

Projeto Final

Gerado por Doxygen 1.12.0

1 UI	1
1.1 Objetivo	1
1.2 Solução	1
1.2.1 O workflow REACT	1
1.2.2 Telas	1
1.2.2.1 Seleção de jogadores	1
1.3 Como rodar?	1
2 Índice Hierárquico	3
2.1 Hierarquia de Classes	3
3 Índice dos Componentes	5
3.1 Lista de Classes	5
4 Índice dos Arquivos	7
4.1 Lista de Arquivos	7
5 Classes	9
5.1 Referência da Classe BatalhaNaval	9
5.1.1 Construtores e Destrutores	10
5.1.1.1 BatalhaNaval()	10
5.1.2 Documentação das funções	10
5.1.2.1 anunciarInicioPartida()	10
5.1.2.2 checarEmpate()	10
5.1.2.3 checarPosicaoValida()	10
5.1.2.4 checarVencedor() [1/2]	11
5.1.2.5 checarVencedor() [2/2]	11
5.1.2.6 inserirBarcos()	11
5.1.2.7 Jogar()	11
5.1.2.8 lerBarcos()	11
5.1.2.9 lerJogada()	11
5.1.2.10 marcarTabuleiro()	12
5.1.2.11 quantidadeBarcosDisponiveis()	12
5.1.2.12 verificarEntrada()	12
5.1.2.13 verificarTamanhoDoBarco()	12
5.2 Referência da Classe CentralDeJogos	12
5.2.1 Construtores e Destrutores	13
5.2.1.1 CentralDeJogos()	13
5.2.1.2 ~CentralDeJogos()	13
5.2.2 Documentação das funções	13
5.2.2.1 buscarJogador()	13
5.2.2.2 cadastrarJogador()	13
5.2.2.3 executarPartida()	13
5.2.2.4 listarJogadores()	13

5.2.2.5 ordenarJogadores()	14
5.2.2.6 removerJogador()	14
5.2.2.7 validarEntrada()	14
5.3 Referência da Classe Estatisticas	14
5.3.1 Construtores e Destrutores	14
5.3.1.1 Estatisticas() [1/2]	14
5.3.1.2 Estatisticas() [2/2]	14
5.3.2 Documentação das funções	15
5.3.2.1 getDerrotas()	15
5.3.2.2 getEmpates()	15
5.3.2.3 getHistorico()	15
5.3.2.4 getVitorias()	15
5.3.2.5 mostrarEstatisticas()	15
5.3.2.6 registrarDerrota()	15
5.3.2.7 registrarEmpate()	15
5.3.2.8 registrarVitoria()	15
5.4 Referência da Classe ExcecaoPosicionamentodeBarco	16
5.4.1 Documentação das funções	16
5.4.1.1 what()	16
5.5 Referência da Classe ExcecaoTipodeBarcoInvalido	16
5.5.1 Documentação das funções	17
5.5.1.1 what()	17
5.6 Referência da Classe Jogador	17
5.6.1 Construtores e Destrutores	17
5.6.1.1 Jogador() [1/2]	17
5.6.1.2 Jogador() [2/2]	17
5.6.2 Documentação das funções	18
5.6.2.1 getApelido()	18
5.6.2.2 getDerrotas()	18
5.6.2.3 getEmpates()	18
5.6.2.4 getNome()	18
5.6.2.5 getVitorias()	18
5.6.2.6 mostrarEstatisticas()	18
5.6.2.7 registrarDerrota()	18
5.6.2.8 registrarEmpate()	18
5.6.2.9 registrarVitoria()	19
5.7 Referência da Classe JogoDaVelha	19
5.7.1 Construtores e Destrutores	20
5.7.1.1 JogoDaVelha() [1/2]	20
5.7.1.2 JogoDaVelha() [2/2]	20
5.7.2 Documentação das funções	20
5.7.2.1 anunciarInicioPartida()	20

5.7.2.2	checarColunas()	20
5.7.2.3	checarDiagonal()	21
5.7.2.4	checarEmpate()	21
5.7.2.5	checarLinhas()	21
5.7.2.6	checarVencedor()	21
5.7.2.7	lerJogada()	21
5.7.3	Documentação dos símbolos amigos e relacionados	21
5.7.3.1	JogoDaVelhaAi	21
5.8	Referência da Classe JogoDaVelhaAi	22
5.8.1	Construtores e Destrutores	22
5.8.1.1	JogoDaVelhaAi()	22
5.8.2	Documentação das funções	22
5.8.2.1	Jogar()	22
5.8.2.2	playGame()	22
5.9	Referência da Classe Jogos	22
5.9.1	Documentação das funções	23
5.9.1.1	anunciarInicioPartida()	23
5.9.1.2	anunciarTurnoJogador()	23
5.9.1.3	checarEmpate()	23
5.9.1.4	checarJogadaExistente()	24
5.9.1.5	checarPosicaoValida()	24
5.9.1.6	checarVencedor()	24
5.9.1.7	gerarDivisoriaTabuleiro()	24
5.9.1.8	Jogar()	24
5.9.1.9	lerJogada()	24
5.9.1.10	limparTabuleiro()	24
5.9.1.11	marcarTabuleiro() [1/2]	25
5.9.1.12	marcarTabuleiro() [2/2]	25
5.9.1.13	mostrarTabuleiro() [1/2]	25
5.9.1.14	mostrarTabuleiro() [2/2]	25
5.9.1.15	sorteioTurno()	25
5.9.2	Atributos	25
5.9.2.1	tabuleiro	25
5.10	Referência da Classe Lig4	26
5.10.1	Construtores e Destrutores	27
5.10.1.1	Lig4() [1/2]	27
5.10.1.2	Lig4() [2/2]	27
5.10.2	Documentação das funções	27
5.10.2.1	anunciarInicioPartida()	27
5.10.2.2	checarColunas()	27
5.10.2.3	checarDiagonal()	27
5.10.2.4	checarEmpate()	27

5.10.2.5	checarLinhas()	27
5.10.2.6	checarVencedor()	28
5.10.2.7	lerJogada()	28
5.11	Referência da Classe Reversi	28
5.11.1	Construtores e Destrutores	29
5.11.1.1	Reversi() [1/2]	29
5.11.1.2	Reversi() [2/2]	29
5.11.2	Documentação das funções	29
5.11.2.1	anunciarInicioPartida()	29
5.11.2.2	checarEmpate()	30
5.11.2.3	checarVencedor() [1/2]	30
5.11.2.4	checarVencedor() [2/2]	30
5.11.2.5	haMovimentosDisponiveis()	30
5.11.2.6	jogadorInicial()	30
5.11.2.7	Jogar()	30
5.11.2.8	lerJogada()	30
5.11.2.9	limparTabuleiro()	31
5.11.2.10	marcarTabuleiro()	31
5.11.2.11	movimentoValido()	31
6	Arquivos	33
6.1	Referência do Arquivo include/BatalhaNaval.hpp	33
6.2	BatalhaNaval.hpp	33
6.3	Referência do Arquivo include/CentralDeJogos.hpp	34
6.4	CentralDeJogos.hpp	34
6.5	Referência do Arquivo include/Estatisticas.hpp	35
6.6	Estatisticas.hpp	35
6.7	Referência do Arquivo include/Jogador.hpp	35
6.8	Jogador.hpp	36
6.9	Referência do Arquivo include/JogoDaVelha.hpp	36
6.10	JogoDaVelha.hpp	36
6.11	Referência do Arquivo include/JogoDaVelhaAi.hpp	37
6.11.1	Variáveis	37
6.11.1.1	BOARD_SIZE	37
6.11.1.2	EMPTY	37
6.11.1.3	PLAYER_O	37
6.11.1.4	PLAYER_X	37
6.12	JogoDaVelhaAi.hpp	38
6.13	Referência do Arquivo include/Jogos.hpp	38
6.14	Jogos.hpp	38
6.15	Referência do Arquivo include/Lig4.hpp	39
6.16	Lig4.hpp	39

6.17 Referência do Arquivo include/Reversi.hpp	40
6.18 Reversi.hpp	40
6.19 Referência do Arquivo src/BatalhaNaval.cpp	40
6.19.1 Funções	41
6.19.1.1 inserirBarcos()	41
6.20 Referência do Arquivo src/CentralDeJogos.cpp	41
6.21 Referência do Arquivo src/Estatisticas.cpp	41
6.22 Referência do Arquivo src/Jogador.cpp	41
6.23 Referência do Arquivo src/JogoDaVelha.cpp	41
6.24 Referência do Arquivo src/JogoDaVelhaAi.cpp	41
6.25 Referência do Arquivo src/Jogos.cpp	42
6.26 Referência do Arquivo src/Lig4.cpp	42
6.27 Referência do Arquivo src/main.cpp	42
6.27.1 Funções	42
6.27.1.1 exibirMenu()	42
6.27.1.2 main()	42
6.27.1.3 validarEntrada()	42
6.28 Referência do Arquivo src/Reversi.cpp	42
6.29 Referência do Arquivo UI/README.md	42
Índice Remissivo	43

Capítulo 1

UI

1.1 Objetivo

O objetivo da UI do projeto é melhorar a experiência do usuário ao interagir com o código em c++.

O código em c++ pode ser tanto rodado usando a UI como diretamente no terminal. Para executar o código com a UI, siga as instruções disponíveis mais a frente.

1.2 Solução

A solução usada para desenvolver a UI foi o REACT. Usando código em javascript, desenvolvemos uma interface gráfica para o projeto que pode ser executada diretamente em qualquer navegador.

Podemos dividir, de uma forma geral, a UI entre telas e componentes.

1.2.1 O workflow REACT

O react trabalha usando html, css, e javascript para criar páginas web.

O diferencial que o uso do REACT proporciona é a capacidade de criar "componentes": Funções que podem ser chamadas como tags html. A criação de componentes proporciona uma facilidade imensa em criar partes reutilizáveis para o site.

Além disso, o REACT proporciona diversos outros benefícios, como a renderização condicional (usando "state variables" é possível re-renderizar apenas as partes do site que sofreram mudanças, melhorando o desempenho da aplicação) e as routes (forma de navegar entre as páginas do site sem ter que recarregar a janela).

1.2.2 Telas

1.2.2.1 Seleção de jogadores

1.3 Como rodar?

- 1- Instale o package manager NPM na sua máquina
- 2- Abra um terminal no diretório UI
- 2- Digite `npm install` e pressione enter
- 3- No mesmo terminal, digite `npm run dev`
- 4- Copie o link resultante e cole-o em um navegador de sua escolha

IMPORTANTE: não feche a janela do terminal enquanto estiver usando a UI

Capítulo 2

Índice Hierárquico

2.1 Hierarquia de Classes

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

CentralDeJogos	12
Estatisticas	14
std::exception	
ExcecaoPosicionamentodeBarco	16
ExcecaoTipodeBarcoInvalido	16
Jogador	17
JogoDaVelhaAi	22
Jogos	22
BatalhaNaval	9
JogoDaVelha	19
Lig4	26
Reversi	28

Capítulo 3

Índice dos Componentes

3.1 Lista de Classes

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

BatalhaNaval	9
CentralDeJogos	12
Estatisticas	14
ExcecaoPosicionamentodeBarco	16
ExcecaoTipodeBarcoInvalido	16
Jogador	17
JogoDaVelha	19
JogoDaVelhaAi	22
Jogos	22
Lig4	26
Reversi	28

Capítulo 4

Índice dos Arquivos

4.1 Lista de Arquivos

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

include/BatalhaNaval.hpp	33
include/CentralDeJogos.hpp	34
include/Estatisticas.hpp	35
include/Jogador.hpp	35
include/JogoDaVelha.hpp	36
include/JogoDaVelhaAi.hpp	37
include/Jogos.hpp	38
include/Lig4.hpp	39
include/Reversi.hpp	40
src/BatalhaNaval.cpp	40
src/CentralDeJogos.cpp	41
src/Estatisticas.cpp	41
src/Jogador.cpp	41
src/JogoDaVelha.cpp	41
src/JogoDaVelhaAi.cpp	41
src/Jogos.cpp	42
src/Lig4.cpp	42
src/main.cpp	42
src/Reversi.cpp	42

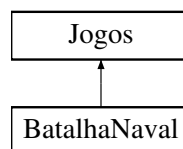
Capítulo 5

Classes

5.1 Referência da Classe BatalhaNaval

```
#include <BatalhaNaval.hpp>
```

Diagrama de hierarquia da classe BatalhaNaval:



Membros Públicos

- [BatalhaNaval](#) ()

Membros Públicos herdados de [Jogos](#)

- virtual void [mostrarTabuleiro](#) ()
- virtual void [mostrarTabuleiro](#) (const std::vector< std::vector< char > > &[tabuleiro](#))

Membros protegidos

- void [anunciarInicioPartida](#) ([Jogador](#) &Jogador1, [Jogador](#) &Jogador2, bool &turno) override
- bool [checarVencedor](#) (std::vector< std::pair< int, int > > &jogadas, [Jogador](#) &vencedor, [Jogador](#) &perdedor) override
- bool [checarVencedor](#) (std::vector< std::pair< int, int > > &jogadasAtacante, std::vector< std::pair< int, int > > &barcosOponente, [Jogador](#) &vencedor, [Jogador](#) &perdedor)
- std::pair< int, int > [lerJogada](#) (std::vector< std::vector< char > > &tabuleiroJogador)
- void [lerBarcos](#) (std::vector< std::pair< int, int > > &barcosJogador, [Jogador](#) &[Jogador](#))
- void [inserirBarcos](#) (std::vector< std::pair< int, int > > &barcosJogador, char tipo, int linhaInicial, int colunaInicial, int linhaFinal, int colunaFinal)
- bool [quantidadeBarcosDisponiveis](#) (std::map< char, int > &countBarcos, char tipo)
- bool [verificarEntrada](#) (char tipo, int linhaInicial, int colunaInicial, int linhaFinal, int colunaFinal)
- bool [verificarTamanhoDoBarco](#) (char tipo, int linhaInicial, int colunaInicial, int linhaFinal, int colunaFinal)
- void [checarPosicaoValida](#) (std::vector< std::vector< char > > &[tabuleiro](#))
- void [Jogar](#) ([Jogador](#) &Jogador1, [Jogador](#) &Jogador2) override
- void [marcarTabuleiro](#) (std::pair< int, int > &jogada, bool &turno, std::vector< std::vector< char > > &tabuleiroJogador, std::vector< std::pair< int, int > > &barcosJogador) override
- void [checarEmpate](#) ()

Membros protegidos herdados de **Jogos**

- virtual void `marcarTabuleiro` (std::pair< int, int > &jogada, bool &turno)
- virtual void `limparTabuleiro` ()
- void `anunciarTurnoJogador` (Jogador &Jogador)
- virtual bool `sorteioTurno` ()
- virtual bool `checarJogadaExistente` (std::vector< std::pair< int, int > > &jogadas, int linha, int coluna)
- virtual bool `checarPosicaoValida` (int linha, int coluna)
- virtual bool `checarEmpate` (int numeroJogadas, Jogador &jogador_01, Jogador &jogador_02)=0
- std::string `gerarDivisoriaTabuleiro` ()
- virtual std::pair< int, int > `lerJogada` ()

Outros membros herdados

Atributos Protegidos herdados de **Jogos**

- std::vector< std::vector< char > > `tabuleiro`

5.1.1 Construtores e Destrutores

5.1.1.1 BatalhaNaval()

```
BatalhaNaval::BatalhaNaval ()
```

5.1.2 Documentação das funções

5.1.2.1 anunciarInicioPartida()

```
void BatalhaNaval::anunciarInicioPartida (  
    Jogador & Jogador1,  
    Jogador & Jogador2,  
    bool & turno) [override], [protected], [virtual]
```

Implementa `Jogos`.

5.1.2.2 checarEmpate()

```
void BatalhaNaval::checarEmpate () [protected]
```

5.1.2.3 checarPosicaoValida()

```
void BatalhaNaval::checarPosicaoValida (  
    std::vector< std::vector< char > > & tabuleiro) [protected]
```

5.1.2.4 checarVencedor() [1/2]

```
bool BatalhaNaval::checarVencedor (
    std::vector< std::pair< int, int > > & jogadas,
    Jogador & vencedor,
    Jogador & perdedor) [override], [protected], [virtual]
```

Implementa [Jogos](#).

5.1.2.5 checarVencedor() [2/2]

```
bool BatalhaNaval::checarVencedor (
    std::vector< std::pair< int, int > > & jogadasAtacante,
    std::vector< std::pair< int, int > > & barcosOponente,
    Jogador & vencedor,
    Jogador & perdedor) [protected]
```

5.1.2.6 inserirBarcos()

```
void BatalhaNaval::inserirBarcos (
    std::vector< std::pair< int, int > > & barcosJogador,
    char tipo,
    int linhaInicial,
    int colunaInicial,
    int linhaFinal,
    int colunaFinal) [protected]
```

5.1.2.7 Jogar()

```
void BatalhaNaval::Jogar (
    Jogador & Jogador1,
    Jogador & Jogador2) [override], [protected], [virtual]
```

Reimplementa [Jogos](#).

5.1.2.8 lerBarcos()

```
void BatalhaNaval::lerBarcos (
    std::vector< std::pair< int, int > > & barcosJogador,
    Jogador & Jogador) [protected]
```

5.1.2.9 lerJogada()

```
std::pair< int, int > BatalhaNaval::lerJogada (
    std::vector< std::vector< char > > & tabuleiroJogador) [protected]
```

5.1.2.10 marcarTabuleiro()

```
void BatalhaNaval::marcarTabuleiro (
    std::pair< int, int > & jogada,
    bool & turno,
    std::vector< std::vector< char > > & tabuleiroJogador,
    std::vector< std::pair< int, int > > & barcosJogador) [override], [protected],
[virtual]
```

Reimplementa [Jogos](#).

5.1.2.11 quantidadeBarcosDisponiveis()

```
bool BatalhaNaval::quantidadeBarcosDisponiveis (
    std::map< char, int > & countBarcos,
    char tipo) [protected]
```

5.1.2.12 verificarEntrada()

```
bool BatalhaNaval::verificarEntrada (
    char tipo,
    int linhaInicial,
    int colunaInicial,
    int linhaFinal,
    int colunaFinal) [protected]
```

5.1.2.13 verificarTamanhoDoBarco()

```
bool BatalhaNaval::verificarTamanhoDoBarco (
    char tipo,
    int linhaInicial,
    int colunaInicial,
    int linhaFinal,
    int colunaFinal) [protected]
```

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

- [include/BatalhaNaval.hpp](#)
- [src/BatalhaNaval.cpp](#)

5.2 Referência da Classe CentralDeJogos

```
#include <CentralDeJogos.hpp>
```

Membros Públicos

- [CentralDeJogos](#) ()
- [~CentralDeJogos](#) ()
- `std::string` [validarEntrada](#) ()
- `bool` [buscarJogador](#) (`std::string` &apelido)
- `void` [cadastrarJogador](#) (`std::string` &apelido, `std::string` &nome)
- `void` [removerJogador](#) (`std::string` &apelido)
- `void` [ordenarJogadores](#) ()
- `void` [listarJogadores](#) ()
- `void` [executarPartida](#) ()

5.2.1 Construtores e Destrutores

5.2.1.1 CentralDeJogos()

```
CentralDeJogos::CentralDeJogos ()
```

5.2.1.2 ~CentralDeJogos()

```
CentralDeJogos::~~CentralDeJogos ()
```

5.2.2 Documentação das funções

5.2.2.1 buscarJogador()

```
bool CentralDeJogos::buscarJogador (  
    std::string & apelido)
```

5.2.2.2 cadastrarJogador()

```
void CentralDeJogos::cadastrarJogador (  
    std::string & apelido,  
    std::string & nome)
```

5.2.2.3 executarPartida()

```
void CentralDeJogos::executarPartida ()
```

5.2.2.4 listarJogadores()

```
void CentralDeJogos::listarJogadores ()
```

5.2.2.5 ordenarJogadores()

```
void CentralDeJogos::ordenarJogadores ()
```

5.2.2.6 removerJogador()

```
void CentralDeJogos::removerJogador (  
    std::string & apelido)
```

5.2.2.7 validarEntrada()

```
std::string CentralDeJogos::validarEntrada ()
```

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

- [include/CentralDeJogos.hpp](#)
- [src/CentralDeJogos.cpp](#)

5.3 Referência da Classe Estatísticas

```
#include <Estatisticas.hpp>
```

Membros Públicos

- [Estatisticas](#) ()
- [Estatisticas](#) (int vitórias, int derrotas, int empates)
- void [registrarVitoria](#) ()
- void [registrarDerrota](#) ()
- void [registrarEmpate](#) ()
- int [getVitorias](#) () const
- int [getDerrotas](#) () const
- int [getEmpates](#) () const
- std::vector< char > [getHistorico](#) () const
- void [mostrarEstatisticas](#) () const

5.3.1 Construtores e Destrutores

5.3.1.1 Estatisticas() [1/2]

```
Estatisticas::Estatisticas ()
```

5.3.1.2 Estatisticas() [2/2]

```
Estatisticas::Estatisticas (  
    int vitórias,  
    int derrotas,  
    int empates)
```

5.3.2 Documentação das funções

5.3.2.1 getDerrotas()

```
int Estatisticas::getDerrotas () const
```

5.3.2.2 getEmpates()

```
int Estatisticas::getEmpates () const
```

5.3.2.3 getHistorico()

```
std::vector< char > Estatisticas::getHistorico () const
```

5.3.2.4 getVitorias()

```
int Estatisticas::getVitorias () const
```

5.3.2.5 mostrarEstatisticas()

```
void Estatisticas::mostrarEstatisticas () const
```

5.3.2.6 registrarDerrota()

```
void Estatisticas::registrarDerrota ()
```

5.3.2.7 registrarEmpate()

```
void Estatisticas::registrarEmpate ()
```

5.3.2.8 registrarVitoria()

```
void Estatisticas::registrarVitoria ()
```

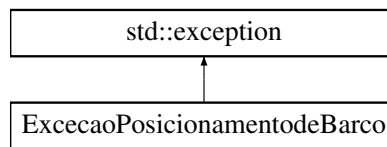
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- [include/Estatisticas.hpp](#)
- [src/Estatisticas.cpp](#)

5.4 Referência da Classe ExcecaoPosicionamentodeBarco

```
#include <BatalhaNaval.hpp>
```

Diagrama de hierarquia da classe ExcecaoPosicionamentodeBarco:



Membros Públicos

- `const char * what () const override throw ()`

5.4.1 Documentação das funções

5.4.1.1 what()

```
const char * ExcecaoPosicionamentodeBarco::what () const throw ( ) [inline], [override]
```

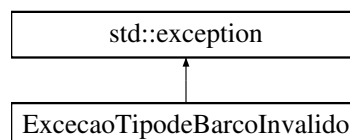
A documentação para essa classe foi gerada a partir do seguinte arquivo:

- `include/BatalhaNaval.hpp`

5.5 Referência da Classe ExcecaoTipodeBarcoInvalido

```
#include <BatalhaNaval.hpp>
```

Diagrama de hierarquia da classe ExcecaoTipodeBarcoInvalido:



Membros Públicos

- `const char * what () const override throw ()`

5.5.1 Documentação das funções

5.5.1.1 what()

```
const char * ExcecaoTipodeBarcoInvalido::what () const throw ( ) [inline], [override]
```

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

- include/[BatalhaNaval.hpp](#)

5.6 Referência da Classe Jogador

```
#include <Jogador.hpp>
```

Membros Públicos

- [Jogador](#) (const std::string &apelido, const std::string &nome)
- [Jogador](#) (const std::string &apelido, const std::string &nome, int vitoriasJogoDaVelha, int derrotasJogoDaVelha, int empatesJogoDaVelha, int vitoriasLig4, int derrotasLig4, int empatesLig4, int vitoriasReversi, int derrotasReversi, int empatesReversi, int vitoriasBatalhaNaval, int derrotasBatalhaNaval, int empatesBatalhaNaval)
- void [registrarVitoria](#) (const std::string &nomeJogo)
- void [registrarDerrota](#) (const std::string &nomeJogo)
- void [registrarEmpate](#) (const std::string &nomeJogo)
- std::string [getApelido](#) () const
- std::string [getNome](#) () const
- int [getVitorias](#) (std::string jogo)
- int [getDerrotas](#) (std::string jogo)
- int [getEmpates](#) (std::string jogo)
- void [mostrarEstatisticas](#) (const std::string &nomeJogo) const

5.6.1 Construtores e Destrutores

5.6.1.1 Jogador() [1/2]

```
Jogador::Jogador (
    const std::string & apelido,
    const std::string & nome)
```

5.6.1.2 Jogador() [2/2]

```
Jogador::Jogador (
    const std::string & apelido,
    const std::string & nome,
    int vitoriasJogoDaVelha,
    int derrotasJogoDaVelha,
    int empatesJogoDaVelha,
    int vitoriasLig4,
    int derrotasLig4,
    int empatesLig4,
    int vitoriasReversi,
    int derrotasReversi,
    int empatesReversi,
    int vitoriasBatalhaNaval,
    int derrotasBatalhaNaval,
    int empatesBatalhaNaval)
```

5.6.2 Documentação das funções

5.6.2.1 getApelido()

```
std::string Jogador::getApelido () const
```

5.6.2.2 getDerrotas()

```
int Jogador::getDerrotas (  
    std::string jogo)
```

5.6.2.3 getEmpates()

```
int Jogador::getEmpates (  
    std::string jogo)
```

5.6.2.4 getNome()

```
std::string Jogador::getNome () const
```

5.6.2.5 getVitorias()

```
int Jogador::getVitorias (  
    std::string jogo)
```

5.6.2.6 mostrarEstatisticas()

```
void Jogador::mostrarEstatisticas (  
    const std::string & nomeJogo) const
```

5.6.2.7 registrarDerrota()

```
void Jogador::registrarDerrota (  
    const std::string & nomeJogo)
```

5.6.2.8 registrarEmpate()

```
void Jogador::registrarEmpate (  
    const std::string & nomeJogo)
```

5.6.2.9 registrarVitoria()

```
void Jogador::registrarVitoria (
    const std::string & nomeJogo)
```

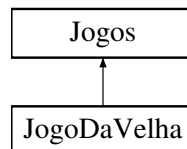
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- [include/Jogador.hpp](#)
- [src/Jogador.cpp](#)

5.7 Referência da Classe JogoDaVelha

```
#include <JogoDaVelha.hpp>
```

Diagrama de hierarquia da classe JogoDaVelha:



Membros Públicos

- [JogoDaVelha](#) ()
- [JogoDaVelha](#) (int tamanhoTabuleiro)

Membros Públicos herdados de [Jogos](#)

- virtual void [mostrarTabuleiro](#) ()
- virtual void [mostrarTabuleiro](#) (const std::vector< std::vector< char > > &tabuleiro)
- virtual void [Jogar](#) ([Jogador](#) &Jogador1, [Jogador](#) &Jogador2)

Membros protegidos

- void [anunciarInicioPartida](#) ([Jogador](#) &Jogador1, [Jogador](#) &Jogador2, bool &turno) override
- std::pair< int, int > [lerJogada](#) () override
- bool [checarDiagonal](#) (std::vector< std::pair< int, int > > &jogadas)
- bool [checarColunas](#) (std::vector< std::pair< int, int > > &jogadas)
- bool [checarLinhas](#) (std::vector< std::pair< int, int > > &jogadas)
- bool [checarVencedor](#) (std::vector< std::pair< int, int > > &jogadas, [Jogador](#) &vencedor, [Jogador](#) &perdedor) override
- bool [checarEmpate](#) (int numeroJogadas, [Jogador](#) &jogador_01, [Jogador](#) &jogador_02) override

Membros protegidos herdados de **Jogos**

- virtual void `marcarTabuleiro` (std::pair< int, int > &jogada, bool &turno)
- virtual void `marcarTabuleiro` (std::pair< int, int > &jogada, bool &turno, std::vector< std::vector< char > > &tabuleiroJogador, std::vector< std::pair< int, int > > &barcosJogador)
- virtual void `limparTabuleiro` ()
- void `anunciarTurnoJogador` (Jogador &Jogador)
- virtual bool `sorteioTurno` ()
- virtual bool `checarJogadaExistente` (std::vector< std::pair< int, int > > &jogadas, int linha, int coluna)
- virtual bool `checarPosicaoValida` (int linha, int coluna)
- std::string `gerarDivisoriaTabuleiro` ()

Amigos

- class `JogoDaVelhaAi`

Outros membros herdados

Atributos Protegidos herdados de **Jogos**

- std::vector< std::vector< char > > `tabuleiro`

5.7.1 Construtores e Destrutores

5.7.1.1 JogoDaVelha() [1/2]

```
JogoDaVelha::JogoDaVelha ()
```

5.7.1.2 JogoDaVelha() [2/2]

```
JogoDaVelha::JogoDaVelha (
    int tamanhoTabuleiro)
```

5.7.2 Documentação das funções

5.7.2.1 anunciarInicioPartida()

```
void JogoDaVelha::anunciarInicioPartida (
    Jogador & Jogador1,
    Jogador & Jogador2,
    bool & turno) [override], [protected], [virtual]
```

Implementa `Jogos`.

5.7.2.2 checarColunas()

```
bool JogoDaVelha::checarColunas (
    std::vector< std::pair< int, int > > & jogadas) [protected]
```

5.7.2.3 checarDiagonal()

```
bool JogoDaVelha::checarDiagonal (
    std::vector< std::pair< int, int > > & jogadas) [protected]
```

5.7.2.4 checarEmpate()

```
bool JogoDaVelha::checarEmpate (
    int numeroJogadas,
    Jogador & jogador_01,
    Jogador & jogador_02) [override], [protected], [virtual]
```

Implementa [Jogos](#).

5.7.2.5 checarLinhas()

```
bool JogoDaVelha::checarLinhas (
    std::vector< std::pair< int, int > > & jogadas) [protected]
```

5.7.2.6 checarVencedor()

```
bool JogoDaVelha::checarVencedor (
    std::vector< std::pair< int, int > > & jogadas,
    Jogador & vencedor,
    Jogador & perdedor) [override], [protected], [virtual]
```

Implementa [Jogos](#).

5.7.2.7 lerJogada()

```
std::pair< int, int > JogoDaVelha::lerJogada () [override], [protected], [virtual]
```

Reimplementa [Jogos](#).

5.7.3 Documentação dos símbolos amigos e relacionados

5.7.3.1 JogoDaVelhaAi

```
friend class JogoDaVelhaAi [friend]
```

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

- include/[JogoDaVelha.hpp](#)
- src/[JogoDaVelha.cpp](#)

5.8 Referência da Classe JogoDaVelhaAi

```
#include <JogoDaVelhaAi.hpp>
```

Membros Públicos

- [JogoDaVelhaAi](#) ()
- void [playGame](#) ()
- void [Jogar](#) ([Jogador](#) &Jogador1, [Jogador](#) &Jogador2)

5.8.1 Construtores e Destrutores

5.8.1.1 JogoDaVelhaAi()

```
JogoDaVelhaAi::JogoDaVelhaAi ()
```

5.8.2 Documentação das funções

5.8.2.1 Jogar()

```
void JogoDaVelhaAi::Jogar (  
    Jogador & Jogador1,  
    Jogador & Jogador2)
```

5.8.2.2 playGame()

```
void JogoDaVelhaAi::playGame ()
```

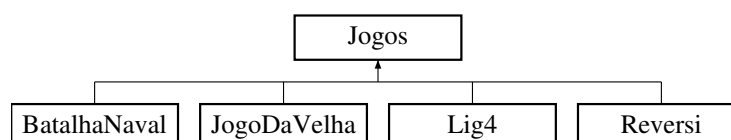
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- include/[JogoDaVelhaAi.hpp](#)
- src/[JogoDaVelhaAi.cpp](#)

5.9 Referência da Classe Jogos

```
#include <Jogos.hpp>
```

Diagrama de hierarquia da classe Jogos:



Membros Públicos

- virtual void `mostrarTabuleiro` ()
- virtual void `mostrarTabuleiro` (const std::vector< std::vector< char > > &`tabuleiro`)
- virtual void `Jogar` (`Jogador` &`Jogador1`, `Jogador` &`Jogador2`)

Membros protegidos

- virtual void `marcarTabuleiro` (std::pair< int, int > &`jogada`, bool &`turno`)
- virtual void `marcarTabuleiro` (std::pair< int, int > &`jogada`, bool &`turno`, std::vector< std::vector< char > > &`tabuleiroJogador`, std::vector< std::pair< int, int > > &`barcosJogador`)
- virtual void `limparTabuleiro` ()
- virtual void `anunciarInicioPartida` (`Jogador` &`Jogador1`, `Jogador` &`Jogador2`, bool &`turno`)=0
- void `anunciarTurnoJogador` (`Jogador` &`Jogador`)
- virtual bool `sorteioTurno` ()
- virtual bool `checarJogadaExistente` (std::vector< std::pair< int, int > > &`jogadas`, int `linha`, int `coluna`)
- virtual bool `checarPosicaoValida` (int `linha`, int `coluna`)
- virtual bool `checarVencedor` (std::vector< std::pair< int, int > > &`jogadas`, `Jogador` &`vencedor`, `Jogador` &`perdedor`)=0
- virtual bool `checarEmpate` (int `numeroJogadas`, `Jogador` &`jogador_01`, `Jogador` &`jogador_02`)=0
- std::string `gerarDivisoriaTabuleiro` ()
- virtual std::pair< int, int > `lerJogada` ()

Atributos Protegidos

- std::vector< std::vector< char > > `tabuleiro`

5.9.1 Documentação das funções

5.9.1.1 anunciarInicioPartida()

```
virtual void Jogos::anunciarInicioPartida (  
    Jogador & Jogador1,  
    Jogador & Jogador2,  
    bool & turno) [protected], [pure virtual]
```

Implementado por [BatalhaNaval](#), [JogoDaVelha](#), [Lig4](#) e [Reversi](#).

5.9.1.2 anunciarTurnoJogador()

```
void Jogos::anunciarTurnoJogador (  
    Jogador & Jogador) [protected]
```

5.9.1.3 checarEmpate()

```
virtual bool Jogos::checarEmpate (  
    int numeroJogadas,  
    Jogador & jogador_01,  
    Jogador & jogador_02) [protected], [pure virtual]
```

Implementado por [JogoDaVelha](#), [Lig4](#) e [Reversi](#).

5.9.1.4 `checarJogadaExistente()`

```
bool Jogos::checarJogadaExistente (
    std::vector< std::pair< int, int > > & jogadas,
    int linha,
    int coluna) [protected], [virtual]
```

5.9.1.5 `checarPosicaoValida()`

```
bool Jogos::checarPosicaoValida (
    int linha,
    int coluna) [protected], [virtual]
```

5.9.1.6 `checarVencedor()`

```
virtual bool Jogos::checarVencedor (
    std::vector< std::pair< int, int > > & jogadas,
    Jogador & vencedor,
    Jogador & perdedor) [protected], [pure virtual]
```

Implementado por [BatalhaNaval](#), [JogoDaVelha](#), [Lig4](#) e [Reversi](#).

5.9.1.7 `gerarDivisoriaTabuleiro()`

```
std::string Jogos::gerarDivisoriaTabuleiro () [protected]
```

5.9.1.8 `Jogar()`

```
void Jogos::Jogar (
    Jogador & Jogador1,
    Jogador & Jogador2) [virtual]
```

Reimplementado por [BatalhaNaval](#) e [Reversi](#).

5.9.1.9 `lerJogada()`

```
virtual std::pair< int, int > Jogos::lerJogada () [inline], [protected], [virtual]
```

Reimplementado por [JogoDaVelha](#) e [Lig4](#).

5.9.1.10 `limparTabuleiro()`

```
void Jogos::limparTabuleiro () [protected], [virtual]
```

Reimplementado por [Reversi](#).

5.9.1.11 marcarTabuleiro() [1/2]

```
void Jogos::marcarTabuleiro (
    std::pair< int, int > & jogada,
    bool & turno) [protected], [virtual]
```

Reimplementado por [Reversi](#).

5.9.1.12 marcarTabuleiro() [2/2]

```
virtual void Jogos::marcarTabuleiro (
    std::pair< int, int > & jogada,
    bool & turno,
    std::vector< std::vector< char > > & tabuleiroJogador,
    std::vector< std::pair< int, int > > & barcosJogador) [protected], [virtual]
```

Reimplementado por [BatalhaNaval](#).

5.9.1.13 mostrarTabuleiro() [1/2]

```
void Jogos::mostrarTabuleiro () [virtual]
```

5.9.1.14 mostrarTabuleiro() [2/2]

```
virtual void Jogos::mostrarTabuleiro (
    const std::vector< std::vector< char > > & tabuleiro) [virtual]
```

5.9.1.15 sorteioTurno()

```
bool Jogos::sorteioTurno () [protected], [virtual]
```

5.9.2 Atributos**5.9.2.1 tabuleiro**

```
std::vector<std::vector<char> > Jogos::tabuleiro [protected]
```

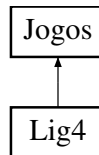
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- [include/Jogos.hpp](#)
- [src/Jogos.cpp](#)

5.10 Referência da Classe Lig4

```
#include <Lig4.hpp>
```

Diagrama de hierarquia da classe Lig4:



Membros Públicos

- [Lig4](#) (int tamanhoTabuleiro)
- [Lig4](#) ()

Membros Públicos herdados de [Jogos](#)

- virtual void [mostrarTabuleiro](#) ()
- virtual void [mostrarTabuleiro](#) (const std::vector< std::vector< char > > &tabuleiro)
- virtual void [Jogar](#) ([Jogador](#) &Jogador1, [Jogador](#) &Jogador2)

Membros protegidos

- void [anunciarInicioPartida](#) ([Jogador](#) &Jogador1, [Jogador](#) &Jogador2, bool &turno) override
- bool [checarDiagonal](#) (std::vector< std::pair< int, int > > &jogadas)
- bool [checarColunas](#) (std::vector< std::pair< int, int > > &jogadas)
- bool [checarLinhas](#) (std::vector< std::pair< int, int > > &jogadas)
- bool [checarVencedor](#) (std::vector< std::pair< int, int > > &jogadas, [Jogador](#) &vencedor, [Jogador](#) &perdedor) override
- bool [checarEmpate](#) (int numeroJogadas, [Jogador](#) &jogador_01, [Jogador](#) &jogador_02) override
- std::pair< int, int > [lerJogada](#) () override

Membros protegidos herdados de [Jogos](#)

- virtual void [marcarTabuleiro](#) (std::pair< int, int > &jogada, bool &turno)
- virtual void [marcarTabuleiro](#) (std::pair< int, int > &jogada, bool &turno, std::vector< std::vector< char > > &tabuleiroJogador, std::vector< std::pair< int, int > > &barcosJogador)
- virtual void [limparTabuleiro](#) ()
- void [anunciarTurnoJogador](#) ([Jogador](#) &Jogador)
- virtual bool [sorteioTurno](#) ()
- virtual bool [checarJogadaExistente](#) (std::vector< std::pair< int, int > > &jogadas, int linha, int coluna)
- virtual bool [checarPosicaoValida](#) (int linha, int coluna)
- std::string [gerarDivisoriaTabuleiro](#) ()

Outros membros herdados

Atributos Protegidos herdados de [Jogos](#)

- std::vector< std::vector< char > > [tabuleiro](#)

5.10.1 Construtores e Destrutores

5.10.1.1 Lig4() [1/2]

```
Lig4::Lig4 (
    int tamanhoTabuleiro)
```

5.10.1.2 Lig4() [2/2]

```
Lig4::Lig4 ()
```

5.10.2 Documentação das funções

5.10.2.1 anunciarInicioPartida()

```
void Lig4::anunciarInicioPartida (
    Jogador & Jogador1,
    Jogador & Jogador2,
    bool & turno) [override], [protected], [virtual]
```

Implementa [Jogos](#).

5.10.2.2 checarColunas()

```
bool Lig4::checarColunas (
    std::vector< std::pair< int, int > > & jogadas) [protected]
```

5.10.2.3 checarDiagonal()

```
bool Lig4::checarDiagonal (
    std::vector< std::pair< int, int > > & jogadas) [protected]
```

5.10.2.4 checarEmpate()

```
bool Lig4::checarEmpate (
    int numeroJogadas,
    Jogador & jogador_01,
    Jogador & jogador_02) [override], [protected], [virtual]
```

Implementa [Jogos](#).

5.10.2.5 checarLinhas()

```
bool Lig4::checarLinhas (
    std::vector< std::pair< int, int > > & jogadas) [protected]
```

5.10.2.6 `checarVencedor()`

```
bool Lig4::checarVencedor (
    std::vector< std::pair< int, int > > & jogadas,
    Jogador & vencedor,
    Jogador & perdedor) [override], [protected], [virtual]
```

Implementa `Jogos`.

5.10.2.7 `lerJogada()`

```
std::pair< int, int > Lig4::lerJogada () [override], [protected], [virtual]
```

Reimplementa `Jogos`.

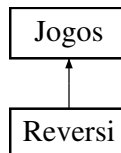
A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- `include/Lig4.hpp`
- `src/Lig4.cpp`

5.11 Referência da Classe Reversi

```
#include <Reversi.hpp>
```

Diagrama de hierarquia da classe Reversi:



Membros Públicos

- `Reversi ()`
- `Reversi (int tamanhoTabuleiro)`
- `void Jogar (Jogador &Jogador1, Jogador &Jogador2) override`

Membros Públicos herdados de `Jogos`

- `virtual void mostrarTabuleiro ()`
- `virtual void mostrarTabuleiro (const std::vector< std::vector< char > > &tabuleiro)`

Membros protegidos

- void `anunciarInicioPartida` (`Jogador &Jogador1`, `Jogador &Jogador2`, `bool &turno`) override
- bool `jogadorInicial` (`bool &turno`)
- `std::pair< int, int >` `lerJogada` (`bool turno`)
- void `marcarTabuleiro` (`std::pair< int, int > &jogada`, `bool &turno`) override
- bool `movimentoValido` (`std::pair< int, int > &jogada`, `char jogador`, `std::vector< std::pair< int, int > > &flips`)
- bool `haMovimentosDisponiveis` (`char Jogador`)
- void `limparTabuleiro` () override
- bool `checarVencedor` (`std::vector< std::pair< int, int > > &jogadas`, `Jogador &vencedor`, `Jogador &perdedor`)
- bool `checarVencedor` (`std::vector< std::pair< int, int > > &jogadas`, `Jogador &jogador1`, `Jogador &jogador2`, `bool &PrimeiroJogador`)
- bool `checarEmpate` (`int numeroJogadas`, `Jogador &jogador_01`, `Jogador &jogador_02`) override

Membros protegidos herdados de Jogos

- virtual void `marcarTabuleiro` (`std::pair< int, int > &jogada`, `bool &turno`, `std::vector< std::vector< char > > &tabuleiroJogador`, `std::vector< std::pair< int, int > > &barcosJogador`)
- void `anunciarTurnoJogador` (`Jogador &Jogador`)
- virtual bool `sorteioTurno` ()
- virtual bool `checarJogadaExistente` (`std::vector< std::pair< int, int > > &jogadas`, `int linha`, `int coluna`)
- virtual bool `checarPosicaoValida` (`int linha`, `int coluna`)
- `std::string` `gerarDivisoriaTabuleiro` ()
- virtual `std::pair< int, int >` `lerJogada` ()

Outros membros herdados

Atributos Protegidos herdados de Jogos

- `std::vector< std::vector< char > >` `tabuleiro`

5.11.1 Construtores e Destrutores

5.11.1.1 Reversi() [1/2]

```
Reversi::Reversi ()
```

5.11.1.2 Reversi() [2/2]

```
Reversi::Reversi (
    int tamanhoTabuleiro)
```

5.11.2 Documentação das funções

5.11.2.1 anunciarInicioPartida()

```
void Reversi::anunciarInicioPartida (
    Jogador & Jogador1,
    Jogador & Jogador2,
    bool & turno) [override], [protected], [virtual]
```

Implementa `Jogos`.

5.11.2.2 `checarEmpate()`

```
bool Reversi::checarEmpate (
    int numeroJogadas,
    Jogador & jogador_01,
    Jogador & jogador_02) [inline], [override], [protected], [virtual]
```

Implementa [Jogos](#).

5.11.2.3 `checarVencedor()` [1/2]

```
bool Reversi::checarVencedor (
    std::vector< std::pair< int, int > > & jogadas,
    Jogador & jogador1,
    Jogador & jogador2,
    bool & PrimeiroJogador) [protected]
```

5.11.2.4 `checarVencedor()` [2/2]

```
bool Reversi::checarVencedor (
    std::vector< std::pair< int, int > > & jogadas,
    Jogador & vencedor,
    Jogador & perdedor) [inline], [protected], [virtual]
```

Implementa [Jogos](#).

5.11.2.5 `haMovimentosDisponiveis()`

```
bool Reversi::haMovimentosDisponiveis (
    char Jogador) [protected]
```

5.11.2.6 `jogadorInicial()`

```
bool Reversi::jogadorInicial (
    bool & turno) [protected]
```

5.11.2.7 `Jogar()`

```
void Reversi::Jogar (
    Jogador & Jogador1,
    Jogador & Jogador2) [override], [virtual]
```

Reimplementa [Jogos](#).

5.11.2.8 `lerJogada()`

```
std::pair< int, int > Reversi::lerJogada (
    bool turno) [protected]
```

5.11.2.9 limparTabuleiro()

```
void Reversi::limparTabuleiro () [override], [protected], [virtual]
```

Reimplementa [Jogos](#).

5.11.2.10 marcarTabuleiro()

```
void Reversi::marcarTabuleiro (  
    std::pair< int, int > & jogada,  
    bool & turno) [override], [protected], [virtual]
```

Reimplementa [Jogos](#).

5.11.2.11 movimentoValido()

```
bool Reversi::movimentoValido (  
    std::pair< int, int > & jogada,  
    char jogador,  
    std::vector< std::pair< int, int > > & flips) [protected]
```

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

- include/[Reversi.hpp](#)
- src/[Reversi.cpp](#)

Capítulo 6

Arquivos

6.1 Referência do Arquivo include/BatalhaNaval.hpp

```
#include "Jogos.hpp"
```

Componentes

- class [BatalhaNaval](#)
- class [ExcecaoTipodeBarcoInvalido](#)
- class [ExcecaoPosicionamentodeBarco](#)

6.2 BatalhaNaval.hpp

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

```
00001 #ifndef BATALHA_NAVAL
00002 #define BATALHA_NAVAL
00003
00004 #include "Jogos.hpp"
00005
00006 class BatalhaNaval : public Jogos
00007 {
00008     public:
00009         BatalhaNaval();
00010
00011     protected:
00012         void anunciarInicioPartida(Jogador &Jogador1, Jogador &Jogador2, bool &turno) override;
00013
00014         bool checarVencedor(std::vector<std::pair<int, int> &jogadas, Jogador &vencedor, Jogador
&perdedor) override;
00015         bool checarVencedor(std::vector<std::pair<int, int> &jogadasAtacante, std::vector<std::pair<int,
int> &barcosOponente, Jogador &vencedor, Jogador &perdedor);
00016         std::pair<int, int> lerJogada(std::vector<std::vector<char> &tabuleiroJogador);
00017
00018         void lerBarcos(std::vector<std::pair<int, int> &barcosJogador, Jogador &Jogador);
00019         void inserirBarcos(std::vector<std::pair<int, int> &barcosJogador, char tipo, int linhaInicial,
int colunaInicial,
00020                             int linhaFinal, int colunaFinal);
00021         bool quantidadeBarcosDisponiveis(std::map<char, int> &countBarcos, char tipo);
00022         bool verificarEntrada(char tipo, int linhaInicial, int colunaInicial, int linhaFinal, int
colunaFinal);
00023         bool verificarTamanhoDoBarco(char tipo, int linhaInicial, int colunaInicial, int linhaFinal, int
colunaFinal);
00024         void checarPosicaoValida(std::vector<std::vector<char> &tabuleiro);
00025         void Jogar(Jogador &Jogador1, Jogador &Jogador2) override;
00026         void marcarTabuleiro(std::pair<int, int> &jogada, bool &turno, std::vector<std::vector<char>
&tabuleiroJogador,
```

```

00027         std::vector<std::pair<int, int> &barcosJogador) override;
00028     void checarEmpate();
00029 };
00030
00031 class ExcecaoTipodeBarcoInvalido : public std::exception {
00032 public:
00033     const char* what() const throw() override {
00034         return "ERRO! Tipo de barco invalido.";
00035     }
00036 };
00037
00038 class ExcecaoPosicionamentodeBarco : public std::exception {
00039 public:
00040     const char* what() const throw() override {
00041         return "ERRO! Barcos devem ser colocados horizontalmente ou verticalmente.";
00042     }
00043 };
00044
00045 #endif

```

6.3 Referência do Arquivo include/CentralDeJogos.hpp

```

#include "JogoDaVelha.hpp"
#include "JogoDaVelhaAi.hpp"
#include "Jogos.hpp"
#include "Lig4.hpp"
#include "Reversi.hpp"

```

Componentes

- class [CentralDeJogos](#)

6.4 CentralDeJogos.hpp

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

```

00001 #ifndef CENTRALDEJOGOS_HPP
00002 #define CENTRALDEJOGOS_HPP
00003
00004 #include "JogoDaVelha.hpp"
00005 #include "JogoDaVelhaAi.hpp"
00006 #include "Jogos.hpp"
00007 #include "Lig4.hpp"
00008 #include "Reversi.hpp"
00009
00010 class CentralDeJogos
00011 {
00012     private:
00013         std::vector<Jogador> jogadoresCadastrados;
00014         JogoDaVelhaAi Ai;
00015         JogoDaVelha velha;
00016         Lig4 lig4;
00017         Reversi reversi;
00018
00019     public:
00020         CentralDeJogos();
00021         ~CentralDeJogos();
00022
00023         std::string validarEntrada();
00024
00025         bool buscarJogador(std::string &apelido);
00026
00027         void cadastrarJogador(std::string &apelido, std::string &nome);
00028         void removerJogador(std::string &apelido);
00029         void ordenarJogadores();
00030         void listarJogadores();
00031
00032         void executarPartida();
00033 };
00034
00035 #endif

```

6.5 Referência do Arquivo include/Estatisticas.hpp

```
#include <vector>
```

Componentes

- class [Estatisticas](#)

6.6 Estatisticas.hpp

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

```
00001 #ifndef ESTATISTICAS_HPP
00002 #define ESTATISTICAS_HPP
00003
00004 #include <vector>
00005
00006 class Estatisticas
00007 {
00008     private:
00009         int vitorias;
00010         int derrotas;
00011         int empates;
00012     public:
00013         Estatisticas();
00014         Estatisticas(int vitorias, int derrotas, int empates);
00015
00016         void registrarVitoria();
00017         void registrarDerrota();
00018         void registrarEmpate();
00019
00020         int getVitorias() const;
00021         int getDerrotas() const;
00022         int getEmpates() const;
00023         std::vector<char> getHistorico() const;
00024         void mostrarEstatisticas() const;
00025 };
00026
00027 #endif
```

6.7 Referência do Arquivo include/Jogador.hpp

```
#include <map>
#include <string>
#include <unordered_map>
#include "Estatisticas.hpp"
```

Componentes

- class [Jogador](#)

6.8 Jogador.hpp

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

```
00001 #ifndef JOGADOR_HPP
00002 #define JOGADOR_HPP
00003
00004 #include <map>
00005 #include <string>
00006 #include <unordered_map>
00007
00008 #include "Estatisticas.hpp"
00009
00010 class Jogador {
00011 private:
00012     std::string apelido;
00013     std::string nome;
00014     std::unordered_map<std::string, Estatisticas> estatisticasPorJogo;
00015
00016 public:
00017     Jogador(const std::string &apelido, const std::string &nome);
00018     Jogador(const std::string &apelido, const std::string &nome,
00019             int vitoriasJogoDaVelha, int derrotasJogoDaVelha, int empatesJogoDaVelha,
00020             int vitoriasLig4, int derrotasLig4, int empatesLig4,
00021             int vitoriasReversi, int derrotasReversi, int empatesReversi,
00022             int vitoriasBatalhaNaval, int derrotasBatalhaNaval, int empatesBatalhaNaval);
00023
00024     void registrarVitoria(const std::string &nomeJogo);
00025     void registrarDerrota(const std::string &nomeJogo);
00026     void registrarEmpate(const std::string &nomeJogo);
00027
00028     std::string getApelido() const;
00029     std::string getNome() const;
00030     int getVitorias(std::string jogo);
00031     int getDerrotas(std::string jogo);
00032     int getEmpates(std::string jogo);
00033
00034     void mostrarEstatisticas(const std::string &nomeJogo) const;
00035 };
00036
00037 #endif
```

6.9 Referência do Arquivo include/JogoDaVelha.hpp

```
#include "Jogos.hpp"
```

Componentes

- class [JogoDaVelha](#)

6.10 JogoDaVelha.hpp

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

```
00001 #ifndef JOGODAVELHA_HPP
00002 #define JOGODAVELHA_HPP
00003
00004 #include "Jogos.hpp"
00005
00006 class JogoDaVelha : public Jogos
00007 {
00008 public:
00009     JogoDaVelha();
00010     JogoDaVelha(int tamanhoTabuleiro);
00011
00012 protected:
00013     void anunciarInicioPartida(Jogador &Jogador1, Jogador &Jogador2, bool &turno) override;
00014
00015     std::pair<int, int> lerJogada() override;
00016 }
```

```

00017     bool  checarDiagonal(std::vector<std::pair<int, int> &jogadas);
00018     bool  checarColunas(std::vector<std::pair<int, int> &jogadas);
00019     bool  checarLinhas(std::vector<std::pair<int, int> &jogadas);
00020     bool  checarVencedor(std::vector<std::pair<int, int> &jogadas, Jogador &vencedor, Jogador
&perdedor) override;
00021     bool  checarEmpate(int numeroJogadas, Jogador &jogador_01, Jogador &jogador_02) override;
00022
00023     friend class JogoDaVelhaAi;
00024 };
00025
00026 #endif

```

6.11 Referência do Arquivo include/JogoDaVelhaAi.hpp

```

#include "JogoDaVelha.hpp"
#include "Jogos.hpp"
#include <iostream>
#include <vector>

```

Componentes

- class `JogoDaVelhaAi`

Variáveis

- const int `BOARD_SIZE` = 9
- const char `EMPTY` = ''
- const char `PLAYER_X` = 'X'
- const char `PLAYER_O` = 'O'

6.11.1 Variáveis

6.11.1.1 BOARD_SIZE

```
const int BOARD_SIZE = 9
```

6.11.1.2 EMPTY

```
const char EMPTY = ' '
```

6.11.1.3 PLAYER_O

```
const char PLAYER_O = 'O'
```

6.11.1.4 PLAYER_X

```
const char PLAYER_X = 'X'
```

6.12 JogoDaVelhaAi.hpp

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

```
00001 #ifndef JOGODAVELHAAI_HPP
00002 #define JOGODAVELHAAI_HPP
00003
00004 #include "JogoDaVelha.hpp"
00005 #include "Jogos.hpp"
00006 #include <iostream>
00007 #include <vector>
00008
00009 const int BOARD_SIZE = 9;
00010 const char EMPTY = ' ';
00011 const char PLAYER_X = 'X';
00012 const char PLAYER_O = 'O';
00013 class JogoDaVelhaAi
00014 {
00015     public:
00016         JogoDaVelhaAi();
00017         void playGame();
00018         void Jogar(Jogador &Jogador1, Jogador &Jogador2);
00019
00020     private:
00021         int MAX_DEPTH;
00022
00023         JogoDaVelha jogo;
00024         std::vector<char> board;
00025
00026         bool checkWin(char player) const;
00027         bool isBoardFull() const;
00028         int minimax(bool isMaximizing, int depth);
00029         int getBestMove();
00030
00031         std::pair<int, int> humanMove(bool turno);
00032         std::pair<int, int> aiMove(bool turno);
00033 };
00034
00035 #endif
```

6.13 Referência do Arquivo include/Jogos.hpp

```
#include <algorithm>
#include <ctime>
#include <fstream>
#include <iostream>
#include <limits>
#include <random>
#include <string>
#include <utility>
#include <vector>
#include "Jogador.hpp"
```

Componentes

- class [Jogos](#)

6.14 Jogos.hpp

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

```
00001 #ifndef JOGOS_HPP
00002 #define JOGOS_HPP
00003
00004 #include <algorithm>
```

```

00005 #include <ctime>
00006 #include <fstream>
00007 #include <iostream>
00008 #include <limits>
00009 #include <random>
00010 #include <string>
00011 #include <utility>
00012 #include <vector>
00013
00014 #include "Jogador.hpp"
00015
00016 class Jogos
00017 {
00018     protected:
00019         std::vector<std::vector<char> > tabuleiro;
00020
00021         virtual void marcarTabuleiro(std::pair<int, int> &jogada, bool &turno);
00022         virtual void marcarTabuleiro(std::pair<int, int> &jogada, bool &turno,
00023                                     std::vector<std::vector<char> > &tabuleiroJogador,
00024                                     std::vector<std::pair<int, int> > &barcosJogador);
00025         virtual void limparTabuleiro();
00026         virtual void anunciarInicioPartida(Jogador &Jogador1, Jogador &Jogador2, bool &turno) = 0;
00027
00028         void anunciarTurnoJogador(Jogador &Jogador);
00029
00030         virtual bool sorteioTurno();
00031         virtual bool checarJogadaExistente(std::vector<std::pair<int, int> > &jogadas, int linha, int
00032     coluna);
00033         virtual bool checarPosicaoValida(int linha, int coluna);
00034
00035         virtual bool checarVencedor(std::vector<std::pair<int, int> > &jogadas, Jogador &vencedor, Jogador
00036     &perdedor) = 0;
00037         virtual bool checarEmpate(int numeroJogadas, Jogador &jogador_01, Jogador &jogador_02) = 0;
00038
00039         std::string gerarDivisoriaTabuleiro();
00040
00041         virtual std::pair<int, int> lerJogada() { return {0, 0}; };
00042
00043     public:
00044         virtual void mostrarTabuleiro();
00045         virtual void mostrarTabuleiro(const std::vector<std::vector<char> > &tabuleiro);
00046         virtual void Jogar(Jogador &Jogador1, Jogador &Jogador2);
00047 };
00048 #endif

```

6.15 Referência do Arquivo include/Lig4.hpp

```
#include "Jogos.hpp"
```

Componentes

- class [Lig4](#)

6.16 Lig4.hpp

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

```

00001 #ifndef LIG4_HPP
00002 #define LIG4_HPP
00003
00004 #include "Jogos.hpp"
00005
00006 class Lig4 : public Jogos
00007 {
00008     public:
00009         Lig4(int tamanhoTabuleiro);
00010         Lig4();
00011
00012     protected:
00013         void anunciarInicioPartida(Jogador &Jogador1, Jogador &Jogador2, bool &turno) override;

```

```

00014
00015     bool  checarDiagonal(std::vector<std::pair<int, int> &jogadas);
00016     bool  checarColunas(std::vector<std::pair<int, int> &jogadas);
00017     bool  checarLinhas(std::vector<std::pair<int, int> &jogadas);
00018     bool  checarVencedor(std::vector<std::pair<int, int> &jogadas, Jogador &vencedor, Jogador
&perdedor) override;
00019     bool  checarEmpate(int numeroJogadas, Jogador &jogador_01, Jogador &jogador_02) override;
00020
00021     std::pair<int, int> lerJogada() override;
00022 };
00023
00024 #endif

```

6.17 Referência do Arquivo include/Reversi.hpp

```
#include "Jogos.hpp"
```

Componentes

- class [Reversi](#)

6.18 Reversi.hpp

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

```

00001 #ifndef REVERSI_HPP
00002 #define REVERSI_HPP
00003
00004 #include "Jogos.hpp"
00005 class Reversi : public Jogos {
00006     public:
00007         Reversi();
00008         Reversi(int tamanhoTabuleiro);
00009         void Jogar(Jogador &Jogador1, Jogador &Jogador2) override;
00010     protected:
00011         void anunciarInicioPartida(Jogador &Jogador1, Jogador &Jogador2, bool &turno) override;
00012         bool jogadorInicial(bool &turno);
00013
00014         std::pair<int, int> lerJogada(bool turno);
00015
00016         void marcarTabuleiro(std::pair<int, int> &jogada, bool &turno) override;
00017         bool movimentoValido(std::pair<int, int> &jogada, char jogador, std::vector<std::pair<int, int>
&flips);
00018         bool haMovimentosDisponiveis(char Jogador);
00019
00020         void limparTabuleiro() override;
00021         bool checarVencedor(std::vector<std::pair<int, int> &jogadas, Jogador &vencedor, Jogador &perdedor)
{ return false;};
00022         bool checarVencedor(std::vector<std::pair<int, int> &jogadas, Jogador &jogador1, Jogador &Jogador2,
bool &PrimeiroJogador);
00023         bool checarEmpate(int numeroJogadas, Jogador &jogador_01, Jogador &jogador_02) override { return
false; };
00024 };
00025
00026 #endif

```

6.19 Referência do Arquivo src/BatalhaNaval.cpp

```
#include "BatalhaNaval.hpp"
```


Funções

- void `inserirBarcos` (std::vector< std::pair< int, int > > &barcosJogador, char tipo, int linhaInicial, int colunaInicial, int linhaFinal, int colunaFinal)

6.19.1 Funções

6.19.1.1 inserirBarcos()

```
void inserirBarcos (
    std::vector< std::pair< int, int > > & barcosJogador,
    char tipo,
    int linhaInicial,
    int colunaInicial,
    int linhaFinal,
    int colunaFinal)
```

6.20 Referência do Arquivo src/CentralDeJogos.cpp

```
#include <CentralDeJogos.hpp>
```

6.21 Referência do Arquivo src/Estatisticas.cpp

```
#include "Estatisticas.hpp"
#include <iostream>
```

6.22 Referência do Arquivo src/Jogador.cpp

```
#include "Jogador.hpp"
#include <iostream>
```

6.23 Referência do Arquivo src/JogoDaVelha.cpp

```
#include "JogoDaVelha.hpp"
```

6.24 Referência do Arquivo src/JogoDaVelhaAi.cpp

```
#include "JogoDaVelhaAi.hpp"
```

6.25 Referência do Arquivo src/Jogos.cpp

```
#include "Jogos.hpp"  
#include "Jogador.hpp"
```

6.26 Referência do Arquivo src/Lig4.cpp

```
#include "Lig4.hpp"
```

6.27 Referência do Arquivo src/main.cpp

```
#include <CentralDeJogos.hpp>  
#include <iostream>  
#include <stdexcept>  
#include <vector>
```

Funções

- std::string `validarEntrada()`
- void `exibirMenu()`
- int `main()`

6.27.1 Funções

6.27.1.1 `exibirMenu()`

```
void exibirMenu ()
```

6.27.1.2 `main()`

```
int main ()
```

6.27.1.3 `validarEntrada()`

```
std::string validarEntrada ()
```

6.28 Referência do Arquivo src/Reversi.cpp

```
#include "Reversi.hpp"
```

6.29 Referência do Arquivo UI/README.md

Índice Remissivo

- ~CentralDeJogos
 - CentralDeJogos, [13](#)
- anunciarInicioPartida
 - BatalhaNaval, [10](#)
 - JogoDaVelha, [20](#)
 - Jogos, [23](#)
 - Lig4, [27](#)
 - Reversi, [29](#)
- anunciarTurnoJogador
 - Jogos, [23](#)
- BatalhaNaval, [9](#)
 - anunciarInicioPartida, [10](#)
 - BatalhaNaval, [10](#)
 - checarEmpate, [10](#)
 - checarPosicaoValida, [10](#)
 - checarVencedor, [10](#), [11](#)
 - inserirBarcos, [11](#)
 - Jogar, [11](#)
 - lerBarcos, [11](#)
 - lerJogada, [11](#)
 - marcarTabuleiro, [11](#)
 - quantidadeBarcosDisponiveis, [12](#)
 - verificarEntrada, [12](#)
 - verificarTamanhoDoBarco, [12](#)
- BatalhaNaval.cpp
 - inserirBarcos, [41](#)
- BOARD_SIZE
 - JogoDaVelhaAi.hpp, [37](#)
- buscarJogador
 - CentralDeJogos, [13](#)
- cadastrarJogador
 - CentralDeJogos, [13](#)
- CentralDeJogos, [12](#)
 - ~CentralDeJogos, [13](#)
 - buscarJogador, [13](#)
 - cadastrarJogador, [13](#)
 - CentralDeJogos, [13](#)
 - executarPartida, [13](#)
 - listarJogadores, [13](#)
 - ordenarJogadores, [13](#)
 - removerJogador, [14](#)
 - validarEntrada, [14](#)
- checarColunas
 - JogoDaVelha, [20](#)
 - Lig4, [27](#)
- checarDiagonal
 - JogoDaVelha, [20](#)
- Lig4, [27](#)
- checarEmpate
 - BatalhaNaval, [10](#)
 - JogoDaVelha, [21](#)
 - Jogos, [23](#)
 - Lig4, [27](#)
 - Reversi, [29](#)
- checarJogadaExistente
 - Jogos, [23](#)
- checarLinhas
 - JogoDaVelha, [21](#)
 - Lig4, [27](#)
- checarPosicaoValida
 - BatalhaNaval, [10](#)
 - Jogos, [24](#)
- checarVencedor
 - BatalhaNaval, [10](#), [11](#)
 - JogoDaVelha, [21](#)
 - Jogos, [24](#)
 - Lig4, [27](#)
 - Reversi, [30](#)
- EMPTY
 - JogoDaVelhaAi.hpp, [37](#)
- Estatisticas, [14](#)
 - Estatisticas, [14](#)
 - getDerrotas, [15](#)
 - getEmpates, [15](#)
 - getHistorico, [15](#)
 - getVitorias, [15](#)
 - mostrarEstatisticas, [15](#)
 - registrarDerrota, [15](#)
 - registrarEmpate, [15](#)
 - registrarVitoria, [15](#)
- ExcecaoPosicionamentodeBarco, [16](#)
 - what, [16](#)
- ExcecaoTipodeBarcoInvalido, [16](#)
 - what, [17](#)
- executarPartida
 - CentralDeJogos, [13](#)
- exibirMenu
 - main.cpp, [42](#)
- gerarDivisoriaTabuleiro
 - Jogos, [24](#)
- getApelido
 - Jogador, [18](#)
- getDerrotas
 - Estatisticas, [15](#)
 - Jogador, [18](#)

- getEmpates
 - Estatísticas, 15
 - Jogador, 18
- getHistorico
 - Estatísticas, 15
- getNome
 - Jogador, 18
- getVitorias
 - Estatísticas, 15
 - Jogador, 18
- haMovimentosDisponiveis
 - Reversi, 30
- include/BatalhaNaval.hpp, 33
- include/CentralDeJogos.hpp, 34
- include/Estatísticas.hpp, 35
- include/Jogador.hpp, 35, 36
- include/JogoDaVelha.hpp, 36
- include/JogoDaVelhaAi.hpp, 37, 38
- include/Jogos.hpp, 38
- include/Lig4.hpp, 39
- include/Reversi.hpp, 40
- inserirBarcos
 - BatalhaNaval, 11
 - BatalhaNaval.cpp, 41
- Jogador, 17
 - getApelido, 18
 - getDerrotas, 18
 - getEmpates, 18
 - getNome, 18
 - getVitorias, 18
 - Jogador, 17
 - mostrarEstatísticas, 18
 - registrarDerrota, 18
 - registrarEmpate, 18
 - registrarVitoria, 18
- jogadorInicial
 - Reversi, 30
- Jogar
 - BatalhaNaval, 11
 - JogoDaVelhaAi, 22
 - Jogos, 24
 - Reversi, 30
- JogoDaVelha, 19
 - anunciarInicioPartida, 20
 - checarColunas, 20
 - checarDiagonal, 20
 - checarEmpate, 21
 - checarLinhas, 21
 - checarVencedor, 21
 - JogoDaVelha, 20
 - JogoDaVelhaAi, 21
 - lerJogada, 21
- JogoDaVelhaAi, 22
 - Jogar, 22
 - JogoDaVelha, 21
 - JogoDaVelhaAi, 22
 - playGame, 22
- JogoDaVelhaAi.hpp
 - BOARD_SIZE, 37
 - EMPTY, 37
 - PLAYER_O, 37
 - PLAYER_X, 37
- Jogos, 22
 - anunciarInicioPartida, 23
 - anunciarTurnoJogador, 23
 - checarEmpate, 23
 - checarJogadaExistente, 23
 - checarPosicaoValida, 24
 - checarVencedor, 24
 - gerarDivisoriaTabuleiro, 24
 - Jogar, 24
 - lerJogada, 24
 - limparTabuleiro, 24
 - marcarTabuleiro, 24, 25
 - mostrarTabuleiro, 25
 - sorteioTurno, 25
 - tabuleiro, 25
- lerBarcos
 - BatalhaNaval, 11
- lerJogada
 - BatalhaNaval, 11
 - JogoDaVelha, 21
 - Jogos, 24
 - Lig4, 28
 - Reversi, 30
- Lig4, 26
 - anunciarInicioPartida, 27
 - checarColunas, 27
 - checarDiagonal, 27
 - checarEmpate, 27
 - checarLinhas, 27
 - checarVencedor, 27
 - lerJogada, 28
 - Lig4, 27
- limparTabuleiro
 - Jogos, 24
 - Reversi, 30
- listarJogadores
 - CentralDeJogos, 13
- main
 - main.cpp, 42
- main.cpp
 - exibirMenu, 42
 - main, 42
 - validarEntrada, 42
- marcarTabuleiro
 - BatalhaNaval, 11
 - Jogos, 24, 25
 - Reversi, 31
- mostrarEstatísticas
 - Estatísticas, 15
 - Jogador, 18
- mostrarTabuleiro

- Jogos, [25](#)
- movimentoValido
 - Reversi, [31](#)
- ordenarJogadores
 - CentralDeJogos, [13](#)
- PLAYER_O
 - JogoDaVelhaAi.hpp, [37](#)
- PLAYER_X
 - JogoDaVelhaAi.hpp, [37](#)
- playGame
 - JogoDaVelhaAi, [22](#)
- quantidadeBarcosDisponiveis
 - BatalhaNaval, [12](#)
- registrarDerrota
 - Estatisticas, [15](#)
 - Jogador, [18](#)
- registrarEmpate
 - Estatisticas, [15](#)
 - Jogador, [18](#)
- registrarVitoria
 - Estatisticas, [15](#)
 - Jogador, [18](#)
- removerJogador
 - CentralDeJogos, [14](#)
- Reversi, [28](#)
 - anunciarInicioPartida, [29](#)
 - checarEmpate, [29](#)
 - checarVencedor, [30](#)
 - haMovimentosDisponiveis, [30](#)
 - jogadorInicial, [30](#)
 - Jogar, [30](#)
 - lerJogada, [30](#)
 - limparTabuleiro, [30](#)
 - marcarTabuleiro, [31](#)
 - movimentoValido, [31](#)
 - Reversi, [29](#)
- sorteioTurno
 - Jogos, [25](#)
- src/BatalhaNaval.cpp, [40](#)
- src/CentralDeJogos.cpp, [41](#)
- src/Estatisticas.cpp, [41](#)
- src/Jogador.cpp, [41](#)
- src/JogoDaVelha.cpp, [41](#)
- src/JogoDaVelhaAi.cpp, [41](#)
- src/Jogos.cpp, [42](#)
- src/Lig4.cpp, [42](#)
- src/main.cpp, [42](#)
- src/Reversi.cpp, [42](#)
- tabuleiro
 - Jogos, [25](#)
- UI, [1](#)
- UI/README.md, [42](#)
- validarEntrada
 - CentralDeJogos, [14](#)
 - main.cpp, [42](#)
- verificarEntrada
 - BatalhaNaval, [12](#)
- verificarTamanhoDoBarco
 - BatalhaNaval, [12](#)
- what
 - ExcecaoPosicionamentodeBarco, [16](#)
 - ExcecaoTipodeBarcoInvalido, [17](#)