



UNIVERSIDADE
FEDERAL DE
SERGIPE



A r q u i v o

PROGRAMAÇÃO IMPERATIVA



Situação Problema

Aplicação

Escrever programa em C para ler nome, notas e frequência dos 80 alunos do projeto “AprendaProgramação”, calcular a média destes e emitir em tela a situação final da turma. São três notas, e são aprovados todos os que ficam com média acima da média da turma e têm frequência acima de 65%. Deve ser exibido o nome de cada aluno, seguido das notas, média e da situação final: APROVADO ou REPROVADO.

Como manter os dados desse programa considerando que o curso ocorre ao longo de 18 semanas, as inscrições ocorrem antes do início das aulas, a 1ª nota é definida na 6ª semana, a 2ª nota, na 12ª semana e a 3ª nota na 18ª semana?



ARQUIVO

Definição

- Os arquivos podem ser entendidos como coleções de registros; e residem em memória secundária, ou seja, memória não volátil.
- Portanto são diferentes dos itens de armazenamento de dados estudados até então: **registros**, vetores, matrizes; os quais não preservam os dados neles depositados após o término da execução dos programas que os definem.

ARQUIVO

Vantagens de Uso

São vantagens de uso de arquivos:

- armazenamento permanente de dados;
- possibilidade de armazenamento de grande quantidade de dados;
- disponibilidade dos dados para vários programas.

ARQUIVO

Tipos de Arquivos em C

Em C há dois tipos de arquivos:

- 1. Arquivo texto** – que armazena caracteres os quais podem ser exibidos diretamente em tela e modificados através do uso de editor de texto.
- 2. Arquivo binário** – que contém sequência de bits e obedece a “regras” do programa que o gerou, a exemplo dos c, py, pptx, docx.

São objetos de estudo desta disciplina os **arquivos binários**.

ARQUIVO

Sintaxe de Declaração

Sob o ponto de vista lógico, os **arquivos** binários em C podem ser entendidos como **coleções de registros** e são declarados:

```
FILE *<nomeArquivo>;
```

Onde:

- `nomeArquivo` – nome do arquivo (interno), o qual deve ser precedido por `*`;
- `FILE` – palavra reservada, em maiúsculo.

ARQUIVO

Aplicação

Considerando, por exemplo,
a necessidade de cadastrar
os pacientes a serem
atendidos por um
determinado médico; onde
as consultas são marcadas
com antecedência de
semanas, até meses.



*Um array de
registros poderia
ser usado para
atender a este
propósito?*

ARQUIVO

Aplicação

Considerando a necessidade de manter a agenda médica por meio de arquivos, este poderia ser composto por registros estruturados como segue:

```
struct TpMed{  
    char Nome[21];  
    char Convenio[11],  
        Fone[12]; //com DDD  
    float Pago;  
    int Retorno; //1 retorno 0 primeira vez  
} Consulta;
```


ARQUIVO

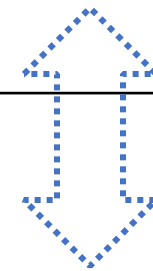
Lembrando REGISTRO

```
struct <nomeDoTipo>
{
    <tipo1> <campo1>[,...<campo2>];
    <tipo2> <campo3>[...];
    [...]
} <IdentificaVariavel>[...];
```

Os registros correspondem a estruturas heterogêneas de armazenamento de dados.

Os arquivos são compostos por registros.

Nome	Zuleika
Telefone	7932574923
Financiamento	17.000,00
Tipo	A
Fidelidade	26
Cliente	



Declaração

```
struct TpCliente{
    char Nome[21];
    char Telefone[12];
    float Financiamento;
    char Tipo;//aposentado
    int Fidelidade; //anos
} Cliente;
```

```

struct TpMed{
    char Nome[21];
    char Convenio[11],
        Fone[12]; // com DDD
    float Pago;
    int Retorno; //1 retorno 0 primeira vez
} Consulta;

```

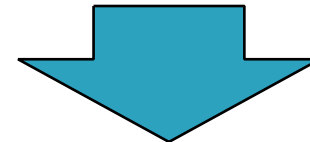
```
FILE *Arquivo;
```

0	≡
1	≡
2	≡
3	≡
4	≡
5	≡
6	≡
7	≡
...	
N	≡
	#

Arquivo

ARQUIVO

Esboço



Nom e	Convênio	Fone	Pago	Retorno
<i>Leide</i>	<i>PoliMed</i>	<i>793...</i>	<i>185,50</i>	<i>1</i>

Marca de fim de arquivo.

ARQUIVO

Função para Abrir Arquivo

- Para gerenciamento dos dados mantidos em arquivos, são necessários comandos específicos; de manipulação de arquivos.
- Um destes refere-se à abertura e associação a nome externo.
- O `fopen` é o procedimento que possibilita a abertura e associação a nome externo do arquivo.

`<arqLógico>=fopen(<arqFísico>,<modo de abrir>);`

ARQUIVO

Nome Externo

- Os arquivos de dados, agora estudados, existem independente do programa que o manipula (são dois programas: o fonte e o de dados). Portanto, os arquivos de dados apresentam nome externo, através do qual, por exemplo, outros programas os identificam.
- Através do nome externo o arquivo mantém-se disponível para outros programas.

 Farmacos.dat PaguePouco.c

ARQUIVO

Modos de Abrir

São modos de abrir arquivos binários em C:

- `r+b` – o arquivo deve existir, **permite apenas ler e alterar os dados existentes**; na abertura ponteiro posicionado no início do arquivo.
- `w+b` – permite leitura e escrita de dados do arquivo, se na abertura o arquivo não existe, este é criado; **se existir, o arquivo é recriado (arquivo anterior apagado)**; na abertura ponteiro posicionado no início do arquivo.
- `a+b` – permite leitura e escrita de dados do arquivo, se na abertura o arquivo não existe, este é criado vazio; se existir, **permite acrescentar dados**; na abertura ponteiro posicionado no fim do arquivo.

ARQUIVO

Modos de Abrir

Modo	Existência do Arquivo	Operações sobre Arquivo	Posicionamento do Ponteiro ao Abrir
$r + b$	arquivo deve existir	leitura e alteração dos dados	ponteiro no início do arquivo ao abrir
$w + b$	<ul style="list-style-type: none">• se não existir, é criado• se existir, é apagado	leitura e escrita dos dados	ponteiro no início do arquivo ao abrir
$a + b$	<ul style="list-style-type: none">• se não existir, é criado• se existir, permite adicionar dados	leitura e escrita de dados	ponteiro no início do arquivo ao abrir

*Convém abrir um arquivo onde se mantém contatos
(nome, telefone, e-mail) com o modo $w + b$?*



ARQUIVO

Modos de Abrir

Modo	Existência do Arquivo	Operações sobre Arquivo	Posicionamento do Ponteiro ao Abrir
r + b	arquivo deve existir	leitura e alteração dos dados	ponteiro no início do arquivo ao abrir
w + b	<ul style="list-style-type: none">• se não existir, é criado• se existir, é apagado	leitura e escrita dos dados	ponteiro no início do arquivo ao abrir
a + b	<ul style="list-style-type: none">• se não existir, é criado• se existir, permite adicionar dados	leitura e escrita de dados	ponteiro no início do arquivo ao abrir

Diante da necessidade de criação de um cadastro de alunos interessados em bolsa PIBIC, com inclusão, e emissão de relatório, vale usar o modo de abertura a + b?



ARQUIVO

Modos de Abrir

Modo	Existência do Arquivo	Operações sobre Arquivo	Posicionamento do Ponteiro ao Abrir
r + b	arquivo deve existir	leitura e alteração dos dados	ponteiro no início do arquivo ao abrir
w + b	<ul style="list-style-type: none">• se não existir, é criado• se existir, é apagado	leitura e escrita dos dados	ponteiro no início do arquivo ao abrir
a + b	<ul style="list-style-type: none">• se não existir, é criado• se existir, permite adicionar dados	leitura e escrita de dados	ponteiro no início do arquivo ao abrir

Diante da necessidade de criação de um controle de estoque, com inclusão, edição/alteração e emissão de relatório, que modo de abertura aplicar?



ARQUIVO

Função para Abrir Arquivo

```
Arquivo = fopen("Consultas.dat", "a+b");
```

- `"Consultas.dat"` – nome externo, através do qual, mantém-se disponível para outros programas.
- `a+b` – permite leitura e escrita de dados do arquivo, se na abertura o arquivo não existe, este é criado; se existir, **permite acrescentar dados**; na abertura ponteiro posicionado no fim do arquivo.

ARQUIVO

Função para Abrir Arquivo

```
Arquivo = fopen("Consultas.dat", "a+b");
```

- "Consultas.dat" – nome externo, através do qual, mantém-se disponível para outros programas.
- a+b – permite leitura e escrita de dados do arquivo, se na abertura o arquivo não existe, este é criado; se existir, **permite acrescentar dados**; na abertura ponteiro posicionado no fim do arquivo.

E quanto à edição/alteração de dados (na agenda médica), não é necessária? Como disponibilizar essas operações?



ARQUIVO

Função para Abrir Arquivo

```
Arquivo = fopen("Consultas.dat", "a+b") ;
```

- "Consultas.dat" – nome externo.
- a+b – permite leitura e escrita de dados do arquivo, se na abertura o arquivo não existe, este é criado; se existir, permite acrescentar dados.

*O arquivo
Consultas.dat
é criado na pasta
corrente.*



ARQUIVO

Função para Fechar Arquivo

```
fclose (<arqLógico>) ;
```

- Através do comando `fclose` é possível fechar um arquivo.
- Vale esclarecer que a ausência do `fclose` não é identificada nem no processo de compilação, nem no de execução . Porém pode provocar danos nos dados mantidos na estrutura.

ARQUIVO

Função para Fechar Arquivo

```
fclose(<arqLógico>) ;
```

- O processo de fechar um arquivo deve ser efetuado ao final da manipulação deste e uma única vez. Ou seja, não deve ser mantido em laços, assim como o procedimento de abrir arquivos, para evitar perdas de tempo de processamento.

ARQUIVO

Gravando Registro em Arquivos

Para escrever dados em arquivo binário em C,
usar:

```
fwrite(<dado>, <tamanhoDado>, <quant>, <arqLógico>);
```

Onde:

- `<dado>` – dado de qualquer tipo a ser escrito no arquivo;
- `<tamanhoDado>` – tamanho do dado, para tanto deve-se fazer uso da função `sizeof()`.
- `<quant>` – quantidade de dados a ser escrito;
- `< arqLógico >` – nome do arquivo interno.

ARQUIVO

Gravando Registro em Arquivos

```
struct TpMed{  
    char Nome[21];  
    char Convenio[11],  
        Fone[12]; // com DDD  
    float Pago;  
    int Retorno; //1 retorno 0 primeira vez  
} Consulta;
```

```
FILE *Arquivo;
```

0	≡
1	≡
2	≡
3	≡
4	≡
5	≡
6	≡
7	≡
...	
N	≡
	#

Arquivo

ARQUIVO Esboço



Nom e	Convênio	Fone	Pago	Retorno
<u>Leide</u>	<u>PoliMed</u>	793...	185,50	1

Marca de fim de arquivo.

Para gerenciamento de um arquivo é mantido um ponteiro numérico que indica a posição (corrente) do registro a ser manipulado.

Para escrever dados em arquivo binário em C:

```
fwrite (<dado>,<tamanhoDado>,<quant>,<arqLógico>) ;
```

```
int Tamanho=42*sizeof(char)+sizeof(float)+sizeof(int);
```

```
fwrite (&Consulta,Tamanho,1,Arquivo) ;
```

ARQUIVO

Aplicação

```
FILE *Arquivo;
```

```
struct TpMed{  
    char Nome[20];  
    char Convenio[10],  
        Fone[10];  
    float Pago;  
    int Retorno;  
}Consulta;
```

```
Arquivo =  
    fopen ("Dados.dat", "a+b");
```



*Onde a variável
Consulta reside?*

Para manter agenda médica:

```
printf("Paciente: ");  
gets(Consulta.Nome);  
printf("Convenio: ");  
gets(Consulta.Convenio);  
printf("Telefone: ");  
gets(Consulta.Fone);  
printf("Valor: ");  
scanf("%f",&Consulta.Pago);  
printf("Retorno? S/N ");  
scanf("%d",&Consulta.Retorno);  
int Tamanho = 40 * sizeof(char)  
    + sizeof(float) +  
    sizeof(int); // dos campos  
do registro  
fwrite(&Consulta,Tamanho,1,  
    Arquivo);  
  
...  
fclose(Arquivo);
```


ARQUIVO

Aplicação

```
FILE *Arquivo;
```

```
struct TpMed{  
    char Nome[20];  
    char Convenio[10],  
        Fone[10];  
    float Pago;  
    int Retorno;  
}Consulta;  
  
Arquivo =  
    fopen ("Dados.dat", "a+b");
```



*Qual o nome externo
do arquivo?*

Para manter agenda médica:

```
printf("Paciente: ");  
gets(Consulta.Nome);  
printf("Convenio: ");  
gets(Consulta.Convenio);  
printf("Telefone: ");  
gets(Consulta.Fone);  
printf("Valor: ");  
scanf("%f",&Consulta.Pago);  
printf("Retorno? S/N ");  
scanf("%d",&Consulta.Retorno);  
int Tamanho = 40 * sizeof(char)  
    + sizeof(float) +  
    sizeof(int); // dos campos  
do registro  
fwrite (&Consulta,Tamanho,1,  
    Arquivo);  
  
...  
fclose (Arquivo);
```

ARQUIVO

Aplicação

```
FILE *Arquivo;
```

```
struct TpMed{
```

```
    char Nome[20];
```

```
    char Convenio[10],
```

```
        Fone[10];
```

```
    float Pago;
```

```
    int Retorno;
```

```
    }Consulta;
```

```
Arquivo =
```

```
    fopen ("Dados.dat", "a+b");
```



*Quais os dados
mantidos no arquivo?*

Para manter agenda médica:

```
printf("Paciente: ");
```

```
gets(Consulta.Nome);
```

```
printf("Convenio: ");
```

```
gets(Consulta.Convenio);
```

```
printf("Telefone: ");
```

```
gets(Consulta.Fone);
```

```
printf("Valor: ");
```

```
scanf("%f",&Consulta.Pago);
```

```
printf("Retorno? S/N ");
```

```
scanf("%d",&Consulta.Retorno);
```

```
int Tamanho = 40 * sizeof(char) +  
    sizeof(float) +  
    sizeof(int); // dos campos  
do registro
```

```
fwrite (&Consulta,Tamanho,1,  
    Arquivo);
```

```
...
```

```
fclose (Arquivo);
```

ARQUIVO

Aplicação

```
FILE *Arquivo;

struct TpMed{
    char Nome[20];
    char Convenio[10],
        Fone[10];
    float Pago;
    int Retorno;
}Consulta;

Arquivo =
    fopen ("Dados.dat", "a+b");
```



*Código suficiente para
manter a agenda?*

Para manter agenda médica:

```
printf("Paciente: ");
gets(Consulta.Nome);
printf("Convenio: ");
gets(Consulta.Convenio);
printf("Telefone: ");
gets(Consulta.Fone);
printf("Valor: ");
scanf("%f",&Consulta.Pago);
printf("Retorno? S/N ");
scanf("%d",&Consulta.Retorno);
int Tamanho = 40 * sizeof(char)
    + sizeof(float) +
    sizeof(int); // dos campos
do registro
fwrite (&Consulta,Tamanho,1,
    Arquivo);

...
fclose (Arquivo);
```

ARQUIVO

Lendo Registro de Arquivos

Para ler dados em arquivo binário em C, usar:

```
fread(<dado>, <tamanhoDado>, <quant>, <arqLógico>);
```

Onde:

- `<dado>` – dado de qualquer tipo a ser escrito no arquivo;
- `<tamanhoDado>` – tamanho do dado, para tanto deve-se fazer uso da função `sizeof()`.
- `<quant>` – quantidade de dados a ser escrito;
- `< arqLógico >` – nome do arquivo interno.

ARQUIVO

Função que Indica Alcance de Fim de Arquivo

Em geral, no processo de inclusão de novos registros, estes devem ser adicionados ao final do arquivo, após o último registro. Para tanto, é útil a função `feof`.

```
int feof(<arqLógico>);
```

A função `feof` retorna não zero (`true`) quando se tenta executar algo sobre a marca de fim do arquivo e zero (`false`) quando não encontrado.

```
if (!feof(Arquivo)) //se não for fim de arquivo
{ ...
}
```

Caso seja necessário manipular um registro a partir de sua posição no arquivo, é útil procedimento `fseek`.

```
int fseek(<arqLógico>, <salto>, <origem>);
```

Onde:

- `arqLógico` – arquivo onde será efetuado o posicionamento;
- `salto` – do tipo `long`, em bytes, correspondente o quanto se deseja deslocar no arquivo, se positivo avança em direção ao final do arquivo, se negativo volta ao início;
- `origem` – define o ponto de início do deslocamento, se 0 (SEEK_SET) o deslocamento inicia na início do arquivo, se 1 (SEEK_CUR) inicia da posição corrente, se 2 (SEEK_END) do final do arquivo.

0	≡
1	≡
2	≡
3	≡
4	≡
5	≡
6	≡
7	≡
...	
N	≡
	#

Arquivo

ARQUIVO

Aplicação

```
FILE *Arquivo;
```

```
struct TpMed{  
    char Nome[20];  
    char Convenio[10],  
        Fone[10];  
    float Pago;  
    int Retorno;  
}Consulta;
```

```
Arquivo = fopen("Dados.dat", "a+b");
```

Para manter agenda médica:

...

```
fseek(Arquivo, 0, 0);  
fread(&Consulta, Tamanho, 1, Arquivo);  
printf("Paciente:  
        %s\n", Consulta.Nome);  
printf("Convenio:  
        %s\n", Consulta.Convenio);  
printf("Telefone:  
        %s\n", Consulta.Fone);  
printf("Valor: %f", Consulta.Pago);  
printf("Retorno:  
        %d", Consulta.Retorno);  
fclose(Arquivo);
```



*O que é exibido
em tela?*

ARQUIVO

Aplicação

```
FILE *Arquivo;
```

```
struct TpMed{  
    char Nome[20];  
    char Convenio[10],  
        Fone[10];  
    float Pago;  
    int Retorno;  
}Consulta;
```

```
Arquivo = fopen("Dados.dat", "a+b");
```

Para manter agenda médica:

...

```
fseek(Arquivo, 0, 0);  
fread(&Consulta, Tamanho, 1, Arquivo);  
printf("Paciente:  
        %s\n", Consulta.Nome);  
printf("Convenio:  
        %s\n", Consulta.Convenio);  
printf("Telefone:  
        %s\n", Consulta.Fone);  
printf("Valor: %f", Consulta.Pago);  
printf("Retorno:  
        %d", Consulta.Retorno);  
fclose(Arquivo);
```



*Quantos registros
são exibidos?*

ARQUIVO

Aplicação

```
FILE *Arquivo;
```

```
struct TpMed{  
    char Nome[20];  
    char Convenio[10],  
        Fone[10];  
    float Pago;  
    int Retorno;  
}Consulta;
```

```
Arquivo = fopen("Dados.dat", "a+b");
```

Para manter agenda médica:

...

```
fseek(Arquivo, 0, 0);  
fread(&Consulta, Tamanho, 1, Arquivo);  
printf("Paciente:  
        %s\n", Consulta.Nome);  
printf("Convenio:  
        %s\n", Consulta.Convenio);  
printf("Telefone:  
        %s\n", Consulta.Fone);  
printf("Valor: %f", Consulta.Pago);  
printf("Retorno:  
        %d", Consulta.Retorno);  
fclose(rquivo);
```



*O que ajustar
para exibir todos
os registros do
arquivo?*

ARQUIVO

Aplicação

```
FILE *Arquivo;
```

```
struct TpMed{  
    char Nome[20];  
    char Convenio[10],  
        Fone[10];  
    float Pago;  
    int Retorno;  
}Consulta;  
...
```

Para manter agenda médica:

```
fseek(Arquivo, 0, 0);  
do {  
    fread(&Consulta, Tamanho, 1, Arquivo);  
    if (!feof(Arquivo)) {  
        printf("Paciente:  
               %s\n", Consulta.Nome);  
        printf("Convenio:  
               %s\n", Consulta.Convenio);  
        printf("Telefone:  
               %s\n", Consulta.Fone);  
        printf("Valor:  
               %f\n", Consulta.Pago);  
        printf("Retorno:  
               %d\n", Consulta.Retorno);  
        printf("***\n\n");  
    }  
} while (!feof(Arquivo));  
...
```

ARQUIVO

Aplicação

Arquivos devem ser usados em aplicações computacionais que manipulam dados que devem ser mantidos permanentemente:

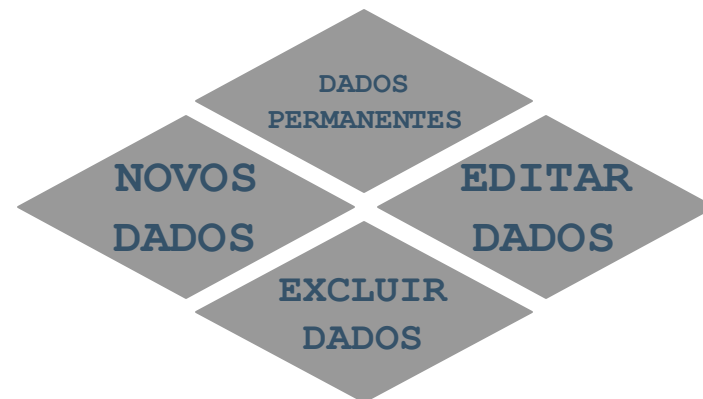
- agenda telefônica
- estoque
- cadastros... notas, alunos, fornecedores

Vale considerar que a memória permanente tem também grande capacidade de armazenamento de dados, mas é lenta em relação à memória principal.

MEDICAMENTOS



Implementar programa
de gerenciamento
dos medicamentos
de uma farmácia.



CADASTRO FARMÁCIA principais dados

```
struct TpFarma{  
    char Nome[21];  
    float Preco;  
    int QEstoque;  
};
```

```
typedef struct TpFarma TpFARMA;
```

```
TpFARMA RgFarma;
```

```
FILE *ArqFarma;
```



```
int main() {
```

```
    ArqFarma=fopen("Famacos.dat", "a+b");
```

```
    char Opcao;
```

```
    do{
```

```
        system("cls");
```

```
        printf("\n\n> > > Pague Pouco < < < \n\n");
```

```
        printf("Que deseja fazer? \n\n");
```

```
        printf("I - Incluir \n");
```

```
        printf("E - Excluir \n");
```

```
        printf("A - Alterar \n");
```

```
        printf("C - Consultar \n");
```

```
        printf("T - Listar Todos \n");
```

```
        printf("S - Sair \n\n");
```

```
        scanf(" %c", &Opcao);
```

```
        Opcao=toupper(Opcao);
```

```
        switch (Opcao) {
```

```
            case 'I': Incluir(); break;
```

```
            case 'E': Excluir(); break;
```

```
            case 'A': Alterar(); break;
```

```
            case 'C': Consultar(); break;
```

```
            case 'T': LTodos(); break;}}
```

```
    while (Opcao != 'S');
```

```
    fclose(ArqFarma);
```

```
    return 0;}
```

CADASTRO FARMÁCIA

principal



```
int main() {
    ArqFarma=fopen("Famacos.dat", "a+b");
    char Opcao;
    do{
        system("cls");
        printf("\n\n> > > Pague Pouco < < < \n\n");
        printf("Que deseja fazer? \n\n");
        printf("I - Incluir \n");
        printf("E - Excluir \n");
        printf("A - Alterar \n");
        printf("C - Consultar \n");
        printf("T - Listar Todos \n");
        printf("S - Sair \n\n");
        scanf(" %c", &Opcao);
        Opcao=toupper(Opcao);
        switch (Opcao) {
            case 'I': Incluir(); break;
            case 'E': Excluir(); break;
            case 'A': Alterar(); break;
            case 'C': Consultar(); break;
            case 'T': LTodos(); break;}}
    while (Opcao != 'S');
    fclose(ArqFarma);
    return 0;}

```



```
void Incluir() {  
    char R;  
    do{  
        //system("cls");  
        printf("*** inclusao ***\n\n");  
        printf("Nome: ");  
        scanf("%s", RgFarma.Nome);  
        fflush(stdin);  
        printf("Preco: ");  
        scanf("%f", &RgFarma.Preco);  
        printf("Estoque: ");  
        scanf("%d", &RgFarma.QEstoque);  
        fseek(ArqFarma, 0, 2);  
        fwrite(&RgFarma, Tamanho, 1, ArqFarma);  
        printf("\nNova inclusao? S/N ");  
        scanf(" %c", &R);  
        R=toupper(R);  
    } while (R!='N');  
    return;}  

```

CADASTRO FARMÁCIA
inclusão

CADASTRO FARMÁCIA

inclusão

```
void Incluir() {  
    char R;  
    do{  
        //system("cls");  
        printf("*** inclusao ***\n\n");  
        printf("Nome: ");  
        scanf("%s", RgFarma.Nome);  
        fflush(stdin);  
        printf("Preco: ");  
        scanf("%f", &RgFarma.Preco);  
        printf("Estoque: ");  
        scanf("%d", &RgFarma.QEstoque);  
        fseek(ArqFarma, 0, 2);  
        fwrite(&RgFarma, Tamanho, 1, ArqFarma);  
        printf("\nNova inclusao? S/N ");  
        scanf(" %c", &R);  
        R=toupper(R);  
    } while (R!='N');  
    return;}  

```

*Os registros (novos) estão
sendo adicionados no início
do arquivo?*



```

...
void Incluir() {
    char R;
    do{
        system("cls");
        printf("*** inclusao ***\n\n");
        printf("Nome: ");
        scanf("%s",RgFarma.Nome);
        printf("Preco: ");
        scanf("%f",&RgFarma.Preco);
        printf("Estoque: ");
        scanf("%d",&RgFarma.QEstoque);
        fseek(ArqFarma,0,2);
        fwrite(&RgFarma,Tamanho,1,ArqFarma)
        printf("\nNova inclusao? S/N ");
        scanf(" %c",&R);
        R=toupper(R);
    }while (R!='N');
    return;}

```

```

struct TpFarma{
    char Nome[21];
    float Preco;
    int QEstoque;};

typedef struct
    TpFarma TpFARMA;

TpFARMA RgFarma;

FILE *ArqFarma;

long int Tamanho=
    20*sizeof(char)+
    sizeof(float)+
    sizeof(int);

```



CADASTRO FARMÁCIA
inclusão

CADASTRO FARMÁCIA

consulta total



```
...  
void LTodos () {  
    //system("cls");  
    printf("*** lista todos ***\n\n");  
    fseek(ArqFarma, 0, 0);  
    do {  
        fread(&RgFarma, Tamanho, 1, ArqFarma);  
        if (!feof(ArqFarma)) {  
            printf("Nome: %s\n", RgFarma.Nome);  
            printf("Valor: %f\n", RgFarma.Preco);  
            printf("Estoque: %d\n", RgFarma.QEstoque);  
            printf("***\n\n");  
        }  
        while (!feof(ArqFarma));  
    }  
    //system("pause");  
    return;  
}
```

CADASTRO FARMÁCIA

consulta total

```
...  
void LTodos () {  
    //system("cls");  
    printf("*** lista todos ***\n\n");  
    fseek(ArqFarma, 0, 0);  
    do {  
        fread(&RgFarma, Tamanho, 1, ArqFarma);  
        if (!feof(ArqFarma)) {  
            printf("Nome: %s\n", RgFarma.Nome);  
            printf("Valor: %f\n", RgFarma.Preco);  
            printf("Estoque: %d\n", RgFarma.QEstoque);  
            printf("***\n\n");  
        }  
        while (!feof(ArqFarma));  
        //system("pause");  
    }  
    return;  
}
```



*Quais registros do
arquivo são exibidos?*

CADASTRO FARMÁCIA

consulta total

```
1.  ...
2.  void LTodos () {
3.      //system("cls");
4.      printf("*** lista todos ***\n\n");
5.      fseek(ArqFarma,0,0);
6.      do {
7.          fread(&RgFarma,Tamanho,1,ArqFarma);
8.          if (!feof(ArqFarma)) {
9.              printf("Nome: %s\n",RgFarma.Nome);
10.             printf("Valor: %f\n",RgFarma.Preco);
11.             printf("Estoque: %d\n",RgFarma.QEstoque);
12.             printf("***\n\n");}}
13.      while (!feof(ArqFarma));
14.      //system("pause");
15.      return; }
```



Se fseek fosse removido da linha 5, quais os efeitos sobre o resultado?



```

void Efetua() {
    system("cls");
    printf("*** efetuar ***\n\n");
    fseek(ArqFarma,0,0);
    printf("Qual farmaco? ");
    char Farmaco[20];
    scanf("%s",Farmaco);
    int Achou=0;
    do {
        fread(&RgFarma,Tamanho,1,ArqFarma);
        if (strcmp(RgFarma.Nome,Farmaco)==0) {
            Achou=1;
            printf("Nome: %s\n",RgFarma.Nome);
            printf("Valor: %f\n",RgFarma.Preco);
            printf("Estoque:
%d\n",RgFarma.QEstoque);}}
        while (!feof(ArqFarma) && (Achou==0));
    if (Achou==0)
        printf("Registro inexistente!");
    system("pause");
    return;}

```

CADASTRO FARMÁCIA

efetua



*Qual a operação
é realizada por
meio desse
subprograma?*

```
void Consultar() {
    system("cls");
    printf("*** consultar ***\n\n");
    fseek(ArqFarma,0,0);
    printf("Qual farmaco? ");
    char Farmaco[20];
    scanf("%s",Farmaco);
    int Achou=0;
    do {
        fread(&RgFarma,Tamanho,1,ArqFarma);
        if (strcmp(RgFarma.Nome,Farmaco)==0) {
            Achou=1;
            printf("Nome: %s\n",RgFarma.Nome);
            printf("Valor: %f\n",RgFarma.Preco);
            printf("Estoque:
%d\n",RgFarma.QEstoque);}}
    while (!feof(ArqFarma)&&(Achou==0));
    if (Achou==0)
        printf("Registro inexistente!");
    system("pause");
    return;}

```



*Havendo 5
medicamento com
mesmo nome?
Quantos são
exibidos?*



Discutir a
implementação
de procedimento
de alteração
(ajuste do preço e
estoque).



CADASTRO FARMÁCIA

alteração

```
long int TArquivo() {  
    fseek(ArqFarma, 0, 2);  
    long int R=ftell(ArqFarma)/Tamanho;  
    return R;}
```

```
void Alterar() {  
    if (TArquivo() != 0) {  
        fclose(ArqFarma);  
        ArqFarma=fopen("Famacos.dat", "r+b");  
        ... Localizará registro!!!  
        ... Alterará registro!!!  
        fclose(ArqFarma);  
        ArqFarma=fopen("Famacos.dat", "a+b");  
    }  
    else {  
        printf("Arquivo Vazio.  Nao existem dados a alterar.");  
        system("pause");  
    }  
    return;}
```



CADASTRO FARMÁCIA

alteração

```
void Alterar() {
```

...Checou que há dados no arquivo!!!

...Abriu arquivo para alterar!!!

```
//system("cls");  
printf("*** alterar ***\n\n");  
fseek(ArqFarma,0,0);  
printf("Qual farmaco? ");  
char Farmaco[20];  
scanf("%s",Farmaco);  
int Achou=0;  
do {  
    fread(&RgFarma,Tamanho,1,ArqFarma);  
    if (strcmp(RgFarma.Nome,Farmaco)==0) {  
        Achou=1;  
        printf("Nome: %s\n",RgFarma.Nome);  
        printf("Valor: %f\n",RgFarma.Preco);  
        printf("Estoque: %d\n",RgFarma.QEstoque);}}  
while (!feof(ArqFarma) && (Achou==0));  
if (Achou==0)  
    printf("Registro inexistente!");  
else{
```

...Alteração será efetuada!!!



CADASTRO FARMÁCIA

alteração

```
void Alterar(){  
    ... Checou que há dados no arquivo!!!  
    ... Abriu arquivo para alterar!!!  
    ... Localizou o registro!!!  
    if (Achou==0)  
        printf("Registro inexistente!");  
    else{  
        printf("Qual o novo preco? \n");  
        scanf("%f",&RgFarma.Preco);  
        printf("Qual a nova quantidade? \n");  
        scanf("%d",&RgFarma.QEstoque);  
        fseek(ArqFarma,-Tamanho,1);  
        fwrite(&RgFarma,Tamanho,1,ArqFarma);  
        printf(">>> Alteracao efetuada com sucesso! <<<\n");  
        //system("pause");  
        fclose(ArqFarma);  
        ArqFarma=fopen("Famacos.dat","a+b");  
    }  
    else{ //if (TArquivo()!=0){  
        printf("Arquivo Vazio.  Nao existem dados a alterar.");  
        //system("pause");  
    }  
    return;}  
}
```



ARQUIVO

Lendo Registro de Arquivos

São vantagens de uso de arquivos:

- armazenamento permanente de dados;
- possibilidade de armazenamento de grande quantidade dados;
- disponibilidade dos dados para vários programas.



Discutir

implementação de
procedimento de
exclusão (lógica).



Discutir se
implementação de
procedimento de
exclusão (lógica) gera
efeito colateral sobre:

a) inclusão

b) consulta total

CADASTRO FARMÁCIA

inclusão

```
void Incluir() {
    char R;
    do{
        system("cls");
        printf("*** inclusao ***\n\n");
        printf("Nome: ");
        scanf("%s",RgFarma.Nome);
        fflush(stdin);
        printf("Preco: ");
        scanf("%f",&RgFarma.Preco);
        printf("Estoque: ");
        scanf("%d",&RgFarma.QEstoque);
        fseek(ArqFarma,0,2);
        fwrite(&RgFarma,Tamanho,1,ArqFarma);
        printf("\nNova inclusao? S/N ");
        scanf(" %c",&R);
        R=toupper(R);
    }while (R!='N');
    return;}

```

Discutir o ajuste do procedimento de inclusão de forma a evitar redundância de dados.





Exercício 1

Implementar no projeto “Cadastro Farmácia” (TestaArquivo.c) as operações a seguir: T03-G02 & T09

- (1) exclusão lógica de fármaco dado pelo usuário, bem como os ajustes dos procedimentos que sofrerem efeitos colaterais destes – consulta total, por exemplo;
- (2) ajuste da inclusão de forma a impossibilitar fármacos repetidos.
- (3) ajustar a consulta total de forma a não exibir os registros excluídos.



Exercícios

2 No "Motos Zondas", implementar histórico de atendimento com data, placa e valor (do atendimento). T02-G01(até J)

3 No "Estacionamento", implementar histórico de saídas com data, placa e valor (pago). T02-G02

4 No dos "Frezers", manter dados a matriz em arquivo (um dado por linha) e em caso de alteração, o arquivo deve ser alterado. T03-G01(até I)

Complementar Estudos...



Fundamentos da Programação de Computadores

Ana Fernanda Gomes Ascencio

Edilene Aparecida Veneruchi de Campos



Capítulo Registros e Arquivos

Complementar Estudos...



Linguagem C: Completa e Descomplicada

André Backes



Capítulo **12** Arquivos

Próximo passo...



*Estudar, praticar, obtendo êxito em
Programação Imperativa e aplicando as
competências adquiridas na vida prática.*