

My Complex - Projeto de Engenharia II

Generated by Doxygen 1.9.1

1 My Complex	1
1.1 Integrantes	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 complexo Struct Reference	7
4.1.1 Member Data Documentation	7
4.1.1.1 img	7
4.1.1.2 real	7
5 File Documentation	9
5.1 main.c File Reference	9
5.1.1 Function Documentation	9
5.1.1.1 main()	10
5.2 mainpage.md File Reference	10
5.3 matrizes.c File Reference	10
5.3.1 Function Documentation	11
5.3.1.1 allocateComplexMatrix()	11
5.3.1.2 conjugada()	11
5.3.1.3 hermetiana()	11
5.3.1.4 multcomp()	11
5.3.1.5 printComplex()	11
5.3.1.6 produto_interno()	12
5.3.1.7 produto_matricial()	12
5.3.1.8 produto_por_escalar()	12
5.3.1.9 soma()	12
5.3.1.10 soma_complexo()	12
5.3.1.11 subtracao()	13
5.3.1.12 teste_conjugada()	13
5.3.1.13 teste_hermetiana()	13
5.3.1.14 teste_produto_interno()	13
5.3.1.15 teste_produto_matricial()	13
5.3.1.16 teste_produto_por_escalar()	13
5.3.1.17 teste_soma()	14
5.3.1.18 teste_subtracao()	14
5.3.1.19 teste_todos()	14
5.3.1.20 teste_transposta()	14
5.3.1.21 transposta()	14
5.4 matrizes.h File Reference	15

5.4.1 Typedef Documentation	16
5.4.1.1 complexo	16
5.4.2 Function Documentation	16
5.4.2.1 allocateComplexMatrix()	16
5.4.2.2 conjugada()	16
5.4.2.3 hermetiana()	17
5.4.2.4 LiberarMatriz()	17
5.4.2.5 multcomp()	17
5.4.2.6 printComplex()	17
5.4.2.7 produto_interno()	17
5.4.2.8 produto_matricial()	17
5.4.2.9 soma()	18
5.4.2.10 soma_complexo()	18
5.4.2.11 subtracao()	18
5.4.2.12 teste_conjugada()	18
5.4.2.13 teste_hermetiana()	18
5.4.2.14 teste_produto_interno()	18
5.4.2.15 teste_produto_matricial()	19
5.4.2.16 teste_soma()	19
5.4.2.17 teste_subtracao()	19
5.4.2.18 teste_todos()	19
5.4.2.19 teste_transposta()	19
5.4.2.20 transposta()	19
Index	21

Chapter 1

My Complex

Projeto feito para a disciplina de Projeto de Engenharia ministrada no segundo semestre na Faculdade de Computação e Telecomunicações da Universidade Federal do Pará. O objetivo deste projeto é desenvolver uma biblioteca em C contendo funções que implementam operações matriciais. Essas funções serão testadas e validadas, levando em consideração as dimensões dos operandos. O arquivo "matrizes.c" conterà a implementação dessas funções, enquanto outro arquivo C terá uma função "main" para realizar testes em cada uma delas. A biblioteca resultante será utilizada em atividades futuras da disciplina de Projeto de Engenharia II.

Para mais informações, visite <https://github.com/jvictorferreira3301/Projeto-de-Engenharia-II>.

1.1 Integrantes

- João Victor Ferreira
-
-
-

Chapter 2

Class Index

2.1 Class List

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

complexo	7
------------------------------------	-------------------

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

main.c	9
matrizes.c	10
matrizes.h	15

Chapter 4

Class Documentation

4.1 `complexo` Struct Reference

```
#include <matrizes.h>
```

Public Attributes

- float [real](#)
- float [img](#)

4.1.1 Member Data Documentation

4.1.1.1 `img`

```
float complexo::img
```

4.1.1.2 `real`

```
float complexo::real
```

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

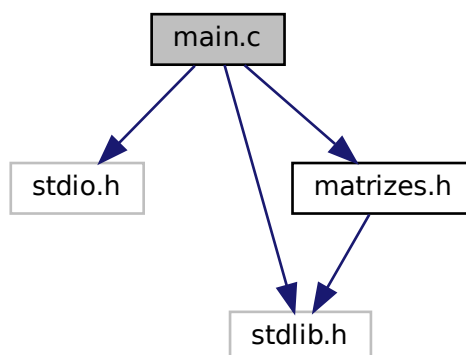
- [matrizes.h](#)

Chapter 5

File Documentation

5.1 main.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "matrizes.h"
Include dependency graph for main.c:
```



Functions

- int `main` (void)

5.1.1 Function Documentation

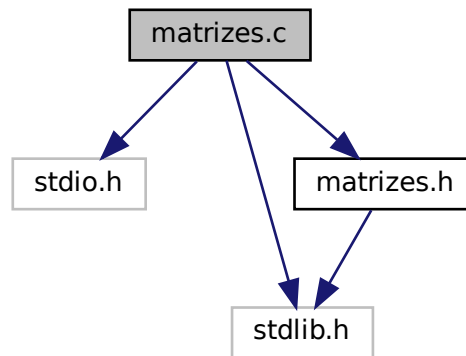
5.1.1.1 main()

```
int main (  
    void )
```

5.2 mainpage.md File Reference

5.3 matrizes.c File Reference

```
#include <stdio.h>  
#include <stdlib.h>  
#include "matrizes.h"  
Include dependency graph for matrizes.c:
```



Functions

- `complexo ** allocateComplexMatrix (int linhas, int colunas)`
- `void printComplex (complexo c)`
- `complexo soma_complexo (complexo c1, complexo c2)`
- `complexo ** conjugada (complexo **mtx, int linhas, int colunas)`
- `void teste_conjugada (void)`
- `complexo produto_interno (complexo **v1, complexo **v2, int linhas, int colunas)`
- `void teste_produto_interno (void)`
- `complexo ** transposta (complexo **matriz, int linhas, int colunas)`
- `complexo ** hermetiana (complexo **mtx, int linhas, int colunas)`
- `void teste_hermetiana (void)`
- `complexo ** soma (complexo **c2, complexo **c1, int linhas, int colunas)`
- `complexo ** subtracao (complexo **c1, complexo **c2, int linhas, int colunas)`
- `complexo ** produto_matricial (complexo **mtx_a, complexo **mtx_b, int linhas, int colunas)`
- `complexo ** produto_por_escalar (complexo **mtx, int linhas, int colunas, int k)`
- `void teste_produto_por_escalar (void)`
- `void teste_produto_matricial (void)`

- [complexo multcomp](#) ([complexo](#) c, [complexo](#) c1)
- void [teste_transposta](#) (void)
- void [teste_soma](#) (void)
- void [teste_subtracao](#) (void)
- void [teste_todos](#) (void)

5.3.1 Function Documentation

5.3.1.1 allocateComplexMatrix()

```
complexo** allocateComplexMatrix (  
    int linhas,  
    int colunas )
```

5.3.1.2 conjugada()

```
complexo** conjugada (  
    complexo ** mtx,  
    int linhas,  
    int colunas )
```

5.3.1.3 hermetiana()

```
complexo** hermetiana (  
    complexo ** mtx,  
    int linhas,  
    int colunas )
```

5.3.1.4 multcomp()

```
complexo multcomp (  
    complexo c,  
    complexo c1 )
```

5.3.1.5 printComplex()

```
void printComplex (  
    complexo c )
```

5.3.1.6 produto_interno()

```
complexo produto_interno (
    complexo ** v1,
    complexo ** v2,
    int linhas,
    int colunas )
```

5.3.1.7 produto_matricial()

```
complexo** produto_matricial (
    complexo ** mtx_a,
    complexo ** mtx_b,
    int linhas,
    int colunas )
```

5.3.1.8 produto_por_escalar()

```
complexo** produto_por_escalar (
    complexo ** mtx,
    int linhas,
    int colunas,
    int k )
```

5.3.1.9 soma()

```
complexo** soma (
    complexo ** c2,
    complexo ** c1,
    int linhas,
    int colunas )
```

5.3.1.10 soma_complexo()

```
complexo soma_complexo (
    complexo c1,
    complexo c2 )
```


5.3.1.11 subtracao()

```
complexo** subtracao (
    complexo ** c1,
    complexo ** c2,
    int linhas,
    int colunas )
```

5.3.1.12 teste_conjugada()

```
void teste_conjugada (
    void )
```

5.3.1.13 teste_hermetiana()

```
void teste_hermetiana (
    void )
```

5.3.1.14 teste_produto_interno()

```
void teste_produto_interno (
    void )
```

5.3.1.15 teste_produto_matricial()

```
void teste_produto_matricial (
    void )
```

5.3.1.16 teste_produto_por_escalar()

```
void teste_produto_por_escalar (
    void )
```

5.3.1.17 teste_soma()

```
void teste_soma (
    void )
```

5.3.1.18 teste_subtracao()

```
void teste_subtracao (
    void )
```

5.3.1.19 teste_todos()

```
void teste_todos (
    void )
```

5.3.1.20 teste_transposta()

```
void teste_transposta (
    void )
```

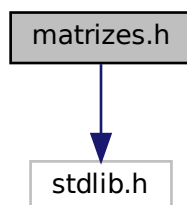
5.3.1.21 transposta()

```
complexo** transposta (
    complexo ** matriz,
    int linhas,
    int colunas )
```

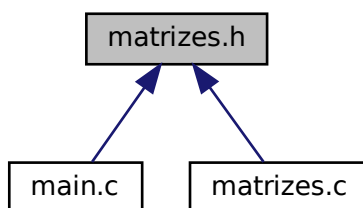
5.4 matrizes.h File Reference

```
#include <stdlib.h>
```

Include dependency graph for matrizes.h:



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



Classes

- struct [complexo](#)

Typedefs

- typedef struct [complexo](#) [complexo](#)

Functions

- [complexo](#) [soma_complexo](#) ([complexo](#) c1, [complexo](#) c2)
- [complexo](#) ** [allocateComplexMatrix](#) (int linhas, int colunas)
- void [LiberarMatriz](#) ([complexo](#) **mtx, int linhas)
- [complexo](#) ** [transposta](#) ([complexo](#) **c, int linhas, int colunas)
- void [printComplex](#) ([complexo](#) c)

- `complexo multcomp (complexo c, complexo c1)`
- `complexo ** soma (complexo **c, complexo **c1, int linhas, int colunas)`
- `complexo ** subtracao (complexo **c, complexo **c2, int linhas, int colunas)`
- `complexo ** produto_matricial (complexo **mtx_a, complexo **mtx_b, int linhas, int colunas)`
- `complexo produto_interno (complexo **v1, complexo **v2, int linhas, int colunas)`
- `complexo ** conjugada (complexo **mtx, int linhas, int colunas)`
- `complexo ** hermetiana (complexo **mtx, int linhas, int colunas)`
- `void teste_conjugada (void)`
- `void teste_transposta (void)`
- `void teste_hermetiana (void)`
- `void teste_soma (void)`
- `void teste_subtracao (void)`
- `void teste_todos (void)`
- `void teste_produto_interno (void)`
- `void teste_produto_matricial (void)`

5.4.1 Typedef Documentation

5.4.1.1 complexo

```
typedef struct complexo complexo
```

5.4.2 Function Documentation

5.4.2.1 allocateComplexMatrix()

```
complexo** allocateComplexMatrix (  
    int linhas,  
    int colunas )
```

5.4.2.2 conjugada()

```
complexo** conjugada (  
    complexo ** mtx,  
    int linhas,  
    int colunas )
```

5.4.2.3 hermetiana()

```
complexo** hermetiana (
    complexo ** mtx,
    int linhas,
    int colunas )
```

5.4.2.4 LiberarMatriz()

```
void LiberarMatriz (
    complexo ** mtx,
    int linhas )
```

5.4.2.5 multcomp()

```
complexo multcomp (
    complexo c,
    complexo c1 )
```

5.4.2.6 printComplex()

```
void printComplex (
    complexo c )
```

5.4.2.7 produto_interno()

```
complexo produto_interno (
    complexo ** v1,
    complexo ** v2,
    int linhas,
    int colunas )
```

5.4.2.8 produto_matricial()

```
complexo** produto_matricial (
    complexo ** mtx_a,
    complexo ** mtx_b,
    int linhas,
    int colunas )
```

5.4.2.9 soma()

```
complexo** soma (
    complexo ** c,
    complexo ** c1,
    int linhas,
    int colunas )
```

5.4.2.10 soma_complexo()

```
complexo soma_complexo (
    complexo c1,
    complexo c2 )
```

5.4.2.11 subtracao()

```
complexo** subtracao (
    complexo ** c,
    complexo ** c2,
    int linhas,
    int colunas )
```

5.4.2.12 teste_conjugada()

```
void teste_conjugada (
    void )
```

5.4.2.13 teste_hermetiana()

```
void teste_hermetiana (
    void )
```

5.4.2.14 teste_produto_interno()

```
void teste_produto_interno (
    void )
```

5.4.2.15 teste_produto_matricial()

```
void teste_produto_matricial (
    void )
```

5.4.2.16 teste_soma()

```
void teste_soma (
    void )
```

5.4.2.17 teste_subtracao()

```
void teste_subtracao (
    void )
```

5.4.2.18 teste_todos()

```
void teste_todos (
    void )
```

5.4.2.19 teste_transposta()

```
void teste_transposta (
    void )
```

5.4.2.20 transposta()

```
complexo** transposta (
    complexo ** c,
    int linhas,
    int colunas )
```


Index

- allocateComplexMatrix
 - matrizes.c, [11](#)
 - matrizes.h, [16](#)
- complexo, [7](#)
 - img, [7](#)
 - matrizes.h, [16](#)
 - real, [7](#)
- conjugada
 - matrizes.c, [11](#)
 - matrizes.h, [16](#)
- hermetiana
 - matrizes.c, [11](#)
 - matrizes.h, [16](#)
- img
 - complexo, [7](#)
- LiberarMatriz
 - matrizes.h, [17](#)
- main
 - main.c, [9](#)
- main.c, [9](#)
 - main, [9](#)
- mainpage.md, [10](#)
- matrizes.c, [10](#)
 - allocateComplexMatrix, [11](#)
 - conjugada, [11](#)
 - hermetiana, [11](#)
 - multcomp, [11](#)
 - printComplex, [11](#)
 - produto_interno, [11](#)
 - produto_matricial, [12](#)
 - produto_por_escalar, [12](#)
 - soma, [12](#)
 - soma_complexo, [12](#)
 - subtracao, [12](#)
 - teste_conjugada, [13](#)
 - teste_hermetiana, [13](#)
 - teste_produto_interno, [13](#)
 - teste_produto_matricial, [13](#)
 - teste_produto_por_escalar, [13](#)
 - teste_soma, [13](#)
 - teste_subtracao, [14](#)
 - teste_todos, [14](#)
 - teste_transposta, [14](#)
 - transposta, [14](#)
- matrizes.h, [15](#)
 - allocateComplexMatrix, [16](#)
- complexo, [16](#)
- conjugada, [16](#)
- hermetiana, [16](#)
- LiberarMatriz, [17](#)
- multcomp, [17](#)
- printComplex, [17](#)
- produto_interno, [17](#)
- produto_matricial, [17](#)
- soma, [17](#)
- soma_complexo, [18](#)
- subtracao, [18](#)
- teste_conjugada, [18](#)
- teste_hermetiana, [18](#)
- teste_produto_interno, [18](#)
- teste_produto_matricial, [18](#)
- teste_soma, [19](#)
- teste_subtracao, [19](#)
- teste_todos, [19](#)
- teste_transposta, [19](#)
- transposta, [19](#)
- multcomp
 - matrizes.c, [11](#)
 - matrizes.h, [17](#)
- printComplex
 - matrizes.c, [11](#)
 - matrizes.h, [17](#)
- produto_interno
 - matrizes.c, [11](#)
 - matrizes.h, [17](#)
- produto_matricial
 - matrizes.c, [12](#)
 - matrizes.h, [17](#)
- produto_por_escalar
 - matrizes.c, [12](#)
- real
 - complexo, [7](#)
- soma
 - matrizes.c, [12](#)
 - matrizes.h, [17](#)
- soma_complexo
 - matrizes.c, [12](#)
 - matrizes.h, [18](#)
- subtracao
 - matrizes.c, [12](#)
 - matrizes.h, [18](#)
- teste_conjugada

- matrizes.c, [13](#)
- matrizes.h, [18](#)
- teste_hermetiana
 - matrizes.c, [13](#)
 - matrizes.h, [18](#)
- teste_produto_interno
 - matrizes.c, [13](#)
 - matrizes.h, [18](#)
- teste_produto_matricial
 - matrizes.c, [13](#)
 - matrizes.h, [18](#)
- teste_produto_por_escalar
 - matrizes.c, [13](#)
- teste_soma
 - matrizes.c, [13](#)
 - matrizes.h, [19](#)
- teste_subtracao
 - matrizes.c, [14](#)
 - matrizes.h, [19](#)
- teste_todos
 - matrizes.c, [14](#)
 - matrizes.h, [19](#)
- teste_transposta
 - matrizes.c, [14](#)
 - matrizes.h, [19](#)
- transposta
 - matrizes.c, [14](#)
 - matrizes.h, [19](#)