



UNIVERSIDADE ANHANGUERA – UNIDERP
TRABALHO 3

Curso: Ciência da Computação, Engenharia da Computação e CTS em Análise e Desenvolvimento de Sistemas

Disciplina: Algoritmos e Estrutura de Dados

Docente: Paulo Eduardo Suriano Tiene

Orientações para o a atividade avaliativa:

- A atividade pode ser realizada em grupos de até 4 pessoas.
- Nota: valor máximo de 1500 pontos;
- Prazo máximo para entrega: até as 23:59 horas do dia 13/05/2025;
- Forma de entrega do trabalho: deverá ser enviada apenas o arquivo .c para o endereço de e-mail: paulo.tiene@anhanguera.com
- No assunto do e-mail coloque o nome do aluno (ou alunos) e o nome da disciplina, exemplo:
- “SEUNOME / Trabalho 3 / AED - Segunda” (para alunos que tem aula na segunda-feira)
- “SEUNOME / Trabalho 3 / AED - Terça” (para alunos que tem aula na terça-feira)

O que deve ser entregue:

- Código fonte do programa em C (**bem identada e comentada**). Obs: não envie o executável compilado, apenas o código fonte.

Comentários Gerais:

- Clareza, indentação e comentários no programa também vão valer pontos.
- Trabalhos copiados serão penalizados conforme anunciado.
- Alguns grupos serão sorteados aleatoriamente para entrevista sobre o trabalho. O participante ou os participantes do grupo deverão saber responder a TODAS as perguntas feitas.

Descrição do Trabalho

O objetivo deste trabalho é implementar uma **lista duplamente encadeada** para gerenciar um sistema de **notas de alunos**, permitindo inserção, remoção, edição e listagem de forma dinâmica.

Critérios de Avaliação:

- **(375 pontos):** Inserção ordenada pelo CPF (sem duplicatas).
- **(375 pontos):** Remoção de um aluno em qualquer posição da lista, utilizando o CPF.
- **(375 pontos):** Edição dos dados de um aluno (Nome, Nota1 e Nota2) com atualização automática da média e validação da existência do CPF.
- **(375 pontos):** Saída correta das informações (Endereço do nó, CPF, Nome, Nota1, Nota2, Média, endereço do nó anterior e endereço do próximo nó).

Requisitos:

- A lista deve ser **duplamente encadeada**, utilizando **alocação dinâmica de memória**.
- Cada aluno deve conter os seguintes dados:
 - **CPF** (único e ordenado),
 - **Nome**,
 - **Nota1**,
 - **Nota2**,
 - **Média** (calculada automaticamente como a média das duas notas).

```
5 struct Aluno {  
6     long long int cpf;  
7     char nome[100];  
8     float nota1;  
9     float nota2;  
10    float media;  
11  
12    struct Aluno* anterior;  
13    struct Aluno* proximo;  
14 };  
15 typedef struct Aluno node;
```

Funcionalidades:

Inserir Aluno:

- Insere um aluno na lista de forma **ordenada pelo CPF**.
- **Não permite duplicatas**. Se um CPF já estiver cadastrado, uma mensagem de alerta deve ser exibida.

Remover Aluno:

- Remove um aluno da lista com base no **CPF** informado.
- Caso o CPF não exista na lista, uma mensagem informando que o aluno não foi encontrado deve ser exibida.

Editar Aluno:

- Permite buscar um aluno pelo CPF e **editar os campos Nome, Nota1 e Nota2**.
- A **média** deve ser recalculada automaticamente após qualquer alteração nas notas.
- Caso o CPF não seja encontrado, deve exibir uma mensagem de erro.

Exibir Lista:

- Mostra todos os dados dos alunos, incluindo:
 - CPF, Nome, Nota1, Nota2 e Média.
 - Endereço do nó atual, do **nó anterior** e do **próximo nó**, facilitando o acompanhamento da estrutura da lista.

O sistema deve garantir eficiência na manipulação da lista, permitindo fácil manutenção e visualização dos dados.

Exemplos de saída:

Inserir aluno:

```
=== MENU ===
1 - Inserir Aluno
2 - Remover Aluno
3 - Editar Aluno
4 - Listar Alunos
0 - Sair
Escolha: 1
CPF: 12345678901
Nome: Paulo
Nota 1: 9
Nota 2: 8
Aluno inserido com sucesso!
```

CPF já cadastrado ao inserir:

```
=== MENU ===
1 - Inserir Aluno
2 - Remover Aluno
3 - Editar Aluno
4 - Listar Alunos
0 - Sair
Escolha: 1
CPF: 12345678901
Erro: CPF já cadastrado.
```

Remoção:

```
=== MENU ===
1 - Inserir Aluno
2 - Remover Aluno
3 - Editar Aluno
4 - Listar Alunos
0 - Sair
Escolha: 2
CPF do aluno a remover: 12345678901
Aluno removido com sucesso!
```

CPF não encontrado ao remover:

```
=== MENU ===
1 - Inserir Aluno
2 - Remover Aluno
3 - Editar Aluno
4 - Listar Alunos
0 - Sair
Escolha: 2
CPF do aluno a remover: 98765432101
Aluno não encontrado.
```

Impressão:

```
=== MENU ===
1 - Inserir Aluno
2 - Remover Aluno
3 - Editar Aluno
4 - Listar Alunos
0 - Sair
Escolha: 4
=== Lista de Alunos ===
Endereco atual: 0x616df6bc9ac0
CPF: 12345678901
Nome: Paulo
Notas: 9.00 | Nota2: 8.00 | Média: 8.50
Anterior: (nil) | Próximo: 0x616df6bc9be0

Endereco atual: 0x616df6bc9be0
CPF: 56728174812
Nome: Eraldo
Notas: 7.00 | Nota2: 7.00 | Média: 7.00
Anterior: 0x616df6bc9ac0 | Próximo: 0x616df6bc9b50

Endereco atual: 0x616df6bc9b50
CPF: 98738218491
Nome: Cassia
Notas: 7.00 | Nota2: 4.00 | Média: 5.50
Anterior: 0x616df6bc9be0 | Próximo: (nil)
```

Editar aluno:

```
=== MENU ===
1 - Inserir Aluno
2 - Remover Aluno
3 - Editar Aluno
4 - Listar Alunos
0 - Sair
Escolha: 3
CPF do aluno a editar: 98738218491
Editando aluno: Cassia
Novo nome: Kassia
Nova nota1: 7
Nova nota2: 6
Dados atualizados com sucesso!
```

Caso não encontre o aluno para editar:

```
=== MENU ===
1 - Inserir Aluno
2 - Remover Aluno
3 - Editar Aluno
4 - Listar Alunos
0 - Sair
Escolha: 3
CPF do aluno a editar: 237297358121
Aluno com CPF 237297358121 não encontrado.
```

BOA SORTE E BOM TRABALHO!!!!