

Especificação do Projeto 1b: Controle de Tráfego Aéreo

versão 1.0

1 Introdução

Este projeto tem como objetivo desenvolver um sistema de *controle de tráfego aéreo* utilizando uma lista de prioridades implementada como um **heap máximo**. A estrutura *heap* será utilizada para gerenciar as prioridades das aeronaves de acordo com critérios específicos, permitindo a rápida identificação e remoção da aeronave de maior prioridade. Além disso, o sistema permitirá a atualização das prioridades das aeronaves, possibilitando ajustes dinâmicos conforme novas informações sejam inseridas.

2 Objetivos

O sistema deve:

- carregar informações das aeronaves a partir de um arquivo `.csv`.
- permitir a inserção manual de novas aeronaves pelo usuário.
- calcular a prioridade de cada aeronave com base em critérios definidos.
- gerenciar as aeronaves em uma lista de prioridades, garantindo que a aeronave com maior prioridade esteja sempre acessível.
- implementar as operações básicas de um *heap*: inserção, remoção, visualização e atualização de prioridade.

3 Especificações Técnicas

3.1 Estrutura dos Dados

Cada aeronave será representada por uma estrutura (**struct**) contendo:

- **identificador**: identificador único da aeronave (ex.: “TPR123”).
- **combustivel**: quantidade de combustível restante em litros.
- **horario**: horário previsto de chegada ou partida (em minutos do dia).
- **tipo**: tipo de operação (0 para decolagem, 1 para pouso).
- **emergencia**: indica se a aeronave está em situação de emergência (0 para não, 1 para sim).
- **prioridade**: prioridade calculada com base nos dados da aeronave.

3.2 Cálculo da Prioridade

A prioridade será calculada pela fórmula:

$$\text{Prioridade} = (1000 - \text{combustivel}) + (1440 - \text{horario}) + 500 \times (\text{peso do tipo}) + 5000 \times (\text{peso da emergencia}),$$

onde:

- **Combustível**: quanto menor, maior a prioridade.
- **Horário**: quanto mais cedo, maior a prioridade.
- **Tipo de operação**: 0 para decolagem, 500 para pouso.
- **Emergência**: 5000 para emergências, 0 caso contrário.

3.2.1 Exemplo de Cálculo da Prioridade

Considere os seguintes dados de uma aeronave:

- **Identificador**: TPR123
- **Tipo de operação (T)**: pouso (1)
- **Combustível restante (C)**: 300 litros

- **Horário previsto (H):** 540 minutos (9h00)
- **Situação de emergência (E):** sem emergência (0)

A prioridade será calculada com a fórmula:

$$\text{Prioridade} = (1000 - C) + (1440 - H) + 500 \times T + 5000 \times E$$

Substituindo os valores:

$$\text{Prioridade} = (1000 - 300) + (1440 - 540) + 500 \times 1 + 5000 \times 0$$

$$\text{Prioridade} = 700 + 900 + 500 + 0$$

$$\text{Prioridade} = 2100$$

Neste exemplo, a aeronave TPR123 terá uma prioridade de 2100, sendo mais prioritária em comparação a aeronaves com valores menores de prioridade.

3.3 Operações do Sistema

O sistema deve implementar as seguintes funcionalidades:

- **Carregar aeronaves:** ler os dados de um arquivo `.csv` e construir o *heap* inicial.
- **Inserir aeronave:** adicionar uma nova aeronave ao *heap* e ajustar a estrutura para manter a propriedade de *heap*.
- **Consultar maior prioridade:** exibir a aeronave com maior prioridade (raiz do *heap*).
- **Remover maior prioridade:** remover a aeronave de maior prioridade e ajustar a estrutura.
- **Atualizar prioridade:** recalculer a prioridade de uma aeronave já inserida e reorganizar o *heap*.
- **Exibir *heap*:** mostrar todas as aeronaves na ordem atual do *heap*.

3.4 Exemplo de Arquivo `.csv`

```
identificador,combustivel,horario,tipo,emergencia
FL123,200,600,1,0
AB456,50,800,2,1
CD789,300,500,1,0
```

3.5 Processo de Atualização de Prioridade

A prioridade das aeronaves pode ser alterada ao longo do tempo devido a mudanças em parâmetros como o combustível, horário de chegada ou a situação de emergência. Para garantir que a lista de prioridades seja sempre correta após a atualização, será necessário:

- Localizar a aeronave no *heap* (identificada por seu `identificador`).
- Recalcular sua prioridade com base nos novos dados.
- Reorganizar o *heap*, mantendo a propriedade de *heap*.

Para a reorganização, será utilizado o algoritmo de `heapify`, que garante que a estrutura do *heap* seja mantida após a atualização da prioridade.

4 Regras a seguir

1. Este projeto vale 5,0 pontos.
2. O projeto poderá ser implementado em dupla.
3. Os nomes dos participantes deverão ser enviados via tarefa a ser informada.
4. Idem caso o projeto for implementado por apenas um aluno.
5. A data limite para envio do projeto será 30/11/24.
6. A forma de envio do projeto será via tarefa a ser informada.
7. O sorteio dos dias de apresentação será realizado no dia 01/12/24 e seu resultado divulgado no Sigaa.
8. Apenas um dos participantes ficará responsável por enviar o projeto compactado.
9. Este documento está sujeito a alterações e qualquer atualização será informada pelo Sigaa.