|  |
| --- |
| IPCA |
| Projeto Fase 1 |
| Estruturas de Dados Avançadas |

|  |
| --- |
| Telmo Silva | a20456  30/3/2025 |

# Índice

1. Introdução

2. Objetivos do Projeto

3. Estrutura de Dados

4. Arquitetura e Modularização do Projeto

5. Funcionalidades Implementadas

6. Demonstração Visual (Prints)

7. Documentação Técnica com Doxygen

8. Conclusão

9. Anexos

# 1. Introdução

O presente relatório descreve o desenvolvimento da Fase 1 do projeto da unidade curricular de Estruturas de Dados Avançadas (EDA). O projeto consiste na criação de uma aplicação que simula a gestão de antenas numa matriz, detetando efeitos nefastos de interferência entre elas. A aplicação foi desenvolvida em linguagem C, utilizando listas ligadas, modularização e leitura de ficheiros.

# 2. Objetivos do Projeto

* Utilizar estruturas de dados dinâmicas (listas ligadas) para armazenar antenas e efeitos nefastos.
* Implementar funcionalidades como inserção, remoção, listagem e deteção de interferências.
* Ler uma configuração inicial a partir de um ficheiro.
* Representar graficamente os dados numa grelha 2D.
* Modularizar o projeto com ficheiros separados e gerar documentação com Doxygen.

# 3. Estrutura de Dados

O projeto baseia-se em três estruturas principais:

* Coordenada: Representa uma posição (linha, coluna).
* Antena: Contém a frequência (caracter), uma coordenada e ponteiro para a próxima antena.
* EfeitoNefasto: Armazena posições onde há interferência, semelhante à estrutura de antenas.

# 4. Arquitetura e Modularização do Projeto

O código está dividido em vários ficheiros, permitindo maior organização e reutilização de código:

* main.c: Contém o menu principal e chamadas às funções.
* main\_sem\_menu.c: Versão automática para testes diretos.
* antenas.c / antenas.h: Implementa e declara todas as funções relacionadas com antenas, listas e efeitos nefastos.
* README.txt: Descreve a compilação e utilização do programa.

# 5. Funcionalidades Implementadas

Nesta secção são detalhadas as principais funcionalidades implementadas no projeto, com explicações técnicas e exemplos práticos.

## 5.1 Inserção de Antenas

A função 'inserirAntena' permite inserir uma nova antena na lista ligada. Antes da inserção, é feita uma verificação para garantir que a posição não está ocupada, utilizando a função auxiliar 'jaExisteAntena'. Em caso positivo, a antena é criada e ligada no início da lista.

Exemplo: Inserção da antena 'O' na posição (4, 3) com sucesso.

## 5.2 Remoção de Antenas

A função 'removerAntena' percorre a lista ligada à procura de uma antena com as coordenadas especificadas. Se encontrada, é removida e libertada da memória. É dada especial atenção ao caso da cabeça da lista ser a antena alvo.

## 5.3 Listagem de Antenas

A listagem percorre a lista de antenas e imprime de forma tabular a frequência e as coordenadas de cada antena. Esta informação é apresentada ao utilizador sempre que necessário no menu.

## 5.4 Cálculo de Efeitos Nefastos

Esta funcionalidade verifica interferências entre antenas do tipo 'A'. Quando duas antenas estão alinhadas (horizontal, vertical ou diagonal) e equidistantes de um ponto vazio, esse ponto é considerado uma localização com efeito nefasto. A função 'calcularEfeitoNefasto' adiciona essas localizações numa nova lista ligada.

## 5.5 Visualização da Grelha

São utilizadas duas grelhas 2D para mostrar a disposição das antenas (12x12) e os efeitos nefastos (10x10). Cada célula é representada por '.', 'a', 'A', 'O' ou '#'. Esta funcionalidade é útil para validação visual do sistema.

## 5.6 Leitura de Ficheiro

O ficheiro de configuração inicial ('antenas\_base.txt') é lido linha a linha, e cada antena é detetada com base nos caracteres 'a', 'A' ou 'O'. As coordenadas são atribuídas dinamicamente com base na posição da leitura.

## 5.7 Interface com Menu

O programa apresenta ao utilizador um menu interativo com várias opções. Cada escolha chama uma função respetiva que executa a operação desejada. O menu é apresentado continuamente até que o utilizador escolha sair.

# 6. Demonstração Visual (Prints)

Abaixo são apresentados os principais momentos da execução do programa, com capturas do terminal e funcionalidades em funcionamento.

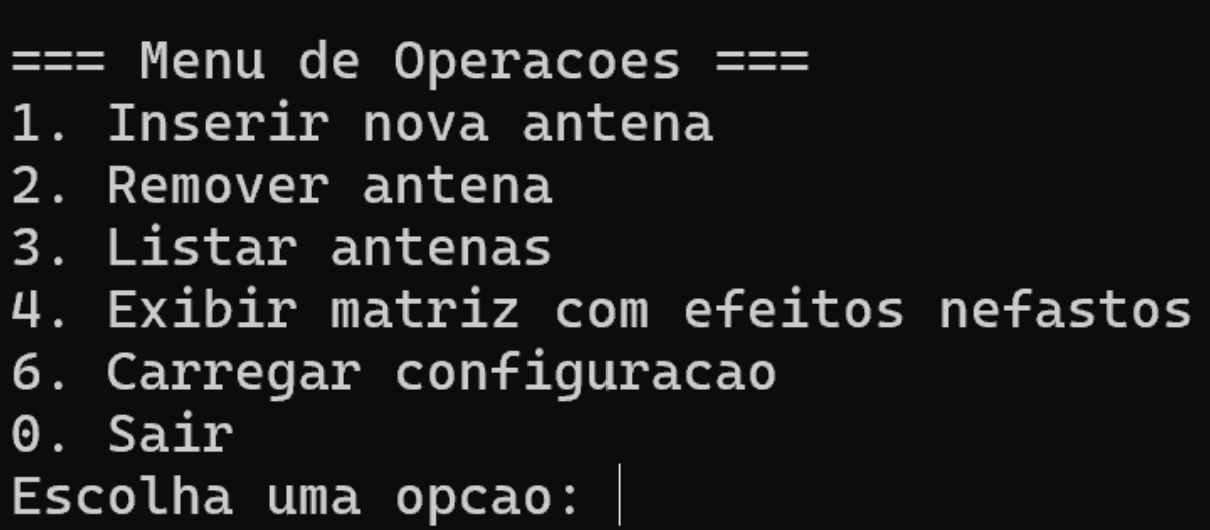


Figura – Menu de Operações

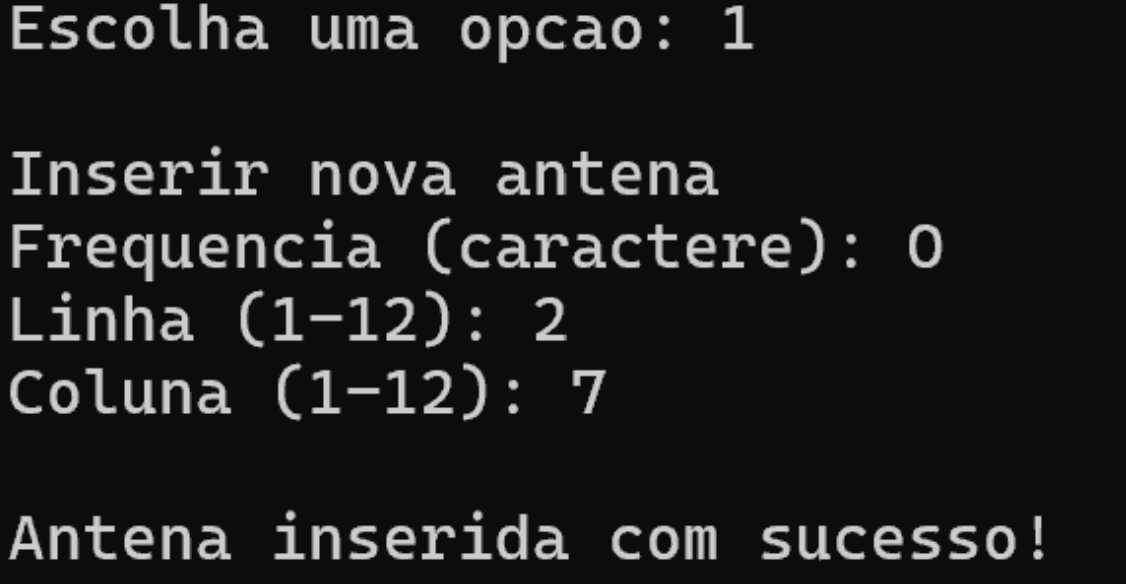


Figura – Inserção de uma nova antena na posição (2,7)

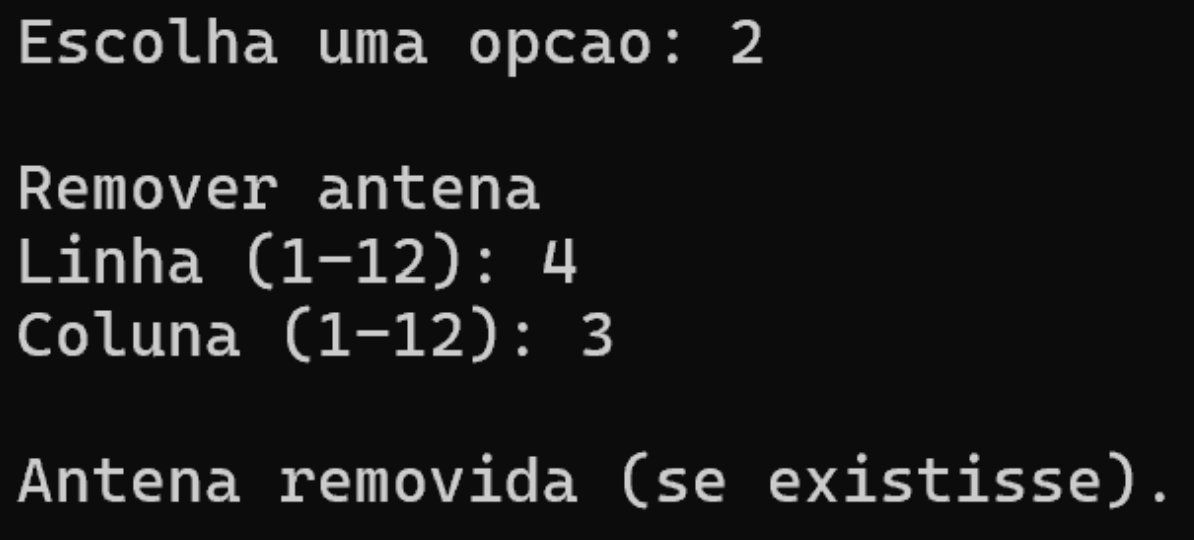


Figura – Remoção da antena na posição (4,3)

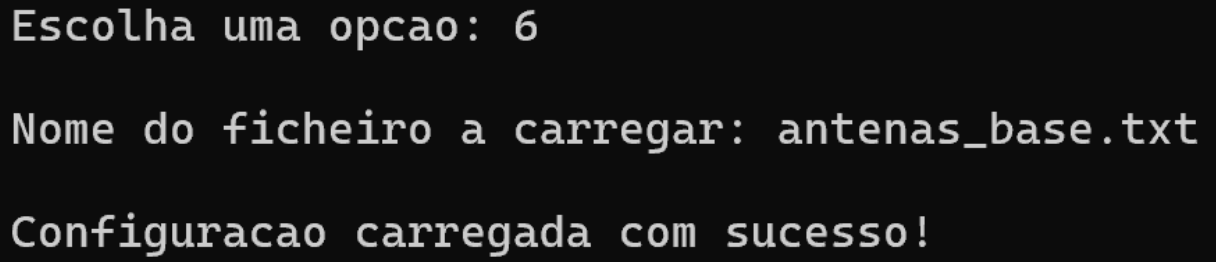


Figura – Configuração inicial carregada a partir do ficheiro

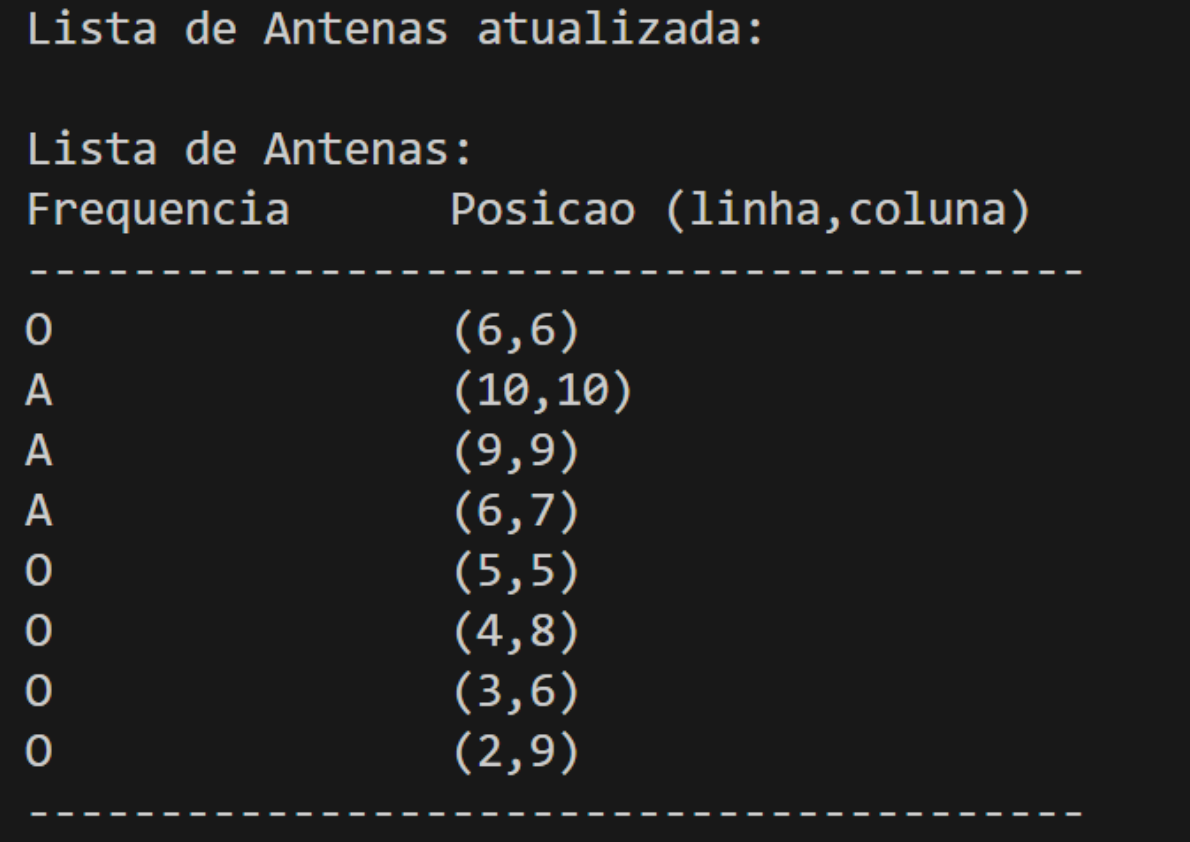


Figura – Lista de antenas atualizada após carregamento

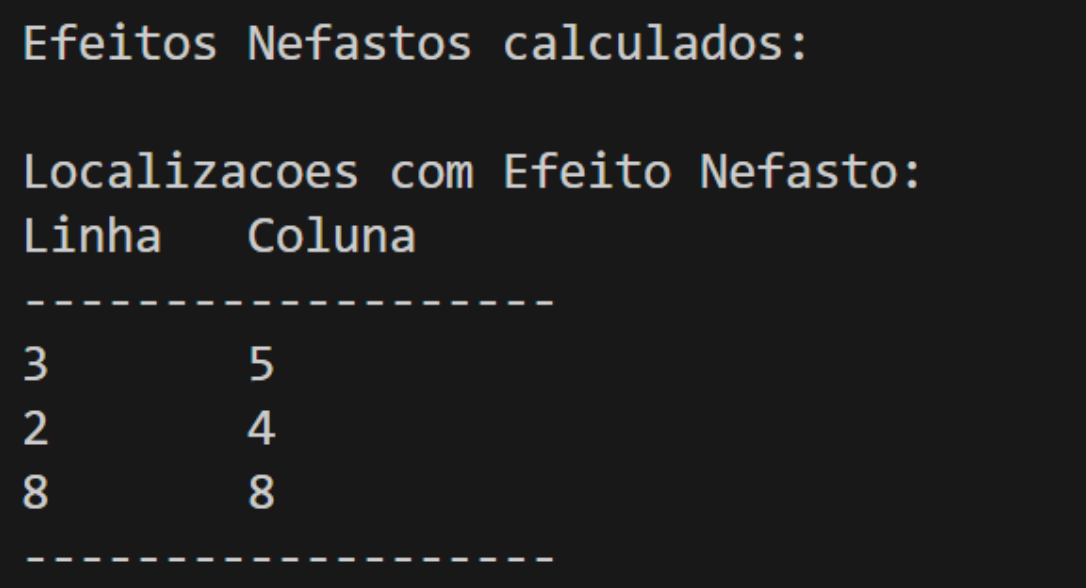


Figura – Localizações detetadas com efeito nefasto

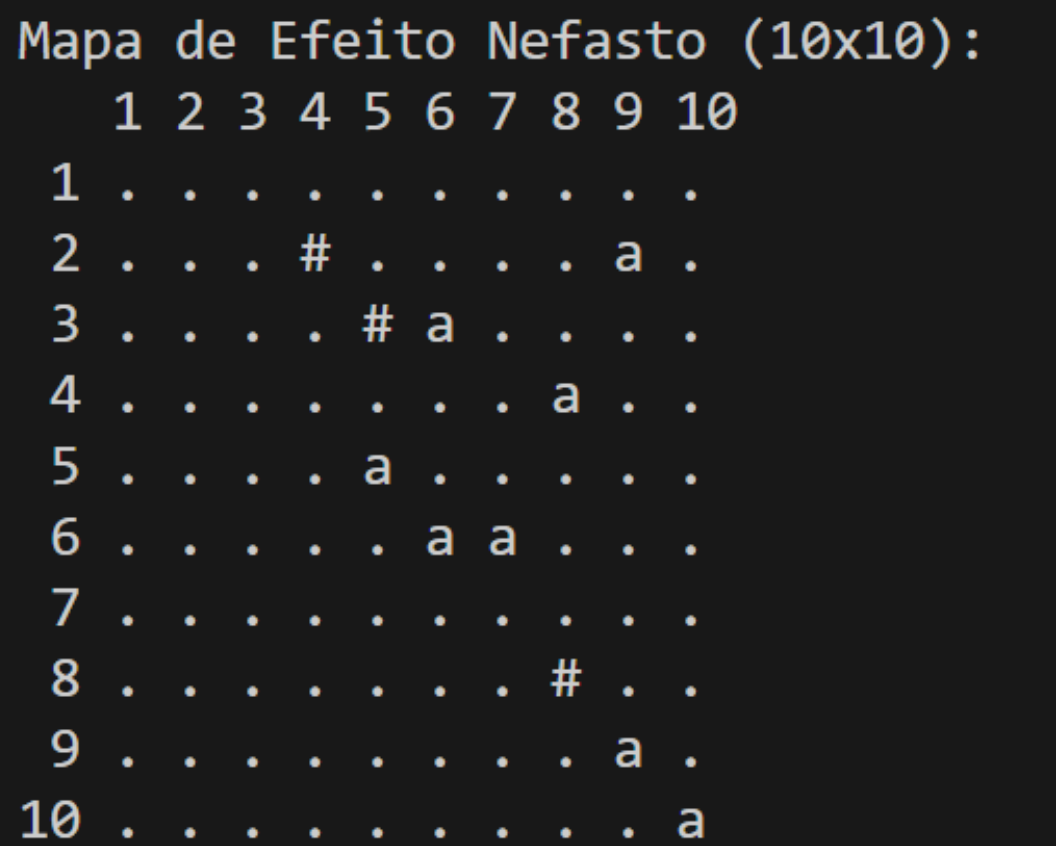


Figura – Mapa 10x10 com efeitos nefastos ('#')

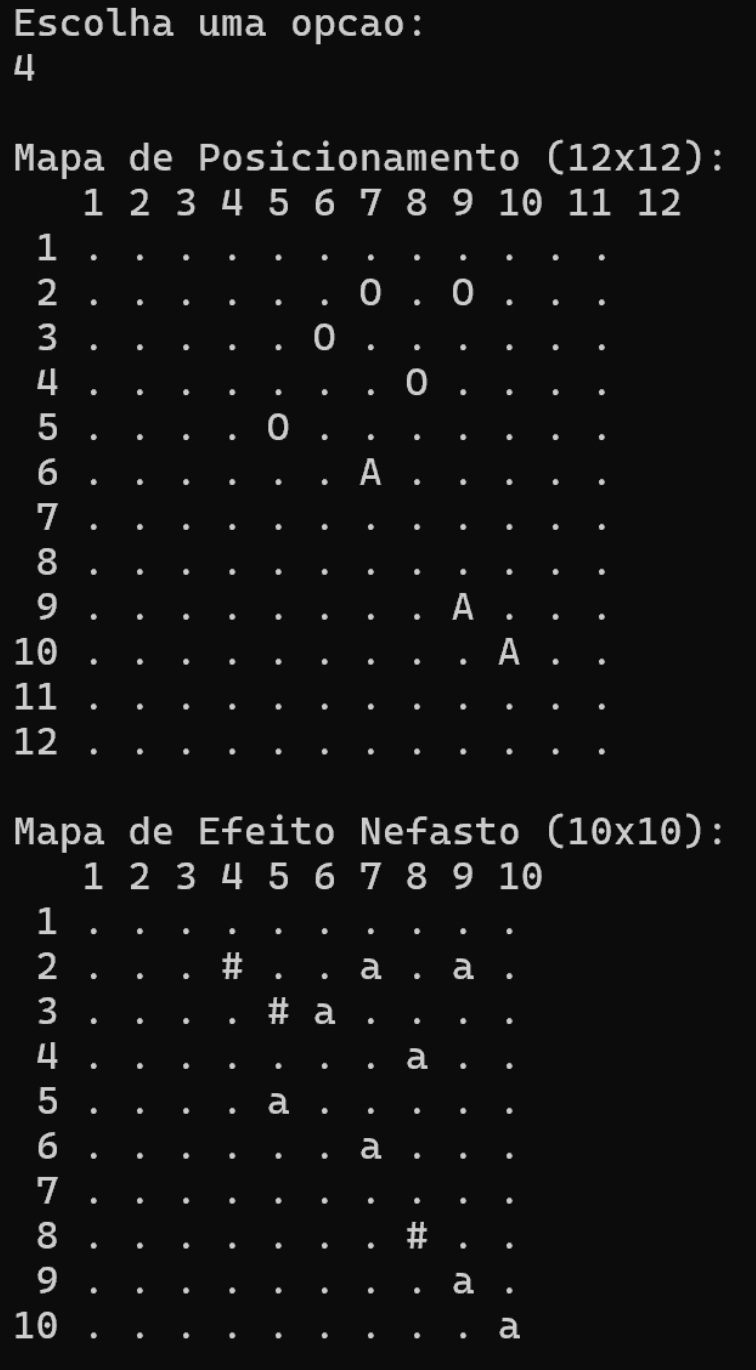


Figura – Mapa 12x12 com antenas e efeitos

# 7. Documentação Técnica com Doxygen

A documentação do projeto foi gerada com a ferramenta Doxygen. Esta ferramenta permite criar documentação HTML a partir de comentários estruturados no código-fonte. Abaixo seguem capturas das principais secções da documentação gerada automaticamente.

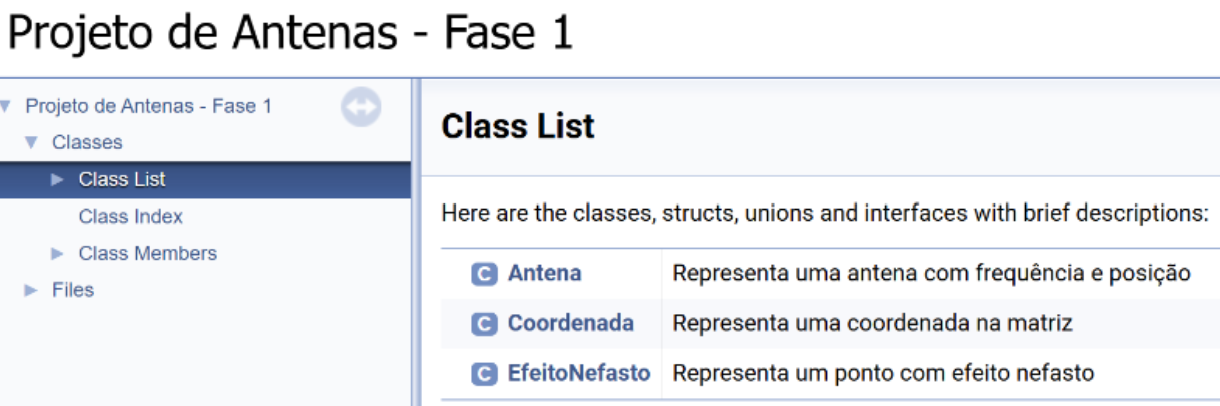


Figura – Página inicial da documentação Doxygen

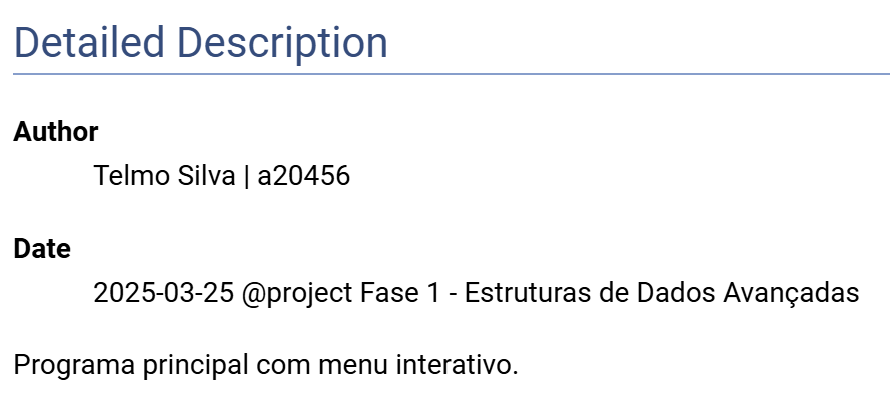


Figura – Descrição geral do projeto e autores

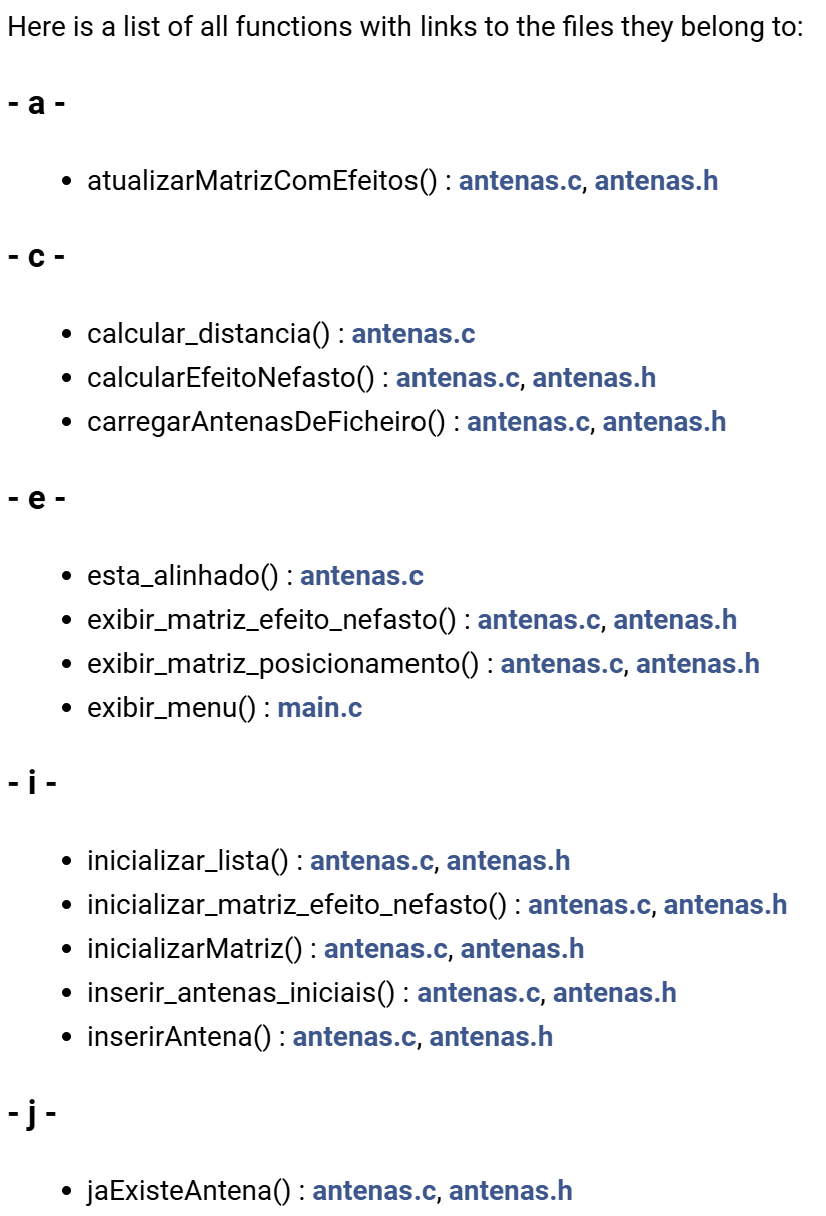


Figura – Lista de funções documentadas (parte 1)

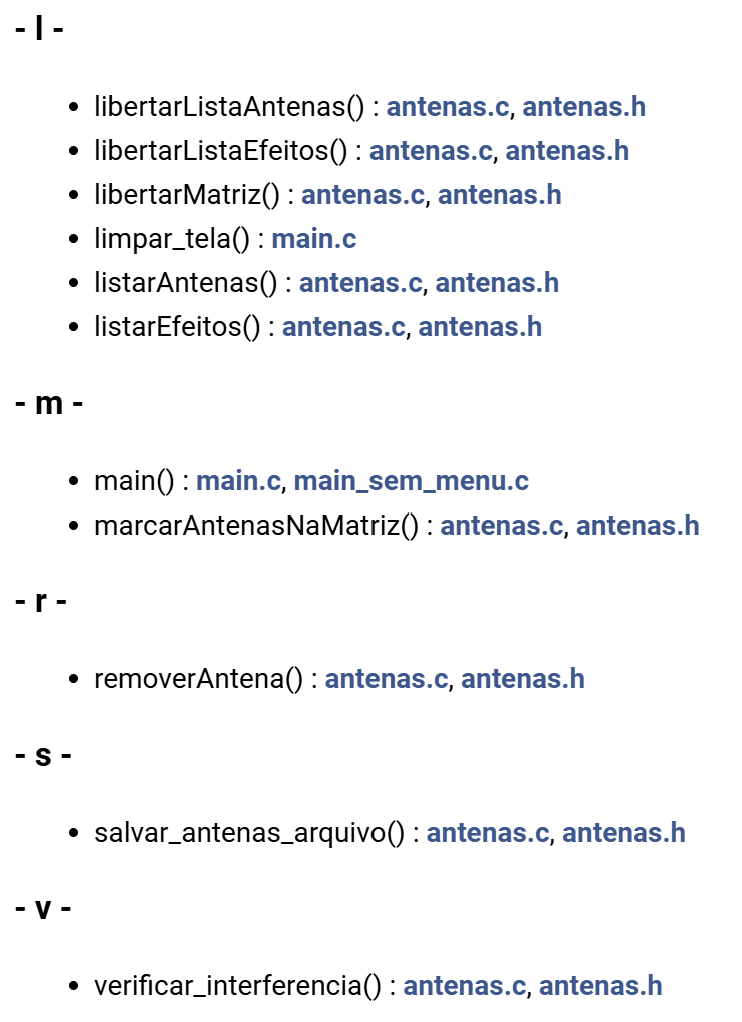


Figura – Lista de funções documentadas (parte 2)

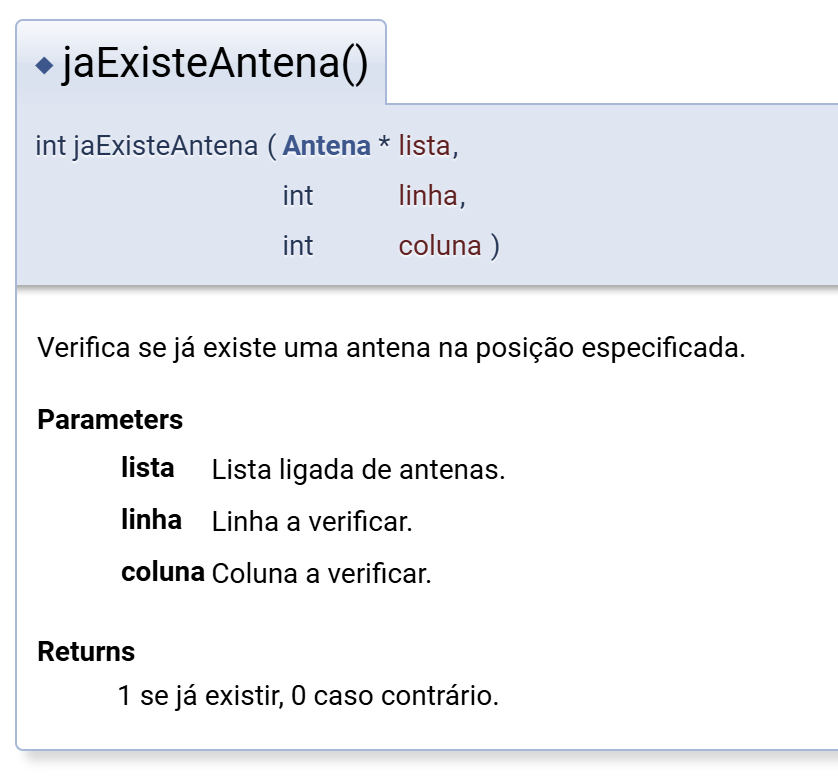


Figura – Função documentada: jaExisteAntena()

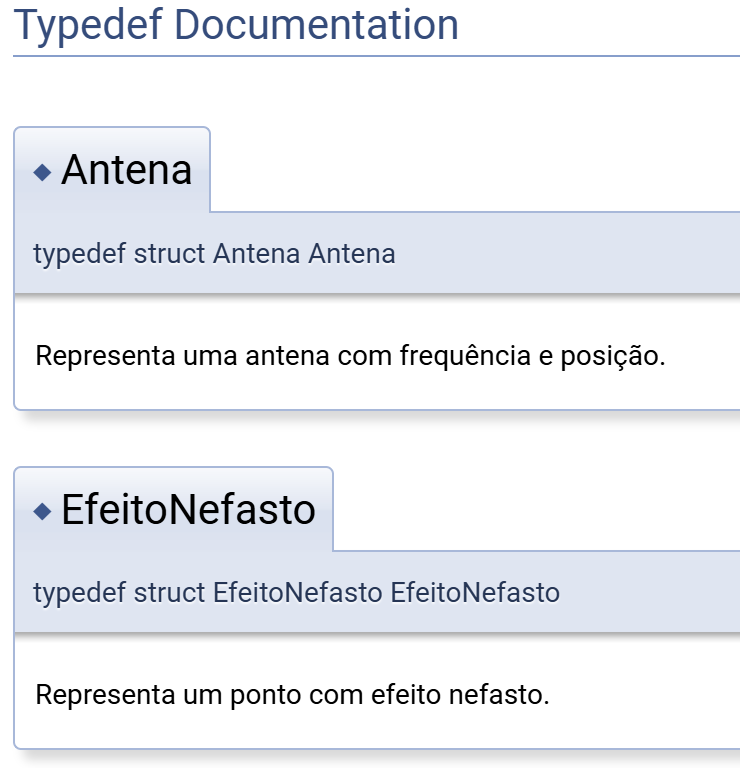


Figura – Definições de tipos (typedefs)

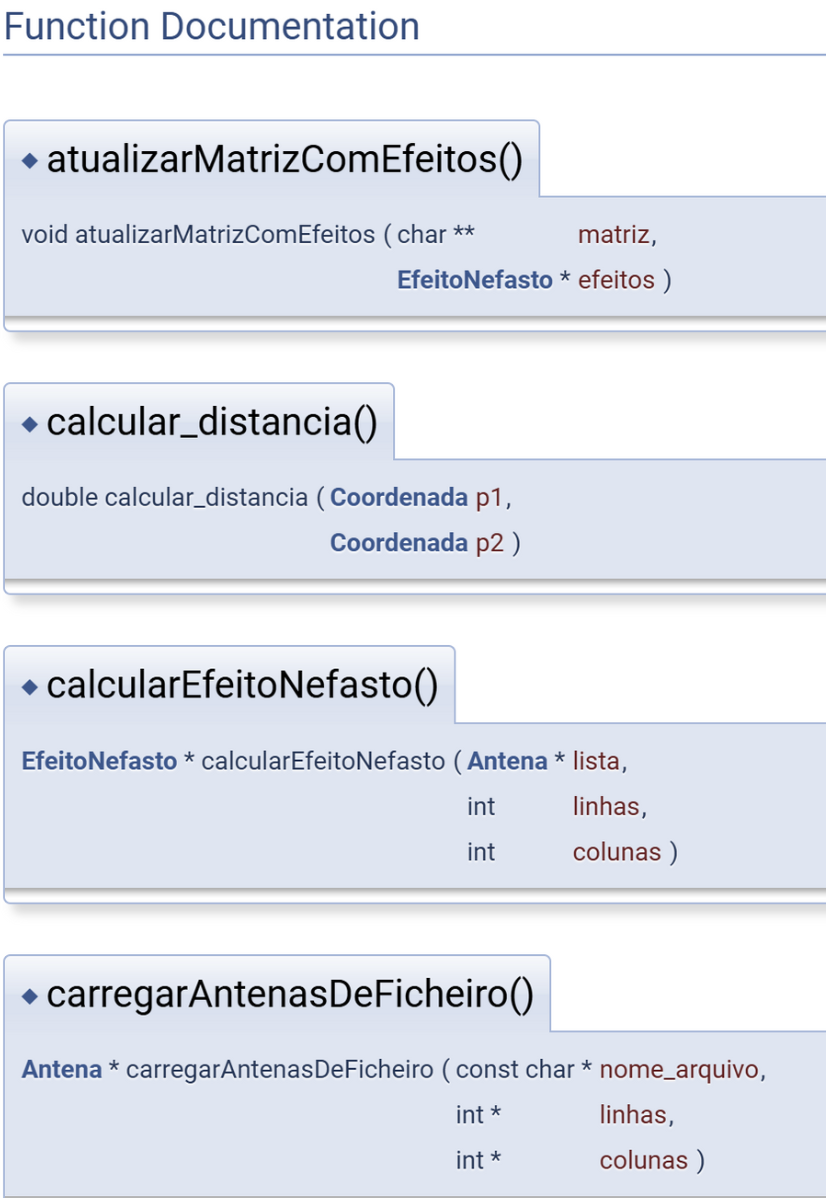


Figura – Documentação das funções principais

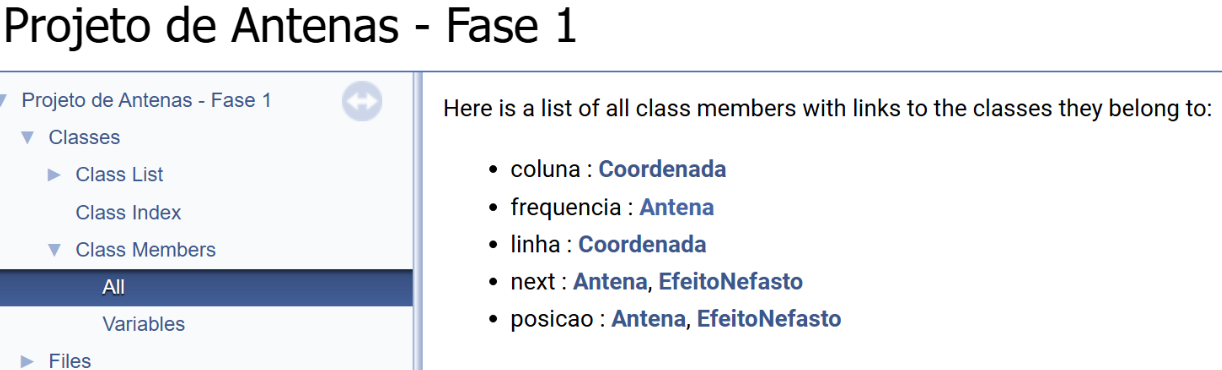


Figura – Lista de membros das estruturas

# 8. Conclusão

A Fase 1 do projeto de Estruturas de Dados Avançadas permitiu aplicar, de forma prática, os conceitos fundamentais abordados ao longo da unidade curricular, nomeadamente a utilização de listas ligadas, leitura e escrita de ficheiros, modularização de código e documentação técnica.

O projeto foi desenvolvido em linguagem C, cumprindo todos os requisitos definidos no enunciado:

* Implementação de uma estrutura dinâmica (lista ligada) para armazenar antenas e localizações com efeito nefasto;
* Desenvolvimento de funcionalidades como inserção, remoção, listagem e cálculo de interferências;
* Leitura automática de configurações a partir de ficheiros de texto;
* Representação tabular e visual de dados numa grelha bidimensional;
* Estrutura modular com separação clara entre ficheiros .h e .c;
* Geração de documentação técnica automática com a ferramenta Doxygen;
* Elaboração de um relatório técnico descritivo de todo o processo.

O código desenvolvido encontra-se devidamente comentado e organizado, facilitando a sua manutenção e extensão futura. A documentação foi gerada com Doxygen e permite uma navegação clara sobre as funções e estruturas utilizadas.

O trabalho foi compactado num único ficheiro ZIP, contendo:

* O código fonte (.c e .h);
* A documentação gerada automaticamente;
* O relatório técnico com prints ilustrativos e explicações detalhadas.

Para consulta de todo o trabalho desenvolvido, incluindo o código-fonte, documentação técnica e versões anteriores, o repositório Git está disponível em:

<https://github.com/TelmoSilva20456/TP_Fase1_EDA_A20456>

# 9. Anexos

* **Código-fonte do projeto:**

main.c – Interface interativa com menu

main\_sem\_menu.c – Versão automática para testes

antenas.c – Implementação das funcionalidades principais

antenas.h – Definições de estruturas e protótipos

README.txt – Instruções de compilação e execução

* **Ficheiro de configuração:**

antenas\_base.txt – Mapa inicial de antenas carregado pelo sistema

* **Prints de execução:**

Execução das funcionalidades no terminal com menus, inserção, remoção e grelha

* **Documentação técnica:**

Documentação gerada com Doxygen (.html + imagens)

* **Relatório técnico:**

Documento .docx completo com descrição, estrutura e justificação técnica.