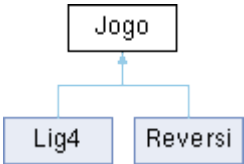


# Referência da Classe Jogo abstract

Classe responsável pela gestão do jogo, incluindo o tabuleiro e os jogadores. [Mais...](#)

```
#include <Jogo.h>
```

Diagrama de hierarquia da classe Jogo:



## Membros Públicos

	<b>Jogo ()</b>	Construtor padrão da classe <b>Jogo</b> .
virtual void	<b>inicializarTabuleiro ()</b>	Inicializa o tabuleiro do jogo.
virtual bool	<b>validarJogada</b> (int _linha, int _coluna, <b>Jogador</b> *jogador, char peca)=0	Valida a jogada de um jogador.
virtual int	<b>verificarCondicaoVitoria</b> (char peca)	Verifica a condição de vitória.
void	<b>imprimirTabuleiro ()</b>	Imprime o estado atual do jogo.
virtual void	<b>realizarJogada</b> ( <b>Jogador</b> *jogador, char peca)=0	Realiza a jogada de um jogador em uma coluna (usado por <b>Lig4</b> ).
<b>Jogador</b> *	<b>getJogadorAtual ()</b>	Realiza uma jogada no tabuleiro (usado por <b>Reversi</b> ).
	<b>~Jogo ()</b>	Destrutor da classe <b>Jogo</b> .

## Descrição detalhada

Classe responsável pela gestão do jogo, incluindo o tabuleiro e os jogadores.

A classe **Jogo** gerencia as operações principais como inicializar o tabuleiro, validar jogadas, verificar condições de vitória e realizar jogadas.

## Construtores e Destrutores

◆ Jogo()

Jogo::Jogo ( )

Construtor padrão da classe **Jogo**.

Inicializa o ponteiro do tabuleiro e o ponteiro do jogador atual.

◆ ~Jogo()

Jogo::~~Jogo ( )

Destrutor da classe **Jogo**.

Libera a memória alocada para o tabuleiro e o jogador atual.

## Documentação das funções

◆ getJogadorAtual()

**Jogador** \* Jogo::getJogadorAtual ( )

Realiza uma jogada no tabuleiro (usado por **Reversi**).

Este método deve ser implementado pelas classes derivadas que usam coordenadas x e y, e a referência ao jogador para realizar a jogada.

### Parâmetros

**x** Posição X no tabuleiro.

**y** Posição Y no tabuleiro.

**jogador** Referência ao jogador que está fazendo a jogada.

Obtém o jogador atual.

### Retorna

Jogador\* Ponteiro para o jogador atual.

◆ imprimirTabuleiro()

void Jogo::imprimirTabuleiro ( )

Imprime o estado atual do jogo.

Usa o método imprimir() da classe **Tabuleiro** para mostrar o tabuleiro e as jogadas.

◆ inicializarTabuleiro()

void Jogo::inicializarTabuleiro ( )

virtual

Inicializa o tabuleiro do jogo.

Método virtual que deve ser implementado para definir o estado inicial do tabuleiro.

Reimplementado por **Lig4** e **Reversi**.

◆ realizarJogada()

virtual void Jogo::realizarJogada ( **Jogador** \* jogador ,  
char peca )

pure virtual

Realiza a jogada de um jogador em uma coluna (usado por **Lig4**).

Este método deve ser implementado pelas classes derivadas que usam apenas a coluna para realizar a jogada.

#### Parâmetros

**coluna** A coluna onde o jogador deseja realizar a jogada.

Implementado por **Lig4** e **Reversi**.

◆ validarJogada()

```
virtual bool Jogo::validarJogada ( int      _linha,  
                                   int      _coluna,  
                                   Jogador * jogador,  
                                   char      peca )
```

pure virtual

Valida a jogada de um jogador.

#### Parâmetros

**\_linha** Linha onde o jogador deseja jogar.

**\_coluna** Coluna onde o jogador deseja jogar.

**jogador** Objeto que representa o jogador que está realizando a jogada.

#### Retorna

true se a jogada for válida, false caso contrário.

Verifica se a jogada feita é válida com base nas regras do jogo.

Implementado por **Lig4** e **Reversi**.

### ◆ verificarCondicaoVitoria()

```
int Jogo::verificarCondicaoVitoria ( char peca )
```

virtual

Verifica a condição de vitória.

Analisa o tabuleiro para determinar o estado atual do jogo.

#### Retorna

int Retorna 1 se algum jogador venceu, -1 em caso de empate, ou 0 se o jogo deve continuar.

Reimplementado por **Lig4** e **Reversi**.

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- /Users/iangodoi/Desktop/TP-jogosTabuleiro-cpp-desenvolvimento 5/include/**Jogo.hpp**
- /Users/iangodoi/Desktop/TP-jogosTabuleiro-cpp-desenvolvimento 5/src/**Jogo.cpp**