



CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER
ESCOLA SUPERIOR POLITÉCNICA
ANALISE E DESENVOLVIMENTO DE SISTEMAS
DISCIPLINA DE ESTRUTURA DE DADOS

ATIVIDADE PRÁTICA

ALYSSON - 3050279
VINICIUS BORIN

CURITIBA - PARANÁ
2021

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

```
#include <stdlib.h> // inclui as bibliotecas
#include <stdio.h>
#include <locale.h>
#include <string.h>

void inicializarTabela(); // Inicializa a tabela
void inserir(); // inserir um aluno na tabela.
void listarTabela(); // listar todos os alunos da tabela.
int criarCodigoHash(int numeroRU); // função para gerar um código hash para cada elemento da tabela.

#define M 15 // "M" para representar o tamanho da tabela

// estrutura dadosAluno com nome, matrícula e email
typedef struct
{
    int ruAluno;
    char nome[45];
    char email[45];
}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, 45 - 1, stdin);
    printf("Digite o e-mail do Aluno:\n");
    fgets(Aluno.email, 45 - 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"); // prepara o código para portugues.

    int opcao, numeroRU;
    dadosAluno* Aluno;

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

    // menu de opções
    do
    {
        system("cls");
        printf("SISTEMA DE CADASTRO DE ALUNOS \n\n");
        printf("\tMENU PRINCIPAL\n\n");
        printf("(1) - Cadastrar novo aluno\n");
        printf("(2) - Procurar aluno por RU\n");
        printf("(3) - Listar todos os alunos\n");
        printf("(4) - Sair\n\n");
        printf("Digite a opção:\n");
        scanf_s("%d", &opcao);

        switch (opcao)
        {
            case 1:

```

```

        system("cls");
        printf("SISTEMA DE CADASTRO DE ALUNOS \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 \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.
        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 nao encontrado!\n");
            system("pause");
        }
        break;
    case 3:
        listarTabela(); // listagem dos alunos cadastrados.
        break;
    case 4:
        system("cls"); // encerra o código.
        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. 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;
}

```

```

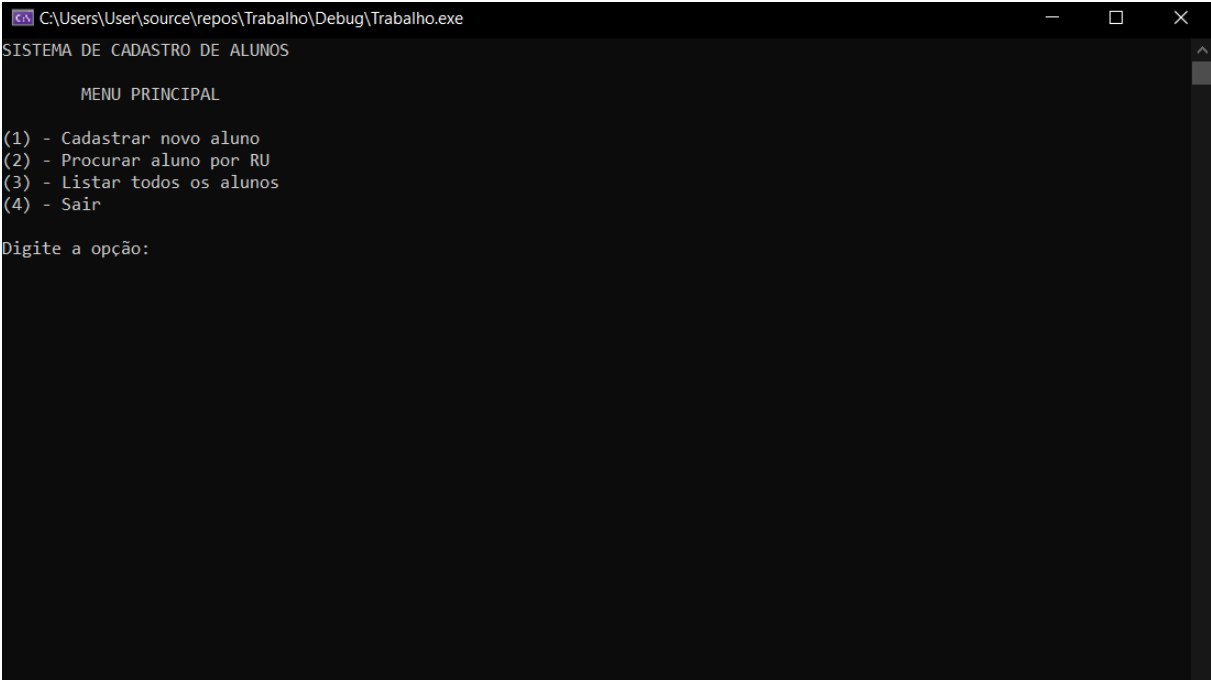
}

//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;
}

```

Tela principal



The screenshot shows a Windows command prompt window titled "C:\Users\User\source\repos\Trabalho\Debug\Trabalho.exe". The output of the program is as follows:

```

SISTEMA DE CADASTRO DE ALUNOS

      MENU PRINCIPAL

(1) - Cadastrar novo aluno
(2) - Procurar aluno por RU
(3) - Listar todos os alunos
(4) - Sair

Digite a opção:

```

Inserir aluno no sistema

```
C:\Users\User\source\repos\Trabalho\Debug\Trabalho.exe
SISTEMA DE CADASTRO DE ALUNOS

    MENU DE INSERÇÃO DE ALUNO

Digite o RU do aluno:
3050279
Digite o nome do Aluno:
Alysson
Digite o e-mail do Aluno:
alyssonduma@hotmail.com

Dados inseridos com sucesso!

RU:      3050279
Aluno:   Alysson
E-mail   alyssonduma@hotmail.com
Pressione qualquer tecla para continuar. . .
```

Listagem de alunos cadastrados

```
C:\Users\User\source\repos\Trabalho\Debug\Trabalho.exe
SISTEMA DE CADASTRO DE ALUNOS UNINTER

    MENU DE LISTAGEM

RU:      7654321
Nome:    pedro
E-mail   pedro@email.com

RU:      7894561
Nome:    leticio
E-mail   leticio@email.com

RU:      1472583
Nome:    paulo
E-mail   paulo@email.com

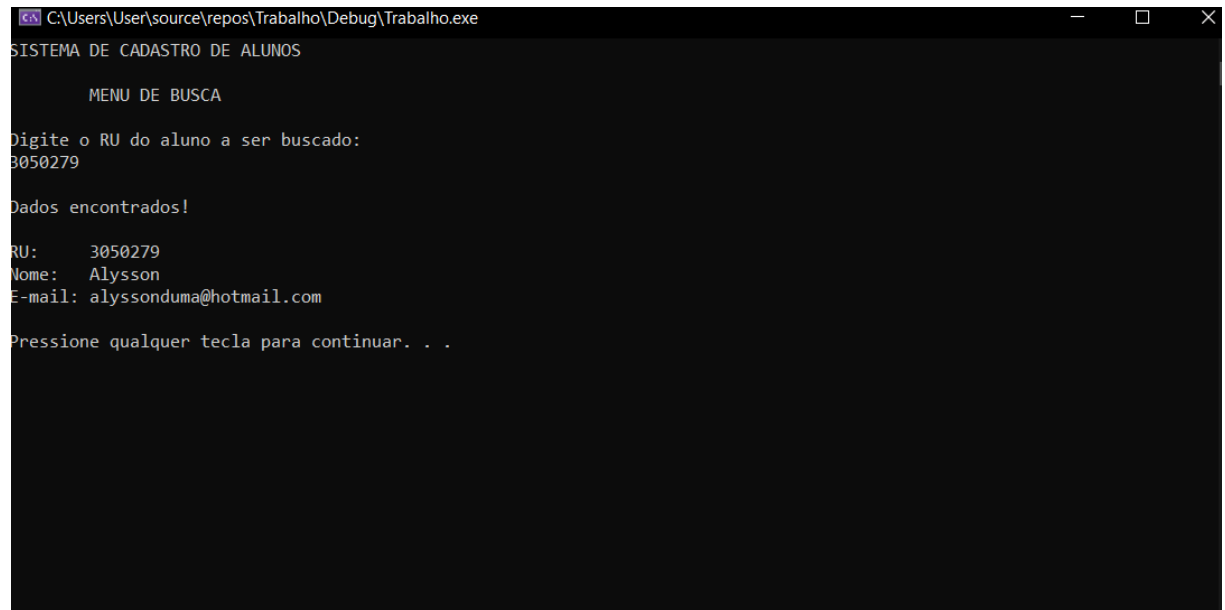
RU:      1234567
Nome:    joao
E-mail   joao@email.com

RU:      7418529
Nome:    maria
E-mail   maria@email.com

RU:      3050279
Nome:    Alysson
E-mail   alyssonduma@hotmail.com

Pressione qualquer tecla para continuar. . .
```

Teste de busca pelo RU



A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\User\source\repos\Trabalho\Debug\Trabalho.exe. The window contains the following text:

```
SISTEMA DE CADASTRO DE ALUNOS

MENU DE BUSCA

Digite o RU do aluno a ser buscado:
3050279

Dados encontrados!

RU:      3050279
Nome:    Alysson
E-mail:  alyssonduma@hotmail.com

Pressione qualquer tecla para continuar. . .
```