

Billar

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 Bola Class Reference	5
3.1.1 Detailed Description	6
3.1.2 Constructor & Destructor Documentation	6
3.1.2.1 Bola()	6
3.1.3 Member Function Documentation	6
3.1.3.1 ChoqueElastico()	6
3.1.3.2 Getm()	7
3.1.3.3 Getr()	7
3.1.3.4 Getvx()	7
3.1.3.5 Getvy()	7
3.1.3.6 Getx()	7
3.1.3.7 Gety()	7
3.1.3.8 Inicie()	7
3.1.3.9 Muevase()	8
3.1.3.10 ResuelvaColisionParedesRobusto()	8
3.1.3.11 ResuelvaColisionParedesSimple()	9
3.2 Caja Class Reference	10
3.2.1 Detailed Description	10
3.2.2 Constructor & Destructor Documentation	10
3.2.2.1 Caja()	11
3.2.3 Member Function Documentation	11
3.2.3.1 Defina()	11
3.2.3.2 GetH()	11
3.2.3.3 GetW()	12
3.3 Sistema Class Reference	13
3.3.1 Detailed Description	13
3.3.2 Member Function Documentation	13
3.3.2.1 DefinaCaja()	13
3.3.2.2 Encabezado()	14
3.3.2.3 Guarde()	14
3.3.2.4 InicialiceRejilla()	14
3.3.2.5 Paso()	15
3.3.2.6 Reserve()	16
3.3.2.7 SeleccioneIntegrador()	16
4 File Documentation	17

4.1 Bola.h File Reference	17
4.1.1 Detailed Description	18
4.2 Bola.h	18
4.3 Caja.h File Reference	18
4.3.1 Detailed Description	19
4.4 Caja.h	19
4.5 Sistema.h File Reference	20
4.5.1 Detailed Description	21
4.5.2 Enumeration Type Documentation	21
4.5.2.1 Integrador	21
4.6 Sistema.h	21
4.7 Bola.cpp File Reference	22
4.7.1 Detailed Description	22
4.8 Caja.cpp File Reference	22
4.8.1 Detailed Description	23
4.9 Sistema.cpp File Reference	23
4.9.1 Detailed Description	24
Index	25

Chapter 1

Class Index

1.1 Class List

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

Bola	Representa una bola en una simulación de billar 2D	5
Caja	Representa el recinto rectangular donde se mueven las bolas	10
Sistema	Representa el sistema completo de simulación de un billar de N bolas	13

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Bola.h	Define la clase Bola que representa una partícula (bola) en el sistema de billar	17
Caja.h	Define la clase Caja que representa el contenedor rectangular del sistema de billar	18
Sistema.h	Define la clase Sistema , que representa el conjunto de bolas y la caja donde se realiza la simulación	20
Bola.cpp	Implementación de los métodos de la clase Bola	22
Caja.cpp	Implementación de la clase Caja	22
Sistema.cpp	Implementación de la clase Sistema , que coordina la simulación del billar	23

Chapter 3

Class Documentation

3.1 Bola Class Reference

Representa una bola en una simulación de billar 2D.

```
#include <Bola.h>
```

Public Member Functions

- **Bola ()**
Constructor por defecto. Inicializa una bola en el origen con masa y radio por defecto.
- void **Inicie** (double x0, double y0, double vx0, double vy0, double m0, double r0)
Inicializa los parámetros físicos de la bola.
- void **Muevase** (double dt)
Actualiza la posición de la bola usando integración simple.
- void **ResuelvaColisionParedesSimple** (const **Caja** &C)
Resuelve colisiones simples con las paredes de la caja.
- void **ResuelvaColisionParedesRobusto** (const **Caja** &C)
Resuelve colisiones robustas con las paredes de la caja.
- void **ChoqueElastico** (**Bola** &otra)
Resuelve una colisión elástica con otra bola.
- double **Getx** () const
Retorna la coordenada x.
- double **Gety** () const
Retorna la coordenada y.
- double **Getvx** () const
Retorna la componente vx.
- double **Getvy** () const
Retorna la componente vy.
- double **Getm** () const
Retorna la masa.
- double **Getr** () const
Retorna el radio.

3.1.1 Detailed Description

Representa una bola en una simulación de billar 2D.

Cada bola posee propiedades físicas básicas (posición, velocidad, masa y radio) y puede interactuar elásticamente con otras bolas y con las paredes de una caja.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Bola()

```
Bola::Bola ( )
```

Constructor por defecto. Inicializa una bola en el origen con masa y radio por defecto.

Constructor por defecto.

Inicializa la bola en el origen con velocidad cero, masa unitaria y radio 0.1.

3.1.3 Member Function Documentation

3.1.3.1 ChoqueElastico()

```
void Bola::ChoqueElastico (  
    Bola & otra )
```

Resuelve una colisión elástica con otra bola.

Resuelve una colisión elástica entre dos bolas.

Conserva el momento lineal y la energía cinética en el sistema de dos bolas.

Parameters

<i>otra</i>	Referencia a la otra bola.
-------------	----------------------------

Conserva el momento lineal y la energía cinética del sistema de dos bolas. Si las bolas se solapan, también realiza una corrección de posición para separarlas.

Parameters

<i>otra</i>	Referencia a la otra bola con la que colisiona.
-------------	---

< Impulso escalar.

3.1.3.2 Getm()

```
double Bola::Getm ( ) const [inline]
```

Retorna la masa.

3.1.3.3 Getr()

```
double Bola::Getr ( ) const [inline]
```

Retorna el radio.

3.1.3.4 Getvx()

```
double Bola::Getvx ( ) const [inline]
```

Retorna la componente vx.

3.1.3.5 Getvy()

```
double Bola::Getvy ( ) const [inline]
```

Retorna la componente vy.

3.1.3.6 Getx()

```
double Bola::Getx ( ) const [inline]
```

Retorna la coordenada x.

3.1.3.7 Gety()

```
double Bola::Gety ( ) const [inline]
```

Retorna la coordenada y.

3.1.3.8 Inicie()

```
void Bola::Inicie (
    double x0,
    double y0,
    double vx0,
    double vy0,
    double m0,
    double r0 )
```

Inicializa los parámetros físicos de la bola.

Parameters

<i>x0</i>	Posición inicial en x.
<i>y0</i>	Posición inicial en y.
<i>vx0</i>	Velocidad inicial en x.
<i>vy0</i>	Velocidad inicial en y.
<i>m0</i>	Masa.
<i>r0</i>	Radio.

3.1.3.9 Muevase()

```
void Bola::Muevase (
    double dt )
```

Actualiza la posición de la bola usando integración simple.

Avanza la posición de la bola según su velocidad.

Parameters

<i>dt</i>	Paso de tiempo.
-----------	-----------------

Este método no considera colisiones; simplemente actualiza la posición usando el paso de tiempo *dt*.

Parameters

<i>dt</i>	Paso de tiempo.
-----------	-----------------

3.1.3.10 ResuelvaColisionParedesRobusto()

```
void Bola::ResuelvaColisionParedesRobusto (
    const Caja & C )
```

Resuelve colisiones robustas con las paredes de la caja.

Corrige tanto la posición como la velocidad para evitar que la bola atraviese las paredes.

Parameters

<i>C</i>	Caja con la que colisiona.
----------	----------------------------

Además de invertir la velocidad, este método ajusta la posición de la bola para evitar que atraviese las paredes, mejorando la estabilidad numérica de la simulación.

Parameters

C	Caja con la que colisiona.
---	----------------------------

< Corrige posición en x. Here is the call graph for this function:



3.1.3.11 ResuelvaColisionParedesSimple()

```
void Bola::ResuelvaColisionParedesSimple (const Caja & C)
```

Resuelve colisiones simples con las paredes de la caja.

Invierte la velocidad al chocar con una pared, pero puede causar adherencia si el tiempo de integración es grande.

Parameters

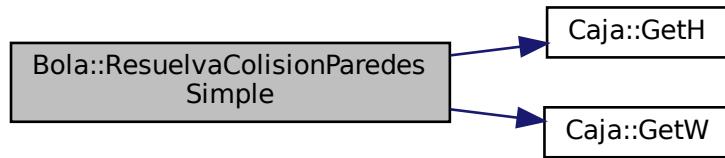
C	Caja con la que colisiona.
---	----------------------------

Si la bola toca una pared y se mueve hacia ella, su velocidad se invierte. Este método no corrige la posición y puede causar “pegado” si el paso de tiempo es grande.

Parameters

C	Caja con la que colisiona.
---	----------------------------

Here is the call graph for this function:



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

- [Bola.h](#)
- [Bola.cpp](#)

3.2 Caja Class Reference

Representa el recinto rectangular donde se mueven las bolas.

```
#include <Caja.h>
```

Public Member Functions

- [Caja \(\)](#)
Constructor por defecto. Crea una caja sin dimensiones definidas.
- void [Defina \(double W_, double H_\)](#)
Define las dimensiones de la caja.
- double [GetW \(\) const](#)
Retorna el ancho de la caja.
- double [GetH \(\) const](#)
Retorna el alto de la caja.

3.2.1 Detailed Description

Representa el recinto rectangular donde se mueven las bolas.

La caja define los límites espaciales del sistema de billar y se usa para detectar y resolver colisiones con las paredes.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 Caja()

```
Caja::Caja ( )
```

Constructor por defecto. Crea una caja sin dimensiones definidas.

Constructor por defecto.

Inicializa una caja de dimensiones 1x1.

3.2.3 Member Function Documentation

3.2.3.1 Defina()

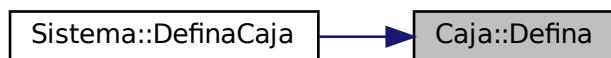
```
void Caja::Defina (
    double W,
    double H )
```

Define las dimensiones de la caja.

Parameters

$W \leftarrow$	Ancho de la caja.
$H \leftarrow$	Alto de la caja.

Here is the caller graph for this function:

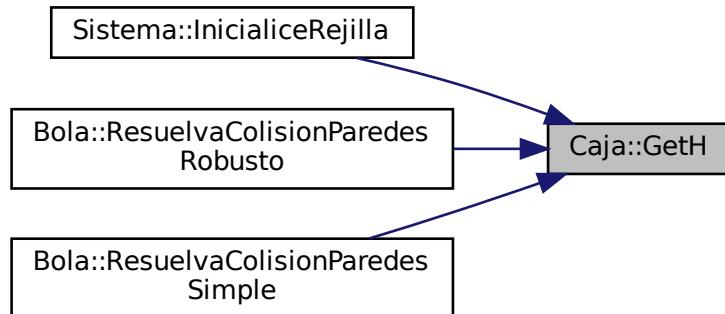


3.2.3.2 GetH()

```
double Caja::GetH ( ) const [inline]
```

Retorna el alto de la caja.

Here is the caller graph for this function:

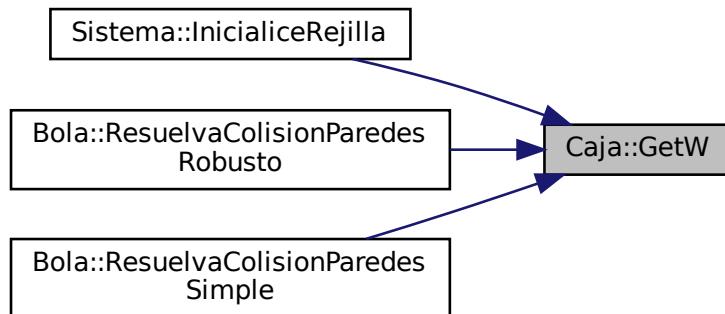


3.2.3.3 GetW()

```
double Caja::GetW ( ) const [inline]
```

Retorna el ancho de la caja.

Here is the caller graph for this function:



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

- [Caja.h](#)
- [Caja.cpp](#)

3.3 Sistema Class Reference

Representa el sistema completo de simulación de un billar de N bolas.

```
#include <Sistema.h>
```

Public Member Functions

- void **DefinaCaja** (double W, double H)
Define las dimensiones de la caja contenedora.
- void **Reserve** (int N)
Reserva memoria para un número determinado de bolas.
- void **InicialiceRejilla** (double m, double r, double v_max, bool alterna=false)
Inicializa las bolas en una configuración de rejilla.
- void **SeleccioneIntegrador** (const std::string &nombre)
Selecciona el integrador a utilizar.
- void **Paso** (double dt)
Ejecuta un paso temporal del sistema según el integrador actual.
- void **Encabezado** (std::ofstream &f)
Escribe el encabezado de columnas en un archivo de salida.
- void **Guarde** (std::ofstream &f, double t)
Guarda el estado actual del sistema en un archivo.

3.3.1 Detailed Description

Representa el sistema completo de simulación de un billar de N bolas.

Contiene la caja, las bolas y el método de integración seleccionado. Se encarga de actualizar el sistema en cada paso de tiempo y guardar los resultados.

3.3.2 Member Function Documentation

3.3.2.1 DefinaCaja()

```
void Sistema::DefinaCaja (
    double W,
    double H )
```

Define las dimensiones de la caja contenedora.

Define las dimensiones de la caja de simulación.

Parameters

<i>W</i>	Ancho de la caja.
<i>H</i>	Alto de la caja.

Here is the call graph for this function:



3.3.2.2 Encabezado()

```
void Sistema::Encabezado (
    std::ofstream & f )
```

Escribe el encabezado de columnas en un archivo de salida.

Parameters

<i>f</i>	Flujo de salida (archivo abierto).
<i>f</i>	Archivo de salida abierto.

3.3.2.3 Guarde()

```
void Sistema::Guarda (
    std::ofstream & f,
    double t )
```

Guarda el estado actual del sistema en un archivo.

Parameters

<i>f</i>	Flujo de salida (archivo abierto).
<i>t</i>	Tiempo actual de la simulación.
<i>f</i>	Archivo de salida abierto.
<i>t</i>	Tiempo actual de la simulación.

3.3.2.4 InicialiceRejilla()

```
void Sistema::InicialiceRejilla (
    double m,
```

```
double r,
double vmax,
bool alterna = false )
```

Inicializa las bolas en una configuración de rejilla.

Inicializa las bolas en una distribución de rejilla.

Las posiciones se distribuyen uniformemente y las velocidades pueden asignarse aleatoriamente hasta un valor máximo dado.

Parameters

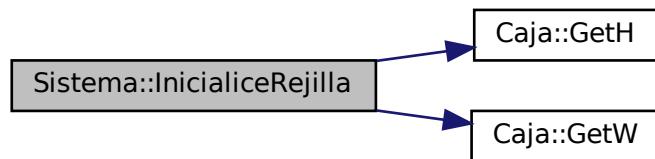
<i>m</i>	Masa de cada bola.
<i>r</i>	Radio de cada bola.
<i>v_max</i>	Velocidad máxima inicial.
<i>alterna</i>	Si es verdadero, alterna el sentido de las velocidades.

Las bolas se ubican uniformemente dentro de la caja y se les asignan velocidades aleatorias hasta un máximo *vmax*.

Parameters

<i>m</i>	Masa de cada bola.
<i>r</i>	Radio de cada bola.
<i>vmax</i>	Velocidad máxima inicial.
<i>alterna</i>	Si es true, alterna la dirección de las velocidades.

Here is the call graph for this function:



3.3.2.5 Paso()

```
void Sistema::Paso (
    double dt )
```

Ejecuta un paso temporal del sistema según el integrador actual.

Realiza un paso temporal del sistema según el integrador actual.

Parameters

<i>dt</i>	Paso de tiempo.
-----------	-----------------

3.3.2.6 Reserve()

```
void Sistema::Reserve (
    int N )
```

Reserva memoria para un número determinado de bolas.

Reserva memoria para N bolas en la simulación.

Parameters

<i>N</i>	Número de bolas a reservar.
<i>N</i>	Número de bolas.

3.3.2.7 SeleccionIntegrador()

```
void Sistema::SeleccionIntegrador (
    const std::string & nombre )
```

Selecciona el integrador a utilizar.

Selecciona el método de integración temporal.

Parameters

<i>nombre</i>	Nombre del integrador ("Euler" o "Verlet").
<i>nombre</i>	Nombre del integrador ("euler" o "verlet").

Exceptions

<i>std::invalid_argument</i>	Si el nombre no es válido.
------------------------------	----------------------------

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

- [Sistema.h](#)
- [Sistema.cpp](#)

Chapter 4

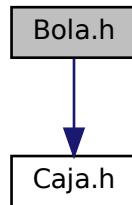
File Documentation

4.1 Bola.h File Reference

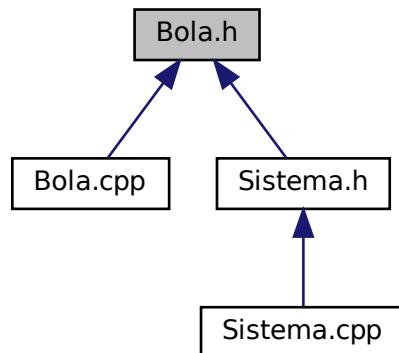
Define la clase **Bola** que representa una partícula (bola) en el sistema de billar.

```
#include "Caja.h"
```

Include dependency graph for Bola.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **Bola**
Representa una bola en una simulación de billar 2D.

4.1.1 Detailed Description

Define la clase **Bola** que representa una partícula (bola) en el sistema de billar.

Esta clase contiene la posición, velocidad, masa y radio de una bola, así como los métodos necesarios para moverla y resolver colisiones con otras bolas o con las paredes.

4.2 Bola.h

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

```

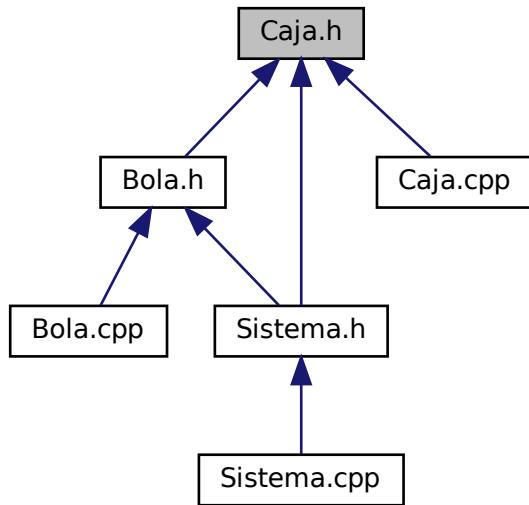
1
9 #ifndef BOLA_H
10 #define BOLA_H
11
12 #include "Caja.h"
13
21 class Bola {
22 private:
23     double x, y;
24     double vx, vy;
25     double m;
26     double r;
27
28 public:
30     Bola();
31
41     void Inicie(double x0, double y0, double vx0, double vy0, double m0, double r0);
42
47     void Muevase(double dt);
48
57     void ResuelvaColisionParedesSimple(const Caja& C);
58
66     void ResuelvaColisionParedesRobusto(const Caja& C);
67
74     void ChoqueElastico(Bola& otra);
75
76     // ===== Getters =====
77     double Getx() const { return x; }
78     double Gety() const { return y; }
79     double Getvx() const { return vx; }
80     double Getvy() const { return vy; }
81     double Getm() const { return m; }
82     double Getr() const { return r; }
83 };
84
85 #endif

```

4.3 Caja.h File Reference

Define la clase **Caja** que representa el contenedor rectangular del sistema de billar.

This graph shows which files directly or indirectly include this file:



Classes

- class [Caja](#)
Representa el recinto rectangular donde se mueven las bolas.

4.3.1 Detailed Description

Define la clase [Caja](#) que representa el contenedor rectangular del sistema de billar.

La clase [Caja](#) define las dimensiones del área de simulación (ancho y alto), dentro de la cual se mueven las bolas y con cuyas paredes pueden colisionar.

4.4 Caja.h

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

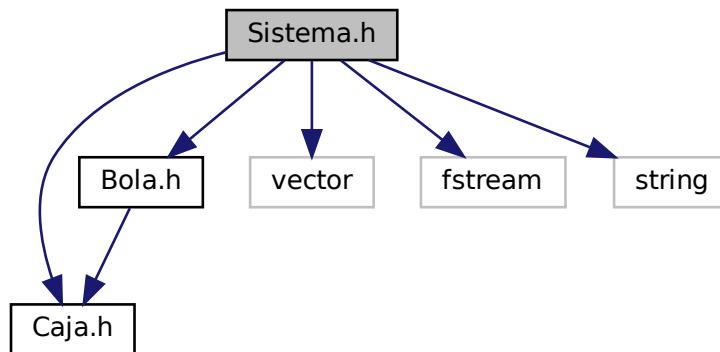
```

1
2 #ifndef CAJA_H
3 #define CAJA_H
4
5 class Caja {
6 private:
7     double W;
8     double H;
9
10 public:
11     Caja();
12
13     void Defina(double W_, double H_);
14
15     double GetW() const { return W; }
16
17     double GetH() const { return H; }
18 };
19 #endif
  
```

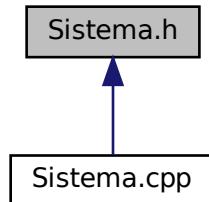
4.5 Sistema.h File Reference

Define la clase [Sistema](#), que representa el conjunto de bolas y la caja donde se realiza la simulación.

```
#include "Caja.h"
#include "Bola.h"
#include <vector>
#include <fstream>
#include <string>
Include dependency graph for Sistema.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [Sistema](#)
Representa el sistema completo de simulación de un billar de N bolas.

Enumerations

- enum class [Integrador](#) { [Euler](#) , [Verlet](#) }
Enumeración para seleccionar el método de integración temporal.

4.5.1 Detailed Description

Define la clase [Sistema](#), que representa el conjunto de bolas y la caja donde se realiza la simulación.

La clase [Sistema](#) coordina la evolución temporal de todas las bolas, maneja la detección y resolución de colisiones, y gestiona la salida de datos.

4.5.2 Enumeration Type Documentation

4.5.2.1 Integrador

```
enum class Integrador [strong]
```

Enumeración para seleccionar el método de integración temporal.

Permite elegir entre los esquemas de Euler y Verlet.

Enumerator

Euler	Integración mediante el método de Euler explícito.
Verlet	Integración mediante el método de Verlet.

4.6 Sistema.h

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

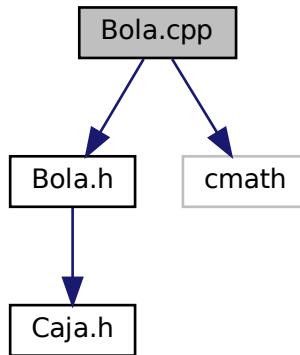
```
1
9 #ifndef SISTEMA_H
10 #define SISTEMA_H
11
12 #include "Caja.h"
13 #include "Bola.h"
14 #include <vector>
15 #include <fstream>
16 #include <string>
17
24 enum class Integrador {
25     Euler,
26     Verlet
27 };
28
36 class Sistema {
37 private:
38     Caja caja;
39     std::vector<Bola> bolas;
40     Integrador integrador_actual = Integrador::Verlet;
41
46     void PasoEuler(double dt);
47
52     void PasoVerlet(double dt);
53
54 public:
60     void DefinaCaja(double W, double H);
61
66     void Reserve(int N);
67
79     void InicialiceRejilla(double m, double r, double v_max, bool alterna = false);
80
86     void SeleccionaIntegrador(const std::string& nombre);
```

```
87     void Paso(double dt);
88
89     void Encabezado(std::ofstream& f);
90
91     void Guarda(std::ofstream& f, double t);
92 };
93
94 #endif
95
96
97
98
99
100
101
102
103
104
105
106
107
108 #endif
109
```

4.7 Bola.cpp File Reference

Implementación de los métodos de la clase [Bola](#).

```
#include "Bola.h"
#include <cmath>
Include dependency graph for Bola.cpp:
```



4.7.1 Detailed Description

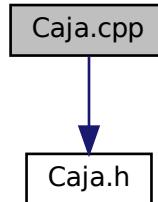
Implementación de los métodos de la clase [Bola](#).

Contiene la lógica para el movimiento de las bolas, las colisiones con las paredes y los choques elásticos entre bolas.

4.8 Caja.cpp File Reference

Implementación de la clase [Caja](#).

```
#include "Caja.h"
Include dependency graph for Caja.cpp:
```



4.8.1 Detailed Description

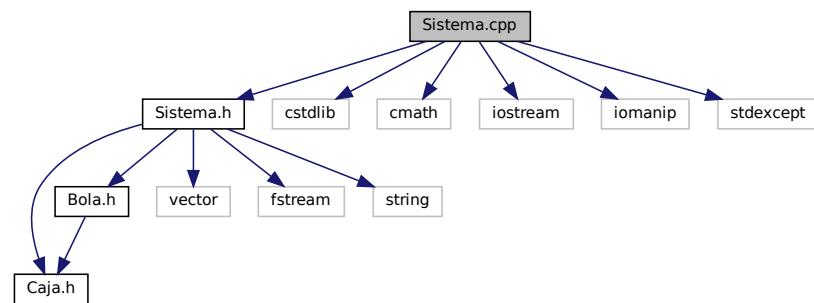
Implementación de la clase [Caja](#).

Contiene la definición de los métodos que inicializan y configuran las dimensiones del recinto donde se mueven las bolas.

4.9 Sistema.cpp File Reference

Implementación de la clase [Sistema](#), que coordina la simulación del billar.

```
#include "Sistema.h"
#include <cstdlib>
#include <cmath>
#include <iostream>
#include <iomanip>
#include <stdexcept>
Include dependency graph for Sistema.cpp:
```



4.9.1 Detailed Description

Implementación de la clase [Sistema](#), que coordina la simulación del billar.

Este archivo contiene la lógica para el avance temporal, las colisiones entre bolas, y la escritura de resultados a archivos de salida.

Index

Bola, 5
 Bola, 6
 ChoqueElastico, 6
 Getm, 7
 Getr, 7
 Getvx, 7
 Getvy, 7
 Getx, 7
 Gety, 7
 Inicie, 7
 Muevase, 8
 ResuelvaColisionParedesRobusto, 8
 ResuelvaColisionParedesSimple, 9
Bola.cpp, 22
Bola.h, 17, 18

Caja, 10
 Caja, 10
 Defina, 11
 GetH, 11
 GetW, 12
Caja.cpp, 22
Caja.h, 18, 19
ChoqueElastico
 Bola, 6

Defina
 Caja, 11
DefinaCaja
 Sistema, 13

Encabezado
 Sistema, 14
Euler
 Sistema.h, 21

GetH
 Caja, 11
Getm
 Bola, 7
Getr
 Bola, 7
Getvx
 Bola, 7
Getvy
 Bola, 7
GetW
 Caja, 12
Getx
 Bola, 7

 Gety
 Bola, 7
 Guarda
 Sistema, 14
 InicialiceRejilla
 Sistema, 14
 Inicie
 Bola, 7
 Integrador
 Sistema.h, 21
 Muevase
 Bola, 8
 Paso
 Sistema, 15
 Reserve
 Sistema, 16
 ResuelvaColisionParedesRobusto
 Bola, 8
 ResuelvaColisionParedesSimple
 Bola, 9
 SeleccionelIntegrador
 Sistema, 16
 Sistema, 13
 DefinaCaja, 13
 Encabezado, 14
 Guarda, 14
 InicialiceRejilla, 14
 Paso, 15
 Reserve, 16
 SeleccionelIntegrador, 16
 Sistema.cpp, 23
 Sistema.h, 20, 21
 Euler, 21
 Integrador, 21
 Verlet, 21
 Verlet
 Sistema.h, 21