

```
//Inclusão de Bibliotecas
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <exception>

#define END_ARQUIVO "LACadastroEmArquivo.dat"

#define M_ABRE_LEITURA "rb"
#define M_ABRE_LEITURA_ESCRITA "rb+"

#define M_ANEXA_ESCRITA "ab"
#define M_ANEXA_ESCRITA_LEITURA "ab+"

#define M_SOBRESCREVE_ESCRITA "wb"
#define M_SOBRESCREVE_ESCRITA_LEITURA "wb+"

typedef struct
{
    char nome[40];
    char endereco[100];
    long telefone;
    long cpf;
    bool ativo;
} tipoCliente;

void exibirMensagemDeClienteJaCadastrado()
{
    printf("\n\nCliente ja cadastrado!\n\n");
}

void exibirMensagemDeClienteNaoCadastrado()
{
    printf("\n\nCliente nao cadastrado!\n\n");
}

void exibirMensagemDeNenhumClienteCadastrado()
{
    printf("\n\nNenhum cliente cadastrado!\n\n");
}

void exibirMensagemDeConfirmacaoDeOperacao(bool operacaoRealizadaComSucesso)
{
    if (operacaoRealizadaComSucesso)
    {
        printf("\n\nOperacao concluida com SUCESSO!\n\n");
    }
    else
    {
        printf("\n\nOperacao nao concluida - Existem ERROS!\n\n");
    }
}

void exibirCliente(tipoCliente cliente)
{
    printf("\n\n-----");
    printf("\nNome: %s", cliente.nome);
    printf("\nEndereco: %s", cliente.endereco);
    printf("\nTelefone: %d", cliente.telefone);
    printf("\nCpf: %d", cliente.cpf);
    printf("\n-----\n\n");
}

tipoCliente consultar(long cpf)
{
    tipoCliente cliente;
```

```
bool encontrou = false;

FILE * arq = fopen(END_ARQUIVO, M_ABRE_LEITURA);

while (!feof(arq))
{
    fread(&cliente, sizeof(tipoCliente), 1, arq);

    if (cpf == cliente.cpf && cliente.ativo == true)
    {
        encontrou = true;
        break;
    }
}

if (!encontrou)
{
    cliente.ativo = false;
}

fclose(arq);

return cliente;
}

bool cadastrar(tipoCliente cliente)
{
    bool operacaoRealizadaComSucesso = true;

    try
    {
        if (!consultar(cliente.cpf).ativo)
        {
            FILE * arq = fopen(END_ARQUIVO, M_ANEXA_ESCRITA);

            fwrite(&cliente, sizeof(tipoCliente), 1, arq);

            fclose(arq);
        }
        else
        {
            exibirMensagemDeClienteJaCadastrado();
        }
    }
    catch (const std::exception& ex)
    {
        operacaoRealizadaComSucesso = false;
    }

    return operacaoRealizadaComSucesso;
}

bool alterar(tipoCliente cliente)
{
    bool operacaoRealizadaComSucesso = true;

    try
    {
        tipoCliente clientePesquisado;

        FILE * arq = fopen(END_ARQUIVO, M_ABRE_LEITURA_ESCRITA);

        long contador = 0;

        while (!feof(arq))
        {
            fread(&clientePesquisado, sizeof(tipoCliente), 1, arq);

            contador++;
        }
    }
}
```

```
        if (clientePesquisado.cpf == cliente.cpf)
        {
            fseek(arq, (contador - 1) * sizeof(tipoCliente), SEEK_SET);

            fwrite(&cliente, sizeof(tipoCliente), 1, arq);
            break;
        }
    }

    fclose(arq);
}
catch (const std::exception& ex)
{
    operacaoRealizadaComSucesso = false;
}

return operacaoRealizadaComSucesso;
}

bool excluir(long cpf)
{
    bool operacaoRealizadaComSucesso = false;

    tipoCliente cliente;

    cliente = consultar(cpf);

    if (cliente.ativo)
    {
        cliente.ativo = false;

        operacaoRealizadaComSucesso = alterar(cliente);
    }
    else
    {
        exibirMensagemDeClienteNaoCadastrado();
    }

    return operacaoRealizadaComSucesso;
}

void relatorio()
{
    tipoCliente cliente;

    FILE * arq = fopen(END_ARQUIVO, M_ABRE_LEITURA);

    long contador = 0;

    while (!feof(arq))
    {
        fread(&cliente, sizeof(tipoCliente), 1, arq);

        if (!feof(arq) && cliente.ativo)
        {
            contador++;
            exibirCliente(cliente);
        }
    }

    fclose(arq);

    if (contador == 0)
    {
        exibirMensagemDeNenhumClienteCadastrado();
    }
    else
    {

```

```
        exibirMensagemDeConfirmacaoDeOperacao(true);
    }
}

//Método Main - Entry Point do Programa
int main()
{
    //Título do Programa e autor
    printf(" << UEFS - PGCA 2014.1 - Programa Arquivo com Cadastro em Arquivos>>");
    printf("\n\n ## Autor: Leonardo Melo\n\n");

    int escolha = -1;

    do
    {
        printf("\nMENU\n");
        printf("\n1 - Cadastrar");
        printf("\n2 - Alterar");
        printf("\n3 - Excluir");
        printf("\n4 - Consultar");
        printf("\n5 - Relatorio");
        printf("\n0 - Sair");

        printf("\n\nDigite sua opcao: ");
        scanf("%d", &escolha);
        fflush(stdin);

        tipoCliente cliente;

        switch (escolha)
        {
            case 1:
                printf("\n\nNome: ");
                gets(cliente.nome);
                fflush(stdin);

                printf("\n\nEndereco: ");
                gets(cliente.endereco);
                fflush(stdin);

                printf("\n\nTelefone: ");
                scanf("%d", &cliente.telefone);
                fflush(stdin);

                printf("\n\nCpf: ");
                scanf("%d", &cliente.cpf);
                fflush(stdin);

                cliente.ativo = true;

                exibirMensagemDeConfirmacaoDeOperacao(cadastrar(cliente));

                break;
            case 2:
                printf("\n\nCpf: ");
                scanf("%d", &cliente.cpf);
                fflush(stdin);

                cliente = consultar(cliente.cpf);

                if (!cliente.ativo)
                {
                    exibirMensagemDeClienteNaoCadastrado();
                }
                else
                {
                    printf("\n\nNovo Nome: ");
                    gets(cliente.nome);
                    fflush(stdin);
                }
            }
        }
    }
}
```

```
        printf("\n\nNovo Endereco: ");
        gets(cliente.endereco);
        fflush(stdin);

        printf("\n\nNovo Telefone: ");
        scanf("%d", &cliente.telefone);
        fflush(stdin);

        exibirMensagemDeConfirmacaoDeOperacao(alterar(cliente));
    }

    break;
case 3:
    printf("\n\nCpf: ");
    scanf("%d", &cliente.cpf);
    fflush(stdin);

    cliente = consultar(cliente.cpf);

    if (!cliente.ativo)
    {
        exibirMensagemDeClienteNaoCadastrado();
    }
    else
    {
        exibirMensagemDeConfirmacaoDeOperacao(excluir(cliente.cpf));
    }

    break;
case 4:
    printf("\n\nCpf: ");
    scanf("%d", &cliente.cpf);
    fflush(stdin);

    cliente = consultar(cliente.cpf);

    if (!cliente.ativo)
    {
        exibirMensagemDeClienteNaoCadastrado();
    }
    else
    {
        exibirCliente(cliente);
    }

    break;
case 5:
    relatorio();

    break;
}

} while (escolha != 0);

//Pula duas linhas e Pausa a Tela (Utilizando comandos DOS)
printf("\n\n\n");
system("pause");

//Retorno do método main
return(EXIT_SUCCESS);
}
```

