



Relatório do Projeto

Parte 2

Nome do Integrante	RA
Matteo Domiciano Varnier	10390247
André Akio Morita Osakawa	10340045
Rafael De Souza Oliveira Cerqueira Tinôco	10401436

Relatório

Projeto linhas aéreas

Autores

Matteo Domiciano Varnier

André Akio Morita Osakawa

Rafael de Souza Oliveira Cerqueira Tinoco

Descrição do Projeto:

Observamos que os aeroportos frequentemente não fornecem uma lista completa de todos os destinos disponíveis diretamente em seus sites. Isso significa que, muitas vezes, é necessário pesquisar individualmente em cada companhia aérea para descobrir se é possível voar de um aeroporto para outro.

Nosso projeto visa, com a ajuda de modelagem de rotas aéreas e aeroportos, mostrar as melhores rotas e conexões possíveis de um destino ao outro para ajudar usuários a planejar melhor suas viagens.

Os aeroportos e rotas escolhidas são do território nacional.

OBJETIVO:

Para facilitar a busca e planejamento de viagens, decidimos desenvolver uma ferramenta que permita identificar de forma mais eficiente quais aeroportos do país podem ser alcançados a partir de um determinado ponto de partida. Além disso, a ferramenta vai ajudar a determinar quais aeroportos precisam ser incluídos em uma conexão para atingir um destino desejado, caso o aeroporto inicial não ofereça voos diretos para o destino final.

Com essa abordagem, você poderá rapidamente encontrar rotas diretas e conexões necessárias simplificando o planejamento de suas viagens.

ODS

Nosso trabalho se encaixaria no ODS 9: Indústria, Inovação e infraestrutura. O projeto propõe uma solução inovadora para melhorar a eficiência de transporte aéreos, facilitando conexões entre aeroportos e melhorando o planejamento.



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



MODELAGEM DOS GRAFOS:

A modelagem das linhas aéreas do país como grafo tem como objetivo transformar os aeroportos e rotas em um modelo que facilite as buscas e análises de rotas.

Os aeroportos escolhidos foram baseados nos 20 mais visitados do país e o restante foram pegos, por ordem alfabética, aeroportos menores de cidades menores, para mostrar as melhores rotas que pessoas que moram nesses lugares mais isolados devem pegar para viajar, totalizando 50 aeroportos.

Os aeroportos escolhidos foram:

Guarulhos
Brasília
Fortaleza
Maceio
Manaus
Florianópolis
Goiania
Galeao
Confins
Belem
Porto Alegre
Santos Dumont
Cuiaba
Salvador
Recife
Vitoria
Curitiba
Natal
Vira-copos
Congonhas
João Pessoa
Uberlândia
Campo_Grande
Foz de Iguaçu
São Luiz
Aracaju
Água Boa
Almerim
Almerim M.Dourado
Alta Floresta
Altamira
Apucarana
Aracatuba
Araguaina
Barreiras
Bauru-Arealva
Boa Vista
Bonito
Breves
Cacador



Caldas Nova
Campina Grande
Campos dos Goytacazes
Carajas
Caruaru
Caruaru
Cascavel
Caixias do Sul
Chapeco
Coari Urucu
Cruz

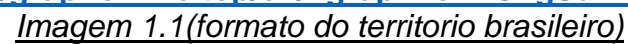
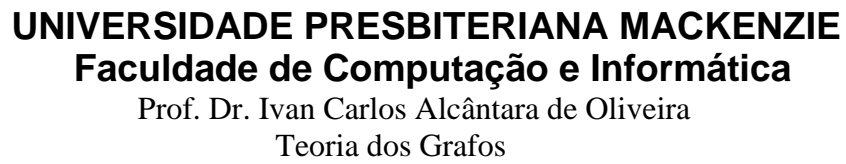
Para a modelagem foi construído um grafo não orientado com peso nas arestas utilizando o aplicativo graphonline.com, que foi mostrado durante aula para a turma, seguindo as seguintes representações:

- Vértice: Na modelagem do grafo, cada aeroporto é um vértice, identificado por seu nome.
- Arestas: No grafo, cada aresta é uma rota de um aeroporto para o outro, com peso nelas representando a distância entre cada aeroporto.

Grafo com escopo reduzido:

Para melhor visualização, abaixo está uma modelagem de escopo menor:







Tecnologias:

As tecnologias usadas durante o trabalho foram:

- Para o desenvolvimento do código do projeto foi utilizada a linguagem Python, por se tratar de uma linguagem mais simples e com muitas ferramentas uteis.
- Bloco de notas, foi utilizado como arquivo de entrada para o código funcionar, contendo os aeroportos e suas rotas.
- Graphonline, plataforma online para montagem dos grafos
- Sites para acompanhar as rotas dos aviões e site para calcular as distancias entre os aeroportos:
 - Site mais utilizado para o acompanhamento das rotas dos aviões foi feita pelo site <https://aeroin.net/rotas-domestico/> , onde pudemos ver todas as rotas dos aeroportos de todas as companhias nacionais. O site está com dados de 2021, por conta disso algumas informações podem estar desatualizadas, o que é uma possível limitação do projeto.
 - Para calcular a distância entre eles, foi utilizado o site <https://pt.distance.to/Guarulhos/Manaus> , onde pudemos verificar a distância quase real entre um aeroporto e outro.

Algoritmos Utilizados:


Para auxiliar na resolução do problema, foram implementados alguns algoritmos:

- Algoritmo de Dijkstra: Foi utilizado para calcular o menor caminho entre dois aeroportos, com base nas rotas e distancias, e assim encontrando a melhor rota possível para uma pessoa poder planejar sua viagem.
- Algoritmo de coloração: Implementado para visualizar segmentações dentro do grafo, pode ajudar no planejamento de rotas.
- Algoritmo de percursos abrangentes (Caminho Hamiltoniano e Euleriano): Implementado para ver se o grafo possui algum percurso euleriano e hamiltoniano.



Compilação e testes

Foi utilizado uma extensão do Visual Code feita pela própria microsoft para rodar aplicações em Python, porém foi necessário fazer a instalação do próprio Python. Python instalado: Python 3.12.6

Feito os passos acima abra o código chamado Teste_matriz.py, pode clicar no botão de play() para começar a rodar o código.

Ao iniciar o código, essa será a primeira aparição do código:

```
--- Menu de Opções ---  
1. Ler dados do arquivo de entrada  
2. Gravar dados no arquivo de saída  
3. Inserir vértice  
4. Inserir aresta  
5. Remover vértice  
6. Remover aresta  
7. Mostrar conteúdo do arquivo  
8. Mostrar grafo  
9. Verificar conexividade do grafo  
10. Coloracao  
11. Dijkstra  
12. Caminho Euleriano  
13. Caminho Hamiltoniano  
14. Encerrar a aplicação  
-----  
Escolha uma opção:
```

O usuário irá digitar 1 e apertar enter para que ele reconheça que será feita a opção 1.

A opção 1 deve ser obrigatoriamente a primeira, pois será carregado as informações do grafo.

- Ao apertar enter irá aparecer para o usuário digitar o nome do arquivo de entrada, será digitado: grafos.txt



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



```
--- Menu de Opções ---
1. Ler dados do arquivo de entrada
2. Gravar dados no arquivo de saída
3. Inserir vértice
4. Inserir aresta
5. Remover vértice
6. Remover aresta
7. Mostrar conteúdo do arquivo
8. Mostrar grafo
9. Verificar conexividade do grafo
10. Coloracao
11. Dijkstra
12. Caminho Euleriano
13. Caminho Hamiltoniano
14. Encerrar a aplicação
-----
Escolha uma opção: 1

Digite o nome do arquivo de entrada: grafos.txt
Agora carregue o arquivo de saída, que irá salvar o grafo
```

Após a primeira opção ser concluída o usuário irá digitar 2, para que as alterações feitas no código fiquem salvas em um outro arquivo.

A opção 2 deve ser obrigatória após a primeira etapa

- Após apertar entrar, irá aparecer para digitar o nome do arquivo de saída, pode ser digitado qualquer nome.

```
--- Menu de Opções ---
1. Ler dados do arquivo de entrada
2. Gravar dados no arquivo de saída
3. Inserir vértice
4. Inserir aresta
5. Remover vértice
6. Remover aresta
7. Mostrar conteúdo do arquivo
8. Mostrar grafo
9. Verificar conexividade do grafo
10. Coloracao
11. Dijkstra
12. Caminho Euleriano
13. Caminho Hamiltoniano
14. Encerrar a aplicação
-----
Escolha uma opção: 2

Carregue o arquivo de saída: output.txt
```

Após essas duas ações feitas, o usuário poderá ter **interações livres com o código**. Irei fazer os testes em ordem para melhor entendimento.



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



A opção 3 será para adicionar vértices(aeroportos). Ele irá pedir para que o usuário digite um nome de um novo aeroporto e irá salvar na lista de aeroportos já existente.

Aeroporto de Barbacena adicionado

```
--- Menu de Opções ---
1. Ler dados do arquivo de entrada
2. Gravar dados no arquivo de saída
3. Inserir vértice
4. Inserir aresta
5. Remover vértice
6. Remover aresta
7. Mostrar conteúdo do arquivo
8. Mostrar grafo
9. Verificar conexividade do grafo
10. Coloracao
11. Dijkstra
12. Caminho Euleriano
13. Caminho Hamiltoniano
14. Encerrar a aplicação
-----
Escolha uma opção: 3

Nome do aeroporto desejado:
Nome: Barbacena
```

Aeroporto de marilia adicionado

```
--- Menu de Opções ---
1. Ler dados do arquivo de entrada
2. Gravar dados no arquivo de saída
3. Inserir vértice
4. Inserir aresta
5. Remover vértice
6. Remover aresta
7. Mostrar conteúdo do arquivo
8. Mostrar grafo
9. Verificar conexividade do grafo
10. Coloracao
11. Dijkstra
12. Caminho Euleriano
13. Caminho Hamiltoniano
14. Encerrar a aplicação
-----
Escolha uma opção: 3

Nome do aeroporto desejado: Marília
Marília Adicionado
```




UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

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



A opção 4 irá criar a ligação entre dois vértices. Ao selecionar a opção ele irá pedir para digitar o primeiro vértice desejado e depois o segundo que deseja ir. O peso utilizado será a distância entre os dois aeroportos.

Aresta entre Barbacena e Maceió criado com sucesso

```
--- Menu de Opções ---
1. Ler dados do arquivo de entrada
2. Gravar dados no arquivo de saída
3. Inserir vértice
4. Inserir aresta
5. Remover vértice
6. Remover aresta
7. Mostrar conteúdo do arquivo
8. Mostrar grafo
9. Verificar conexividade do grafo
10. Coloracao
11. Dijkstra
12. Caminho Euleriano
13. Caminho Hamiltoniano
14. Encerrar a aplicação
-----
Escolha uma opção: 4

Nomes dos aeroportos das arestas e valor do peso:
Nome do aeroporto 1: Maceio
Nome do aeroporto 2: Barbacena
Peso da aresta: 1234
Aresta adicionada com sucesso
```

Aresta entre Marília e Guarulhos

```
--- Menu de Opções ---
1. Ler dados do arquivo de entrada
2. Gravar dados no arquivo de saída
3. Inserir vértice
4. Inserir aresta
5. Remover vértice
6. Remover aresta
7. Mostrar conteúdo do arquivo
8. Mostrar grafo
9. Verificar conexividade do grafo
10. Coloracao
11. Dijkstra
12. Caminho Euleriano
13. Caminho Hamiltoniano
14. Encerrar a aplicação
-----
Escolha uma opção: 4

Nomes dos aeroportos das arestas e valor do peso:
Nome do aeroporto 1: Marilia
Nome do aeroporto 2: Guarulhos
Peso da aresta: 400
Aresta adicionada com sucesso
```



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

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



A opção 5 permite com que o usuário possa remover um vértice(aeroporto). Ao selecionar a opção ele irá pedir qual número do vértice que deseja remover do grafo.

Remoção do vértice Barbacena

```
--- Menu de Opções ---
1. Ler dados do arquivo de entrada
2. Gravar dados no arquivo de saída
3. Inserir vértice
4. Inserir aresta
5. Remover vértice
6. Remover aresta
7. Mostrar conteúdo do arquivo
8. Mostrar grafo
9. Verificar conexividade do grafo
10. Coloracao
11. Dijkstra
12. Caminho Euleriano
13. Caminho Hamiltoniano
14. Encerrar a aplicação
-----
Escolha uma opção: 5

Nome do aeroporto: Barbacena
Vértice Barbacena removido com sucesso.
Vértice removido com sucesso
```

Remoção do vertice Marilia

```
--- Menu de Opções ---
1. Ler dados do arquivo de entrada
2. Gravar dados no arquivo de saída
3. Inserir vértice
4. Inserir aresta
5. Remover vértice
6. Remover aresta
7. Mostrar conteúdo do arquivo
8. Mostrar grafo
9. Verificar conexividade do grafo
10. Coloracao
11. Dijkstra
12. Caminho Euleriano
13. Caminho Hamiltoniano
14. Encerrar a aplicação
-----
Escolha uma opção: 5

Nome do aeroporto: Marilia
Vértice Marilia removido com sucesso.
Vértice removido com sucesso
```



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

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



A opção 6 ele permite que o usuário remova a aresta (conexão) entre dois vértices, com isso ocorre a remoção da conexão entre eles e o peso (distância) entre eles também.

Remoção da aresta Barbacena-Maceio

```
--- Menu de Opções ---
1. Ler dados do arquivo de entrada
2. Gravar dados no arquivo de saída
3. Inserir vértice
4. Inserir aresta
5. Remover vértice
6. Remover aresta
7. Mostrar conteúdo do arquivo
8. Mostrar grafo
9. Verificar conexividade do grafo
10. Coloracao
11. Dijkstra
12. Caminho Euleriano
13. Caminho Hamiltoniano
14. Encerrar a aplicação
-----
Escolha uma opção: 6

Nome do aeroporto 1: Barbacena
Nome do aeroporto 2: Maceio
Aresta removida com sucesso
```

Remoção da aresta Marília-Guarulhos

```
--- Menu de Opções ---
1. Ler dados do arquivo de entrada
2. Gravar dados no arquivo de saída
3. Inserir vértice
4. Inserir aresta
5. Remover vértice
6. Remover aresta
7. Mostrar conteúdo do arquivo
8. Mostrar grafo
9. Verificar conexividade do grafo
10. Coloracao
11. Dijkstra
12. Caminho Euleriano
13. Caminho Hamiltoniano
14. Encerrar a aplicação
-----
Escolha uma opção: 6

Nome do aeroporto 1: Marília
Nome do aeroporto 2: Guarulhos
Aresta removida com sucesso
```

A opção 7 mostra o conteúdo inteiro que está no txt, mostrando o tipo de grafo, quantidade de vértices, nome dos vértices, as arestas e os pesos dos vértices. Como o arquivo é grande, iremos mostrar alguns pequenos exemplos.

2	
52	
Guarulhos	0
Brasilia	1
Fortaleza	2
Maceio	3
Manaus	4
Florianopolis	5
Goiania	6
Galeao	7
Confins	8
Belem	9
Porto Alegre	10
Santos Dumont	11

0 1 864
0 2 2357
0 3 1915
0 4 2700
0 5 502
0 6 805
0 7 345
0 8 527
0 9 2475
0 10 840
0 11 350
0 12 1330
0 13 1467
0 14 2143
0 15 757
0 16 326
0 17 2307

Vértices e numeração deles

Arestas e pesos

A opção 8 mostra essas informações acima em formato de matriz, para melhor entendimento do grafo. Como a saída ficou grande, iremos mostrar pequenos exemplos

[illegible][illegible]



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

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



A opção 9 ele irá mostrar se o grafo é conexo ou não, caso não seja irá receber a mensagem não conexo, caso contrário irá mostrar que o grafo é conexo. No nosso caso, por ser um grafo não direcionado ele apenas informa se é ou não conexo.

```
--- Menu de Opções ---
1. Ler dados do arquivo de entrada
2. Gravar dados no arquivo de saída
3. Inserir vértice
4. Inserir aresta
5. Remover vértice
6. Remover aresta
7. Mostrar conteúdo do arquivo
8. Mostrar grafo
9. Verificar conexividade do grafo
10. Coloracao
11. Dijkstra
12. Caminho Euleriano
13. Caminho Hamiltoniano
14. Encerrar a aplicação
-----
Escolha uma opção: 9

Grafo conexo
```

A opção 10 irá mostrar a coloração do grafo, mostrando aeroporto por aeroporto qual seria sua cor. Como ficou grande, irei mostrar uma pequena parte da saída.

```
Vértice Bauru-Arealva -> Cor 2
Vértice Boa Vista -> Cor 1
Vértice Bonito -> Cor 2
Vértice Breves -> Cor 1
Vértice Cacador -> Cor 2
Vértice Caldas Nova -> Cor 2
Vértice Campina Grande -> Cor 2
Vértice Campos dos Goytacazes -> Cor 1
Vértice Carajas -> Cor 1
Vértice Caruaru -> Cor 1
Vértice Caruaru -> Cor 1
Vértice Cascavel -> Cor 2
Vértice Caixias do Sul -> Cor 2
Vértice Chapeco -> Cor 2
Vértice Coari Urucu -> Cor 1
Vértice Cruz -> Cor 2
Vértice Barbacena -> Cor 1

Número total de cores usadas: 7
```



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

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



Opção 11 irá fazer descobrir o caminho mínimo de um aeroporto até outro usando o algoritmo de dijkstra. Ao selecionar essa opção ele irá perguntar qual aeroporto o usuário gostaria de sair e qual seria o destino. Ao fazer as escolhas, ele irá mostrar quantos km o usuário iria viajar e se será necessário fazer conexão para chegar ao destino.

Guarulhos até Araçatuba

```
Digite o nome do aeroporto de origem: Guarulhos
Digite o nome do aeroporto de destino: Aracatuba
Distância mínima de Guarulhos para Aracatuba: 473 km
Caminho: ['Guarulhos', 'Aracatuba']
```

Porto Alegre – Goiania

```
Caruaru
Cascavel
Caixias do Sul
Chapeco
Coari Urucu
Cruz
Marilia

Digite o nome do aeroporto de origem: Porto Alegre
Digite o nome do aeroporto de destino: Goiania
Distância mínima de Porto Alegre para Goiania: 1544 km
Caminho: ['Porto Alegre', 'Florianopolis', 'Vira-copos', 'Goiania']
```

A opção 12 irá mostrar se o grafo tem caminho euleriano. Caso não tenha, irá retornar a mensagem que não é euleriano. Caso tenha, irá dizer que é euleriano.

```
Escolha uma opção: 12

O grafo não tem um caminho euleriano.
```

A opção 13 irá mostrar se o grafo tem caminho hamiltoniano. Caso não tenha, irá retornar a mensagem que não é hamiltoniano. Caso tenha, irá dizer que é hamiltoniano

```
Escolha uma opção: 13

O grafo tem um caminho hamiltoniano.
```

Menu de Opções



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

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



A opção 14 sai do código e com isso finalizando as ações

```
-----  
Escolha uma opção: 14
```

```
Saindo...
```

Apêndice

LINK DO GITHUB

<https://github.com/matteovar/Projeto-Grafos>

Link do vídeo do youtube

<https://youtu.be/18X-KvVioFc>