



## Programação

### Gestão Metro Mondego

Docente: Francisco Pereira

# Índice

|      |                               |    |
|------|-------------------------------|----|
| 1.   | Introdução .....              | 3  |
| 2.   | Código Fonte.....             | 4  |
| 3.   | Estruturas de Dados .....     | 6  |
| 3.1. | Estrutura Stop .....          | 6  |
| 3.2. | Estrutura Line .....          | 6  |
| 3.3. | Estrutura LineStop .....      | 7  |
| 3.4. | Estrutura LineList .....      | 7  |
| 4.   | Implementação.....            | 8  |
| 5.   | . Anexos .....                | 10 |
| 5.1. | Anexo 1 –Menus .....          | 10 |
| 5.2. | Anexo 2-Funcionalidades ..... | 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, desenvolvido no Visual Studio Code.

O objetivo principal do projeto é permitir aos usuários visualizar informações sobre as paragens, linhas e realizar cálculos de percursos .

Também tem como objetivo ser de fácil utilização, através da implementação de uma interface simples, e que esclareça o utilizador em tudo o que pode fazer em cada menu.

Para alcançar estes objetivos, foram usados conceitos da linguagem C, como arrays de estruturas dinâmicas, estruturas ligadas, leitura e escrita de ficheiros binários e leitura 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 tres ficheiros do tipo header (extensão .h), separados em menu e ficheiros, operacoes de paragens e linhas, utilidades e estruturas:

- main.c
  - Contém um menu a base de switch...case onde são chamadas a maioria das funções ou os respetivos sub-menus para aceder as diferentes funcionalidades;
  - Neste ficheiro são inicializados o array dinâmico das paragens e a lista ligada das linhas;
  - Contém as funções relativas aos ficheiros binarios que permitem guardar os dados das paragens e linhas do sistema;
- stops.c/.h
  - Contêm a estrutura das paragens, que permite guardar em cada paragem as suas informacoes;
  - Contêm as funções relacionadas às paragens que permitem aos usuários listar todas as paragens existentes, adicionar novas paragens e remover paragens existentes.
  - Contêm algumas funções de apoio como por exemplo a função getName(), que faz um prompt para pedir o nome da paragem, ou a função generateCode() que gera o codigo alfanumerico para as paragens;

- `lines.c/.h`
  - Contêm as estruturas para o funcionamento da lista ligada das linhas e das paragens de cada linha;
  - Contêm as funções relacionadas às linhas permitem aos usuários ver uma lista de todas as linhas disponíveis, adicionar uma nova linha ao sistema, atualizar informações de uma linha existente e remover linhas indesejadas;
  - Além disso, há a opção de adicionar novas linhas por meio de um ficheiro de texto externo;
- `percursos.c/.h`
  - Contêm as funções relativas aos percursos que permitem aos usuários calcular percursos diretos ou com escala entre duas paragens. Os usuários podem inserir o ponto de partida e o destino desejado para calcular o percurso;

## 3. Estruturas de Dados

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

- Stop - renomeada para Stop;
- Line - renomeada para Line;
- LineStop - renomeada para LineStop;
- LineList - renomeada para LineList;

Estas estruturas foram declaradas no ficheiro stops.h e no ficheiro lines.h.

As suas declarações e usos vão ser apresentados nas próximas páginas.

### 3.1 Estrutura Stop

```
typedef struct Stop Stop; // Estrutura para as paragens
struct Stop
{
    int valid;
    char name[MAX_NAME_LENGTH];
    char codigo[MAX_CODE_LENGTH + 1];
};
```

A estrutura Stop contém um inteiro que determina se a paragem é válida, um array de caracteres que guarda o nome da paragem e um array de caracteres que guarda o código identificador da paragem.

### 3.2 Estrutura Line

```
typedef struct Line Line; // Estrutura para as linhas
struct Line
{
    char name[MAX_NAME_LENGTH];
    int nStops;
    LineStop *nextStop;
};
```

A estrutura Line contém um array de caracteres que guarda o nome da linha, um inteiro que guarda o número de paragens pertencentes à linha e um ponteiro para uma estrutura do tipo LineStop referente à primeira paragem da linha.

### 3.3 Estrutura LineStop

```
typedef struct LineStop LineStop; // Estrutura para as paragens da linha
struct LineStop
{
    Stop stop;
    LineStop *nextStop;
    LineStop *prevStop;
};
```

A estrutura LineStop é usada para guardar e percorrer as paragens que pertencem a uma respetiva linha. Contem uma estrutura do tipo Stop que guarda as informações da paragem e dois ponteiros para estruturas do tipo LineStop que apontam respetivamente para a paragem seguinte e para a paragem anterior.

### 3.4 Estrutura LineList

```
typedef struct LineList LineList; // Lista das Linhas
struct LineList
{
    Line line;
    LineList *nextLine;
};
```

A estrutura LineList é usada para guardar e percorrer todas as linhas existentes no sistema. Contem uma estrutura do tipo Line que guarda as informações da linha e um ponteiros para uma estrutura do tipo LineList que aponta para a linha seguinte. Em retrospectiva apercebi-me que esta estrutura não era necessária, poderia colocar o ponteiro diretamente na estrutura do tipo Line.

## 4. Implementação

A implementação do programa segue a seguinte estrutura:

Inicialização do array dinâmico das paragens usando o malloc e da lista ligada das linhas.

Verificação do arquivo "metromondego.dat":

Ao iniciar o programa, é verificado se o arquivo "metromondego.dat" existe.

Se o arquivo existir, o programa adiciona os dados lá guardados e adiciona as paragens e linhas às estruturas criadas previamente.

Menu de opções:

Após a verificação do arquivo, o programa apresenta um menu com as seguintes opções:

- 1 - Paragens
- 2 - Linhas
- 3 - Percursos
- 4 - Sair

A opção "Paragens" leva-nos para um sub-menu onde é possível, ver todas as paragens, adicionar novas paragens e remover paragens existentes.

A opção "Linhas" leva-nos para um sub-menu onde é possível, ver uma lista das linha escolhendo a que deseja visualizar, adicionar novas linhas, atualizar uma linha existente, ou seja, adicionar ou remover paragens da respetiva linha, remover linhas existentes e adicionar novas linhas por meio de um ficheiro de texto .

A opção "Percursos" vai solicitar ao utilizador a sua paragem de partida e qual o seu destino, mostrando depois a opção de procurar um percurso direto ou com uma escala.



Guardar dados:

A opção "Sair" permite que o utilizador guarde num ficheiro binário todos os dados que foram adicionados ao sistema.

As informações necessárias são armazenadas no ficheiro, permitindo manter essa estrutura de dados para uma próxima utilização.

No ficheiro são guardados o número de paragens seguido pelo array dinâmico das paragens. Depois guarda o número de linhas e percorre as linhas todas guardando, respetivamente, o nome da linha, o número de paragens pertencentes á linha e percorre as paragens guardando a posição que a paragem tem no array dinâmico das paragens

De seguida termina o programa.

Essa é uma visão geral da implementação do programa, abordando os principais pontos e funcionalidades.

## 5. Anexos

### 5.1 Anexo 1 – Menus

Os Menus têm o seguinte aspeto:

```
--Operacoes Paragens--  
1 - Listar todas as paragens  
2 - Adicionar nova paragem  
3 - Remover paragem existente  
4 - Voltar ao menu principal
```

*Figura 1: Menu Operacoes Paragens*

```
--- Lista das Linhas ---  
  
1 - Linha Norte  
2 - Linha Litoral  
3 - Linha Sul
```

*Figura 2: Vizualizar Linhas*

```
--Operacoes Linhas--  
1 - Ver Linhas  
2 - Adicionar nova Linha  
3 - Atualizar Linha existente  
4 - Apagar Linha existente  
5 - Adicionar nova Linha por Ficheiro  
6 - Voltar ao menu principal
```

*Figura 3: Menu Operacoes Linhas*

```
--- MENU ---  
1 - Paragens  
2 - Linhas  
3 - Percursos  
4 - Exit
```

*Figura 4: Menu Inicial*

## 5.2 Anexo 2 – Funcionalidades

As imagens seguintes apresentam o aspeto de algumas funcionalidades principais:

```
Linha da Baixa - 5 Paragens  
[ Parque da Cidade <> Portagem <> Loja do Cidadao <> Casa do Sal <> Coimbra-B ]
```

*Figura 5: Vizualizar Linha da Baixa*

```
----- Linha Norte -> Linha Litoral -----  
Linha Norte  
[ Braga -> Porto -> Aveiro ]  
  
Linha Litoral  
[ Aveiro -> Figueira da Foz -> Coimbra -> Lisboa ]
```

*Figura 6: Calcular Percurso de Braga para Lisboa*

```
--- Adicionar Paragem ---  
  
Nome da Paragem: Coimbra Shopping  
  
Nome: Coimbra Shopping  
Codigo: C216
```

*Figura 7: Adicionar Paragem*