

# SLTEDOS – Exercícios TADS

Fábio Modesto

Bacharelado em Ciência da Computação  
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo  
fabiomodesto@ifsp.edu.br  
slt.ifsp.edu.br

## Exercício Único

Desenvolver um pequeno sistema de cadastro de pessoas em C++, organizando o código em três arquivos (header, implementação e main), empregando struct, enum e um vetor estático de 10 posições.

### Requisitos

#### 1. Defina em cadastro.h:

- Um enum Sexo { Masculino, Feminino, Outro };
- Uma struct Pessoa com os campos:
  - string nome;
  - int idade;
  - Sexo sexo;
- Protótipos das funções:
  - bool cadastrarPessoa(Pessoa vetor[], int tamanho, string\* nome, int idade, Sexo sexo, int\* proximaPosicao);
  - void listarPessoas(Pessoa vetor[], int tamanho);

#### 2. Implemente em cadastro.cpp:

- cadastrarPessoa:
  - Recebe o vetor de Pessoa, seu tamanho, os dados da nova pessoa e uma variável proximaPosicao que indica onde inserir.
  - Se houver espaço ( $\text{proximaPosicao} < \text{tamanho}$ ), insere os dados, incrementa proximaPosicao e retorna true. Caso contrário, retorna false.
- listarPessoas:

- Percorre todo o vetor e imprime os dados de cada posição, convertendo o enum Sexo em texto.

### 3. Crie em main.cpp

- Um vetor `Pessoa pessoas[10]`; e um `int proximaPosicao = 0`;

- Um laço que exiba um menu com opções:

- 1) Cadastrar pessoa

- 2) Listar pessoas

- 3) Sair

- Para “Cadastrar pessoa”, solicite ao usuário o nome (linha), a idade (inteiro) e a opção de sexo (0, 1 ou 2), chame `cadastrarPessoa` e informe sucesso ou cadastro cheio.

- Para “Listar pessoas”, chame `listarPessoas`.

- Trate entradas inválidas de menu e finalize ao escolher Sair.

### Pontos de atenção

- Use `namespace std`; apenas nos arquivos `.cpp`, não dentro do header.

- Organize bem seus includes e proteções de header ( `#ifndef`, `#define`, `#endif` ).

- Garanta que, quando o vetor estiver cheio, novas tentativas de cadastro sejam rejeitadas.

- Separe claramente cada arquivo em seu papel (interface, implementação e fluxo principal).

### Critérios de avaliação

- Correção da lógica de inserção e listagem.

- Adesão ao enunciado (struct, enum, vetor de 10, menu).

- Qualidade do código: nomes claros, formatação consistente e tratamento de erros mínimos (p. ex., verificar limites do menu).

- Organização em arquivos `.h` e `.cpp`.

### Entregue:

- `cadastro.h`

- `cadastro.cpp`

- `main.cpp`

