

Laboratorio 5

Generated by Doxygen 1.8.12

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	5
3.1	Lista< T > Class Template Reference	5
3.2	ListaConArreglo< T > Class Template Reference	6
3.2.1	Detailed Description	7
3.2.2	Constructor & Destructor Documentation	7
3.2.2.1	ListaConArreglo() [1/2]	7
3.2.2.2	ListaConArreglo() [2/2]	7
3.2.3	Member Function Documentation	9
3.2.3.1	agregar()	9
3.2.3.2	agregark()	9
3.2.3.3	buscar()	9
3.2.3.4	eliminar()	10
3.2.3.5	eliminarK()	10
3.2.3.6	sumark()	10
3.3	Mesa Class Reference	11
3.3.1	Constructor & Destructor Documentation	12
3.3.1.1	Mesa() [1/2]	12
3.3.1.2	Mesa() [2/2]	12
3.3.2	Member Function Documentation	12
3.3.2.1	agregar_jugador()	12
3.3.2.2	Blackjack()	12
3.3.2.3	llenar()	13
3.3.2.4	sacar_jugador()	13
3.4	Pila< T > Class Template Reference	13
	Index	15

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Lista< T >	5
ListaConArreglo< T >	6
Lista< char >	5
ListaConArreglo< char >	6
Lista< int >	5
ListaConArreglo< int >	6
Mesa	11
Pila< T >	13
Pila< int >	13

Chapter 2

Class Index

2.1 Class List

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

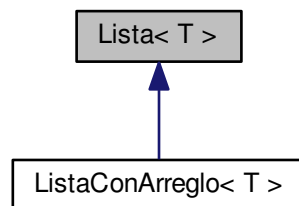
Lista< T >	5
ListaConArreglo< T >	
Plantilla para construir una Lista basada en arreglos	6
Mesa	11
Pila< T >	13

Chapter 3

Class Documentation

3.1 `Lista< T >` Class Template Reference

Inheritance diagram for `Lista< T >`:



Public Member Functions

- **Lista** (const [Lista](#) &orig)
- virtual void **agregar** (T const &e)=0
- virtual void **eliminar** (T const &e)=0
- virtual void **eliminarK** (int k)=0
- virtual int **buscar** (T const &e)=0
- virtual T **siguienteK** (int k)=0
- virtual T **anteriorK** (int k)=0
- virtual T **recuperar** (int k)=0
- virtual void **imprimir** ()=0

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

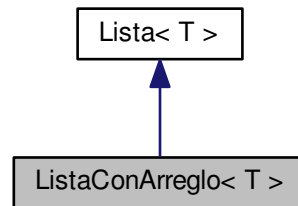
- `Lista.h`

3.2 ListaConArreglo< T > Class Template Reference

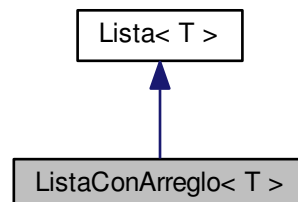
Plantilla para construir una [Lista](#) basada en arreglos.

```
#include <ListaConArreglo.h>
```

Inheritance diagram for ListaConArreglo< T >:



Collaboration diagram for ListaConArreglo< T >:



Public Member Functions

- [ListaConArreglo](#) ()
Constructor de [ListaConArreglo](#).
- [ListaConArreglo](#) (int N)
Constructor sobrecargado de la clase [ListaConArreglo](#).
- [ListaConArreglo](#) (const [ListaConArreglo](#) &orig)
Constructor sobrecargado de la clase [ListaConArreglo](#).
- [~ListaConArreglo](#) ()
Destructor de la clase [ListaConArreglo](#).
- void [agregar](#) (const T &e)
Agrega un elemento.
- void [sumark](#) (int k, const T &e)

- *Suma un elemento.*
- void [agregark](#) (int k, const T &e)
- *Agrega en una posicion un elemento.*
- void [eliminar](#) (const T &e)
- *Elimina un elemento.*
- void [eliminarK](#) (int k)
- *Elimina una posicion.*
- int [buscar](#) (const T &e)
- *Busca un elemento.*
- T [siguienteK](#) (int k)
- T [anteriorK](#) (int k)
- T [recuperar](#) (int k)
- void [imprimir](#) ()
- *Imprime la lista.*
- bool [vacio](#) ()

Public Attributes

- int [tam](#) = 0

3.2.1 Detailed Description

```
template<class T>
class ListaConArreglo< T >
```

Plantilla para construir una [Lista](#) basada en arreglos.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 ListaConArreglo() [1/2]

```
template<class T>
ListaConArreglo< T >::ListaConArreglo (
    int N ) [inline]
```

Constructor sobrecargado de la clase [ListaConArreglo](#).

Parameters

<i>N</i>	Dimension de Lista .
----------	--------------------------------------

3.2.2.2 ListaConArreglo() [2/2]

```
template<class T>
ListaConArreglo< T >::ListaConArreglo (
    const ListaConArreglo< T > & orig ) [inline]
```

Constructor sobrecargado de la clase [ListaConArreglo](#).

Parameters

<i>ListaConArreglo</i> &	Objeto del tipo ListaConArreglo .
--------------------------	---

3.2.3 Member Function Documentation

3.2.3.1 agregar()

```
template<class T>
void ListaConArreglo< T >::agregar (
    const T & e ) [inline], [virtual]
```

Agrega un elemento.

Parameters

& <i>e</i>	Elemento por agregar.
------------	-----------------------

Implements [Lista< T >](#).

3.2.3.2 agregark()

```
template<class T>
void ListaConArreglo< T >::agregark (
    int k,
    const T & e ) [inline]
```

Agrega en una posicion un elemento.

Parameters

<i>k</i>	Posicion.
& <i>e</i>	Elemento por agregar.

3.2.3.3 buscar()

```
template<class T>
int ListaConArreglo< T >::buscar (
    const T & e ) [inline], [virtual]
```

Busca un elemento.

Parameters

& <i>e</i>	Elemento por buscar.
------------	----------------------

Implements [Lista< T >](#).

3.2.3.4 eliminar()

```
template<class T>
void ListaConArreglo< T >::eliminar (
    const T & e ) [inline], [virtual]
```

Elimina un elemento.

Parameters

<code>&e</code>	Elemento por eliminar
---------------------	-----------------------

Implements [Lista< T >](#).

3.2.3.5 eliminarK()

```
template<class T>
void ListaConArreglo< T >::eliminarK (
    int k ) [inline], [virtual]
```

Elimina una posicion.

Parameters

<code>k</code>	Posicion.
----------------	-----------

Implements [Lista< T >](#).

3.2.3.6 sumark()

```
template<class T>
void ListaConArreglo< T >::sumark (
    int k,
    const T & e ) [inline]
```

Suma un elemento.

Parameters

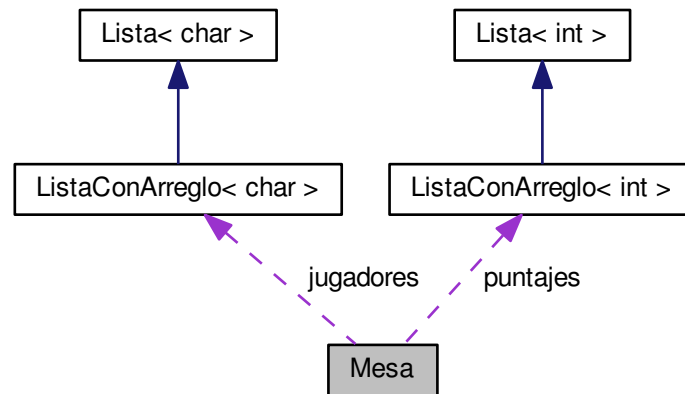
<code>k</code>	Elemento al cual sumar.
<code>&e</code>	Elemento por sumar

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

- [ListaConArreglo.h](#)

3.3 Mesa Class Reference

Collaboration diagram for Mesa:



Public Member Functions

- [Mesa](#) ()
Constructor de clase [Mesa](#).
- [Mesa](#) (int N)
Constructor de clase [Mesa](#) sobrecargado.
- [Mesa](#) (const [Mesa](#) &orig)
Imprime una cadena de caracteres dada por el usuario.
- virtual [~Mesa](#) ()
Destructor de la clase [Mesa](#).
- void [Blackjack](#) ([ListaConArreglo](#)< char > *admisión)
Función principal de distribución para el juego de Blackjack.
- void [agregar_jugador](#) ([ListaConArreglo](#)< char > *admisión)
Agrega jugador a la mesa de juego.
- void [sacar_jugador](#) (int i)
Elimina jugador a la mesa de juego.
- void [barajar](#) ()
Prepara las cartas para el juego.
- bool [lleno](#) ()
Revisa si la mesa está llena o no.
- void [llenar](#) ([ListaConArreglo](#)< char > *Admisión)
Llena las mesas para el juego.
- void [vaciar](#) ()
Vacía la mesa.

Public Attributes

- [ListaConArreglo](#)< char > * **jugadores**
- [ListaConArreglo](#)< int > * **puntajes**
- int **ronda**

3.3.1 Constructor & Destructor Documentation

3.3.1.1 Mesa() [1/2]

```
Mesa::Mesa (
    int N )
```

Constructor de clase [Mesa](#) sobrecargado.

Parameters

<i>N</i>	dimension del arreglo.
----------	------------------------

3.3.1.2 Mesa() [2/2]

```
Mesa::Mesa (
    const Mesa & orig )
```

Imprime una cadena de caracteres dada por el usuario.

Parameters

<i>Mesa&</i>	Objeto del tipo Mesa .
------------------	--

3.3.2 Member Function Documentation

3.3.2.1 agregar_jugador()

```
void Mesa::agregar_jugador (
    ListaConArreglo< char > * admission )
```

Agrega jugador a la mesa de juego.

Parameters

<i>Admission</i>	Lista de personas por entrar al juego.
------------------	--

3.3.2.2 Blackjack()

```
void Mesa::Blackjack (
```



```
ListaConArreglo< char > * Admision )
```

Funcion principal de distribucion para el juego de Blackjack.

Parameters

<i>Admision</i>	Lista de personas por entrar al juego.
-----------------	--

3.3.2.3 llenar()

```
void Mesa::llenar (
    ListaConArreglo< char > * Admision )
```

Llena las mesas para el juego.

Parameters

<i>Admision</i>	Lista de personas por entrar al juego.
-----------------	--

3.3.2.4 sacar_jugador()

```
void Mesa::sacar_jugador (
    int i )
```

Elimina jugador a la mesa de juego.

Parameters

<i>i</i>	Orden del jugador.
----------	--------------------

The documentation for this class was generated from the following files:

- Mesa.h
- Mesa.cpp

3.4 Pila< T > Class Template Reference

Public Member Functions

- **Pila** (int N)
- **Pila** (const [Pila](#) &orig)
- **T pop** ()
- void **push** (T const &e)
- void **imprimir** ()
- void **aleatorizar** ()
- bool **vacio** ()

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

- Pila.h

Index

- agregar
 - ListaConArreglo, [9](#)
- agregar_jugador
 - Mesa, [12](#)
- agregark
 - ListaConArreglo, [9](#)
- Blackjack
 - Mesa, [12](#)
- buscar
 - ListaConArreglo, [9](#)
- eliminar
 - ListaConArreglo, [9](#)
- eliminarK
 - ListaConArreglo, [10](#)
- Lista< T >, [5](#)
- ListaConArreglo
 - agregar, [9](#)
 - agregark, [9](#)
 - buscar, [9](#)
 - eliminar, [9](#)
 - eliminarK, [10](#)
 - ListaConArreglo, [7](#)
 - sumark, [10](#)
- ListaConArreglo< T >, [6](#)
- llenar
 - Mesa, [13](#)
- Mesa, [11](#)
 - agregar_jugador, [12](#)
 - Blackjack, [12](#)
 - llenar, [13](#)
 - Mesa, [12](#)
 - sacar_jugador, [13](#)
- Pila< T >, [13](#)
- sacar_jugador
 - Mesa, [13](#)
- sumark
 - ListaConArreglo, [10](#)