

# Reference Manual

Generated by Doxygen 1.6.1

Sun May 2 21:48:34 2010



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	elemento Struct Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Member Data Documentation . . . . .	5
3.1.2.1	cor . . . . .	5
3.1.2.2	letra . . . . .	5
3.2	erro Struct Reference . . . . .	6
3.2.1	Member Data Documentation . . . . .	6
3.2.1.1	err_code . . . . .	6
3.2.1.2	mensagem . . . . .	6
3.3	gravados Struct Reference . . . . .	7
3.3.1	Detailed Description . . . . .	7
3.3.2	Member Data Documentation . . . . .	7
3.3.2.1	its . . . . .	7
3.3.2.2	jgs . . . . .	7
3.3.2.3	mvs . . . . .	7
3.3.2.4	totalGr . . . . .	7
3.4	It Struct Reference . . . . .	8
3.4.1	Detailed Description . . . . .	8
3.4.2	Member Data Documentation . . . . .	8
3.4.2.1	next . . . . .	8
3.4.2.2	pos . . . . .	8
3.5	jogadas Struct Reference . . . . .	9

3.5.1	Detailed Description	9
3.5.2	Member Data Documentation	9
3.5.2.1	col	9
3.5.2.2	color	9
3.5.2.3	next	9
3.5.2.4	row	9
3.6	movimentos Struct Reference	10
3.6.1	Detailed Description	10
3.6.2	Member Data Documentation	10
3.6.2.1	next	10
3.6.2.2	pos	10
<b>4</b>	<b>File Documentation</b>	<b>11</b>
4.1	ajuda.c File Reference	11
4.1.1	Detailed Description	11
4.1.2	Define Documentation	12
4.1.2.1	CHUNK_SIZE	12
4.1.3	Function Documentation	12
4.1.3.1	ajuda	12
4.1.3.2	auto_black	12
4.1.3.3	auto_white	12
4.1.3.4	control	12
4.1.3.5	p_branças	13
4.1.3.6	p_preta	13
4.2	ajuda.h File Reference	14
4.2.1	Function Documentation	14
4.2.1.1	ajuda	14
4.2.1.2	auto_black	14
4.2.1.3	auto_white	14
4.2.1.4	control	15
4.2.1.5	p_branças	15
4.2.1.6	p_preta	15
4.3	comandos.c File Reference	16
4.3.1	Detailed Description	16
4.3.2	Function Documentation	16
4.3.2.1	executa_comando	16
4.3.2.2	executaCriaJogo	17

4.3.2.3	<a href="#">executaGravaJogo</a>	17
4.3.2.4	<a href="#">executaJogada</a>	17
4.4	<a href="#">comandos.h File Reference</a>	19
4.4.1	<a href="#">Function Documentation</a>	19
4.4.1.1	<a href="#">executa_comando</a>	19
4.4.1.2	<a href="#">executaCriaJogo</a>	19
4.4.1.3	<a href="#">executaGravaJogo</a>	20
4.4.1.4	<a href="#">executaJogada</a>	20
4.5	<a href="#">erro.c File Reference</a>	21
4.5.1	<a href="#">Typedef Documentation</a>	21
4.5.1.1	<a href="#">ERRO</a>	21
4.5.2	<a href="#">Function Documentation</a>	21
4.5.2.1	<a href="#">mensagem_de_erro</a>	21
4.6	<a href="#">erro.h File Reference</a>	22
4.6.1	<a href="#">Define Documentation</a>	22
4.6.1.1	<a href="#">E_ARGS</a>	22
4.6.1.2	<a href="#">E_COMMAND</a>	22
4.6.1.3	<a href="#">E_COORDS</a>	22
4.6.1.4	<a href="#">E_FILE_FORMAT</a>	22
4.6.1.5	<a href="#">E_INVARGS</a>	22
4.6.1.6	<a href="#">E_NO_BOARD</a>	22
4.6.1.7	<a href="#">E_NO_FILE</a>	22
4.6.1.8	<a href="#">E_NO_MOVES</a>	22
4.6.1.9	<a href="#">E_NOERR</a>	22
4.6.1.10	<a href="#">E_SAVE</a>	22
4.6.1.11	<a href="#">E_WRONG_SOLUTION</a>	22
4.6.2	<a href="#">Function Documentation</a>	22
4.6.2.1	<a href="#">mensagem_de_erro</a>	22
4.7	<a href="#">estado_tabuleiro.c File Reference</a>	24
4.7.1	<a href="#">Detailed Description</a>	24
4.7.2	<a href="#">Function Documentation</a>	24
4.7.2.1	<a href="#">caminho</a>	24
4.7.2.2	<a href="#">free_tab</a>	24
4.7.2.3	<a href="#">prepara_tab</a>	25
4.7.2.4	<a href="#">st</a>	25
4.7.2.5	<a href="#">vb</a>	25

4.7.2.6	vl	25
4.7.2.7	vp	25
4.8	estado_tabuleiro.h File Reference	26
4.8.1	Function Documentation	26
4.8.1.1	caminho	26
4.8.1.2	free_tab	26
4.8.1.3	prepara_tab	26
4.8.1.4	st	26
4.8.1.5	vb	26
4.8.1.6	vl	27
4.8.1.7	vp	27
4.9	jogadas.c File Reference	28
4.9.1	Detailed Description	28
4.9.2	Function Documentation	29
4.9.2.1	anc	29
4.9.2.2	anm	29
4.9.2.3	E_Par	29
4.9.2.4	E_Par_des	29
4.9.2.5	E_trp_col	30
4.9.2.6	E_trp_row	30
4.9.2.7	free_jogadasGR	30
4.9.2.8	freeIts	31
4.9.2.9	freeJgs	31
4.9.2.10	freeMvs	31
4.9.2.11	grava_jogada	31
4.9.2.12	grava_movimento	31
4.9.2.13	inicia_GR	32
4.9.2.14	it	32
4.9.2.15	jogada	32
4.9.2.16	jogadaManual	32
4.9.2.17	pds	33
4.9.2.18	pis	33
4.9.2.19	rb	34
4.9.2.20	snd	34
4.9.2.21	trp	34
4.10	jogadas.h File Reference	35

4.10.1	Typedef Documentation	36
4.10.1.1	Gravados	36
4.10.1.2	IT	36
4.10.1.3	Jogadas	36
4.10.1.4	Movimentos	36
4.10.2	Function Documentation	36
4.10.2.1	anc	36
4.10.2.2	anm	36
4.10.2.3	E_Par	36
4.10.2.4	E_Par_des	37
4.10.2.5	E_trp_col	37
4.10.2.6	E_trp_row	37
4.10.2.7	free_jogadasGR	38
4.10.2.8	grava_jogada	38
4.10.2.9	grava_movimento	38
4.10.2.10	inicia_GR	38
4.10.2.11	it	38
4.10.2.12	jogada	39
4.10.2.13	jogadaManual	39
4.10.2.14	pds	39
4.10.2.15	pis	40
4.10.2.16	rb	40
4.10.2.17	snd	40
4.10.2.18	trp	41
4.11	main.c File Reference	42
4.11.1	Detailed Description	42
4.11.2	Function Documentation	42
4.11.2.1	main	42
4.12	tabuleiro.c File Reference	43
4.12.1	Detailed Description	43
4.12.2	Function Documentation	43
4.12.2.1	criarTabuleiro	43
4.12.2.2	dimencao	43
4.12.2.3	ficheiroValido	44
4.12.2.4	freeTab	44
4.12.2.5	gravaJogo	44

4.12.2.6	lerficheiro . . . . .	44
4.12.2.7	printTabuleiro . . . . .	45
4.13	tabuleiro.h File Reference . . . . .	46
4.13.1	Detailed Description . . . . .	46
4.13.2	Typedef Documentation . . . . .	46
4.13.2.1	Elem . . . . .	46
4.13.2.2	Elemento . . . . .	46
4.13.3	Function Documentation . . . . .	46
4.13.3.1	criarTabuleiro . . . . .	46
4.13.3.2	dimensao . . . . .	46
4.13.3.3	ficheiroValido . . . . .	47
4.13.3.4	freeTab . . . . .	47
4.13.3.5	gravaJogo . . . . .	47
4.13.3.6	lerficheiro . . . . .	48
4.13.3.7	printTabuleiro . . . . .	48



# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">elemento</a> (É constituída por dois chars, que correspondem às características de cada célula do tabuleiro, a cor e a letra ) . . . . .	5
<a href="#">erro</a> . . . . .	6
<a href="#">gravados</a> (Estrutura onde é armazenada toda a informação relativa a <a href="#">movimentos</a> e <a href="#">jogadas</a> guardadas ) . . . . .	7
<a href="#">It</a> (Estrutura responsável por guardar a posição a que corresponde o tabuleiro nas cores guardadas até ao momento que é invocada ) . . . . .	8
<a href="#">jogadas</a> (Estrutura responsável por guardar a coluna, linha e cor de uma célula a gravar ) . . . . .	9
<a href="#">movimentos</a> (Estrutura responsável por guardar a posição a que corresponde o tabuleiro nas cores guardadas até ao momento que é invocada ) . . . . .	10



# Chapter 2

## File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">ajuda.c</a> (Ajuda do jogo ) . . . . .	11
<a href="#">ajuda.h</a> . . . . .	14
<a href="#">comandos.c</a> (Comandos do jogo ) . . . . .	16
<a href="#">comandos.h</a> . . . . .	19
<a href="#">erro.c</a> . . . . .	21
<a href="#">erro.h</a> . . . . .	22
<a href="#">estado_tabuleiro.c</a> (Comandos de ajuda ao jogo ) . . . . .	24
<a href="#">estado_tabuleiro.h</a> . . . . .	26
<a href="#">jogadas.c</a> (Jogadas possiveis de efectuar e gravar do jogo e ) . . . . .	28
<a href="#">jogadas.h</a> . . . . .	35
<a href="#">main.c</a> (Main do jogo ) . . . . .	42
<a href="#">tabuleiro.c</a> (Tabuleiro do jogo ) . . . . .	43
<a href="#">tabuleiro.h</a> (Header file do <a href="#">tabuleiro.c</a> ) . . . . .	46



## Chapter 3

# Class Documentation

### 3.1 elemento Struct Reference

É constituída por dois chars, que correspondem às características de cada célula do tabuleiro, a cor e a letra.

```
#include <tabuleiro.h>
```

#### Public Attributes

- char [letra](#)  
*Letra da célula do tabuleiro.*
- char [cor](#)  
*Cor da célula do tabuleiro.*

#### 3.1.1 Detailed Description

É constituída por dois chars, que correspondem às características de cada célula do tabuleiro, a cor e a letra.

#### 3.1.2 Member Data Documentation

##### 3.1.2.1 elemento::cor

Cor da célula do tabuleiro.

##### 3.1.2.2 elemento::letra

Letra da célula do tabuleiro.

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

- [tabuleiro.h](#)

## 3.2 erro Struct Reference

### Public Attributes

- char \* [err\\_code](#)
- char \* [mensagem](#)

### 3.2.1 Member Data Documentation

#### 3.2.1.1 char\* erro::err\_code

#### 3.2.1.2 char\* erro::mensagem

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

- [erro.c](#)

## 3.3 gravados Struct Reference

Estrutura onde é armazenada toda a informação relativa a [movimentos](#) e [jogadas](#) guardadas.

```
#include <jogadas.h>
```

### Public Attributes

- [int](#) [totalGr](#)
- [IT](#) \* [its](#)
- [Jogadas](#) \* [jgs](#)
- [Movimentos](#) \* [mvs](#)

### 3.3.1 Detailed Description

Estrutura onde é armazenada toda a informação relativa a [movimentos](#) e [jogadas](#) guardadas.

### 3.3.2 Member Data Documentation

#### 3.3.2.1 [IT](#)\* [gravados::its](#)

#### 3.3.2.2 [Jogadas](#)\* [gravados::jgs](#)

#### 3.3.2.3 [Movimentos](#)\* [gravados::mvs](#)

#### 3.3.2.4 [int](#) [gravados::totalGr](#)

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

- [jogadas.h](#)

## 3.4 It Struct Reference

Estrutura responsavel por guardar a posição a que corresponde o tabuleiro nas cores guardadas até ao momento que é invocada.

```
#include <jogadas.h>
```

### Public Attributes

- int [pos](#)
- struct [It](#) \* [next](#)

### 3.4.1 Detailed Description

Estrutura responsavel por guardar a posição a que corresponde o tabuleiro nas cores guardadas até ao momento que é invocada.

### 3.4.2 Member Data Documentation

#### 3.4.2.1 struct [It](#)\* [It::next](#) [[read](#)]

#### 3.4.2.2 int [It::pos](#)

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

- [jogadas.h](#)



## 3.5 jogadas Struct Reference

Estrutura responsavel por guardar a coluna,linha e cor de uma celula a gravar.

```
#include <jogadas.h>
```

### Public Attributes

- int [col](#)
- int [row](#)
- char [color](#)
- struct [jogadas](#) \* [next](#)

### 3.5.1 Detailed Description

Estrutura responsavel por guardar a coluna,linha e cor de uma celula a gravar.

### 3.5.2 Member Data Documentation

**3.5.2.1** int [jogadas::col](#)

**3.5.2.2** char [jogadas::color](#)

**3.5.2.3** struct [jogadas](#)\* [jogadas::next](#) [[read](#)]

**3.5.2.4** int [jogadas::row](#)

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

- [jogadas.h](#)

## 3.6 movimentos Struct Reference

Estrutura responsavel por guardar a posição a que corresponde o tabuleiro nas cores guardadas até ao momento que é invocada.

```
#include <jogadas.h>
```

### Public Attributes

- int [pos](#)
- struct [movimentos](#) \* [next](#)

### 3.6.1 Detailed Description

Estrutura responsavel por guardar a posição a que corresponde o tabuleiro nas cores guardadas até ao momento que é invocada.

### 3.6.2 Member Data Documentation

**3.6.2.1** `struct movimentos* movimentos::next` [[read](#)]

**3.6.2.2** `int movimentos::pos`

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

- [jogadas.h](#)

# Chapter 4

## File Documentation

### 4.1 ajuda.c File Reference

```
ajuda do jogo #include "tabuleiro.h"
#include "erro.h"
#include "jogadas.h"
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
```

#### Defines

- #define [CHUNK\\_SIZE](#) 32

#### Functions

- int [ajuda](#) ()
- int [p\\_branças](#) ([Elem](#) \*\*Tab, int i, int j, int \*DIM, [Gravados](#) \*gr)
- int [auto\\_black](#) ([Elem](#) \*\*Tab, int \*DIM, [Gravados](#) \*gr)
- int [p\\_preta](#) ([Elem](#) \*\*Tab, int i, int j, char letra, int \*DIM, [Gravados](#) \*gr)
- int [auto\\_white](#) ([Elem](#) \*\*Tab, int \*DIM, [Gravados](#) \*gr)
- int [control](#) ([Elem](#) \*\*Tab, int \*DIM, int \*flagAB, int \*flagAP, [Gravados](#) \*gr)

#### 4.1.1 Detailed Description

ajuda do jogo

#### Author:

André Pimenta, João Gomes, Nelson Carvalho

**Date:**

1 Maio de 2010

**4.1.2 Define Documentation****4.1.2.1 #define CHUNK\_SIZE 32****4.1.3 Function Documentation****4.1.3.1 int ajuda ()**

Função que vai imprimir as intruções de ajuda guardada no ficheiro com as instruções de ajuda

**4.1.3.2 int auto\_black (Elem \*\* *Tab*, int \* *DIM*, Gravados \* *gr*)**

A função auto\_black procura no tabuleiro uma letra preta e chama a função p\_branças,

**Parameters:**

*Tab* tabuleiro.

*DIM* dimensão do tabuleiro.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

**Returns:**

Retorna um inteiro que irá indicar que houve alguma alteração no tabuleiro.

**4.1.3.3 int auto\_white (Elem \*\* *Tab*, int \* *DIM*, Gravados \* *gr*)**

A função auto\_white procura no tabuleiro uma letra branca e chama a função p\_preta,

**Parameters:**

*Tab* tabuleiro.

*DIM* dimensão do tabuleiro.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

**Returns:**

Retorna um inteiro que irá indicar que houve alguma alteração no tabuleiro.

**4.1.3.4 int control (Elem \*\* *Tab*, int \* *DIM*, int \* *flagAB*, int \* *flagAP*, Gravados \* *gr*)**

A função control recebe, o tabuleiro as dimensoes do mesmo, bem como as flags dos automaticos, e dependendo das flags que estão activas chama auto\_white e/ou auto\_black sucessivamente até que estas já não façam nenhuma alteração no tabuleiro

**Parameters:**

*Tab* tabuleiro.

*DIM* dimensão do tabuleiro.

*flagAB* flag do automático branco.

*flagAP* flag do automático preto.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

**Returns:**

Retorna um inteiro que irá indicar que foram introduzidas todas as cores possíveis.

**4.1.3.5 int p\_branca (Elem \*\* *Tab*, int *i*, int *j*, int \* *DIM*, Gravados \* *gr*)**

A função p\_branca pinta de branco todas as letras avolta da letra da posição (i,j).

**Parameters:**

*Tab* tabuleiro.

*i* linha.

*j* coluna.

*DIM* dimensão do tabuleiro.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

**Returns:**

Retorna um inteiro que irá indicar que houve alguma alteração no tabuleiro.

**4.1.3.6 int p\_preta (Elem \*\* *Tab*, int *i*, int *j*, char *letra*, int \* *DIM*, Gravados \* *gr*)**

A função p\_preta procura na linha (i) e na coluna (j) uma letra igual á letra passada e pinta essa letra de preto.

**Parameters:**

*Tab* tabuleiro.

*i* linha.

*j* coluna.

*letra* letra.

*DIM* dimensão do tabuleiro.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

**Returns:**

Retorna um inteiro que irá indicar que houve alguma alteração no tabuleiro.

## 4.2 ajuda.h File Reference

### Functions

- int [ajuda](#) ()
- int [p\\_branças](#) (Elem \*\*Tab, int i, int j, int \*DIM, Gravados \*gr)
- int [auto\\_black](#) (Elem \*\*Tab, int \*DIM, Gravados \*gr)
- int [p\\_preta](#) (Elem \*\*Tab, int i, int j, char letra, int \*DIM, Gravados \*gr)
- int [auto\\_white](#) (Elem \*\*Tab, int \*DIM, Gravados \*gr)
- int [control](#) (Elem \*\*Tab, int \*DIM, int \*flagAB, int \*flagAP, Gravados \*gr)

### 4.2.1 Function Documentation

#### 4.2.1.1 int ajuda ()

Função que vai imprimir as intruções de ajuda guardada no ficheiro com as instruções de ajuda

#### 4.2.1.2 int auto\_black (Elem \*\* *Tab*, int \* *DIM*, Gravados \* *gr*)

A função auto\_black procura no tabuleiro uma letra preta e chama a função p\_branças,

##### Parameters:

*Tab* tabuleiro.

*DIM* dimensão do tabuleiro.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

##### Returns:

Retorna um inteiro que irá indicar que houve alguma alteração no tabuleiro.

#### 4.2.1.3 int auto\_white (Elem \*\* *Tab*, int \* *DIM*, Gravados \* *gr*)

A função auto\_white procura no tabuleiro uma letra branca e chama a função p\_preta,

##### Parameters:

*Tab* tabuleiro.

*DIM* dimensão do tabuleiro.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

##### Returns:

Retorna um inteiro que irá indicar que houve alguma alteração no tabuleiro.

#### 4.2.1.4 int control (Elem \*\* *Tab*, int \* *DIM*, int \* *flagAB*, int \* *flagAP*, Gravados \* *gr*)

A função control recebe, o tabuleiro as dimensoes do mesmo, bem como as flags dos automaticos, e dependendo das flags que estão activas chama auto\_white e/ou auto\_black sucessivamente até que estas já não façam nenhuma alteração no tabuleiro

##### Parameters:

*Tab* tabuleiro.

*DIM* dimensão do tabuleiro.

*flagAB* flag do automático branco.

*flagAP* flag do automático preto.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

##### Returns:

Retorna um inteiro que irá indicar que foram introduzidas todas as cores possíveis.

#### 4.2.1.5 int p\_branca (Elem \*\* *Tab*, int *i*, int *j*, int \* *DIM*, Gravados \* *gr*)

A função p\_branca pinta de branco todas as letras avolta da letra da posição (i,j).

##### Parameters:

*Tab* tabuleiro.

*i* linha.

*j* coluna.

*DIM* dimensão do tabuleiro.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

##### Returns:

Retorna um inteiro que irá indicar que houve alguma alteração no tabuleiro.

#### 4.2.1.6 int p\_preta (Elem \*\* *Tab*, int *i*, int *j*, char *letra*, int \* *DIM*, Gravados \* *gr*)

A função p\_preta procura na linha (i) e na coluna (j) uma letra igual á letra passada e pinta essa letra de preto.

##### Parameters:

*Tab* tabuleiro.

*i* linha.

*j* coluna.

*letra* letra.

*DIM* dimensão do tabuleiro.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

##### Returns:

Retorna um inteiro que irá indicar que houve alguma alteração no tabuleiro.

## 4.3 comandos.c File Reference

```
Comandos do jogo. #include "tabuleiro.h"
#include "erro.h"
#include "jogadas.h"
#include "ajuda.h"
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include "estado_tabuleiro.h"
```

### Functions

- int [executaJogada](#) (char \*args, char \*cor, int \*DIM, [Elem](#) \*\*Tab, int \*flagAP, int \*flagAB, [Gravados](#) \*gr)
- int [executaCriaJogo](#) (char \*args, int \*DIM, [Elem](#) \*\*Tab, [Gravados](#) \*gr, int \*flagAP, int \*flagAB)
- int [executaGravaJogo](#) (char \*args, int \*DIM, [Elem](#) \*\*Tab)
- int [executa\\_comando](#) (char \*linha, int \*DIM, [Elem](#) \*\*Tab, int \*flagAP, int \*flagAB, [Gravados](#) \*gr)

### 4.3.1 Detailed Description

Comandos do jogo.

#### Author:

André Pimenta, João Gomes, Nelson Carvalho

#### Date:

1 Abril de 2010

### 4.3.2 Function Documentation

#### 4.3.2.1 int [executa\\_comando](#) (char \* *linha*, int \* *DIM*, [Elem](#) \*\* *Tab*, int \* *flagAP*, int \* *flagAB*, [Gravados](#) \* *gr*)

A função [executa\\_comando](#) é uma das funções mais importantes pois é ela que vai reagir aos comandos dados pelo utilizador. A função irá receber a linha de comando separa la em dois, no comando e nos argumentos do comando, e passa los para as funções associadas aos comandos.

#### Parameters:

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*linha* Recebe uma string que corresponde á linha lida da shell.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).



*flagAP* flag para controlo do automatico preto.

*flagAB* flag para controlo do automatico branco.

*gr* apontador para a estrutura de dados onde se guarda as jogadas e moviemntos efectuados.

**Returns:**

torna um inteiro que verifica se foi ou não bem sucedida a função.

**4.3.2.2 int executaCriaJogo (char \* args, int \* DIM, Elem \*\* Tab, Gravados \* gr, int \* flagAP, int \* flagAB)**

A função executaCriaJog recebe os argumentos da linha de comando, vai verificar se tem os argumentos, certos se tal ocorre, executa criarJogo com o nome do argumento no formato ".ltr".

**Parameters:**

*args* Recebe uma string com os argumentos associados ao comando em questão ("cr").

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*gr* apontador para a estrutura de dados onde se encontram guardados todos os movimentos e jogadas.

*flagAP* flag para controlo do automatico preto.

*flagAB* flag para controlo do automatico branco.

**Returns:**

torna um inteiro que verifica se foi ou não bem sucedida a função.

**4.3.2.3 int executaGravaJogo (char \* args, int \* DIM, Elem \*\* Tab)**

A função executaGravaJogo recebe os argumentos da linha de comando e vai verificar se tem os argumetos certos, se tal for verdade, executa gravaJogo com o nome do argumento no formato ".ltr".

**Parameters:**

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*args* Recebe uma string com os argumentos associados ao comando em questão("gr").

**Returns:**

torna um inteiro que verifica se foi ou não bem sucedida a função.

**4.3.2.4 int executaJogada (char \* args, char \* cor, int \* DIM, Elem \*\* Tab, int \* flagAP, int \* flagAB, Gravados \* gr)**

A função executajogada recebe os argumentos da linha de comandos e a cor que vai passar para a jogada. Vai verificar se tem os argumetos certos, e se tal vou verdade, chama a função jogada, para que esta se possa realizar.

**Parameters:**

*args* Recebe uma string com os argumentos do comando do utilizador.

*cor* Corresponde ao próprio comando que irá ser utilizado como a cor que se pretende mudar.

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

*flagAP* flag para controlo do automatico preto.

*flagAB* flag para controlo do automatico branco.

**Returns:**

Retorna um inteiro que verifica se foi ou não bem sucedida a função.

## 4.4 comandos.h File Reference

### Functions

- int [executaJogada](#) (char \*args, char \*cor, int \*DIM, [Elem](#) \*\*Tab, int \*flagAP, int \*flagAB, [Gravados](#) \*gr)
- int [executaCriaJogo](#) (char \*args, int \*DIM, [Elem](#) \*\*Tab, [Gravados](#) \*gr, int \*flagAP, int \*flagAB)
- int [executaGravaJogo](#) (char \*args, int \*DIM, [Elem](#) \*\*Tab)
- int [executa\\_comando](#) (char \*linha, int \*DIM, [Elem](#) \*\*Tab, int \*flagAP, int \*flagAB, [Gravados](#) \*gr)

### 4.4.1 Function Documentation

#### 4.4.1.1 int [executa\\_comando](#) (char \* *linha*, int \* *DIM*, [Elem](#) \*\* *Tab*, int \* *flagAP*, int \* *flagAB*, [Gravados](#) \* *gr*)

A função [executa\\_comando](#) é uma das funções mais importantes pois é ela que vai reagir aos comandos dados pelo utilizador. A função irá receber a linha de comando separa la em dois, no comando e nos argumentos do comando, e passa los para as funções associadas aos comandos.

##### Parameters:

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*linha* Recebe uma string que corresponde á linha lida da shell.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

*flagAP* flag para controlo do automatico preto.

*flagAB* flag para controlo do automatico branco.

*gr* apontador para a estrutura de dados onde se guarda as [jogadas](#) e moviemntos efectuados.

##### Returns:

torna um inteiro que verifica se foi ou não bem sucedida a função.

#### 4.4.1.2 int [executaCriaJogo](#) (char \* *args*, int \* *DIM*, [Elem](#) \*\* *Tab*, [Gravados](#) \* *gr*, int \* *flagAP*, int \* *flagAB*)

A função [executaCriaJog](#) recebe os argumentos da linha de comando, vai verificar se tem os argumentos, certos se tal ocorre, executa [criarJogo](#) com o nome do argumento no formato ".ltr".

##### Parameters:

*args* Recebe uma string com os argumentos associados ao comando em questão ("cr").

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

*flagAP* flag para controlo do automatico preto.

*flagAB* flag para controlo do automatico branco.

##### Returns:

torna um inteiro que verifica se foi ou não bem sucedida a função.

#### 4.4.1.3 `int executaGravaJogo(char * args, int * DIM, Elem ** Tab)`

A função `executaGravaJogo` recebe os argumentos da linha de comando e vai verificar se tem os argumentos certos, se tal for verdade, executa `gravaJogo` com o nome do argumento no formato ".ltr".

##### Parameters:

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*args* Recebe uma string com os argumentos associados ao comando em questão("gr").

##### Returns:

torna um inteiro que verifica se foi ou não bem sucedida a função.

#### 4.4.1.4 `int executaJogada(char * args, char * cor, int * DIM, Elem ** Tab, int * flagAP, int * flagAB, Gravados * gr)`

A função `executaJogada` recebe os argumentos da linha de comandos e a cor que vai passar para a jogada. Vai verificar se tem os argumentos certos, e se tal for verdade, chama a função `jogada`, para que esta se possa realizar.

##### Parameters:

*args* Recebe uma string com os argumentos do comando do utilizador.

*cor* Corresponde ao próprio comando que irá ser utilizado como a cor que se pretende mudar.

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

*flagAP* flag para controlo do automatico preto.

*flagAB* flag para controlo do automatico branco.

##### Returns:

Retorna um inteiro que verifica se foi ou não bem sucedida a função.

## 4.5 erro.c File Reference

```
#include <stdio.h>
#include "erro.h"
```

### Classes

- struct [erro](#)

### Typedefs

- typedef struct [erro](#) [ERRO](#)

### Functions

- int [mensagem\\_de\\_erro](#) (int *erro\_num*)

#### 4.5.1 Typedef Documentation

##### 4.5.1.1 typedef struct erro ERRO

#### 4.5.2 Function Documentation

##### 4.5.2.1 int mensagem\_de\_erro (int *erro\_num*)

Reporta o [erro](#) escrevendo uma mensagem no standard error

#### Parameters:

*erro\_num* O número do [erro](#) que vem da tabela de erros. Usar SEMPRE o define e não o valor numérico para maior legibilidade.

## 4.6 erro.h File Reference

### Defines

- #define [E\\_NOERR](#) 0
- #define [E\\_COMMAND](#) 1
- #define [E\\_ARGS](#) 2
- #define [E\\_NO\\_BOARD](#) 3
- #define [E\\_NO\\_FILE](#) 4
- #define [E\\_FILE\\_FORMAT](#) 5
- #define [E\\_WRONG\\_SOLUTION](#) 6
- #define [E\\_NO\\_MOVES](#) 7
- #define [E\\_COORDS](#) 8
- #define [E\\_SAVE](#) 9
- #define [E\\_INVARGS](#) 10

### Functions

- int [mensagem\\_de\\_erro](#) (int erro\_num)

#### 4.6.1 Define Documentation

4.6.1.1 #define [E\\_ARGS](#) 2

4.6.1.2 #define [E\\_COMMAND](#) 1

4.6.1.3 #define [E\\_COORDS](#) 8

4.6.1.4 #define [E\\_FILE\\_FORMAT](#) 5

4.6.1.5 #define [E\\_INVARGS](#) 10

4.6.1.6 #define [E\\_NO\\_BOARD](#) 3

4.6.1.7 #define [E\\_NO\\_FILE](#) 4

4.6.1.8 #define [E\\_NO\\_MOVES](#) 7

4.6.1.9 #define [E\\_NOERR](#) 0

4.6.1.10 #define [E\\_SAVE](#) 9

4.6.1.11 #define [E\\_WRONG\\_SOLUTION](#) 6

#### 4.6.2 Function Documentation

4.6.2.1 int [mensagem\\_de\\_erro](#) (int *erro\_num*)

Reporta o [erro](#) escrevendo uma mensagem no standard error

**Parameters:**

*erro\_num* O número do [erro](#) que vem da tabela de erros. Usar SEMPRE o define e não o valor numérico para maior legibilidade.

## 4.7 estado\_tabuleiro.c File Reference

Comandos de ajuda ao jogo. #include "tabuleiro.h"

#include "erro.h"

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

### Functions

- int [vb](#) ([Elem](#) \*\*Tab, int \*DIM)
- int [vp](#) ([Elem](#) \*\*Tab, int \*DIM)
- void [free\\_tab](#) (int \*\*tab\_aux, int \*DIM)
- int [prepara\\_tab](#) ([Elem](#) \*\*Tab, int \*\*tab\_aux, int \*DIM)
- int [caminho](#) (int \*\*tab\_aux, int \*DIM)
- int [vl](#) ([Elem](#) \*\*Tab, int \*DIM)
- int [st](#) ([Elem](#) \*\*Tab, int \*DIM)

### 4.7.1 Detailed Description

Comandos de ajuda ao jogo.

#### Author:

André Pimenta, João Gomes, Nelson Carvalho

#### Date:

3 Abril de 2010

### 4.7.2 Function Documentation

#### 4.7.2.1 int [caminho](#) (int \*\* *tab\_aux*, int \* *DIM*)

#### 4.7.2.2 void [free\\_tab](#) (int \*\* *tab\_aux*, int \* *DIM*)

Função que procura a existencia de caminho entre todas as células brABas ou indefinidas. Caso não exista retorna uma mensagem de [erro](#).

#### Parameters:

*tab\_aux* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

#### Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#), é exibido através da função mensagem de [erro](#).



**4.7.2.3** `int prepara_tab (Elem ** Tab, int ** tab_aux, int * DIM)`

**4.7.2.4** `int st (Elem ** Tab, int * DIM)`

**4.7.2.5** `int vb (Elem ** Tab, int * DIM)`

Função que percorre todas as posições do tabuleiro caso sejam pretas, verifica se não têm vizinas na vertical ou na horizontal com a cor preta. Caso existam retorna uma mensagem de [erro](#).

**Parameters:**

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

**Returns:**

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#), é exibido através da função mensagem de [erro](#).

**4.7.2.6** `int vl (Elem ** Tab, int * DIM)`

**4.7.2.7** `int vp (Elem ** Tab, int * DIM)`

Função que percorre uma linha/coluna e vê se existem duas letras iguais com mesma cor (brABa), fazendo o mesmo para todas as linhas/colunas. Caso existam retorna uma mensagem de [erro](#).

**Parameters:**

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

**Returns:**

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#), é exibido através da função mensagem de [erro](#).

## 4.8 estado\_tabuleiro.h File Reference

### Functions

- int [vb](#) ([Elem](#) \*\*Tab, int \*DIM)
- int [vp](#) ([Elem](#) \*\*tab, int \*DIM)
- int [vl](#) ([Elem](#) \*\*Tab, int \*DIM)
- int [st](#) ([Elem](#) \*\*Tab, int \*DIM)
- int [caminho](#) (int \*tab\_aux, int \*DIM)
- int [prepara\\_tab](#) ([Elem](#) \*\*Tab, int \*\*tab\_aux, int \*DIM)
- void [free\\_tab](#) (int \*\*tab\_aux, int \*DIM)

### 4.8.1 Function Documentation

#### 4.8.1.1 int [caminho](#) (int \* *tab\_aux*, int \* *DIM*)

#### 4.8.1.2 void [free\\_tab](#) (int \*\* *tab\_aux*, int \* *DIM*)

Função que procura a existencia de caminho entre todas as celulas brABas ou indefinidas. Caso não exista retorna uma mensagem de [erro](#).

##### Parameters:

*tab\_aux* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

##### Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#),é exibido atraves da função mensagem de [erro](#).

#### 4.8.1.3 int [prepara\\_tab](#) ([Elem](#) \*\* *Tab*, int \*\* *tab\_aux*, int \* *DIM*)

#### 4.8.1.4 int [st](#) ([Elem](#) \*\* *Tab*, int \* *DIM*)

#### 4.8.1.5 int [vb](#) ([Elem](#) \*\* *Tab*, int \* *DIM*)

Função que percorre todas as posições do tabuleiro caso sejam pretas, verifica se não têm vizinas na vertical ou na horinzontal com a cor preta. Caso exitam retorna uma mensagem de [erro](#).

##### Parameters:

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

##### Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#),é exibido atraves da função mensagem de [erro](#).

**4.8.1.6 int vl (Elem \*\* *Tab*, int \* *DIM*)****4.8.1.7 int vp (Elem \*\* *Tab*, int \* *DIM*)**

Função que percorre uma linha/coluna e ve se existem duas letras iguais com mesma cor(brABa), fazendo o mesmo para todas as linhas/colunas. Caso existam retorna uma mensagem de [erro](#).

**Parameters:**

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

**Returns:**

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#), é exibido através da função mensagem de [erro](#).

## 4.9 jogadas.c File Reference

Jogadas possiveis de efectuar e gravar do jogo e. #include "tabuleiro.h"

```
#include "erro.h"
#include "jogadas.h"
#include "ajuda.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

### Functions

- int [jogadaManual](#) (char cor, int x, int y, int \*DIM, Elem \*\*Tab, Gravados \*gr)
- int [jogada](#) (char cor, int x, int y, Elem \*\*Tab, Gravados \*gr)
- int [E\\_trp\\_col](#) (Elem \*\*Tab, int \*DIM, int i, int j, Gravados \*gr)
- int [E\\_trp\\_row](#) (Elem \*\*Tab, int \*DIM, int i, int j, Gravados \*gr)
- int [trp](#) (Elem \*\*Tab, int \*DIM, int \*flagAP, int \*flagAB, Gravados \*gr)
- int [snd](#) (Elem \*\*Tab, int \*DIM, int \*flagAP, int \*flagAB, Gravados \*gr)
- int [E\\_Par](#) (Elem \*\*Tab, int pos, char letra, int \*DIM, char \*local)
- int [pis](#) (Elem \*\*Tab, int \*DIM, int \*flagAP, int \*flagAB, Gravados \*gr)
- int [E\\_Par\\_des](#) (Elem \*\*Tab, char letra1, char letra2, int \*DIM, int line1, int cl1, int line2, int cl2)
- int [pds](#) (Elem \*\*Tab, int \*DIM, int \*flagAP, int \*flagAB, Gravados \*gr)
- int [grava\\_jogada](#) (int row, int col, char color, Gravados \*gr)
- int [anc](#) (Elem \*\*Tab, Gravados \*gr)
- int [it](#) (Gravados \*gr)
- int [rb](#) (Elem \*\*Tab, Gravados \*gr)
- int [inicia\\_GR](#) (Gravados \*gr)
- int [grava\\_movimento](#) (Gravados \*gr)
- int [anm](#) (Elem \*\*Tab, Gravados \*gr)
- int [freeMvs](#) (Gravados \*gr)
- int [freeJgs](#) (Gravados \*gr)
- int [freeIts](#) (Gravados \*gr)
- int [free\\_jogadasGR](#) (Gravados \*gr)

### 4.9.1 Detailed Description

Jogadas possiveis de efectuar e gravar do jogo e.

#### Author:

André Pimenta, João Gomes, Nelson Carvalho

#### Date:

1 Maio de 2010

## 4.9.2 Function Documentation

### 4.9.2.1 int anc (Elem \*\* *Tab*, Gravados \* *gr*)

Função responsável por anular uma cor guardada quando é chamada. A função vai anular a ultima cor guardada.

**Parameters:**

*Tab* tabuleiro onde vão ser efectuadas as mudanças de cor até chegar ao estado desejado.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

### 4.9.2.2 int anm (Elem \*\* *Tab*, Gravados \* *gr*)

Função responsável por anular um movimento quando é chamada. A função vai anular cores até chegar á ultima cor gravada quando foi guardado o movimento.

**Parameters:**

*Tab* tabuleiro onde vão ser efectuadas as mudanças de cor até chegar ao estado desejado.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

### 4.9.2.3 int E\_Par (Elem \*\* *Tab*, int *pos*, char *letra*, int \* *DIM*, char \* *local*)

Função auxiliar que verifica se existem pares de letras numa linha ou numa coluna Para tal vai percorrer a linha ou coluna e verifica se existe. A função tem em conta o caso de tres letras seguidas não ser um par mas sim um triplo.

**Parameters:**

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*pos* Linha ou coluna onde vai procurar os pares

*letra* a letra que se pretende saber se existe pares ou não

*local* Para decidir se se procura nas linhas ou nas colunas

**Returns:**

Inteiro que confirma se existe ou não um para na coluna/linha.

### 4.9.2.4 int E\_Par\_des (Elem \*\* *Tab*, char *letra1*, char *letra2*, int \* *DIM*, int *line1*, int *cl1*, int *line2*, int *cl2*)

Função auxiliar que verifica se existem pares de letras tanto nas linhas como nas colunas passdas como parametro. Para tal vai percorrer a linha ou coluna e verifica se existe.

**Parameters:**

*Tab* Tabuleiro do jogo

*letra1* Primeira letra da diagonal.

*letra2* Segunda letra da diagonal

*DIM* dimensão do tabuleiro

*line1* Linha da primeira letra

*line2* Linha da segunda letra

*cl1* Coluna da primeira letra

*cl2* Coluna da segunda letra

#### Returns:

Inteiro que confirma se existe ou não um para na coluna/linha.

#### 4.9.2.5 int E\_trp\_col (Elem \*\* *Tab*, int \* *DIM*, int *i*, int *j*, Gravados \* *gr*)

Função responsável por verificar se existe um triplo a partir das coordenadas *i* e *j* na vertical. Se existir atribui as cores correspondentes.

##### Parameters:

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*i* linha da célula.

*j* coluna da célula.

*gr* apontador para a estrutura de dados onde se guardam as jogadas e movimentos.

#### 4.9.2.6 int E\_trp\_row (Elem \*\* *Tab*, int \* *DIM*, int *i*, int *j*, Gravados \* *gr*)

Função responsável por verificar se existe um triplo a partir das coordenadas *i* e *j* na horizontal. Se existir atribui as cores correspondentes.

##### Parameters:

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*i* linha da célula.

*j* coluna da célula.

*gr* apontador para a estrutura de dados onde se guardam as jogadas e movimentos.

#### 4.9.2.7 int free\_jogadasGR (Gravados \* *gr*)

Função responsável por limpar da memória todos os dados guardados, em relação às cores e movimentos guardados.

##### Parameters:

*gr* apontador para a estrutura de dados onde se encontram guardados todos os movimentos e jogadas.

#### 4.9.2.8 int freeIts (Gravados \* *gr*)

Função responsável por limpar da memória todos os estados de tabuleiros marcados com o comando it.

**Parameters:**

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

#### 4.9.2.9 int freeJgs (Gravados \* *gr*)

Função responsável por limpar da memória todas as [jogadas](#) guardadas em memória.

**Parameters:**

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

#### 4.9.2.10 int freeMvs (Gravados \* *gr*)

Função responsável por limpar da memória todos os [movimentos](#) guardados em memória.

**Parameters:**

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

#### 4.9.2.11 int grava\_jogada (int *row*, int *col*, char *color*, Gravados \* *gr*)

Função que irá guardar a cor a linha e a coluna de um tabuleiro que correspondente á célula que vai ser substituída. Esta função será chamada sempre que se mude uma cor.

**Parameters:**

*row* linha da célula que se vai guardar.

*col* coluna da célula que se vai guardar.

*color* Cor da célula que vai ser substituída

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

#### 4.9.2.12 int grava\_movimento (Gravados \* *gr*)

Função que irá guardar na estrutura de dados de [movimentos](#) o número de cores que foram guardadas até ao momento da sua chamada. Para posteriormente poder se recuperar o estado do tabuleiro.

**Parameters:**

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

#### 4.9.2.13 int inicia\_GR (Gravados \* *gr*)

Função reponsavel por inicializar e alocar espaço para as estruturas de dados responsaveis por guardar informação relativa a [movimentos](#) e [jogadas](#) efectuadas. Aloca espaço para [It](#), para moviemntos e para [jogadas](#)

##### Parameters:

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

#### 4.9.2.14 int it (Gravados \* *gr*)

Função reonsavel tal como a guarda [movimentos](#), de guardar na estrutura de dados de IT's o numero de cores que foram guardadas até ao momento da sua chamada. Para postriormente poder se recoperar o estado do tabuleiro.

##### Parameters:

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

#### 4.9.2.15 int jogada (char *cor*, int *x*, int *y*, Elem \*\* *Tab*, Gravados \* *gr*)

A função jogada é responsável por alterar os dados do tabuleiro que são atribuidos automaticamente. Ela irá receber o novo dado e as poisções onde vai ficar o novo dado, actualizando assim o tabuleiro com os novos dados

##### Parameters:

*Tab* Tabuleiro do jogo

*cor* nova cor para um [elemento](#) do tabuleiro

*x* Corresponde á linha que vamos substituir no tabuleiro.

*y* Corresponde á coluna que vamos substituir no tabuleiro.

*gr* apontador para a estrutura de dados onde se guardas as [jogadas](#) e [movimentos](#).

##### Returns:

Retorna um inteiro que verifica se a jogada foi ou não bem sucedida.

#### 4.9.2.16 int jogadaManual (char *cor*, int *x*, int *y*, int \* *DIM*, Elem \*\* *Tab*, Gravados \* *gr*)

A função jogada é responsável por alterar os dados do tabuleiro introduzidos pelo utilizador. Ela irá receber o novo dado e as poisções onde vai ficar o novo dado, actualizando assim o tabuleiro com os novos dados

##### Parameters:

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*cor* nova cor para um [elemento](#) do tabuleiro

*x* Corresponde á linha que vamos substituir no tabuleiro.

*y* Corresponde á coluna que vamos substituir no tabuleiro.



*gr* apontador para a estrutura de dados onde se guardas as [jogadas](#) e [movimentos](#).

**Returns:**

Retorna um inteiro que verifica se a jogada foi ou não bem sucedida.

**4.9.2.17 int pds (Elem \*\* *Tab*, int \* *DIM*, int \* *flagAP*, int \* *flagAB*, Gravados \* *gr*)**

Função responsável por ver se existem ou não pares isolados no tabuleiro. Para tal esta função vai pegar em quatro células, formando um quadrado. Neste quadrado verifica se as duas diagonais pertencem ou não a regra pares desencontrados, com a ajuda da função auxiliar EpParDes. Caso pertence ver se a diagonal contrária tem cores contrárias, se tiver dá uma mensagem de [erro](#), se não muda as cores.

**Parameters:**

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*flagAP* flag para controlo do automatico preto.

*flagAB* flag para controlo do automatico branco.

*gr* apontador para a estrutura de dados onde se guarda as [jogadas](#) e [movimentos](#) efectuados.

**Returns:**

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#), é exibido através da função mensagem de [erro](#).

**4.9.2.18 int pis (Elem \*\* *Tab*, int \* *DIM*, int \* *flagAP*, int \* *flagAB*, Gravados \* *gr*)**

Função que procura pares isolados no tabuleiro (numa coluna ou numa linha). Para tal vai se percorrer todas as posições do tabuleiro, e ver se a letra é diferente das que a rodeiam na horizontal ou na vertical. Confirmando se a presença de uma letra isola, chama a função auxiliar Epar que vai ver se existe um par dessa letra na coluna ou na linha. Caso se verifiquem as duas condições irá ver se a letra não está de uma cor contrária à que se vai atribuir, caso não esteja atribui-se uma nova cor, se não uma mensagem de [erro](#).

**Parameters:**

*Tab* Tabuleiro do jogo

*flagAP* flag para controlo do automatico preto.

*flagAB* flag para controlo do automatico branco.

*gr* apontador para a estrutura de dados onde se guarda as [jogadas](#) e [movimentos](#) efectuados.

*DIM* dimensão do tabuleiro

**Returns:**

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#), é exibido através da função mensagem de [erro](#).

#### 4.9.2.19 int rb (Elem \*\* *Tab*, Gravados \* *gr*)

Função responsável por anular cores até chegar ao estado de tabuleiro marcado pela função *it*. Desta forma será possível retornar o estado do tabuleiro tal como foi marcado na função *it*.

##### Parameters:

*Tab* tabuleiro onde vão ser efectuadas as mudanças de cor até chegar ao estado desejado.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

#### 4.9.2.20 int snd (Elem \*\* *Tab*, int \* *DIM*, int \* *flagAP*, int \* *flagAB*, Gravados \* *gr*)

Função que procura sandes no tabuleiro(tres letras onde apenas a do meio é diferente na mesma linha/-coluna). Para tal vai se percorrer todas as posições do tabuleiro, ver caso existam quais letras de cima e de baixo, para verificar se existe triplo na vertical, e a da esquerda e a da direita para a horizontal. Confirmando se a presença de uma sandes de letras, atribui se as cores correctas caso nenhuma das celulas não tenha um cor contrária, se não envia uma mensagem de [erro](#).

##### Parameters:

*Tab* Tabuleiro do jogo

*flagAP* flag para controlo do automatico preto.

*flagAB* flag para controlo do automatico branco.

*gr* apontador para a estrutura de dados onde se guarda as [jogadas](#) e moviemntos efectuados.

*DIM* dimensão do tabuleiro

##### Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#),é exibido atraves da função mensagem de [erro](#).

#### 4.9.2.21 int trp (Elem \*\* *Tab*, int \* *DIM*, int \* *flagAP*, int \* *flagAB*, Gravados \* *gr*)

Função que procura triplos no tabuleiro(tres letras iguais na mesma linha/coluna). Para tal vai se percorrer todas as posições do tabuleiro, ver caso existam quais letras de cima e de baixo, para verificar se existe triplo na vertical, e a da esquerda e a da direita para a horizontal. É também tido em contam o caso de existirem quadroplos não serem considerados triplos. Confirmando se a presença de um triplo atribui se as cores correctas caso nenhuma das celulas não tenha um cor contrária, se não envia uma mensagem de [erro](#).

##### Parameters:

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*flagAP* flag para controlo do automatico preto.

*flagAB* flag para controlo do automatico branco.

*gr* apontador para a estrutura de dados onde se guarda as [jogadas](#) e moviemntos efectuados.

##### Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um [erro](#),é exibido atraves da função mensagem de [erro](#).

## 4.10 jogadas.h File Reference

### Classes

- struct [jogadas](#)  
*Estrutura responsavel por guardar a coluna,linha e cor de uma celula a gravar.*
- struct [movimentos](#)  
*Estrutura responsavel por guardar a posição a que corresponde o tabuleiro nas cores guardadas até ao momento que é invocada.*
- struct [It](#)  
*Estrutura responsavel por guardar a posição a que corresponde o tabuleiro nas cores guardadas até ao momento que é invocada.*
- struct [gravados](#)  
*Estrutura onde é armazenada toda a informação relativa a [movimentos](#) e [jogadas](#) guardadas.*

### Typedefs

- typedef struct [jogadas](#) [Jogadas](#)
- typedef struct [movimentos](#) [Movimentos](#)
- typedef struct [It](#) [IT](#)
- typedef struct [gravados](#) [Gravados](#)

### Functions

- int [grava\\_jogada](#) (int col, int row, char cor, [Gravados](#) \*gr)
- int [jogada](#) (char cor, int y, int x, [Elem](#) \*\*Tab, [Gravados](#) \*gr)
- int [E\\_Par\\_des](#) ([Elem](#) \*\*Tab, char letra1, char letra2, int \*DIM, int line1, int cl1, int line2, int col2)
- int [jogadaManual](#) (char cor, int x, int y, int \*DIM, [Elem](#) \*\*Tab, [Gravados](#) \*gr)
- int [pds](#) ([Elem](#) \*\*Tab, int \*DIM, int \*flagAP, int \*flagAB, [Gravados](#) \*gr)
- int [E\\_trp\\_row](#) ([Elem](#) \*\*Tab, int \*DIM, int i, int j, [Gravados](#) \*gr)
- int [E\\_trp\\_col](#) ([Elem](#) \*\*Tab, int \*DIM, int i, int j, [Gravados](#) \*gr)
- int [trp](#) ([Elem](#) \*\*Tab, int \*DIM, int \*flagAP, int \*flagAB, [Gravados](#) \*gr)
- int [snd](#) ([Elem](#) \*\*Tab, int \*DIM, int \*flagAP, int \*flagAB, [Gravados](#) \*gr)
- int [E\\_Par](#) ([Elem](#) \*\*Tab, int pos, char letra, int \*DIM, char \*local)
- int [pis](#) ([Elem](#) \*\*Tab, int \*DIM, int \*flagAP, int \*flagAB, [Gravados](#) \*gr)
- int [anc](#) ([Elem](#) \*\*Tab, [Gravados](#) \*gr)
- int [free\\_jogadasGR](#) ([Gravados](#) \*gr)
- int [it](#) ([Gravados](#) \*gr)
- int [rb](#) ([Elem](#) \*\*Tab, [Gravados](#) \*gr)
- int [inicia\\_GR](#) ([Gravados](#) \*gr)
- int [grava\\_movimento](#) ([Gravados](#) \*gr)
- int [anm](#) ([Elem](#) \*\*Tab, [Gravados](#) \*gr)

## 4.10.1 Typedef Documentation

### 4.10.1.1 typedef struct gravados Gravados

### 4.10.1.2 typedef struct It IT

### 4.10.1.3 typedef struct jogadas Jogadas

### 4.10.1.4 typedef struct movimentos Movimentos

## 4.10.2 Function Documentation

### 4.10.2.1 int anc (Elem \*\* *Tab*, Gravados \* *gr*)

Função responsável por anular uma cor guardada quando é chamada. A função vai anular a ultima cor guardada.

#### Parameters:

*Tab* tabuleiro onde vão ser efectuadas as mudanças de cor até chegar ao estado desejado.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

### 4.10.2.2 int anm (Elem \*\* *Tab*, Gravados \* *gr*)

Função responsável por anular um movimento quando é chamada. A função vai anular cores até chegar á ultima cor gravada quando foi guardado o movimento.

#### Parameters:

*Tab* tabuleiro onde vão ser efectuadas as mudanças de cor até chegar ao estado desejado.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

### 4.10.2.3 int E\_Par (Elem \*\* *Tab*, int *pos*, char *letra*, int \* *DIM*, char \* *local*)

Função auxiliar que verifica se existem pares de letras numa linha ou numa coluna Para tal vai percorrer a linha ou coluna e verifica se existe. A função tem em conta o caso de tres letras seguidas não ser um par mas sim um triplo.

#### Parameters:

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*pos* Linha ou coluna onde vai procurar os pares

*letra* a letra que se pretende saber se existe pares ou não

*local* Para decidir se se procura nas linhas ou nas colunas

#### Returns:

Inteiro que confirma se existe ou não um para na coluna/linha.

#### 4.10.2.4 int E\_Par\_des (Elem \*\* *Tab*, char *letra1*, char *letra2*, int \* *DIM*, int *line1*, int *cl1*, int *line2*, int *cl2*)

Função auxiliar que verifica se existem pares de letras tanto nas linhas como nas colunas passadas como parametro. Para tal vai percorrer a linha ou coluna e verifica se existe.

##### Parameters:

*Tab* Tabuleiro do jogo  
*letra1* Primeira letra da diagonal.  
*letra2* Segunda letra da diagonal  
*DIM* dimensão do tabuleiro  
*line1* Linha da primeira letra  
*line2* Linha da segunda letra  
*cl1* Coluna da primeira letra  
*cl2* Coluna da segunda letra

##### Returns:

Inteiro que confirma se existe ou não um par na coluna/linha.

#### 4.10.2.5 int E\_trp\_col (Elem \*\* *Tab*, int \* *DIM*, int *i*, int *j*, Gravados \* *gr*)

Função responsável por verificar se existe um triplo a partir das coordenadas *i* e *j* na vertical. Se existir atribui as cores correspondentes.

##### Parameters:

*Tab* Tabuleiro do jogo  
*DIM* dimensão do tabuleiro  
*i* linha da célula.  
*j* coluna da célula.  
*gr* apontador para a estrutura de dados onde se guardam as [jogadas](#) e [movimentos](#).

#### 4.10.2.6 int E\_trp\_row (Elem \*\* *Tab*, int \* *DIM*, int *i*, int *j*, Gravados \* *gr*)

Função responsável por verificar se existe um triplo a partir das coordenadas *i* e *j* na horizontal. Se existir atribui as cores correspondentes.

##### Parameters:

*Tab* Tabuleiro do jogo  
*DIM* dimensão do tabuleiro  
*i* linha da célula.  
*j* coluna da célula.  
*gr* apontador para a estrutura de dados onde se guardam as [jogadas](#) e [movimentos](#).

#### 4.10.2.7 int free\_jogadasGR (Gravados \* *gr*)

Função responsável por limpar da memória todos os dados guardados, em relação às cores e movimentos guardados.

**Parameters:**

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

#### 4.10.2.8 int grava\_jogada (int *row*, int *col*, char *color*, Gravados \* *gr*)

Função que irá guardar a cor a linha e a coluna de um tabuleiro que correspondente à célula que vai ser substituída. Esta função será chamada sempre que se mude uma cor.

**Parameters:**

*row* linha da célula que se vai guardar.

*col* coluna da célula que se vai guardar.

*color* Cor da célula que vai ser substituída

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

#### 4.10.2.9 int grava\_movimento (Gravados \* *gr*)

Função que irá guardar na estrutura de dados de [movimentos](#) o número de cores que foram guardadas até ao momento da sua chamada. Para posteriormente poder se recuperar o estado do tabuleiro.

**Parameters:**

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

#### 4.10.2.10 int inicia\_GR (Gravados \* *gr*)

Função responsável por inicializar e alocar espaço para as estruturas de dados responsáveis por guardar informação relativa a [movimentos](#) e [jogadas](#) efectuadas. Aloca espaço para [It](#), para movimentos e para [jogadas](#)

**Parameters:**

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

#### 4.10.2.11 int it (Gravados \* *gr*)

Função responsável tal como a guarda [movimentos](#), de guardar na estrutura de dados de [IT's](#) o número de cores que foram guardadas até ao momento da sua chamada. Para posteriormente poder se recuperar o estado do tabuleiro.

**Parameters:**

*gr* apontador para a estrutura de dados onde se encontram guardados todos os [movimentos](#) e [jogadas](#).

#### 4.10.2.12 int jogada (char *cor*, int *x*, int *y*, Elem \*\* *Tab*, Gravados \* *gr*)

A função jogada é responsável por alterar os dados do tabuleiro que são atribuídos automaticamente. Ela irá receber o novo dado e as posições onde vai ficar o novo dado, actualizando assim o tabuleiro com os novos dados

##### Parameters:

*Tab* Tabuleiro do jogo

*cor* nova cor para um elemento do tabuleiro

*x* Corresponde á linha que vamos substituir no tabuleiro.

*y* Corresponde á coluna que vamos substituir no tabuleiro.

*gr* apontador para a estrutura de dados onde se guardas as jogadas e movimentos.

##### Returns:

Retorna um inteiro que verifica se a jogada foi ou não bem sucedida.

#### 4.10.2.13 int jogadaManual (char *cor*, int *x*, int *y*, int \* *DIM*, Elem \*\* *Tab*, Gravados \* *gr*)

A função jogada é responsável por alterar os dados do tabuleiro introduzidos pelo utilizador. Ela irá receber o novo dado e as posições onde vai ficar o novo dado, actualizando assim o tabuleiro com os novos dados

##### Parameters:

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*cor* nova cor para um elemento do tabuleiro

*x* Corresponde á linha que vamos substituir no tabuleiro.

*y* Corresponde á coluna que vamos substituir no tabuleiro.

*gr* apontador para a estrutura de dados onde se guardas as jogadas e movimentos.

##### Returns:

Retorna um inteiro que verifica se a jogada foi ou não bem sucedida.

#### 4.10.2.14 int pds (Elem \*\* *Tab*, int \* *DIM*, int \* *flagAP*, int \* *flagAB*, Gravados \* *gr*)

Função responsável por ver se existem ou não pares isolados no tabuleiro. Para tal esta função vai pegar em quatro células, formando um quadrado. Neste quadrado verifica se as duas diagonais pertencem ou não a regra pares desconstruídos, com a ajuda da função auxiliar EpParDes. Caso pertence ver se a diagonal contrária tem cores contrárias, se tiver dá uma mensagem de erro, se não muda as cores.

##### Parameters:

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*flagAP* flag para controlo do automatico preto.

*flagAB* flag para controlo do automatico branco.

*gr* apontador para a estrutura de dados onde se guarda as jogadas e moviemntos efectuados.

**Returns:**

Inteiro que em caso de correr tudo bem retorna 0, caso exista um erro, é exibido através da função mensagem de erro.

**4.10.2.15 int pis (Elem \*\* Tab, int \* DIM, int \* flagAP, int \* flagAB, Gravados \* gr)**

Função que procura pares isolados no tabuleiro (numa coluna ou numa linha). Para tal vai se percorrer todas as posições do tabuleiro, e ver se a letra é diferente das que a rodeiam na horizontal ou na vertical. Confirmando se a presença de uma letra isola, chama a função auxiliar Epar que vai ver se existe um par dessa letra na coluna ou na linha. Caso se verifiquem as duas condições irá ver se a letra não está de uma cor contrária à que se vai atribuir, caso não esteja atribui se uma nova cor, se não uma mensagem de erro.

**Parameters:**

*Tab* Tabuleiro do jogo

*flagAP* flag para controlo do automatico preto.

*flagAB* flag para controlo do automatico branco.

*gr* apontador para a estrutura de dados onde se guarda as jogadas e moviemntos efectuados.

*DIM* dimensão do tabuleiro

**Returns:**

Inteiro que em caso de correr tudo bem retorna 0, caso exista um erro, é exibido através da função mensagem de erro.

**4.10.2.16 int rb (Elem \*\* Tab, Gravados \* gr)**

Função responsável por anular cores até chegar ao estado de tabuleiro marcado pela função it. Desta forma será possível retornar o estado do tabuleiro tal como foi marcado na função it.

**Parameters:**

*Tab* tabuleiro onde vão ser efectuadas as mudanças de cor até chegar ao estado desejado.

*gr* apontador para a estrutura de dados onde se encontram guardados todos os movimentos e jogadas.

**4.10.2.17 int snd (Elem \*\* Tab, int \* DIM, int \* flagAP, int \* flagAB, Gravados \* gr)**

Função que procura sandes no tabuleiro (três letras onde apenas a do meio é diferente na mesma linha/-coluna). Para tal vai se percorrer todas as posições do tabuleiro, ver caso existam quais letras de cima e de baixo, para verificar se existe triplo na vertical, e a da esquerda e a da direita para a horizontal. Confirmando se a presença de uma sandes de letras, atribui se as cores correctas caso nenhuma das células não tenha um cor contrária, se não envia uma mensagem de erro.

**Parameters:**

*Tab* Tabuleiro do jogo

*flagAP* flag para controlo do automatico preto.



*flagAB* flag para controlo do automatico branco.

*gr* apontador para a estrutura de dados onde se guarda as jogadas e moviemntos efectuados.

*DIM* dimensão do tabuleiro

#### Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um erro, é exibido através da função mensagem de erro.

#### 4.10.2.18 int trp (Elem \*\* Tab, int \* DIM, int \* flagAP, int \* flagAB, Gravados \* gr)

Função que procura triplos no tabuleiro (tres letras iguais na mesma linha/coluna). Para tal vai se percorrer todas as posições do tabuleiro, ver caso existam quais letras de cima e de baixo, para verificar se existe triplo na vertical, e a da esquerda e a da direita para a horizontal. É também tido em conta o caso de existirem quadroplos não serem considerados triplos. Confirmando se a presença de um triplo atribui se as cores correctas caso nenhuma das células não tenha um cor contraria, se não envia uma mensagem de erro.

#### Parameters:

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*flagAP* flag para controlo do automatico preto.

*flagAB* flag para controlo do automatico branco.

*gr* apontador para a estrutura de dados onde se guarda as jogadas e moviemntos efectuados.

#### Returns:

Inteiro que em caso de correr tudo bem retorna 0, caso exista um erro, é exibido através da função mensagem de erro.

## 4.11 main.c File Reference

```
main do jogo #include "tabuleiro.h"
#include "jogadas.h"
#include "comandos.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <readline/readline.h>
#include <readline/history.h>
```

### Functions

- `int main ()`

#### 4.11.1 Detailed Description

main do jogo

##### Author:

André Pimenta, João Gomes, Nelson Carvalho

##### Date:

5 Março de 2010

#### 4.11.2 Function Documentation

##### 4.11.2.1 `int main ()`

A função main é um ciclo que apenas termina quando o utilizador insere o comando "q" Esta função vai então ser responsável por receber os comandos continuamente, e imprimir o resultado enquanto o jogo decorrer

## 4.12 tabuleiro.c File Reference

```
Tabuleiro do jogo. #include "tabuleiro.h"
#include "erro.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <readline/readline.h>
#include <readline/history.h>
```

### Functions

- int [dimensao](#) (FILE \*fp, int \*DIM)
- int [ficheiroValido](#) (FILE \*fp, int \*DIM)
- void [freeTab](#) (int \*DIM, Elem \*\*Tab)
- int [lerficheiro](#) (int \*DIM, Elem \*\*Tab, FILE \*fp)
- int [criarTabuleiro](#) (char \*nome, int \*DIM, Elem \*\*Tab)
- void [printTabuleiro](#) (int \*DIM, Elem \*\*Tab)
- int [gravaJogo](#) (char \*nome, int \*DIM, Elem \*\*Tab)

### 4.12.1 Detailed Description

Tabuleiro do jogo.

#### Author:

André Pimenta, João Gomes, Nelson Carvalho

#### Date:

5 Março de 2010

### 4.12.2 Function Documentation

#### 4.12.2.1 int criarTabuleiro (char \* *nome*, int \* *DIM*, Elem \*\* *Tab*)

#### 4.12.2.2 int dimensao (FILE \* *fp*, int \* *DIM*)

A função `dimensao` é responsável por carregar a dimensão do tabuleiro que estamos a abrir, para a variável `DIM`. Para isso ela usa o `fscanf` da biblioteca `stdio`.

#### Parameters:

*DIM* dimensão do tabuleiro

*fp* Recebe o ficheiro do qual irá verificar se existe o valor da dim e passa-lo para a variável Global `DIM`.

#### Returns:

retorna um valor lógico 1 ou 0 que confirma se conseguiu ou não ler o valor correspondente á dimensão.

#### 4.12.2.3 int `ficheiroValido` (FILE \* *fp*, int \* *DIM*)

A função `ficheiroValido` recebe o ficheiro do qual queremos importar o tabuleiro e verifica se este é válido para o nosso jogo. Para ser válido irá verificar se tem o número de elementos correspondentes a dimensão, se são elementos do tipo predefinido do tabuleiro(ex: "[A]")

##### Parameters:

*DIM* dimensão do tabuleiro

*fp* Recebe o ficheiro que está aberto na função `criarTabuleiro`

##### Returns:

Retorna um inteiro que irá confirmar se o ficheiro é válido ou não.

#### 4.12.2.4 void `freeTab` (int \* *DIM*, Elem \*\* *Tab*)

A função `freeTab` irá apagar da memória do tabuleiro antigo caso estejamos a abrir um novo tabuleiro.

##### Parameters:

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

##### Remarks:

Esta função apenas é usada se existir um tabuleiro carregado e queremos carregar outro.

#### 4.12.2.5 int `gravaJogo` (char \* *nome*, int \* *DIM*, Elem \*\* *Tab*)

A função `gravaJogo` é responsável por gravar o nosso tabuleiro num ficheiro cujo o nome é dado como argumento e que está no formato ".ltr". O tabuleiro será imprimido para o ficheiro seguindo as regras de representação deste, para que possa ser lido de novo e seja considerado válido.

##### Parameters:

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*nome* Recebe uma string que representa o nome que vai dar ao ficheiro que vai guardar

#### 4.12.2.6 int `lerficheiro` (int \* *DIM*, Elem \*\* *Tab*, FILE \* *fp*)

A função `criarTabuleiro` recebe o nome do ficheiro que queremos abrir e irá tentar abrir este. Depois irá chamar a função `freeTab` caso exista já um tabuleiro carregado e em seguida chama a função `ficheiroValido` para ver verificar se o mesmo é válido. Após todas verificações a função cria espaço para o tabuleiro na memória e carrega este para lá, ficando guardado no apontador *Tab*. Os valores lidos do ficheiro irão ser associados por duas características, a letra e a cor, esta última será "i" correspondente a indefinida caso a letra esteja rodeada por dois espaços brancos, preta caso esteja rodeada por dois parênteses rectos e branca caso esteja rodeada por dois parênteses curvos.

**Parameters:**

*Tab* Tabuleiro do jogo  
*DIM* dimensão do tabuleiro  
*fp* ficheiro a ser lido

**Returns:**

Retorna um inteiro que verifica se conseguiu ou não criar o Tabuleiro.

**4.12.2.7 void printTabuleiro (int \* *DIM*, Elem \*\* *Tab*)**

A função printTabuleiro é responsável por imprimir no ecrã o nosso tabuleiro.

**Parameters:**

*Tab* Tabuleiro do jogo  
*DIM* dimensão do tabuleiro

## 4.13 tabuleiro.h File Reference

header file do [tabuleiro.c](#) `#include <stdio.h>`

### Classes

- struct [elemento](#)

*É constituída por dois chars, que correspondem às características de cada célula do tabuleiro, a cor e a letra.*

### Typedefs

- typedef struct [elemento](#) [Elemento](#)
- typedef struct [elemento](#) \* [Elem](#)

### Functions

- int [dimensao](#) (FILE \*fp, int \*DIM)
- int [ficheiroValido](#) (FILE \*fp, int \*DIM)
- void [freeTab](#) (int \*DIM, [Elem](#) \*\*Tab)
- int [lerficheiro](#) (int \*DIM, [Elem](#) \*\*Tab, FILE \*fp)
- int [criarTabuleiro](#) (char \*nome, int \*DIM, [Elem](#) \*\*Tab)
- void [printTabuleiro](#) (int \*DIM, [Elem](#) \*\*Tab)
- int [gravaJogo](#) (char \*nome, int \*DIM, [Elem](#) \*\*Tab)

#### 4.13.1 Detailed Description

header file do [tabuleiro.c](#)

##### Author:

André Pimenta,Miguel Gomes,Nelson Cravalho

##### Date:

5 Março de 2010

#### 4.13.2 Typedef Documentation

##### 4.13.2.1 typedef struct elemento\* Elem

##### 4.13.2.2 typedef struct elemento Elemento

#### 4.13.3 Function Documentation

##### 4.13.3.1 int criarTabuleiro (char \* nome, int \* DIM, Elem \*\* Tab)

##### 4.13.3.2 int dimensao (FILE \* fp, int \* DIM)

A função `dimensao` é responsável por carregar a dimensão do tabuleiro que estamos a abrir, para a variável `DIM`. Para isso ela usa o `fscanf` da biblioteca `stdio`.

**Parameters:**

*DIM* dimensão do tabuleiro

*fp* Recebe o ficheiro do qual irá verificar se existe o valor da dim e passa-lo para a variável Global DIM.

**Returns:**

retorna um valor lógico 1 ou 0 que confirma se conseguiu ou não ler o valor correspondente á dimensão.

**4.13.3.3 int ficheiroValido (FILE \* *fp*, int \* *DIM*)**

A função ficheiroValido recebe o ficheiro do qual queremos importar o tabuleiro e verifica se este é válido para o nosso jogo. Para ser válido irá verficar se tem o número de elementos correspondentes a dimensão, se são elementos do tipo predefenido do tabuleiro(ex: "[A]")

**Parameters:**

*DIM* dimensão do tabuleiro

*fp* Recebe o ficheiro que está aberto na função criarTabuleiro

**Returns:**

Retorna um inteiro que irá confirmar se o ficheiro é válido ou não.

**4.13.3.4 void freeTab (int \* *DIM*, Elem \*\* *Tab*)**

A função freeTab irá apagar da memória do tabuleiro antigo caso estejamos a abrir um novo tabuleiro.

**Parameters:**

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

**Remarks:**

Esta função apenas é usada se existir um tabuleiro carregado e queremos carregar outro.

**4.13.3.5 int gravaJogo (char \* *nome*, int \* *DIM*, Elem \*\* *Tab*)**

A função gravaJogo é responsável por gravar o nosso tabuleiro num ficheiro cujo o nome é dado como argumento e que está no formato ".ltr". O tabuleiro será imprimido para o ficheiro seguindo as regras de representação deste, para que possa ser lido de novo e seja considerado válido.

**Parameters:**

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*nome* Recebe uma string que representa o nome que vai dar ao ficheiro que vai guardar

#### 4.13.3.6 `int lerficheiro (int * DIM, Elem ** Tab, FILE * fp)`

A função `criarTabuleiro` recebe o nome do ficheiro que queremos abrir e irá tentar abrir este. Depois irá chamar a função `freeTab` caso exista já um tabuleiro carregado e em seguida chama a função `ficheiroValido` para ver verificar se o mesmo é válido. Após todas verificações a função cria espaço para o tabuleiro na memória e carrega este para lá, ficando guardado no apontador `Tab`. Os valores lidos do ficheiro irão ser associados por duas características, a letra e a cor, esta última será "i" correspondente a indefinida caso a letra esteja rodeada por dois espaços brancos, preta caso esteja rodeada por dois parênteses rectos e branca caso esteja rodeada por dois parênteses curvos.

##### Parameters:

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro

*fp* ficheiro a ser lido

##### Returns:

Retorna um inteiro que verifica se conseguiu ou não criar o Tabuleiro.

#### 4.13.3.7 `void printTabuleiro (int * DIM, Elem ** Tab)`

A função `printTabuleiro` é responsável por imprimir no ecrã o nosso tabuleiro.

##### Parameters:

*Tab* Tabuleiro do jogo

*DIM* dimensão do tabuleiro



# Index

- ajuda
  - ajuda.c, [12](#)
  - ajuda.h, [14](#)
- ajuda.c, [11](#)
  - ajuda, [12](#)
  - auto\_black, [12](#)
  - auto\_white, [12](#)
  - CHUNK\_SIZE, [12](#)
  - control, [12](#)
  - p\_branças, [13](#)
  - p\_preta, [13](#)
- ajuda.h, [14](#)
  - ajuda, [14](#)
  - auto\_black, [14](#)
  - auto\_white, [14](#)
  - control, [14](#)
  - p\_branças, [15](#)
  - p\_preta, [15](#)
- anc
  - jogadas.c, [29](#)
  - jogadas.h, [36](#)
- anm
  - jogadas.c, [29](#)
  - jogadas.h, [36](#)
- auto\_black
  - ajuda.c, [12](#)
  - ajuda.h, [14](#)
- auto\_white
  - ajuda.c, [12](#)
  - ajuda.h, [14](#)
- caminho
  - estado\_tabuleiro.c, [24](#)
  - estado\_tabuleiro.h, [26](#)
- CHUNK\_SIZE
  - ajuda.c, [12](#)
- col
  - jogadas, [9](#)
- color
  - jogadas, [9](#)
- comandos.c, [16](#)
  - executa\_comando, [16](#)
  - executaCriaJogo, [17](#)
  - executaGravaJogo, [17](#)
  - executaJogada, [17](#)
- comandos.h, [19](#)
  - executa\_comando, [19](#)
  - executaCriaJogo, [19](#)
  - executaGravaJogo, [19](#)
  - executaJogada, [20](#)
- control
  - ajuda.c, [12](#)
  - ajuda.h, [14](#)
- cor
  - elemento, [5](#)
- criarTabuleiro
  - tabuleiro.c, [43](#)
  - tabuleiro.h, [46](#)
- dimensao
  - tabuleiro.c, [43](#)
  - tabuleiro.h, [46](#)
- E\_ARGS
  - erro.h, [22](#)
- E\_COMMAND
  - erro.h, [22](#)
- E\_COORDS
  - erro.h, [22](#)
- E\_FILE\_FORMAT
  - erro.h, [22](#)
- E\_INVARGS
  - erro.h, [22](#)
- E\_NO\_BOARD
  - erro.h, [22](#)
- E\_NO\_FILE
  - erro.h, [22](#)
- E\_NO\_MOVES
  - erro.h, [22](#)
- E\_NOERR
  - erro.h, [22](#)
- E\_Par
  - jogadas.c, [29](#)
  - jogadas.h, [36](#)
- E\_Par\_des
  - jogadas.c, [29](#)
  - jogadas.h, [36](#)
- E\_SAVE
  - erro.h, [22](#)
- E\_trp\_col

- jogadas.c, 30
- jogadas.h, 37
- E\_trp\_row
  - jogadas.c, 30
  - jogadas.h, 37
- E\_WRONG\_SOLUTION
  - erro.h, 22
- Elem
  - tabuleiro.h, 46
- Elemento
  - tabuleiro.h, 46
- elemento, 5
  - cor, 5
  - letra, 5
- err\_code
  - erro, 6
- ERRO
  - erro.c, 21
- erro, 6
  - err\_code, 6
  - mensagem, 6
- erro.c, 21
  - ERRO, 21
  - mensagem\_de\_erro, 21
- erro.h, 22
  - E\_ARGS, 22
  - E\_COMMAND, 22
  - E\_COORDS, 22
  - E\_FILE\_FORMAT, 22
  - E\_INVARGS, 22
  - E\_NO\_BOARD, 22
  - E\_NO\_FILE, 22
  - E\_NO\_MOVES, 22
  - E\_NOERR, 22
  - E\_SAVE, 22
  - E\_WRONG\_SOLUTION, 22
  - mensagem\_de\_erro, 22
- estado\_tabuleiro.c, 24
  - caminho, 24
  - free\_tab, 24
  - prepara\_tab, 24
  - st, 25
  - vb, 25
  - vl, 25
  - vp, 25
- estado\_tabuleiro.h, 26
  - caminho, 26
  - free\_tab, 26
  - prepara\_tab, 26
  - st, 26
  - vb, 26
  - vl, 26
  - vp, 27
- executa\_comando
  - comandos.c, 16
  - comandos.h, 19
- executaCriaJogo
  - comandos.c, 17
  - comandos.h, 19
- executaGravaJogo
  - comandos.c, 17
  - comandos.h, 19
- executaJogada
  - comandos.c, 17
  - comandos.h, 20
- ficheiroValido
  - tabuleiro.c, 43
  - tabuleiro.h, 47
- free\_jogadasGR
  - jogadas.c, 30
  - jogadas.h, 37
- free\_tab
  - estado\_tabuleiro.c, 24
  - estado\_tabuleiro.h, 26
- freeIts
  - jogadas.c, 30
- freeJgs
  - jogadas.c, 31
- freeMvs
  - jogadas.c, 31
- freeTab
  - tabuleiro.c, 44
  - tabuleiro.h, 47
- grava\_jogada
  - jogadas.c, 31
  - jogadas.h, 38
- grava\_movimento
  - jogadas.c, 31
  - jogadas.h, 38
- Gravados
  - jogadas.h, 36
- gravados, 7
  - its, 7
  - jgs, 7
  - mvs, 7
  - totalGr, 7
- gravaJogo
  - tabuleiro.c, 44
  - tabuleiro.h, 47
- inicia\_GR
  - jogadas.c, 31
  - jogadas.h, 38
- IT
  - jogadas.h, 36
- It, 8

- next, 8
- pos, 8
- it
  - jogadas.c, 32
  - jogadas.h, 38
- its
  - gravados, 7
- jgs
  - gravados, 7
- jogada
  - jogadas.c, 32
  - jogadas.h, 38
- jogadaManual
  - jogadas.c, 32
  - jogadas.h, 39
- Jogadas
  - jogadas.h, 36
- jogadas, 9
  - col, 9
  - color, 9
  - next, 9
  - row, 9
- jogadas.c, 28
  - anc, 29
  - anm, 29
  - E\_Par, 29
  - E\_Par\_des, 29
  - E\_trp\_col, 30
  - E\_trp\_row, 30
  - free\_jogadasGR, 30
  - freeIts, 30
  - freeJgs, 31
  - freeMvs, 31
  - grava\_jogada, 31
  - grava\_movimento, 31
  - inicia\_GR, 31
  - it, 32
  - jogada, 32
  - jogadaManual, 32
  - pds, 33
  - pis, 33
  - rb, 33
  - snd, 34
  - trp, 34
- jogadas.h, 35
  - anc, 36
  - anm, 36
  - E\_Par, 36
  - E\_Par\_des, 36
  - E\_trp\_col, 37
  - E\_trp\_row, 37
  - free\_jogadasGR, 37
  - grava\_jogada, 38
  - grava\_movimento, 38
  - Gravados, 36
  - inicia\_GR, 38
  - IT, 36
  - it, 38
  - jogada, 38
  - jogadaManual, 39
  - Jogadas, 36
  - Movimentos, 36
  - pds, 39
  - pis, 40
  - rb, 40
  - snd, 40
  - trp, 41
- lerficheiro
  - tabuleiro.c, 44
  - tabuleiro.h, 47
- letra
  - elemento, 5
- main
  - main.c, 42
- main.c, 42
  - main, 42
- mensagem
  - erro, 6
- mensagem\_de\_erro
  - erro.c, 21
  - erro.h, 22
- Movimentos
  - jogadas.h, 36
- movimentos, 10
  - next, 10
  - pos, 10
- mvs
  - gravados, 7
- next
  - It, 8
  - jogadas, 9
  - movimentos, 10
- p\_branças
  - ajuda.c, 13
  - ajuda.h, 15
- p\_preta
  - ajuda.c, 13
  - ajuda.h, 15
- pds
  - jogadas.c, 33
  - jogadas.h, 39
- pis
  - jogadas.c, 33

- jogadas.h, 40
- pos
  - It, 8
  - movimentos, 10
- prepara\_tab
  - estado\_tabuleiro.c, 24
  - estado\_tabuleiro.h, 26
- printTabuleiro
  - tabuleiro.c, 45
  - tabuleiro.h, 48
- rb
  - jogadas.c, 33
  - jogadas.h, 40
- row
  - jogadas, 9
- snd
  - jogadas.c, 34
  - jogadas.h, 40
- st
  - estado\_tabuleiro.c, 25
  - estado\_tabuleiro.h, 26
- tabuleiro.c, 43
  - criarTabuleiro, 43
  - dimensao, 43
  - ficheiro Valido, 43
  - freeTab, 44
  - gravaJogo, 44
  - lerficheiro, 44
  - printTabuleiro, 45
- tabuleiro.h, 46
  - criarTabuleiro, 46
  - dimensao, 46
  - Elem, 46
  - Elemento, 46
  - ficheiro Valido, 47
  - freeTab, 47
  - gravaJogo, 47
  - lerficheiro, 47
  - printTabuleiro, 48
- totalGr
  - gravados, 7
- trp
  - jogadas.c, 34
  - jogadas.h, 41
- vb
  - estado\_tabuleiro.c, 25
  - estado\_tabuleiro.h, 26
- vl
  - estado\_tabuleiro.c, 25
  - estado\_tabuleiro.h, 26
- vp
  - estado\_tabuleiro.c, 25
  - estado\_tabuleiro.h, 27