

## Projeto de EDA

Gerado por Doxygen 1.9.1



<b>1 Índice dos componentes</b>	<b>1</b>
1.1 Lista de componentes . . . . .	1
<b>2 Índice dos ficheiros</b>	<b>3</b>
2.1 Lista de ficheiros . . . . .	3
<b>3 Documentação da classe</b>	<b>5</b>
3.1 Referência à estrutura Antena . . . . .	5
3.1.1 Descrição detalhada . . . . .	5
3.2 Referência à estrutura AntenaF . . . . .	6
3.2.1 Descrição detalhada . . . . .	6
3.3 Referência à estrutura EfeitoNefasto . . . . .	6
3.3.1 Descrição detalhada . . . . .	6
<b>4 Documentação do ficheiro</b>	<b>7</b>
4.1 Referência ao ficheiro funcoes.c . . . . .	7
4.1.1 Descrição detalhada . . . . .	8
4.1.2 Documentação das funções . . . . .	8
4.1.2.1 CalcularEfeito() . . . . .	8
4.1.2.2 CarregarFicheiro() . . . . .	9
4.1.2.3 CriarAntena() . . . . .	9
4.1.2.4 CriarEfeitoNefasto() . . . . .	10
4.1.2.5 DestroiLista() . . . . .	10
4.1.2.6 GravarFicheiroBin() . . . . .	11
4.1.2.7 InserirAntena() . . . . .	11
4.1.2.8 InserirEfeitoNefasto() . . . . .	12
4.1.2.9 LerFicheiroBin() . . . . .	12
4.1.2.10 RemoverAntena() . . . . .	13
4.1.2.11 RemoverAntenasComConflito() . . . . .	13
4.2 Referência ao ficheiro funcoes.h . . . . .	14
4.2.1 Descrição detalhada . . . . .	16
4.2.2 Documentação das funções . . . . .	16
4.2.2.1 CalcularEfeito() . . . . .	16
4.2.2.2 CarregarFicheiro() . . . . .	17
4.2.2.3 CriarAntena() . . . . .	17
4.2.2.4 CriarEfeitoNefasto() . . . . .	18
4.2.2.5 DestroiLista() . . . . .	19
4.2.2.6 GravarFicheiroBin() . . . . .	20
4.2.2.7 InserirAntena() . . . . .	20
4.2.2.8 InserirEfeitoNefasto() . . . . .	21
4.2.2.9 LerFicheiroBin() . . . . .	22
4.2.2.10 RemoverAntena() . . . . .	23
4.2.2.11 RemoverAntenasComConflito() . . . . .	23

---

4.3 Referência ao ficheiro main.c . . . . .	24
4.3.1 Descrição detalhada . . . . .	25
4.3.2 Documentação das funções . . . . .	26
4.3.2.1 main() . . . . .	26
4.4 Referência ao ficheiro matriz.c . . . . .	26
4.4.1 Descrição detalhada . . . . .	27
4.4.2 Documentação das funções . . . . .	28
4.4.2.1 CalcularDimensaoAntenas() . . . . .	28
4.4.2.2 CalcularDimensaoAntenasEEfeitos() . . . . .	28
4.4.2.3 MostrarAntenas() . . . . .	29
4.4.2.4 MostrarAntenasEEfeitos() . . . . .	29
4.5 Referência ao ficheiro matriz.h . . . . .	29
4.5.1 Descrição detalhada . . . . .	31
4.5.2 Documentação das funções . . . . .	31
4.5.2.1 CalcularDimensaoAntenas() . . . . .	31
4.5.2.2 CalcularDimensaoAntenasEEfeitos() . . . . .	32
4.5.2.3 MostrarAntenas() . . . . .	32
4.5.2.4 MostrarAntenasEEfeitos() . . . . .	33
<b>Índice</b>	<b>35</b>

# Capítulo 1

## Índice dos componentes

### 1.1 Lista de componentes

Lista de classes, estruturas, uniões e interfaces com uma breve descrição:

<a href="#">Antena</a>	Esta estrutura é usada numa lista ligada para armazenar informações sobre antenas Como, a frequência, as coordenadas na matriz e um ponteiro para a próxima antena na lista . . . . .	5
<a href="#">AntenaF</a>	Estrutura usada para armazenar uma antena, com a frequência (representada por uma letra) e as suas coordenadas na matriz num ficheiro . . . . .	6
<a href="#">EfeitoNefasto</a>	Esta estrutura é utilizada numa lista ligada para armazenar as coordenadas de pontos da matriz onde há efeitos nefastos assim como a ligação para o próximo elemento da lista Assim, representa um efeito nefasto na matriz . . . . .	6



## Capítulo 2

# Índice dos ficheiros

### 2.1 Lista de ficheiros

Lista de todos os ficheiros documentados com uma breve descrição:

<a href="#">funcoes.c</a>	Implementação de todas as funções . . . . .	7
<a href="#">funcoes.h</a>	Define as funções e estruturas . . . . .	14
<a href="#">main.c</a>	Programa principal que testa as funções . . . . .	24
<a href="#">matriz.c</a>	Implementação das funções para mostrar as matrizes no ecrã . . . . .	26
<a href="#">matriz.h</a>	Define as funções para apresentar as matrizes no ecrã . . . . .	29





## Capítulo 3

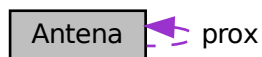
# Documentação da classe

### 3.1 Referência à estrutura Antena

Esta estrutura é usada numa lista ligada para armazenar informações sobre antenas. Como, a frequência, as coordenadas na matriz e um ponteiro para a próxima antena na lista.

```
#include <funcoes.h>
```

Diagrama de colaboração para Antena:



#### Atributos Públicos

- char **freq**
- int **x**
- int **y**
- struct [Antena](#) \* **prox**

#### 3.1.1 Descrição detalhada

Esta estrutura é usada numa lista ligada para armazenar informações sobre antenas. Como, a frequência, as coordenadas na matriz e um ponteiro para a próxima antena na lista.

A documentação para esta estrutura foi gerada a partir do seguinte ficheiro:

- [funcoes.h](#)

## 3.2 Referência à estrutura AntenaF

Estrutura usada para armazenar uma antena, com a frequência (representada por uma letra) e as suas coordenadas na matriz num ficheiro.

```
#include <funcoes.h>
```

### Atributos Públicos

- char **freq**
- int **x**
- int **y**

#### 3.2.1 Descrição detalhada

Estrutura usada para armazenar uma antena, com a frequência (representada por uma letra) e as suas coordenadas na matriz num ficheiro.

A documentação para esta estrutura foi gerada a partir do seguinte ficheiro:

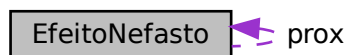
- [funcoes.h](#)

## 3.3 Referência à estrutura EfeitoNefasto

Esta estrutura é utilizada numa lista ligada para armazenar as coordenadas de pontos da matriz onde há efeitos nefastos assim como a ligação para o próximo elemento da lista Assim, representa um efeito nefasto na matriz.

```
#include <funcoes.h>
```

Diagrama de colaboração para EfeitoNefasto:



### Atributos Públicos

- int **x**
- int **y**
- struct [EfeitoNefasto](#) \* **prox**

#### 3.3.1 Descrição detalhada

Esta estrutura é utilizada numa lista ligada para armazenar as coordenadas de pontos da matriz onde há efeitos nefastos assim como a ligação para o próximo elemento da lista Assim, representa um efeito nefasto na matriz.

A documentação para esta estrutura foi gerada a partir do seguinte ficheiro:

- [funcoes.h](#)

## Capítulo 4

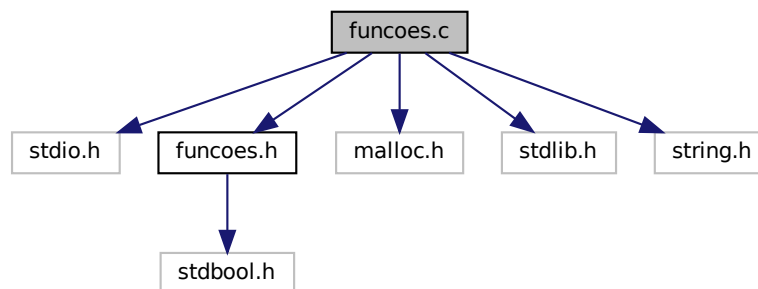
# Documentação do ficheiro

### 4.1 Referência ao ficheiro funcoes.c

Implementação de todas as funções.

```
#include <stdio.h>
#include "funcoes.h"
#include <malloc.h>
#include <stdlib.h>
#include <string.h>
```

Diagrama de dependências de inclusão para funcoes.c:



### Funções

- **Antena \* CriarAntena** (char freq, int x, int y, bool \*validar)  
*Cria uma nova antena e aloca na memória.*
- **Antena \* InserirAntena** (Antena \*inicio, Antena \*nova, bool \*validar)  
*Insera uma antena nova na lista ligada, ordenada corretamente pelas coordenadas (x, y)*
- **Antena \* RemoverAntena** (Antena \*inicio, int x, int y, bool \*validar)  
*Remove uma antena da lista ligada com base nas coordenadas (x, y)*
- **Antena \* DestroiLista** (Antena \*inicio, bool \*validar)

- Liberta e destrói a memória da lista ligada das antenas.*
- **EfeitoNefasto** \* **CriarEfeitoNefasto** (int x, int y, bool \*validar)  
*Função para criar efeitos nefastos e alocar na memória com os parâmetros especificados.*
- **EfeitoNefasto** \* **InserirEfeitoNefasto** (**EfeitoNefasto** \*inicio, **EfeitoNefasto** \*n, bool \*validar)  
*Insere um novo efeito nefasto numa lista ligada de efeitos, caso ainda não exista um efeito nessa posição.*
- **EfeitoNefasto** \* **CalcularEfeito** (**Antena** \*inicio, bool \*validar)  
*Calcula o efeito nefasto entre o sinal das antenas com a mesma frequência e insere os efeitos nefastos nas devidas posições.*
- **Antena** \* **RemoverAntenasComConflito** (**EfeitoNefasto** \*Lista\_Efeitos, **Antena** \*inicio, bool \*validar)  
*Remove as antenas que estejam nas posições onde correspondem efeitos nefastos.*
- **Antena** \* **CarregarFicheiro** (char \*nomeficheiro, bool \*validar)  
*Carrega apenas as antenas a partir de um ficheiro e cria uma lista ligada com a representação da cidade.*
- bool **GravarFicheiroBin** (char \*nomeficheiro, **Antena** \*inicio)  
*Grava a lista de antenas num ficheiro binário.*
- **Antena** \* **LerFicheiroBin** (char \*nomeficheiro, bool \*validar)  
*Lê a lista de antenas a partir do ficheiro binário.*

### 4.1.1 Descrição detalhada

Implementação de todas as funções.

Autor

David Faria ( [a31517@alunos.ipca.pt](mailto:a31517@alunos.ipca.pt) )

Versão

0.1

Data

2025-03-19

Copyright

Copyright (c) 2025

### 4.1.2 Documentação das funções

#### 4.1.2.1 CalcularEfeito()

```
EfeitoNefasto* CalcularEfeito (
    Antena * inicio,
    bool * validar )
```

Calcula o efeito nefasto entre o sinal das antenas com a mesma frequência e insere os efeitos nefastos nas devidas posições.

Calcula os efeitos nefastos gerados pelas antenas e devolve uma lista com esses efeitos.

Esta função percorre todas as antenas da lista e compara as antenas com a mesma frequência. Quando encontra antenas que estão a uma distância até 2 unidades numa direção horizontal, vertical ou diagonal, é gerado um efeito nefasto nas posições ao lado dessas antenas. Para cada par de antenas que estão a uma distância de 2 unidades, a função insere efeitos nefastos antes e depois das antenas nas direções horizontal, vertical ou diagonal. A função guarda os efeitos nefastos numa lista ligada e os efeitos são inseridos nas posições de interferência de forma ordenada. A interferência é apenas entre antenas com a mesma frequência.

## Parâmetros

<i>inicio</i>	Ponteiro para o início da lista de antenas
<i>validar</i>	Ponteiro para variável booleana que indica se o processo foi bem-sucedido ou não

## Retorna

EfeitoNefasto\* Retorna a lista de efeitos nefastos após inserir os mesmos

## 4.1.2.2 CarregarFicheiro()

```
Antena* CarregarFicheiro (
    char * nomeficheiro,
    bool * validar )
```

Carrega apenas as antenas a partir de um ficheiro e cria uma lista ligada com a representação da cidade.

Carrega as antenas a partir de um ficheiro e as insere numa lista ligada.

Esta função lê um ficheiro de texto que contém uma matriz que representa uma cidade com antenas. Apenas as antenas (!='.') são copiadas para a lista ligada, enquanto os pontos '.' representam posições vazias. A função chama 'InserirAntena()' para adicionar as antenas à lista.

## Parâmetros

<i>nomeficheiro</i>	Nome do ficheiro a ser lido
<i>validar</i>	Ponteiro para a variável booleana que indica se a operação foi bem-sucedida

## Retorna

Retorna o ponteiro para a primeira antena da lista ligada, ou NULL em caso de falha.

## 4.1.2.3 CriarAntena()

```
Antena* CriarAntena (
    char freq,
    int x,
    int y,
    bool * validar )
```

Cria uma nova antena e aloca na memória.

Função para criar antena e alocar a mesma na memória.

Esta função aloca dinamicamente memória para uma nova antena, inicializa os seus campos com os valores fornecidos e retorna um ponteiro para a antena criada. Caso a alocação de memória falhe, o ponteiro *validar* é definido como falso.

**Parâmetros**

<i>freq</i>	Letra que representa a frequência da antena
<i>x</i>	Coordenada X da antena na matriz
<i>y</i>	Coordenada Y da antena na matriz
<i>validar</i>	Ponteiro para uma variável booleana que será definido como verdadeiro se a criação for bem-sucedida, ou falso caso contrário

**Retorna**

Ponteiro para a antena criada ou NULL em caso de falha na alocação

**4.1.2.4 CriarEfeitoNefasto()**

```
EfeitoNefasto* CriarEfeitoNefasto (
    int x,
    int y,
    bool * validar )
```

Função para criar efeitos nefastos e alocar na memória com os parâmetros especificados.

Cria um novo efeito nefasto e devolve um ponteiro para ele.

Esta função aloca dinamicamente memória para um novo efeito da estrutura 'EfeitoNefasto' Coloca os respetivos valores nas coordenadas (x, y)

**Parâmetros**

<i>x</i>	Coordenada X da localização com efeito nefasto
<i>y</i>	Coordenada Y da localização com efeito nefasto
<i>validar</i>	Ponteiro para uma variável booleana que será definido como true se a alocação for bem-sucedida, ou false caso contrário

**Retorna**

EfeitoNefasto\* Retorna o ponteiro para o novo efeito, ou NULL caso a alocação na memória falhe

**4.1.2.5 DestroiLista()**

```
Antena* DestroiLista (
    Antena * inicio,
    bool * validar )
```

Liberta e destrói a memória da lista ligada das antenas.

Liberta a memória de todas as antenas na lista ligada.

Esta função percorre a lista ligada de antenas toda e liberta a memória de cada antena

## Parâmetros

<i>inicio</i>	Ponteiro para o início da lista ligada de antenas
<i>validar</i>	Ponteiro para uma variável booleana que será definida como true se a lista for destruída corretamente, ou false caso contrário

## Retorna

Antena\* Retorna NULL após destruir a lista, o que indica que a memória foi libertada e a lista destruída corretamente

## 4.1.2.6 GravarFicheiroBin()

```
bool GravarFicheiroBin (
    char * nomeficheiro,
    Antena * inicio )
```

Grava a lista de antenas num ficheiro binário.

Grava as antenas numa lista ligada para um ficheiro binário.

Esta função percorre a lista ligada de antenas e grava as antenas num ficheiro binário

## Parâmetros

<i>nomeficheiro</i>	Nome do ficheiro onde as antenas serão guardadas
<i>inicio</i>	Ponteiro para a lista ligada de antenas a ser gravada

## Retorna

true Se a gravação das antenas no ficheiro ocorrer corretamente  
false Caso ocorra erro ao abrir o ficheiro

## 4.1.2.7 InserirAntena()

```
Antena* InserirAntena (
    Antena * inicio,
    Antena * nova,
    bool * validar )
```

Inserir uma antena nova na lista ligada, ordenada corretamente pelas coordenadas (x, y)

Inserir uma nova antena no início da lista ligada de antenas.

Esta função insere uma antena na posição correta dentro da lista ligada, mantendo a ordem crescente das coordenadas (y, x). A ordenação é feita primeiramente pelo valor de 'y' (linha), e, caso 'y' seja igual, a ordenação é feita pelo valor de 'x' (coluna). Se já existir uma antena nas mesmas coordenadas, a nova antena não será adicionada, e a memória alocada para ela será libertada.

**Parâmetros**

<i>inicio</i>	Ponteiro para o início da lista ligada de antenas
<i>nova</i>	Ponteiro para a nova antena a ser inserida
<i>validar</i>	Ponteiro para uma variável booleana que indica se a inserção foi bem-sucedida

**Retorna**

Retorna a lista de antenas atualizada

**4.1.2.8 InserirEfeitoNefasto()**

```
EfeitoNefasto* InserirEfeitoNefasto (
    EfeitoNefasto * inicio,
    EfeitoNefasto * n,
    bool * validar )
```

Insere um novo efeito nefasto numa lista ligada de efeitos, caso ainda não exista um efeito nessa posição.

Insere um novo efeito nefasto na lista ligada.

Esta função cria um novo efeito nefasto e insere o efeito na lista de forma ordenada (pelas coordenadas (x, y)), a menos que já exista um na posição especificada. Se o efeito já estiver na posição fornecida (com as mesmas coordenadas 'x' e 'y'), o novo efeito não será inserido e a lista será retornada sem modificações. Caso a lista esteja vazia, o novo efeito será o primeiro da lista.

**Parâmetros**

<i>inicio</i>	Ponteiro para o início da lista de efeitos nefastos
<i>n</i>	Ponteiro para o novo efeito a ser inserido
<i>validar</i>	Ponteiro para uma variável booleana que será definida como true se a inserção for bem-sucedida ou false caso contrário

**Retorna**

EfeitoNefasto\* Retorna o início da lista de efeitos nefastos após a inserção ou sem modificações caso o efeito já exista

**4.1.2.9 LerFicheiroBin()**

```
Antena* LerFicheiroBin (
    char * nomeficheiro,
    bool * validar )
```

Lê a lista de antenas a partir do ficheiro binário.

Lê um ficheiro binário e carrega as antenas para uma lista ligada.

Esta função abre o ficheiro binário e lê os dados das antenas e reconstrói a lista ligada. As antenas são inseridas na lista através da função 'InserirAntena'.



## Parâmetros

<i>nomeficheiro</i>	Nome do ficheiro binário de onde as antenas serão lidas
<i>validar</i>	Um ponteiro para variável booleano de sucesso

## Retorna

Antena\* Retorna um ponteiro para a lista ligada reconstruída (antenas) ou 'NULL' caso o ficheiro não exista

## 4.1.2.10 RemoverAntena()

```
Antena* RemoverAntena (
    Antena * inicio,
    int x,
    int y,
    bool * validar )
```

Remove uma antena da lista ligada com base nas coordenadas (x, y)

Remove uma antena da lista ligada com base nas suas coordenadas.

Esta função percorre a lista ligada de antenas e caso encontre uma antena na posição (x, y) remove-a da lista e liberta a memória

## Parâmetros

<i>inicio</i>	Ponteiro para o primeiro elemento da lista ligada de antenas
<i>x</i>	Coordenada x da antena a ser removida
<i>y</i>	Coordenada y da antena a ser removida
<i>validar</i>	Ponteiro para uma variável booleana que será definido como true se a remoção for bem-sucedida, ou false caso contrário

## Retorna

Antena\* Retorna a lista ligada das antenas já atualizada

## 4.1.2.11 RemoverAntenasComConflito()

```
Antena* RemoverAntenasComConflito (
    EfeitoNefasto * Lista_Efeitos,
    Antena * inicio,
    bool * validar )
```

Remove as antenas que estejam nas posições onde correspondem efeitos nefastos.

Remove as antenas da lista que estão em locais de conflito com efeitos nefastos.

Esta função percorre e compara as posições da lista de efeitos nefastos e da lista de antenas Remove todas as antenas que estejam na mesma posição de um efeito nefasto ao chamar a função 'RemoverAntena'

**Parâmetros**

<i>Lista_Efeitos</i>	Ponteiro para a lista ligada de efeitos nefastos
<i>inicio</i>	Ponteiro para a lista ligada de antenas
<i>validar</i>	Ponteiro para a variável booleana que indica se houve alguma remoção

**Retorna**

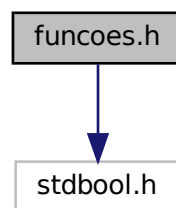
Antena\* Retorna o ponteiro atualizado para a lista de antenas após remover

## 4.2 Referência ao ficheiro funcoes.h

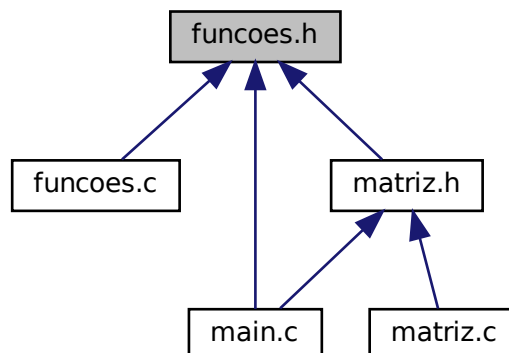
Define as funções e estruturas.

```
#include <stdbool.h>
```

Diagrama de dependências de inclusão para funcoes.h:



Este grafo mostra quais são os ficheiros que incluem directamente ou indirectamente este ficheiro:



## Componentes

- struct [Antena](#)  
*Esta estrutura é usada numa lista ligada para armazenar informações sobre antenas Como, a frequência, as coordenadas na matriz e um ponteiro para a próxima antena na lista.*
- struct [EfeitoNefasto](#)  
*Esta estrutura é utilizada numa lista ligada para armazenar as coordenadas de pontos da matriz onde há efeitos nefastos assim como a ligação para o próximo elemento da lista Assim, representa um efeito nefasto na matriz.*
- struct [AntenaF](#)  
*Estrutura usada para armazenar uma antena, com a frequência (representada por uma letra) e as suas coordenadas na matriz num ficheiro.*

## Macros

- #define **TAM\_MAX\_LINHA** 200

## Definições de tipos

- typedef struct [Antena](#) [Antena](#)  
*Esta estrutura é usada numa lista ligada para armazenar informações sobre antenas Como, a frequência, as coordenadas na matriz e um ponteiro para a próxima antena na lista.*
- typedef struct [EfeitoNefasto](#) [EfeitoNefasto](#)  
*Esta estrutura é utilizada numa lista ligada para armazenar as coordenadas de pontos da matriz onde há efeitos nefastos assim como a ligação para o próximo elemento da lista Assim, representa um efeito nefasto na matriz.*
- typedef struct [AntenaF](#) [AntenaF](#)  
*Estrutura usada para armazenar uma antena, com a frequência (representada por uma letra) e as suas coordenadas na matriz num ficheiro.*

## Funções

- [Antena](#) \* [CriarAntena](#) (char freq, int x, int y, bool \*validar)  
*Função para criar antena e alocar a mesma na memória.*
- [Antena](#) \* [InserirAntena](#) ([Antena](#) \*inicio, [Antena](#) \*nova, bool \*validar)  
*Insere uma nova antena no início da lista ligada de antenas.*
- [Antena](#) \* [RemoverAntena](#) ([Antena](#) \*inicio, int x, int y, bool \*validar)  
*Remove uma antena da lista ligada com base nas suas coordenadas.*
- [EfeitoNefasto](#) \* [CriarEfeitoNefasto](#) (int x, int y, bool \*validar)  
*Cria um novo efeito nefasto e devolve um ponteiro para ele.*
- [EfeitoNefasto](#) \* [InserirEfeitoNefasto](#) ([EfeitoNefasto](#) \*inicio, [EfeitoNefasto](#) \*n, bool \*validar)  
*Insere um novo efeito nefasto na lista ligada.*
- [EfeitoNefasto](#) \* [CalcularEfeito](#) ([Antena](#) \*inicio, bool \*validar)  
*Calcula os efeitos nefastos gerados pelas antenas e devolve uma lista com esses efeitos.*
- [Antena](#) \* [RemoverAntenasComConflito](#) ([EfeitoNefasto](#) \*Lista\_Efeitos, [Antena](#) \*inicio, bool \*validar)  
*Remove as antenas da lista que estão em locais de conflito com efeitos nefastos.*
- [Antena](#) \* [CarregarFicheiro](#) (char \*nomeficheiro, bool \*validar)  
*Carrega as antenas a partir de um ficheiro e as insere numa lista ligada.*
- bool [GravarFicheiroBin](#) (char \*nomeficheiro, [Antena](#) \*inicio)  
*Grava as antenas numa lista ligada para um ficheiro binário.*
- [Antena](#) \* [DestroiLista](#) ([Antena](#) \*inicio, bool \*validar)  
*Liberta a memória de todas as antenas na lista ligada.*
- [Antena](#) \* [LerFicheiroBin](#) (char \*nomeficheiro, bool \*validar)  
*Lê um ficheiro binário e carrega as antenas para uma lista ligada.*

### 4.2.1 Descrição detalhada

Define as funções e estruturas.

#### Autor

David Faria ( [a31517@alunos.ipca.pt](mailto:a31517@alunos.ipca.pt) )

#### Versão

0.1

#### Data

2025-03-18

#### Copyright

Copyright (c) 2025

### 4.2.2 Documentação das funções

#### 4.2.2.1 CalcularEfeito()

```
EfeitoNefasto* CalcularEfeito (
    Antena * inicio,
    bool * validar )
```

Calcula os efeitos nefastos gerados pelas antenas e devolve uma lista com esses efeitos.

#### Parâmetros

<i>inicio</i>	Ponteiro para o início da lista de antenas
<i>validar</i>	Ponteiro para uma variável booleana que indicará se o cálculo foi bem-sucedido

#### Retorna

EfeitoNefasto\* Ponteiro para o início da lista de efeitos nefastos

Calcula os efeitos nefastos gerados pelas antenas e devolve uma lista com esses efeitos.

Esta função percorre todas as antenas da lista e compara as antenas com a mesma frequência. Quando encontra antenas que estão a uma distância até 2 unidades numa direção horizontal, vertical ou diagonal, é gerado um efeito nefasto nas posições ao lado dessas antenas. Para cada par de antenas que estão a uma distância de 2 unidades, a função insere efeitos nefastos antes e depois das antenas nas direções horizontal, vertical ou diagonal. A função guarda os efeitos nefastos numa lista ligada e os efeitos são inseridos nas posições de interferência de forma ordenada. A interferência é apenas entre antenas com a mesma frequência.

## Parâmetros

<i>inicio</i>	Ponteiro para o início da lista de antenas
<i>validar</i>	Ponteiro para variável booleana que indica se o processo foi bem-sucedido ou não

## Retorna

EfeitoNefasto\* Retorna a lista de efeitos nefastos após inserir os mesmos

## 4.2.2.2 CarregarFicheiro()

```
Antena* CarregarFicheiro (
    char * nomeficheiro,
    bool * validar )
```

Carrega as antenas a partir de um ficheiro e as insere numa lista ligada.

## Parâmetros

<i>nomeficheiro</i>	Nome do ficheiro a ser lido
<i>validar</i>	Ponteiro para uma variável booleana que indicará se o carregamento foi bem-sucedido

## Retorna

Antena\* Ponteiro para o início da lista de antenas carregadas

Carrega as antenas a partir de um ficheiro e as insere numa lista ligada.

Esta função lê um ficheiro de texto que contém uma matriz que representa uma cidade com antenas. Apenas as antenas (!='.') são copiadas para a lista ligada, enquanto os pontos '.' representam posições vazias. A função chama 'InserirAntena()' para adicionar as antenas à lista.

## Parâmetros

<i>nomeficheiro</i>	Nome do ficheiro a ser lido
<i>validar</i>	Ponteiro para a variável booleana que indica se a operação foi bem-sucedida

## Retorna

Retorna o ponteiro para a primeira antena da lista ligada, ou NULL em caso de falha.

## 4.2.2.3 CriarAntena()

```
Antena* CriarAntena (
    char freq,
```

```

    int x,
    int y,
    bool * validar )

```

Função para criar antena e alocar a mesma na memória.

#### Parâmetros

<i>freq</i>	Letra que representa a frequência da antena
<i>x</i>	Coordenada X da antena na matriz
<i>y</i>	Coordenada Y da antena na matriz
<i>validar</i>	Ponteiro para uma variável booleana que será definida como verdadeiro se a criação for bem-sucedida, ou falso caso contrário

#### Retorna

Antena\* Ponteiro para a antena criada ou NULL em caso de falha

Função para criar antena e alocar a mesma na memória.

Esta função aloca dinamicamente memória para uma nova antena, inicializa os seus campos com os valores fornecidos e retorna um ponteiro para a antena criada. Caso a alocação de memória falhe, o ponteiro *validar* é definido como falso

#### Parâmetros

<i>freq</i>	Letra que representa a frequência da antena
<i>x</i>	Coordenada X da antena na matriz
<i>y</i>	Coordenada Y da antena na matriz
<i>validar</i>	Ponteiro para uma variável booleana que será definido como verdadeiro se a criação for bem-sucedida, ou falso caso contrário

#### Retorna

Ponteiro para a antena criada ou NULL em caso de falha na alocação

#### 4.2.2.4 CriarEfeitoNefasto()

```

EfeitoNefasto* CriarEfeitoNefasto (
    int x,
    int y,
    bool * validar )

```

Cria um novo efeito nefasto e devolve um ponteiro para ele.

#### Parâmetros

<i>x</i>	Coordenada X do efeito nefasto na matriz
<i>y</i>	Coordenada Y do efeito nefasto na matriz
<i>validar</i>	Ponteiro para uma variável booleana que será definida como verdadeiro se a criação for bem-sucedida, ou falso caso contrário

**Retorna**

EfeitoNefasto\* Ponteiro para o efeito nefasto criado ou NULL em caso de falha

Cria um novo efeito nefasto e devolve um ponteiro para ele.

Esta função aloca dinamicamente memória para um novo efeito da estrutura 'EfeitoNefasto' Coloca os respetivos valores nas coordenadas (x, y)

**Parâmetros**

<i>x</i>	Coordenada X da localização com efeito nefasto
<i>y</i>	Coordenada Y da localização com efeito nefasto
<i>validar</i>	Ponteiro para uma variável booleana que será definido como true se a alocação for bem-sucedida, ou false caso contrário

**Retorna**

EfeitoNefasto\* Retorna o ponteiro para o novo efeito, ou NULL caso a alocação na memória falhe

**4.2.2.5 DestroiLista()**

```
Antena* DestroiLista (
    Antena * inicio,
    bool * validar )
```

Liberta a memória de todas as antenas na lista ligada.

**Parâmetros**

<i>inicio</i>	Ponteiro para o início da lista de antenas a ser destruída
<i>validar</i>	Ponteiro para uma variável booleana que indicará se a destruição foi bem-sucedida

**Retorna**

Antena\* Ponteiro nulo após a libertação da memória de todas as antenas

Liberta a memória de todas as antenas na lista ligada.

Esta função percorre a lista ligada de antenas toda e liberta a memória de cada antena

**Parâmetros**

<i>inicio</i>	Ponteiro para o início da lista ligada de antenas
<i>validar</i>	Ponteiro para uma variável booleana que será definida como true se a lista for destruída corretamente, ou false caso contrário

**Retorna**

Antena\* Retorna NULL após destruir a lista, o que indica que a memória foi libertada e a lista destruída corretamente

**4.2.2.6 GravarFicheiroBin()**

```
bool GravarFicheiroBin (
    char * nomeficheiro,
    Antena * inicio )
```

Grava as antenas numa lista ligada para um ficheiro binário.

**Parâmetros**

<i>nomeficheiro</i>	Nome do ficheiro onde os dados das antenas serão gravados
<i>inicio</i>	Ponteiro para o início da lista de antenas a ser gravada

**Retorna**

true Se a gravação for bem-sucedida

false Se a gravação não for bem-sucedida

Grava as antenas numa lista ligada para um ficheiro binário.

Esta função percorre a lista ligada de antenas e grava as antenas num ficheiro binário

**Parâmetros**

<i>nomeficheiro</i>	Nome do ficheiro onde as antenas serão guardadas
<i>inicio</i>	Ponteiro para a lista ligada de antenas a ser gravada

**Retorna**

true Se a gravação das antenas no ficheiro ocorrer corretamente

false Caso ocorra erro ao abrir o ficheiro

**4.2.2.7 InserirAntena()**

```
Antena* InserirAntena (
    Antena * inicio,
    Antena * nova,
    bool * validar )
```

Insere uma nova antena no início da lista ligada de antenas.



## Parâmetros

<i>inicio</i>	Ponteiro para o início atual da lista de antenas
<i>nova</i>	Ponteiro para a nova antena a ser inserida
<i>validar</i>	Ponteiro para uma variável booleana que será definida como verdadeiro se a inserção for bem-sucedida, ou falso caso contrário

## Retorna

Antena\* Ponteiro para o novo início da lista

Insere uma nova antena no início da lista ligada de antenas.

Esta função insere uma antena na posição correta dentro da lista ligada, mantendo a ordem crescente das coordenadas (y, x). A ordenação é feita primeiramente pelo valor de 'y' (linha), e, caso 'y' seja igual, a ordenação é feita pelo valor de 'x' (coluna). Se já existir uma antena nas mesmas coordenadas, a nova antena não será adicionada, e a memória alocada para ela será libertada.

## Parâmetros

<i>inicio</i>	Ponteiro para o início da lista ligada de antenas
<i>nova</i>	Ponteiro para a nova antena a ser inserida
<i>validar</i>	Ponteiro para uma variável booleana que indica se a inserção foi bem-sucedida

## Retorna

Retorna a lista de antenas atualizada

## 4.2.2.8 InserirEfeitoNefasto()

```
EfeitoNefasto* InserirEfeitoNefasto (  
    EfeitoNefasto * inicio,  
    EfeitoNefasto * n,  
    bool * validar )
```

Insere um novo efeito nefasto na lista ligada.

## Parâmetros

<i>inicio</i>	Ponteiro para o início atual da lista de efeitos nefastos
<i>n</i>	Ponteiro para o novo efeito nefasto a ser inserido
<i>validar</i>	Ponteiro para uma variável booleana que indicará se a inserção foi bem-sucedida

## Retorna

EfeitoNefasto\* Ponteiro para o novo início da lista

Insere um novo efeito nefasto na lista ligada.

Esta função cria um novo efeito nefasto e insere o efeito na lista de forma ordenada (pelas coordenadas (x, y)), a menos que já exista um na posição especificada. Se o efeito já estiver na posição fornecida (com as mesmas coordenadas 'x' e 'y'), o novo efeito não será inserido e a lista será retornada sem modificações. Caso a lista esteja vazia, o novo efeito será o primeiro da lista.

#### Parâmetros

<i>inicio</i>	Ponteiro para o início da lista de efeitos nefastos
<i>n</i>	Ponteiro para o novo efeito a ser inserido
<i>validar</i>	Ponteiro para uma variável booleana que será definida como true se a inserção for bem-sucedida ou false caso contrário

#### Retorna

EfeitoNefasto\* Retorna o início da lista de efeitos nefastos após a inserção ou sem modificações caso o efeito já exista

#### 4.2.2.9 LerFicheiroBin()

```
Antena* LerFicheiroBin (
    char * nomeficheiro,
    bool * validar )
```

Lê um ficheiro binário e carrega as antenas para uma lista ligada.

#### Parâmetros

<i>nomeficheiro</i>	Nome do ficheiro binário a ser lido
<i>validar</i>	Ponteiro para uma variável booleana que indicará se a leitura foi bem-sucedida

#### Retorna

Antena\* Ponteiro para o início da lista de antenas carregadas

Lê um ficheiro binário e carrega as antenas para uma lista ligada.

Esta função abre o ficheiro binário e lê os dados das antenas e reconstrói a lista ligada. As antenas são inseridas na lista através da função 'InserirAntena'.

#### Parâmetros

<i>nomeficheiro</i>	Nome do ficheiro binário de onde as antenas serão lidas
<i>validar</i>	Um ponteiro para variável booleana de sucesso

**Retorna**

Antena\* Retorna um ponteiro para a lista ligada reconstruída (antenas) ou 'NULL' caso o ficheiro não exista

**4.2.2.10 RemoverAntena()**

```
Antena* RemoverAntena (
    Antena * inicio,
    int x,
    int y,
    bool * validar )
```

Remove uma antena da lista ligada com base nas suas coordenadas.

**Parâmetros**

<i>inicio</i>	Ponteiro para o início da lista de antenas
<i>x</i>	Coordenada X da antena a remover
<i>y</i>	Coordenada Y da antena a remover
<i>validar</i>	Ponteiro para uma variável booleana que indicará se a remoção foi bem-sucedida ou não

**Retorna**

Antena\* Ponteiro para o novo início da lista após a remoção

Remove uma antena da lista ligada com base nas suas coordenadas.

Esta função percorre a lista ligada de antenas e caso encontre uma antena na posição (x, y) remove-a da lista e liberta a memória

**Parâmetros**

<i>inicio</i>	Ponteiro para o primeiro elemento da lista ligada de antenas
<i>x</i>	Coordenada x da antena a ser removida
<i>y</i>	Coordenada y da antena a ser removida
<i>validar</i>	Ponteiro para uma variável booleana que será definido como true se a remoção for bem-sucedida, ou false caso contrário

**Retorna**

Antena\* Retorna a lista ligada das antenas já atualizada

**4.2.2.11 RemoverAntenasComConflito()**

```
Antena* RemoverAntenasComConflito (
    EfeitoNefasto * Lista_Efeitos,
```

```
Antena * inicio,  
bool * validar )
```

Remove as antenas da lista que estão em locais de conflito com efeitos nefastos.

#### Parâmetros

<i>Lista_Efeitos</i>	Ponteiro para o início da lista de efeitos nefastos
<i>inicio</i>	Ponteiro para o início da lista de antenas
<i>validar</i>	Ponteiro para uma variável booleana que indicará se a remoção foi bem-sucedida

#### Retorna

Antena\* Ponteiro para o novo início da lista de antenas após a remoção

Remove as antenas da lista que estão em locais de conflito com efeitos nefastos.

Esta função percorre e compara as posições da lista de efeitos nefastos e da lista de antenas Remove todas as antenas que estejam na mesma posição de um efeito nefasto ao chamar a função 'RemoverAntena'

#### Parâmetros

<i>Lista_Efeitos</i>	Ponteiro para a lista ligada de efeitos nefastos
<i>inicio</i>	Ponteiro para a lista ligada de antenas
<i>validar</i>	Ponteiro para a variável booleana que indica se houve alguma remoção

#### Retorna

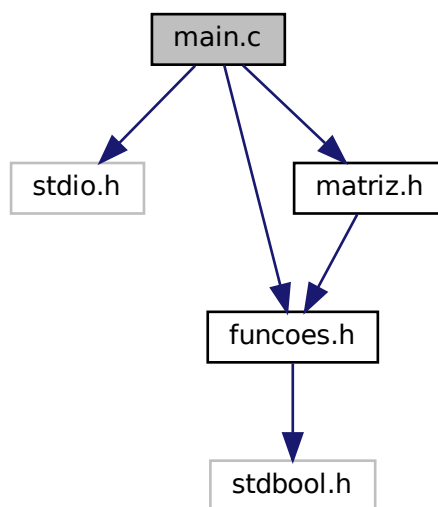
Antena\* Retorna o ponteiro atualizado para a lista de antenas após remover

## 4.3 Referência ao ficheiro main.c

Programa principal que testa as funções.

```
#include <stdio.h>  
#include "funcoes.h"  
#include "matriz.h"
```

Diagrama de dependências de inclusão para main.c:



## Funções

- int `main()`

*Executa as seguintes operações:*

### 4.3.1 Descrição detalhada

Programa principal que testa as funções.

#### Autor

David Faria ( [a31517@alunos.ipca.pt](mailto:a31517@alunos.ipca.pt) )

#### Versão

0.1

#### Data

2025-03-21

#### Copyright

Copyright (c) 2025

## 4.3.2 Documentação das funções

### 4.3.2.1 main()

```
int main ( )
```

Executa as seguintes operações:

1. Carrega as antenas a partir do ficheiro "cidade.txt"
2. Insere antenas em posições específicas
3. Remove antenas em determinadas posições
4. Calcula os efeitos nefastos das antenas e insere os mesmos
5. Remove as antenas em conflito de posições com efeito nefasto e atualiza estes
6. Grava a lista ligada das antenas num ficheiro binário
7. Destroi a lista ligada das antenas
8. Lê os dados das antenas do ficheiro binário e guarda-os na lista ligada

Retorna

\* int Retorna 0 para indicar que o programa foi executado com sucesso

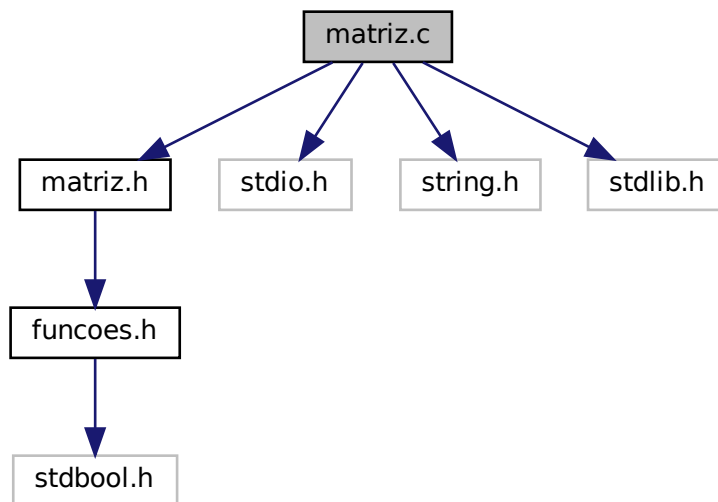
## 4.4 Referência ao ficheiro matriz.c

Implementação das funções para mostrar as matrizes no ecrã

```
#include "matriz.h"  
#include <stdio.h>  
#include <string.h>
```

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

Diagrama de dependências de inclusão para matriz.c:



## Funções

- void `CalcularDimensaoAntenas` (`Antena *inicio`, `int *linhas`, `int *colunas`)  
*Calcula as dimensões máximas (linhas e colunas) de uma matriz baseada nas coordenadas das antenas.*
- void `MostrarAntenas` (`Antena *inicio`)  
*Mostra as antenas na matriz e imprime no terminal.*
- void `CalcularDimensaoAntenasEEfeitos` (`Antena *inicio`, `EfeitoNefasto *efeitos`, `int *linhas`, `int *colunas`)  
*Calcula as dimensões (linhas e colunas) da matriz, considerando tanto as antenas como os efeitos nefastos.*
- void `MostrarAntenasEEfeitos` (`Antena *antenas`, `EfeitoNefasto *efeitos`)  
*Exibe as antenas e os efeitos nefastos numa matriz, representando as suas localizações.*

### 4.4.1 Descrição detalhada

Implementação das funções para mostrar as matrizes no ecrã

Autor

David Faria ( [a31517@alunos.ipca.pt](mailto:a31517@alunos.ipca.pt) )

Versão

0.1

Data

2025-05-01

Copyright

Copyright (c) 2025

## 4.4.2 Documentação das funções

### 4.4.2.1 CalcularDimensaoAntenas()

```
void CalcularDimensaoAntenas (
    Antena * inicio,
    int * linhas,
    int * colunas )
```

Calcula as dimensões máximas (linhas e colunas) de uma matriz baseada nas coordenadas das antenas.

Calcula as dimensões da matriz necessária para mostrar as antenas.

Esta função percorre a lista de antenas e encontra as coordenadas máximas (x, y), de modo a determinar a dimensão necessária para representar todas as antenas

#### Parâmetros

<i>inicio</i>	Ponteiro para o início da lista ligada de antenas
<i>linhas</i>	Ponteiro para armazenar o número máximo de linhas
<i>colunas</i>	Ponteiro para armazenar o número máximo de colunas

### 4.4.2.2 CalcularDimensaoAntenasEEfeitos()

```
void CalcularDimensaoAntenasEEfeitos (
    Antena * inicio,
    EfeitoNefasto * efeitos,
    int * linhas,
    int * colunas )
```

Calcula as dimensões (linhas e colunas) da matriz, considerando tanto as antenas como os efeitos nefastos.

Calcula as dimensões da matriz necessária para mostrar as antenas e os efeitos nefastos.

Esta função percorre a lista de antenas e efeitos nefastos para determinar as dimensões mínimas necessárias para a matriz que pode conter todas as antenas e efeitos, levando em consideração as suas coordenadas. As dimensões calculadas correspondem ao número total de linhas e colunas da matriz, ajustadas para cobrir todas as posições das antenas e efeitos.

#### Parâmetros

<i>inicio</i>	Ponteiro para a lista ligada das antenas
<i>efeitos</i>	Ponteiro para a lista ligada dos efeitos nefastos
<i>linhas</i>	Ponteiro para o número de linhas da matriz (será atualizado pela função)
<i>colunas</i>	Ponteiro para o número de colunas da matriz (será atualizado pela função)



#### 4.4.2.3 MostrarAntenas()

```
void MostrarAntenas (
    Antena * inicio )
```

Mostra as antenas na matriz e imprime no terminal.

Mostra as antenas presentes na lista ligada numa matriz.

Esta função calcula as dimensões da cidade (matriz) com base nas antenas, aloca dinamicamente uma matriz para armazenar a cidade, e exibe-a no terminal.

##### Parâmetros

<i>inicio</i>	Ponteiro para o início da lista ligada de antenas
---------------	---

#### 4.4.2.4 MostrarAntenasEEfeitos()

```
void MostrarAntenasEEfeitos (
    Antena * antenas,
    EfeitoNefasto * efeitos )
```

Exibe as antenas e os efeitos nefastos numa matriz, representando as suas localizações.

Mostra as antenas e os efeitos nefastos presentes nas listas ligadas.

Esta função calcula as dimensões da matriz necessária para acomodar tanto as antenas quanto os efeitos nefastos. A matriz resultante é preenchida com as posições das antenas (representadas pelo seu identificador de frequência) e os efeitos nefastos (representados pelo símbolo '#'). As antenas são colocadas primeiro, seguidas pelos efeitos.

As posições das antenas são inseridas nas coordenadas correspondentes na matriz, enquanto os efeitos são representados nas mesmas posições onde ocorrem, sobrepondo as antenas caso necessário.

##### Parâmetros

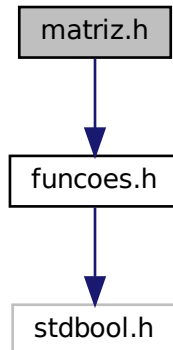
<i>antenas</i>	Ponteiro para a lista ligada de antenas a ser exibida.
<i>efeitos</i>	Ponteiro para a lista ligada de efeitos nefastos a ser exibidos.

## 4.5 Referência ao ficheiro matriz.h

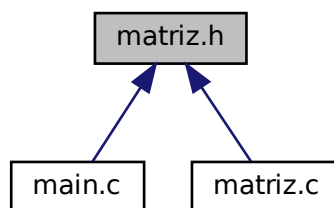
Define as funções para apresentar as matrizes no ecrã

```
#include "funcoes.h"
```

Diagrama de dependências de inclusão para matriz.h:



Este grafo mostra quais são os ficheiros que incluem directamente ou indirectamente este ficheiro:



## Funções

- void [CalcularDimensaoAntenas](#) ([Antena](#) \*inicio, int \*linhas, int \*colunas)  
*Calcula as dimensões da matriz necessária para mostrar as antenas.*
- void [MostrarAntenas](#) ([Antena](#) \*inicio)  
*Mostra as antenas presentes na lista ligada numa matriz.*
- void [CalcularDimensaoAntenasEEfeitos](#) ([Antena](#) \*inicio, [EfeitoNefasto](#) \*efeitos, int \*linhas, int \*colunas)  
*Calcula as dimensões da matriz necessária para mostrar as antenas e os efeitos nefastos.*
- void [MostrarAntenasEEfeitos](#) ([Antena](#) \*antenas, [EfeitoNefasto](#) \*efeitos)  
*Mostra as antenas e os efeitos nefastos presentes nas listas ligadas.*

### 4.5.1 Descrição detalhada

Define as funções para apresentar as matrizes no ecrã

#### Autor

David Faria ( [a31517@alunos.ipca.pt](mailto:a31517@alunos.ipca.pt) )

#### Versão

0.1

#### Data

2025-05-01

#### Copyright

Copyright (c) 2025

### 4.5.2 Documentação das funções

#### 4.5.2.1 CalcularDimensaoAntenas()

```
void CalcularDimensaoAntenas (
    Antena * inicio,
    int * linhas,
    int * colunas )
```

Calcula as dimensões da matriz necessária para mostrar as antenas.

#### Parâmetros

<i>inicio</i>	Ponteiro para o início da lista de antenas
<i>linhas</i>	Ponteiro para a variável que irá armazenar o número de linhas da matriz
<i>colunas</i>	Ponteiro para a variável que irá armazenar o número de colunas da matriz

Calcula as dimensões da matriz necessária para mostrar as antenas.

Esta função percorre a lista de antenas e encontra as coordenadas máximas (x, y), de modo a determinar a dimensão necessária para representar todas as antenas

#### Parâmetros

<i>inicio</i>	Ponteiro para o início da lista ligada de antenas
<i>linhas</i>	Ponteiro para armazenar o número máximo de linhas
<i>colunas</i>	Ponteiro para armazenar o número máximo de colunas

#### 4.5.2.2 CalcularDimensaoAntenasEEfeitos()

```
void CalcularDimensaoAntenasEEfeitos (
    Antena * inicio,
    EfeitoNefasto * efeitos,
    int * linhas,
    int * colunas )
```

Calcula as dimensões da matriz necessária para mostrar as antenas e os efeitos nefastos.

##### Parâmetros

<i>inicio</i>	Ponteiro para o início da lista de antenas
<i>efeitos</i>	Ponteiro para o início da lista de efeitos nefastos
<i>linhas</i>	Ponteiro para a variável que irá armazenar o número de linhas da matriz
<i>colunas</i>	Ponteiro para a variável que irá armazenar o número de colunas da matriz

Calcula as dimensões da matriz necessária para mostrar as antenas e os efeitos nefastos.

Esta função percorre a lista de antenas e efeitos nefastos para determinar as dimensões mínimas necessárias para a matriz que pode conter todas as antenas e efeitos, levando em consideração as suas coordenadas. As dimensões calculadas correspondem ao número total de linhas e colunas da matriz, ajustadas para cobrir todas as posições das antenas e efeitos.

##### Parâmetros

<i>inicio</i>	Ponteiro para a lista ligada das antenas
<i>efeitos</i>	Ponteiro para a lista ligada dos efeitos nefastos
<i>linhas</i>	Ponteiro para o número de linhas da matriz (será atualizado pela função)
<i>colunas</i>	Ponteiro para o número de colunas da matriz (será atualizado pela função)

#### 4.5.2.3 MostrarAntenas()

```
void MostrarAntenas (
    Antena * inicio )
```

Mostra as antenas presentes na lista ligada numa matriz.

##### Parâmetros

<i>inicio</i>	Ponteiro para o início da lista de antenas a ser exibida
---------------	--

Mostra as antenas presentes na lista ligada numa matriz.

Esta função calcula as dimensões da cidade (matriz) com base nas antenas, aloca dinamicamente uma matriz para armazenar a cidade, e exibe-a no terminal.

## Parâmetros

<i>inicio</i>	Ponteiro para o início da lista ligada de antenas
---------------	---

## 4.5.2.4 MostrarAntenasEEfeitos()

```
void MostrarAntenasEEfeitos (
    Antena * antenas,
    EfeitoNefasto * efeitos )
```

Mostra as antenas e os efeitos nefastos presentes nas listas ligadas.

## Parâmetros

<i>antenas</i>	Ponteiro para o início da lista de antenas a ser exibida
<i>efeitos</i>	Ponteiro para o início da lista de efeitos nefastos a ser exibida

Mostra as antenas e os efeitos nefastos presentes nas listas ligadas.

Esta função calcula as dimensões da matriz necessária para acomodar tanto as antenas quanto os efeitos nefastos. A matriz resultante é preenchida com as posições das antenas (representadas pelo seu identificador de frequência) e os efeitos nefastos (representados pelo símbolo '#'). As antenas são colocadas primeiro, seguidas pelos efeitos.

As posições das antenas são inseridas nas coordenadas correspondentes na matriz, enquanto os efeitos são representados nas mesmas posições onde ocorrem, sobrepondo as antenas caso necessário.

## Parâmetros

<i>antenas</i>	Ponteiro para a lista ligada de antenas a ser exibida.
<i>efeitos</i>	Ponteiro para a lista ligada de efeitos nefastos a ser exibidos.



# Índice

- Antena, [5](#)
- AntenaF, [6](#)
- CalcularDimensaoAntenas
  - matriz.c, [28](#)
  - matriz.h, [31](#)
- CalcularDimensaoAntenasEEfeitos
  - matriz.c, [28](#)
  - matriz.h, [32](#)
- CalcularEfeito
  - funcoes.c, [8](#)
  - funcoes.h, [16](#)
- CarregarFicheiro
  - funcoes.c, [9](#)
  - funcoes.h, [17](#)
- CriarAntena
  - funcoes.c, [9](#)
  - funcoes.h, [17](#)
- CriarEfeitoNefasto
  - funcoes.c, [10](#)
  - funcoes.h, [18](#)
- DestroiLista
  - funcoes.c, [10](#)
  - funcoes.h, [19](#)
- EfeitoNefasto, [6](#)
- funcoes.c, [7](#)
  - CalcularEfeito, [8](#)
  - CarregarFicheiro, [9](#)
  - CriarAntena, [9](#)
  - CriarEfeitoNefasto, [10](#)
  - DestroiLista, [10](#)
  - GravarFicheiroBin, [11](#)
  - InserirAntena, [11](#)
  - InserirEfeitoNefasto, [12](#)
  - LerFicheiroBin, [12](#)
  - RemoverAntena, [13](#)
  - RemoverAntenasComConflito, [13](#)
- funcoes.h, [14](#)
  - CalcularEfeito, [16](#)
  - CarregarFicheiro, [17](#)
  - CriarAntena, [17](#)
  - CriarEfeitoNefasto, [18](#)
  - DestroiLista, [19](#)
  - GravarFicheiroBin, [20](#)
  - InserirAntena, [20](#)
  - InserirEfeitoNefasto, [21](#)
  - LerFicheiroBin, [22](#)
- RemoverAntena, [23](#)
- RemoverAntenasComConflito, [23](#)
- GravarFicheiroBin
  - funcoes.c, [11](#)
  - funcoes.h, [20](#)
- InserirAntena
  - funcoes.c, [11](#)
  - funcoes.h, [20](#)
- InserirEfeitoNefasto
  - funcoes.c, [12](#)
  - funcoes.h, [21](#)
- LerFicheiroBin
  - funcoes.c, [12](#)
  - funcoes.h, [22](#)
- main
  - main.c, [26](#)
- main.c, [24](#)
  - main, [26](#)
- matriz.c, [26](#)
  - CalcularDimensaoAntenas, [28](#)
  - CalcularDimensaoAntenasEEfeitos, [28](#)
  - MostrarAntenas, [29](#)
  - MostrarAntenasEEfeitos, [29](#)
- matriz.h, [29](#)
  - CalcularDimensaoAntenas, [31](#)
  - CalcularDimensaoAntenasEEfeitos, [32](#)
  - MostrarAntenas, [32](#)
  - MostrarAntenasEEfeitos, [33](#)
- MostrarAntenas
  - matriz.c, [29](#)
  - matriz.h, [32](#)
- MostrarAntenasEEfeitos
  - matriz.c, [29](#)
  - matriz.h, [33](#)
- RemoverAntena
  - funcoes.c, [13](#)
  - funcoes.h, [23](#)
- RemoverAntenasComConflito
  - funcoes.c, [13](#)
  - funcoes.h, [23](#)