

# Métodos de Programação 2

---

MiETI :: 1º Ano

**2018/2019**



- Eduardo Paiva, 85312  
[eduardopaiva@live.com.pt](mailto:eduardopaiva@live.com.pt)



- Tiago Ferreira, 85392  
[tiago.ferreira.19@hotmail.com](mailto:tiago.ferreira.19@hotmail.com)



# ÍNDICE

INTRODUÇÃO .....	1
ENUNCIADO .....	2
DESCRIÇÃO DA SOLUÇÃO .....	3
CÓDIGO EM C .....	4
EXEMPLOS DE UTILIZAÇÃO DA SOLUÇÃO .....	16
CONCLUSÃO .....	17

# INTRODUÇÃO

Este projeto realizado no âmbito da cadeira de Métodos de Programação II visa criar um programa que armazene os valores de uma lista de análises clínicas de pacientes bem como os valores registados para cada uma dessas análises.

Neste relatório iremos apresentar a descrição do problema, bem como a nossa solução para o problema proposto e exemplos da sua implementação.

# ENUNCIADO

Neste problema pretende-se criar um sistema que permita armazenar os valores de uma lista de análises clínicas de um ou vários pacientes, bem como os valores registados para cada uma dessas análises, no exame anterior. Caso não exista valor anterior para um dado tipo de análise deve registar-se o valor -1.

Cada análise a registar terá associado um código alfanumérico, o resultado do exame (um número real), a data da colheita atual e a data da colheita anterior.

O registo poderá ser feito para um ou vários pacientes. Cada paciente é identificado pelo nome e pelo número nacional de saúde (NNS). O sistema deve emitir um alerta sempre que o valor atual de uma análise seja pelo menos 20% superior ou pelos menos 20% inferior ao valor do exame anterior. Além disso, o programa deve mostrar o boletim da colheita atual de um paciente, dado o seu NNS.

Os dados introduzidos devem ser gravados num ficheiro binário para que possam ser recuperados numa sessão de trabalho posterior.

Os resultados produzidos devem ser enviados para o standard output.

# DESCRIÇÃO DA SOLUÇÃO

Em primeiro lugar, através do enunciado retiramos as informações mais importantes.

Verificamos logo que na nossa solução iríamos utilizar uma estrutura onde iríamos guardar os dados de cada utente, neste caso o NNS, nome, valor e datas das análises.

Optamos por criar quatro funções para responder as necessidades pedidas, uma para adicionar utentes, uma para editar utentes, uma para apagar utentes e uma para consultar os dados do utente bem como os valores e datas das análises. Criamos também um pequeno menu que permite ao utilizador utilizar as funcionalidades do nosso programa.

No final acrescentamos a função para guardar os dados introduzidos, para numa secção posterior poderem ser carregados e serem utilizados.

# CÓDIGO EM C

```
/* MP2 - G23 */
/* Eduardo Paiva A85312 */
/* Tiago Ferreira A85392 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX 500

FILE *ficheiro;

typedef struct estrutura
{

    char nome[MAX];
    int NNS;
    int colesterol;
    int colesterolAnt;
    char DataColesterol[MAX];
    char DataColesterolAnterior[MAX];
    int glicose;
    int glicoseAnt;
    char DataGlicose[MAX];
    char DataGlicoseAnterior[MAX];
    int triglicerideos;
    int triglicerideosAnt;
    char DataTriglicerideos[MAX];
    char DataTriglicerideosAnterior[MAX];
    int acido;
    int acidoAnt;
    char DataAcido[MAX];
    char DataAcidoAnterior[MAX];
```

```
} boletim;

int Nutente=0;

boletim utente[MAX];

void AdicionarUtente()
{

    int esc = 1;
    int d=0;

    while(esc != 0)
    {
        printf("Insira o NSN:");
        scanf(" %d", &utente[Nutente].NNS);
        printf("Insira o Nome:");
        fflush(stdin);
        fgets(utente[Nutente].nome,MAX,stdin);

        while(esc!=0)
        {
            printf("Que tipo de analise quer adicionar:\n1 - Colestrol\n2 - Glicose\n3 - Triglecerideos\n4 - Acido Urico\n0 - Nenhuma\n");
            scanf("%d", &d);

            switch(d)
            {

                case 1:
                    system("cls");
                    printf("Valor analise colestrol:");
                    scanf(" %d", &utente[Nutente].colestrol);
```



```
printf("Data da colheita(dd/mm/aa):");  
fflush(stdin);  
fgets(utente[Nutente].DataColestrol,MAX,stdin);  
break;
```

case 2:

```
system("cls");  
printf("Valor analise Glicose:");  
scanf(" %d", &utente[Nutente].glicose);  
printf("Data da colheita(dd/mm/aa):");  
fflush(stdin);  
fgets(utente[Nutente].DataGlicose, MAX,stdin);  
break;
```

case 3:

```
system("cls");  
printf("Valor analise Triglicerideos:");  
scanf(" %d", &utente[Nutente].triglicerideos);  
printf("Data da colheita(dd/mm/aa):");  
fflush(stdin);  
fgets(utente[Nutente].DataTriglicerideos, MAX,stdin);  
break;
```

case 4:

```
system("cls");  
printf("Valor analise Acido Urico:");  
scanf(" %d", &utente[Nutente].acido);  
printf("Data da colheita(dd/mm/aa):");  
fflush(stdin);  
fgets(utente[Nutente].DataAcido, MAX,stdin);  
break;
```

case 0:

```
utente[Nutente].colestrol = -1;
```

```
        utente[Nutente].glicose = -1;
        utente[Nutente].triglicerideos = -1;
        utente[Nutente].acido = -1;
        break;

    }

    system("cls");
    printf("Quer adicionar outra analise?\n1 - sim\n0 - nao\n");
    scanf(" %d", &esc);

}

if(utente[Nutente].glicose == 0)
{
    utente[Nutente].glicose = -1;
}

if(utente[Nutente].colesterol == 0)
{
    utente[Nutente].colesterol = -1;
}

if(utente[Nutente].triglicerideos == 0)
{
    utente[Nutente].triglicerideos = -1;
}

if(utente[Nutente].acido == 0)
{
    utente[Nutente].acido = -1;
}

system("cls");
printf("Quer adicionar outro utente?\n1 - sim\n0 - nao\n");
scanf(" %d", &esc);
Nutente++;
```

```
}

}

void editarUtente()
{
    int Bole;
    int c;
    int esc = 1;
    int i;
    int f;
    int g;

    system("cls");
    printf("Insira o NNS que quer alterar:\n");
    scanf("%d", &Bole);

    for(i=0; i<Nutente; i++)
    {
        if(Bole==utente[i].NNS)
        {
            printf("O que quer editar?\n1 - NSS\n2 - Nome\n3 - Analise Colestrol\n4 - Analise Glicose\n5 - Analise Triglicerideos\n6 - Acido Urico\n ");
            scanf("%d", &c);
            switch(c)
            {
                case 1:
                    system("cls");
                    printf("NSS atual:%d\nInsira o novo NSS:", utente[i].NNS);
                    scanf(" %d", &utente[i].NNS);
                    break;

                case 2:
                    system("cls");
                    printf("Nome atual:%s\nInsira o novo nome:", utente[i].nome);
```

```
fflush(stdin);
fgets(utente[i].nome,MAX,stdin);
break;

case 3:
    system("cls");
    printf("Ultimo Valor Colestrol:%d\nInsira o novo valor:",
utente[i].colestrol);
    utente[i].colestrolAnt= utente[i].colestrol;
    scanf(" %d", &utente[i].colestrol);
    printf("Data da ultima colheita Colestrol:%s\nInsira a nova
data(dd/mm/aa):", utente[i].DataColestrol);
    strcpy(utente[i].DataColestrolAnterior, utente[i].DataColestrol);
    fflush(stdin);
    fgets(utente[i].DataColestrol,MAX,stdin);

    if(utente[i].colestrolAnt != -1)
    {
        f = (utente[i].colestrolAnt*0.20)+utente[i].colestrolAnt;
        g= utente[i].colestrolAnt-(utente[i].colestrolAnt*0.20);
        if(utente[i].colestrol>f|| utente[i].colestrol<g)
        {
            printf("\nATENCAO, O NOVO VALOR ESTA ACIMA OU ABAIXO 20
PORCENTO COMPARADO COM O VALOR ANTERIOR\n\n");
        }
    }

    break;

case 4:
    system("cls");
    printf("Ultimo valor Glicose:%d\nInsira o novo valor:",
utente[i].glicose);
    utente[i].glicoseAnt= utente[i].glicose;
    scanf(" %d", &utente[i].glicose);
```

```
printf("Data da ultima colheita Glicose:%s\nInsira a nova
data(dd/mm/aa):", utente[i].DataGlicose);

strcpy(utente[i].DataGlicoseAnterior, utente[i].DataGlicose);
fflush(stdin);
fgets(utente[i].DataCoolestrol,MAX,stdin);

if(utente[i].glicoseAnt != -1)
{
    f = (utente[i].glicoseAnt*0,20)+utente[i].glicoseAnt;
    g= utente[i].glicoseAnt-(utente[i].glicoseAnt*0,20);
    if(utente[i].glicose>f || utente[i].glicose<g)
    {
        printf("\nATENCAO, O NOVO VALOR ESTA ACIMA OU ABAIXO 20
PORCENTO COMPARADO COM O VALOR ANTERIOR\n\n");
    }

}

break;

case 5:
system("cls");
printf("Ultimo valor Triglicerideos:%d\nInsira o novo valor:",
utente[i].triglicerideos);
utente[i].triglicerideosAnt= utente[i].triglicerideos;
scanf(" %d", &utente[i].triglicerideos);
printf("Data da ultima colheita triglicerideos:%s\nInsira a nova
data(dd/mm/aa):", utente[i].DataTriglicerideos);
strcpy(utente[i].DataTriglicerideosAnterior, utente[i].DataTriglicerideos);
fflush(stdin);
fgets(utente[i].DataTriglicerideos,MAX,stdin);

if(utente[i].triglicerideosAnt != -1)
{
    f = (utente[i].triglicerideosAnt*0,20)+utente[i].triglicerideosAnt;
    g= utente[i].triglicerideosAnt-(utente[i].triglicerideosAnt*0,20);
```

```
        if(utente[i].triglicerideos>f || utente[i].triglicerideos<g)
        {
            printf("\nATENCAO, O NOVO VALOR ESTA ACIMA OU ABAIXO 20
PORCENTO COMPARADO COM O VALOR ANTERIOR\n\n");
        }

    }
    break;

    case 6:
        system("cls");
        printf("Ultimo valor Acido Urico:%d\nInsira o novo valor:",
utente[i].acido);
        utente[i].acidoAnt= utente[i].acido;
        scanf(" %d", &utente[i].acido);
        printf("Data da ultima colheita Acido Urico:%s\nInsira a nova
data(dd/mm/aa):", utente[i].DataAcido);
        strcpy(utente[i].DataAcidoAnterior, utente[i].DataAcido);
        fflush(stdin);
        fgets(utente[i].DataAcido,MAX,stdin);

        if(utente[i].acidoAnt != -1)
        {
            f = (utente[i].acidoAnt*0,20)+utente[i].acidoAnt;
            g= utente[i].acidoAnt-(utente[i].acidoAnt*0,20);
            if(utente[i].acido>f || utente[i].acido<g)
            {
                printf("\nATENCAO, O NOVO VALOR ESTA ACIMA OU ABAIXO 20
PORCENTO COMPARADO COM O VALOR ANTERIOR\n\n");
            }
        }

    }
    break;
}
}
```

```
}

printf("Quer editar mais algum utente?\n1 - sim\n0 - nao\n");
scanf("%d", &esc);
switch(esc)
{
case 1:
    system("cls");
    editarUtente();
    break;

case 0:
    break;
}

}

void apagarUtente()
{
    int bole;
    int esc = 1;
    int i;
    while(esc!= 0)
    {
        printf("Insira o NNS do utente que quer apagar:");
        scanf("%d", &bole);
        for(i=0; i<Nutente; i++)
        {
            if(bole==utente[i].NNS)
            {
                utente[i] = utente[MAX];
            }
        }
    }
}
```

```
printf("Quer apagar mais algum utente?\n1 - Sim\n0 - Nao\n");
scanf("%d", &esc);

}
}
void ConsultarBoletim()
{
    int Bole;
    int esc= 1;
    int i=0;
    while(esc != 0)
    {
        printf("Digite o NNS do utente que quer consultar:");
        scanf("%d", &Bole);
        for(i=0; i<Nutente; i++)
        {
            if(Bole == utente[i].NNS)
            {
                printf("\nNome: %s\n", utente[i].nome);
                printf("NNS: %d\n", utente[i].NNS);
                printf("Colestrol: %d| %s| %s \n", utente[i].colestrol,
utente[i].DataColestrol, utente[i].DataColestrolAnterior);
                printf("Glicose: %d| %s| %s \n", utente[i].glicose, utente[i].DataGlicose,
utente[i].DataGlicoseAnterior);
                printf("Triglicerideos: %d| %s| %s\n",
utente[i].triglicerideos, utente[i].DataTriglicerideos,
utente[i].DataTriglicerideosAnterior);
                printf("Acido Urico: %d| %s| %s\n", utente[i].acido, utente[i].DataAcido,
utente[i].DataAcidoAnterior);
            }
        }

        printf("\nQuer consultar outro boletim?\n1 - sim \n0 - nao\n");
        scanf(" %d", &esc);
        system("cls");
    }
}
```



```
    }  
}  
  
int main()  
{  
  
    int esc;  
  
    ficheiro = fopen("dados.bin","rb");  
    if(ficheiro != NULL)  
    {  
        fread (&Nutente, sizeof(int), 1, ficheiro);  
        fread(utente,sizeof(boletim),MAX, ficheiro);  
    }  
    fclose(ficheiro);  
  
    while(esc != 0)  
    {  
        system("cls");  
        printf("\n -----\\n");  
        printf("|  Adicionar Utente:    1 |\\n");  
        printf("|  Editar utente:       2 |\\n");  
        printf("|  Apagar utente:       3 |\\n");  
        printf("|  Consultar Boletim:   4 |\\n");  
        printf("|  Sair e guardar dados: 0 |\\n");  
        printf(" -----\\n");  
        scanf("%d", &esc);  
  
        switch(esc)  
        {  
        case 1:  
            system("cls");  
            AdicionarUtente();  
            break;
```

```
case 2:
    system("cls");
    editarUtente();
    break;
case 3:
    system("cls");
    apagarUtente();
    break;
case 4:
    system("cls");
    ConsultarBoletim();
    break;
case 0:
    ficheiro = fopen("dados.bin","wb");
    fwrite(&Nutente, sizeof(int), 1, ficheiro);
    fwrite(utente,sizeof(boletim),MAX, ficheiro);
    fclose(ficheiro);
    break;
}
}
}
```

# EXEMPLOS DE UTILIZAÇÃO DA SOLUÇÃO

Entrada:

NSS: 1897899

Nome: Carlos Oliveira

Valor analise colesterol: 150

Data da análise: 12/07/18

Valor analise glicose: 121

Data da análise: 14/07/18

Output:

(no caso de pedir para consultar o boletim)

NSS: 1897899

Nome: Carlos Oliveira

Colesterol: 150|12/07/18|

Glicose: 121|14/07/18|

Triglicerídeos: -1| |

Ácido Úrico: -1| |

# CONCLUSÃO

Através da solução que desenvolvemos foi possível resolver o problema apresentado de forma eficaz e precisa.

O processo mais trabalhoso foi a passagem do algoritmo para código.

Foi um projeto positivo, pois foi possível consolidar os conhecimentos de estruturas e manipulação de ficheiros que adquirimos ao longo do semestre.