

Laboratorio\_3

1.0

Generated by Doxygen 1.8.8

Wed Sep 28 2016 00:17:30



# Contents

<b>1</b>	<b>Programación genérica en C++</b>	<b>1</b>
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	Calculadora< data > Class Template Reference . . . . .	7
4.1.1	Constructor & Destructor Documentation . . . . .	7
4.1.1.1	Calculadora . . . . .	7
4.1.1.2	~Calculadora . . . . .	7
4.1.2	Member Function Documentation . . . . .	7
4.1.2.1	add . . . . .	7
4.1.2.2	div . . . . .	7
4.1.2.3	mul . . . . .	8
4.1.2.4	print . . . . .	8
4.1.2.5	sub . . . . .	8
4.2	Fraccion Class Reference . . . . .	8
4.2.1	Constructor & Destructor Documentation . . . . .	8
4.2.1.1	Fraccion . . . . .	8
4.2.1.2	Fraccion . . . . .	8
4.2.1.3	~Fraccion . . . . .	8
4.2.2	Member Function Documentation . . . . .	9
4.2.2.1	operator* . . . . .	9
4.2.2.2	operator+ . . . . .	9
4.2.2.3	operator- . . . . .	9
4.2.2.4	operator/ . . . . .	9
4.2.2.5	operator~ . . . . .	10
4.2.3	Member Data Documentation . . . . .	10
4.2.3.1	den . . . . .	10

4.2.3.2	num	10
4.3	Matriz Class Reference	10
4.3.1	Constructor & Destructor Documentation	10
4.3.1.1	Matriz	10
4.3.1.2	Matriz	11
4.3.1.3	~Matriz	11
4.3.2	Member Function Documentation	11
4.3.2.1	operator*	11
4.3.2.2	operator+	11
4.3.2.3	operator-	11
4.3.2.4	operator/	11
4.3.2.5	operator~	12
4.3.3	Member Data Documentation	12
4.3.3.1	m	12
4.3.3.2	matrix	12
4.3.3.3	n	12
4.4	Polinomio Class Reference	12
4.4.1	Constructor & Destructor Documentation	13
4.4.1.1	Polinomio	13
4.4.1.2	Polinomio	13
4.4.1.3	~Polinomio	13
4.4.2	Member Function Documentation	13
4.4.2.1	operator*	13
4.4.2.2	operator+	13
4.4.2.3	operator-	14
4.4.2.4	operator/	15
4.4.2.5	operator~	15
4.4.3	Member Data Documentation	15
4.4.3.1	coef	15
4.4.3.2	tam	15
4.4.3.3	var	15
<b>5</b>	<b>File Documentation</b>	<b>17</b>
5.1	code/Calculadora.h File Reference	17
5.2	code/Fraccion.cpp File Reference	17
5.3	code/Fraccion.h File Reference	17
5.4	code/main.cpp File Reference	17
5.4.1	Function Documentation	18
5.4.1.1	main	18
5.5	code/Matriz.cpp File Reference	18

---

5.6	<a href="#">code/Matriz.h File Reference</a>	18
5.7	<a href="#">code/Polinomio.cpp File Reference</a>	18
5.8	<a href="#">code/Polinomio.h File Reference</a>	18
<b>Index</b>		<b>19</b>



# Chapter 1

## Programación genérica en C++

### Author

Dunia Barahona [s4si@hotmail.com](mailto:s4si@hotmail.com)

### Date

27 de setiembre de 2016

### Version

1.0

### Descripción

Clase emplantillada [Calculadora](#), en la cual se pueden sumar, restar, multiplicar o dividir objetos del tipo [Frac-cion](#), [Polinomio](#) y [Matriz](#). Todas estas clases tiene su respectivo archivo de encabezados y estan implementadas en el [main](#).





## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">Calculadora&lt; data &gt;</a>	7
<a href="#">Fraccion</a>	8
<a href="#">Matriz</a>	10
<a href="#">Polinomio</a>	12



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

code/ <a href="#">Calculadora.h</a>	17
code/ <a href="#">Fraccion.cpp</a>	17
code/ <a href="#">Fraccion.h</a>	17
code/ <a href="#">main.cpp</a>	17
code/ <a href="#">Matriz.cpp</a>	18
code/ <a href="#">Matriz.h</a>	18
code/ <a href="#">Polinomio.cpp</a>	18
code/ <a href="#">Polinomio.h</a>	18



## Chapter 4

# Class Documentation

### 4.1 Calculadora< data > Class Template Reference

```
#include <Calculadora.h>
```

#### Public Member Functions

- [Calculadora](#) ()
- [~Calculadora](#) ()
- data [add](#) (data d1, const data d2)
- data [sub](#) (data d1, const data d2)
- data [mul](#) (data d1, const data d2)
- data [div](#) (data d1, const data d2)
- void [print](#) (data d)

#### 4.1.1 Constructor & Destructor Documentation

4.1.1.1 `template<typename data > Calculadora< data >::Calculadora ( ) [inline]`

Constructor de la clase [Calculadora](#).

4.1.1.2 `template<typename data > Calculadora< data >::~~Calculadora ( ) [inline]`

Destructor de la clase [Calculadora](#).

#### 4.1.2 Member Function Documentation

4.1.2.1 `template<typename data > data Calculadora< data >::add ( data d1, const data d2 ) [inline]`

Método suma de la clase [Calculadora](#).

4.1.2.2 `template<typename data > data Calculadora< data >::div ( data d1, const data d2 ) [inline]`

Método división de la clase [Calculadora](#).

4.1.2.3 `template<typename data > data Calculadora< data >::mul ( data d1, const data d2 )` `[inline]`

Método multiplicación de la clase [Calculadora](#).

4.1.2.4 `template<typename data > void Calculadora< data >::print ( data d )` `[inline]`

Método que imprime.

4.1.2.5 `template<typename data > data Calculadora< data >::sub ( data d1, const data d2 )` `[inline]`

Método resta de la clase [Calculadora](#).

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

- `code/Calculadora.h`

## 4.2 Fraccion Class Reference

```
#include <Fraccion.h>
```

### Public Member Functions

- [Fraccion](#) ()
- [Fraccion](#) (double n, double d)
- [~Fraccion](#) ()
- [Fraccion operator+](#) (const [Fraccion](#) f2)
- [Fraccion operator-](#) (const [Fraccion](#) f2)
- [Fraccion operator\\*](#) (const [Fraccion](#) f2)
- [Fraccion operator/](#) (const [Fraccion](#) f2)
- void [operator~](#) ()

### Public Attributes

- double [num](#)  
*Numerador de la fracción.*
- double [den](#)  
*Denominador de la fracción.*

### 4.2.1 Constructor & Destructor Documentation

4.2.1.1 `Fraccion::Fraccion ( )`

El **constructor** es una función que se llama igual que la clase

4.2.1.2 `Fraccion::Fraccion ( double n, double d )`

Constructor que recibe los atributos como parámetros

4.2.1.3 `Fraccion::~~Fraccion ( )`

Destructor, sirve para destruir un objeto de la clase

## 4.2.2 Member Function Documentation

### 4.2.2.1 Fraccion Fraccion::operator\* ( const Fraccion f2 )

Sobrecarga del operador \*

Método que pertenece a la clase [Fraccion](#) y devuelve un objeto de tipo [Fraccion](#)

Parameters

<i>f2</i>	objeto de tipo <a href="#">Fraccion</a>
-----------	---

Returns

Multiplicación de las fracciones dadas

### 4.2.2.2 Fraccion Fraccion::operator+ ( const Fraccion f2 )

Sobrecarga del operador +

Método que pertenece a la clase [Fraccion](#) y devuelve un objeto de tipo [Fraccion](#)

Parameters

<i>f2</i>	objeto de tipo <a href="#">Fraccion</a>
-----------	---

Returns

Suma de las fracciones dadas

### 4.2.2.3 Fraccion Fraccion::operator- ( const Fraccion f2 )

Sobrecarga del operador -

Método que pertenece a la clase [Fraccion](#) y devuelve un objeto de tipo [Fraccion](#)

Parameters

<i>f2</i>	objeto de tipo <a href="#">Fraccion</a>
-----------	---

Returns

Resta de las fracciones dadas

### 4.2.2.4 Fraccion Fraccion::operator/ ( const Fraccion f2 )

Sobrecarga del operador /

Método que pertenece a la clase [Fraccion](#) y devuelve un objeto de tipo [Fraccion](#)

Parameters

<i>f2</i>	objeto de tipo <a href="#">Fraccion</a>
-----------	---

Returns

División de las fracciones dadas

#### 4.2.2.5 void Fraccion::operator~ ( )

Sobrecarga del operador ~

Método que pertenece a la clase [Fraccion](#)

Imprime la fracción especificada

### 4.2.3 Member Data Documentation

#### 4.2.3.1 double Fraccion::den

Denominador de la fracción.

#### 4.2.3.2 double Fraccion::num

Numerador de la fracción.

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

- code/[Fraccion.h](#)
- code/[Fraccion.cpp](#)

## 4.3 Matriz Class Reference

```
#include <Matriz.h>
```

### Public Member Functions

- [Matriz](#) ()
- [Matriz](#) (int [m](#), int [n](#), double \*\*matriz)
- [~Matriz](#) ()
- [Matriz operator+](#) (const [Matriz](#) f2)
- [Matriz operator-](#) (const [Matriz](#) f2)
- [Matriz operator\\*](#) (const [Matriz](#) f2)
- [Matriz operator/](#) (const [Matriz](#) f2)
- void [operator~](#) ()

### Public Attributes

- int [m](#)  
*Filas de la matriz.*
- int [n](#)  
*Columnas de la matriz.*
- double \*\* [matrix](#)  
*Coeficientes de la matriz.*

### 4.3.1 Constructor & Destructor Documentation

#### 4.3.1.1 Matriz::Matriz ( )

Constructor simple



#### 4.3.1.2 `Matriz::Matriz ( int m, int n, double ** matriz )`

Constructor con atributos

#### 4.3.1.3 `Matriz::~~Matriz ( )`

Destructor

### 4.3.2 Member Function Documentation

#### 4.3.2.1 `Matriz Matriz::operator* ( const Matriz f2 )`

Sobrecarga del operador `*`

Método que pertenece a la clase [Matriz](#) y que devuelve un objeto de tipo [Matriz](#)

Parameters

<code>m2</code>	objeto de tipo <a href="#">Matriz</a>
-----------------	---------------------------------------

Returns

Multiplicación de las matrices dadas

#### 4.3.2.2 `Matriz Matriz::operator+ ( const Matriz f2 )`

Sobrecarga del operador `+`

Método que pertenece a la clase [Matriz](#) y que devuelve un objeto de tipo [Matriz](#)

Parameters

<code>m2</code>	objeto de tipo <a href="#">Matriz</a>
-----------------	---------------------------------------

Returns

Suma de las matrices dadas

#### 4.3.2.3 `Matriz Matriz::operator- ( const Matriz f2 )`

Sobrecarga del operador `-`

Método que pertenece a la clase [Matriz](#) y que devuelve un objeto de tipo [Matriz](#)

Parameters

<code>m2</code>	objeto de tipo <a href="#">Matriz</a>
-----------------	---------------------------------------

Returns

Resta de las matrices dadas

#### 4.3.2.4 `Matriz Matriz::operator/ ( const Matriz f2 )`

Sobrecarga del operador `/`

Método que pertenece a la clase [Matriz](#) y que devuelve un objeto de tipo [Matriz](#).

**Parameters**

<i>m2</i>	objeto de tipo <a href="#">Matriz</a>
-----------	---------------------------------------

**Returns**

División de las matrices dadas

**4.3.2.5 void Matriz::operator~ ( )**

Sobrecarga del operador ~

Método que pertenece a la clase [Matriz](#)

Imprime la matriz especificada

**4.3.3 Member Data Documentation****4.3.3.1 int Matriz::m**

Filas de la matriz.

**4.3.3.2 double\*\* Matriz::matrix**

Coeficientes de la matriz.

**4.3.3.3 int Matriz::n**

Columnas de la matriz.

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

- [code/Matriz.h](#)
- [code/Matriz.cpp](#)

**4.4 Polinomio Class Reference**

```
#include <Polinomio.h>
```

**Public Member Functions**

- [Polinomio \(\)](#)
- [Polinomio \(int tam, char var, double \\*coef\)](#)
- [~Polinomio \(\)](#)
- [Polinomio operator+ \(const Polinomio p2\)](#)
- [Polinomio operator- \(const Polinomio p2\)](#)
- [Polinomio operator\\* \(const Polinomio p2\)](#)
- [Polinomio operator/ \(const Polinomio p2\)](#)
- void [operator~ \(\)](#)

## Public Attributes

- int [tam](#)  
*Tamaño del polinomio.*
- char [var](#)  
*Variable del polinomio.*
- double \* [coef](#)  
*Coefficientes numéricos del polinomio.*

## 4.4.1 Constructor & Destructor Documentation

### 4.4.1.1 Polinomio::Polinomio ( )

Constructor simple

### 4.4.1.2 Polinomio::Polinomio ( int *tam*, char *var*, double \* *coef* )

Constructor con atributos

### 4.4.1.3 Polinomio::~~Polinomio ( )

Destructor

## 4.4.2 Member Function Documentation

### 4.4.2.1 Polinomio Polinomio::operator\* ( const Polinomio *p2* )

Sobrecarga del operador \*

Método que pertenece a la clase [Polinomio](#) y que devuelve un objeto de tipo [Polinomio](#)

Parameters

<i>p2</i>	objeto de tipo <a href="#">Polinomio</a>
-----------	--

Returns

Multiplicación de los polinomios dados

### 4.4.2.2 Polinomio Polinomio::operator+ ( const Polinomio *p2* )

Sobrecarga del operador +

Método que pertenece a la clase [Polinomio](#) y que devuelve un objeto de tipo [Polinomio](#)

Parameters

<i>p2</i>	objeto de tipo <a href="#">Polinomio</a>
-----------	--

Returns

Suma de los polinomios dados

#### 4.4.2.3 `Polinomio` `Polinomio::operator-` ( `const Polinomio p2` )

Sobrecarga del operador -

Método que pertenece a la clase `Polinomio` y que devuelve un objeto de tipo `Polinomio`

## Parameters

<i>p2</i>	objeto de tipo <a href="#">Polinomio</a>
-----------	--

## Returns

Resta de los polinomios dados

**4.4.2.4 Polinomio Polinomio::operator/ ( const Polinomio *p2* )**

Sobrecarga del operador /

Método que pertenece a la clase [Polinomio](#) y que devuelve un objeto de tipo [Polinomio](#)

## Parameters

<i>p2</i>	objeto de tipo <a href="#">Polinomio</a>
-----------	--

## Returns

División de los polinomios dados

**4.4.2.5 void Polinomio::operator~ ( )**

Sobrecarga del operador ~

Método que pertenece a la clase [Polinomio](#)

Imprime el polinomio especificado

**4.4.3 Member Data Documentation****4.4.3.1 double\* Polinomio::coef**

Coeficientes numéricos del polinomio.

**4.4.3.2 int Polinomio::tam**

Tamaño del polinomio.

**4.4.3.3 char Polinomio::var**

Variable del polinomio.

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

- code/[Polinomio.h](#)
- code/[Polinomio.cpp](#)



## Chapter 5

# File Documentation

### 5.1 code/Calculadora.h File Reference

```
#include "Fraccion.h"  
#include "Polinomio.h"  
#include "Matriz.h"
```

#### Classes

- class [Calculadora](#)< data >

### 5.2 code/Fraccion.cpp File Reference

```
#include "Fraccion.h"
```

### 5.3 code/Fraccion.h File Reference

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

#### Classes

- class [Fraccion](#)

### 5.4 code/main.cpp File Reference

```
#include "Fraccion.h"  
#include "Polinomio.h"  
#include "Matriz.h"  
#include "Calculadora.h"
```

## Functions

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

### 5.4.1 Function Documentation

#### 5.4.1.1 int main ( int argc, char \*\* argv )

Llamado a la clase emplantillada [Calculadora](#):

```
1 Calculadora<Fraccion>* c1= new Calculadora<Fraccion>();
2 Calculadora<Matriz>* c2= new Calculadora<Matriz>();
3 Calculadora<Polinomio>* c3= new Calculadora<Polinomio>();
```

Se suman, restan, multiplican y dividen fracciones, polinomios y matrices.

## 5.5 code/Matriz.cpp File Reference

```
#include "Matriz.h"
```

## 5.6 code/Matriz.h File Reference

```
#include <cstdlib>
#include <iostream>
#include "string"
```

## Classes

- class [Matriz](#)

## 5.7 code/Polinomio.cpp File Reference

```
#include "Polinomio.h"
```

## 5.8 code/Polinomio.h File Reference

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

## Classes

- class [Polinomio](#)



# Index

add  
    Calculadora, 7

Calculadora  
    add, 7  
    Calculadora, 7  
    div, 7  
    mul, 7  
    print, 8  
    sub, 8  
Calculadora < data >, 7  
coef  
    Polinomio, 15

den  
    Fraccion, 10

div  
    Calculadora, 7

Fraccion, 8  
    den, 10  
    Fraccion, 8  
    num, 10  
    operator\*, 9  
    operator+, 9  
    operator-, 9  
    operator/, 9  
    operator~, 9

m  
    Matriz, 12

matrix  
    Matriz, 12

Matriz, 10  
    m, 12  
    matrix, 12  
    Matriz, 10  
    n, 12  
    operator\*, 11  
    operator+, 11  
    operator-, 11  
    operator/, 11  
    operator~, 12

mul  
    Calculadora, 7

n  
    Matriz, 12

num  
    Fraccion, 10

operator\*  
    Fraccion, 9  
    Matriz, 11  
    Polinomio, 13

operator+  
    Fraccion, 9  
    Matriz, 11  
    Polinomio, 13

operator-  
    Fraccion, 9  
    Matriz, 11  
    Polinomio, 13

operator/  
    Fraccion, 9  
    Matriz, 11  
    Polinomio, 15

operator~  
    Fraccion, 9  
    Matriz, 12  
    Polinomio, 15

Polinomio, 12  
    coef, 15  
    operator\*, 13  
    operator+, 13  
    operator-, 13  
    operator/, 15  
    operator~, 15  
    Polinomio, 13  
    tam, 15  
    var, 15

print  
    Calculadora, 8

sub  
    Calculadora, 8

tam  
    Polinomio, 15

var  
    Polinomio, 15