vértices definidos)
- **e)** Remover vértice (e atualizar a matriz de adjacência)
- **f)** Remover aresta
- **g)** Mostrar conteúdo do arquivo
- **h)** Mostrar grafo (estado atual)
- **i)** Apresentar conexidade do grafo
- **j)** Encerrar aplicação

As funcionalidades interagem diretamente com o grafo em memória e com o arquivo `grafo.txt`, permitindo edições persistentes e dinâmicas na estrutura do grafo.



## 5. Testes Recomendados

- Testar a inserção e remoção de vértices em diferentes posições.

- Testar leitura e gravação com grafos de diferentes tamanhos.
- Verificar o comportamento de conexidade em grafos com componentes isolados.
- Testar simetria com grafos direcionados e não-direcionados.

## Conclusão

Este sistema apresenta uma abordagem funcional e educacional para o estudo e manipulação de grafos direcionados. Ele pode ser expandido para suportar diferentes aplicações, como redes sociais, rotas de tráfego ou mapas de navegação, como proposto no Cajo Traffic System.

Se quiser, posso gerar uma versão em PDF desse relatório ou exportar para um .docx também.  
Deseja isso?