

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

ARTHUR GRIZONE SILVESTRE DE OLIVEIRA - 25008341

LUCAS RODRIGUES XAVIER - 25000508

MATEUS OLIVEIRA RAFAEL - 25001369

WANDELEY CELESTINO VIEIRA FILHO - 25007121

ALGORITMO DE PROGRAMAÇÃO - PRÁTICA

CAMPINAS

2025

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

**ESCOLA POLITÉCNICA
ENGENHARIA DE SOFTWARE**

**ARTHUR GRIZONE SILVESTRE DE OLIVEIRA - 25008341
LUCAS RODRIGUES XAVIER - 25000508
MATEUS OLIVEIRA RAFAEL - 25001369
WANDELEY CELESTINO VIEIRA FILHO - 25007121**

ALGORITMO DE PROGRAMAÇÃO - PRÁTICA

**CAMPINAS
2025**

SUMÁRIO

1.	INTRODUÇÃO	4
2.	APRESENTAÇÃO DO PROJETO	5
3.	ESTRUTURA DO SISTEMA	5
4.	DIFICULDADES E SOLUÇÕES	6
5.	COMPILADOR UTILIZADO	8
6.	REFERÊNCIA BIBLIOGRÁFICA	8

1. INTRODUÇÃO

O problema a ser resolvido envolve o desenvolvimento de um sistema de gerenciamento de voos e passageiros, com o objetivo de facilitar a organização e a consulta de informações relacionadas às viagens aéreas. Este sistema deve ser capaz de cadastrar voos, registrar passageiros e realizar consultas baseadas em diferentes critérios, como código do voo, cidade de origem ou cidade de destino.

A principal necessidade deste sistema é garantir uma maneira eficiente de gerenciar os dados dos voos e dos passageiros, assegurando que as informações possam ser acessadas de forma rápida e organizada. Cada voo deve conter informações essenciais, como código, origem, destino, número de escalas, preço e lugares disponíveis. Os passageiros devem ser identificados de maneira única, utilizando o CPF, e ser capazes de comprar mais de uma passagem ao longo do tempo.

O sistema deve permitir que o usuário consulte os voos de diversas formas, como por código, origem ou destino, e também possibilitar a listagem de passageiros de um voo específico. Além disso, deve permitir a adição de novos voos, o cadastro de passageiros, a venda e o cancelamento de passagens.

A principal dificuldade desse sistema reside na organização eficiente dos dados e na criação de uma estrutura flexível que permita consultas e alterações rápidas, sem comprometer a integridade das informações.

2. APRESENTAÇÃO DO PROJETO

O sistema é dividido em dois componentes principais: voos e passageiros. Os voos são armazenados em um dicionário, onde a chave é o código do voo e o valor é outro dicionário com informações como cidade de origem, cidade de destino, número de escalas, preço e lugares disponíveis. Cada voo também contém uma lista de passageiros, que são registrados com seu CPF, nome e telefone. Essa abordagem permite que os dados sejam acessados de maneira rápida e organizada.

3. ESTRUTURA DO SISTEMA

O sistema utiliza dois dicionários principais para organizar as informações:

voos: chave é o código do voo, e o valor é um dicionário com dados como origem, destino, escalas, preço e lugares disponíveis.

passageiros: chave é o CPF do passageiro, e o valor é uma lista dos códigos dos voos que ele comprou.

Além disso, há um dicionário auxiliar `dados_passageiros` que armazena as informações pessoais dos passageiros (nome e telefone) associadas a cada CPF.

4. DIFICULDADES E SOLUÇÕES

- Controle de disponibilidade de lugares nos voos:

Garantir que a quantidade de lugares disponíveis fosse corretamente atualizada nas operações de venda e cancelamento de passagens exigiu atenção para evitar inconsistências e garantir que voos não fossem vendidos além da capacidade.

- Relacionamento entre passageiros e voos:

A necessidade de manter um registro que relacionasse passageiros com múltiplos voos exigiu uma organização clara entre os dicionários, para que fosse possível consultar rapidamente tanto os passageiros de um voo quanto os voos comprados por um passageiro.

- Tratamento de entradas inconsistentes de texto:

Para evitar erros causados por diferenças entre maiúsculas e minúsculas nas entradas, como códigos de voo e nomes de cidades, utilizamos o método `.lower()` para padronizar todas as entradas. Isso garantiu maior consistência e facilitou as buscas e comparações de dados no sistema.

Outra dificuldade enfrentada ocorreu na estruturação inicial do sistema. A primeira tentativa foi usar a mesma chave (código do voo) tanto no dicionário de voos quanto no dicionário de passageiros, assumindo que isso facilitaria o relacionamento entre os dados. No entanto, essa abordagem não funcionava corretamente, pois os passageiros deveriam ser identificados pelo CPF, e não por voo. Essa estrutura não permitia gerenciar corretamente situações como um passageiro comprando múltiplas passagens ou listagens por CPF.

Para resolver o problema, reorganizamos a estrutura utilizando o CPF como chave no dicionário passageiros, com uma lista de códigos de voo como valor. Assim, foi possível registrar corretamente todos os voos que um passageiro comprou. Paralelamente, o dicionário voos manteve o código do voo como chave, com seus dados e número de lugares disponíveis. Essa separação de responsabilidades entre os dicionários trouxe mais clareza e eficiência ao gerenciamento dos dados no sistema.

5. COMPILADOR UTILIZADO

O sistema foi desenvolvido e testado principalmente no Visual Studio Code (VSCode), utilizando o interpretador padrão do Python 3.11. Também foram realizados testes pontuais no IDLE do Python para validação rápida de funcionalidades

6. REFERÊNCIA BIBLIOGRÁFICA

Pontifícia Universidade Católica de Campinas. "Notas de aula da disciplina Algoritmos de Programação, Projetos e Computação – APPC". 2025.