

# Roteiro Aula Prática



**ALGORÍTMOS E LÓGICA  
DE PROGRAMAÇÃO**

# ROTEIRO DE AULA PRÁTICA

**NOME DA DISCIPLINA:** ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

**Unidade:** 4 – APLICAÇÕES DE PROGRAMAÇÃO

**Aula:** 4 – Registros e arquivos

## OBJETIVOS

- ✓ Interpretação dos requisitos e desenvolvimento correto do algoritmo abrangendo todas as possibilidades distintas na execução do programa;
- ✓ Aplicar conceitos de manipulação de “arquivos”, além da criação e acesso a registros “Structs” na Linguagem de Programação “C”.

## SOLUÇÃO DIGITAL

- Visual Studio Code (VS Code)

**LINK SOLUÇÃO DIGITAL:** <https://code.visualstudio.com/download>

O VS Code é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS. Conta com a instalação de extensões para programação em diversas linguagens de programação.

- Tipo de licença: Código-fonte: Licença MIT; Binários: Freeware  
Computador com acesso à internet, e com o mínimo de 4 GB de Memória RAM.
- Instalação do Visual Studio Code + extensões:
  - C/C++ for Visual Studio Code;
  - C/C++ Compile Run extension.

## PROCEDIMENTO/ATIVIDADE

### ATIVIDADE PROPOSTA:

Você foi designado pela sua instituição de ensino para desenvolver um programa de classificação dos alunos. O programa deverá solicitar alguns dados, inclusive as médias das disciplinas de cada aluno para que ele possa fazer o cálculo da média do aluno. Deverá solicitar também o nome do aluno.

### Descrição do Problema:

Para realização da atividade você deve utilizar os recursos de Structs e Arquivos. A Struct armazenará os dados do aluno como: Nome, média da primeira disciplina, média da segunda disciplina, média geral por aluno e sua classificação.

Um arquivo do tipo “.txt” deve ser utilizado para armazenar as informações no final da execução do programa. O layout do arquivo deve conter:

- **Nome do Aluno:**
- **Média Disciplina 1:**
- **Média Disciplina 2:**
- **Média Geral:**
- **Classificação:**

## PROCEDIMENTOS PARA A REALIZAÇÃO DA ATIVIDADE:

Faça uma análise detalhada de todas as ações que o programa deve executar, bem como, as estruturas algorítmicas necessárias para o desenvolvimento do código. Aplique os fundamentos da lógica de programação e algoritmos na criação de uma solução para o cenário apresentado utilizando a Linguagem de Programação “C” tendo como apoio o editor de código “VS Code”.

A seguir, os procedimentos que devem ser realizados:

### 1. Passo 1: Definição das estruturas de dados:

- a. Criar a estrutura(Struct) “Aluno” com os campos necessários (nome, médias das disciplinas, média geral, classificação).

Você pode optar pelo nome da Struct ser “Aluno”, ou outro nome qualquer desde que você respeite as regras para nomenclatura de variáveis: evitando acentuação, sem utilização de espaços em branco, não podendo utilizar palavras reservadas à linguagem, exemplo: “void” e não começar com caractere numérico.

Aqui optei pelo nome “Aluno”.

```
typedef struct {
    char nome[MAX_NOME];
    float media_disciplina1;
    float media_disciplina2;
    float media_geral;
    int classificacao;
} Aluno;
```

### 2. Passo 2: Implementação das funções de cálculo:

- a. Implementar uma função para calcular a média geral de um aluno com base nas médias das disciplinas.

Um exemplo da função para calcular a média de um aluno. Passa-se por parâmetro as médias das disciplinas e depois retorna o cálculo da média.

```
float calcularMediaGeral(float media_disciplina1, float media_disciplina2) {
    return (media_disciplina1 + media_disciplina2) / 2;
}
```

- b. Implementar uma função de comparação para ser utilizada na ordenação da turma.

Essa função (**compararMedias**) é chamada dentro de outra função que salva o arquivo .txt que tem o objetivo de ordenar o aluno dentro do arquivo de acordo com sua classificação por nota. Exemplo:

```
int compararMedias(const void *a, const void *b) {
    const Aluno *alunoA = (const Aluno *)a;
    const Aluno *alunoB = (const Aluno *)b;
    if (alunoA->media_geral < alunoB->media_geral) {...
} else if (alunoA->media_geral > alunoB->media_geral) {
    return -1;
} else {
    return 0;
}
}
```

### 3. Passo 3: Entrada de dados:

- a. Pedir ao usuário que informe quantos alunos serão registrados.

Isso é feito dentro da função main(). Uma sugestão de como resolver está logo a seguir:

```
69 ∨ int main() {
70     system("clear");
71     int num_alunos;
72     printf("Quantos alunos deseja registrar? (maximo %d): ", MAX_ALUNOS);
73     scanf("%d", &num_alunos);
```

- b. Usar um loop para solicitar o nome, médias das disciplinas 1 e 2 de cada aluno e armazenar esses dados na estrutura Aluno.

Dentro de um laço qualquer na linguagem de programação C, solicite do usuário esses dados.

### 4. Passo 4: Cálculo das médias e classificação:

- a. Calcular a média geral de cada aluno utilizando uma função.

A função que faz esse cálculo é a **calcularMediaGeral()** já exibida no item 2.

- b. Ordenar a turma em ordem decrescente de acordo com as médias dos alunos.

Você deve comparar as médias gerais de cada aluno e colocar a classificação de cada um dentro do arquivo .txt ao lado dos dados do aluno.

- c. Atribuir as classificações aos alunos com base na ordem da turma.

Inserir dentro do arquivo um campo para a classificação do aluno. Utilizar o campo classificação da Struct.

```
typedef struct {
    char nome[MAX_NOME];
    float media_disciplina1;
    float media_disciplina2;
    float media_geral;
    int classificacao;
} Aluno;
```

#### 5. Passo 5: Saída de dados:

- a. Para cada aluno, escrever no arquivo seu nome, médias das disciplinas, média geral e classificação.
- b. Calcular e escrever no arquivo a média geral da turma.

Uma sugestão para a saída do arquivo texto logo a seguir:

```
notas_alunos.txt
1 Nome: C
2 Media Disciplina 1: 70.00
3 Media Disciplina 2: 80.00
4 Media Geral: 75.00
5 Ordem: 1
6
7 Nome: B
8 Media Disciplina 1: 60.00
9 Media Disciplina 2: 70.00
10 Media Geral: 65.00
11 Ordem: 2
```

#### 6. Passo 6: Finalização:

- a. Fechar o arquivo.

Utilizar a classe "FILE" da linguagem C para manipular dados no arquivo.

#### 7. Execução do Programa:

Apresentar o programa, demonstrando o seu funcionamento.

Exemplo do programa sendo executado:

```
PROBLEMAS 1 SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL
Quantos alunos deseja registrar? (maximo 50): 2

Aluno 1:
Nome: Anderson
Media Disciplina 1: 100
Media Disciplina 2: 90

Aluno 2:
Nome: Renata
Media Disciplina 1: 85
Media Disciplina 2: 48

Dados dos alunos salvos no arquivo 'notas_alunos.txt'.
```

Na sequência, o arquivo .txt gerado pelo programa.

```
notas_alunos.txt
1  Nome: Anderson
2  Media Disciplina 1: 100.00
3  Media Disciplina 2: 90.00
4  Media Geral: 95.00
5  Ordem: 1
6
7  Nome: Renata
8  Media Disciplina 1: 85.00
9  Media Disciplina 2: 48.00
10 Media Geral: 66.50
11 Ordem: 2
12
13
14 Media geral da turma: 161.50
15
```

Munido de todas essas informações, coloque suas habilidades de programação em prática e crie um programa que atenda a todos os requisitos solicitados pela sua instituição de ensino.

### CHECKLIST:

- Analisar o cenário para criação do código-fonte do programa;
- Instalar o VS Code em seu dispositivo. Baixe a IDE escolhendo o sistema operacional por meio do link: <https://code.visualstudio.com/download>
- Desenvolver o algoritmo na Linguagem de Programação C;
- Considerar todos os critérios solicitados para construção do algoritmo.
- Executar testes e possíveis correções;
- Apresentar o programa e verificar seu correto funcionamento.

### RESULTADOS

O estudante deve entregar um arquivo em PDF contendo toda a codificação do exercício documentada, ou seja, para cada trecho do código-fonte, o estudante deve anexar um texto explicando o que acontece quando o trecho do código é executado.

O arquivo deverá conter:

- Capa;
- Folha de rosto com os dados da disciplina e do aluno;
- Codificação completa dos exercícios;
- Referências bibliográficas (quando houver).

## **RESULTADOS DE APRENDIZAGEM:**

O objetivo da atividade é aplicar conceitos das “estruturas de seleção”, “laços de repetições”, “funções”, “Structs” e “arquivos”, para coletar os dados, realizar os cálculos e fazer a classificação dos alunos. Diante do cenário apresentado, ser capaz de aplicar os conceitos de lógica de programação desenvolvendo o raciocínio lógico de forma prática em um contexto do mundo real.