

CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER
ESCOLA SUPERIOR POLITÉCNICA
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
ESTRUTURAS DE DADOS

ATIVIDADE PRÁTICA

Leandro Roberto De Souza Freitas – RU: 3462061

NOME PROFESSOR: Vinicius Pozzobon Borin

Poços de caldas – MG 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. 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. 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. 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 soluções. Decida também como você irá implementar a inserção (no início, no fim ou no meio da lista encadeada). 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<stdlib.h>

#include<locale.h>

#include<string.h>


//Função Menu

int menu();

void inserirmusica();

void procurar_musca();

void limpar_lista();

void visualizar_lista();

using namespace std;

//Estrutura heterogênea

struct playlist_musical {

char musica[50], cantor[50];
```

```
float tempo;

struct playlist_musical* prox;

}; struct playlist_musical* inicio, * meio, * fim,
* aux;

int main(int argc, char* argv[])
{
    setlocale(LC_ALL, "");
    int opc, c;

    // Esse laço de repetição faz com que o
    programa seja executado pelo menos uma vez na
    tela para escolher a opção desejada
    do
    {
        printf("1 Inserir Musica\n");
        printf("2 Procurar Musica\n");
        printf("3 Ver Lista\n");
        printf("4 Limpar Lista\n");
        printf("5 Sair\n");
        printf("Escolha uma opção :");
        scanf_s("%d", &opc);
        while ((c = getchar()) != '\n' && c != EOF); {}
        switch (opc)
        {
```

```
case 1:
{
    inserirmusica();
    scanf_s("%d", &opc);
    system("pause");
    break;
}

case 2:
{
    procurar_musica();
    system("pause");
    break;
}

case 3:
{
    visualizar_lista();
    system("pause");
    break;
}

case 4:
{
    limpar_lista();
    system("pause");
```

```
break;
}

case 5:
{
system(" pause");
printf("Saindo do menu...\n");
break;
} default:
printf("Opcao nao existe");
break;
}

system("cls");
} while (opc != 5);
return 0;

system("PAUSE");
return EXIT_SUCCESS;
}

//Função inserir musica
void inserirmusica()
{
int c;

playlist_musical* novo = (struct
playlist_musical*)malloc(sizeof(struct
playlist_musical));
```

```
printf("Digite o nome da musica: \n");
scanf_s("%d", &novo->musica);
printf("Digite o nome do  artista: \n");
scanf_s("%d", &novo->cantor);
printf("Duracao da musica");
scanf_s("%f", &novo->tempo);
while ((c = getchar()) != 'n' && c != EOF) {}
novo->prox = NULL;
if (inicio == NULL)
{
    inicio = fim = novo;
}
else
{
    fim->prox = novo;
    fim = novo;
}
}

//Função Limpar a lista
void limpar_lista()
{
    char ch;
    if (inicio == NULL)
```

```
{  
printf("\nLista");  
}  
else  
{  
printf("s\n");  
ch = getchar();  
if (ch == 's');  
while (inicio != NULL)  
{  
aux = inicio;  
inicio = inicio->prox;  
free(aux); // limpar memoria  
}  
printf("\n Lista vazia");  
}  
}  
  
//Função visualizar a lista  
void visualizar_lista()  
{  
aux = inicio;  
if (inicio == NULL)  
{
```

```
printf("\n Lista Vazia");
}
else
{
while (aux != NULL)
{
printf("musica:%s", aux->musica);
printf("cantor:%s", aux->cantor);
printf("duracao: %f\n", aux->tempo);
aux = aux->prox;
}
}
}
//função procurar musica
void procurar_musica()
{
aux = inicio;
if (inicio == NULL)
{
printf("\n Lista vazia");
}
else
{
```



```
char musica[50];
printf("Digiete a faixa:");
scanf_s("%s", &aux);
while (aux != NULL)
{
    if (musica == aux->musica)
    {
        printf("\n Musica %s encontrada\n", aux->musica);
        printf("musica: %s", aux->musica);
        printf("cantor:%s\n", aux->cantor);
        return;
    }
    else
    {
        aux = aux->prox;
    }
}
if (aux == NULL)
{
    printf("\n Musica %s nao encontrada", musica);
}
}

printf("\n");
```

}

CÓDIGO RODANDO (MENU INICIAL)

```
C:\Users\55359\Source\Repos\PlaylistMusica\x64\Debug\PlaylistMusica.exe
1 Inserir Musica
2 Procurar Musica
3 Ver Lista
4 Limpar Lista
5 Sair
Escolha uma opção :
```

CÓDIGO RODANDO (INSERINDO MÚSICA)

```
C:\Users\55359\Source\Repos\PlaylistMusica\x64\Debug\PlaylistMusica.exe
1 Inserir Musica
2 Procurar Musica
3 Ver Lista
4 Limpar Lista
5 Sair
Escolha uma opção :1
Digite o nome da musica:
É o amor
Digite o nome do artista:
Duracao da musica 2.50
-
```

CÓDIGO RODANDO (PROCURAR MÚSICA)

```
C:\Users\55359\Source\Repos\PlaylistMusica\x64\Debug\PlaylistMusica.exe
1 Inserir Musica
2 Procurar Musica
3 Ver Lista
4 Limpar Lista
5 Sair
Escolha uma opção :2

Lista vazia
Pressione qualquer tecla para continuar. . . _
```

CÓDIGO RODANDO (VER LISTA)

```
C:\Users\55359\Source\Repos\PlaylistMusica\x64\Debug\PlaylistMusica.exe
1 Inserir Musica
2 Procurar Musica
3 Ver Lista
4 Limpar Lista
5 Sair
Escolha uma opção :3

Lista VaziaPressione qualquer tecla para continuar. . .
```

EXERCICIO 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 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, 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<string.h>

#include<iostream>

int main() {

    // Declaracao das variaveis

    int RuALuno;

    int opcao;

    char pause[1];

    // Estrutura de dados

    struct alunos {

        char nome[40];

        char email[40];

        int Ru;

    };

    struct alunos aluno_1, aluno_2, aluno_3, aluno_4,
    aluno_5, aluno_6, aluno_7, aluno_8, aluno_9,
    aluno_10;

    // Alimentando a estrutura criada com os
    cadastros (nome,e-mail,ru)
```

```
// Primeiro cadastro

strcpy_s(aluno_1.nome, "Leandro Roberto de SOuza
Freitas \n");

strcpy_s(aluno_1.email,
"leandro.freitas@colegiocomercial.net \n");

aluno_1.Ru = 3462061;

//Segundo Cadastro

strcpy_s(aluno_2.nome, "Jose Da Silva \n");

strcpy_s(aluno_2.email,
"JoseSilva231@gmail.com\n");

aluno_2.Ru = 3262062;

// Terceiro Cadastro

strcpy_s(aluno_3.nome, "Luan Pereira \n");

strcpy_s(aluno_3.email, "luanpereira@gmail.com
\n");

aluno_3.Ru = 3362063;

// Quarto Cadastro

strcpy_s(aluno_4.nome, "Maria Pereira \n");

strcpy_s(aluno_4.email, "mariapereira@gmail.com
\n ");

aluno_4.Ru = 3562064;

// Quinto cadastro

strcpy_s(aluno_5.nome, "Eduarda Fernandes \n");

strcpy_s(aluno_5.email,
```

```
"eduardafernandes@gmail.com \n");
aluno_5.Ru = 3662065;

// Sexto Cadastro
strcpy_s(aluno_6.nome, "Marcela Fernandes \n");
strcpy_s(aluno_6.email,
"marcela_fernandes@gmail.com \n");
aluno_6.Ru = 3762066;

// Setimo cadastro
strcpy_s(aluno_7.nome, "Izabela Alves \n");
strcpy_s(aluno_7.email,
"izabelaalves@gmail.com\n");
aluno_7.Ru = 3862067;

// Oitavo cadastro
strcpy_s(aluno_8.nome, "Alef Oliveira \n");
strcpy_s(aluno_8.email,
"alefoliveira@gmail.com\n");
aluno_8.Ru = 3962068;

// Nono Cadastro
strcpy_s(aluno_9.nome, "Ana Zilda \n");
strcpy_s(aluno_9.email, "ana@gmail.com \n");
aluno_9.Ru = 3962069;

// Decimo cadastro
strcpy_s(aluno_10.nome, "Carlos Roberto \n");
```

```
strcpy_s(aluno_10.email, "carlos- roberto@gmail.com\n");

aluno_10.Ru = 4062069;

// Menu Inicial

while (1) {

printf("Cadastro de aluno \n");

printf("Digite a opcao desejada: \n\n");

printf("1 Mostrar todos os alunos cadastrados \n");

printf("2 Buscar aluno \n");

printf("3 Sair \n");

scanf_s("%d", &opcao);

// Utilizando o switch case para fazer a busca das opções cadastradas

switch (opcao) {

case 1:

printf("%s" "%s" "%d \n", aluno_1.nome, aluno_1.email, aluno_1.Ru);

printf("%s" "%s" "%d \n", aluno_2.nome, aluno_2.email, aluno_2.Ru);

printf("%s" "%s" "%d \n", aluno_3.nome, aluno_3.email, aluno_3.Ru);

printf("%s" "%s" "%d \n", aluno_4.nome, aluno_4.email, aluno_4.Ru);

printf("%s" "%s" "%d \n", aluno_5.nome,
```

```
aluno_5.email, aluno_5.Ru);

printf("%s" "%s" "%d \n", aluno_6.nome,
aluno_6.email, aluno_6.Ru);

printf("%s" "%s" "%d \n", aluno_7.nome,
aluno_7.email, aluno_7.Ru);

printf("%s" "%s" "%d \n", aluno_8.nome,
aluno_8.email, aluno_8.Ru);

printf("%s" "%s" "%d \n", aluno_9.nome,
aluno_9.email, aluno_9.Ru);

printf("%s" "%s" "%d \n", aluno_10.nome,
aluno_10.email,
aluno_10.Ru);

system("pause");
system("cls");
break;

//Fazendo a busca pelo RU para verificar se o aluno
tem cadastro

case 2:

printf("Digite o RU do aluno: ");

scanf_s("%d", &RuALuno);

if (RuALuno == aluno_1.Ru)
{
printf("%s" "%s" "%d \n",
aluno_1.nome, aluno_1.email, aluno_1.Ru);
```



```
system("pause");
system("cls");
}
else if (RuALuno == aluno_2.Ru)
{
printf("%s" "%s" "%d \n", aluno_2.nome,
aluno_2.email, aluno_2.Ru);
system("pause");
system("cls");
}
else if (RuALuno == aluno_3.Ru)
{
printf("%s" "%s" "%d \n", aluno_3.nome,
aluno_3.email, aluno_3.Ru);
system("pause");
system("cls");
}
else if (RuALuno == aluno_4.Ru)
{
printf("%s" "%s" "%d \n", aluno_4.nome,
aluno_4.email, aluno_4.Ru);
system("pause");
system("cls");
}
```

```
else if (RuALuno == aluno_5.Ru)
{
printf("%s" "%s" "%d \n", aluno_5.nome,
aluno_5.email, aluno_5.Ru);

system("pause");
system("cls");
}

else if (RuALuno == aluno_6.Ru)
{
printf("%s" "%s" "%d \n",
aluno_6.nome, aluno_6.email, aluno_6.Ru);
system("pause");
system("cls");
}

else if (RuALuno == aluno_7.Ru)
{
printf("%s" "%s" "%d \n", aluno_7.nome,
aluno_7.email, aluno_7.Ru);

system("pause");
system("cls");
}

else if (RuALuno == aluno_8.Ru)
{
```

```
printf("%s" "%s" "%d \n", aluno_8.nome,
aluno_8.email, aluno_8.Ru);

system("pause");

system("cls");

}

else if (RuALuno == aluno_9.Ru)

{

printf("%s" "%s" "%d \n", aluno_9.nome,
aluno_9.email, aluno_9.Ru);

system("pause");

system("cls");

}

else if (RuALuno == aluno_10.Ru)

{

printf("%s" "%s" "%d \n", aluno_10.nome,
aluno_10.email, aluno_10.Ru);

system("pause");

system("cls");

}

break;

// Se o Ru não tiver cadastro cai direto nessa
opção e exibe uma mensagem para o usuario

case 3:

return 0;
```

```
default:

printf("Ru nao cadastrado!!! \n\n ");

system("pause");

system("cls");

    }

}

}
```

Código rodando (Tela Inicial)



The screenshot shows a Windows command prompt window titled "C:\Users\55359\Source\Repos\Trabalho_Busca_Alunos\x64\Debug\Trabalho_Busca_Alunos.exe". The window displays the following text:

```
Cadastro de aluno
Digite a opcao desejada:

1 Mostrar todos os alunos cadastrados
2 Buscar aluno
3 Sair
```

Código rodando (Busca Pelo RU)

C:\Users\55359\Source\Repos\Trabalho_Busca_Alunos\x64\Debug\Trabalho_Busca_Alunos.exe

Cadastro de aluno

Digite a opção desejada:

1 Mostrar todos os alunos cadastrados

2 Buscar aluno

3 Sair

2

Digite o RU do aluno: 3462061

Leandro Roberto de Souza Freitas

leandro.freitas@colegiocomercial.net

3462061

Pressione qualquer tecla para continuar. . .