MyVensim

Generated by Doxygen 1.9.6

Chapter 1

bcc-322

Código referente ao trabalho prático desenvolvido para a disciplina Engenharia de Software I.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

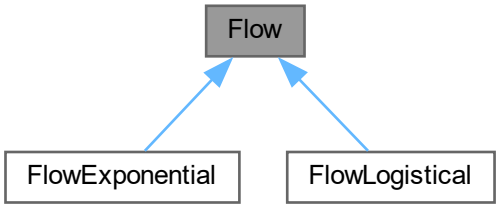
Chapter 5

Class Documentation

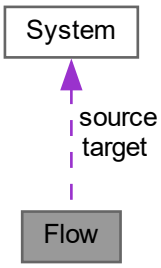
5.1 Flow Class Reference

`#include <flow.h>`

Inheritance diagram for Flow:



Collaboration diagram for Flow:



Public Member Functions

- Flow ()
- Flow (Flow &obj)
- Flow (const string name, System *source, System *target)
- virtual ~Flow ()
- string getName () const
- void setName (const string name)
- System * getSource () const
- void setSource (System *source)
- System * getTarget () const
- void setTarget (System *target)
- bool operator== (const Flow &obj) const
- bool operator!= (const Flow &obj) const
- Flow & operator= (const Flow &obj)
- virtual float execute ()=0

Protected Attributes

- string name
- System * source
- System * target

5.1.1 Constructor & Destructor Documentation

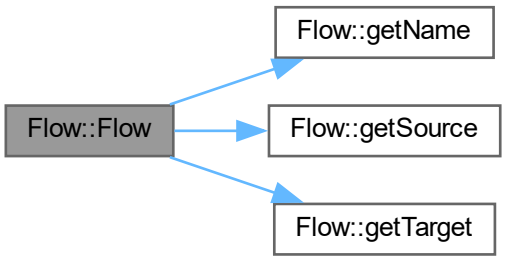
5.1.1.1 Flow() [1/3]

```
Flow::Flow ( )
```

5.1.1.2 Flow() [2/3]

```
Flow::Flow (
            Flow & obj )
```

Here is the call graph for this function:



5.1.1.3 Flow() [3/3]

```
Flow::Flow (
            const string name,
            System * source,
            System * target )
```

5.1.1.4 ~Flow()

```
Flow::~Flow ( )  [virtual]
```

5.1.2 Member Function Documentation

5.1.2.1 execute()

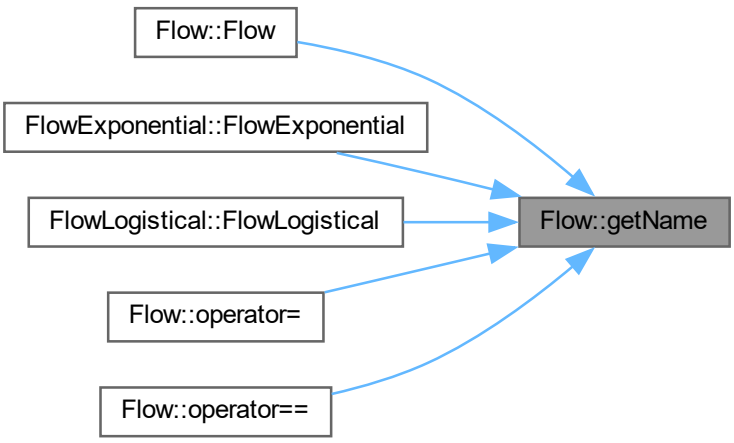
```
virtual float Flow::execute ( )  [pure virtual]
```

Implemented in FlowExponential, and FlowLogistical.

5.1.2.2 getName()

```
string Flow::getName ( ) const
```

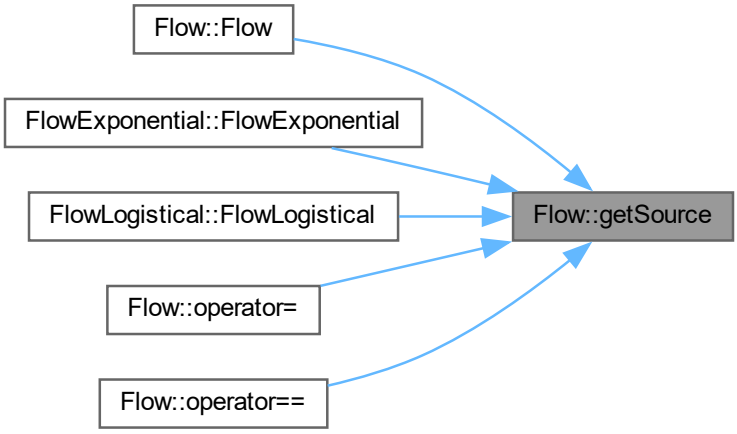
Here is the caller graph for this function:



5.1.2.3 getSource()

```
System * Flow::getSource ( ) const
```

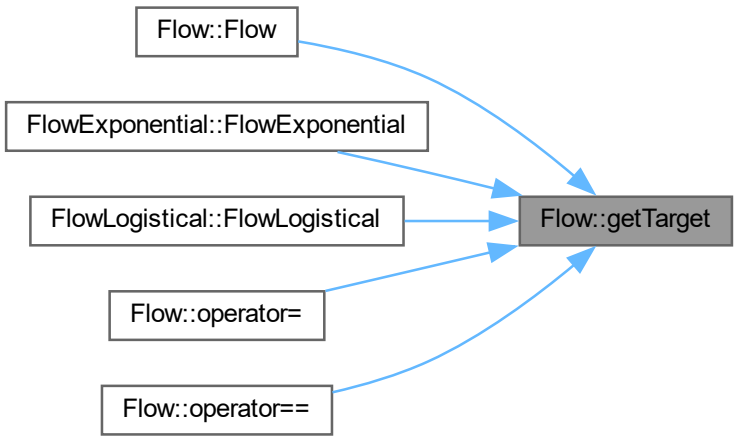
Here is the caller graph for this function:



5.1.2.4 getTarget()

```
System * Flow::getTarget ( ) const
```

Here is the caller graph for this function:



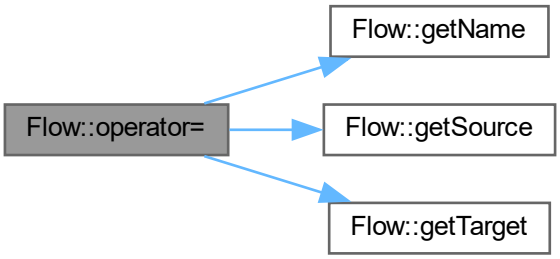
5.1.2.5 operator"!=()

```
bool Flow::operator!= (
            const Flow & obj ) const
```

5.1.2.6 operator=()

```
Flow & Flow::operator= (
            const Flow & obj )
```

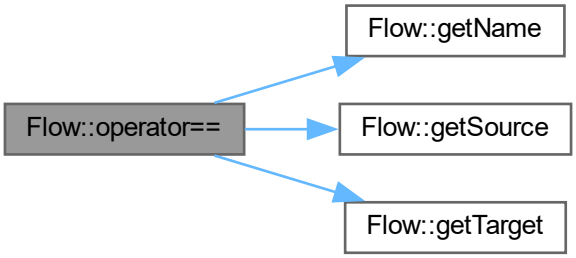
Here is the call graph for this function:



5.1.2.7 operator==()

```
bool Flow::operator== (
            const Flow & obj ) const
```

Here is the call graph for this function:



5.1.2.8 setName()

```
void Flow::setName (
            const string name )
```

5.1.2.9 setSource()

```
void Flow::setSource (
            System * source )
```

5.1.2.10 setTarget()

```
void Flow::setTarget (
            System * target )
```

5.1.3 Member Data Documentation

5.1.3.1 name

```
string Flow::name  [protected]
```

5.1.3.2 source

```
System* Flow::source  [protected]
```

5.1.3.3 target

```
System* Flow::target  [protected]
```

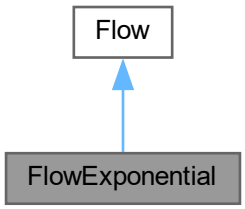
The documentation for this class was generated from the following files:

- src/flow.h
- src/flow.cpp

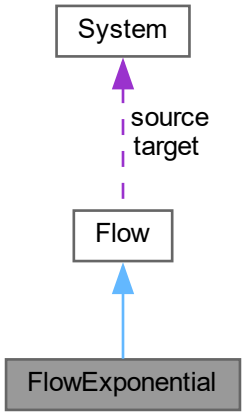
5.2 FlowExponential Class Reference

`#include <flowExponential.h>`

Inheritance diagram for FlowExponential:



Collaboration diagram for FlowExponential:



Public Member Functions

- FlowExponential ()
- FlowExponential (Flow &obj)
- FlowExponential (const string name, System *source, System *target)
- virtual ~FlowExponential ()
- virtual float execute ()

Public Member Functions inherited from Flow

- Flow ()
- Flow (Flow &obj)
- Flow (const string name, System *source, System *target)
- virtual ~Flow ()
- string getName () const
- void setName (const string name)
- System * getSource () const
- void setSource (System *source)
- System * getTarget () const
- void setTarget (System *target)
- bool operator== (const Flow &obj) const
- bool operator!= (const Flow &obj) const
- Flow & operator= (const Flow &obj)
- virtual float execute ()=0

Additional Inherited Members

Protected Attributes inherited from Flow

- string name
- System * source
- System * target

5.2.1 Constructor & Destructor Documentation

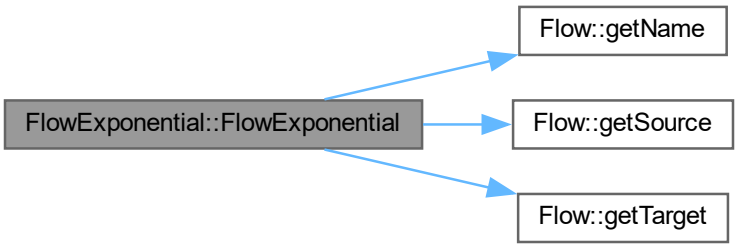
5.2.1.1 FlowExponential() [1/3]

```
FlowExponential::FlowExponential ( )
```

5.2.1.2 FlowExponential() [2/3]

```
FlowExponential::FlowExponential (
            Flow & obj )
```

Here is the call graph for this function:



5.2.1.3 FlowExponential() [3/3]

```
FlowExponential::FlowExponential (
            const string name,
            System * source,
            System * target )
```

5.2.1.4 ~FlowExponential()

```
FlowExponential::~FlowExponential ( )  [virtual]
```

5.2.2 Member Function Documentation

5.2.2.1 execute()

```
float FlowExponential::execute ( )  [virtual]
```

Implements Flow.

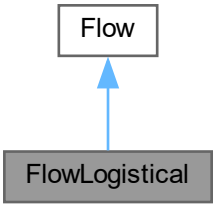
The documentation for this class was generated from the following files:

- test/funcional/flowExponential.h
- test/funcional/flowExponential.cpp

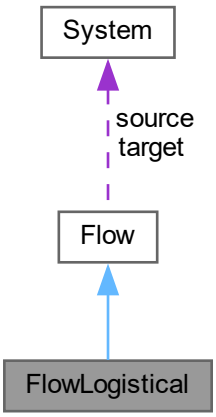
5.3 FlowLogistical Class Reference

```
#include <flowLogistical.h>
```

Inheritance diagram for FlowLogistical:



Collaboration diagram for FlowLogistical:



Public Member Functions

- FlowLogistical ()
- FlowLogistical (Flow &obj)
- FlowLogistical (const string name, System *source, System *target)
- virtual ~FlowLogistical ()
- virtual float execute ()

Public Member Functions inherited from Flow

- Flow ()
- Flow (Flow &obj)
- Flow (const string name, System *source, System *target)
- virtual ~Flow ()
- string getName () const
- void setName (const string name)
- System * getSource () const
- void setSource (System *source)
- System * getTarget () const
- void setTarget (System *target)
- bool operator== (const Flow &obj) const
- bool operator!= (const Flow &obj) const
- Flow & operator= (const Flow &obj)
- virtual float execute ()=0

Additional Inherited Members

Protected Attributes inherited from Flow

- string name
- System * source
- System * target

5.3.1 Constructor & Destructor Documentation

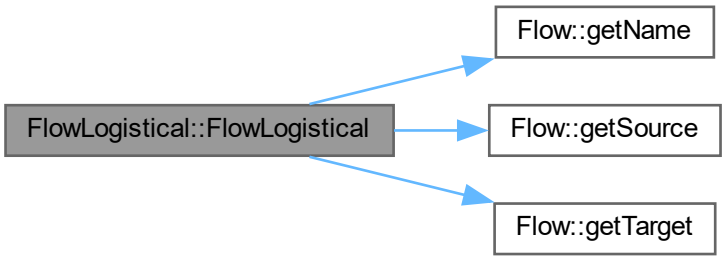
5.3.1.1 FlowLogistical() [1/3]

```
FlowLogistical::FlowLogistical ( )
```

5.3.1.2 FlowLogistical() [2/3]

```
FlowLogistical::FlowLogistical (
            Flow & obj )
```

Here is the call graph for this function:



5.3.1.3 FlowLogistical() [3/3]

```
FlowLogistical::FlowLogistical (
            const string name,
            System * source,
            System * target )
```

5.3.1.4 ~FlowLogistical()

```
FlowLogistical::~FlowLogistical ( )  [virtual]
```

5.3.2 Member Function Documentation

5.3.2.1 execute()

```
float FlowLogistical::execute ( )  [virtual]
```

Implements Flow.

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

- test/funcional/flowLogistical.h
- test/funcional/flowLogistical.cpp

5.4 Model Class Reference

```
#include <model.h>
```

Public Types

- typedef vector< Flow * >::iterator itFlow
- typedef vector< System * >::iterator itSystem

Public Member Functions

- Model ()
- Model (const string name)
- Model (const string name, vector< Flow * > &flows, vector< System * > &systems)
- virtual ~Model ()
- string getName () const
- void setName (const string name)
- itFlow getFlowBegin ()
- itFlow getFlowEnd ()
- int getFlowSize ()
- itSystem getSystemBegin ()
- itSystem getSystemEnd ()
- int getSystemSize ()
- void add (System *)
- void add (Flow *)
- bool remove (System *)
- bool remove (Flow *)
- void clear ()
- void show ()
- void run (int, int, int)

Protected Attributes

- string name
- vector< Flow * > flows
- vector< System * > systems

5.4.1 Member Typedef Documentation

5.4.1.1 itFlow

```
typedef vector<Flow*>::iterator Model::itFlow
```

5.4.1.2 itSystem

```
typedef vector<System*>::iterator Model::itSystem
```

5.4.2 Constructor & Destructor Documentation

5.4.2.1 Model() [1/3]

```
Model::Model ( )
```

5.4.2.2 Model() [2/3]

```
Model::Model (
            const string name )
```

5.4.2.3 Model() [3/3]

```
Model::Model (
            const string name,
            vector< Flow * > & flows,
            vector< System * > & systems )
```

5.4.2.4 ~Model()

```
Model::~Model ( )  [virtual]
```

5.4.3 Member Function Documentation

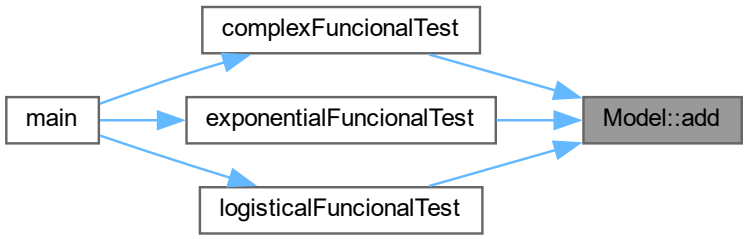
5.4.3.1 add() [1/2]

```
void Model::add (
            Flow * flow )
```

5.4.3.2 add() [2/2]

```
void Model::add (
            System * subSystem )
```

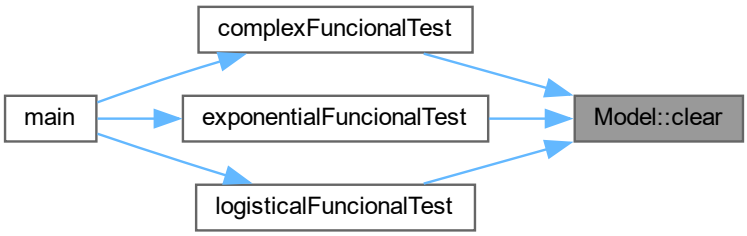
Here is the caller graph for this function:



5.4.3.3 clear()

```
void Model::clear ( )
```

Here is the caller graph for this function:



5.4.3.4 getFlowBegin()

```
Model::itFlow Model::getFlowBegin ( )
```

5.4.3.5 getFlowEnd()

```
Model::itFlow Model::getFlowEnd ( )
```

5.4.3.6 getFlowSize()

```
int Model::getFlowSize ( )
```

5.4.3.7 getName()

```
string Model::getName ( ) const
```

5.4.3.8 getSystemBegin()

```
Model::itSystem Model::getSystemBegin ( )
```

5.4.3.9 getSystemEnd()

```
Model::itSystem Model::getSystemEnd ( )
```

5.4.3.10 getSystemSize()

```
int Model::getSystemSize ( )
```

5.4.3.11 remove() [1/2]

```
bool Model::remove (
            Flow * obj )
```

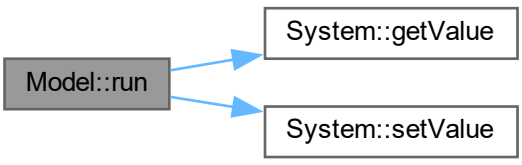
5.4.3.12 remove() [2/2]

```
bool Model::remove (
            System * obj )
```

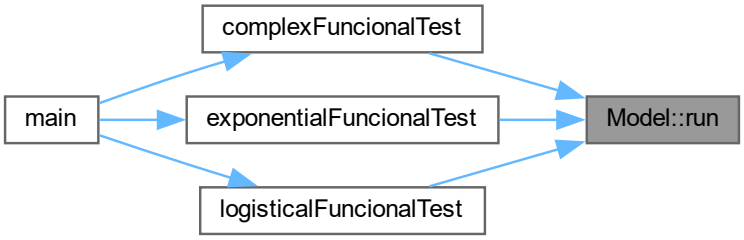
5.4.3.13 run()

```
void Model::run (
            int start,
            int finish,
            int increment )
```

Here is the call graph for this function:



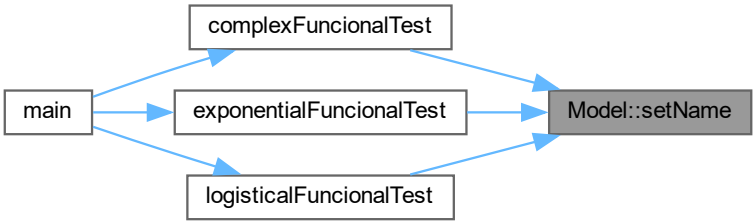
Here is the caller graph for this function:



5.4.3.14 setName()

```
void Model::setName (
            const string name )
```

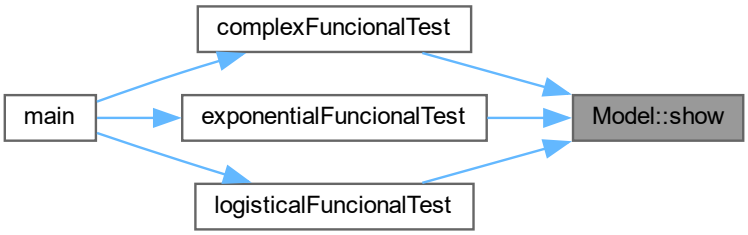
Here is the caller graph for this function:



5.4.3.15 show()

```
void Model::show ( )
```

Here is the caller graph for this function:



5.4.4 Member Data Documentation

5.4.4.1 flows

```
vector<Flow*> Model::flows  [protected]
```

5.4.4.2 name

```
string Model::name  [protected]
```

5.4.4.3 systems

```
vector<System*> Model::systems  [protected]
```

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

- src/model.h
- src/model.cpp

5.5 System Class Reference

```
#include <system.h>
```

Public Member Functions

- System ()
- System (const string name)
- System (float value)
- System (System &obj)
- System (const string name, float value)
- virtual ~System ()
- string getName () const
- void setName (const string name)
- float getValue () const
- void setValue (float value)
- System & operator= (const System &obj)

Protected Attributes

- string name
- float value

5.5.1 Constructor & Destructor Documentation

5.5.1.1 System() [1/5]

```
System::System ( )
```

5.5.1.2 System() [2/5]

```
System::System (
            const string name )
```

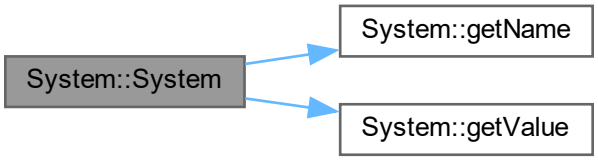
5.5.1.3 System() [3/5]

```
System::System (
            float value )
```

5.5.1.4 System() [4/5]

```
System::System (
            System & obj )
```

Here is the call graph for this function:



5.5.1.5 System() [5/5]

```
System::System (
            const string name,
            float value )
```

5.5.1.6 ~System()

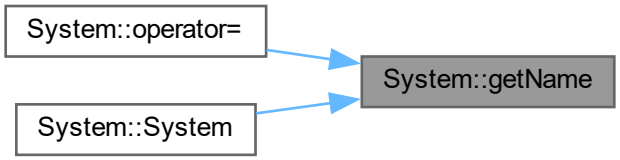
```
System::~System ( )  [virtual]
```

5.5.2 Member Function Documentation

5.5.2.1 getName()

```
string System::getName ( ) const
```

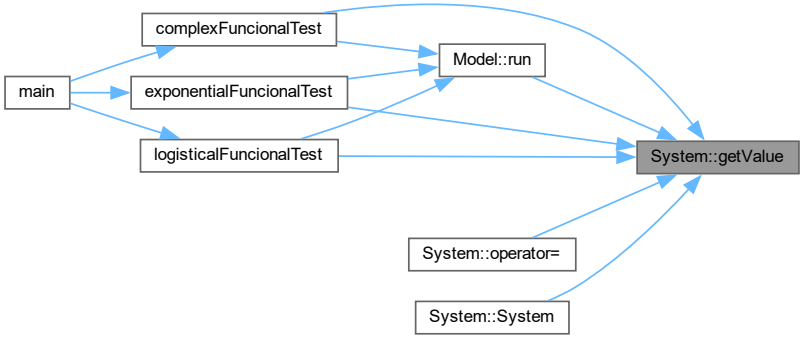
Here is the caller graph for this function:



5.5.2.2 getValue()

```
float System::getValue ( ) const
```

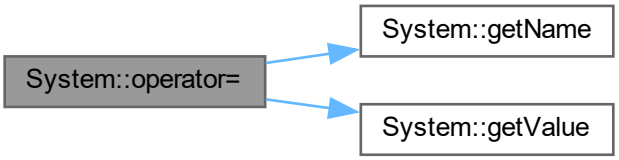
Here is the caller graph for this function:



5.5.2.3 operator=()

```
System & System::operator= (
            const System & obj )
```

Here is the call graph for this function:



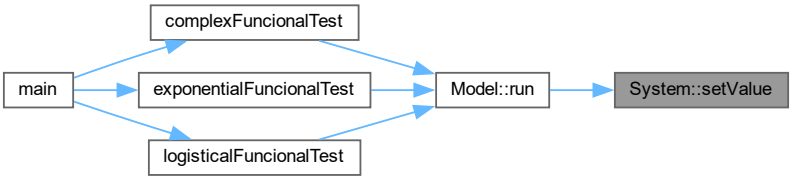
5.5.2.4 setName()

```
void System::setName (
            const string name )
```

5.5.2.5 setValue()

```
void System::setValue (
            float value )
```

Here is the caller graph for this function:



5.5.3 Member Data Documentation

5.5.3.1 name

```
string System::name  [protected]
```

5.5.3.2 value

```
float System::value  [protected]
```

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

- src/system.h
- src/system.cpp

Chapter 6

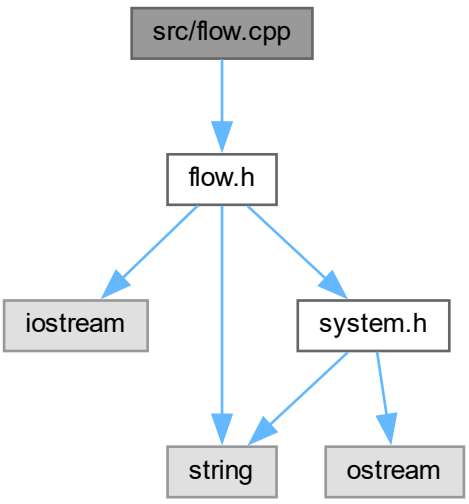
File Documentation

6.1 README.md File Reference

6.2 src/flow.cpp File Reference

```
#include "flow.h"
```

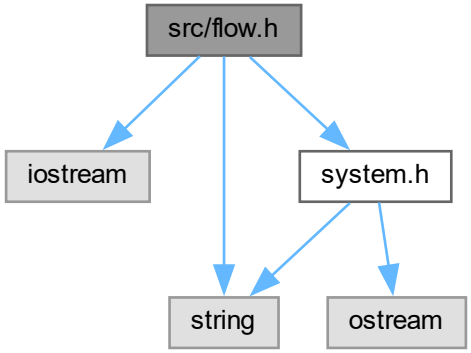
Include dependency graph for flow.cpp:



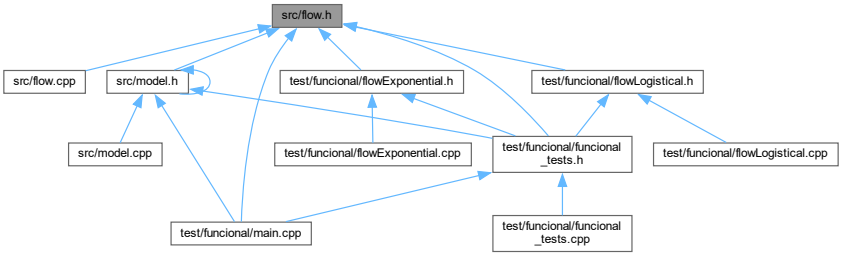
6.3 src/flow.h File Reference

```
#include <iostream>
#include <string>
```

`#include "system.h"`
Include dependency graph for flow.h:



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



Classes

- class Flow

6.4 flow.h

Go to the documentation of this file.

```
00001 #ifndef FLOW_H
00002 #define FLOW_H
00003
00004 #include <iostream>
00005 #include <string>
00006 #include "system.h"
00007
00008 class Flow {
00009     protected:
00010         string name;
00011         System *source;
00012         System *target;
00013     public:
00014         Flow();
```

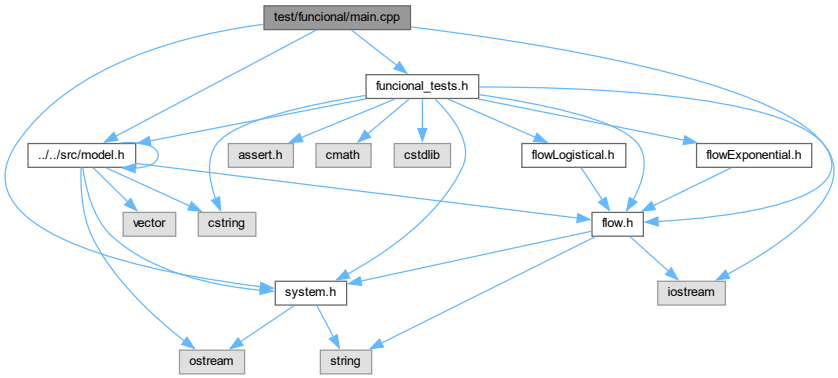
```
00015         Flow(Flow &obj);
00016         Flow(const string name, System *source, System *target);
00017         virtual ~Flow();
00018 
00019         string getName() const;
00020         void setName(const string name);
00021         System *getSource() const;
00022         void setSource(System *source);
00023         System *getTarget() const;
00024         void setTarget(System *target);
00025 
00026         bool operator==(const Flow &obj) const;
00027         bool operator!=(const Flow &obj) const;
00028         Flow &operator= (const Flow &obj);
00029 
00030         virtual float execute() = 0;
00031 };
00032 
00033 #endif
```

6.5 src/main.cpp File Reference

6.6 test/funcional/main.cpp File Reference

```
#include "funcional_tests.h"
#include "..\..\src\model.h"
#include "..\..\src\system.h"
#include "..\..\src\flow.h"
```

Include dependency graph for main.cpp:



Macros

- #define MAIN_FUNCIONAL_TESTS

Functions

- int main ()

6.6.1 Macro Definition Documentation

6.6.1.1 MAIN_FUNCIONAL_TESTS

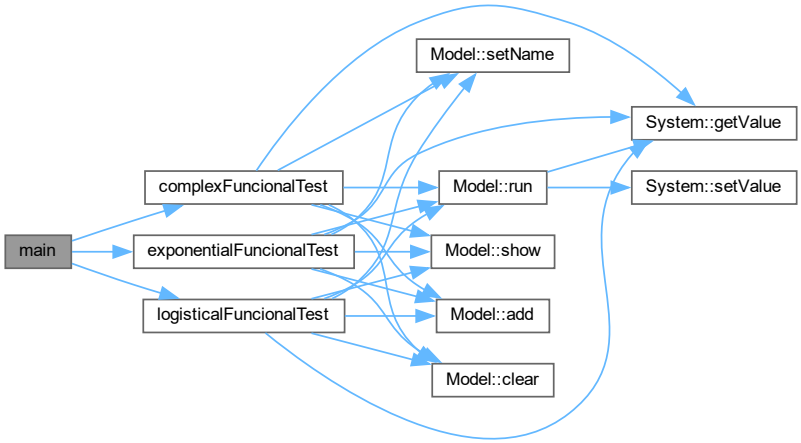
```
#define MAIN_FUNCIONAL_TESTS
```

6.6.2 Function Documentation

6.6.2.1 main()

```
int main ( )
```

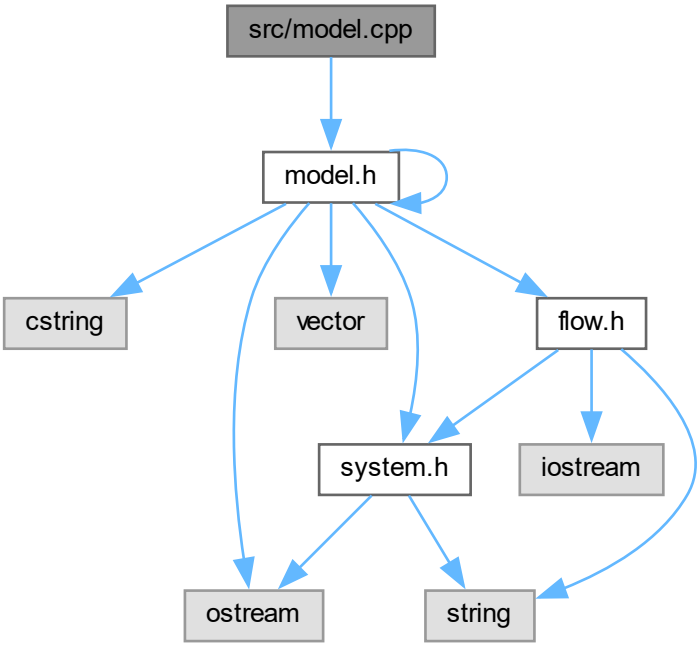
Here is the call graph for this function:



6.7 src/model.cpp File Reference

```
#include "model.h"
```

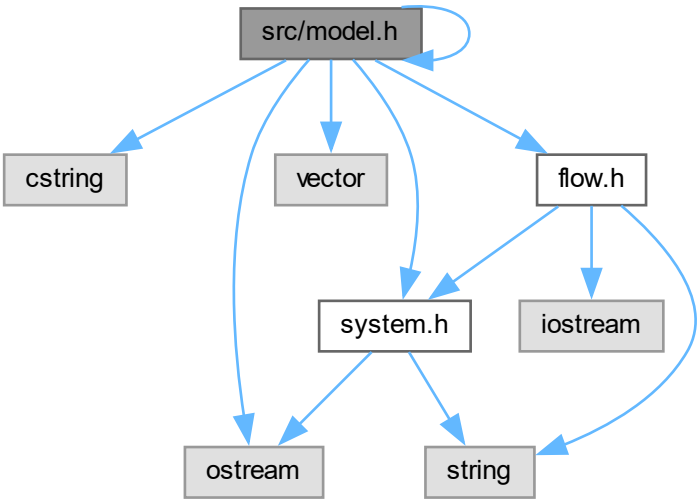
Include dependency graph for model.cpp:



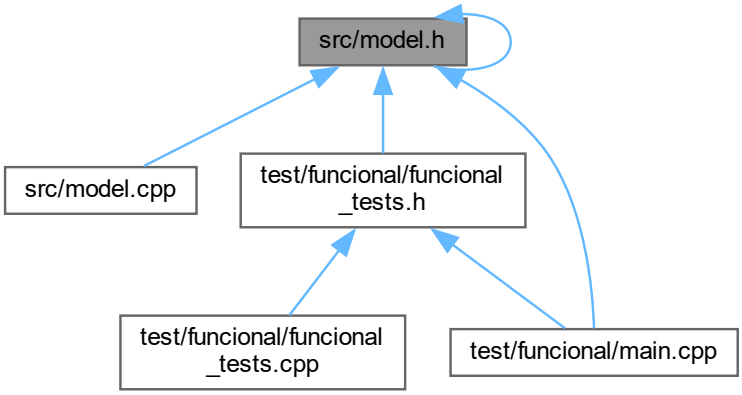
6.8 src/model.h File Reference

```
#include <cstring>
#include <ostream>
#include <vector>
#include "flow.h"
#include "system.h"
#include "model.h"
```

Include dependency graph for model.h:



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



Classes

- class Model

6.9 model.h

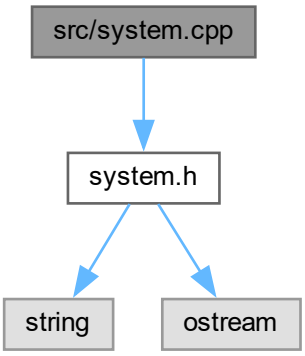
Go to the documentation of this file.

```
00001 #ifndef MODEL_H
00002 #define MODEL_H
00003 #include <cstring>
00004 #include <ostream>
00005 #include <vector>
00006 #include "flow.h"
00007 //#include "flowExponential.h"
00008 //#include "flowLogistic.h"
00009 #include "system.h"
00010 #include "model.h"
00011
00012 class Model {
00013     protected:
00014         string name;
00015         vector<Flow*> flows;
00016         vector<System*> systems;
00017
00018     private:
00019         Model(Model& obj);
00020         Model& operator= (const Model& obj);
00021
00022     public:
00023         Model();
00024         Model(const string name);
00025         Model(const string name, vector<Flow*> &flows, vector<System*> &systems);
00026         virtual ~Model();
00027
00028         typedef typename vector<Flow*> :: iterator itFlow;
00029         typedef typename vector<System*> :: iterator itSystem;
00030
00031         string getName() const;
00032         void setName(const string name);
00033
00034         itFlow getFlowBegin();
00035         itFlow getFlowEnd();
00036         int getFlowSize();
00037
00038         itSystem getSystemBegin();
00039         itSystem getSystemEnd();
00040         int getSystemSize();
00041
00042         void add(System*);
00043         void add(Flow*);
00044         bool remove(System*);
00045         bool remove(Flow*);
00046         void clear();
00047         void show();
00048         void run(int, int, int);
00049 };
00050
00051 #endif
```

6.10 src/system.cpp File Reference

```
#include "system.h"
```

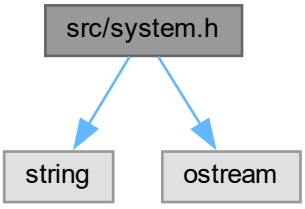
Include dependency graph for system.cpp:



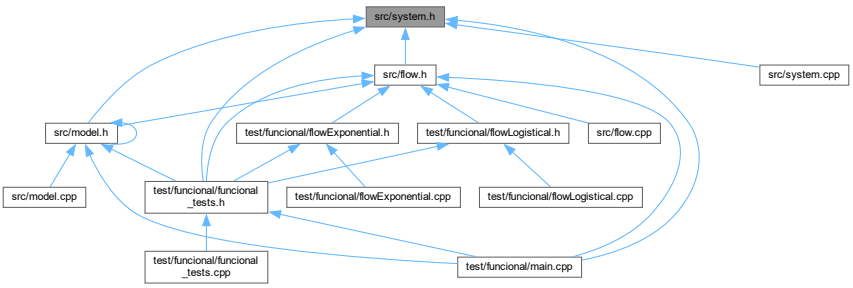
6.11 src/system.h File Reference

```
#include <string>
#include <ostream>
```

Include dependency graph for system.h:



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



Classes

- class System

6.12 system.h

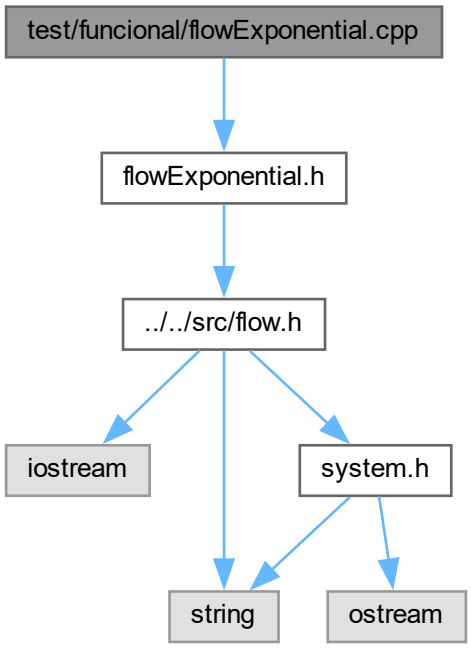
Go to the documentation of this file.

```
00001 #ifndef SYSTEM_H
00002 #define SYSTEM_H
00003
00004 #include <string>
00005 #include <ostream>
00006
00007 using namespace std;
00008
00009 class System {
00010     protected:
00011         string name;
00012         float value;
00013
00014     public:
00015         System();
00016         System(const string name);
00017         System(float value);
00018         System(System& obj);
00019         System(const string name, float value);
00020         virtual ~System();
00021
00022         string getName() const;
00023         void setName(const string name);
00024         float getValue() const;
00025         void setValue(float value);
00026
00027         System& operator= (const System& obj);
00028 };
00029
00030 #endif
```

6.13 test/funcional/flowExponential.cpp File Reference

```
#include "flowExponential.h"
```

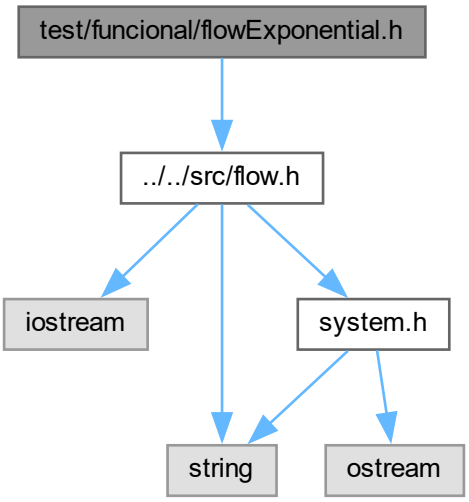
Include dependency graph for flowExponential.cpp:



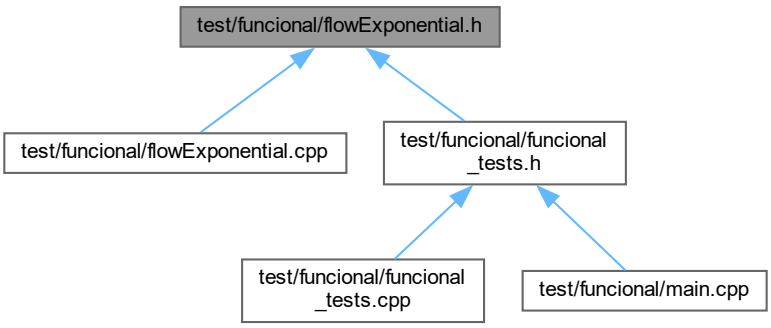
6.14 test/funcional/flowExponential.h File Reference

```
#include "../../src/flow.h"
```

Include dependency graph for flowExponential.h:



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



Classes

- class FlowExponential

6.15 flowExponential.h

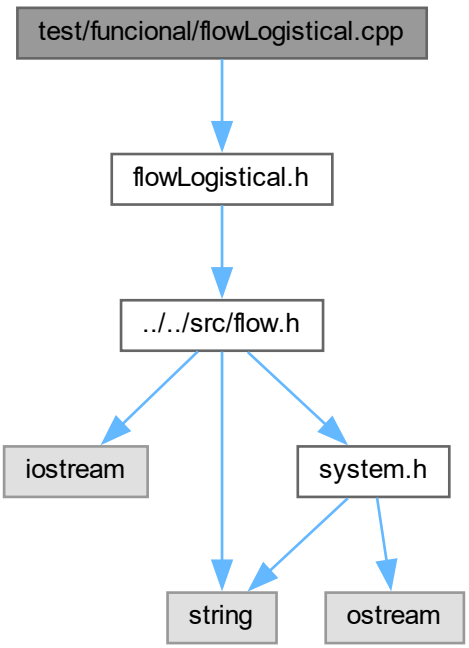
Go to the documentation of this file.

```
00001 #ifndef FLOWEXPONENTIAL_H
00002 #define FLOWEXPONENTIAL_H
00003
00004 #include "../../src/flow.h"
00005
00006 class FlowExponential : public Flow {
00007     public:
00008         FlowExponential();
00009         FlowExponential(Flow &obj);
00010         FlowExponential(const string name, System *source, System *target);
00011         virtual ~FlowExponential();
00012
00013         virtual float execute();
00014 };
00015
00016 #endif
```

6.16 test/funcional/flowLogistical.cpp File Reference

```
#include "flowLogistical.h"
```

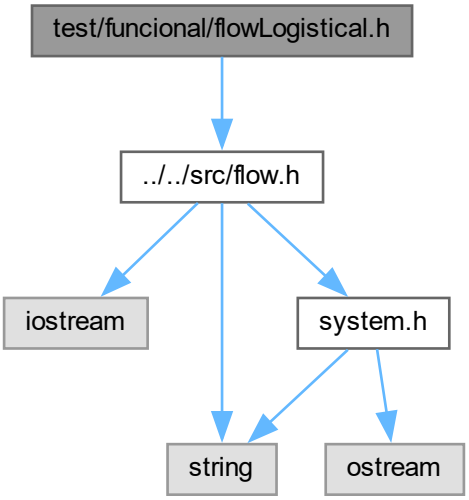
Include dependency graph for flowLogistical.cpp:



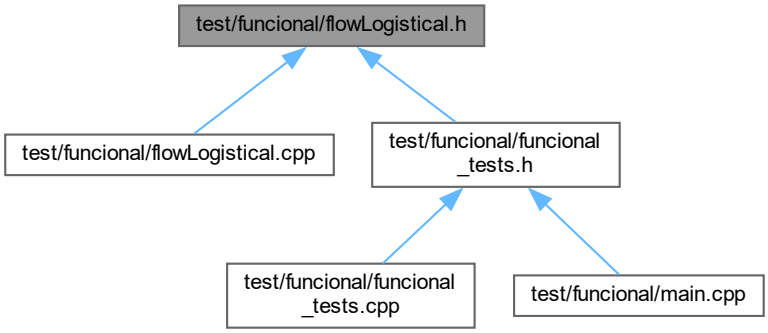
6.17 test/funcional/flowLogistical.h File Reference

```
#include "../../src/flow.h"
```

Include dependency graph for flowLogistical.h:



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



Classes

- class FlowLogistical

Macros

- #define FLOWLOGISTIC_H

6.17.1 Macro Definition Documentation

6.17.1.1 FLOWLOGISTIC_H

```
#define FLOWLOGISTIC_H
```

6.18 flowLogistical.h

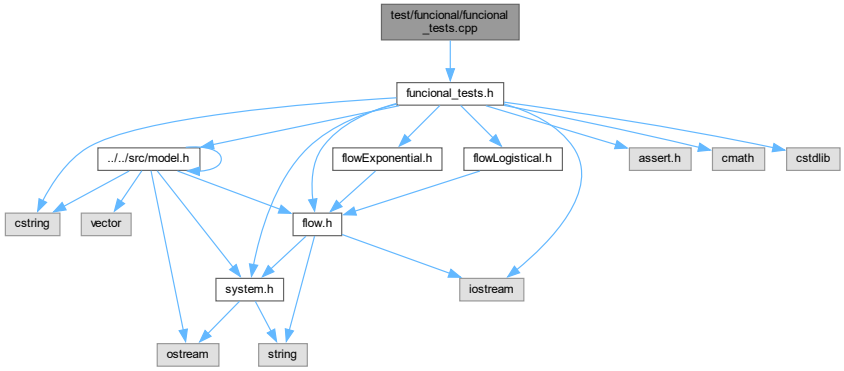
Go to the documentation of this file.

```
00001 #ifndef FLOWLOGISTICAL_H
00002 #define FLOWLOGISTIC_H
00003
00004 #include "../../src/flow.h"
00005
00006 class FlowLogistical : public Flow {
00007 public:
00008     FlowLogistical();
00009     FlowLogistical(Flow &obj);
00010     FlowLogistical(const string name, System *source, System *target);
00011     virtual ~FlowLogistical();
00012
00013     virtual float execute();
00014 };
00015
00016 #endif
```

6.19 test/funcional/funcional_tests.cpp File Reference

```
#include "funcional_tests.h"
```

Include dependency graph for funcional_tests.cpp:



Functions

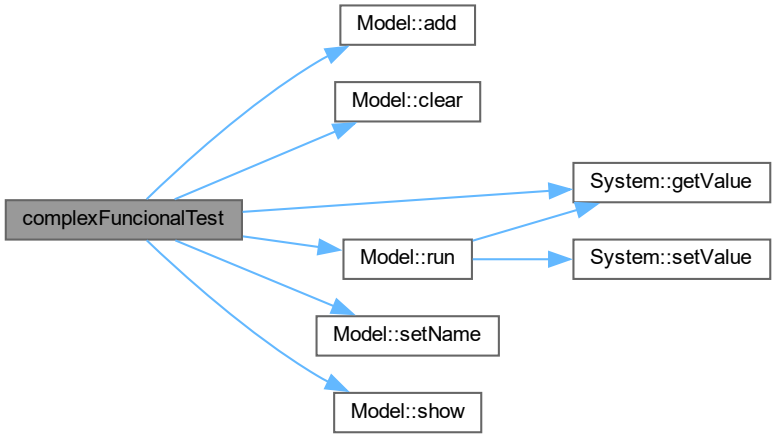
- void exponentialFuncionalTest ()
- void logisticalFuncionalTest ()
- void complexFuncionalTest ()

6.19.1 Function Documentation

6.19.1.1 complexFuncionalTest()

```
void complexFuncionalTest ( )
```

Here is the call graph for this function:

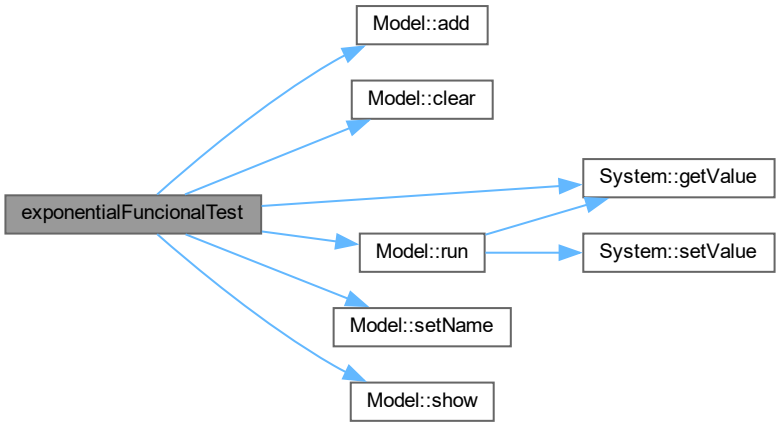


Here is the caller graph for this function:

6.19.1.2 exponentialFuncionalTest()

```
void exponentialFuncionalTest ( )
```

Here is the call graph for this function:

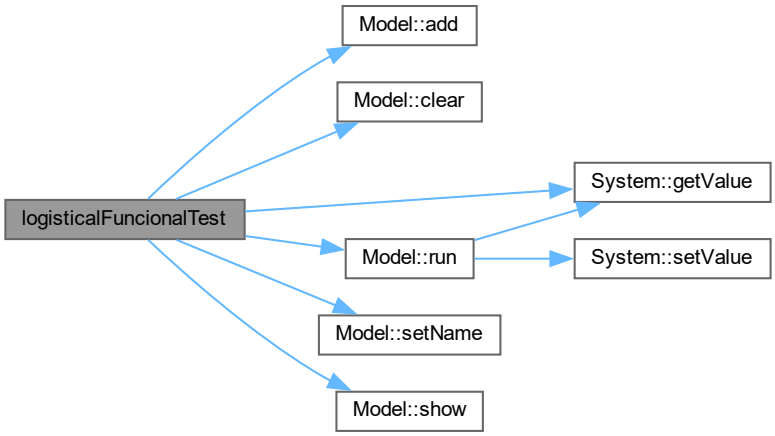


Here is the caller graph for this function:

6.19.1.3 logisticalFuncionalTest()

```
void logisticalFuncionalTest ( )
```

Here is the call graph for this function:

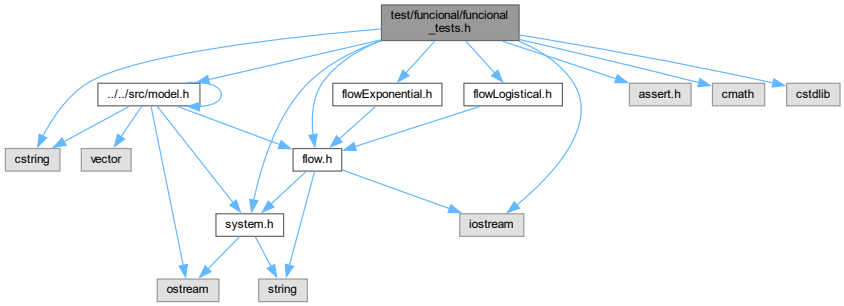


Here is the caller graph for this function:

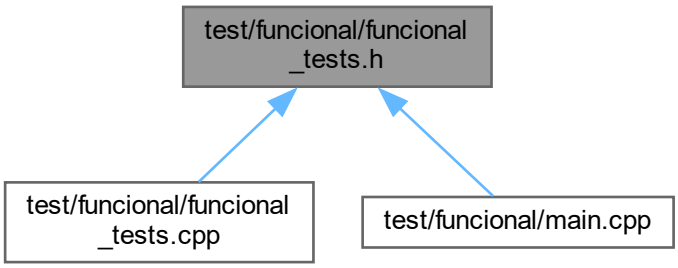
6.20 test/funcional/funcional_tests.h File Reference

```
#include "../../src/model.h"
#include "../../src/system.h"
#include "../../src/flow.h"
#include "flowExponential.h"
#include "flowLogistical.h"
#include <assert.h>
#include <cmath>
#include <iostream>
#include <cstdlib>
#include <cstring>
```

Include dependency graph for funcional_tests.h:



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



Functions

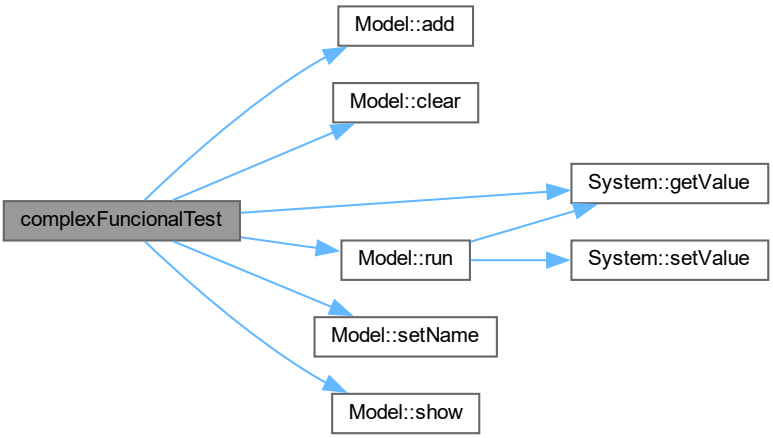
- void exponentialFuncionalTest ()
- void logisticalFuncionalTest ()
- void complexFuncionalTest ()

6.20.1 Function Documentation

6.20.1.1 complexFuncionalTest()

```
void complexFuncionalTest ( )
```

Here is the call graph for this function:

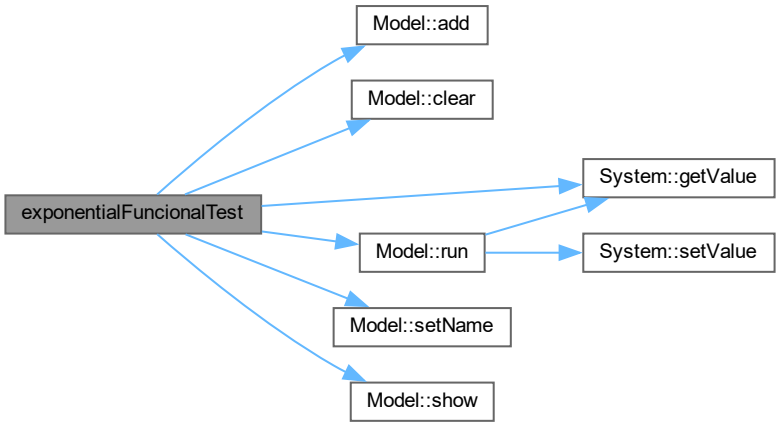


Here is the caller graph for this function:

6.20.1.2 exponentialFuncionalTest()

```
void exponentialFuncionalTest ( )
```

Here is the call graph for this function:

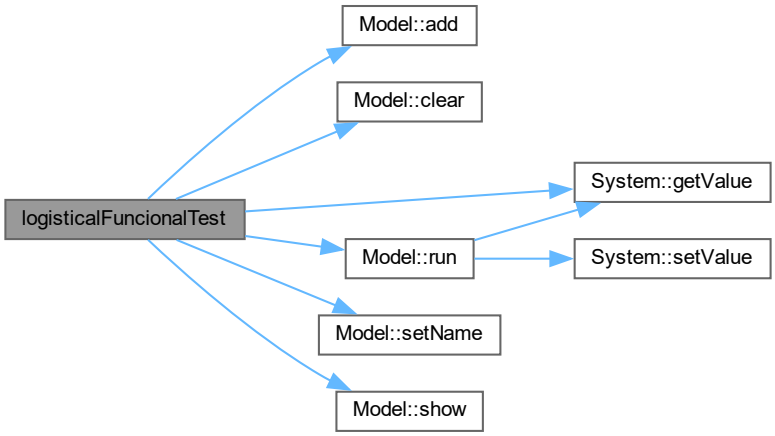


Here is the caller graph for this function:

6.20.1.3 logisticalFuncionalTest()

```
void logisticalFuncionalTest ( )
```

Here is the call graph for this function:



Here is the caller graph for this function:

6.21 funcional_tests.h

Go to the documentation of this file.

```
00001 #include "../../src/model.h"
00002 #include "../../src/system.h"
00003 #include "../../src/flow.h"
00004 #include "flowExponential.h"
00005 #include "flowLogistical.h"
00006
00007 #include <assert.h>
00008 #include <cmath>
00009 #include <iostream>
00010 #include <cstdlib>
00011 #include <cstring>
00012
00013 #ifndef FUNCIONAL_TESTS
00014 #define FUNCIONAL_TESTS
00015
00016 void exponentialFuncionalTest();
00017 void logisticalFuncionalTest();
00018 void complexFuncionalTest();
00019
00020 #endif
```

Index