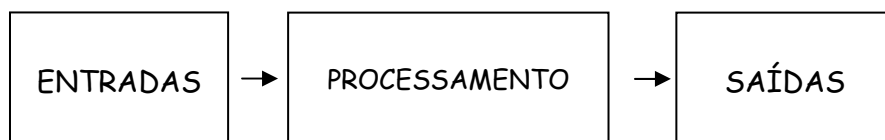


**Objetivo:**

Exercitar as habilidades e conceitos de programação desenvolvidos ao longo da disciplina através da implementação de uma aplicação em C, desenvolvida por um grupo de 1 ou 2 alunos.

O programa deve ser estruturado de forma a receber um conjunto de entradas (simultaneamente ou durante o uso do programa), cuja consistência deve ser verificada, processá-las e fornecer uma ou mais saídas.

**Conteúdos:**

A aplicação desenvolvida deverá demonstrar os seguintes conteúdos:

1. (2 pontos) Habilidade em estruturar programas pela decomposição da tarefa em sub-tarefas, implementadas através do uso de subprogramas.
2. (2 pontos) Documentação adequada de programas (indentação, utilização de nomes de variáveis, abstração dos procedimentos para obter maior clareza, uso de comentários no código).
3. (2 ponto) Domínio na utilização de tipos de dados simples e estruturados (arranjos, estruturas) e passagem de parâmetros.
4. (1 ponto) Formatação e controle de entrada e saída, com orientação correta ao usuário.
5. (1 ponto) Utilização de arquivos.
6. (2 pontos) Atendimento aos requisitos do programa.

**Aplicação a ser desenvolvida:**

O trabalho a ser desenvolvido será uma simulação de um sistema simplificado de controle aéreo. O usuário deve fazer o roteamento de uma lista de aeronaves para as respectivas pistas, seguindo algumas regras.

Para isso, o programa deve iniciar por um menu inicial de opções para o usuário, no formato de uma barra de menu no topo da tela, de acordo com a figura a seguir.

Iniciar Continuar Abrir <b>Salvar</b> Relatório Sair			Menu
TAM9869 NHT156 ...	TAM São Paulo THL Santa Maria ...	Boeing 737, 2000m, 7t LET 410, 565m, 6,6t ...	Lista dos vãos
>> NHT156 PISTA4			Linha de comando
Pista 1 – 3500m – Capacidade 250t	Status – Ocupada		Lista de pistas
Pista 2 – 1500m – Capacidade 100t	Status – Livre		
Pista 3 – 600m – Capacidade 50t	Status – Livre		
Pista 4 – 4000m – Capacidade 500t	Status – Livre		

Para mostrar as informações, o programa deverá carregar três arquivos texto fornecidos ao aluno. O arquivo **PISTAS.TXT** conterá as informações das pistas disponíveis no aeroporto. O arquivo **AVIOES.TXT** informará os modelos de aviões que serão pousados. E o arquivo **VOOS.TXT** informará os vôos que devem ser escalonados. A estrutura dos arquivos é como segue:

#### **PISTAS.TXT**

<CODIGO-PISTA>, <EXTENSAO (metros)>, <PESOMAXIMO (kg)>

#### **AVIOES.TXT**

<CODIGO-AVIAO>, <MODELO>, <METROS-POUSO>, <PESOMAXIMO (kg)>

#### **VOOS.TXT**

<CODIGO-VOO>, <CIDADE-ORIGEM>, <CODIGO-AVIAO>

Com o programa carregado, a interface deve disponibilizar uma lista com os próximos vôos a serem escalonados (não todos). O usuário digita na linha de comando o código do vôo e o nome da pista para pouso, conforme o exemplo da figura. Se o pouso for válido, então o estado da pista deve ser posto em ocupado. A seguir, o vôo deve ser eliminado da lista de vôos e a lista atualizada. Cada pouso corresponde a um “turno” do sistema. O procedimento deve seguir as seguintes regras:

1. Um avião não pode pousar em uma pista que:
  - a. For menor que o tamanho requerido para pouso;
  - b. Não comportar o peso do avião;
  - c. Estiver ocupada;
2. Uma pista deve ser desocupada após o pouso seguinte (ou seja, após o “turno” seguinte);
3. Cada vôo deve gerar um evento em um log de pouso, descrito mais adiante.
4. A linha de comando deve aceitar o comando “parar”, que irá parar a simulação e colocar ativar o menu superior.

### **Detalhamento das opções (disponíveis no menu)**

#### **Iniciar**

- a. O sistema ativa o escalonamento de aviões, colocando o cursor na linha de comando;
- b. Se o programa já estava parado, o comando iniciar deve reiniciar a simulação do começo.

#### **Continuar**

O sistema reativa o escalonamento de aviões no estado em que está, colocando o cursor na linha de comando.

#### **Abrir**

O usuário deve selecionar se quer começar uma simulação nova ou continuar uma anterior.

- a. Se for uma simulação nova:
  - i. Abre o arquivo formatado PISTAS.TXT somente para leitura e carrega um array de estruturas com as pistas e os seus respectivos estados. O número de pistas pode variar.
  - ii. Abre o arquivo formatado AVIOES.TXT para leitura e carrega seu conteúdo para um arranjo de estruturas em memória.
  - iii. Abre o arquivo formatado VOOS.TXT para leitura e carrega seu conteúdo para um arranjo de estruturas em memória.

- iv. Mostra o conteúdo de VOOS.TXT na janela da Lista de Voos, um embaixo do outro. Se o conteúdo for maior que janela, mostrar as os primeiros vôos do arquivo apenas.
  - v. Mostra o conteúdo de PISTAS.TXT na janela da Lista de PISTAS, um embaixo do outro e em ordem alfabética.
  - vi. **O usuário usará a tecla TAB para passar o cursor entre as opções do menu.**
  - vii. Por *default*, na barra de tarefa, será enfatizada a opção Abrir.
- b. Se for uma simulação antiga
- i. Abrir os arquivos binários com os registros anteriores salvos (ver item **Salvar**) e carregar o estado da aplicação novamente.

### Salvar

Deve salvar o estado atual da aplicação (registros de vôos, pistas, e aviões) em arquivos binários, para que a simulação possa ser retomada. O formato e a divisão dos arquivos deve ser definida por cada aluno.

### Relatórios

Oferece ao usuário a opção de realizar as seguintes estatísticas sobre os dados (salvar **em arquivo TEXTO formatado**):

- a. Tentativas de pouso em cada pista;
- b. Tentativas de pouso bem sucedidas;
- c. Maior avião que pousou em cada pista;
- d. Quantidade de pousos de cada modelo de avião;

### Log

Cada execução do sistema deve ser registrada em um arquivo de log (TEXTO) e mantida a cada execução do programa. O formato do log de cada execução deve ser:

```
<data e hora do inicio da execução do programa – AAAA/MM/DD hh:mm:ss>
<carga de arquivos>
<tentativas de pouso no formato no formato codVoo-codPista-Sucesso/Falha>
<data e hora do fim da execução do programa – AAAA/MM/DD hh:mm:ss>
```

Exemplo:

```
Inicio Execução - 2008/05/16 14:07:34
Carga de arquivo: c:\aviões.txt
Carga de arquivo: c:\pistas.txt
Carga de arquivo: c:\voos.txt
Pouso <TAM1234-34/R-S>
Pouso <VRG3245-34/R-F>
Pouso <VRG3245-164/R-F>
...
Fim Execução - 2008/05/16 15:13:26
Inicio Execução - 2008/05/17 11:07:14
Carga de arquivo: c:\aviões.txt
...
```

### Sair

- a. Dá mensagem ao usuário de encerramento.
- b. Fecha todos os arquivos abertos e encerra o sistema.

## Tipo básico das Estruturas

```
typedef struct
{
    char codpista[10];
    int extensao;
    int peso;
} PISTA;

typedef struct
{
    int codaviao;
    char modelo[30];
    int extensaoparapouso;
    int pesomaximo;
} AVIAO;

typedef struct
{
    char codvoo[30];
    char origem[20]
    int codaviao;
} VOO;
```

## Produto

1. Programa executável.
2. Código documentado (arquivos .c, .cpp)
3. Instruções de uso: texto curto explicando como o programa funciona, como a entrada de dados é informada ao programa, exemplos, etc, suficiente para que seja executado sem o autor.
4. Dados de teste (opcional): dados usados para testar o programa

## Proposta: dia 9 de junho (segunda-feira)

Entregar em aula a composição do Grupo (duplas).

## Relatório de andamento: dia 16 de junho (segunda-feira)

Entrega em aula de relatório escrito descrevendo as funcionalidades solicitadas para o Menu do usuário ABRIR e BUSCAR, com as respectivas funções de manipulação de arquivos, que equivale cerca de 50% do trabalho.

## Entrega: até dia 23 de junho (segunda-feira), incluindo:

1. Código documentado (arquivos.cpp) contendo em seu cabeçalho o nome dos alunos.
2. Programa executável
3. Instruções de uso: texto curto explicando como o funcionamento do programa, como a entrada de dados é informada ao programa, exemplos, etc.
4. Dados de teste (opcional): dados usados para testar o programa

## Apresentação

Os trabalhos serão apresentados nos dias **24 e 26 de junho**, na aula prática. Somente serão apresentados programas que tenham sido entregues, através de *upload* no Moodle. Nenhuma alteração será permitida.

**Avaliação:**

- Cada um dos conteúdos do trabalho será avaliado separadamente segundo a pontuação descrita no item conteúdos acima.
- O programa deve atender todos os requisitos listados abaixo, não deve apresentar erros de compilação e rodar normalmente, pontos serão deduzidos caso contrário.
- A nota do trabalho prático corresponderá a 50% da nota de laboratório.