



TDA Sudoku Killer

Yeray Lopez Ramirez

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 Bloque Class Reference	5
3.1.1 Detailed Description	6
3.1.2 Constructor & Destructor Documentation	6
3.1.2.1 Bloque() [1/2]	6
3.1.2.2 Bloque() [2/2]	6
3.1.3 Member Function Documentation	6
3.1.3.1 aniadirCasilla()	6
3.1.3.2 eliminarCasilla()	7
3.1.3.3 esCasillaContigua()	7
3.1.3.4 getCasilla()	8
3.1.3.5 getSumaBloque()	8
3.1.3.6 getTamanio()	8
3.1.3.7 setCasilla()	9
3.2 BloquesSudoku Class Reference	9
3.2.1 Detailed Description	10
3.2.2 Constructor & Destructor Documentation	10
3.2.2.1 BloquesSudoku()	10
3.2.3 Member Function Documentation	10
3.2.3.1 aniadirBloque()	10
3.2.3.2 eliminarBloque()	10
3.2.3.3 getBloquesSudoku()	11
3.2.3.4 setBloquesSudoku() [1/2]	11
3.2.3.5 setBloquesSudoku() [2/2]	11
3.2.3.6 sumaBloques()	13
3.3 Casilla Class Reference	13
3.3.1 Detailed Description	14
3.3.2 Constructor & Destructor Documentation	14
3.3.2.1 Casilla() [1/2]	14
3.3.2.2 Casilla() [2/2]	14
3.3.3 Member Function Documentation	15
3.3.3.1 getColumna()	15
3.3.3.2 getFila()	15
3.3.3.3 getValor()	15
3.3.3.4 isEmpty()	16
3.3.3.5 setCasilla()	16
3.3.3.6 setColumna()	16

3.3.3.7 setFila()	16
3.3.3.8 setPosicion()	17
3.3.3.9 setValor()	17
3.3.3.10 to_string()	17
3.4 Matriz< M > Class Template Reference	18
3.4.1 Detailed Description	18
3.4.2 Constructor & Destructor Documentation	18
3.4.2.1 Matriz()	18
3.4.3 Member Function Documentation	19
3.4.3.1 eliminarValor()	19
3.4.3.2 getCasilla()	19
3.4.3.3 getColumna()	20
3.4.3.4 getFila()	20
3.4.3.5 insertarValor()	20
3.5 ResolverSudokuKiller Class Reference	21
3.5.1 Detailed Description	21
3.5.2 Member Function Documentation	22
3.5.2.1 DescartaSecuencias()	22
3.5.2.2 ResuelveSudoku()	22
3.5.2.3 SecuenciaBloqueKiller()	22
3.5.2.4 SecuenciaCaja()	23
3.5.2.5 SecuenciasPosibles()	23
3.6 Sudoku Class Reference	24
3.6.1 Detailed Description	24
3.6.2 Constructor & Destructor Documentation	24
3.6.2.1 Sudoku()	24
3.6.3 Member Function Documentation	25
3.6.3.1 GenerarSudoku()	25
3.6.3.2 getSudoku()	25
3.6.3.3 getValorCasilla()	25
3.7 SudokuKiller Class Reference	26
3.7.1 Detailed Description	26
3.7.2 Constructor & Destructor Documentation	26
3.7.2.1 SudokuKiller() [1/2]	26
3.7.2.2 SudokuKiller() [2/2]	27
3.7.3 Member Function Documentation	27
3.7.3.1 EstaResuelto()	27
4 File Documentation	29
4.1 include/BloquesSudoku.h File Reference	29
4.1.1 Detailed Description	30
4.1.2 Function Documentation	30

4.1.2.1 esMismoBloque()	30
4.1.2.2 seSolapaBloques()	30
4.2 BloquesSudoku.h	31
4.3 include/Casilla.h File Reference	32
4.3.1 Detailed Description	32
4.3.2 Function Documentation	32
4.3.2.1 mismaCasilla()	32
4.4 Casilla.h	33
4.5 include/Matriz.h File Reference	33
4.5.1 Detailed Description	34
4.6 Matriz.h	34
4.7 include/ResolverSudokuKiller.h File Reference	34
4.7.1 Detailed Description	35
4.8 ResolverSudokuKiller.h	35
4.9 include/SudokuKiller.h File Reference	35
4.9.1 Detailed Description	36
4.9.2 Function Documentation	36
4.9.2.1 BloquesValido()	36
4.9.2.2 SudokuKillerValido()	37
4.9.2.3 SudokuValido()	37
4.10 SudokuKiller.h	37
4.11 src/main.cpp File Reference	38
4.11.1 Detailed Description	38
4.11.2 Function Documentation	39
4.11.2.1 main()	39
Index	41

Chapter 1

Class Index

1.1 Class List

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

Bloque	
Bloque	Una clase que agrupa las casillas de una Matriz , se utiliza en la clase BloquesSudoku para la clase SudokuKiller
	5
BloquesSudoku	
BloquesSudoku	Una clase que contiene los bloques del Sudoku , se utiliza para la clase SudokuKiller y ResolverSudokuKiller
	9
Casilla	
Casilla	Una clase que contiene los datos básicos de una casilla del sudoku, a partir de una matriz sudoku proporcionada por la clase SudokuKiller
	13
Matriz< M >	
Matriz	Un template para el tipo de datos abstracto Matriz , se utiliza para las clases BloquesSudoku , SudokuKiller y ResolverSudokuKiller
	18
ResolverSudokuKiller	
ResolverSudokuKiller	Una clase que resuelve el juego del Sudoku Killer a partir de algoritmos logicos humanos, utiliza la clase SudokuKiller
	21
Sudoku	
Sudoku	Clase basica del cual se creará la version killer del sudoku
	24
SudokuKiller	
SudokuKiller	Una clase para crear y gestionar el sudoku killer completo, se utiliza la clase Matriz y la utiliza ResolverSudokuKiller
	26

Chapter 2

File Index

2.1 File List

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

include/ BloquesSudoku.h	
Fichero cabecera de las clases Bloque y BloquesSudoku	29
include/ Casilla.h	
Fichero cabecera de la clase Casilla	32
include/ Matriz.h	
Fichero cabecera de la clase Matriz	33
include/ ResolverSudokuKiller.h	
Fichero cabecera de la clase ResolverSudokuKiller	34
include/ SudokuKiller.h	
Fichero cabecera de la clase SudokuKiller	35
src/ main.cpp	
Fichero main.cpp	38

Chapter 3

Class Documentation

3.1 Bloque Class Reference

Bloque Una clase que agrupa las casillas de una [Matriz](#), se utiliza en la clase [BloquesSudoku](#) para la clase [SudokuKiller](#).

```
#include <BloquesSudoku.h>
```

Public Member Functions

- **Bloque** ()
Constructor por defecto.
- **Bloque** (int tam)
Constructor con parametros.
- **Bloque** (const **Bloque** &b)
Constructor de copia.
- int **getTamanio** ()
Obtiene el tamaño del bloque.
- int **getSumaBloque** ()
Obtiene la suma de los valores bloque, llama a sumaValores.
- **Casilla** **getCasilla** (int i)
Obtiene un objeto [Casilla](#).
- void **setCasilla** (int i, int fil, int col, int val)
Establece los parametros de una casilla.
- bool **aniadirCasilla** (int val, int fil, int col)
Añade una nueva casilla al bloque, tama++.
- bool **eliminarCasilla** (int i)
Elimina la casilla del bloque.
- bool **esCasillaContigua** (**Casilla** nueva)
Comprueba que la nueva casilla sea contigua a alguna existente en el bloque.
- void **sumaValores** ()
Suma los valores de las casillas del bloque y modifica el atributo suma.

3.1.1 Detailed Description

[Bloque](#) Una clase que agrupa las casillas de una [Matriz](#), se utiliza en la clase [BloquesSudoku](#) para la clase [SudokuKiller](#).

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Bloque() [1/2]

```
Bloque::Bloque (
    int tam )
```

Constructor con parametros.

Parameters

<i>tam</i>	: int, Tamaño del bloque
------------	--------------------------

3.1.2.2 Bloque() [2/2]

```
Bloque::Bloque (
    const Bloque & b )
```

Constructor de copia.

Parameters

<i>b</i>	: Bloque , El bloque a copiar
----------	---

3.1.3 Member Function Documentation

3.1.3.1 aniadirCasilla()

```
bool Bloque::aniadirCasilla (
    int val,
    int fil,
    int col )
```

Añade una nueva casilla al bloque, tama++.

Parameters

<i>val</i>	: int, El valor de la casilla
<i>fil</i>	: int, La posicion en la fila de la casilla
<i>col</i>	: int, La posicion en la columna de la casilla

Returns

true si es añadida, false en otro caso

Precondition

`val != null && val > 0 && val < 0`
`tama < TAMA_MAX`
`esCasillaContigua==true`

3.1.3.2 eliminarCasilla()

```
bool Bloque::eliminarCasilla (  
    int i )
```

Elimina la casilla del bloque.

Parameters

<i>i</i>	: int, Posicion de la casilla a eliminar
----------	--

Returns

true si se elimina correctamente, false en otro caso

Precondition

`i < tama && i >= 0`

3.1.3.3 esCasillaContigua()

```
bool Bloque::esCasillaContigua (  
    Casilla nueva )
```

Comprueba que la nueva casilla sea contigua a alguna existente en el bloque.

Parameters

<i>nueva</i>	: Casilla , La casilla a comprobar
--------------	--

Precondition

Si tama==0, devuelve true

Returns

true si la casilla es contigua, false en otro caso

3.1.3.4 getCasilla()

```
Casilla Bloque::getCasilla (
    int i )
```

Obtiene un objeto [Casilla](#).

Parameters

<i>i</i>	: int, posición de la casilla en el vector
----------	--

Returns

el objeto [Casilla](#)

3.1.3.5 getSumaBloque()

```
int Bloque::getSumaBloque ( )
```

Obtiene la suma de los valores bloque, llama a sumaValores.

Returns

la suma del valor en las casillas del bloque

3.1.3.6 getTamano()

```
int Bloque::getTamano ( )
```

Obtiene el tamaño del bloque.

Returns

el tamaño del bloque

3.1.3.7 setCasilla()

```
void Bloque::setCasilla (
    int i,
    int fil,
    int col,
    int val )
```

Establece los parametros de una casilla.

Parameters

<i>i</i>	: int, Posicion de la casilla a modificar
<i>fil</i>	: int, La posicion en la fila de la casilla
<i>col</i>	: int, La posicion en la columna de la casilla
<i>val</i>	: int, El valor de la casilla

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

- include/[BloquesSudoku.h](#)

3.2 BloquesSudoku Class Reference

[BloquesSudoku](#) Una clase que contiene los bloques del [Sudoku](#), se utiliza para la clase [SudokuKiller](#) y [ResolverSudokuKiller](#).

```
#include <BloquesSudoku.h>
```

Public Member Functions

- **BloquesSudoku ()**
Constructor por defecto.
- **BloquesSudoku (int num)**
Constructor con parametros.
- **Bloque getBloquesSudoku (int i)**
Obtiene un objeto [Bloque](#) del vector de objetos de [BloquesSudoku](#).
- void **setBloquesSudoku (int i, Bloque b)**
Modifica un bloque a partir de otro.
- void **setBloquesSudoku (int i, vector< [Casilla](#) >)**
Modifica un bloque a partir de un vector de casillas.
- bool **añadirBloque (Bloque b)**
Añade un bloque al vector de [BloquesSudoku](#).
- bool **eliminarBloque (int i)**
elimina un bloque del vector
- int **sumaBloques (vector< int > ibloques)**
Calcula el total de los atributos suma de los bloques pedidos.

3.2.1 Detailed Description

[BloquesSudoku](#) Una clase que contiene los bloques del [Sudoku](#), se utiliza para la clase [SudokuKiller](#) y [ResolverSudokuKiller](#).

3.2.2 Constructor & Destructor Documentation

3.2.2.1 BloquesSudoku()

```
BloquesSudoku::BloquesSudoku (
    int num )
```

Constructor con parametros.

Parameters

<i>num</i>	: int, Numero de bloques a crear
------------	----------------------------------

3.2.3 Member Function Documentation

3.2.3.1 aniadirBloque()

```
bool BloquesSudoku::aniadirBloque (
    Bloque b )
```

Añade un bloque al vector de [BloquesSudoku](#).

Parameters

<i>b</i>	: Bloque , El bloque a añadir
----------	---

Returns

true si se inserta correctamente, false en otro caso

3.2.3.2 eliminarBloque()

```
bool BloquesSudoku::eliminarBloque (
    int i )
```

elimina un bloque del vector

Parameters

<i>i</i>	: int, Posicion del bloque a eliminar
----------	---------------------------------------

Returns

true si se borra correctamente, false en otro caso

3.2.3.3 getBloquesSudoku()

```
Bloque BloquesSudoku::getBloquesSudoku (
    int i )
```

Obtiene un objeto [Bloque](#) del vector de objetos de [BloquesSudoku](#).

Parameters

<i>i</i>	: int, Posicion del objeto a obtener en el vector
----------	---

Returns

un objeto [Bloque](#)

3.2.3.4 setBloquesSudoku() [1/2]

```
void BloquesSudoku::setBloquesSudoku (
    int i,
    Bloque b )
```

Modifica un bloque a partir de otro.

Parameters

<i>i</i>	: int, La posicion del bloque a modificar
<i>b</i>	: Bloque , El bloque de copia

3.2.3.5 setBloquesSudoku() [2/2]

```
void BloquesSudoku::setBloquesSudoku (
    int i,
    vector< Casilla > )
```

Parameters

<i>i</i>	: int, La posicion del bloque a modificar
<Casilla>	: Vector, Las casillas a copiar

3.2.3.6 sumaBloques()

```
int BloquesSudoku::sumaBloques (
    vector< int > ibloques )
```

Calcula el total de los atributos suma de los bloques pedidos.

Parameters

<i>ibloques</i>	: vector<int>, Posiciones de los bloques a calcular
-----------------	---

Returns

la suma total de los bloques dados

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

- include/[BloquesSudoku.h](#)

3.3 Casilla Class Reference

[Casilla](#) Una clase que contiene los datos básicos de una casilla del sudoku, a partir de una matriz sudoku proporcionada por la clase [SudokuKiller](#).

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

Public Member Functions

- **Casilla ()**
Constructor por defecto.
- **Casilla (int fil, int col)**
Constructor con parametros.
- **Casilla (int fil, int col, int valor)**
Constructor con parametros.
- bool **isEmpty ()** const
Detecta pares invalidos.
- const int **getFila ()** const
Obtiene la fila.
- const int **getColumna ()** const

- Obtiene la columna.*
 - const int `getValor ()` const
- Obtiene el valor de la casilla.*
 - void `setFila` (const int fil)
- Modifica el valor de la fila al valor dado.*
 - void `setColumna` (const int col)
- Modifica el valor de la columna al valor dado.*
 - void `setValor` (const int valor)
- Modifica el valor de la casilla al valor dado.*
 - void `setPosicion` (int fil, int col)
- Modifica la fila y columna.*
 - void `setCasilla` (int fil, int col, int valor)
- modifica todos los valores propios de una casilla*
 - std::string `to_string ()` const
- Convierte el objeto `Casilla` en un string.*
 - void `initDefault ()`
- Establece el par por defecto. Filas, columnas, valor == -1.*

3.3.1 Detailed Description

`Casilla` Una clase que contiene los datos básicos de una casilla del sudoku, a partir de una matriz sudoku proporcionada por la clase `SudokuKiller`.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 `Casilla()` [1/2]

```
Casilla::Casilla (
    int fil,
    int col )
```

Constructor con parametros.

Parameters

<i>fil</i>	: int
<i>col</i>	: int

3.3.2.2 `Casilla()` [2/2]

```
Casilla::Casilla (
    int fil,
```

```
int col,  
int valor )
```

Constructor con parametros.

Parameters

<i>fil</i>	: int
<i>col</i>	: int
<i>valor</i>	: int

Precondition

valor>0 && valor<10

3.3.3 Member Function Documentation

3.3.3.1 getColumna()

```
const int Casilla::getColumna ( ) const
```

Obtiene la columna.

Returns

La columna de la casilla

3.3.3.2 getFila()

```
const int Casilla::getFila ( ) const
```

Obtiene la fila.

Returns

La fila de la casilla

3.3.3.3 getValor()

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

Obtiene el valor de la casilla.

Returns

El valor

3.3.3.4 isEmpty()

```
bool Casilla::isEmpty ( ) const
```

Detecta pares invalidos.

Returns

Devuelve true cuando sus parametros son -1, y falso en otro caso

3.3.3.5 setCasilla()

```
void Casilla::setCasilla (
    int fil,
    int col,
    int valor )
```

modifica todos los valores propios de una casilla

Parameters

<i>fil</i>	: int
<i>col</i>	: int
<i>valor</i>	: int

Precondition

`valor>0 && valor<10`

3.3.3.6 setColumna()

```
void Casilla::setColumna (
    const int col )
```

Modifica el valor de la columna al valor dado.

Parameters

<i>col</i>	: int
------------	-------

3.3.3.7 setFila()

```
void Casilla::setFila (
```

```
const int fil )
```

Modifica el valor de la fila al valor dado.

Parameters

<i>fil</i>	: int
------------	-------

3.3.3.8 setPosicion()

```
void Casilla::setPosicion (
    int fil,
    int col )
```

Modifica la fila y columna.

Parameters

<i>fil</i>	: int
<i>col</i>	: int

3.3.3.9 setValor()

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

Modifica el valor de la casilla al valor dado.

Parameters

<i>valor</i>	: int
--------------	-------

3.3.3.10 to_string()

```
std::string Casilla::to_string ( ) const
```

Convierte el objeto [Casilla](#) en un string.

Returns

un string en el formato -> Fila: *fil* , Columna: *col* , Valor: *val*

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

- include/[Casilla.h](#)

3.4 Matriz< M > Class Template Reference

[Matriz](#) Un template para el tipo de datos abstracto [Matriz](#), se utiliza para las clases [BloquesSudoku](#), [SudokuKiller](#) y [ResolverSudokuKiller](#).

```
#include <Matriz.h>
```

Public Member Functions

- **Matriz ()**
Constructor sin parametros.
- **Matriz (int f, int c)**
Constructor de dos parámetros: filas y columnas.
- **~Matriz ()**
Destructor de la clase [Matriz](#).
- **int getCasilla (int fil, int col)**
Obtiene el valor que almacena la matriz en la fila y columna que se pasan como parametro.
- **int getFila ()**
Obtiene el numero de filas.
- **int getColumna ()**
Obtiene el numero de columnas.
- **void insertarValor (int fil, int col, const M &valor)**
Inserta un nuevo valor en una casilla de la [Matriz](#).
- **void eliminarValor (int fil, int col)**
Borrar un valor dentro de la matriz.

3.4.1 Detailed Description

```
template<class M>
class Matriz< M >
```

[Matriz](#) Un template para el tipo de datos abstracto [Matriz](#), se utiliza para las clases [BloquesSudoku](#), [SudokuKiller](#) y [ResolverSudokuKiller](#).

3.4.2 Constructor & Destructor Documentation

3.4.2.1 Matriz()

```
template<class M >
Matriz< M >::Matriz (
    int f,
    int c )
```

Constructor de dos parámetros: filas y columnas.

Parameters

<i>f</i>	: int, Numero de filas
<i>c</i>	: int, Numero de columnas

Precondition

`f > 0 && f != null`

`c > 0 && c != null`

3.4.3 Member Function Documentation

3.4.3.1 eliminarValor()

```
template<class M >
void Matriz< M >::eliminarValor (
    int fil,
    int col )
```

Borrar un valor dentro de la matriz.

Parameters

<i>f</i>	: int Posicion de la fila
<i>c</i>	: int Posicion de la columna

Precondition

`fil > 0 && fil < filas`

`col > 0 && col < columnas`

3.4.3.2 getCasilla()

```
template<class M >
int Matriz< M >::getCasilla (
    int fil,
    int col )
```

Obtiene el valor que almacena la matriz en la fila y columna que se pasan como parametro.

Parameters

<i>f</i>	: int, Numero de filas
<i>c</i>	: int, Numero de columnas

Precondition

`fil > 0 && fil != null`

`col > 0 && col != null`

Returns

La casilla en la fila y columna dadas

3.4.3.3 getColumna()

```
template<class M >
int Matriz< M >::getColumna ( )
```

Obtiene el numero de columnas.

Returns

El numero de columnas de la [Matriz](#)

3.4.3.4 getFila()

```
template<class M >
int Matriz< M >::getFila ( )
```

Obtiene el numero de filas.

Returns

El numero de filas de la [Matriz](#)

3.4.3.5 insertarValor()

```
template<class M >
void Matriz< M >::insertarValor (
    int fil,
    int col,
    const M & valor )
```

Inserta un nuevo valor en una casilla de la [Matriz](#).

Parameters

<i>fil</i>	: int, Posicion de la fila
<i>col</i>	: int, Posicion de la columna
<i>valor</i>	: M, El valor que queremos insertar

Precondition

```

fil > 0 && fil < filas
col > 0 && col < columnas
valor != null

```

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

- include/[Matriz.h](#)

3.5 ResolverSudokuKiller Class Reference

[ResolverSudokuKiller](#) Una clase que resuelve el juego del [Sudoku](#) Killer a partir de algoritmos logicos humanos, utiliza la clase [SudokuKiller](#).

```
#include <ResolverSudokuKiller.h>
```

Public Member Functions

- **ResolverSudokuKiller ()**
Constructor por defecto.
- [SudokuKiller ResuelveSudoku](#) ([SudokuKiller](#) aResolver)
utiliza los metodos y algoritmos de la clase para resolver el sudoku dado
- vector< int > [SecuenciaCaja](#) (int i)
Devuelve los posibles valores de la caja/region de 3x3 del sudoku Si el vector resultante contiene un solo elemento, lo inserta automaticamente.
- vector< int > [SecuenciaBloqueKiller](#) (int i)
Obtiene todos los valores posibles dentro de un bloque killer a partir de la suma. Tendrá en cuenta los valores de [SecuenciaCaja\(\)](#) Si el vector resultante contiene un solo elemento, lo inserta automaticamente.
- void **BuscaSecuenciasUnicas** ()
Una Secuencia Única implica que para un bloque el numero de valores es igual a su tamaño, la disposicion de los valores es desconocida. Matemáticamente ciertas sumas de valores para determinados tamaños dan siempre una secuencia Única. Se pueden buscar a fuerza bruta o almacenarlas en una lista Escribe los valores posibles en las casillas.
- void **Regla45** ()
*Comprueba si existen cajas, filas o columnas que contengan bloques killer completos excepto una casilla, cuyo valor se calcula 45*numero de caja, fila, columna - la suma de los bloques killer(metodo sumaBloques)*
- void [DescartaSecuencias](#) (int i)
Descarta las combinaciones de numeros que contengan un numero que ya existe en la caja, fila, columna o bloque killer. Regla basica de sudoku.
- vector< int > [SecuenciasPosibles](#) (int valor, int tama)
Calcula las combinaciones posibles de numeros para un valor concreto y un tamaño de bloque.

3.5.1 Detailed Description

[ResolverSudokuKiller](#) Una clase que resuelve el juego del [Sudoku](#) Killer a partir de algoritmos logicos humanos, utiliza la clase [SudokuKiller](#).

3.5.2 Member Function Documentation

3.5.2.1 DescartaSecuencias()

```
void ResolverSudokuKiller::DescartaSecuencias (
    int i )
```

Descarta las combinaciones de numeros que contengan un numero que ya existe en la caja, fila, columna o bloque killer. Regla basica de sudoku.

Parameters

<i>i</i>	
----------	--

3.5.2.2 ResuelveSudoku()

```
SudokuKiller ResolverSudokuKiller::ResuelveSudoku (
    SudokuKiller aResolver )
```

utiliza los metodos y algoritmos de la clase para resolver el sudoku dado

Parameters

<i>aResolver</i>	: Sudokukiller, El sudoku devuelto por el metodo GenerarSudokuKiller para resolverlo
------------------	--

Returns

el sudoku completamente resuelto

3.5.2.3 SecuenciaBloqueKiller()

```
vector< int > ResolverSudokuKiller::SecuenciaBloqueKiller (
    int i )
```

Obtiene todos los valores posibles dentro de un bloque killer a partir de la suma. Tendrá en cuenta los valores de [SecuenciaCaja\(\)](#) Si el vector resultante contiene un solo elemento, lo inserta automáticamente.

Parameters

<i>i</i>	: int, La posicion del bloque dentro del vector de bloquerKiller
----------	--

Precondition

$i \geq 0 \ \&\& \ i < \text{bloquesKiller.getTama}()$

Returns

Los valores posibles en el bloque dado

3.5.2.4 SecuenciaCaja()

```
vector< int > ResolverSudokuKiller::SecuenciaCaja (
    int i )
```

Devuelve los posibles valores de la caja/region de 3x3 del sudoku Si el vector resultante contiene un solo elemento, lo inserta automaticamente.

Parameters

<i>i</i>	: int, la posicion de la caja solicitada
----------	--

Precondition

$i \geq 0 \ \&\& \ i < \text{caja.getTama}()$

Returns

Los valores posibles de la caja

3.5.2.5 SecuenciasPosibles()

```
vector< int > ResolverSudokuKiller::SecuenciasPosibles (
    int valor,
    int tama )
```

Calcula las combinaciones posibles de numeros para un valor concreto y un tamaño de bloque.

Parameters

<i>valor</i>	: int, El valor total de la secuencia
<i>tama</i>	: int, El tamaño de la secuencia

Returns

las posibles combinaciones para un total y tamaño

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

- include/[ResolverSudokuKiller.h](#)

3.6 Sudoku Class Reference

[Sudoku](#) Clase basica del cual se creará la version killer del sudoku.

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

Public Member Functions

- **Sudoku** ()
Constructor por defecto. Genera un sudoku aleatorio con el metodo GenerarSudoku.
- **Sudoku** ([Matriz](#) m)
Constructor con parámetros.
- int [getValorCasilla](#) (int fil, int col)
Obtiene el valor en la casilla dada.
- [Matriz](#) [getSudoku](#) ()
Devuelve la matriz sudoku.
- [Matriz](#) [GenerarSudoku](#) ()
Genera un sudoku normal terminado a partir de un algoritmo aleatorio y llama a SudokuValido para comprobar si es correcto.

3.6.1 Detailed Description

[Sudoku](#) Clase basica del cual se creará la version killer del sudoku.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 Sudoku()

```
Sudoku::Sudoku (
    Matriz m )
```

Constructor con parámetros.

Parameters

<i>m</i>	: Matriz , La matriz a partir se crea el sudoku, se verifica con el metodo SudokuValido
----------	---

3.6.3 Member Function Documentation

3.6.3.1 GenerarSudoku()

```
Matriz Sudoku::GenerarSudoku ( )
```

Genera un sudoku normal terminado a partir de un algoritmo aleatorio y llama a SudokuValido para comprobar si es correcto.

Returns

La matriz [Sudoku](#) generada

3.6.3.2 getSudoku()

```
Matriz Sudoku::getSudoku ( )
```

Devuelve la matriz sudoku.

Returns

3.6.3.3 getValorCasilla()

```
int Sudoku::getValorCasilla (
    int fil,
    int col )
```

Obtiene el valor en la casilla dada.

Parameters

<i>fil</i>	: int, Posicion en la fila
<i>col</i>	: int, Posicion en la columna

Returns

El valor de la casilla

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

- include/[SudokuKiller.h](#)

3.7 SudokuKiller Class Reference

SudokuKiller Una clase para crear y gestionar el sudoku killer completo, se utiliza la clase [Matriz](#) y la utiliza [ResolverSudokuKiller](#).

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

Public Member Functions

- **SudokuKiller ()**
Constructor por defecto, llama a Generarsudoku de la clase [Sudoku](#) y lo devuelve como parametro a Generar← SudokuKillerCompleto.
- **SudokuKiller ([Matriz](#) sudoku)**
Constructor con parametros. Crea un sudoku nuevo a partir de la matriz dada y llama al siguiente constructor. Si es incorrecta, se llama al constructor por defecto.
- **SudokuKiller ([Sudoku](#) sudoku)**
Constructor con parametros, llama al metodo GenerarSudokuKillerCompleto.
- **SudokuKiller [GenerarSudokuKiller](#) ()**
Devuelve un objeto [SudokuKiller](#) vacio listo para jugar. Usa el metodo GenerarSudokuKillerCompleto, copia el objeto e inicializa sus valores a null para el jugador.
- **bool [EstaResuelto](#) ([SudokuKiller](#) resuelto)**
Verifica si el sudoku dado está completo y cumple las reglas para considerarse resuelto.

3.7.1 Detailed Description

SudokuKiller Una clase para crear y gestionar el sudoku killer completo, se utiliza la clase [Matriz](#) y la utiliza [ResolverSudokuKiller](#).

3.7.2 Constructor & Destructor Documentation

3.7.2.1 SudokuKiller() [1/2]

```
SudokuKiller::SudokuKiller (
    Matriz sudoku )
```

Constructor con parametros. Crea un sudoku nuevo a partir de la matriz dada y llama al siguiente constructor. Si es incorrecta, se llama al constructor por defecto.

Parameters

<code>sudoku</code>	: Matriz
---------------------	--------------------------

3.7.2.2 SudokuKiller() [2/2]

```
SudokuKiller::SudokuKiller (
    Sudoku sudoku )
```

Constructor con parametros, llama al metodo GenerarSudokuKillerCompleto.

Parameters

<i>sudoku</i>	: Sudoku
---------------	--------------------------

3.7.3 Member Function Documentation

3.7.3.1 EstaResuelto()

```
bool SudokuKiller::EstaResuelto (
    SudokuKiller resuelto )
```

Verifica si el sudoku dado está completo y cumple las reglas para considerarse resuelto.

Parameters

<i>resuelto</i>	: SudokuKiller , El sudoku a comprobar
-----------------	--

Returns

true si está completamente resuelto, false en otro caso

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

- include/[SudokuKiller.h](#)

Chapter 4

File Documentation

4.1 include/BloquesSudoku.h File Reference

Fichero cabecera de las clases [Bloque](#) y [BloquesSudoku](#).

```
#include <iostream>
#include <vector>
#include "Casilla.h"
```

Classes

- class [Bloque](#)
[Bloque](#) Una clase que agrupa las casillas de una [Matriz](#), se utiliza en la clase [BloquesSudoku](#) para la clase [SudokuKiller](#).
- class [BloquesSudoku](#)
[BloquesSudoku](#) Una clase que contiene los bloques del [Sudoku](#), se utiliza para la clase [SudokuKiller](#) y [ResolverSudokuKiller](#).

Macros

- `#define TAMA_MAX = 5`

Functions

- bool [seSolapaBloques](#) ([Bloque](#) izq, [Bloque](#) dcha)
Comprueba que 2 bloques no se solapan.
- bool [esMismoBloque](#) ([Bloque](#) izq, [Bloque](#) dcha)
Compara si 2 bloques son iguales.

4.1.1 Detailed Description

Fichero cabecera de las clases `Bloque` y `BloquesSudoku`.

Author

López Ramírez, Yeray

Date

Octubre 2021

4.1.2 Function Documentation

4.1.2.1 `esMismoBloque()`

```
bool esMismoBloque (
    Bloque izq,
    Bloque dcha )
```

Compara si 2 bloques son iguales.

Parameters

<i>izq</i>	
<i>dcha</i>	

Returns

true cuando las casillas que lo forman son iguales, false en otro caso

4.1.2.2 `seSolapaBloques()`

```
bool seSolapaBloques (
    Bloque izq,
    Bloque dcha )
```

Comprueba que 2 bloques no se solapan.

Parameters

<i>izq</i>	
<i>dcha</i>	

Returns

true si alguna casilla coincide, false en otro caso

4.2 BloquesSudoku.h

[Go to the documentation of this file.](#)

```

1
2
3
4
5
6
7
8 #ifndef BLOQUESSUDOKU_H
9 #define BLOQUESSUDOKU_H
10
11 #include <iostream>
12 #include <vector>
13 #include "Casilla.h"
14
15 #define TAMA_MAX = 5
16
17 using namespace std;
18
19 class Bloque{
20     private:
21         int tama, suma; //tamaño de bloque y la suma de sus valores
22         vector<pair> casillas; //vector de pares de posiciones de casillas(fil, col)
23
24     public:
25
26         Bloque();
27
28         Bloque(int tam);
29
30         Bloque(const Bloque &b);
31
32         int getTamanio();
33
34         int getSumaBloque();
35
36         Casilla getCasilla(int i);
37
38         void setCasilla(int i, int fil, int col, int val);
39
40         bool aniadirCasilla(int val, int fil, int col);
41
42         bool eliminarCasilla(int i);
43
44         bool esCasillaContigua(Casilla nueva);
45
46         void sumaValores();
47
48 };
49
50 class BloquesSudoku{
51     private:
52         int numBloques;
53         vector<Bloque> BloquesSudoku; //vector de bloques
54     public:
55
56         BloquesSudoku();
57
58         BloquesSudoku(int num);
59
60         Bloque getBloquesSudoku(int i);
61
62         void setBloquesSudoku(int i, Bloque b);
63
64         void setBloquesSudoku(int i, vector<Casilla>);
65
66         bool aniadirBloque(Bloque b);
67
68         bool eliminarBloque(int i);
69
70         int sumaBloques(vector<int> ibloques);
71
72 };
73
74 bool seSolapaBloques(Bloque izq, Bloque dcha);
75
76 bool esMismoBloque(Bloque izq, Bloque dcha);
77
78 #endif /* BLOQUESSUDOKU_H */
79
80

```

4.3 include/Casilla.h File Reference

Fichero cabecera de la clase [Casilla](#).

```
#include "BloquesSudoku.h"
```

Classes

- class [Casilla](#)

[Casilla](#) Una clase que contiene los datos básicos de una casilla del sudoku, a partir de una matriz sudoku proporcionada por la clase [SudokuKiller](#).

Functions

- bool [mismaCasilla](#) (const [Casilla](#) &izq, const [Casilla](#) &dcha)
comprueba que dos casillas son iguales

4.3.1 Detailed Description

Fichero cabecera de la clase [Casilla](#).

Author

López Ramírez, Yeray

Date

28 de Octubre de 2021

4.3.2 Function Documentation

4.3.2.1 mismaCasilla()

```
bool mismaCasilla (  
    const Casilla & izq,  
    const Casilla & dcha )
```

comprueba que dos casillas son iguales

Parameters

<i>izq</i>	: Casilla
<i>dcha</i>	: Casilla

Returns

true cuando sus posiciones son iguales, false en otro caso

4.4 Casilla.h

[Go to the documentation of this file.](#)

```

1
9 #ifndef CASILLA_H
10 #define CASILLA_H
11
12 #include "BloquesSudoku.h"
13
14 class Casilla
15 {
16 private:
17     int fila, columna, valor;
18
19 public:
20     Casilla();
21
22     Casilla(int fil, int col);
23
24     Casilla(int fil, int col, int valor);
25     bool isEmpty() const;
26
27     const int getFila() const;
28     const int getColumna() const;
29     const int getValor() const;
30
31     void setFila(const int fil);
32     void setColumna(const int col);
33     void setValor(const int valor);
34
35     void setPosicion(int fil, int col);
36
37     void setCasilla(int fil, int col, int valor);
38
39     std::string to_string() const;
40
41     void initDefault();
42 };
43
44 bool mismaCasilla(const Casilla & izq, const Casilla & dcha);
45
46 #endif /* CASILLA_H */

```

4.5 include/Matriz.h File Reference

Fichero cabecera de la clase [Matriz](#).

```
#include <iostream>
```

Classes

- class [Matriz< M >](#)

Matriz Un template para el tipo de datos abstracto [Matriz](#), se utiliza para las clases [BloquesSudoku](#), [SudokuKiller](#) y [ResolverSudokuKiller](#).

4.5.1 Detailed Description

Fichero cabecera de la clase [Matriz](#).

Author

estudiante1: López Ramírez, Yeray

Date

28 de Octubre de 2021

4.6 Matriz.h

[Go to the documentation of this file.](#)

```
1
9 #ifndef MATRIZ_H
10 #define MATRIZ_H
11 #include <iostream>
12
13 using namespace std;
14
15
21 template <class M>
22 class Matriz
23 {
24     private:
25         M **matriz;
26         int filas, columnas;
27
31         void init();
32
33     public:
34
39         Matriz();
40
48         Matriz(int f, int c);
49
53         ~Matriz();
54
64         int getCasilla(int fil, int col);
65
70         int getFila();
71
76         int getColumna();
77
87         void insertarValor(int fil, int col, const M &valor);
88
96         void eliminarValor(int fil, int col);
97
98 };
99
100 #endif /* MATRIZ_H */
```

4.7 include/ResolverSudokuKiller.h File Reference

Fichero cabecera de la clase [ResolverSudokuKiller](#).

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

Classes

- class [ResolverSudokuKiller](#)

[ResolverSudokuKiller](#) Una clase que resuelve el juego del [Sudoku](#) Killer a partir de algoritmos logicos humanos, utiliza la clase [SudokuKiller](#).

4.7.1 Detailed Description

Fichero cabecera de la clase [ResolverSudokuKiller](#).

Author

López Ramírez, Yeray

Date

Octubre 2021

4.8 ResolverSudokuKiller.h

[Go to the documentation of this file.](#)

```
1
2
3
4
5
6
7
8
9 #ifndef RESOLVERSUDOKUKILLER_H
10 #define RESOLVERSUDOKUKILLER_H
11
12 #include <SudokuKiller.h>
13
14
15
16
17
18
19 class ResolverSudokuKiller{
20     private:
21         int ***sudoku; //Matriz tridimensional ya que cada casilla contendrá un vector de posibles
22         valores
23         BloquesSudoku caja;
24         BloquesSudoku bloquesKiller;
25     public:
26
27         ResolverSudokuKiller();
28
29         SudokuKiller ResuelveSudoku(SudokuKiller aResolver);
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46     vector<int> SecuenciaCaja(int i);
47
48
49
50
51
52
53
54
55
56     vector<int> SecuenciaBloqueKiller(int i);
57
58
59
60
61
62
63
64
65     void BuscaSecuenciasUnicas();
66
67
68
69
70
71
72     void Regla45();
73
74
75
76
77
78
79     void DescartaSecuencias(int i);
80
81
82
83
84
85
86
87
88     vector<int> SecuenciasPosibles(int valor, int tama);
89
90 };
91
92
93 #endif /* RESOLVERSUDOKUKILLER_H */
94
```

4.9 include/SudokuKiller.h File Reference

Fichero cabecera de la clase [SudokuKiller](#).

```
#include <iostream>
#include <vector>
#include "Matriz.h"
```

Classes

- class [Sudoku](#)
[Sudoku](#) Clase basica del cual se creará la version killer del sudoku.
- class [SudokuKiller](#)
[SudokuKiller](#) Una clase para crear y gestionar el sudoku killer completo, se utiliza la clase [Matriz](#) y la utiliza [ResolverSudokuKiller](#).

Functions

- bool [SudokuValido](#) ([Matriz](#) M)
Comprueba que la matriz dada cumple las reglas del sudoku.
- bool [BloquesValido](#) ([SudokuKiller](#) sudoku)
verifica que los numeros de la matriz sudoku coinciden con el valor suma de los bloques killer
- bool [SudokuKillerValido](#) ([SudokuKiller](#) sudoku)
Verifica que el sudoku dado es un killer y que todo está correcto.

4.9.1 Detailed Description

Fichero cabecera de la clase [SudokuKiller](#).

Author

López Ramírez, Yeray

Date

Octubre 2021

4.9.2 Function Documentation

4.9.2.1 BloquesValido()

```
bool BloquesValido (
    SudokuKiller sudoku )
```

verifica que los numeros de la matriz sudoku coinciden con el valor suma de los bloques killer

Parameters

<i>sudoku</i>	: SudokuKiller ,
---------------	----------------------------------

Returns

true si suma==sumaValores(), false en otro caso

4.9.2.2 SudokuKillerValido()

```
bool SudokuKillerValido (
    SudokuKiller sudoku )
```

Verifica que el sudoku dado es un killer y que todo está correcto.

Parameters

<i>sudoku</i>	: SudokuKiller , El sudoku a verificar
---------------	--

Returns

true si cumple las condiciones del killer y los metodos SudokuValido y Bloques valido dan true, false en otro caso

4.9.2.3 SudokuValido()

```
bool SudokuValido (
    Matriz M )
```

Comprueba que la matriz dada cumple las reglas del sudoku.

Returns

true si los valores cumplen las reglas del sudoku, false en otro caso -No se repiten numeros de la misma fila o columna o caja(3x3)

4.10 SudokuKiller.h

[Go to the documentation of this file.](#)

```
1
9 #ifndef SUDOKUKILLER_H
10 #define SUDOKUKILLER_H
11
12 #include <iostream>
13 #include <vector>
14 #include "Matriz.h"
15
16 using namespace std;
17
22 class Sudoku
23 {
24     private:
25         Matriz sudoku();
26         BloquesSudoku cajas;
27
31         void init();
32     public:
37         Sudoku();
38
44         Sudoku(Matriz m);
45
52         int getValorCasilla(int fil, int col);
```

```

53
54     Matriz getSudoku();
55
56     Matriz GenerarSudoku();
57 };
58
59 class SudokuKiller{
60     private:
61         Matriz sudokuKiller;
62         BloquesSudoku cajas;
63         BloquesSudoku bloquesKiller;
64
65     void GenerarSudokuKillerCompleto(Sudoku sudoku);
66
67     public:
68         SudokuKiller();
69
70         SudokuKiller(Matriz sudoku);
71
72         SudokuKiller(Sudoku sudoku);
73
74         SudokuKiller GenerarSudokuKiller();
75
76         bool EstaResuelto(SudokuKiller resuelto);
77 };
78
79 bool SudokuValido(Matriz M);
80
81 bool BloquesValido(SudokuKiller sudoku);
82
83 bool SudokuKillerValido(SudokuKiller sudoku);
84
85 #endif /* SUDOKUKILLER_H */
86

```

4.11 src/main.cpp File Reference

Fichero [main.cpp](#).

```

#include <iostream>
#include <ResolverSudokuKiller.h>

```

Functions

- int [main](#) (int argc, char **argv)
Pseudo main de ejemplo para resolver un sudokuKiller.

4.11.1 Detailed Description

Fichero [main.cpp](#).

Author

López Ramírez, Yeray
Castillo Ucles, Jaime

Date

Octubre 2021

4.11.2 Function Documentation

4.11.2.1 main()

```
int main (
    int argc,
    char ** argv )
```

Pseudo main de ejemplo para resolver un sudokuKiller.

Parameters

<i>argc</i>	Numero de argumentos pasados al main
<i>argv</i>	Cadenas de string que representan los argumentos

Returns

Index

- aniadirBloque
 - BloquesSudoku, [10](#)
- aniadirCasilla
 - Bloque, [6](#)
- Bloque, [5](#)
 - aniadirCasilla, [6](#)
 - Bloque, [6](#)
 - eliminarCasilla, [7](#)
 - esCasillaContigua, [7](#)
 - getCasilla, [8](#)
 - getSumaBloque, [8](#)
 - getTamanio, [8](#)
 - setCasilla, [8](#)
- BloquesSudoku, [9](#)
 - aniadirBloque, [10](#)
 - BloquesSudoku, [10](#)
 - eliminarBloque, [10](#)
 - getBloquesSudoku, [11](#)
 - setBloquesSudoku, [11](#)
 - sumaBloques, [13](#)
- BloquesSudoku.h
 - esMismoBloque, [30](#)
 - seSolapaBloques, [30](#)
- BloquesValido
 - SudokuKiller.h, [36](#)
- Casilla, [13](#)
 - Casilla, [14](#)
 - getColumna, [15](#)
 - getFila, [15](#)
 - getValor, [15](#)
 - isEmpty, [15](#)
 - setCasilla, [16](#)
 - setColumna, [16](#)
 - setFila, [16](#)
 - setPosicion, [17](#)
 - setValor, [17](#)
 - to_string, [17](#)
- Casilla.h
 - mismaCasilla, [32](#)
- DescartaSecuencias
 - ResolverSudokuKiller, [22](#)
- eliminarBloque
 - BloquesSudoku, [10](#)
- eliminarCasilla
 - Bloque, [7](#)
- eliminarValor
 - Matriz< M >, [19](#)
- esCasillaContigua
 - Bloque, [7](#)
- esMismoBloque
 - BloquesSudoku.h, [30](#)
- EstaResuelto
 - SudokuKiller, [27](#)
- GenerarSudoku
 - Sudoku, [25](#)
- getBloquesSudoku
 - BloquesSudoku, [11](#)
- getCasilla
 - Bloque, [8](#)
 - Matriz< M >, [19](#)
- getColumna
 - Casilla, [15](#)
 - Matriz< M >, [20](#)
- getFila
 - Casilla, [15](#)
 - Matriz< M >, [20](#)
- getSudoku
 - Sudoku, [25](#)
- getSumaBloque
 - Bloque, [8](#)
- getTamanio
 - Bloque, [8](#)
- getValor
 - Casilla, [15](#)
- getValorCasilla
 - Sudoku, [25](#)
- include/BloquesSudoku.h, [29](#), [31](#)
- include/Casilla.h, [32](#), [33](#)
- include/Matriz.h, [33](#), [34](#)
- include/ResolverSudokuKiller.h, [34](#), [35](#)
- include/SudokuKiller.h, [35](#), [37](#)
- insertarValor
 - Matriz< M >, [20](#)
- isEmpty
 - Casilla, [15](#)
- main
 - main.cpp, [39](#)
- main.cpp
 - main, [39](#)
- Matriz
 - Matriz< M >, [18](#)
- Matriz< M >, [18](#)
 - eliminarValor, [19](#)

- getCasilla, [19](#)
- getColumna, [20](#)
- getFila, [20](#)
- insertarValor, [20](#)
- Matriz, [18](#)
- mismaCasilla
 - Casilla.h, [32](#)
- ResolverSudokuKiller, [21](#)
 - DescartaSecuencias, [22](#)
 - ResuelveSudoku, [22](#)
 - SecuenciaBloqueKiller, [22](#)
 - SecuenciaCaja, [23](#)
 - SecuenciasPosibles, [23](#)
- ResuelveSudoku
 - ResolverSudokuKiller, [22](#)
- SecuenciaBloqueKiller
 - ResolverSudokuKiller, [22](#)
- SecuenciaCaja
 - ResolverSudokuKiller, [23](#)
- SecuenciasPosibles
 - ResolverSudokuKiller, [23](#)
- seSolapaBloques
 - BloquesSudoku.h, [30](#)
- setBloquesSudoku
 - BloquesSudoku, [11](#)
- setCasilla
 - Bloque, [8](#)
 - Casilla, [16](#)
- setColumna
 - Casilla, [16](#)
- setFila
 - Casilla, [16](#)
- setPosicion
 - Casilla, [17](#)
- setValor
 - Casilla, [17](#)
- src/main.cpp, [38](#)
- Sudoku, [24](#)
 - GenerarSudoku, [25](#)
 - getSudoku, [25](#)
 - getValorCasilla, [25](#)
 - Sudoku, [24](#)
- SudokuKiller, [26](#)
 - EstaResuelto, [27](#)
 - SudokuKiller, [26](#)
- SudokuKiller.h
 - BloquesValido, [36](#)
 - SudokuKillerValido, [37](#)
 - SudokuValido, [37](#)
- SudokuKillerValido
 - SudokuKiller.h, [37](#)
- SudokuValido
 - SudokuKiller.h, [37](#)
- sumaBloques
 - BloquesSudoku, [13](#)
- to_string