

# Roteiro de Prática em C — Sistema de Controle de Alunos

O projeto desta prática será construir um **Sistema de Controle de Alunos**, que roda em terminal, e permitirá cadastrar alunos, listar, buscar, calcular estatísticas e salvar informações em arquivo.

## Objetivos da Prática

Aplicar na prática:

- Variáveis e tipos primitivos
- Entrada e saída
- Estruturas de decisão (if, switch)
- Estruturas de repetição (while, for)
- **Vetores**
- **Structs**
- **Funções**
- Arquivos (opcional, para os mais avançados)
- Modularização (opcional)

## Contexto do Projeto

Você deverá implementar um sistema em C para gerenciar alunos de uma turma. Cada aluno deve possuir:

- Nome
- Matrícula
- Idade
- Nota 1
- Nota 2
- Média (calculada)

Estruture o sistema para:

1. Cadastrar alunos
2. Listar alunos
3. Buscar aluno por matrícula
4. Exibir estatísticas gerais da turma
5. Remover aluno (opcional)
6. Salvar/Carregar de arquivo (opcional)

## ❖❖ Etapa 1 — Planejamento da Struct

Crie uma struct chamada **Aluno** contendo os atributos acima. Reflita sobre:

- Tipo apropriado para nome (string → vetor de char)
- Tipos das notas e média (float ou double)
- Ordem dos campos

## Etapa 2 — Vetor de alunos

O sistema deve armazenar **até 100 alunos** em um **vetor de structs**. Defina:

- Aluno alunos[100];
- Variável para controlar quantos alunos foram cadastrados

Desafios:

- Evitar cadastro acima do limite
- Garantir que matrículas não se repitam (opcional)

## Etapa 3 — Funções

Crie funções separadas para organizar melhor o programa. Sugestões:

1. void cadastrarAluno(...)
2. void listarAlunos(...)
3. int buscarAluno(...)
4. float calcularMedia(float, float)
5. void mostrarEstatisticas(...)

E funções extras opcionais:

- void removerAluno(...)
- void salvarEmArquivo(...)
- void carregarArquivo(...)

## Etapa 4 — Menu Interativo

Crie um menu usando `while` e `switch`:

- 1 - Cadastrar aluno
- 2 - Listar alunos
- 3 - Buscar aluno
- 4 - Estatísticas da turma
- 5 - Remover aluno (opcional)
- 0 - Sair

Sempre retornar ao menu até a opção **0**.

O menu deve chamar as funções criadas na etapa

anterior. Desafios:

- Validar entradas
- Não deixar o programa quebrar com entradas inválidas

# Etapa 5 — Implementar cada funcionalidade

## Cadastrar aluno

- Preencher todos os campos
- Calcular média chamando função específica
- Armazenar no vetor
- Incrementar contador

## Listar alunos

- Percorrer o vetor
- Mostrar dados de forma organizada
- Mostrar mensagem quando não houver nenhum aluno

## cadastrado **Buscar aluno**

- Solicitar matrícula
- Percorrer vetor
- Se encontrar → exibir aluno
- Se não → avisar

## Estatísticas da turma

Calcular:

- Quantidade de alunos
- Média geral da turma
- Maior média
- Menor média

Usar funções e laços de repetição.

## Remover aluno (opcional)

- Encontrar aluno pela matrícula
- "Remover" deslocando o vetor para trás

# **Etapa 6 — Recursos Avançados (opcional)**

Para alunos mais avançados, adicionar:

## **Salvar em arquivo (txt ou binário)**

- Gravar todos os alunos
- Ler os alunos ao iniciar o programa

## **Modularização**

Criar:

- aluno.h
- aluno.c
- main.c

Separando funções e a struct.

# **Etapa Final — Testes e Validação**

Testar:

- Cadastro de vários alunos
- Busca existente e inexistente
- Estatísticas com diferentes valores
- Limite máximo de alunos
- Entradas inválidas (texto no lugar de números)

# **Entrega**

O aluno deve entregar:

- Código completo
- Arquivo “relatório.txt” contendo:
  - O que foi implementado
  - Dificuldades encontradas
  - O que aprendeu

]