****

**Projeto Final I.P**

***Gestão de Requisição e Devolução de Equipamentos Informáticos***

**Relatório**

**Objetivo:**

O código implementa um sistema simples de gerenciamento de requisições e devoluções de equipamentos informáticos. Utilizando estruturas e funções, o programa oferece funcionalidades como adição de requisições, aprovação ou rejeição, registro de devolução, e listagem de requisições por diferentes critérios.

**Estrutura do Código:**

- Estrutura de Dados (`Requisicao`):

- Armazena informações sobre cada requisição, incluindo usuário, equipamento, estado (pendente, aprovada, rejeitada), e indicador de devolução.

**- Funções:**

- `adicionarRequisicao`: Adiciona uma nova requisição ao sistema.

- `listarRequisicoes`: Exibe a lista de todas as requisições com detalhes.

- `aprovarRequisicao` e `rejeitarRequisicao`: Aprovam ou rejeitam uma requisição com base no nome do usuário.

- `registrarDevolucao`: Registra a devolução de um equipamento aprovado.

- `listarRequisicoesPorUsuario`: Lista as requisições de um usuário específico.

- `listarRequisicoesAprovadas` e `listarRequisicoesDevolvidas`: Mostram as requisições aprovadas e devolvidas, respectivamente.

**Execução Principal (`main`):**

**- Menu Interativo:**

- Apresenta um menu de opções numeradas para o usuário.

- Aceita a entrada do usuário para escolher a operação desejada.

- Chama as funções correspondentes com base na escolha do usuário.

**- Array de Requisições:**

- Utiliza um array de estruturas `Requisicao` para armazenar até 100 requisições.

***Observações:***

- A estrutura do código é modular, facilitando a manutenção e a adição de novas funcionalidades.

- O código possui verificações para garantir que ações como aprovação, rejeição e devolução só ocorram em situações válidas.

- A mensagem informativa no caso de uma opção inválida ajuda na usabilidade do programa.

**Conclusão:**

O código fornece uma implementação funcional de um sistema básico de gestão de requisições e devoluções de equipamentos informáticos. Ele serve como uma base sólida para extensões e melhorias futuras, proporcionando uma experiência interativa para o usuário.

**Código C:**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

typedef struct {

char usuario[50];

char equipamento[20];

char estado[30];

int devolvido;

} Requisicao;

void adicionarRequisicao(Requisicao requisicoes[], int \*numRequisicoes) {

printf("Nome do usuario: ");

scanf("%s", requisicoes[\*numRequisicoes].usuario);

printf("Equipamento desejado: ");

scanf("%s", requisicoes[\*numRequisicoes].equipamento);

strcpy(requisicoes[\*numRequisicoes].estado, "Pendente");

requisicoes[\*numRequisicoes].devolvido = 0;

(\*numRequisicoes)++;

}

void listarRequisicoes(Requisicao requisicoes[], int numRequisicoes) {

printf("\nLista de Requisicoes:\n");

for (int i = 0; i < numRequisicoes; i++) {

printf("Usuario: %s, Equipamento: %s, Estado: %s, Devolvido: %s\n",

requisicoes[i].usuario, requisicoes[i].equipamento,

requisicoes[i].estado, (requisicoes[i].devolvido) ? "Sim" : "Nao");

}

}

void aprovarRequisicao(Requisicao requisicoes[], int numRequisicoes) {

char usuario[50];

printf("Digite o nome do usuario para aprovar a requisicao: ");

scanf("%s", usuario);

for (int i = 0; i < numRequisicoes; i++) {

if (strcmp(requisicoes[i].usuario, usuario) == 0 && strcmp(requisicoes[i].estado,"Pendente") == 0) {

strcpy(requisicoes[i].estado, "Aprovada");

printf("Requisicao aprovada!\n");

return;

}

}

printf("Requisicao nao encontrada ou ja aprovada/rejeitada.\n");

}

void rejeitarRequisicao(Requisicao requisicoes[], int numRequisicoes) {

char usuario[50];

printf("Digite o nome do usuario para rejeitar a requisicao: ");

scanf("%s", usuario);

for (int i = 0; i < numRequisicoes; i++) {

if (strcmp(requisicoes[i].usuario, usuario) == 0 && strcmp(requisicoes[i].estado,"Pendente") == 0) {

strcpy(requisicoes[i].estado, "Rejeitada");

printf("Requisicao rejeitada!\n");

return;

}

}

printf("Requisicao nao encontrada ou ja foi aprovada/rejeitada.\n"); }

void registrarDevolucao(Requisicao requisicoes[], int numRequisicoes) {

char usuario[50];

printf("Introduza o nome do usuario para registrar a devolucao: ");

scanf("%s", usuario);

for (int i = 0; i < numRequisicoes; i++) {

if (strcmp(requisicoes[i].usuario, usuario) == 0 && strcmp(requisicoes[i].estado, "Aprovada") == 0) {

requisicoes[i].devolvido = 1;

printf("Devolucao registrada com sucesso!\n");

return;

}

}

printf("Usuario nao foi encontrado ou requisicao nao foi aprovada.\n");

}

void listarRequisicoesPorUsuario(Requisicao requisicoes[], int numRequisicoes) {

char usuario[50];

printf("Introduza o nome do usuario para listar as requisicoes: ");

scanf("%s", usuario);

printf("\nRequisicoes de %s:\n", usuario);

for (int i = 0; i < numRequisicoes; i++) {

if (strcmp(requisicoes[i].usuario, usuario) == 0) {

printf("Equipamento: %s, Estado: %s, Devolvido: %s\n",

requisicoes[i].equipamento, requisicoes[i].estado,

(requisicoes[i].devolvido) ? "Sim" : "Nao");

}

}

}

void listarRequisicoesAprovadas(Requisicao requisicoes[], int numRequisicoes) {

printf("\nRequisicoes Aprovadas:\n");

for (int i = 0; i < numRequisicoes; i++) {

if (strcmp(requisicoes[i].estado, "Aprovada") == 0) {

printf("Usuario: %s, Equipamento: %s, Devolvido: %s\n",

requisicoes[i].usuario, requisicoes[i].equipamento,

(requisicoes[i].devolvido) ? "Sim" : "Nao");

}

}

}

void listarRequisicoesDevolvidas(Requisicao requisicoes[], int numRequisicoes) {

printf("\nRequisicoes Devolvidas:\n");

for (int i = 0; i < numRequisicoes; i++) {

if (requisicoes[i].devolvido) {

printf("Usuario: %s, Equipamento: %s\n",

requisicoes[i].usuario, requisicoes[i].equipamento);

}

}

}

int main() {

Requisicao requisicoes[100];

int numRequisicoes = 0;

int opcao;

do {

printf("\nSistema Gestao de Requisicao e Devolucao de Equipamentos Informaticos\n");

printf("----Menu----\n\n");

printf("1. Adicionar Requisicao\n");

printf("2. Listar Requisicoes\n");

printf("3. Aprovar Requisicao\n");

printf("4. Rejeitar Requisicao\n");

printf("5. Registrar Devolucao\n");

printf("6. Listar Requisicoes por Usuario\n");

printf("7. Listar Requisicoes Aprovadas\n");

printf("8. Listar Requisicoes Devolvidas\n");

printf("9.+Info.\n");

printf("10. Sair\n\n");

printf("Digite a opcao que deseja: ");

scanf("%d", &opcao);

switch (opcao) {

case 1:

adicionarRequisicao(requisicoes, &numRequisicoes);

break;

case 2:

listarRequisicoes(requisicoes, numRequisicoes);

break;

case 3:

aprovarRequisicao(requisicoes, numRequisicoes);

break;

case 4:

rejeitarRequisicao(requisicoes, numRequisicoes);

break;

case 5:

registrarDevolucao(requisicoes, numRequisicoes);

break;

case 6:

listarRequisicoesPorUsuario(requisicoes, numRequisicoes);

break;

case 7:

listarRequisicoesAprovadas(requisicoes, numRequisicoes);

break;

case 8:

listarRequisicoesDevolvidas(requisicoes, numRequisicoes);

break;

case 9:

printf("\nSistema de Gestao desenvolvido por:\n-ERIC REIS\n-LIEDSON DELGADO\n-TIAGO DONGO\nFoi criado em:30/01/2024\n\n");

break;

case 10:

printf("Encerrando o programa.\n");

break;

default:

printf("[OPCAO INVALIDA]\nTente novamente.\n\n");

}

} while (opcao != 10);

}

***Elementos do Grupo:***

*-Eric Reis*

*-Liedson Delgado*

*-Tiago Dongo*