```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "estruturas.h"

/*****************************************/
/*Samuel Anjos nº 13305                  */
/*Luis Camilo nº13987                   */
/*****************************************/

/*
*   adiciona o novo funcionario
*/
struct listaFuncionarios * adicionarFuncionario ( struct listaFuncionarios * head){

    struct listaFuncionarios *current;
    current = (struct listaFuncionarios*) malloc (sizeof(struct listaFuncionarios));

    /* verificar se é o primeiro node, se for criar o funcionario */
    if (head == NULL)
        head = menuCriaNovoFuncionario();

    else{
        /* ir até ao ultimo node */
        current = head;
        while (current->next != NULL)
            current = current->next;

        /* criar novo node */

        current->next = menuCriaNovoFuncionario();
    }

    return head;

}// fim função

/*
*   imprime a lista de funcionarios
*/
struct listaFuncionarios * imprimeFuncionarios(struct listaFuncionarios * head){

    struct listaFuncionarios * current;

    /* verificar se a lista tem nodes */
    if (head == NULL){
        printf("%61s", " "); printf("Não existem funcionarios!\n");
    }

    current = head;

    while (current != NULL){
        /* imprime node */
        printFuncionario(current);
        /* proximo node */
        current = current->next;
    }
    printf("\n");
    alteraCorTexto(8);
    printf("%0s\n", "----------------------------------------------------------------\n");
    alteraCorTexto(15);
    system("pause");

    return head;

}// fim função

/*
*   imprime no ecra um node
*/
void printFuncionario(struct listaFuncionarios * current){
```

```
    printf("\n");
    alteraCorTexto(8);
    printf("%0s\n", "------------------------------------------------------------\n");
    alteraCorTexto(15);
    printf("Nome proprio:%s", current->funcionarios.dadosPessoais.nomeProprio);
    printf("\nApelido:%s", current->funcionarios.dadosPessoais.apelido);
    printf("\nNif:%d", current->funcionarios.dadosPessoais.nif);
    printf("\nTelefone:%d", current->funcionarios.dadosPessoais.telefone);
    printf("\nCargo:%d", current->funcionarios.cargo);
    printf("\nUsername para aceder o sistema:%s", current->funcionarios.username);
    printf("\nPassword para aceder o sistema:%s\n", current->funcionarios.password);

}// fim função
/*
*   editar o funcionarios atravez do seu nif
*/
void editarFuncionario(struct listaFuncionarios * head){

    /* campo pelo qual pretendemos pesquisa a lista ligada */
    int nif;
    /* struct que pretendemos para  editar */
    struct listaFuncionarios * funcionarioPretendido;
    /* perguntar ao utilizador qual o nif do funcionario */
    printf("%61s", " "); printf("Introduza o nif do funcionario:");
    scanf ("%d", &nif);
    fflush(stdin);

    /* funcionario pretendido para editar */
    funcionarioPretendido = procuraNodeFuncionario (head, &nif);
    /* funcionario existe*/
    if (funcionarioPretendido!=NULL)
        /* editar funcionario */
        menuEditarFuncionario(funcionarioPretendido);

    else{
        printf("%61s", " ");
        alteraCorTexto(12); printf("Esse funcionario nao exite.\n"); alteraCorTexto(15);
    }
    system("pause");

}// fim função

/*
*   procura um node na lista ligada por nif
*/
struct listaFuncionarios * procuraNodeFuncionario( struct listaFuncionarios * head , int *nif ){

    struct listaFuncionarios * current;

    current = head;
    /* traversa a lista */
    while (current != NULL){
        if (current->funcionarios.dadosPessoais.nif == *nif)
            return current;
        /* avança para o proximo node */
        current = current->next;
    }

    return NULL;

}// fim função

/*
*   função para procurar o utilizador por username e password na lista ligada
*/
struct listaFuncionarios * procuraFuncionarioLogin(struct listaFuncionarios * head, char * username, char *
    password){

    struct listaFuncionarios * utilizador;

    utilizador = head;
    /* traversa a lista */
    while (utilizador != NULL){
        if ((strcmp(utilizador->funcionarios.username, username) == 0) && (strcmp(utilizador->funcionarios.
```

```
    password, password) == 0))
            return utilizador;
        /* avança para o proximo node */
        utilizador = utilizador->next;
    }

    return utilizador = NULL;

}// fim função


/*
*   pede ao utilizador o nif do funcionario que deseja apagar do sistema
*/
struct listaFuncionarios * apagaFuncionario(struct listaFuncionarios * head){
    /* campo pelo qual pretendemos pesquisa a lista ligada */
    int nif;
    /* struct que pretendemos para  editar */
    struct listaFuncionarios * funcionarioPretendido;
    /* perguntar ao utilizador qual o nif do funcionario */
    printf("%61s", " "); printf("Introduza o nif do funcionario:");
    scanf ("%d", &nif);
    fflush(stdin);

    /* funcionario pretendido para apagar */
    funcionarioPretendido = procuraNodeFuncionario (head, &nif);
    /* funcionario existe*/
    if (funcionarioPretendido != NULL){
        /*apagar funcionario da lista */
        head = removerFuncionario(head, funcionarioPretendido);

        printf("%61s", " ");
        alteraCorTexto(10); printf("O funcionario foi apagado com sucesso!\n"); alteraCorTexto(15);

    }/* funcionario não exite*/
    else{
        printf("%61s", " ");
        alteraCorTexto(12); printf("Esse funcionario nao exist.\n"); alteraCorTexto(15);
    }
    system("pause");

    return head;
}// fim função

/*
*   remover a cabeça da lista ligada
*/
struct listaFuncionarios* deleteHeadFuncionario(struct listaFuncionarios*head){
    struct listaFuncionarios * primeiro = head;

    //struct funcionario * primeiro = head;
    head = head->next;

    /* confirmar que exite mais que um node */
    if (primeiro == head)
        head = NULL;

    primeiro = NULL;
    free (primeiro);

    return head;
}//fim função
/*
*   remover o ultimo node da lista ligada
*/
struct listaFuncionarios * deleteUltimoFuncionario (struct listaFuncionarios * head){
    struct listaFuncionarios * current = head;
    struct listaFuncionarios * ultimo = NULL;

    /* transversar a lista até ao penultimo */
    while ( current->next != NULL){
        ultimo = current ;
        current = current->next;
    }
```

```
    if (ultimo != NULL)
        ultimo->next = NULL;

    free(current);

    return head;

}// fim função

/*
*   remove qualque node
*/
struct listaFuncionarios * removerFuncionario(struct listaFuncionarios *head, struct listaFuncionarios *
    current){
    struct listaFuncionarios * temp = head;

    if (current == NULL)
        return NULL;

    /*se o node é o primeiro */
    if (current == head){
        //return head;
        return deleteHeadFuncionario(head);
    }

    /*se o node é o ultimo */
    if (current->next == NULL)
        return deleteUltimoFuncionario (head);

    /*se o node é no meio da lista ligada */
    while (temp != NULL){
        if (temp->next == current)
            break;
        temp = temp->next;
    }

    struct listaFuncionarios * temp2 = temp->next;
    temp->next = temp2->next;
    temp2->next = NULL;
    free (temp2);

    return head;

}//fim função

/*
*   grava funcionarios no ficheiro binario
*/
void escreveFuncionarios ( struct listaFuncionarios * head){
    FILE *fp;
    struct listaFuncionarios *current;
    /* chama a função para abri o ficheiro */
    fp = abreFicheiroBinario ("funcionarios.dat", "wb");

    current = head;
    /* escreve no ficheiro um node de cada vez da lista */
    while (current != NULL){
        fwrite (current, sizeof (struct listaFuncionarios),1, fp);
        current = current->next;
    }

    fclose (fp);

}// fim função

/*
*   função para liebertar a lista de memoria
*/
void libertaLista(struct listaFuncionarios *head){
    struct listaFuncionarios *temp, *current = head;

    while (current != NULL){
        temp = current->next;
```

```
        free(current);  // liberta o node
        current = temp;
    }
}// fim função

/*
*   carregamento de funcionarios de ficheiro binarios
*/
struct listaFuncionarios * carregaFuncionarios (struct listaFuncionarios * head){

    FILE *fp;
    struct listaFuncionarios  *temp;
    struct listaFuncionarios *current ;

    fp = abreFicheiroBinario ( "funcionarios.dat" , "rb");

    /* percorre o ficheio até ao fim */
    while ( !feof(fp)){
        //current = (struct funcionario *) malloc (sizeof (struct funcionario));
        if ((current = (struct listaFuncionarios *) malloc (sizeof (struct listaFuncionarios)))==NULL){
            printf("Erro: reservad de memoria carrega funcionarios.");
            exit (1);
        }
        /* devolve 0 qunado não tem sucesso */
        if (fread (current, sizeof(struct listaFuncionarios),1 , fp)>0){

            current->next = NULL;
            temp = head;

            /* preencher o primeiro node */
            if (head == NULL){
                head = current;
            }
            /* preencher o resto dos nodes */
            else{

                while (temp->next != NULL)
                    temp = temp->next;

                temp->next = current;
            }

        }// fim if
    } // fim while

    fclose (fp);

    return head;
}// fim função

/*
* função para realizar o login in e saber quem é o utilizador
*/

struct listaFuncionarios * loginFuncionarios(struct  listaFuncionarios * head){
    char username[20];
    char password[10];

    struct listaFuncionarios * utilizador =NULL;

    while (utilizador == NULL){

        menuLogin(username, password);
        utilizador = procuraFuncionarioLogin(head, username, password);

    }

    return utilizador;
}// fim função
```