

## Documentação Xadrez

Gerado por Doxygen 1.9.4



<b>1 Índice Hierárquico</b>	<b>1</b>
1.1 Hierarquia de Classes	1
<b>2 Índice dos Componentes</b>	<b>3</b>
2.1 Lista de Classes	3
<b>3 Índice dos Arquivos</b>	<b>5</b>
3.1 Lista de Arquivos	5
<b>4 Classes</b>	<b>7</b>
4.1 Referência da Classe Bispo	7
4.1.1 Descrição detalhada	7
4.1.2 Construtores e Destrutores	7
4.1.2.1 Bispo()	7
4.1.2.2 ~Bispo()	8
4.1.3 Funções membros	8
4.1.3.1 get_representacao()	8
4.2 Referência da Classe Cavalo	8
4.2.1 Descrição detalhada	9
4.2.2 Construtores e Destrutores	9
4.2.2.1 Cavalo()	9
4.2.2.2 ~Cavalo()	9
4.2.3 Funções membros	9
4.2.3.1 get_representacao()	9
4.3 Referência da Classe Dama	10
4.3.1 Descrição detalhada	10
4.3.2 Construtores e Destrutores	10
4.3.2.1 Dama()	10
4.3.2.2 ~Dama()	10
4.3.3 Funções membros	11
4.3.3.1 get_representacao()	11
4.4 Referência da Classe ErroAoAbrirException	11
4.4.1 Funções membros	11
4.4.1.1 what()	11
4.5 Referência da Classe ErroAoFecharException	12
4.5.1 Funções membros	12
4.5.1.1 what()	12
4.6 Referência da Classe ErroDeJogoException	12
4.6.1 Funções membros	12
4.6.1.1 what()	13
4.7 Referência da Classe ErroDeLogicaException	13
4.7.1 Funções membros	13
4.7.1.1 what()	13

4.8 Referência da Classe ErroDeRoqueException . . . . .	13
4.8.1 Funções membros . . . . .	14
4.8.1.1 what() . . . . .	14
4.9 Referência da Classe ErroPgnException . . . . .	14
4.9.1 Funções membros . . . . .	14
4.9.1.1 what() . . . . .	14
4.10 Referência da Classe EscritorDePgn . . . . .	15
4.10.1 Construtores e Destrutores . . . . .	15
4.10.1.1 EscritorDePgn() . . . . .	15
4.10.1.2 ~EscritorDePgn() . . . . .	15
4.10.2 Funções membros . . . . .	15
4.10.2.1 abre_arquivo() . . . . .	16
4.10.2.2 fecha_arquivo() . . . . .	16
4.11 Referência da Classe ForaDoMapaException . . . . .	16
4.11.1 Funções membros . . . . .	16
4.11.1.1 what() . . . . .	16
4.12 Referência da Classe Jogador . . . . .	17
4.12.1 Construtores e Destrutores . . . . .	17
4.12.1.1 Jogador() . . . . .	17
4.12.1.2 ~Jogador() . . . . .	17
4.12.2 Funções membros . . . . .	18
4.12.2.1 atualiza_pecas() . . . . .	18
4.12.2.2 constroi_pecas() . . . . .	18
4.12.2.3 get_cor() . . . . .	18
4.12.2.4 get_pecas() . . . . .	18
4.12.2.5 imprime() . . . . .	18
4.12.2.6 zera_pecas() . . . . .	19
4.13 Referência da Classe Jogo . . . . .	19
4.13.1 Descrição detalhada . . . . .	19
4.13.2 Construtores e Destrutores . . . . .	19
4.13.2.1 Jogo() . . . . .	20
4.13.2.2 ~Jogo() . . . . .	20
4.13.3 Funções membros . . . . .	20
4.13.3.1 ativo() . . . . .	20
4.13.3.2 encerrar() . . . . .	20
4.13.3.3 imprime() . . . . .	20
4.13.3.4 jogada() . . . . .	20
4.13.3.5 processa_jogada() . . . . .	20
4.13.3.6 processa_terminal() . . . . .	21
4.14 Referência da Classe LeitorDePgn . . . . .	21
4.14.1 Construtores e Destrutores . . . . .	21
4.14.1.1 LeitorDePgn() . . . . .	21

4.14.1.2 ~LeitorDePgn()	22
4.14.2 Funções membros	22
4.14.2.1 abre_arquivo()	22
4.14.2.2 fecha_arquivo()	22
4.14.2.3 imprime_comandos()	22
4.15 Referência da Classe Movimento	23
4.15.1 Construtores e Destrutores	23
4.15.1.1 Movimento()	23
4.15.2 Funções membros	24
4.15.2.1 checa_movimento_bispo()	24
4.15.2.2 checa_movimento_cavalo()	24
4.15.2.3 checa_movimento_dama()	24
4.15.2.4 checa_movimento_peao()	24
4.15.2.5 checa_movimento_peca()	24
4.15.2.6 checa_movimento_rei()	25
4.15.2.7 checa_movimento_torre()	25
4.15.2.8 executar_movimento()	25
4.15.2.9 promove_peao()	25
4.15.2.10 roque()	25
4.15.2.11 validar_movimento()	26
4.16 Referência da Classe MovimentoInvalidoException	26
4.16.1 Funções membros	26
4.16.1.1 what()	26
4.17 Referência da Classe NaoESuaVezException	26
4.17.1 Funções membros	27
4.17.1.1 what()	27
4.18 Referência da Classe Peao	27
4.18.1 Descrição detalhada	27
4.18.2 Construtores e Destrutores	27
4.18.2.1 Peao()	27
4.18.2.2 ~Peao()	28
4.18.3 Funções membros	28
4.18.3.1 get_representacao()	28
4.18.4 Atributos	28
4.18.4.1 en_passant	28
4.18.4.2 primeiro_movimento	28
4.19 Referência da Classe Peca	29
4.19.1 Descrição detalhada	29
4.19.2 Construtores e Destrutores	29
4.19.2.1 Peca()	29
4.19.2.2 ~Peca()	30
4.19.3 Funções membros	30

4.19.3.1	get_casas()	30
4.19.3.2	get_cor()	30
4.19.3.3	get_en_passant()	30
4.19.3.4	get_primeiro_movimento()	31
4.19.3.5	get_representacao()	31
4.19.3.6	set_casas()	31
4.19.3.7	set_en_passant()	31
4.19.3.8	set_primeiro_movimento()	32
4.20	Referência da Classe PecaAliadaException	32
4.20.1	Funções membros	32
4.20.1.1	what()	32
4.21	Referência da Classe PecaNaFrenteException	32
4.21.1	Funções membros	33
4.21.1.1	what()	33
4.22	Referência da Classe Pgn	33
4.22.1	Construtores e Destrutores	33
4.22.1.1	Pgn()	33
4.22.1.2	~Pgn()	34
4.22.2	Funções membros	34
4.22.2.1	abre_arquivo()	34
4.22.2.2	fecha_arquivo()	34
4.22.3	Atributos	34
4.22.3.1	casa_final	34
4.22.3.2	comandos_da_rodada	35
4.22.3.3	conteudo	35
4.22.3.4	jogo_valido	35
4.22.3.5	nome_do_arquivo	35
4.22.3.6	rodada	35
4.23	Referência da Classe Rei	35
4.23.1	Descrição detalhada	36
4.23.2	Construtores e Destrutores	36
4.23.2.1	Rei()	36
4.23.2.2	~Rei()	36
4.23.3	Funções membros	36
4.23.3.1	get_representacao()	36
4.24	Referência da Classe SemPecaParaMovimentarException	37
4.24.1	Funções membros	37
4.24.1.1	what()	37
4.25	Referência da Classe Tabuleiro	37
4.25.1	Construtores e Destrutores	38
4.25.1.1	Tabuleiro()	38
4.25.1.2	~Tabuleiro()	38

4.25.2 Funções membros	38
4.25.2.1 adiciona_peca()	38
4.25.2.2 adiciona_peca_existente()	39
4.25.2.3 constroi_tabuleiro()	39
4.25.2.4 get_casa()	39
4.25.2.5 get_tamanho()	39
4.25.2.6 imprime()	39
4.25.2.7 limpa_casa()	39
4.25.2.8 limpa_tabuleiro()	40
4.26 Referência da Classe Torre	40
4.26.1 Descrição detalhada	40
4.26.2 Construtores e Destrutores	40
4.26.2.1 Torre()	40
4.26.2.2 ~Torre()	41
4.26.3 Funções membros	41
4.26.3.1 get_representacao()	41
<b>5 Arquivos</b>	<b>43</b>
5.1 Referência do Arquivo include/bispo.hpp	43
5.2 bispo.hpp	43
5.3 Referência do Arquivo include/cavalo.hpp	43
5.4 cavalo.hpp	44
5.5 Referência do Arquivo include/dama.hpp	44
5.6 dama.hpp	44
5.7 Referência do Arquivo include/excecoes.hpp	45
5.8 excecoes.hpp	45
5.9 Referência do Arquivo include/jogador.hpp	46
5.10 jogador.hpp	47
5.11 Referência do Arquivo include/jogo.hpp	47
5.12 jogo.hpp	48
5.13 Referência do Arquivo include/movimento.hpp	48
5.14 movimento.hpp	48
5.15 Referência do Arquivo include/peao.hpp	49
5.16 peao.hpp	49
5.17 Referência do Arquivo include/peca.hpp	50
5.17.1 Enumerações	50
5.17.1.1 Cor	50
5.18 peca.hpp	50
5.19 Referência do Arquivo include/pgn.hpp	51
5.20 pgn.hpp	51
5.21 Referência do Arquivo include/rei.hpp	52
5.22 rei.hpp	52

5.23 Referência do Arquivo <code>include/tabuleiro.hpp</code> . . . . .	53
5.23.1 Definições e macros . . . . .	53
5.23.1.1 <code>TAMANHO_DO_TABULEIRO</code> . . . . .	53
5.24 <code>tabuleiro.hpp</code> . . . . .	54
5.25 Referência do Arquivo <code>include/torre.hpp</code> . . . . .	54
5.26 <code>torre.hpp</code> . . . . .	54
<b>Índice Remissivo</b>	<b>57</b>



# Capítulo 1

## Índice Hierárquico

### 1.1 Hierarquia de Classes

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

std::exception	
ErroDeJogoException . . . . .	12
ForaDoMapaException . . . . .	16
MovimentoInvalidoException . . . . .	26
NaoESuaVezException . . . . .	26
PecaAliadaException . . . . .	32
PecaNaFrenteException . . . . .	32
SemPecaParaMovimentarException . . . . .	37
ErroDeLogicaException . . . . .	13
ErroDeRoqueException . . . . .	13
ErroPgnException . . . . .	14
ErroAoAbrirException . . . . .	11
ErroAoFecharException . . . . .	12
Jogador . . . . .	17
Jogo . . . . .	19
Movimento . . . . .	23
Peca . . . . .	29
Bispo . . . . .	7
Cavalo . . . . .	8
Dama . . . . .	10
Peao . . . . .	27
Rei . . . . .	35
Torre . . . . .	40
Pgn . . . . .	33
EscritorDePgn . . . . .	15
LeitorDePgn . . . . .	21
Tabuleiro . . . . .	37



## Capítulo 2

# Índice dos Componentes

### 2.1 Lista de Classes

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

Bispo	Classe que define o comportamento do bispo . . . . .	7
Cavalo	Classe que define o comportamento do cavalo . . . . .	8
Dama	Classe que define o comportamento da dama . . . . .	10
ErroAoAbrirException	. . . . .	11
ErroAoFecharException	. . . . .	12
ErroDeJogoException	. . . . .	12
ErroDeLogicaException	. . . . .	13
ErroDeRoqueException	. . . . .	13
ErroPgnException	. . . . .	14
EscritorDePgn	. . . . .	15
ForaDoMapaException	. . . . .	16
Jogador	. . . . .	17
Jogo	Define os funcionamentos de um jogo padrao . . . . .	19
LeitorDePgn	. . . . .	21
Movimento	. . . . .	23
MovimentoInvalidoException	. . . . .	26
NaoESuaVezException	. . . . .	26
Peao	Classe que define o comportamento do peao . . . . .	27
Peca	Classe que cuida de todas as funcionalidades gerais de uma peca . . . . .	29
PecaAliadaException	. . . . .	32
PecaNaFrenteException	. . . . .	32
Pgn	. . . . .	33
Rei	Classe que define o comportamento do rei . . . . .	35
SemPecaParaMovimentarException	. . . . .	37
Tabuleiro	. . . . .	37
Torre	Classe que define o comportamento da torre . . . . .	40



## Capítulo 3

# Índice dos Arquivos

### 3.1 Lista de Arquivos

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

<a href="#">include/bispo.hpp</a>	43
<a href="#">include/cavalo.hpp</a>	43
<a href="#">include/dama.hpp</a>	44
<a href="#">include/excecoes.hpp</a>	45
<a href="#">include/jogador.hpp</a>	46
<a href="#">include/jogo.hpp</a>	47
<a href="#">include/movimento.hpp</a>	48
<a href="#">include/peao.hpp</a>	49
<a href="#">include/peca.hpp</a>	50
<a href="#">include/pgn.hpp</a>	51
<a href="#">include/rei.hpp</a>	52
<a href="#">include/tabuleiro.hpp</a>	53
<a href="#">include/torre.hpp</a>	54



## Capítulo 4

# Classes

### 4.1 Referência da Classe Bispo

classe que define o comportamento do bispo

```
#include <bispo.hpp>
```

Subclasse de [Peca](#).

#### Membros Públicos

- [Bispo](#) (int linha, int coluna, [Cor](#) cor)  
*Construtor de um objeto bispo.*
- virtual [~Bispo](#) () override  
*Destrutor de um objeto bispo.*
- virtual std::string [get\\_representacao](#) () override

#### 4.1.1 Descrição detalhada

classe que define o comportamento do bispo

#### 4.1.2 Construtores e Destrutores

##### 4.1.2.1 Bispo()

```
Bispo::Bispo (  
    int  linha,  
    int  coluna,  
    Cor cor )
```

Construtor de um objeto bispo.

**Parâmetros**

<i>linha</i>	Posicao inicial da linha que o bispo ocupa no tabuleiro
<i>coluna</i>	Posicao inicial da coluna que o bispo ocupa no tabuleiro
<i>cor</i>	Cor da peca

**4.1.2.2 ~Bispo()**

```
virtual Bispo::~~Bispo ( ) [override], [virtual]
```

Destrutor de um objeto bispo.

**4.1.3 Funções membros****4.1.3.1 get\_representacao()**

```
virtual std::string Bispo::get_representacao ( ) [override], [virtual]
```

**Retorna**

retorna a representacao do bispo em forma de string

Implementa [Peca](#).

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- [include/bispo.hpp](#)

**4.2 Referência da Classe Cavalo**

classe que define o comportamento do cavalo

```
#include <cavalo.hpp>
```

Subclasse de [Peca](#).

**Membros Públicos**

- [Cavalo](#) (int linha, int coluna, [Cor](#) cor)  
*Construtor de um objeto cavalo.*
- virtual [~Cavalo](#) () override  
*Destrutor de um objeto cavalo.*
- virtual std::string [get\\_representacao](#) () override



### 4.2.1 Descrição detalhada

classe que define o comportamento do cavalo

### 4.2.2 Construtores e Destrutores

#### 4.2.2.1 Cavalo()

```
Cavalo::Cavalo (
    int linha,
    int coluna,
    Cor cor )
```

Construtor de um objeto cavalo.

##### Parâmetros

<i>linha</i>	Posicao inicial da linha que o cavalo ocupa no tabuleiro
<i>coluna</i>	Posicao inicial da coluna que o cavalo ocupa no tabuleiro
<i>cor</i>	Cor da peca

#### 4.2.2.2 ~Cavalo()

```
virtual Cavalo::~Cavalo ( ) [override], [virtual]
```

Destrutor de um objeto cavalo.

### 4.2.3 Funções membros

#### 4.2.3.1 get\_representacao()

```
virtual std::string Cavalo::get_representacao ( ) [override], [virtual]
```

##### Retorna

retorna a representacao do cavalo em forma de string

Implementa [Peca](#).

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- include/[cavalo.hpp](#)

## 4.3 Referência da Classe Dama

classe que define o comportamento da dama

```
#include <dama.hpp>
```

Subclasse de [Peca](#).

### Membros Públicos

- [Dama](#) (int linha, int coluna, [Cor](#) cor)  
*Construtor de um objeto dama.*
- virtual [~Dama](#) () override  
*Destrutor de um objeto dama.*
- virtual std::string [get\\_representacao](#) () override

#### 4.3.1 Descrição detalhada

classe que define o comportamento da dama

#### 4.3.2 Construtores e Destrutores

##### 4.3.2.1 Dama()

```
Dama::Dama (
    int  linha,
    int  coluna,
    Cor  cor )
```

Construtor de um objeto dama.

##### Parâmetros

<i>linha</i>	Posicao inicial da linha que a dama ocupa no tabuleiro
<i>coluna</i>	Posicao inicial da coluna que a dama ocupa no tabuleiro
<i>cor</i>	Cor da peca

##### 4.3.2.2 ~Dama()

```
virtual Dama::~~Dama ( ) [override], [virtual]
```

Destrutor de um objeto dama.

### 4.3.3 Funções membros

#### 4.3.3.1 get\_representacao()

```
virtual std::string Dama::get_representacao ( ) [override], [virtual]
```

##### Retorna

retorna a representacao da dama em forma de string

Implementa [Peca](#).

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- include/[dama.hpp](#)

## 4.4 Referência da Classe ErroAoAbrirException

```
#include <excecoes.hpp>
```

Subclasse de [ErroPgnException](#).

### Membros Públicos

- virtual const char \* [what](#) () const noexcept

### 4.4.1 Funções membros

#### 4.4.1.1 what()

```
virtual const char * ErroAoAbrirException::what ( ) const [inline], [virtual], [noexcept]
```

Reimplementa [ErroPgnException](#).

```
104     {  
105         return "Não foi possivel realizar a abertura do arquivo";  
106     }
```

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- include/[excecoes.hpp](#)

## 4.5 Referência da Classe ErroAoFecharException

```
#include <excecoes.hpp>
```

Subclasse de [ErroPgnException](#).

### Membros Públicos

- virtual const char \* [what](#) () const noexcept

### 4.5.1 Funções membros

#### 4.5.1.1 what()

```
virtual const char * ErroAoFecharException::what ( ) const [inline], [virtual], [noexcept]
```

Reimplementa [ErroPgnException](#).

```
113     {  
114         return "Não foi possível realizar o fechamento do arquivo";  
115     }
```

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- include/[excecoes.hpp](#)

## 4.6 Referência da Classe ErroDeJogoException

```
#include <excecoes.hpp>
```

Subclasse de `std::exception`.

Superclasse de [ForaDoMapaException](#), [MovimentoInvalidoException](#), [NaoESuaVezException](#), [PecaAliadaException](#), [PecaNaFrenteException](#) e [SemPecaParaMovimentarException](#).

### Membros Públicos

- virtual const char \* [what](#) () const noexcept

### 4.6.1 Funções membros

#### 4.6.1.1 what()

```
virtual const char * ErroDeJogoException::what ( ) const [inline], [virtual], [noexcept]
```

Reimplementado por [ForaDoMapaException](#), [SemPecaParaMovimentarException](#), [NaoESuaVezException](#), [PecaNaFrenteException](#), [PecaAliadaException](#) e [MovimentoInvalidoException](#).

```
11     {  
12         return "Movimentação Inválida!";  
13     }
```

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- include/[excecoes.hpp](#)

## 4.7 Referência da Classe ErroDeLogicaException

```
#include <excecoes.hpp>
```

Subclasse de `std::exception`.

### Membros Públicos

- virtual const char \* [what](#) () const noexcept

#### 4.7.1 Funções membros

##### 4.7.1.1 what()

```
virtual const char * ErroDeLogicaException::what ( ) const [inline], [virtual], [noexcept]  
74     {  
75         return "Procedimento inesperado.";  
76     }
```

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- include/[excecoes.hpp](#)

## 4.8 Referência da Classe ErroDeRoqueException

```
#include <excecoes.hpp>
```

Subclasse de `std::exception`.

## Membros Públicos

- virtual const char \* [what](#) () const noexcept

### 4.8.1 Funções membros

#### 4.8.1.1 what()

```
virtual const char * ErroDeRoqueException::what ( ) const [inline], [virtual], [noexcept]
83     {
84         return "Não é possível realizar o roque nesta situação";
85     }
```

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- include/[excecoes.hpp](#)

## 4.9 Referência da Classe ErroPgnException

```
#include <excecoes.hpp>
```

Subclasse de `std::exception`.

Superclasse de [ErroAoAbrirException](#) e [ErroAoFecharException](#).

## Membros Públicos

- virtual const char \* [what](#) () const noexcept

### 4.9.1 Funções membros

#### 4.9.1.1 what()

```
virtual const char * ErroPgnException::what ( ) const [inline], [virtual], [noexcept]
```

Reimplementado por [ErroAoAbrirException](#) e [ErroAoFecharException](#).

```
95     {
96         return "Erro ao manipular os arquivos!";
97     }
```

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- include/[excecoes.hpp](#)

## 4.10 Referência da Classe EscritorDePgn

```
#include <pgn.hpp>
```

Subclasse de [Pgn](#).

### Membros Públicos

- [EscritorDePgn](#) (const std::string &nome\_do\_arquivo)  
*Construtor do objeto Esritor de [Pgn](#).*
- virtual [~EscritorDePgn](#) () override  
*Destrutor do objeto Escritor de [Pgn](#).*
- virtual void [abre\\_arquivo](#) () override  
*Abre o arquivo para escrita.*
- virtual void [fecha\\_arquivo](#) () override  
*fechamento do arquivo utilizado para escrita*

### Outros membros herdados

#### 4.10.1 Construtores e Destrutores

##### 4.10.1.1 EscritorDePgn()

```
EscritorDePgn::EscritorDePgn (  
    const std::string & nome_do_arquivo )
```

Construtor do objeto Esritor de [Pgn](#).

##### Parâmetros

<code>nome_do_arquivo</code>	Recebe o nome do arquivo em que realizará a gravação dos dados da partida
------------------------------	---

##### 4.10.1.2 ~EscritorDePgn()

```
virtual EscritorDePgn::~~EscritorDePgn ( ) [override], [virtual]
```

Destrutor do objeto Escritor de [Pgn](#).

#### 4.10.2 Funções membros

#### 4.10.2.1 abre\_arquivo()

```
virtual void EscriitorDePgn::abre_arquivo ( ) [override], [virtual]
```

Abre o arquivo para escrita.

Implementa [Pgn](#).

#### 4.10.2.2 fecha\_arquivo()

```
virtual void EscriitorDePgn::fecha_arquivo ( ) [override], [virtual]
```

fechamento do arquivo utilizado para escrita

Implementa [Pgn](#).

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- [include/pgn.hpp](#)

### 4.11 Referência da Classe ForaDoMapaException

```
#include <excecoes.hpp>
```

Subclasse de [ErroDeJogoException](#).

#### Membros Públicos

- virtual const char \* [what](#) () const noexcept

#### 4.11.1 Funções membros

##### 4.11.1.1 what()

```
virtual const char * ForaDoMapaException::what ( ) const [inline], [virtual], [noexcept]
```

Reimplementa [ErroDeJogoException](#).

```
20     {  
21         return "A peça não pode ficar fora do tabuleiro.";  
22     }
```

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- [include/excecoes.hpp](#)



## 4.12 Referência da Classe Jogador

```
#include <jogador.hpp>
```

### Membros Públicos

- `Jogador (Tabuleiro *tabuleiro, Cor cor)`  
*Construtor de um objeto jogador.*
- `~Jogador ()`  
*Destrutor de um objeto jogador.*
- `void imprime ()`  
*Imprime o status do Jogador.*
- `void constroi_pecas (Tabuleiro *tabuleiro)`  
*Constroi as pecas do jogador no tabuleiro de acordo com a cor fornecida.*
- `Cor get_cor ()`  
*retorna cor do jogador*
- `void atualiza_pecas (Tabuleiro *tabuleiro)`  
*Atualiza a quantidades de peças do jogador.*
- `void zera_pecas ()`  
*Zera os valores de todas pecas.*
- `map< string, int > get_pecas ()`  
*Retorna um mapa com as peças do jogador.*

### 4.12.1 Construtores e Destrutores

#### 4.12.1.1 Jogador()

```
Jogador::Jogador (
    Tabuleiro * tabuleiro,
    Cor cor )
```

Construtor de um objeto jogador.

#### Parâmetros

<i>tabuleiro</i>	Endereço do tabuleiro em que serão colocadas as pecas do jogador
<i>cor</i>	Define se o jogador jogará com as brancas ou com as pecas pretas

#### 4.12.1.2 ~Jogador()

```
Jogador::~Jogador ( )
```

Destrutor de um objeto jogador.

## 4.12.2 Funções membros

### 4.12.2.1 atualiza\_pecas()

```
void Jogador::atualiza_pecas (
    Tabuleiro * tabuleiro )
```

Atualiza a quantidades de peças do jogador.

### 4.12.2.2 constroi\_pecas()

```
void Jogador::constroi_pecas (
    Tabuleiro * tabuleiro )
```

Constroi as pecas do jogador no tabuleiro de acordo com a cor fornecida.

#### Parâmetros

<i>tabuleiro</i>	endereço do tabuleiro utilizado
------------------	---------------------------------

### 4.12.2.3 get\_cor()

```
Cor Jogador::get_cor ( )
```

retorna cor do jogador

### 4.12.2.4 get\_pecas()

```
map< string, int > Jogador::get_pecas ( )
```

Retorna um mapa com as peças do jogador.

### 4.12.2.5 imprime()

```
void Jogador::imprime ( )
```

Imprime o status do [Jogador](#).

#### 4.12.2.6 zera\_pecas()

```
void Jogador::zera_pecas ( )
```

Zera os valores de todas pecas.

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- [include/jogador.hpp](#)

## 4.13 Referência da Classe Jogo

Define os funcionamentos de um jogo padrao.

```
#include <jogo.hpp>
```

### Membros Públicos

- [Jogo \(\)](#)  
*Construtor do jogo.*
- [~Jogo \(\)](#)  
*Destrutor do jogo.*
- void [imprime \(\)](#)  
*Imprime o estado atual do jogo.*
- bool [ativo \(\)](#)  
*retorna verdadeiro se o jogo estiver atibo*
- void [encerrar \(\)](#)  
*encerra o jogo*
- void [jogada](#) (std::string entrada)  
*realiza os procedimentos do jogador da vez*
- const std::string [processa\\_terminal \(\)](#)  
*Processa os comandos do terminal.*
- const std::string [processa\\_jogada](#) (const std::string entrada)  
*Processa a jogada valida.*

#### 4.13.1 Descrição detalhada

Define os funcionamentos de um jogo padrao.

#### 4.13.2 Construtores e Destrutores

#### 4.13.2.1 Jogo()

```
Jogo::Jogo ( )
```

Construtor do jogo.

#### 4.13.2.2 ~Jogo()

```
Jogo::~~Jogo ( )
```

Destrutor do jogo.

### 4.13.3 Funções membros

#### 4.13.3.1 ativo()

```
bool Jogo::ativo ( )
```

retorna verdadeiro se o jogo estiver atibo

#### 4.13.3.2 encerrar()

```
void Jogo::encerrar ( )
```

encerra o jogo

#### 4.13.3.3 imprime()

```
void Jogo::imprime ( )
```

Imprime o estado atual do jogo.

#### 4.13.3.4 jogada()

```
void Jogo::jogada (
    std::string entrada )
```

realiza os procedimentos do jogador da vez

#### 4.13.3.5 processa\_jogada()

```
const std::string Jogo::processa_jogada (
    const std::string entrada )
```

Processa a jogada valida.

## Parâmetros

<i>entrada</i>	Paramentro que contem o comando
----------------	---------------------------------

## Retorna

const std::string o comando valido

## 4.13.3.6 processa\_terminal()

```
const std::string Jogo::processa_terminal ( )
```

Processa os comandos do terminal.

## Retorna

retorna uma string com a jogada

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- [include/jogo.hpp](#)

## 4.14 Referência da Classe LeitorDePgn

```
#include <pgn.hpp>
```

Subclasse de [Pgn](#).

## Membros Públicos

- [LeitorDePgn](#) (const std::string &nome\_do\_arquivo)  
*Construtor do objeto Leitor De [Pgn](#).*
- virtual [~LeitorDePgn](#) () override  
*Destrutor do objeto Leitor De [Pgn](#).*
- virtual void [abre\\_arquivo](#) () override  
*Abre o arquivo para leitura.*
- virtual void [fecha\\_arquivo](#) () override  
*fechamento do arquivo utilizado para leitura*
- void [imprime\\_comandos](#) ()  
*Função teste para verificar o que esta sendo lido.*

## Outros membros herdados

## 4.14.1 Construtores e Destrutores

## 4.14.1.1 LeitorDePgn()

```
LeitorDePgn::LeitorDePgn (
    const std::string & nome_do_arquivo )
```

Construtor do objeto Leitor De [Pgn](#).

**Parâmetros**

<code>nome_do_arquivo</code>	Recebe o nome do arquivo em que realizará leitura dos dados da partida
------------------------------	--

**4.14.1.2 ~LeitorDePgn()**

```
virtual LeitorDePgn::~~LeitorDePgn ( ) [override], [virtual]
```

Destrutor do objeto Leitor De [Pgn](#).

**4.14.2 Funções membros****4.14.2.1 abre\_arquivo()**

```
virtual void LeitorDePgn::abre_arquivo ( ) [override], [virtual]
```

Abre o arquivo para leitura.

Implementa [Pgn](#).

**4.14.2.2 fecha\_arquivo()**

```
virtual void LeitorDePgn::fecha_arquivo ( ) [override], [virtual]
```

fechamento do arquivo utilizado para leitura

Implementa [Pgn](#).

**4.14.2.3 imprime\_comandos()**

```
void LeitorDePgn::imprime_comandos ( )
```

Função teste para verificar o que esta sendo lido.

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- [include/pgn.hpp](#)

## 4.15 Referência da Classe Movimento

```
#include <movimento.hpp>
```

### Membros Públicos

- **Movimento** (**Tabuleiro** \*tabuleiro, **Jogador** \*jogador, int linha\_inicial, int coluna\_inicial, int linha\_final, int coluna\_final)  
*construtor do movimento*
- void **executar\_movimento** ()  
*executa o movimento, modificando o tabuleiro*
- void **validar\_movimento** ()
- void **checa\_movimento\_peca** ()  
*Checa se a movimentação da peça é válida.*
- void **checa\_movimento\_peao** ()  
*Checa se a movimentação do peão é válida.*
- void **checa\_movimento\_cavalo** ()  
*Checa se a movimentação do cavalo é válida.*
- void **checa\_movimento\_torre** ()  
*Checa se a movimentação da torre é válida.*
- void **checa\_movimento\_bispo** ()  
*Checa se a movimentação do bispo é válida.*
- void **checa\_movimento\_dama** ()  
*Checa se a movimentação da dama é válida.*
- void **checa\_movimento\_rei** ()  
*Checa se a movimentação do rei é válida.*
- void **promove\_peao** (bool mensagem=true)  
*Promove o peão.*
- void **roque** ()  
*Realiza o roque.*

### 4.15.1 Construtores e Destrutores

#### 4.15.1.1 Movimento()

```
Movimento::Movimento (
    Tabuleiro * tabuleiro,
    Jogador * jogador,
    int linha_inicial,
    int coluna_inicial,
    int linha_final,
    int coluna_final )
```

construtor do movimento

### 4.15.2 Funções membros

#### 4.15.2.1 `checa_movimento_bispo()`

```
void Movimento::checa_movimento_bispo ( )
```

Checa se a movimentação do bispo é válida.

#### 4.15.2.2 `checa_movimento_cavalo()`

```
void Movimento::checa_movimento_cavalo ( )
```

Checa se a movimentação do cavalo é válida.

#### 4.15.2.3 `checa_movimento_dama()`

```
void Movimento::checa_movimento_dama ( )
```

Checa se a movimentação da dama é válida.

#### 4.15.2.4 `checa_movimento_peao()`

```
void Movimento::checa_movimento_peao ( )
```

Checa se a movimentação do peão é válida.

#### 4.15.2.5 `checa_movimento_peca()`

```
void Movimento::checa_movimento_peca ( )
```

Checa se a movimentação da peça é válida.



#### 4.15.2.6 `checa_movimento_rei()`

```
void Movimento::checa_movimento_rei ( )
```

Checa se a movimentação do rei é válida.

#### 4.15.2.7 `checa_movimento_torre()`

```
void Movimento::checa_movimento_torre ( )
```

Checa se a movimentação da torre é válida.

#### 4.15.2.8 `executar_movimento()`

```
void Movimento::executar_movimento ( )
```

executa o movimento, modificando o tabuleiro

#### 4.15.2.9 `promove_peao()`

```
void Movimento::promove_peao (
    bool mensagem = true )
```

Promove o peão.

##### Parâmetros

<i>mensagem</i>	Se verdadeiro, a mensagem é exibida
-----------------	-------------------------------------

#### 4.15.2.10 `roque()`

```
void Movimento::roque ( )
```

Realiza o roque.

#### 4.15.2.11 validar\_movimento()

```
void Movimento::validar_movimento ( )
```

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- include/[movimento.hpp](#)

## 4.16 Referência da Classe MovimentoInvalidoException

```
#include <excecoes.hpp>
```

Subclasse de [ErroDeJogoException](#).

### Membros Públicos

- virtual const char \* [what](#) () const noexcept

#### 4.16.1 Funções membros

##### 4.16.1.1 what()

```
virtual const char * MovimentoInvalidoException::what ( ) const [inline], [virtual], [noexcept]
```

Reimplementa [ErroDeJogoException](#).

```
65     {  
66         return "A peça não pode realizar este movimento.";  
67     }
```

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- include/[excecoes.hpp](#)

## 4.17 Referência da Classe NaoESuaVezException

```
#include <excecoes.hpp>
```

Subclasse de [ErroDeJogoException](#).

### Membros Públicos

- virtual const char \* [what](#) () const noexcept

### 4.17.1 Funções membros

#### 4.17.1.1 what()

```
virtual const char * NaoESuaVezException::what ( ) const [inline], [virtual], [noexcept]
```

Reimplementa [ErroDeJogoException](#).

```
38 {
39     return "Não é sua vez de realizar a movimentação.";
40 }
```

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- include/[excecoes.hpp](#)

## 4.18 Referência da Classe Peao

classe que define o comportamento do peao

```
#include <peao.hpp>
```

Subclasse de [Peca](#).

### Membros Públicos

- [Peao](#) (int linha, int coluna, [Cor](#) cor)  
*Construtor de um objeto peao.*
- virtual [~Peao](#) () override  
*Destrutor de um objeto peao.*
- virtual std::string [get\\_representacao](#) () override

### Atributos Públicos

- bool [primeiro\\_movimento](#)  
*guarda a informação se esta é a primeira movimentação do peão*
- bool [en\\_passant](#)

### 4.18.1 Descrição detalhada

classe que define o comportamento do peao

### 4.18.2 Construtores e Destrutores

#### 4.18.2.1 Peao()

```
Peao::Peao (
    int linha,
    int coluna,
    Cor cor )
```

Construtor de um objeto peao.

**Parâmetros**

<i>linha</i>	Posicao inicial na linha que o peao ocupa no tabuleiro
<i>coluna</i>	Posicao inicial na coluna que o peao ocupa no tabuleiro
<i>cor</i>	Cor da peca

**4.18.2.2 ~Peao()**

```
virtual Peao::~~Peao ( ) [override], [virtual]
```

Destrutor de um objeto peao.

**4.18.3 Funções membros****4.18.3.1 get\_representacao()**

```
virtual std::string Peao::get_representacao ( ) [override], [virtual]
```

**Retorna**

retorna a representacao do peao em forma de string

Implementa [Peca](#).

**4.18.4 Atributos****4.18.4.1 en\_passant**

```
bool Peao::en_passant
```

**4.18.4.2 primeiro\_movimento**

```
bool Peao::primeiro_movimento
```

guarda a informação se esta é a primeira movimentação do peão

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- include/[peao.hpp](#)

## 4.19 Referência da Classe Peca

Classe que cuida de todas as funcionalidades gerais de uma peca.

```
#include <peca.hpp>
```

Superclasse de [Bispo](#), [Cavalo](#), [Dama](#), [Peao](#), [Rei](#) e [Torre](#).

### Membros Públicos

- [Peca](#) (int linha, int coluna, [Cor](#) cor)  
*construtor padrão de uma peca*
- virtual [~Peca](#) ()  
*Destrutor da peça.*
- virtual std::string [get\\_representacao](#) ()=0
- [Cor](#) [get\\_cor](#) ()  
*retorna Cor da peça*
- bool [get\\_primeiro\\_movimento](#) ()  
*retorna true se a peça ainda não se movimentou*
- void [set\\_primeiro\\_movimento](#) (bool primeiro\_movimento)  
*seta o primeiro\_movimento*
- void [set\\_en\\_passant](#) (bool en\_passant)  
*seta o movimento de en passant*
- bool [get\\_en\\_passant](#) ()  
*retorna true se o movimento foi de en passant*
- std::pair< int, int > [get\\_casas](#) ()  
*retorna um pair-> first que representa a linha atual da peça e pair-> second que representa a coluna atual da peça*
- void [set\\_casas](#) (int linha, int coluna)  
*Seta as posições da peça.*

### 4.19.1 Descrição detalhada

Classe que cuida de todas as funcionalidades gerais de uma peca.

### 4.19.2 Construtores e Destrutores

#### 4.19.2.1 Peca()

```
Peca::Peca (  
    int linha,  
    int coluna,  
    Cor cor )
```

construtor padrão de uma peca

**Parâmetros**

<i>linha</i>	Posicao inicial da linha que a peca ocupa no tabuleiro
<i>coluna</i>	Posicao inicial da coluna que a peca ocupa no tabuleiro
<i>cor</i>	Cor da peca

**4.19.2.2 ~Peca()**

```
virtual Peca::~~Peca ( ) [inline], [virtual]
```

Destrutor da peça.

```
30 {};
```

**4.19.3 Funções membros****4.19.3.1 get\_casas()**

```
std::pair< int, int > Peca::get_casas ( )
```

retorna um pair->first que representa a linha atual da peça e pair->second que representa a coluna atual da peça

**4.19.3.2 get\_cor()**

```
Cor Peca::get_cor ( )
```

retorna Cor da peça

**4.19.3.3 get\_en\_passant()**

```
bool Peca::get_en_passant ( )
```

retorna true se o movimento foi de en passant

#### 4.19.3.4 get\_primeiro\_movimento()

```
bool Peca::get_primeiro_movimento ( )
```

retorna true se a peça ainda não se movimentou

#### 4.19.3.5 get\_representacao()

```
virtual std::string Peca::get_representacao ( ) [pure virtual]
```

##### Retorna

retorna a representação da peça em forma de string A representação da peça deve possuir 5 caracteres incluindo um espaço antes e depois

Implementado por [Bispo](#), [Cavalo](#), [Dama](#), [Peao](#), [Rei](#) e [Torre](#).

#### 4.19.3.6 set\_casas()

```
void Peca::set_casas (
    int linha,
    int coluna )
```

Seta as posições da peça.

##### Parâmetros

<i>linha</i>	atual
<i>coluna</i>	atual

#### 4.19.3.7 set\_en\_passant()

```
void Peca::set_en_passant (
    bool en_passant )
```

seta o movimento de en passant

#### 4.19.3.8 set\_primeiro\_movimento()

```
void Peca::set_primeiro_movimento (
    bool primeiro_movimento )
```

seta o primeiro\_movimento

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- include/[peca.hpp](#)

## 4.20 Referência da Classe PecaAliadaException

```
#include <excecoes.hpp>
```

Subclasse de [ErroDeJogoException](#).

### Membros Públicos

- virtual const char \* [what](#) () const noexcept

#### 4.20.1 Funções membros

##### 4.20.1.1 what()

```
virtual const char * PecaAliadaException::what ( ) const [inline], [virtual], [noexcept]
```

Reimplementa [ErroDeJogoException](#).

```
56     {
57         return "O jogador não pode tomar as suas próprias pecas.";
58     }
```

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- include/[excecoes.hpp](#)

## 4.21 Referência da Classe PecaNaFrenteException

```
#include <excecoes.hpp>
```

Subclasse de [ErroDeJogoException](#).



## Membros Públicos

- virtual const char \* [what](#) () const noexcept

### 4.21.1 Funções membros

#### 4.21.1.1 what()

```
virtual const char * PecaNaFrenteException::what ( ) const [inline], [virtual], [noexcept]
```

Reimplementa [ErroDeJogoException](#).

```
47     {
48         return "A peça não pode ser movimentada para a casa selecionada porque tem uma peça no
    caminho.";
49     }
```

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- include/[excecoes.hpp](#)

## 4.22 Referência da Classe Pgn

```
#include <pgn.hpp>
```

Superclasse de [EscritorDePgn](#) e [LeitorDePgn](#).

## Membros Públicos

- [Pgn](#) (const std::string &nome\_do\_arquivo)  
*Construtor de um objeto [Pgn](#).*
- virtual [~Pgn](#) ()  
*Destrutor de objetos [Pgn](#).*
- virtual void [abre\\_arquivo](#) ()=0  
*Abre o arquivo.*
- virtual void [fecha\\_arquivo](#) ()=0  
*fecha arquivo*

## Atributos Protegidos

- std::string [nome\\_do\\_arquivo](#)
- std::regex [jogo\\_valido](#)
- std::regex [comandos\\_da\\_rodada](#)
- std::regex [casa\\_final](#)
- std::regex [rodada](#)
- std::string [conteudo](#)

### 4.22.1 Construtores e Destrutores

#### 4.22.1.1 Pgn()

```
Pgn::Pgn (
    const std::string & nome_do_arquivo )
```

Construtor de um objeto [Pgn](#).

**Parâmetros**

<i>nome_do_arquivo</i>	Recebe o nome do arquivo em que realizará a gravação ou leitura dos dados da partida
------------------------	--

**4.22.1.2 ~Pgn()**

```
virtual Pgn::~~Pgn ( ) [virtual]
```

Destrutor de objetos [Pgn](#).

**4.22.2 Funções membros****4.22.2.1 abre\_arquivo()**

```
virtual void Pgn::abre_arquivo ( ) [pure virtual]
```

Abre o arquivo.

Implementado por [LeitorDePgn](#) e [EscritorDePgn](#).

**4.22.2.2 fecha\_arquivo()**

```
virtual void Pgn::fecha_arquivo ( ) [pure virtual]
```

fecha arquivo

Implementado por [LeitorDePgn](#) e [EscritorDePgn](#).

**4.22.3 Atributos****4.22.3.1 casa\_final**

```
std::regex Pgn::casa_final [protected]
```

#### 4.22.3.2 comandos\_da\_rodada

```
std::regex Pgn::comandos_da_rodada [protected]
```

#### 4.22.3.3 conteudo

```
std::string Pgn::conteudo [protected]
```

#### 4.22.3.4 jogo\_valido

```
std::regex Pgn::jogo_valido [protected]
```

#### 4.22.3.5 nome\_do\_arquivo

```
std::string Pgn::nome_do_arquivo [protected]
```

#### 4.22.3.6 rodada

```
std::regex Pgn::rodada [protected]
```

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- [include/pgn.hpp](#)

## 4.23 Referência da Classe Rei

classe que define o comportamento do rei

```
#include <rei.hpp>
```

Subclasse de [Peca](#).

### Membros Públicos

- [Rei](#) (int linha, int coluna, [Cor](#) cor)  
*Construtor de um objeto rei.*
- virtual [~Rei](#) () override  
*Destrutor de um objeto rei.*
- virtual std::string [get\\_representacao](#) () override

### 4.23.1 Descrição detalhada

classe que define o comportamento do rei

### 4.23.2 Construtores e Destrutores

#### 4.23.2.1 Rei()

```
Rei::Rei (
    int linha,
    int coluna,
    Cor cor )
```

Construtor de um objeto rei.

##### Parâmetros

<i>linha</i>	Posicao inicial da linha que o rei ocupa no tabuleiro
<i>coluna</i>	Posicao inicial da coluna que o rei ocupa no tabuleiro
<i>cor</i>	Cor da peca

#### 4.23.2.2 ~Rei()

```
virtual Rei::~~Rei ( ) [override], [virtual]
```

Destrutor de um objeto rei.

### 4.23.3 Funções membros

#### 4.23.3.1 get\_representacao()

```
virtual std::string Rei::get_representacao ( ) [override], [virtual]
```

##### Retorna

retorna a representacao do rei em forma de string

Implementa [Peca](#).

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- [include/rei.hpp](#)

## 4.24 Referência da Classe SemPecaParaMovimentarException

```
#include <excecoes.hpp>
```

Subclasse de [ErroDeJogoException](#).

### Membros Públicos

- virtual const char \* [what](#) () const noexcept

#### 4.24.1 Funções membros

##### 4.24.1.1 what()

```
virtual const char * SemPecaParaMovimentarException::what ( ) const [inline], [virtual], [noexcept]
```

Reimplementa [ErroDeJogoException](#).

```
29 {  
30     return "Não há peça na casa selecionada para realizar a movimentação.";  
31 }
```

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- include/[excecoes.hpp](#)

## 4.25 Referência da Classe Tabuleiro

```
#include <tabuleiro.hpp>
```

### Membros Públicos

- [Tabuleiro](#) (int tamanho)  
*Construtor de um objeto tabuleiro.*
- [~Tabuleiro](#) ()  
*Destrutor de um objeto tabuleiro.*
- void [imprime](#) ()  
*Imprime o estado atual do tabuleiro.*
- void [adiciona\\_peca](#) (const std::string &peca, int linha, int coluna, [Cor](#) cor)  
*Adiciona a peça naquela posicao.*
- void [adiciona\\_peca\\_existente](#) ([Peca](#) \*peca, int linha, int coluna)  
*Adiciona uma peça existente ao tabuleiro.*
- void [limpa\\_casa](#) (int linha, int coluna)  
*Deleta a peça se houver.*
- [Peca](#) \* [get\\_casa](#) (int linha, int coluna)  
*Retorna o ponteiro da casa do tabuleiro.*
- int [get\\_tamanho](#) ()  
*Retorna o tamanho do tabuleiro.*
- void [constroi\\_tabuleiro](#) ()  
*Constroi cada casa do tabuleiro.*
- void [limpa\\_tabuleiro](#) ()  
*Limpa o tabuleiro.*

## 4.25.1 Construtores e Destrutores

### 4.25.1.1 Tabuleiro()

```
Tabuleiro::Tabuleiro (
    int tamanho )
```

Construtor de um objeto tabuleiro.

#### Parâmetros

<i>tamanho</i>	define o tamanho para as n colunas e n linhas do tabuleiro
----------------	--

### 4.25.1.2 ~Tabuleiro()

```
Tabuleiro::~~Tabuleiro ( )
```

Destrutor de um objeto tabuleiro.

## 4.25.2 Funções membros

### 4.25.2.1 adiciona\_peca()

```
void Tabuleiro::adiciona_peca (
    const std::string & peca,
    int linha,
    int coluna,
    Cor cor )
```

Adiciona a peca naquela posicao.

#### Parâmetros

<i>peca</i>	string do nome da peca
<i>linha</i>	posição na linha
<i>coluna</i>	posição na coluna
<i>cor</i>	cor da peca

#### 4.25.2.2 adiciona\_peca\_existente()

```
void Tabuleiro::adiciona_peca_existente (
    Peca * peca,
    int linha,
    int coluna )
```

Adiciona uma peça existente ao tabuleiro.

##### Parâmetros

<i>peca</i>	o endereço da peça que vai ser adicionada
<i>linha</i>	posição na linha
<i>coluna</i>	posição na coluna

#### 4.25.2.3 constroi\_tabuleiro()

```
void Tabuleiro::constroi_tabuleiro ( )
```

Constroi cada casa do tabuleiro.

#### 4.25.2.4 get\_casa()

```
Peca * Tabuleiro::get_casa (
    int linha,
    int coluna )
```

Retorna o ponteiro da casa do tabuleiro.

#### 4.25.2.5 get\_tamanho()

```
int Tabuleiro::get_tamanho ( )
```

Retorna o tamanho do tabuleiro.

#### 4.25.2.6 imprime()

```
void Tabuleiro::imprime ( )
```

Imprime o estado atual do tabuleiro.

#### 4.25.2.7 limpa\_casa()

```
void Tabuleiro::limpa_casa (
    int linha,
    int coluna )
```

Deleta a peça se houver.

## Parâmetros

<i>linha</i>	posição na linha
<i>coluna</i>	posição na coluna

#### 4.25.2.8 limpa\_tabuleiro()

```
void Tabuleiro::limpa_tabuleiro ( )
```

Limpa o tabuleiro.

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- [include/tabuleiro.hpp](#)

## 4.26 Referência da Classe Torre

classe que define o comportamento da torre

```
#include <torre.hpp>
```

Subclasse de [Peca](#).

### Membros Públicos

- [Torre](#) (int linha, int coluna, [Cor](#) cor)  
*Construtor de um objeto torre.*
- virtual [~Torre](#) () override  
*Destrutor de um objeto torre.*
- virtual std::string [get\\_representacao](#) () override

#### 4.26.1 Descrição detalhada

classe que define o comportamento da torre

#### 4.26.2 Construtores e Destrutores

##### 4.26.2.1 Torre()

```
Torre::Torre (
    int linha,
    int coluna,
    Cor cor )
```

Construtor de um objeto torre.



## Parâmetros

<i>linha</i>	Posicao inicial da linha que a torre ocupa no tabuleiro
<i>coluna</i>	Posicao inicial da coluna que a torre ocupa no tabuleiro
<i>cor</i>	Cor da peca

**4.26.2.2 ~Torre()**

```
virtual Torre::~~Torre ( ) [override], [virtual]
```

Destrutor de um objeto torre.

**4.26.3 Funções membros****4.26.3.1 get\_representacao()**

```
virtual std::string Torre::get_representacao ( ) [override], [virtual]
```

**Retorna**

retorna a representacao da torre em forma de string

Implementa [Peca](#).

A documentação para essa classe foi gerada a partir do seguinte arquivo:

- [include/torre.hpp](#)



## Capítulo 5

# Arquivos

### 5.1 Referência do Arquivo include/bispo.hpp

```
#include "peca.hpp"
```

#### Componentes

- class [Bispo](#)  
*classe que define o comportamento do bispo*

### 5.2 bispo.hpp

[Vá para a documentação desse arquivo.](#)

```
1 #ifndef BISPO_HPP
2 #define BISPO_HPP
3
4 #include "peca.hpp"
5
6
10 class Bispo : public Peca
11 {
12     public:
19     Bispo(int linha, int coluna, Cor cor);
20
25     virtual ~Bispo() override;
26
30     virtual std::string get_representacao() override;
31 };
32
33 #endif
```

### 5.3 Referência do Arquivo include/cavalo.hpp

```
#include "peca.hpp"
```

## Componentes

- class [Cavalo](#)

*classe que define o comportamento do cavalo*

## 5.4 cavalo.hpp

[Vá para a documentação desse arquivo.](#)

```
1 #ifndef CAVALO_HPP
2 #define CAVALO_HPP
3
4 #include "peca.hpp"
5
6
10 class Cavalo : public Peca
11 {
12     public:
19         Cavalo(int linha, int coluna, Cor cor);
20
25         virtual ~Cavalo() override;
26
30         virtual std::string get_representacao() override;
31 };
32
33 #endif
```

## 5.5 Referência do Arquivo include/dama.hpp

```
#include "peca.hpp"
```

## Componentes

- class [Dama](#)

*classe que define o comportamento da dama*

## 5.6 dama.hpp

[Vá para a documentação desse arquivo.](#)

```
1 #ifndef DAMA_HPP
2 #define DAMA_HPP
3
4 #include "peca.hpp"
5
6
10 class Dama : public Peca
11 {
12     public:
19         Dama(int linha, int coluna, Cor cor);
20
25         virtual ~Dama() override;
26
30         virtual std::string get_representacao() override;
31 };
32
33 #endif
```

## 5.7 Referência do Arquivo include/excecoes.hpp

```
#include <exception>
```

### Componentes

- class [ErroDeJogoException](#)
- class [ForaDoMapaException](#)
- class [SemPecaParaMovimentarException](#)
- class [NaoESuaVezException](#)
- class [PecaNaFrenteException](#)
- class [PecaAliadaException](#)
- class [MovimentoInvalidoException](#)
- class [ErroDeLogicaException](#)
- class [ErroDeRoqueException](#)
- class [ErroPgnException](#)
- class [ErroAoAbrirException](#)
- class [ErroAoFecharException](#)

## 5.8 excecoes.hpp

Vá para a documentação desse arquivo.

```
1 #ifndef EXCECOES_HPP
2 #define EXCECOES_HPP
3
4 #include <exception>
5
6
7 class ErroDeJogoException : public std::exception
8 {
9     public:
10         virtual const char* what() const noexcept
11         {
12             return "Movimentação Inválida!";
13         }
14 };
15
16 class ForaDoMapaException : public ErroDeJogoException
17 {
18     public:
19         virtual const char* what() const noexcept
20         {
21             return "A peça não pode ficar fora do tabuleiro.";
22         }
23 };
24
25 class SemPecaParaMovimentarException : public ErroDeJogoException
26 {
27     public:
28         virtual const char* what() const noexcept
29         {
30             return "Não há peça na casa selecionada para realizar a movimentação.";
31         }
32 };
33
34 class NaoESuaVezException : public ErroDeJogoException
35 {
36     public:
37         virtual const char* what() const noexcept
38         {
39             return "Não é sua vez de realizar a movimentação.";
40         }
41 };
42
43 class PecaNaFrenteException : public ErroDeJogoException
44 {
45     public:
```

```

46     virtual const char* what() const noexcept
47     {
48         return "A peça não pode ser movimentada para a casa selecionada porque tem uma peça no
        caminho.";
49     }
50 };
51
52 class PecaAliadaException : public ErroDeJogoException
53 {
54     public:
55     virtual const char* what() const noexcept
56     {
57         return "O jogador não pode tomar as suas próprias peças.";
58     }
59 };
60
61 class MovimentoInvalidoException : public ErroDeJogoException
62 {
63     public:
64     virtual const char* what() const noexcept
65     {
66         return "A peça não pode realizar este movimento.";
67     }
68 };
69
70 class ErroDeLogicaException : public std::exception
71 {
72     public:
73     virtual const char* what() const noexcept
74     {
75         return "Procedimento inesperado.";
76     }
77 };
78
79 class ErroDeRoqueException : public std::exception
80 {
81     public:
82     virtual const char* what() const noexcept
83     {
84         return "Não é possível realizar o roque nesta situação";
85     }
86 };
87
88
89
90
91 class ErroPgnException : public std::exception
92 {
93     public:
94     virtual const char* what() const noexcept
95     {
96         return "Erro ao manipular os arquivos!";
97     }
98 };
99
100 class ErroAoAbrirException : public ErroPgnException
101 {
102     public:
103     virtual const char* what() const noexcept
104     {
105         return "Não foi possível realizar a abertura do arquivo";
106     }
107 };
108
109 class ErroAoFecharException : public ErroPgnException
110 {
111     public:
112     virtual const char* what() const noexcept
113     {
114         return "Não foi possível realizar o fechamento do arquivo";
115     }
116 };
117
118 #endif

```

## 5.9 Referência do Arquivo include/jogador.hpp

```

#include "tabuleiro.hpp"
#include "peca.hpp"
#include <string>

```

```
#include <map>
```

## Componentes

- class [Jogador](#)

## 5.10 jogador.hpp

[Vá para a documentação desse arquivo.](#)

```
1 #ifndef JOGADOR_HPP
2 #define JOGADOR_HPP
3
4 #include "tabuleiro.hpp"
5 #include "peca.hpp"
6
7 #include <string>
8 #include <map>
9 using namespace std;
10
11 class Jogador
12 {
13     public:
14         Jogador(Tabuleiro* tabuleiro, Cor cor);
15
16         ~Jogador();
17
18         void imprime();
19
20         void constroi_pecas(Tabuleiro* tabuleiro);
21
22         Cor get_cor();
23
24         void atualiza_pecas(Tabuleiro* tabuleiro);
25
26         void zera_pecas();
27
28         map<string, int> get_pecas();
29
30     private:
31         std::map <std::string, int> pecas;
32
33         Cor cor;
34 };
35
36 #endif
```

## 5.11 Referência do Arquivo include/jogo.hpp

```
#include "jogador.hpp"
#include "movimento.hpp"
#include "tabuleiro.hpp"
#include <iostream>
#include <regex>
#include <sstream>
#include <algorithm>
```

## Componentes

- class [Jogo](#)

*Define os funcionamentos de um jogo padrao.*

## 5.12 jogo.hpp

Vá para a documentação desse arquivo.

```
1 #ifndef JOGO_HPP
2 #define JOGO_HPP
3
4 #include "jogador.hpp"
5 #include "movimento.hpp"
6 #include "tabuleiro.hpp"
7
8 #include <iostream>
9 #include <regex>
10 #include <sstream>
11 #include <algorithm>
12
13 class Jogo
14 {
15     public:
16         Jogo();
17         ~Jogo();
18
19         void imprime();
20
21         bool ativo();
22
23         void encerrar();
24
25         void jogada(std::string entrada);
26
27         const std::string processa_terminal();
28
29         const std::string processa_jogada(const std::string entrada);
30
31     private:
32         Jogador* p1 = nullptr;
33         Jogador* p2 = nullptr;
34
35         Jogador* da_vez;
36
37         Tabuleiro* tabuleiro = nullptr;
38
39         bool jogo_ativo;
40
41         std::regex comando_valido;
42 };
43 #endif
```

## 5.13 Referência do Arquivo include/movimento.hpp

```
#include "tabuleiro.hpp"
#include "jogador.hpp"
#include <iostream>
```

### Componentes

- class [Movimento](#)

## 5.14 movimento.hpp

Vá para a documentação desse arquivo.

```
1 #ifndef MOVIMENTO_HPP
2 #define MOVIMENTO_HPP
3
4 #include "tabuleiro.hpp"
```



```

5 #include "jogador.hpp"
6
7 #include <iostream>
8
9
10 class Movimento
11 {
12     public:
13         Movimento(
14             Tabuleiro *tabuleiro,
15             Jogador *jogador,
16             int linha_inicial,
17             int coluna_inicial,
18             int linha_final,
19             int coluna_final);
20
21     void executar_movimento();
22
23     void validar_movimento();
24     void checa_movimento_peca();
25
26     void checa_movimento_peao();
27
28     void checa_movimento_cavalo();
29
30     void checa_movimento_torre();
31
32     void checa_movimento_bispo();
33
34     void checa_movimento_dama();
35
36     void checa_movimento_rei();
37
38     void promove_peao(bool mensagem = true);
39
40     void roque();
41
42     private:
43         Tabuleiro *tabuleiro;
44         Jogador *jogador;
45         Peca *peca;
46         int linha_inicial;
47         int coluna_inicial;
48         int linha_final;
49         int coluna_final;
50 };
51 #endif

```

## 5.15 Referência do Arquivo include/peao.hpp

```
#include "peca.hpp"
```

### Componentes

- class **Peao**  
*classe que define o comportamento do peao*

## 5.16 peao.hpp

Vá para a documentação desse arquivo.

```

1 #ifndef PEAO_HPP
2 #define PEAO_HPP
3
4 #include "peca.hpp"
5
6
10 class Peao : public Peca
11 {

```

```

12     public:
19         Peao(int linha, int coluna, Cor cor);
20
24         virtual ~Peao() override;
25
29         virtual std::string get_representacao() override;
30
34         bool primeiro_movimento;
35
36         bool en_passant;
37     };
38
39 #endif

```

## 5.17 Referência do Arquivo include/peca.hpp

```

#include <string>
#include "excecoes.hpp"

```

### Componentes

- class [Peca](#)

*Classe que cuida de todas as funcionalidades gerais de uma peca.*

### Enumerações

- enum [Cor](#) { [BRANCAS](#) , [PRETAS](#) }

#### 5.17.1 Enumerações

##### 5.17.1.1 Cor

```
enum Cor
```

##### Enumeradores

BRANCAS	
PRETAS	

```

9 {
10     BRANCAS, PRETAS
11 };

```

## 5.18 peca.hpp

[Vá para a documentação desse arquivo.](#)

```
1 #ifndef PECA_HPP
```

```

2 #define PECA_HPP
3
4 #include <string>
5 #include "excecoes.hpp"
6
7
8 enum Cor
9 {
10     BRANCAS, PRETAS
11 };
12
13 class Peca
14 {
15     public:
16         Peca(int linha, int coluna, Cor cor);
17
18         virtual ~Peca() {};
19
20         virtual std::string get_representacao() = 0;
21
22         Cor get_cor();
23
24         bool get_primeiro_movimento();
25
26         void set_primeiro_movimento(bool primeiro_movimento);
27
28         void set_en_passant(bool en_passant);
29
30         bool get_en_passant();
31
32         std::pair<int, int> get_casas();
33
34         void set_casas(int linha, int coluna);
35
36     private:
37         unsigned int linha, coluna;
38
39         Cor cor;
40
41         bool primeiro_movimento;
42
43         bool en_passant;
44 };
45 #endif

```

## 5.19 Referência do Arquivo include/pgn.hpp

```

#include "excecoes.hpp"
#include <string>
#include <chrono>
#include <regex>
#include <fstream>
#include <iostream>

```

### Componentes

- class [Pgn](#)
- class [LeitorDePgn](#)
- class [EscritorDePgn](#)

## 5.20 pgn.hpp

Vá para a documentação desse arquivo.

```
1 #ifndef PGN_HPP
```

```

2 #define PGN_HPP
3
4 #include "excecoes.hpp"
5
6 #include <string>
7 #include <chrono>
8 #include <regex>
9 #include <fstream>
10 #include <iostream>
11
12 class Pgn
13 {
14     public:
15         Pgn(const std::string &nome_do_arquivo);
16         virtual ~Pgn();
17         virtual void abre_arquivo() = 0;
18         virtual void fecha_arquivo() = 0;
19
20     protected:
21         std::string nome_do_arquivo;
22         std::regex jogo_valido;
23         std::regex comandos_da_rodada;
24         std::regex casa_final;
25         std::regex rodada;
26         std::string conteudo;
27 };
28
29 class LeitorDePgn: public Pgn
30 {
31     public:
32         LeitorDePgn(const std::string &nome_do_arquivo);
33         virtual ~LeitorDePgn() override;
34         virtual void abre_arquivo() override;
35         virtual void fecha_arquivo() override;
36         void imprime_comandos();
37
38     private:
39         std::ifstream arquivo;
40 };
41
42 class EscritorDePgn: public Pgn
43 {
44     public:
45         EscritorDePgn(const std::string &nome_do_arquivo);
46         virtual ~EscritorDePgn() override;
47         virtual void abre_arquivo() override;
48         virtual void fecha_arquivo() override;
49
50     private:
51         std::ofstream arquivo;
52 };
53 #endif

```

## 5.21 Referência do Arquivo include/rei.hpp

```
#include "peca.hpp"
```

### Componentes

- class [Rei](#)  
*classe que define o comportamento do rei*

## 5.22 rei.hpp

Vá para a documentação desse arquivo.

```
1 #ifndef REI_HPP
2 #define REI_HPP
3
4 #include "peca.hpp"
5
6
10 class Rei : public Peca
11 {
12     public:
19     Rei(int linha, int coluna, Cor cor);
20
25     virtual ~Rei() override;
26
30     virtual std::string get_representacao() override;
31 };
32
33 #endif
```

## 5.23 Referência do Arquivo include/tabuleiro.hpp

```
#include "peca.hpp"
#include "peao.hpp"
#include "bispo.hpp"
#include "cavalo.hpp"
#include "torre.hpp"
#include "dama.hpp"
#include "rei.hpp"
#include <vector>
#include <iostream>
```

### Componentes

- class [Tabuleiro](#)

### Definições e Macros

- #define [TAMANHO\\_DO\\_TABULEIRO](#) 8

#### 5.23.1 Definições e macros

##### 5.23.1.1 TAMANHO\_DO\_TABULEIRO

```
#define TAMANHO_DO_TABULEIRO 8
```

## 5.24 tabuleiro.hpp

Vá para a documentação desse arquivo.

```
1 #ifndef TABULEIRO_HPP
2 #define TABULEIRO_HPP
3
4 #include "peca.hpp"
5 #include "peao.hpp"
6 #include "bispo.hpp"
7 #include "cavalo.hpp"
8 #include "torre.hpp"
9 #include "dama.hpp"
10 #include "rei.hpp"
11
12 #include <vector>
13 #include <iostream>
14
15 #define TAMANHO_DO_TABULEIRO 8
16
17
18 class Tabuleiro
19 {
20     public:
21         Tabuleiro(int tamanho);
22         ~Tabuleiro();
23
24         void imprime();
25
26         void adiciona_peca(const std::string &peca, int linha, int coluna, Cor cor);
27
28         void adiciona_peca_existente(Peca *peca, int linha, int coluna);
29
30         void limpa_casa( int linha, int coluna);
31
32         Peca* get_casa(int linha, int coluna);
33
34         int get_tamanho();
35
36         void constroi_tabuleiro();
37
38         void limpa_tabuleiro();
39     private:
40         std::vector <std::vector <Peca *>> casas;
41
42         const int tamanho;
43
44         std::vector<std::string> L {"A", "B", "C", "D", "E", "F", "G", "H"};
45 };
46
47 #endif
```

## 5.25 Referência do Arquivo include/torre.hpp

```
#include "peca.hpp"
```

### Componentes

- class [Torre](#)  
*classe que define o comportamento da torre*

## 5.26 torre.hpp

Vá para a documentação desse arquivo.

```
1 #ifndef TORRE_HPP
2 #define TORRE_HPP
```

```
3
4 #include "peca.hpp"
5
6
10 class Torre : public Peca
11 {
12     public:
19         Torre(int linha, int coluna, Cor cor);
20
25         virtual ~Torre() override;
26
30         virtual std::string get_representacao() override;
31 };
32
33 #endif
```





# Índice Remissivo

- ~Bispo
  - Bispo, [8](#)
- ~Cavalo
  - Cavalo, [9](#)
- ~Dama
  - Dama, [10](#)
- ~EscritorDePgn
  - EscritorDePgn, [15](#)
- ~Jogador
  - Jogador, [17](#)
- ~Jogo
  - Jogo, [20](#)
- ~LeitorDePgn
  - LeitorDePgn, [22](#)
- ~Peao
  - Peao, [28](#)
- ~Peca
  - Peca, [30](#)
- ~Pgn
  - Pgn, [34](#)
- ~Rei
  - Rei, [36](#)
- ~Tabuleiro
  - Tabuleiro, [38](#)
- ~Torre
  - Torre, [41](#)
- abre\_arquivo
  - EscritorDePgn, [15](#)
  - LeitorDePgn, [22](#)
  - Pgn, [34](#)
- adiciona\_pecas
  - Tabuleiro, [38](#)
- adiciona\_pecas\_existente
  - Tabuleiro, [38](#)
- ativo
  - Jogo, [20](#)
- atualiza\_pecas
  - Jogador, [18](#)
- Bispo, [7](#)
  - ~Bispo, [8](#)
  - Bispo, [7](#)
  - get\_representacao, [8](#)
- BRANCAS
  - peca.hpp, [50](#)
- casa\_final
  - Pgn, [34](#)
- Cavalo, [8](#)
- ~Cavalo, [9](#)
  - Cavalo, [9](#)
  - get\_representacao, [9](#)
- checa\_movimento\_bispo
  - Movimento, [24](#)
- checa\_movimento\_cavalo
  - Movimento, [24](#)
- checa\_movimento\_dama
  - Movimento, [24](#)
- checa\_movimento\_peao
  - Movimento, [24](#)
- checa\_movimento\_pecas
  - Movimento, [24](#)
- checa\_movimento\_rei
  - Movimento, [24](#)
- checa\_movimento\_torre
  - Movimento, [25](#)
- comandos\_da\_rodada
  - Pgn, [34](#)
- constroi\_pecas
  - Jogador, [18](#)
- constroi\_tabuleiro
  - Tabuleiro, [39](#)
- conteudo
  - Pgn, [35](#)
- Cor
  - peca.hpp, [50](#)
- Dama, [10](#)
  - ~Dama, [10](#)
  - Dama, [10](#)
  - get\_representacao, [11](#)
- en\_passant
  - Peao, [28](#)
- encerrar
  - Jogo, [20](#)
- ErroAoAbrirException, [11](#)
  - what, [11](#)
- ErroAoFecharException, [12](#)
  - what, [12](#)
- ErroDeJogoException, [12](#)
  - what, [12](#)
- ErroDeLogicaException, [13](#)
  - what, [13](#)
- ErroDeRoqueException, [13](#)
  - what, [14](#)
- ErroPgnException, [14](#)
  - what, [14](#)
- EscritorDePgn, [15](#)

- ~EscritorDePgn, 15
- abre\_arquivo, 15
- EscritorDePgn, 15
- fecha\_arquivo, 16
- executar\_movimento
  - Movimento, 25
- fecha\_arquivo
  - EscritorDePgn, 16
  - LeitorDePgn, 22
  - Pgn, 34
- ForaDoMapaException, 16
  - what, 16
- get\_casa
  - Tabuleiro, 39
- get\_casas
  - Peca, 30
- get\_cor
  - Jogador, 18
  - Peca, 30
- get\_en\_passant
  - Peca, 30
- get\_pecas
  - Jogador, 18
- get\_primeiro\_movimento
  - Peca, 30
- get\_representacao
  - Bispo, 8
  - Cavalo, 9
  - Dama, 11
  - Peao, 28
  - Peca, 31
  - Rei, 36
  - Torre, 41
- get\_tamanho
  - Tabuleiro, 39
- imprime
  - Jogador, 18
  - Jogo, 20
  - Tabuleiro, 39
- imprime\_comandos
  - LeitorDePgn, 22
- include/bispo.hpp, 43
- include/cavalo.hpp, 43, 44
- include/dama.hpp, 44
- include/excecoes.hpp, 45
- include/jogador.hpp, 46, 47
- include/jogo.hpp, 47, 48
- include/movimento.hpp, 48
- include/peao.hpp, 49
- include/peca.hpp, 50
- include/pgn.hpp, 51
- include/rei.hpp, 52
- include/tabuleiro.hpp, 53, 54
- include/torre.hpp, 54
- jogada
  - Jogo, 20
- Jogador, 17
  - ~Jogador, 17
  - atualiza\_pecas, 18
  - constroi\_pecas, 18
  - get\_cor, 18
  - get\_pecas, 18
  - imprime, 18
  - Jogador, 17
  - zera\_pecas, 18
- Jogo, 19
  - ~Jogo, 20
  - ativo, 20
  - encerrar, 20
  - imprime, 20
  - jogada, 20
  - Jogo, 19
  - processa\_jogada, 20
  - processa\_terminal, 21
- jogo\_valido
  - Pgn, 35
- LeitorDePgn, 21
  - ~LeitorDePgn, 22
  - abre\_arquivo, 22
  - fecha\_arquivo, 22
  - imprime\_comandos, 22
  - LeitorDePgn, 21
- limpa\_casa
  - Tabuleiro, 39
- limpa\_tabuleiro
  - Tabuleiro, 40
- Movimento, 23
  - checa\_movimento\_bispo, 24
  - checa\_movimento\_cavalo, 24
  - checa\_movimento\_dama, 24
  - checa\_movimento\_peao, 24
  - checa\_movimento\_peca, 24
  - checa\_movimento\_rei, 24
  - checa\_movimento\_torre, 25
  - executar\_movimento, 25
  - Movimento, 23
  - promove\_peao, 25
  - roque, 25
  - validar\_movimento, 25
- MovimentoInvalidoException, 26
  - what, 26
- NaoESuaVezException, 26
  - what, 27
- nome\_do\_arquivo
  - Pgn, 35
- Peao, 27
  - ~Peao, 28
  - en\_passant, 28
  - get\_representacao, 28
  - Peao, 27

- primeiro\_movimento, 28
- Peca, 29
  - ~Peca, 30
  - get\_casas, 30
  - get\_cor, 30
  - get\_en\_passant, 30
  - get\_primeiro\_movimento, 30
  - get\_representacao, 31
  - Peca, 29
  - set\_casas, 31
  - set\_en\_passant, 31
  - set\_primeiro\_movimento, 31
- peca.hpp
  - BRANCAS, 50
  - Cor, 50
  - PRETAS, 50
- PecaAliadaException, 32
  - what, 32
- PecaNaFrenteException, 32
  - what, 33
- Pgn, 33
  - ~Pgn, 34
  - abre\_arquivo, 34
  - casa\_final, 34
  - comandos\_da\_rodada, 34
  - conteudo, 35
  - fecha\_arquivo, 34
  - jogo\_valido, 35
  - nome\_do\_arquivo, 35
  - Pgn, 33
  - rodada, 35
- PRETAS
  - peca.hpp, 50
- primeiro\_movimento
  - Peao, 28
- processa\_jogada
  - Jogo, 20
- processa\_terminal
  - Jogo, 21
- promove\_peao
  - Movimento, 25
- Rei, 35
  - ~Rei, 36
  - get\_representacao, 36
  - Rei, 36
- rodada
  - Pgn, 35
- roque
  - Movimento, 25
- SemPecaParaMovimentarException, 37
  - what, 37
- set\_casas
  - Peca, 31
- set\_en\_passant
  - Peca, 31
- set\_primeiro\_movimento
  - Peca, 31
- Tabuleiro, 37
  - ~Tabuleiro, 38
  - adiciona\_peca, 38
  - adiciona\_peca\_existente, 38
  - constroi\_tabuleiro, 39
  - get\_casa, 39
  - get\_tamanho, 39
  - imprime, 39
  - limpa\_casa, 39
  - limpa\_tabuleiro, 40
  - Tabuleiro, 38
- tabuleiro.hpp
  - TAMANHO\_DO\_TABULEIRO, 53
- TAMANHO\_DO\_TABULEIRO
  - tabuleiro.hpp, 53
- Torre, 40
  - ~Torre, 41
  - get\_representacao, 41
  - Torre, 40
- validar\_movimento
  - Movimento, 25
- what
  - ErroAoAbrirException, 11
  - ErroAoFecharException, 12
  - ErroDeJogoException, 12
  - ErroDeLogicaException, 13
  - ErroDeRoqueException, 14
  - ErroPgnException, 14
  - ForaDoMapaException, 16
  - MovimentoInvalidoException, 26
  - NaoESuaVezException, 27
  - PecaAliadaException, 32
  - PecaNaFrenteException, 33
  - SemPecaParaMovimentarException, 37
- zera\_pecas
  - Jogador, 18