

CircuitSim

Gerado por Doxygen 1.9.1



<b>1 Descrição</b>	<b>1</b>
<b>2 &lt;em&gt;Namespaces&lt;/em&gt;</b>	<b>3</b>
2.1 Lista de Namespaces	3
<b>3 Índice Hierárquico</b>	<b>5</b>
3.1 Hierarquia de Classes	5
<b>4 Índice dos Componentes</b>	<b>7</b>
4.1 Lista de Classes	7
<b>5 Índice dos Arquivos</b>	<b>9</b>
5.1 Lista de Arquivos	9
<b>6 &lt;em&gt;Namespace&lt;/em&gt;</b>	<b>11</b>
6.1 Referência do Namespace CCT	11
6.2 Referência do Namespace CMP	11
6.2.1 Enumerações	11
6.2.1.1 type	11
6.3 Referência do Namespace GRF	12
6.4 Referência do Namespace NM	12
<b>7 Classes</b>	<b>13</b>
7.1 Referência da Classe GRF::adjacencyMatrix	13
7.1.1 Construtores e Destrutores	13
7.1.1.1 adjacencyMatrix() [1/2]	13
7.1.1.2 adjacencyMatrix() [2/2]	14
7.1.2 Funções membros	14
7.1.2.1 getVertexNumber()	14
7.1.2.2 insertEdge()	14
7.1.2.3 print()	14
7.1.2.4 query()	14
7.1.2.5 removeVertex()	14
7.1.3 Atributos	14
7.1.3.1 adjMatrix	15
7.1.3.2 vertexNumber	15
7.2 Referência da Classe CCT::Circuit	15
7.2.1 Descrição detalhada	16
7.2.2 Construtores e Destrutores	16
7.2.2.1 Circuit()	16
7.2.2.2 ~Circuit()	16
7.2.3 Funções membros	17
7.2.3.1 addComponent()	17
7.2.3.2 editComponent() [1/2]	17

7.2.3.3 editComponent() [2/2]	18
7.2.3.4 getCurrent()	18
7.2.3.5 getVoltage()	19
7.2.3.6 initialize()	19
7.2.3.7 removeComponent()	19
7.2.3.8 reset()	20
7.2.3.9 Solve()	20
7.2.3.10 updateComponents()	20
7.2.4 Atributos	20
7.2.4.1 chords	21
7.2.4.2 circuitMatrix	21
7.2.4.3 components	21
7.3 Referência da Classe CMP::Component	21
7.3.1 Construtores e Destrutores	22
7.3.1.1 Component()	22
7.3.1.2 ~Component()	22
7.3.2 Funções membros	22
7.3.2.1 getCurrent()	22
7.3.2.2 getLabel()	22
7.3.2.3 getNodes()	22
7.3.2.4 getType()	23
7.3.2.5 getVoltage()	23
7.3.2.6 setCurrent()	23
7.3.2.7 setLabel()	23
7.3.2.8 setVoltage()	23
7.3.3 Atributos	23
7.3.3.1 current	23
7.3.3.2 label	24
7.3.3.3 voltage	24
7.3.3.4 vtxs	24
7.4 Referência da Classe Diagram	24
7.4.1 Construtores e Destrutores	25
7.4.1.1 Diagram()	26
7.4.2 Funções membros	26
7.4.2.1 clickedControl()	26
7.4.2.2 edit	26
7.4.2.3 editMode	26
7.4.2.4 freeAllocatedMemory()	26
7.4.2.5 getFileName()	26
7.4.2.6 getPixmap()	27
7.4.2.7 getStatus()	27
7.4.2.8 initializeDiagram()	27

7.4.2.9 insert()	27
7.4.2.10 leftButtonActions()	27
7.4.2.11 leftButtonClicked	27
7.4.2.12 load()	27
7.4.2.13 loadError	28
7.4.2.14 modified	28
7.4.2.15 mouseMoveEvent()	28
7.4.2.16 mousePressEvent()	28
7.4.2.17 paintEvent()	28
7.4.2.18 query	28
7.4.2.19 queryMode	28
7.4.2.20 remove	29
7.4.2.21 rightButtonActions()	29
7.4.2.22 rightButtonClicked	29
7.4.2.23 save()	29
7.4.2.24 setFileName()	29
7.4.2.25 setSelectedButton()	29
7.4.2.26 setStatus()	29
7.4.2.27 showEditDialog	30
7.4.3 Atributos	30
7.4.3.1 circuit	30
7.4.3.2 clickedStack	30
7.4.3.3 connections	30
7.4.3.4 cursorLocation	30
7.4.3.5 drawList	30
7.4.3.6 editButton	30
7.4.3.7 editMenu	31
7.4.3.8 fileName	31
7.4.3.9 mode	31
7.4.3.10 playButton	31
7.4.3.11 selectedButton	31
7.4.3.12 selectedComponent	31
7.4.3.13 selectedPrev	31
7.4.3.14 status	31
7.4.3.15 wireCounter	32
7.5 Referência da Classe NM::EquationSystem	32
7.5.1 Construtores e Destrutores	32
7.5.1.1 EquationSystem()	32
7.5.2 Funções membros	33
7.5.2.1 findPivot()	33
7.5.2.2 gaussJordan()	33
7.5.2.3 gaussSeidel()	33

7.5.2.4	<code>getSolution()</code>	33
7.5.2.5	<code>print()</code>	33
7.5.2.6	<code>printSolution()</code>	33
7.5.2.7	<code>sassenfeldCriteria()</code>	34
7.5.3	Atributos	34
7.5.3.1	A	34
7.5.3.2	B	34
7.5.3.3	x	34
7.6	Referência da Classe <code>GraphicComponent</code>	34
7.6.1	Construtores e Destrutores	36
7.6.1.1	<code>GraphicComponent()</code> [1/3]	36
7.6.1.2	<code>GraphicComponent()</code> [2/3]	36
7.6.1.3	<code>GraphicComponent()</code> [3/3]	36
7.6.2	Funções membros	36
7.6.2.1	<code>clicked</code>	36
7.6.2.2	<code>clickedArea()</code>	37
7.6.2.3	<code>clickedVertex</code>	37
7.6.2.4	<code>draw()</code>	37
7.6.2.5	<code>getBottom()</code>	37
7.6.2.6	<code>getBoundRect()</code>	37
7.6.2.7	<code>getHeight()</code>	37
7.6.2.8	<code>getLabel()</code>	37
7.6.2.9	<code>getLeft()</code>	38
7.6.2.10	<code>getMap()</code>	38
7.6.2.11	<code>getOrientation()</code>	38
7.6.2.12	<code>getRight()</code>	38
7.6.2.13	<code>getTop()</code>	38
7.6.2.14	<code>getType()</code>	38
7.6.2.15	<code>getValue()</code>	38
7.6.2.16	<code>getVertex1()</code>	39
7.6.2.17	<code>getVertex1Point()</code>	39
7.6.2.18	<code>getVertex2()</code>	39
7.6.2.19	<code>getVertex2Point()</code>	39
7.6.2.20	<code>getVertexNum()</code>	39
7.6.2.21	<code>getWidth()</code>	39
7.6.2.22	<code>setValue()</code>	39
7.6.2.23	<code>setVertex1()</code>	40
7.6.2.24	<code>setVertex2()</code>	40
7.6.3	Atributos	40
7.6.3.1	<code>boundRect</code>	40
7.6.3.2	<code>componentType</code>	40
7.6.3.3	<code>label</code>	40

7.6.3.4 map	40
7.6.3.5 orientation	40
7.6.3.6 value	41
7.6.3.7 vertex1	41
7.6.3.8 vertex2	41
7.6.3.9 vertexArea1	41
7.6.3.10 vertexArea2	41
7.6.3.11 vertexNum	41
7.6.3.12 x	41
7.6.3.13 y	42
7.7 Referência da Classe GRF::incidenceMatrix	42
7.7.1 Construtores e Destrutores	43
7.7.1.1 incidenceMatrix() [1/2]	43
7.7.1.2 incidenceMatrix() [2/2]	43
7.7.2 Funções membros	43
7.7.2.1 addEdge()	43
7.7.2.2 getConNum()	43
7.7.2.3 getEdgeNumber()	43
7.7.2.4 getEdges() [1/2]	43
7.7.2.5 getEdges() [2/2]	44
7.7.2.6 getLoop()	44
7.7.2.7 getSpanningTree()	44
7.7.2.8 getVertex()	44
7.7.2.9 getVertexCon()	44
7.7.2.10 getVertexNumber()	44
7.7.2.11 makeCon()	44
7.7.2.12 operator[]()	45
7.7.2.13 print()	45
7.7.2.14 removeEdge()	45
7.7.2.15 removeVertex()	45
7.7.3 Atributos	45
7.7.3.1 edgeNumber	45
7.7.3.2 inMatrix	45
7.7.3.3 vertexNumber	46
7.8 Referência da Classe MainWindow	46
7.8.1 Construtores e Destrutores	47
7.8.1.1 MainWindow()	47
7.8.2 Funções membros	47
7.8.2.1 closeFile	47
7.8.2.2 drawRes180	47
7.8.2.3 drawRes90	48
7.8.2.4 drawVcc180	48

7.8.2.5 drawVcc90 . . . . .	48
7.8.2.6 initializeMenu() . . . . .	48
7.8.2.7 initializeStatusBar() . . . . .	48
7.8.2.8 initializeTabs() . . . . .	48
7.8.2.9 initializeToolBar() . . . . .	48
7.8.2.10 newFile . . . . .	48
7.8.2.11 openFile [1/2] . . . . .	49
7.8.2.12 openFile() [2/2] . . . . .	49
7.8.2.13 preferences . . . . .	49
7.8.2.14 saveFile . . . . .	49
7.8.2.15 saveFileAs . . . . .	49
7.8.2.16 setTabStatus . . . . .	49
7.8.3 Atributos . . . . .	49
7.8.3.1 diagrams . . . . .	49
7.8.3.2 fileMenu . . . . .	50
7.8.3.3 helpMenu . . . . .	50
7.8.3.4 initialTab . . . . .	50
7.8.3.5 mainBar . . . . .	50
7.8.3.6 newFileAct . . . . .	50
7.8.3.7 openFileAct . . . . .	50
7.8.3.8 prefAct . . . . .	50
7.8.3.9 prefMenu . . . . .	50
7.8.3.10 saveFileAct . . . . .	51
7.8.3.11 saveFileAsAct . . . . .	51
7.8.3.12 statusBar . . . . .	51
7.8.3.13 tabs . . . . .	51
7.8.3.14 toolbar . . . . .	51
7.8.3.15 tutorialAct . . . . .	51
7.9 Referência da Classe NM::Matrix . . . . .	51
7.9.1 Construtores e Destrutores . . . . .	52
7.9.1.1 Matrix() [1/2] . . . . .	52
7.9.1.2 Matrix() [2/2] . . . . .	52
7.9.2 Funções membros . . . . .	52
7.9.2.1 Abs() . . . . .	53
7.9.2.2 getCol() . . . . .	53
7.9.2.3 getColNumber() . . . . .	53
7.9.2.4 getRowNumber() . . . . .	53
7.9.2.5 operator*() . . . . .	53
7.9.2.6 operator*=( ) . . . . .	53
7.9.2.7 operator+( ) . . . . .	53
7.9.2.8 operator+=( ) . . . . .	54
7.9.2.9 operator-( ) [1/2] . . . . .	54



7.9.2.10 operator-() [2/2]	54
7.9.2.11 operator-=( )	54
7.9.2.12 operator=( )	54
7.9.2.13 operator[]()	54
7.9.2.14 print() [1/2]	54
7.9.2.15 print() [2/2]	55
7.9.2.16 swapLines()	55
7.9.2.17 transpose()	55
7.9.3 Atributos	55
7.9.3.1 colNumber	55
7.9.3.2 realMatrix	55
7.9.3.3 rowNumber	55
7.10 Referência da Classe CMP::Resistor	56
7.10.1 Construtores e Destrutores	56
7.10.1.1 Resistor()	56
7.10.1.2 ~Resistor()	56
7.10.2 Funções membros	57
7.10.2.1 getResistance()	57
7.10.2.2 getType()	57
7.10.2.3 setCurrent()	57
7.10.2.4 setResistance()	57
7.10.2.5 setVoltage()	57
7.10.3 Atributos	57
7.10.3.1 resistance	58
7.11 Referência da Classe Resistor	58
7.11.1 Construtores e Destrutores	58
7.11.1.1 Resistor()	58
7.11.2 Funções membros	59
7.11.2.1 getType()	59
7.11.3 Atributos	59
7.11.3.1 resCounter	59
7.12 Referência da Classe CMP::Vcc	59
7.12.1 Construtores e Destrutores	60
7.12.1.1 Vcc()	60
7.12.1.2 ~Vcc()	60
7.12.2 Funções membros	60
7.12.2.1 getType()	60
7.12.2.2 setCurrent()	60
7.12.2.3 setVoltage()	61
7.13 Referência da Classe Vcc	61
7.13.1 Construtores e Destrutores	61
7.13.1.1 Vcc()	61

7.13.2 Funções membros	62
7.13.2.1 getType()	62
7.13.3 Atributos	62
7.13.3.1 vccCounter	62
<b>8 Arquivos</b>	<b>63</b>
8.1 Referência do Arquivo Circuit.cpp	63
8.1.1 Descrição detalhada	63
8.2 Referência do Arquivo Circuit.h	64
8.2.1 Descrição detalhada	64
8.3 Referência do Arquivo Component.cpp	64
8.3.1 Descrição detalhada	65
8.4 Referência do Arquivo Component.h	65
8.4.1 Descrição detalhada	65
8.5 Referência do Arquivo Diagram.cpp	66
8.5.1 Descrição detalhada	66
8.6 Referência do Arquivo Diagram.h	66
8.6.1 Descrição detalhada	67
8.6.2 Enumerações	67
8.6.2.1 mode	67
8.6.2.2 sts	67
8.6.2.3 typeOrientation	68
8.7 Referência do Arquivo Graph.cpp	68
8.7.1 Descrição detalhada	68
8.8 Referência do Arquivo Graph.h	69
8.8.1 Descrição detalhada	69
8.9 Referência do Arquivo GraphicComponent.cpp	69
8.9.1 Descrição detalhada	70
8.10 Referência do Arquivo GraphicComponent.h	70
8.10.1 Descrição detalhada	71
8.10.2 Definições e macros	71
8.10.2.1 HEIGHT	71
8.10.2.2 WIDTH	71
8.10.3 Enumerações	71
8.10.3.1 orien	71
8.11 Referência do Arquivo main.cpp	72
8.11.1 Funções	72
8.11.1.1 main()	72
8.12 Referência do Arquivo MainWindow.cpp	72
8.12.1 Descrição detalhada	73
8.13 Referência do Arquivo MainWindow.h	73
8.13.1 Descrição detalhada	73

---

8.14 Referência do Arquivo Numeric.cpp . . . . .	74
8.14.1 Descrição detalhada . . . . .	74
8.15 Referência do Arquivo Numeric.h . . . . .	74
8.15.1 Descrição detalhada . . . . .	75
<b>Índice Remissivo</b>	<b>77</b>



# Capítulo 1

## Descrição

Este projeto consiste em um simulador de circuitos compostos por resistores e fontes de tensão contínua com interface gráfica.



## Capítulo 2

# <em>Namespaces</em>

### 2.1 Lista de Namespaces

Esta é a lista de todos os Namespaces com suas respectivas descrições:

CCT	.....	11
CMP	.....	11
GRF	.....	12
NM	.....	12





## Capítulo 3

# Índice Hierárquico

### 3.1 Hierarquia de Classes

Esta lista de hierarquias está parcialmente ordenada (ordem alfabética):

GRF::adjacencyMatrix . . . . .	13
CMP::Component . . . . .	21
CMP::Resistor . . . . .	56
CMP::Vcc . . . . .	59
NM::EquationSystem . . . . .	32
GRF::incidenceMatrix . . . . .	42
CCT::Circuit . . . . .	15
NM::Matrix . . . . .	51
QMainWindow	
MainWindow . . . . .	46
QObject	
GraphicComponent . . . . .	34
Resistor . . . . .	58
Vcc . . . . .	61
QWidget	
Diagram . . . . .	24



## Capítulo 4

# Índice dos Componentes

### 4.1 Lista de Classes

Aqui estão as classes, estruturas, uniões e interfaces e suas respectivas descrições:

GRF::adjacencyMatrix	13
CCT::Circuit	
Implementação da classe <a href="#">Circuit</a>	15
CMP::Component	21
Diagram	24
NM::EquationSystem	32
GraphicComponent	34
GRF::incidenceMatrix	42
MainWindow	46
NM::Matrix	51
CMP::Resistor	56
Resistor	58
CMP::Vcc	59
Vcc	61



## Capítulo 5

# Índice dos Arquivos

### 5.1 Lista de Arquivos

Esta é a lista de todos os arquivos e suas respectivas descrições:

<a href="#">Circuit.cpp</a>	
Implementação da classe <a href="#">Circuit</a> . . . . .	63
<a href="#">Circuit.h</a>	
Implementação da classe <a href="#">Circuit</a> . . . . .	64
<a href="#">Component.cpp</a>	
Implementação da classe <a href="#">Component</a> . . . . .	64
<a href="#">Component.h</a>	
Declaração da classe <a href="#">Component</a> . . . . .	65
<a href="#">Diagram.cpp</a>	
Declaração da classe <a href="#">Numeric</a> . . . . .	66
<a href="#">Diagram.h</a>	
Declaração da classe <a href="#">Numeric</a> . . . . .	66
<a href="#">Graph.cpp</a>	
Implementação das classes de grafo utilizadas no projeto . . . . .	68
<a href="#">Graph.h</a>	
Declaração das classes de grafo utilizadas no projeto . . . . .	69
<a href="#">GraphicComponent.cpp</a>	
Implementação da classe <a href="#">GraphicComponent</a> para a inserção gráfica de componentes . . . . .	69
<a href="#">GraphicComponent.h</a>	
Declaração da classe <a href="#">GraphicComponent</a> para a inserção gráfica de componentes . . . . .	70
<a href="#">main.cpp</a> . . . . .	72
<a href="#">MainWindow.cpp</a>	
Implementação da classe <a href="#">MainWindow</a> . . . . .	72
<a href="#">MainWindow.h</a>	
Declaração da classe <a href="#">MainWindow</a> . . . . .	73
<a href="#">Numeric.cpp</a>	
Declaração da classe <a href="#">Numeric</a> . . . . .	74
<a href="#">Numeric.h</a>	
Implementação da classe <a href="#">Numeric</a> . . . . .	74



## Capítulo 6

# <em>Namespace</em>

### 6.1 Refência do Namespace CCT

#### Componentes

- class [Circuit](#)  
*Implementação da classe [Circuit](#).*

### 6.2 Refência do Namespace CMP

#### Componentes

- class [Component](#)
- class [Resistor](#)
- class [Vcc](#)

#### Enumerações

- enum [type](#) { [RESISTOR](#) , [VCC](#) , [WIRE](#) , [GROUND](#) }

#### 6.2.1 Enumerações

##### 6.2.1.1 type

enum [CMP::type](#)

#### Enumeradores

RESISTOR	
VCC	
WIRE	
GROUND	

## 6.3 Referência do Namespace GRF

### Componentes

- class [incidenceMatrix](#)
- class [adjacencyMatrix](#)

## 6.4 Referência do Namespace NM

### Componentes

- class [Matrix](#)
- class [EquationSystem](#)



## Capítulo 7

# Classes

### 7.1 Referência da Classe GRF::adjacencyMatrix

```
#include <Graph.h>
```

#### Membros Públicos

- [adjacencyMatrix](#) ()
- [adjacencyMatrix](#) (unsigned int vertexNum)
- void [insertEdge](#) (unsigned int i, unsigned int j)
- void [removeVertex](#) (unsigned int i)
- int [query](#) (unsigned int i, unsigned int j)
- unsigned int [getVertexNumber](#) ()
- void [print](#) ()

#### Atributos Privados

- unsigned int [vertexNumber](#)
- std::vector< std::vector< int > > [adjMatrix](#)

#### 7.1.1 Construtores e Destrutores

##### 7.1.1.1 adjacencyMatrix() [1/2]

```
GRF::adjacencyMatrix::adjacencyMatrix ( )
```

### 7.1.1.2 adjacencyMatrix() [2/2]

```
GRF::adjacencyMatrix::adjacencyMatrix (
    unsigned int vertexNum )
```

## 7.1.2 Funções membros

### 7.1.2.1 getVertexNumber()

```
unsigned int GRF::adjacencyMatrix::getVertexNumber ( )
```

### 7.1.2.2 insertEdge()

```
void GRF::adjacencyMatrix::insertEdge (
    unsigned int i,
    unsigned int j )
```

### 7.1.2.3 print()

```
void GRF::adjacencyMatrix::print ( ) [inline]
```

### 7.1.2.4 query()

```
int GRF::adjacencyMatrix::query (
    unsigned int i,
    unsigned int j )
```

### 7.1.2.5 removeVertex()

```
void GRF::adjacencyMatrix::removeVertex (
    unsigned int i )
```

## 7.1.3 Atributos

### 7.1.3.1 adjMatrix

```
std::vector<std::vector<int> > GRF::adjacencyMatrix::adjMatrix [private]
```

### 7.1.3.2 vertexNumber

```
unsigned int GRF::adjacencyMatrix::vertexNumber [private]
```

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

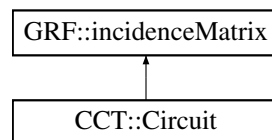
- [Graph.h](#)
- [Graph.cpp](#)

## 7.2 Referência da Classe CCT::Circuit

Implementação da classe [Circuit](#).

```
#include <Circuit.h>
```

Diagrama de hierarquia para CCT::Circuit:



### Membros Públicos

- [Circuit](#) ()  
*Construtor para a classe Circuito.*
- [~Circuit](#) ()  
*Desconstrutor para a classe Circuito.*
- void [initialize](#) ()  
*Obtém os circuitos fundamentais do grafo de circuito e armazena em circuitMatrix, bem como, atualiza o vector chords contendo as arestas que não estão contidas na árvore geradora do grafo e após isso, resolve o circuito e atualiza os valores de cada componente.*
- void [reset](#) ()  
*Reseta a matriz de circuitos fundamentais contidas no vector circuitMatrix e o vector chords de arestas que não estão contidas na árvore geradora.*
- void [addComponent](#) (CMP::type t, std::string l, double value, unsigned int vtx1, unsigned int vtx2)  
*Adiciona um objeto da classe Component no circuito.*
- void [editComponent](#) (std::string label, double value)  
*Edita o valor de um objeto da classe Component pertencente ao circuito.*
- void [editComponent](#) (std::string label, std::string newLabel)  
*Edita a etiqueta de um objeto da classe Component pertencente ao circuito.*
- void [removeComponent](#) (std::string l)  
*Remove um objeto da classe Component pertencente ao circuito.*
- double [getVoltage](#) (std::string l)  
*Retorna a tensão através de um componente.*
- double [getCurrent](#) (std::string l)  
*Retorna a corrente através de um componente.*

## Membros Privados

- void `updateComponents` (std::vector< double > currents)  
*Atualiza os valores dos componentes.*
- void `Solve` ()  
*Obtém os valores das correntes em cada componente e atualiza cada componente.*

## Atributos Privados

- std::vector< `CMP::Component` \* > `components`  
*Armazena os componentes inseridos.*
- std::vector< std::vector< int > > `circuitMatrix`  
*Armazena os circuitos fundamentais do grafo.*
- std::vector< unsigned int > `chords`  
*Armazena as arestas que não estão na árvore geradora do grafo.*

## Outros membros herdados

### 7.2.1 Descrição detalhada

Implementação da classe `Circuit`.

Representa um circuito de componentes eletrônicos através de uma matriz de incidência.

### 7.2.2 Construtores e Destrutores

#### 7.2.2.1 `Circuit()`

```
CCT::Circuit::Circuit ( )
```

Construtor para a classe `Circuito`.

Implementação dos métodos da classe `Circuit`.

Constroi um elemento da Classe `Circuit`.

Representa um circuito de componentes eletrônicos através de uma matriz de incidência.

#### 7.2.2.2 `~Circuit()`

```
CCT::Circuit::~~Circuit ( )
```

Desconstrutor para a classe `Circuito`.

Destrói um elemento da Classe `Circuit`.

### 7.2.3 Funções membros

#### 7.2.3.1 addComponent()

```
void CCT::Circuit::addComponent (
    CMP::type t,
    std::string l,
    double value,
    unsigned int vtx1,
    unsigned int vtx2 )
```

Adiciona um objeto da classe Component no circuito.

Os vértices do componente representam nós com que outros componentes podem ser adicionados, isto é, se dois componentes estão conectados, então, estes componentes possuem um vértice em comum.

##### Parâmetros

<i>t</i>	tipo do componente a ser adicionado no circuito.
<i>l</i>	etiqueta do componente a ser adicionado.
<i>value</i>	valor associado ao componente a ser adicionado (resistência/tensão).
<i>vtx1</i>	número do primeiro vértice do componente.
<i>vtx2</i>	número do segundo vértice do componente.

##### Retorna

void

#### 7.2.3.2 editComponent() [1/2]

```
void CCT::Circuit::editComponent (
    std::string label,
    double value )
```

Edita o valor de um objeto da classe Component pertencente ao circuito.

Caso o componente esteja no circuito, altera o valor relacionado ao componente (resistência/tensão) cuja etiqueta é label.

##### Parâmetros

<i>value</i>	novo valor do componente a ser editado.
<i>label</i>	etiqueta do componente a ser editado.

**Retorna**

void

**7.2.3.3 editComponent() [2/2]**

```
void CCT::Circuit::editComponent (
    std::string label,
    std::string newLabel )
```

Edita a etiqueta de um objeto da classe Component pertencente ao circuito.

Caso o componente esteja no circuito, altera a etiqueta do componente a ser editado.

**Parâmetros**

<i>label</i>	etiqueta do componente a ser editado.
<i>newLabel</i>	nova etiqueta do componente a ser editado.

**Retorna**

void

Esse é um método provido por conveniência. Ele difere do método acima apenas na lista de argumentos que devem ser utilizados.

**7.2.3.4 getCurrent()**

```
double CCT::Circuit::getCurrent (
    std::string l )
```

Retorna a corrente através de um componente.

Caso o componente exista no circuito, retorna o valor da corrente através do componente com a etiqueta fornecida.

**Parâmetros**

<i>l</i>	etiqueta do componente consultado.
----------	------------------------------------

**Retorna**

valor da corrente através do componente consultado.

### 7.2.3.5 getVoltage()

```
double CCT::Circuit::getVoltage (
    std::string l )
```

Retorna a tensão através de um componente.

Caso o componente exista no circuito, retorna o valor da tensão do componente com a etiqueta fornecida.

#### Parâmetros

/	etiqueta do componente consultado.
---	------------------------------------

#### Retorna

valor da tensão através do componente consultado.

### 7.2.3.6 initialize()

```
void CCT::Circuit::initialize ( )
```

Obtém os circuitos fundamentais do grafo de circuito e armazena em circuitMAtrix, bem como, atualiza o vector chords contendo as arestas que não estão contidas na árvore geradora do grafo e após isso, resolve o circuito e atualiza os valores de cada componente.

#### Retorna

void

### 7.2.3.7 removeComponent()

```
void CCT::Circuit::removeComponent (
    std::string l )
```

Remove um objeto da classe Component pertencente ao circuito.

Caso o componente exista no circuito, remove do circuito o componente com a etiqueta fornecida.

#### Parâmetros

/	etiqueta do componente a ser removido.
---	--

#### Retorna

void

#### 7.2.3.8 reset()

```
void CCT::Circuit::reset ( )
```

Reseta a matriz de circuitos fundamentais contidas no vector `circuitMatrix` e o vector `chords` de arestas que não estão contidas na árvore geradora.

**Retorna**

void

#### 7.2.3.9 Solve()

```
void CCT::Circuit::Solve ( ) [private]
```

Obtém os valores das correntes em cada componente e atualiza cada componente.

**Retorna**

void

#### 7.2.3.10 updateComponents()

```
void CCT::Circuit::updateComponents (
    std::vector< double > currents ) [private]
```

Atualiza os valores dos componentes.

**Parâmetros**

<i>currents</i>	um vector contendo os novos valores para as correntes dos componentes
-----------------	---

**Retorna**

void

### 7.2.4 Atributos



#### 7.2.4.1 chords

```
std::vector<unsigned int> CCT::Circuit::chords [private]
```

Armazena as arestas que não estão na árvore geradora do grafo.

#### 7.2.4.2 circuitMatrix

```
std::vector<std::vector<int> > CCT::Circuit::circuitMatrix [private]
```

Armazena os circuitos fundamentais do grafo.

#### 7.2.4.3 components

```
std::vector<CMP::Component*> CCT::Circuit::components [private]
```

Armazena os componentes inseridos.

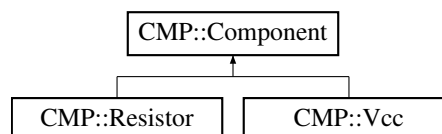
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

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

## 7.3 Referência da Classe CMP::Component

```
#include <Component.h>
```

Diagrama de hierarquia para CMP::Component:



### Membros Públicos

- [Component](#) (std::string l, unsigned int vtx1, unsigned int vtx2)
- virtual [~Component](#) ()
- void [setLabel](#) (std::string newLabel)
- std::string [getLabel](#) ()
- double [getVoltage](#) ()
- double [getCurrent](#) ()
- std::pair< unsigned int, unsigned int > [getNodes](#) ()
- virtual enum [type](#) [getType](#) ()=0
- virtual void [setCurrent](#) (double cur)=0
- virtual void [setVoltage](#) (double vol)=0

## Atributos Protegidos

- `std::string` `label`
- `std::pair< unsigned int, unsigned int >` `vtxs`
- `double` `voltage`
- `double` `current`

### 7.3.1 Construtores e Destrutores

#### 7.3.1.1 Component()

```
CMP::Component::Component (
    std::string l,
    unsigned int vtx1,
    unsigned int vtx2 )
```

#### 7.3.1.2 ~Component()

```
CMP::Component::~~Component ( ) [virtual]
```

### 7.3.2 Funções membros

#### 7.3.2.1 getCurrent()

```
double CMP::Component::getCurrent ( )
```

#### 7.3.2.2 getLabel()

```
std::string CMP::Component::getLabel ( )
```

#### 7.3.2.3 getNodes()

```
std::pair< unsigned int, unsigned int > CMP::Component::getNodes ( )
```

#### 7.3.2.4 getType()

```
virtual enum type CMP::Component::getType ( ) [pure virtual]
```

Implementado por [CMP::Vcc](#) e [CMP::Resistor](#).

#### 7.3.2.5 getVoltage()

```
double CMP::Component::getVoltage ( )
```

#### 7.3.2.6 setCurrent()

```
virtual void CMP::Component::setCurrent (
    double cur ) [pure virtual]
```

Implementado por [CMP::Vcc](#) e [CMP::Resistor](#).

#### 7.3.2.7 setLabel()

```
void CMP::Component::setLabel (
    std::string newLabel )
```

#### 7.3.2.8 setVoltage()

```
virtual void CMP::Component::setVoltage (
    double vol ) [pure virtual]
```

Implementado por [CMP::Vcc](#) e [CMP::Resistor](#).

### 7.3.3 Atributos

#### 7.3.3.1 current

```
double CMP::Component::current [protected]
```

### 7.3.3.2 label

```
std::string CMP::Component::label [protected]
```

### 7.3.3.3 voltage

```
double CMP::Component::voltage [protected]
```

### 7.3.3.4 vtxs

```
std::pair<unsigned int, unsigned int> CMP::Component::vtxs [protected]
```

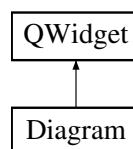
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

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

## 7.4 Referência da Classe Diagram

```
#include <Diagram.h>
```

Diagrama de hierarquia para Diagram:



### Slots Públicos

- void [queryMode](#) ()
- void [editMode](#) ()
- void [rightButtonClicked](#) (int x, int y, int cArea)
- void [leftButtonClicked](#) (int x, int y, int cArea)
- void [showEditDialog](#) ()
- void [edit](#) (double newValue)
- void [remove](#) ()
- void [query](#) ()

### Sinais

- void [modified](#) (bool checked=false)
- void [loadError](#) (bool checked=false)

## Membros Públicos

- [Diagram](#) (QWidget \*parent=nullptr)
- void [setFileName](#) (QString file)
- QString [getFileName](#) ()
- void [load](#) ()
- void [save](#) ()
- enum [sts](#) [getStatus](#) ()
- void [setSelectedButton](#) (enum [typeOrientation](#))
- void [insert](#) (int x, int y)

## Membros Protegidos

- void [mousePressEvent](#) (QMouseEvent \*event) override
- void [mouseMoveEvent](#) (QMouseEvent \*event) override

## Membros Privados

- void [initializeDiagram](#) ()
- void [setStatus](#) (enum [sts](#) newStatus)
- void [paintEvent](#) (QPaintEvent \*event) override
- void [freeAllocatedMemory](#) ()
- void [rightButtonActions](#) (int x, int y)
- void [leftButtonActions](#) (int x, int y)
- void [clickedControl](#) (int, int, int)
- std::pair< QRect, QPixmap > [getPixMap](#) (enum [typeOrientation](#) type)

## Atributos Privados

- QPushButton \* [playButton](#)
- QPushButton \* [editButton](#)
- std::string [fileName](#)
- enum [sts](#) [status](#)
- CCT::Circuit [circuit](#)
- enum [mode](#) [mode](#)
- std::vector< [GraphicComponent](#) \* > [drawList](#)
- [GraphicComponent](#) \* [selectedComponent](#)
- enum [typeOrientation](#) [selectedButton](#)
- std::stack< std::pair< int, [GraphicComponent](#) \* > > [clickedStack](#)
- GRF::adjacencyMatrix [connections](#)
- unsigned int [wireCounter](#)
- QMenu \* [editMenu](#)
- QPoint [cursorLocation](#)
- QPoint [selectedPrev](#)

### 7.4.1 Construtores e Destrutores

#### 7.4.1.1 Diagram()

```
Diagram::Diagram (
    QWidget * parent = nullptr ) [explicit]
```

### 7.4.2 Funções membros

#### 7.4.2.1 clickedControl()

```
void Diagram::clickedControl (
    int x,
    int y,
    int index ) [private]
```

#### 7.4.2.2 edit

```
void Diagram::edit (
    double newValue ) [slot]
```

#### 7.4.2.3 editMode

```
void Diagram::editMode ( ) [slot]
```

#### 7.4.2.4 freeAllocatedMemory()

```
void Diagram::freeAllocatedMemory ( ) [private]
```

#### 7.4.2.5 getFileName()

```
QString Diagram::getFileName ( )
```

#### 7.4.2.6 getPixmap()

```
std::pair< QRect, QPixmap > Diagram::getPixmap (
    enum typeOrientation type ) [private]
```

#### 7.4.2.7 getStatus()

```
enum sts Diagram::getStatus ( )
```

#### 7.4.2.8 initializeDiagram()

```
void Diagram::initializeDiagram ( ) [private]
```

#### 7.4.2.9 insert()

```
void Diagram::insert (
    int x,
    int y )
```

#### 7.4.2.10 leftButtonActions()

```
void Diagram::leftButtonActions (
    int x,
    int y ) [private]
```

#### 7.4.2.11 leftButtonClicked

```
void Diagram::leftButtonClicked (
    int x,
    int y,
    int cArea ) [slot]
```

#### 7.4.2.12 load()

```
void Diagram::load ( )
```

#### 7.4.2.13 loadError

```
void Diagram::loadError (
    bool checked = false ) [signal]
```

#### 7.4.2.14 modified

```
void Diagram::modified (
    bool checked = false ) [signal]
```

#### 7.4.2.15 mouseMoveEvent()

```
void Diagram::mouseMoveEvent (
    QMouseEvent * event ) [override], [protected]
```

#### 7.4.2.16 mousePressEvent()

```
void Diagram::mousePressEvent (
    QMouseEvent * event ) [override], [protected]
```

#### 7.4.2.17 paintEvent()

```
void Diagram::paintEvent (
    QPaintEvent * event ) [override], [private]
```

#### 7.4.2.18 query

```
void Diagram::query ( ) [slot]
```

#### 7.4.2.19 queryMode

```
void Diagram::queryMode ( ) [slot]
```



#### 7.4.2.20 remove

```
void Diagram::remove ( ) [slot]
```

#### 7.4.2.21 rightButtonActions()

```
void Diagram::rightButtonActions (
    int x,
    int y ) [private]
```

#### 7.4.2.22 rightButtonClicked

```
void Diagram::rightButtonClicked (
    int x,
    int y,
    int cArea ) [slot]
```

#### 7.4.2.23 save()

```
void Diagram::save ( )
```

#### 7.4.2.24 setFileName()

```
void Diagram::setFileName (
    QString file )
```

#### 7.4.2.25 setSelectedButton()

```
void Diagram::setSelectedButton (
    enum typeOrientation button )
```

#### 7.4.2.26 setStatus()

```
void Diagram::setStatus (
    enum sts newStatus ) [private]
```

#### 7.4.2.27 showEditDialog

```
void Diagram::showEditDialog ( ) [slot]
```

### 7.4.3 Atributos

#### 7.4.3.1 circuit

```
CCT::Circuit Diagram::circuit [private]
```

#### 7.4.3.2 clickedStack

```
std::stack<std::pair<int,GraphicComponent*> > Diagram::clickedStack [private]
```

#### 7.4.3.3 connections

```
GRF::adjacencyMatrix Diagram::connections [private]
```

#### 7.4.3.4 cursorLocation

```
QPoint Diagram::cursorLocation [private]
```

#### 7.4.3.5 drawList

```
std::vector<GraphicComponent*> Diagram::drawList [private]
```

#### 7.4.3.6 editButton

```
QPushButton* Diagram::editButton [private]
```

#### 7.4.3.7 editMenu

```
QMenu* Diagram::editMenu [private]
```

#### 7.4.3.8 fileName

```
std::string Diagram::fileName [private]
```

#### 7.4.3.9 mode

```
enum mode Diagram::mode [private]
```

#### 7.4.3.10 playButton

```
QPushButton* Diagram::playButton [private]
```

#### 7.4.3.11 selectedButton

```
enum typeOrientation Diagram::selectedButton [private]
```

#### 7.4.3.12 selectedComponent

```
GraphicComponent* Diagram::selectedComponent [private]
```

#### 7.4.3.13 selectedPrev

```
QPoint Diagram::selectedPrev [private]
```

#### 7.4.3.14 status

```
enum sts Diagram::status [private]
```

#### 7.4.3.15 wireCounter

```
unsigned int Diagram::wireCounter [private]
```

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

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

## 7.5 Referência da Classe NM::EquationSystem

```
#include <Numeric.h>
```

### Membros Públicos

- [EquationSystem](#) ([Matrix](#) a, [Matrix](#) b)
- [Matrix](#) [getSolution](#) (double tol, unsigned int maxIter)
- void [printSolution](#) ()
- void [print](#) ()

### Membros Privados

- void [gaussSeidel](#) (double tol, unsigned int maxIter)
- bool [sassenfeldCriteria](#) ()
- void [findPivot](#) (unsigned int startI)
- void [gaussJordan](#) ()

### Atributos Privados

- [Matrix](#) A
- [Matrix](#) B
- [Matrix](#) x

### 7.5.1 Construtores e Destrutores

#### 7.5.1.1 EquationSystem()

```
NM::EquationSystem::EquationSystem (  
    Matrix a,  
    Matrix b )
```

## 7.5.2 Funções membros

### 7.5.2.1 findPivot()

```
void NM::EquationSystem::findPivot (
    unsigned int startI ) [private]
```

### 7.5.2.2 gaussJordan()

```
void NM::EquationSystem::gaussJordan ( ) [private]
```

### 7.5.2.3 gaussSeidel()

```
void NM::EquationSystem::gaussSeidel (
    double tol,
    unsigned int maxIter ) [private]
```

### 7.5.2.4 getSolution()

```
Matrix NM::EquationSystem::getSolution (
    double tol,
    unsigned int maxIter )
```

### 7.5.2.5 print()

```
void NM::EquationSystem::print ( )
```

### 7.5.2.6 printSolution()

```
void NM::EquationSystem::printSolution ( )
```

### 7.5.2.7 sassenfeldCriteria()

```
bool NM::EquationSystem::sassenfeldCriteria ( ) [private]
```

## 7.5.3 Atributos

### 7.5.3.1 A

```
Matrix NM::EquationSystem::A [private]
```

### 7.5.3.2 B

```
Matrix NM::EquationSystem::B [private]
```

### 7.5.3.3 x

```
Matrix NM::EquationSystem::x [private]
```

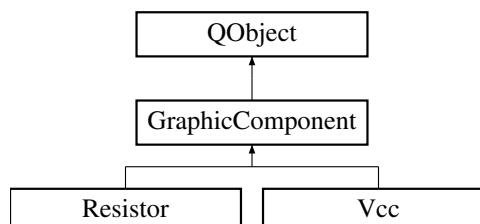
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

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

## 7.6 Referência da Classe GraphicComponent

```
#include <GraphicComponent.h>
```

Diagrama de hierarquia para GraphicComponent:



## Sinais

- void `clickedVertex` (int `Area`, `GraphicComponent` \*)
- void `clicked` (bool `checked`=false)

## Membros Públicos

- `GraphicComponent` (int `x`, int `y`, enum `orien` s, `QObject` \*parent=nullptr)
- `GraphicComponent` (`QObject` \*parent=nullptr)
- `GraphicComponent` (`GraphicComponent` &`C`, `QObject` \*paren=nullptr)
- int `clickedArea` (int `x`, int `y`)
- void `draw` (`QPainter` \*painter)
- int `getHeight` ()
- int `getWidth` ()
- `QPoint` `getBottom` ()
- `QPoint` `getTop` ()
- `QPoint` `getLeft` ()
- `QPoint` `getRight` ()
- `QPixmap` `getMap` ()
- `QRect` `getBoundRect` ()
- enum `orien` `getOrientation` ()
- void `setVertex1` (unsigned int `vtx`)
- void `setVertex2` (unsigned int `vtx`)
- unsigned int `getVertex1` ()
- unsigned int `getVertex2` ()
- std::string `getLabel` ()
- virtual `CMP::type` `getType` ()=0
- double `getValue` ()
- void `setValue` (double `newValue`)
- `QPoint` `getVertex1Point` ()
- `QPoint` `getVertex2Point` ()

## Membros Públicos Estáticos

- static unsigned int `getVertexNum` ()

## Atributos Públicos

- enum `orien` `orientation`

## Atributos Protegidos

- int `x`
- int `y`
- `QRect` `vertexArea1`
- `QRect` `vertexArea2`
- `QRect` `boundRect`
- `QPixmap` `map`
- unsigned int `vertex1`
- unsigned int `vertex2`
- `CMP::type` `componentType`
- `QString` `label`
- double `value`

## Atributos Privados Estáticos

- static unsigned int `vertexNum` = 0

## 7.6.1 Construtores e Destrutores

### 7.6.1.1 GraphicComponent() [1/3]

```
GraphicComponent::GraphicComponent (
    int x,
    int y,
    enum orientation s,
    QObject * parent = nullptr ) [explicit]
```

### 7.6.1.2 GraphicComponent() [2/3]

```
GraphicComponent::GraphicComponent (
    QObject * parent = nullptr ) [explicit]
```

### 7.6.1.3 GraphicComponent() [3/3]

```
GraphicComponent::GraphicComponent (
    GraphicComponent & C,
    QObject * paren = nullptr ) [explicit]
```

## 7.6.2 Funções membros

### 7.6.2.1 clicked

```
void GraphicComponent::clicked (
    bool checked = false ) [signal]
```



### 7.6.2.2 clickedArea()

```
int GraphicComponent::clickedArea (
    int x,
    int y )
```

### 7.6.2.3 clickedVertex

```
void GraphicComponent::clickedVertex (
    int Area,
    GraphicComponent * ) [signal]
```

### 7.6.2.4 draw()

```
void GraphicComponent::draw (
    QPainter * painter )
```

### 7.6.2.5 getBottom()

```
QPoint GraphicComponent::getBottom ( )
```

### 7.6.2.6 getBoundRect()

```
QRect GraphicComponent::getBoundRect ( )
```

### 7.6.2.7 getHeight()

```
int GraphicComponent::getHeight ( )
```

### 7.6.2.8 getLabel()

```
std::string GraphicComponent::getLabel ( )
```

#### 7.6.2.9 getLeft()

```
QPoint GraphicComponent::getLeft ( )
```

#### 7.6.2.10 getMap()

```
QPixmap GraphicComponent::getMap ( )
```

#### 7.6.2.11 getOrientation()

```
enum orien GraphicComponent::getOrientation ( )
```

#### 7.6.2.12 getRight()

```
QPoint GraphicComponent::getRight ( )
```

#### 7.6.2.13 getTop()

```
QPoint GraphicComponent::getTop ( )
```

#### 7.6.2.14 getType()

```
virtual CMP::type GraphicComponent::getType ( ) [pure virtual]
```

Implementado por [Vcc](#) e [Resistor](#).

#### 7.6.2.15 getValue()

```
double GraphicComponent::getValue ( )
```

**7.6.2.16 getVertex1()**

```
unsigned int GraphicComponent::getVertex1 ( )
```

**7.6.2.17 getVertex1Point()**

```
QPoint GraphicComponent::getVertex1Point ( )
```

**7.6.2.18 getVertex2()**

```
unsigned int GraphicComponent::getVertex2 ( )
```

**7.6.2.19 getVertex2Point()**

```
QPoint GraphicComponent::getVertex2Point ( )
```

**7.6.2.20 getVertexNum()**

```
static unsigned int GraphicComponent::getVertexNum ( ) [inline], [static]
```

**7.6.2.21 getWidth()**

```
int GraphicComponent::getWidth ( )
```

**7.6.2.22 setValue()**

```
void GraphicComponent::setValue (
    double newValue )
```

#### 7.6.2.23 setVertex1()

```
void GraphicComponent::setVertex1 (
    unsigned int vtx )
```

#### 7.6.2.24 setVertex2()

```
void GraphicComponent::setVertex2 (
    unsigned int vtx )
```

### 7.6.3 Atributos

#### 7.6.3.1 boundRect

```
QRect GraphicComponent::boundRect [protected]
```

#### 7.6.3.2 componentType

```
CMP::type GraphicComponent::componentType [protected]
```

#### 7.6.3.3 label

```
QString GraphicComponent::label [protected]
```

#### 7.6.3.4 map

```
QPixmap GraphicComponent::map [protected]
```

#### 7.6.3.5 orientation

```
enum orien GraphicComponent::orientation
```

#### 7.6.3.6 value

```
double GraphicComponent::value [protected]
```

#### 7.6.3.7 vertex1

```
unsigned int GraphicComponent::vertex1 [protected]
```

#### 7.6.3.8 vertex2

```
unsigned int GraphicComponent::vertex2 [protected]
```

#### 7.6.3.9 vertexArea1

```
QRect GraphicComponent::vertexArea1 [protected]
```

#### 7.6.3.10 vertexArea2

```
QRect GraphicComponent::vertexArea2 [protected]
```

#### 7.6.3.11 vertexNum

```
unsigned int GraphicComponent::vertexNum = 0 [static], [private]
```

#### 7.6.3.12 x

```
int GraphicComponent::x [protected]
```

### 7.6.3.13 y

```
int GraphicComponent::y [protected]
```

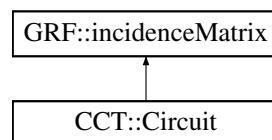
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

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

## 7.7 Referência da Classe GRF::incidenceMatrix

```
#include <Graph.h>
```

Diagrama de hierarquia para GRF::incidenceMatrix:



### Membros Públicos

- [incidenceMatrix](#) ()
- [incidenceMatrix](#) (unsigned int v, unsigned int e)
- `std::vector< int > & operator\[\]` (unsigned int vtx)
- void [makeCon](#) (unsigned int vtx1, unsigned int vtx2, unsigned int edge)
- void [addEdge](#) (unsigned int vtx1, unsigned int vtx2)
- void [removeEdge](#) (unsigned int edg)
- void [removeVertex](#) (unsigned int vtx)
- unsigned int [getVertexNumber](#) ()
- unsigned int [getEdgeNumber](#) ()
- `std::vector< int > getLoop` (unsigned int key)
- [incidenceMatrix](#) [getSpanningTree](#) (unsigned int key)
- unsigned int [getVertexCon](#) (unsigned int vtx, unsigned int i)
- `std::vector< unsigned int > getEdges` (unsigned int vtx)
- `std::vector< unsigned int > getEdges` (unsigned int vtx1, unsigned int vtx2)
- `std::pair< unsigned int, unsigned int > getVertex` (unsigned int edg)
- unsigned int [getConNum](#) (unsigned int vtx)
- void [print](#) ()

### Atributos Protegidos

- `std::vector< std::vector< int > > inMatrix`

### Atributos Privados

- unsigned int [vertexNumber](#)
- unsigned int [edgeNumber](#)

## 7.7.1 Construtores e Destrutores

### 7.7.1.1 incidenceMatrix() [1/2]

```
GRF::incidenceMatrix::incidenceMatrix ( )
```

### 7.7.1.2 incidenceMatrix() [2/2]

```
GRF::incidenceMatrix::incidenceMatrix (
    unsigned int v,
    unsigned int e )
```

## 7.7.2 Funções membros

### 7.7.2.1 addEdge()

```
void GRF::incidenceMatrix::addEdge (
    unsigned int vtx1,
    unsigned int vtx2 )
```

### 7.7.2.2 getConNum()

```
unsigned int GRF::incidenceMatrix::getConNum (
    unsigned int vtx )
```

### 7.7.2.3 getEdgeNumber()

```
unsigned int GRF::incidenceMatrix::getEdgeNumber ( )
```

### 7.7.2.4 getEdges() [1/2]

```
std::vector< unsigned int > GRF::incidenceMatrix::getEdges (
    unsigned int vtx )
```

#### 7.7.2.5 getEdges() [2/2]

```
std::vector< unsigned int > GRF::incidenceMatrix::getEdges (
    unsigned int vtx1,
    unsigned int vtx2 )
```

#### 7.7.2.6 getLoop()

```
std::vector< int > GRF::incidenceMatrix::getLoop (
    unsigned int key )
```

#### 7.7.2.7 getSpanningTree()

```
incidenceMatrix GRF::incidenceMatrix::getSpanningTree (
    unsigned int key )
```

#### 7.7.2.8 getVertex()

```
std::pair< unsigned int, unsigned int > GRF::incidenceMatrix::getVertex (
    unsigned int edg )
```

#### 7.7.2.9 getVertexCon()

```
unsigned int GRF::incidenceMatrix::getVertexCon (
    unsigned int vtx,
    unsigned int i )
```

#### 7.7.2.10 getVertexNumber()

```
unsigned int GRF::incidenceMatrix::getVertexNumber ( )
```

#### 7.7.2.11 makeCon()

```
void GRF::incidenceMatrix::makeCon (
    unsigned int vtx1,
    unsigned int vtx2,
    unsigned int edge )
```



#### 7.7.2.12 operator[]()

```
std::vector< int > & GRF::incidenceMatrix::operator[] (
    unsigned int vtx )
```

#### 7.7.2.13 print()

```
void GRF::incidenceMatrix::print ( )
```

#### 7.7.2.14 removeEdge()

```
void GRF::incidenceMatrix::removeEdge (
    unsigned int edg )
```

#### 7.7.2.15 removeVertex()

```
void GRF::incidenceMatrix::removeVertex (
    unsigned int vtx )
```

### 7.7.3 Atributos

#### 7.7.3.1 edgeNumber

```
unsigned int GRF::incidenceMatrix::edgeNumber [private]
```

#### 7.7.3.2 inMatrix

```
std::vector<std::vector<int> > GRF::incidenceMatrix::inMatrix [protected]
```

### 7.7.3.3 vertexNumber

```
unsigned int GRF::incidenceMatrix::vertexNumber [private]
```

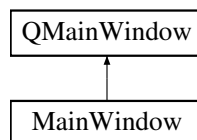
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

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

## 7.8 Referência da Classe MainWindow

```
#include <MainWindow.h>
```

Diagrama de hierarquia para MainWindow:



### Slots Públicos

- void [newFile](#) ()
- void [openFile](#) ()
- void [saveFile](#) ()
- void [saveFileAs](#) ()
- void [preferences](#) ()
- void [setTabStatus](#) (bool modified)
- void [closeFile](#) (int index)
- void [drawRes90](#) ()
- void [drawRes180](#) ()
- void [drawVcc90](#) ()
- void [drawVcc180](#) ()

### Membros Públicos

- [MainWindow](#) (QWidget \*parent=nullptr)

### Membros Privados

- void [initializeMenu](#) ()
- void [initializeToolbar](#) ()
- void [initializeTabs](#) ()
- void [openFile](#) (QString fileName)
- void [initializeStatusBar](#) ()

## Atributos Privados

- `std::list< Diagram * > diagrams`
- `QTabWidget * tabs`
- `QMenuBar * mainBar`
- `QMenu * fileMenu`
- `QMenu * prefMenu`
- `QMenu * helpMenu`
- `QAction * openFileAct`
- `QAction * saveFileAct`
- `QAction * saveFileAsAct`
- `QAction * newFileAct`
- `QAction * prefAct`
- `QAction * tutorialAct`
- `QToolBar * toolbar`
- `QStatusBar * statusBar`
- `QWidget * initialTab`

## 7.8.1 Construtores e Destrutores

### 7.8.1.1 MainWindow()

```
MainWindow::MainWindow (  
    QWidget * parent = nullptr ) [explicit]
```

## 7.8.2 Funções membros

### 7.8.2.1 closeFile

```
void MainWindow::closeFile (  
    int index ) [slot]
```

### 7.8.2.2 drawRes180

```
void MainWindow::drawRes180 ( ) [slot]
```

### 7.8.2.3 drawRes90

```
void MainWindow::drawRes90 ( ) [slot]
```

### 7.8.2.4 drawVcc180

```
void MainWindow::drawVcc180 ( ) [slot]
```

### 7.8.2.5 drawVcc90

```
void MainWindow::drawVcc90 ( ) [slot]
```

### 7.8.2.6 initializeMenu()

```
void MainWindow::initializeMenu ( ) [private]
```

### 7.8.2.7 initializeStatusBar()

```
void MainWindow::initializeStatusBar ( ) [private]
```

### 7.8.2.8 initializeTabs()

```
void MainWindow::initializeTabs ( ) [private]
```

### 7.8.2.9 initializeToolbar()

```
void MainWindow::initializeToolbar ( ) [private]
```

### 7.8.2.10 newFile

```
void MainWindow::newFile ( ) [slot]
```

#### 7.8.2.11 openFile [1/2]

```
void MainWindow::openFile ( ) [slot]
```

#### 7.8.2.12 openFile() [2/2]

```
void MainWindow::openFile (
    QString fileName ) [private]
```

#### 7.8.2.13 preferences

```
void MainWindow::preferences ( ) [slot]
```

#### 7.8.2.14 saveFile

```
void MainWindow::saveFile ( ) [slot]
```

#### 7.8.2.15 saveFileAs

```
void MainWindow::saveFileAs ( ) [slot]
```

#### 7.8.2.16 setTabStatus

```
void MainWindow::setTabStatus (
    bool modified ) [slot]
```

### 7.8.3 Atributos

#### 7.8.3.1 diagrams

```
std::list<Diagram*> MainWindow::diagrams [private]
```

### 7.8.3.2 fileMenu

```
QMenu* MainWindow::fileMenu [private]
```

### 7.8.3.3 helpMenu

```
QMenu* MainWindow::helpMenu [private]
```

### 7.8.3.4 initialTab

```
QWidget* MainWindow::initialTab [private]
```

### 7.8.3.5 mainBar

```
QMenuBar* MainWindow::mainBar [private]
```

### 7.8.3.6 newFileAct

```
QAction* MainWindow::newFileAct [private]
```

### 7.8.3.7 openFileAct

```
QAction* MainWindow::openFileAct [private]
```

### 7.8.3.8 prefAct

```
QAction* MainWindow::prefAct [private]
```

### 7.8.3.9 prefMenu

```
QMenu* MainWindow::prefMenu [private]
```

#### 7.8.3.10 saveFileAct

```
QAction* MainWindow::saveFileAct [private]
```

#### 7.8.3.11 saveFileAsAct

```
QAction* MainWindow::saveFileAsAct [private]
```

#### 7.8.3.12 statusBar

```
QStatusBar* MainWindow::statusBar [private]
```

#### 7.8.3.13 tabs

```
QTabWidget* MainWindow::tabs [private]
```

#### 7.8.3.14 toolbar

```
QToolBar* MainWindow::toolbar [private]
```

#### 7.8.3.15 tutorialAct

```
QAction* MainWindow::tutorialAct [private]
```

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

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

## 7.9 Referência da Classe NM::Matrix

```
#include <Numeric.h>
```

## Membros Públicos

- [Matrix](#) (unsigned int r)
- [Matrix](#) (unsigned int r, unsigned int c)
- [Matrix operator\\*](#) ([Matrix](#) &m1)
- void [operator\\*=](#) ([Matrix](#) &m1)
- [Matrix operator+](#) ([Matrix](#) &m1)
- void [operator+=](#) ([Matrix](#) &m1)
- [Matrix operator-](#) ([Matrix](#) &m1)
- void [operator-=](#) ([Matrix](#) &m1)
- void [operator=](#) (const [Matrix](#) &m1)
- std::vector< double > & [operator\[\]](#) (unsigned int index)
- [Matrix operator-](#) ()
- [Matrix transpose](#) ()
- [Matrix Abs](#) ()
- std::vector< double > [getCol](#) (unsigned int col)
- unsigned int [getColNumber](#) () const
- unsigned int [getRowNumber](#) () const
- void [swapLines](#) (double l1, double l2)
- void [print](#) ()
- void [print](#) (std::vector< [CMP::Component](#) \* > C)

## Atributos Privados

- std::vector< std::vector< double > > [realMatrix](#)
- unsigned int [rowNumber](#)
- unsigned int [colNumber](#)

## 7.9.1 Construtores e Destrutores

### 7.9.1.1 [Matrix\(\)](#) [1/2]

```
NM::Matrix::Matrix (
    unsigned int r )
```

### 7.9.1.2 [Matrix\(\)](#) [2/2]

```
NM::Matrix::Matrix (
    unsigned int r,
    unsigned int c )
```

## 7.9.2 Funções membros



### 7.9.2.1 Abs()

```
Matrix NM::Matrix::Abs ( )
```

### 7.9.2.2 getCol()

```
std::vector< double > NM::Matrix::getCol (
    unsigned int col )
```

### 7.9.2.3 getColNumber()

```
unsigned int NM::Matrix::getColNumber ( ) const
```

### 7.9.2.4 getRowNumber()

```
unsigned int NM::Matrix::getRowNumber ( ) const
```

### 7.9.2.5 operator\*()

```
Matrix NM::Matrix::operator* (
    Matrix & m1 )
```

### 7.9.2.6 operator\*=( )

```
void NM::Matrix::operator*= (
    Matrix & m1 )
```

### 7.9.2.7 operator+( )

```
Matrix NM::Matrix::operator+ (
    Matrix & m1 )
```

#### 7.9.2.8 operator+=()

```
void NM::Matrix::operator+= (
    Matrix & m1 )
```

#### 7.9.2.9 operator-() [1/2]

```
Matrix NM::Matrix::operator- ( )
```

#### 7.9.2.10 operator-() [2/2]

```
Matrix NM::Matrix::operator- (
    Matrix & m1 )
```

#### 7.9.2.11 operator-=()

```
void NM::Matrix::operator-= (
    Matrix & m1 )
```

#### 7.9.2.12 operator=()

```
void NM::Matrix::operator= (
    const Matrix & m1 )
```

#### 7.9.2.13 operator[]()

```
std::vector< double > & NM::Matrix::operator[] (
    unsigned int index )
```

#### 7.9.2.14 print() [1/2]

```
void NM::Matrix::print ( )
```

#### 7.9.2.15 print() [2/2]

```
void NM::Matrix::print (
    std::vector< CMP::Component * > C )
```

#### 7.9.2.16 swapLines()

```
void NM::Matrix::swapLines (
    double l1,
    double l2 )
```

#### 7.9.2.17 transpose()

```
Matrix NM::Matrix::transpose ( )
```

### 7.9.3 Atributos

#### 7.9.3.1 colNumber

```
unsigned int NM::Matrix::colNumber [private]
```

#### 7.9.3.2 realMatrix

```
std::vector<std::vector<double> > NM::Matrix::realMatrix [private]
```

#### 7.9.3.3 rowNumber

```
unsigned int NM::Matrix::rowNumber [private]
```

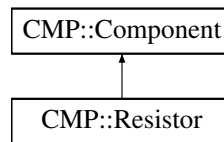
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

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

## 7.10 Referência da Classe CMP::Resistor

```
#include <Component.h>
```

Diagrama de hierarquia para CMP::Resistor:



### Membros Públicos

- [Resistor](#) (std::string l, double res, unsigned int vtx1, unsigned int vtx2)
- [~Resistor](#) ()
- double [getResistance](#) ()
- void [setResistance](#) (double res)
- enum [type](#) [getType](#) ()
- void [setCurrent](#) (double cur)
- void [setVoltage](#) (double cur)

### Atributos Privados

- double [resistance](#)

### Outros membros herdados

#### 7.10.1 Construtores e Destrutores

##### 7.10.1.1 Resistor()

```
Resistor::Resistor (
    std::string l,
    double res,
    unsigned int vtx1,
    unsigned int vtx2 )
```

##### 7.10.1.2 ~Resistor()

```
Resistor::~~Resistor ( )
```

## 7.10.2 Funções membros

### 7.10.2.1 getResistance()

```
double Resistor::getResistance ( )
```

### 7.10.2.2 getType()

```
enum type Resistor::getType ( ) [virtual]
```

Implementa [CMP::Component](#).

### 7.10.2.3 setCurrent()

```
void Resistor::setCurrent (
    double cur ) [virtual]
```

Implementa [CMP::Component](#).

### 7.10.2.4 setResistance()

```
void Resistor::setResistance (
    double res )
```

### 7.10.2.5 setVoltage()

```
void Resistor::setVoltage (
    double cur ) [virtual]
```

Implementa [CMP::Component](#).

## 7.10.3 Atributos

### 7.10.3.1 resistance

```
double CMP::Resistor::resistance [private]
```

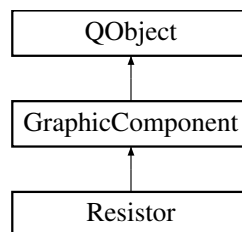
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

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

## 7.11 Referência da Classe Resistor

```
#include <GraphicComponent.h>
```

Diagrama de hierarquia para Resistor:



### Membros Públicos

- [Resistor](#) (int *x*, int *y*, enum *orien* *s*=[VERTICAL](#), QObject \**parent*=nullptr)
- [CMP::type](#) [getType](#) ()

### Atributos Privados Estáticos

- static unsigned int [resCounter](#) = 0

### Outros membros herdados

### 7.11.1 Construtores e Destrutores

#### 7.11.1.1 Resistor()

```
Resistor::Resistor (  
    int x,  
    int y,  
    enum orien s = VERTICAL,  
    QObject * parent = nullptr ) [explicit]
```

### 7.11.2 Funções membros

#### 7.11.2.1 getType()

```
CMP::type Resistor::getType ( ) [virtual]
```

Implementa [GraphicComponent](#).

### 7.11.3 Atributos

#### 7.11.3.1 resCounter

```
unsigned int Resistor::resCounter = 0 [static], [private]
```

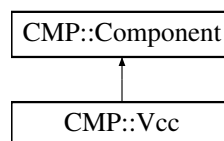
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

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

## 7.12 Referência da Classe CMP::Vcc

```
#include <Component.h>
```

Diagrama de hierarquia para CMP::Vcc:



### Membros Públicos

- [Vcc](#) (std::string l, double vol, unsigned int negative, unsigned int positive)
- [~Vcc](#) ()
- enum [type](#) [getType](#) ()
- void [setCurrent](#) (double cur)
- void [setVoltage](#) (double vol)

## Outros membros herdados

### 7.12.1 Construtores e Destrutores

#### 7.12.1.1 Vcc()

```
Vcc::Vcc (
    std::string l,
    double vol,
    unsigned int negative,
    unsigned int positive )
```

#### 7.12.1.2 ~Vcc()

```
Vcc::~Vcc ( )
```

### 7.12.2 Funções membros

#### 7.12.2.1 getType()

```
enum type Vcc::getType ( ) [virtual]
```

Implementa [CMP::Component](#).

#### 7.12.2.2 setCurrent()

```
void Vcc::setCurrent (
    double cur ) [virtual]
```

Implementa [CMP::Component](#).



### 7.12.2.3 setVoltage()

```
void Vcc::setVoltage (
    double vol ) [virtual]
```

Implementa [CMP::Component](#).

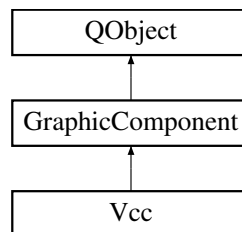
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

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

## 7.13 Referência da Classe Vcc

```
#include <GraphicComponent.h>
```

Diagrama de hierarquia para Vcc:



### Membros Públicos

- [Vcc](#) (int *x*, int *y*, enum *orien* *s*=[VERTICAL](#), QObject \**parent*=nullptr)
- [CMP::type](#) *getType* ()

### Atributos Privados Estáticos

- static unsigned int [vccCounter](#) = 0

### Outros membros herdados

## 7.13.1 Construtores e Destrutores

### 7.13.1.1 Vcc()

```
Vcc::Vcc (
    int x,
    int y,
    enum orien s = VERTICAL,
    QObject * parent = nullptr ) [explicit]
```

### 7.13.2 Funções membros

#### 7.13.2.1 getType()

```
CMP::type Vcc::getType ( ) [virtual]
```

Implementa [GraphicComponent](#).

### 7.13.3 Atributos

#### 7.13.3.1 vccCounter

```
unsigned int Vcc::vccCounter = 0 [static], [private]
```

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

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

## Capítulo 8

# Arquivos

### 8.1 Referência do Arquivo Circuit.cpp

Implementação da classe Circuit.

```
#include "Circuit.h"  
#include "Graph.h"  
#include "Numeric.h"  
#include <vector>  
#include <string>  
#include <stack>  
#include <fstream>
```

**<em>Namespaces</em>**

- [CCT](#)

#### 8.1.1 Descrição detalhada

Implementação da classe Circuit.

**Autores**

: Lucas Carvalho; Rafael Marasca Martins

**Data**

: 20 04 2021

Este arquivo contém as implementações dos métodos e membros da classe Circuit.

A classe Circuit fornece os métodos para se interligar objetos da classe Component e obter os valores relacionados a cada um destes.

## 8.2 Referência do Arquivo Circuit.h

Implementação da classe Circuit.

```
#include "Graph.h"
#include "Component.h"
#include <vector>
#include <string>
```

### Componentes

- class [CCT::Circuit](#)  
*Implementação da classe [Circuit](#).*

**<em>Namespaces</em>**

- [CCT](#)

### 8.2.1 Descrição detalhada

Implementação da classe Circuit.

#### Autores

: Lucas Carvalho; Rafael Marasca Martins

#### Data

: 20 04 2021

Este arquivo contém as declarações dos métodos e membros da classe Circuit.

A classe Circuit fornece os métodos para se interligar objetos da classe Component e obter os valores relacionados a cada um destes.

## 8.3 Referência do Arquivo Component.cpp

Implementação da classe Component.

```
#include "Component.h"
#include <string>
#include <iostream>
```

**<em>Namespaces</em>**

- [CMP](#)

### 8.3.1 Descrição detalhada

Implementação da classe Component.

#### Autores

: Lucas Carvalho; Rafael Marasca Martins

#### Data

: 30 04 2021

Este arquivo contém as implementações dos métodos e membros da classe Component.

A classe Component fornece os métodos e definições para se criar e alterar objetos da classe Component. Estes objetos são necessários para se resolver o circuito.

## 8.4 Referência do Arquivo Component.h

Declaração da classe Component.

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

### Componentes

- class `CMP::Component`
- class `CMP::Resistor`
- class `CMP::Vcc`

### <em>Namespaces</em>

- `CMP`

### Enumerações

- enum `CMP::type` { `CMP::RESISTOR` , `CMP::VCC` , `CMP::WIRE` , `CMP::GROUND` }

### 8.4.1 Descrição detalhada

Declaração da classe Component.

#### Autores

: Lucas Carvalho; Rafael Marasca Martins

#### Data

: 30 04 2021

Este arquivo contém as ideclarações dos métodos e membros da classe Component.

A classe Component fornece os métodos e definições para se criar e alterar objetos da classe Component. Estes objetos são necessários para se resolver o circuito.

## 8.5 Referência do Arquivo Diagram.cpp

Declaração da classe Numeric.

```
#include "Diagram.h"
#include <fstream>
#include <QPushButton>
#include <QVBoxLayout>
#include <QPainter>
#include <QScrollBar>
#include <QDebug>
#include <QMenu>
#include <QAction>
#include <QMessageBox>
#include <QInputDialog>
```

### 8.5.1 Descrição detalhada

Declaração da classe Numeric.

#### Autores

: Lucas Carvalho; Rafael Marasca Martins

#### Data

: 30 04 2021

Este arquivo contém as implementações dos métodos e membros da classe [Diagram](#).

A classe [Diagram](#) fornece as facilidades para inserir objetos da classe GraphicsComponent graficamente, identificar as interações do usuário com estes objetos, bem como, interligar a implementação gráfica com a parte numérica.

## 8.6 Referência do Arquivo Diagram.h

Declaração da classe Numeric.

```
#include "Circuit.h"
#include "Graph.h"
#include "GraphicComponent.h"
#include <QWidget>
#include <QObject>
#include <QScrollBar>
#include <QPushButton>
#include <QScrollArea>
#include <QMouseEvent>
#include <stack>
#include <QMessageBox>
#include <QPropertyAnimation>
```

## Componentes

- class [Diagram](#)

## Enumerações

- enum [typeOrientation](#) {  
    [VCC90](#) , [VCC180](#) , [RES90](#) , [RES180](#) ,  
    [NONE](#) }
- enum [sts](#) { [UNSAVED](#) , [MODIFIED](#) , [OK](#) , [ERROR](#) }
- enum [mode](#) { [EDIT](#) , [QUERY](#) }

### 8.6.1 Descrição detalhada

Declaração da classe Numeric.

#### Autores

: Lucas Carvalho; Rafael Marasca Martins

#### Data

: 30 04 2021

Este arquivo contém as declarações dos métodos e membros da classe [Diagram](#).

A classe [Diagram](#) fornece as facilidades para inserir objetos da classe GraphicsComponent graficamente, identificar as interações do usuário com estes objetos, bem como, interligar a implementação gráfica com a parte numérica.

### 8.6.2 Enumerações

#### 8.6.2.1 mode

enum [mode](#)

##### Enumeradores

EDIT	
QUERY	

#### 8.6.2.2 sts

enum [sts](#)

## Enumeradores

UNSAVED	
MODIFIED	
OK	
ERROR	

## 8.6.2.3 typeOrientation

```
enum typeOrientation
```

## Enumeradores

VCC90	
VCC180	
RES90	
RES180	
NONE	

## 8.7 Referência do Arquivo Graph.cpp

Implementação das classes de grafo utilizadas no projeto.

```
#include "Graph.h"
#include <vector>
#include <stack>
```

## &lt;em&gt;Namespaces&lt;/em&gt;

- [GRF](#)

## 8.7.1 Descrição detalhada

Implementação das classes de grafo utilizadas no projeto.

## Autores

: Lucas Carvalho; Rafael Marasca Martins

## Data

: 19 04 2021

Este arquivo contém as implementações dos métodos e membros utilizados por duas classes de grafo distintas.

A classe incidenceMatrix implementa um grafo que utiliza matriz de incidência como estrutura base.

A classe adjacencyMatrix implemeta um grafo utilizando matriz de adjacência como estrutura base.



## 8.8 Referência do Arquivo Graph.h

Declaração das classes de grafo utilizadas no projeto.

```
#include <iostream>
#include <vector>
```

### Componentes

- class [GRF::incidenceMatrix](#)
- class [GRF::adjacencyMatrix](#)

### <em>Namespaces</em>

- [GRF](#)

### 8.8.1 Descrição detalhada

Declaração das classes de grafo utilizadas no projeto.

#### Autores

: Lucas Carvalho; Rafael Marasca Martins

#### Data

: 19 04 2021

Este arquivo contém as declarações dos métodos e membros utilizados por duas classes de grafo distintas.

A classe incidenceMatrix implementa um grafo que utiliza matriz de incidência como estrutura base.

A classe adjacencyMatrix implemeta um grafo utilizando matriz de adjacência como estrutura base.

## 8.9 Referência do Arquivo GraphicComponent.cpp

Implementação da classe [GraphicComponent](#) para a inserção gráfica de componentes.

```
#include "GraphicComponent.h"
#include <QtDebug>
#include <QPen>
#include <QPoint>
```

### 8.9.1 Descrição detalhada

Implementação da classe [GraphicComponent](#) para a inserção gráfica de componentes.

#### Autores

: Lucas Carvalho; Rafael Marasca Martins

#### Data

: 30 04 2021

Este arquivo contém as implementações dos métodos e membros que são utilizados para inserir componenetes graficamente em objeto da classe Diagrama através dos métodos da classe [Diagram](#).

## 8.10 Referência do Arquivo GraphicComponent.h

Declaração da classe [GraphicComponent](#) para a inserção gráfica de componentes.

```
#include "Graph.h"
#include "Component.h"
#include <QObject>
#include <QRect>
#include <QPoint>
#include <QPainter>
```

### Componentes

- class [GraphicComponent](#)
- class [Resistor](#)
- class [Vcc](#)

### Definições e Macros

- #define [HEIGHT](#) 115
- #define [WIDTH](#) 50

### Enumerações

- enum [orien](#) { [VERTICAL](#) , [HORIZONTAL](#) }

### 8.10.1 Descrição detalhada

Declaração da classe [GraphicComponent](#) para a inserção gráfica de componentes.

#### Autores

: Lucas Carvalho; Rafael Marasca Martins

#### Data

: 30 04 2021

Este arquivo contém as declarações necessárias para se inserir componentes graficamente em um objeto da classe [Diagram](#).

### 8.10.2 Definições e macros

#### 8.10.2.1 HEIGHT

```
#define HEIGHT 115
```

#### 8.10.2.2 WIDTH

```
#define WIDTH 50
```

### 8.10.3 Enumerações

#### 8.10.3.1 orien

```
enum orien
```

#### Enumeradores

VERTICAL	
HORIZONTAL	

## 8.11 Referência do Arquivo main.cpp

```
#include "MainWindow.h"  
#include <QApplication>
```

### Funções

- int [main](#) (int argc, char \*argv[])

### 8.11.1 Funções

#### 8.11.1.1 main()

```
int main (  
    int argc,  
    char * argv[] )
```

## 8.12 Referência do Arquivo MainWindow.cpp

Implementação da classe [MainWindow](#).

```
#include "MainWindow.h"  
#include "Diagram.h"  
#include <QMenuBar>  
#include <QMenu>  
#include <QAction>  
#include <QFileDialog>  
#include <QMessageBox>  
#include <QString>  
#include <QDebug>  
#include <QPushButton>  
#include <QTabWidget>  
#include <QTabBar>  
#include <QFileInfo>  
#include <QPixmap>  
#include <QIcon>  
#include <QStatusBar>  
#include <QVBoxLayout>  
#include <QLabel>
```

### 8.12.1 Descrição detalhada

Implementação da classe [MainWindow](#).

#### Autores

: Lucas Carvalho; Rafael Marasca Martins

#### Data

: 28 04 2021

Este arquivo contém as implementações dos métodos e membros da classe [MainWindow](#).

A classe [MainWindow](#) controla o fluxo de execução com base nas interações do usuário.

## 8.13 Referência do Arquivo MainWindow.h

Declaração da classe [MainWindow](#).

```
#include "Diagram.h"
#include <QMainWindow>
#include <QMenuBar>
#include <QMenu>
#include <QAction>
#include <QFileDialog>
#include <QString>
#include <QToolBar>
#include <list>
#include <QPushButton>
```

### Componentes

- class [MainWindow](#)

### 8.13.1 Descrição detalhada

Declaração da classe [MainWindow](#).

#### Autores

: Lucas Carvalho; Rafael Marasca Martins

#### Data

: 28 04 2021

Este arquivo contém as declarações dos métodos e membros da classe [MainWindow](#), que representa a janela principal do programa.

A classe [MainWindow](#) controla o fluxo de execução com base nas interações do usuário.

## 8.14 Referência do Arquivo Numeric.cpp

Declaração da classe Numeric.

```
#include "Numeric.h"  
#include <algorithm>  
#include <cmath>  
#include <string>
```

### <em>Namespaces</em>

- [NM](#)

### 8.14.1 Descrição detalhada

Declaração da classe Numeric.

#### Autores

: Lucas Carvalho; Rafael Marasca Martins

#### Data

: 19 04 2021

Este arquivo contém as declarações dos métodos e membros da classe Numeric.

A classe Numeric fornece as classes e métodos para se criar e resolver sistemas de equações.

## 8.15 Referência do Arquivo Numeric.h

Implementação da classe Numeric.

```
#include "Circuit.h"  
#include <vector>  
#include <iostream>
```

### Componentes

- class [NM::Matrix](#)
- class [NM::EquationSystem](#)

### <em>Namespaces</em>

- [NM](#)

### 8.15.1 Descrição detalhada

Implementação da classe Numeric.

**Autores**

: Lucas Carvalho; Rafael Marasca Martins

**Data**

: 19 04 2021

Este arquivo contém a implementação dos métodos e membros da classe Numeric.

A classe Numeric fornece as classes e métodos para se criar e resolver sistemas de equações.





# Índice Remissivo

- ~Circuit
  - CCT::Circuit, [16](#)
- ~Component
  - CMP::Component, [22](#)
- ~Resistor
  - CMP::Resistor, [56](#)
- ~Vcc
  - CMP::Vcc, [60](#)
- A
  - NM::EquationSystem, [34](#)
- Abs
  - NM::Matrix, [52](#)
- addComponent
  - CCT::Circuit, [17](#)
- addEdge
  - GRF::incidenceMatrix, [43](#)
- adjacencyMatrix
  - GRF::adjacencyMatrix, [13](#)
- adjMatrix
  - GRF::adjacencyMatrix, [14](#)
- B
  - NM::EquationSystem, [34](#)
- boundRect
  - GraphicComponent, [40](#)
- CCT, [11](#)
- CCT::Circuit, [15](#)
  - ~Circuit, [16](#)
  - addComponent, [17](#)
  - chords, [20](#)
  - Circuit, [16](#)
  - circuitMatrix, [21](#)
  - components, [21](#)
  - editComponent, [17](#), [18](#)
  - getCurrent, [18](#)
  - getVoltage, [18](#)
  - initialize, [19](#)
  - removeComponent, [19](#)
  - reset, [20](#)
  - Solve, [20](#)
  - updateComponents, [20](#)
- chords
  - CCT::Circuit, [20](#)
- Circuit
  - CCT::Circuit, [16](#)
- circuit
  - Diagram, [30](#)
- Circuit.cpp, [63](#)
- Circuit.h, [64](#)
- circuitMatrix
  - CCT::Circuit, [21](#)
- clicked
  - GraphicComponent, [36](#)
- clickedArea
  - GraphicComponent, [36](#)
- clickedControl
  - Diagram, [26](#)
- clickedStack
  - Diagram, [30](#)
- clickedVertex
  - GraphicComponent, [37](#)
- closeFile
  - MainWindow, [47](#)
- CMP, [11](#)
  - GROUND, [11](#)
  - RESISTOR, [11](#)
  - type, [11](#)
  - VCC, [11](#)
  - WIRE, [11](#)
- CMP::Component, [21](#)
  - ~Component, [22](#)
  - Component, [22](#)
  - current, [23](#)
  - getCurrent, [22](#)
  - getLabel, [22](#)
  - getNodes, [22](#)
  - getType, [22](#)
  - getVoltage, [23](#)
  - label, [23](#)
  - setCurrent, [23](#)
  - setLabel, [23](#)
  - setVoltage, [23](#)
  - voltage, [24](#)
  - vtxs, [24](#)
- CMP::Resistor, [56](#)
  - ~Resistor, [56](#)
  - getResistance, [57](#)
  - getType, [57](#)
  - resistance, [57](#)
  - Resistor, [56](#)
  - setCurrent, [57](#)
  - setResistance, [57](#)
  - setVoltage, [57](#)
- CMP::Vcc, [59](#)
  - ~Vcc, [60](#)
  - getType, [60](#)
  - setCurrent, [60](#)

- setVoltage, 60
  - Vcc, 60
- colNumber
  - NM::Matrix, 55
- Component
  - CMP::Component, 22
- Component.cpp, 64
- Component.h, 65
- components
  - CCT::Circuit, 21
- componentType
  - GraphicComponent, 40
- connections
  - Diagram, 30
- current
  - CMP::Component, 23
- cursorLocation
  - Diagram, 30
- Diagram, 24
  - circuit, 30
  - clickedControl, 26
  - clickedStack, 30
  - connections, 30
  - cursorLocation, 30
  - Diagram, 25
  - drawList, 30
  - edit, 26
  - editButton, 30
  - editMenu, 30
  - editMode, 26
  - fileName, 31
  - freeAllocatedMemory, 26
  - getFileName, 26
  - getPixMap, 26
  - getStatus, 27
  - initializeDiagram, 27
  - insert, 27
  - leftButtonActions, 27
  - leftButtonClicked, 27
  - load, 27
  - loadError, 27
  - mode, 31
  - modified, 28
  - mouseMoveEvent, 28
  - mousePressEvent, 28
  - paintEvent, 28
  - playButton, 31
  - query, 28
  - queryMode, 28
  - remove, 28
  - rightButtonActions, 29
  - rightButtonClicked, 29
  - save, 29
  - selectedButton, 31
  - selectedComponent, 31
  - selectedPrev, 31
  - setFileName, 29
  - setSelectedButton, 29
  - setStatus, 29
  - showEditDialog, 29
  - status, 31
  - wireCounter, 31
- Diagram.cpp, 66
- Diagram.h, 66
  - EDIT, 67
  - ERROR, 68
  - mode, 67
  - MODIFIED, 68
  - NONE, 68
  - OK, 68
  - QUERY, 67
  - RES180, 68
  - RES90, 68
  - sts, 67
  - typeOrientation, 68
  - UNSAVED, 68
  - VCC180, 68
  - VCC90, 68
- diagrams
  - MainWindow, 49
- draw
  - GraphicComponent, 37
- drawList
  - Diagram, 30
- drawRes180
  - MainWindow, 47
- drawRes90
  - MainWindow, 47
- drawVcc180
  - MainWindow, 48
- drawVcc90
  - MainWindow, 48
- edgeNumber
  - GRF::incidenceMatrix, 45
- EDIT
  - Diagram.h, 67
- edit
  - Diagram, 26
- editButton
  - Diagram, 30
- editComponent
  - CCT::Circuit, 17, 18
- editMenu
  - Diagram, 30
- editMode
  - Diagram, 26
- EquationSystem
  - NM::EquationSystem, 32
- ERROR
  - Diagram.h, 68
- fileMenu
  - MainWindow, 49
- fileName
  - Diagram, 31
- findPivot

NM::EquationSystem, 33  
 freeAllocatedMemory  
     Diagram, 26  
 gaussJordan  
     NM::EquationSystem, 33  
 gaussSeidel  
     NM::EquationSystem, 33  
 getBottom  
     GraphicComponent, 37  
 getBoundRect  
     GraphicComponent, 37  
 getCol  
     NM::Matrix, 53  
 getColNumber  
     NM::Matrix, 53  
 getConNum  
     GRF::incidenceMatrix, 43  
 getCurrent  
     CCT::Circuit, 18  
     CMP::Component, 22  
 getEdgeNumber  
     GRF::incidenceMatrix, 43  
 getEdges  
     GRF::incidenceMatrix, 43  
 getFileName  
     Diagram, 26  
 getHeight  
     GraphicComponent, 37  
 getLabel  
     CMP::Component, 22  
     GraphicComponent, 37  
 getLeft  
     GraphicComponent, 37  
 getLoop  
     GRF::incidenceMatrix, 44  
 getMap  
     GraphicComponent, 38  
 getNodes  
     CMP::Component, 22  
 getOrientation  
     GraphicComponent, 38  
 getPixMap  
     Diagram, 26  
 getResistance  
     CMP::Resistor, 57  
 getRight  
     GraphicComponent, 38  
 getRowNumber  
     NM::Matrix, 53  
 getSolution  
     NM::EquationSystem, 33  
 getSpanningTree  
     GRF::incidenceMatrix, 44  
 getStatus  
     Diagram, 27  
 getTop  
     GraphicComponent, 38  
 getType  
     CMP::Component, 22  
     CMP::Resistor, 57  
     CMP::Vcc, 60  
     GraphicComponent, 38  
     Resistor, 59  
     Vcc, 62  
 getValue  
     GraphicComponent, 38  
 getVertex  
     GRF::incidenceMatrix, 44  
 getVertex1  
     GraphicComponent, 38  
 getVertex1Point  
     GraphicComponent, 39  
 getVertex2  
     GraphicComponent, 39  
 getVertex2Point  
     GraphicComponent, 39  
 getVertexCon  
     GRF::incidenceMatrix, 44  
 getVertexNum  
     GraphicComponent, 39  
 getVertexNumber  
     GRF::adjacencyMatrix, 14  
     GRF::incidenceMatrix, 44  
 getVoltage  
     CCT::Circuit, 18  
     CMP::Component, 23  
 getWidth  
     GraphicComponent, 39  
 Graph.cpp, 68  
 Graph.h, 69  
 GraphicComponent, 34  
     boundRect, 40  
     clicked, 36  
     clickedArea, 36  
     clickedVertex, 37  
     componentType, 40  
     draw, 37  
     getBottom, 37  
     getBoundRect, 37  
     getHeight, 37  
     getLabel, 37  
     getLeft, 37  
     getMap, 38  
     getOrientation, 38  
     getRight, 38  
     getTop, 38  
     getType, 38  
     getValue, 38  
     getVertex1, 38  
     getVertex1Point, 39  
     getVertex2, 39  
     getVertex2Point, 39  
     getVertexNum, 39  
     getWidth, 39  
     GraphicComponent, 36  
     label, 40

- map, 40
- orientation, 40
- setValue, 39
- setVertex1, 39
- setVertex2, 40
- value, 40
- vertex1, 41
- vertex2, 41
- vertexArea1, 41
- vertexArea2, 41
- vertexNum, 41
- x, 41
- y, 41
- GraphicComponent.cpp, 69
- GraphicComponent.h, 70
  - HEIGHT, 71
  - HORIZONTAL, 71
  - orien, 71
  - VERTICAL, 71
  - WIDTH, 71
- GRF, 12
- GRF::adjacencyMatrix, 13
  - adjacencyMatrix, 13
  - adjMatrix, 14
  - getVertexNumber, 14
  - insertEdge, 14
  - print, 14
  - query, 14
  - removeVertex, 14
  - vertexNumber, 15
- GRF::incidenceMatrix, 42
  - addEdge, 43
  - edgeNumber, 45
  - getConNum, 43
  - getEdgeNumber, 43
  - getEdges, 43
  - getLoop, 44
  - getSpanningTree, 44
  - getVertex, 44
  - getVertexCon, 44
  - getVertexNumber, 44
  - incidenceMatrix, 43
  - inMatrix, 45
  - makeCon, 44
  - operator[], 44
  - print, 45
  - removeEdge, 45
  - removeVertex, 45
  - vertexNumber, 45
- GROUND
  - CMP, 11
- HEIGHT
  - GraphicComponent.h, 71
- helpMenu
  - MainWindow, 50
- HORIZONTAL
  - GraphicComponent.h, 71
- incidenceMatrix
  - GRF::incidenceMatrix, 43
- initialize
  - CCT::Circuit, 19
- initializeDiagram
  - Diagram, 27
- initializeMenu
  - MainWindow, 48
- initializeStatusBar
  - MainWindow, 48
- initializeTabs
  - MainWindow, 48
- initializeToolBar
  - MainWindow, 48
- initialTab
  - MainWindow, 50
- inMatrix
  - GRF::incidenceMatrix, 45
- insert
  - Diagram, 27
- insertEdge
  - GRF::adjacencyMatrix, 14
- label
  - CMP::Component, 23
  - GraphicComponent, 40
- leftButtonActions
  - Diagram, 27
- leftButtonClicked
  - Diagram, 27
- load
  - Diagram, 27
- loadError
  - Diagram, 27
- main
  - main.cpp, 72
- main.cpp, 72
  - main, 72
- mainBar
  - MainWindow, 50
- MainWindow, 46
  - closeFile, 47
  - diagrams, 49
  - drawRes180, 47
  - drawRes90, 47
  - drawVcc180, 48
  - drawVcc90, 48
  - fileMenu, 49
  - helpMenu, 50
  - initializeMenu, 48
  - initializeStatusBar, 48
  - initializeTabs, 48
  - initializeToolBar, 48
  - initialTab, 50
  - mainBar, 50
  - MainWindow, 47
  - newFile, 48
  - newFileAct, 50

- openFile, [48](#), [49](#)
- openFileAct, [50](#)
- prefAct, [50](#)
- preferences, [49](#)
- prefMenu, [50](#)
- saveFile, [49](#)
- saveFileAct, [50](#)
- saveFileAs, [49](#)
- saveFileAsAct, [51](#)
- setTabStatus, [49](#)
- statusBar, [51](#)
- tabs, [51](#)
- toolbar, [51](#)
- tutorialAct, [51](#)
- MainWindow.cpp, [72](#)
- MainWindow.h, [73](#)
- makeCon
  - GRF::incidenceMatrix, [44](#)
- map
  - GraphicComponent, [40](#)
- Matrix
  - NM::Matrix, [52](#)
- mode
  - Diagram, [31](#)
  - Diagram.h, [67](#)
- MODIFIED
  - Diagram.h, [68](#)
- modified
  - Diagram, [28](#)
- mouseMoveEvent
  - Diagram, [28](#)
- mousePressEvent
  - Diagram, [28](#)
- newFile
  - MainWindow, [48](#)
- newFileAct
  - MainWindow, [50](#)
- NM, [12](#)
- NM::EquationSystem, [32](#)
  - A, [34](#)
  - B, [34](#)
  - EquationSystem, [32](#)
  - findPivot, [33](#)
  - gaussJordan, [33](#)
  - gaussSeidel, [33](#)
  - getSolution, [33](#)
  - print, [33](#)
  - printSolution, [33](#)
  - sassenfeldCriteria, [33](#)
  - x, [34](#)
- NM::Matrix, [51](#)
  - Abs, [52](#)
  - colNumber, [55](#)
  - getCol, [53](#)
  - getColNumber, [53](#)
  - getRowNumber, [53](#)
  - Matrix, [52](#)
  - operator\*, [53](#)
  - operator\*=, [53](#)
  - operator+, [53](#)
  - operator+=, [53](#)
  - operator-, [54](#)
  - operator-=, [54](#)
  - operator=, [54](#)
  - operator[], [54](#)
  - print, [54](#)
  - realMatrix, [55](#)
  - rowNumber, [55](#)
  - swapLines, [55](#)
  - transpose, [55](#)
- NONE
  - Diagram.h, [68](#)
- Numeric.cpp, [74](#)
- Numeric.h, [74](#)
- OK
  - Diagram.h, [68](#)
- openFile
  - MainWindow, [48](#), [49](#)
- openFileAct
  - MainWindow, [50](#)
- operator\*
  - NM::Matrix, [53](#)
- operator\*=
  - NM::Matrix, [53](#)
- operator+
  - NM::Matrix, [53](#)
- operator+=
  - NM::Matrix, [53](#)
- operator-
  - NM::Matrix, [54](#)
- operator-=
  - NM::Matrix, [54](#)
- operator=
  - NM::Matrix, [54](#)
- operator[]
  - GRF::incidenceMatrix, [44](#)
  - NM::Matrix, [54](#)
- orien
  - GraphicComponent.h, [71](#)
- orientation
  - GraphicComponent, [40](#)
- paintEvent
  - Diagram, [28](#)
- playButton
  - Diagram, [31](#)
- prefAct
  - MainWindow, [50](#)
- preferences
  - MainWindow, [49](#)
- prefMenu
  - MainWindow, [50](#)
- print
  - GRF::adjacencyMatrix, [14](#)
  - GRF::incidenceMatrix, [45](#)
  - NM::EquationSystem, [33](#)

NM::Matrix, 54  
 printSolution  
   NM::EquationSystem, 33  
 QUERY  
   Diagram.h, 67  
 query  
   Diagram, 28  
   GRF::adjacencyMatrix, 14  
 queryMode  
   Diagram, 28  
 realMatrix  
   NM::Matrix, 55  
 remove  
   Diagram, 28  
 removeComponent  
   CCT::Circuit, 19  
 removeEdge  
   GRF::incidenceMatrix, 45  
 removeVertex  
   GRF::adjacencyMatrix, 14  
   GRF::incidenceMatrix, 45  
 RES180  
   Diagram.h, 68  
 RES90  
   Diagram.h, 68  
 resCounter  
   Resistor, 59  
 reset  
   CCT::Circuit, 20  
 resistance  
   CMP::Resistor, 57  
 RESISTOR  
   CMP, 11  
 Resistor, 58  
   CMP::Resistor, 56  
   getType, 59  
   resCounter, 59  
   Resistor, 58  
 rightButtonActions  
   Diagram, 29  
 rightButtonClicked  
   Diagram, 29  
 rowNumber  
   NM::Matrix, 55  
 sassensfeldCriteria  
   NM::EquationSystem, 33  
 save  
   Diagram, 29  
 saveFile  
   MainWindow, 49  
 saveFileAct  
   MainWindow, 50  
 saveFileAs  
   MainWindow, 49  
 saveFileAsAct  
   MainWindow, 51  
 selectedButton  
   Diagram, 31  
 selectedComponent  
   Diagram, 31  
 selectedPrev  
   Diagram, 31  
 setCurrent  
   CMP::Component, 23  
   CMP::Resistor, 57  
   CMP::Vcc, 60  
 setFileName  
   Diagram, 29  
 setLabel  
   CMP::Component, 23  
 setResistance  
   CMP::Resistor, 57  
 setSelectedButton  
   Diagram, 29  
 setStatus  
   Diagram, 29  
 setTabStatus  
   MainWindow, 49  
 setValue  
   GraphicComponent, 39  
 setVertex1  
   GraphicComponent, 39  
 setVertex2  
   GraphicComponent, 40  
 setVoltage  
   CMP::Component, 23  
   CMP::Resistor, 57  
   CMP::Vcc, 60  
 showEditDialog  
   Diagram, 29  
 Solve  
   CCT::Circuit, 20  
 status  
   Diagram, 31  
 statusBar  
   MainWindow, 51  
 sts  
   Diagram.h, 67  
 swapLines  
   NM::Matrix, 55  
 tabs  
   MainWindow, 51  
 toolbar  
   MainWindow, 51  
 transpose  
   NM::Matrix, 55  
 tutorialAct  
   MainWindow, 51  
 type  
   CMP, 11  
 typeOrientation  
   Diagram.h, 68  
 UNSAVED

- Diagram.h, 68
- updateComponents
  - CCT::Circuit, 20
- value
  - GraphicComponent, 40
- VCC
  - CMP, 11
- Vcc, 61
  - CMP::Vcc, 60
  - getType, 62
  - Vcc, 61
  - vccCounter, 62
- VCC180
  - Diagram.h, 68
- VCC90
  - Diagram.h, 68
- vccCounter
  - Vcc, 62
- vertex1
  - GraphicComponent, 41
- vertex2
  - GraphicComponent, 41
- vertexArea1
  - GraphicComponent, 41
- vertexArea2
  - GraphicComponent, 41
- vertexNum
  - GraphicComponent, 41
- vertexNumber
  - GRF::adjacencyMatrix, 15
  - GRF::incidenceMatrix, 45
- VERTICAL
  - GraphicComponent.h, 71
- voltage
  - CMP::Component, 24
- vtxs
  - CMP::Component, 24
- WIDTH
  - GraphicComponent.h, 71
- WIRE
  - CMP, 11
- wireCounter
  - Diagram, 31
- x
  - GraphicComponent, 41
  - NM::EquationSystem, 34
- y
  - GraphicComponent, 41