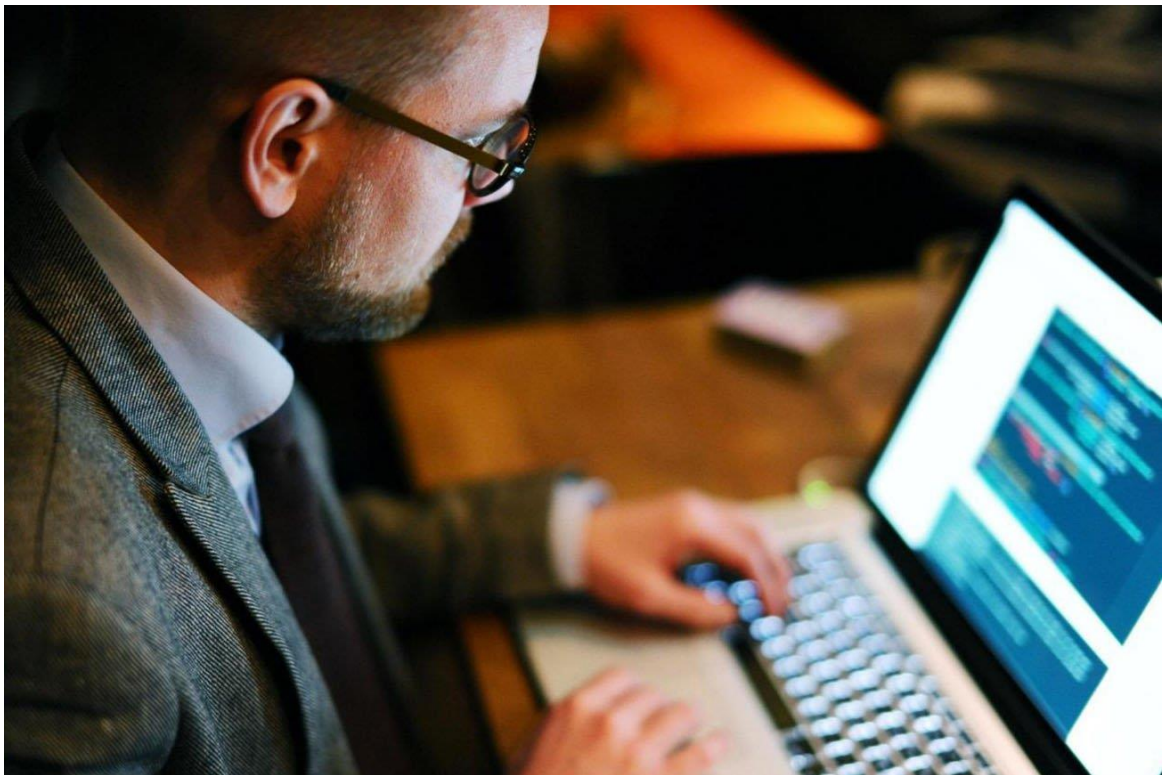


MANUAL DE PROGRAMADOR

Football Life



Tomás Silva

ÍNDICE

Bibliotecas e Headers.....	1
Titulo.h	2
Main.c	3
Login.H.....	4
Menu_Login.h	4
Direção.h	5
Equipa Técnica.....	6
Atletas.....	7

BIBLIOTECAS E HEADERS

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <conio.c>
4 #include <locale.h>
```

Adicionei 4 bibliotecas ao programa, onde o <conio.c> serve para os textcolor(); e os Sleep(); e o <locale.h> para que o texto seja apresentado em português, com acentos adequados.

Adicionei também headers correspondentes às funções que eu criei, cada uma delas têm o nome da sua função no programa.

```
6 #include "Login.h"
7 #include "Menu_Login.h"
8 #include "Titulo.h"
9 #include "Equipa_Tecnica.h"
10 #include "Direcao.h"
11 #include "Atletas.h"
```

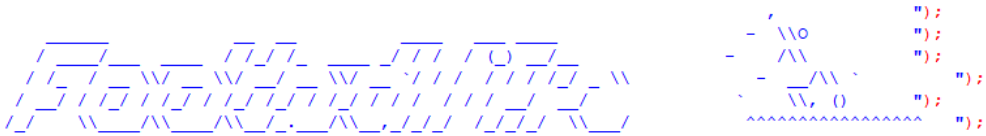
```
13 #define cod_direcao 9999
14 #define cod_equipa_tecnica 2500
15 #define cod_atleta 1000
```

E ainda alguns defines para corresponder um número a uma certa variável, cada uma delas correspondente ao código de acesso dos nomes correspondentes.

TITULO.H

No header "Titulo.h" temos o que corresponde ao título, ou seja, existem duas funções onde a "titulo();" é usada para o início do programa apareça as várias cores na palavra FOOTBALL LIFE em art ascii, usando vários textcolor, e a função "titulo2();" serve para que em todas as páginas seja apresentado o titulo/logo (FOOTBALL LIFE em art ascii) em verde.

```
69 titulo2() {
70     printf("\n\n");
71     textcolor(LIGHTGREEN);
72     puts("
73     puts("
74     puts("
75     puts("
76     puts("
77     puts("
78     textcolor(WHITE);
79 }
```



Nota: Este título não está muito perceptível por que programação C tem de ser usar duas barras em vez de uma quando esta aparece: (\ -> \\\).

MAIN.C

O "main.c" é a principal página do programa e, é lá, que está o menu inicial, já com os códigos de acesso.

```

19 setlocale(LC_ALL, "Portuguese");
20
21 titulo();
22
23 int menu, cod;
24
25
26 do{
27     titulo2();
28     printf("\n\n\n\n\n");
29     printf("
30     printf("
31     printf("
32     printf("
33     printf("
34     printf("
35     printf("
36     printf("
37     printf("
38     scanf("%i", &menu);

```

Começo por usar o "setlocale" para o texto aparecer em português e assim ter uso da biblioteca "locale.h"; declaro as variáveis inteiras do menu e do código de acesso e então segui para a construção do menu com o uso de um "do{...switch(){...}}while(menu!=0);"

De seguida, quando o utilizador digitar o número pretendido este terá de meter o código de acesso. Basicamente, o programa lê o número digitado pelo utilizador e compara-o com o "#define" correspondente, isto usando um "if() else()". Se o utilizador acertar ele será direcionado para o "menu_login_(correspondente)();" que veremos mais a frente.

```

41     case 1: printf("\n
42             scanf("%i",&cod);
43             if(cod==cod_direcao)
44             {
45                 textcolor(LIGHTGREEN);
46                 printf("
47                 textcolor(WHITE);
48                 Sleep(1000);
49                 system("CLS");
50                 menu_login_direcao();
51             }
52     else
53     {
54         textcolor(LIGHTRED);
55         printf("
56         textcolor(WHITE);
57         Sleep(1000);
58         system("cls");
59     }break:

```

LOGIN.H

Para criar o login, no header "Login.h", utilizei uma struct para as variáveis do utilizador e da password do mesmo.

```
4 struct login
5 {
6     int a;
7     char utilizador[20], pass[15];
8 }
9
10
11 struct login log;
```

Neste header existem duas funções: "registar();" e "login_(correspondente)();". A função "registar();", tal como

```
30 strcat(log.utilizador,log.pass);
31 FILE*a;
32 a = fopen(log.utilizador,"w");
33 fprintf(a, " Bem Vindo! ");
34 fclose(a);
```

vemos na imagem, junta todos os caracteres introduzidos pelo utilizador com uma string e então cria um file com esse mesmo nome (utilizador e password), já na função "login();" também junta os caracteres introduzidos pelo utilizador e vê se há algum ficheiro txt com esse mesmo nome, se sim vai para o função correspondente, seja esta a função para a direcção, para a equipa técnica ou para os atletas.

MENU_LOGIN.H

No header "Menu_Login.h" simplesmente existem 3 funções, uma para cada função (direcção, equipa técnica ou atletas), com um menu e dependente da sua escolha este poderá fazer o registo ou logar na sua conta já criada.

```
22 case 1: login_direcao();break;
23
24 case 2: registar();break;
```

Imagem da função "menu_login_direcao();".

DIREÇÃO.H

Logo depois de introduzir 4 structs, criei uma página com algumas opções, sendo algumas delas verificar o saldo do clube e as suas transações, pagar e receber quantias de dinheiro, que estão exemplificadas já a seguir.

Ver Saldo do Clube

Quando o user escolhe a opção para ver o saldo do clube vai deparar-se com um outro menu em que poderá consultar o saldo do clube abrindo o "saldo_do_clube.txt" e também poderá consultar as suas ações no ficheiro "Transações_do_Clube.txt". Estes ficheiros são desenvolvidos pelas outras duas funções.

Pagar Despesas

Para o membro da direção conseguir pagar as despesas, este terá um menu com as mais regulares despesas de clubes, e ainda uma outra opção para quaisquer outras despesas, como a luz, a água...

Para isso, abri o "Saldo_do_Clube.txt" e li o saldo existente no clube, ou seja, o número escrito no txt e então foi só subtrair esse mesmo valor com o valor que o user tinha digitado, volto a escrever o novo saldo no txt e também abro outro ("Transações_do_Clube.txt") para escrever a despesa paga e assim aparecer o que o membro da direção pagou neste mesmo txt e, assim, ele pode consultá-lo na "verificação do saldo do clube".

```
181 printf("\n                                Indique a nome da despesa a pagar: "); fflush(stdin); gets(desp.nome);
182 printf("                                Indique o valor a pagar: "); scanf("%f", &desp.outras);
183 b = fopen("saldo_do_clube.txt", "w");
184 while(fscanf(b, "%f", &md.total) != EOF)
185 {
186 }
187 }
188 md.total = md.total - desp.outras;
189 fprintf(b, "\n Saldo Do Clube: %.2f", md.total);
190 fclose(b);
191 textcolor(LIGHTGREEN);
192 printf("                                [!] Despesa de %s paga!", desp.nome);
193 textcolor(WHITE);
194 Sleep(1000);
195 c = fopen("Transações_do_Clube.txt", "a");
196 fprintf(c, "\n Despesas:");
197 fprintf(c, "\n > %s: - %.2f euros", desp.nome, desp.outras);
198 fclose(c);
```

Receber Pagamentos

Esta opção não varia muito que a anterior, apenas o membro da direção consegue adicionar dinheiro ao clube, seja ele por cotas dos jogadores ou de qualquer outro meio. Assim, apenas mudei a subtração para uma soma e mais uns pequenos ajustes, tal como no ficheiro das transações.

EQUIPA TÉCNICA

Normalmente, os treinadores e os atletas precisam de um meio de comunicação para saberem se são convocados e a que horas e a concentração antes do jogo e, para isso, utilizar o mesmo programa que a direção era a melhor opção.

Para não desviar da estrutura do programa, continuei com um menu com algumas opções, sendo algumas delas inserir informações sobre o próximo jogo e fazer a convocatória, que estão exemplificadas já a seguir.

Inserir informações sobre o próximo jogo

Consegui que o user da equipa técnica inserisse algumas informações para o próximo jogo e para fazê-lo bastou pedir a este que digitasse o pretendido e, a medida que este ia escrevendo, digitava a mesma coisa no "Jogo.txt" para que os atletas pudessem visualizá-lo sempre que quisessem.

```
20     printf("\n                               Equipa Adversária: ");
21     fflush(stdin);
22     gets(adv);
23     fprintf(Jogo, "\n EQUIPA ADVERSÁRIA: %s", adv);
24     printf("                               Dia (DD/MM/AAAA): ");
25     scanf("%i%i%i", &dia, &mes, &ano);
26     fprintf(Jogo, "\n DIA: %i/%i/%i", dia, mes, ano);
27     printf("                               Hora do Jogo (HH:MM): ");
28     scanf("%i%i", &hora, &minuto);
29     fprintf(Jogo, "\n HORA DO JOGO: %i:%i", hora, minuto);
30     printf("                               Hora de concentração em casa (HH:MM): ");
31     scanf("%i%i", &hora, &minuto);
32     fprintf(Jogo, "\n HORA DE CONCENTRAÇÃO (CASA): %i:%i", hora, minuto);
33     printf("                               Campo: ");
34     fflush(stdin);
35     gets(campo);
36     fprintf(Jogo, "\n CAMPO: %s", campo);
37     fclose(Jogo);
```


Fazer a Convocatória

Consegui que o user da equipa técnica inserisse o nome dos 18 jogadores que iam jogar no próximo jogo e para fazê-lo bastou usar uma struct com vetores. Tal como na opção anterior, os nomes aparecem na "Convocatória.txt" para que os atletas pudessem visualizá-la sempre que quisessem.

```
41 struct convocato
42 {
43     char nome[18][18];
44 }
```

```
46 filecall3()
47 {
48     int i;
49     struct convocato conv;
50
51     FILE*op;
52     op = fopen("Convocatoria.txt", "w+");
53     system("cls");
54     titulo2();
55     printf("\n\n\n\n\n");
56     printf("////////////////////////////////////////\n");
57     printf("//////// CONVOCADOS (18) //////////\n");
58     printf("////////////////////////////////////////\n");
59     fprintf(op, " CONVOCADOS (18): \n");
60     for(i=0; i<18; i++)
61     {
62         printf("                                > ");
63         fflush(stdin);
64
65         gets(conv.nome[i]);
66         fprintf(op, "\n > %s", conv.nome[i]);
67     }
68     fclose(op);
69     system("cls");
70 }
```

ATLETAS

A parte mais simples é a dos atletas, pois apenas tem a abertura de dois filles. Apesar de ser muito simples tem de se mudar de computador para computador, pois diferentes computadores têm diferentes users.

```
4 filecall()
5 {
6     system("C:\\Users\\PSI17F20\\Desktop\\Football_Life\\Jogo.txt");
7     system("cls");
8 }
```

