**UNIVERSIDADE DE UBERABA**

**LABORATÓRIO DE PROGRAMAÇÃO COMPETITIVA**

**PROJETO PRÁTICO N3**

**ALUNOS:** VITOR DE AZAMBUJA RIBEIRO FRANCO

PEDRO HENRIQUE RESENDE

**MALHA DE TRANSPORTE AÉREO: OTIMIZAÇÃO DE ROTAS AÉREAS**

**RESUMO:**

O projeto utiliza Python e bibliotecas como Networkx, Geopy, Matplotlib, Pandas,Basemap, Sys e PyQt para implementar uma interface gráfica que permite a visualização de um mapa da malha aérea do Brasil. O mapa consiste em uma representação na forma de um grafo ponderado fazendo o uso de aeroportos do Brasil como nós, rotas aéreas simuladas como arestas e a distância de cada rota como pesos, para simular um tráfego real. Na aplicação, o usuário é recebido com uma visualização da malha aérea brasileira e caixas de seleção para escolher o Estado de origem e o aeroporto de origem dentro do Estado de origem escolhido. Em seguida, o usuário deve escolher o Estado de destino e o aeroporto de destino dentro do Estado de destino escolhido.

A screenshot of a computer screen

Description automatically generated

Figura - Interface inicial da aplicação

Após escolher as opções de origem e destino, o usuário deve apertar o botão de “Calcular Rota”. Quando apertado, o botão acionará o cálculo da menor rota possível e da distância total entre os dois aeroportos escolhidos usando o algoritmo de Dijkstra. Logo, o mapa atualizará e o menor caminho possível será mostrado em vermelho e o aeroporto de origem, de destino e a distância total em quilômetros serão mostrados acima do mapa.

A screenshot of a computer screen

Description automatically generated

Figura - Interface após calcular a distância entre dois aeroportos

**IMPLEMENTAÇÃO EM PYTHON:**

O código pode ser dividido em importação de dependências, funções auxiliares, classe que implementa a interface com o cálculo de rotas e função principal.

A screen shot of a computer

Description automatically generated

Figura 3 - Importação das bibliotecas

Após importar as bibliotecas sys, pandas, networkx, geopy, matplotlib, basemap e PyQt5, temos as funções auxiliares que são criar\_grafo e plotar\_gráfico. A função criar\_grafo utiliza das bibliotecas Networkx e Geopy e recebe os aeroportos principais agrupados por Estado, cria o grafo, adiciona cada aeroporto e sua sigla como um nó no grafo tendo a sua posição equivalente às coordenadas geográficas reais. Então transforma esses nós em uma lista que é salvada na variável lista\_nos. Para adicionar as arestas, é usado primeiramente um “for” do índice e da origem em um enumerate, obtido a partir da lista criada anteriormente. Logo, é criado um loop dentro desse “for” para adicionar as arestas de fato e garantir que um nó se conecte somente com outros 3 nós, para simular uma malha aérea. Então, dentro desse loop, é estabelecido um índice cíclico de destino, é obtido o nó de destino da lista usando o índice de destino, depois são obtidas as coordenadas geográficas da origem e do destino e a distância entre esses dois pontos é calculada usando o Geopy e convertida em quilômetros. Finalmente, são adicionadas as arestas entre a origem e o destino tendo a distância arredondada em duas casas decimais como peso e o grafo completo é retornado.

A screenshot of a computer

Description automatically generated

Figura 4 - Função criar\_grafo

Abaixo da função criar\_grafo têm-se a função plotar\_grafico, que recebe os aeroportos principais agrupados por Estado e o grafo. A função faz o uso das bibliotecas Matplotlib e Basemap para configurar a figura e o eixo do gráfico, criar o mapa, receber especificações de como o mapa deve ser plotado, iterar sobre o DataFrame que guarda os aeroportos principais por Estado, obter a posição dos aeroportos usando sua sigla e suas respectivas coordenadas geográficas e salvá-las em um dicionário para serem utilizadas usando o Basemap. Em seguida, o grafo é plotado no mapa e a função retorna a figura e a posição dos nós.

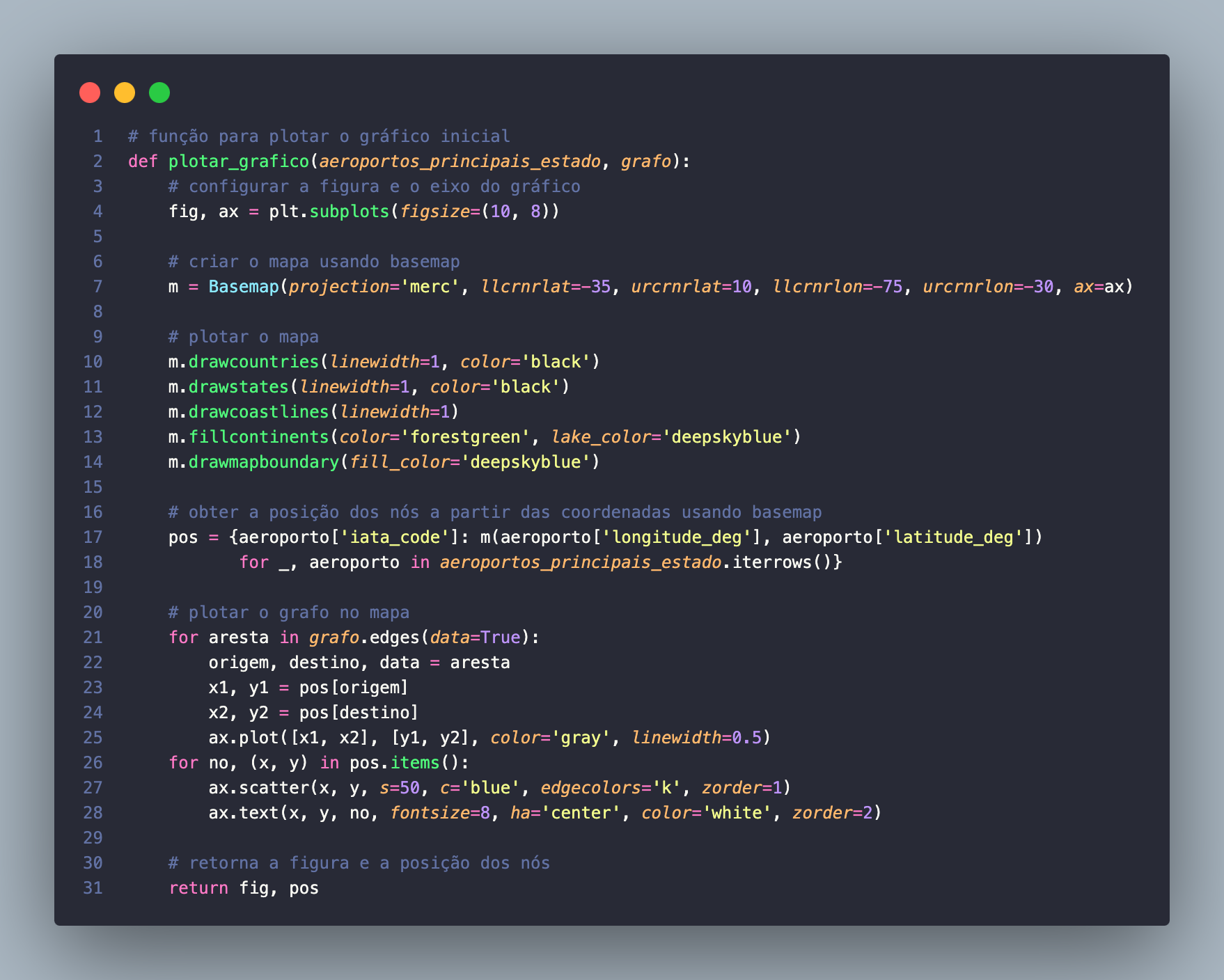


Figura 5 - Função plotar\_grafico

Para a classe principal chamada de MainWindow, temos na função de inicialização a definição das variáveis dentro da classe, a configuração da janela, a exibição do gráfico inicial, a criação do layout e de seus itens como ComboBoxes, botões e Labels, a conexão das Comboboxes e botões com seus respectivos dados e funções a serem chamadas.

A screen shot of a computer program

Description automatically generated

Figura 6 - Função de inicialização da classe MainWindow

Abaixo da função de inicialização, têm-se as funções atualizar\_aeroportos\_origem e atualizar\_aeroportos\_destino, que são usadas na função de inicialização e servem para obter o texto dentro da ComboBox, filtrar e validar os aeroportos dentro desse Estado, adicionar os aeroportos válidos resultantes à segunda ComboBox e limpar os itens da ComboBox quando o Estado selecionado for trocado.

A screenshot of a computer program

Description automatically generated

Figura - Funções atualizar\_aeroportos de origem e destino

Por fim, temos a última função da classe MainWindow, chamada de calcular\_rota. A função calcular\_rota utiliza as bibliotecas Matplotib e NetworkX, é chamada ao apertar o botão e calcula a menor rota entre dois nós no grafo usando o algoritmo de Dijkstra, obtém a distância entre esses dois nós e arredonda para duas casas decimais, plota um segundo gráfico com a menor rota destacada e atualiza o gráfico na interface. Primeiro, a função define os aeroportos de origem com base no texto da ComboBox, então é feito o cálculo do menor\_caminho e da distância desse caminho. Em seguida, o segundo gráfico é plotado com as arestas que compõem o menor caminho destacadas em vermelho e o restante do gráfico é plotado igual ao primeiro. Por fim, é definido o título do gráfico para exibir o aeroporto de origem e o de destino e a distância do menor caminho entre eles e então a interface é atualizada para remover o primeiro gráfico e inserir o segundo, porém o efeito que se obtém é somente de grifar a menor rota em vermelho e mostrar o título do gráfico, fazendo a troca de gráficos ficar quase imperceptível.

*A computer screen shot of a program code

Description automatically generated*

Figura - Função calcular\_rota

Finalizando a classe que implementa a interface, temos a função principal, a função main que utiliza as bibliotecas Pandas, PyQt e Sys. Dentro dessa função, temos a leitura do arquivo .csv que contém os dados (sigla, coordenadas, classificação) de todos aeroportos do Brasil e pode ser obtido através do site <https://ourairports.com/countries/BR/> por meio de download. Também é feita a filtragem dos aeroportos grandes e médios e a formatação e agrupamento da sigla dos Estados do Brasil que cada aeroporto pertence. O grafo então é criado a partir da função criar\_grafo já abordada anteriormente e a aplicação é inicializada. Como última linha do código, temos a chamada da função main.

A computer screen shot of a program

Description automatically generated

Figura - Função main e sua chamada

**GUIA DE USO:**

Primeiramente, deve-se instalar as dependências usadas:

A black rectangle with white text

Description automatically generated

Figura - Comando para instalar as dependências necessárias

Então, salve o código Python em um arquivo chamado malha\_aerea.py em um diretório. Logo após, deve-se obter o arquivo .csv do site <https://ourairports.com/countries/BR/> e inserí-lo no mesmo diretório em que está o arquivo malha\_aerea.py. Finalmente, o código pode ser executado por qualquer IDE ou pelo terminal através do comando:

A black rectangular with yellow green and red circles and white text

Description automatically generated

Figura - Comando para executar o código

**CONCLUSÃO:**

Este projeto combina o uso de leitura de DataFrames, interface gráfica, estrutura de dados, modelagem e visualização de grafos em um mapa com projeções baseadas em coordenadas reais para demonstrar um possível uso em análise de malhas aéreas para achar a menor rota entre dois aeroportos. Logo, também pode ser utilizado para estudos e ser adaptado para uma eventual escalabilidade de dados. O projeto pode ser encontrado para download no repositório <https://github.com/vitordeazambuja/malha-aerea>.