



Programação

Metro Mondego

Docente: Francisco Pereira

Rodrigo Dinis Lourenço nº 2021155662 – LEI

Coimbra, 10 de Junho de 2023

Índice

1. Introdução.....	2
2. Código Fonte	3
3. Estruturas de Dados.....	4
3.1 Estrutura Paragem	4
3.2 Estrutura Linha.....	5
3.3 Estrutura Sistema	6
3.4 Estrutura NoLinha.....	6
4. Decisões.....	6
4.1 Estruturas Dinâmicas	6
4.1.1 Paragem	7
4.1.2 Linha	7
4.1.3 Lista Ligada de Linhas	7
4.2 Percursos	8
5. Outras Funcionalidades	9
5.1 Registrar Paragem	9
5.2 Eliminar Paragem.....	9
5.3 Visualizar Paragens.....	9
5.4 Adicionar Linhas.....	10
5.5 Atualizar Linha	10
5.6 Visualizar Linhas	11

1. Introdução

Este trabalho surge no âmbito da Unidade Curricular de Programação, no 2º Semestre do ano letivo de 2022/2023. Feito em linguagem C standard, respeitando a norma C99, tem como objetivos principais:

1. Simular, a um nível básico, um sistema de mobilidade do Metro.

Para alcançar estes objetivos, foram usados conceitos da linguagem C, como estruturas dinâmicas, estruturas ligadas, leitura de ficheiros binários e leitura e escrita de ficheiros de texto, entre outros.

2. Código Fonte

O código fonte do programa encontra-se dividido por quatro ficheiros de código (extensão .c) e três ficheiros do tipo *header* (extensão .h), separados em menus e preparação, utilidades gerais, utilidades de estruturas e utilidades de ficheiros:

- main.c
 - Contém a inicialização das estruturas Sistema e Paragem.
 - E chama a função menu para começar a receber instruções do utilizador.
- Funcs.c
 - Contêm todas as funções.
- Header.h
 - Contêm as estruturas.

3. Estruturas de Dados

Neste trabalho, foram utilizadas, no total, quatro estruturas diferentes:

- Paragem
- Linha
- Sistema
- NoLinha

Estas estruturas foram todas declaradas no ficheiro header.h. As suas declarações e usos vão ser apresentados nas próximas páginas.

3.1 Estrutura Paragem

```
typedef struct{  
    char **nomep;  
    char **codigop;  
    int num_paragensp;  
}Paragem;
```

A estrutura Paragem é usada para guardar todas as Paragens adicionadas ao Sistema. Guardando o nome e o código num array de ponteiros incrementando sempre o num_paragensp.

3.2 Estrutura Linha

```
typedef struct
{
    char *nome;
    char **paragens; //
    char **codigo; //Ar
    int num_paragens;
}Linha;
```

A estrutura Linha é, como o nome indica, guarda as informações da Linha, nome, as suas paragens, os codigos das mesmas e quantidade total de paragens da mesma.

3.3 Estrutura Sistema

```
typedef struct{  
    Paragem *paragens; //  
    Linha *linhas; //Arr  
    int num_linhas;  
    int Snumparagens;  
}Sistema;
```

A estrutura Sistema sendo a estrutura principal do projeto “guarda”/aponta todas as paragens e Linhas, guardando também o numero de linhas total e de paragens.

3.4 Estrutura NoLinha

```
typedef struct NoLinha {  
    Linha *linha;  
    struct NoLinha *prox;  
} NoLinha;
```

A estrutura NoLinha aponta para uma estrutura Linha.

Inclui também um ponteiro para outra Linha, de modo a possibilitar a formação de uma lista ligada de Linhas.

4. Decisões

4.1 Estruturas Dinâmicas

Foram, ao longo deste trabalho, implementadas diferentes estruturas dinâmicas e alteradas várias vezes. Decidi, por fim, utilizar as estruturas de dados já referidas, formando as seguintes estruturas dinâmicas:

4.1.1 Paragem

Como referido anteriormente a estrutura Paragem contém um ponteiro para um array de ponteiros de strings “nomep” e “codigop” que foram manipulados ao longo do projeto mas principalmente durante a adição de paragens e linhas pelo ficheiro.txt de inicialização.

```
if (paragens->nomep == NULL // paragens->num_paragensp == 0) {  
  
    paragens->nomep = malloc(sizeof(char *));  
    paragens->codigop = malloc(sizeof(char *));  
    if (paragens->nomep == NULL // paragens->codigop == NULL) {  
        // erro ao alocar memória  
        printf("Erro a alocar memoria do tamanho do ponteiro no adicionaparagemutilizador\n");  
        exit(1);  
    }  
}  
else {  
  
    paragens->nomep = realloc(paragens->nomep, (paragens->num_paragensp + 1) * sizeof(char *));  
    paragens->codigop = realloc(paragens->codigop, (paragens->num_paragensp + 1) * sizeof(char *));  
  
    if (paragens->nomep == NULL // paragens->codigop == NULL) {  
        // erro ao alocar memória  
        printf("Erro a realocar memoria do tamanho do ponteiro no adicionaparagemutilizador\n");  
        exit(1);  
    }  
}
```

4.1.2 Linha

Já a estrutura Linha é constantemente manipulada pois qualquer alteração a uma linha ou paragens é refletida na mesma tendo em conta os seus dois arrays de ponteiros “paragens” e “código”.

4.1.3 Lista Ligada de Linhas

Serve para guardar a informação das Linhas, cada “Nó” representando uma Linha.

4.2 Percursos

```
---Menu---
1-->Registar Paragem
2-->Eliminar Paragem
3-->Visualizar Paragens
4-->Adicionar Linhas
5-->Atualizar Linha
6-->Visualizar Linhas
7-->Percurso teste
8-->Sair do Programa
█
```

Após a escolha da opção 7 o utilizador irá inserir a paragem de partida e a paragem de chegada e receberá como resposta todos os percursos que cumprem os requisitos quer seja um percurso que passe nas duas paragens desejadas numa linha única ou em duas linhas (percurso com transbordo). Sendo as linhas percorráveis em dois sentidos o output poderá ser o seguinte, ex:

```
Introduza as paragens que deseja incluir no seu percurso:
Paragem de Partida:
viaduto
Paragem Destino:
aviario

-----
Percurso linha unica [Sentido Ida]
Linha Alta : Sentido normal
Viaduto -> Paragem -> Flores -> Loja do Cidadao -> Zona Alta -> Aviario
-----
Percurso linha unica [Sentido Volta]
Nao foi encontrado percurso entre viaduto e aviario.
-----
Percurso duas linhas [Sentido Ida]
Linha Secundaria: Sentido Normal
Viaduto -> Ferrovia -> Casa do Mel -> Flores
Linha Alta: Flores -> Loja do Cidadao -> Zona Alta -> Aviario

Linha Secundaria: Sentido Normal
Viaduto -> Ferrovia -> Casa do Mel -> Flores
Linha da Baixa: Flores -> Aviario

Linha Alta: Sentido Normal
Viaduto -> Paragem -> Flores
Linha da Baixa: Flores -> Aviario
-----
Percurso duas linhas [Sentido Volta]
Linha Alta: Sentido Inverso
Viaduto -> Paragem Fim
Linha da Baixa: Paragem Fim -> Casa do Mel -> Casa do Sal -> Loja do Cidadao -> Aviario
-----
```


5. Outras Funcionalidades

5.1 Registrar Paragem

O utilizador introduz o nome de uma nova paragem e após verificação nas paragens já adicionadas ao sistema a mesma é adicionada como uma paragem do utilizador logo com Cod iniciado por U (todas as outras paragens terão um código iniciado pelas restantes letras do alfabeto).

5.2 Eliminar Paragem

Caso já existam paragens do Utilizador no sistema o utilizador pode elimina-las caso não pertençam a nenhuma linha.

5.3 Visualizar Paragens

Mostra todas as paragens presentes no momento no Sistema.

```
Linha -> [-Linha Odivelas-]
:Odivelas: Y6921
:Senhor Roubado: O7192
:Oriente: W6877
:Campo Grande: A5414
:Intendente: T7289
:Anjos: W8966
:Praca da Espanha: H2375

Linha -> [-Linha Campolide-]
:Rossio: S3948
:Martin Moniz: T9744
:Baixa-Chiado: J6135
:Terreiro do Paco: R2166
:Santa Apolonia: J8874
:Avenida: Q6462
:Picoas: P6985
:Sete Rios: Y5331
:Alto dos Moinhos: G2884

Paragens do utilizador:
47->Paragem do Utilizador U7184
Bem vindo!
```

5.4 Adicionar Linhas

Nesta opção o utilizador pode adicionar novas Linhas através de um ficheiro txt há sua escolha ou introduz um novo nome para a sua Linha e depois poderá “montar” a sua linha com paragens do Sistema.

5.5 Atualizar Linha

Nesta opção o utilizador escolhe uma linha e poderá adicionar novas paragens ou remover alguma das suas paragens:

```
Paragens do utilizador:
47->Paragem Teste U9319
48->Paragem 2 U9329
49->Paragem Utilizador U9342
50->Tudo Paragens Utilizador U9368
Introduza o numero da paragem a adicionar e a posicao escolhida
[0]:Parque Verde:[1]:Paragem Fim:[2]:Viaduto:[3]:Paragem:[4]:Flores:[5]:Loja do Cidadao:[6]:Zona Alta:[7]:Aviario:[8]
[Exemplo:20 3
Escolheu a paragem 20 que ira ser colocada na posicao 3
Quanto terminar introduza :Sair:]
Linha -> [-Linha da Baixa-]
:Parque da Cidade: P011
:Portagem: P09
:Flores: A544
:Aviario: I221
:Loja do Cidadao: P013
:Casa do Sal: F554
:Casa do Mel: F514
:Paragem Fim: H123
:Coimbra-B: H123
Introduza o cod da paragem a remover
Quanto terminar introduza :Sair:
```

5.6 Visualizar Linhas

Por último o Utilizador pode imprimir todas as linhas que estão no Sistema e imprimir as Linhas que passam numa paragem há sua escolha.

```
Opcao 6 seleccionada
0 -> [-Linha Inicial-]
1 -> [-Linha Secundaria-]
2 -> [-Linha Alta-]
3 -> [-Linha da Baixa-]
4 -> [-Linha Odivelas-]
5 -> [-Linha Campolide-]
6 -> [-Linha dos Barcos-]
Pretende visualizar linhas que passem em alguma paragem especifica?
1-->Sim
2-->Nao
1
```

```
46->Alto dos Moinhos G2884
47->Paragem Teste U9319
48->Paragem 2 U9329
49->Paragem Utilizador U9342
50->Tudo Paragens Utilizador U9368
Introduza o numero da paragem que pretende seleccionar
25
2 -> [-Linha Alta-]
3 -> [-Linha da Baixa-]
```