

Projeto 1  
19.20.37.42

Generated by Doxygen 1.8.18



---

<b>1 File Index</b>	<b>1</b>
1.1 File List . . . . .	1
<b>2 File Documentation</b>	<b>3</b>
2.1 Proj1.cpp File Reference . . . . .	3
2.1.1 Macro Definition Documentation . . . . .	4
2.1.1.1 owner . . . . .	4
2.1.2 Function Documentation . . . . .	4
2.1.2.1 get_altura() . . . . .	4
2.1.2.2 pegar a altura e colocar na lista . . . . .	4
2.1.2.3 get_comprimento() . . . . .	5
2.1.2.4 pegar o comprimento e colocar na lista . . . . .	5
2.1.2.5 get_data() . . . . .	6
2.1.2.6 pegar os dados e colocar na lista . . . . .	6
2.1.2.7 get_nome() . . . . .	6
2.1.2.8 pegar os nomes que estão no arquivo e colocar na lista . . . . .	6
2.1.2.9 main() . . . . .	7
2.1.3 MAIN . . . . .	7
2.1.3.1 obter dados dos arquivos . . . . .	7
2.1.3.2 matrix_Q() . . . . .	7
2.1.3.3 retorna a quantidade de elementos da matriz . . . . .	7
2.1.3.4 nome_arq() . . . . .	8
2.1.3.5 verificar() . . . . .	8
2.1.3.6 verificar se existe o arq xml, caso não, retorna 0 junto com uma mensagem de erro . . . . .	8
<b>Index</b>	<b>11</b>



# Chapter 1

## File Index

### 1.1 File List

Here is a list of all files with brief descriptions:

<a href="#">Proj1.cpp</a> . . . . .	3
-------------------------------------	---



## Chapter 2

# File Documentation

### 2.1 Proj1.cpp File Reference

```
#include <iostream>
#include <string>
#include <fstream>
#include <stack>
#include <list>
#include <queue>
```

#### Macros

- `#define owner = < antoniomontagner309@gmail.com >`

#### Functions

- `std::string nome_arq ()`  
*retornar o nome do arquivo xml a ser utilizado*
- `int get_nome (std::fstream &file, std::list< std::string > &list_nome)`  
*instancias que serão utilizadas e a serem completadas*
- `int get_altura (std::fstream &file, std::list< std::string > &list_altura)`  
*get\_altura abre o arquivo e conta suas linhas para pegar a altura e colocar na lista de retorno.*
- `int get_comprimento (std::fstream &file, std::list< std::string > &list_comprimento)`  
*abre o arquivo e conta suas linhas para pegar o comprimento e colocar na lista*
- `int get_data (std::fstream &file, std::list< std::string > &list_dado, std::list< std::string > list_altura)`  
*verificar se o arquivo XML passado "file" é valido,*
- `int matrix_Q (std::string name, std::string altura, std::string comprimento, std::string data)`  
*retorna a quantidade de elementos da matriz em um escopo de altura X comprimento e utilizando a lista de dados do XML passado*
- `bool verificar (std::fstream &file)`  
*verificar o se existe o arq xml, caso não, retorna erro junto com uma mensagem de erro.*
- `int main ()`

## 2.1.1 Macro Definition Documentation

### 2.1.1.1 owner

```
#define owner = < antoniomontagner309@gmail.com >
```

#### 2.1.1.1.1 Copyright [2021] <Antonio S M>

Version

19.20.37.42

#### 2.1.1.1.2 Projeto I - processamento de XML com imagens binárias

## 2.1.2 Function Documentation

### 2.1.2.1 get\_altura()

```
int get_altura (
    std::fstream & file,
    std::list< std::string > & list_altura )
```

get\_altura abre o arquivo e conta suas linhas para pegar a altura e colocar na lista de retorno.

#### 2.1.2.2 pegar a altura e colocar na lista

Parameters

<i>file</i>	
<i>list_altura</i>	

verificar se esta aberto o arquivo e pegar altura

Parameters

<i>comprimento</i>	
--------------------	--

enquanto a posicao da tag "height" estiver diferente da posição atual

"line.find() " pegar altura



"line[pos+1]" adicionar no fim um marcador "@"

atualizar comprimento + linha atual

fechar o arquivo apos criar a lista com os dados para retorno

#### Returns

list\_altura.size()

### 2.1.2.3 get\_comprimento()

```
int get_comprimento (
    std::fstream & file,
    std::list< std::string > & list_comprimento )
```

abre o arquivo e conta suas linhas para pegar o comprimento e colocar na lista

### 2.1.2.4 pegar o comprimento e colocar na lista

#### Parameters

<i>file</i>	
<i>list_comprimento</i>	

pegar comprimento

altura

"line.find()" procurar tag de comprimento

#### Parameters

<i>pos</i>	
------------	--

pegar comprimento

"line[pos+1]" adicionar no fim um marcador "@"

fechar o arquivo apos criar a lista com os dados para retorno

#### Returns

list\_comprimento.size()

### 2.1.2.5 get\_data()

```
int get_data (
    std::fstream & file,
    std::list< std::string > & list_dado,
    std::list< std::string > list_altura )
```

verificar se o arquivo XML passado "file" é valido,

### 2.1.2.6 pegar os dados e colocar na lista

**2.1.2.6.1 get\_data** pegar os dados do arquivo passado em um escopo diferente do primeiro elemento de "list\_↵\_altura",

retorna-los em uma lista "list\_dado"

#### Parameters

<i>file</i>	<i>list_dado</i>	<i>list_altura</i>	
-------------	------------------	--------------------	--

#### Returns

list\_dado.size()

### 2.1.2.7 get\_nome()

```
int get_nome (
    std::fstream & file,
    std::list< std::string > & list_nome )
```

instancias que serão utilizadas e a serem completadas

pegar os nomes que estão no arquivo XML e colocar em uma lista

### 2.1.2.8 pegar os nomes que estão no arquivo e colocar na lista

#### Parameters

<i>file</i>	
<i>list_nome</i>	

pega nome da imagem

"line.find()" procura o inicio do nome usando a tag <name>

"line[pos+1]" adicionar no fim um marcador "@"

"list\_nome.push\_back(name)" adiciona o nome na lista de nomes

fechar o arquivo apor fazer a verificação e obter o nome

#### Returns

list\_nome.size()

#### 2.1.2.9 main()

```
int main ( )
```

### 2.1.3 MAIN

"verificar(file)" retorna 0 junto com uma mensagem de erro caso o arquivo nao seja válido

#### Parameters

<i>nome_lista</i>	
<i>altura_lista</i>	
<i>comprimento_lista</i>	
<i>dados_lista</i>	recebe o nome da imagem, altura, comprimento, dados

#### 2.1.3.1 obter dados dos arquivos

pegar os dados nome, altura, comprimento e a matriz

abrir a imagem

pegar o nome da imagem do arquivo

pegar altura da imagem do arquivo

pegar comprimento da imagem do arquivo

pegar dados da imagem do arquivo

retorno da quantidade de elementos específico

#### 2.1.3.2 matrix\_Q()

```
int matrix_Q (
    std::string name,
    std::string altura,
    std::string comprimento,
    std::string data )
```

retorna a quantidade de elementos da matriz em um escopo de altura X comprimento e utilizando a lista de dados do XML passado

#### 2.1.3.3 retorna a quantidade de elementos da matriz

**Parameters**

<i>name, altura, comprimento, data</i>	
--	--

**Parameters**

<i>comp</i> <sub>↔</sub>	
<i>_</i>	
<i>alt_</i>	

cria uma lista para armazenar os elementos da matriz

armazena o elemento da coordenada "push(std::make\_pair(i, k))"

armazenar a lista em uma temporaria para retorno e limpar a lista principal

verificar se o comprimento da lista temporaria e maior ou igual a 0

verificar se a altura da lista temporaria e menor que a altura passada

verificar se a altura da lista temporaria e maior ou igual a 0

verificar se o comprimento da lista temporaria e menor que o comprimento passado

enfileira

**Returns**

n\_elementos

**2.1.3.4 nome\_arq()**

```
std::string nome_arq ( )
```

retornar o nome do arquivo xml a ser utilizado

**2.1.3.4.1 retornar nome do arquivo****2.1.3.5 verificar()**

```
bool verificar (
    std::fstream & file )
```

verificar o se existe o arq xml, caso não, retorna erro junto com uma mensagem de erro.

**2.1.3.6 verificar se existe o arq xml, caso não, retorna 0 junto com uma mensagem de erro**

**Parameters**

<i>file</i>	
-------------	--

primeiro vamos verificar os erros

**Parameters**

<i>stack</i>	
<i>read</i>	
<i>error</i>	

usar getline para percorrer a string

checar se é topo da lista para continuar ou nao

caso nao seja topo da lista, passa erro e retorna

fechar o arquivo apor fazer a verificação

**Returns**

error



# Index

- get\_altura
  - Proj1.cpp, [4](#)
- get\_comprimento
  - Proj1.cpp, [5](#)
- get\_data
  - Proj1.cpp, [5](#)
- get\_nome
  - Proj1.cpp, [6](#)
- main
  - Proj1.cpp, [7](#)
- matrix\_Q
  - Proj1.cpp, [7](#)
- nome\_arq
  - Proj1.cpp, [8](#)
- owner
  - Proj1.cpp, [4](#)
- Proj1.cpp, [3](#)
  - get\_altura, [4](#)
  - get\_comprimento, [5](#)
  - get\_data, [5](#)
  - get\_nome, [6](#)
  - main, [7](#)
  - matrix\_Q, [7](#)
  - nome\_arq, [8](#)
  - owner, [4](#)
  - verificar, [8](#)
- verificar
  - Proj1.cpp, [8](#)