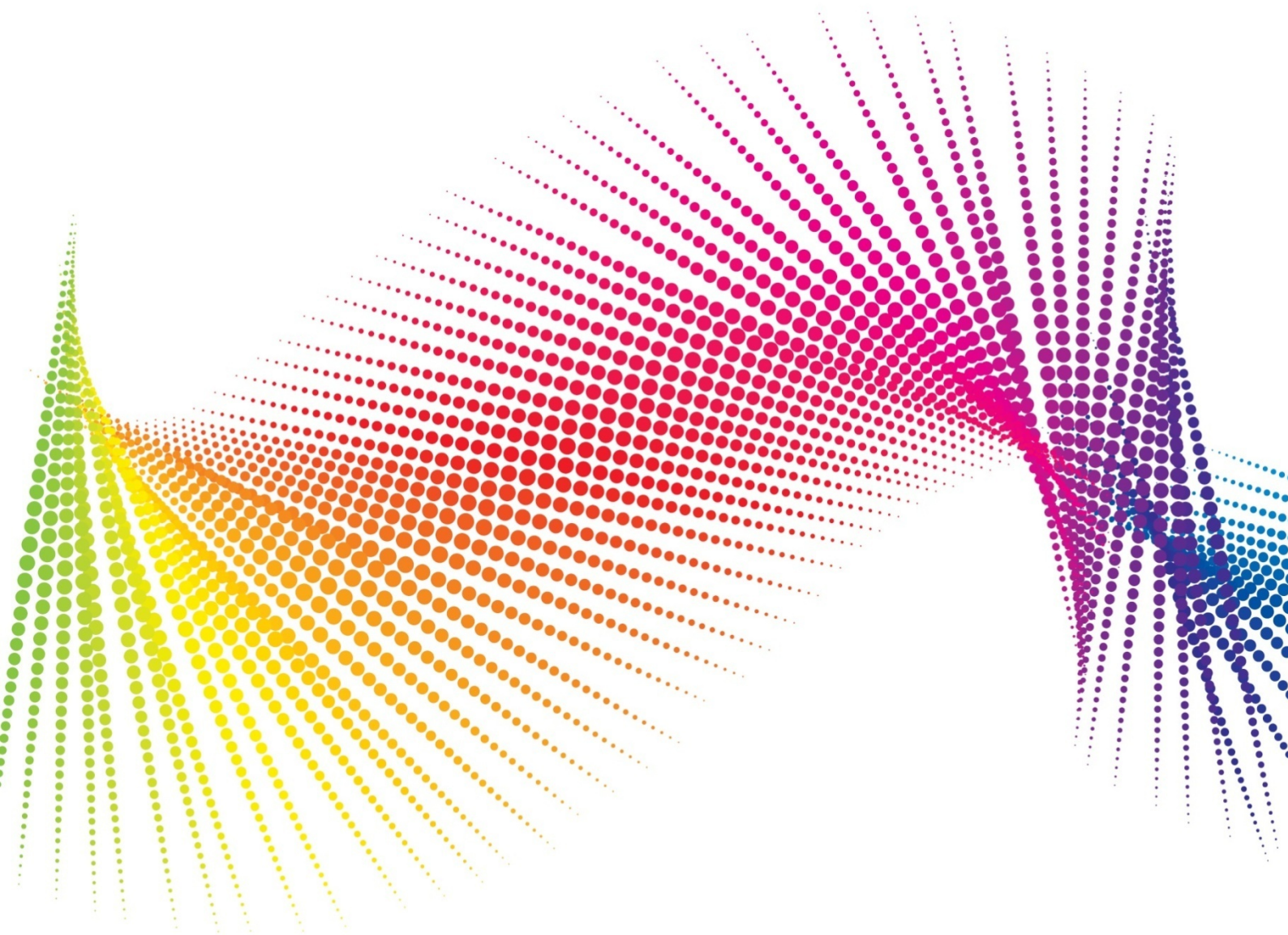


# Estrutura de Dados

Aula 18



Este material é parte integrante da disciplina oferecida pela UNINOVE.

O acesso às atividades, conteúdos multimídia e interativo, encontros virtuais, fóruns de discussão e a comunicação com o professor devem ser feitos diretamente no ambiente virtual de aprendizagem UNINOVE.

Uso consciente do papel.

Cause boa impressão, imprima menos.

## Aula 18: Arquivos, tabelas, ordenação e pesquisa

**Objetivo:** Aprender a ler os dados de um arquivo, armazená-los em tabelas, ordená-los e fazer pesquisa.

### Exemplo de um programa que trabalha com arquivo texto, tabela, método da ordenação e pesquisa binária

Caro aluno, este programa tem intenção bem didática, então, vamos inserir comentários ao longo do código explicando o que está sendo feito, ok? Preste bastante atenção para que consiga entender bem!

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

Cria um tipo registro com nome de uma pessoa e sua nota.

```
struct pessoa{
    char nome[40];
    float nota;
};
```

Cria um vetor de registro com 1000 células, ou seja, pode armazenar até 1000 notas.

```
struct pessoa dados[1000];
char arq[]="teste.txt";
int linhas=0;
```

Cria 2 variáveis externas. Elas existirão em qualquer lugar do programa.

```
void grava();
void le();
void ordena();
void insere();
void lista();
int pesquisa();
int pesquisabinaria(char *,int , int );
int menu();
```

Protótipos das funções que usaremos. Há funções que retornam valor e funções que não retornam valor.

```
main()
{
```

Laço infinito.

```
    int i;
    for(;;){
        i=menu();
        switch(i){
```

Assim que entra no laço é chamado um menu. Ao escolher a opção correta ele chama a respectiva função que executa o que foi selecionado.

```

        case 1:insere(); break;
        case 2:le(); break;
        case 3:grava(); break;
        case 4:pesquisa(); break;
        case 5:ordena(); break;
        case 6:lista(); break;
        case 7:exit(1);
    }//fim switch
} //fim for
} //fim main

```

```

int menu()
{
    int i,f;
    system("cls");
    printf("\t\t MENU\n\n\n\n");
    printf("\t 1. Insere Nomes\n");
    printf("\t 2. Le Arquivo\n");
    printf("\t 3. Grava\n");
    printf("\t 4. Pesquisa\n");
    printf("\t 5. Ordena\n");
    printf("\t 6. Lista\n");
    printf("\t 7. Sai\n");
    printf("\n Entre com a opcao:");
    do{
        fflush(stdin);
        f=scanf("%d",&i);
    }while(!f);
    return i;
}

```

Menu com as respectivas opções.

Temos aí um laço repita (do-while). Enquanto o usuário não digitar um número ele não sai do laço.

```

void lista()
{
    int i;
    system("cls");
    printf("\nListagem...");
    for(i=0;i<linhas ;i++)
        if(dados[i].nome[0])
            printf("\nNome%d)%s -
.2f\n",i+1,dados[i].nome,dados[i].nota);

    system("pause");
} //fim lista
void insere()

```

Função que lista os nomes e notas cadastrados nos registros.

O laço varre o vetor de registros e mostra somente registros que tenham um nome.

A função que insere os nomes e notas.

```

{
    int i=0;
    system("cls");
    printf("\n Entre com alguns nomes. tecle apenas enter sem
digitar nada para sair:");
    for(; i){
        printf("\nNome %d:",i+1);
        fflush(stdin);
        gets(dados[linhas].nome);
        if(!dados[linhas].nome[0])
            break;
        printf("\n Nota:");
        scanf("%f",&dados[linhas].nota);
        linhas++;
        i++;
    }//fim for
}

```

Pegando um nome digitado e armazenando no registro. A variável linhas é global para facilitar a atualização dos índices.

Quando não for digitado nada a condição vai para o break e sai do laço encerrando a inserção.

```

void grava()
{
    FILE * fp;
    int i=0;
    //abrindo o arquivo no modo gravação
    if((fp=fopen(arq,"wb"))==NULL){
        printf("\n Nao pode abrir %s\n",arq);
        exit(1);
    }
    printf("\n gravando...");
    for(i=0;i<linhas;i++)
        fwrite(&dados[i],sizeof(struct pessoa),1,fp);
    fclose(fp);
}

```

Esta é a função que grava cada registro no arquivo. Os dados são armazenados no modo binário.

```

void le()
{
    FILE *fp;
    int i;
    //agora abrindo de novo o arquivo para leitura
    if((fp=fopen(arq,"rb"))==NULL){
        printf("\n Nao pode abrir %s\n",arq);
        exit(1);
    }
    system("cls");
}

```

```

printf("\n Listando ...\n\n");
linhas=0;
while(!feof(fp)){
    fread(&dados[linhas],sizeof(struct pessoa),1,fp);
    if(dados[linhas].nome[0]) {
        printf("\n%d)          %s          -          %.2f\n",
linhas+1,dados[linhas].nome,dados[linhas].nota);
    }
    linhas++;
} //fim while
fclose(fp);
printf("\n Fim ...\n\n");
system("pause");
}

void ordena()
{
    int i,j;
    struct pessoa temp;
    for(i=0;i <linhas ; i++)
        for(j=i+1;j<linhas;j++)
            if(strcmp(dados[i].nome,dados[j].nome) > 0){
                temp = dados[i];
                dados[i]=dados[j];
                dados[j]=temp;
            }
} //fim le

int pesquisa()
{
    char nome[40];
    int achou;
    system("cls");
    printf("\n\n Pesquisa\n\n");
    printf("\nEntre com o nome a pesquisar:");
    fflush(stdin);
    gets(nome);
    ordena();
    achou = pesquisabinaria(nome,0,linhas);
    if(achou != -1)
        printf("\n      Nome      encontrado:\nNome:      %s\nNota:
%.2f\n",dados[achou].nome,dados[achou].nota);
    else printf("\n %s nao encontrado\n",nome);
    system("pause");
}

```

Esta é a função que lê cada registro no arquivo. Os dados são lidos no modo binário.

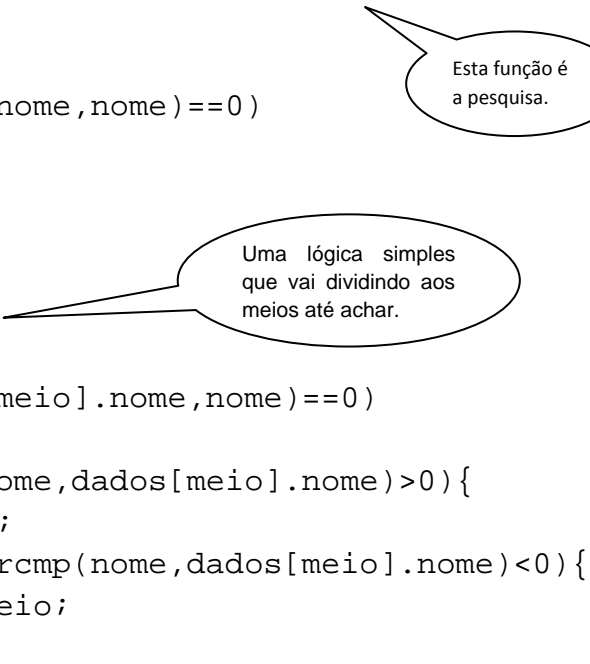
Esta é a função que ordena em ordem alfabética os registros.

Esta função chama a pesquisa binária e mostra se achou ou não.

```
}

int pesquisabinaria(char *nome,int i, int j)
{
    int meio,achou=0;
    if(strcmp(dados[i].nome,nome)==0)
        return i;

    while(1){
        meio=(i+j)/2;
        if(meio==i)
            return -1;
        if(strcmp(dados[meio].nome,nome)==0)
            return meio;
        else if(strcmp(nome,dados[meio].nome)>0){
            i = meio;
        }else if(strcmp(nome,dados[meio].nome)<0){
            j = meio;
        }
    }
    //fim while
    return achou;
} //fim pesquisabinaria
```



Esta função é a pesquisa.

Uma lógica simples que vai dividindo aos meios até achar.

Agora, caro aluno, vamos praticar mais resolvendo os exercícios propostos. **Leia a lista, resolva os exercícios e verifique seu conhecimento.** Caso fique alguma dúvida, leve a questão ao Fórum e divida com seus colegas e professor.

## REFERÊNCIAS

SCHILDT, H. *C Completo e total*. São Paulo: Makron Books, 1997.

TENEMBAUM, Aaron M., et al. *Estruturas de dados usando C*. São Paulo: Pearson Makron Books, 1995.