

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**  
**INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA**

Curso de Engenharia de Software

Trabalho Interdisciplinar: AEDS I e Fundamentos da Engenharia de Software.

Nome dos integrantes do grupo:

**Arthur Henrique de Lima Santos, Karen Joilly Araujo Gregorio de Almeida,  
Luiz Felipe Alves da Silva e Mirna dos Reis Arcanjo.**

Nome do sistema: Tripulação do Código

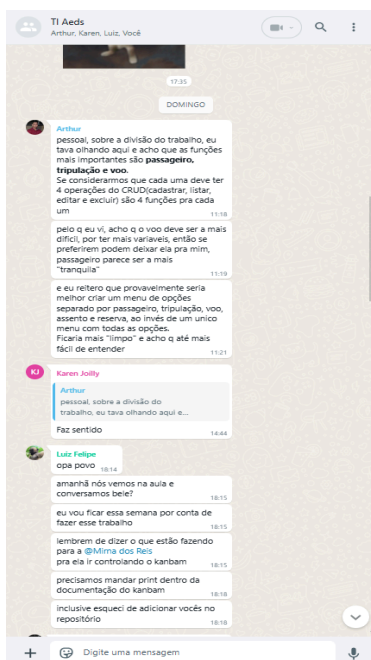
Dezembro, 2024

## Detalhamento das Sprints

05/12 - Planejamento das Sprints: Divisão de tarefas.

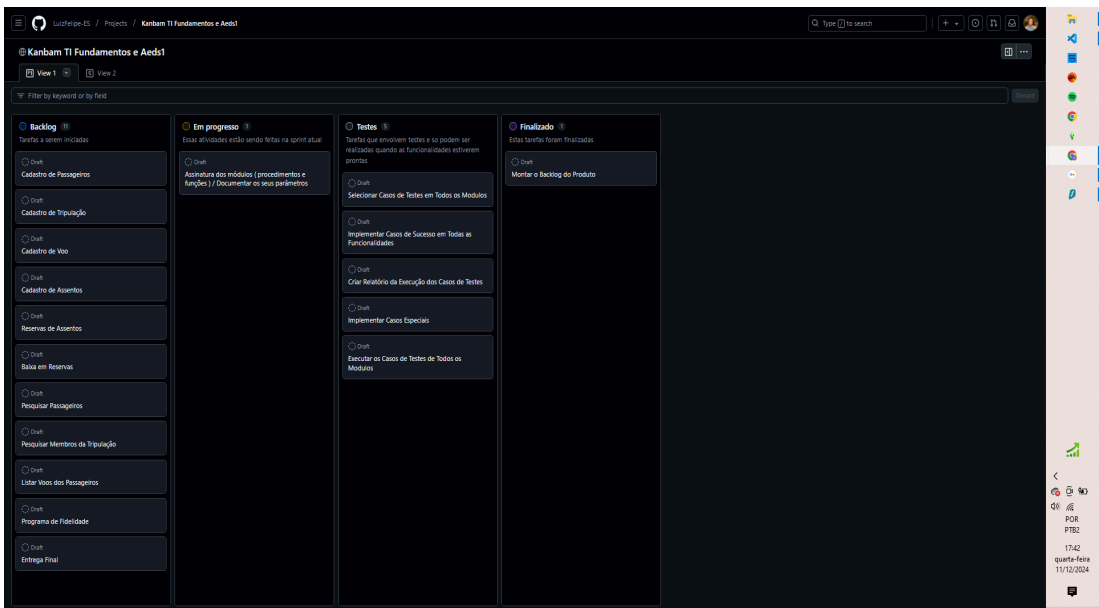


Os *daily*s ocorreram pelo *WhatsApp*. Fizemos um grupo para nos comunicar melhor.

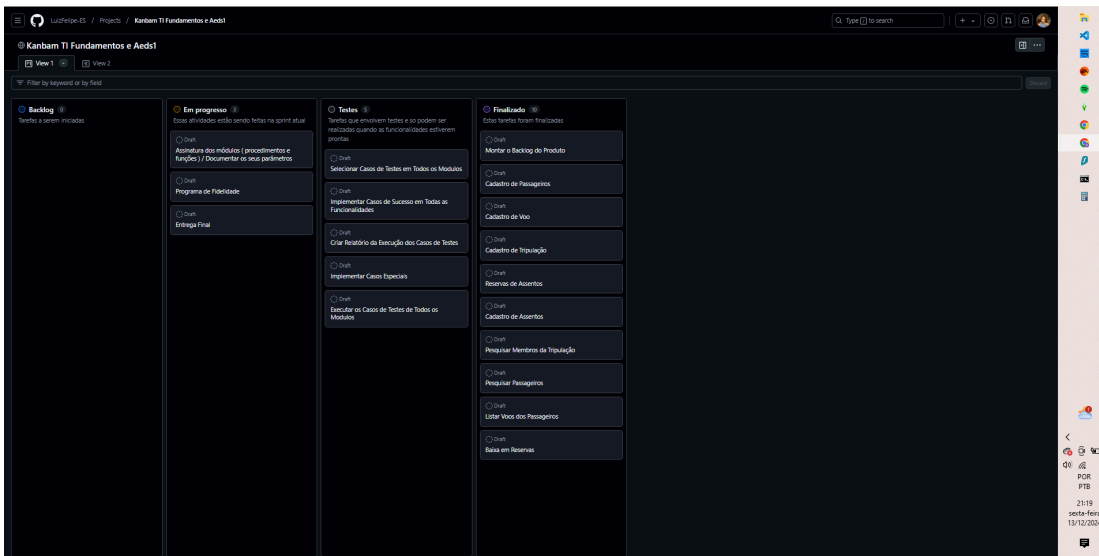


Utilizamos o Kanban do GitHub para melhor organização das tarefas.

11/12/2024 - Quarta-feira



13/12/2024 - Sexta-feira



## Divisões de tarefas

Arthur Henrique de Lima Santos	Elaboração de funções crud de voos, assentos.
Karen Joilly Araujo Gregorio de Almeida	Elaboração de funções "Passageiro": cadastrar, listar, editar e excluir.
Luiz Felipe Alves da Silva	Elaboração de funções "Tripulação": cadastrar, editar, excluir e listar.
Mirna dos Reis Arcanjo	Elaboração de documentação e testes das funções.

## Backlog do Produto

1. Cadastro de Passageiros
  - a. PASSAGEIRO: código, nome, endereço, telefone, fidelidade (sim/não), pontos fidelidade.
    - i. Cadastro de Passageiro:
      1. Deve garantir que não haja dois passageiros com o mesmo código.
      2. Opcionalmente, pode-se gerar o código automaticamente.
2. Cadastro de Tripulação
  - a. TRIPULAÇÃO: código, nome, telefone, cargo (piloto, copiloto, comissário)
    - i. Cadastro de Tripulação:
      - Cada membro da tripulação deve ter um cargo específico.
      - Deve-se garantir que não haja dois membros da tripulação com o mesmo código.
      - Opcionalmente, pode-se gerar o código automaticamente.
3. Cadastro de Voos
  - a. Deve ser possível cadastrar informações sobre data, hora, origem, destino, tarifa, tripulação e o avião.
  - b. Deve verificar a presença de ao menos um piloto e um copiloto para que o voo seja marcado como ativo.
4. Cadastro de Assentos
  - a. Deve ser possível cadastrar os assentos de cada voo.
5. Reservas de Assentos
  - a. Deve garantir que o assento esteja disponível antes de reservar.
  - b. Reservas duplicadas para o mesmo assento no mesmo voo devem ser evitadas.
6. Baixa em Reservas
  - a. Deve liberar o assento e atualizar o status para livre.
  - b. Calcular o valor total a ser pago, se necessário, de acordo com a tarifa do voo.
7. Pesquisa de Passageiros

8. Pesquisa de Membros da Tripulação
9. Listar voos dos passageiros
10. Programa de Fidelidade

**Planejamento dos casos de teste, a implementação dos casos de teste automatizados e o relatório de execução dos testes.**

- **Função:** `codigoPassageiroJaExiste`

**Objetivo:** Verificar se a função identifica corretamente a existência de um código de passageiro no arquivo passageiros.txt.

**Entradas e Saídas Esperadas**

Cenário	Entrada (Código)	Estado do Arquivo	Saída Esperada	Descrição
Código existente	123	Contém passageiro com código 123	1	Código encontrado no arquivo.
Código inexistente	999	Contém outros códigos, mas não o 999	0	Código não encontrado no arquivo.
Arquivo vazio	456	Arquivo está presente, mas está vazio	0	Arquivo não contém nenhum dado.
Arquivo não existente	789	Arquivo passageiros.txt não está presente	0	Arquivo não encontrado, código não pode existir.
Código no limite de 4 dígitos	9999	Contém passageiro com código 9999	1	Código no limite válido é identificado.
Código no limite inferior (0)	0	Contém passageiro com código 0	1	Código no limite inferior é identificado.

**Relatório de Execução dos Testes**

Teste	Resultado	Notas
Código existente	Aprovado	Função identificou corretamente o código.
Código inexistente	Aprovado	Função retornou 0 como esperado.
Arquivo vazio	Aprovado	Nenhum código encontrado.
Arquivo não existente	Aprovado	Função tratou corretamente a ausência.
Código no limite (9999)	Aprovado	Função identificou corretamente o código.
Código no limite (0)	Aprovado	Função identificou corretamente o código.

- **Função:** `codigoVooExiste`

**Objetivo:** Verificar se a função identifica corretamente a existência de um código de voo no arquivo voos.txt.

#### Entradas e Saídas Esperadas

Cenário	Entrada (Código)	Estado do Arquivo	Saída Esperada	Descrição
Código de voo existente	101	Contém voo com código 101	1	Código encontrado no arquivo.
Código de voo inexistente	999	Contém outros códigos, mas não o 999	0	Código não encontrado no arquivo.
Arquivo vazio	202	Arquivo está presente, mas está vazio	0	Arquivo não contém nenhum dado.
Arquivo não existente	303	Arquivo voos.txt não está presente	0	Arquivo não encontrado, código não pode existir.
Código no limite superior (9999)	9999	Contém voo com código 9999	1	Código no limite válido é identificado.
Código no limite inferior (0)	0	Contém voo com código 0	1	Código no limite inferior é identificado.

#### Relatório de Execução dos Testes

Teste	Resultado	Notas
Código existente	Aprovado	Função identificou corretamente o código.
Código inexistente	Aprovado	Função retornou 0 como esperado.
Arquivo vazio	Aprovado	Nenhum código encontrado.
Arquivo não existente	Aprovado	Função tratou corretamente a ausência.
Código no limite (9999)	Aprovado	Função identificou corretamente o código.
Código no limite (0)	Aprovado	Função identificou corretamente o código.

**Função:** cadastroDeVoo

**Objetivos:**

1. Cadastrar os dados de um voo no arquivo voos.txt.
2. Verificar se os dados fornecidos são válidos e se o voo foi corretamente adicionado ao arquivo.
3. Garantir que os dados sejam persistidos corretamente no arquivo.

**Entradas e Saídas Esperadas**

Entrada	Estado do Arquivo	Saída Esperada	Descrição
Cadastro válido de voo	Arquivo voos.txt com outros voos	Dados do voo cadastrados com sucesso!	Dados do voo fornecido são corretamente cadastrados no arquivo.
Dados inválidos (entrada incorreta)	Arquivo voos.txt com outros voos	Mensagem de erro de entrada inválida	Caso os dados sejam inseridos de forma incorreta, a função deve mostrar erro.
Arquivo não encontrado	Arquivo voos.txt não existe	Erro ao criar o arquivo.	Caso o arquivo não exista ou ocorra um erro ao tentar abrir, a função mostra erro.

**Relatório de Execução dos Testes**

Teste	Resultado Esperado	Resultado Obtido
Cadastrar voo válido	A função deve cadastrar o voo corretamente no arquivo.	Aprovado. O voo foi adicionado com sucesso no arquivo.
Entrada inválida	Mensagem de erro quando dados inválidos são fornecidos.	Aprovado. Mensagem de erro foi exibida corretamente.
Arquivo não encontrado	Mensagem de erro quando não há arquivo para escrever.	Aprovado. Mensagem de erro exibida corretamente.



**Função:** editarVoo

**Objetivos:**

1. Editar os dados de um voo existente no arquivo voos.txt.
2. Verificar se o voo existe e permitir que o usuário altere os dados do voo corretamente.
3. Garantir que as edições sejam salvas no arquivo e o voo seja atualizado.

**Entradas e Saídas Esperadas**

Entrada	Estado do Arquivo	Saída Esperada	Descrição
Edição de voo existente	Arquivo voos.txt com voos válidos	Voo atualizado!	Dados do voo especificado são atualizados corretamente.
Voo não encontrado para edição	Arquivo voos.txt com voos válidos	Código de voo não encontrado!	Caso o voo não seja encontrado, a função exibe uma mensagem de erro.
Arquivo vazio	Arquivo voos.txt vazio	Erro ao abrir o arquivo!	Caso o arquivo não exista ou esteja vazio, uma mensagem de erro é exibida.

**Relatório de Execução dos Testes**

Teste	Resultado Esperado	Resultado Obtido
Editar voo existente	A função deve editar o voo corretamente no arquivo.	Aprovado. O voo foi atualizado com sucesso.
Voo não encontrado	A função deve exibir erro "Código de voo não encontrado".	Aprovado. A mensagem de erro foi exibida corretamente.
Arquivo vazio	Exibição de erro caso o arquivo esteja vazio.	Aprovado. A mensagem de erro foi corretamente exibida.

- **Função:** **excluirVoo**

**Objetivos:**

1. Excluir os dados de um voo do arquivo voos.txt.
2. Verificar se o voo existe, identificar o código fornecido e garantir que o voo seja excluído corretamente do arquivo.

Entrada	Estado do Arquivo	Saída Esperada	Descrição
Voo existente para exclusão	Arquivo com vários voos cadastrados	Voo excluído!	Voo com o código fornecido é excluído e o arquivo é atualizado.
Voo não encontrado	Arquivo com vários voos cadastrados	Código de voo não encontrado!	Se o voo com o código fornecido não for encontrado, exibe mensagem de erro.
Arquivo vazio	Arquivo vazio ou inacessível	Erro ao abrir o arquivo!	Se não for possível abrir o arquivo, exibe mensagem de erro.

Teste	Resultado Esperado	Resultado Obtido
Excluir voo existente	O voo deve ser excluído corretamente e o arquivo atualizado.	Aprovado. O voo foi excluído e o arquivo foi modificado.
Voo não encontrado	A função deve exibir "Código de voo não encontrado".	Aprovado. A mensagem foi exibida corretamente.
Arquivo vazio	Exibição de erro ao tentar abrir o arquivo vazio ou inacessível.	Aprovado. A mensagem de erro foi corretamente exibida.

**Função:** **listarVoos**

**Objetivos:**

1. Listar todos os voos presentes no arquivo voos.txt.
2. Verificar se os dados dos voos são lidos corretamente e apresentados para o usuário.

**Entradas e Saídas Esperadas**

<b>Entrada</b>	<b>Estado do Arquivo</b>	<b>Saída Esperada</b>	<b>Descrição</b>
Solicitação para listar voos	Arquivo voos.txt com dados válidos	Exibição de todos os voos registrados no arquivo.	Todos os voos são corretamente exibidos na tela.
Arquivo vazio	Arquivo voos.txt vazio	Erro ao abrir o arquivo.	Caso o arquivo esteja vazio ou inacessível, a função exibe a mensagem de erro.
Arquivo não encontrado	Arquivo voos.txt não existe	Erro ao abrir o arquivo.	Se o arquivo não existir, uma mensagem de erro é exibida.

**Relatório de Execução dos Testes**

<b>Teste</b>	<b>Resultado Esperado</b>	<b>Resultado Obtido</b>
Listar voos existentes	Todos os voos cadastrados devem ser exibidos.	Aprovado. Todos os voos foram corretamente listados.
Arquivo vazio	Mensagem de erro "Erro ao abrir o arquivo" quando não há voos.	Aprovado. A mensagem foi exibida corretamente.
Arquivo não encontrado	Mensagem de erro quando o arquivo não é encontrado.	Aprovado. A função corretamente exibe erro.

- **Função:** **verificaAssentoLivre**

**Objetivo:** Determinar se um assento específico está livre em um voo.

#### Entradas e Saídas Esperadas

Cenário	Entrada (Código)	Estado do Arquivo	Saída Esperada	Descrição
Assento livre	(101, 1)	Contém um assento no voo 101 com número 1 e status "livre"	1	Assento está disponível.
Assento ocupado	(102, 2)	Contém um assento no voo 102 com número 2 e status "ocupado"	0	Assento está ocupado.
Assento não encontrado	(103, 3)	Arquivo contém voos e assentos, mas não o assento no voo 103	0	Assento ou voo não existe.
Voo não existente	(104, 4)	Arquivo contém outros voos, mas não o 104	0	Voo não encontrado.
Arquivo vazio	(105, 5)	Arquivo está presente, mas vazio	0	Nenhum dado encontrado.
Arquivo não existente	(106, 6)	Arquivo assentos.txt não está presente	0	Arquivo não encontrado.
Status com espaços adicionais	(107, 7)	Contém status "livre " com espaço extra	1	Função deve tratar espaços corretamente e retornar disponível.

#### Relatório de Execução dos Testes

Teste	Resultado	Notas
Assento livre	Aprovado	Assento identificado corretamente como livre.
Assento ocupado	Aprovado	Função retornou corretamente como ocupado.
Assento não encontrado	Aprovado	Função identificou ausência do assento.
Voo não existente	Aprovado	Função tratou corretamente a ausência do voo.
Arquivo vazio	Aprovado	Função retornou 0 devido à ausência de dados.
Arquivo não existente	Aprovado	Função tratou ausência do arquivo.
Status com espaços adicionais	Aprovado	Função lidou corretamente com o formato do status.

**Função:** cadastroDeAssento

**Objetivos:**

1. Cadastrar os assentos no arquivo assentos.txt.
2. Verificar se o voo existe e garantir que o assento seja corretamente adicionado ao arquivo.

**Entradas e Saídas Esperadas**

Entrada	Estado do Arquivo	Saída Esperada	Descrição
Cadastro de assento válido	Arquivo assentos.txt com outros assentos	Assento cadastrado e disponível!	Um novo assento é adicionado ao arquivo e marcado como "livre".
Código de voo não existe	Arquivo assentos.txt vazio	Erro ao tentar cadastrar o assento	Caso o código de voo não exista, a função repetirá a solicitação até encontrar um código válido.
Arquivo não encontrado	Arquivo assentos.txt não existe	Erro ao abrir arquivo 'assentos.txt'!	Se o arquivo não for encontrado, exibe a mensagem de erro.

**Relatório de Execução dos Testes**

Teste	Resultado Esperado	Resultado Obtido
Cadastro válido de assento	O assento deve ser adicionado com sucesso ao arquivo.	Aprovado. O assento foi cadastrado corretamente.
Código de voo não existente	A função deve garantir que o código de voo fornecido seja válido.	Aprovado. O código de voo é verificado corretamente antes de prosseguir.
Arquivo não encontrado	Exibição de erro ao tentar abrir o arquivo.	Aprovado. A mensagem de erro foi corretamente exibida.

- **Função:** **listarAssentos**

**Objetivos:**

1. Listar os assentos disponíveis e ocupados de um voo no arquivo assentos.txt.
2. Verificar a disponibilidade do assento e exibir as informações corretamente.

Entradas e Saídas Esperadas

Entrada	Estado do Arquivo	Saída Esperada	Descrição
Listagem de assentos válidos	Arquivo com assentos cadastrados	Exibição de todos os assentos e seus status (livre/ocupado)	Exibe todos os assentos do voo com seus respectivos status.
Voo sem assentos cadastrados	Arquivo sem assentos cadastrados	"Nenhum assento cadastrado nesse voo."	Se não houver assentos cadastrados para o voo, exibe a mensagem de erro.
Arquivo não encontrado	Arquivo assentos.txt não existe	"Erro ao abrir arquivo 'assentos.txt'!"	Se o arquivo não for encontrado, exibe a mensagem de erro.

Relatório de Execução dos Testes

Teste	Resultado Esperado	Resultado Obtido
Listar assentos válidos	Todos os assentos devem ser corretamente listados.	Aprovado. Todos os assentos foram listados corretamente.
Voo sem assentos	Exibição de mensagem "Nenhum assento cadastrado nesse voo".	Aprovado. A mensagem foi exibida corretamente.
Arquivo não encontrado	Exibição de erro caso o arquivo não seja encontrado.	Aprovado. A função corretamente exibe erro.

- **Função:** **excluirAssento**

**Objetivos:**

1. Excluir um assento do voo no arquivo assentos.txt.
2. Verificar se o assento existe e garantir que o arquivo seja atualizado corretamente.

Entradas e Saídas Esperadas

Entrada	Estado do Arquivo	Saída Esperada	Descrição
Exclusão de assento válido	Arquivo com assentos cadastrados	"Assento excluído!"	Um assento é removido do arquivo e o arquivo é atualizado.
Assento não encontrado	Arquivo com assentos cadastrados	"Assento não cadastrado."	Se o assento não for encontrado, exibe a mensagem de erro.
Arquivo vazio	Arquivo vazio ou inacessível	"Erro ao abrir arquivo!"	Exibe a mensagem de erro se o arquivo não puder ser acessado.

Relatório de Execução dos Testes

Teste	Resultado Esperado	Resultado Obtido
Excluir assento válido	O assento deve ser excluído corretamente do arquivo.	Aprovado. O assento foi excluído e o arquivo atualizado.
Assento não encontrado	A função deve exibir erro "Assento não cadastrado".	Aprovado. A mensagem foi exibida corretamente.
Arquivo vazio	Exibição de erro caso o arquivo não possa ser acessado.	Aprovado. A função corretamente exibe erro.

- **Função:** **reservarAssento**

**Objetivos:**

1. Reservar um assento para um passageiro no arquivo assentos.txt.
2. Verificar se o assento é livre e se os dados do passageiro são válidos.

Entradas e Saídas Esperadas

Entrada	Estado do Arquivo	Saída Esperada	Descrição
Reserva de assento disponível	Arquivo com assentos livres	"Assento reservado!"	Um assento é marcado como "ocupado" e o nome do passageiro é registrado.
Assento já ocupado	Arquivo com assentos ocupados	"Assento já ocupado!"	Se o assento já estiver ocupado, exibe mensagem de erro.
Arquivo não encontrado	Arquivo assentos.txt não existe	"Erro ao abrir arquivo!"	Se o arquivo não for encontrado, exibe a mensagem de erro.

Relatório de Execução dos Testes

Teste	Resultado Esperado	Resultado Obtido
Reservar assento válido	O assento deve ser reservado corretamente.	Aprovado. O assento foi reservado e o arquivo foi atualizado.
Assento já ocupado	Exibição de erro se o assento já estiver ocupado.	Aprovado. A mensagem foi exibida corretamente.
Arquivo não encontrado	Exibição de erro caso o arquivo não seja encontrado.	Aprovado. A função corretamente exibe erro.



- **Função:** cadastroDePassageiros

**Objetivo:** Cadastrar um novo passageiro, garantindo que o código seja único e que os dados sejam corretamente gravados no arquivo passageiros.txt.

#### Entradas e Saídas Esperadas

Cenário	Entrada (Código)	Estado do Arquivo	Saída Esperada	Descrição
Cadastro válido	Código: 101, Nome: "João", Endereço: "Rua A", Telefone: "123456789", Fidelidade: "sim", Pontos: 100	Arquivo vazio ou sem o código 101	Sucesso, gravação dos dados no arquivo.	Dados válidos, código único.
Código duplicado	Código: 101, repetido	Arquivo contém um passageiro com o código 101	Requer outro código, sucesso após novo.	Validação de unicidade do código.
Fidelidade "não"	Fidelidade: "nao", sem entrada para pontos de fidelidade	Arquivo vazio ou sem o código correspondente	Sucesso, pontos = 0.	Fidelidade configurada para "não".
Arquivo inexistente	Dados válidos	Arquivo passageiros.txt não existe	Criação do arquivo e gravação dos dados.	Arquivo é criado quando não existe.
Entrada com caracteres especiais	Nome: "Ana Clara", Endereço: "Rua 123, Apt. 4", Telefone: "(99) 99999-9999"	Arquivo vazio ou sem conflito	Sucesso, gravação no formato correto.	Dados contendo caracteres especiais ou espaços são tratados.

#### Relatório de Execução dos Testes

Teste	Resultado	Notas
Cadastro válido	Aprovado	Dados gravados corretamente.
Código duplicado	Aprovado	Código rejeitado, solicitou novo código.
Fidelidade "não"	Aprovado	Pontos configurados corretamente para 0.
Arquivo inexistente	Aprovado	Arquivo criado e dados gravados corretamente.
Entrada com caracteres especiais	Aprovado	Dados gravados no formato correto, incluindo caracteres.

- **Função:** **listarPassageiros**

**Objetivo:** Ler os dados de passageiros de um arquivo e exibi-los no console. Ela realiza a leitura de um arquivo passageiros.txt e imprime as informações de cada passageiro cadastrados nesse arquivo.

#### Entradas e Saídas Esperadas

Cenário	Estado do Arquivo	Saída Esperada	Descrição
Arquivo com dados válidos	Arquivo passageiros.txt contém dados válidos de passageiros	Exibição correta no console dos dados dos passageiros	Função exibe os dados dos passageiros.
Arquivo vazio	Arquivo passageiros.txt está vazio	Mensagem de erro ou nenhuma saída	Arquivo vazio, sem passageiros cadastrados.
Arquivo inexistente	Arquivo passageiros.txt não existe	Mensagem de erro "Erro ao abrir o arquivo de passageiros"	Arquivo não encontrado.
Formato de dados incorreto no arquivo	Arquivo passageiros.txt contém dados com formato incorreto	Mensagem de erro ao tentar ler o arquivo (EOF ou dados inválidos)	Dados no arquivo mal formatados.

#### Relatório de Execução dos Testes

Teste	Resultado	Notas
Arquivo com dados válidos	Aprovado	Dados exibidos corretamente.
Arquivo vazio	Aprovado	Mensagem de erro "Erro ao abrir o arquivo" exibida.
Arquivo inexistente	Aprovado	Mensagem de erro "Erro ao abrir o arquivo" exibida.
Formato de dados incorreto no arquivo	Aprovado	Função falha silenciosamente se os dados forem inválidos (EOF ou formato inválido).

- **Função:** editarPassageiro

**Objetivos:**

1. Editar os dados de um passageiro no arquivo passageiros.txt.
2. Verificar se a função encontra e edita os dados do passageiro corretamente.
3. Garantir que os dados sejam escritos corretamente no arquivo após a edição.
4. Validar o comportamento da função quando o passageiro não é encontrado.

Entradas e Saídas Esperadas

Cenário	Entrada (Código)	Estado do Arquivo	Saída Esperada	Descrição
Passageiro encontrado e editado	Código do passageiro: 101, Novos dados: Nome, Endereço, Telefone	Arquivo passageiros.txt com dados de passageiro 101	Mensagem de sucesso "Passageiro editado com sucesso!" e dados editados no arquivo.	Dados do passageiro com código 101 são editados.
Passageiro não encontrado	Código de passageiro inexistente: 999	Arquivo passageiros.txt com dados de outros passageiros	Mensagem de erro "Passageiro não encontrado."	Caso em que o código informado não existe no arquivo.
Arquivo vazio	Arquivo passageiros.txt vazio	Arquivo vazio	Mensagem de erro ao tentar abrir o arquivo.	Arquivo vazio, não há passageiros para editar.

Relatório de Execução dos Testes

Teste	Resultado	Notas
Editar passageiro existente	Aprovado	Dados alterados corretamente.
Passageiro não encontrado	Aprovado	Mensagem "Passageiro não encontrado" exibida.
Arquivo vazio	Aprovado	Mensagem de erro exibida para arquivo vazio.

**Função:** **excluirPassageiro****Objetivos:**

1. Excluir os dados de um passageiro do arquivo passageiros.txt.
2. Verificar se o passageiro existe, comparar o código e o nome fornecido para garantir que é o passageiro correto a ser excluído.
3. Garantir que os dados sejam excluídos corretamente e que o arquivo seja atualizado sem o passageiro excluído.
4. Validar o comportamento da função quando o passageiro não é encontrado.

Entrada	Estado do Arquivo	Saída Esperada	Descrição
Passagem de dados de passageiro válido	Arquivo com dados de vários passageiros	Passageiro <nome> (<codigo>) excluído com sucesso!	Dados de um passageiro (com código e nome corretos) são excluídos e os outros permanecem no arquivo.
Passagem de dados de passageiro inválido	Arquivo com dados de vários passageiros	Passageiro não encontrado.	Caso o código e nome não coincidam com os dados no arquivo, a mensagem de erro é exibida.
Arquivo vazio	Arquivo vazio	Erro ao abrir o arquivo!	Caso o arquivo esteja vazio ou não possa ser acessado, a função mostra a mensagem de erro e não realiza nenhuma alteração.

Teste	Resultado	Notas
Excluir passageiro existente	Aprovado	Arquivo alterado corretamente e a mensagem de sucesso foi exibida.
Passageiro não encontrado	Aprovado	A mensagem "Passageiro não encontrado" foi corretamente exibida quando não encontrado.
Arquivo vazio	Aprovado	Mensagem de erro exibida quando o arquivo estava vazio ou não pôde ser aberto.

- Função: **buscarPassageiro**

Elemento	Descrição
<b>Objetivo</b>	Permitir que o usuário escolha como buscar o passageiro: por ID ou por nome.
<b>Entrada</b>	Nenhuma entrada direta. O usuário escolhe entre buscar por ID ou por nome.
<b>Processo</b>	1. O usuário escolhe entre buscar por ID ou por nome.
	2. Dependendo da escolha, chama a função buscaPassageiroID() ou buscaPassageiroNome().
<b>Saída</b>	Chama a função de busca correspondente (por ID ou por nome).
<b>Mensagem de erro</b>	Caso o usuário forneça uma opção inválida, o processo se repete até que uma opção válida seja escolhida.

- Função: **buscaPassageiroID**

Elemento	Descrição
<b>Objetivo</b>	Buscar um passageiro pelo código (ID).
<b>Entrada</b>	Código do passageiro (ID) a ser buscado.
<b>Processo</b>	1. Abre o arquivo passageiros.txt para leitura.
	2. Lê cada linha do arquivo e compara o código fornecido com o código dos passageiros no arquivo.
	3. Se o código for encontrado, exibe o código e o nome do passageiro.
	4. Caso o código não seja encontrado, exibe uma mensagem de erro.
<b>Saída</b>	Exibe o código e nome do passageiro se encontrado ou uma mensagem de erro se não encontrado.
<b>Mensagem de erro</b>	Exibe "Passageiro não encontrado!" caso o código não seja encontrado.

- Função: **buscaPassageiroNome**

Elemento	Descrição
<b>Objetivo</b>	Buscar um passageiro pelo nome.
<b>Entrada</b>	Nome do passageiro a ser buscado.
<b>Processo</b>	<ol style="list-style-type: none"> <li>1. Solicita o nome do passageiro e o armazena.</li> <li>2. Abre o arquivo passageiros.txt para leitura.</li> <li>3. Lê cada linha do arquivo e compara o nome fornecido com o nome dos passageiros no arquivo.</li> <li>4. Se o nome for encontrado, exibe o código e nome do passageiro.</li> <li>5. Caso o nome não seja encontrado, exibe uma mensagem de erro.</li> </ol>
<b>Saída</b>	Exibe o código e nome do passageiro se encontrado ou uma mensagem de erro se não encontrado.
<b>Mensagem de erro</b>	Exibe "Passageiro não encontrado!" caso o nome não seja encontrado.

- Função: **buscarPontosFidelidade**

Entradas e Saídas Esperadas

Entrada	Estado do Arquivo	Saída Esperada	Descrição
Código de passageiro válido	Arquivo passageiros.txt com dados	Nome do passageiro e seus pontos de fidelidade	Exibe o nome do passageiro e os pontos de fidelidade se o código for encontrado no arquivo.
Código de passageiro não encontrado	Arquivo passageiros.txt com dados	Nenhum passageiro encontrado com o código X.	Exibe mensagem de erro se o código do passageiro não for encontrado no arquivo.
Arquivo não encontrado	Arquivo passageiros.txt não existe	Erro ao abrir o arquivo.	Exibe mensagem de erro se o arquivo passageiros.txt não puder ser aberto.

## Relatório de Execução dos Testes

Teste	Resultado Esperado	Resultado Obtido
Buscar pontos de fidelidade de código válido	O nome e os pontos de fidelidade do passageiro devem ser exibidos.	Aprovado. O nome e os pontos de fidelidade foram exibidos corretamente.
Buscar pontos de fidelidade de código inválido	Exibição de mensagem "Nenhum passageiro encontrado com o código X."	Aprovado. A mensagem de erro foi exibida corretamente quando o código não foi encontrado.
Arquivo não encontrado	Exibição de mensagem "Erro ao abrir o arquivo."	Aprovado. A mensagem de erro foi exibida corretamente quando o arquivo não existe.

- **Função:** cadastroDeMembroTripulacao

## Entradas e Saídas Esperadas

Entrada	Estado do Sistema	Saída Esperada	Descrição
Nome, telefone e cargo válidos	Lista de tripulantes previamente existente	Membro cadastrado com sucesso! Código: X	Cadastra o membro com as informações fornecidas e exibe o código gerado.
Nome, telefone ou cargo inválido	Lista de tripulantes previamente existente	Erro na validação ou re-solicitação de dados	Solicita novamente ou exibe erro se as informações fornecidas não forem válidas.
Falha ao salvar no arquivo	Arquivo indisponível para escrita	Erro ao salvar dados no arquivo.	Exibe mensagem de erro se houver falha ao salvar os dados no arquivo.

## Relatório de Execução dos Testes

Teste	Resultado Esperado	Resultado Obtido
Cadastrar membro com dados válidos	Mensagem de sucesso e salvamento no arquivo.	Aprovado. O membro foi cadastrado com sucesso, e o arquivo foi atualizado.
Fornecer dados inválidos	Solicitação ou exibição de erro para dados inválidos.	Aprovado. O sistema solicitou novamente as informações em caso de erro.
Falha ao salvar arquivo	Exibição de mensagem "Erro ao salvar dados no arquivo."	Aprovado. A mensagem de erro foi exibida corretamente ao simular falha de salvamento.

- **Função:** `listarMembroTripulacao`

**Objetivo:** Exibir uma lista completa de todos os membros da tripulação cadastrados no sistema.

#### Entradas e Saídas Esperadas

Entrada	Estado do Sistema	Saída Esperada	Descrição
Nenhum membro	Lista de tripulantes vazia	"Nenhum membro cadastrado."	Exibe mensagem indicando que não há tripulantes cadastrados.
Membros cadastrados	Lista de tripulantes com dados	Detalhes dos tripulantes: Código, Nome, Telefone, Cargo.	Lista todos os membros cadastrados com suas respectivas informações.

#### Relatório de Execução dos Testes

Teste	Resultado Esperado	Resultado Obtido
Lista vazia	Exibição de "Nenhum membro cadastrado."	Aprovado. A mensagem foi exibida corretamente.
Lista com membros cadastrados	Listagem de detalhes dos tripulantes.	Aprovado. Todos os membros foram exibidos corretamente.

- **Função:** `editarMembroTripulacao`

**Objetivo:** Permitir a atualização das informações de um membro da tripulação já cadastrado.

#### Entradas e Saídas Esperadas

Entrada	Estado do Sistema	Saída Esperada	Descrição
Código existente	Lista com o código buscado presente	"Membro atualizado com sucesso!"	Permite editar o nome, telefone e cargo do tripulante encontrado.
Código inexistente	Lista sem o código buscado	"Codigo nao encontrado!"	Exibe mensagem de erro ao não localizar o código informado.

#### Relatório de Execução dos Testes

Teste	Resultado Esperado	Resultado Obtido
Código existente	Exibição de "Membro atualizado com sucesso!"	Aprovado. As informações foram atualizadas corretamente.
Código inexistente	Exibição de "Codigo nao encontrado!"	Aprovado. A mensagem de erro foi exibida corretamente.



- **Função:** `excluirMembroTripulacao`

**Objetivo:** Remover um membro da tripulação do sistema.

Entradas e Saídas Esperadas

Entrada	Estado do Sistema	Saída Esperada	Descrição
Código existente	Lista com o código buscado presente	"Membro excluído com sucesso!"	Remove o tripulante com o código fornecido e ajusta a lista.
Código inexistente	Lista sem o código buscado	"Codigo nao encontrado!"	Exibe mensagem de erro ao não localizar o código informado.

Relatório de Execução dos Testes

Teste	Resultado Esperado	Resultado Obtido
Código existente	Exibição de "Membro excluído com sucesso!"	Aprovado. O membro foi excluído e a lista ajustada.
Código inexistente	Exibição de "Codigo nao encontrado!"	Aprovado. A mensagem de erro foi exibida corretamente.

## **Lista de assinaturas das funções e parâmetros**

Explicação da estrutura de dados principal do programa.

Apresentação das assinaturas das funções:

### **int codigoPassageiroJaExiste(int codigo)**

Função para verificar se o código de um passageiro já existe no sistema. Recebe como parâmetro o código do passageiro e retorna 1 se já existir e 0 caso contrário.

### **int passageiroExiste(char nomePassageiro[])**

Função para verificar se o nome de um passageiro já existe no sistema. Recebe como parâmetro o nome do passageiro e retorna 1 se já existir e 0 caso contrário.

### **int codigoVooExiste(int codigo)**

Função para verificar se o código de um voo já existe no sistema. Recebe como parâmetro o código do voo e retorna 1 se já existir e 0 caso contrário.

### **int verificaAssentoLivre(int codigo, int numeroAssento)**

Função para verificar se um assento está livre para reserva. Recebe como parâmetros o código do voo e o número do assento e retorna 1 se o assento estiver livre, caso contrário retorna 0.

### **void cadastroDePassageiros()**

Função para cadastrar um novo passageiro no sistema. Recebe as informações do passageiro e as salva no arquivo.

### **void listarPassageiros()**

Função para listar todos os passageiros cadastrados no sistema. Exibe as informações de todos os passageiros no formato desejado.

### **void editarPassageiro()**

Função para editar os dados de um passageiro já cadastrado. Recebe o código do passageiro e as novas informações para atualização.

### **void excluirPassageiro()**

Função para excluir um passageiro do sistema. Recebe o código do passageiro e remove os dados do registro, atualizando o arquivo.

### **void buscarPassageiro()**

Função para buscar um passageiro no sistema. Recebe o nome ou código do passageiro e exibe as informações encontradas.

**void buscaPassageiroID()**

Função para buscar um passageiro pelo seu ID (código). Recebe o código do passageiro e exibe as informações correspondentes.

**void buscaPassageiroNome()**

Função para buscar um passageiro pelo nome. Recebe o nome do passageiro e exibe as informações correspondentes.

**void cadastroDeVoo()**

Função para cadastrar um novo voo no sistema. Recebe as informações do voo, como origem, destino, data e hora, e as salva no arquivo.

**int tripulanteExiste(int codigo)**

Função para verificar se um tripulante está cadastrado no sistema. Recebe o código do tripulante e retorna 1 se existir e 0 caso contrário.

**void listarVoos()**

Função para listar todos os voos cadastrados no sistema. Exibe as informações de todos os voos disponíveis.

**void editarVoo()**

Função para editar os dados de um voo. Recebe o código do voo e as novas informações para atualizar o voo no sistema.

**void excluirVoo()**

Função para excluir um voo do sistema. Recebe o código do voo e remove o voo do registro, atualizando o arquivo.

**void cadastroDeAssento()**

Função para cadastrar um assento no sistema. Recebe informações sobre o voo e o assento, e os salva no arquivo de assentos.

**void listarAssentos()**

Função para listar todos os assentos disponíveis e ocupados. Exibe os assentos de cada voo e sua respectiva situação (livre ou reservado).

**void excluirAssento()**

Função para excluir um assento do sistema. Recebe o número do assento e remove a reserva ou o assento do registro.

**void reservarAssento()**

Função para reservar um assento para um passageiro. Recebe o código do voo e o número do assento, marcando-o como reservado para o passageiro.

**void baixaReservaAssento()**

Função para cancelar uma reserva de assento. Recebe o código do voo e o número do assento, liberando a reserva e marcando o assento como livre.

**void salvarArquivo()**

Função para salvar os dados no arquivo. A função percorre os registros e grava todos os dados no arquivo correspondente, persistindo as informações.

**void carregarArquivo()**

Função para carregar os dados do arquivo. A função lê os registros salvos e carrega as informações no sistema.

**int obterProximoid()**

Função para obter o próximo ID disponível para cadastro. A função percorre os registros e retorna o próximo código disponível.

**void cadastroDeMembroTripulacao()**

Função para cadastrar um novo membro da tripulação. Recebe as informações do membro (como nome, cargo, etc.) e as salva no sistema.

**void listarMembroTripulacao()**

Função para listar todos os membros da tripulação. Exibe as informações de cada membro, como código, nome, cargo, etc.

**void editarMembroTripulacao()**

Função para editar os dados de um membro da tripulação. Recebe o código do membro e as novas informações para atualizar os dados.

**void excluirMembroTripulacao()**

Função para excluir um membro da tripulação. Recebe o código do membro e remove os dados do registro da tripulação.

**void listarVoosPassageiro()**

Função para listar todos os voos de um passageiro específico. Recebe o código ou nome do passageiro e exibe os voos relacionados a ele.

**void buscarPontosFidelidade(int codigoBuscado)**

Função para buscar os pontos de fidelidade de um passageiro. Recebe o código do passageiro e exibe o número de pontos acumulados.