

# Relatório do Trabalho Prático

FASE 1: LISTAS LIGADAS

EDGAR ALEXANDRE VASQUEZ CASAL – A31026

UC – ESTRUTURAS DE DADOS AVANÇADAS

## **1. Introdução**

Este relatório descreve a implementação da Fase 1 do projeto, que consiste num sistema de gestão de antenas e cálculo de efeitos nefastos, desenvolvido em C. O objetivo foi aplicar estruturas de dados dinâmicas (listas ligadas) para resolver um problema clássico de alinhamento de antenas.

## 2. Estruturas de Dados Utilizadas

### 2.1. Antena

```
typedef struct Antena {  
    char freq;  
    int x, y;  
    struct Antena* prox;  
}Antena;
```

**Objetivo:** Armazenar as antenas e suas posições numa lista ligada simples.

### 2.2. Efeito Nefasto

```
typedef struct Nefasto {  
    int x, y;  
    struct Nefasto* prox;  
}Nefasto;
```

**Objetivo:** Listar locais com interferência causada por alinhamento de antenas.

## 3. Funcionalidades Implementadas

### 3.1. Carregamento de Antenas

#### Função:

Antena\* carregarAntenas(char\* nomeFicheiro)

#### Descrição:

- Lê um ficheiro de texto com o mapa de antenas (ex: ..A..\n.B...).
- Cria uma lista ligada onde cada nó representa uma antena

### 3.2. Inserção/Remoção de Antenas

#### Funções:

Antena\* inserirAntena(Antena\* h, char freq, int x, int y);  
Antena\* removerAntena(Antena\* h, int x, int y);

#### Descrição:

- Inserção no início da lista
- Remoção por busca de coordenadas

### 3.3. Cálculo de Efeitos Nefastos

#### Função:

Nefasto\* efeitoNefasto(Antena\* h)

#### Descrição:

1) Para cada par de antenas com mesma frequência, calcula pontos alinhados onde:

- Antena A está ao dobro da distância de Antena B (e vice-versa).
- Fórmula: Ponto =  $(2*x1 - x2, 2*y1 - y2)$ .

2) Armazena coordenadas válidas numa lista ligada.

### 3.4. Visualização

#### Funções:

```
void imprimirAntenas(Antena* h);  
void imprimirNefasto(Nefasto* h);  
void imprimirAntenasNefasto(char* nomeFicheiro, Nefasto* h);
```

#### Descrição:

- Lista das antenas em forma tabular
- Lista dos efeitos nefastos em forma tabular
- Lista do mapa original com o mapeamento dos efeitos nefastos (#)

## 4. Estrutura do Código

O código está em /src e contém os seguintes ficheiros:

### 4.1. main.c

#### Função principal:

- Testa as funções básicas (inserção, remoção, cálculo de efeitos nefastos).

Exemplo de uso:

```
int main () {
    Antena* lista = NULL;
    Nefasto* listafeitoNefasto = NULL;

    //Teste de criar/inserir antenas numa lista ligada, remoção de uma antena e imprimir antenas de uma lista ligada em forma tabular
    lista = inserirAntena(lista, 'A', 0, 0);
    lista = inserirAntena(lista, 'A', 0, 1);
    lista = inserirAntena(lista, 'A', 1, 1);
    lista = inserirAntena(lista, 'B', 1, 2);
    imprimirAntenas(lista);
    printf("\n");

    removerAntena(lista, 1, 1);
    imprimirAntenas(lista);
    printf("\n");
}
```

### 4.2. funcoes.h

#### Contém:

- Declarações das estruturas Antena e Nefasto.
- Protótipos de todas as funções (ex: inserirAntena, gravarFicheiroBinario).

### 4.3. funcoes.c

#### Implementação das funções:

- **carregarAntenas()**: Lê um ficheiro de texto com o mapa de antenas e cria uma lista ligada. Ignora caracteres não alfabéticos.

- **Antena\* inserirAntena(Antena\* h, char freq, int x, int y):** Insere uma nova antena no início da lista. Devolve o novo início.
- **Antena\* removerAntena(Antena\* h, int x, int y):** Remove uma antena com as coordenadas (x,y). Devolve o início da lista (atualizado se necessário).
- **Nefasto\* efeitoNefasto(Antena\* h):** Calcula pontos com interferência ao usar a fórmula  $(2 \cdot x_1 - x_2, 2 \cdot y_1 - y_2)$  para antenas com mesma frequência.
- **Nefasto\* inserirNefasto(Nefasto\* h, int x, int y):** Adiciona um novo efeito à lista, evitando duplicados. Devolve o novo início.
- **void imprimirAntenas(Antena\* h):** Imprime a lista de antenas em formato tabular.
- **void imprimirNefasto(Nefasto\* h):** Imprime os efeitos nefastos em formato tabular.
- **bool gravarFicheiroBinario(char\* nomeFicheiro, Antena\* h):** Grava as antenas num ficheiro binário (frequência + coordenadas).
- **void imprimirAntenasNefasto(char\* nomeFicheiro, Nefasto\* h):** Exibe o mapa original com # nas posições de efeito nefasto.

## 5. Testes Realizados

### Caso 1: Testar a inserção, remoção e imprimir em forma tabular

main.c:

```
int main () {
    Antena* lista = NULL;
    Nefasto* listaEfeitoNefasto = NULL;

    //Teste de criar/inserir antenas numa lista ligada, remoção de uma antena e imprimir antenas de uma lista ligada em forma tabular
    lista = inserirAntena(lista, 'A', 0, 0);
    lista = inserirAntena(lista, 'A', 0, 1);
    lista = inserirAntena(lista, 'A', 1, 1);
    lista = inserirAntena(lista, 'B', 1, 2);
    imprimirAntenas(lista);
    printf("\n");

    removerAntena(lista, 1, 1);
    imprimirAntenas(lista);
    printf("\n");
}
```

output:

```
root@quedas: /mnt/c/Users/edgar/Desktop/IPCA
ANTENAS:
| Frequência | Posição |
|-----|-----|
|      B      | ( 1, 2) |
|      A      | ( 1, 1) |
|      A      | ( 0, 1) |
|      A      | ( 0, 0) |

ANTENAS:
| Frequência | Posição |
|-----|-----|
|      B      | ( 1, 2) |
|      A      | ( 0, 1) |
|      A      | ( 0, 0) |
```



## Caso 2: Testar o carregamento de ficheiro, cálculo de efeitos nefastos e a visualização dos resultados

Conteúdo do ficheiro “antenas.txt”:

```
antenas.txt
.....
...|. ...o...
...o.....
...o.....
...o.....
.....
.....
.....
.....A...
.....A..
.....
.....
```

main.c:

```
//Teste de carregar um ficheiro txt, detetar efeito nefasto, e imprimir o conteúdo do ficheiro com os efeitos nefastos (#) adicionados
printf("\n");
lista = carregarAntenas("antenas.txt");
listaEfeitoNefasto = efeitoNefasto(lista);
imprimirAntenasNefasto("antenas.txt", listaEfeitoNefasto);
printf("\n");
imprimirNefasto(listaEfeitoNefasto);
printf("\n");
imprimirAntenas(lista);|
```

output:

|               |                   |                      |
|---------------|-------------------|----------------------|
| .....#....#   | EFEITOS NEFASTOS: | ANTENAS:             |
| ...#....O...  | Posição           | Frequência   Posição |
| .....O....#.  | -----             | -----                |
| ..#....O....  | ( 0,11)           |                      |
| ....O....#..  | ( 3, 2)           |                      |
| ..#....#..... | ( 5, 6)           |                      |
| ....O....#..  | ( 1, 3)           | A ( 9, 9)            |
| ..#....#..... | ( 4, 9)           | A ( 8, 8)            |
| ...#.....     | ( 7, 0)           | O ( 4, 4)            |
| #.....#.....  | ( 0, 6)           | O ( 3, 7)            |
| .....A....    | ( 6, 3)           | O ( 2, 5)            |
| .....A....    | ( 2,10)           | O ( 1, 8)            |
| .....#.       | ( 5, 1)           |                      |
| .....         | ( 7, 7)           |                      |
| .....         | (10,10)           |                      |

## 6. Anexos

**Código fonte:** Disponível em [https://github.com/edgarcasal/projeto\\_EDA.git](https://github.com/edgarcasal/projeto_EDA.git)

**Documentação Técnica:** Disponível em </doxdoc/html/index.html>