Instituto Politécnico do Cávado e do Ave

Escola Superior de Tecnologia

Estrutura de Dados Avançadas

Licenciatura em Engenharia de Sistemas Informáticos

Trabalho Prático

Gestão de antenas e efeitos nefastos

Pedro Lourenço Morais Rocha – a23009

março de 2025

Resumo

Este relatório apresenta o desenvolvimento da Fase 1 do projeto de avaliação da Unidade Curricular Estruturas de Dados Avançadas (EDA), focado na implementação de listas ligadas em C para gerenciar antenas em uma cidade fictícia. O sistema carrega dados do arquivo antenas.txt (matriz 12x12), permite inserção e remoção de antenas, deteta efeitos nefastos baseados em interferências de frequência e exibe resultados em formato tabular. A solução utiliza estruturas dinâmicas (Antena e Nefasto), modularização em quatro arquivos e documentação *Doxygen*, atendendo aos requisitos do enunciado.

Índice

Índice de Ilustrações

1. Introdução

No âmbito da cadeira de Estrutura de Dados Avançadas, lecionada por Luís Ferreira no curso de Engenharia de Sistemas Informáticos, apresentamos este relatório sobre o desenvolvimento de um programa de gestão de antenas e efeitos nefastos. objetivo desta fase é aplicar os conhecimentos adquiridos sobre estruturas de dados dinâmicas em C, especificamente listas ligadas, para gerir antenas em uma cidade fictícia, detetar efeitos nefastos decorrentes de interferências entre antenas de mesma frequência e representá-los em uma estrutura tabular. O projeto utiliza o arquivo antenas.txt como entrada e segue os requisitos do enunciado, incluindo modularização, armazenamento em arquivo e documentação com padrões *Doxygen*.

2. Descrição do projeto

2.1 Objetivos

Conforme o enunciado, a Fase 1 visa:

- Definir uma estrutura de dados (ED) como lista ligada para representar antenas com frequência e coordenadas.
- Carregar dados de antenas de um arquivo de texto (antenas.txt) em uma matriz de dimensões arbitrárias.
- Implementar operações de manipulação da lista ligada:
    - Inserção de antenas.
    - Remoção de antenas.
    - Dedução automática de efeitos nefastos em uma lista ligada.
    - Listagem tabular das antenas e efeitos nefastos.

2.2 Estrutura do Código

O projeto é composto por quatro arquivos principais:

- **estrutura.h**: Define as estruturas Antena (frequência, coordenadas x e y, ponteiro para próxima antena) e Nefasto (coordenadas x e y, ponteiro para próximo efeito).
- **lista.h**: Declara funções para manipulação de listas ligadas e da matriz, com limites de MAX_LINHAS (100) e MAX_COLUNAS (100).
- **lista.c**: Implementa as funções declaradas, incluindo criação, inserção, remoção, deteção de efeitos e manipulação da matriz.
- **main.c**: Integra as funcionalidades em um fluxo de execução que testa o sistema com o arquivo antenas.txt.

2.3 Funcionalidades Principais

- **Carregamento**: Lê o arquivo antenas.txt para inicializar listas e matriz.
- **Inserção e Remoção**: Adiciona ou remove antenas da lista ligada e atualiza a matriz.

- **Deteção de Efeitos**: Calcula pontos de interferência entre antenas de mesma frequência.
- **Visualização**: Exibe listas e matriz em formato tabular.

3. Análise Técnica

3.1 Estrutura de Dados

- **Lista Ligada de Antenas**: Cada nó (Antena) armazena um caractere (frequência) e coordenadas (x, y), com um ponteiro para o próximo nó. Inserção ocorre no início, e remoção tem complexidade.
- **Lista Ligada de Efeitos Nefastos**: Cada nó (Nefasto) contém apenas coordenadas (x, y), evitando duplicatas durante a inserção.

3.2 Carregamento de Dados

- A função *carregarAntenas* lê o antenas.txt linha a linha, inserindo antenas na lista ligada quando encontra caracteres diferentes de '.'.

```
Antena* carregarAntenas(const char* filename) {
    FILE* file = fopen(filename, "r");
    if (!file) {
        printf("Erro ao abrir o ficheiro!\n");
        return NULL;
    }
    Antena* lista = NULL;
    char linha[100];
    for (int y = 0; fgets(linha, sizeof(linha), file); y++) {
        for (int x = 0; linha[x] != '\0' && linha[x] != '\n'; x++) {
            if (linha[x] != '.') inserirAntena(&lista, linha[x], x, y);
        }
    }
    fclose(file);
    return lista;
}
```

Figura 1: Função para ler o antena.txt

- Suporta matrizes de qualquer dimensão até 100x100, conforme definido por *MAX_LINHAS* e *MAX_COLUNAS*.

3.3 Operações de manipulação

- **Inserção**: *inserirAntena* adiciona uma antena no início da lista, alocando memória dinamicamente.

```c
int inserirAntena(Antena** lista, char freq, int x, int y) {
    Antena* nova = criarAntena(freq, x, y);
    if (!nova) return 0;
    nova->prox = *lista;
    *lista = nova;
    return 1;
}
```

Figura 2: Função para inserir antenas

- **Remoção**: *removerAntena* busca por coordenadas e libertar o nó correspondente.

```c
int removerAntena(Antena** lista, int x, int y) {
    Antena* atual = *lista, *anterior = NULL;
    while (atual && (atual->x != x || atual->y != y)) {
        anterior = atual;
        atual = atual->prox;
    }
    if (!atual) return 0;
    if (!anterior) *lista = atual->prox;
    else anterior->prox = atual->prox;
    free(atual);
    return 1;
}
```

Figura 3: Função para remover antena

- **Deteção de Efeitos**: *detetarEfeitosNefastos* compara pares de antenas de mesma frequência, calculando pontos simétricos com base na regra do enunciado (distância dupla). Complexidade $O(n^2)$.

```
Nefasto* detetarEfeitosNefastos(Antena* lista) {
    Nefasto* efeitos = NULL;
    for (Antena* a1 = lista; a1; a1 = a1->prox) {
        for (Antena* a2 = lista; a2; a2 = a2->prox) {
            if (a1 != a2 && a1->frequencia == a2->frequencia) {
                int dx = abs(a1->x - a2->x);
                int dy = abs(a1->y - a2->y);
                int ex1 = a1->x > a2->x ? a1->x + dx : a1->x - dx;
                int ey1 = a1->y > a2->y ? a1->y + dy : a1->y - dy;
                int ex2 = a2->x > a1->x ? a2->x + dx : a2->x - dx;
                int ey2 = a2->y > a1->y ? a2->y + dy : a2->y - dy;
                inserirNefasto(&efeitos, ex1, ey1);
                inserirNefasto(&efeitos, ex2, ey2);
            }
        }
    }
    return efeitos;
}
```

Figura 4: Função para detetar efeitos nefastos

- **Listagem**: *imprimirAntenas* e *imprimirEfeitosNefastos* exibem os dados em tabelas na consola.

```
void imprimirAntenas(Antena* lista) {
    printf("\nLista de Antenas:\n");
    printf("Frequência | Coordenadas\n");
    printf("-----------------------\n");
    while (lista) {
        printf("    %c      | (X:%d, Y:%d)\n", lista->frequencia, lista->x, lista->y);
        lista = lista->prox;
    }
    printf("-----------------------\n");
}
```

Figura 5: Função que exibe as antenas

```c
void imprimirEfeitosNefastos(Nefasto* lista) {
    printf("\nEfeitos Nefastos:\n");
    printf("Coordenadas\n");
    printf("------------\n");
    while (lista) {
        printf("# |  (X:%d, Y:%d)\n", lista->x, lista->y);
        lista = lista->prox;
    }
    printf("------------\n");
}
```

Figura 6: Função que exibe os efeitos nefastos

4. Conclusão

O desenvolvimento da Fase 1 do projeto resultou em uma solução robusta e funcional que atende plenamente aos requisitos do enunciado, consolidando os conhecimentos sobre estruturas de dados dinâmicas em C. A utilização de listas ligadas para representar antenas e efeitos nefastos demonstrou eficiência nas operações de inserção e adequação para manipulação dinâmica, enquanto a modularização em quatro arquivos reflete boas práticas de organização de código. A documentação no formato *Doxygen*, presente nos arquivos, oferece clareza sobre a lógica das funções.

5. Repositório GitHub

No seguinte link encontra-se a versão mais atualizada do projeto, com um *ReadMe* bem definido que explicita o que se encontra no repositório *Git*:
https://github.com/PieWolfie/ProjetoEDA.git