

```

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

typedef struct
{
    char observacoes[140];
}prontuario;
typedef struct
{
    int data;
    int horario;
    char medicacaoaplicada[140];
    char refeicao[20];
    char comportamento[140];
    int temperatura;
    int freqcardiaca;
    int taxaglicose;
    int taxaoxigenio;
    char rh[20];
    int codigodiagnostico;
    int banho;
    int TempoBanho;
    int GraudeDependencia;
    int assaduras;
    int defecou;
    int tipodefezes;
    int qtdafezes;
    int urinou;
    int tipourina;
    int qtdaurina;
    int vomitou;
    int codDiagnostico;
    //struct diagnostico*proximo;//encadeamento
}diagnostico;
typedef struct
{
    char nome[50];
    int CPF;
    int idade;
    int RG;
    char EstadoCivil[15];
    char CidadeNatal[20];
    char Estado[20];
    char anamnese[140];
    char estadoClinico[20];
    char alergias[140];
    char tipoSangue[3];
    char rh[20];
    char enfermidade[140];
    int codpaciente;
    int codProntuario;
    int qtdadediagnostico;
    diagnostico diagpct[100];
    prontuario prontpct;
    struct paciente*proximo;//encadeamento
}paciente;

```

```

typedef struct
{
    char nome[50];
    int CPF;
    int idade;
    int RG;
    char EstadoCivil[15];
    char CidadeNatal[20];
    char Estado[20];
    char historico[140];
    int tempoprofissao;
    int cargahoraria;
    int pisosalarial;
    int carteiradetrabalho;
    char fichacriminal[140];
    int codfuncionario;
    int senha;
    struct funcionario* proximo; //encadeamento
} funcionario;

paciente *F = NULL; //auxiliares
paciente *Atual = NULL;
paciente *Anterior = NULL;
funcionario *Fun = NULL; //auxiliares
funcionario *AtualFun = NULL;
funcionario *AnteriorFun = NULL;
int i=0, np=0, teste=0;

paciente *alocarNovoPaciente() //alocando o espaço para um novo paciente
{
    paciente *novopaciente;
    novopaciente = (paciente *) malloc(sizeof(paciente));
    novopaciente->proximo=NULL;
    return novopaciente;
}

funcionario *alocarNovoFuncionario() //alocando o espaço para um novo funcionario
{
    funcionario *novofuncionario;
    novofuncionario = (funcionario *) malloc(sizeof(funcionario));
    novofuncionario->proximo=NULL;
    return novofuncionario;
}

int inserirNovoPaciente(char *nome, int idade, int CPF, int RG, char *EstadoCivil,
char *CidadeNatal, char *Estado, char *anamnese, char *estadoClinico, char
*alergias, char *tipoSangue, char *rh, char *enfermidade) //inserindo o nome do paciente
e seu problema na fila
{
    if(F == NULL)
    {
        F = alocarNovoPaciente();
        strcpy(F->nome, nome);
        F->idade=idade;
        F->CPF=CPF;
        F->RG=RG;
        strcpy(F->EstadoCivil, EstadoCivil);
        strcpy(F->CidadeNatal, CidadeNatal);
        strcpy(F->Estado, Estado);
    }
}

```

```

        strcpy(F->anamnese, anamnese);
        strcpy(F->estadoClinico, estadoClinico);
        strcpy(F->alergias, alergias);
        strcpy(F->tipoSangue, tipoSangue);
        strcpy(F->rh, rh);
        strcpy(F->enfermidade, enfermidade);
        F->proximo = NULL;
        F->codpaciente=np;
        F->qtdadediagnostico =-1;
        np++;
        return F->codpaciente;
    }
    else
    {
        Atual = F;
        while(Atual->proximo != NULL)
        {
            Atual = (paciente*)Atual->proximo;
        }
        strcpy(Atual->nome, nome);
        Atual->idade=idade;
        Atual->CPF=CPF;
        Atual->RG=RG;
        strcpy(Atual->EstadoCivil, EstadoCivil);
        strcpy(Atual->CidadeNatal, CidadeNatal);
        strcpy(Atual->Estado, Estado);
        strcpy(Atual->anamnese, anamnese);
        strcpy(Atual->estadoClinico, estadoClinico);
        strcpy(Atual->alergias, alergias);
        strcpy(Atual->tipoSangue, tipoSangue);
        strcpy(Atual->rh, rh);
        strcpy(Atual->enfermidade, enfermidade);
        Atual->qtdadediagnostico =-1;
        Atual->proximo = NULL;
        Atual->codpaciente=np;;
        np++;
        return Atual->codpaciente;
    }
}
int inserirNovoFuncionario(char *nome, int idade, int CPF,int RG, char *EstadoCivil,
char *CidadeNatal,char *Estado,char *historico,int tempoprofissao,int
cargahoraria,int pisolarial,int carteiradetrabalho,char *fichacriminal)
{
    if(Fun == NULL)
    {
        Fun = alocarNovoFuncionario();
        AtualFun = Fun;
        strcpy(AtualFun->nome, nome);
        Fun->idade=idade;
        Fun->CPF=CPF;
        Fun->RG=RG;
        strcpy(Fun->EstadoCivil, EstadoCivil);
        strcpy(Fun->CidadeNatal, CidadeNatal);
        strcpy(Fun->Estado, Estado);
        strcpy(Fun->historico, historico);
        Fun->tempoprofissao= tempoprofissao;
    }
}

```

```

        Fun->cargahoraria=cargahoraria;
        Fun->pisosalarial=pisosalarial;
        Fun->carteiradetrabalho=carteiradetrabalho;
        strcpy(Fun->fichacriminal, fichacriminal);
        Fun->proximo = NULL;
        Fun->codfuncionario=1000 + i;
        i++;
        return Fun->codfuncionario;
    }
else
{
    AtualFun = Fun;
    while(AtualFun->proximo != NULL)
    {
        AtualFun = (funcionario*)Atual->proximo;
    }

    strcpy(AtualFun->nome, nome);
    AtualFun->idade=idade;
    AtualFun->CPF=CPF;
    AtualFun->RG=RG;
    strcpy(AtualFun->EstadoCivil, EstadoCivil);
    strcpy(AtualFun->CidadeNatal, CidadeNatal);
    strcpy(AtualFun->Estado, Estado);
    strcpy(AtualFun->historico, historico);
    AtualFun->tempoprofissao= tempoprofissao;
    AtualFun->cargahoraria=cargahoraria;
    AtualFun->pisosalarial=pisosalarial;
    AtualFun->carteiradetrabalho=carteiradetrabalho;
    strcpy(AtualFun->fichacriminal, fichacriminal);
    AtualFun->proximo = NULL;
    AtualFun->codfuncionario=1000 + i;
    i++;
    return AtualFun->codfuncionario;
}

}

void CadastrarPaciente()
{
    char *nome[50];
    int CPF;
    int idade;
    int RG;
    char *EstadoCivil[15];
    char *CidadeNatal[20];
    char *Estado[20];
    char *anamnese[140];
    char *estadoClinico[20];
    char *alergias[140];
    char *tipoSangue[3];
    char *rh[20];
    char *enfermidade[140];
    int codigopaciente;

    system("cls");

```

```

printf("-- CADASTRAR PACIENTE --\n\n");
printf("-- Dados Pessoais --\n");
printf("\n Nome do paciente:  ");
fflush(stdin);
gets(nome);

printf("\nIdade:          ");
scanf("%i",&idade);

printf("\nCPF:            ");
scanf("%i",&CPF);

printf("\nRG:             ");
scanf("%i",&RG);

printf("\nEstado Civil:   ");
fflush(stdin);
gets(EstadoCivil);

printf("\nCidade Natal:   ");
fflush(stdin);
gets(CidadeNatal);

printf("\nEstado Natal:   ");
fflush(stdin);
gets(Estado);

printf("-- Dados Saude --\n");
printf("\nAnamnese:      ");
fflush(stdin);
gets(anamnese);

printf("\nEstado Clinico: ");
fflush(stdin);
gets(estadoclinico);

printf("\nAlergias:      ");
fflush(stdin);
gets(alergias);

printf("\nTipo Sangue:   ");
fflush(stdin);
gets(tipoSangue);

printf("\nRH:           ");
fflush(stdin);
gets(rh);

printf("\nEnfermidade:   ");
fflush(stdin);
gets(enfermidade);

codigopaciente=inserirNovoPaciente(&nome,idade,CPF,RG,&EstadoCivil,&CidadeNatal,&Estado,&anamnese,&estadoclinico,&alergias,&tipoSangue,&rh,&enfermidade);
printf("Cadastro realizado com sucesso! Codigo Paciente: %i",codigopaciente);
system("PAUSE");
system("cls");

```

```

}
void CadastrarFuncionario()
{
    char *nome[50];
    int CPF;
    int idade;
    int RG;
    char *EstadoCivil[15];
    char *CidadeNatal[20];
    char *Estado[20];
    char *historico[140];
    int tempoprofissao;
    int cargahoraria;
    int pisosalarial;
    int carteiradetrabalho;
    char *fichacriminal[140];
    int codfuncionario;
    int senha,login;

    system("cls");
    printf("-- CADASTRAR Funcionario --\n\n");
    printf("\n Nome do Funcionario:  ");
    fflush(stdin);
    gets(nome);

    printf("\nIdade:          ");
    scanf("%i",&idade);

    printf("\nCPF:              ");
    scanf("%i",&CPF);

    printf("\nRG:                ");
    scanf("%i",&RG);

    printf("\n Estado Civil:    ");
    fflush(stdin);
    gets(EstadoCivil);

    printf("\n Cidade Natal:   ");
    fflush(stdin);
    gets(CidadeNatal);

    printf("\n Estado Natal:   ");
    fflush(stdin);
    gets(Estado);

    printf("\n Historico:      ");
    fflush(stdin);
    gets(historico);

    printf("\nTempo Profissao:          ");
    scanf("%i",&tempoprofissao);

    printf("\nCarga Horária:          ");
    scanf("%i",&cargahoraria);

    printf("\nPiso Salarial:          ");
    scanf("%i",&pisosalarial);

```

```

printf("\nCarteira de Trabalho:      ");
scanf("%i",&carteiradetrabalho);

printf("\n Ficha Criminal:  ");
fflush(stdin);
gets(fichacriminal);

login=inserirNovoFuncionario(&nome, idade, CPF, RG, &EstadoCivil,
&CidadeNatal,&Estado,&historico,tempoprofissao,cargahoraria,pisosalarial,carteiradet
rabalho,&fichacriminal);
senha=login;
printf("\nLogin: %i\nSenha: %i",login,senha);
}
int buscarPaciente(int codPaciente)
{
    int aux=0;
    aux=0;
    if(F==NULL)
    {
        printf("\nNenhum Paciente cadastrado");
        aux=0;
    }
    else
    {
        Atual = F;
        if(Atual->proximo==NULL)
        {
            if(Atual->codpaciente == codPaciente)
            {
                aux++;
            }
        }
        else
        {
            while(Atual->proximo !=NULL || aux==0)
            {
                if(Atual->codpaciente == codPaciente)
                {
                    aux++;
                }
                Atual = (paciente*)Atual->proximo;
            }
        }
    }
    return aux;
}
void CadastrarOutros()
{
}
void mostrarPrincipaisdadosPaciente(int codPaciente)
{
    paciente *P;
    int aux=0;
    system("CLS");
    if(F==NULL)

```

```

{
    printf("\nNenhum Paciente cadastrado");
    aux=0;
}
else
{
    Atual = F;
    if(Atual->proximo==NULL)
    {
        if(Atual->codpaciente == codPaciente)
        {
            aux++;
        }
    }
    else
    {
        while(Atual->proximo !=NULL || aux==0)
        {
            if(Atual->codpaciente == codPaciente)
            {
                aux++;
            }
            Atual = (paciente*)Atual->proximo;
        }
    }
    P=Atual;
    printf("\n----PRONTUARIO DE:  %s  CODIGO: %i----\n",P->nome,Atual-
>codpaciente);
    printf("\nAnamnese %s",P->anamnese);
    printf("\nMedicacoes %s",P->diagpct->medicacaoaplicada);
    printf("\nEstado Clinico %s",P->estadoClinico);
    printf("\nAlergias %s",P->alergias);
    printf("\nTipo Sanguineo %s - Rh %s",P->tipoSangue,P->rh);
}
}
void mostrarDiagnosticosPaciente(int codPaciente)
{
    int a=0;
    int j=0,sair=0;
    int aux=0;
    if(F==NULL)
    {
        printf("\nNenhum Paciente cadastrado\n");
        aux=0;
    }
    else
    {
        Atual = F;
        if(Atual->proximo==NULL)
        {
            if(Atual->codpaciente == codPaciente)
            {
                aux++;
            }
        }
        else
        {

```



```

while(Atual->proximo !=NULL || aux==0)
{
    if(Atual->codpaciente == codPaciente)
    {
        aux++;
    }
    Atual = (paciente*)Atual->proximo;
}
}
do
{
    printf("\n\n---DIAGNOSTICO %i---",j);
    printf("\nData: %i",Atual->diagpct[j].data);
    printf("\nHorario: %i",Atual->diagpct[j].horario);

    printf("\n_____");
    printf("\nMedicacao Aplicada: %s",&Atual-
>diagpct[j].medicacaoaplicada);

    printf("\n_____");
    printf("\nRefeicao: %s",&Atual->diagpct[j].refeicao);

    printf("\n_____");
    printf("\nComportamento: %s",&Atual->diagpct[j].comportamento);

    printf("\n_____");
    printf("\nTemperatura: %i",Atual->diagpct[j].temperatura);
    printf("\nFrequencia Cardiaca: %i",Atual-
>diagpct[j].freqcardiaca);
    printf("\nTaxa de Glicose: %i",Atual->diagpct[j].taxaglicose);
    printf("\nTaxa Oxigenio: %i",Atual->diagpct[j].taxaoxigenio);

    printf("\n_____");

    if(Atual->diagpct[j].banho == 0)
    {
        printf("\nNao tomou banho!");
    }
    else
    {
        printf("\nTomou banho!");
        printf("\nTempo Banho: %i",Atual->diagpct[j].TempoBanho);
        printf("\nGrau de Dependencia: %i",Atual-
>diagpct[j].GraudeDependencia);
    }
    if(Atual->diagpct[j].assaduras==0)
    {
        printf("\nNao Apresentou assaduras");
    }
    else
    {
        printf("\nApresentou assaduras:");
    }
    if(Atual->diagpct[j].defecou==0)
    {
        printf("\nNao Defecou");
    }
}

```

```

        }
        else
        {
            printf("\nDefecou");
            printf("\nTipo de fezes: %i",Atual-
>diagpct[j].tipodefezes);
            printf("\nQuantidade de Fezes: %i",Atual-
>diagpct[j].qtdafezes);
        }
        if(Atual->diagpct[j].urinou==0)
        {
            printf("\nNao Urinou");
        }
        else
        {
            printf("\nUrinou");
            printf("\nTipo de Urina: %i",Atual-
>diagpct[j].tipourina);
            printf("\nQuantidade de Urina: %i",Atual-
>diagpct[j].qtdaurina);
        }

        if(Atual->diagpct[j].vomitou==0)
        {
            printf("\nNao Vomitou");
        }
        else
        {
            printf("\nVomitou");
        }

        printf("\n=====
=====\\n");
        j++;

    }while(j<=Atual->qtdadediagnostico);

}
j=0;
system("Pause");
system("CLS");
}
void fazerConsultaProntuario()
{
    int codPaciente;
    system("CLS");
    printf("\nDigite oCodigo do prontuario ou do Paciente");
    scanf("%i",&codPaciente);
    teste=buscarPaciente(codPaciente);
    if(teste==0)
    {
        printf("==\npaciente nao encontrado!!==");
    }
    else
    {
        mostrarPrincipaisdadosPaciente(codPaciente);
        mostrarDiagnosticosPaciente(codPaciente);
    }
}

```

```

    }
}
void AnexarDiagnostico(int data,int horario,char *medicacaoaplicada,char
*refeicao,char *comportamento,int temperatura,int freqcardiaca,int taxaglicose,int
taxaoxigenio,int banho,int TempoBanho,int GraudeDependencia,int assaduras,int
defecou,int tipodefezes,int qtdafezes,int urinou,int tipourina,int qtdaurina,int
vomitou)
{
    int j=0;
    Atual->qtdadediagnostico=Atual->qtdadediagnostico+1;
    if(j<Atual->qtdadediagnostico)
    {
        while(j!=Atual->qtdadediagnostico)
        {
            j++;
        }
    }
    Atual->diagpct[j].data=data;
    Atual->diagpct[j].horario=horario;
    strcpy(Atual->diagpct[j].medicacaoaplicada,medicacaoaplicada);
    strcpy(Atual->diagpct[j].refeicao,refeicao);
    strcpy(Atual->diagpct[j].comportamento,comportamento);
    Atual->diagpct[j].temperatura=temperatura;
    Atual->diagpct[j].freqcardiaca=freqcardiaca;
    Atual->diagpct[j].taxaglicose=taxaglicose;
    Atual->diagpct[j].taxaoxigenio=taxaoxigenio;
    Atual->diagpct[j].banho=banho;
    Atual->diagpct[j].TempoBanho=TempoBanho;
    Atual->diagpct[j].GraudeDependencia=GraudeDependencia;
    Atual->diagpct[j].assaduras=assaduras;
    Atual->diagpct[j].defecou=defecou;
    Atual->diagpct[j].tipodefezes=tipodefezes;
    Atual->diagpct[j].qtdafezes=qtdafezes;
    Atual->diagpct[j].urinou=urinou;
    Atual->diagpct[j].tipourina=tipourina;
    Atual->diagpct[j].qtdaurina=qtdaurina;
    Atual->diagpct[j].vomitou=vomitou;
    j=0;
}
void fazerDiagnostico()
{
    int data;
    int horario;
    char *medicacaoaplicada[140];
    char *refeicao[20];
    char *comportamento[140];
    int temperatura;
    int freqcardiaca;
    int taxaglicose;
    int taxaoxigenio;
    int codigodiagnostico;
    int banho;
    int TempoBanho;
    int GraudeDependencia;
    int assaduras;
    int defecou;
}

```

```

int tipodefezes;
int qtdafezes;
int urinou;
int tipourina;
int qtdaurina;
int vomitou;
int codDiagnostico;
int aux=0,codPaciente;
int q=0;
aux=0;
printf("\nCodigo Paciente: ");
scanf("%i",&codPaciente);
if(F==NULL)
{
    printf("\nNenhum Paciente cadastrado");
    aux=0;
}
else
{
    Atual = F;
    if(Atual->proximo==NULL)
    {
        if(Atual->codpaciente == codPaciente)
        {
            aux++;
        }
    }
    else
    {
        while(Atual->proximo !=NULL || aux==0)
        {
            if(Atual->codpaciente == codPaciente)
            {
                aux++;
            }
            else
            {
                Atual = (paciente*)Atual->proximo;
            }
        }
    }
}

//aux=buscarPacienteParaDiagnostico(codPaciente);
if(aux==0)
{
    printf("paciente nao encontrado");
}
else
{
    system("CLS");
    printf("---DIAGNOSTICO PACIENTE %i--- ",codPaciente);

    printf("\nData: ");
    scanf("%i",&data);
    printf("\nHorario: ");
}

```

```

scanf("%i",&horario);
printf("----- ");
printf("\nMedicacao Aplicada: ");
fflush(stdin);
gets(medicacaoaplicada);

printf("\nRefeicao: ");
fflush(stdin);
gets(refeicao);

printf("\nComportamento: ");
fflush(stdin);
gets(comportamento);

printf("\nTemperatura: ");
scanf("%i",&temperatura);

printf("Frequencia Cardiaca: ");
scanf("%i",&freqcardiaca);

printf("Taxa Glicose: ");
scanf("%i",&taxaglicose);

printf("Taxa Oxigenio: ");
scanf("%i",&taxaoxigenio);

printf("\n----- ");

printf("\nBanho 0-NAO 1-SIM: ");
scanf("%i",&banho);
if(banho!=0)
{
    printf("\nTempo Banho: ");
    scanf("%i",&TempoBanho);
    printf("\nGrau de dependencia: ");
    scanf("%i",&GraudeDependencia);
    printf("\nAssaduras 0-NAO 1-SIM: ");
    scanf("%i",&assaduras);
    printf("\n----- ");
}
else
{
    TempoBanho=0;
    GraudeDependencia=0;
    assaduras=0;
}
printf("\nDefecou 0-NAO 1-SIM: ");
scanf("%i",&defecou);
if(defecou!=0)
{
    printf("\nTipo de fezes ");
    scanf("%i",&tipodefezes);
    printf("\nQuantidade de Fezes ");
    scanf("%i",&qtddefezes);
    printf("\n----- ");
}
else
{

```

```

        tipodefezes=0;
        qtdafezes=0;
    }
    printf("\nUrinou 0-NAO 1-SIM: ");
    scanf("%i",&urinou);
    if(urinou!=0)
    {
        printf("\nTipo de Urina");
        scanf("%i",&tipourina);

        printf("\nQuantidade de Urina ");
        scanf("%i",&qtdaurina);
    }
    else
    {
        tipourina=0;
        qtdaurina=0;
    }

    printf("\nVomitou 0-NAO 1-SIM: ");
    scanf("%i",&vomitou);

```

AnexarDiagnostico(data,horario,&medicacaoaplicada,&refeicao,&comportamento,temperatura,freqcardiaca,taxaglicose,taxaoxigenio,banho,TempoBanho,GraudeDependencia,assaduras,defecou,tipodefezes,qtdafezes,urinou,tipourina,qtdaurina,vomitou);

```

        printf("\n");

```

```

    }
    system("PAUSE");
    system("CLS");
}
void fazerCadastro()
{
    int op;
    printf("1-Paciente\n2-Funcionario\n3-Outros\n");
    scanf("%i",&op);
    switch(op)
    {
        case 1:
        {
            CadastrarPaciente();
            break;
        }
        case 2:
        {
            CadastrarFuncionario();
            break;
        }
        case 3:
        {
            CadastrarOutros();
            break;
        }
        default:
        {
            printf("Escolha invalida");

```

```

        break;
    }
}
printf("\n");
system("PAUSE");
system("CLS");
}
void Alterar()
{
}
void main()
{
    int op;
    while(1)
    {
        system("CLS");
        printf("\n1-Gerar Diagnostico\n2-Cadastro\n3-Consultar Prontuario\n4-
Alterar\n");
        scanf("%i",&op);
        switch(op)
        {
            case 1:
            {
                fazerDiagnostico();
                break;
            }
            case 2:
            {
                fazerCadastro();
                break;
            }
            case 3:
            {
                fazerConsultaProntuario();
                break;
            }
            case 4:
            {
                Alterar();
                break;
            }
            default:
            {
                printf("\nopcao invalida");
                break;
            }
        }
    }
}

```