My second Doxygen test

Generated by Doxygen 1.9.1

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

- Camera . . . . **??**
  - RGB . . . . **??**
  - Thermic . . . . **??**
- Dashboard . . . . **??**
- std::exception
  - InvalidUserException . . . . **??**
- Sensor . . . . **??**
  - AirQ . . . . **??**
  - Humidity . . . . **??**
  - IlluminationLv . . . . **??**
  - Temperature . . . . **??**
- User . . . . **??**

Chapter 2

Class Index

2.1 Class List

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

AirQ
Declaracion de Clase AirQ hija de Sensor . . . . ??
Camera
Declaracion de la clase padre Camera . . . . ??
Dashboard
Declaracion de clase Dashboard . . . . ??
Humidity
Declaracion de la clase Humidity hija de Sensor . . . . ??
IlluminationLv
Declaracion clase IlluminationLv hija de Sensor . . . . ??
InvalidUserException
Declaracion de clase InvalidUserException . . . . ??
RGB
Declaracion de la clase RGB hija de Camera . . . . ??
Sensor
Declaracion de clase padre Sensor . . . . ??
Temperature
Declaracion de clase Temperature hija de Sensor . . . . ??
Thermic
Declaracion de clase Thermic hija de Camera . . . . ??
User
Declaracion de clase User . . . . ??

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

AirQ.cpp . . . ??
AirQ.h
Herencia Sensor y AirQ . . . ??
Camera.cpp . . . ??
Camera.h
Genera el padre Camera . . . ??
Dashboard.cpp . . . ??
Dashboard.h
Libreria para la seleccion y demostracion de la interface . . . ??
Humidity.cpp . . . ??
Humidity.h
Herencia Sensor y Humidity . . . ??
IlluminationLv.cpp . . . ??
IlluminationLv.h
Herencia Sensor y IlluminationLv . . . ??
InvalidUserExcepcion.h
Detecta una excepcion tipo Usuario Invalido . . . ??
main.cpp . . . ??
RGB.cpp . . . ??
RGB.h
Herencia Camera y RGB . . . ??
Sensor.cpp . . . ??
Sensor.h
Genera el padre Sensor . . . ??
Temperature.cpp . . . ??
Temperature.h
Herencia Sensor y Temperature . . . ??
Thermic.cpp . . . ??
Thermic.h . . . ??
User.cpp . . . ??
User.h . . . ??

Chapter 4

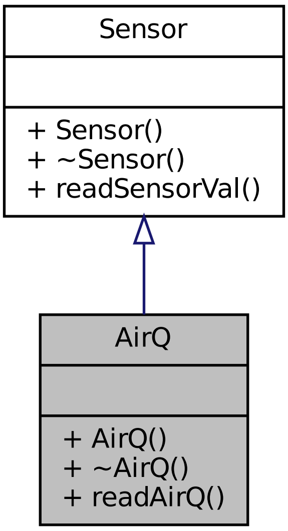
Class Documentation

4.1 AirQ Class Reference

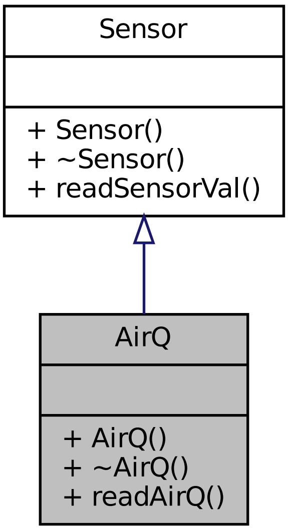
Declaracion de Clase AirQ hija de Sensor.

```
#include <AirQ.h>
```

Inheritance diagram for AirQ:



Collaboration diagram for AirQ:



Public Member Functions

- AirQ ()

  Constructor de clase AirQ.
- ∼AirQ ()

  Destructor de clase AirQ.
- void readAirQ ()

  Lectura de AirQ.

4.1.1 Detailed Description

Declaracion de Clase AirQ hija de Sensor.

Definition at line 23 of file AirQ.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 AirQ()

```
AirQ::AirQ ( )
```

Constructor de clase AirQ.

Definition at line 5 of file AirQ.cpp.

```
5 {}                //Constructor
```

4.1.2.2 ∼AirQ()

```
AirQ::∼AirQ ( )
```

Destructor de clase AirQ.

Definition at line 13 of file AirQ.cpp.

```
13 {}               //Destructor
```

4.1.3 Member Function Documentation

4.1.3.1 readAirQ()

```
void AirQ::readAirQ ( )
```

Lectura de AirQ.

Returns

void

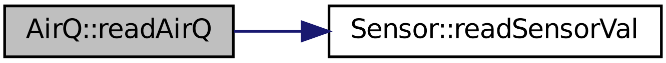
Definition at line 8 of file AirQ.cpp.

```
8                     {
9     readSensorVal();
10     std::cout « "The AirQ is of 78%" « std::endl;
11 }
```

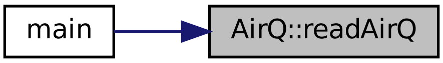
References Sensor::readSensorVal().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



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

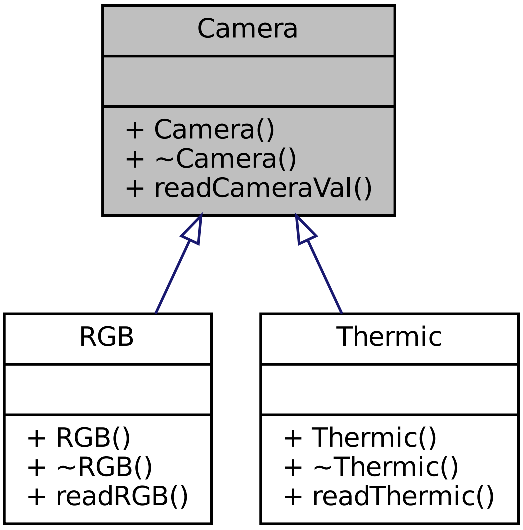
- AirQ.h
- AirQ.cpp

4.2 Camera Class Reference

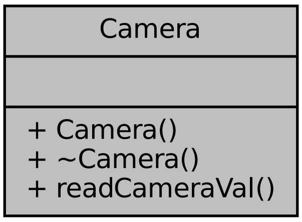
Declaracion de la clase padre Camera.

`#include <Camera.h>`

Inheritance diagram for Camera:



Collaboration diagram for Camera:



Public Member Functions

- Camera ()

  Constructor de clase Camera.
- ∼Camera ()

  Destructor de clase Camera.
- void readCameraVal ()

  Lectura de datos Camera.

4.2.1 Detailed Description

Declaracion de la clase padre Camera.

Definition at line 15 of file Camera.h.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Camera()

```
Camera::Camera ( )
```

Constructor de clase Camera.

Definition at line 4 of file Camera.cpp.

```
4 {}                //Constructor
```

4.2.2.2 ∼Camera()

```
Camera::∼Camera ( )
```

Destructor de clase Camera.

Definition at line 5 of file Camera.cpp.

```
5 {}               //Destructor
```

4.2.3 Member Function Documentation

4.2.3.1 readCameraVal()

```
void Camera::readCameraVal ( )
```

Lectura de datos Camera.

Returns

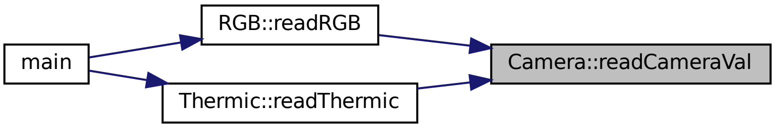
void

Definition at line 8 of file Camera.cpp.

```
                        {           //Falta agregar esto
    std::cout « "Seeing data of Camera..." « std::endl;
 }
```

Referenced by RGB::readRGB(), and Thermic::readThermic().

Here is the caller graph for this function:



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

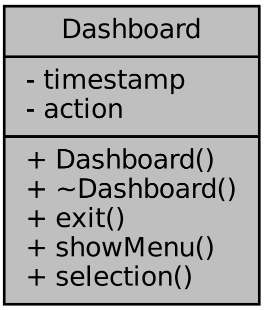
- Camera.h
- Camera.cpp

4.3 Dashboard Class Reference

Declaracion de clase Dashboard.

```
#include <Dashboard.h>
```

Collaboration diagram for Dashboard:



Public Member Functions

- Dashboard ()

  Constructor de clase Dashboard.
- ∼Dashboard ()

  Destructor de clase Dashboard.
- void exit ()

  La salida de la interface.
- int showMenu ()

  Muestra las oopciones de la intefaz.
- void selection (int)

  Marca la opcion de lo que se quiere observar de la interfaz.

Private Attributes

- float timestamp
- int action

4.3.1 Detailed Description

Declaracion de clase Dashboard.

Definition at line 17 of file Dashboard.h.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 Dashboard()

```
Dashboard::Dashboard ( )
```

Constructor de clase Dashboard.

Definition at line 13 of file Dashboard.cpp.

```
13 {}                //Constructor
```

4.3.2.2 ∼Dashboard()

```
Dashboard::∼Dashboard ( )
```

Destructor de clase Dashboard.

Definition at line 15 of file Dashboard.cpp.

```
15 {}
```

4.3.3 Member Function Documentation

4.3.3.1 exit()

```
void Dashboard::exit ( )
```

La salida de la interface.

Returns

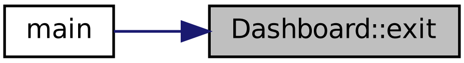
Cierre de la interfaz

Definition at line 18 of file Dashboard.cpp.

```
18                {
19     std::cout « "End of program "« std::endl;
20 }
```

Referenced by main().

Here is the caller graph for this function:



4.3.3.2 selection()

```
void Dashboard::selection (
             int   )
```

Marca la opcion de lo que se quiere observar de la interfaz.

Parameters

int	Numero entre 0 y 7

Returns

Lectura de la opcion seleccionada

4.3.3.3 showMenu()

```
int Dashboard::showMenu ( )
```

Muestra las oopciones de la intefaz.

Returns

Una interfaz de 8 opciones para escoger de sensores, camaras o actualizar base de datos

Definition at line 22 of file Dashboard.cpp.

```
22                          {
23     int action;
24 
25     std::cout « "----------------------------------------" « std::endl;
26     std::cout « "Select the corresponding option to see: " « std::endl « std::endl;
27     std::cout « "0) New Authorization" « std::endl;
28     std::cout « "1) Temperature" « std::endl;
29     std::cout « "2) Humidity" « std::endl;
30     std::cout « "3) Air Quality" « std::endl;
31     std::cout « "4) Illumination" « std::endl;
32     std::cout « "5) RGB Camera" « std::endl;
33     std::cout « "6) Thermica Camera" « std::endl;
34     std::cout « "7) Exit" « std::endl;
35     std::cout « "----------------------------------------" « std::endl « std:: endl;
36 
37     std::cin » action;
38     return action;
39 }
```

References action.

Referenced by main().

Here is the caller graph for this function:

main
Dashboard::showMenu

4.3.4 Member Data Documentation

4.3.4.1 action

```
int Dashboard::action [private]
```

Definition at line 47 of file Dashboard.h.

Referenced by showMenu().

4.3.4.2 timestamp

```
float Dashboard::timestamp [private]
```

Definition at line 46 of file Dashboard.h.

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

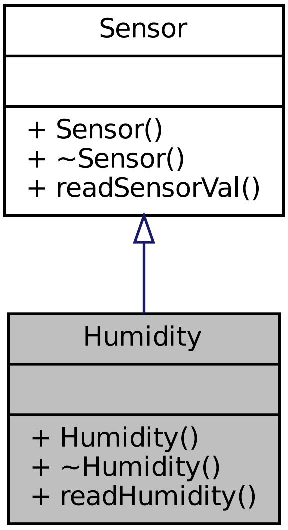
- Dashboard.h
- Dashboard.cpp

4.4 Humidity Class Reference

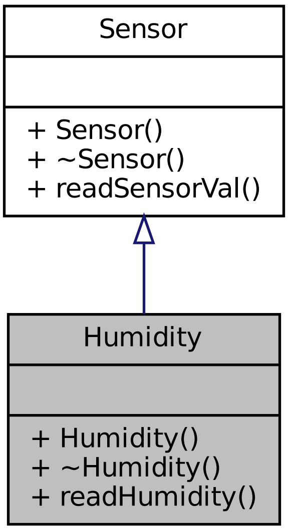
Declaracion de la clase Humidity hija de Sensor.

```
#include <Humidity.h>
```

Inheritance diagram for Humidity:



Collaboration diagram for Humidity:



Public Member Functions

- Humidity ()

    Constructor de clase Humidity.
- ∼Humidity ()

    Destructor de clase Humidity.
- void readHumidity ()

    Lectura de Humedad.

4.4.1 Detailed Description

Declaracion de la clase Humidity hija de Sensor.

Definition at line 23 of file Humidity.h.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 Humidity()

```
Humidity::Humidity ( )
```

Constructor de clase Humidity.

Definition at line 5 of file Humidity.cpp.

```
5 {}                //Constructor
```

4.4.2.2 ∼Humidity()

```
Humidity::∼Humidity ( )
```

Destructor de clase Humidity.

Definition at line 13 of file Humidity.cpp.

```
13 {}
```

4.4.3 Member Function Documentation

4.4.3.1 readHumidity()

```
void Humidity::readHumidity ( )
```

Lectura de Humedad.

Returns

void

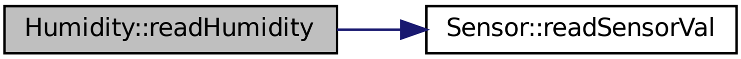
Definition at line 8 of file Humidity.cpp.

```
8                             {
9     readSensorVal();
10     std::cout « "The Humidity is of 88%" « std::endl;
11 }
```

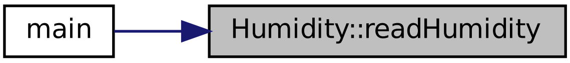
References Sensor::readSensorVal().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



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

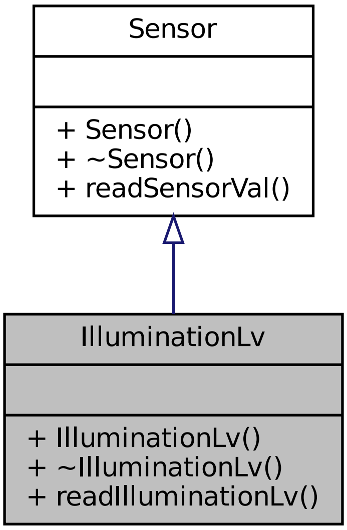
- Humidity.h
- Humidity.cpp

4.5 IlluminationLv Class Reference

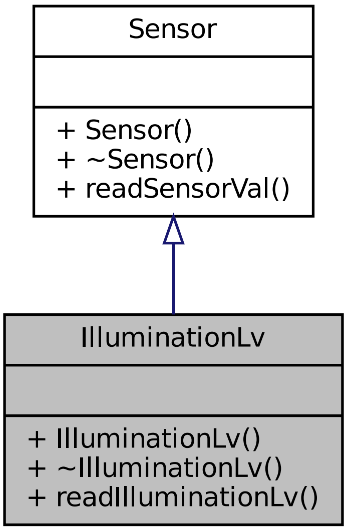
Declaracion clase IlluminationLv hija de Sensor.

```
#include <IlluminationLv.h>
```

Inheritance diagram for IlluminationLv:



Collaboration diagram for IlluminationLv:



Public Member Functions

- IlluminationLv ()

  Constructor de clase IlluminationLv.
- ∼IlluminationLv ()

  Destructor de clase IlluminationLv.
- void readIlluminationLv ()

  Lectura del nivel de Illumination.

4.5.1 Detailed Description

Declaracion clase IlluminationLv hija de Sensor.

Definition at line 23 of file IlluminationLv.h.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 IlluminationLv()

```
IlluminationLv::IlluminationLv ( )
```

Constructor de clase IlluminationLv.

Definition at line 5 of file IlluminationLv.cpp.

```
5 {}                //Constructor
```

4.5.2.2 ∼IlluminationLv()

```
IlluminationLv::∼IlluminationLv ( )
```

Destructor de clase IlluminationLv.

Definition at line 13 of file IlluminationLv.cpp.

```
13 {}               //Destructor
```

4.5.3 Member Function Documentation

4.5.3.1 readIlluminationLv()

```
void IlluminationLv::readIlluminationLv ( )
```

Lectura del nivel de Illumination.

Returns

void

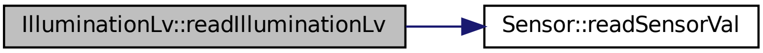
Definition at line 8 of file IlluminationLv.cpp.

```
8                                       {
9     readSensorVal();
10     std::cout « "The IlluminationLv is of 100%" « std::endl;
11 }
```

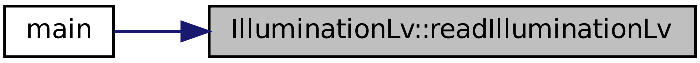
References Sensor::readSensorVal().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



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

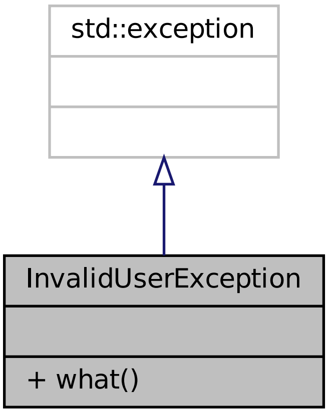
- IlluminationLv.h
- IlluminationLv.cpp

4.6 InvalidUserException Class Reference

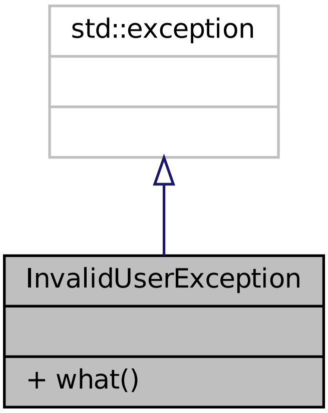
Declaracion de clase InvalidUserException.

```
#include <InvalidUserExcepcion.h>
```

Inheritance diagram for InvalidUserException:



Collaboration diagram for InvalidUserException:



Public Member Functions

- virtual const char * what () const throw ()

  Declaracion de la excepcion para Usuarios incorrectos.

4.6.1 Detailed Description

Declaracion de clase InvalidUserException.

Definition at line 44 of file InvalidUserExcepcion.h.

4.6.2 Member Function Documentation

4.6.2.1 what()

```
virtual const char* InvalidUserException::what ( ) const throw ( )   [inline], [virtual]
```

Declaracion de la excepcion para Usuarios incorrectos.

Definition at line 49 of file InvalidUserExcepcion.h.

```
49                                              {
50              return "Invalid User Information";
51          }
```

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

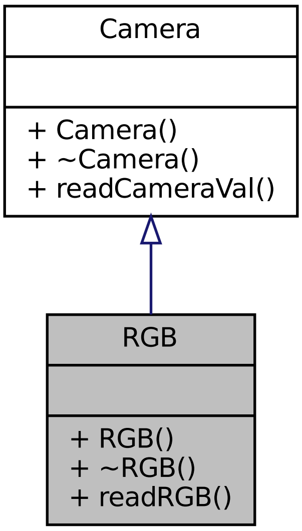
- InvalidUserExcepcion.h

4.7 RGB Class Reference

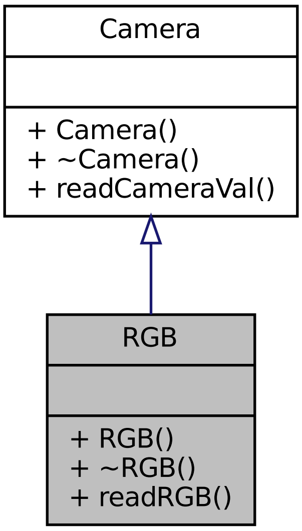
Declaracion de la clase RGB hija de Camera.

```
#include <RGB.h>
```

Inheritance diagram for RGB:



Collaboration diagram for RGB:



Public Member Functions

- RGB ()

  Constructor de clase RGB.
- ∼RGB ()

  Destructor de clase RGB.
- void readRGB ()

  Lectura de RGB.

4.7.1 Detailed Description

Declaracion de la clase RGB hija de Camera.

Definition at line 23 of file RGB.h.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 RGB()

```
RGB::RGB ( )
```

Constructor de clase RGB.

Definition at line 5 of file RGB.cpp.

```
5 {}                //Constructor
```

4.7.2.2 ∼RGB()

```
RGB::∼RGB ( )
```

Destructor de clase RGB.

Definition at line 13 of file RGB.cpp.

```
{}                //Destructor
```

4.7.3 Member Function Documentation

4.7.3.1 readRGB()

```
void RGB::readRGB ( )
```

Lectura de RGB.

Returns

void

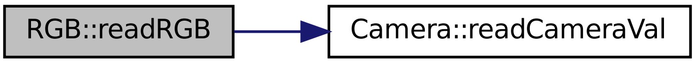
Definition at line 8 of file RGB.cpp.

```
               {
     readCameraVal();
      std::cout « "The RGB camera has a view of 100%" « std::endl;
 }
```

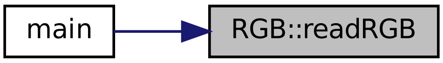
References Camera::readCameraVal().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



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

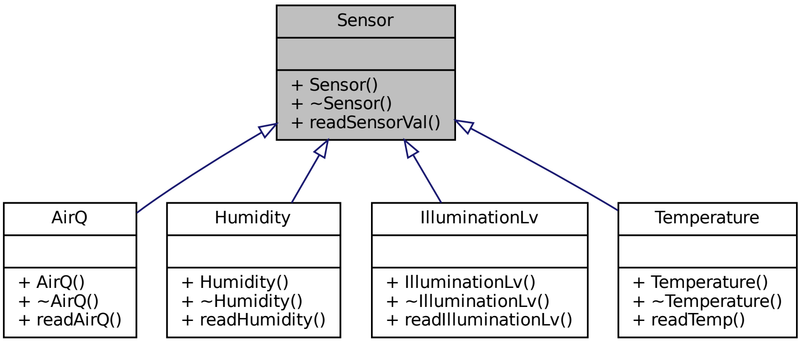
- RGB.h
- RGB.cpp

4.8 Sensor Class Reference

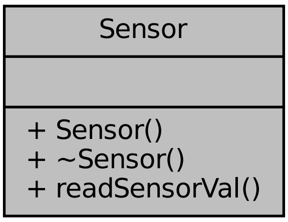
Declaracion de clase padre Sensor.

```
#include <Sensor.h>
```

Inheritance diagram for Sensor:



Collaboration diagram for Sensor:



Public Member Functions

- Sensor ()

  Constructor de clase Sensor.
- ∼Sensor ()

  Destructor de clase Sensor.
- void readSensorVal ()

  Lectura de datos de Sensores.

4.8.1 Detailed Description

Declaracion de clase padre Sensor.

Definition at line 15 of file Sensor.h.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 Sensor()

```
Sensor::Sensor ( )
```

Constructor de clase Sensor.

Definition at line 4 of file Sensor.cpp.

```
4 {}                //Constructor
```

4.8.2.2 ∼Sensor()

```
Sensor::∼Sensor ( )
```

Destructor de clase Sensor.

Definition at line 11 of file Sensor.cpp.

```
11 {}               //Destructor
```

4.8.3 Member Function Documentation

4.8.3.1 readSensorVal()

```
void Sensor::readSensorVal ( )
```

Lectura de datos de Sensores.

Returns

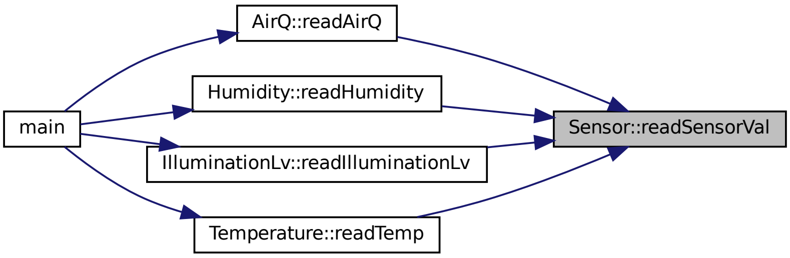
void

Definition at line 7 of file Sensor.cpp.

```
7                           {           //Falta agregar esto
8   std::cout « "Seeing data of Sensor..." « std::endl;
9 }
```

Referenced by AirQ::readAirQ(), Humidity::readHumidity(), IlluminationLv::readIlluminationLv(), and Temperature↩ ::readTemp().

Here is the caller graph for this function:



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

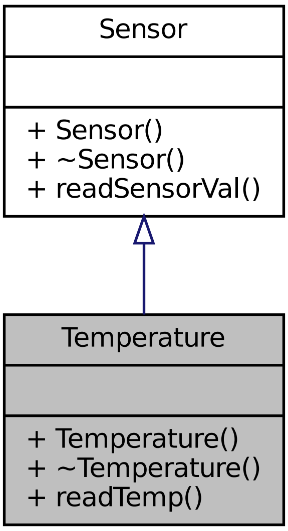
- Sensor.h
- Sensor.cpp

4.9 Temperature Class Reference

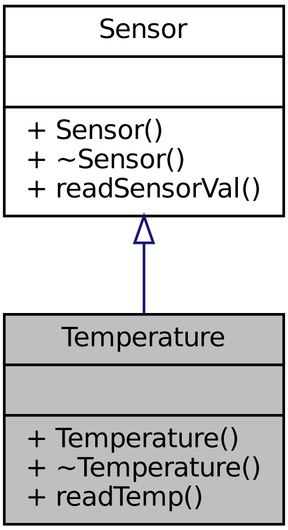
Declaracion de clase Temperature hija de Sensor.

```
#include <Temperature.h>
```

Inheritance diagram for Temperature:



Collaboration diagram for Temperature:



Public Member Functions

- Temperature ()

  Constructor de clase Temperature.
- ∼Temperature ()

  Destructor de clase Temperature.
- void readTemp ()

  Lectura de Temperature.

4.9.1 Detailed Description

Declaracion de clase Temperature hija de Sensor.

Definition at line 23 of file Temperature.h.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 Temperature()

```
Temperature::Temperature ( )
```

Constructor de clase Temperature.

Definition at line 5 of file Temperature.cpp.

```
5 {}                //Constructor
```

4.9.2.2 ∼Temperature()

```
Temperature::∼Temperature ( )
```

Destructor de clase Temperature.

Definition at line 6 of file Temperature.cpp.

```
6 {}                //Destructor
```

4.9.3 Member Function Documentation

4.9.3.1 readTemp()

```
void Temperature::readTemp ( )
```

Lectura de Temperature.

Returns

void

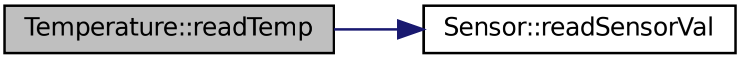
Definition at line 9 of file Temperature.cpp.

```
9                            {
10     readSensorVal();
11     std::cout « "The temperature is of 28 °C" « std::endl;
12 }
```

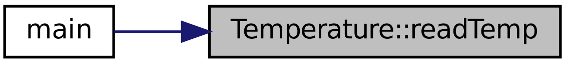
References Sensor::readSensorVal().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



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

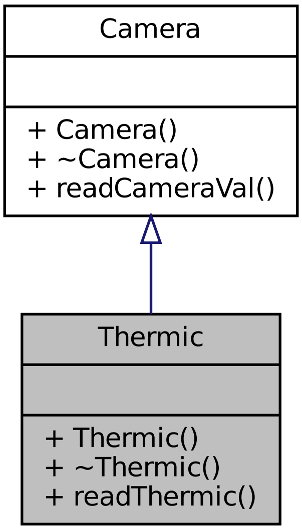
- Temperature.h
- Temperature.cpp

4.10 Thermic Class Reference

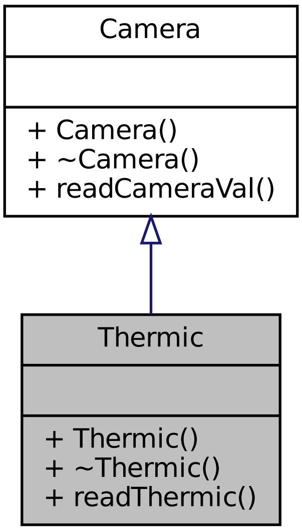
Declaracion de clase Thermic hija de Camera.

```
#include <Thermic.h>
```

Inheritance diagram for Thermic:



Collaboration diagram for Thermic:



Public Member Functions

- Thermic ()

  Constructor de clase Thermic.
- ∼Thermic ()

  Destructor de clase Thermic.
- void readThermic ()

  Lectura de camaras Termicas.

4.10.1 Detailed Description

Declaracion de clase Thermic hija de Camera.

Definition at line 23 of file Thermic.h.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 Thermic()

```
Thermic::Thermic ( )
```

Constructor de clase Thermic.

Definition at line 5 of file Thermic.cpp.

```
5 {}                //Constructor
```

4.10.2.2 ∼Thermic()

```
Thermic::∼Thermic ( )
```

Destructor de clase Thermic.

Definition at line 13 of file Thermic.cpp.

```
13 {}
```

4.10.3 Member Function Documentation

4.10.3.1 readThermic()

```
void Thermic::readThermic ( )
```

Lectura de camaras Termicas.

Returns

void

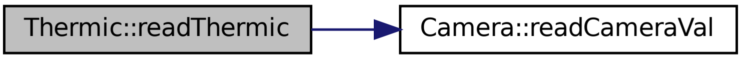
Definition at line 8 of file Thermic.cpp.

```
8                             {
9      readCameraVal();
10      std::cout « "The Thermic camera detects no movement" « std::endl;
11 }
```

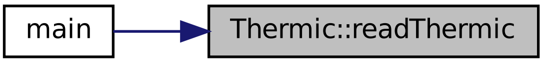
References Camera::readCameraVal().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



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

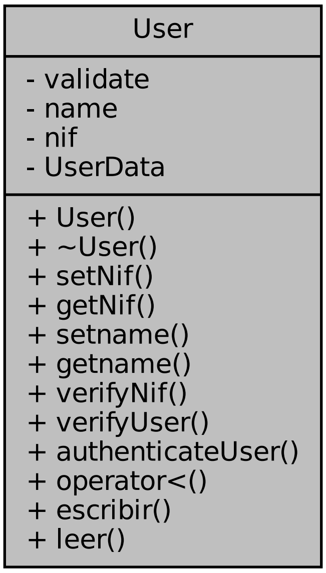
- Thermic.h
- Thermic.cpp

4.11 User Class Reference

Declaracion de clase User.

```
#include <User.h>
```

Collaboration diagram for User:



Public Member Functions

- User (string &, string &)

  Constructor de clase User.
- ∼User ()

  Destructor de clase User.
- void setNif (string &)

  Establece el NIF del usuario.
- string getNif ()

  Regresa el Nif del usuario.
- void setname (string &)

  Estable el nombre del usuario.
- string getname ()

  Regresa el nombre del usuario.
- void verifyNif (bool)

  Verifica si el NIF introducido es permitido.
- void verifyUser (bool)

  Verifica si el numero de usuario introducido es permitido.
- bool authenticateUser ()

Verifica si el NIF y el name son validos.
- bool operator< (User &)

  Sobrecarga del User para comparar usuarios.
- void escribir (string, string)

  Escribe el fichero de Usuarios.
- void leer ()

  Lee el fichero de Usuarios.

Private Attributes

- bool validate
- string name
- string nif
- set< pair< string, string > > UserData

Friends

- ostream & operator<< (ostream &, User &)

4.11.1 Detailed Description

Declaracion de clase User.

Definition at line 45 of file User.h.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 User()

```
User::User (
              string & name,
              string & nif )
```

Constructor de clase User.

Parameters

Strings	del Nombre y Nif del usuario

Definition at line 14 of file User.cpp.

```
14                                      {
15
16      setNif(nif);
17      setname(name);
18 }
```

4.11.2.2 ∼User()

```
User::∼User ( )
```

Destructor de clase User.

Definition at line 21 of file User.cpp.

```
21 {}
```

4.11.3 Member Function Documentation

4.11.3.1 authenticateUser()

```
bool User::authenticateUser ( )
```

Verifica si el NIF y el name son validos.

Returns

permiso para proseguir a la interface o no

Definition at line 58 of file User.cpp.

```
58                           {
59     bool validate = false;
60 
61 //-----MODIFICACION: EXCEPCIONES-----
62     try{
63         if(UserData.find({name,nif}) == UserData.end()){
64             throw InvalidUserException();
65         }
66         std::cout « "\nVALID USER" « endl;
67         std::cout « "Loading..." « endl « endl;
68         sleep(2);
69         system("clear");
70 
71         validate = true;
72         auto date = chrono::system_clock::now();
73         time_t actual_time = chrono::system_clock::to_time_t(date);
74         std::cout « "Accessed time: " « ctime(&actual_time);
75         return validate;
76     }
77     catch(InvalidUserException& e){
78         std::cout « endl « "The USER:" « name « " IS NOT VALID" « endl;
79         validate = false;
80         std::cout « "The NIF:" « nif « " DOESN'T EXIST" « endl « endl;
81         validate = false;
82         sleep(2);
83         throw system("./main");
84 
85     }
86     return validate;
87 }
```

Referenced by main().

Here is the caller graph for this function:

main
User::authenticateUser

4.11.3.2 escribir()

```
void User::escribir (
              string name,
              string nif )
```

Escribe el fichero de Usuarios.

Parameters

String	de name y nif del usuario

Returns

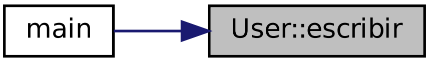
Fichero users.dat

Definition at line 118 of file User.cpp.

```
118                                         {
119     ofstream fichero;
120     fichero.open("users.dat", ios::app); //Abrir el archivo en modo escritura
121 
122     if (fichero.fail()){
123         cout«"Wasn't possible to open file...";
124         exit(1);
125     }
126     if (UserData.find({name,nif}) == UserData.end()){
127         cout « "New Registration Done Correctly!!"« endl;
128         fichero« "Name " « name « " Nif " « nif « endl;
129         fichero.close();
130     }
131     else{
132         cout « "Name or NIF already exist" « endl;
133     }
134 }
```

Referenced by main().

Here is the caller graph for this function:



4.11.3.3 getname()

```
string User::getname ( )
```

Regresa el nombre del usuario.

Returns

string del nombre

Definition at line 39 of file User.cpp.

```
39                      {
40      return name;
41 }
```

4.11.3.4 getNif()

```
string User::getNif ( )
```

Regresa el Nif del usuario.

Returns

String del NIF

Definition at line 29 of file User.cpp.

```
                     {
    return nif;
}
```

4.11.3.5 leer()

```
void User::leer ( )
```

Lee el fichero de Usuarios.

Returns

Lectura del fichero users.dat

Definition at line 90 of file User.cpp.

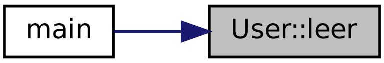
```
                 {
    ifstream fichero;
    string texto;
    fichero.open("users.dat", ios::in); //Abrir el archivo en modo escritura

    if (fichero.fail()){
        cout«"Wasn't possible to open file...";
        exit(1);
    }
    while(!fichero.eof()){
         getline(fichero,texto);
         char delimeter = ' ';

         std::vector<std::string> tokens;
         std::string token;
         std::istringstream tokenStream(texto);
         while (getline(tokenStream, token, delimeter))
         {
             tokens.push_back(token);
         }
         if(tokens.size() > 3){
             UserData.insert({tokens.at(1),tokens.at(3)});
         }
     }
     fichero.close();
}
```

Referenced by main().

Here is the caller graph for this function:



4.11.3.6 operator<()

```
bool User::operator< (
             User & other )
```

Sobrecarga del User para comparar usuarios.

Parameters

Usuario	a comparar

Returns

bool

Definition at line 138 of file User.cpp.

```
                               {
    return(this -> name > other.name);
}
```

References name.

4.11.3.7 setname()

```
void User::setname (
             string & name )
```

Estable el nombre del usuario.

Parameters

String	& del nombre

Returns

void

Definition at line 34 of file User.cpp.

```
                               {
    this -> name = name;
}
```

4.11.3.8 setNif()

```
void User::setNif (
             string & nif )
```

Establece el NIF del usuario.

Parameters

String	& del NIF

Returns

void

Definition at line 24 of file User.cpp.

```
24                              {
25     this -> nif = nif;
26 }
```

4.11.3.9 verifyNif()

```
void User::verifyNif (
             bool validate )
```

Verifica si el NIF introducido es permitido.

Parameters

bool	

Returns

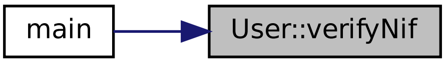
Da la bienvenida al usuario o indica que no es correcto

Definition at line 44 of file User.cpp.

```
44                                   {
45     if (validate == true){
46         std::cout « "Employee with NIF: " « nif « endl;
47     }
48 }
```

Referenced by main().

Here is the caller graph for this function:



4.11.3.10 verifyUser()

```
void User::verifyUser (
             bool validate )
```

Verifica si el numero de usuario introducido es permitido.

Parameters

bool	

Returns

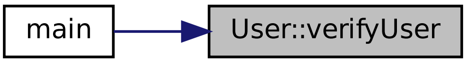
Da la bienvenida al usuario o indica que no es correcto

Definition at line 51 of file User.cpp.

```
51                                {
52     if(validate == true){
53         std::cout « "Welcome User: " « name « endl;
54     }
55 }
```

Referenced by main().

Here is the caller graph for this function:



4.11.4 Friends And Related Function Documentation

4.11.4.1 operator<<

```
ostream& operator<< (
             ostream & output,
             User & a )   [friend]
```

Definition at line 142 of file User.cpp.

```
142                                     {
143     cout « "[User's Name: " « a.getname() « "-- User's NIF: " « a.getNif() « "]" « endl;
144     return output;
145 }
```

4.11.5 Member Data Documentation

4.11.5.1 name

```
string User::name [private]
```

Definition at line 117 of file User.h.

Referenced by operator<().

4.11.5.2 nif

```
string User::nif [private]
```

Definition at line 118 of file User.h.

4.11.5.3 UserData

```
set<pair<string,string> > User::UserData [private]
```

Definition at line 119 of file User.h.

4.11.5.4 validate

```
bool User::validate [private]
```

Definition at line 116 of file User.h.

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

- User.h
- User.cpp

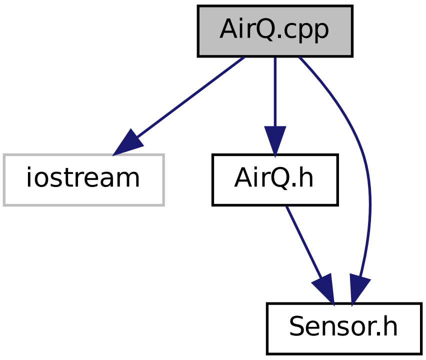
Chapter 5

File Documentation

5.1 AirQ.cpp File Reference

```
#include <iostream>
#include "AirQ.h"
#include "Sensor.h"
```

Include dependency graph for AirQ.cpp:

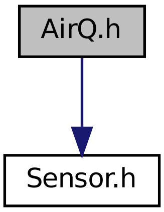


5.2 AirQ.h File Reference

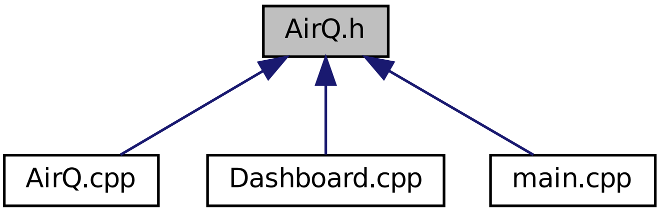
Herencia Sensor y AirQ.

```
#include "Sensor.h"
```

Include dependency graph for AirQ.h:



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



Classes

- class AirQ

    Declaracion de Clase AirQ hija de Sensor.

5.2.1 Detailed Description

Herencia Sensor y AirQ.

Version

1.0

Date

21/12/2022

Author

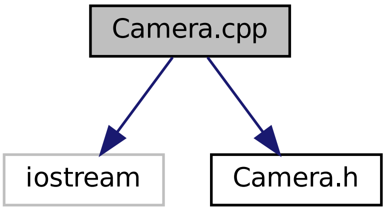
Rodrigo Rodas Barrera @title Detector de Cualidad de Aire

```
int main(){
 AirQ airQ = AirQ();
 airQ.readAirQ();
}
```

5.3 Camera.cpp File Reference

```
#include <iostream>
#include "Camera.h"
```

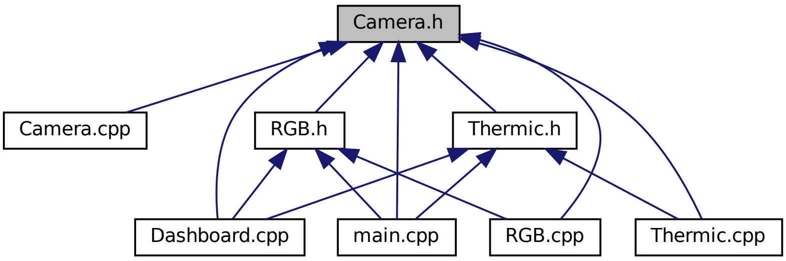
Include dependency graph for Camera.cpp:



5.4 Camera.h File Reference

Genera el padre Camera.

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



Classes

- class Camera

  Declaracion de la clase padre Camera.

5.4.1 Detailed Description

Genera el padre Camera.

Version

1.0

Date

21/12/2022

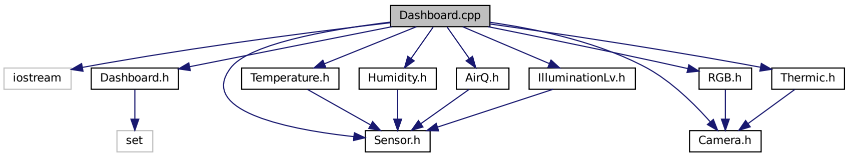
Author

Rodrigo Rodas Barrera @title Verificacion de Camaras

5.5 Dashboard.cpp File Reference

```
#include <iostream>
#include "Dashboard.h"
#include "Camera.h"
#include "Sensor.h"
#include "Temperature.h"
#include "Humidity.h"
#include "AirQ.h"
#include "IlluminationLv.h"
#include "RGB.h"
#include "Thermic.h"
```

Include dependency graph for Dashboard.cpp:

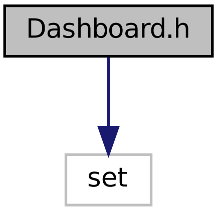


5.6 Dashboard.h File Reference

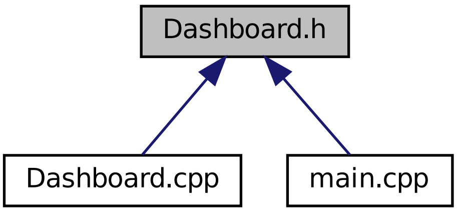
Libreria para la seleccion y demostracion de la interface.

```
#include <set>
```

Include dependency graph for Dashboard.h:



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



Classes

- class Dashboard

  Declaracion de clase Dashboard.

5.6.1 Detailed Description

Libreria para la seleccion y demostracion de la interface.

Version

1.0

Date

21/12/2022

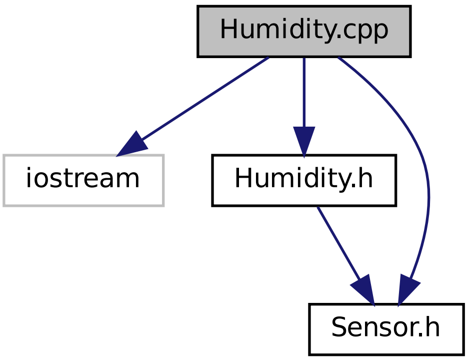
Author

Rodrigo Rodas Barrera @title Interfaz para la lectura de datos

5.7 Humidity.cpp File Reference

```
#include <iostream>
#include "Humidity.h"
#include "Sensor.h"
```

Include dependency graph for Humidity.cpp:

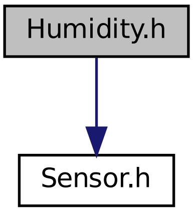


5.8 Humidity.h File Reference

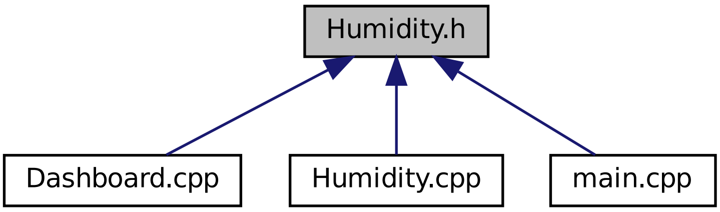
Herencia Sensor y Humidity.

```
#include "Sensor.h"
```

Include dependency graph for Humidity.h:



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



Classes

- class Humidity

  Declaracion de la clase Humidity hija de Sensor.

5.8.1 Detailed Description

Herencia Sensor y Humidity.

Version

1.0

Date

21/12/2022

Author

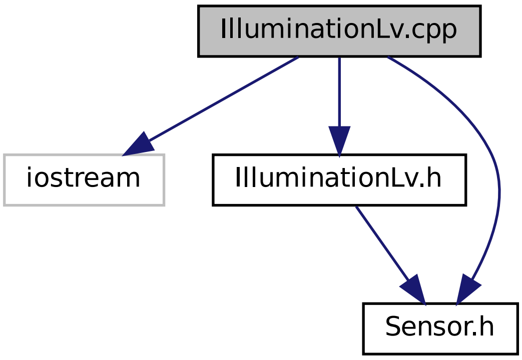
Rodrigo Rodas Barrera @title Detector de Humedad

```
int main(){
 Humidity humid = Humidity();
 humid.readHumidity();
}
```

5.9 IlluminationLv.cpp File Reference

```
#include <iostream>
#include "IlluminationLv.h"
```

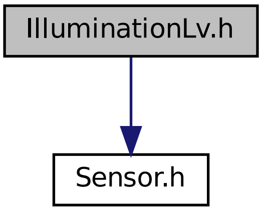
`#include "Sensor.h"`
Include dependency graph for IlluminationLv.cpp:



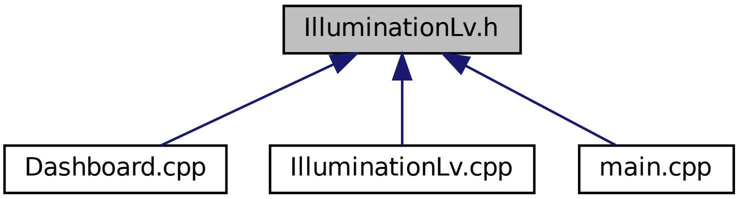
5.10 IlluminationLv.h File Reference

Herencia Sensor y IlluminationLv.

`#include "Sensor.h"`
Include dependency graph for IlluminationLv.h:



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



Classes

- class IlluminationLv

  Declaracion clase IlluminationLv hija de Sensor.

5.10.1 Detailed Description

Herencia Sensor y IlluminationLv.

Version

1.0

Date

21/12/2022

Author

Rodrigo Rodas Barrera @title Detector la Iluminacion

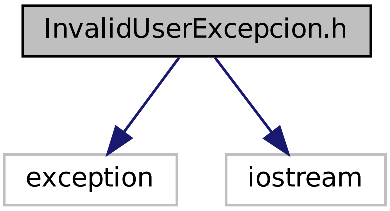
```
int main(){
 IlluminationLv illumination = IlluminationLv();
 illumination.readIlluminationLv();
}
```

5.11 InvalidUserExcepcion.h File Reference

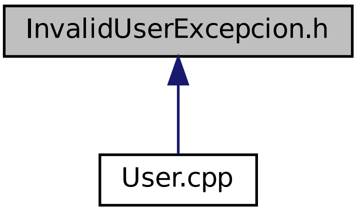
Detecta una excepcion tipo Usuario Invalido.

```
#include <exception>
#include <iostream>
```

Include dependency graph for InvalidUserExcepcion.h:



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



Classes

- class InvalidUserException

  Declaracion de clase InvalidUserException.

5.11.1 Detailed Description

Detecta una excepcion tipo Usuario Invalido.

Version

1.0

Date

21/12/2022

Author

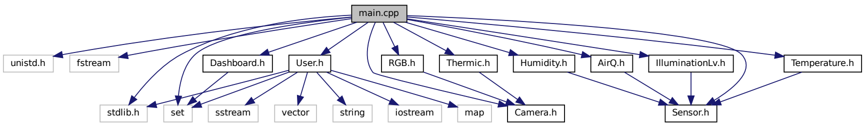
Rodrigo Rodas Barrera @title Detector de Excepciones

```
try{
    if(UserData.find({name,nif}) == UserData.end()){
        throw InvalidUserException();
    }
    std::cout « "\nVALID USER" « endl;
    std::cout « "Loading..." « endl « endl;
    sleep(2);
    system("clear");
    validate = true;
    auto date = chrono::system_clock::now();
    time_t actual_time = chrono::system_clock::to_time_t(date);
    std::cout « "Accessed time: " « ctime(&actual_time);
    return validate;
}
catch(InvalidUserException& e){
    std::cout « endl « "The USER:" « name « " IS NOT VALID" « endl;
    validate = false;
    std::cout « "The NIF:" « nif « " DOESN'T EXIST" « endl « endl;
    validate = false;
    sleep(2);
    throw system("./main");
}
return validate;
```

5.12 main.cpp File Reference

```
#include <unistd.h>
#include <fstream>
#include <stdlib.h>
#include <set>
#include "User.h"
#include "Dashboard.h"
#include "Camera.h"
#include "Sensor.h"
#include "Temperature.h"
#include "Humidity.h"
#include "AirQ.h"
#include "IlluminationLv.h"
#include "RGB.h"
#include "Thermic.h"
```

Include dependency graph for main.cpp:



Functions

- int main ()

5.12.1 Function Documentation

5.12.1.1 main()

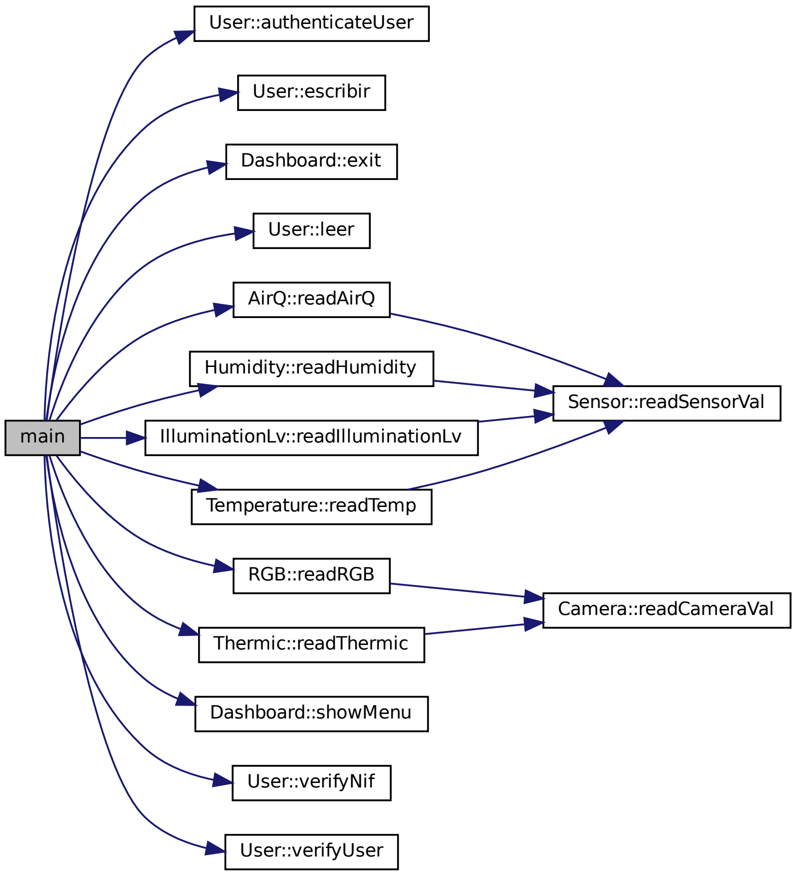
```
int main ( )
```

Definition at line 20 of file main.cpp.

```
20         {
21     system("clear");
22 
23     string _name = "";
24     std::cout « "Write your user name: ";
25     std::cin » _name;
26 
27     string _nif = "";
28     std::cout « "Write your NIF: ";
29     std::cin » _nif;
30 
31     bool valid;
32 
33     User p1 = User(_name, _nif);      //Crea un usuario y prueba si sus datos son correctos.
34     p1.leer();
35     valid = p1.authenticateUser();
36     p1.verifyUser(valid);
37     p1.verifyNif(valid);
38     std::cout « "\n";
39 
40     while(valid == true){
41         Dashboard menu = Dashboard();             //Crea el menu del dashboard
42         int option = menu.showMenu();
43 
44         if(option == 0){
45             string newName, newNif;
46             cout « "New Name: ";
47             cin » newName;
48             cout « "New Nif: ";
49             cin » newNif;
50             p1.escribir(newName, newNif);
51         }
52         else if(option == 1){
53             Temperature temp = Temperature();
54             temp.readTemp();
55         }
56         else if(option == 2){
57             Humidity humid = Humidity();
58             humid.readHumidity();
59         }
60         else if(option == 3){
61             AirQ airQ = AirQ();
62             airQ.readAirQ();
63         }
64         else if(option == 4){
65             IlluminationLv illumination = IlluminationLv();
66             illumination.readIlluminationLv();
67         }
68         else if(option == 5){
69             RGB rgb = RGB();
70             rgb.readRGB();
71         }
72         else if(option == 6){
73             Thermic therm = Thermic();
74             therm.readThermic();
75         }
76         else if(option == 7){
77             Dashboard exit = Dashboard();
78             exit.exit();
79             break;
80         }
81         sleep(5);
82         system("clear");
83 
84     }
85     if (valid == false){
86         std::cout « "Error at loging in..." « std::endl;
87         system("./main");
88     }
89 }
```

References User::authenticateUser(), User::escribir(), Dashboard::exit(), User::leer(), AirQ::readAirQ(), Humidity::readHumidity(), IlluminationLv::readIlluminationLv(), RGB::readRGB(), Temperature::readTemp(), Thermic::readThermic(), Dashboard::showMenu(), User::verifyNif(), and User::verifyUser().

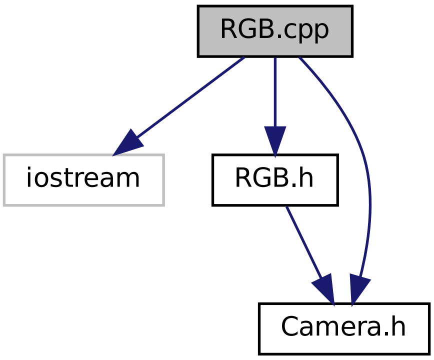
Here is the call graph for this function:



5.13 RGB.cpp File Reference

```
#include <iostream>
#include "RGB.h"
#include "Camera.h"
```

Include dependency graph for RGB.cpp:

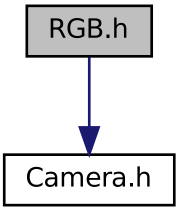


5.14 RGB.h File Reference

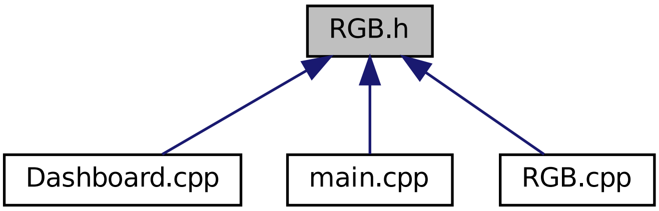
Herencia Camera y RGB.

```
#include "Camera.h"
```

Include dependency graph for RGB.h:



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



Classes

- class RGB

  Declaracion de la clase RGB hija de Camera.

5.14.1 Detailed Description

Herencia Camera y RGB.

Herencia Camera y Thermic.

Version

1.0

Date

21/12/2022

Author

Rodrigo Rodas Barrera @title Detector de RGB

```
int main(){
 RGB rgb = RGB();
 rgb.readRGB();
}
```

Version

1.0

Date

21/12/2022

Author

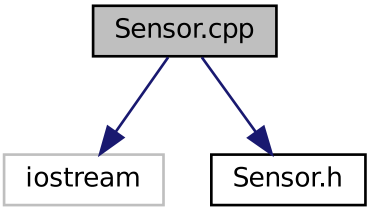
Rodrigo Rodas Barrera @title Detector de Thermicos

```
int main(){
 Thermic therm = Thermic();
 therm.readThermic();
}
```

5.15 Sensor.cpp File Reference

```
#include <iostream>
#include "Sensor.h"
```

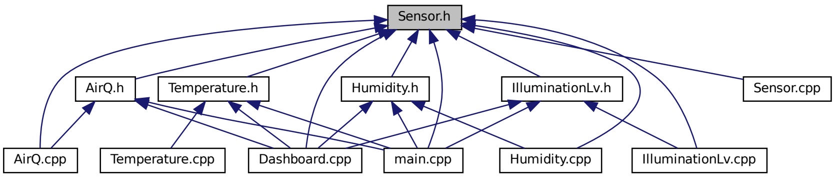
Include dependency graph for Sensor.cpp:



5.16 Sensor.h File Reference

Genera el padre Sensor.

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



Classes

- class Sensor

    Declaracion de clase padre Sensor.

5.16.1 Detailed Description

Genera el padre Sensor.

Version

1.0

Date

21/12/2022

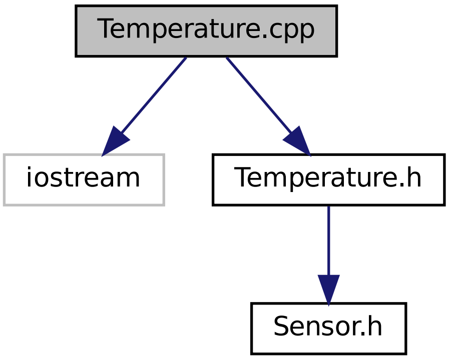
Author

Rodrigo Rodas Barrera @title Verificacion de Sensores

5.17 Temperature.cpp File Reference

```
#include <iostream>
#include "Temperature.h"
```

Include dependency graph for Temperature.cpp:

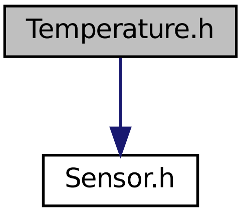


5.18 Temperature.h File Reference

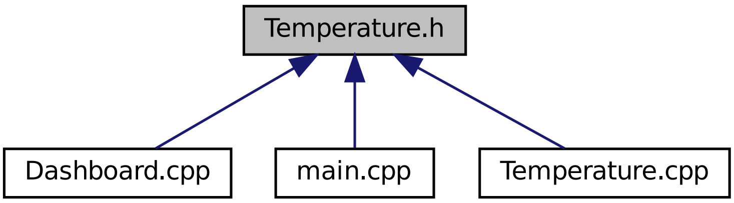
Herencia Sensor y Temperature.

```
#include "Sensor.h"
```

Include dependency graph for Temperature.h:



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



Classes

- class Temperature

  Declaracion de clase Temperature hija de Sensor.

5.18.1 Detailed Description

Herencia Sensor y Temperature.

Verifica las credenciales del usuario para dar permiso de usar la interface o no.

Version

1.0

Date

21/12/2022

Author

Rodrigo Rodas Barrera @title Detector de Temperatura

```
int main(){
 Temperature temp = Temperature();
 temp.readTemp();
}
```

Version

1.0

Date

21/12/2022

Author

Rodrigo Rodas Barrera @title Verificacion de Usuario

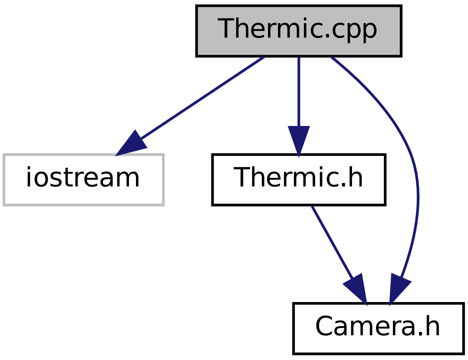
```
int main(){
     string _name = "";
     std::cout « "Write your user name: ";
     std::cin » _name;
     string _nif = "";
     std::cout « "Write your NIF: ";
     std::cin » _nif;
     bool valid;

     User p1 = User(_name, _nif);      //Crea un usuario y prueba si sus datos son correctos.
     p1.leer();
     valid = p1.authenticateUser();
     p1.verifyUser(valid);
     p1.verifyNif(valid);
     std::cout « "\n";
}
```

5.19 Thermic.cpp File Reference

```
#include <iostream>
#include "Thermic.h"
#include "Camera.h"
```

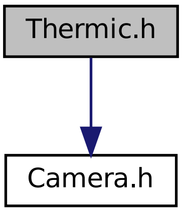
Include dependency graph for Thermic.cpp:



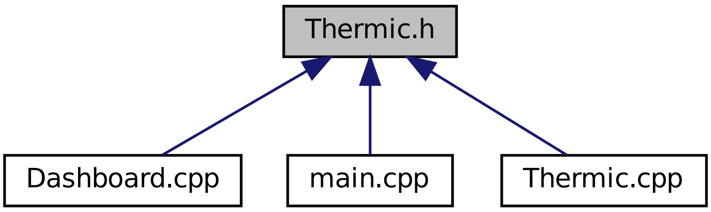
5.20 Thermic.h File Reference

```
#include "Camera.h"
```

Include dependency graph for Thermic.h:



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



Classes

- class Thermic

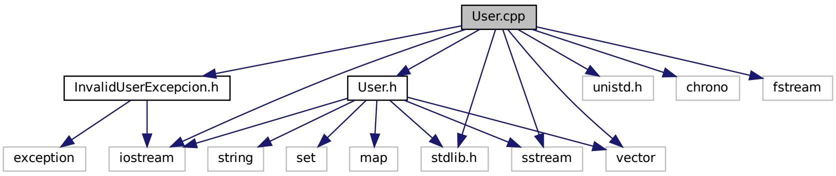
  Declaracion de clase Thermic hija de Camera.

5.21 User.cpp File Reference

```
#include <iostream>
#include <stdlib.h>
#include <unistd.h>
#include <chrono>
#include <fstream>
#include <sstream>
#include <vector>
#include "User.h"
```

`#include "InvalidUserExcepcion.h"`

Include dependency graph for User.cpp:



Functions

- ostream & operator<< (ostream &output, User &a)

5.21.1 Function Documentation

5.21.1.1 operator<<()

```
ostream& operator<< (
            ostream & output,
            User & a )
```

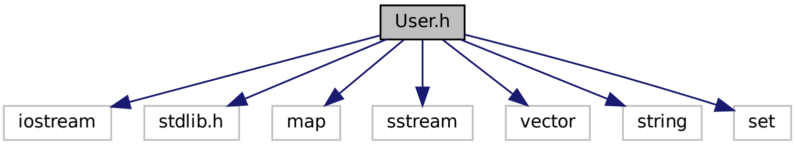
Definition at line 142 of file User.cpp.

```
                                        {
       cout « "[User's Name: " « a.getname() « "-- User's NIF: " « a.getNif() « "]" « endl;
       return output;
   }
```

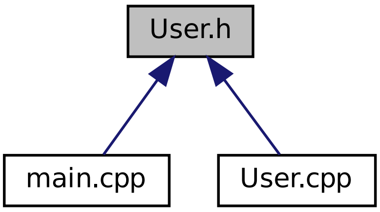
5.22 User.h File Reference

```
#include <iostream>
#include <stdlib.h>
#include <map>
#include <sstream>
#include <vector>
#include <string>
#include <set>
```

Include dependency graph for User.h:



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



Classes

- class User

  Declaracion de clase User.