

# Documentação do Programa *TADS AEROPORTO*



## OBJETIVO DO TRABALHO

O objetivo do trabalho desde o início foi utilizar os conhecimentos teóricos de tipos abstratos de dados, lista e listas encadeadas adquiridos no início do atual semestre na disciplina de algoritmos e estrutura de dados, para criar um sistema de um aeroporto com as principais informações que geralmente aparecem em painéis de aeroportos, como número do voo, aeroporto de chegada e partida, entre outras.

## PROJETO DO SISTEMA E DECISÕES

Não iniciamos o projeto com uma divisão e tarefas muito exata mas seguindo a ordem da especificação do trabalho. Uma das primeiras decisões foi de como criar os números de identificação de voos, seriam estas series randômicas de números inteiros usando na função `rand()` para gerar números entre 0 e 1000, além de usar o tempo como semente para a mesma. As funções de horário de voo por sua vez, utilizam uma string `Char Hr[10]` para armazenar os dois pontos que separam a hora dos minutos na escrita cotidiana. A criação do restantes dos TADs requisitados na descrição do trabalho foi completamente acumulativa, isto é, cada novo TAD iniciado dependia totalmente do uso de funções presentes nos TADs anteriores, garantindo assim uma total interligação e dependência de um TAD com outro.

## MÓDULOS DESENVOLVIDOS

Inicialmente o fizemos esquemas práticos de como o programa se comportaria desde a interface vista pelo usuário até as alocações dinâmicas realizadas pelas funções de inserção e remoção de voos. Então durante as reuniões para decidir o que e como seria feito, o algoritmo foi facilmente implementado usando o número de TADs requisitado pela descrição do trabalho.

## PRINCIPAIS FUNÇÕES IMPLEMENTADAS

No Dados\_Voo:

- SetVID
- SetHora\_Decola
- SetHora\_Previsto
- SetPist
- SetAeroporto\_Partida
- SetAeroporto\_Chegada
- GetVID
- GetHora\_Decola

- GetHora\_Previsto
- GetPista
- GetAeroporto\_Partida
- GetAeroporto\_Chegada

No Lista\_Voo:

- Inicia\_Lista
- Lista\_Vazia
- Insere\_Voo
- Remove\_Voo
- Procura\_Voo
- Imprime\_Lista

No Item\_Matriz:

- Inicializa\_Item\_Matriz
- SetLista\_de\_Voos
- SetNumero\_de\_Voos
- SetUltima\_Atualizacao
- GetLista\_de\_Voos
- GetNumero\_de\_Voos
- GetUltima\_Atualizacao

No Matriz\_Voo:

- SetData
- SetTotal\_De\_Voos
- SetHora\_Ultima\_Atualizacao\_Matriz
- GetData
- GetHora\_Ultima\_Atualizacao\_Matriz
- Inicia\_Matriz
- Insere\_Voo\_Matriz
- Remove\_Voo\_Matriz
- Procurar\_Voo\_Matriz
- Imprimir\_Voos\_Decolagem\_Pouso
- Imprimir\_Voos\_Decolagem
- Imprimir\_Voos\_Pouso
- Imprimir\_Matriz
- Encontrar\_Faixa\_Voos\_Maior
- Encontrar\_Faixa\_Voos\_Menor
- Encontrar\_Lista\_Voos\_Mais\_Recente
- Encontrar\_Lista\_Voos\_Menos\_Recente
- Matriz\_Esparca

## COMPILAÇÃO E EXECUÇÃO DO PROGRAMA

Todo o desenvolvimento do algoritmo do programa foi realizado apenas com o uso de um editor de texto sincronizado com o repositório do GitHub que apenas relatava erros de sintaxe. O código foi sempre compilado pelo terminal do Linux, utilizando os compiladores GCC e CLANG. Devido a isso e a grande quantidade de arquivos (arquivos dos TADS), foi criado um arquivo makefile para facilitar e agilizar a compilação do mesmo.

**Para realizar a compilação do programa pelo terminal usando o GCC:**

(Primeiramente certifique-se que o terminal encontra-se navegado até a pasta principal do programa)

->Execute os seguintes comandos:

```
$ make  
(O makefile irá executar a compilação utilizando o GCC)  
$ ./EXEC_GCC  
(O programa irá começar a rodar)
```

OBS: O programa já se encontra compilado, então caso não ache necessário re-compilar, apenas chame-o executando apenas o ultimo comando.

### **Para realizar a compilação do programa pelo terminal usando o CLANG:**

(Primeiramente certifique-se que o terminal encontra-se navegado até a pasta principal do programa)

->Execute os seguintes comandos:

```
$ make clang  
(O makefile irá executar a compilação utilizando o CLANG)  
$ ./EXEC_CLANG  
(O programa irá começar a rodar)
```

OBS: O programa já se encontra compilado, então caso não ache necessário re-compilar, apenas chame-o executando apenas o ultimo comando.

## **UTILIZAÇÃO DO PROGRAMA**

A execução do programa é uma atividade bastante intuitiva, o programa inicialmente apresenta uma tela de boas vindas e informa 3 opções, sendo duas delas para a escolha da forma de entrada preferida e a última a porta de saída do programa(finalização do programa), tendo esta o número 9 escolhido propositalmente para evitar finalizações acidentais por parte do usuário.

Caso o usuário opte pela opção de entrada interativa(Opção 1), um segundo menu será apresentado, requisitando uma segunda escolha de operação desejada, a qual o usuário deverá decidir de acordo com o que quer realizar no programa(Algumas das opções do segundo menu ainda encaminham o usuário para um outro terceiro menu onde opções mais especificas porem do mesmo gênero da opção selecionada no segundo menu se encontram para serem escolhidas).

Caso o usuário opte pela opção de entrada por arquivo(Opção 2), uma requisição pelo nome do arquivo de entrada será realizada, então o usuário necessitará preencher o campo com o nome do mesmo que deverá estar previamente localizado na mesma pasta em que o programa.(OBS: O nome do arquivo deverá ser escrito obrigatoriamente com a extensão ".txt" no final. Ex: *Nome\_do\_Arquivo.txt*)

O programa ao fim de uma execução de alguma função independente de gênero ou tipo, deverá sempre retornar a tela de boas vindas inicial, a fim de prosseguir com a próxima tarefa ou então a finalização da execução do mesmo.

## **NOTAS FINAIS E AGRADECIMENTOS**

Neste trabalho utilizamos todos os conhecimentos apresentados nas aulas e vimos um pouco de como é desenvolver um projeto maior do que listas de exercícios de programação. Além disso tivemos contato com ferramentas novas de desenvolvimento como o GitHub que serão bem útil ao longo da carreira e do curso. Por fim, trabalhar em grupo é sempre um aprendizado para toda a vida.

Todo o desenvolvimento do programa e disponibilização do mesmo se encontra licenciado pela licença GPL-v3.0 e hospedado em um repositório do GitHub.

-->Link do repositório: <https://github.com/Globson/TP-Aeroporto-AEDS>

Integrantes do grupo de trabalho prático:

Yuri Dimitre - 3485

Samuel Pedro - 3494

Aryel Penido - 3500

Agradecimento especial aos nossos monitores da disciplina de Algoritmos e Estrutura de Dados **Angelo Bernar** e **Kayque Avelar** que auxiliaram e orientaram nas dificuldades enfrentadas durante a implementação do nosso código!

***Obrigado pela atenção e interesse em ler toda a documentação e esperamos que nosso algoritmo não deixe a desejar na execução do mesmo!***