



# Estruturas em C

Higor Neto

Monitoria LPI – 17/05/06




Faça um programa em C que administre clientes de um Banco. O administrador deve poder:

- 1\_ Inserir um cliente;
- 2\_ Modificar um cliente;
- 3\_ Ver os clientes cadastrados;
- 4\_ Deletar um cliente;
- Usar matriz de estruturas para representar os clientes!



## ➤ Atributos dos clientes (Estrutura):

1. Número da conta (Int)
2. Nome do cliente (Char)
3. Endereço do cliente (Char)

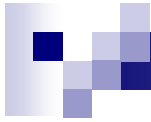


```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
```

```
#define MAX 20
#define MAX_END 30
#define MAX_CLIENTE 50
```

```
int next=0;
int conta=1000;
```

```
struct cliente
{
    char nome[MAX];
    char end[MAX_END];
    int nconta;
};
```



```
struct cliente clientes[MAX_CLIENTE];
```

```
int menu()
```

```
{
```

```
    printf("\n1)-Cadastrar cliente");
```

```
    printf("\n2)-Modificar cliente");
```

```
    printf("\n3)-Ver clientes cadastrados");
```

```
    printf("\n4)-Deletar cliente");
```

```
    printf("\n5)-Sair");
```

```
    printf("\n>> ");
```

```
    return(getche());
```

```
}
```

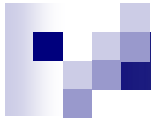


```
void cadastrar( )
```

```
{
```

```
    clientes[next].nconta=conta;
    printf("\nCliente numero %d.",next);
    printf("\nNumero de conta %d",clientes[next].nconta);
    printf("\nNome do cliente:");
    gets(clientes[next].nome);
    printf("\nEndereco do cliente:");
    gets(clientes[next].end);
    printf("\nCliente cadastrado...");
    conta++;
    next++;
    getch();
```

```
}
```




```
int procuracliente(char nome[ ])

{
    int next1=0;
    for(next1=0; next1<=next; next1++)

        {
            if(!strcmp(nome,(clientes[next1].nome)))
                return(next1);
        }

    return(-1);
}
```



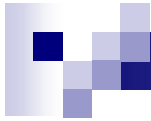
```
void modificar()
{
    char nome1[MAX];
    int cliente1;
    init:
    printf("\nNome do cliente a ser modificado:");
    scanf("%s",&nome1);
    printf(nome1);
    cliente1=procuracliente(nome1);
    if(cliente1==-1)
    {
        printf("\nCliente nao encontrado!!");
        goto init;
    }
    printf("\nNovo nome:");
    fflush(stdin);
    gets(clientes[cliente1].nome);
    printf("\nNovo endereco:");
    gets(clientes[cliente1].end);
    printf("\nCliente modificado...");
    getch();
}
```





```
void verclientes( )
```

```
{  
    int i=0;  
    printf("\nOs clientes cadastrados sao:");  
    for(i=0; i<next; i++)  
  
        {  
            printf("\nNome: %s",clientes[i].nome);  
            printf("\nEndereco: %s",clientes[i].end);  
            printf("\nNumero da conta:%d",clientes[i].nconta);  
        }  
  
    getch();  
}
```



```
void deletar( )

{
    char nome2[MAX];
    int cliente2,i=0;
    printf("\nNome do cliente a ser apagado:");
    gets(nome2);
    cliente2=procuracliente(nome2);
    for(i=0; i<=MAX; i++)


        {
            clientes[cliente2].nome[i]=' ';
            clientes[cliente2].end[i]=' ';
        }

    printf("\nCliente de numero %d.
    apagado..",clientes[cliente2].nconta);
    getch();
}
```



```
void erro()

{
    printf("\nOpcao errada!!");
    getch();
}
```



```
main()
{
    int op;
    while(1)
    {
        system("cls");
        inicio:
        op=menu();
        switch(op)
        {
            case '1':cadastrar();break;
            case '2':modificar();break;
            case '3':verclientes();break;
            case '4':deletar();break;
            case '5':exit(0);
            default: erro(); goto inicio;
        }
    }
}
```