Trabalho prático 1: Sistema de gerenciamento de aeroportos xAero

Leandro Lázaro (3513), Mateus Pinto (3489), Vinícius Júlio (3495)

Ciência da Computação – Universidade Federal de Viçosa – Campus Florestal (UFV- Caf) – Florestal – MG – Brasil

{leandro.lazaro, mateus.p.silva, vinicius.julio}@ufv.br

**Resumo.** Criamos um sistema de gerenciamento de aeroportos batizado de xAero usando, como os Tipos Abstratos de Dados principais, o TAD voo, lista de voos, item matriz e a matriz de voos. Como secundário, foi usado o TAD horário. Ao fim, criamos um programa Main para gerenciar os TADs.

# 1. Introdução

Neste trabalho implementaremos um sistema de gerenciamento completo de aeroporto com a principais funções: criar novo voo, retirar novo voo, verificar voos diários, etc. Todo o sistema será feito em linguagem C através de Tipos abstratos de dados principais e auxiliares, e criaremos uma interface agradável via terminal para o usuário final do sistema.

**2. Nomenclatura Utilizada**

Para uma melhor organização e compreensão do código, adotamos a seguinte nomenclatura em relação aos TADs Principais:

* TADs Principais: “NOME\_DO\_TAD.h” e “NOME\_DO\_TAD.c”
* Tipos (Estruturas, Listas, Matrizes): “NOME\_DO\_TIPO”
* Funções: “NOME\_DO\_TAD\_nomeFuncao”

E em relação aos TADs Auxiliares:

* TADs Auxiliares: “nome\_do\_tad.h” e “nome\_do\_tad.c”
* Tipos (Estruturas, Listas, Matrizes): “nome\_do\_tipo”
* Funções: “nome\_do\_tad\_nomeFuncao”

**3. Tipos Abstratos de Dados**

Tipos Abstratos de Dados utilizados no sistema. Para melhor organização, os TADs Auxiliares e Principais foram separados em duas diferentes pastas: “TAD\_AUXILIAR/” e “TAD/”.

**3.1 Tipos Abstratos de Dados Auxiliares**

Durante o desenvolvimento, notamos a necessidade de TADs auxiliares para melhor organizar os dados. Neste tópico, explicaremos o motivo da implementação de cada um deles e as curiosidades de cada dado dentro das e struturas.

## 3.1.1. Horário

Como era muito recorrente a necessidade de definir o horário de acontecimentos, criamos o TAD horario.

Seu header se encontra em “TAD\_AUXILIAR/horario.h” e define o tipo “horario” que é composto pelas variáveis “hora” e “min”, que são declaradas como unsigned int uma vez que horas são números naturais (inteiros e positivos), além de definir os protótipos das funções a serem executadas.

O source se encontra em “TAD\_AUXILIAR/horario.c” e implementa as funções de inicialização do horário, verificação de os horários estarem em ordem crescente, estabelece o horário no TAD horario, e mostrar o horário.

**3.1.2. Lista Encadeada “Random”**

Os Identificadores de Voo (VIDs), por serem gerados aleatoriamente, são suscetíveis à repetição, e para evitar que isso aconteça, foi criado o TAD lista\_encadeada\_random, que verifica se o VID gerado já é existente comparando-o com VIDs já gerados, salvos em uma lista encadeada.

Seu header se encontra em “TAD\_AUXILIAR/lista\_encadeada\_random.h” e define o item “item\_lista\_encadeada\_random”, que contém o VID, a lista encadeada “lista\_encadeada\_random” e também define os protótipos das funções a serem executadas.

O source é encontrado em “TAD\_AUXILIAR/lista\_encadeada\_random.c” e implementa as funções de criar a lista, inserir o VID na lista, e remover voos.

**3.2 Tipos Abstratos de Dados Principais**

Implementação dos TADs essenciais para o funcionamento do sistema de aeroporto.

**3.2.1. Voo**

O TAD Voo contém as informações de cada voo.

O header é localizado em “TAD/VOO.h” e define o tipo VOO, que é composto pelo VID (que é um unsigned int, já que não é necessário que o código tenha sinal negativo), o horário de decolagem e de pouso, o aeroporto de decolagem e de pouso, e o identificador de pista de decolagem, que também é um unsigned int. O cabeçalho também define os protótipos de funções das operações a serem feitas no tipo VOO.

O source está localizado em “TAD/VOO.c” e contém a implementação das funções de inicializar a estrutura VOO, e funções “set”, “get” e “show” que, respectivamente, alteram, leem e mostram cada uma das informações contidas no tipo VOO.

**3.2.2. Lista de Voos**

O TAD Lista de Voos coloca os voos, definidos pelo TAD Voo, que decolaram em um determinado dia em uma lista encadeada, o que facilita a inserção e remoção de voos caso hajam cancelamentos ou mudanças de horários.

Seu header encontra-se em “TAD/LISTA\_DE\_VOOS.h” e define como item “ITEM\_LISTA\_DE\_VOOS”, que contém o item VOO, e a lista encadeada “LISTA\_DE\_VOOS”. Também define o protótipo das funções de operações a serem realizadas na lista.

O source, localizado em “TAD/LISTA\_DE\_VOOS.c” implementa as funções de criação da lista, inserção e remoção de voos, e também as de retornar e mostrar os voos pelo identificador.

**3.2.3. Item Matriz**

Os voos que partirem de um determinado aeroporto em um mesmo dia serão colocados na MATRIZ\_VOOS, para isso deverá ser criado o TAD Item Matriz contendo as listas de voos e outras informações necessárias.

O header é encontrado em “TAD/ITEM\_MATRIZ.h” e define o item “ITEM\_MATRIZ”, que contém um item LISTA\_DE\_VOOS, o número na lista e o horário da última atualização dessa lista. Também é definido o protótipo das funções que serão implementadas no TAD ITEM\_MATRIZ.

O source se encontra em “TAD/ITEM\_MATRIZ.c” e tem a implementação das funções de inicialização do ITEM\_MATRIZ, funções “set”, “get” e “show” (alterar, ler e mostrar) para cada informação contida no ITEM\_MATRIZ.

**3.2.4. Matriz Voos**