

EDA - Ivo Fernandes 8819 -2025

Generated by Doxygen 1.13.2

1 Estruturas de dados avançadas (EDA) - Projeto - Grupo I	1
1.1 Descrição	1
1.2 Funcionalidades	1
1.3 Instalação	1
1.4 Contribuições	1
1.5 Licença	1
2 Data Structure Index	3
2.1 Data Structures	3
3 File Index	5
3.1 File List	5
4 Data Structure Documentation	7
4.1 Antena Struct Reference	7
4.1.1 Detailed Description	7
4.2 AntenasFile Struct Reference	7
4.3 ListaLigadaAntenas Struct Reference	8
4.3.1 Detailed Description	8
5 File Documentation	9
5.1 antenas.c File Reference	9
5.1.1 Detailed Description	10
5.1.2 Function Documentation	10
5.1.2.1 carregarFicheiro()	10
5.1.2.2 criarLista()	10
5.1.2.3 detectarEfeitoNefasto()	10
5.1.2.4 gravarDadosFicheiro()	11
5.1.2.5 inserirAntena()	11
5.1.2.6 listarAntenas()	12
5.1.2.7 removerAntena()	12
5.2 antenas.c	12
5.3 antenas.h File Reference	14
5.3.1 Detailed Description	14
5.3.2 Function Documentation	15
5.3.2.1 carregarFicheiro()	15
5.3.2.2 criarLista()	15
5.3.2.3 detectarEfeitoNefasto()	15
5.3.2.4 gravarDadosFicheiro()	16
5.3.2.5 inserirAntena()	16
5.3.2.6 listarAntenas()	16
5.3.2.7 removerAntena()	17
5.4 antenas.h	17
5.5 dados.h File Reference	17

5.5.1 Detailed Description	18
5.5.2 Typedef Documentation	18
5.5.2.1 Antena	18
5.6 dados.h	19
Index	21

Chapter 1

Estruturas de dados avançadas (EDA) - Projeto - Grupo I

1.1 Descrição

Este projeto faz parte do trabalho de Estruturas de dados avançadas (EDA), tendo o Grupo I como objetivo gerir antenas e seus efeitos nefastos com base na sua proximidade. O trabalho recorre a apontadores e listas ligadas.

1.2 Funcionalidades

- Carregar Lista de Antenas a partir de ficheiro
- Criar lista ligada
- Listar antenas
- Remover antenas da Lista
- Deteta efeito nefasto em lista
- Gravar em ficheiro

1.3 Instalação

1. Clone o repositório:

```
git clone https://github.com/ivofernandes8819/trabalho-eda-gl.git
```

2. Navegue até ao diretório do projeto:

```
cd trabalho-eda-gl
```

Compilar e executar a executar o ficheiro main.c

1.4 Contribuições

Contribuições são bem-vindas! Faça um fork do repositório e envie um pull request.

1.5 Licença

Este projeto está licenciado sob a Licença MIT. Consulte o ficheiro LICENSE para mais detalhes.

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Antena	7
AntenasFile	7
ListaLigadaAntenas	8

Chapter 3

File Index

3.1 File List

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

antenas.c	9
antenas.h	14
dados.h	17

Chapter 4

Data Structure Documentation

4.1 Antena Struct Reference

```
#include <dados.h>
```

Data Fields

- char **frequencia**
- int **x**
- int **y**
- struct [Antena](#) * **next**

4.1.1 Detailed Description

Gere Elemento da lista Antenas

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

- [dados.h](#)

4.2 AntenasFile Struct Reference

Data Fields

- char **frequencia**
- int **x**
- int **y**

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

- [dados.h](#)

4.3 ListaLigadaAntenas Struct Reference

```
#include <dados.h>
```

Data Fields

- [Antena](#) * head

4.3.1 Detailed Description

Gere a lista de elementos

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

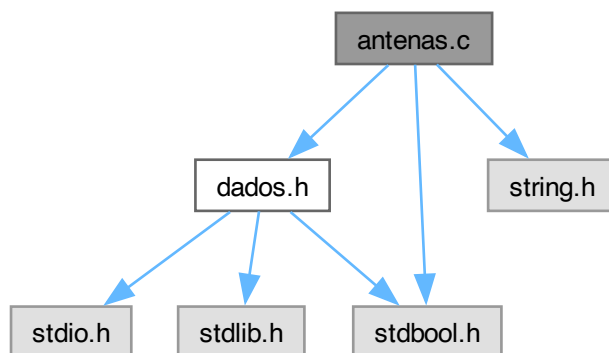
- [dados.h](#)

Chapter 5

File Documentation

5.1 antenas.c File Reference

```
#include "dados.h"  
#include <stdbool.h>  
#include <string.h>  
Include dependency graph for antenas.c:
```



Functions

- `ListaLigadaAntenas * criarLista ()`
Cria e inicializa uma nova lista ligada para armazenar antenas.
- `void inserirAntena (ListaLigadaAntenas *lista, char frequencia, int x, int y)`
insere uma nova antena em uma lista ligada de antenas.
- `void removerAntena (ListaLigadaAntenas *lista, int x, int y)`
Remove uma antena específica da lista ligada.
- `void listarAntenas (ListaLigadaAntenas *lista)`
Lista antenas com base nas coordenadas do caractere no ficheiro.

- void `carregarFicheiro` (`ListaLigadaAntenas *lista`, `const char *nomeFicheiro`)
Função que carrega itens existentes no ficheiro. A posição do caractere no ficheiro representa as suas coordenadas.
- void `detectarEfeitoNefasto` (`ListaLigadaAntenas *lista`)
Detecta o efeito nefasto causado por antenas com a mesma frequência localizadas na mesma linha (coordenada x).
- bool `gravarDadosFicheiro` (`const char *nomeFicheiro`, `ListaLigadaAntenas *lista`)
grava dados no ficheiro binário

5.1.1 Detailed Description

Author

Ivo Fernandes | 8819

Date

2025-03-30 @project Trabalho pratico EDA - Grupo I

5.1.2 Function Documentation

5.1.2.1 `carregarFicheiro()`

```
void carregarFicheiro (  
    ListaLigadaAntenas * lista,  
    const char * nomeFicheiro)
```

Função que carrega itens existentes no ficheiro. A posição do caractere no ficheiro representa as suas coordenadas.

Note

: aprimorar validações do ficheiro: vazio, formato fora do esperado.

5.1.2.2 `criarLista()`

```
ListaLigadaAntenas * criarLista ()
```

Cria e inicializa uma nova lista ligada para armazenar antenas.

Esta função aloca memória para uma nova estrutura de lista ligada e inicializa o seu apontador head para NULL, indicando que a lista está vazia.

Returns

Um apontador para a nova lista ligada de antenas criada.

5.1.2.3 `detectarEfeitoNefasto()`

```
void detectarEfeitoNefasto (  
    ListaLigadaAntenas * lista)
```

Detecta o efeito nefasto causado por antenas com a mesma frequência localizadas na mesma linha (coordenada x).

Esta função percorre uma lista ligada de antenas e verifica se existem múltiplas antenas com a mesma frequência e coordenada x.

Parameters

<i>lista</i>	apontador para a lista ligada de antenas (ListaLigadaAntenas).
--------------	--

Note

Assumiu-se que O efeito nefasto ocorre quando mais do que uma antena opera na mesma frequência e está localizada na mesma linha (coordenada x).

5.1.2.4 gravarDadosFicheiro()

```
bool gravarDadosFicheiro (  
    const char * nomeFicheiro,  
    ListaLigadaAntenas * lista)
```

grava dados no ficheiro binário

Note

detetado que não funciona corretamente. TODO: fazer correção na próxima entrega

5.1.2.5 inserirAntena()

```
void inserirAntena (  
    ListaLigadaAntenas * lista,  
    char frequencia,  
    int x,  
    int y)
```

insere uma nova antena em uma lista ligada de antenas.

1. Aloca dinamicamente memória para uma nova antena usando `malloc`.
2. Inicializa os campos da nova antena (`frequencia`, `x`, `y`) com os valores fornecidos como parâmetros.
3. Faz o apontador `next` da nova antena apontar para o atual primeiro elemento da lista (ou seja, o antigo `head`).
4. Atualiza o apontador `head` da lista para que ele aponte para a nova antena, tornando-a o novo primeiro elemento da lista.

Parameters

<i>lista</i>	Apontador para a lista ligada de antenas.
<i>frequencia</i>	Caractere representando a frequência da antena.
<i>x</i>	Coordenada x da antena.
<i>y</i>	Coordenada y da antena.

5.1.2.6 listarAntenas()

```
void listarAntenas (
    ListaLigadaAntenas * lista)
```

Lista antenas com base nas coordenadas do caractere no ficheiro.

Esta função percorre um ficheiro contendo informações sobre antenas

Returns

Uma lista de antenas listadas pelo antenas.txt. + as inseridas pela função inserir TODO: não bate certo com a inserção na lista!

5.1.2.7 removerAntena()

```
void removerAntena (
    ListaLigadaAntenas * lista,
    int x,
    int y)
```

Remove uma antena específica da lista ligada.

Esta função percorre a lista ligada de antenas e remove a antena que corresponde aos parâmetros fornecidos (frequência, coordenada x e coordenada y).

Parameters

<i>lista</i>	Apontador para a lista ligada de antenas.
<i>frequencia</i>	Caractere representando a frequência da antena a ser removida.
<i>x</i>	Coordenada x da antena a ser removida.
<i>y</i>	Coordenada y da antena a ser removida.

Note

Se a antena não for encontrada, a lista permanece inalterada.

5.2 antenas.c

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

```
00001
00007
00008 //#include "antenas.h"
00009 #include "dados.h"
00010 #include <stdbool.h>
00011 #include <string.h>
00012
00021 ListaLigadaAntenas* criarLista() {
00022     ListaLigadaAntenas* lista = (ListaLigadaAntenas*)malloc(sizeof(ListaLigadaAntenas));
00023     lista->head = NULL;
00024     return lista;
00025 }
00026
00037
00038 void inserirAntena(ListaLigadaAntenas* lista, char frequencia, int x, int y) {
00039     Antena* nova = (Antena*)malloc(sizeof(Antena));
```



```

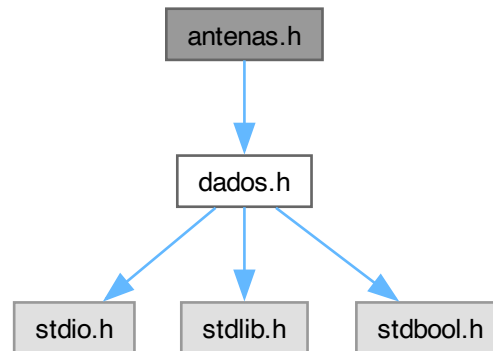
00040     nova->frequencia = frequencia;
00041     nova->x = x;
00042     nova->y = y;
00043     nova->next = lista->head;
00044     lista->head = nova;
00045 }
00046
00060 void removerAntena(ListaLigadaAntenas* lista, int x, int y) {
00061     Antena* atual = lista->head, *anterior = NULL;
00062     while (atual) {
00063         if (atual->x == x && atual->y == y) {
00064             if (anterior) anterior->next = atual->next;
00065             else lista->head = atual->next;
00066             free(atual);
00067             return;
00068         }
00069         anterior = atual;
00070         atual = atual->next;
00071     }
00072 }
00073
00081
00082 void listarAntenas(ListaLigadaAntenas* lista) {
00083     Antena* atual = lista->head;
00084     printf("Lista de Antenas:\n");
00085     while (atual) {
00086         printf("Frequência: %c, Coordenadas: (%d, %d)\n", atual->frequencia, atual->x, atual->y);
00087         atual = atual->next;
00088     }
00089     printf("Fim da lista.\n");
00090 }
00091
00097 void carregarFicheiro(ListaLigadaAntenas* lista, const char* nomeFicheiro) {
00098     FILE* file = fopen(nomeFicheiro, "r");
00099     if (!file) return;
00100
00101     char linha[256];
00102     int y = 0;
00103     while (fgets(linha, sizeof(linha), file)) {
00104         for (int x = 0; linha[x] != '\0' && linha[x] != '\n'; x++) {
00105             if ((linha[x] >= 'A' && linha[x] <= 'Z') || (linha[x] >= 'a' && linha[x] <= 'z') ||
linha[x] == '0') {
00106                 inserirAntena(lista, linha[x], x, y);
00107             }
00108         }
00109         y++;
00110     }
00111     fclose(file);
00112 }
00113
00126 void detectarEfeitoNefasto(ListaLigadaAntenas* lista) {
00127     Antena* a1 = lista->head;
00128     while (a1) {
00129         Antena* a2 = lista->head; // começa pela head
00130         int count = 0;
00131         while (a2) {
00132             if (a1->frequencia == a2->frequencia && a1->x == a2->x) {
00133                 count++;
00134                 if (count > 1) { // mais do que uma antena com a mesma coordenada X
00135                     printf("Efeito nefasto detectado:\n");
00136                     printf("Antena com Frequência: %c, Linha %d, Coordenadas: (%d, %d)\n",
a1->frequencia, a1->x, a1->y);
00137                     break;
00138                 }
00139             }
00140             a2 = a2->next;
00141         }
00142         a1 = a1->next;
00143     }
00144 }
00145
00151 bool gravarDadosFicheiro(const char* nomeFicheiro, ListaLigadaAntenas* lista) {
00152     if (lista == NULL || lista->head == NULL) return false;
00153
00154     FILE* file = fopen(nomeFicheiro, "wb");
00155     if (!file) return false;
00156
00157     Antena* atual = lista->head;
00158     while (atual) {
00159         fprintf(file, " %s \n", atual->frequencia);
00160         atual = atual->next;
00161     }
00162     fclose(file);
00163     return true;
00164 }
00165 }

```

5.3 antenas.h File Reference

```
#include "dados.h"
```

Include dependency graph for antenas.h:



Functions

- `ListaLigadaAntenas * criarLista ()`
Cria e inicializa uma nova lista ligada para armazenar antenas.
- `void inserirAntena (ListaLigadaAntenas *lista, char frequencia, int x, int y)`
insere uma nova antena em uma lista ligada de antenas.
- `void removerAntena (ListaLigadaAntenas *lista, int x, int y)`
Remove uma antena específica da lista ligada.
- `void listarAntenas (ListaLigadaAntenas *lista)`
Lista antenas com base nas coordenadas do caractere no ficheiro.
- `void carregarFicheiro (ListaLigadaAntenas *lista, const char *nomeFicheiro)`
Função que carrega itens existentes no ficheiro. A posição do caractere no ficheiro representa as suas coordenadas.
- `void detectarEfeitoNefasto (ListaLigadaAntenas *lista)`
Detecta o efeito nefasto causado por antenas com a mesma frequência localizadas na mesma linha (coordenada x).
- `bool gravarDadosFicheiro (const char *nomeFicheiro, ListaLigadaAntenas *lista)`
grava dados no ficheiro binário

5.3.1 Detailed Description

Author

Ivo Fernandes | 8819

Date

2025-03-30 @project Trabalho pratico EDA - Grupo I

5.3.2 Function Documentation

5.3.2.1 carregarFicheiro()

```
void carregarFicheiro (  
    ListaLigadaAntenas * lista,  
    const char * nomeFicheiro)
```

Função que carrega itens existentes no ficheiro. A posição do caractere no ficheiro representa as suas coordenadas.

Note

: aprimorar validações do ficheiro: vazio, formato fora do esperado.

5.3.2.2 criarLista()

```
ListaLigadaAntenas * criarLista ()
```

Cria e inicializa uma nova lista ligada para armazenar antenas.

Esta função aloca memória para uma nova estrutura de lista ligada e inicializa o seu apontador head para NULL, indicando que a lista está vazia.

Returns

Um apontador para a nova lista ligada de antenas criada.

5.3.2.3 detectarEfeitoNefasto()

```
void detectarEfeitoNefasto (  
    ListaLigadaAntenas * lista)
```

Detecta o efeito nefasto causado por antenas com a mesma frequência localizadas na mesma linha (coordenada x).

Esta função percorre uma lista ligada de antenas e verifica se existem múltiplas antenas com a mesma frequência e coordenada x.

Parameters

<i>lista</i>	apontador para a lista ligada de antenas (ListaLigadaAntenas).
--------------	--

Note

Assumiu-se que O efeito nefasto ocorre quando mais do que uma antena opera na mesma frequência e está localizada na mesma linha (coordenada x).

5.3.2.4 gravarDadosFicheiro()

```
bool gravarDadosFicheiro (  
    const char * nomeFicheiro,  
    ListaLigadaAntenas * lista)
```

grava dados no ficheiro binário

Note

detetado que não funciona corretamente. TODO: fazer correção na próxima entrega

5.3.2.5 inserirAntena()

```
void inserirAntena (  
    ListaLigadaAntenas * lista,  
    char frequencia,  
    int x,  
    int y)
```

insere uma nova antena em uma lista ligada de antenas.

1. Aloca dinamicamente memória para uma nova antena usando `malloc`.
2. Inicializa os campos da nova antena (`frequencia`, `x`, `y`) com os valores fornecidos como parâmetros.
3. Faz o apontador `next` da nova antena apontar para o atual primeiro elemento da lista (ou seja, o antigo `head`).
4. Atualiza o apontador `head` da lista para que ele aponte para a nova antena, tornando-a o novo primeiro elemento da lista.

Parameters

<i>lista</i>	Apontador para a lista ligada de antenas.
<i>frequencia</i>	Caractere representando a frequência da antena.
<i>x</i>	Coordenada x da antena.
<i>y</i>	Coordenada y da antena.

5.3.2.6 listarAntenas()

```
void listarAntenas (  
    ListaLigadaAntenas * lista)
```

Lista antenas com base nas coordenadas do caractere no ficheiro.

Esta função percorre um ficheiro contendo informações sobre antenas

Returns

Uma lista de antenas listadas pelo `antenas.txt`. + as inseridas pela função `inserir` TODO: não bate certo com a inserção na lista!

5.3.2.7 removerAntena()

```
void removerAntena (
    ListaLigadaAntenas * lista,
    int x,
    int y)
```

Remove uma antena específica da lista ligada.

Esta função percorre a lista ligada de antenas e remove a antena que corresponde aos parâmetros fornecidos (frequência, coordenada x e coordenada y).

Parameters

<i>lista</i>	Apontador para a lista ligada de antenas.
<i>frequencia</i>	Caractere representando a frequência da antena a ser removida.
<i>x</i>	Coordenada x da antena a ser removida.
<i>y</i>	Coordenada y da antena a ser removida.

Note

Se a antena não for encontrada, a lista permanece inalterada.

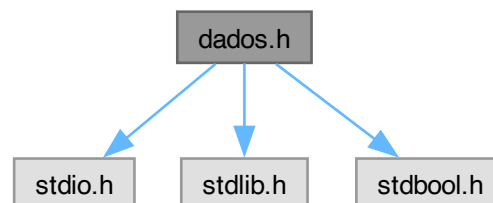
5.4 antenas.h

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

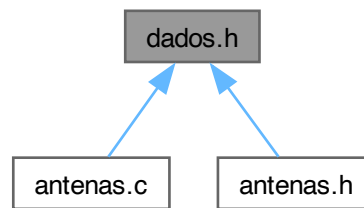
```
00001
00007 #pragma once
00008
00009 #include "dados.h"
00010
00011 ListaLigadaAntenas* criarLista();
00012 void inserirAntena(ListaLigadaAntenas* lista, char frequencia, int x, int y);
00013 void removerAntena(ListaLigadaAntenas* lista, int x, int y);
00014 void listarAntenas(ListaLigadaAntenas* lista);
00015 void carregarFicheiro(ListaLigadaAntenas* lista, const char* nomeFicheiro);
00016 void detectarEfeitoNefasto(ListaLigadaAntenas* lista);
00017 bool gravarDadosFicheiro(const char* nomeFicheiro, ListaLigadaAntenas* lista);
```

5.5 dados.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
Include dependency graph for dados.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [Antena](#)
- struct [ListaLigadaAntenas](#)
- struct [AntenasFile](#)

Typedefs

- typedef struct Antena [Antena](#)
- typedef struct AntenasFile **AntenasFile**

5.5.1 Detailed Description

Author

Ivo Fernandes | 8819

Date

2025-03-30 @project Trabalho pratico EDA - Grupo I

5.5.2 Typedef Documentation

5.5.2.1 Antena

```
typedef struct Antena Antena
```

Gere Elemento da lista Antenas

5.6 dados.h

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

```
00001
00007 #pragma once
00008
00009 #include <stdio.h>
00010 #include <stdlib.h>
00011 #include <stdbool.h>
00012
00016
00017 typedef struct Antena {
00018     char frequencia;
00019     int x, y;
00020     struct Antena* next;
00021 } Antena;
00022
00023
00027 typedef struct {
00028     Antena* head;
00029 } ListaligadaAntenas;
00030
00031 typedef struct AntenasFile {
00032     char frequencia;
00033     int x, y;
00034 }AntenasFile;
```


Index

- Antena, [7](#)
 - dados.h, [18](#)
- antenas.c, [9](#)
 - carregarFicheiro, [10](#)
 - criarLista, [10](#)
 - detectarEfeitoNefasto, [10](#)
 - gravarDadosFicheiro, [11](#)
 - inserirAntena, [11](#)
 - listarAntenas, [11](#)
 - removerAntena, [12](#)
- antenas.h, [14](#)
 - carregarFicheiro, [15](#)
 - criarLista, [15](#)
 - detectarEfeitoNefasto, [15](#)
 - gravarDadosFicheiro, [15](#)
 - inserirAntena, [16](#)
 - listarAntenas, [16](#)
 - removerAntena, [16](#)
- AntenasFile, [7](#)
- carregarFicheiro
 - antenas.c, [10](#)
 - antenas.h, [15](#)
- criarLista
 - antenas.c, [10](#)
 - antenas.h, [15](#)
- dados.h, [17](#)
 - Antena, [18](#)
- detectarEfeitoNefasto
 - antenas.c, [10](#)
 - antenas.h, [15](#)
- Estruturas de dados avançadas (EDA) - Projeto - Grupo I, [1](#)
- gravarDadosFicheiro
 - antenas.c, [11](#)
 - antenas.h, [15](#)
- inserirAntena
 - antenas.c, [11](#)
 - antenas.h, [16](#)
- ListaLigadaAntenas, [8](#)
- listarAntenas
 - antenas.c, [11](#)
 - antenas.h, [16](#)
- removerAntena
 - antenas.c, [12](#)
 - antenas.h, [16](#)