

Laboratorio 6

1

Generated by Doxygen 1.8.13

Contents

1	Data Structure Index	1
1.1	Data Structures	1
2	File Index	3
2.1	File List	3
3	Data Structure Documentation	5
3.1	_lista Struct Reference	5
3.2	_posicion Struct Reference	5
4	File Documentation	7
4.1	main.c File Reference	7
4.2	sourcecode/lista.c File Reference	7
4.2.1	Detailed Description	8
4.2.2	Function Documentation	8
4.2.2.1	AgregarElemento()	8
4.2.2.2	BuscarDato()	8
4.2.2.3	BuscarK()	8
4.2.2.4	CrearLista()	10
4.2.2.5	EliminarElemento()	10
4.2.2.6	EliminarLista()	10
4.2.2.7	Estirar()	11
4.2.2.8	ImprimirLista()	11
4.2.2.9	Vaciar()	11
4.3	sourcecode/posicion.c File Reference	12
4.3.1	Detailed Description	12
4.3.2	Function Documentation	12
4.3.2.1	Anterior()	12
4.3.2.2	CrearPosicion()	13
4.3.2.3	EliminarPosicion()	13
4.3.2.4	Siguiente()	13
	Index	15

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

_lista	5
_posicion	5

Chapter 2

File Index

2.1 File List

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

main.c	Archivo principal, laboratorio que trata sobre listas enlazadas	7
include/ includes.h	??
include/ lista.h	??
include/ posicion.h	??
sourcecode/ lista.c	Archivo que contiene las funciones para manipular listas	7
sourcecode/ posicion.c	Archivo que contiene las funciones para manipular las posiciones de una lista	12

Chapter 3

Data Structure Documentation

3.1 `_lista` Struct Reference

Collaboration diagram for `_lista`:

Data Fields

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

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

- `include/lista.h`

3.2 `_posicion` Struct Reference

Collaboration diagram for `_posicion`:

Data Fields

- 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/lista.c File Reference

Archivo que contiene las funciones para manipular listas.

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

Include dependency graph for lista.c:

Functions

- `lista ** Estirar (lista **listaAnterior, int tamanoViejo, int tamanoNuevo)`
Funcion que agranda la lista que contienen las demas listas.
- `lista * CrearLista ()`
Funcion que crea una lista vacia.
- `void EliminarLista (lista *ls)`
Funcion que elimina una lista.
- `void Vaciar (lista *ls)`
Funcion que llena una lista de ceros.
- `void AgregarElemento (lista *ls, int elemnto)`
Funcion que agrega un elemento a la lista.
- `void EliminarElemento (lista *ls, int elemento)`
Funcion que elimina un elemnto de la lista.
- `void ImprimirLista (lista *ls)`
Funcion que imprime una lista.
- `posicion * BuscarDato (lista *ls, int dato)`
Funcion que busca un dato en la lista.
- `int BuscarK (lista *ls, posicion *laPosicion)`
Funcion que busca una posicion dentro de la lista.

4.2.1 Detailed Description

Archivo que contiene las funciones para manipular listas.

Author

Jesus Zuñiga Mendez

4.2.2 Function Documentation

4.2.2.1 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>elemento</i>	elemento que se va a agregar

4.2.2.2 BuscarDato()

```
posicion* BuscarDato (
    lista * ls,
    int dato )
```

Funcion que busca un dato en la lista.

Parameters

<i>ls</i>	lista donde se va a buscar
<i>dato</i>	es el dato que se va a buscar

4.2.2.3 BuscarK()

```
int BuscarK (
    lista * ls,
    posicion * laPosicion )
```

Funcion que busca una posicion dentro de la lista.

Parameters

<i>ls</i>	lista donde se va a buscar
<i>laPosicion</i>	es la posicion que se va a buscar

4.2.2.4 CrearLista()

```
lista* CrearLista ( )
```

Funcion que crea una lista vacia.

Returns

ls es la lista que se acaba de crear;

4.2.2.5 EliminarElemento()

```
void EliminarElemento (
    lista * ls,
    int elemento )
```

Funcion que elimina un elemnto de la lista.

Parameters

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

4.2.2.6 EliminarLista()

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

Funcion que elimina una lista.

Parameters

<i>ls</i>	lista con la que se va a trabajar
-----------	-----------------------------------

4.2.2.7 Estirar()

```
lista** Estirar (
    lista ** listaAnterior,
    int  tamanoViejo,
    int  tamanoNuevo )
```

Funcion que agranda la lista que contienen las demas listas.

Returns

listaViejo es la lista que se acaba de crear con un espacio mas

Parameters

<i>listaAnterior</i>	es la lista original
<i>tamanoViejo</i>	es la cantidad de listas que existian antes de estirar
<i>TamanoNuevo</i>	es el nuevo numero de listas

4.2.2.8 ImprimirLista()

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

Funcion que imprime una lista.

Parameters

<i>ls</i>	lista que se va a imprimir
-----------	----------------------------

4.2.2.9 Vaciar()

```
void Vaciar (
    lista * ls )
```

Funcion que llena una lista de ceros.

Parameters

<i>ls</i>	lista con la que se va a trabajar
-----------	-----------------------------------

4.3 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 * Siguiente (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 * CrearPosicion (int dato)`
Funcion que crea una posicion vacia.
- `void EliminarPosicion (posicion *p)`
Funcion que elimina una poicion.

4.3.1 Detailed Description

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

Author

Jesus Zuñiga Mendez

4.3.2 Function Documentation

4.3.2.1 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

4.3.2.2 CrearPosision()

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

Funcion que crea una posicion vacia.

Parameters

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

4.3.2.3 EliminarPosicion()

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

Funcion que elimina una poicion.

Parameters

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

4.3.2.4 Siguiete()

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

Funcion que devuelve la posicion siguiente de una lista.

Parameters

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

Index

`_lista`, [5](#)

`_posicion`, [5](#)

AgregarElemento

`lista.c`, [8](#)

Anterior

`posicion.c`, [12](#)

BuscarDato

`lista.c`, [8](#)

BuscarK

`lista.c`, [8](#)

CrearLista

`lista.c`, [10](#)

CrearPosicion

`posicion.c`, [12](#)

EliminarElemento

`lista.c`, [10](#)

EliminarLista

`lista.c`, [10](#)

EliminarPosicion

`posicion.c`, [13](#)

Estirar

`lista.c`, [10](#)

ImprimirLista

`lista.c`, [11](#)

`lista.c`

AgregarElemento, [8](#)

BuscarDato, [8](#)

BuscarK, [8](#)

CrearLista, [10](#)

EliminarElemento, [10](#)

EliminarLista, [10](#)

Estirar, [10](#)

ImprimirLista, [11](#)

Vaciar, [11](#)

`main.c`, [7](#)

`posicion.c`

Anterior, [12](#)

CrearPosicion, [12](#)

EliminarPosicion, [13](#)

Siguiente, [13](#)

Siguiente

`posicion.c`, [13](#)

`sourcecode/lista.c`, [7](#)

`sourcecode/posicion.c`, [12](#)

Vaciar

`lista.c`, [11](#)