

a22997_Projeto_EDA_Pos_Defesa

2

Generated by Doxygen 1.13.2

1 Topic Index	1
1.1 Topics	1
2 Data Structure Index	3
2.1 Data Structures	3
3 File Index	5
3.1 File List	5
4 Topic Documentation	7
4.1 Funções de Manipulação de Antenas	7
4.1.1 Detailed Description	7
4.1.2 Function Documentation	7
4.1.2.1 AdicionarAntena()	7
4.1.2.2 ApresentarLista()	8
4.1.2.3 ApresentarMatrizLista()	9
4.1.2.4 InserirAntena()	9
4.1.2.5 LerLista()	10
4.1.2.6 RemoverAntena()	11
4.2 Funções de Manipulação de Nefastos	11
4.2.1 Detailed Description	12
4.2.2 Function Documentation	12
4.2.2.1 ApresentarListaNef()	12
4.2.2.2 ApresentarMatrizListaNef()	12
4.2.2.3 EncontrarNefastos()	13
4.2.2.4 InserirNefasto()	14
4.3 Funções de Gestão de Memória	15
4.3.1 Detailed Description	15
4.3.2 Function Documentation	15
4.3.2.1 FreeAntNef()	15
4.3.2.2 FreeLista()	15
4.3.2.3 FreeListaNef()	16
5 Data Structure Documentation	17
5.1 Ant Struct Reference	17
5.1.1 Detailed Description	17
5.1.2 Field Documentation	17
5.1.2.1 freqAntena	17
5.1.2.2 id	17
5.1.2.3 proxAntena	18
5.1.2.4 x	18
5.1.2.5 y	18
5.2 AntNef Struct Reference	18
5.2.1 Detailed Description	18

5.2.2 Field Documentation	18
5.2.2.1 lista	18
5.2.2.2 listaNef	19
5.3 Nef Struct Reference	19
5.3.1 Detailed Description	19
5.3.2 Field Documentation	19
5.3.2.1 antena1	19
5.3.2.2 antena2	19
5.3.2.3 proxNef	20
5.3.2.4 x	20
5.3.2.5 y	20
6 File Documentation	21
6.1 projetoeda/funcao.h File Reference	21
6.1.1 Detailed Description	22
6.2 funcao.h	23
6.3 projetoedalib/funcao.h File Reference	23
6.3.1 Detailed Description	25
6.4 funcao.h	25
6.5 projetoeda/main.c File Reference	26
6.5.1 Detailed Description	26
6.5.2 Function Documentation	27
6.5.2.1 main()	27
6.6 projetoedalib/funcao.c File Reference	27
6.6.1 Detailed Description	28
Index	29

Chapter 1

Topic Index

1.1 Topics

Here is a list of all topics with brief descriptions:

Funções de Manipulação de Antenas	7
Funções de Manipulação de Nefastos	11
Funções de Gestão de Memória	15

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Ant	Estrutura que representa uma antena	17
AntNef	Estrutura que agrupa listas de antenas e posições nefastas	18
Nef	Estrutura que representa uma posição nefasta	19

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

projetoeda/funcao.h	Definições de estruturas e protótipos de funções para manipulação de antenas e nefastos . . .	21
projetoeda/main.c	Programa principal para manipulação de antenas e detecção de nefastos	26
projetoedalib/funcao.c	Implementação de funções para manipulação de antenas e nefastos	27
projetoedalib/funcao.h	Definições de estruturas e protótipos de funções para manipulação de antenas e nefastos . .	23

Chapter 4

Topic Documentation

4.1 Funções de Manipulação de Antenas

Conjunto de funções para criação, leitura e manipulação de antenas.

Functions

- `Ant * InserirAntena (Ant *lista, char freq, int x, int y, int id)`
Cria e insere uma nova antena na lista.
- `Ant * LerLista (const char *nomeFicheiro, const char *tipoFicheiro, int *linhas, int *colunas)`
Lê um ficheiro e constrói a lista de antenas.
- `int ApresentarLista (Ant *lista)`
Imprime no ecrã a lista de antenas.
- `int ApresentarMatrizLista (Ant *lista, int linhas, int colunas)`
Exibe uma representação matricial das antenas.
- `AntNef * RemoverAntena (AntNef *listaAntNef, int x, int y)`
Remove uma antena da lista e atualiza posições nefastas.
- `AntNef * AdicionarAntena (AntNef *listaAntNef, char freq, int x, int y, int linhas, int colunas)`
Adiciona ou atualiza uma antena na lista.

4.1.1 Detailed Description

Conjunto de funções para criação, leitura e manipulação de antenas.

4.1.2 Function Documentation

4.1.2.1 AdicionarAntena()

```
AntNef * AdicionarAntena (  
    AntNef * listaAntNef,  
    char freq,  
    int x,  
    int y,  
    int linhas,  
    int colunas)
```

Adiciona ou atualiza uma antena na lista.

Parameters

<i>listaAntNef</i>	Estrutura contendo ambas as listas
<i>freq</i>	Frequência da nova antena
<i>x</i>	Coordenada x da antena
<i>y</i>	Coordenada y da antena
<i>linhas</i>	Total de linhas da área
<i>colunas</i>	Total de colunas da área

Returns

Estrutura atualizada

Adiciona ou atualiza uma antena na lista.

Se já existir uma antena nas coordenadas especificadas, atualiza sua frequência. Caso contrário, insere uma nova antena.

Parameters

<i>listaAntNef</i>	Estrutura contendo listas de antenas e posições nefastas
<i>freq</i>	Frequência da antena
<i>x</i>	Coordenada x da antena
<i>y</i>	Coordenada y da antena
<i>linhas</i>	Número de linhas da área de cobertura
<i>colunas</i>	Número de colunas da área de cobertura

Returns

AntNef* Retorna a estrutura atualizada

4.1.2.2 ApresentarLista()

```
int ApresentarLista (
    Ant * lista)
```

Imprime no ecrã a lista de antenas.

Parameters

<i>lista</i>	Lista de antenas a ser exibida
--------------	--------------------------------

Returns

1 em caso de sucesso

Imprime no ecrã a lista de antenas.

Percorre a lista de antenas e imprime no ecrã a frequência, coordenadas e ID de cada uma.

Parameters

<i>lista</i>	Lista de antenas a apresentar
--------------	-------------------------------

Returns

int Retorna 1 em caso de sucesso

4.1.2.3 ApresentarMatrizLista()

```
int ApresentarMatrizLista (  
    Ant * lista,  
    int linhas,  
    int colunas)
```

Exibe uma representação matricial das antenas.

Parameters

<i>lista</i>	Lista de antenas
<i>linhas</i>	Número de linhas da matriz
<i>colunas</i>	Número de colunas da matriz

Returns

1 em caso de sucesso

Exibe uma representação matricial das antenas.

Gera uma representação matricial onde as antenas são mostradas com sua frequência e os espaços vazios são representados por pontos.

Parameters

<i>lista</i>	Lista de antenas
<i>linhas</i>	Número de linhas da matriz
<i>colunas</i>	Número de colunas da matriz

Returns

int Retorna 1 em caso de sucesso

4.1.2.4 InserirAntena()

```
Ant * InserirAntena (  
    Ant * lista,  
    char freq,  
    int x,  
    int y,  
    int id)
```

Cria e insere uma nova antena na lista.

Parameters

<i>lista</i>	Lista existente de antenas (pode ser NULL)
<i>freq</i>	Carácter representando a frequência da antena
<i>x</i>	Coordenada horizontal da antena
<i>y</i>	Coordenada vertical da antena
<i>id</i>	Identificador único para a nova antena

Returns

Ponteiro para a lista atualizada

Cria e insere uma nova antena na lista.

Aloca memória para uma nova antena e insere-a no início da lista.

Parameters

<i>lista</i>	Lista de antenas existente
<i>freq</i>	Carácter que representa a frequência da antena
<i>x</i>	Coordenada x da antena
<i>y</i>	Coordenada y da antena
<i>id</i>	Identificador único da antena

Returns

Ant* Retorna a lista atualizada com a nova antena

4.1.2.5 LerLista()

```
Ant * LerLista (  
    const char * nomeFicheiro,  
    const char * tipoFicheiro,  
    int * linhas,  
    int * colunas)
```

Lê um ficheiro e constrói a lista de antenas.

Parameters

<i>nomeFicheiro</i>	Nome base do ficheiro (sem extensão)
<i>tipoFicheiro</i>	Extensão do ficheiro (".txt" ou outro)
<i>linhas</i>	Ponteiro para armazenar número de linhas lidas
<i>colunas</i>	Ponteiro para armazenar número máximo de colunas

Returns

Lista de antenas criada ou NULL em caso de erro

Lê um ficheiro e constrói a lista de antenas.

Esta função lê um ficheiro (texto ou binário) e cria uma lista ligada de antenas com base nos caracteres encontrados que não sejam pontos ('.').

Parameters

<i>nomeFicheiro</i>	Nome base do ficheiro (sem extensão)
<i>tipoFicheiro</i>	Extensão do ficheiro (".txt" ou outro para binário)
<i>linhas</i>	Ponteiro para armazenar o número de linhas lidas
<i>colunas</i>	Ponteiro para armazenar o número máximo de colunas encontradas

Returns

Ant* Retorna a lista de antenas criada ou NULL em caso de erro

4.1.2.6 RemoverAntena()

```
AntNef * RemoverAntena (  
    AntNef * listaAntNef,  
    int x,  
    int y)
```

Remove uma antena da lista e atualiza posições nefastas.

Parameters

<i>listaAntNef</i>	Estrutura contendo ambas as listas
<i>x</i>	Coordenada x da antena a remover
<i>y</i>	Coordenada y da antena a remover

Returns

Estrutura atualizada

Remove uma antena da lista e atualiza posições nefastas.

Parameters

<i>listaAntNef</i>	Estrutura contendo listas de antenas e posições nefastas
<i>x</i>	Coordenada x da antena a remover
<i>y</i>	Coordenada y da antena a remover

Returns

AntNef* Retorna a estrutura atualizada

4.2 Funções de Manipulação de Nefastos

Conjunto de funções para cálculo e manipulação de posições nefastas.

Functions

- `Nef * InserirNefasto (Nef *listaNef, int x, int y, int ant1, int ant2, Ant *lista)`
Insere uma nova posição nefasta na lista.
- `Nef * EncontrarNefastos (Ant *lista, int linhas, int colunas)`
Calcula posições nefastas entre antenas.
- `int ApresentarListaNef (Nef *listaNef)`
Imprime no ecrã a lista de posições nefastas.
- `int ApresentarMatrizListaNef (Ant *lista, int linhas, int colunas, Nef *listaNef)`
Exibe matriz com antenas e posições nefastas.

4.2.1 Detailed Description

Conjunto de funções para cálculo e manipulação de posições nefastas.

4.2.2 Function Documentation

4.2.2.1 ApresentarListaNef()

```
int ApresentarListaNef (
    Nef * listaNef)
```

Imprime no ecrã a lista de posições nefastas.

Parameters

<code>listaNef</code>	Lista de nefastos a exibir
-----------------------	----------------------------

Returns

1 em caso de sucesso

Imprime no ecrã a lista de posições nefastas.

Parameters

<code>listaNef</code>	Lista de posições nefastas a apresentar
-----------------------	---

Returns

int Retorna 1 em caso de sucesso

4.2.2.2 ApresentarMatrizListaNef()

```
int ApresentarMatrizListaNef (
    Ant * lista,
    int linhas,
    int colunas,
    Nef * listaNef)
```

Exibe matriz com antenas e posições nefastas.

Parameters

<i>lista</i>	Lista de antenas
<i>linhas</i>	Número de linhas da matriz
<i>colunas</i>	Número de colunas da matriz
<i>listaNef</i>	Lista de posições nefastas

Returns

1 em caso de sucesso

Exibe matriz com antenas e posições nefastas.

Gera uma representação matricial mostrando antenas com sua frequência, posições nefastas com '#' e espaços vazios com '.'.

Parameters

<i>lista</i>	Lista de antenas
<i>linhas</i>	Número de linhas da matriz
<i>colunas</i>	Número de colunas da matriz
<i>listaNef</i>	Lista de posições nefastas

Returns

int Retorna 1 em caso de sucesso

4.2.2.3 EncontrarNefastos()

```
Nef * EncontrarNefastos (  
    Ant * lista,  
    int linhas,  
    int colunas)
```

Calcula posições nefastas entre antenas.

Parameters

<i>lista</i>	Lista de antenas a analisar
<i>linhas</i>	Número de linhas da área
<i>colunas</i>	Número de colunas da área

Returns

Lista de posições nefastas encontradas

Calcula posições nefastas entre antenas.

Identifica posições onde podem ocorrer interferências entre antenas da mesma frequência seguindo padrões geométricos específicos.

Parameters

<i>lista</i>	Lista de antenas a analisar
<i>linhas</i>	Número de linhas da área de cobertura
<i>colunas</i>	Número de colunas da área de cobertura

Returns

Nef* Retorna uma lista de posições nefastas encontradas

4.2.2.4 InserirNefasto()

```
Nef * InserirNefasto (  
    Nef * listaNef,  
    int x,  
    int y,  
    int ant1,  
    int ant2,  
    Ant * lista)
```

Insere uma nova posição nefasta na lista.

Parameters

<i>listaNef</i>	Lista existente de nefastos
<i>x</i>	Coordenada x da posição nefasta
<i>y</i>	Coordenada y da posição nefasta
<i>ant1</i>	ID da primeira antena relacionada
<i>ant2</i>	ID da segunda antena relacionada
<i>lista</i>	Lista de antenas para validação

Returns

Lista atualizada de nefastos

Verifica se a posição já existe na lista antes de inserir uma nova posição nefasta.

Parameters

<i>listaNef</i>	Lista de posições nefastas existente
<i>x</i>	Coordenada x da posição nefasta
<i>y</i>	Coordenada y da posição nefasta
<i>ant1</i>	ID da primeira antena relacionada
<i>ant2</i>	ID da segunda antena relacionada
<i>lista</i>	Lista de antenas para validação

Returns

Nef* Retorna a lista atualizada de posições nefastas

4.3 Funções de Gestão de Memória

Funções para libertação de memória alocada.

Functions

- int `FreeLista` (`Ant` *`lista`)
Liberta memória alocada para lista de antenas.
- int `FreeListaNef` (`Nef` *`listaNef`)
Liberta memória alocada para lista de nefastos.
- int `FreeAntNef` (`AntNef` *`listaAntNef`)
Liberta memória alocada para ambas as listas.

4.3.1 Detailed Description

Funções para libertação de memória alocada.

4.3.2 Function Documentation

4.3.2.1 FreeAntNef()

```
int FreeAntNef (  
    AntNef * listaAntNef)
```

Liberta memória alocada para ambas as listas.

Parameters

<code>lista</code>	Lista de antenas a ser liberada
<code>listaNef</code>	Lista de nefastos a ser liberada

Returns

1 em caso de sucesso

Liberta memória alocada para ambas as listas.

Parameters

<code>listaAntNef</code>	Estrutura <code>AntNef</code> a libertar
--------------------------	--

Returns

int Retorna 1 em caso de sucesso

4.3.2.2 FreeLista()

```
int FreeLista (  
    Ant * lista)
```

Liberta memória alocada para lista de antenas.

Parameters

<i>lista</i>	Lista a ser liberada
--------------	----------------------

Returns

1 em caso de sucesso

Liberta memória alocada para lista de antenas.

Parameters

<i>lista</i>	Lista de antenas a libertar
--------------	-----------------------------

Returns

int Retorna 1 em caso de sucesso

4.3.2.3 FreeListaNef()

```
int FreeListaNef (  
    Nef * listaNef)
```

Liberta memória alocada para lista de nefastos.

Parameters

<i>listaNef</i>	Lista a ser liberada
-----------------	----------------------

Returns

1 em caso de sucesso

Liberta memória alocada para lista de nefastos.

Parameters

<i>listaNef</i>	Lista de posições nefastas a libertar
-----------------	---------------------------------------

Returns

int Retorna 1 em caso de sucesso

Chapter 5

Data Structure Documentation

5.1 Ant Struct Reference

Estrutura que representa uma antena.

```
#include <funcao.h>
```

Data Fields

- char `freqAntena`
- int `id`
- int `x`
- int `y`
- struct `Ant` * `proxAntena`

5.1.1 Detailed Description

Estrutura que representa uma antena.

Armazena informações sobre uma antena incluindo sua frequência, coordenadas de posição e identificador único.

5.1.2 Field Documentation

5.1.2.1 `freqAntena`

```
char freqAntena
```

Frequência da antena (carácter representativo).

5.1.2.2 `id`

```
int id
```

Identificador único da antena.

5.1.2.3 proxAntena

```
struct Ant * proxAntena
```

Ponteiro para a próxima antena na lista ligada.

5.1.2.4 x

```
int x
```

Coordenada horizontal (coluna) da antena.

5.1.2.5 y

```
int y
```

Coordenada vertical (linha) da antena.

The documentation for this struct was generated from the following files:

- projetoeda/[funcao.h](#)
- projetoedalib/[funcao.h](#)

5.2 AntNef Struct Reference

Estrutura que agrupa listas de antenas e posições nefastas.

```
#include <funcao.h>
```

Data Fields

- struct [Ant](#) * [lista](#)
- struct [Nef](#) * [listaNef](#)

5.2.1 Detailed Description

Estrutura que agrupa listas de antenas e posições nefastas.

Permite manipular conjuntamente as listas de antenas e suas posições nefastas relacionadas.

5.2.2 Field Documentation

5.2.2.1 lista

```
struct Ant * lista
```

Ponteiro para a lista de antenas.

5.2.2.2 listaNef

```
struct Nef * listaNef
```

Ponteiro para a lista de posições nefastas.

The documentation for this struct was generated from the following files:

- projetoeda/[funcao.h](#)
- projetoedalib/[funcao.h](#)

5.3 Nef Struct Reference

Estrutura que representa uma posição nefasta.

```
#include <funcao.h>
```

Data Fields

- int [x](#)
- int [y](#)
- int [antena1](#)
- int [antena2](#)
- struct [Nef](#) * [proxNef](#)

5.3.1 Detailed Description

Estrutura que representa uma posição nefasta.

Armazena coordenadas de posições onde podem ocorrer interferências entre antenas, incluindo referências às antenas relacionadas.

5.3.2 Field Documentation

5.3.2.1 antena1

```
int antena1
```

ID da primeira antena associada ao nefasto.

5.3.2.2 antena2

```
int antena2
```

ID da segunda antena associada ao nefasto.

5.3.2.3 proxNef

```
struct Nef * proxNef
```

Ponteiro para o próximo nefasto na lista ligada.

5.3.2.4 x

```
int x
```

Coordenada horizontal da posição nefasta.

5.3.2.5 y

```
int y
```

Coordenada vertical da posição nefasta.

The documentation for this struct was generated from the following files:

- projetoeda/[funcao.h](#)
- projetoedalib/[funcao.h](#)

Chapter 6

File Documentation

6.1 projetoeda/funcao.h File Reference

Definições de estruturas e protótipos de funções para manipulação de antenas e nefastos.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Data Structures

- struct [Ant](#)
Estrutura que representa uma antena.
- struct [Nef](#)
Estrutura que representa uma posição nefasta.
- struct [AntNef](#)
Estrutura que agrupa listas de antenas e posições nefastas.

Macros

- #define **MAX_LINHAS** 100
Número máximo de linhas na matriz.
- #define **MAX_COLUNAS** 100
Número máximo de colunas na matriz.

Typedefs

- typedef struct Ant **Ant**
- typedef struct Nef **Nef**
- typedef struct AntNef **AntNef**

Functions

- **Ant * InserirAntena** (Ant *lista, char freq, int x, int y, int id)
Cria e insere uma nova antena na lista.
- **Ant * LerLista** (const char *nomeFicheiro, const char *tipoFicheiro, int *linhas, int *colunas)
Lê um ficheiro e constrói a lista de antenas.
- int **ApresentarLista** (Ant *lista)
Imprime no ecrã a lista de antenas.
- int **ApresentarMatrizLista** (Ant *lista, int linhas, int colunas)
Exibe uma representação matricial das antenas.
- **AntNef * RemoverAntena** (AntNef *listaAntNef, int x, int y)
Remove uma antena da lista e atualiza posições nefastas.
- **AntNef * AdicionarAntena** (AntNef *listaAntNef, char freq, int x, int y, int linhas, int colunas)
Adiciona ou atualiza uma antena na lista.
- **Nef * InserirNefasto** (Nef *listaNef, int x, int y, int ant1, int ant2, Ant *lista)
Insere uma nova posição nefasta na lista.
- **Nef * EncontrarNefastos** (Ant *lista, int linhas, int colunas)
Calcula posições nefastas entre antenas.
- int **ApresentarListaNef** (Nef *listaNef)
Imprime no ecrã a lista de posições nefastas.
- int **ApresentarMatrizListaNef** (Ant *lista, int linhas, int colunas, Nef *listaNef)
Exibe matriz com antenas e posições nefastas.
- int **FreeLista** (Ant *lista)
Liberta memória alocada para lista de antenas.
- int **FreeListaNef** (Nef *listaNef)
Liberta memória alocada para lista de nefastos.
- int **FreeAntNef** (AntNef *listaAntNef)
Liberta memória alocada para ambas as listas.

6.1.1 Detailed Description

Definições de estruturas e protótipos de funções para manipulação de antenas e nefastos.

Author

Fábio Rafael Gomes Costa @contact a22997@alunos.ipca.pt @course Engenharia Sistemas Informáticos

Version

"Com correções pos defesa"

Date

04/04/2025

6.2 funcao.h

[Go to the documentation of this file.](#)

```

00001
00010
00011
00012 #pragma once
00013 #include <stdio.h>
00014 #include <stdlib.h>
00015 #include <string.h>
00016
00021 #define MAX_LINHAS 100
00022
00027 #define MAX_COLUNAS 100
00028
00036 typedef struct Ant {
00037     char freqAntena;
00038     int id;
00039     int x;
00040     int y;
00041     struct Ant* proxAntena;
00042 } Ant;
00043
00051 typedef struct Nef {
00052     int x;
00053     int y;
00054     int antena1;
00055     int antena2;
00056     struct Nef* proxNef;
00057 } Nef;
00058
00066 typedef struct AntNef {
00067     struct Ant* lista;
00068     struct Nef* listaNef;
00069 } AntNef;
00070
00071 /*-----*/
00072 /*          PROTÓTIPOS DE FUNÇÕES          */
00073 /*-----*/
00074
00080
00090 Ant* InserirAntena(Ant* lista, char freq, int x, int y, int id);
00091
00100 Ant* LerLista(const char* nomeFicheiro, const char* tipoFicheiro, int* linhas, int* colunas);
00101
00107 int ApresentarLista(Ant* lista);
00108
00116 int ApresentarMatrizLista(Ant* lista, int linhas, int colunas);
00117
00125 AntNef* RemoverAntena(AntNef* listaAntNef, int x, int y);
00126
00137 AntNef* AdicionarAntena(AntNef* listaAntNef, char freq, int x, int y, int linhas, int colunas);
00138 // Fim do grupo Antenas
00140
00146
00157 Nef* InserirNefasto(Nef* listaNef, int x, int y, int ant1, int ant2, Ant* lista);
00158
00166 Nef* EncontrarNefastos(Ant* lista, int linhas, int colunas);
00167
00173 int ApresentarListaNef(Nef* listaNef);
00174
00183 int ApresentarMatrizListaNef(Ant* lista, int linhas, int colunas, Nef* listaNef);
00184 // Fim do grupo Nefastos
00186
00192
00198 int FreeLista(Ant* lista);
00199
00205 int FreeListaNef(Nef* listaNef);
00212 int FreeAntNef(AntNef* listaAntNef); // Fim do grupo Memoria

```

6.3 projetoedalib/funcao.h File Reference

Definições de estruturas e protótipos de funções para manipulação de antenas e nefastos.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

```

Data Structures

- struct [Ant](#)
Estrutura que representa uma antena.
- struct [Nef](#)
Estrutura que representa uma posição nefasta.
- struct [AntNef](#)
Estrutura que agrupa listas de antenas e posições nefastas.

Macros

- #define **MAX_LINHAS** 100
Número máximo de linhas na matriz.
- #define **MAX_COLUNAS** 100
Número máximo de colunas na matriz.

Typedefs

- typedef struct Ant **Ant**
- typedef struct Nef **Nef**
- typedef struct AntNef **AntNef**

Functions

- [Ant](#) * [InserirAntena](#) ([Ant](#) *lista, char freq, int x, int y, int id)
Cria e insere uma nova antena na lista.
- [Ant](#) * [LerLista](#) (const char *nomeFicheiro, const char *tipoFicheiro, int *linhas, int *colunas)
Lê um ficheiro e constrói a lista de antenas.
- int [ApresentarLista](#) ([Ant](#) *lista)
Imprime no ecrã a lista de antenas.
- int [ApresentarMatrizLista](#) ([Ant](#) *lista, int linhas, int colunas)
Exibe uma representação matricial das antenas.
- [AntNef](#) * [RemoverAntena](#) ([AntNef](#) *listaAntNef, int x, int y)
Remove uma antena da lista e atualiza posições nefastas.
- [AntNef](#) * [AdicionarAntena](#) ([AntNef](#) *listaAntNef, char freq, int x, int y, int linhas, int colunas)
Adiciona ou atualiza uma antena na lista.
- [Nef](#) * [InserirNefasto](#) ([Nef](#) *listaNef, int x, int y, int ant1, int ant2, [Ant](#) *lista)
Insere uma nova posição nefasta na lista.
- [Nef](#) * [EncontrarNefastos](#) ([Ant](#) *lista, int linhas, int colunas)
Calcula posições nefastas entre antenas.
- int [ApresentarListaNef](#) ([Nef](#) *listaNef)
Imprime no ecrã a lista de posições nefastas.
- int [ApresentarMatrizListaNef](#) ([Ant](#) *lista, int linhas, int colunas, [Nef](#) *listaNef)
Exibe matriz com antenas e posições nefastas.
- int [FreeLista](#) ([Ant](#) *lista)
Liberta memória alocada para lista de antenas.
- int [FreeListaNef](#) ([Nef](#) *listaNef)
Liberta memória alocada para lista de nefastos.
- int [FreeAntNef](#) ([AntNef](#) *listaAntNef)
Liberta memória alocada para ambas as listas.

6.3.1 Detailed Description

Definições de estruturas e protótipos de funções para manipulação de antenas e nefastos.

Author

Fábio Rafael Gomes Costa @contact a22997@alunos.ipca.pt @course Engenharia Sistemas Informáticos

Version

"Com correções pos defesa"

Date

04/04/2025

6.4 funcao.h

[Go to the documentation of this file.](#)

```

00001
00010
00011
00012 #pragma once
00013 #include <stdio.h>
00014 #include <stdlib.h>
00015 #include <string.h>
00016
00021 #define MAX_LINHAS 100
00022
00027 #define MAX_COLUNAS 100
00028
00036 typedef struct Ant {
00037     char freqAntena;
00038     int id;
00039     int x;
00040     int y;
00041     struct Ant* proxAntena;
00042 } Ant;
00043
00051 typedef struct Nef {
00052     int x;
00053     int y;
00054     int antena1;
00055     int antena2;
00056     struct Nef* proxNef;
00057 } Nef;
00058
00066 typedef struct AntNef {
00067     struct Ant* lista;
00068     struct Nef* listaNef;
00069 } AntNef;
00070
00071 /*-----*/
00072 /*          PROTÓTIPOS DE FUNÇÕES          */
00073 /*-----*/
00074
00080
00090 Ant* InserirAntena(Ant* lista, char freq, int x, int y, int id);
00091
00100 Ant* LerLista(const char* nomeFicheiro, const char* tipoFicheiro, int* linhas, int* colunas);
00101
00107 int ApresentarLista(Ant* lista);
00108
00116 int ApresentarMatrizLista(Ant* lista, int linhas, int colunas);
00117
00125 AntNef* RemoverAntena(AntNef* listaAntNef, int x, int y);
00126
00137 AntNef* AdicionarAntena(AntNef* listaAntNef, char freq, int x, int y, int linhas, int colunas);
00138 // Fim do grupo Antenas

```

```

00140
00146
00157 Nef* InserirNefasto(Nef* listaNef, int x, int y, int ant1, int ant2, Ant* lista);
00158
00166 Nef* EncontrarNefastos(Ant* lista, int linhas, int colunas);
00167
00173 int ApresentarListaNef(Nef* listaNef);
00174
00183 int ApresentarMatrizListaNef(Ant* lista, int linhas, int colunas, Nef* listaNef);
00184 // Fim do grupo Nefastos
00186
00192
00198 int FreeLista(Ant* lista);
00199
00205 int FreeListaNef(Nef* listaNef);
00212 int FreeAntNef(AntNef* listaAntNef); // Fim do grupo Memoria

```

6.5 projetoeda/main.c File Reference

Programa principal para manipulação de antenas e detecção de nefastos.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "funcao.h"

```

Functions

- int `main` ()
Função principal do programa.

6.5.1 Detailed Description

Programa principal para manipulação de antenas e detecção de nefastos.

Author

Fábio Rafael Gomes Costa @contact a22997@alunos.ipca.pt @course Engenharia Sistemas Informáticos

Version

"Com correções pos defesa"

Date

04/04/2025

6.5.2 Function Documentation

6.5.2.1 main()

```
int main ()
```

Função principal do programa.

Esta função demonstra o funcionamento do sistema de gestão de antenas e posições nefastas, incluindo operações de leitura, remoção e adição de antenas, além da detecção de interferências.

Returns

int Retorna 0 em caso de sucesso, 1 em caso de erro

6.6 projetoedalib/funcao.c File Reference

Implementação de funções para manipulação de antenas e nefastos.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "funcao.h"
```

Functions

- [Ant * LerLista](#) (const char *nomeFicheiro, const char *tipoFicheiro, int *linhas, int *colunas)
Lê um ficheiro e cria uma lista de antenas.
- [Ant * InserirAntena](#) ([Ant](#) *lista, char freq, int x, int y, int id)
Insere uma nova antena na lista.
- int [ApresentarLista](#) ([Ant](#) *lista)
Apresenta no ecrã a lista de antenas.
- int [ApresentarMatrizLista](#) ([Ant](#) *lista, int linhas, int colunas)
Apresenta uma matriz representando as antenas e espaços vazios.
- [Nef * EncontrarNefastos](#) ([Ant](#) *lista, int linhas, int colunas)
Encontra posições nefastas entre antenas.
- [Nef * InserirNefasto](#) ([Nef](#) *listaNef, int x, int y, int ant1, int ant2, [Ant](#) *lista)
Insere uma nova posição nefasta na lista.
- int [ApresentarListaNef](#) ([Nef](#) *listaNef)
Apresenta no ecrã a lista de posições nefastas.
- int [ApresentarMatrizListaNef](#) ([Ant](#) *lista, int linhas, int colunas, [Nef](#) *listaNef)
Apresenta matriz com antenas e posições nefastas.
- [AntNef * RemoverAntena](#) ([AntNef](#) *listaAntNef, int x, int y)
Remove uma antena da lista e suas posições nefastas relacionadas.
- [AntNef * AdicionarAntena](#) ([AntNef](#) *listaAntNef, char freq, int x, int y, int linhas, int colunas)
Adiciona ou atualiza uma antena e recalcula posições nefastas.
- int [FreeLista](#) ([Ant](#) *lista)
Liberta a memória alocada para a lista de antenas.
- int [FreeListaNef](#) ([Nef](#) *listaNef)
Liberta a memória alocada para a lista de posições nefastas.
- int [FreeAntNef](#) ([AntNef](#) *listaAntNef)
Liberta a memória alocada para a estrutura [AntNef](#).

6.6.1 Detailed Description

Implementação de funções para manipulação de antenas e nefastos.

Author

Fábio Rafael Gomes Costa @contact a22997@alunos.ipca.pt @course Engenharia Sistemas Informáticos

Version

"Com correções pos defesa"

Date

04/04/2025

Index

- AdicionarAntena
 - Funções de Manipulação de Antenas, 7
- Ant, 17
 - freqAntena, 17
 - id, 17
 - proxAntena, 17
 - x, 18
 - y, 18
- antena1
 - Nef, 19
- antena2
 - Nef, 19
- AntNef, 18
 - lista, 18
 - listaNef, 18
- ApresentarLista
 - Funções de Manipulação de Antenas, 8
- ApresentarListaNef
 - Funções de Manipulação de Nefastos, 12
- ApresentarMatrizLista
 - Funções de Manipulação de Antenas, 9
- ApresentarMatrizListaNef
 - Funções de Manipulação de Nefastos, 12
- EncontrarNefastos
 - Funções de Manipulação de Nefastos, 13
- FreeAntNef
 - Funções de Gestão de Memória, 15
- FreeLista
 - Funções de Gestão de Memória, 15
- FreeListaNef
 - Funções de Gestão de Memória, 16
- freqAntena
 - Ant, 17
- Funções de Gestão de Memória, 15
 - FreeAntNef, 15
 - FreeLista, 15
 - FreeListaNef, 16
- Funções de Manipulação de Antenas, 7
 - AdicionarAntena, 7
 - ApresentarLista, 8
 - ApresentarMatrizLista, 9
 - InserirAntena, 9
 - LerLista, 10
 - RemoverAntena, 11
- Funções de Manipulação de Nefastos, 11
 - ApresentarListaNef, 12
 - ApresentarMatrizListaNef, 12
 - EncontrarNefastos, 13
- InserirNefasto, 14
- id
 - Ant, 17
- InserirAntena
 - Funções de Manipulação de Antenas, 9
- InserirNefasto
 - Funções de Manipulação de Nefastos, 14
- LerLista
 - Funções de Manipulação de Antenas, 10
- lista
 - AntNef, 18
- listaNef
 - AntNef, 18
- main
 - main.c, 27
- main.c
 - main, 27
- Nef, 19
 - antena1, 19
 - antena2, 19
 - proxNef, 19
 - x, 20
 - y, 20
- projetoeda/funcao.h, 21, 23
- projetoeda/main.c, 26
- projetoedalib/funcao.c, 27
- projetoedalib/funcao.h, 23, 25
- proxAntena
 - Ant, 17
- proxNef
 - Nef, 19
- RemoverAntena
 - Funções de Manipulação de Antenas, 11
- x
 - Ant, 18
 - Nef, 20
- y
 - Ant, 18
 - Nef, 20