



# **Licenciatura em Engenharia Informática**

**Fundamentos de Programação 2020/2021**

## **Trabalho prático final**

**Realizado em: 02/01/2022**

**Guilherme Rodrigues a2020154390**

# Índice

## Conteúdo

<b>1. Introdução .....</b>	<b>4</b>
<b>2. Elaboração .....</b>	<b>5</b>
<b>4. Funcionalidades do projeto .....</b>	<b>14</b>
<b>5. Tempo despendido no projeto .....</b>	<b>14</b>
<b>6. Conclusão .....</b>	<b>14</b>

## Lista de figuras

Figure 1: Menu do programa .....	5
Figure 2: Comando switch.....	6
Figure 3: Pedido de informações ao utilizador .....	7
Figure 4: Escrita no ficheiro .....	7
Figure 5: Função indicador.....	8
Figure 6: Abertura e copia do conteúdo .....	8
Figure 7: Identificação da tarefa.....	9
Figure 8: Alteração do conteúdo .....	9
Figure 9: Conteúdo original.....	9
Figure 10: Conteúdo alterado .....	9
Figure 11: Menu de consultas.....	10
Figure 12: Consulta de dados de todas as tarefas introduzidas .....	10
Figure 13: Todas as tarefas.....	10
Figure 14: Consulta de dados das tarefas não concluídas, dada uma importância, ou de todas.....	11
Figure 15: Consulta de dados das tarefas não concluídas e atrasadas .....	11
Figure 16: Resumo do estado das tarefas. ....	12
Figure 17: Menu .....	12
Figure 18: Introdução das informações .....	12
Figure 19: Conteúdo do ficheiro.....	13
Figure 20: Introdução do novo estado .....	13
Figure 21: Estado alterado.....	13
Figure 22: Menu de consultas.....	13
Figure 23: Consulta 1 .....	13
Figure 24: Consulta 2 .....	14
Figure 25: Consulta de dados das tarefas não concluídas e atrasadas .....	14
Figure 26: Consulta 4 .....	14

# I. Introdução

O objetivo deste trabalho passa por criar um programa que faça o armazenamento e a pesquisa de dados inseridos pelo utilizador.

A aplicação tem a função de uma agenda e por isso os dados introduzidos são uma data, uma descrição, a importância da tarefa e por fim o seu estado. Todos estes dados são guardados dentro de um ficheiro e as consultas são efetuadas dentro do mesmo.

O projeto foi realizado no IDE Atom em linguagem C.

## 2. Elaboração

O projeto teve início com a criação de um menu de navegação dentro da função “main” onde corre todo o programa.

```
int main()
{
    int a = 0;
    int indLinha = 0;
    while (a != 4) // Ciclo para que volte sempre ao menu cada vez que se inserir uma opcao diferente
    {

        system("cls"); // Limpa o ecrã

        puts("-----");
        puts("                MENU                ");
        puts("                ");
        puts("    O que pretende realizar?          ");
        puts("                ");
        puts("1 -- Inserir uma Tarefa                ");
        puts("2 -- concluir uma tarefa              ");
        puts("3 -- Menu de consultas                ");
        puts("4 -- Sair                             ");
        puts("-----");

        printf("Insira a opcao: ");
        scanf("%d", &a);
        fflush(stdin);

        system("cls");
    }
}
```

Figure 1: Menu do programa

Dentro desta função existe um ciclo “while” que permite que o programa esteja sempre em aberto e que permite ao utilizador voltar ao menu principal quando executa uma função do menu e não tenha de iniciar o programa novamente.

O comando de “fflush” tem a função de limpar algum tipo de informação que o teclado possa ter e evita assim caracteres indesejados, por sua vez o comando “system(“cls”)” limpa todo o ecrã para uma visão mais limpa do mesmo.

Ainda dentro desta função existe um comando de “switch” que permite chamar outras funções e por sua vez executar outras funções da aplicação.

```
switch (a)
{
    //case 1: indLinha = indicador(); inserirTarefa(i); indLinha++; break;
    case 1: inserirTarefa(); break;
    case 2: estado(); break;
    case 3: menuConsulta(); break;
    case 4: break;
    default: printf("Opcao invalida\n");
}

fflush(stdin);
printf("\nOperacao concluida\n");
system("pause > nul"); // Espera para que se carregue numa tecla para continuar
```

Figure 2: Comando switch

- **Inserir uma tarefa**

A função de inserir tarefas, tal como o nome indica, tem a função de inserir novas tarefas dentro do ficheiro de tarefas.

Para isso é pedido ao utilizador as informações de descrição da tarefa, a data e a sua importância o número da tabela é dado automaticamente pelo programa e o estado da tarefa é colocado automaticamente como fazer.

```
printf("Insira a descricao da tarefa: \n");
gets(desc);
if(desc[0] == '\0')
{
    puts("Descricao invalida");
    system("pause > nul");
    return;
}
printf("Insira o dia: \n");
scanf("%d", &dia);
if(dia < 1 || dia > 31)
{
    printf("data invalida");
    system("pause > nul");
    return;
}
printf("Insira o mes: \n");
scanf("%d", &mes);
if(mes < 1 || mes > 12)
{
    printf("data invalida");
    system("pause > nul");
    return;
}
printf("Insira o ano: \n");
scanf("%d", &anos);
if(anos > 2022 || anos < 2022)
{
    printf("data invalida");
    system("pause > nul");
    return;
}
printf("Insira a importancia(1-alta; 2-media; 3-baixa) \n");
scanf("%d", &importancia);
if(importancia < 1 || importancia > 3)
{
    printf("importancia invalida");
    system("pause > nul");
    return;
}
```

**Figure 3: Pedido de informações ao utilizador**

Ao pedir a informação desejada ao utilizador o programa faz de forma automática a validação da informação da data para que não haja datas impossíveis, a importância é introduzida em forma de números de vão de 1 a 3 e por isso é feita a validação para que não sejam introduzidos outros números.

Depois de introduzida a informação a mesma é escrita no ficheiro.

```
fprintf(fp, "%d ", i);
fprintf(fp, "%s ", desc);
fprintf(fp, "%02d-%02d-%d ", dia, mes, anos);
fprintf(fp, "%d ", estado);
fprintf(fp, "%d\n", importancia);
fclose(fp);
printf("\nOperacao concluida\7");
system("pause > nul");
```

**Figure 4: Escrita no ficheiro**

Como referido acima o estado é introduzido automaticamente pelo programa. Ele é introduzido por uma variável iniciada a 0 que depois mais tarde pode ser alterada pelo utilizador. O indicador

da tarefa também é inserido de forma automática. Uma função foi criada para que conte o número de linhas e guarde essa informação, quando uma nova tarefa é inserida o número da linha é colocado no início da nova linha antes da informação da tarefa.

```
int indicador()
{
    int linha = 1;
    int ch;
    FILE * fp;
    fp = fopen("tarefas", "r");

    while ((ch = fgetc(fp)) != EOF)
    {
        if(ch == '\n')
        {
            linha++;
        }
    }
    fclose(fp);
    return linha;
}
```

Figure 5: Função indicador

- **Concluir uma tarefa**

Ainda dentro do menu existe uma função de concluir uma tarefa. Esta função tem o objetivo de apenas mudar o estado de 0 para 1 nas informações da tarefa dentro do ficheiro, para completar este objetivo foi utilizado um ficheiro auxiliar com o nome de “estado”.

O ficheiro original é aberto na opção de leitura e o auxiliar na opção de escrita. Depois de escrito o conteúdo no ficheiro auxiliar ambos são fechados e reabertos, mas com as opções trocadas.

Quando é feita novamente a cópia do conteúdo do ficheiro auxiliar para o original é feita a alteração desejada na secção de estado na tarefa indicada pelo utilizador e assim é feita a alteração de estado.

```
FILE * fp;
FILE * fp1;
fp = fopen("tarefas", "r");
fp1 = fopen("Estado", "wt");
//uso de ficheiro auxiliar para realizar a ação
while(fscanf(fp, "%d %s %d %d %d %d", &i, desc, &dia, &mes, &anos, &estado, &importancia) != EOF)
{
    fprintf(fp1, "%d %s %d %d %d %d\n", i, desc, dia, mes, anos, estado, importancia);
}
fclose(fp);
fclose(fp1);
```

Figure 6: Abertura e cópia do conteúdo



```
puts("Insira o indicador da tarefa: ");
scanf("%d", &a);
puts("insira o novo estado(0-fazer 1-concluido)");
scanf("%d", &b);
if(b < 0 || b > 1)
{
    puts("estado invalido");
    system("pause > nul");
}
```

Figure 7: Identificação da tarefa

```
fp1 = fopen("Estado","r");
fp = fopen("tarefas","wt");
//escreve o novo estado
while(fscanf(fp1, "%d %s %d %d %d %d", &i, desc, &dia, &mes, &anos, &estado, &importancia) != EOF)
{
    if(a == i)
        fprintf(fp, "%d %s %d%d%d %d %d\n", i, desc, dia, mes, anos, b, importancia);
    else
        fprintf(fp, "%d %s %d%d%d %d %d\n", i, desc, dia, mes, anos, estado, importancia);
}

fclose(fp1);
fclose(fp);

printf("\nOperacao concluida\n");
system("pause > nul");
```

Figure 8: Alteração do conteúdo

```
1 1 pe 1-2-2022 0 1
2 2 fe 2-3-2022 1 2
3 3 ge 1-2-2022 1 3
```

Figure 9: Conteúdo original

```
1 1 pe 1-2-2022 0 1
2 2 fe 2-3-2022 1 2
3 3 ge 1-2-2022 0 3
```

Figure 10: Conteúdo alterado

- **Menu de consultas**

O menu de consultas foi criado da mesma maneira do menu principal com a exceção de que este chama as opções de consulta da aplicação

```

while (a != 5) // Ciclo para que volte sempre ao menu cada vez que se inser
{
    system("cls"); // Limpa o ecrã

    puts("-----");
    puts("                MENU                ");
    puts("                ");
    puts("    O que pretende realizar?        ");
    puts("                ");
    puts("1 -- Consultar todas as Tarefas      ");
    puts("2 -- Consulta de dados das tarefas nao concluidas ");
    puts("3 -- Consulta de dados das tarefas nao concluidas e atrasadas");
    puts("4 -- Resumo do estado das tarefas   ");
    puts("5 -- Sair                            ");
    puts("-----");

    printf("Insira a opcao: ");
    scanf("%d", &a);
    fflush(stdin);

    system("cls");

    switch (a)
    {
        case 1: consultarTarefa(); break;
        case 2: naoConcluido(); break;
        case 3: atrasadas(); break;
        case 4: resumo(); break;
        case 5: break;
        default: printf("Opcao invalida\n");
    }
}

```

Figure 11: Menu de consultas

- **Primeira consulta**

A primeira consulta consistira na visualização de todas as tarefas e para isso foi utilizado um ciclo que lê as linhas do ficheiro e transforma a importância em frases para que não sejam vistos os números e seja mais fácil de ler.

```

char desc[50];
int estado = 0, dia = 1, mes = 1, anos = 0, i = 0;
int importancia = 0;
int ch = 0;
FILE *fp;
fp = fopen("tarefas", "r");

if (fp == NULL)
    printf("Erro!!");
else
    printf("Ficheiro aberto com sucesso!!\n");

while(fscanf(fp, "%d %s %d %d %d %d", &i, desc, &dia, &mes, &anos, &estado, &importancia) != EOF)
{
    if(importancia == 1)
        printf("%d %s %d %d %d %d alta\n", i, desc, dia, mes, anos, estado);
    if(importancia == 2)
        printf("%d %s %d %d %d %d media\n", i, desc, dia, mes, anos, estado);
    if(importancia == 3)
        printf("%d %s %d %d %d %d baixa\n", i, desc, dia, mes, anos, estado);
}

fclose(fp);

printf("\nOperacao concluida\n");
system("pause > nul"); // Espera para que se carregue numa tecla para continuar

```

Figure 12: Consulta de dados de todas as tarefas introduzidas

```

Ficheiro aberto com sucesso!!
1 pe 1-2-2022 0 alta
2 fe 2-3-2022 1 media
3 ge 1-2-2022 0 baixa

Operacao concluida

```

Figure 13: Todas as tarefas

## • Segunda consulta

Para a segunda consulta é necessário filtrar as consultas anteriores e mostrar apenas as que não foram concluídas dada uma importância ou todas.

Para realizar esta função é feito um ciclo de leitura das linhas e é feita a comparação dos estados para que quando este seja igual a 1 não seja mostrado e assim é apenas mostrado os que têm o estado a 0.

```
void naoConcluido() //segunda consulta
{
    char desc[50];
    int estado = 0, dia = 1, mes = 1, anos = 0, i = 0, a = 0;
    int importancia = 0;

    FILE * fp;
    fp = fopen("tarefas", "r");

    puts("insira a importancia (0-todas; 1-alta; 2-media; 3-baixa): ");
    scanf("%d", &a);

    while(fscanf(fp, "%d %s %d %d %d %d", &i, desc, &dia, &mes, &anos, &estado, &importancia) != EOF)
    {
        if(a == 0 && estado == 0)
            printf("%d %s %d %d %d %d\n", i, desc, dia, mes, anos, estado, importancia);
        if(a == 1 && importancia == a && estado == 0)
            printf("%d %s %d %d %d %d\n", i, desc, dia, mes, anos, estado, importancia);
        if(a == 2 && importancia == a && estado == 0)
            printf("%d %s %d %d %d %d\n", i, desc, dia, mes, anos, estado, importancia);
        if(a == 3 && importancia == a && estado == 0)
            printf("%d %s %d %d %d %d\n", i, desc, dia, mes, anos, estado, importancia);
    }
    printf("\nOperacao concluida\n");
    system("pause > nul");
}
```

Figure 14: Consulta de dados das tarefas não concluídas, dada uma importância, ou de todas.

## • Terceira Consulta

A terceira consulta passa por listar as tarefas não concluídas e atrasadas, para isso é necessário verificar a data da tarefa e comparar com a data atual.

```
void atrasadas() // terceira consulta
{
    char desc[50];
    int estado = 0, dia = 1, mes = 1, anos = 0, i = 0, a = 0, b = 0, c = 0;
    int importancia = 0;

    puts("insira a data(dd-mm-aa): ");
    scanf("%d %d %d", &a, &b, &c);

    FILE * fp;
    fp = fopen("tarefas", "r");

    while(fscanf(fp, "%d %s %d %d %d %d", &i, desc, &dia, &mes, &anos, &estado, &importancia) != EOF)
    {
        if(a > dia || b > mes || c > anos)
            printf("%d %s %d %d %d %d\n", i, desc, dia, mes, anos, estado, importancia);
    }

    printf("\nOperacao concluida\n");
    system("pause > nul");
}
```

Figure 15: Consulta de dados das tarefas não concluídas e atrasadas

## • Quarta consulta

A quarta consulta por sua vez é pretendido que seja mostrado um resumo das tarefas, ou seja, mostras quantas consultas estão por fazer, quantas consultas estão feitas e por sim quantas consultas estão atrasadas.

Para isso são criadas novas variáveis que farão as contagens e no fim é dado o print do número final de cada contagem.

```
void resumo() //quarta consulta
{
    char desc[50];
    int estado = 0, dia = 1, mes = 1, anos = 0, i = 0;
    int fazer = 0, concluido = 0, fazer1 = 0, concluido1 = 0, fazer2 = 0, concluido2 = 0;
    int importancia = 0;

    FILE * fp;
    fp = fopen("tarefas", "r");

    while(fscanf(fp, "%d %s %d %d %d %d", &i, desc, &dia, &mes, &anos, &estado, &importancia) != EOF)
    {
        if(estado == 0 && importancia == 1)
            fazer++;
        if(estado == 1 && importancia == 1)
            concluido++;
        if(estado == 0 && importancia == 2)
            fazer1++;
        if(estado == 1 && importancia == 2)
            concluido1++;
        if(estado == 0 && importancia == 3)
            fazer2++;
        if(estado == 1 && importancia == 3)
            concluido2++;
    }
}
```

Figure 16: Resumo do estado das tarefas.

### 3. Demonstração do programa

Apos iniciar o programa o menu inicial será mostrado.

```
-----
                        MENU

    O que pretende realizar?

1 -- Inserir uma Tarefa
2 -- concluir uma tarefa
3 -- Menu de consultas
4 -- Sair
-----
Insira a opcao:
```

Figure 17: Menu

De seguida a primeira opção de inserir tarefa deve ser a primeira a ser executada para que haja tarefas a ser pesquisadas nas outras opções.

```
Insira a descricao da tarefa:
TESTE
Insira o dia:
1
Insira o mes:
2
Insira o ano:
2022
Insira a importancia(1-alta; 2-media; 3-baixa)
2
Operacao concluida
```

Figure 18: Introdução das informações

```
1 pe 1-2-2022 0 1
2 fe 2-3-2022 1 2
3 ge 1-2-2022 0 3
4 TESTE 01-02-2022 0 2
```

Figure 19: Conteúdo do ficheiro

De seguida já pode ser alterado o estado da tarefa inserida. Para isso é necessário escolher a segunda opção do menu, depois indicar o número da tarefa e por fim escolher o estado que deseja.

```
Insira o indicador da tarefa:
4
insira o novo estado(0-fazer 1-concluido)
1
Operacao concluida
```

Figure 20: Introdução do novo estado

```
1 pe 1-2-2022 0 1
2 fe 2-3-2022 1 2
3 ge 1-2-2022 0 3
4 TESTE 1-2-2022 1 2
```

Figure 21: Estado alterado

Em seguida já é possível realizar as consultas acima descritas.

```
-----
MENU
O que pretende realizar?
1 -- Consultar todas as Tarefas
2 -- Consulta de dados das tarefas nao concluidas
3 -- Consulta de dados das tarefas nao concluidas e atrasadas
4 -- Resumo do estado das tarefas
5 -- Sair
-----
Insira a opcao:
```

Figure 22: Menu de consultas

```
Ficheiro aberto com sucesso!!
1 pe 1-2-2022 1 alta
2 fe 2-3-2022 1 media
3 ge 1-2-2022 0 baixa
4 TESTE 1-2-2022 1 media
Operacao concluida
```

Figure 23: Consulta 1

```
insira a importancia (0-todas; 1-alta; 2-media; 3-baixa): 0
3 ge 1-2-2022 0 3

Operacao concluida
```

Figure 24: Consulta 2

```
insira a data(dd-mm-aa):
1 1 2022
1 pe 01-1-2022 0 1
2 fe 02-1-2022 0 2
3 ge 02-2-2022 0 3
4 TESTE 03-2-2022 0 2

Operacao concluida
```

Figure 25: Consulta de dados das tarefas não concluídas e atrasadas

```
importancia fazer concluidas
Alta:      0      1
Media:     0      2
Baixa:     1      0
```

Figure 26: Consulta 4

## 4. Funcionalidades do projeto

A aplicação realiza todas as consultas com exceção da terceira. Apesar de serem inseridos os dados da data manualmente e feita a comparação com a data presente na tarefa o programa acaba por mostrar todas as tarefas presentes dentro do ficheiro.

Este inconveniente acabou por prejudicar a última consulta onde seria mostrado o resumo das tarefas por fazes e concluídas, mas, no entanto, a data acabou por não ser incluída dentro da consulta.

Outro erro encontrado foi o facto da descrição poder ter apenas uma palavra.

## 5. Tempo despendido no projeto

Esta aplicação levou cerca de 23 horas para a sua realização e mais 4 para o relatório totalizando assim 34 horas de projeto.

## 6. Conclusão

Para concluir este trabalho demonstrou um nível de dificuldade adequado e um trabalho de pesquisa justificável. Foram aplicados os conhecimentos aprendidos durante o semestre.

Algumas dificuldades foram notadas nomeadamente na realização da terceira consulta para a comparação da data atual com a data-limite da tarefa e, por esta razão, acabou por afetar a resolução da última consulta onde era também pedido a data.

Foi, no entanto, deixado o dentro do ficheiro o código de tentativa de ambas as pesquisas e feitas executáveis apesar de não obter o resultado desejado.

## Referências

Não há referências.