

## My Project

Gerado por Doxygen 1.12.0



<b>1 README</b>	<b>1</b>
<b>2 Índice das estruturas de dados</b>	<b>3</b>
2.1 Estruturas de dados	3
<b>3 Índice dos ficheiros</b>	<b>5</b>
3.1 Lista de ficheiros	5
<b>4 Documentação da estruturas de dados</b>	<b>7</b>
4.1 Referência à estrutura Antena	7
4.1.1 Descrição detalhada	7
4.1.2 Documentação dos campos e atributos	7
4.1.2.1 frequencia	7
4.1.2.2 prox	7
4.1.2.3 x	7
4.1.2.4 y	8
4.2 Referência à estrutura AntenaFicheiro	8
4.2.1 Descrição detalhada	8
4.2.2 Documentação dos campos e atributos	8
4.2.2.1 frequencia	8
4.2.2.2 x	8
4.2.2.3 y	8
4.3 Referência à estrutura Nefasto	9
4.3.1 Descrição detalhada	9
4.3.2 Documentação dos campos e atributos	9
4.3.2.1 prox	9
4.3.2.2 x	9
4.3.2.3 y	9
<b>5 Documentação do ficheiro</b>	<b>11</b>
5.1 Referência ao ficheiro C:/Users/user/source/repos/Antenas01/dados.h	11
5.1.1 Descrição detalhada	11
5.2 C:/Users/user/source/repos/Antenas01/dados.h	12
5.3 Referência ao ficheiro C:/Users/user/source/repos/Antenas01/funcoes.c	12
5.3.1 Descrição detalhada	13
5.3.2 Documentação das macros	13
5.3.2.1 _CRT_SECURE_NO_WARNINGS	13
5.3.3 Documentação das funções	13
5.3.3.1 carregarFicheiro()	13
5.3.3.2 criarAntena()	13
5.3.3.3 criarInserirNefasto()	14
5.3.3.4 GravarBinario()	14
5.3.3.5 inserirAntena()	14
5.3.3.6 LerBinario()	14

---

5.3.3.7	listarAntenas()	14
5.3.3.8	listarNefastos()	14
5.3.3.9	removerAntena()	15
5.3.4	Documentação das variáveis	15
5.3.4.1	maxX	15
5.3.4.2	maxY	15
5.4	C:/Users/user/source/repos/Antenas01/funcoes.c	15
5.5	Referência ao ficheiro C:/Users/user/source/repos/Antenas01/funcoes.h	18
5.5.1	Descrição detalhada	18
5.5.2	Documentação das funções	19
5.5.2.1	carregarFicheiro()	19
5.5.2.2	criarAntena()	19
5.5.2.3	criarInserirNefasto()	19
5.5.2.4	GravarBinario()	19
5.5.2.5	inserirAntena()	19
5.5.2.6	LerBinario()	20
5.5.2.7	listarAntenas()	20
5.5.2.8	listarNefastos()	20
5.5.2.9	removerAntena()	20
5.6	C:/Users/user/source/repos/Antenas01/funcoes.h	21
5.7	Referência ao ficheiro C:/Users/user/source/repos/Antenas01/main.c	21
5.7.1	Descrição detalhada	21
5.7.2	Documentação das funções	22
5.7.2.1	main()	22
5.8	C:/Users/user/source/repos/Antenas01/main.c	22

# Chapter 1

## Descrição Geral do Projeto:

Pretende-se considerar uma cidade com várias antenas. Cada antena é sintonizada numa frequência específica indicada por um carácter. O mapa das antenas com as suas localizações (coordenadas na matriz) e frequências é representado através de uma matriz. Considerando que o sinal de cada antena aplica um efeito nefasto em localizações específicas L com base nas frequências de ressonância das antenas. Em particular, o efeito nefasto ocorre em qualquer localização L que esteja perfeitamente alinhada com duas antenas da mesma frequência - mas apenas quando uma das antenas está duas vezes mais distante que a outra. Isso significa que para qualquer par de antenas com a mesma frequência, existem duas localizações, uma de cada lado das antenas.

### 1.1 Objetivos Específicos:

Nesta primeira fase é pedido que considerando a contextualização acima, procuremos implementar as funcionalidades seguintes: Definição de uma estrutura de dados ED, para a representação das antenas, sob a forma de uma lista ligada simples. Cada registo da lista ligada deverá conter a frequência de ressonância de uma antena e suas coordenadas; Carregamento para uma estrutura de dados ED dos dados das antenas constantes num ficheiro de texto. A operação deverá considerar matrizes de caracteres com qualquer dimensão.

Implementar operações de manipulação da lista ligada do tipo ED, incluindo: a) Inserção de uma nova antena na lista ligada; Remoção de uma antena constante na lista ligada; b) Dedução automática das localizações com efeito nefasto e respetiva representação sob a forma de uma lista ligada; c) Listagem de forma tabular na consola das antenas e localizações com efeito nefasto.



## Chapter 2

# Índice das estruturas de dados

### 2.1 Estruturas de dados

Lista das estruturas de dados com uma breve descrição:

Antena	7
AntenaFicheiro	8
Nefasto	9





## Chapter 3

# Índice dos ficheiros

### 3.1 Lista de ficheiros

Lista de todos os ficheiros documentados com uma breve descrição:

C:/Users/user/source/repos/Antenas01/ <a href="#">dados.h</a>	
Ficheiro cabeho estruturas e variaveis do programa antenas . . . . .	11
C:/Users/user/source/repos/Antenas01/ <a href="#">funcoes.c</a>	
Ficheiro de funs do programa antenas . . . . .	12
C:/Users/user/source/repos/Antenas01/ <a href="#">funcoes.h</a>	
Cabeho de funs e estruturas do programa antenas . . . . .	18
C:/Users/user/source/repos/Antenas01/ <a href="#">main.c</a>	
Ficheiro principal do programa antenas . . . . .	21



## Chapter 4

# Documentação da estruturas de dados

### 4.1 Referência à estrutura Antena

#### Campos de Dados

- char `frequencia`
- int `x`
- int `y`
- struct `Antena` \* `prox`

#### 4.1.1 Descrição detalhada

Definido na linha 10 do ficheiro `dados.h`.

#### 4.1.2 Documentação dos campos e atributos

##### 4.1.2.1 frequencia

```
char frequencia
```

Definido na linha 11 do ficheiro `dados.h`.

##### 4.1.2.2 prox

```
struct Antena* prox
```

Definido na linha 13 do ficheiro `dados.h`.

##### 4.1.2.3 x

```
int x
```

Definido na linha 12 do ficheiro `dados.h`.

#### 4.1.2.4 y

```
int y
```

Definido na linha 12 do ficheiro [dados.h](#).

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

- C:/Users/user/source/repos/Antenas01/[dados.h](#)

## 4.2 Referência à estrutura AntenaFicheiro

### Campos de Dados

- char [frequencia](#)
- int [x](#)
- int [y](#)

### 4.2.1 Descrição detalhada

Definido na linha 17 do ficheiro [dados.h](#).

### 4.2.2 Documentação dos campos e atributos

#### 4.2.2.1 frequencia

```
char frequencia
```

Definido na linha 18 do ficheiro [dados.h](#).

#### 4.2.2.2 x

```
int x
```

Definido na linha 19 do ficheiro [dados.h](#).

#### 4.2.2.3 y

```
int y
```

Definido na linha 19 do ficheiro [dados.h](#).

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

- C:/Users/user/source/repos/Antenas01/[dados.h](#)

## 4.3 Referência à estrutura Nefasto

### Campos de Dados

- int `x`
- int `y`
- struct `Nefasto` \* `prox`

#### 4.3.1 Descrição detalhada

Definido na linha [22](#) do ficheiro [dados.h](#).

#### 4.3.2 Documentação dos campos e atributos

##### 4.3.2.1 `prox`

```
struct Nefasto* prox
```

Definido na linha [24](#) do ficheiro [dados.h](#).

##### 4.3.2.2 `x`

```
int x
```

Definido na linha [23](#) do ficheiro [dados.h](#).

##### 4.3.2.3 `y`

```
int y
```

Definido na linha [23](#) do ficheiro [dados.h](#).

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

- `C:/Users/user/source/repos/Antenas01/dados.h`



## Chapter 5

# Documentação do ficheiro

### 5.1 Referência ao ficheiro

**C:/Users/user/source/repos/Antenas01/dados.h**

ficheiro cabeho estruturas e variaveis do programa antenas

#### Estruturas de Dados

- struct [Antena](#)
- struct [AntenaFicheiro](#)
- struct [Nefasto](#)

#### Definições de tipos

- typedef struct Antena **Antena**
- typedef struct AntenaFicheiro **AntenaFicheiro**
- typedef struct Nefasto **Nefasto**

#### 5.1.1 Descrição detalhada

ficheiro cabeho estruturas e variaveis do programa antenas

#### Autor

\$NSMC1975

#### Data

March 2025

Definido no ficheiro [dados.h](#).

## 5.2 C:/Users/user/source/repos/Antenas01/dados.h

Ir para a documentação deste ficheiro.

```
00001
00008 #pragma once
00009
00010 typedef struct Antena {
00011     char frequencia; // 1 carater que representa uma frequencia de resonancia da antena
00012     int x, y;        // Coordenadas x e y na matriz
00013     struct Antena* prox; // Apontador para a proxima antena
00014 } Antena;
00015
00016 //struct para gravar em ficheiro
00017 typedef struct AntenaFicheiro {
00018     char frequencia; // 1 carater que representa uma frequencia de resonancia da antena
00019     int x, y;        // Coordenadas x e y na matriz
00020 } AntenaFicheiro;
00021
00022 typedef struct Nefasto {
00023     int x, y;        // Coordenadas da localiza com efeito nefasto
00024     struct Nefasto* prox; // Pro efeito nefasto na lista ligada
00025 } Nefasto;
00026
00027
```

## 5.3 Referência ao ficheiro

### C:/Users/user/source/repos/Antenas01/funcoes.c

Ficheiro de funs do programa antenas.

```
#include "funcoes.h"
#include <stdio.h>
#include <malloc.h>
#include <locale.h>
```

### Funções

- **Antena \*** **criarAntena** (char freq, int x, int y)  
*cria antena*
- **Antena \*** **inserirAntena** (**Antena \***inicio, **Antena \***a)  
*fun inserir antena*
- **Antena \*** **carregarFicheiro** (const char \*nomeFicheiro)  
*Fun Carregar Ficheiro de TEXTO.*
- void **listarAntenas** (**Antena \***inicio)  
*Listar antenas para o ecran.*
- **Antena \*** **removerAntena** (**Antena \***inicio, int x, int y)  
*fun remover antenas*
- **Nefasto \*** **criarInserirNefasto** (**Nefasto \***lista, int x, int y)
- void **listarNefastos** (**Nefasto \***lista)
- bool **gravarBinario** (char \*nomeFicheiro, **Antena \***h)  
*Preservar dados em ficheiro Grava localiza das antenas em Ficheiro binario.*
- **Antena \*** **lerBinario** (char \*nomeFicheiro)  
*Lforma de ficheiro.*

### Variáveis

- int **maxX** = 0
- int **maxY** = 0



### 5.3.1 Descrição detalhada

Ficheiro de funs do programa antenas.

---

#### Autor

\$NSMC1975

#### Data

March 2025

Definido no ficheiro [funcoes.c](#).

### 5.3.2 Documentação das macros

#### 5.3.2.1 \_CRT\_SECURE\_NO\_WARNINGS

```
#define _CRT_SECURE_NO_WARNINGS
```

Definido na linha 9 do ficheiro [funcoes.c](#).

### 5.3.3 Documentação das funções

#### 5.3.3.1 carregarFicheiro()

```
Antena * carregarFicheiro (  
    const char * nomeFicheiro)
```

Fun Carregar Ficheiro de TEXTO.

#### Parâmetros

in	nome	e localiza do ficheiro TXT a carregar
----	------	---------------------------------------

#### Retorna

apontador para a antena criada

Definido na linha 77 do ficheiro [funcoes.c](#).

#### 5.3.3.2 criarAntena()

```
Antena * criarAntena (  
    char freq,  
    int x,  
    int y)
```

cria antena

#### Parâmetros

in	freq	frequencia da antena, x e y coordenadas
----	------	---

#### Retorna

apontador para a antena criada

Definido na linha 29 do ficheiro [funcoes.c](#).

### 5.3.3.3 criarInserirNefasto()

```
Nefasto * criarInserirNefasto (
    Nefasto * lista,
    int x,
    int y)
```

Definido na linha 198 do ficheiro [funcoes.c](#).

### 5.3.3.4 gravarBinario()

```
bool gravarBinario (
    char * nomeFicheiro,
    Antena * h)
```

Preservar dados em ficheiro Grava localiza das antenas em Ficheiro binario.

Definido na linha 229 do ficheiro [funcoes.c](#).

### 5.3.3.5 inserirAntena()

```
Antena * inserirAntena (
    Antena * inicio,
    Antena * a)
```

fun inserir antena

#### Parâmetros

in	<i>inicio</i>	apontador para a primeira antena, antena que queremos inserir
----	---------------	---

#### Retorna

apontador para a primeira antena (lista de antenas)

Definido na linha 44 do ficheiro [funcoes.c](#).

### 5.3.3.6 lerBinario()

```
Antena * lerBinario (
    char * nomeFicheiro)
```

Lforma de ficheiro.

Definido na linha 254 do ficheiro [funcoes.c](#).

### 5.3.3.7 listarAntenas()

```
void listarAntenas (
    Antena * inicio)
```

Listar antenas para o ecran.

#### Parâmetros

in	<i>apontador</i>	para a primeira antena
----	------------------	------------------------

#### Retorna

VOID, nao retorna nada

Definido na linha 110 do ficheiro [funcoes.c](#).

### 5.3.3.8 listarNefastos()

```
void listarNefastos (
    Nefasto * lista)
```

Definido na linha 208 do ficheiro [funcoes.c](#).

### 5.3.3.9 removerAntena()

```
Antena * removerAntena (
    Antena * inicio,
    int x,
    int y)
```

fun remover antenas

#### Parâmetros

[n]	inicio apontador para a primeira antena, x e y coordenadas da antena a remover
-----	--

#### Retorna

apontador para a primeira antena da lista

Definido na linha 159 do ficheiro [funcoes.c](#).

## 5.3.4 Documentação das variáveis

### 5.3.4.1 maxX

```
int maxX = 0
```

Definido na linha 17 do ficheiro [funcoes.c](#).

### 5.3.4.2 maxY

```
int maxY = 0
```

Definido na linha 17 do ficheiro [funcoes.c](#).

## 5.4 C:/Users/user/source/repos/Antenas01/funcoes.c

[Ir para a documentação deste ficheiro.](#)

```
00001
00009 #define _CRT_SECURE_NO_WARNINGS
00010 #include "funcoes.h"
00011 #include <stdio.h>
00012 // #include <stdlib.h>
00013 #include <malloc.h>
00014 #include <locale.h>
00015
00016
00017 int maxX = 0, maxY = 0; // Variis globais para limites da matriz
00018
00019
00020 // #define dist =2;
00021
00022 #pragma region OperacoesComAntenas
00023
00029 Antena* criarAntena(char freq, int x, int y) {
00030     Antena* nova = (Antena*)malloc(sizeof(Antena));
00031     if (!nova) return NULL;
00032     nova->frequencia = freq;
00033     nova->x = x;
00034     nova->y = y;
00035     nova->prox = NULL;
00036     return nova;
00037 }
00038
00044 Antena* inserirAntena(Antena* inicio, Antena* a) {
00045     if (a == NULL) return inicio; // Se a antena for invda, retorna a lista inalterada
00046
00047     // Inserir no ino se a lista estiver vazia ou se a antena deve ser a primeira
00048     if (inicio == NULL || a->x < inicio->x || (a->x == inicio->x && a->y < inicio->y)) {
00049         a->prox = inicio;
00050         return a; // Novo ino da lista
00051     }
00052
00053     Antena* aux = inicio;
00054     Antena* aux2 = NULL;
00055
00056     // Percorrer a lista para encontrar o ponto de inser correto
```

```

00057     while (aux != NULL && (a->x > aux->x || (a->x == aux->x && a->y > aux->y))) {
00058         aux2 = aux;
00059         aux = aux->prox;
00060     }
00061
00062     // Garantir que aux2 nNULL antes de alterar os apontadores, segmentation fault
00063     if (aux2 != NULL) {
00064         aux2->prox = a;
00065         a->prox = aux;
00066     }
00067
00068     return inicio; // Retorna a cabeda lista
00069 }
00070
00071
00072 Antena* carregarFicheiro(const char* nomeFicheiro) {
00073     FILE* ficheiro = fopen(nomeFicheiro, "r");
00074     if (!ficheiro) {
00075         printf("Erro ao abrir o ficheiro!\n");
00076         return NULL;
00077     }
00078
00079     Antena* inicio = NULL;
00080     char linha[100]; // Mmo de caracteres por linha
00081     int nrLinha = 0; // Contador de numero linhas da matriz
00082
00083     while (fgets(linha, sizeof(linha), ficheiro)) {
00084         for (int x = 0; linha[x] != '\0'; x++) {
00085             if (linha[x] != '.' && linha[x] != '\n') { // Ignora '.' e quebras de linha
00086                 Antena* nova = criarAntena(linha[x], x, nrLinha);
00087                 if (nova) { // Garante que a antena foi criada com sucesso
00088                     inicio = inserirAntena(inicio, nova);
00089                 }
00090             }
00091         }
00092         nrLinha++; // Pra linha da matriz
00093     }
00094
00095     fclose(ficheiro);
00096     return inicio; // Retorna a nova inicio da lista
00097 }
00098
00099 void listarAntenas(Antena* inicio) {
00100     /* Antena* aux = inicio;
00101     while (aux) {
00102         printf("Antena: %c (%d, %d)\n", aux->frequencia, aux->x, aux->y);
00103         aux = aux->prox;
00104     }
00105     */
00106     if (!inicio) {
00107         printf("Nao ha antenas na lista.\n");
00108         return;
00109     }
00110
00111     // Determinar os limites mmos da matriz
00112     //int maxX = 0, maxY = 0;
00113     Antena* aux = inicio;
00114     while (aux) {
00115         if (aux->x > maxX) maxX = aux->x;
00116         if (aux->y > maxY) maxY = aux->y;
00117         aux = aux->prox;
00118     }
00119     maxX++; maxY++; // Ajustar tamanho para poderem aparecer mais tarde os nefastos
00120
00121     // Percorrer a matriz e imprimir diretamente
00122     for (int y = 0; y <= maxY; y++) {
00123         for (int x = 0; x <= maxX; x++) {
00124             aux = inicio;
00125             char simbolo = '.'; // Por padr imprimimos '.'
00126
00127             // Verificar se existe uma antena nesta posi
00128             while (aux) {
00129                 if (aux->x == x && aux->y == y) {
00130                     simbolo = aux->frequencia;
00131                     break; // Encontramos a antena, podemos parar
00132                 }
00133                 aux = aux->prox;
00134             }
00135
00136             printf("%c", simbolo);
00137         }
00138         printf("\n"); // Nova linha apada linha da matriz
00139     }
00140 }
00141
00142
00143
00144
00145
00146
00147
00148
00149
00150
00151
00152
00153

```

```

00159 Antena* removerAntena(Antena* inicio, int x, int y) {
00160     if (inicio == NULL) {
00161         printf("Erro: Lista vazia, nhntenas para remover.\n");
00162         return NULL;
00163     }
00164
00165     Antena* atual = inicio;
00166     Antena* anterior = NULL;
00167
00168     // Percorrer a lista para encontrar a antena a remover
00169     while (atual != NULL) {
00170         if (atual->x == x && atual->y == y) {
00171             // Encontrou a antena a remover
00172             if (anterior == NULL) {
00173                 // Caso especial: a antena esta ino da lista
00174                 inicio = atual->prox;
00175             }
00176             else {
00177                 // Caso geral: ligar o anterior ao pro n
00178                 anterior->prox = atual->prox;
00179             }
00180             free(atual); // Libertar a mem da antena removida
00181             printf("Antena em (%d, %d) removida com sucesso.\n", x, y);
00182             return inicio;
00183         }
00184         anterior = atual;
00185         atual = atual->prox;
00186     }
00187
00188     // Se chegou aqui, significa que nencontrou a antena
00189     printf("Erro: Nenhuma antena encontrada em (%d, %d).\n", x, y);
00190     return inicio;
00191 }
00192
00193 #pragma endregion
00194
00195 #pragma region EfeitosNefastos
00196 // EFEITOS NEFASTOS
00197 //fun adicionar Nefasto
00198 Nefasto* criarInserirNefasto(Nefasto* lista, int x, int y) {
00199     Nefasto* novo = (Nefasto*)malloc(sizeof(Nefasto));
00200     if (!novo) return lista; // Falha ao alocar mem
00201
00202     novo->x = x;
00203     novo->y = y;
00204     novo->prox = lista; // Adiciona no ino da lista, nao precisa de estar ordenado
00205
00206     return novo;
00207 }
00208 void listarNefastos(Nefasto* lista) {
00209     Nefasto* atual = lista;
00210     while (atual != NULL) {
00211         printf("Efeito nefasto em (%d, %d)\n", atual->x, atual->y);
00212         atual = atual->prox;
00213     }
00214 }
00215
00216 // fun efeitos nefastos REVER ESTA FUNCAO
00217
00218
00219
00220 #pragma endregion
00221
00222 #pragma region FICHEIROS BINARIOS
00223
00224 // ===== Ficheros binarios =====
00229 bool gravarBinario(char* nomeFicheiro, Antena* h) {
00230     FILE* fp;
00231
00232     if (h == NULL) return false;
00233     if ((fp = fopen(nomeFicheiro, "wb")) == NULL) return false;
00234     //grava todos as antenas para ficheiro
00235     Antena* aux = h;
00236     AntenaFicheiro auxAntena = { 0 }; //aqui ariavel auxiliar para gravar em ficheiro!
00237     while (aux) { //enquanto houver antenas
00238         //Colocar no registo de ficheiro a inf que esto registo de mem
00239         auxAntena.x = aux->x;
00240         auxAntena.y = aux->y;
00241         auxAntena.frequencia = aux->frequencia;
00242
00243         fwrite(&auxAntena, sizeof(AntenaFicheiro), 1, fp);
00244
00245         aux = aux->prox;
00246     }
00247     fclose(fp);
00248     return true;
00249 }

```

```

00250
00254 Antena* lerBinario(char* nomeFicheiro) {
00255     FILE* fp;
00256     Antena* h = NULL;
00257     Antena* aux;
00258     AntenaFicheiro auxAntena;
00259
00260     if ((fp = fopen(nomeFicheiro, "rb")) == NULL) return NULL;
00261     //l registos no ficheiro
00262     //AntenaFicheiro auxAntena;
00263     while (fread(&auxAntena, sizeof(AntenaFicheiro), 1, fp)) {
00264         printf("Antena=%c %d %d \n", auxAntena.frequencia, auxAntena.x, auxAntena.y); //para debug
00265         aux = criarAntena(auxAntena.frequencia, auxAntena.x, auxAntena.y);
00266         h = inserirAntena(h, aux);
00267     }
00268     fclose(fp);
00269     return h;
00270 }
00271 #pragma endregion
00272
00273 // falta funcao para libertar Lista de Antenas free

```

## 5.5 Referência ao ficheiro

**C:/Users/user/source/repos/Antenas01/funcoes.h**

cabeço de funs e estruturas do programa antenas

```

#include "dados.h"
#include <malloc.h>
#include <stdbool.h>

```

### Funções

- **Antena \*** **criarAntena** (char freq, int x, int y)  
*cria antena*
- **Antena \*** **inserirAntena** (Antena \*inicio, Antena \*a)  
*fun inserir antena*
- **Antena \*** **removerAntena** (Antena \*inicio, int x, int y)  
*fun remover antenas*
- **Antena \*** **carregarFicheiro** (const char \*nomeFicheiro)  
*Fun Carregar Ficheiro de TEXTO.*
- void **listarAntenas** (Antena \*inicio)  
*Listar antenas para o ecran.*
- **Nefasto \*** **criarInserirNefasto** (Nefasto \*lista, int x, int y)
- void **listarNefastos** (Nefasto \*lista)
- bool **gravarBinario** (char \*nomeFicheiro, Antena \*h)  
*Preservar dados em ficheiro Grava localiza das antenas em Ficheiro binario.*
- **Antena \*** **lerBinario** (char \*nomeFicheiro)  
*Lforma de ficheiro.*

### 5.5.1 Descrição detalhada

cabeço de funs e estruturas do programa antenas

**Autor**

\$NSMC1975

**Data**

March 2025

Definido no ficheiro [funcoes.h](#).

## 5.5.2 Documentação das funções

### 5.5.2.1 carregarFicheiro()

```
Antena * carregarFicheiro (
    const char * nomeFicheiro)
```

Fun Carregar Ficheiro de TEXTO.

#### Parâmetros

in	nome	e localiza do ficheiro TXT a carregar
----	------	---------------------------------------

#### Retorna

apontador para a antena criada

Definido na linha 77 do ficheiro [funcoes.c](#).

### 5.5.2.2 criarAntena()

```
Antena * criarAntena (
    char freq,
    int x,
    int y)
```

cria antena

#### Parâmetros

in	freq	frequencia da antena, x e y coordenadas
----	------	---

#### Retorna

apontador para a antena criada

Definido na linha 29 do ficheiro [funcoes.c](#).

### 5.5.2.3 criarInserirNefasto()

```
Nefasto * criarInserirNefasto (
    Nefasto * lista,
    int x,
    int y)
```

Definido na linha 198 do ficheiro [funcoes.c](#).

### 5.5.2.4 gravarBinario()

```
bool gravarBinario (
    char * nomeFicheiro,
    Antena * h)
```

Preservar dados em ficheiro Grava localiza das antenas em Ficheiro binario.

Definido na linha 229 do ficheiro [funcoes.c](#).

### 5.5.2.5 inserirAntena()

```
Antena * inserirAntena (
    Antena * inicio,
    Antena * a)
```

fun inserir antena

**Parâmetros**

<code>in</code>	<code>inicio</code>	apontador para a primeira antena, antena que queremos inserir
-----------------	---------------------	---

**Retorna**

apontador para a primeira antena (lista de antenas)

Definido na linha 44 do ficheiro [funcoes.c](#).

**5.5.2.6 lerBinario()**

```
Antena * lerBinario (  
    char * nomeFicheiro)
```

Lforma de ficheiro.

Definido na linha 254 do ficheiro [funcoes.c](#).

**5.5.2.7 listarAntenas()**

```
void listarAntenas (  
    Antena * inicio)
```

Listar antenas para o ecran.

**Parâmetros**

<code>in</code>	<code>apontador</code>	para a primeira antena
-----------------	------------------------	------------------------

**Retorna**

VOID, nao retorna nada

Definido na linha 110 do ficheiro [funcoes.c](#).

**5.5.2.8 listarNefastos()**

```
void listarNefastos (  
    Nefasto * lista)
```

Definido na linha 208 do ficheiro [funcoes.c](#).

**5.5.2.9 removerAntena()**

```
Antena * removerAntena (  
    Antena * inicio,  
    int x,  
    int y)
```

fun remover antenas

**Parâmetros**

<code>[n]</code>	inicio apontador para a primeira antena, x e y coordenadas da antena a remover
------------------	--

**Retorna**

apontador para a primeira antena da lista

Definido na linha 159 do ficheiro [funcoes.c](#).



## 5.6 C:/Users/user/source/repos/Antenas01/funcoes.h

Ir para a documentação deste ficheiro.

```

00001
00010 #pragma once
00011
00012 #include "dados.h"
00013 #include <malloc.h>
00014 #include <stdbool.h>
00015
00016 // FUNES
00017 // Criar Antena
00018 Antena* criarAntena(char freq, int x, int y);
00019
00020 // Inserir Antena
00021 Antena* inserirAntena(Antena* inicio, Antena* a);
00022
00023 // funcao Eliminar Antena
00024 Antena* removerAntena(Antena* inicio, int x, int y);
00025
00026 // fun carregar ficheiro
00027 Antena* carregarFicheiro(const char* nomeFicheiro);
00028
00029 // fun Listar Antenas
00030 void listarAntenas(Antena* inicio);
00031
00032 //fun Nefasto
00033 Nefasto* criarInserirNefasto(Nefasto* lista, int x, int y);
00034 void listarNefastos(Nefasto* lista);
00035
00036
00037
00038 //funs com FICHEIROS
00039 bool gravarBinario(char* nomeFicheiro, Antena* h);
00040 Antena* lerBinario(char* nomeFicheiro);
00041
00042
00043
00044
00045 // funcao para libertar Lista de Antenas free
00046
00047
00048
00049

```

## 5.7 Referência ao ficheiro C:/Users/user/source/repos/Antenas01/main.c

Ficheiro principal do programa antenas.

```

#include "funcoes.h"
#include <stdio.h>
#include <locale.h>

```

### Funções

- int [main](#) ()

### 5.7.1 Descrição detalhada

Ficheiro principal do programa antenas.

Pretende-se considerar uma cidade com vas antenas. Cada antena ntonizada numa frequia especca indicada por um caracter.

#### Autor

\$NSMC1975

#### Data

March 2025

Definido no ficheiro [main.c](#).

## 5.7.2 Documentação das funções

### 5.7.2.1 main()

```
int main ()
```

Definido na linha 17 do ficheiro `main.c`.

## 5.8 C:/Users/user/source/repos/Antenas01/main.c

[Ir para a documentação deste ficheiro.](#)

```
00001
00011 #include "funcoes.h"
00012 #include <stdio.h>
00013 #include <locale.h>
00014
00015 #pragma once
00016
00017 int main() {
00018     setlocale(LC_ALL, "Portuguese");
00019     Antena* inicio = NULL; // Lista vazia
00020     Antena* inicio2 = NULL; // para nao estragar a inicio a ler do binario, por seguran
00021     Nefasto* inicio4 = NULL; // Lista de efeitos nefastos vazia
00022
00023     // Carregar as antenas do ficheiro
00024     inicio = carregarFicheiro("antenas.txt");
00025
00026     // Teste: listar as antenas carregadas
00027     printf("Lista de Antenas Ordenada:\n");
00028     listarAntenas(inicio);
00029
00030     // tenta remover a antena na posi (4,4) que nao existe
00031     printf("\nRemover antena na posicao (3,4)...\n");
00032     inicio = removerAntena(inicio, 3, 4);
00033
00034     // remove antena que existe
00035     printf("\nRemover antena na posicao (3,0)...\n");
00036     inicio = removerAntena(inicio, 3, 0);
00037
00038     // Listar novamente
00039     printf("\nLista de Antenas:\n");
00040     listarAntenas(inicio);
00041
00042     //adicionar de novo antena posicao 3,0
00043     printf("\nAdicionar antena na posicao (3,0)...\n");
00044     inicio = inserirAntena(inicio, criarAntena('E', 3, 0));
00045
00046     printf("\nLista de Antenas antes de imprimir para binario:\n");
00047     listarAntenas(inicio);
00048
00049     printf("gravar binario\n");
00050     gravarBinario("antenas1.bin", inicio);
00051
00052     //ler bin e imprimir ecran
00053     printf("lido do binario e escrever na struct inicio2:\n");
00054     inicio2 = lerBinario("antenas1.bin");
00055     listarAntenas(inicio2);
00056
00057     //tratar nefastos
00058     printf("\nEfeitos nefastos:\n");
00059     printf("a criar nefastos manualmente para debug...");
00060     inicio4 = criarInserirNefasto(inicio4, 1, 1);
00061     inicio4 = criarInserirNefasto(inicio4, 1, 3);
00062     inicio4 = criarInserirNefasto(inicio4, 1, 4);
00063
00064     printf("listar nefastos criados:\n");
00065     listarNefastos(inicio4);
00066
00067
00068     // falta juntar os efeitos nefastos, nao esteito
00069
00070     //libertar memoria
00071     free(inicio);
00072     free(inicio2);
00073     return 0;
00074 }
```