



Universidade Federal do Espírito Santo  
Departamento de Informática

## Trabalho Prático

### **SkyManager - Sistema de Gerenciamento de Operações Aéreas**

Programação II - 2025/2

09 de dezembro de 2025

### **Objetivo**

O objetivo deste trabalho é colocar em prática as habilidades de programação na linguagem C adquiridas ao longo do curso. O trabalho foi elaborado para você exercitar diversos conceitos, principalmente a construção de TADs opacos e genéricos, alocação e manipulação de memória de maneira dinâmica. Como consta no plano de ensino da disciplina, o trabalho será realizado em duas etapas (a primeira realizada em casa e a segunda em sala de aula), sendo esta a primeira etapa. O somatório de ambas as etapas **equivale a 50% da nota final do curso**.

### **Introdução**

Companhias aéreas modernas precisam administrar uma quantidade enorme de dados: voos diários, passageiros transportados, funcionários atuando em turnos e aeronaves distribuídas entre diferentes rotas. Essas empresas empregam sistemas computacionais robustos para garantir que reservas sejam feitas corretamente, tripulações sejam alocadas, voos sejam planejados e passageiros recebam as informações adequadas.

Neste trabalho, você implementará o **SkyManager**, um protótipo simplificado de sistema de gestão de operações aéreas.

### **Descrição geral do sistema**

O SkyManager deve controlar digitalmente as operações de uma companhia aérea. Para isso, considere:

- Existem dois atores principais: **passageiros** e **tripulantes**
- O sistema deve ser capaz de cadastrar ambos os atores
- Passageiros podem criar **reservas de voo**
- Tripulantes são alocados para voos com base na disponibilidade
- Cada reserva possui um conjunto específico de informações
- O sistema deve manter uma **fila de reservas**
- O sistema deve criar relatórios de passageiros, tripulantes e voos

Funcionalidades essenciais:

1. Cadastro de Passageiros e Tripulantes
2. Cadastro de Aeronaves e Voos
3. Solicitação de reservas de voos
4. Outras ações

## Cadastro de Passageiros e Tripulantes

A primeira funcionalidade do SkyManager é o cadastro de atores do sistema. Os atores possuem dados em comum e dados específicos. Em comum, todos eles possuem:

- Nome completo (string com no máximo 100 caracteres)
- CPF (no formato: 000.000.000-00)
- Data de nascimento (segundo padrão dd/mm/aaaa)
- Telefone (formato: (00)00000-0000)
- Gênero (MASCULINO, FEMININO, OUTROS)

**O CPF é o identificador único de um ator.** O sistema não deve permitir cadastro de atores do mesmo tipo com um mesmo CPF. Por exemplo, não é permitido o cadastro de dois tripulantes com o mesmo CPF. Caso seja realizada uma tentativa de cadastro com o mesmo CPF o sistema deve **ignorar** e não realizar o cadastro.

Além das informações a seguir, um **passageiro** também possui a seguinte informação:

- Programa de fidelidade (BRONZE, PRATA ou OURO)

No caso de um **tripulante**, ele também possui:

- Cargo (PILOTO ou COMISSARIO)
- Horas trabalhadas (inteiro)
- Salário (float)

O cadastro de ambos os atores é realizado via entrada padrão e são representados pelos caracteres **P** e **T**, cadastro do **P**assageiro e **T**ripulante, respectivamente. Um exemplo de cadastro para ambos os atores é ilustrado a seguir:

```
P (caractere indicando cadastro de um passageiro)
MARIA ESTEVES (nome)
123.456.789-00 (CPF)
10/02/1980 (data de nascimento)
(27)98765-0909 (telefone)
FEMININO (gênero)
OURO (programa de fidelidade)
T (caractere indicando cadastro de um tripulante)
ROBERTO MENEZES (nome)
987.654.321-99 (CPF)
15/03/1970 (data de nascimento)
(11)91234-5678 (telefone)
MASCULINO (gênero)
PILOTO (cargo)
40 (horas trabalhadas)
18000 (salário)
```

## Cadastro de Aeronaves e Voos

O cadastro de aeronave e voo seguem os mesmos princípios anteriores. Uma deve conter as seguintes informações:

- Código (string com no máximo 20 caracteres, ex: AZUL320, BOEING73)
- Modelo (string com no máximo 50 caracteres)
- Capacidade (número inteiro de assentos)

Já um voo, contém:

- Código do voo (string com no máximo 20 caracteres, ex: SM1234)
- Origem (string com no máximo 30 caracteres)
- Destino (string com no máximo 30 caracteres)
- Aeronave utilizada (referenciada pelo código cadastrado)
- Data do voo (seguindo padrão dd/mm/aaaa)
- Hora de saída (inteiro de 0 a 23)
- Hora de chegada (inteiro de 0 a 23)
- Preço do voo (float)

Para que um voo seja cadastrado, a **aeronave solicitada deverá existir e está disponível na data e horário solicitado**. Caso contrário, o voo é ignorado. **Não** é permitido cadastrar aeronaves e voos com o mesmo código. Se isso acontecer, o cadastro deve ser ignorado.

O cadastro de ambos os atores é realizado via entrada padrão e são representados pelos caracteres A e V, cadastro do **A**eronave e **V**oo, respectivamente. Um exemplo de cadastro para ambos os atores é ilustrado a seguir:

```
A (caractere indicando cadastro de uma aeronave)
BOEING73 (código da aeronave)
BOEING 737-800 (modelo da aeronave)
172 (capacidade da aeronave)
V (caractere indicando cadastro de um voo)
SM3410 (código do voo)
VITORIA (origem do voo)
BELO HORIZONTE (destino do voo)
BOEING73 (código da aeronave que vai operar)
10/11/2025 (data do voo)
9 (hora de saída)
11 (hora de chegada)
1200 (preço do voo)
```

## Solicitação de reservas de voos

Passageiros podem solicitar reservas para os voos cadastrados no sistema. Existem três tipos de reserva: econômica, executiva, e prime. Cada reserva possui as seguintes características:

- **Econômica:**
  - N° de bagagens despachadas (inteiro)
  - Necessita de assento especial (Sim ou Não)
- **Executiva:**
  - N° de bagagens despachadas
  - Necessita de refeição especial (Sim ou Não)
  - Acesso ao lounge (Sim ou Não)
- **Prime:**
  - N° de bagagens despachadas
  - Tipo de refeição (COMPLETA, PARCIAL, NENHUMA)
  - Necessita de motorista (Sim ou Não)

O tipo de reserva afeta no preço da passagem. A reserva econômica possui o mesmo preço já cadastrado no voo. A reserva executiva adiciona 30% no valor e a prime adiciona 70%. Por exemplo, se o preço do voo é R\$ 1000,00, uma reserva econômica custará o mesmo valor, uma executiva custará R\$ 1300,00 e uma prime custará R\$ 1700,00. Além disso, é cobrado um valor de R\$ 50,00 por mala despachada.

Por fim, o programa de fidelidade de um passageiro pode garantir um desconto no preço da passagem. Passageiros BRONZE não possuem desconto. Passageiros PRATA possuem 2% de desconto no valor final. Já passageiros OURO possuem 5% de desconto.

Uma reserva pode ser solicitada da seguinte forma:

```
R (caractere indicando a solicitação de uma reserva)
255.942.213-22 (CPF do passageiro solicitante)
SM3410 (código do voo solicitado)
ECONOMICA (tipo da reserva)
2 (quantidade de malas despachadas)
NAO (necessidade de assento especial)
R (caractere indicando a solicitação de uma reserva)
255.942.213-22 (CPF do passageiro solicitante)
SM3410 (código do voo solicitado)
EXECUTIVA (tipo da reserva)
2 (quantidade de malas despachadas)
NAO (necessidade de refeição especial)
SIM (acesso ao lounge)
R (caractere indicando a solicitação de uma reserva)
255.942.213-22 (CPF do passageiro solicitante)
SM3410 (código do voo solicitado)
PRIME (tipo da reserva)
2 (quantidade de malas despachadas)
COMPLETA (tipo de refeição)
NAO (necessidade de motorista)
```

Observe que os dados mudam de acordo com o tipo de reserva solicitada.

Sempre que uma reserva é solicitada, ela entra em uma **fila de processamento**. Quando solicitada, essa fila é processada pelo programa. Uma reserva será confirmada com sucesso se ela cumprir os seguintes requisitos:

- Foi realizada por um passageiro cadastrado no sistema
- Foi solicitado um voo cadastrado no sistema

Se os requisitos não forem cumpridos, a reserva será cancelada. Para todas as reservas concluídas, é calculado o preço final de acordo com os dados do voo e solicitações do passageiro.

Um exemplo da solicitação de processamento das reservas é mostrado a seguir:

```
E (caractere indicando que alguma ação será executada)
```

```
PROCESSA RESERVAS(ação solicitada)
```

## Outras ações disponíveis no sistema

O sistema deve permitir a execução de outras ações baseadas nos dados cadastrados e reservas solicitadas pelos passageiros. Essas ações são descritas a seguir.

## Consulta das reservas

Todas as reservas solicitadas podem ser consultadas. Um exemplo de solicitação de consulta de reservas é mostrado a seguir:

```
E (caractere indicando que alguma ação será executada)
CONSULTA RESERVAS (ação solicitada)
```

Uma vez solicitada, todas as reservas solicitadas devem ser exibidas na saída padrão. Um exemplo de saída é mostrado a seguir:

```
===== RESERVA =====
- ID: Res-1
- Passageiro: 255.942.213-22
- Status: SOLICITADA
- Código do voo: SM3410
- Preço: 0.00
- Tipo: PRIME
- Numero de Bagagens: 2
- Refeicao: COMPLETA
- Motorista: NAO
=====
```

O exemplo anterior é realizado para uma reserva do tipo PRIME. As reservas do tipo ECONOMICA e EXECUTIVA, devem apresentar as informações pertinentes a elas.

Uma reserva pode ter 3 status:

- SOLICITADA: ainda não foi processada, ou seja, ainda não foi executado o processamento das reservas. Neste caso, o preço deve aparecer zerado.
- CONFIRMADA: o processamento foi realizado e a reserva foi concluída com sucesso. Neste caso, o preço deve ser calculado e apresentado.

- CANCELADA: o processamento foi realizado e a reserva foi cancelada por não cumprir todos os requisitos. Neste caso, o preço deve aparecer zerado.

## Banco de passageiros

O sistema deve permitir que todos os passageiros cadastrados sejam exibidos na tela seguindo a ordem de cadastro. Essa ação é solicitada da seguinte forma:

```
E (caractere indicando que alguma ação será executada)
PASSAGEIROS (ação solicitada)
```

Uma vez solicitada, os passageiros cadastrados até aquele momento devem ser exibidos na tela. Um exemplo para um único usuário é mostrado a seguir:

```
- Nome: JOSE ROMEU
- CPF: 123.932.250-46
- Data de Nascimento: 1/9/1955
- Telefone: (34)96373-7004
- Genero: OUTROS
- Classe: BRONZE
- Reservas confirmadas: 2
```

Observe que além das informações do passageiro também é exibido quantas reservas já foram confirmadas para ele (0 ou mais).

## Banco de tripulantes

O sistema deve permitir que todos os tripulantes cadastrados sejam exibidos na tela seguindo a ordem de cadastro. Essa ação é solicitada da seguinte forma:

```
E (caractere indicando que alguma ação será executada)
TRIPULANTES (ação solicitada)
```

Uma vez solicitada, os tripulantes cadastrados até aquele momento devem ser exibidos na tela. Um exemplo para um único tripulante é mostrado a seguir:

```
- Nome: JAYCE BARBOSA
- CPF: 854.632.894-20
```

```
- Data de Nascimento: 22/2/1990
- Telefone: (37)9938-3382
- Genero: MASCULINO
- Cargo: PILOTO
- Horas trabalhadas: 10
- Salario: 22000.00
```

## Rankings

Existe uma variação possível dos bancos de passageiros e tripulantes. A solicitação pode vir com a palavra RANKING à frente, por exemplo:

```
E (caractere indicando que alguma ação será executada)
RANKING TRIPULANTES (ação solicitada)
```

ou

```
E (caractere indicando que alguma ação será executada)
RANKING PASSAGEIROS (ação solicitada)
```

Neste caso a exibição **não** segue a ordem de cadastro. No caso dos tripulantes, eles devem ser exibidos de acordo com a quantidade de tempo trabalhado. Já para os passageiros, eles devem ser exibidos de acordo com a quantidade de reservas confirmadas. Em ambos os casos o ranking segue ordem decrescente, ou seja, para os tripulantes quem trabalhou mais tempo deve aparecer no topo. Naturalmente, para os passageiros, o topo pertence a quem possui mais reservas confirmadas. Para ambos os casos, em caso de empate, os atores devem aparecer em ordem alfabética.

## Relatório geral

Por fim, o sistema deve ser capaz de gerar um relatório geral do sistema. Essa solicitação é realizada da seguinte forma:

```
E (caractere indicando que alguma ação será executada)
RELATORIO (ação solicitada)
```

O relatório deve exibir as seguintes informações:

- Número total de aeronaves cadastradas
- Número total de voos cadastrados
- Número total de reservas



- Número total confirmadas com sucesso
- Número total de reservas canceladas
- Quantidade de passageiros cadastrados
- Média de idade dos passageiros
- Quantidade de tripulantes cadastrados
- Média de horas trabalhadas dos tripulantes (somente parte inteira)

Essas informações devem ser exibidas na tela da seguinte forma

```
===== RELATORIO =====  
- Aeronaves: 5  
- Voos: 7  
- Reservas: 20  
- Reservas confirmadas: 15  
- Passageiros: 20  
- Media de idade passageiros: 48  
- Tripulantes: 8  
- Media de horas trabalhadas: 15  
=====
```

**Importante:** o relatório reflete as informações no momento em que ele foi solicitado.

## Informações gerais

Alguns arquivos de entrada e respectivos arquivos de saída serão fornecidos para o aluno. O aluno deverá utilizar tais arquivos para testes durante a implementação do trabalho. É de responsabilidade do aluno criar novos arquivos para testar outras possibilidades do programa e garantir seu correto funcionamento. O trabalho será corrigido usando, além dos arquivos dados, outros arquivos (específicos para a correção e não disponibilizados para os alunos) seguindo a formatação descrita neste documento. Em caso de dúvida, entre em contato com o professor e/ou monitores.

O uso de arquivos com formatação diferente poderá acarretar em incompatibilidade durante a correção do trabalho e consequentemente na impossibilidade de correção do mesmo (sendo atribuído a nota zero). Portanto, siga estritamente o formato estabelecido.

## Implementação e verificação de resultados

A implementação deverá seguir o *template* de código disponibilizado no repositório do trabalho dentro da organização da disciplina no Github:

- Link para o repositório:

<https://github.com/prog-II-ufes/2025-2-template-TP-etapa-1>

A utilização do template segue os mesmos princípios dos exercícios disponibilizados ao longo da disciplina. Você deve utilizar os arquivos .h disponibilizados e, obrigatoriamente, implementar todas as funcionalidades requisitadas para a construção destes TADs. Todavia, vocês são livres para implementar outros TADs e/ou bibliotecas que julgarem necessárias. Além disso, você já possui acesso ao script de correção automática de exercícios e trabalhos. Ele será utilizado para corrigir o seu trabalho. Desta forma, é extremamente importante que você avalie seu trabalho utilizando o script e os casos de teste disponibilizados.

Para fins de depuração, você também pode utilizar softwares de comparação de arquivos, como diff e o Meld (alternativa gráfica do diff). Diferenças na formatação poderão impossibilitar a comparação e consequentemente impossibilitar a correção do trabalho. O programa será considerado correto se gerar a saída esperada idêntica à fornecida com os casos de teste.

## Pontuação e Processo de Correção

A primeira etapa do trabalho será pontuada de acordo com sua implementação e seguindo os pontos apresentados na tabela a seguir.

Item	Descrição	Peso
fila.h	TAD relacionado a fila	10%
reserva.h	TAD relacionado a uma reserva	10%
economica.h	TAD relacionado ao tipo de reserva ECONOMICA	5%
main.c + restante	Função principal do programa + restante do código necessário	75%

A pontuação da tabela será utilizada para calcular a nota para cada caso de teste. A pontuação final será a média de pontos considerando todos os casos. Haverá casos de teste visíveis e ocultos (que serão divulgados durante a segunda etapa do trabalho em sala).

A pontuação obtida de acordo com a tabela anterior será utilizada para calcular a nota final do seu trabalho. O gerenciamento de memória e modularização são itens obrigatórios. Sendo assim, não basta apenas acertar a saída, é necessário seguir os

templates de código disponibilizados e gerenciar corretamente o uso da memória. Logo, considere que seu trabalho seja capaz de gerar todos os arquivos corretamente e obtenha pontuação igual a 10. A partir desta pontuação a sua nota final será calculada da seguinte maneira:

(1) primeiramente será verificado se você seguiu corretamente o template de código fornecido.

Se você implementou todos os templates corretamente, não haverá nenhum desconto na sua pontuação nesta fase. Porém, se você deixar de implementar algum dos templates, sua pontuação será descontada de acordo com o peso do template.

(2) após a verificação dos templates, será verificado o gerenciamento de memória através do Valgrind. Caso o seu programa apresente vazamento de memória e/ou erro de memória será **descontado 50%** da pontuação que chegar até essa fase.

Sendo assim, é de **suma importância que seu trabalho siga a risca os templates e gerencie corretamente a memória** (lembre-se, esses são os principais objetivos desta disciplina). Todo esse processo de pontuação será realizado por meio do script de correção que vocês já possuem acesso. Observe que a compilação da pasta completo (dentro de resultados) no script não é utilizado para computar a pontuação. Porém, pode ser utilizado para depuração de erros no seu programa.

**Processo de correção do trabalho:** como consta no plano de ensino da disciplina, o trabalho prático é dividido em duas etapas. A primeira etapa é o desenvolvimento do projeto seguindo as regras e descrições apresentadas neste documento. A segunda etapa consiste na correção, atualização e/ou incremento de funcionalidades implementadas na etapa anterior. Esta última etapa é realizada em sala de aula de acordo com cronograma da disciplina. **A nota final do trabalho é a média ponderada entre as duas etapas, de acordo com:**

$$\text{Nota final} = 0.2 * \text{Etapa 1} + 0.8 * \text{Etapa 2}$$

É importante ressaltar que ao iniciar a etapa 2 vocês terão acesso à nota e aos casos de teste ocultos da etapa 1. Portanto, vocês poderão corrigir possíveis erros cometidos durante o desenvolvimento da etapa 1.

## Prazo e submissão

Todas as informações referentes a prazo de entrega e forma de submissão estão disponíveis na atividade **Trabalho Prático - Etapa 1** no classroom/ava da disciplina.

## Regras gerais

Para todas as etapas do trabalho, considere as seguintes regras:

- Trabalhos entregues após o prazo **não** serão corrigidos (recebendo a nota zero)
- O trabalho deverá ser feito **individualmente** e pelo próprio aluno, isto é, o aluno deverá necessariamente conhecer e dominar **todos** os trechos de código criados. Cada aluno deverá trabalhar independente dos outros, não sendo permitido a cópia ou compartilhamento de código.
  - Haverá verificação automatizada de plágio. Trabalhos identificados como iguais, em termos de programação (por exemplo, mudar nomes de variáveis e funções, entre outros, não faz dois trabalhos serem diferentes), **serão penalizados com a nota zero e poderão ser submetidos para penalidades adicionais em instâncias superiores**. Isso também inclui a pessoa que forneceu o trabalho, sendo, portanto, de sua obrigação a proteção de seu trabalho contra cópias ilícitas. Proteja seu trabalho e não esqueça cópias do seu código nas máquinas de uso comum.
- Nenhuma entrada dos dados nos casos de teste vai possuir acentos ou caracteres especiais. Além disso, estarão sempre em caixa alta (ex: JOAO).
- Todos os arquivos de entrada vão seguir sempre o mesmo padrão.
- Todas as entradas de dados serão fornecidas de acordo com a descrição. Por exemplo, se o padrão do CPF é 000.000.000-00, todos CPFs fornecidos vão estar neste padrão.
- Para dados que resulte em flutuante, use sempre a parte inteira. Por exemplo, 3.14 deve ser exibido como 3.
- **Para calcular a idade dos atores considere a data de referência como sendo 10/12/2025**
- Para dados multidimensionais no qual o tamanho é desconhecido, é obrigatório o uso de alocação dinâmica de memória. Liberar a memória alocada é de sua responsabilidade.
  - Caso você tente burlar essa regra, você será penalizado
- O programa deve ser encerrado quando o caractere F (de **F**inalizar) for lido pela entrada padrão.

- Modularização e organização são fundamentais na construção do código. Você é obrigado a implementar os TADs solicitados via o template dos .h. Todavia, isso não impede a criação de outros.
- Todos os trabalhos serão corrigidos utilizando o sistema operacional Linux. **Garanta que seu código funcione nos computadores do Labgrad.** Qualquer discrepância de resultado na correção por conta de Sistema Operacional, os computadores do Labgrad serão utilizados para solucionar o problema.
- **Todos os TADs implementados no trabalho devem ser opacos/encapsulados**
- Possíveis problemas na descrição do trabalho serão solucionados e comunicados via Discord/Classroom. Don't Panic!

## Considerações finais

Esse documento descreve de maneira geral as regras de implementação do trabalho. É de responsabilidade do aluno garantir que o programa funcione de maneira correta e amigável com o usuário. Qualquer alteração nas regras do trabalho será comunicada em sala e no classroom. É responsabilidade do aluno frequentar as aulas e se manter atualizado em relação às especificações do trabalho. Caso seja notada qualquer tipo de inconsistência nos arquivos de testes disponibilizados, comunique imediatamente ao professor para que ela seja corrigida e reenviada para os alunos.