

My Project

Generated by Doxygen 1.9.4

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 Casilla Class Reference	5
3.1.1 Detailed Description	6
3.1.2 Constructor & Destructor Documentation	6
3.1.2.1 Casilla() [1/2]	6
3.1.2.2 ~Casilla()	6
3.1.2.3 Casilla() [2/2]	6
3.1.3 Member Function Documentation	7
3.1.3.1 eliminaPosibilidad()	7
3.1.3.2 esPosible()	7
3.1.3.3 getIndSuma()	8
3.1.3.4 getNPos()	8
3.1.3.5 getValor()	8
3.1.3.6 mismasPosib()	8
3.1.3.7 nuevaPosibilidad()	9
3.1.3.8 operator=()	9
3.1.3.9 setValor()	9
3.2 Posicion Struct Reference	10
3.2.1 Detailed Description	10
3.3 SudokuKiller Class Reference	10
3.3.1 Detailed Description	11
3.3.2 Constructor & Destructor Documentation	11
3.3.2.1 SudokuKiller() [1/2]	11
3.3.2.2 ~SudokuKiller()	12
3.3.2.3 SudokuKiller() [2/2]	12
3.3.3 Member Function Documentation	12
3.3.3.1 AniadirSuma()	12
3.3.3.2 DescartaPosibilidades()	13
3.3.3.3 operator=()	13
3.3.3.4 RellenarCasilla()	13
3.3.3.5 toString()	14
3.4 Suma Class Reference	14
3.4.1 Detailed Description	15
3.4.2 Constructor & Destructor Documentation	15
3.4.2.1 Suma() [1/2]	15
3.4.2.2 ~Suma()	15
3.4.2.3 Suma() [2/2]	15

3.4.3 Member Function Documentation	16
3.4.3.1 AniadirCasilla()	16
3.4.3.2 EliminaCasilla()	16
3.4.3.3 getTam()	17
3.4.3.4 getValor()	17
3.4.3.5 operator=()	17
3.4.3.6 operator[]() [1/2]	17
3.4.3.7 operator[]() [2/2]	18
3.4.3.8 setValor()	18
4 File Documentation	19
4.1 Casilla.h	19
4.2 SudokuKiller.h	19
4.3 Suma.h	20
Index	21

Chapter 1

Class Index

1.1 Class List

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

Casilla	TDA Casilla	5
Posicion	Posición de una casilla en el sudoku	10
SudokuKiller	TDA SudokuKiller	10
Suma	TDA Suma	14

Chapter 2

File Index

2.1 File List

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

Casilla.h	??
SudokuKiller.h	??
Suma.h	??

Chapter 3

Class Documentation

3.1 Casilla Class Reference

TDA [Casilla](#).

```
#include <Casilla.h>
```

Public Member Functions

- [Casilla](#) ()
Constructor por defecto.
- [~Casilla](#) ()
Destructor.
- [Casilla](#) (const [Casilla](#) &otro)
Constructor de copia.
- [Casilla](#) & [operator=](#) (const [Casilla](#) &otro)
Operador de asignación.
- int [getValor](#) () const
Valor de la casilla.
- void [setValor](#) (int valor)
Establece el valor de la casilla.
- int [getIndSuma](#) () const
Índice de la suma a la que pertenece la casilla.
- int [getNPos](#) () const
Número de posibilidades para la casilla.
- void [eliminaPosibilidad](#) (int valor)
Elimina una posibilidad de la casilla.
- void [nuevaPosibilidad](#) (int valor)
Añade una posibilidad a la casilla.
- bool [mismasPosib](#) ([Casilla](#) casilla)
Comprueba si dos casillas tienen las mismas posibilidades.
- bool [esPosible](#) (int valor) const
Comprueba si un valor es posible para la casilla.

3.1.1 Detailed Description

TDA [Casilla](#).

Representa una casilla de un sudoku killer.

- Si la casilla tiene un 0, se considera vacía.

Author

Andrés Alejo Martínez

Julia Miralles Martínez

3.1.2 Constructor & Destructor Documentation

3.1.2.1 [Casilla\(\)](#) [1/2]

```
Casilla::Casilla ( )
```

Constructor por defecto.

Returns

Crea una casilla vacía, con valor 0 y posibilidades = {1,2,3,4,5,6,7,8,9}.

3.1.2.2 [~Casilla\(\)](#)

```
Casilla::~~Casilla ( )
```

Destructor.

Returns

Libera la memoria dinámica reservada.

Postcondition

La casilla está vacía.

3.1.2.3 [Casilla\(\)](#) [2/2]

```
Casilla::Casilla (
    const Casilla & otro )
```

Constructor de copia.

Parameters

<i>otro</i>	Casilla a copiar.
-------------	-------------------

Returns

Casilla, el objeto casilla creada.

3.1.3 Member Function Documentation

3.1.3.1 eliminaPosibilidad()

```
void Casilla::eliminaPosibilidad (
    int valor )
```

Elimina una posibilidad de la casilla.

Parameters

<i>valor</i>	Valor a eliminar.
--------------	-------------------

Precondition

$1 \leq \text{valor} \leq 9$

3.1.3.2 esPosible()

```
bool Casilla::esPosible (
    int valor ) const
```

Comprueba si un valor es posible para la casilla.

Parameters

<i>valor</i>	Valor a comprobar.
--------------	--------------------

Returns

true si el valor es posible, false en caso contrario.

3.1.3.3 getIndSuma()

```
int Casilla::getIndSuma ( ) const
```

Índice de la suma a la que pertenece la casilla.

Returns

Índice de la suma a la que pertenece la casilla.

3.1.3.4 getNPos()

```
int Casilla::getNPos ( ) const
```

Número de posibilidades para la casilla.

Returns

Número de posibilidades para la casilla.

3.1.3.5 getValor()

```
int Casilla::getValor ( ) const
```

Valor de la casilla.

Returns

Valor de la casilla.

3.1.3.6 mismasPosib()

```
bool Casilla::mismasPosib (
    Casilla casilla )
```

Comprueba si dos casillas tienen las mismas posibilidades.

Parameters

<i>casilla</i>	Casilla a comparar.
----------------	-------------------------------------

Returns

true si las casillas tienen las mismas posibilidades, false en caso contrario.

3.1.3.7 nuevaPosibilidad()

```
void Casilla::nuevaPosibilidad (
    int valor )
```

Añade una posibilidad a la casilla.

Parameters

<i>valor</i>	Valor a añadir.
--------------	-----------------

Precondition

$1 \leq \text{valor} \leq 9$

3.1.3.8 operator=()

```
Casilla & Casilla::operator= (
    const Casilla & otro )
```

Operador de asignación.

Parameters

<i>otro</i>	Casilla a copiar.
-------------	-------------------

Returns

Una referencia a la casilla copiada.

3.1.3.9 setValor()

```
void Casilla::setValor (
    int valor )
```

Establece el valor de la casilla.

Parameters

<i>valor</i>	Valor de la casilla.
--------------	----------------------

Precondition

$0 \leq \text{valor} \leq 9$

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

- Casilla.h

3.2 Posicion Struct Reference

Posición de una casilla en el sudoku.

```
#include <SudokuKiller.h>
```

Public Attributes

- int **fila**
- int **columna**

3.2.1 Detailed Description

Posición de una casilla en el sudoku.

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

- SudokuKiller.h

3.3 SudokuKiller Class Reference

TDA [SudokuKiller](#).

```
#include <SudokuKiller.h>
```

Public Member Functions

- [SudokuKiller](#) ()
Constructor por defecto.
- [~SudokuKiller](#) ()
Destructor.
- [SudokuKiller](#) (const [SudokuKiller](#) &otro)
Constructor de copia.
- [SudokuKiller](#) & [operator=](#) (const [SudokuKiller](#) &otro)
Operador de asignación.
- void [RellenarCasilla](#) ([Posicion](#) pos, int num)
Rellena una casilla del sudoku.
- void [AniadirSuma](#) ([Suma](#) &sum)
Añade una suma al tablero del sudoku.
- void [DescartaPosibilidades](#) ([Posicion](#) pos)
Elimina los números que no puedan estar en una casilla (ver clase [Casilla](#)).
- string [toString](#) () const
Formatea el sudoku killer en un string.

3.3.1 Detailed Description

TDA [SudokuKiller](#).

Representa un sudoku killer, un sudoku donde se guardan las sumas de casillas contiguas.

Author

Andrés Alejo Martínez

Julia Miralles Martínez

3.3.2 Constructor & Destructor Documentation

3.3.2.1 [SudokuKiller\(\)](#) [1/2]

```
SudokuKiller::SudokuKiller ( )
```

Constructor por defecto.

Returns

Crea un sudoku killer vacío.

3.3.2.2 ~SudokuKiller()

```
SudokuKiller::~SudokuKiller ( )
```

Destructor.

Returns

Libera la memoria dinámica reservada.

Postcondition

El [SudokuKiller](#) queda vacío.

3.3.2.3 SudokuKiller() [2/2]

```
SudokuKiller::SudokuKiller (
    const SudokuKiller & otro )
```

Constructor de copia.

Parameters

<i>otro</i>	Sudoku killer a copiar.
-------------	-------------------------

Returns

[SudokuKiller](#), el objeto sudoku creado.

3.3.3 Member Function Documentation

3.3.3.1 AniadirSuma()

```
void SudokuKiller::AniadirSuma (
    Suma & sum )
```

Añade una suma al tablero del sudoku.

Parameters

<i>sum</i>	Suma a añadir.
------------	--------------------------------

3.3.3.2 DescartaPosibilidades()

```
void SudokuKiller::DescartaPosibilidades (
    Posicion pos )
```

Elimina los números que no puedan estar en una casilla (ver clase [Casilla](#)).

Parameters

<i>pos</i>	Posición de la casilla a comprobar.
------------	-------------------------------------

Precondition

$0 \leq \text{pos.fila} < 9 \ \&\& \ 0 \leq \text{pos.columna} < 9$

3.3.3.3 operator=()

```
SudokuKiller & SudokuKiller::operator= (
    const SudokuKiller & otro )
```

Operador de asignación.

Parameters

<i>otro</i>	Sudoku killer a copiar.
-------------	-------------------------

Returns

Una referencia al sudoku killer copiado.

3.3.3.4 RellenarCasilla()

```
void SudokuKiller::RellenarCasilla (
    Posicion pos,
    int num )
```

Rellena una casilla del sudoku.

Parameters

<i>pos</i>	Posición de la casilla a rellenar.
<i>num</i>	Número a introducir en la casilla.

Precondition

`0 <= pos.fila < 9 && 0 <= pos.columna < 9 && 1 <= num <= 9`

3.3.3.5 toString()

```
string SudokuKiller::toString ( ) const
```

Formatea el sudoku killer en un string.

Returns

String con el sudoku killer formateado.

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

- SudokuKiller.h

3.4 Suma Class Reference

TDA [Suma](#).

```
#include <Suma.h>
```

Public Member Functions

- [Suma](#) ()
Constructor por defecto.
- [~Suma](#) ()
Destructor.
- [Suma](#) (const [Suma](#) &otro)
Constructor de copia.
- [Suma](#) & [operator=](#) (const [Suma](#) &otro)
Operador de asignación.
- int [getTam](#) () const
Tamaño de la suma.
- int [getValor](#) () const
Valor de la suma.
- void [setValor](#) (int valor)
Establece el valor de la suma.
- void [AniadirCasilla](#) ([Posicion](#) pos)
Añade una casilla a la suma.
- void [EliminaCasilla](#) ([Posicion](#) pos)
Elimina una casilla de la suma.
- [Posicion](#) & [operator\[\]](#) (int i)
Operador de acceso a una casilla.
- const [Posicion](#) & [operator\[\]](#) (int i) const
Operador de acceso a una casilla.

3.4.1 Detailed Description

TDA [Suma](#).

Representa una suma de casillas.

Author

Andrés Alejo Martínez

Julia Miralles Martínez

3.4.2 Constructor & Destructor Documentation

3.4.2.1 Suma() [1/2]

```
Suma::Suma ( )
```

Constructor por defecto.

Returns

Crea una suma vacía.

3.4.2.2 ~Suma()

```
Suma::~~Suma ( )
```

Destructor.

Returns

Libera la memoria dinámica reservada.

Postcondition

La [Suma](#) está vacía.

3.4.2.3 Suma() [2/2]

```
Suma::Suma (
    const Suma & otro )
```

Constructor de copia.

Parameters

<i>otro</i>	Suma a copiar.
-------------	----------------

Returns

Suma, el objeto suma creado.

3.4.3 Member Function Documentation

3.4.3.1 AniadirCasilla()

```
void Suma::AniadirCasilla (  
    Posicion pos )
```

Añade una casilla a la suma.

Parameters

<i>pos</i>	Posición de la casilla añadida.
------------	---------------------------------

Precondition

$0 \leq \text{pos.fila} < 9 \ \&\& \ 0 \leq \text{pos.columna} < 9$ pos tiene que ser una casilla contigua a las demás.

3.4.3.2 EliminaCasilla()

```
void Suma::EliminaCasilla (  
    Posicion pos )
```

Elimina una casilla de la suma.

Parameters

<i>pos</i>	Posición de la casilla a eliminar.
------------	------------------------------------

Precondition

$0 \leq \text{pos.fila} < 9 \ \&\& \ 0 \leq \text{pos.columna} < 9$

3.4.3.3 getTam()

```
int Suma::getTam ( ) const
```

Tamaño de la suma.

Returns

Número de posiciones que forman la suma.

3.4.3.4 getValor()

```
int Suma::getValor ( ) const
```

Valor de la suma.

Returns

Valor de la suma.

3.4.3.5 operator=()

```
Suma & Suma::operator= (
    const Suma & otro )
```

Operador de asignación.

Parameters

<i>otro</i>	Suma a copiar.
-------------	----------------

Returns

Una referencia a la suma copiado.

3.4.3.6 operator[]() [1/2]

```
Posicion & Suma::operator[] (
    int i )
```

Operador de acceso a una casilla.

Parameters

<i>i</i>	Posición de la casilla a acceder.
----------	-----------------------------------

Returns

Una referencia a la posición de la casilla en el sudoku.

3.4.3.7 operator[]() [2/2]

```
const Posicion & Suma::operator[] (
    int i ) const
```

Operador de acceso a una casilla.

Parameters

<i>i</i>	Posición de la casilla a acceder.
----------	-----------------------------------

Returns

Una referencia constante a la posición de la casilla en el sudoku.

3.4.3.8 setValor()

```
void Suma::setValor (
    int valor )
```

Establece el valor de la suma.

Parameters

<i>valor</i>	Valor de la suma.
--------------	-------------------

Precondition

$\text{valor} \geq 3$

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

- Suma.h

Chapter 4

File Documentation

4.1 Casilla.h

```
1
11 class Casilla{
12 private:
20     int valor;
21     int indSuma;
22     int *posibilidades;
23     int nPos;
26 public:
31     Casilla();
32
38     ~Casilla();
39
45     Casilla(const Casilla& otro);
46
52     Casilla& operator=(const Casilla& otro);
53
58     int getValor() const;
59
65     void setValor(int valor);
66
71     int getIndSuma() const;
72
77     int getNPos() const;
78
84     void eliminaPosibilidad(int valor);
85
91     void nuevaPosibilidad(int valor);
92
98     bool mismasPosib(Casilla casilla);
99
105     bool esPosible(int valor) const;
106 };
```

4.2 SudokuKiller.h

```
1 #include <Casilla.h>
2 #include <Suma.h>
3 #include <iostream>
4
5 using namespace std;
6
10 struct Posicion {
11     int fila;
12     int columna;
13 };
14
15
25 class SudokuKiller {
26 private:
27     Casilla sudoku[9][9];
28     Suma *sumas;
29     int nSumas;
38     bool EstaEnFila(int fila, int num) const;
39
```

```
47 bool EstaEnColumna(int columna, int num) const;
48
56 bool EstaEnBloque(Posicion pos, int num) const;
57
58 public:
63   SudokuKiller();
64
70   ~SudokuKiller();
71
77   SudokuKiller(const SudokuKiller& otro);
78
84   SudokuKiller& operator=(const SudokuKiller& otro);
85
93   void RellenarCasilla(Posicion pos, int num);
94
99   void AniadirSuma(Suma &sum);
100
106   void DescartaPosibilidades(Posicion pos);
107
112   string toString() const;
113 };
114
121 ostream& operator<<(ostream &os, const SudokuKiller &sudoku);
```

4.3 Suma.h

```
1
10 class Suma {
11 private:
12
18   int val;
19   Posicion *posiciones;
20   int nPos;
22 public:
27   Suma();
28
34   ~Suma();
35
41   Suma(const Suma& otro);
42
48   Suma& operator=(const Suma& otro);
49
54   int getTam() const;
55
60   int getValor() const;
61
67   void setValor(int valor);
68
75   void AniadirCasilla(Posicion pos);
76
82   void EliminaCasilla(Posicion pos);
83
89   Posicion& operator[](int i);
90
96   const Posicion& operator[](int i) const;
97 };
```


Index

~Casilla
 Casilla, [6](#)
~SudokuKiller
 SudokuKiller, [11](#)
~Suma
 Suma, [15](#)

AniadirCasilla
 Suma, [16](#)
AniadirSuma
 SudokuKiller, [12](#)

Casilla, [5](#)
 ~Casilla, [6](#)
 Casilla, [6](#)
 eliminaPosibilidad, [7](#)
 esPosible, [7](#)
 getIndSuma, [7](#)
 getNPos, [8](#)
 getValor, [8](#)
 mismasPosib, [8](#)
 nuevaPosibilidad, [9](#)
 operator=, [9](#)
 setValor, [9](#)

DescartaPosibilidades
 SudokuKiller, [12](#)

EliminaCasilla
 Suma, [16](#)
eliminaPosibilidad
 Casilla, [7](#)
esPosible
 Casilla, [7](#)

getIndSuma
 Casilla, [7](#)
getNPos
 Casilla, [8](#)
getTam
 Suma, [16](#)
getValor
 Casilla, [8](#)
 Suma, [17](#)

mismasPosib
 Casilla, [8](#)

nuevaPosibilidad
 Casilla, [9](#)

operator=
 Casilla, [9](#)
 SudokuKiller, [13](#)
 Suma, [17](#)
operator[]
 Suma, [17](#), [18](#)

Posicion, [10](#)

RellenarCasilla
 SudokuKiller, [13](#)

setValor
 Casilla, [9](#)
 Suma, [18](#)
SudokuKiller, [10](#)
 ~SudokuKiller, [11](#)
 AniadirSuma, [12](#)
 DescartaPosibilidades, [12](#)
 operator=, [13](#)
 RellenarCasilla, [13](#)
 SudokuKiller, [11](#), [12](#)
 toString, [14](#)
Suma, [14](#)
 ~Suma, [15](#)
 AniadirCasilla, [16](#)
 EliminaCasilla, [16](#)
 getTam, [16](#)
 getValor, [17](#)
 operator=, [17](#)
 operator[], [17](#), [18](#)
 setValor, [18](#)
 Suma, [15](#)

toString
 SudokuKiller, [14](#)