

Desenvolvimento de Programa em Linguagem C

FACULDADE UNYLEYA

PÓS-GRADUAÇÃO EM ENGENHARIA ELETRÔNICA E DE COMPUTAÇÃO

FERMYNO BRAGA GUTIERREZ

fermyno@gmail.com

INTRODUÇÃO

Como foi observado, a disciplina Linguagem C teve por objetivo a apresentação da linguagem, o conhecimento de sua sintaxe, seu funcionamento e a estrutura básica de um programa. As estruturas de controle de fluxo e de repetição, o funcionamento de matrizes e strings, funções e ponteiros, e os comandos mais importantes da linguagem foram demonstrados por meio de exemplos práticos e exercícios.

ENUNCIADO

Por meio do conhecimento adquirido na disciplina, desenvolva um programa em linguagem C para cadastro de pacientes, médicos e consultas de uma clínica médica.

O cadastro deverá conter as seguintes informações:

- **Paciente**
 - Nome
 - Telefone
 - Endereço
 - Idade
 - CPF

- **Médico**
 - Nome
 - Especialidade
 - Telefone
 - CRM
- **Consulta**
 - Paciente
 - Médico
 - Data da Consulta

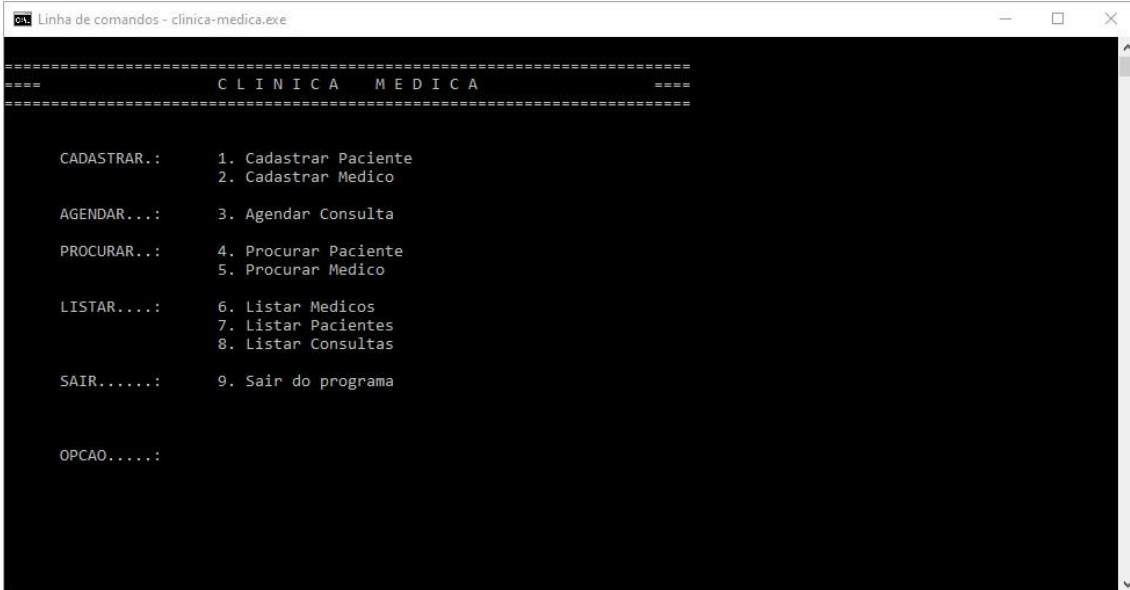
O programa deve ter um menu constante com as seguintes opções:

- Cadastrar um paciente
- Cadastrar um médico
- Agendar uma consulta
- Procurar um paciente pelo CPF
- Procurar um médico pelo CRM
- Sair do programa

ORIENTAÇÕES

- a) Faça a submissão do seu código. Caso encontre problemas ao enviar o arquivo no formato “*.c”, envie-o compactado.
- b) Siga a orientação para envio de arquivo, atente-se ao prazo de entrega.

CAPTURAS DE TELA



```
Linha de comandos - clinica-medica.exe

=====
==== CLINICA MEDICA =====
=====

CADASTRAR.:    1. Cadastrar Paciente
                2. Cadastrar Medico

AGENDAR...:    3. Agendar Consulta

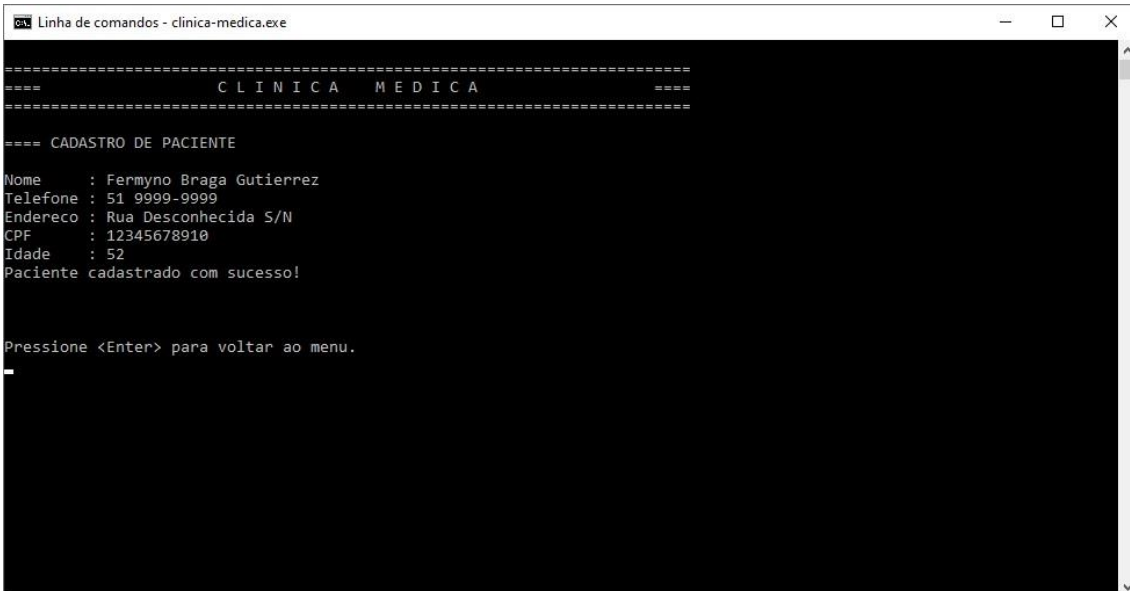
PROCURAR...:   4. Procurar Paciente
                5. Procurar Medico

LISTAR....:    6. Listar Medicos
                7. Listar Pacientes
                8. Listar Consultas

SAIR.....:    9. Sair do programa

OPCAO.....:
```

MENU PRINCIPAL



```
Linha de comandos - clinica-medica.exe

=====
==== CLINICA MEDICA =====
=====

==== CADASTRO DE PACIENTE

Nome       : Fermyno Braga Gutierrez
Telefone   : 51 9999-9999
Endereco   : Rua Desconhecida S/N
CPF        : 12345678910
Idade      : 52
Paciente cadastrado com sucesso!

Pressione <Enter> para voltar ao menu.
_
```

CADASTRAR CLIENTE

```
Linha de comandos - clinica-medica.exe

=====
==== CLINICA MEDICA ====
=====

==== AGENDAR CONSULTA

Digite o CPF do paciente      : 12345678910
Digite o CRM do medico       : 123456789
Data da consulta (DD/MM/AAAA) : 01/04/2023
Consulta agendada com sucesso.

Pressione <Enter> para voltar ao menu.
_
```

AGENDAR CONSULTA

```
Linha de comandos - clinica-medica.exe

=====
==== CLINICA MEDICA ====
=====

==== LOCALIZAR PACIENTE

Digite o CPF do paciente a ser procurado: 12345678910

===== Dados do Paciente =====
Nome      : Fermyno Braga Gutierrez
Telefone  : 51 9999-9999
Endereco  : Rua Desconhecida S/N
Idade     : 52 anos
CPF       : 12345678910
=====

Pressione <Enter> para voltar ao menu.
```

LOCALIZAR PACIENTE

```
Linha de comandos - clinica-medica.exe

=====
==== CLINICA MEDICA ====
=====

==== LOCALIZAR MEDICO

Digite o CRM do medico: 123456789

===== Dados do Medico =====
Nome      : Dr. Pedro Alvares Cabral
Especialidade : Clinico Geral
Telefone   : 51 8888-8888
CRM        : 123456789
=====

Pressione <Enter> para voltar ao menu.
_
```

LOCALIZAR MÉDICO

```
Linha de comandos - clinica-medica.exe

=====
==== CLINICA MEDICA ====
=====

==== MEDICOS CADASTRADOS

Nome      : Dr. Pedro Alvares Cabral. Especialidade: Clinico Geral
Telefone  : 51 8888-8888. CRM: 123456789

Nome      : Dr. Cristovam Colombo Jr. Especialidade: Clinico Geral
Telefone  : 51 9999-9998. CRM: 123456788

2 medicos cadastrados.

=====
_
```

LISTAR MÉDICOS

```
Linha de comandos - clinica-medica.exe

=====
==== C L I N I C A   M E D I C A   ====
=====

==== PACIENTES CADASTRADOS

Nome      : Fermyno Braga Gutierrez. Telefone: 51 9999-9999
Endereco  : Rua Desconhecida s/n
CPF       : 12345678910. Idade: 52 anos

Nome      : Pedro Primeiro. Telefone: 51 9999-9991
Endereco  : Rua Sem Numero
CPF       : 12345678999. Idade: 84 anos

2 pacientes cadastrados.

=====
```

LISTAR PACIENTES

```
Linha de comandos - clinica-medica.exe

=====
==== C L I N I C A   M E D I C A   ====
=====

==== CONSULTAS AGENDADAS

Paciente  : Fermyno Braga Gutierrez
Medico    : Dr. Pedro Alvares Cabral
Data      : 02/04/2023

Paciente  : Pedro Primeiro
Medico    : Dr. Cristovam Colombo Jr.
Data      : 04/04/2023

2 consultas agendadas.

=====
```

LISTAR CONSULTAS



```
Linha de comandos

=====
===== CLINICA MEDICA =====
=====
Programa finalizado com sucesso.

C:\UNYLEYA\Disciplinas\3.Linguagem C>
```

CÓDIGO FONTE

```
// ////////////////////////////////////////
// INFORMACOES UTEIS
// PROGRAMA DESENVOLVIDO POR FERMYN0 BRAGA GUTIERREZ PARA A DISCIPLINA DE
// LINGUAGEM C DO CURSO DE POS-GRADUACAO EM ENGENHARIA ELETRONICA E DE
// COMPUTACAO DA FACULDADE UNILEYA
// ////////////////////////////////////////

// ////////////////////////////////////////
// BIBLIOTECAS
// ////////////////////////////////////////
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// ////////////////////////////////////////
// CONSTANTES
// ////////////////////////////////////////
#define MAX_PACIENTES      10
#define MAX_MEDICOS        10
#define MAX_CONSULTAS      20
#define MAX_NOME            50
#define MAX_TELEFONE        15
#define MAX_ENDERECO        100
#define MAX_CPF             12
#define MAX_ESPECIALIDADE  30
#define MAX_CRM             10

// ////////////////////////////////////////
// ESTRUTURAS
// ////////////////////////////////////////

typedef struct {
    char nome[50];
    char telefone[15];
    char endereco[100];
    int idade;
    char cpf[12];
} Paciente;

typedef struct {
    char nome[50];
    char especialidade[30];
```



```

    char telefone[15];
    char crm[10];
} Medico;

typedef struct {
    Paciente paciente;
    Medico medico;
    char data[11];
} Consulta;

// ////////////////////////////////////////
// VARIÁVEIS GLOBAIS
// ////////////////////////////////////////
Paciente pacientes[MAX_PACIENTES];
Medico medicos[MAX_MEDICOS];
Consulta consultas[MAX_CONSULTAS];
int num_pacientes = 0;
int num_medicos = 0;
int num_consultas = 0;

// ////////////////////////////////////////
// PROTOTIPO DE FUNCOES
// ////////////////////////////////////////
void cadastrar_paciente();
void cadastrar_medico();
void agendar_consulta();
void procurar_paciente_por_cpf();
void procurar_medico_por_crm();
void listar_medicos();
void listar_pacientes();
void listar_consultas();
void espera_tecla();
void cabecalho();

// ////////////////////////////////////////
// PROGRAMA
// ////////////////////////////////////////
int main() {
    int opcao;
    do {
        // Exibindo o menu de opções
        cabecalho();
        printf("\n");
        printf("\n");
        printf("      CADASTRAR.:      1. Cadastrar Paciente      \n");
        printf("                      2. Cadastrar Medico          \n");
        printf("\n");
        printf("      AGENDAR...:      3. Agendar Consulta          \n");
        printf("\n");

```

```

printf("      PROCURAR...:      4. Procurar Paciente      \n");
printf("                        5. Procurar Medico        \n");
printf("\n");
printf("      LISTAR.....:      6. Listar Medicos          \n");
printf("                        7. Listar Pacientes          \n");
printf("                        8. Listar Consultas          \n");
printf("\n");
printf("      SAIR.....:      9. Sair do programa          \n");
printf("\n");
printf("\n");
printf("\n");
printf("      OPCA.....:      ");
scanf("%d", &opcao);
getchar(); // Consumindo o caractere '\n' deixado pelo scanf

// Realizando a ação correspondente a opção escolhida
switch (opcao) {
    case 1:
        cadastrar_paciente();
        break;
    case 2:
        cadastrar_medico();
        break;
    case 3:
        agendar_consulta();
        break;
    case 4:
        procurar_paciente_por_cpf();
        break;
    case 5:
        procurar_medico_por_crm();
        break;
    case 6:
        listar_medicos();
        break;
    case 7:
        listar_pacientes();
        break;
    case 8:
        listar_consultas();
        break;
    case 9:
        cabecalho();
        printf("\nPrograma finalizado com sucesso.\n");
        break;
    default:
        printf("Opcao invalida. Tente novamente.\n");
}

```

```

        printf("\n");
    } while (opcao != 9);

    return 0;
}

// ////////////////////////////////////////
// FUNCOES
// ////////////////////////////////////////

// AGUARDA PRESSIONAMENTO DE TECLA
void espera_tecla() {
    getchar(); //limpar buffer do teclado
    printf("\n\nPressione <Enter> para voltar ao menu.\n");
    getchar();
}

// MOSTRA CABECALHO NA TELA
void cabecalho() {
    // Exibindo o cabecalho do programa
    system("cls");
    printf("\n");
    printf("=====\n");
    printf("====          C L I N I C A      M E D I C A          ====\n");
    printf("=====\n");
}

// CADASTRA PACIENTES
void cadastrar_paciente() {
    cabecalho();
    printf("\n");
    printf("=== CADASTRO DE PACIENTE \n");
    printf("\n");
    printf("Nome      : ");
    fgets(pacientes[num_pacientes].nome, MAX_NOME, stdin);
    strtok(pacientes[num_pacientes].nome, "\n");
    printf("Telefone : ");
    fgets(pacientes[num_pacientes].telefone, MAX_TELEFONE, stdin);
    strtok(pacientes[num_pacientes].telefone, "\n");
    printf("Endereco : ");
    fgets(pacientes[num_pacientes].endereco, MAX_ENDERECO, stdin);
    strtok(pacientes[num_pacientes].endereco, "\n");
    printf("CPF       : ");
    fgets(pacientes[num_pacientes].cpf, MAX_CPF, stdin);
    strtok(pacientes[num_pacientes].cpf, "\n");
}

```

```

printf("Idade      : ");
scanf("%d", &pacientes[num_pacientes].idade);
printf("Paciente cadastrado com sucesso!\n\n");
// Incrementa o número de pacientes cadastrados
num_pacientes++;
espera_tecla();
}

// CADASTRA MEDICOS
void cadastrar_medico() {
    cabecalho();
    printf("\n");
    printf("==== CADASTRO DE MEDICO \n");
    printf("\n");
    // Lê os dados do médico
    printf("Nome          : ");
    fgets(medicos[num_medicos].nome, MAX_NOME, stdin);
    strtok(medicos[num_medicos].nome, "\n");
    printf("Especialidade : ");
    fgets(medicos[num_medicos].especialidade, MAX_ESPECIALIDADE, stdin);
    strtok(medicos[num_medicos].especialidade, "\n");
    printf("Telefone      : ");
    fgets(medicos[num_medicos].telefone, MAX_TELEFONE, stdin);
    strtok(medicos[num_medicos].telefone, "\n");
    printf("CRM           : ");
    fgets(medicos[num_medicos].crm, MAX_CRM, stdin);
    strtok(medicos[num_medicos].crm, "\n");
    printf("\nMedico cadastrado com sucesso!\n\n");
    // Incrementa o número de médicos cadastrados
    num_medicos++;
    espera_tecla();
}

// AGENDA CONSULTAS
void agendar_consulta() {
    int i;
    int indice_paciente = -1;
    int indice_medico = -1;
    char data[11];
    char cpf[12];
    char crm[10];

    cabecalho();
    printf("\n");
    printf("==== AGENDAR CONSULTA \n");
    printf("\n");

```

```

printf("Digite o CPF do paciente      : ");
scanf("%s", cpf);
// Busca pelo paciente pelo CPF
for (i = 0; i < MAX_PACIENTES; i++) {
    if(strcmp(pacientes[i].cpf, cpf) == 0){
        indice_paciente = i;
        break;
    }
}
if (indice_paciente == -1) {
    printf("Paciente nao encontrado.\n");
    espera_tecla();
    return;
}

printf("Digite o CRM do medico        : ");
scanf("%s", crm);
// Busca pelo medico pelo CRM
for (i = 0; i < MAX_MEDICOS; i++) {
    if(strcmp(medicos[i].crm, crm) == 0){
        indice_medico = i;
        break;
    }
}

if (indice_medico == -1) {
    printf("Medico nao encontrado.\n");
    espera_tecla();
    return;
}

printf("Data da consulta (DD/MM/AAAA) : ");
scanf("%s", data);

// Adiciona a consulta
consultas[num_consultas].paciente = pacientes[indice_paciente];
consultas[num_consultas].medico = medicos[indice_medico];
strcpy(consultas[num_consultas].data, data);
num_consultas++;
printf("Consulta agendada com sucesso.\n");
espera_tecla();
return;

// printf("Nao foi possivel agendar a consulta.\n");
espera_tecla();
}

// PROCURA PACIENTE POR CPF

```

```

void procurar_paciente_por_cpf() {
    int i;
    char cpf[12];
    int encontrado = 0;

    cabecalho();
    printf("\n");
    printf("==== LOCALIZAR PACIENTE \n");
    printf("\n");

    printf("Digite o CPF do paciente a ser procurado: ");
    scanf("%s", cpf);

    for(i=0; i<num_pacientes; i++){
        if(strcmp(pacientes[i].cpf, cpf) == 0){
            printf("\n");
            printf("==== Dados do Paciente =====\n\n");
            printf("Nome      : %s\n", pacientes[i].nome);
            printf("Telefone : %s\n", pacientes[i].telefone);
            printf("Endereco : %s\n", pacientes[i].endereco);
            printf("Idade    : %d anos\n", pacientes[i].idade);
            printf("CPF      : %s\n", pacientes[i].cpf);
            printf("===== \n\n");
            encontrado = 1;
            break;
        }
    }

    if(!encontrado){
        printf("Paciente nao encontrado.\n");
    }
    espera_tecla();
}

// PROCURA MEDICO POR CRM
void procurar_medico_por_crm() {
    int i;
    char crm[10];
    int encontrado = 0;

    cabecalho();
    printf("\n");
    printf("==== LOCALIZAR MEDICO \n");
    printf("\n");
    printf("Digite o CRM do medico: ");
    scanf("%s", crm);

    for(i=0; i<MAX_MEDICOS; i++) {

```

```

        if(strcmp(medicos[i].crm, crm) == 0) {
            printf("\n");
            printf("===== Dados do Medico =====\n\n");
            printf("Nome           : %s\n", medicos[i].nome);
            printf("Especialidade : %s\n", medicos[i].especialidade);
            printf("Telefone      : %s\n", medicos[i].telefone);
            printf("CRM           : %s\n", medicos[i].crm);
            printf("===== \n\n");
            encontrado = 1;
            break;
        }
    }

    if(!encontrado) {
        printf("\n--- Medico nao encontrado ---\n");
    }
    espera_tecla();
}

// LISTA MEDICOS CADASTRADOS
void listar_medicos() {
    int i;

    cabecalho();
    printf("\n");
    printf("==== MEDICOS CADASTRADOS \n");
    printf("\n");

    for (i = 0; i < MAX_MEDICOS; i++) {
        // nao mostrar registros em branco
        if (strcmp(medicos[i].nome, "") != 0) {
            printf("Nome       : %s. Especialidade: %s\n", medicos[i].nome,
medicos[i].especialidade);
            printf("Telefone : %s. CRM: %s\n", medicos[i].telefone, medicos[i].crm);
            printf("\n");
        }
    }
    printf("\n%d medicos cadastrados.\n", num_medicos);
    printf("\n===== \n\n");
};
    espera_tecla();
}

// LISTA PACIENTES CADASTRADOS
void listar_pacientes() {
    int i;

```

```

cabecalho();
printf("\n");
printf("==== PACIENTES CADASTRADOS \n");
printf("\n");

for (i = 0; i < MAX_PACIENTES; i++) {
    // nao mostrar registros em branco
    if (strcmp(pacientes[i].nome, "") != 0) {
        printf("Nome      : %s. Telefone: %s\n", pacientes[i].nome, pacientes[i].telefone);
        printf("Endereco  : %s\n", pacientes[i].endereco);
        printf("CPF       : %s. Idade: %d anos\n", pacientes[i].cpf, pacientes[i].idade);
        printf("\n");
    }
}

printf("\n%d pacientes cadastrados.\n", num_pacientes);
printf("\n===== \n\n");
};
espera_tecla();
}

// LISTA CONSULTAS AGENDADAS
void listar_consultas() {
    int i;

    cabecalho();
    printf("\n");
    printf("==== CONSULTAS AGENDADAS \n");
    printf("\n");
    for (i = 0; i < num_consultas; i++) {
        // nao mostrar registros em branco
        printf("Paciente : %s\n", consultas[i].paciente);
        printf("Medico   : %s\n", consultas[i].medico);
        printf("Data     : %s\n", consultas[i].data);
        printf("\n");
    }
    printf("\n%d consultas agendadas.\n", num_consultas);
    printf("\n===== \n\n");
};
espera_tecla();
}

```