

Relatório

# **Casos de Teste**

Projeto Chess

Niterói, RJ  
Julho de 2022

## Testes Unitários:

### I. Board

#### **test\_chessNotation**

Testa se a função **Board.chessNotation** retorna a posição correspondente do tabuleiro, informada como um par de inteiros, em notação convencional de xadrez.

Estado do teste:

Passou	SIM
Falhou	

#### **test\_arrayNotation**

Testa se a função **Board.arrayNotation** retorna a posição correspondente do tabuleiro, informada em notação convencional de xadrez, como um par de inteiros.

Estado do teste:

Passou	SIM
Falhou	

#### **test\_getKingTile**

Testa se as funções **Board.getBlackKingTile** e **Board.getWhiteKingTile** retornam a posição correspondente ao rei preto e branco, respectivamente.

Estado do teste:

Passou	SIM
Falhou	

#### **test\_isKingInCheck**

Testa se a função **Board.isKingInCheck** informa corretamente se o rei está em xeque.

Estado do teste:

Passou	SIM
Falhou	

### **test\_isKingInCheckmate**

Testa se a função **Board.isKingInCheckmate** informa corretamente se o rei está em xeque-mate.

Estado do teste:

Passou	SIM
Falhou	

### **test\_getSequencesPawnBlack**

Testa se a função **Board.getSequencesPawnBlack** informa corretamente os movimentos válidos de um peão do time preto.

Estado do teste:

Passou	SIM
Falhou	

### **test\_getSequencesPawnwhite**

Testa se a função **Board.getSequencesPawnWhite** informa corretamente os movimentos válidos de um peão do time branco.

Estado do teste:

Passou	SIM
Falhou	

### **test\_getSequencesRook**

Testa se a função **Board.getSequencesRook** informa corretamente os movimentos válidos de uma torre.

Estado do teste:

Passou	SIM
Falhou	

### **test\_getSequencesKnight**

Testa se a função **Board.getSequencesKnight** informa corretamente os movimentos válidos de um cavalo.

Estado do teste:

Passou	SIM
Falhou	

### **test\_getSequencesBishop**

Testa se a função **Board.getSequencesBishop** informa corretamente os movimentos válidos de um bispo.

Estado do teste:

Passou	SIM
Falhou	

### **test\_getSequencesQueen**

Testa se a função **Board.getSequencesQueen** informa corretamente os movimentos válidos de uma rainha.

Estado do teste:

Passou	SIM
Falhou	

### **test\_getSequencesKing**

Testa se a função **Board.getSequencesKing** informa corretamente os movimentos válidos de um rei.

Estado do teste:

Passou	SIM
Falhou	

### **test\_getSequencesKing**

Testa se a função **Board.getSequencesKing** informa corretamente os movimentos válidos de um rei.

Estado do teste:

Passou	SIM
Falhou	

### **test\_getSequencesKing**

Testa se a função **Board.getSequencesKing** informa corretamente os movimentos válidos de um rei.

Estado do teste:

Passou	SIM
Falhou	

### test\_getSequencesKing

Testa se a função **Board.getSequencesKing** informa corretamente os movimentos válidos de um rei.

Estado do teste:

Passou	SIM
Falhou	

### test\_getCastling

Testa se a função **Board.getCastling** informa corretamente os movimentos válidos de roque.

Estado do teste:

Passou	SIM
Falhou	

### Evidência de testes:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

(venv) PS C:\Users\karin\OneDrive\Documentos\chess> python -m unittest tests/test_board.py
.....
-----
Ran 13 tests in 0.040s

OK
(venv) PS C:\Users\karin\OneDrive\Documentos\chess> █
```

## Testes de Integração:

### I. Serviços

#### test\_start\_game

Testa se a função **Services.start\_game** informa corretamente o tabuleiro inicial bem como atribui os valores corretos para cada um dos atributos da classe **Services**.

Estado do teste:

Passou	SIM
Falhou	

#### test\_get\_legal\_moves

Testa se a função **Services.get\_legal\_moves** informa corretamente as movimentações possíveis no formato desejado para o frontend.

Estado do teste:

Passou	SIM
Falhou	

#### **test\_execute\_move**

Testa se a função **Services.execute\_move** atualiza corretamente o objeto Board da classe Services, além de retornar as posições de todas as peças para atualizar no front.

Estado do teste:

Passou	SIM
Falhou	

#### **test\_legalSequences**

Testa se a função **Services.legalSequences** identifica a possibilidade do jogador específico ter alguma movimentação válida.

Estado do teste:

Passou	SIM
Falhou	

#### **test\_draw**

Testa se a função **Services.draw** identifica corretamente os casos de empates.

Estado do teste:

Passou	SIM
Falhou	

#### **Evidência de testes:**

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

(venv) PS C:\Users\karin\OneDrive\Documentos\chess> python -m unittest tests/test_services.py
.....
-----
Ran 5 tests in 0.163s

OK
(venv) PS C:\Users\karin\OneDrive\Documentos\chess> 
```

## II. IA

### `test_rand`

Testa se a função **rand** gera um movimento aleatório válido com a cor correta

Estado do teste:

Passou	SIM
Falhou	

### `test_min_max`

Testa se a função **min\_max** gera um movimento que captura a peça inimiga ocom a melhor pontuação bem, como se a pontuação gerada estava correta

Estado do teste:

Passou	SIM
Falhou	

### `test_in_min_max`

Testa se a função **in\_min\_max** gera a pontuação correta após capturar uma peça.

Estado do teste:

Passou	SIM
Falhou	

### `test_alpha_beta`

Testa se a função **min\_max** gera um movimento que captura a peça inimiga com a melhor pontuação,em uma situação mais complexa que a min\_max.

Estado do teste:

Passou	SIM
Falhou	

### Evidência de testes:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

(venv) PS C:\Users\karin\OneDrive\Documentos\chess> python -m unittest tests/test_ia.py
....
-----
Ran 4 tests in 0.229s

OK
(venv) PS C:\Users\karin\OneDrive\Documentos\chess> 
```

## Teste de Sistema:

test\_routeToEasyGame => test\_routeToEasyGame

Estado do teste:

Passou	SIM
Falhou	

test\_routeToHardGame => Acessar jogo de nível difícil

Estado do teste:

Passou	SIM
Falhou	

test\_routeToLevels => Acessar página de níveis do jogo

Estado do teste:

Passou	SIM
Falhou	

test\_routeToRules => Acessar página de regras do jogo

Estado do teste:

Passou	SIM
Falhou	

test\_routeToAbout => Acessar página Sobre

Estado do teste:

Passou	SIM
Falhou	

test\_routeToIaFight => Acessar página de briga de IA's

Estado do teste:

Passou	SIM
Falhou	



test\_backToMenuFromEasyGame => Retorno ao menu dentro de um jogo fácil

Estado do teste:

Passou	SIM
Falhou	

test\_backToMenuFromHardGame => Retorno ao menu dentro de um jogo difícil

Estado do teste:

Passou	SIM
Falhou	

### Evidência de testes:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

(venv) PS C:\Users\karin\OneDrive\Documentos\chess> python -m unittest tests/test_ia.py
....
-----
Ran 4 tests in 0.229s

OK
(venv) PS C:\Users\karin\OneDrive\Documentos\chess> 
```