

GRAFOS

Percursos internacionais: Principais rotas
aéreas internacionais

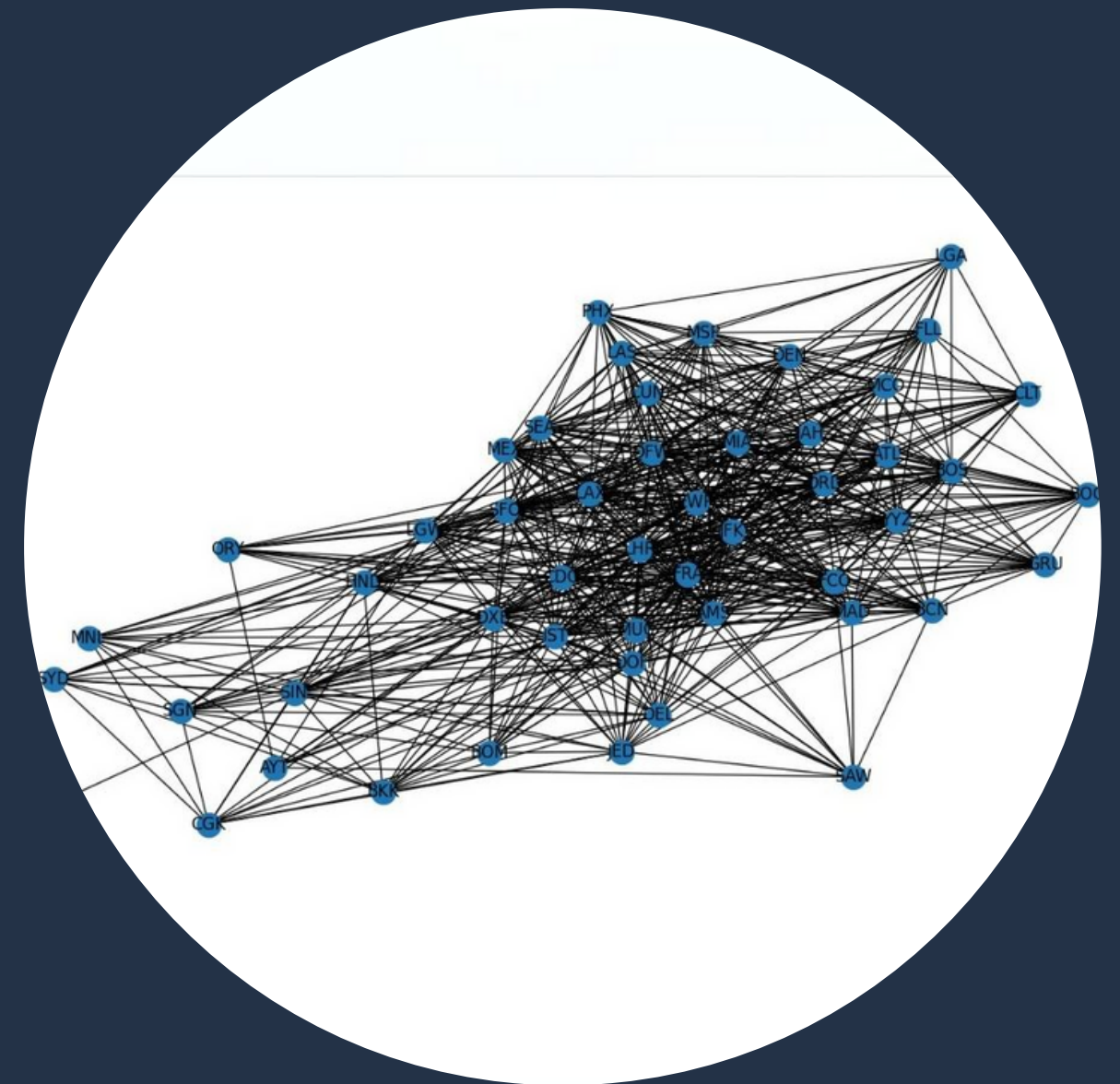
Leonardo Pinheiro de Souza - 32127391
Lucas Paulo da Rocha - 32196628
Luiz Octavio Tassinari Saraiva - 32030411
Thiago Aidar Figueiredo - 32144547



O PROJETO

Nosso projeto é uma aplicação que usa o sistema de grafos não direcionados e com pesos na arestas para representar as principais linhas aéreas entre os principais aeroportos internacionais do mundo. O objetivo do projeto está em promover uma forma de análise de quais rotas existem e analisar quais aeronaves podem ser usadas de forma mais efetiva entre as rotas.

01



GRAFO GERADO PELA BIBLIOTECA *NETWORKX*

GRAFO NÃO ORIENTADO PONDERADO

50

Vértices

Vértices são representados por siglas dos 50 aeroportos mais populosos do mundo.

656

Arestas

Indicam possíveis rotas aéreas entre os aeroportos

```
class GrafoND:
    TAM_MAX_DEFAULT = 100
    MODELOS_DEFAULT = ()
    def __init__(self, n=TAM_MAX_DEFAULT, mapa={}):
        self.n = n # número de vértices
        self.m = 0 # número de arestas
        self.mapa = mapa
        self.adj = [[float('inf') for i in range(n)] for j in range(n)]
```

Saida Nome Aeroporto	Chegada	Chegada Nome
Tan Son Nhat International Airport	DXB	Dubai International Airp
Tan Son Nhat International Airport	SFO	San Francisco Internationa
Tan Son Nhat International Airport	IST	Istanbul Airport
Tan Son Nhat International Airport	BKK	Bangkok Suvarnabhumi Airport
Tan Son Nhat International Airport	SIN	Singapore Changi Airport
Tan Son Nhat International Airport	DOH	Hamad International Airport
Tan Son Nhat International Airport	MNL	Ninoy Aquino International Airport
Tan Son Nhat International Airport	HND	Tokyo Haneda Airport
Tan Son Nhat International Airport	CGK	Soekarno–Hatta International Airp
Tan Son Nhat International Airport	DEL	Indira Gandhi International Airport
Tan Son Nhat International Airport	BOM	Chhatrapati Shivaji Maharaj Intern
Tan Son Nhat International Airport	FRA	Frankfurt Airport
Tan Son Nhat International Airport	CDG	Charles de Gaulle Airport
Tan Son Nhat International Airport	SYD	Sydney Airport
Tan Son Nhat International Airport	LHR	Heathrow Airport
Tan Son Nhat International Airport	LGW	London Gatwick Airport
Hartsfield–Jackson Atlanta International Airport	BOS	Logan International Airport
Hartsfield–Jackson Atlanta International Airport	CLT	Charlotte Douglas International Ai
Hartsfield–Jackson Atlanta International Airport	DEN	Denver International Airport
Hartsfield–Jackson Atlanta International Airport	DFW	Dallas Fort Worth International Air
Hartsfield–Jackson Atlanta International Airport	EWR	Newark Liberty International Airpo
Hartsfield–Jackson Atlanta International Airport	FLL	Fort Lauderdale–Hollywood Intern
Hartsfield–Jackson Atlanta International Airport	IAH	George Bush Intercontinental Airp
Hartsfield–Jackson Atlanta International Airport	JFK	John F. Kennedy International Airp
Hartsfield–Jackson Atlanta International Airport	LAS	Harry Reid International Airport
Hartsfield–Jackson Atlanta International Airport	LAX	Los Angeles International Airport
Hartsfield–Jackson Atlanta International Airport	LGA	LaGuardia Airport
Hartsfield–Jackson Atlanta International Airport	MCO	Orlando International Airport
Hartsfield–Jackson Atlanta International Airport	MIA	Miami International Airport
Hartsfield–Jackson Atlanta International Airport	MSP	Minneapolis/St Paul International
Hartsfield–Jackson Atlanta International Airport	ORD	O'Hare International Airport
Hartsfield–Jackson Atlanta International Airport	PHX	Phoenix Sky Harbor International
Hartsfield–Jackson Atlanta International Airport	SEA	Seattle–Tacoma International Airp

DATA SET CONSTRUÍDO

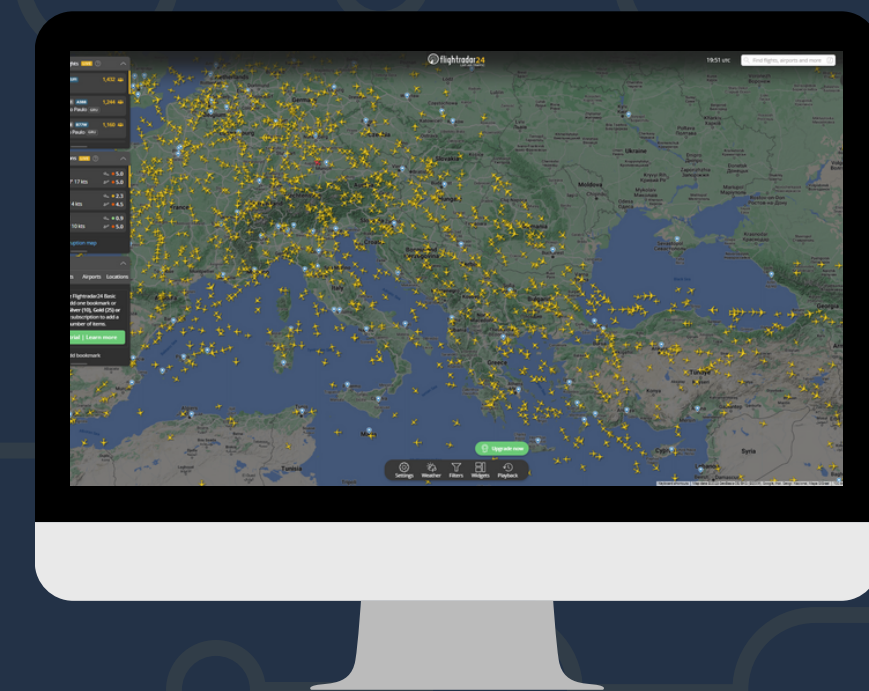
657

Linhas

7

Colunas

Informações retiradas do site
flightradar24.com



MENU

----- GRAFO PRINCIPAIS ROTAS AEREAS INTERNACIONAIS -----

- 1 - Ler dados do arquivo grafo.txt
 - 2 - Gravar dados no arquivo grafo.txt;
 - 3 - Inserir vértice;
 - 4 - Inserir aresta
 - 5 - Remove vértice
 - 6 - Remove aresta
 - 7 - Mostrar conteúdo do arquivo
 - 8 - Mostrar grafo
 - 9 - Conexidade do grafo
 - 10 - Encerrar a aplicação.
- />

MENU

```
def menu():  
    g1 = GrafoND(0, {})  
    resp = 0  
    tipo = 0  
    while (resp != 10):  
        print('\n\n')  
        print("\n----- GRAFO PRINCIPAIS ROTAS AEREAS INTERNACIONAIS -----\n")  
        print("1 - Ler dados do arquivo grafo.txt")  
        print("2 - Gravar dados no arquivo grafo.txt")  
        print("3 - Inserir vértice")  
        print("4 - Inserir aresta")  
        print("5 - Remove vértice")  
        print("6 - Remove aresta")  
        print("7 - Mostrar conteúdo do arquivo")  
        print("8 - Mostrar grafo")  
        print("9 - Conexidade do grafo")  
        print("10 - Encerrar a aplicação.")  
        resp = int(input("/> ").strip())  
  
    match resp:  
        case 1:  
            tipo, g1 = leituraTXT()  
        case 2:  
            g1.gravarTXT()  
        case 3:  
            sigla = input("Sigla do aeroporto: ").strip().upper()  
            nome = input("Nome do aeroporto: ").strip()  
            g1.insereV(sigla, nome)
```

```
case 4:  
    sigla_saida = input("Sigla de saida: ").strip().upper()  
    sigla_chegada = input("Sigla de chegada: ").strip().upper()  
    distancia = float(input("Distancia: "))  
    qtdModelos = int(input("Quantidade Modelos: ").strip())  
    modelos = []  
    for i in range(qtdModelos):  
        modelos.append((input("Modelos: ")).strip())  
    qtdAirlines = int(input("Quantidade de Airlines: ").strip())  
    Airlines = []  
    for i in range(qtdAirlines):  
        Airlines.append((input("Airline: ")).strip())  
    dict = {  
        "distancia": distancia,  
        "modelos": modelos,  
        "airlines": Airlines  
    }  
    g1.insereA(sigla_saida, sigla_chegada, dict)  
case 5:  
    nome = input("Sigla do aeroporto: ").strip().upper()  
    g1.removeV(nome)  
case 6:  
    saida = input("Sigla do aeroporto de saida: ").strip().upper()  
    chegada = input("Sigla do aeroporto de chegada: ").strip().upper()  
    g1.removeA(saida, chegada)  
case 7:  
    showTXT()  
case 8:  
    g1.showMin()  
case 9:  
    if g1.conexo():  
        print("Conexo")  
    else:  
        print("Desconexo")  
case 10:  
    continue
```

OP. 1 LEITURA DO ARQUIVO

"Programmers are very important in a company for system development and marketing."

```
----- GRAFO PRINCIPAIS ROTAS AEREAS INTERNACIONAIS -----  
1 - Ler dados do arquivo grafo.txt  
2 - Gravar dados no arquivo grafo.txt;  
3 - Inserir vértice;  
4 - Inserir aresta  
5 - Remove vértice  
6 - Remove aresta  
7 - Mostrar conteúdo do arquivo  
8 - Mostrar grafo  
9 - Conexidade do grafo  
10 - Encerrar a aplicação.  
/> 1  
Aresta inserida!  
Aresta inserida!  
Aresta inserida!  
Aresta inserida!  
Aresta inserida!  
Aresta inserida!  
Aresta inserida!  
Aresta inserida!  
Aresta inserida!
```

```
def leituraTXT():  
    mapa = {}  
    with open("grafo.txt", "r") as arquivo:  
        tipo = arquivo.readline().strip('\n')  
        n = int(arquivo.readline())  
        for i in range(n):  
            linha = arquivo.readline().strip('\n')  
            linha = linha.split(" ",1)  
            mapa[linha[0]] = [i,linha[1]]  
        Grafo = GrafoND(n,mapa)  
        m = int(arquivo.readline())  
        for w in range(m):  
            linha = arquivo.readline().strip('\n')  
            linha = linha.split(" ",2)  
            linha[2] = linha[2].replace("'", '')  
            linha[2] = linha[2].strip("'")  
            linha[2] = linha[2].split(';')  
            linha[2][1]=linha[2][1].replace("[", "")  
            linha[2][1]=linha[2][1].replace("]", "")  
            linha[2][2]=linha[2][2].replace("]", "")  
            linha[2][2]=linha[2][2].replace("[", "")  
            dict = {  
                "distancia": linha[2][0],  
                "modelos": linha[2][2].strip().split(", "),  
                "airlines": linha[2][1].strip().split(", ")  
            }  
            Grafo.insererA(linha[0],linha[1],dict)  
    return tipo, Grafo
```

OP. 2 GRAVAR DADOS TXT

"Programmers are very important in a company for system development and marketing."

Grafo salvo no arquivo.

----- GRAFO PRINCIPAIS ROTAS AEREAS INTERNACIONAIS -----

```
1 - Ler dados do arquivo grafo.txt
2 - Gravar dados no arquivo grafo.txt;
3 - Inserir vértice;
4 - Inserir aresta
5 - Remove vértice
6 - Remove aresta
7 - Mostrar conteúdo do arquivo
8 - Mostrar grafo
9 - Conexidade do grafo
10 - Encerrar a aplicação.
/> 2
```

Grafo salvo no arquivo.

----- GRAFO PRINCIPAIS ROTAS AEREAS INTERNACIONAIS -----

```
def gravarTXT(self):
    lista =(list(self.mapa))
    with open("grafo.txt", "w") as arquivo:
        arquivo.write("2\n")
        arquivo.write(f"{self.n:2d}\n")
        for key,value in self.mapa.items():
            arquivo.write(f"{key} {self.mapa[key][1]}\n")
        arquivo.write(f"{self.m:2d}\n")
        for i in range(self.n):
            for w in range(i+1,self.n):
                if isinstance(self.adj[i][w], dict):
                    arquivo.write(f"{lista[i]} {lista[w]} {self.adj[i][w]
['distancia']};{self.adj[i][w]['airlines']};{self.adj[i][w]
['modelos']}\n")
```


OP. 3 E 5

INSERIR/REMOVER VÉRTICE

```
def insereV(self, sigla, nome):
    if sigla not in self.mapa:
        self.mapa[sigla] = [len(self.mapa), nome]
        self.n = len(self.mapa)
        for i in range(self.n - 1):
            self.adj[i].append(float('inf'))
        self.adj.append([float('inf') for z in range(self.n)])
        print("Vértice inserido")

def removeV(self, nome):
    if nome in self.mapa:
        for i in range(self.n):
            self.adj[i].pop(self.mapa[nome][0])
        self.adj.pop(self.mapa[nome][0])
        for i in self.mapa.keys():
            if (self.mapa[nome][0] < self.mapa[i][0]):
                self.mapa[i][0] = self.mapa[i][0] - 1
        self.mapa.pop(nome, True)
        self.n = len(self.mapa)
        print("Vértice removido!")
```

```
Aresta inserida!
Aresta inserida!
Aresta inserida!
Aresta inserida!
Aresta inserida!
Aresta inserida!
```

----- GRAFO PRINCIPAIS ROTAS AEREAS INTERNACIONAIS -----

```
1 - Ler dados do arquivo grafo.txt
2 - Gravar dados no arquivo grafo.txt;
3 - Inserir vértice;
4 - Inserir aresta
5 - Remove vértice
6 - Remove aresta
7 - Mostrar conteúdo do arquivo
8 - Mostrar grafo
9 - Conexidade do grafo
10 - Encerrar a aplicação.
/> █
```

OP. 4 E 6

INSERIR/REMOVER ARESTA

```
def insereA(self, v, w, dict):
    if v in self.mapa and w in self.mapa:
        if self.adj[self.mapa[v][0]][self.mapa[w][0]] == float('inf'):
            self.adj[self.mapa[v][0]][self.mapa[w][0]] = dict
            self.adj[self.mapa[w][0]][self.mapa[v][0]] = dict
            self.m += 1
            print("Aresta inserida!")

def removeA(self, v, w):
    if v in self.mapa and w in self.mapa:
        if self.adj[self.mapa[v][0]][self.mapa[w][0]] != float('inf'):
            self.adj[self.mapa[v][0]][self.mapa[w][0]] = float('inf')
            self.adj[self.mapa[w][0]][self.mapa[v][0]] = float('inf')
            self.m -= 1
            print("Aresta removida!")
```

----- GRAFO PRINCIPAIS ROTAS AEREAS INTERNACIONAIS -----

```
1 - Ler dados do arquivo grafo.txt
2 - Gravar dados no arquivo grafo.txt;
3 - Inserir vértice;
4 - Inserir aresta
5 - Remove vértice
6 - Remove aresta
7 - Mostrar conteúdo do arquivo
8 - Mostrar grafo
9 - Conexidade do grafo
10 - Encerrar a aplicação.
/> 3
Sigla do aeroporto: Vertice
Nome do aeroporto: Vertice Teste
Vértice inserido
```

----- GRAFO PRINCIPAIS ROTAS AEREAS INTERNACIONAIS -----

```
1 - Ler dados do arquivo grafo.txt
2 - Gravar dados no arquivo grafo.txt;
3 - Inserir vértice;
4 - Inserir aresta
5 - Remove vértice
6 - Remove aresta
7 - Mostrar conteúdo do arquivo
8 - Mostrar grafo
9 - Conexidade do grafo
10 - Encerrar a aplicação.
/> █
```

OP. 7

MOSTRAR ARQUIVO

```
def showTXT():  
    with open("grafo.txt", "r") as arquivo:  
        Linhas = arquivo.readlines()  
        for linha in Linhas:  
            print("{}".format(linha.strip()))
```

----- GRAFO PRINCIPAIS ROTAS AEREAS INTERNACIONAIS -----

```
1 - Ler dados do arquivo grafo.txt  
2 - Gravar dados no arquivo grafo.txt;  
3 - Inserir vértice;  
4 - Inserir aresta  
5 - Remove vértice  
6 - Remove aresta  
7 - Mostrar conteúdo do arquivo  
8 - Mostrar grafo  
9 - Conexidade do grafo  
10 - Encerrar a aplicação.  
/> []
```

OP. 8 MOSTRAR GRAFO

```
def showMin(self):
    lista =(list(self.mapa))
    print(f"\n n: {self.n:2d} ", end="")
    print(f"m: {self.m:2d}\n")
    for i in range(self.n):
        for w in range(i+1,self.n):
            if isinstance(self.adj[i][w], dict):
                print(f"{lista[i]} {lista[w]} {self.adj[i][w]['distancia']}")
    print("\nfim da impressao do grafo.")
```

```
>_ Console x Shell x +
6 - Remove aresta
7 - Mostrar conteúdo do arquivo
8 - Mostrar grafo
9 - Conexidade do grafo
10 - Encerrar a aplicação.
/> 3
Sigla do aeroporto: Vertice
Nome do aeroporto: Vertice Teste
Vértice inserido

----- GRAFO PRINCIPAIS ROTAS AEREAS INTERNACIONAIS -----

1 - Ler dados do arquivo grafo.txt
2 - Gravar dados no arquivo grafo.txt;
3 - Inserir vértice;
4 - Inserir aresta
5 - Remove vértice
6 - Remove aresta
7 - Mostrar conteúdo do arquivo
8 - Mostrar grafo
9 - Conexidade do grafo
10 - Encerrar a aplicação.
/> 4
Sigla de saida: GRU
Sigla de chegada: SYD
Distancia: 89000
Quantidade Modelos: 1
Modelos: TRATOR
Quantidade de Airlines: 1
Airline: MACKENZIE
Aresta inserida!

----- GRAFO PRINCIPAIS ROTAS AEREAS INTERNACIONAIS -----

1 - Ler dados do arquivo grafo.txt
2 - Gravar dados no arquivo grafo.txt;
3 - Inserir vértice;
4 - Inserir aresta
5 - Remove vértice
6 - Remove aresta
7 - Mostrar conteúdo do arquivo
8 - Mostrar grafo
9 - Conexidade do grafo
10 - Encerrar a aplicação.
/> █
```

OP. 9 CONEXIDADE DO GRAFO

```
def conexo(self): #C0
    bool = True
    for i in range(1,self.n):
        teste = self.atingivelPercurso(0, i, [0] * self.n)
        if not teste:
            bool = False
    return bool
```

----- GRAFO PRINCIPAIS ROTAS AEREAS INTERNACIONAIS -----

```
1 - Ler dados do arquivo grafo.txt
2 - Gravar dados no arquivo grafo.txt;
3 - Inserir vértice;
4 - Inserir aresta
5 - Remove vértice
6 - Remove aresta
7 - Mostrar conteúdo do arquivo
8 - Mostrar grafo
9 - Conexidade do grafo
10 - Encerrar a aplicação.
/> 9
Conexo
```


OBRIGADO



[CLIQUE AQUI](#)