

Referência da Classe SistemaDeJogo

Classe responsável pela gestão e execução das partidas. [Mais...](#)

```
#include <SistemaDeJogo.hpp>
```

Membros Públicos

SistemaDeJogo ()

Construtor da classe **SistemaDeJogo**.

void **executarPartida** ()

Método para executar a partida.

void **escolherJogo** ()

Método para escolher o jogo.

~SistemaDeJogo ()

Destrutor da classe **SistemaDeJogo**.

Atributos Públicos

Jogador * **jogador1**

Jogador * **jogador2**

Descrição detalhada

Classe responsável pela gestão e execução das partidas.

A classe **SistemaDeJogo** gerencia a interação entre o jogo e o cadastro de jogadores, permitindo a escolha de jogadores e o controle do fluxo da partida.

Construtores e Destrutores

◆ SistemaDeJogo()

SistemaDeJogo::SistemaDeJogo ()

Construtor da classe **SistemaDeJogo**.

Inicializa o sistema de jogo e carrega os jogadores.

```
4      {
5      cadastroDeJogadores.criarArquivoJogadores();
6      cadastroDeJogadores.carregarJogadoresDeArquivo("jogadores.txt");
7      jogador1 = nullptr;
8      jogador2 = nullptr;
9  }
```

◆ ~SistemaDeJogo()

SistemaDeJogo::~~SistemaDeJogo ()

Destrutor da classe **SistemaDeJogo**.

Libera os recursos alocados.

```
109     {
110     // Os recursos são automaticamente liberados por causa dos smart
111     pointers
112 }
```

Documentação das funções

◆ escolherJogo()

```
void SistemaDeJogo::escolherJogo ( )
```

Método para escolher o jogo.

Permite ao usuário selecionar o tipo de jogo que deseja jogar.

```
11         {
12             int escolha;
13             std::cout << "Escolha o jogo: " << std::endl;
14             std::cout << "1. Reversi" << std::endl;
15             std::cout << "2. Lig4" << std::endl;
16             std::cin >> escolha;
17
18             switch (escolha) {
19                 case 1:
20                     jogo = std::make_unique<Reversi>(jogador1, jogador2); //
Inicializa o Reversi
21                     break;
22                 case 2:
23                     jogo = std::make_unique<Lig4>(); // Inicializa o Lig4
24                     break;
25                 default:
26                     std::cout << "Opção inválida!" << std::endl;
27                     break;
28             }
29         }
```

♦ executarPartida()

void SistemaDeJogo::executarPartida ()

Método para executar a partida.

Permite ao usuário escolher os jogadores, inicializa o tabuleiro e controla o fluxo do jogo.

```

31         {
32             std::string nomeJogador1, nomeJogador2;
33
34             std::cout << "Digite o nome do primeiro jogador: ";
35             std::cin >> nomeJogador1;
36             Jogador* jogador1 = cadastroDeJogadores.getJogadorPorNome(nomeJogador1);
37
38             if (jogador1 == nullptr) {
39                 char opcao;
40                 std::cout << "Jogador " << nomeJogador1 << " não encontrado. Deseja
cadastrá-lo? (s/n): ";
41                 std::cin >> opcao;
42                 if (opcao == 's' || opcao == 'S') {
43                     cadastroDeJogadores.adicionarJogadorNoArquivo(nomeJogador1);
44                     jogador1 = cadastroDeJogadores.getJogadorPorNome(nomeJogador1);
45                 } else {
46                     return;
47                 }
48             }
49
50             std::cout << "Digite o nome do segundo jogador: ";
51             std::cin >> nomeJogador2;
52             Jogador* jogador2 = cadastroDeJogadores.getJogadorPorNome(nomeJogador2);
53
54             if (jogador2 == nullptr) {
55                 char opcao;
56                 std::cout << "Jogador " << nomeJogador2 << " não encontrado. Deseja
cadastrá-lo? (s/n): ";
57                 std::cin >> opcao;
58                 if (opcao == 's' || opcao == 'S') {
59                     cadastroDeJogadores.adicionarJogadorNoArquivo(nomeJogador2);
60                     jogador2 = cadastroDeJogadores.getJogadorPorNome(nomeJogador2);
61                 } else {
62                     return;
63                 }
64             }
65
66             escolherJogo(); //main ?
67
68             if (jogo) {
69                 jogo->inicializarTabuleiro();
70                 bool turnoJogador1 = true; // Inicia com o jogador 1
71                 int condicaoVitoria = 0;
72
73                 while (condicaoVitoria == 0) {
74                     jogo->imprimirTabuleiro();
75
76                     if (turnoJogador1) {
77                         std::cout << "Vez de " << nomeJogador1 << std::endl;
78                         jogo->realizarJogada();
79                     } else {
80                         std::cout << "Vez de " << nomeJogador2 << std::endl;
81                         jogo->realizarJogada();
82                     }
83
84                     turnoJogador1 = !turnoJogador1; // Alterna o turno entre os
jogadores
85
86                     condicaoVitoria = jogo->verificarCondicaoVitoria();

```

```
86         }
87
88         jogo->imprimirTabuleiro();
89         std::cout << "Fim de jogo!" << std::endl;
90
91         if (condicaoVitoria == -1) {
92             std::cout << "O jogo terminou em empate!" << std::endl;
93             jogador1->registrarEmpate();
94             jogador2->registrarEmpate();
95         } else if (condicaoVitoria == 1) {
96             if (turnoJogador1) {
97                 std::cout << "O jogador " << nomeJogador2 << " é o campeão!"
98                 << std::endl;
99                 jogador1->registrarDerrota();
100                jogador2->registrarVitoria();
101            } else {
102                std::cout << "O jogador " << nomeJogador1 << " é o campeão!"
103                << std::endl;
104                jogador1->registrarVitoria();
105                jogador2->registrarDerrota();
106            }
107        }
```

Atributos

♦ jogador1

Jogador* SistemaDeJogo::jogador1

♦ jogador2

Jogador* SistemaDeJogo::jogador2

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

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

Gerado por **doxygen** 1.12.0