

GESTÃO DE ANTENAS

Luís Filipe Ferreira Leite

Docente

Prof. Luís Ferreira

março, 2025

GESTÃO DE ANTENAS

Luís Filipe Ferreira Leite

Docente

Prof. Luís Ferreira

março, 2025

DECLARAÇÃO

Nome: Luís Filipe Ferreira Leite

Endereço eletrónico: a10232@alunos.ipca.pt

Título do Projeto: Gestão de Antenas

Docente: Prof. Luís Ferreira

Ano do projeto: março, 2025

Designação do Curso: Licenciatura em Engenharia de Sistemas Informáticos Nome do Mestrado.

Nos exemplares das Dissertações /Projetos/ Relatórios de Estágio de mestrado ou de outros trabalhos entregues para prestação de Provas Públicas, e dos quais é obrigatoriamente enviado exemplares para depósito legal, deve constar uma das seguintes declarações:

☒ É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA DISSERTAÇÃO/TRABALHO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

☐ É AUTORIZADA A REPRODUÇÃO PARCIAL DESTA DISSERTAÇÃO/TRABALHO (indicar, caso tal seja necessário, nº máximo de páginas, ilustrações, gráficos, etc.), APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

☐ DE ACORDO COM A LEGISLAÇÃO EM VIGOR, NÃO É PERMITIDA A REPRODUÇÃO DE QUALQUER PARTE DESTA DISSERTAÇÃO/TRABALHO

Instituto Politécnico do Cávado e do Ave, ____/____/____

Assinatura: _____

ÍNDICE

ÍNDICE	III
INTRODUÇÃO	1
1. ESTRUCTURA DO PROJETO	3
1.1 DESCRIÇÃO DOS FICHEIROS	3
1.1.1 ESTRUTURAS.H	3
1.1.2 FUNCOES.C	3
1.1.2.1 Funções Implementadas	3
1.1.3 FUNCOES.H	4
1.1.4 MAIN.C	4
CONCLUSÕES	5

INTRODUÇÃO

Este projeto, desenvolvido no âmbito da Unidade Curricular de Estruturas de Dados Avançadas (EDA), tem como principal objetivo consolidar os conhecimentos adquiridos ao longo do semestre, ao aplicar conceitos fundamentais sobre estruturas de dados dinâmicas na linguagem de programação C. O código-fonte deste projeto está disponível publicamente no GitHub, que pode ser consultado no repositório <https://github.com/a10232/EDA-TrabalhoPratico>.

Para isso, foi implementado um sistema que modela uma cidade com várias antenas, em que cada uma delas opera numa frequência específica e distribuída num mapa representado por uma matriz. A solução desenvolvida recorre ao uso de estruturas de dados dinâmicas, ao armazenamento em ficheiro, sendo também garantida a documentação clara e organizada através do Doxygen.

Os resultados obtidos demonstram uma implementação eficiente das estruturas de dados necessárias para gestão de antenas.

No final, conclui-se que a utilização de estruturas de dados dinâmicas contribui significativamente para uma gestão eficaz da informação, sendo essencial para o desenvolvimento de aplicações mais robustas e escaláveis. O projeto também reforçou a importância da modularização e da documentação bem estruturada, aspetos fundamentais para a criação de código mais organizado, compreensível e reutilizável.

1. ESTRUTURA DO PROJETO

O projeto é composto por quatro ficheiros principais: *Funcoes.c*, *Funcoes.h*, *Estruturas.h* e *Main.c*.

1.1 DESCRIÇÃO DOS FICHEIROS

1.1.1 ESTRUTURAS.H

Nele estão declaradas duas estruturas de dados *Antena* e *Localizacao*, que permitem armazenar informações relacionadas com as antenas e locais com efeitos nefastos. Ambas as estruturas utilizam listas ligadas para armazenar múltiplos elementos, de forma a permitir a gestão dinâmica das antenas e localizações neste projeto.

1.1.2 FUNCOES.C

Neste ficheiro estão implementadas as funções necessárias para a gestão de antenas de uma cidade.

1.1.2.1 Funções Implementadas

- carregarAntenasFile - Função que carrega as antenas de um ficheiro de texto para uma lista ligada. Ela abre o ficheiro especificado pelo nome, lê cada linha do ficheiro e cria uma nova antena para cada caractere que não seja um ponto('.') ou uma nova linha ('\n'). Uma nova antena é adicionada à lista ligada. Se ocorrer um erro ao abrir o ficheiro ou ao alocar memória para uma nova antena, a função imprime uma mensagem de erro e retorna.
- criarAntena - Função que cria uma nova antena com a frequência, linha e coluna especificadas. Ela aloca memória para a nova antena e inicializa os seus campos. Se a alocação de memória falhar, a função imprime uma mensagem de erro e termina o programa.
- adicionarAntena - Esta função adiciona uma antena à lista ligada, inserindo-a no início da lista. Ela ajusta o apontador *next* da nova antena para apontar para a antena atualmente no início da lista e atualiza o apontador da lista para apontar para a nova antena.
- removerAntena - Esta função remove uma antena da lista ligada com base na frequência, linha e coluna especificadas. Ela percorre a lista ligada procurando uma antena que corresponda aos critérios especificados. Se encontrar a antena, ajusta os apontadores da lista para removê-la e liberta a memória da antena removida. A função retorna 1 se a antena foi removida com sucesso e 0 caso contrário.

- listarAntLocalNef - Esta função lista de forma tabular na consola as antenas e localizações com efeito nefasto. Ela percorre a lista de antenas e imprime a frequência, linha e coluna de cada antena. Em seguida, percorre a lista de localizações com efeito nefasto e imprime a linha e coluna de cada localização.
- libertaMemoria - Esta função liberta a memória alocada para a lista ligada de antenas. Ela percorre a lista ligada, faz com que a memória de cada antena até que a lista esteja vazia.

1.1.3 FUNCOES.H

Este ficheiro contém as declarações das funções implementadas em Funcoes.c. Este permite que outros ficheiros do projeto utilizem as funções implementadas, ao fornecer as assinaturas das funções e os tipos de dados necessários.

1.1.4 MAIN.C

No ficheiro Main.c são realizadas várias operações para gerir antenas e localizações. No início da função main são declaradas duas listas ligadas: uma para armazenar as antenas e outra para as localizações, ambas inicializadas a NULL. As listas são configuradas com NULL para confirmar que se encontram vazias.

A primeira operação trata-se do carregamento das antenas através de um ficheiro de texto *file-antenas.txt*. A função `carregarAntenasFile` é responsável por ler as informações do ficheiro e inserir as antenas na lista ligada. Dessa forma, as antenas são carregadas dinamicamente, com detalhes como a sua frequência e posição sendo automaticamente adicionadas à lista.

Além disso, o código também permite a manipulação manual das antenas. Uma nova antena, com a frequência 'B' e na posição (linha 5, coluna 5), é criada e adicionada à lista de antenas. Para isso, são utilizadas as funções `criarAntena` e `adicionarAntena`, que garantem que a antena seja corretamente inserida na lista. Também é possível remover uma antena específica, como a que possui frequência 'A' e posição (linha 3, coluna 3), ao utilizar a função `removerAntena`.

Por fim, o código mostra as antenas e localizações afetadas por efeitos nefastos, através da função `listarAntLocalNef`. Após as modificações, a memória alocada para a lista de antenas é limpa com a função `libertaMemoria`, evitando lixo na memória do equipamento após utilização do programa.

CONCLUSÕES

Neste projeto, foi desenvolvido um sistema para gerir antenas de uma cidade, usando listas ligadas para armazenar e manipular dados de antenas e localizações com efeitos nefastos. O código foi dividido em quatro ficheiros principais: `Funcoes.c`, `Funcoes.h`, `Estruturas.h` e `Main.c`, que organizam e implementam as funcionalidades do sistema.

A estrutura `Antena` armazena dados como frequência e localização de cada antena, enquanto a estrutura `Localizacao` guarda informações sobre as áreas nefastas. Foram implementadas funções para criar, adicionar, remover e listar antenas e localizações, além de limpar a memória alocada. A função principal, no `Main.c`, realiza toda a integração, que permite carregar dados de um ficheiro, adicionar e remover antenas, listar informações e libertar memória.

Na Fase 1 os objetivos do projeto foram alcançados com sucesso, exceto pela dedução automática das localizações com efeito nefasto e a respetiva representação sob a forma de uma lista ligada (3.c.). Foi conseguido implementar um sistema funcional que permite carregar antenas de um ficheiro de texto para uma lista ligada, adicionar novas antenas manualmente, remover antenas específicas da lista, listar as antenas e as localizações com efeito nefasto na consola, e libertar a memória alocada para as listas ligadas.

Este projeto contribui para a compreensão e aplicação de conceitos fundamentais de estruturas de dados, como listas ligadas, em contexto prático. Além disso, demonstra a importância de uma boa organização do código e da modularidade, facilitando a manutenção e a expansão do sistema.

No futuro próximo, será dada continuidade ao projeto de Estruturas de Dados e Algoritmos (EDA) ao desenvolver exercícios sobre a matéria de grafos.