

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS**



**PUC**  
**CAMPINAS**  
**PONTIFÍCIA UNIVERSIDADE CATÓLICA**

<b>Enzo Cinto Quatrochi Ra</b>	<b>23015904</b>
<b>Gustavo Toledo Bicaletto</b>	<b>21011066</b>
<b>João Pedro Giaretta</b>	<b>23008717</b>
<b>Thiago Luiz Fossa</b>	<b>23010116</b>

**Algoritmos de Programação - Prática**

**CAMPINAS, 2023**

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS**



**PUC**  
**CAMPINAS**  
**PONTIFÍCIA UNIVERSIDADE CATÓLICA**

<b>Enzo Cinto Quatrochi</b>	<b>23015904</b>
<b>Gustavo Toledo Bicaletto</b>	<b>21011066</b>
<b>João Pedro Giaretta</b>	<b>23008717</b>
<b>Thiago Luiz Fossa</b>	<b>23010116</b>

**Programa Startup – Venda de Passagens Aéreas**

**Trabalho apresentado à professora Lucia Filomena como objetivo de obtenção de nota  
na matéria APPC.**

**CAMPINAS, 2023**

# Introdução

Nosso time foi designado para construir um aplicativo para uma companhia aérea, que consta com específicas funções como, incluir voos, alterar as informações sobre um determinado voo, apagar um voo, dada a cidade origem determinar quantos voos saem dessa cidade, dada a cidade origem e destino, determinar o voo com menor número de escalar e imprimir todas as informações sobre esse voo e dada a cidade destino, determinar todos os voos com a respectiva cidade origem que chegam nesse destino.

O código foi desenvolvido em duas etapas, a etapa 1 foi focada no desenvolvimento das funções responsáveis por cada funcionalidade do Sistema, como demonstrado abaixo

Inclur voos:

[illegible]

Alterar voos:

[illegible]

Apagar voos:

```
def apagar(Voos):  
    while True:  
        try:  
            codbusca = int(input('Digite o código do voo que deseja apagar: '))  
            break  
        except ValueError:  
            print('Tem que ser número inteiro...')  
  
    for keys, values in voos.items():  
        if codbusca == keys:  
            del Voos[codbusca]  
            print("Voo Apagado com Sucesso!!!")  
  
    return Voos
```

Quantos voos saem de uma cidade para outra:

```
def cidade_origem(Voos):  
    while True:  
        try:  
            Perg3 = input('Digite a cidade ORIGEM para saber quantos voos saem dela: ').upper()  
            break  
        except ValueError:  
            print('Tem que digitar A CIDADE...')  
  
    i = 0  
  
    for keys, values in voos.items():  
        if Perg3 == values[0]:  
            i += 1  
        else:  
            print('Não há voos que saem desta cidade...')  
  
    print(f'Quantidade de voos que saem desta cidade origem: {i}')  
    return i
```

Qual o voo com menor número de escalas para o destino:

```
def Voo_menor_escalas(Voos):  
    while True:  
        try:  
            Perg4 = input('Digite a cidade ORIGEM para determinar o voo com menor número de escalas: ').upper()  
            Perg5 = input('Digite a cidade DESTINO para determinar o voo com menor número de escalas: ').upper()  
            break  
        except ValueError:  
            print('Tem que digitar A CIDADE...')  
  
    menorNescalas = 0  
  
    for keys, values in voos.items():  
        if Perg4 == values[0] and Perg5 == values[1]:  
            if values[2] < menorNescalas:  
                menorNescalas = values[2]  
                voomenorescala = keys, values  
            else:  
                print("Não há voos entre essas cidades")  
  
    print(voomenorescala)  
    return voomenorescala
```

Voos que saem da cidade de origem para a de destino:

```
def cidade_origem2(Voos):  
    while True:  
        try:  
            destino = input('Digite a cidade DESTINO para determinar quais cidades Origem possuem esse destino: ').upper()  
            break  
        except ValueError:  
            print("Digite o nome da cidade")  
  
    city_origem = []  
  
    for values in voos.values():  
        if destino == values[1]:  
            city_origem.append(values[0])  
  
    print(city_origem)  
    return city_origem
```

Já a etapa 2, foi realizada a criação do menu interativo e da chamada das funções, dependendo da opção selecionada pelo usuário.

```
voos = {}
while True:
    print("Menu\n")
    print("1. Inserir Voo")
    print("2. Alterar Informações")
    print("3. Apagar Voo")
    print("4. Determinar Voos que saiem da cidade")
    print("5. Determinar Voo com menor numero de Escalas")
    print("6. Determinar as cidades de origem que possuem o mesmo destino")
    print("7. Sair\n")

    while True:
        try:
            num = int(input("Digite o numero da opção que deseja acessar: "))
            break
        except ValueError:
            print("Digite o número da opção")

    if num == 1:
        while True:
            inserir(voos)
            break

    if num == 2:
        while True:
            alterar(voos)
            break

    if num == 3:
        while True:
            apagar(voos)
            break

    if num == 4:
        while True:
            cidade_origem(voos)
            break
```

## **Apresentação do projeto – Erros e Soluções**

O time encontrou dificuldades em se comunicar entre os membros, e organizar as devidas funções de cada membro. Depois de organizado, o time trabalhou com sinergia e concluiu todos os objetivos do projeto.

### **Apresentação do projeto – Como utilizar o Programa**

1. Ao Iniciar o programa, será apresentado o Menu principal como todas as funções disponíveis no sistema.
2. Será pedido para que o usuário digite o número correspondente a função que ele deseja utilizar.
3. Caso digite 1, o sistema irá para a tela de inserção de Voos, nela será pedido para que o usuário insira as informações do voo, como, código, cidade de origem, cidade de destino e número de escalas. Caso o número de escalas seja superior a 0, o sistema pedirá também o nome dessas cidades.
4. Caso digite 2, o sistema irá para a tela de alteração de informações, ao entrar, será disponibilizado as informações sobre os voos que já estão cadastrados, e pedirá para o usuário o código do voo que deseja alterar, após isso, ele pedirá as novas informações sobre o respectivo voo.
5. Caso digite 3, o sistema irá para a tela de apagar informações, ao entrar, será disponibilizado as informações sobre os voos que já estão cadastrados, e pedirá para o usuário o código do voo que deseja excluir.
6. Caso digite 4, o sistema irá para a tela que determina a quantidade de voos que saem de determinada cidade, nela será pedido apenas o nome da cidade de origem.
7. Caso digite 5, o sistema irá para a tela que determina o voo com menor quantidade de escalas entre duas cidades, nela será pedido o nome da cidade de origem e de destino.
8. Caso digite 6, o sistema irá para a tela que determina quais cidades possuem o mesmo destino, nessa tela será necessário inserir apenas o nome da cidade de destino.
9. Caso digite 7, o sistema será encerrado.

## **Referências Bibliográficas**

CURSO Python #20 - Funções (Parte 1). [S. l.: s. n.], 2019. Disponível em: [https://youtu.be/ezfr9d7wd\\_k](https://youtu.be/ezfr9d7wd_k). Acesso em: 31 maio 2023