



Programação

Metro Mondego

Docente: Francisco Pereira

Miguel Umbelino da Mota Roma nº2021138955 – LEI

Coimbra, 11 de junho de 2023

Índice

1.	Introdução.....	3
2.	Organização do Código	4
3.	Estruturas de Dados.....	5
3.1	Estrutura das Paragens.....	5
3.2	Estrutura das Linhas	5
4.	Sequência de Funcionamento	6
5.	Utilização do Programa	7
6.	Conclusão	10

1. Introdução

Relatório do trabalho realizado no âmbito da Unidade Curricular de Programação, no 2º semestre do ano letivo 2022/23 utilizando a linguagem C Standard com recurso ao Clion como ambiente de desenvolvimento (IDE).

Com este documento pretende-se destacar as funcionalidades implementadas e as soluções aplicadas para resolver cada objetivo proposto no enunciado do trabalho, detalhando algumas das implementações e explicando o processo de funcionamento do mesmo.

2. Organização do Código

O programa é composto por sete ficheiros source do tipo .c e seis ficheiros *header* do tipo .h, separados em menus e utilidades gerais, utilidades de estruturas e utilidades de ficheiros:

- **main.c** -> menu principal do sistema de Metro que trata da chamada das funções associadas a cada submenu (linhas, paragens, ...) e leitura dos ficheiros binários
- **menus.c/.h** -> nestes ficheiros são tratadas a parte “gráfica” do programa, onde estão as funções relativas aos menus a serem mostrados ao utilizador e que devolvem a opção por este introduzida.
- **paragem.c/.h** -> contêm as funções de manipulação e interação das paragens no sistema do Metro Mondego, que são guardadas num array dinâmico.
- **linha.c/.h** -> onde se encontram as funções relativas à manipulação e tratamento das linhas e das suas respetivas paragens permitindo várias operações, como adição, remoção, listar, entre outras.
- **percurso.c/.h** -> possui as funções de cálculo dos percursos de linha única e apresenta o trajeto, se disponível, entre a partida e a chegada (apresentando todas as paragens entre estas).
- **ficheiros.c/.h** -> engloba todas as funções de tratamento de ficheiros, quer dos binários quer dos de texto. Permite recuperar e guardar informação das linhas e das paragens no início e no termino da execução do programa, respetivamente. Assegura também que caso o utilizador pretenda carregar um ficheiro de texto este respeita as normas definidas.
- **utils.c/.h** -> é responsável por gerar um código alfanumérico de 4 caracteres a ser utilizado pelas paragens e é onde são definidas as bibliotecas a usar para evitar repetição constante nos outros ficheiros.

3. Estruturas de Dados

De forma a organizar e otimizar o código foram utilizadas duas estruturas diferentes de acordo com a necessidade da informação a armazenar do sistema do metro:

- **Paragens** -> struct paragem (ponteiro pparagem)
- **Linhas** -> struct linha (ponteiro plinha)

3.1 Estrutura das Paragens

```
typedef struct paragem paragem, *pparagem;  
struct paragem{  
    char nome[100];  
    char codigo[5];  
    int numLinhas;  
  
    pparagem ant;  
    pparagem prox;  
};
```

Estrutura das paragens é utilizada para guardar, num array dinâmico, as paragens registadas pelo utilizador no sistema. Para tal a informação de cada paragem é composta pelo nome, código e as linhas a que pertence. Os ponteiros ant e prox foram adicionadas para serem usados no acesso às paragens nas funções das linhas.

3.2 Estrutura das Linhas

```
typedef struct linha linha, *plinha;  
struct linha{  
    char nome[100];  
    int nParag;  
  
    pparagem lista;  
    plinha prox;  
};
```

Estrutura das Linhas é usada para guardar as linhas registadas no sistema (lista ligada) e suas respetivas paragens (lista duplamente ligada) para facilitar o acesso tanto a cada paragem como a cada linha. A informação armazenada é o número de paragens de cada linha, o seu nome e a informação relativa a cada paragem (vai buscar do array dinâmico) e guarda essa informação na lista de paragens “lista”.

4. Sequência de funcionamento

A sequência de funcionamento do sistema de metro foi implementada de forma a tentar simplificar e criar uma forma mais interativa do utilizador navegar pelas várias opções disponíveis. Para tal foram criados múltiplos menus organizados por categoria do que se pretende realizar no sistema do Metro Mondego.

O programa quando é inicializado tenta ler os ficheiros binários tanto das linhas como das paragens e, caso estes existam, são adicionadas as paragens ao array dinâmico (sistema) e as linhas e suas respetivas paragens à lista ligada (linhas).

Na eventualidade de ser a primeira vez que o programa é corrido não vai ter nenhuma paragem nem nenhuma linha registada. Para tal, de forma a começar a explorar todas as funcionalidades é necessário primeiro entrar na aba das paragens e adicionar, remover e listar todas as paragens ao critério do utilizador, que vão ficar armazenadas no array dinâmico.

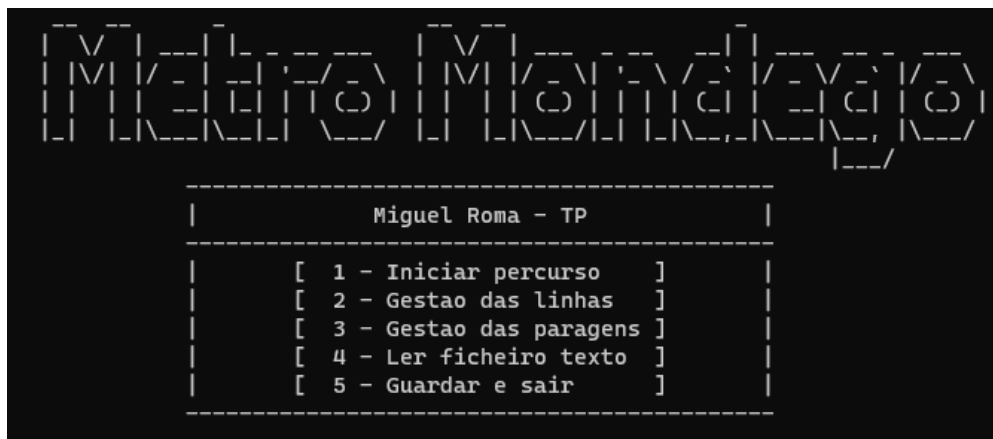
Após ter as paragens registadas, o utilizador pode avançar para o menu das linhas, onde poderá criar e gerir as diferentes linhas do metro, atribuindo-lhes as paragens anteriores. Pode também optar por ler uma linha e suas paragens diretamente de um ficheiro de texto para o programa, desde que estas respeitem as normas definidas.

Por fim com as paragens e linhas devidamente configuradas é possível utilizar a funcionalidade de pesquisa de percursos. Nesta opção após introduzir a paragem de partida e destino o sistema irá fornecer o percurso correspondente, considerando as linhas disponíveis sendo que apenas foi implementado o percurso de linha única.

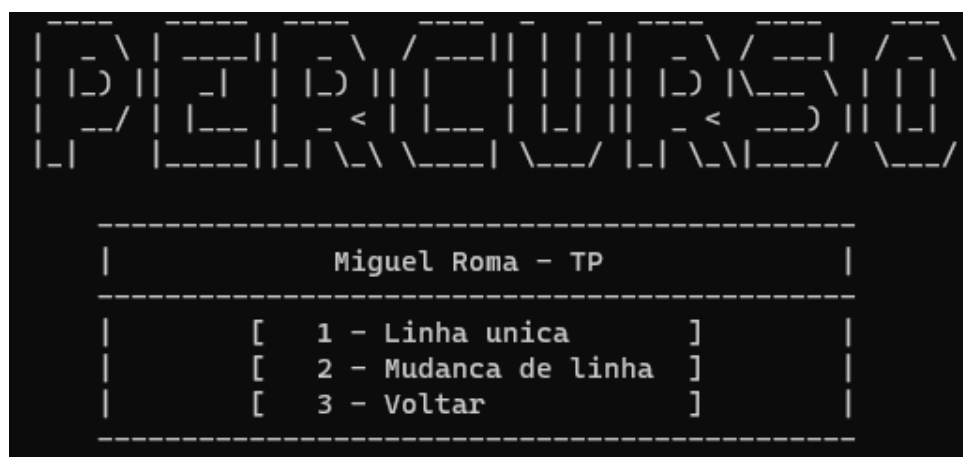
Quando a execução é terminada as linhas e as paragens são guardadas em ficheiros binários para posteriormente, numa nova execução, poder fazer o descrito acima e assim continuar o ciclo do sistema do Metro Mondego.

5. Utilização do Programa

Ao executar o programa, é apresentado um menu inicial com várias opções agrupadas por categoria de acordo com o que se pretende fazer/alterar:



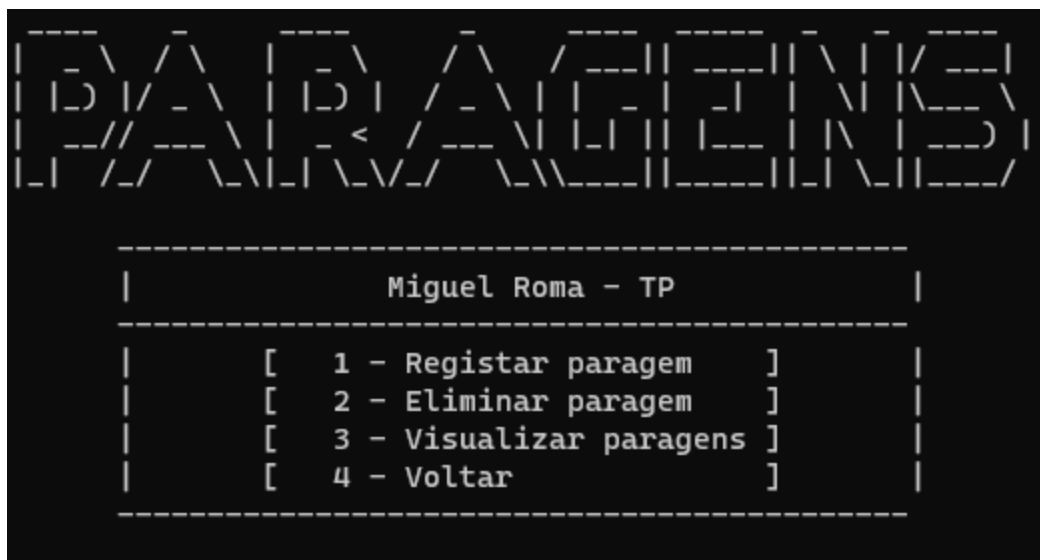
Opção 1 [Iniciar percurso] -> Abre um submenu que permite ao utilizador escolher se pretende procurar o trajeto numa única linha ou se pretende permitir que haja mudança de linha (embora esta funcionalidade não tenha sido implementada está apresentada no menu apenas para mostrar o que seria expectável) no percurso por ele definido (partida e chegada):



Opção 2 [Gestão das linhas] -> É aberto um outro menu que apresenta várias opções relativa à parte de interação e modificação das linhas:



Opção 3 [Gestão das paragens] -> Leva o utilizador a um submenu que permite a interação e modificação das paragens no sistema:



Opção 4 [Ler ficheiro texto] -> A linha e as respetivas paragens guardadas num ficheiro de texto, que segue certas normas de formatação para ser válida a sua leitura, são lidas e introduzidas no sistema do Metro Mondego, como apresentado no exemplo:

```
Linha da Baixa
Parque da Cidade # P011
Portagem # P123
Loja do Cidadao # Q998
Casa do Sal # F554
Coimbra-B # H123
```

Opção 5 [Guardar e Sair] -> Todas as alterações tanto das paragens como das linhas são guardadas em ficheiros binários (separados), que são restauradas aquando da abertura novamente do programa, e termina a execução.

- Ficheiro “**paragens.dat**” -> guarda a informação relativa às paragens, desde nome, código e número de paragens que permite a sua posterior recuperação na abertura do programa.
- Ficheiro “**linhas.dat**” -> guarda a informação relativamente a cada linha, como o nome e o número de paragens, e também a informação relativa a cada paragem que se encontra inserida nessa linha para ser possível reler a informação numa nova execução do programa.

6. Conclusão

Acerca do projeto desenvolvido, os objetivos principais propostos foram atingidos, permitindo ao utilizador a utilização do sistema, embora haja sempre espaço para melhoramento do código desenvolvido quer na eficiência geral como na redução da utilização de ponteiros como auxiliares. É de sublinhar apenas que o único ponto não conseguido no trabalho foi a procura de trajeto com mudança de linha.

Após ter terminado é possível dizer que este trabalho me permitiu não só aplicar e consolidar a matéria que tem vindo a ser lecionada ao longo do semestre, como também ganhar uma maior capacidade de organização de código que pode vir a ser útil em projetos futuros.