

Projeto EDA Fase 1 – Listas Ligadas

Github: github.com/danielvilaca/EDA

Adelino Daniel da Rocha Vilaça
IPCA-EST
Março 2025

Table of Contents

Índice

Projeto EDA Fase 1 – Listas Ligadas	i
Github: github.com/danielvilaca/EDA	i
Adelino Daniel da Rocha Vilaça	i
IPCA-EST	i
Março 2025.....	i
Table of Contents	ii
Class Documentation.....	3
Antena Struct Reference.....	3
Public Attributes	3
Detailed Description	3
Member Data Documentation	3
Node Struct Reference.....	4
Public Attributes	4
Detailed Description	4
Member Data Documentation	4
File Documentation	5
antena.c File Reference.....	5
Functions	5
Detailed Description	5
Function Documentation	5
antena.c.....	7
antena.h File Reference	9
Classes	9
Typedefs	9
Functions	9
Detailed Description	9
Typedef Documentation	9
Function Documentation	9
antena.h.....	12
main.c File Reference	13
Functions	13
Detailed Description	13
Function Documentation	13
main.c	14
matriz.c File Reference.....	15
Functions	15
Detailed Description	15
Function Documentation	15
matriz.c	17
matriz.h File Reference.....	19
Classes	19
Typedefs	19
Functions	19
Detailed Description	19
Typedef Documentation	19
Function Documentation	19
matriz.h.....	21
Index.....	22

Class Documentation

Antena Struct Reference

Estrutura que representa uma antena com tipo de Frequência, linha de posição, coluna de posição e apontador de saída.

Public Attributes

- char **freq**
- int **linha**
- int **coluna**
- struct **Antena** * **next**

Detailed Description

Estrutura que representa uma antena com tipo de Frequência, linha de posição, coluna de posição e apontador de saída.

Member Data Documentation

int Antena::coluna

Coluna de posição da antena.

char Antena::freq

Frequência / Tipo de antena.

int Antena::linha

Linha de posição da antena.

struct Antena* Antena::next

Apontador de saída para acesso às linked list de antenas

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

antena.h

Node Struct Reference

Estrutura de um nó da lista ligada de efeitos nefastos.

Public Attributes

- `int linha`
- `int coluna`
- `struct Node * next`

Detailed Description

Estrutura que representa uma posição na matriz afetada por um efeito nefasto causado por interferência entre antenas. Contém as coordenadas (linha e coluna) da posição e um ponteiro `next` para formar uma lista ligada.

Member Data Documentation

`int Node::coluna`

Coluna de posição da antena.

`int Node::linha`

Linha de posição da antena.

`struct Node* Node::next`

Apontador de saída para acesso às linked list.

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

`matriz.h`

File Documentation

antena.c File Reference

Functions

- **Antena * criarAntena** (char freq, int linha, int coluna)
Cria uma nova antena com os dados fornecidos.
- void **inserirAntena** (**Antena **aux**, char freq, int linha, int coluna)
Insere uma nova antena no final da lista ligada de antenas.
- void **removerAntena** (**Antena **aux**, int linha, int coluna)
Remove uma antena localizada numa posição específica da matriz.
- void **listarAntenas** (**Antena *aux**)
Lista todas as antenas presentes na lista ligada.
- void **libertarAntenas** (**Antena *aux**)
Liberta a memória alocada para a lista de antenas.

Detailed Description

criarAntena: Aloca dinamicamente memória para uma nova antena, inicializa os seus campos e retorna um ponteiro para ela;

inserirAntena: Insere uma nova antena no fim da lista ligada de antenas;

removerAntena: Remove da lista a antena que está na posição (linha, coluna);

listarAntenas: Percorre a lista ligada de antenas e imprime os seus dados (frequência, linha e coluna) numa tabela;

libertarAntenas: Liberta toda a memória alocada para a lista de antenas.

Este ficheiro implementa todas as operações sobre a estrutura de dados Antena, representando uma lista ligada simples de antenas com frequência e posição. Inclui funcionalidades para criação, inserção, remoção, listagem e libertação de memória associada às antenas.

Function Documentation

void listarAntenas (Antena * aux)

Lista todas as antenas presentes na lista ligada.

Parameters

<i>aux</i>	Apontador para a lista de antenas
------------	-----------------------------------

Definition at line **90** of file **antena.c**.

Antena * criarAntena (char freq, int linha, int coluna)

Cria uma nova antena com os dados fornecidos.

Função que cria uma nova antena.

Parameters

<i>freq</i>	Char que representa a frequência/tipo de antena
<i>linha</i>	Linha da matriz onde a antena será colocada
<i>coluna</i>	Coluna da matriz onde a antena será colocada

Returns

Antena* Apontador para a nova antena criada

Definition at line **26** of file **antena.c**.

void inserirAntena (Antena ** aux, char freq, int linha, int coluna)

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

Função que insere uma nova antena na lista.

Parameters

<i>aux</i>	Apontador para o apontador da lista de antenas
<i>freq</i>	Char que representa a frequência/tipo de antena
<i>linha</i>	Linha onde será inserida
<i>coluna</i>	Coluna onde será inserida

Definition at line **48** of file **antena.c**.

void libertarAntenas (Antena * aux)

Liberta a memória alocada para a lista de antenas.

Liberta a memória ocupada por todas as antenas da lista.

Parameters

<i>aux</i>	Apontador para a lista de antenas
------------	-----------------------------------

Definition at line **106** of file **antena.c**.

void removerAntena (Antena ** aux, int linha, int coluna)

Remove uma antena localizada numa posição específica da matriz.

Função que remove uma antena localizada numa posição específica da matriz.

Parameters

<i>aux</i>	Apontador para o apontador da lista de antenas
<i>linha</i>	Linha da antena a remover
<i>coluna</i>	Coluna da antena a remover

Definition at line **69** of file **antena.c**.

antena.c

Go to the documentation of this file.

```
00001
00011
00012 #include <stdio.h>
00013 #include <stdlib.h>
00014 #include <locale.h>
00015 #include "antena.h"
00016
00017
00026 Antena *criarAntena(char freq, int linha, int coluna) {
00027     Antena *nova = (Antena *)malloc(sizeof(Antena));
00028     if (!nova) {
00029         perror("Erro ao alocar memória");
00030         exit(1);
00031     }
00032     nova->freq = freq;
00033     nova->linha = linha;
00034     nova->coluna = coluna;
00035     nova->next = NULL;
00036     return nova;
00037 }
00038
00039
00048 void inserirAntena(Antena **aux, char freq, int linha, int coluna) {
00049     Antena *nova = criarAntena(freq, linha, coluna);
00050     nova->next = NULL;
00051     if (*aux == NULL) {
00052         *aux = nova;
00053     } else {
00054         Antena *atual = *aux;
00055         while (atual->next != NULL)
00056             atual = atual->next;
00057         atual->next = nova;
00058     }
00059 }
00060
00061
00069 void removerAntena(Antena **aux, int linha, int coluna) {
00070     Antena *atual = *aux, *anterior = NULL;
00071     while (atual) {
00072         if (atual->linha == linha && atual->coluna == coluna) {
00073             if (anterior) anterior->next = atual->next;
00074             else *aux = atual->next;
00075             free(atual);
00076             return;
00077         }
00078         anterior = atual;
00079         atual = atual->next;
00080     }
00081     printf("Antena em (%d, %d) não encontrada.\n", linha + 1, coluna + 1);
00082 }
00083
00084
00090 void auxrAntenas(Antena *aux) {
00091     setlocale(LC_ALL, "C");
00092     printf("\n%-12s %-10s %-10s\n", "Frequencia", "Linha", "Coluna");
00093     printf("%-12s %-10s %-10s\n", "-----", "-----", "-----");
00094     while (aux) {
00095         printf("%-12c %-10d %-10d\n", aux->freq, aux->linha + 1, aux->coluna + 1);
00096         aux = aux->next;
00097     }
00098 }
00099
00100
00106 void libertarAntenas(Antena *aux) {
00107     Antena *temp;
00108     while (aux) {
00109         temp = aux;
00110         aux = aux->next;
00111         free(temp);
00112     }
}
```

00113 }

antena.h File Reference

Classes

struct **Antena**: *Estrutura que representa uma antena.*

Typedefs

- typedef struct Antena **Antena**
Estrutura que representa uma antena.

Functions

- **Antena * criarAntena** (char freq, int linha, int coluna)
Função que cria uma nova antena.
- void **inserirAntena** (**Antena **aux**, char freq, int linha, int coluna)
Função que insere uma nova antena na lista.
- void **removerAntena** (**Antena **aux**, int linha, int coluna)
Função que remove uma antena localizada numa posição específica da matriz.
- void **listarAntenas** (**Antena *aux**)
Lista todas as antenas presentes na lista ligada.
- void **libertarAntenas** (**Antena *aux**)
Liberta a memória ocupada por todas as antenas da lista.

Detailed Description

Define a estrutura Antena e os protótipos das funções associadas, permitindo a gestão de uma lista ligada de antenas. Este ficheiro serve como interface pública para manipular dinamicamente as antenas na matriz.

Typedef Documentation

typedef struct Antena Antena

Estrutura que representa uma antena.

Function Documentation

Antena * criarAntena (char freq, int linha, int coluna)

Função que cria uma nova antena.

Parameters

<i>freq</i>	Char que representa a frequência/tipo de antena
<i>linha</i>	Linha da matriz onde a antena será colocada
<i>coluna</i>	Coluna da matriz onde a antena será colocada

Returns

Antena* Apontador para a nova antena criada

Função que cria uma nova antena.

Parameters

<i>freq</i>	Char que representa a frequência/tipo de antena
<i>linha</i>	Linha da matriz onde a antena será colocada
<i>coluna</i>	Coluna da matriz onde a antena será colocada

Returns

Antena* Apontador para a nova antena criada

Definition at line **26** of file **antena.c**.

void inserirAntena (Antena ** aux, char freq, int linha, int coluna)

Função que insere uma nova antena na lista.

Parameters

<i>aux</i>	Apontador para o apontador da lista de antenas
<i>freq</i>	Char que representa a frequência/tipo da nova antena
<i>linha</i>	Linha onde será inserida
<i>coluna</i>	Coluna onde será inserida

Função que insere uma nova antena na lista.

Parameters

<i>aux</i>	Apontador para o apontador da lista de antenas
<i>freq</i>	Char que representa a frequência/tipo de antena
<i>linha</i>	Linha onde será inserida
<i>coluna</i>	Coluna onde será inserida

Definition at line **48** of file **antena.c**.

void libertarAntenas (Antena * aux)

Liberta a memória ocupada por todas as antenas da lista.

Parameters

<i>aux</i>	Apontador para a lista de antenas.
------------	------------------------------------

Liberta a memória ocupada por todas as antenas da lista.

Parameters

<i>aux</i>	Apontador para a lista de antenas
------------	-----------------------------------

Definition at line **106** of file **antena.c**.

void listarAntenas (Antena * aux)

Lista todas as antenas presentes na lista ligada.

Parameters

<i>aux</i>	Apontador para a lista de antenas
------------	-----------------------------------

void removerAntena (Antena ** aux, int linha, int coluna)

Função que remove uma antena localizada numa posição específica da matriz.

Parameters

<i>lista</i>	Apontador para o apontador da lista de antenas
<i>linha</i>	Linha da antena a remover
<i>coluna</i>	Coluna da antena a remover

Função que remove uma antena localizada numa posição específica da matriz.

Parameters

<i>aux</i>	Apontador para o apontador da lista de antenas
<i>linha</i>	Linha da antena a remover
<i>coluna</i>	Coluna da antena a remover

Definition at line **69** of file **antena.c**.

antena.h

Go to the documentation of this file.

```
00001
00011
00012 #ifndef ANTENA_H
00013 #define ANTENA_H
00014
00015
00020 typedef struct Antena {
00021     char freq;
00022     int linha, coluna;
00023     struct Antena *next;
00024 } Antena;
00025
00026
00035 Antena *criarAntena(char freq, int linha, int coluna);
00036
00037
00046 void inserirAntena(Antena **aux, char freq, int linha, int coluna);
00047
00048
00056 void removerAntena(Antena **aux, int linha, int coluna);
00057
00058
00064 void listarAntenas(Antena *aux);
00065
00066
00072 void libertarAntenas(Antena *aux);
00073
00074 #endif
```

main.c File Reference

Functions

- `int main ()`
Função principal do programa. Exibe menu para o utilizador interagir com a matriz e antenas.

Detailed Description

Função principal que controla o ciclo do programa com um menu interativo para o utilizador inserir, remover antenas e/ou visualizar os resultados e sair.

Function Documentation

`int main ()`

Função principal do programa. Exibe menu para o utilizador interagir com a matriz e antenas.

Permite carregar matriz de ficheiro, inserir/remover antenas, visualizar matriz e os efeitos nefastos

Returns

`int` Código de retorno do programa

Definition at line **25** of file **main.c**.

main.c

Go to the documentation of this file.

```
00001
00011
00012 #include <stdio.h>
00013 #include <stdlib.h>
00014 #include "antena.h"
00015 #include "matriz.h"
00016
00017
00025 int main() {
00026
00027     Antena *listaAntenas = NULL;
00028     int linhas = 0, colunas = 0;
00029     int opcao;
00030     char nomeFicheiro[100];
00031
00032     do {
00033
00034         printf("\n1. Load da matriz de ficheiro\n2. Inserir antena\n3. Remover
00035 antena\n4. Exibir Matriz\n5. Sair\nEscolha: ");
00036         scanf("%d", &opcao);
00037
00038         if (opcao == 1) {
00039
00039             printf("Nome do ficheiro: ");
00040             scanf("%s", nomeFicheiro);
00041             libertarAntenas(listaAntenas);
00042             listaAntenas = NULL;
00043             carregarMatrizDoFicheiro(nomeFicheiro, &listaAntenas, &linhas,
00044 &colunas);
00045
00046         } else if (opcao == 2) {
00047
00047             char freq;
00048             int linha, coluna;
00049
00050             printf("Frequencia: ");
00051             scanf(" %c", &freq);
00052             printf("Linha: ");
00053             scanf("%d", &linha);
00054             printf("Coluna: ");
00055             scanf("%d", &coluna);
00056             inserirAntena(&listaAntenas, freq, linha - 1, coluna - 1);
00057
00058         } else if (opcao == 3) {
00059
00060             int linha, coluna;
00061
00062             printf("Linha: ");
00063             scanf("%d", &linha);
00064             printf("Coluna: ");
00065             scanf("%d", &coluna);
00066             removerAntena(&listaAntenas, linha - 1, coluna - 1);
00067
00068         } else if (opcao == 4) {
00069
00070             exibirMatriz(listaAntenas, linhas, colunas);
00071         }
00072
00073     } while (opcao != 5);
00074
00075     libertarAntenas(listaAntenas);
00076
00077     return 0;
00078 }
```

matriz.c File Reference

Functions

- void **inserirNefasto** (**Node** **aux, int linha, int coluna)
Insere uma posição nefasta na lista ligada.
- void **exibirMatriz** (**Antena** *listaAntenas, int linhas, int colunas)
Exibe a matriz no terminal, com as antenas e os efeitos nefastos causados por pares de antenas com a mesma frequência.
- void **gerarEfeitosNefastos** (**Antena** *listaAntenas, char matriz[][100], int linhas, int colunas)
Gera efeitos nefastos causados por pares de antenas com a mesma frequência.
- void **carregarMatrizDoFicheiro** (const char *nomeFicheiro, **Antena** **aux, int *linhas, int *colunas)
Carrega uma matriz de um ficheiro, criando a lista de antenas a partir dos dados encontrados.
- void **listarNefastos** (**Node** *aux)
Lista todas as posições afetadas por efeitos nefastos.

Detailed Description

Contém as funções responsáveis pela gestão e visualização da matriz, incluindo o carregamento de antenas a partir de ficheiros, o cálculo de efeitos nefastos e a sua representação visual. Também gere a memória associada aos nós das posições nefastas.

Function Documentation

void carregarMatrizDoFicheiro (const char * nomeFicheiro, **Antena** ** aux, int * linhas, int * colunas)

Carrega uma matriz de um ficheiro, criando a lista de antenas a partir dos dados encontrados.

Parameters

<i>nomeFicheiro</i>	Path do ficheiro a ser carregado
<i>aux</i>	Apontador para a lista onde as antenas serão armazenadas
<i>linhas</i>	Apontador para armazenar o número de linhas da matriz carregada
<i>colunas</i>	Apontador para armazenar o número de colunas da matriz carregada

Definition at line **130** of file **matriz.c**.

void exibirMatriz (**Antena** * listaAntenas, int linhas, int colunas)

Exibe a matriz no terminal, com as antenas e os efeitos nefastos causados por pares de antenas com a mesma frequência.

Parameters

<i>listaAntenas</i>	Lista de antenas a serem exibidas
<i>linhas</i>	Número de linhas da matriz
<i>colunas</i>	Número de colunas da matriz

Definition at line **45** of file **matriz.c**.

void gerarEfeitosNefastos (Antena * listaAntenas, char matriz[][100], int linhas, int colunas)

Gera efeitos nefastos causados por pares de antenas com a mesma frequência.

Parameters

<i>listaAntenas</i>	Apontador Lista de antenas
<i>matriz</i>	Matriz onde serão marcados os efeitos ("#")
<i>linhas</i>	Número de linhas da matriz
<i>colunas</i>	Número de colunas da matriz

Definition at line **94** of file **matriz.c**.

void inserirNefasto (Node ** aux, int linha, int coluna)

Insere uma posição nefasta na lista ligada.

Parameters

<i>aux</i>	Apontador para o apontador da lista de efeitos nefastos
<i>linha</i>	Linha da célula afetada
<i>coluna</i>	Coluna da célula afetada

Definition at line **26** of file **matriz.c**.

void listarNefastos (Node * aux)

Lista todas as posições afetadas por efeitos nefastos.

Parameters

<i>aux</i>	Lista ligada de posições nefastas
------------	-----------------------------------

Definition at line **159** of file **matriz.c**.

matriz.c

Go to the documentation of this file.

```
00001
00011
00012 #include <stdio.h>
00013 #include <stdlib.h>
00014 #include <string.h>
00015 #include <locale.h>
00016 #include "matriz.h"
00017
00018
00026 void inserirNefasto(Node **aux, int linha, int coluna) {
00027     for (Node *n = *aux; n != NULL; n = n->next)
00028         if (n->linha == linha && n->coluna == coluna) return;
00029
00030     Node *novo = (Node *)malloc(sizeof(Node));
00031     novo->linha = linha;
00032     novo->coluna = coluna;
00033     novo->next = *aux;
00034     *aux = novo;
00035 }
00036
00037
00045 void exibirMatriz(Antena *listaAntenas, int linhas, int colunas) {
00046     char matriz[100][100]; // Máximo
00047
00048     // Inicializar matriz com '.'
00049     for (int i = 0; i < linhas; i++)
00050         for (int j = 0; j < colunas; j++)
00051             matriz[i][j] = '.';
00052
00053     // Antenas na matriz
00054     for (Antena *a = listaAntenas; a != NULL; a = a->next)
00055         matriz[a->linha][a->coluna] = a->freq;
00056
00057     // Gerar efeitos nefastos
00058     gerarEfeitosNefastos(listaAntenas, matriz, linhas, colunas);
00059
00060     // Exibir matriz
00061     //setlocale(LC_ALL, "C");
00062     printf("\nMatriz:\n");
00063     for (int i = 0; i < linhas; i++) {
00064         for (int j = 0; j < colunas; j++)
00065             printf("%c", matriz[i][j]);
00066         printf("\n");
00067     }
00068
00069     // Listar antenas
00070     printf("\nAntenas:\n");
00071     for (Antena *a = listaAntenas; a != NULL; a = a->next)
00072         printf("Frequencia: %c | Linha: %d | Coluna: %d\n", a->freq, a->linha + 1,
a->coluna + 1);
00073
00074     // Listar efeitos nefastos
00075     printf("\nEfeitos nefastos:\n");
00076     for (int i = 0; i < linhas; i++) {
00077         for (int j = 0; j < colunas; j++) {
00078             if (matriz[i][j] == '#') {
00079                 printf("Efeito\t\t\t Linha: %d\t\t\t Coluna: %d\n", i + 1, j + 1);
00080             }
00081         }
00082     }
00083 }
00084
00085
00094 void gerarEfeitosNefastos(Antena *listaAntenas, char matriz[][100], int linhas, int
colunas) {
00095     for (Antena *a1 = listaAntenas; a1 != NULL; a1 = a1->next) {
00096         for (Antena *a2 = a1->next; a2 != NULL; a2 = a2->next) {
00097             if (a1->freq == a2->freq) {
00098                 int dx = a2->linha - a1->linha;
00099                 int dy = a2->coluna - a1->coluna;
```

```

00100
00101 // Geração antes da primeira antena
00102 int linhaAntes = a1->linha - dx;
00103 int colunaAntes = a1->coluna - dy;
00104 if (linhaAntes >= 0 && linhaAntes < linhas &&
00105     colunaAntes >= 0 && colunaAntes < colunas) {
00106     matriz[linhaAntes][colunaAntes] = '#';
00107 }
00108
00109 // Geração depois da segunda antena
00110 int linhaDepois = a2->linha + dx;
00111 int colunaDepois = a2->coluna + dy;
00112 if (linhaDepois >= 0 && linhaDepois < linhas &&
00113     colunaDepois >= 0 && colunaDepois < colunas) {
00114     matriz[linhaDepois][colunaDepois] = '#';
00115 }
00116 }
00117 }
00118 }
00119 }
00120
00121
00130 void carregarMatrizDoFicheiro(const char *nomeFicheiro, Antena **aux, int *linhas,
int *colunas) {
00131     FILE *file = fopen(nomeFicheiro, "r");
00132     if (!file) {
00133         perror("Erro ao abrir ficheiro");
00134         exit(1);
00135     }
00136
00137     char linha[256];
00138     int i = 0, maxCol = 0;
00139     while (fgets(linha, sizeof(linha), file)) {
00140         int j;
00141         for (j = 0; linha[j] != '\0' && linha[j] != '\n'; j++) {
00142             if (linha[j] != '.')
00143                 inserirAntena(aux, linha[j], i, j);
00144         }
00145         if (j > maxCol) maxCol = j;
00146         i++;
00147     }
00148     *linhas = i;
00149     *colunas = maxCol;
00150     fclose(file);
00151 }
00152
00153
00159 void listarNefastos(Node *aux) {
00160     //setlocale(LC_ALL, "C");
00161     printf("%-10s %-10s\n", "Linha", "Coluna");
00162     printf("%-10s %-10s\n", "-----", "-----");
00163     for (; aux; aux = aux->next)
00164         printf("%-10d %-10d\n", aux->linha, aux->coluna);
00165 }
00166
00167

```

matriz.h File Reference

Classes

struct **Node**: *Estrutura de um nó da lista ligada de efeitos nefastos.*

Typedefs

- typedef struct Node **Node**
Estrutura de um nó da lista ligada de efeitos nefastos.

Functions

- void **exibirMatriz** (**Antena** *listaAntenas, int linhas, int colunas)
Exibe a matriz no terminal, com as antenas e os efeitos nefastos causados por pares de antenas com a mesma frequência.
- void **gerarEfeitosNefastos** (**Antena** *listaAntenas, char matriz[][100], int linhas, int colunas)
Gera efeitos nefastos causados por pares de antenas com a mesma frequência.
- void **carregarMatrizDoFicheiro** (const char *nomeFicheiro, **Antena** **listaAntenas, int *linhas, int *colunas)
Carrega uma matriz de um ficheiro, criando a lista de antenas a partir dos dados encontrados.
- void **listarNefastos** (**Node** *lista)
Lista todas as posições afetadas por efeitos nefastos.

Detailed Description

Define a estrutura **Nodo** usada para representar posições afetadas por interferência e declara as funções que interagem com a matriz e com os efeitos nefastos. Atua como interface funcional entre a matriz e a lógica do programa.

Typedef Documentation

typedef struct Node Node

Estrutura de um nó da lista ligada de efeitos nefastos.

Function Documentation

void carregarMatrizDoFicheiro (const char * nomeFicheiro, **Antena** ** aux, int * linhas, int * colunas)

Carrega uma matriz de um ficheiro, criando a lista de antenas a partir dos dados encontrados.

Parameters

<i>nomeFicheiro</i>	Path do ficheiro a ser carregado
<i>listaAntenas</i>	Apontador para a lista de antenas
<i>linhas</i>	Número de linhas da matriz
<i>colunas</i>	Número de colunas da matriz
<i>nomeFicheiro</i>	Path do ficheiro a ser carregado
<i>aux</i>	Apontador para a lista onde as antenas serão armazenadas
<i>linhas</i>	Apontador para armazenar o número de linhas da matriz carregada
<i>colunas</i>	Apontador para armazenar o número de colunas da matriz carregada

Definition at line **130** of file **matriz.c**.

void exhibirMatriz (Antena * listaAntenas, int linhas, int colunas)

Exibe a matriz no terminal, com as antenas e os efeitos nefastos causados por pares de antenas com a mesma frequência.

Parameters

<i>listaAntenas</i>	Lista de antenas a serem exibidas
<i>linhas</i>	Número de linhas da matriz
<i>colunas</i>	Número de colunas da matriz

Definition at line **45** of file **matriz.c**.

void gerarEfeitosNefastos (Antena * listaAntenas, char matriz[][100], int linhas, int colunas)

Gera efeitos nefastos causados por pares de antenas com a mesma frequência.

Parameters

<i>listaAntenas</i>	Apontador Lista de antenas
<i>matriz</i>	Matriz onde serão marcados os efeitos ("#")
<i>linhas</i>	Número de linhas da matriz
<i>colunas</i>	Número de colunas da matriz

Definition at line **94** of file **matriz.c**.

void listarNefastos (Node * aux)

Lista todas as posições afetadas por efeitos nefastos.

Parameters

<i>lista</i>	Apontador para a lista de efeitos nefastos
<i>aux</i>	Lista ligada de posições nefastas

Definition at line **159** of file **matriz.c**.

matriz.h

Go to the documentation of this file.

```
00001
00011
00012 #ifndef MATRIZ_H
00013 #define MATRIZ_H
00014
00015 #include "antena.h"
00016
00017
00022 typedef struct Node {
00023     int linha, coluna;
00024     struct Node *next;
00025 } Node;
00026
00027
00035 void exibirMatriz(Antena *listaAntenas, int linhas, int colunas);
00036
00045 void gerarEfeitosNefastos(Antena *listaAntenas, char matriz[][100], int linhas, int
colunas);
00046
00055 void carregarMatrizDoFicheiro(const char *nomeFicheiro, Antena **listaAntenas, int
*linhas, int *colunas);
00056
00062 void listarNefastos(Node *lista);
00063
00064
00065
00066 #endif
```

Index

Antena, 4
 antena.h, 10
 coluna, 4
 freq, 4
 linha, 4
 next, 4
antena.c
 auxrAntenas, 6
 criarAntena, 7
 inserirAntena, 7
 libertarAntenas, 7
 removerAntena, 7
antena.h, 10
 Antena, 10
 criarAntena, 10
 inserirAntena, 11
 libertarAntenas, 11
 listarAntenas, 11
 removerAntena, 12
auxrAntenas
 antena.c, 6
carregarMatrizDoFicheiro
 matriz.c, 16
 matriz.h, 20
coluna
 Antena, 4
 Node, 5
criarAntena
 antena.c, 7
 antena.h, 10
exibirMatriz
 matriz.c, 16
 matriz.h, 21
freq
 Antena, 4
gerarEfeitosNefastos
 matriz.c, 17
 matriz.h, 21
inserirAntena
 antena.c, 7
 antena.h, 11
inserirNefasto
 matriz.c, 17
libertarAntenas
 antena.c, 7
 antena.h, 11
linha
 Antena, 4
 Node, 5
listarAntenas
 antena.h, 11
listarNefastos
 matriz.c, 17
 matriz.h, 21
main
 main.c, 14
main.c
 main, 14
matriz.c
 carregarMatrizDoFicheiro, 16
 exibirMatriz, 16
 gerarEfeitosNefastos, 17
 inserirNefasto, 17
 listarNefastos, 17
matriz.h
 carregarMatrizDoFicheiro, 20
 exibirMatriz, 21
 gerarEfeitosNefastos, 21
 listarNefastos, 21
 Node, 20
next
 Antena, 4
 Node, 5
Node, 5
 coluna, 5
 linha, 5
 matriz.h, 20
 next, 5
removerAntena
 antena.c, 7
 antena.h, 12