

```
#define _CRT_SECURE_NO_WARNINGS
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <conio.h>
```

```
#include <windows.h>
```

```
#include "estruturas.h"
```

```
/*
*****
*/
```

```
/*Samuel Anjos nº 13305 */
```

```
/*Luis Camilo nº13987 */
```

```
*****
*/
```

```
#define PADX 154 // largura do ecrã de output
```

```
#define PADY 45 // altura do ecrã de output
```

```
/*
```

```
menu para gerir funcionarios
```

```
*/
```

```
struct listaFuncionarios * menuControloFuncionarios(struct listaFuncionarios * head ){
    int escolha; // input do utilizador
```

```
do{
```

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

```
    printf ("%93s\n", "_____");
```

```
    printf ("%87s\n", "GESTAO FUNCIONARIOS");
```

```
    printf ("%93s\n", "_____");
```

```
    printf ("%61s", " ");
```

```
    alteraCorTexto(8); printf("1- "); alteraCorTexto(15); printf("Criar\n");
```

```
    printf ("%61s", " ");
```

```
    alteraCorTexto(8); printf("2- "); alteraCorTexto(15); printf("Editar\n");
```

```
    printf ("%61s", " ");
```

```
    alteraCorTexto(8); printf("3- "); alteraCorTexto(15); printf("Listar\n");
```

```
    printf ("%61s", " ");
```

```
    alteraCorTexto(8); printf("4- "); alteraCorTexto(15); printf("Apagar\n");
```

```
    printf ("%61s", " ");
```

```
    alteraCorTexto(8); printf("9- "); alteraCorTexto(15); printf("Sair\n");
```

```
    printf ("%93s\n", "_____");
```

```
    printf ("%61s", " "); escolha = _getche();
```

```
    fflush (stdin);
```

```
    printf ("\n");
```

```
    switch (escolha){
```

```
        case '1' : head = adicionarFuncionario (head);
```

```
        break;
```

```
        case '2' : editarFuncionario( head);
```

```
        break;
```

```
        case '3' : head = imprimeFuncionarios (head);
```

```
        break;
```

```
        case '4' : head = apagaFuncionario( head);
```

```
        break;
```

```
        case '9' : escreveFuncionarios (head);
```

```
        break;
```

```
        default:
```

```
        break;
```

```
    } // fim switch
```

```
    }while (escolha != '9');
```

```
    return head;
```

```
} // fim função
```

```
/*
```

```
menu para inserir os dados para cada funcionario
```

```
*/
```

```
struct listaFuncionarios * menuCriaNovoFuncionario(){
```

```
    int tamanho; // variavel para validar tamanhos de inteiros
```

```
    struct listaFuncionarios *novoFuncionario =(struct listaFuncionarios *) malloc (sizeof (struct
    listaFuncionarios));
```

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

```

printf("%93s\n", "_____");
printf ("%88s", "CRIAR NOVO FUNCIONARIO\n");
printf("%93s\n", "_____");

do{
    printf("%61s", " ");
    alteraCorTexto(8); printf("Nome proprio: "); alteraCorTexto(15);
} while (!validaLetras(gets(novoFuncionario->funcionarios.dadosPessoais.nomeProprio)));

do{
    printf("%61s", " ");
    alteraCorTexto(8); printf("Apelido: "); alteraCorTexto(15);
} while (!validaLetras(gets(novoFuncionario->funcionarios.dadosPessoais.apelido)));

do{
    printf("%61s", " ");
    alteraCorTexto(8); printf("Nif: "); alteraCorTexto(15);
    scanf("%d", &novoFuncionario->funcionarios.dadosPessoais.nif);
    tamanho = validaNumInteiros(novoFuncionario->funcionarios.dadosPessoais.nif);
    fflush(stdin);
} while (tamanho != 9);

do{
    printf("%61s", " ");
    alteraCorTexto(8); printf("Telefone: "); alteraCorTexto(15);
    scanf("%d", &novoFuncionario->funcionarios.dadosPessoais.telefone);
    tamanho = validaNumInteiros(novoFuncionario->funcionarios.dadosPessoais.telefone);
    fflush(stdin);
} while (tamanho != 9);

printf("%61s", " "); alteraCorTexto(8); printf("Cargo:\n");
printf("%61s", " "); printf("\tNORMAL-0 \n");
printf("%61s", " "); printf("\tADMINISTRATIVO-1\n"); alteraCorTexto(15);
do{
    printf("%61s", " "); scanf("%d", &novoFuncionario->funcionarios.cargo);
    fflush(stdin);
} while (novoFuncionario->funcionarios.cargo != NORMAL && novoFuncionario->funcionarios.cargo != ADMINISTRATIVO);

printf("%61s", " ");
alteraCorTexto(8); printf("Username para aceder o sistema: "); alteraCorTexto(15);
gets(novoFuncionario->funcionarios.username);
printf("%61s", " ");
alteraCorTexto(8); printf("Password para aceder o sistema: "); alteraCorTexto(15);
gets(novoFuncionario->funcionarios.password);
novoFuncionario->next = NULL;

return novoFuncionario;

} // fim função

/*
 * menu para perguntar ao utilizador qual os campos do funcionario
 * que são alterados
 */
void menuEditarFuncionario( struct listaFuncionarios * current ){
    int tamanho = 0; // index para o num de inteiro num numero

    printf("%61s", " ");
    alteraCorTexto(8); printf("Nif: %d\n", current->funcionarios.dadosPessoais.nif);
    printf("%61s", " ");
    printf("Nome proprio: %s\n", current->funcionarios.dadosPessoais.nomeProprio);

    printf("%61s", " ");
    printf("Apelido: "); alteraCorTexto(15); printf("%s\n", current->funcionarios.dadosPessoais.apelido);
    if (perguntaUtilizador()){
        do{
            printf("%61s", " ");
        } while (!validaLetras(gets(current->funcionarios.dadosPessoais.apelido)));
    }
    printf("%61s", " ");

```

```

    alteraCorTexto(8); printf("Telefone: "); alteraCorTexto(15); printf("%d\n", current->funcionarios.
    dadosPessoais.telefone);
    if (perguntaUtilizador()){
        do{
            printf("%61s", " "); scanf("%d", &current->funcionarios.dadosPessoais.telefone);
            tamanho = validaNumInteiros(current->funcionarios.dadosPessoais.telefone);
            fflush(stdin);
        } while (tamanho != 9);
    }
    printf("%61s", " ");
    alteraCorTexto(8); printf("Cargo: "); alteraCorTexto(15); printf("%d\n", current->funcionarios.cargo);
    if (perguntaUtilizador()){
        do{
            printf("%61s", " "); scanf("%d", &current->funcionarios.cargo);
            fflush(stdin);
        } while (current->funcionarios.cargo != NORMAL && current->funcionarios.cargo != ADMINISTRATIVO);
    }
    printf("%61s", " ");
    alteraCorTexto(8); printf("Username para aceder o sistema: "); alteraCorTexto(15); printf("%s\n",
    current->funcionarios.username);
    if (perguntaUtilizador()){
        printf("%61s", " "); gets(current->funcionarios.username);
    }
    printf("%61s", " ");
    alteraCorTexto(8); printf("Password para aceder o sistema: "); alteraCorTexto(15); printf("%s\n",
    current->funcionarios.password);
    if (perguntaUtilizador()){
        printf("%61s", " "); gets(current->funcionarios.password);
    }
    printf ("\n");

} // fim função

/*
* função utilizada para perguntar ao utilizador se pretende editar (sim/não)
*/
int perguntaUtilizador(){
    char alterar;

    printf("%61s", " "); printf(" Pretende alterar (s/n):");
    alterar = _getche();
    fflush(stdin);
    printf("\n");

    if ( alterar == 's' || alterar == 'S'){
        return 1;
    }

    return 0;
} // fim função

/*
* função utilizada para login do sistema
*/
void menuLogin(char * username, char * password){
    int i= 0; // index para construí a password um char de cada vez
    char ch;
    int static index = 0; // numero de tentativas de login

    system("cls");
    setEcra(); // função para o tamanho do command prompt

    printf("%93s\n", " ");
    //primeiro login
    if (index == 0)
        printf("%78s\n", "LOGIN");
    // login incorreto
    else{
        alteraCorTexto(12);
        printf("%85s", "LOGIN INCORRETO\n");
        alteraCorTexto(15);
    }
}

```

```

printf("%93s\n", "_____");
printf("%61s", " "); alteraCorTexto(8); printf("Username:"); alteraCorTexto(15);
gets(username);
printf ("\n");
printf("%61s", " "); alteraCorTexto(8); printf("Password:"); alteraCorTexto(15);

while (1){
    ch = _getch();
    // qualquer char
    if (ch != 13 && ch != 8){
        password [i++] = ch;
        printf ("* \b");
    }//tecla Backspace
    else if (ch == 8){
        printf("\b \b");
        i--;
    }//tecla Enter
    else{
        password[i] = '\0';
        index++; // incrementar o numero de tentativas
        break;
    }
} // fim while

} // fim função

/*
 * função utilizada para alterar a cor do texto
 */
void alteraCorTexto(int cor){

    HANDLE hConsole;
    hConsole = GetStdHandle(STD_OUTPUT_HANDLE);

    SetConsoleTextAttribute(hConsole, cor);
} // fim função

/*
 * menu para o funcionario administrativo
 */
void menuAdministrativos(struct listaFuncionarios * utilizador,
                        struct listaFuncionarios * funcionariosHead,
                        struct listaClientes * clientesHead,
                        struct listaSimulacoes * simulacoesHead){

    struct listaFuncionarios * headFunc = funcionariosHead;

    char user[50];
    char escolha;

    strcpy(user, utilizador->funcionarios.dadosPessoais.nomeProprio);
    strcat(user, " ");
    strcat(user, utilizador->funcionarios.dadosPessoais.apelido);

    do{

        system("cls");
        printf("Utilizador: %s \n", user);
        printf("\n\n");
        printf("%93s\n", "_____");
        printf("%79s", "MENU\n");
        printf("%93s\n", "_____");
        printf("%61s", " ");
        alteraCorTexto(8); printf("1-"); alteraCorTexto(15); printf(" Funcionarios\n");
        printf("%61s", " ");
        alteraCorTexto(8); printf("2-"); alteraCorTexto(15); printf(" Clientes\n");
        printf("%61s", " ");
        alteraCorTexto(8); printf("3-"); alteraCorTexto(15); printf(" Simular Credito\n");
        printf("%61s", " ");
        alteraCorTexto(8); printf("9-"); alteraCorTexto(15); printf(" Sair\n");
        printf("%93s\n", "_____");
    }
}

```

```

    printf("%61s", " "); escolha = _getche();
    fflush(stdin);

    switch (escolha){
        case '1': headFunc = menuControloFuncionarios(headFunc);//menu para controlar funções dos
funcionarios
        break;
        case '2': clientesHead = menuControloCliente(clientesHead);
        break;
        case '3': menuSimularCredito(user, clientesHead, simulacoesHead);
        break;
        case '9': libertaLista(headFunc); // função para libertar a memoria reservada
Funcionarios
        libertaListaClientes(clientesHead); // função para libertar a memoria reservada
Clientes
        printf("sair");

        break;
        default:
        break;
    }// fim switch

} while (escolha != '9');

} // fim função

void menuSimularCredito(char user[50], struct listaClientes * head, struct listaSimulacoes * simulacoesHead)
{
    //struct listaClientes * clienteAtual = NULL;
    char escolha;

    do{

        system("cls");
        printf("Utilizador: %s \n", user);
        printf("\n\n");
        printf("%93s\n", " ");
        printf("%85s", "SIMULADOR CREDITO\n");
        printf("%93s\n", " ");
        printf("%61s", " ");
        alteraCorTexto(8); printf("1-"); alteraCorTexto(15); printf(" Numero Identificacao Fiscal\n");
        printf("%61s", " ");
        alteraCorTexto(8); printf("2-"); alteraCorTexto(15); printf(" Apelido\n");
        printf("%61s", " ");
        alteraCorTexto(8); printf("9-"); alteraCorTexto(15); printf(" Sair\n");
        printf("%93s\n", " ");

        printf("%61s", " "); escolha = _getche();
        fflush(stdin);
        printf("\n");

        switch (escolha){
            case '1': pedeDadosPessoais(user, head, simulacoesHead);
            break;
            case '2': printf ("procura por apelido\n");
            break;
            case '9': printf("sair");
            break;
            default:
            break;
        }// fim switch

    } while (escolha != '9');

} // fim função

/*
* menu para gestão do cliente
*/
struct listaClientes * menuControloCliente(struct listaClientes * head){
    int escolha; // input do utilizador

```

```

do{
    system("cls");

    printf("%93s\n", "_____");
    printf("%84s", "GESTAO CLIENTES\n");
    printf("%93s\n", "_____");
    printf("%61s", " ");
    alteraCorTexto(8); printf("1- "); alteraCorTexto(15); printf("Criar\n");
    printf("%61s", " ");
    alteraCorTexto(8); printf("2- "); alteraCorTexto(15); printf("Editar\n");
    printf("%61s", " ");
    alteraCorTexto(8); printf("3- "); alteraCorTexto(15); printf("Listar\n");
    printf("%61s", " ");
    alteraCorTexto(8); printf("4- "); alteraCorTexto(15); printf("Apagar\n");
    printf("%61s", " ");
    alteraCorTexto(8); printf("9- "); alteraCorTexto(15); printf("Sair\n");
    printf("%93s\n", "_____");

    printf("%61s", " "); escolha = _getche();
    fflush(stdin);
    printf("\n");

    switch (escolha){
        case '1': head = adicionarCliente(head);
            break;
        case '2': editarCliente(head);
            break;
        case '3': head = imprimeClientes(head);
            break;
        case '4': head = apagaCliente (head);
            break;
        case '9': escreveClientes(head);
            break;
        default:
            break;
    }// fim switch

} while (escolha != '9');

return head;
} //fim função

/*
menu para inserir os dados para cada cliente
*/
struct listaClientes * menuCriaNovoCliente(){
    int tamanho = 0;

    system("cls");

    struct listaClientes *novoCliente = (struct listaClientes *) malloc(sizeof(struct listaClientes));
    system("cls");

    printf("%93s\n", "_____");
    printf("%86s", "CRIAR NOVO CLIENTE\n");
    printf("%93s\n", "_____");

    do{
        printf("%61s", " "); alteraCorTexto(8); printf("Nome proprio:"); alteraCorTexto(15);
    } while (!validaLetras(gets(novoCliente->cliente.dadosClt.nomeProprio)));

    do{
        printf("%61s", " "); alteraCorTexto(8); printf("Apelido:"); alteraCorTexto(15);
    }while (!validaLetras(gets(novoCliente->cliente.dadosClt.apelido)));

    do{
        printf("%61s", " "); alteraCorTexto(8); printf("Nif:"); alteraCorTexto(15);
        scanf("%d", &novoCliente->cliente.dadosClt.nif);
        tamanho = validaNumInteiros(novoCliente->cliente.dadosClt.nif);
        fflush(stdin);
    } while (tamanho != 9);

```

```

do{
    printf("%61s", " "); alteraCorTexto(8); printf("Telefone:"); alteraCorTexto(15);
    scanf("%d", &novoCliente->cliente.dadosClt.telefone);
    tamanho = validaNumInteiros(novoCliente->cliente.dadosClt.telefone);
    fflush(stdin);
} while (tamanho != 9);

do{
    printf("%61s", " "); alteraCorTexto(8); printf("Data nascimento(dia/mes/ano):"); alteraCorTexto(15);
    scanf("%d/%d/%d", &novoCliente->cliente.dataNascimento.dia,
        &novoCliente->cliente.dataNascimento.mes,
        &novoCliente->cliente.dataNascimento.ano);
    fflush(stdin);
} while (!validaData(novoCliente->cliente.dataNascimento.dia,
    novoCliente->cliente.dataNascimento.mes,
    novoCliente->cliente.dataNascimento.ano));
printf("%61s", " "); alteraCorTexto(8); printf("Morada\tRua:"); alteraCorTexto(15);
gets(novoCliente->cliente.moradaClt.rua);
printf("%61s", " "); alteraCorTexto(8); printf("\tLocalidade:"); alteraCorTexto(15);
gets(novoCliente->cliente.moradaClt.localidade);
printf("%61s", " "); alteraCorTexto(8); printf("\tCodigo Postal:"); alteraCorTexto(15);
gets(novoCliente->cliente.moradaClt.codigoPostal);

novoCliente->next = NULL;

return novoCliente;

} // fim função

void setEcra(){

    struct small_rect test;

    HANDLE hStdout;
    COORD coord;
    BOOL ok;

    hStdout = GetStdHandle(STD_OUTPUT_HANDLE);
    coord.X = PADX;
    coord.Y = PADY;
    ok = SetConsoleScreenBufferSize(hStdout, coord);

    test.Left = 0;
    test.Top = 0;
    test.Right = coord.X - 1;
    test.Bottom = coord.Y - 1;

    SetConsoleWindowInfo(hStdout, ok, &test);
} // fim funcao

/*
* menu para perguntar ao utilizador qual os campos do cliente
* que são alterados
*/
void menuEditarCliente(struct listaClientes *current){
    int tamanho = 0; // index para o numero de inteiros em um numero
    printf("\n*****\n");

    /* não pode ser alterado */
    printf("%61s", " "); alteraCorTexto(8); printf("Nif: %d\n", current->cliente.dadosClt.nif);

    printf("%61s", " "); printf("Data nascimento(dia/mes/ano): %d/%d/%d\n", current->cliente.dataNascimento.
    dia,
        current->cliente.dataNascimento.mes,
        current->cliente.dataNascimento.ano);
    printf("%61s", " "); printf("Nome proprio: %s\n", current->cliente.dadosClt.nomeProprio);

    printf("%61s", " ");
    alteraCorTexto(8); printf("Apelido: "); alteraCorTexto(15); printf("%s\n", current->cliente.dadosClt.
    apelido);

    if (perguntaUtilizador()){
        do{

```

```

        printf("%61s", " ");
    } while (!validaLetras(gets(current->cliente.dadosClt.apelido)));
}
printf("%61s", " ");
alteraCorTexto(8); printf("Telefone: "); alteraCorTexto(15); printf("%d\n", &current->cliente.dadosClt.
telefone);
if (perguntaUtilizador()){
    do{
        printf("%61s", " "); scanf("%d", &current->cliente.dadosClt.telefone);
        tamanho = validaNumInteiros(current->cliente.dadosClt.telefone);
        fflush(stdin);
    } while (tamanho != 9);
}

printf("%61s", " ");
alteraCorTexto(8); printf("Morada\tRua: "); alteraCorTexto(15); printf("%s\n", current->cliente.
moradaClt.rua);
if (perguntaUtilizador()){
    printf("%61s", " "); gets(current->cliente.moradaClt.rua);
}
printf("%61s", " ");
alteraCorTexto(8); printf("\tLocalidade: "); alteraCorTexto(15); printf("%s\n", current->cliente.
moradaClt.localidade);
if (perguntaUtilizador()){
    printf("%61s", " "); gets(current->cliente.moradaClt.localidade);
}
printf("%61s", " ");
alteraCorTexto(8); printf("\tCodigo Postal: "); alteraCorTexto(15); printf("%s\n", current->cliente.
moradaClt.codigoPostal);
if (perguntaUtilizador()){
    printf("%61s", " "); gets(current->cliente.moradaClt.codigoPostal);
}
printf("\n");

} // fim função

/*
*  menu para o funcionario normal sem poderes administrativos
*/
void menuFuncionario(struct listaFuncionarios * utilizador, struct listaClientes * clientesHead , struct
listaSimulacoes * simulacoesHead){

    //struct listaFuncionarios * headFunc = funcionariosHead;

    char user[50];
    char escolha;

    strcpy(user, utilizador->funcionarios.dadosPessoais.nomeProprio);
    strcat(user, " ");
    strcat(user, utilizador->funcionarios.dadosPessoais.apelido);

    do{

        system("cls");
        printf("Utilizador: %s \n", user);
        printf("\n\n");
        printf("%93s\n", "_____");
        printf("%79s", "MENU\n");
        printf("%93s\n", "_____");
        printf("%61s", " ");
        alteraCorTexto(8); printf("1-"); alteraCorTexto(15); printf(" Menu Clientes\n");
        printf("%61s", " ");
        alteraCorTexto(8); printf("2-"); alteraCorTexto(15); printf(" Simular Credito\n");
        printf("%61s", " ");
        alteraCorTexto(8); printf("9-"); alteraCorTexto(15); printf(" Sair\n");
        printf("%93s\n", "_____");

        printf("%61s", " "); escolha = _getche();
        fflush(stdin);

        switch (escolha){
            case '1': clientesHead = menuControloCliente(clientesHead);
                break;

```



```

        case '2': menuSimularCredito(user, clientesHead, simulacoesHead);
            break;

        case '9': libertaListaClientes(clientesHead); // função para libertar a memoria reservada Clientes

            break;
        default:
            break;
    } // fim switch

} while (escolha != '9');

} // fim função

void menuOutputSimulacao(char user[50], struct dadosPessoaisSimulacao * dadosSimulacao, struct
    dadosPessoaisSimulacao * simulacaoHead){
    //struct listaClientes * clienteAtual = NULL;
    char escolha;

    do{

        system("cls");
        printf("Utilizador: %s \n", user);
        printf("\n\n");
        printf("%93s\n", "_____");
        printf("%89s", "OUTPUT SIMULACAO CREDITO\n");
        printf("%93s\n", "_____");
        printf("%61s", " ");
        alteraCorTexto(8); printf("1-"); alteraCorTexto(15); printf(" Ecargo Mensal\n");
        printf("%61s", " ");
        alteraCorTexto(8); printf("2-"); alteraCorTexto(15); printf(" Custo do Empréstimo-nao funciona\n");
        printf("%61s", " ");
        alteraCorTexto(8); printf("3-"); alteraCorTexto(15); printf(" Dados do Credito-nao funciona\n");
        printf("%61s", " ");
        alteraCorTexto(8); printf("4-"); alteraCorTexto(15); printf(" Resumo da Simulacao-nao funciona\n");
        printf("%61s", " ");
        alteraCorTexto(8); printf("5-"); alteraCorTexto(15); printf(" Documentos Necessarios-nao funciona\n"
    );
        printf("%61s", " ");
        alteraCorTexto(8); printf("6-"); alteraCorTexto(15); printf(" Plano Financeiro do Empréstimo-nao
funciona\n");
        printf("%61s", " ");
        alteraCorTexto(8); printf("7-"); alteraCorTexto(15); printf(" Guardar Simulacao-nao funciona\n");
        printf("%61s", " ");
        alteraCorTexto(8); printf("8-"); alteraCorTexto(15); printf(" Carrega Anterior-nao funciona\n");
        printf("%61s", " ");
        alteraCorTexto(8); printf("9-"); alteraCorTexto(15); printf(" Sair\n");
        printf("%93s\n", "_____");

        printf("%61s", " "); escolha = _getche();
        fflush(stdin);
        printf("\n");

        switch (escolha){
            case '1': encargosMensais(dadosSimulacao);
                break;
            case '2': printf("2\n");
                break;
            case '3': printf("3\n");
                break;
            case '4': printf("4\n");
                break;
            case '5': printf("5\n");
                break;
            case '6': printf("6\n");
                break;
            case '7': //guardarSimulacao(dadosSimulacao,simulacaoHead);
                break;
            case '8': printf("8\n");
                break;
            case '9': printf("sair");
                break;
            default:

```

```
        break;
    }// fim switch

} while (escolha != '9');

system("pause");

}// fim função
```