



Relatório do Projeto

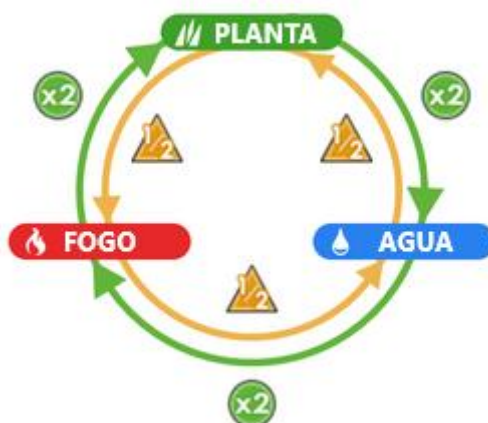
Parte 1

Nome do Integrante	RA
Raphael Iniesta Reis	10396285
Tomás Fiorelli Barbosa	10395687

Relatório

1. DEFINIÇÃO DO PROBLEMA

O tema escolhido pelo grupo foi a interação de ataque e defesa dos tipos elementais presentes no jogo *Pokémon*. Em resumo, um ataque de qualquer elemento possui um coeficiente de efetividade (sem efeito - 0, pouco efetivo - 0.5, efetivo - 1 ou super efetivo - 2) contra um *Pokémon* de um certo tipo – como por exemplo, um ataque do elemento água é super efetivo contra um *Pokémon* do tipo fogo e é pouco efetivo contra um *Pokémon* do tipo planta (podemos falar também que o *Pokémon* do tipo planta é resistente à ataques do tipo água). Abaixo está uma representação simples do “triângulo inicial” de elementos, pois consiste nos tipos de Pokémon que podem ser escolhidos no começo dos jogos da franquia.



Representação do triângulo de efetividade dos tipos iniciais de Pokémon

Considerando todos os elementos, são definidos 18 tipos únicos e, em média, cada um se relaciona com os outros 17 tipos (há exceções, como por exemplo, ataques elétricos não tem efeito em *Pokémons* terrestre), totalizando um valor de 298 interações entre os elementos.

OBS: para este projeto, será desconsiderado dois fatores em função de como será trabalhado o grafo produzido. O primeiro é que será desconsiderado a interação de elementos com eles mesmos, algo que pode acontecer normalmente no jogo (por exemplo, fogo é pouco efetivo contra fogo), já que fomos instruídos a não trabalhar com laços; e o segundo é que *Pokémons* podem (ou não) possuir 2 tipos elementais, ocasionando de gerar mais vértices e arestas (171 combinações em que cada interagiria com um dos 18 possíveis tipos de ataques – totalizando em aproximadamente 3.000 interações). Trabalharemos com 1 tipo para reduzir o escopo do trabalho

2. MODELAGEM DO PROBLEMA



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



O arquivo "grafo.txt" enviado junto de todos os documentos possui a modelagem para o problema das interações dos tipos de *Pokémon*. Ele foi desenvolvido seguindo todos os critérios requisitados pelo enunciado, em que cada linha representa um certo elemento do grafo. Segue abaixo uma parte do arquivo (devido a sua quantidade de linhas, retiramos a maioria das arestas):

```
6
18
NORMAL
FOGO
ÁGUA
ELÉTRICO
PLANTA
GELO
LUTADOR
VENENOSO
TERRESTRE
VOADOR
PSÍQUICO
INSETO
PEDRA
FANTASMA
DRAGÃO
SOMBRIO
AÇO
FADA
306
NORMAL FOGO 1
NORMAL ÁGUA 1
NORMAL ELÉTRICO 1
NORMAL PLANTA 1
NORMAL GELO 1
NORMAL LUTADOR 1
NORMAL VENENOSO 1
NORMAL TERRESTRE 1
NORMAL Voador 1
NORMAL PSÍQUICO 1
NORMAL INSETO 1
NORMAL PEDRA 0.5
NORMAL FANTASMA 0
NORMAL DRAGÃO 1
NORMAL SOMBRIO 1
NORMAL AÇO 0.5
NORMAL FADA 1
FOGO NORMAL 1
FOGO ÁGUA 0.5
FOGO ELÉTRICO 1
FOGO PLANTA 2
FOGO GELO 2
FOGO LUTADOR 1
FOGO VENENOSO 1
FOGO TERRESTRE 1
...
FADA AÇO 0.5
```



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

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



3. DESENVOLVIMENTO DA APLICAÇÃO

Nessa etapa, será apresentado as funcionalidades do programa desenvolvido, seguindo as opções disponíveis no menu

Menu de opções

```
=== RELAÇÃO DOS TIPOS DE POKÉMON EM GRAFO ===  
  
Selecione uma opção:  
[1] Ler dados do arquivo grafo.txt  
[2] Gravar dados no arquivo grafo.txt  
[3] Inserir vértice  
[4] Inserir aresta  
[5] Remover vértice  
[6] Remover aresta  
[7] Mostrar conteúdo do arquivo  
[8] Mostrar grafo  
[9] Apresentar a conexidade do grafo e o reduzido  
[0] Encerrar a aplicação  
  
Opção: █
```

Ler dados do arquivo grafo.txt

```
69     def lerArquivo(self, nome_arquivo):  
70         with open(nome_arquivo, 'r', encoding="utf-8") as arquivo:  
71             self.tipo_grafo = int(arquivo.readline())  
72  
73             quantidade_vertices = int(arquivo.readline().strip())  
74             for _ in range(quantidade_vertices):  
75                 self.insereV(arquivo.readline().strip())  
76  
77             quantidade_arestas = int(arquivo.readline())  
78             for _ in range(quantidade_arestas):  
79                 origem, destino, peso = arquivo.readline().split()  
80                 peso_float = float(peso)  
81                 self.insereA(origem, destino, peso_float)
```

```
Opção: 1  
Digite o nome do arquivo: grafo.txt  
Dados do arquivo grafo.txt lidos com sucesso.  
  
Pressione [ENTER] para voltar █
```

Gravar dados no arquivo grafo.txt



```
83  def salvarArquivo(self, nome_arquivo):
84      with open(nome_arquivo, 'w', encoding="utf-8") as arquivo:
85          # Escreve o tipo de grafo
86          arquivo.write("6\n")
87
88          # Escreve a quantidade de vértices
89          arquivo.write(f"{len(self.adj)}\n")
90
91          # Escreve os rótulos dos vértices
92          for vertice in self.adj.keys():
93              arquivo.write(f"{vertice}\n")
94
95          # Escreve a quantidade de arestas
96          total_arestas = sum(len(vizinhos) for vizinhos in self.adj.values())
97          arquivo.write(f"{total_arestas}\n")
98
99          # Escreve as arestas e pesos
100         for vertice, vizinhos in self.adj.items():
101             for vizinho, peso in vizinhos.items():
102                 arquivo.write(f"{vertice} {vizinho} {peso}\n")
```

Opção: 2

Digite o nome do arquivo: grafo.txt

Dados gravados no arquivo grafo.txt com sucesso.

Pressione [ENTER] para voltar

Inserir vértice; Remove vértice

```
11  def insereV(self, v):
12      if v not in self.adj:
13          self.adj[v] = {}
14          self.qtde_vertices += 1
15
16  def removeV(self, vertice):
17      if vertice in self.adj:
18          del self.adj[vertice]
19          self.qtde_vertices -= 1
20          for v in self.adj: # Remove todas as arestas que envolvem o vértice
21              if vertice in self.adj[v]:
22                  del self.adj[v][vertice]
23                  self.qtde_arestas -= 1
24          print(f"Vértice '{vertice}' removido com sucesso e todas as suas arestas relacionadas.")
25      else:
26          print(f"0 vértice '{vertice}' não existe no grafo.")
```



Inserir aresta; Remove aresta

```
28     def insereA(self, v, w, peso=1.0):
29         if v not in self.adj:
30             self.adj[v] = {}
31         if w not in self.adj:
32             self.adj[w] = {}
33         self.adj[v][w] = peso
34         self.qtde_arestas += 1
35
36     def removeA(self, v, w):
37         if v in self.adj and w in self.adj[v]:
38             del self.adj[v][w]
39         self.qtde_arestas -= 1
```

Opção: 3
Digite o nome do vértice: KENZO
Vértice 'KENZO' inserido com sucesso.
Pressione [ENTER] para voltar

Opção: 8
=== MATRIZ DE ADJACÊNCIA ===
KEN
KEN -

Opção: 5
Digite o nome do vértice a ser removido: KENZO
Vértice 'KENZO' removido com sucesso e todas as suas arestas relacionadas.
Pressione [ENTER] para voltar

Mostrar conteúdo do arquivo

```
41     def show(self):
42         print("\nGRAFO [6] - grafo orientado com peso na aresta")
43         print(f"Quantidade de vértices: {self.qtde_vertices}")
44         print(f"Quantidade de arestas: {self.qtde_arestas}\n")
45         for v, vizinhos in self.adj.items():
46             print(f"{v[:3]}: ", end="")
47             for vizinho, peso in vizinhos.items():
48                 print(f"[{vizinho[:3]}] = {peso}", end=" | ")
49             print()
50         print("\nFim da impressao do grafo.\n\n")
```

Opção: 7

GRAFO [6] - grafo orientado com peso na aresta
Quantidade de vértices: 18
Quantidade de arestas: 306

NOR:	[FOR] - 1.0	[ÁQU] - 1.0	[ELÉ] - 1.0	[PLA] - 1.0	[GEL] - 1.0	[LUT] - 1.0	[VEN] - 1.0	[TER] - 1.0	[VOA] - 1.0	[PSI] - 1.0	[INS] - 1.0	[PED] - 0.5	[FAN] - 0.0	[DRA] - 1.0	[SOM] - 1.0	[ACO] - 0.5	[FAO] - 1.0
FOR:	[NOR] - 1.0	[ÁQU] - 0.5	[ELÉ] - 1.0	[PLA] - 2.0	[GEL] - 2.0	[LUT] - 1.0	[VEN] - 1.0	[TER] - 1.0	[VOA] - 1.0	[PSI] - 1.0	[INS] - 2.0	[PED] - 0.5	[FAN] - 1.0	[DRA] - 0.5	[SOM] - 1.0	[ACO] - 2.0	[FAO] - 1.0
ÁQU:	[NOR] - 1.0	[FOR] - 2.0	[ELÉ] - 1.0	[PLA] - 0.5	[GEL] - 1.0	[LUT] - 1.0	[VEN] - 1.0	[TER] - 2.0	[VOA] - 1.0	[PSI] - 1.0	[INS] - 1.0	[PED] - 2.0	[FAN] - 1.0	[DRA] - 0.5	[SOM] - 1.0	[ACO] - 1.0	[FAO] - 1.0
ELÉ:	[NOR] - 1.0	[FOR] - 1.0	[ÁQU] - 2.0	[PLA] - 0.5	[GEL] - 1.0	[LUT] - 1.0	[VEN] - 1.0	[TER] - 0.0	[VOA] - 2.0	[PSI] - 1.0	[INS] - 1.0	[PED] - 1.0	[FAN] - 1.0	[DRA] - 0.5	[SOM] - 1.0	[ACO] - 1.0	[FAO] - 1.0
PLA:	[NOR] - 1.0	[FOR] - 0.5	[ÁQU] - 2.0	[ELÉ] - 1.0	[GEL] - 1.0	[LUT] - 1.0	[VEN] - 2.0	[TER] - 0.5	[VOA] - 0.5	[PSI] - 1.0	[INS] - 2.0	[PED] - 1.0	[FAN] - 1.0	[DRA] - 0.5	[SOM] - 1.0	[ACO] - 0.5	[FAO] - 1.0
GEL:	[NOR] - 1.0	[FOR] - 1.0	[ÁQU] - 1.0	[ELÉ] - 1.0	[PLA] - 2.0	[LUT] - 1.0	[VEN] - 1.0	[TER] - 1.0	[VOA] - 2.0	[PSI] - 1.0	[INS] - 1.0	[PED] - 1.0	[FAN] - 1.0	[DRA] - 2.0	[SOM] - 1.0	[ACO] - 0.5	[FAO] - 1.0
LUT:	[NOR] - 1.0	[FOR] - 1.0	[ÁQU] - 1.0	[ELÉ] - 1.0	[PLA] - 1.0	[GEL] - 2.0	[VEN] - 1.0	[TER] - 1.0	[VOA] - 0.5	[PSI] - 0.0	[INS] - 0.5	[PED] - 1.0	[FAN] - 0.0	[DRA] - 1.0	[SOM] - 1.0	[ACO] - 1.0	[FAO] - 1.0
VEN:	[NOR] - 1.0	[FOR] - 1.0	[ÁQU] - 1.0	[ELÉ] - 1.0	[PLA] - 0.5	[GEL] - 2.0	[LUT] - 1.0	[TER] - 1.0	[VOA] - 1.0	[PSI] - 1.0	[INS] - 0.5	[PED] - 1.0	[FAN] - 1.0	[DRA] - 1.0	[SOM] - 1.0	[ACO] - 1.0	[FAO] - 1.0
TER:	[NOR] - 1.0	[FOR] - 2.0	[ÁQU] - 1.0	[ELÉ] - 2.0	[PLA] - 1.0	[GEL] - 1.0	[LUT] - 1.0	[VEN] - 0.5	[VOA] - 0.0	[PSI] - 1.0	[INS] - 1.0	[PED] - 2.0	[FAN] - 1.0	[DRA] - 1.0	[SOM] - 1.0	[ACO] - 0.5	[FAO] - 1.0
VOA:	[NOR] - 1.0	[FOR] - 1.0	[ÁQU] - 1.0	[ELÉ] - 0.5	[PLA] - 2.0	[GEL] - 1.0	[LUT] - 2.0	[VEN] - 1.0	[TER] - 1.0	[PSI] - 1.0	[INS] - 0.5	[PED] - 0.0	[FAN] - 1.0	[DRA] - 1.0	[SOM] - 1.0	[ACO] - 0.5	[FAO] - 1.0
PSI:	[NOR] - 1.0	[FOR] - 1.0	[ÁQU] - 1.0	[ELÉ] - 1.0	[PLA] - 1.0	[GEL] - 1.0	[LUT] - 0.0	[VEN] - 1.0	[TER] - 1.0	[VOA] - 1.0	[INS] - 1.0	[PED] - 1.0	[FAN] - 0.0	[DRA] - 1.0	[SOM] - 2.0	[ACO] - 0.5	[FAO] - 1.0
INS:	[NOR] - 1.0	[FOR] - 0.5	[ÁQU] - 1.0	[ELÉ] - 1.0	[PLA] - 0.5	[GEL] - 1.0	[LUT] - 0.5	[VEN] - 2.0	[TER] - 1.0	[VOA] - 2.0	[PSI] - 1.0	[PED] - 1.0	[FAN] - 1.0	[DRA] - 1.0	[SOM] - 1.0	[ACO] - 1.0	[FAO] - 1.0
PED:	[NOR] - 1.0	[FOR] - 2.0	[ÁQU] - 1.0	[ELÉ] - 0.5	[PLA] - 2.0	[GEL] - 1.0	[LUT] - 0.5	[VEN] - 1.0	[TER] - 1.0	[VOA] - 2.0	[PSI] - 1.0	[INS] - 1.0	[FAN] - 1.0	[DRA] - 1.0	[SOM] - 1.0	[ACO] - 0.5	[FAO] - 1.0
FAN:	[NOR] - 0.0	[FOR] - 1.0	[ÁQU] - 1.0	[ELÉ] - 1.0	[PLA] - 1.0	[GEL] - 1.0	[LUT] - 0.0	[VEN] - 1.0	[TER] - 1.0	[VOA] - 1.0	[PSI] - 2.0	[INS] - 1.0	[PED] - 1.0	[DRA] - 1.0	[SOM] - 0.5	[ACO] - 1.0	[FAO] - 1.0
DRA:	[NOR] - 1.0	[FOR] - 1.0	[ÁQU] - 1.0	[ELÉ] - 1.0	[PLA] - 1.0	[GEL] - 1.0	[LUT] - 1.0	[VEN] - 1.0	[TER] - 1.0	[VOA] - 1.0	[PSI] - 1.0	[INS] - 1.0	[PED] - 1.0	[FAN] - 1.0	[SOM] - 1.0	[ACO] - 0.5	[FAO] - 2.0
SOM:	[NOR] - 1.0	[FOR] - 1.0	[ÁQU] - 1.0	[ELÉ] - 1.0	[PLA] - 1.0	[GEL] - 1.0	[LUT] - 0.5	[VEN] - 1.0	[TER] - 1.0	[VOA] - 1.0	[PSI] - 2.0	[INS] - 2.0	[PED] - 1.0	[FAN] - 2.0	[DRA] - 0.0	[ACO] - 1.0	[FAO] - 1.0
ACO:	[NOR] - 2.0	[FOR] - 0.5	[ÁQU] - 0.5	[ELÉ] - 0.5	[PLA] - 1.0	[GEL] - 2.0	[LUT] - 2.0	[VEN] - 0.0	[TER] - 2.0	[VOA] - 0.5	[PSI] - 0.5	[INS] - 0.5	[PED] - 0.5	[FAN] - 1.0	[DRA] - 0.0	[SOM] - 1.0	[FAO] - 1.0
FAO:	[NOR] - 1.0	[FOR] - 0.5	[ÁQU] - 1.0	[ELÉ] - 1.0	[PLA] - 1.0	[GEL] - 1.0	[LUT] - 2.0	[VEN] - 0.5	[TER] - 1.0	[VOA] - 1.0	[PSI] - 1.0	[INS] - 1.0	[PED] - 1.0	[FAN] - 1.0	[DRA] - 2.0	[SOM] - 2.0	[ACO] - 0.5

Fim da impressao do grafo.



Mostrar grafo

```
52 def showMin(self):
53     vertices = self.adj.keys()
54     print("\n === MATRIZ DE ADJACÊNCIA ===")
55     print(" " * 4, end="")
56     for v in vertices:
57         print(f"{v[:3]:^6}", end="")
58     print()
59     for v1 in vertices:
60         print(f"{v1[:3]:<4}", end="")
61         for v2 in vertices:
62             if v2 in self.adj.get(v1, {}):
63                 print(f"{self.adj[v1][v2]:^6.1f}", end="")
64             else:
65                 print(f"'-' :^6", end="")
66         print()
67     print("\nFim da impressao da matriz de adjacência.\n\n")
```

Opção: 8

=== MATRIZ DE ADJACÊNCIA ===

	NOR	FOG	ÁGU	ELÉ	PLA	GEL	LUT	VEN	TER	VOA	PSÍ	INS	PED	FAN	DRA	SOM	AÇO	FAD
NOR	-	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.0	1.0	1.0	0.5	1.0
FOG	1.0	-	0.5	1.0	2.0	2.0	1.0	1.0	1.0	1.0	1.0	2.0	0.5	1.0	0.5	1.0	2.0	1.0
ÁGU	1.0	2.0	-	1.0	0.5	1.0	1.0	1.0	2.0	1.0	1.0	1.0	2.0	1.0	0.5	1.0	1.0	1.0
ELÉ	1.0	1.0	2.0	-	0.5	1.0	1.0	1.0	0.0	2.0	1.0	1.0	1.0	1.0	0.5	1.0	1.0	1.0
PLA	1.0	0.5	2.0	1.0	-	1.0	1.0	2.0	0.5	0.5	1.0	2.0	1.0	1.0	0.5	1.0	0.5	1.0
GEL	1.0	1.0	1.0	1.0	2.0	-	1.0	1.0	1.0	2.0	1.0	1.0	1.0	1.0	2.0	1.0	0.5	1.0
LUT	1.0	1.0	1.0	1.0	1.0	2.0	-	1.0	1.0	0.5	0.0	0.5	1.0	0.0	1.0	1.0	1.0	1.0
VEN	1.0	1.0	1.0	1.0	0.5	2.0	1.0	-	1.0	1.0	1.0	0.5	1.0	1.0	1.0	1.0	1.0	1.0
TER	1.0	2.0	1.0	2.0	1.0	1.0	1.0	0.5	-	0.0	1.0	1.0	2.0	1.0	1.0	1.0	0.5	1.0
VOA	1.0	1.0	1.0	0.5	2.0	1.0	2.0	1.0	1.0	-	1.0	0.5	0.0	1.0	1.0	1.0	0.5	1.0
PSÍ	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	-	1.0	1.0	0.0	1.0	2.0	0.5	1.0
INS	1.0	0.5	1.0	1.0	0.5	1.0	0.5	2.0	1.0	2.0	1.0	-	1.0	1.0	1.0	1.0	1.0	1.0
PED	1.0	2.0	1.0	0.5	2.0	1.0	0.5	1.0	1.0	2.0	1.0	1.0	-	1.0	1.0	1.0	0.5	1.0
FAN	0.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	2.0	1.0	1.0	-	1.0	0.5	1.0	1.0
DRA	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	-	1.0	0.5	2.0
SOM	1.0	1.0	1.0	1.0	1.0	1.0	0.5	1.0	1.0	1.0	2.0	2.0	1.0	2.0	0.0	-	1.0	1.0
AÇO	2.0	0.5	0.5	0.5	1.0	2.0	2.0	0.0	2.0	0.5	0.5	0.5	0.5	1.0	0.0	1.0	-	1.0
FAD	1.0	0.5	1.0	1.0	1.0	1.0	2.0	0.5	1.0	1.0	1.0	1.0	1.0	1.0	2.0	2.0	0.5	-

Apresentar a conexidade do grafo e o reduzido

OBS: ler texto a seguir

```
Opção: 9
O grafo é fortemente conexo (C3).
Apresentando o grafo reduzido:

GRAFO [6] - grafo orientado com peso na aresta
Quantidade de vértices: 18
Quantidade de arestas: 306

NOR: [FOG] = 1.0 | [ÁGU] = 1.0 | [ELÉ] = 1.0 | [PLA] = 1.0 | [GEL] = 1.0 | [LUT] = 1.0 | [VEN] = 1.0 | [TER] = 1.0 | [VOA] = 1.0 | [PSÍ] = 1.0 | [INS] = 1.0 | [PED] = 0.5 | [FAN] = 0.0 | [DRA] = 1.0 | [SOM] = 1.0 | [AÇO] = 0.5 | [FAD] = 1.0 |
FOG: [NOR] = 1.0 | [ÁGU] = 0.5 | [ELÉ] = 1.0 | [PLA] = 2.0 | [GEL] = 2.0 | [LUT] = 1.0 | [VEN] = 1.0 | [TER] = 1.0 | [VOA] = 1.0 | [PSÍ] = 1.0 | [INS] = 2.0 | [PED] = 0.5 | [FAN] = 1.0 | [DRA] = 0.5 | [SOM] = 1.0 | [AÇO] = 2.0 | [FAD] = 1.0 |
ÁGU: [NOR] = 1.0 | [FOG] = 2.0 | [ELÉ] = 1.0 | [PLA] = 0.5 | [GEL] = 1.0 | [LUT] = 1.0 | [VEN] = 1.0 | [TER] = 2.0 | [VOA] = 1.0 | [PSÍ] = 1.0 | [INS] = 1.0 | [PED] = 2.0 | [FAN] = 1.0 | [DRA] = 0.5 | [SOM] = 1.0 | [AÇO] = 1.0 | [FAD] = 1.0 |
ELÉ: [NOR] = 1.0 | [FOG] = 1.0 | [ÁGU] = 2.0 | [PLA] = 0.5 | [GEL] = 1.0 | [LUT] = 1.0 | [VEN] = 1.0 | [TER] = 0.0 | [VOA] = 2.0 | [PSÍ] = 1.0 | [INS] = 1.0 | [PED] = 1.0 | [FAN] = 1.0 | [DRA] = 0.5 | [SOM] = 1.0 | [AÇO] = 1.0 | [FAD] = 1.0 |
PLA: [NOR] = 1.0 | [FOG] = 0.5 | [ÁGU] = 2.0 | [ELÉ] = 1.0 | [GEL] = 1.0 | [LUT] = 1.0 | [VEN] = 2.0 | [TER] = 0.5 | [VOA] = 0.5 | [PSÍ] = 1.0 | [INS] = 2.0 | [PED] = 1.0 | [FAN] = 1.0 | [DRA] = 0.5 | [SOM] = 1.0 | [AÇO] = 0.5 | [FAD] = 1.0 |
GEL: [NOR] = 1.0 | [FOG] = 1.0 | [ÁGU] = 1.0 | [ELÉ] = 1.0 | [PLA] = 2.0 | [LUT] = 1.0 | [VEN] = 1.0 | [TER] = 1.0 | [VOA] = 2.0 | [PSÍ] = 1.0 | [INS] = 1.0 | [PED] = 1.0 | [FAN] = 1.0 | [DRA] = 2.0 | [SOM] = 1.0 | [AÇO] = 0.5 | [FAD] = 1.0 |
LUT: [NOR] = 1.0 | [FOG] = 1.0 | [ÁGU] = 1.0 | [ELÉ] = 1.0 | [PLA] = 1.0 | [GEL] = 2.0 | [VEN] = 2.0 | [TER] = 1.0 | [VOA] = 0.5 | [PSÍ] = 0.0 | [INS] = 0.5 | [PED] = 1.0 | [FAN] = 0.0 | [DRA] = 1.0 | [SOM] = 1.0 | [AÇO] = 1.0 | [FAD] = 1.0 |
VEN: [NOR] = 1.0 | [FOG] = 1.0 | [ÁGU] = 1.0 | [ELÉ] = 1.0 | [PLA] = 0.5 | [GEL] = 2.0 | [LUT] = 1.0 | [TER] = 1.0 | [VOA] = 1.0 | [PSÍ] = 1.0 | [INS] = 0.5 | [PED] = 1.0 | [FAN] = 1.0 | [DRA] = 1.0 | [SOM] = 1.0 | [AÇO] = 1.0 | [FAD] = 1.0 |
TER: [NOR] = 1.0 | [FOG] = 2.0 | [ÁGU] = 1.0 | [ELÉ] = 2.0 | [PLA] = 1.0 | [GEL] = 1.0 | [LUT] = 1.0 | [VEN] = 0.5 | [VOA] = 0.0 | [PSÍ] = 1.0 | [INS] = 1.0 | [PED] = 2.0 | [FAN] = 1.0 | [DRA] = 1.0 | [SOM] = 1.0 | [AÇO] = 0.5 | [FAD] = 1.0 |
VOA: [NOR] = 1.0 | [FOG] = 1.0 | [ÁGU] = 1.0 | [ELÉ] = 0.5 | [PLA] = 2.0 | [GEL] = 1.0 | [LUT] = 2.0 | [VEN] = 1.0 | [TER] = 1.0 | [PSÍ] = 1.0 | [INS] = 0.5 | [PED] = 0.0 | [FAN] = 1.0 | [DRA] = 1.0 | [SOM] = 1.0 | [AÇO] = 0.5 | [FAD] = 1.0 |
PSÍ: [NOR] = 1.0 | [FOG] = 1.0 | [ÁGU] = 1.0 | [ELÉ] = 1.0 | [PLA] = 1.0 | [GEL] = 1.0 | [LUT] = 0.0 | [VEN] = 1.0 | [TER] = 1.0 | [VOA] = 1.0 | [INS] = 1.0 | [PED] = 1.0 | [FAN] = 0.0 | [DRA] = 1.0 | [SOM] = 2.0 | [AÇO] = 0.5 | [FAD] = 1.0 |
INS: [NOR] = 1.0 | [FOG] = 0.5 | [ÁGU] = 1.0 | [ELÉ] = 1.0 | [PLA] = 0.5 | [GEL] = 1.0 | [LUT] = 0.5 | [VEN] = 2.0 | [TER] = 1.0 | [VOA] = 2.0 | [PSÍ] = 1.0 | [PED] = 1.0 | [FAN] = 1.0 | [DRA] = 1.0 | [SOM] = 1.0 | [AÇO] = 1.0 | [FAD] = 1.0 |
PED: [NOR] = 1.0 | [FOG] = 2.0 | [ÁGU] = 1.0 | [ELÉ] = 0.5 | [PLA] = 2.0 | [GEL] = 1.0 | [LUT] = 0.5 | [VEN] = 1.0 | [TER] = 1.0 | [VOA] = 2.0 | [PSÍ] = 1.0 | [INS] = 1.0 | [FAN] = 1.0 | [DRA] = 1.0 | [SOM] = 1.0 | [AÇO] = 0.5 | [FAD] = 1.0 |
FAN: [NOR] = 0.0 | [FOG] = 1.0 | [ÁGU] = 1.0 | [ELÉ] = 1.0 | [PLA] = 1.0 | [GEL] = 1.0 | [LUT] = 0.0 | [VEN] = 1.0 | [TER] = 1.0 | [VOA] = 1.0 | [PSÍ] = 2.0 | [INS] = 1.0 | [PED] = 1.0 | [FAN] = 1.0 | [DRA] = 1.0 | [SOM] = 0.5 | [AÇO] = 1.0 | [FAD] = 1.0 |
DRA: [NOR] = 1.0 | [FOG] = 1.0 | [ÁGU] = 1.0 | [ELÉ] = 1.0 | [PLA] = 1.0 | [GEL] = 1.0 | [LUT] = 1.0 | [VEN] = 1.0 | [TER] = 1.0 | [VOA] = 1.0 | [PSÍ] = 1.0 | [INS] = 1.0 | [PED] = 1.0 | [FAN] = 1.0 | [SOM] = 1.0 | [AÇO] = 0.5 | [FAD] = 2.0 |
SOM: [NOR] = 1.0 | [FOG] = 1.0 | [ÁGU] = 1.0 | [ELÉ] = 1.0 | [PLA] = 1.0 | [GEL] = 1.0 | [LUT] = 0.5 | [VEN] = 1.0 | [TER] = 1.0 | [VOA] = 1.0 | [PSÍ] = 2.0 | [INS] = 2.0 | [PED] = 1.0 | [FAN] = 2.0 | [DRA] = 0.0 | [AÇO] = 1.0 | [FAD] = 1.0 |
AÇO: [NOR] = 2.0 | [FOG] = 0.5 | [ÁGU] = 0.5 | [ELÉ] = 0.5 | [PLA] = 1.0 | [GEL] = 2.0 | [LUT] = 2.0 | [VEN] = 0.0 | [TER] = 2.0 | [VOA] = 0.5 | [PSÍ] = 0.5 | [INS] = 0.5 | [PED] = 0.5 | [FAN] = 1.0 | [DRA] = 0.0 | [SOM] = 1.0 | [FAD] = 1.0 |
FAD: [NOR] = 1.0 | [FOG] = 0.5 | [ÁGU] = 1.0 | [ELÉ] = 1.0 | [PLA] = 1.0 | [GEL] = 1.0 | [LUT] = 2.0 | [VEN] = 0.5 | [TER] = 1.0 | [VOA] = 1.0 | [PSÍ] = 1.0 | [INS] = 1.0 | [PED] = 1.0 | [FAN] = 1.0 | [DRA] = 2.0 | [SOM] = 2.0 | [AÇO] = 0.5 |
```

Fim da impressao do grafo.



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



Para esta última funcionalidade, não foi possível implementar corretamente o algoritmo capaz de gerar o grafo reduzido, apesar de conseguir definir corretamente o tipo de grafo que está sendo trabalhado (neste caso, C3 – fortemente conexo).

Contudo, para gerarmos algum tipo de resultado nesta função, implementamos, a partir de requisições para o ChatGPT, um código funcional para que a aplicação não travasse, pois não conseguimos desenvolver uma solução a tempo da entrega