

Recursividad, Salir del LAberinto

1.0

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	_lista Struct Reference . . . . .	5
3.2	_matriz Struct Reference . . . . .	5
3.3	_posicion Struct Reference . . . . .	6
<b>4</b>	<b>File Documentation</b>	<b>7</b>
4.1	main.c File Reference . . . . .	7
4.2	sourcecode/Laberinto.c File Reference . . . . .	7
4.2.1	Detailed Description . . . . .	8
4.2.2	Function Documentation . . . . .	8
4.2.2.1	CrearTablero() . . . . .	8
4.2.2.2	EscribirCaracter() . . . . .	9
4.2.2.3	Esperar() . . . . .	9
4.2.2.4	GuardarLaberinto() . . . . .	9
4.2.2.5	HacerCamino() . . . . .	9
4.2.2.6	ImprimirMatriz() . . . . .	10
4.2.2.7	LlenarGatos() . . . . .	10
4.2.2.8	MePuedoMover() . . . . .	10
4.3	sourcecode/lista.c File Reference . . . . .	11

4.3.1	Detailed Description	11
4.3.2	Function Documentation	11
4.3.2.1	AgregarElemento()	11
4.3.2.2	CrearLista()	12
4.3.2.3	EliminarLista()	12
4.3.2.4	ImprimirLista()	12
4.4	sourcecode/posicion.c File Reference	12
4.4.1	Detailed Description	13
4.4.2	Function Documentation	13
4.4.2.1	Anterior()	13
4.4.2.2	CrearPosicion()	13
4.4.2.3	EliminarPosicion()	14
4.4.2.4	Siguiente()	14
4.5	sourcecode/Recorrer.c File Reference	14
4.5.1	Detailed Description	15
4.5.2	Function Documentation	15
4.5.2.1	buscarQueso()	15
4.5.2.2	LlenarLaberinto()	15
4.5.2.3	LlenarTablero()	15
4.5.2.4	UbicarRaton()	17
4.5.2.5	VerificarLaberinto()	17
<b>Index</b>		<b>19</b>

# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">_lista</a>	.....	5
<a href="#">_matriz</a>	.....	5
<a href="#">_posicion</a>	.....	6



## Chapter 2

# File Index

### 2.1 File List

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

<a href="#">main.c</a>	Archivo principal, laboratorio que trata sobre listas enlazadas . . . . .	7
include/ <b>includes.h</b>	. . . . .	??
include/ <b>Laberinto.h</b>	. . . . .	??
include/ <b>lista.h</b>	. . . . .	??
include/ <b>posicion.h</b>	. . . . .	??
include/ <b>Recorrer.h</b>	. . . . .	??
sourcecode/ <a href="#">Laberinto.c</a>	Archivo que contiene las funciones para Crear un laberinto . . . . .	7
sourcecode/ <a href="#">lista.c</a>	Archivo que contiene las funciones para manipular listas . . . . .	11
sourcecode/ <a href="#">posicion.c</a>	Archivo que contiene las funciones para manipular las posiciones de una lista . . . . .	12
sourcecode/ <a href="#">Recorrer.c</a>	Archivo que contiene las funciones para Recorrer un laberinto . . . . .	14





## Chapter 3

# Class Documentation

### 3.1 `_lista` Struct Reference

Collaboration diagram for `_lista`:

#### Public Attributes

- unsigned int **items**
- `posicion` \* **primero**

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

- `include/lista.h`

### 3.2 `_matriz` Struct Reference

Collaboration diagram for `_matriz`:

#### Public Attributes

- char \*\* **tablero**
- int **filas**
- int **columnas**
- int **semilla**
- int **posicion** [2]
- `lista` \* **pasos**

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

- `include/includes.h`

### 3.3 `_posicion` Struct Reference

Collaboration diagram for `_posicion`:

#### Public Attributes

- `int d`
- `struct _posicion * siguiente`

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

- `include/posicion.h`

## Chapter 4

# File Documentation

### 4.1 main.c File Reference

Archivo principal, laboratorio que trata sobre listas enlazadas.

```
#include "../include/includes.h"
```

Include dependency graph for main.c:

### 4.2 sourcecode/Laberinto.c File Reference

Archivo que contiene las funciones para Crear un laberinto.

```
#include "../include/includes.h"
```

Include dependency graph for Laberinto.c:

#### Macros

- `#define ANSI_COLOR_RED "\x1b[31m"`
- `#define ANSI_COLOR_GREEN "\x1b[32m"`
- `#define ANSI_COLOR_YELLOW "\x1b[33m"`
- `#define ANSI_COLOR_BLUE "\x1b[34m"`
- `#define ANSI_COLOR_MAGENTA "\x1b[35m"`
- `#define ANSI_COLOR_CYAN "\x1b[36m"`
- `#define ANSI_COLOR_RESET "\x1b[0m"`
- `#define ANSI_COLOR_BOLD_YELLOW "\x1b[01;33m"`
- `#define ANSI_COLOR_BOLD_GREEN "\x1b[01;32m"`

## Functions

- char \*\* [CrearTablero](#) (int filas, int columnas)  
*Funcion que crea una matriz del tamaño deseado por el usuario.*
- void [Esperar](#) (float segundos)  
*Funcion que detiene la ejecucion cierta cantidad de segundos.*
- void [LlenarGatos](#) (matriz \*laberinto)  
*Funcion que llena una matriz de simbolos de # exeptuando las filas y columnas externas las cuales llena con espacio vacío.*
- void [ImprimirMatriz](#) (matriz \*laberinto)  
*Funcion que permite imprimir una matriz.*
- void [EscribirCaracter](#) (matriz \*laberinto, char caracter)  
*Funcion que escribe un caracter en una coordenada indicada.*
- void [GuardarLaberinto](#) (matriz \*laberinto)  
*Funcion que guarda una matriz en un archivo.*
- void [HacerCamino](#) (matriz \*laberinto)  
*Funcion que crea el camino principal es decir el camino ganador aleatoriamente.*
- int [MePuedoMover](#) (matriz \*laberinto, int lado)  
*Funcion que verifica que se cumplan los requisitos para moverse.*

### 4.2.1 Detailed Description

Archivo que contiene las funciones para Crear un laberinto.

#### Author

Jesus Zuñiga Mendez

### 4.2.2 Function Documentation

#### 4.2.2.1 CrearTablero()

```
char** CrearTablero (
    int filas,
    int columnas )
```

Funcion que crea una matriz del tamaño deseado por el usuario.

#### Parameters

<i>filas</i>	es el tamaño de filas de la matriz
<i>columnas</i>	es el tamaño de columnas de la matriz

#### Returns

devuelve un puntero con la matriz creada

#### 4.2.2.2 EscribirCaracter()

```
void EscribirCaracter (
    matriz * laberinto,
    char caracter )
```

Funcion que escribe un caracter en una coordenada indicada.

##### Parameters

<i>laberinto</i>	es la estructura tipo matriz que contiene la matriz de caracteres
<i>caracter</i>	es el caracter que se quiere escribir

#### 4.2.2.3 Esperar()

```
void Esperar (
    float segundos )
```

Funcion que detiene la ejecucion cierta cantidad de segundos.

##### Parameters

<i>tiempo</i>	el el tiempo en espera en segundos codigo recuperado de <a href="https://stackoverflow.com/questions/7684359/how-to-use-nanosleep-in-c-what-are-tim-tv-sec-and-tim-tv-n">https://stackoverflow.com/questions/7684359/how-to-use-nanosleep-in-c-what-are-tim-tv-sec-and-tim-tv-n</a> presentado con modificaciones
---------------	---

#### 4.2.2.4 GuardarLaberinto()

```
void GuardarLaberinto (
    matriz * laberinto )
```

Funcion que guarda una matriz en un archivo.

##### Parameters

<i>laberinto</i>	es la estructura tipo matriz que contiene la matriz de caracteres
------------------	---

#### 4.2.2.5 HacerCamino()

```
void HacerCamino (
```

```
matriz * laberinto )
```

Funcion que crea el camino principal es decir el camino ganador aleatoriamente.

#### Parameters

<i>laberinto</i>	es la estructura tipo matriz que contiene la matriz de caracteres
------------------	---

#### 4.2.2.6 ImprimirMatriz()

```
void ImprimirMatriz (
    matriz * laberinto )
```

Funcion que permite imprimir una matriz.

#### Parameters

<i>recibe</i>	un puntero a la matriz
---------------	------------------------

#### 4.2.2.7 LlenarGatos()

```
void LlenarGatos (
    matriz * laberinto )
```

Funcion que llena una matriz de simbolos de # exeptuando las filas y columnas externas las cuales llena con espacio vacio.

#### Parameters

<i>recibe</i>	un puntero a la estructura matriz
---------------	-----------------------------------

#### 4.2.2.8 MePuedoMover()

```
int MePuedoMover (
    matriz * laberinto,
    int lado )
```

Funcion que verifica que se cumplan los requisitos para moverse.

#### Parameters

<i>laberinto</i>	es la estructura tipo matriz que contiene la matriz de caracteres Qparam lado es la direccion, 0 = izquierda, 1= derecha, 2= arriba, 3= abajo
------------------	---

## 4.3 sourcecode/lista.c File Reference

Archivo que contiene las funciones para manipular listas.

```
#include "../include/includes.h"
```

Include dependency graph for lista.c:

### Functions

- `lista * CrearLista ()`  
*Funcion que crea una lista vacia.*
- `void EliminarLista (lista *ls)`  
*Funcion que elimina una lista.*
- `void AgregarElemento (lista *ls, int elemnto)`  
*Funcion que agrega un elemento a la lista.*
- `void ImprimirLista (lista *ls)`  
*Funcion que imprime una lista.*

### 4.3.1 Detailed Description

Archivo que contiene las funciones para manipular listas.

#### Author

Jesus Zuñiga Mendez

### 4.3.2 Function Documentation

#### 4.3.2.1 AgregarElemento()

```
void AgregarElemento (  
    lista * ls,  
    int elemnto )
```

Funcion que agrega un elemento a la lista.

#### Parameters

<i>ls</i>	lista con la que se va a trabajar
<i>elemnto</i>	elemento que se va a agregar

#### 4.3.2.2 CrearLista()

```
lista* CrearLista ( )
```

Funcion que crea una lista vacia.

##### Returns

Is es la lista que se acaba de crear;

#### 4.3.2.3 EliminarLista()

```
void EliminarLista (
    lista * ls )
```

Funcion que elimina una lista.

##### Parameters

/s	lista con la que se va a trabajar
----	-----------------------------------

#### 4.3.2.4 ImprimirLista()

```
void ImprimirLista (
    lista * ls )
```

Funcion que imprime una lista.

##### Parameters

/s	lista que se va a imprimir
----	----------------------------

## 4.4 sourcecode/posicion.c File Reference

Archivo que contiene las funciones para manipular las posiciones de una ñista.

```
#include "../include/includes.h"
```

Include dependency graph for posicion.c:

### Functions

- [posicion](#) \* [Siguiete](#) ([posicion](#) \*p)



- Funcion que devuelve la posicion siguiente de una lista.*
  - `posicion * Anterior (posicion *p, lista *ls)`
    - Funcion que devuelve la posicion anterior de una lista.*
  - `posicion * CrearPosision (int dato)`
    - Funcion que crea una posicion vacia.*
  - `void EliminarPosicion (posicion *p)`
    - Funcion que elimina una poicion.*

#### 4.4.1 Detailed Description

Archivo que contiene las funciones para manipular las posiciones de una ñista.

Author

Jesus Zuñiga Mendez

#### 4.4.2 Function Documentation

##### 4.4.2.1 Anterior()

```
posicion* Anterior (  
    posicion * p,  
    lista * ls )
```

Funcion que devuelve la posicion anterior de una lista.

Parameters

<i>p</i>	posicion inicial
<i>ls</i>	lista en la que se va a buscar

##### 4.4.2.2 CrearPosision()

```
posicion* CrearPosision (  
    int dato )
```

Funcion que crea una posicion vacia.

Parameters

<i>dato</i>	Es el dato que se ve a agregar
-------------	--------------------------------

#### 4.4.2.3 EliminarPosicion()

```
void EliminarPosicion (
    posicion * p )
```

Funcion que elimina una poicion.

##### Parameters

<i>p</i>	es el posicion que se ve a eliminar
----------	-------------------------------------

#### 4.4.2.4 Siguiete()

```
posicion* Siguiete (
    posicion * p )
```

Funcion que devuelve la posicion siguiente de una lista.

##### Parameters

<i>p</i>	posicion inicial
----------	------------------

## 4.5 sourcecode/Recorrer.c File Reference

Archivo que contiene las funciones para Recorrer un laberinto.

```
#include "../include/includes.h"
Include dependency graph for Recorrer.c:
```

### Functions

- void [LLenarLaberinto](#) ([matriz](#) \*laberinto, char \*\*Archivo)  
*Funcion que llena un laberinto con los datos del archivo sellecionado por el usuario.*
- void [UbicarRaton](#) ([matriz](#) \*laberinto)  
*Funcion que ubica la posicion inicial del raton.*
- void [buscarQueso](#) ([matriz](#) \*laberinto, int fila, int columna)  
*Funcion que ubica la posicion inicial del raton.*
- void [LlenarTablero](#) ([matriz](#) \*laberinto, char \*\*Archivo)  
*Funcion que llena el tablero con los datos del laberinto.*
- int [VerificarLaberinto](#) ([matriz](#) \*laberinto, char \*\*Archivo)  
*Funcion que revisa que el laberinto este escrito de forma correcta, ademas modifica el puntero del laberinto incluyendo las filas y columnas correpondientes.*

### 4.5.1 Detailed Description

Archivo que contiene las funciones para Recorrer un laberinto.

#### Author

Jesus Zuñiga Mendez

### 4.5.2 Function Documentation

#### 4.5.2.1 buscarQueso()

```
void buscarQueso (
    matriz * laberinto,
    int fila,
    int columna )
```

Funcion que ubica la posicion inicial del raton.

#### Parameters

<i>laberinto</i>	es la estructura que contiene todos los datos
------------------	---

#### 4.5.2.2 LLenarLaberinto()

```
void LLenarLaberinto (
    matriz * laberinto,
    char ** Archivo )
```

Funcion que llena un laberinto con los datos del archivo seleccionado por el usuario.

#### Parameters

<i>laberinto</i>	es la estructura que contiene todos los datos
<i>archivo</i>	es la direccion donde esta el archivo

#### 4.5.2.3 LlenarTablero()

```
void LlenarTablero (
    matriz * laberinto,
    char ** Archivo )
```

Funcion que llena el tablero con los datos del laberinto.

**Parameters**

<i>laberinto</i>	es la estructura que contiene todos los datos
<i>Archivo</i>	es el arreglo que contien la ruta del archivo

**4.5.2.4 UbicarRaton()**

```
void UbicarRaton (
    matriz * laberinto )
```

Funcion que ubica la posicion inicial del raton.

**Parameters**

<i>laberinto</i>	es la estructura que contiene todos los datos
------------------	---

**4.5.2.5 VerificarLaberinto()**

```
int VerificarLaberinto (
    matriz * laberinto,
    char ** Archivo )
```

Funcion que revisa que el laberinto este escrito de forma correcta, ademas modifica el puntero del laberinto incluyendo las filas y columnas correpondientes.

**Parameters**

<i>laberinto</i>	es la estructura que contiene todos los datos
<i>Archivo</i>	es el arreglo que contien la ruta del archivo

**Returns**

rspuesta es un indicador si el laberinto es correcto o no



# Index

- [\\_lista](#), [5](#)
  - [\\_matriz](#), [5](#)
  - [\\_posicion](#), [6](#)
- AgregarElemento
  - [lista.c](#), [11](#)
- Anterior
  - [posicion.c](#), [13](#)
- buscarQueso
  - [Recorrer.c](#), [15](#)
- CrearLista
  - [lista.c](#), [11](#)
- CrearPosicion
  - [posicion.c](#), [13](#)
- CrearTablero
  - [Laberinto.c](#), [8](#)
- EliminarLista
  - [lista.c](#), [12](#)
- EliminarPosicion
  - [posicion.c](#), [14](#)
- EscribirCaracter
  - [Laberinto.c](#), [9](#)
- Esperar
  - [Laberinto.c](#), [9](#)
- GuardarLaberinto
  - [Laberinto.c](#), [9](#)
- HacerCamino
  - [Laberinto.c](#), [9](#)
- ImprimirLista
  - [lista.c](#), [12](#)
- ImprimirMatriz
  - [Laberinto.c](#), [10](#)
- LLenarLaberinto
  - [Recorrer.c](#), [15](#)
- Laberinto.c
  - [CrearTablero](#), [8](#)
  - [EscribirCaracter](#), [9](#)
  - [Esperar](#), [9](#)
  - [GuardarLaberinto](#), [9](#)
  - [HacerCamino](#), [9](#)
  - [ImprimirMatriz](#), [10](#)
  - [LlenarGatos](#), [10](#)
  - [MePuedoMover](#), [10](#)
- [lista.c](#)
- [AgregarElemento](#), [11](#)
- [CrearLista](#), [11](#)
- [EliminarLista](#), [12](#)
- [ImprimirLista](#), [12](#)
- [LlenarGatos](#)
  - [Laberinto.c](#), [10](#)
- [LlenarTablero](#)
  - [Recorrer.c](#), [15](#)
- [main.c](#), [7](#)
- [MePuedoMover](#)
  - [Laberinto.c](#), [10](#)
- [posicion.c](#)
  - [Anterior](#), [13](#)
  - [CrearPosicion](#), [13](#)
  - [EliminarPosicion](#), [14](#)
  - [Siguiente](#), [14](#)
- [Recorrer.c](#)
  - [buscarQueso](#), [15](#)
  - [LLenarLaberinto](#), [15](#)
  - [LlenarTablero](#), [15](#)
  - [UbicarRaton](#), [17](#)
  - [VerificarLaberinto](#), [17](#)
- [Siguiente](#)
  - [posicion.c](#), [14](#)
- [sourcecode/Laberinto.c](#), [7](#)
- [sourcecode/Recorrer.c](#), [14](#)
- [sourcecode/lista.c](#), [11](#)
- [sourcecode/posicion.c](#), [12](#)
- [UbicarRaton](#)
  - [Recorrer.c](#), [17](#)
- [VerificarLaberinto](#)
  - [Recorrer.c](#), [17](#)