



CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER
ESCOLA SUPERIOR POLITÉCNICA
TECNOLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
DISCIPLINA DE ESTRUTURA DE DADOS

ATIVIDADE PRÁTICA

PHELIPE SANTOS – RU: 3316094
PROF. VINICIUS POZZOBON BORIN

GUARUJA – SÃO PAULO
2022

Exercício 1: Faça um algoritmo em linguagem C que emule as características de um player de músicas sendo executado em modo texto, via prompt de comando.

1. Deve-se criar uma playlist das músicas utilizando uma lista encadeada. A lista encadeada poderá ser simples ou dupla, circular ou não circular. Fica a critério do aluno decidir
2. Deve-se armazenar o nome de cada música, do artista/banda e a duração da faixa. Para o armazenamento utilize uma estrutura heterogênea de dados.
3. Para inserção dos dados, você pode criar uma leitura dos dados através de um menu na tela ou já deixá-los armazenados em um arquivo texto no seu computador e só carregar este arquivo ao executar o programa. Ou ambas as soluções. Decida também como você irá implementar a inserção (no início, no fim ou no meio da lista encadeada);
4. Deve existir um menu na tela. Este menu deve permitir a inserção de novas músicas (caso optado pela inserção manual de dados), deve ter a opção de listar todas as músicas da playlist (listagem de uma lista encadeada) na tela e encerrar o programa;

Código:

```
#include <stdio.h>
#include <locale.h>
#include <stdlib.h>
#include <string.h>

typedef struct Lista {
    char musica[40], cantor[40]; //estrutura da lista
    float duracao;
    struct Lista* prox;
}Lista;

Lista* inicio, * fim, * aux;

void inserirMusica() { //logica da inserção de musica
    int c;
    Lista* novo = (struct Lista*)malloc(sizeof(struct Lista));
    printf("\nQual musica? ");
    fgets(novo->musica, 39, stdin);
    printf("Quem é o artista? ");
    fgets(novo->cantor, 39, stdin);
    printf("qual a sua duração ? ");
    scanf_s("%f", &novo->duracao);
    while ((c = getchar()) != '\n' && c != EOF) {}
    novo->prox = NULL;
    if (inicio == NULL) {
        inicio = fim = novo;
    }
    else {
        fim->prox = novo;
    }
}
```

```

        fim = novo;
    }
}

void esvaziarLista() { //logica da exclusão de musica

    char ch;
    if (inicio == NULL) {
        printf("\nA fila esta vazia!\n\n");
    }
    else {
        printf("\nTem certeza que deseja esvaziar toda a lista?(s/n): ");
        ch = getchar();
        if (ch == 's' || ch == 'S') {
            while (inicio != NULL) {
                aux = inicio;
                inicio = inicio->prox;
                free(aux); // Libera o espaco na memoria
            }
            printf("\nA lista foi esvaziada com sucesso!!\n\n");
        }
        else {
            printf("\nA lista nao foi esvaziada!!\n\n");
        }
    }
}

void mostrarLista() { //lista as musicas armazenadas
    aux = inicio;
    if (inicio == NULL) {
        printf("\nA lista esta vazia!");
    }
    else {
        while (aux != NULL) {
            printf("musica: %s", aux->musica);
            printf("cantor: %s", aux->cantor);
            printf("duracao: %f\n\n", aux->duracao);
            aux = aux->prox;
        }
        printf("\n\n");
    }
}

void localizarMusica() {
    aux = inicio;
    char procurada[40];
    if (inicio == NULL) {
        printf("\nA lista estah vazia!");
    }
    else {
        char musica[40];
        printf("Digite a musica que procura: ");
        fgets(procurada, 39, stdin);
        while (aux != NULL) {
            if (strcmp(aux->musica, procurada) == 0) {
                printf("\nMusica %s localizada!\n\n", aux->musica);
                printf("musica %s", aux->musica);
                printf("cantor: %s\n\n", aux->cantor);
                return;
            }
            else {
                aux = aux->prox;
            }
        }
    }
}

```

```

        }
    }
    if (aux == NULL) {
        printf("\nMusica  %s nao localizada!\n", musica);
    }
}
printf("\n\n");
}

int main() { //menu do player
    setlocale(LC_ALL, "");
    system("color 0E");
    int opt, c;

    do {
        printf("*=====*\n");
        printf("| Simulador de Player de Musica      |\n");
        printf("*=====*\n");
        printf("| 1. Inserir Musica          |\n");
        printf("| 2. Pesquisar Musicas       |\n");
        printf("| 3. listar Musicas          |\n");
        printf("| 4. Limpar Lista do Player  |\n");
        printf("| 5. Sair                    |\n");
        printf("*=====*\n");
        printf("Digite sua escolha: ");
        scanf_s("%d", &opt);
        while ((c = getchar()) != '\n' && c != EOF) {}

        switch (opt) {
            case 1:
                inserirMusica();
                printf("\nMusica inserida com sucesso!!\n\n");
                system("pause");
                break;
            case 2:
                localizarMusica();
                system("pause");
                break;
            case 3:
                mostrarLista();
                system("pause");
                break;
            case 4:
                esvaziarLista();
                system("pause");
                break;
            case 5:
                printf("\nO programa foi encerrado com sucesso!!\n\n");
                system("pause");
                break;
            default:
                printf("\nEscolha Invalida!!\n\n");
                system("pause");
                break;
        }
        system("cls");
    } while (opt != 5);

    return 0;
}

```

Imagens do código funcionando exercício 1:

```
C:\Users\phelipe\Documents\Trabalho\lista_2\Debug\lista_2...
|=====|
| Simulador de Player de Musica |
|=====|
| 1. Inserir Musica |
| 2. Pesquisar Musicas |
| 3. listar Musicas |
| 4. Limpar Lista do Player |
| 5. Sair |
|=====|
Digite sua escolha: 1

Qual musica? call out my name
Quem é o artista? the weekend
qual a sua duração ? 3:00

Musica inserida com sucesso!!

Pressione qualquer tecla para continuar. . .
```

```
C:\Users\phelipe\Documents\Trabalho\lista_2\Debug\lista_2...
|=====|
| Simulador de Player de Musica |
|=====|
| 1. Inserir Musica |
| 2. Pesquisar Musicas |
| 3. listar Musicas |
| 4. Limpar Lista do Player |
| 5. Sair |
|=====|
Digite sua escolha: 2
Digite a musica que procura: sad but true

Musica sad but true
localizada!

musica sad but true
cantor: Metalica

Pressione qualquer tecla para continuar. . .
```

```
C:\Users\phelipe\Documents\Trabalho\lista_2\Debug\lista_2...
|=====|
| Simulador de Player de Musica |
|=====|
| 1. Inserir Musica |
| 2. Pesquisar Musicas |
| 3. listar Musicas |
| 4. Limpar Lista do Player |
| 5. Sair |
|=====|
Digite sua escolha: 5

O programa foi encerrado com sucesso!!

Pressione qualquer tecla para continuar. . .
```

Exercício 2: Faça um algoritmo em linguagem C que realiza a busca de um aluno da UNINTER no AVA. A busca deve ser realizada utilizando uma estrutura de dados bastante eficiente para esta tarefa de busca. Dentre as estruturas que podem ser empregados estão: árvore binária ou hash.

1. Deve-se armazenar o nome do aluno, seu e-mail e seu RU. Para o armazenamento utilize uma estrutura heterogênea de dados.
2. Não é necessário fazer a leitura dos dados dos alunos manualmente. Você já pode deixar pré-cadastrado os dados no seu código. Cadastre pelo menos uns 10 contatos de alunos na sua estrutura de dados. Um dos contatos deverá ser o seu próprio nome e o seu RU da UNINTER;
3. Em um menu na tela, peça para o usuário digitar um RU. O programa deverá realizar a busca por este RU na estrutura de dados e, caso localize o RU, deverá mostrar o nome correspondente do aluno e o e-mail deste contato. Caso não localize, uma mensagem de erro deve ser apresentada.
4. Para testar o programa, teste a busca com o seu RU e coloque a captura de tela.

Código:

```
#include <stdlib.h>
#include <stdio.h>
#include <locale.h>
#include <string.h>

void inicializarTabela(); // função de espalhamento (resto da divisão do numeroRU por M). Inicializa a tabela com o valor -1.
void inserir(); // função para inserir um aluno na tabela.
void listarTabela(); //funcao para listar todos os alunos da tabela.
int criarCodigoHash(int numeroRU); // função para gerar um código hash para cada elemento da tabela.

#define M 11 // valor constante chamado "M" para representar o tamanho da tabela

// estrutura dadosAluno com nome, matrícula e email
typedef struct
{
    int ruAluno;
    char nome[50];
    char email[50];
}dadosAluno;

// tabela hash do tipo dadosAluno
dadosAluno tabelaHash[M];

// função para ler e retornar os dados do aluno
dadosAluno lerdadosAluno()
{
    dadosAluno Aluno;
```

```

printf("Digite o RU do aluno:\n");
scanf_s("%d", &Aluno.ruAluno);
scanf_s("%*c");
printf("Digite o nome do Aluno:\n");
fgets(Aluno.nome, 50 - 1, stdin);
printf("Digite o e-mail do Aluno:\n");
fgets(Aluno.email, 50 - 1, stdin);
printf("\nDados inseridos com sucesso!\n");
printf("\nRU:\t%d\n", Aluno.ruAluno);
printf("Aluno:\t%s", Aluno.nome);
printf("E-mail\t%s", Aluno.email);
system("pause");
return Aluno;
}

// função para procurar um aluno na tabela
dadosAluno* procurar(int numeroRU)
{
    int indice = criarCodigoHash(numeroRU);
    while (tabelaHash[indice].ruAluno != -1)
    {
        if (tabelaHash[indice].ruAluno == numeroRU)
        {
            return &tabelaHash[indice];
        }
        else
        {
            indice = criarCodigoHash(indice + 1);
        }
    }
    return NULL;
}

//função principal
int main()
{
    setlocale(LC_ALL, "Portuguese");
    system("color 70"); // muda a cor do fundo de tela para azul e a fote para branco.

    int opcao, numeroRU;
    dadosAluno* Aluno;

    inicializarTabela(); // inicializa a tabela com os índices para cada aluno.

    // menu de opções
    do
    {

        system("cls");
        // interface que será apresentada ao usuário utilizando printf

        printf("\n\n *** CENTRO UNIVERSITÁRIO UNINTER 2022 ***");
        printf("\n\n *** ANÁLISE E DESENVOLVIMENTO DE SISTEMAS ***");
        printf("\n\n *** MATÉRIA: ESTRUTURA DE DADOS ***");
        printf("\n\n *** CADASTROS DE ALUNOS *** \n\n\n");
        printf("*-----*\n");
        printf("\tMENU PRINCIPAL\n");
        printf("*-----*\n\n");
        printf("| 1. Inseir novo aluno          |\n");
        printf("| 2. Procurar aluno por RU      |\n");
        printf("| 3. Listar todos os alunos     |\n");
    }

```

```

printf("| 4. Sair | \n");
printf("*-----*\n");
printf("\n");
printf("Digite sua escolha: ");
scanf_s("%d", &opcao);

switch (opcao)
{
case 1:
    system("cls");
    printf("SISTEMA DE CADASTRO DE ALUNOS UNINTER\n\n");
    printf("MENU DE INSERÇÃO DE ALUNO\n\n");
    inserir(); // chama a função para inserir.
    break;
case 2:
    system("cls");
    printf("SISTEMA DE CADASTRO DE ALUNOS UNINTER\n\n");
    printf("\tMENU DE BUSCA\n\n");
    printf("Digite o RU do aluno a ser buscado:\n");
    scanf_s("%d", &numeroRU);
    Aluno = procurar(numeroRU); // chama a função para procurar o ru, usando
    como parâmetro o ru informado.
    if (Aluno)
    {
        printf("\nDados encontrados!\n");
        printf("\nRU: \t%d\n", Aluno->ruAluno);
        printf("Nome: \t%s", Aluno->nome);
        printf("E-mail: \t%s", Aluno->email);
        printf("\n");
        system("pause");
    }
    else
    {
        printf("\nRU não encontrado!\n");
        system("pause");
    }
    break;
case 3:
    listarTabela(); // chama a função para listagem dos alunos cadastrados.
    break;
case 4:
    system("cls"); // encerra o código.
    system("color 07"); // retorna as cores para fundo preto e letras brancas.
    printf("SISTEMA ENCERRADO!\n\n");
    break;
default:
    printf("Opção inválida!\n");
}

} while (opcao != 4); // fica no menu até que o usuário escolha sair
return 0;
}

// função de espalhamento (resto da divisão do numeroRU por M). Inicializa a tabela
com o valor -1.
void inicializarTabela()
{
    int i;
    for (i = 0; i < M; i++)
    {
        tabelaHash[i].ruAluno = -1;
    }
}

```



```

}

// função para inserir um aluno na tabela.
void inserir()
{
    dadosAluno alu = lerdadosAluno();
    int indice = criarCodigoHash(alu.ruAluno);
    while (tabelaHash[indice].ruAluno != -1)
    {
        indice = criarCodigoHash(indice + 1);
    }
    tabelaHash[indice] = alu;
}

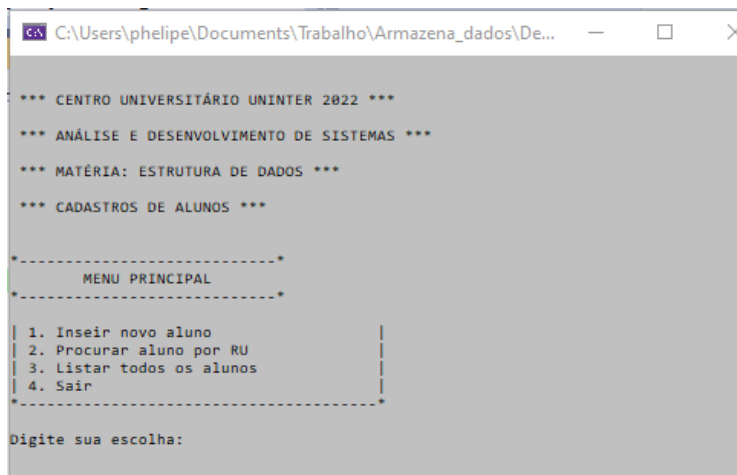
//funcao para listar todos os alunos da tabela.
void listarTabela()
{
    system("cls");
    printf("SISTEMA DE CADASTRO DE ALUNOS UNINTER\n\n");
    printf("\tMENU DE LISTAGEM\n\n");
    int i;
    for (i = 0; i < M; i++)
    {
        if (tabelaHash[i].ruAluno != -1)
        {
            printf("RU:\t%d\n", tabelaHash[i].ruAluno);
            printf("Nome:\t%s", tabelaHash[i].nome);
            printf("E-mail\t%s\n", tabelaHash[i].email);
        }
    }
    printf("\n");
    system("pause");
}

// função para gerar um código hash para cada elemento da tabela.
int criarCodigoHash(int numeroRU)
{
    return numeroRU % M;
}

```

Imagens do código funcionando exercício 2:

Menu:



1.

```
C:\Users\phelipe\Documents\Trabalho\Armazena_dados\De...
SISTEMA DE CADASTRO DE ALUNOS UNINTER

MENU DE INSERÇÃO DE ALUNO

Digite o RU do aluno:
3316094
Digite o nome do Aluno:
Phelipe Santos
Digite o e-mail do Aluno:
phe2111@hotmail.com

Dados inseridos com sucesso!

RU:      3316094
Aluno:   Phelipe Santos
E-mail   phe2111@hotmail.com
Pressione qualquer tecla para continuar. . .
```

2.

```
C:\Users\phelipe\Documents\Trabalho\Armazena_dados\Debug\Armazena...
SISTEMA DE CADASTRO DE ALUNOS UNINTER

MENU DE LISTAGEM

RU:      6678210
Nome:    bart simpson
E-mail   zeruela@elbarto.com

RU:      3316094
Nome:    Phelipe Santos
E-mail   phe2111@hotmail.com

RU:      5678334
Nome:    viviane lima
E-mail   vivil@hotmail.com

RU:      9996724
Nome:    homer simpson
E-mail   dohhh@huhuuu.com

RU:      1456789
Nome:    luna silva
E-mail   luna34@gmail.com

RU:      -858993460
Nome:    biwan kenobi
E-mail

RU:      5567893
Nome:    juanito jones
E-mail   jones@gmail.com

RU:      -858993460
Nome:    nakin skywalker
E-mail

RU:      7899020
Nome:    renato cariane
E-mail   cariane@gmail.com

RU:      1234567
Nome:    Abner Ribeiro
E-mail   ab123@hotmail.com

Pressione qualquer tecla para continuar. . .
```

3.

4.