



PROJETO DE ESTRUTURA DE DADOS AVANÇADA

João Arantes a21600



18 DE MARÇO DE 2023

ESTRUTURA DE DADOS AVANÇADA
Engenharia de Sitemas Informáticos

Índice

Índice	1
Introdução	3
Apresentação do Código	4

Indice de imagens

Figura 1.....	4
Figura 2.....	5
Figura 3.....	6
Figura 4.....	8
Figura 5.....	9
Figura 6.....	10
Figura 7.....	12
Figura 8.....	13

Introdução

Este projeto de avaliação de realização individual da Unidade Curricular (UC) Estruturas de Dados Avançadas (EDA), integrada no 2º semestre do 1º ano das licenciaturas ESI/ESIPL/EIM/EEC/EDJD, visa o reforço e a aplicação dos conhecimentos adquiridos ao longo do semestre. Com este projeto de avaliação pretende-se sedimentar os conhecimentos relativos à definição e manipulação de estruturas de dados dinâmicas na linguagem de programação C. O âmbito deste projeto reside no desenvolvimento de uma solução de software na área da micromobilidade. O crescente ecossistema de novas formas de mobilidade social, nomeadamente aquelas que ocorrem entre distâncias curtas, tem promovido a necessária integração de múltiplos meios de deslocação. Esta transformação na forma como a mobilidade é realizada, fator essencial para o desenvolvimento dos espaços, cidades e outros, irá depender de ações que permitam agilizar a utilização dos meios de transporte que suportem uma mobilidade mais fácil, rápida, limpa e económica, como por exemplo os meios de mobilidade elétrica (trotinetes, bicicletas, etc.)

Apresentação do Código

Este é um trecho de código em linguagem C que apresenta um menu de opções para o usuário. As bibliotecas `stdio.h` e `string.h` são incluídas no início do código. Em seguida, é declarado um ponteiro para a estrutura `Cliente`, `Gestor` e `Meio`, bem como algumas variáveis do tipo `char`, `int` e `float`. Em seguida, são criados três objetos `Meio` com valores pré-definidos e um objeto `Cliente` e `Gestor` com nomes de usuário e senhas padrão. Em seguida, o menu é apresentado ao usuário por meio de um loop `do-while`. O usuário pode selecionar uma das três opções disponíveis digitando o número correspondente e pressionando `Enter`.

- A opção 1 é para fazer login.
- A opção 2 é para registrar um novo usuário.
- A opção 0 é para sair do programa.

```
#include <stdio.h>
#include <string.h>
#include "registos.h"

int main() {

    Cliente* clientes = NULL;
    Gestor* gestores = NULL;
    Meio* meios = NULL;
    char nome_meio[50], tipo[50], nome_utilizador[50], password[20], nome[50], email[50], endereco[100], id[50];
    int user_tipo, idade, disponivel, quantidade, nMeios, n;
    float saldo, preco, autonomia;

    Meio meioss[3] = {
        {1, 5.0, 1},
        {2, 7.5, 1},
        {3, 10.0, 0}
    };

    Cliente cliente = {"cliente1", "123", 0};
    Gestor gestor = {"gestor", "123", 1};

    int opcao;
    do {
        printf("-----BEM-VINDO AO MENU-----\n");
        printf("1 - Login\n");
        printf("2 - Registrar\n");
        printf("0 - Sair\n");
        printf("\n");
        printf("Selecione a opcao:\n");
        scanf("%d", &opcao);
```

Figura 1

Este trecho de código é um bloco switch-case que é executado quando o usuário seleciona a opção 1 no menu anterior. O código solicita ao usuário que insira seu nome de usuário e senha por meio de dois prompts printf e um scanf. Em seguida, verifica se o nome de usuário e a senha inseridos correspondem ao nome de usuário e senha do objeto cliente criado anteriormente. Se as informações de login forem corretas, o programa imprime uma mensagem de boas-vindas ao usuário. Em seguida, um loop while é iniciado e apresenta um novo menu ao usuário, que pode selecionar uma das opções disponíveis digitando o número correspondente e pressionando Enter.

- A opção 1 é para alugar um meio de transporte.
- A opção 2 é para visualizar os meios de transporte disponíveis.
- A opção 3 é para visualizar o saldo da conta do cliente.
- A opção 4 é para carregar saldo na conta do cliente.
- A opção 0 é para sair do programa.

```
switch (opcao) {
    case 1:
        printf("----- \n");
        printf("Insira o nome de utilizador:\n");
        scanf("%s", &nome_utilizador);
        printf("Insira a palavra passe:\n");
        scanf("%s", &password);

        if (strcmp(nome_utilizador, cliente.nome_utilizador) == 0 && strcmp(password, cliente.password) == 0) {
            printf("----- \n");
            printf("Bem-vindo, cliente!\n");

            while (opcao != 0) {
                printf("\nEscolha uma opcao:\n");
                printf("1 - Alugar meio\n");
                printf("2 - Ver meios disponiveis\n");
                printf("3 - Ver saldo\n");
                printf("4 - Carregar saldo\n");
                printf("0 - Sair\n");

                scanf("%d", &opcao);
            }
        }
}
```

Figura 2

Este trecho de código é um bloco switch-case que é executado quando o usuário seleciona uma das opções 1 a 4 no menu anterior. Cada opção do menu é mapeada para uma das cláusulas case no bloco switch. Cada cláusula chama uma função específica para executar a ação correspondente: A cláusula case 1 chama a função `alugarMeio()` e passa como argumentos o ponteiro para o array de meios de transporte, a quantidade de meios no array e o endereço da variável `saldo`. A cláusula case 2 chama a função `verMeiosDisponiveis()` e passa como argumentos o ponteiro para o array de meios de transporte e a quantidade de meios no array. A cláusula case 3 chama a função `verSaldo()` e passa como argumento o valor da variável `saldo`. A cláusula case 4 chama a função `carregarSaldo()` e passa como argumento o endereço da variável `saldo`. Cada função executará a ação correspondente com base nos argumentos recebidos.

```
switch (opcao) {  
    case 1:  
        alugarMeio(meios, quantidade, &saldo);  
        break;  
    case 2:  
        verMeiosDisponiveis(meios, quantidade);  
        break;  
    case 3:  
        verSaldo(saldo);  
        break;  
    case 4:  
        carregarSaldo(&saldo);  
        break;  
}
```

Figura 3

Este trecho de código é executado quando o usuário for um gestor e fornecer as credenciais corretas no login. O código define um novo menu de opções para o gestor realizar diversas operações no sistema, tais como inserção de novos dados, remoção e alteração de dados, registro de aluguel de meios de transporte, listagem de meios de transporte, etc. O bloco do-while é utilizado para manter o menu em execução enquanto o usuário não selecionar a opção "Sair" (opção 0). Dentro do bloco do-while, há outro bloco switch-case que é executado com base na opção selecionada pelo usuário no menu. Cada cláusula case no bloco switch chama uma função específica para executar a ação correspondente:

- A cláusula case 1 exibe um novo menu com opções para inserir um novo cliente, um novo gestor ou um novo meio de transporte elétrico. A opção 4 permite salvar os dados inseridos, enquanto a opção 5 sai do menu.
- A cláusula case 2 permite remover dados do sistema.
- A cláusula case 3 permite alterar dados no sistema.
- A cláusula case 4 permite registrar o aluguel de um determinado meio de transporte.
- A cláusula case 5 permite listar os meios de transporte elétricos por ordem decrescente de autonomia.
- A cláusula case 6 permite listar os meios de transporte elétricos existentes em uma determinada localização.


```

}
// código para o menu do cliente
} else if (strcmp(nome_utilizador, gestor.nome_utilizador) == 0 && strcmp(password, gestor.password) == 0) {
printf("----- \n");
printf("Bem-vindo, gestor!\n");
// código para o menu do administrador

do {
printf("\n");
printf("Escolha uma opcao:\n");
printf("1. Insercao de novos dados\n");
printf("2. Remocao de dados\n");
printf("3. Alteracao de dados\n"); //acabar
printf("4. Registo do aluguer de um determinado meio de mobilidade\n"); //acabar
printf("5. Listagem dos meios de mobilidade eletrica por ordem decrescente de autonomia\n"); //acabar
printf("6. Listagem dos meios de mobilidade eletrica existentes numa localizacao\n"); //acabar
printf("0. Sair\n");
printf("----- \n");

scanf("%d", &opcao);

switch (opcao) {
case 1:
do {
printf("1. Inserir novo cliente\n");
printf("2. Inserir novo gestor\n");
printf("3. Inserir novo meio de mobilidade eletrica\n");
printf("4. Guardar dados inseridos\n");
printf("5. Sair\n");
printf("Selecione uma opcao: ");

scanf("%d", &opcao);

```

Figura 4

O trecho de código apresenta um menu de remoção de dados com quatro opções: remover cliente, remover gestor, remover meio de mobilidade elétrica e sair do menu. Quando o usuário seleciona uma opção válida, o switch case trata a opção selecionada e chama a função correspondente para remover o cliente, gestor ou meio de mobilidade elétrica da lista correspondente. Caso a opção selecionada seja a opção "4", o menu é finalizado com a mensagem "Saindo...". Se a opção selecionada não for nenhuma das opções válidas, é exibida a mensagem "Opção inválida! Tente novamente." e o menu é apresentado novamente até que o usuário selecione uma opção válida (com exceção da opção "4" que finaliza o menu). O laço do-while na linha 23 é responsável por manter o menu ativo enquanto o usuário não selecionar a opção "4".

```
case 2:
    // Remover dados
    do {
        printf("1. Remover cliente\n"); //Exibe a opção de remover cliente.
        printf("2. Remover gestor\n"); //Exibe a opção de remover gestor.
        printf("3. Remover meio de mobilidade eletrica\n"); //Exibe a opção de remover meio de mobilidade elétrica.
        printf("4. Sair\n"); //Exibe a opção de sair do menu de remoção de dados.
        printf("Selecione uma opcao: "); //Solicita ao usuário que selecione uma opção do menu.

        scanf("%d", &opcao); //Recebe a opção selecionada pelo usuário.
```

Figura 5

Este trecho de código apresenta um menu para o usuário selecionar qual tipo de dado ele deseja alterar (cliente, gestor ou meio de mobilidade elétrica) e, em seguida, chama a função correspondente a cada tipo de dado selecionado. O menu é apresentado em um laço do-while que só é finalizado quando o usuário seleciona a opção "4" para sair do menu. Se o usuário selecionar uma opção inválida, uma mensagem é exibida e o menu é apresentado novamente. As funções "alterarDadosCliente", "alterarDadosGestor" e "alterarDadosMeio" recebem como parâmetro a lista correspondente ao tipo de dado que o usuário deseja alterar e solicitam ao usuário que informe o nome do dado a ser alterado. Em seguida, as funções apresentam as opções de campos que podem ser alterados e solicitam ao usuário que informe qual campo ele deseja alterar. Depois de selecionado o campo a ser alterado, a função solicita ao usuário que informe o novo valor a ser atribuído ao campo. Após a alteração, a lista de dados é atualizada com as novas informações.

```
case 3:
// Alterar dados
do {
    printf("1. Alterar dados do cliente\n");
    printf("2. Alterar dados do gestor\n");
    printf("3. Alterar dados do meio de mobilidade eletrica\n");
    printf("4. Sair\n");
    printf("Selecione uma opcao: ");

    scanf("%d", &opcao);

    switch (opcao) {
        case 1:
            alterarDadosCliente(clientes);
            break;
        case 2:
            alterarDadosGestor(gestores);
            break;
        case 3:
            alterarDadosMeio(meios);
            break;
        case 4:
            printf("Saíndo...\n");
            break;
        default:
            printf("Opção inválida! Tente novamente.\n");
            break;
    }
} while (opcao != 4);
```

Figura 6

Este trecho de código representa o menu de opções da aplicação. Quando o usuário escolhe uma opção, a aplicação executa uma determinada função correspondente à opção escolhida. O menu apresenta as seguintes opções:

- Opção 1: Cadastrar novo cliente;
- Opção 2: Cadastrar novo gestor;
- Opção 3: Alterar dados de um cliente, gestor ou meio de mobilidade elétrica;
- Opção 4: Registrar aluguel de um meio de mobilidade elétrica;
- Opção 5: Ordenar a lista de meios de mobilidade elétrica por autonomia;
- Opção 6: Listar os meios de mobilidade elétrica em uma determinada localização;
- Opção 0: Sair da aplicação.

Dependendo da opção selecionada pelo usuário, a aplicação executa uma determinada função. Por exemplo, se o usuário escolher a opção 1, a função "cadastrarCliente" será chamada para cadastrar um novo cliente. Se o usuário escolher a opção 4, a função "alugarMeio" será chamada para registrar o aluguel de um meio de mobilidade elétrica. Se o usuário escolher a opção 0, a aplicação finaliza. Se o usuário escolher uma opção inválida, a aplicação exibe a mensagem "Opção inválida." e o menu é apresentado novamente.

```
case 4:
    // Registrar aluguer
    alugarMeio(meios, quantidade, &saldo);
    break;
case 5:
    ordenarMeiosPorAutonomia(meios, n);
    break;
case 6:
    // Listar meios de mobilidade elétrica em uma localização
    printf("Opcao 6 selecionada.\n");
    break;
case 0:
    // Sair
    printf("Saindo...\n");
    break;
default:
    printf("Opcao inválida.\n");
```

Figura 7

Este trecho de código representa a opção 2 do menu, que permite registrar um novo cliente no sistema. O usuário é solicitado a inserir o nome de usuário, senha, nome, email e endereço do cliente. Após a inserção dos dados, a função `inserirCliente` é chamada para inserir o novo cliente na lista de clientes. Caso a operação seja realizada com sucesso, é exibida a mensagem "Cliente registado com sucesso!". Se a opção selecionada não for válida, é exibida a mensagem "Opcao invalida.". Se a opção selecionada for 0, é exibida a mensagem "A sair do programa." e o programa é encerrado.

```
case 2:
    printf("Insira o nome de utilizador: \n");
    scanf("%s",&nome_utilizador);
    printf("Insira a password: \n");
    scanf("%s",&password);
    printf("Insira o nome: \n");
    scanf("%s",&nome);
    printf("Insira o email: \n");
    scanf("%s",&email);
    printf("Insira o endereco: \n");
    scanf("%s",&endereco);
    printf("Cliente registado com sucesso! \n");

    clientes = inserircliente(clientes,nome_utilizador,password,nome,email,endereco);

    break;

case 0:
    printf("A sair do programa.\n");
    break;

default:
    printf("Opcao invalida.\n");
```

Figura 8