

UNIVERSIDADE FEDERAL DE OURO PRETO
CAMPUS MORRO DO CRUZEIRO - OURO PRETO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

DIOGO TADEU CAMPOS
THIAGO HENRIQUE DE OLIVEIRA GONÇALVES
WENDEL CAUÃ SILVA DE OLIVEIRA

TRABALHO PRÁTICO
BCC 221 - PROGRAMAÇÃO ORIENTADA À OBJETOS (POO)
SISTEMA DE CONTROLE DE VOO EM AEROPORTO

OURO PRETO
2025

DIOGO TADEU CAMPOS
THIAGO HENRIQUE DE OLIVEIRA GONÇALVES
WENDEL CAUÃ SILVA DE OLIVEIRA

SISTEMA DE CONTROLE DE VOO EM AEROPORTO

OURO PRETO
2025

SUMÁRIO

1. INTRODUÇÃO.....	4
1.1. Objetivo Geral.....	5
1.2. Objetivos Específicos.....	6
1.3. Justificativa.....	7
2. DESENVOLVIMENTO.....	8
2.1. Especificações de Requisitos.....	8
2.1.1. Objetivo.....	8
2.1.2. Requisitos Funcionais.....	8
2.1.2.1. Cadastro.....	8
2.1.2.2. Criação de Voo.....	8
2.1.2.3. Embarque de Passageiro.....	8
2.1.2.4. Listagem.....	8
2.1.2.5. Persistência.....	9
2.1.2.6. Validação.....	9
2.1.2.7. Relatórios e Estatísticas.....	9
2.1.3. Requisitos Não Funcionais.....	10
2.1.4. Fluxo Básico de Uso.....	10
2.2. Diagrama de Classes.....	11
3. CONCLUSÃO.....	12

1.INTRODUÇÃO

O presente relatório técnico detalha o desenvolvimento e a implementação de um Sistema de Controle de Voo em Aeroporto, concebido como trabalho prático da disciplina de Programação Orientada a Objetos (POO). Este projeto tem como propósito central a aplicação prática e a consolidação de diversos conceitos fundamentais da programação orientada a objetos, tais como a criação e manipulação de classes e objetos, o emprego do encapsulamento para garantir a integridade dos dados, o estabelecimento de associações entre diferentes classes, a implementação de herança e polimorfismo para promover a flexibilidade e reusabilidade do código, além da manipulação de arquivos para persistência de dados e a utilização eficiente da Standard Template Library (STL) do C++. O sistema proposto visa simular de forma básica as operações de controle de voo em um ambiente aeroportuário, englobando o gerenciamento de informações críticas sobre voos, aeronaves, passageiros e pilotos, e permitindo o registro detalhado de embarques. A interação com o usuário será realizada por meio de uma interface de linha de comando, e a persistência dos dados será assegurada através do armazenamento em arquivos no formato CSV. Ao longo deste documento, serão apresentadas as decisões de design e arquitetura adotadas, a descrição detalhada das funcionalidades implementadas, os métodos de teste empregados e as conclusões relevantes extraídas durante o processo de desenvolvimento do sistema.

1.1. Objetivo Geral

Desenvolver um sistema em C++ que simule o controle básico de voos em um aeroporto, aplicando os princípios de Programação Orientada a Objetos, como classes e objetos, encapsulamento, associação, herança e polimorfismo. O sistema deverá gerenciar informações cruciais sobre voos, aeronaves, passageiros e pilotos, permitindo o registro e a associação de passageiros a voos. A interação com o usuário ocorrerá por meio de uma interface de linha de comando, e a persistência dos dados será assegurada através da leitura e gravação automática em arquivos CSV

1.2. Objetivos Específicos

Implementar um sistema capaz de cadastrar aeronaves, registrando código, modelo, capacidade, velocidade média (em milhas por hora) e autonomia de voo (distância em milhas). Desenvolver a funcionalidade de cadastro de pessoas, distinguindo pilotos (com nome, matrícula, brevê, horas de voo) e passageiros (com nome, CPF, número do bilhete). Criar voos, associando um código, origem, destino, distância da viagem, aeronave, hora de saída prevista, número de escalas estimado (calculado com base na distância do voo e na autonomia da aeronave) e tempo estimado de voo (calculado com base na distância e velocidade da aeronave, com 1 hora adicional por escala). Permitir a designação de um comandante e um primeiro oficial (ambos pilotos) para cada voo. Desenvolver a funcionalidade de embarque de passageiros em voos, respeitando a capacidade máxima da aeronave. Implementar a listagem de voos, exibindo o código do voo, o código e modelo da aeronave, a matrícula do piloto comandante, origem e destino, o número de passageiros, e as horas de saída e chegada previstas. Permitir a listagem de passageiros de um voo específico, exibindo o código e modelo da aeronave e o nome de cada passageiro no voo, mediante a informação do código do voo. Garantir a persistência dos dados através do salvamento e carregamento automático de informações sobre aeronaves, pessoas e voos em arquivos no formato CSV (aeronaves.csv, pessoas.csv, voos.csv). Construir uma interface interativa via linha de comando (terminal) com um menu numerado para acesso às funcionalidades do sistema.

1.3. Justificativa

A justificativa para o desenvolvimento deste sistema reside na sua relevância como ferramenta de aprendizado e aplicação prática dos fundamentos de Programação Orientada a Objetos (POO). A disciplina de POO é central na formação de profissionais da computação, e a capacidade de modelar problemas do mundo real em termos de classes e objetos é uma habilidade essencial.

Este projeto, ao simular um sistema de controle de voo, oferece um cenário complexo e multidisciplinar que exige a aplicação de conceitos como encapsulamento, herança, polimorfismo e associação entre classes. A manipulação de arquivos para persistência de dados e o uso da Standard Template Library (STL) do C++ complementam o aprendizado, proporcionando experiência prática com ferramentas e técnicas amplamente utilizadas no desenvolvimento de *software*.

Portanto, o valor deste trabalho transcende a simulação em si, configurando-se como um exercício acadêmico fundamental para solidificar a compreensão teórica de POO, desenvolver habilidades de resolução de problemas e preparar os estudantes para desafios futuros no campo da engenharia de software.

2. DESENVOLVIMENTO

2.1. Especificações de Requisitos

2.1.1. Objetivo

O sistema deve gerenciar aeronaves, pilotos, passageiros e voos, permitindo cadastro, associação e persistência dos dados em arquivos CSV.

2.1.2 Requisitos Funcionais

2.1.2.1. Cadastro

- **Aeronave:** O usuário pode cadastrar aeronaves informando código, modelo, capacidade, velocidade média e autonomia.
- **Piloto:** O usuário pode cadastrar pilotos informando nome, matrícula, brevê e horas de voo.
- **Passageiro:** O usuário pode cadastrar passageiros informando nome, CPF e número do bilhete.

2.1.2.2. Criação de Voo

- O usuário pode criar voos informando código do voo, origem, destino, distância, hora de saída, código da aeronave, matrícula do comandante e matrícula do primeiro oficial.
- O sistema verifica se a aeronave e os pilotos existem antes de criar o voo.

2.1.2.3. Embarque de Passageiro

- O usuário pode embarcar um passageiro em um voo, informando o código do voo e o CPF do passageiro.
- O sistema verifica se o voo, o passageiro e a aeronave existem e se há capacidade disponível.

2.1.2.4. Listagem

- Voos: O sistema lista todos os voos cadastrados, mostrando informações detalhadas, incluindo aeronave, pilotos, origem, destino, passageiros, hora de saída, tempo estimado e número de escalas.
- Passageiros de um Voo: O sistema lista todos os passageiros embarcados em um voo específico.

2.1.2.5. Persistência

- Todos os dados (aeronaves, pessoas, voos) são salvos em arquivos CSV.
- Ao iniciar, o sistema carrega os dados dos arquivos.
- Ao cadastrar, criar ou embarcar, os dados são atualizados nos arquivos.

2.1.2.6. Validação

- Não é permitido cadastrar aeronaves, pilotos, passageiros ou voos com identificadores já existentes.
- Não é permitido embarcar passageiros inexistentes ou duplicados em um voo.
- Não é permitido criar voos com aeronaves ou pilotos inexistentes.

2.1.2.7. Relatórios e Estatísticas:

Oferece os seguintes relatórios detalhados:

1. **Número Total de Voos Cadastrados:** Apresenta a quantidade completa de voos registrados no sistema.
2. **Média de Passageiros por Voo:** Calcula e exibe a média de passageiros a bordo por voo.
3. **Aeronaves Mais Utilizadas:** Lista as aeronaves mais frequentemente empregadas, baseando-se no número de voos realizados por cada uma.
4. **Passageiros com Múltiplos Voos:** Identifica e lista os passageiros que embarcaram em mais de um voo, detalhando o número de voos para cada um.
5. **Voos com Alta Ocupação ($\geq 90\%$):** Exibe os voos cuja taxa de ocupação atingiu 90% ou mais da capacidade total da aeronave.
6. **Distância Total Percorrida por Aeronave:** Mostra a soma das distâncias de todos os voos para cada aeronave individualmente.

Além dessas funcionalidades, o sistema permite que qualquer relatório gerado seja salvo em um arquivo `.txt` no local desejado pelo usuário.

2.1.3. Requisitos Não Funcionais

- Persistência: Os dados devem ser armazenados em arquivos CSV separados para aeronaves, pessoas e voos.
- Interface: O sistema deve ser operado via terminal, com menus e prompts claros.
- Robustez: O sistema deve validar entradas do usuário e tratar erros de leitura/gravação de arquivos.
- Portabilidade: O sistema deve funcionar em ambientes Windows e Linux.

2.1.4. Fluxo Básico de Uso

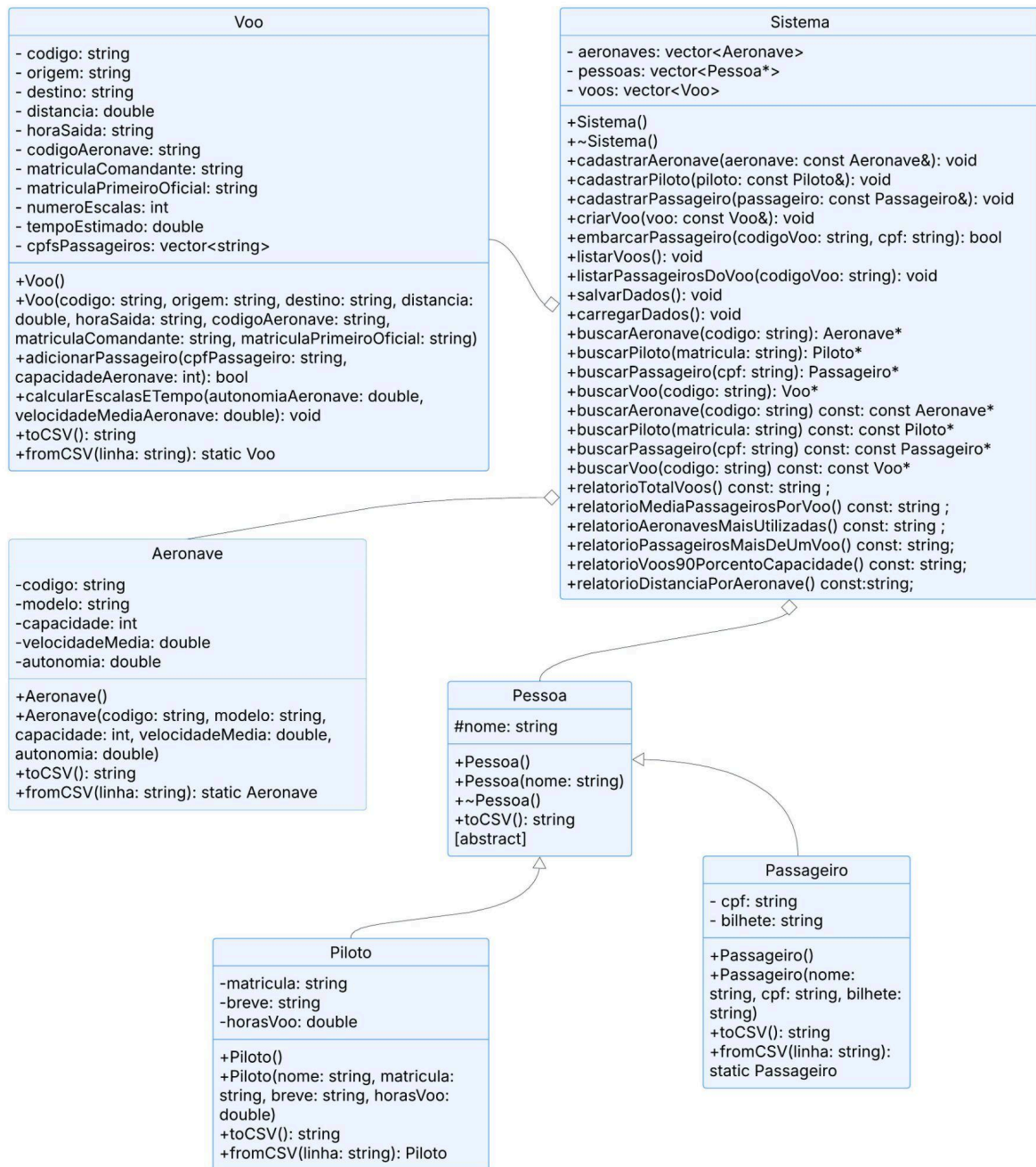
1. Ao iniciar, o sistema carrega os dados dos arquivos.
2. O usuário escolhe uma opção no menu:

```
===== SISTEMA DE CONTROLE DE VOOS =====  
1. Cadastrar aeronave  
2. Cadastrar piloto  
3. Cadastrar passageiro  
4. Criar voo  
5. Embarcar passageiro em voo  
6. Listar voos  
7. Listar passageiros de um voo  
8. Relatórios e Estatísticas  
9. Salvar dados e sair  
=====
```

Escolha uma opção:

3. O sistema executa a operação, valida os dados e salva as alterações.
4. 4. Ao sair, todos os dados são persistidos.

2.2. Diagrama de Classes



3. CONCLUSÃO

O desenvolvimento do Sistema de Controle de Voo em Aeroporto permitiu a aplicação prática dos conceitos fundamentais da Programação Orientada a Objetos (POO) abordados ao longo da disciplina. A implementação em C++ proporcionou uma experiência completa no uso de técnicas como encapsulamento, herança, polimorfismo e associação entre classes, além da manipulação de arquivos para garantir a persistência dos dados.

O sistema atendeu aos requisitos funcionais e não funcionais propostos, permitindo o cadastro de aeronaves, pilotos e passageiros, a criação de voos, o embarque de passageiros e a listagem de informações de forma clara e organizada por meio de uma interface de linha de comando. A utilização de arquivos CSV garantiu a portabilidade e a persistência das informações, assegurando que os dados sejam preservados entre as execuções do programa.

Além disso, a implementação de relatórios e estatísticas no sistema permitiu ao usuário obter uma visão gerencial dos dados, como o total de voos cadastrados, a média de passageiros por voo, a identificação das aeronaves mais utilizadas e o acompanhamento da participação dos passageiros em múltiplos voos. Esses recursos reforçaram a utilidade prática do sistema, aproximando-o de demandas reais de gestão e análise em ambientes aeroportuários.

Durante o desenvolvimento, desafios como o controle de validações e a consistência das associações entre os objetos foram superados, reforçando a compreensão da importância do design de software robusto e modular.

Conclui-se que o projeto cumpriu seu papel acadêmico ao consolidar conhecimentos teóricos em uma aplicação prática, além de fornecer uma base sólida para o desenvolvimento de sistemas mais complexos no futuro. O sistema desenvolvido representa uma solução funcional e didática para simular um ambiente simplificado de gerenciamento de operações aeroportuárias, sendo um exercício valioso para o aprimoramento das habilidades de programação e modelagem de sistemas orientados a objetos.