

My Project

Generated by Doxygen 1.9.8

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 CuentaBancaria Class Reference	5
3.1.1 Constructor & Destructor Documentation	5
3.1.1.1 CuentaBancaria()	5
3.1.2 Member Function Documentation	6
3.1.2.1 depositar()	6
3.1.2.2 obtenerSaldo()	6
3.1.2.3 retirar()	6
4 File Documentation	7
4.1 CuentaBancaria.h	7
4.2 main.cpp File Reference	8
4.2.1 Detailed Description	8
4.2.2 Function Documentation	9
4.2.2.1 main()	9
Index	11

Chapter 1

Class Index

1.1 Class List

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

CuentaBancaria	5
--	---

Chapter 2

File Index

2.1 File List

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

CuentaBancaria.h	7
main.cpp	
Programa para gestionar cuentas bancarias	8

Chapter 3

Class Documentation

3.1 CuentaBancaria Class Reference

Public Member Functions

- [CuentaBancaria](#) (string numC, float s)
constructor de la clase [CuentaBancaria](#)
- string **GetNumeroCuenta** ()
Getter para el número de cuenta.
- float **GetSaldo** ()
Getter para el saldo.
- void [depositar](#) (float cantidad)
depositar dinero
- void [retirar](#) (float cantidad)
Retira dinero de la cuenta si hay saldo suficiente y no está bloqueada.
- void **bloquearRetiros** ()
Bloquea Retiros.
- float [obtenerSaldo](#) ()
obtiene el saldo actual
- void **mostrarCuenta** ()
Muestra el número de cuenta y el saldo inicial.

3.1.1 Constructor & Destructor Documentation

3.1.1.1 CuentaBancaria()

```
CuentaBancaria::CuentaBancaria (
    string numC,
    float s ) [inline]
```

constructor de la clase [CuentaBancaria](#)

Parameters

<i>numeroCuenta</i>	numero de cuenta bancaria
<i>saldo</i>	saldo inicial

3.1.2 Member Function Documentation

3.1.2.1 depositar()

```
void CuentaBancaria::depositar (
    float cantidad ) [inline]
```

depositar dinero

Parameters

<i>cantidad</i>	cantidad depositada
-----------------	---------------------

3.1.2.2 obtenerSaldo()

```
float CuentaBancaria::obtenerSaldo ( ) [inline]
```

obtiene el saldo actual

Returns

saldo disponible

3.1.2.3 retirar()

```
void CuentaBancaria::retirar (
    float cantidad ) [inline]
```

Retira dinero de la cuenta si hay saldo suficiente y no está bloqueada.

Parameters

<i>cantidad</i>	cantidad a retirar
-----------------	--------------------

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

- CuentaBancaria.h

Chapter 4

File Documentation

4.1 CuentaBancaria.h

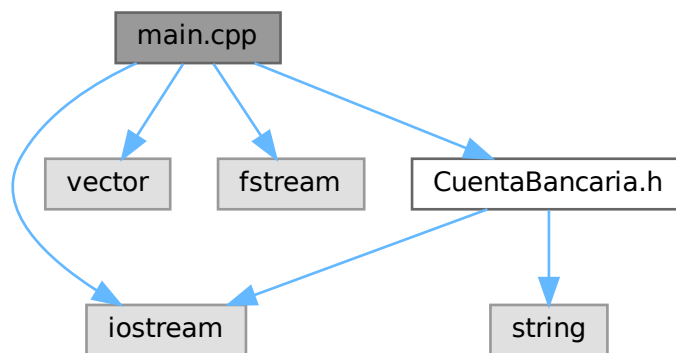
```
00001 #ifndef CUENTA_BANCARIA_H
00002 #define CUENTA_BANCARIA_H
00003
00004 #include <iostream>
00005 #include <string>
00006 using namespace std;
00007
00008 class CuentaBancaria{
00009     private:
00010         string numeroCuenta;
00011         float saldo;
00012         bool retirosBloqueados;
00013
00014     public:
00015
00021         CuentaBancaria(string numC, float s): numeroCuenta(numC), saldo(s), retirosBloqueados(false)
00022     {}
00025         string GetNumeroCuenta(){
00026             return numeroCuenta;
00027         }
00031         float GetSaldo(){
00032             return saldo;
00033         }
00038         void depositar(float cantidad){
00039             if (cantidad > 0 && retirosBloqueados==false){
00040                 saldo = saldo + cantidad;
00041                 cout << "Nuevo Saldo: " << saldo << endl;
00042             }
00043             else if(cantidad < 0) {
00044                 cout << "Valor No Válido" << endl;
00045             }
00046             else if (retirosBloqueados == true){
00047                 cout << "La cuenta está Bloqueada" << endl;
00048             }
00049         }
00054         void retirar(float cantidad){
00055             if(retirosBloqueados){
00056                 cout << "Retiros Bloqueados" << endl;
00057             }
00058             else if (cantidad > 0 && cantidad <= saldo){
00059                 saldo = saldo - cantidad;
00060                 cout << "Retiro Exitoso, Saldo Nuevo: " << saldo << endl;
00061             }
00062             else {
00063                 cout << "Fondos Insuficientes" << endl;
00064             }
00065         }
00069         void bloquearRetiros(){
00070             retirosBloqueados = true;
00071             cout << "Cuenta Bloqueada" << endl;
00072         }
00078         float obtenerSaldo(){
00079             return saldo;
00080         }
00084         void mostrarCuenta(){
00085             cout << numeroCuenta << ", " << saldo << endl;
00086         }
00087     };
```

```
00088
00089
00090 #endif
```

4.2 main.cpp File Reference

Programa para gestionar cuentas bancarias.

```
#include <iostream>
#include <vector>
#include <fstream>
#include "CuentaBancaria.h"
Include dependency graph for main.cpp:
```



Functions

- int [main](#) ()
Función principal del programa.

4.2.1 Detailed Description

Programa para gestionar cuentas bancarias.

Este programa permite crear cuentas bancarias, realizar depósitos, retiros, bloquear cuentas y generar un archivo con los datos finales de las cuentas.

Author

Lenin Enrique García Caraballo y David Orlando Riaño Morales

Date

2025-04-04

4.2.2 Function Documentation

4.2.2.1 main()

```
int main ( )
```

Función principal del programa.

Solicita al usuario ingresar cuentas bancarias, realiza operaciones de prueba como depósitos, retiros y bloqueos, y genera un archivo con los datos finales.

Returns

int Código de salida del programa.

Index

- CuentaBancaria, [5](#)
 - CuentaBancaria, [5](#)
 - depositar, [6](#)
 - obtenerSaldo, [6](#)
 - retirar, [6](#)
- depositar
 - CuentaBancaria, [6](#)
- main
 - main.cpp, [9](#)
- main.cpp, [8](#)
 - main, [9](#)
- obtenerSaldo
 - CuentaBancaria, [6](#)
- retirar
 - CuentaBancaria, [6](#)