

Laboratorio 2

Generated by Doxygen 1.8.12

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	Circulo Class Reference	8
4.1.1	Constructor & Destructor Documentation	10
4.1.1.1	Circulo() [1/2]	10
4.1.1.2	Circulo() [2/2]	10
4.1.2	Member Function Documentation	10
4.1.2.1	calcularArea()	10
4.1.2.2	calcularPerimetro()	11
4.2	Cuadrado Class Reference	12
4.2.1	Constructor & Destructor Documentation	14
4.2.1.1	Cuadrado() [1/2]	14
4.2.1.2	Cuadrado() [2/2]	14
4.3	Figuras Class Reference	15
4.3.1	Constructor & Destructor Documentation	16
4.3.1.1	Figuras() [1/2]	16
4.3.1.2	Figuras() [2/2]	17
4.4	Triangulo Class Reference	18
4.4.1	Constructor & Destructor Documentation	20
4.4.1.1	Triangulo() [1/2]	20
4.4.1.2	Triangulo() [2/2]	20
4.4.2	Member Function Documentation	21
4.4.2.1	calcularArea()	21
4.4.2.2	calcularPerimetro()	21

5 File Documentation	23
5.1 main.cpp File Reference	23
5.1.1 Detailed Description	24
Index	25

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Figuras	15
Circulo	8
Cuadrado	12
Triangulo	18

Chapter 2

Class Index

2.1 Class List

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

Circulo	8
Cuadrado	12
Figuras	15
Triangulo	18

Chapter 3

File Index

3.1 File List

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

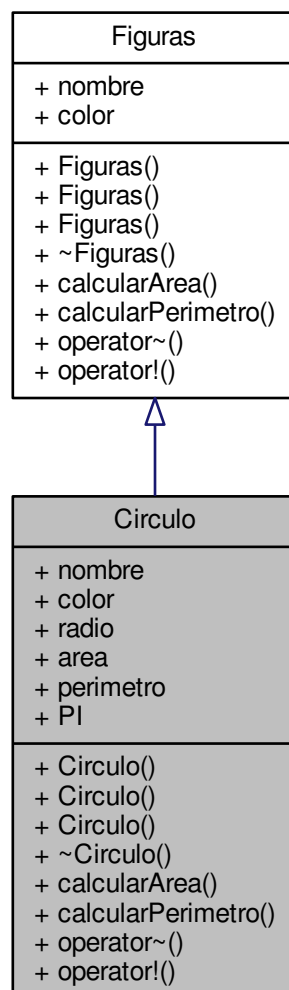
Figuras.h	??
main.cpp	Este programa realiza el calculo del area y el perimetro con los parametros establecidos de cada figura geometrica. Cada una de estas figuras posee sus caracteristicas propias de color y nombre	23

Chapter 4

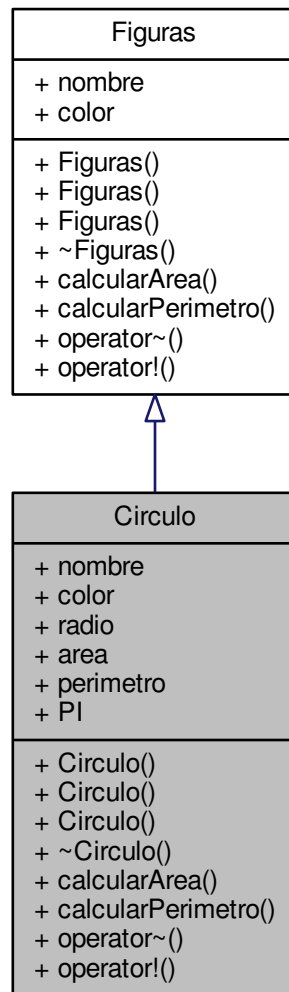
Class Documentation

4.1 Circulo Class Reference

Inheritance diagram for Circulo:



Collaboration diagram for Circulo:



Public Member Functions

- [Circulo](#) ()
Constructor vacio de la clase [Circulo](#).
- [Circulo](#) (string nombre, string color, double radio)
Constructor sobrecargado de la clase [Triangulo](#).
- [Circulo](#) (const [Circulo](#) &orig)
Constructor de la clase [Circulo](#).
- virtual [~Circulo](#) ()
Destructor de la clase [Circulo](#).
- virtual void [calcularArea](#) (double radio)
Calcula el area del circulo con sus parametros.
- virtual void [calcularPerimetro](#) (double radio)
Calcula el perimetro del circulo con sus parametros.

- virtual void `operator~` ()
Sobrecarga el operador ~ para imprimir los atributos de la clase.
- virtual void `operator!` ()
Sobrecarga el operador ~ para imprimir los atributos de la clase.

Public Attributes

- string **nombre**
- string **color**
- double **radio**
- double **area**
- double **perimetro**
- const double **PI** = 3.141592653589793238463

4.1.1 Constructor & Destructor Documentation

4.1.1.1 `Circulo()` [1/2]

```
Circulo::Circulo (
    string nombre,
    string color,
    double radio )
```

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

Parameters

<i>nombre</i>	Nombre de la figura.
<i>color</i>	Color de la figura.
<i>radio</i>	Valor del radio de la figura.

4.1.1.2 `Circulo()` [2/2]

```
Circulo::Circulo (
    const Circulo & orig )
```

Constructor de la clase [Circulo](#).

Parameters

<i>Circulo&</i>	Constante objeto.
---------------------	-------------------

4.1.2 Member Function Documentation

4.1.2.1 `calcularArea()`

```
void Circulo::calcularArea (
```

```
double radio ) [virtual]
```

Calcula el area del circulo con sus parametros.

Parameters

<i>radio</i>	Valor del radio de la figura.
--------------	-------------------------------

4.1.2.2 calcularPerimetro()

```
void Circulo::calcularPerimetro (  
    double radio ) [virtual]
```

Calcula el perimetro del circulo con sus parametros.

Parameters

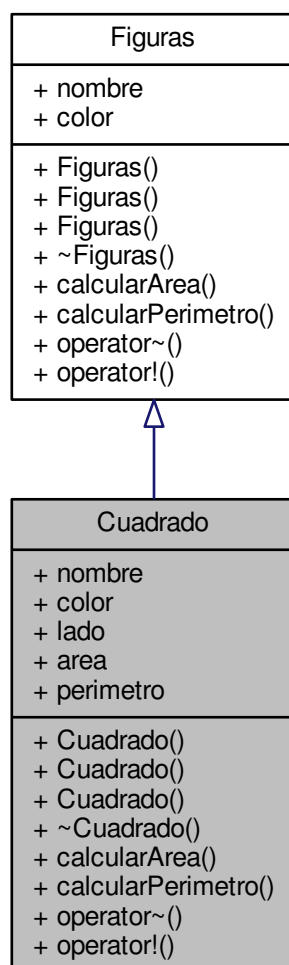
<i>radio</i>	Valor del radio de la figura.
--------------	-------------------------------

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

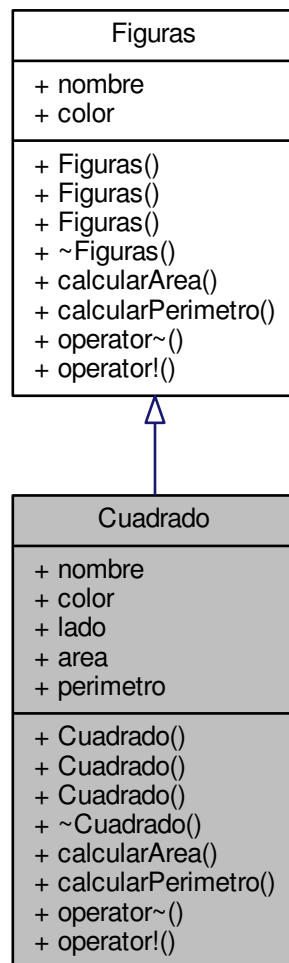
- Figuras.h
- Figuras.cpp

4.2 Cuadrado Class Reference

Inheritance diagram for Cuadrado:



Collaboration diagram for Cuadrado:



Public Member Functions

- [Cuadrado](#) ()
Constructor vacio de la clase [Cuadrado](#).
- [Cuadrado](#) (string nombre, string color, double lado)
Constructor sobrecargado de la clase [Cuadrado](#).
- [Cuadrado](#) (const [Cuadrado](#) &orig)
Constructor de la clase [Cuadrado](#).
- virtual [~Cuadrado](#) ()
Destructor de la clase [Cuadrado](#).
- virtual void [calcularArea](#) (double lado)
Llama a la funcion en la clase [Figuras](#).
- virtual void [calcularPerimetro](#) (double lado)
Llama a la funcion en la clase [Figuras](#).

- virtual void `operator~` ()
Sobrecarga el operador ~ para imprimir los atributos de la clase.
- virtual void `operator!` ()
Sobrecarga el operador ! para imprimir los calculos de la clase.

Public Attributes

- string **nombre**
- string **color**
- double **lado**
- double **area**
- double **perimetro**

4.2.1 Constructor & Destructor Documentation

4.2.1.1 Cuadrado() [1/2]

```
Cuadrado::Cuadrado (
    string nombre,
    string color,
    double lado )
```

Constructor sobrecargado de la clase `Cuadrado`.

Parameters

<i>nombre</i>	Nombre de la figura.
<i>color</i>	Color de la figura.
<i>lado</i>	Valor del lado de la figura.

4.2.1.2 Cuadrado() [2/2]

```
Cuadrado::Cuadrado (
    const Cuadrado & orig )
```

Constructor de la clase `Cuadrado`.

Parameters

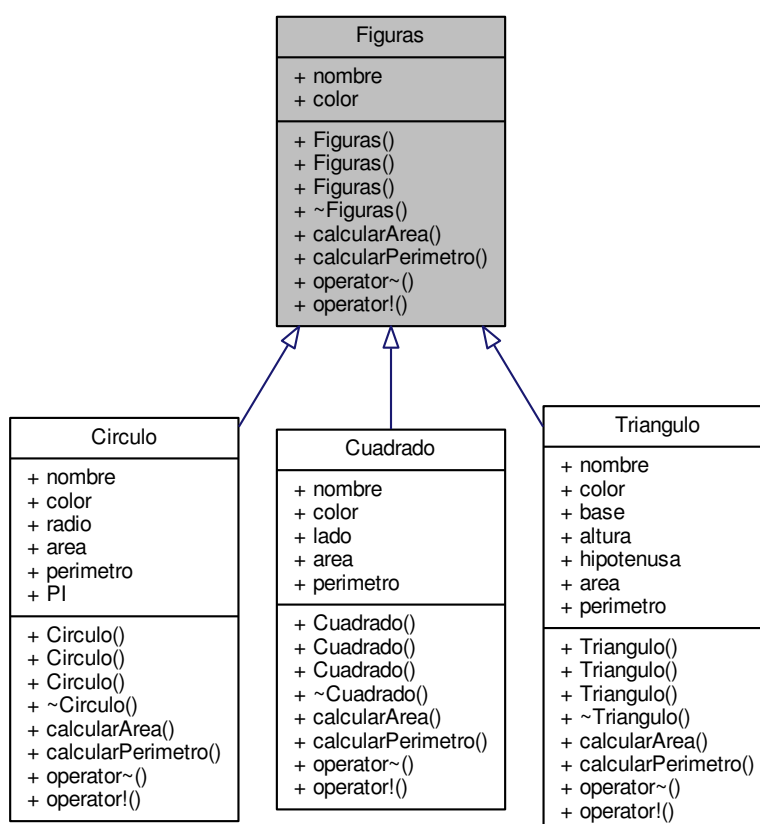
<i>Cuadrado&</i>	Constante objeto.
----------------------	-------------------

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

- Figuras.h
- Figuras.cpp

4.3 Figuras Class Reference

Inheritance diagram for Figuras:



Collaboration diagram for Figuras:

Figuras
+ nombre + color
+ Figuras() + Figuras() + Figuras() + ~Figuras() + calcularArea() + calcularPerimetro() + operator~() + operator!()

Public Member Functions

- [Figuras \(\)](#)
Constructor vacio de clase [Figuras](#).
- [Figuras \(string nombre, string color\)](#)
Constructor sobrecargado de la clase [Figuras](#).
- [Figuras \(const \[Figuras\]\(#\) &orig\)](#)
Constructor de la clase [Figuras](#).
- virtual [~Figuras \(\)](#)
Destructor de la clase [Figuras](#).
- virtual void [calcularArea \(\)](#)
Llama a la funcion en la clase [Figuras](#).
- virtual void [calcularPerimetro \(\)](#)
Llama a la funcion en la clase [Figuras](#).
- virtual void [operator~ \(\)](#)
Sobrecarga el operador ~ para imprimir los atributos de la clase.
- virtual void [operator! \(\)](#)
Sobrecarga el operador ! para imprimir los calculos de la clase.

Public Attributes

- string **nombre**
- string **color**

4.3.1 Constructor & Destructor Documentation

4.3.1.1 [Figuras\(\)](#) [1/2]

```
Figuras::Figuras (
    string nombre,
    string color )
```

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

Parameters

<i>nombre</i>	Nombre de la figura.
<i>color</i>	Color de la figura.

4.3.1.2 Figuras() [2/2]

```
Figuras::Figuras (  
    const Figuras & orig )
```

Constructor de la clase Figuras.

Parameters

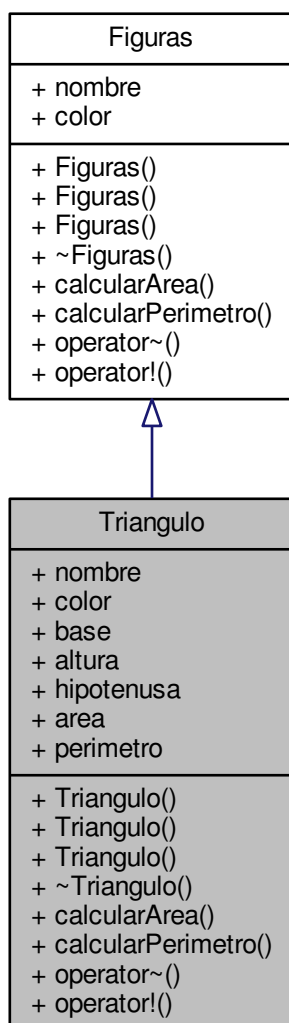
<i>Figuras&</i>	Constante objeto.
---------------------	-------------------

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

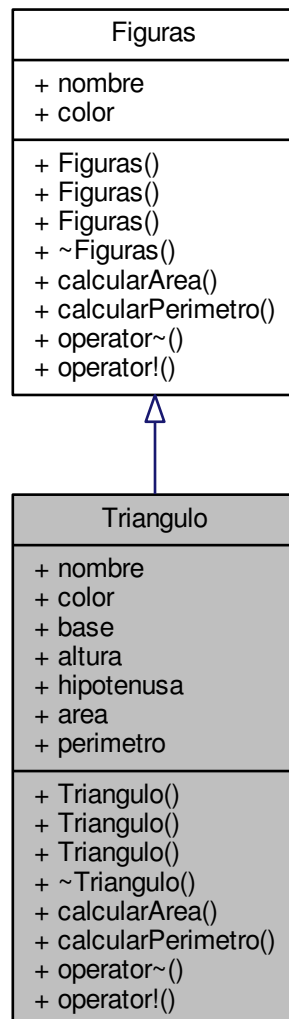
- Figuras.h
- Figuras.cpp

4.4 Triangulo Class Reference

Inheritance diagram for Triangulo:



Collaboration diagram for Triangulo:



Public Member Functions

- [Triangulo](#) ()
Constructor vacio de la clase [Triangulo](#).
- [Triangulo](#) (string nombre, string color, double base, double altura, double hipotenusa)
Constructor sobrecargado de la clase [Triangulo](#).
- [Triangulo](#) (const [Triangulo](#) &orig)
Constructor de la clase [Triangulo](#).
- virtual [~Triangulo](#) ()
Destructor de la clase [Cuadrado](#).
- virtual void [calcularArea](#) (double base, double altura)
Calcula el area del triangulo con sus parametros.
- void [calcularPerimetro](#) (double base, double altura, double hipotenusa)

Calcula el perimetro del triangulo con sus parametros.

- virtual void `operator~()`

Sobrecarga el operador ~ para imprimir los atributos de la clase.

- virtual void `operator!()`

Sobrecarga el operador ! para imprimir los calculos de la clase.

Public Attributes

- string **nombre**
- string **color**
- double **base**
- double **altura**
- double **hipotenusa**
- double **area**
- double **perimetro**

4.4.1 Constructor & Destructor Documentation

4.4.1.1 Triangulo() [1/2]

```
Triangulo::Triangulo (
    string nombre,
    string color,
    double base,
    double altura,
    double hipotenusa )
```

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

Parameters

<i>nombre</i>	Nombre de la figura.
<i>color</i>	Color de la figura.
<i>base</i>	Valor de la base de la figura.
<i>altura</i>	Valor de la altura de la figura.
<i>hipotenusa</i>	Valor de la hipotenusa de la figura.

4.4.1.2 Triangulo() [2/2]

```
Triangulo::Triangulo (
    const Triangulo & orig )
```

Constructor de la clase [Triangulo](#).

Parameters

<i>Triangulo&</i>	Constante objeto.
-----------------------	-------------------

4.4.2 Member Function Documentation

4.4.2.1 calcularArea()

```
void Triangulo::calcularArea (
    double base,
    double altura ) [virtual]
```

Calcula el area del triangulo con sus parametros.

Parameters

<i>base</i>	Valor de la base de la figura.
<i>altura</i>	Valor de la altura de la figura.

4.4.2.2 calcularPerimetro()

```
void Triangulo::calcularPerimetro (
    double base,
    double altura,
    double hipotenusa )
```

Calcula el perimetro del triangulo con sus parametros.

Parameters

<i>base</i>	Valor de la base de la figura.
<i>altura</i>	Valor de la altura de la figura.
<i>hipotenusa</i>	Valor de la hipotenusa de la figura.

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

- Figuras.h
- Figuras.cpp

Chapter 5

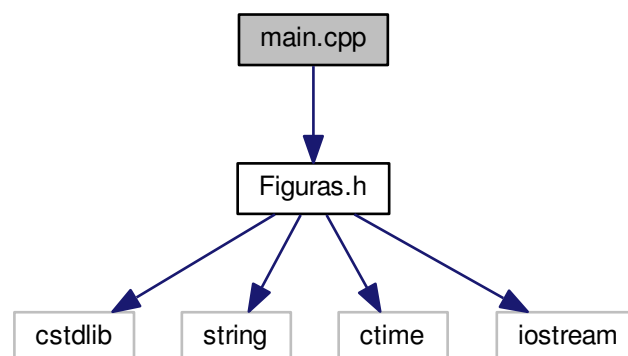
File Documentation

5.1 main.cpp File Reference

Este programa realiza el calculo del area y el perimetro con los parametros establecidos de cada figura geometrica. Cada una de estas figuras posee sus caracteristicas propias de color y nombre.

```
#include "Figuras.h"
```

Include dependency graph for main.cpp:



Functions

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

5.1.1 Detailed Description

Este programa realiza el calculo del area y el perimetro con los parametros establecidos de cada figura geometrica. Cada una de estas figuras posee sus caracteristicas propias de color y nombre.

Author

Jose Fernando Gonzalez Salas & Isaac Gomez Sanchez

Date

12 de setiembre, 2016

Index

- calcularArea
 - Circulo, [10](#)
 - Triangulo, [21](#)
- calcularPerimetro
 - Circulo, [11](#)
 - Triangulo, [21](#)
- Circulo, [8](#)
 - calcularArea, [10](#)
 - calcularPerimetro, [11](#)
 - Circulo, [10](#)
- Cuadrado, [12](#)
 - Cuadrado, [14](#)
- Figuras, [15](#)
 - Figuras, [16](#), [17](#)
- main.cpp, [23](#)
- Triangulo, [18](#)
 - calcularArea, [21](#)
 - calcularPerimetro, [21](#)
 - Triangulo, [20](#)