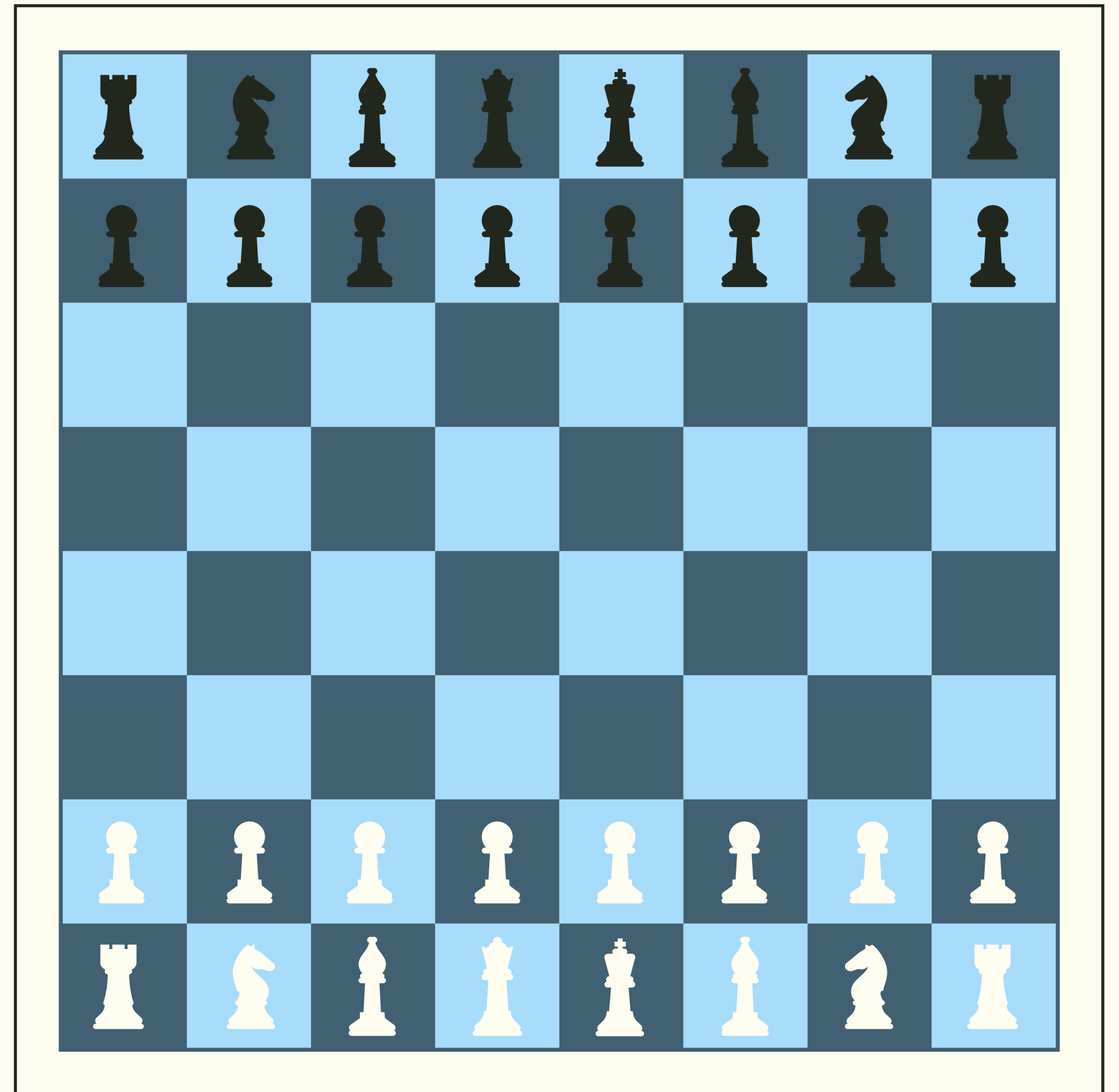
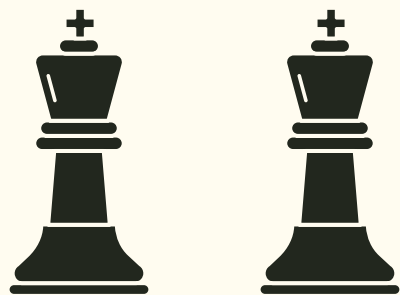


PROJETO CHESS GAME

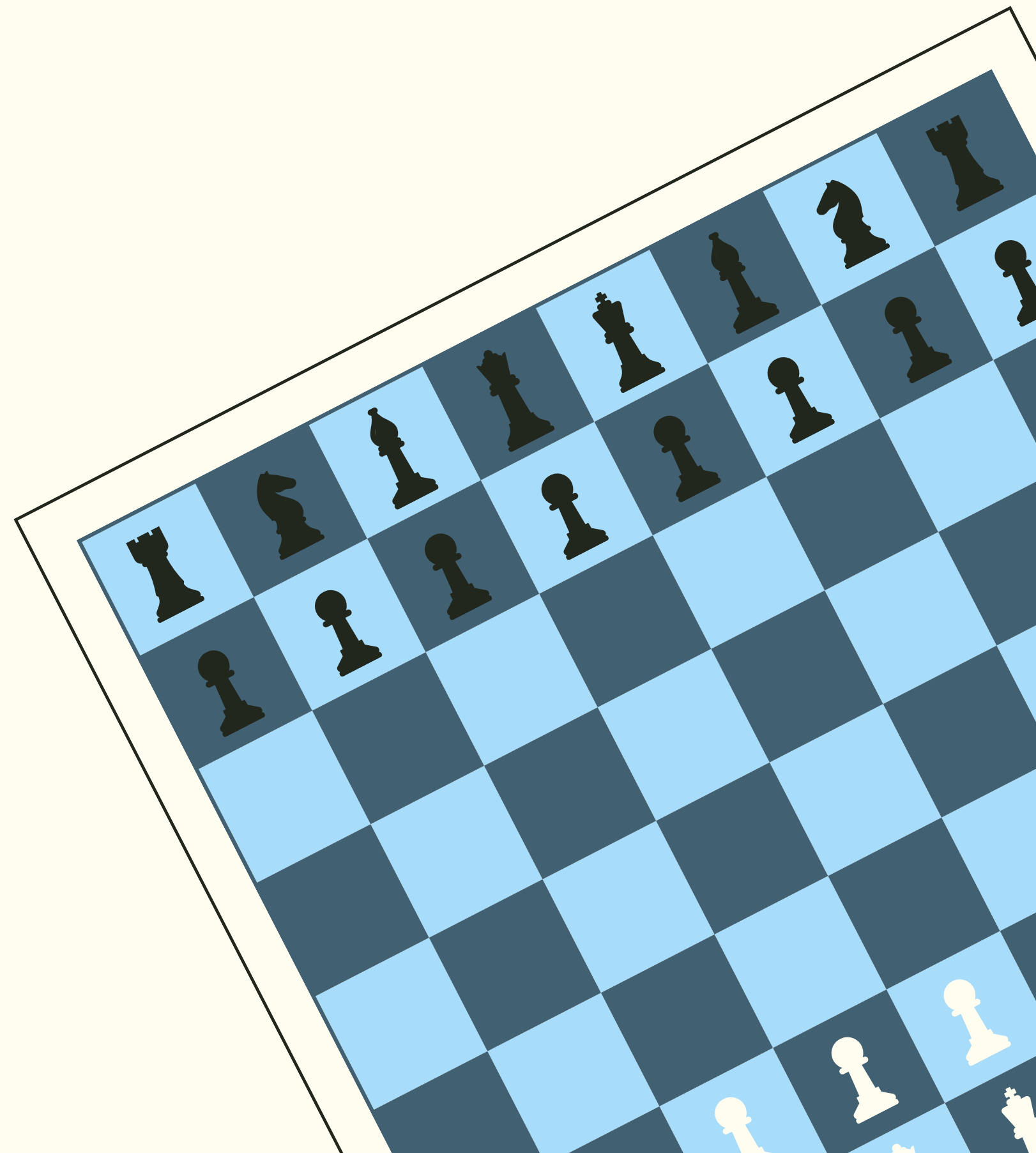
Grupo: Gabriela Pinheiro
Karina Lemos
Thiago Nascimento
Tiago Lacerda
Valesca Moura

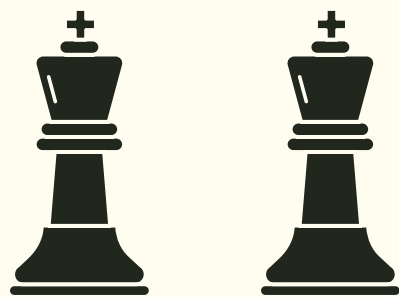




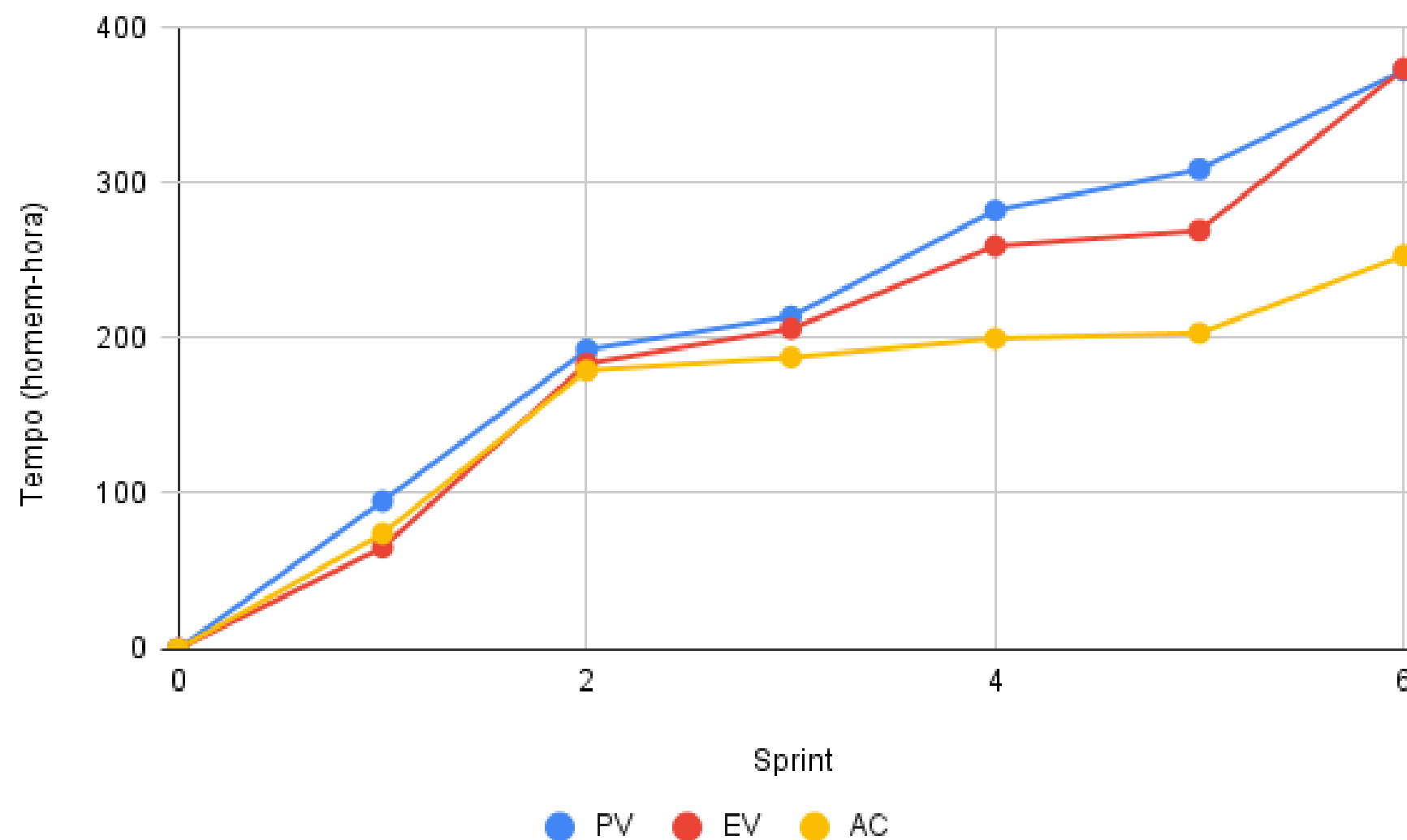
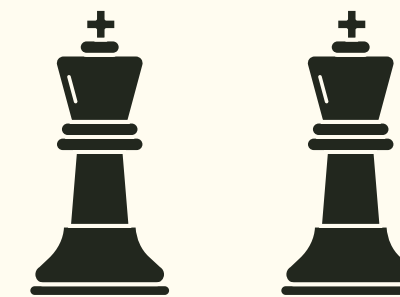
OVERVIEW

- **Aplicação web**
 - Django
 - Deploy no Heroku
- **Backend em python**
- **Frontend em javascript**
- **IA Alpha-beta**
- **Unittest**



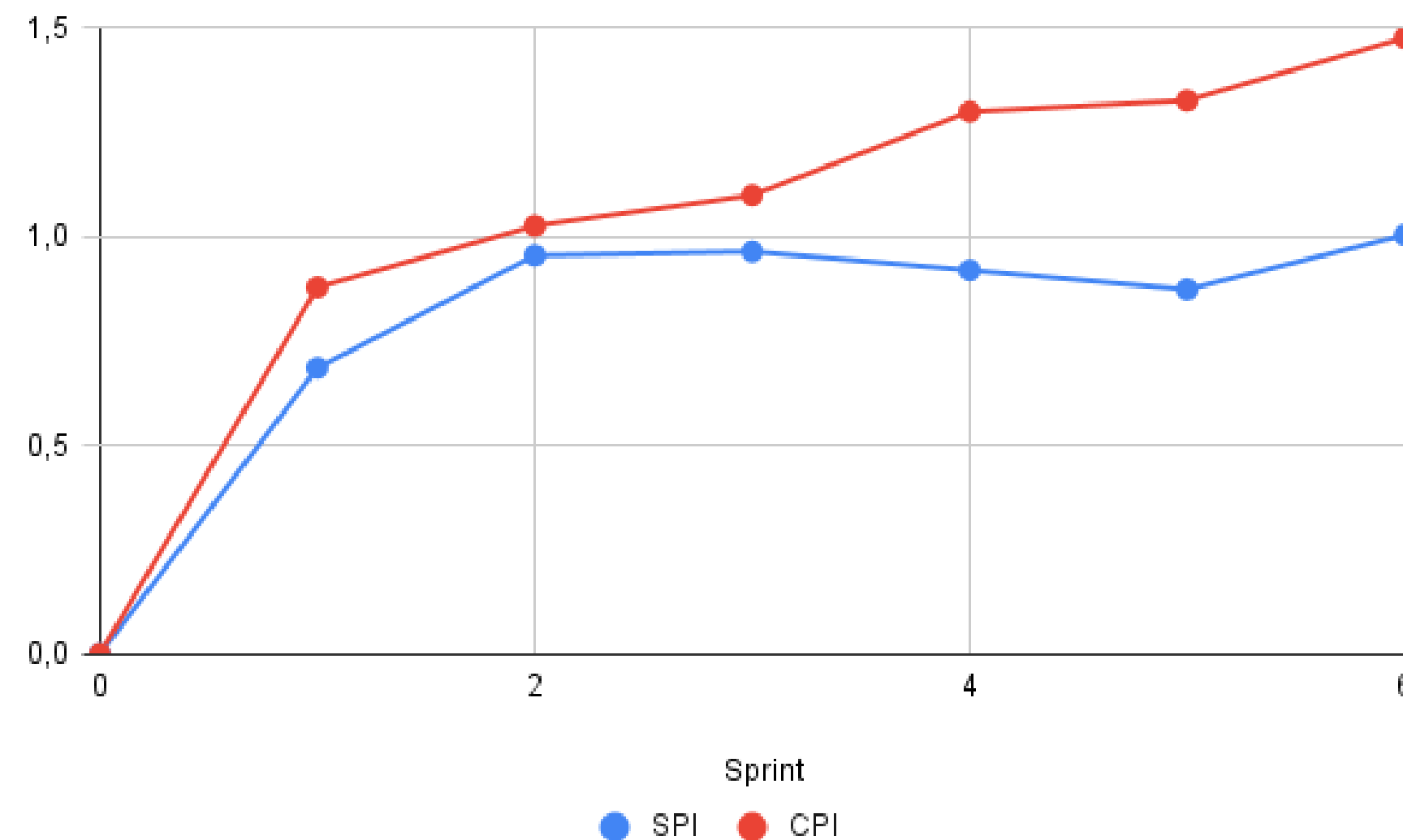


MONITORAMENTO E CONTROLE DO PROJETO



Resultados finais:

- PV: 372,35 homem-hora
- EV: 373,35 homem-hora
- AC: 253,17 homem-hora

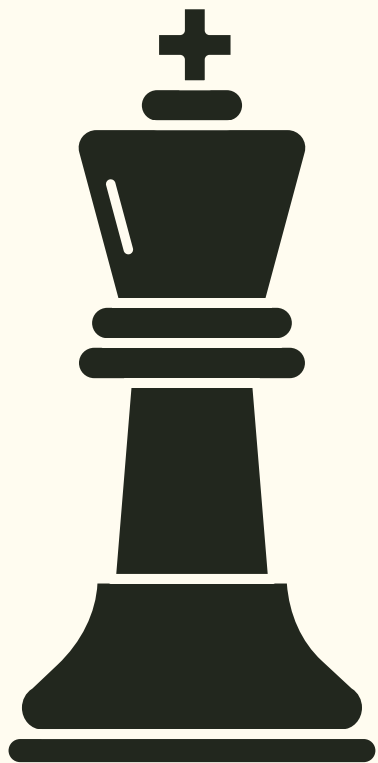


Resultados finais:

- SPI: 1,003
- CPI: 1,475

PLANO DE TESTES

- Escopo
- Objetivos de Qualidade
- Papéis e Responsabilidades
- Fases de Teste
- Critérios de Suspensão e Requisitos de Retomada
- Completude do Teste
- Estimativas de cronograma
- Entregáveis de Teste
- Ferramentas de Teste
- Ferramenta de cobertura de testes
- Ferramenta de Teste de performance



CASOS DE TESTES

- Relatório

Relatorio_CasosdeTeste.pdf

Arquivo | C:/Users/karin/Downloads/Relatorio_CasosdeTeste.pdf

6 de 9

Exibição de página | Ler em voz alta | Adicionar texto | Desenhar | Realçar | Apagar

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>



TESTES UNITÁRIOS

Classe board:

- 13 Casos de Teste

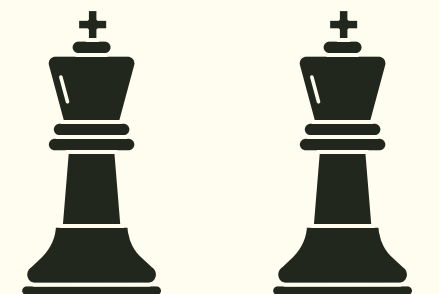
- test_chessNotation
- test_arrayNotation
- test_getKingTile
- test_isKingInCheck
- test_isKingInCheckmate
- test_getSequencesPawnBlack
- test_getSequencesPawnwhite
- test_getSequencesRook
- test_getSequencesKnight
- test_getSequencesBishop
- test_getSequencesQueen
- test_getSequencesKing
- test_getCastling

- Evidência dos Teste:

```
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 0.048s

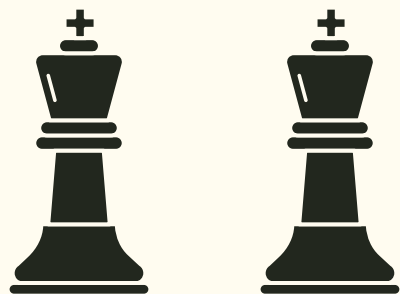
OK
(venv) PS C:\Users\karin\OneDrive\Documentos\chess> 
```



TESTES DE INTEGRAÇÃO

Classe Services

- 5 Casos de Teste
 - test_start_game
 - test_get_legal_moves
 - test_execute_move
 - test_legalSequences
 - test_draw



- Evidência de Teste:

```
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 0.145s

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

TESTES USANDO MOCK/PATCH

```
def get_legal_moves(self, piece):
    sequences = self.chess.getLegalSequences(piece)
    result = []
    for i in range(len(sequences)):
        sequences[i] = sequences[i][0].split(" ")
        result.append(sequences[i][2])

    return result
```

```
def test_get_legal_moves(self):

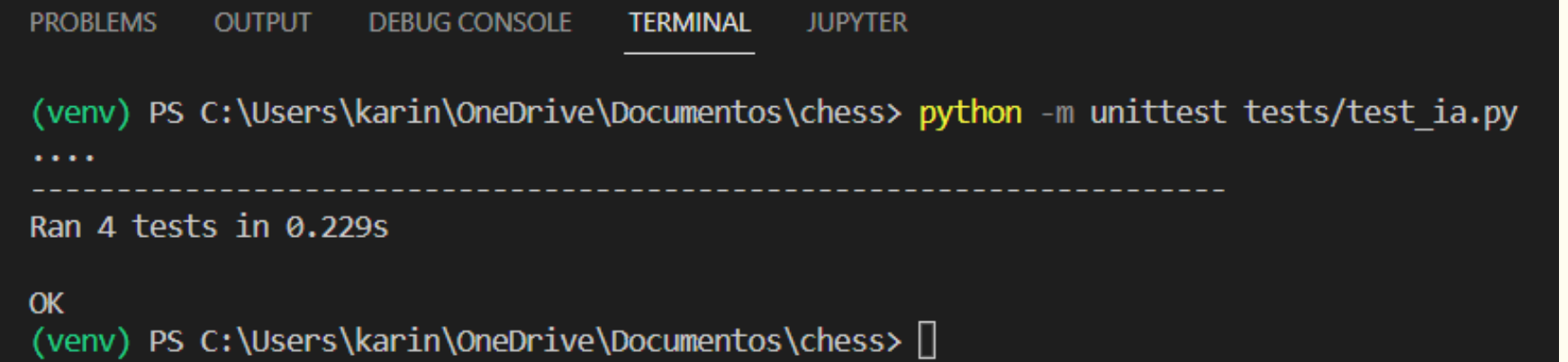
    # tabuleiro inicial
    teste = Services()
    with patch.object(Board, 'getLegalSequences', return_value=[['move a2 a3'], ['move a2 a4']]):
        assert teste.get_legal_moves("a2") == ["a3", "a4"]
```



ia.py

- **5 Casos de Teste**
 - test_rand
 - test_min_max
 - test_in_min_max
 - test_alpha_beta

• Evidência de Teste:



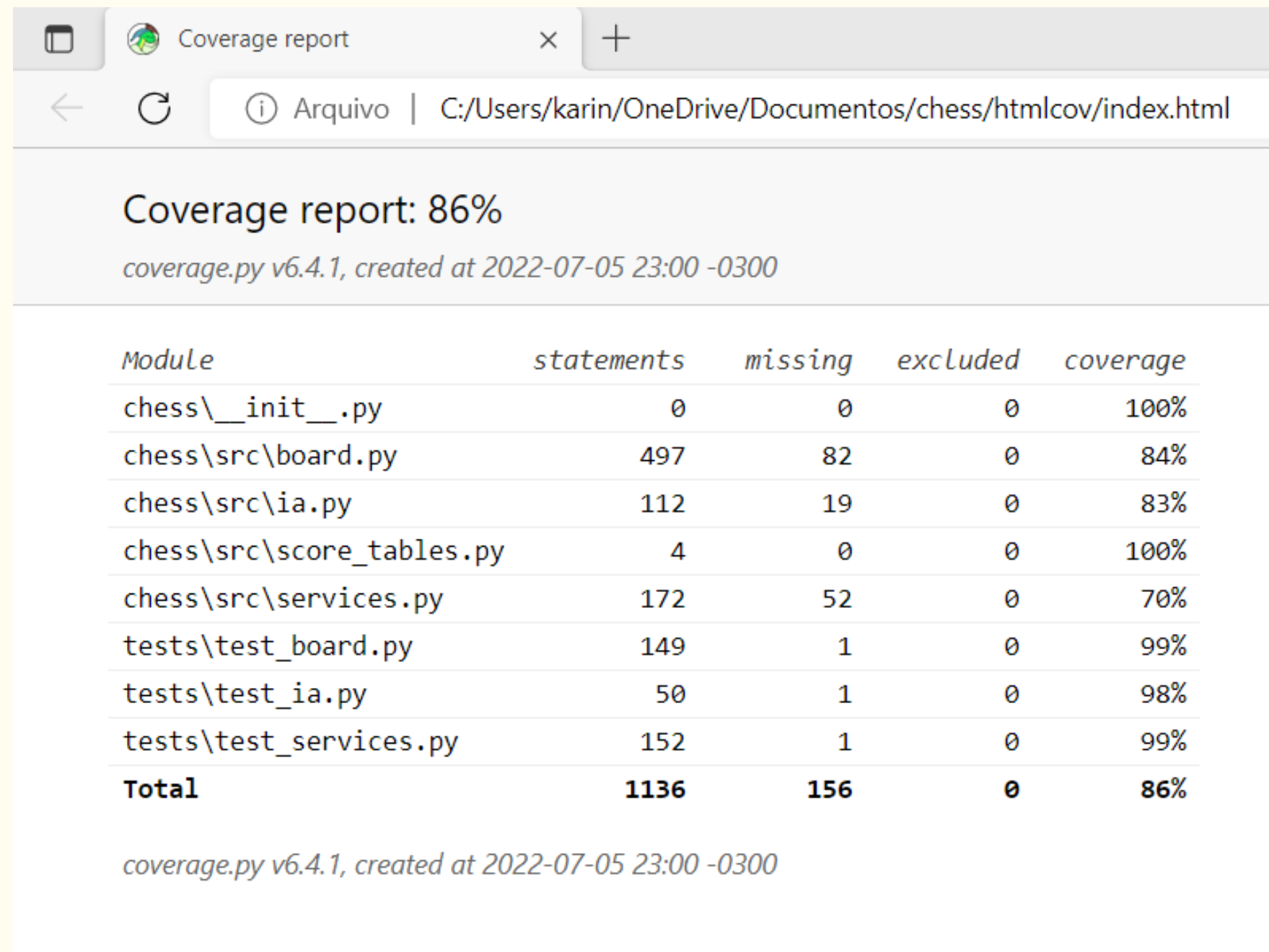
A screenshot of a terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and JUPYTER. The terminal shows the command `python -m unittest tests/test_ia.py` being executed in a PowerShell prompt. The output consists of four dots, a dashed line, and the message "Ran 4 tests in 0.229s". Below this, the word "OK" is displayed, followed by the prompt returning to the PowerShell shell.

```
(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> 
```

COBERTURA DOS TESTES

- Biblioteca coverage.py



The screenshot shows a web browser window with a single tab titled 'Coverage report'. The address bar shows the file path 'C:/Users/karin/OneDrive/Documentos/chess/htmlcov/index.html'. The main content area displays a 'Coverage report: 86%' with a subtitle 'coverage.py v6.4.1, created at 2022-07-05 23:00 -0300'. Below this is a table with 5 columns: 'Module', 'statements', 'missing', 'excluded', and 'coverage'. The table lists several modules including 'chess__init__.py', 'chess\src\board.py', 'chess\src\ia.py', 'chess\src\score_tables.py', 'chess\src\services.py', and test files 'tests\test_board.py', 'tests\test_ia.py', and 'tests\test_services.py'. A 'Total' row at the bottom summarizes the overall statistics: 1136 statements, 156 missing, 0 excluded, resulting in 86% coverage. The same version and creation information is repeated at the bottom of the table.

Module	statements	missing	excluded	coverage
chess__init__.py	0	0	0	100%
chess\src\board.py	497	82	0	84%
chess\src\ia.py	112	19	0	83%
chess\src\score_tables.py	4	0	0	100%
chess\src\services.py	172	52	0	70%
tests\test_board.py	149	1	0	99%
tests\test_ia.py	50	1	0	98%
tests\test_services.py	152	1	0	99%
Total	1136	156	0	86%

coverage.py v6.4.1, created at 2022-07-05 23:00 -0300

TESTES DE SISTEMA

- 8 Casos de Teste:

test_routeToEasyGame → Acessar jogo de nível fácil

test_routeToHardGame → Acessar jogo de nível difícil

test_routeToLevels → Acessar página de níveis do jogo

test_routeToRules → Acessar página de regras do jogo

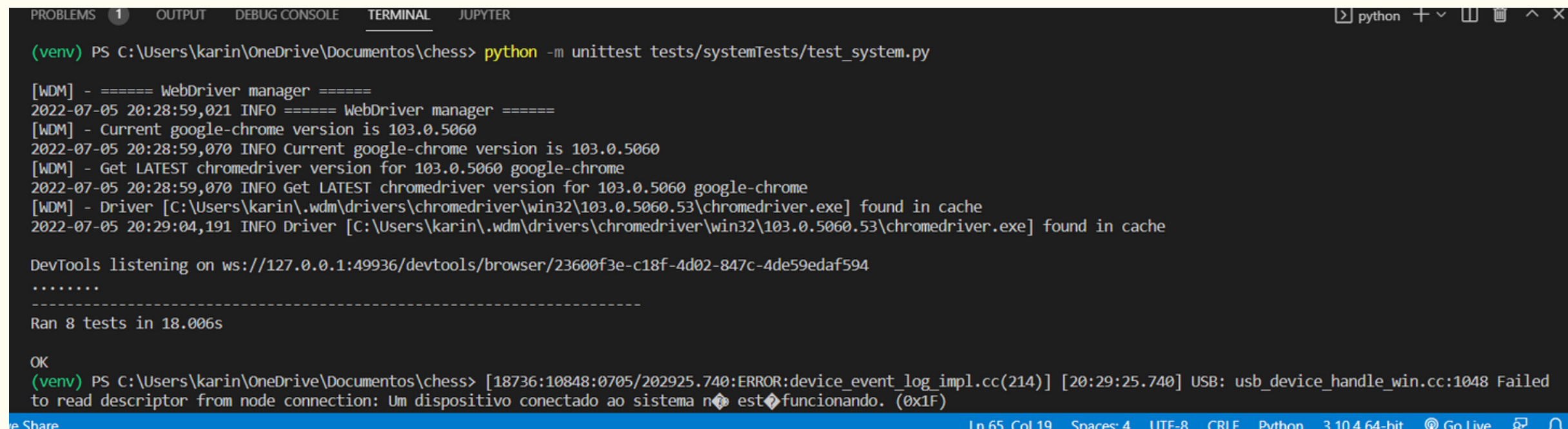
test_routeToAbout → Acessar página Sobre

test_routeToFight → Acessar página de briga de IA's

test_backToMenuFromEasyGame → Retorno ao menu dentro de um jogo fácil

test_backToMenuFromHardGame → Retorno ao menu dentro de um jogo difícil

- Evidência do teste:



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
(venv) PS C:\Users\karin\OneDrive\Documentos\chess> python -m unittest tests/systemTests/test_system.py

[WDM] - ===== WebDriver manager =====
2022-07-05 20:28:59,021 INFO ===== WebDriver manager =====
[WDM] - Current google-chrome version is 103.0.5060
2022-07-05 20:28:59,070 INFO Current google-chrome version is 103.0.5060
[WDM] - Get LATEST chromedriver version for 103.0.5060 google-chrome
2022-07-05 20:28:59,070 INFO Get LATEST chromedriver version for 103.0.5060 google-chrome
[WDM] - Driver [C:\Users\karin\wdm\drivers\chromedriver\win32\103.0.5060.53\chromedriver.exe] found in cache
2022-07-05 20:29:04,191 INFO Driver [C:\Users\karin\wdm\drivers\chromedriver\win32\103.0.5060.53\chromedriver.exe] found in cache

DevTools listening on ws://127.0.0.1:49936/devtools/browser/23600f3e-c18f-4d02-847c-4de59edaf594
.....
-----
Ran 8 tests in 18.006s

OK
(venv) PS C:\Users\karin\OneDrive\Documentos\chess> [18736:10848:0705/202925.740:ERROR:device_event_log_impl.cc(214)] [20:29:25.740] USB: usb_device_handle_win.cc:1048 Failed
to read descriptor from node connection: Um dispositivo conectado ao sistema n  est  funcionando. (0x1F)
```

TESTES DE PERFORMANCE

- Ferramenta: 

testeDePerformanceChessGame.jmx (C:\Users\karin\Downloads\apache-jmeter-5.5\apache-jmeter-5.5\bin\testeDePerformanceChessGame.jmx) - Apache JMeter (5.5)

Arquivo Editar Search Executar Opções Tools Ajuda

00:00:59 0 0/100000000

Plano de Teste

- Grupo de Usuários
 - Requisição HTTP
 - Ver Resultados em Tabela

Ver Resultados em Tabela

Nome: Ver Resultados em Tabela

Comentários:

Escrever resultados para arquivo / Ler a partir do arquivo

Nome do arquivo Procurar... Apenas Logar/Exibir Erros Sucessos Configurar

Amostra #	Tempo de início	Nome do Usuário...	Rótulo	Tempo da amostr...	Estado	Bytes	Sent Bytes	Latency	Connect Time(ms)
15828	00:36:54.531	Grupo de Usuári...	Requisição HTTP	34539	✓	1587	134	34539	135
15829	00:36:54.545	Grupo de Usuári...	Requisição HTTP	34525	✓	1587	134	34525	131
15830	00:36:54.389	Grupo de Usuári...	Requisição HTTP	34681	✓	1587	134	34681	132
15831	00:36:54.671	Grupo de Usuári...	Requisição HTTP	34400	✓	1587	134	34400	244
15832	00:36:54.584	Grupo de Usuári...	Requisição HTTP	34506	✓	1587	134	34506	362
15833	00:37:00.273	Grupo de Usuári...	Requisição HTTP	28820	✗	2556	0	0	159
15834	00:36:54.918	Grupo de Usuári...	Requisição HTTP	34177	✓	1587	134	34177	131
15835	00:36:56.544	Grupo de Usuári...	Requisição HTTP	32555	✗	2556	0	0	134
15836	00:36:54.634	Grupo de Usuári...	Requisição HTTP	34468	✓	1587	134	34468	298
15837	00:36:54.580	Grupo de Usuári...	Requisição HTTP	34522	✓	1587	134	34522	372
15838	00:36:54.587	Grupo de Usuári...	Requisição HTTP	34516	✓	1587	134	34516	364
15839	00:36:54.613	Grupo de Usuári...	Requisição HTTP	34491	✓	1587	134	34491	330
15840	00:36:57.364	Grupo de Usuári...	Requisição HTTP	31741	✗	2556	0	0	132
15841	00:36:54.609	Grupo de Usuári...	Requisição HTTP	34496	✓	1587	134	34496	335
15842	00:36:54.620	Grupo de Usuári...	Requisição HTTP	34488	✓	1587	134	34488	323
15843	00:36:54.536	Grupo de Usuári...	Requisição HTTP	34572	✓	1587	134	34572	414
15844	00:36:58.240	Grupo de Usuári...	Requisição HTTP	30873	✗	2556	0	0	134
15845	00:36:54.587	Grupo de Usuári...	Requisição HTTP	34527	✓	1587	134	34527	361
15846	00:36:54.621	Grupo de Usuári...	Requisição HTTP	34493	✓	1587	134	34493	317
15847	00:36:54.580	Grupo de Usuári...	Requisição HTTP	34537	✓	1587	134	34537	372
15848	00:36:54.586	Grupo de Usuári...	Requisição HTTP	34532	✓	1587	134	34532	366
15849	00:36:55.227	Grupo de Usuári...	Requisição HTTP	33892	✗	2556	0	0	134
15850	00:36:54.585	Grupo de Usuári...	Requisição HTTP	34537	✓	1587	134	34537	367

☐ Scroll automatically? ☐ Child samples?

Núm. de Amostras 18863 Última Amostra 29270 Média 17015 Desvio 15513

TESTES DE ACEITAÇÃO

Feedback recebido:

- Decisão de movimentação da IA;
- Priorização de tipo de movimentação da IA;
- Tempo de reposta.

DECISÕES TOMADAS

Aplicativo Mobile vs Aplicação WEB

Demanda de desempenho do ambiente de desenvolvimento

Experiências prévias do grupo com as plataformas

TBD - Trunk Based Development

Minimizar o esforço de merge e integração

Deploy automático

Revisão frequente do código pelos membros do grupo

DECISÕES TOMADAS

Heroku

Opção gratuita de hospedagem

Experiência prévia da equipe

Algoritmo de I.A. Min-Max

Extensamente abordada na literatura

Unittest vs Pytest

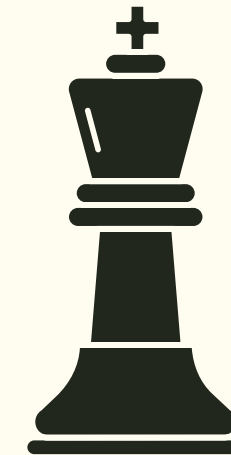
Interface simplificada

DIFICULDADES

- Criação da WBS;
- Desenvolvimento do gráfico de Gantt;
- Monitoramento e controle do projeto;
- Testes (função de mock);
- Integração com o Heroku.

DIVISÃO DE TAREFAS

- **Pelo menos um responsável primário e secundário para cada setor.**
 - **Back**
 - **Máquina de Regras: Tiago Lacerda**
 - **IA: Gabriela Pinheiro**
 - **Serviços: Thiago Lopes**
 - **Configuração Django e Controllers: Valesca**
 - **Front**
 - **UI/UX: Karina Lemos**
 - **Requisições jQuery e funções Javascript: Valesca**
 - **Testes:**
 - **Gerência: Karina Lemos**
 - **Testes unitários: Gabriela, Tiago Lacerda e Thiago Lopes**
 - **Gerência:**
 - **Grupo**
 - **Documentação:**
 - **Monitoramento e controle: Thiago Lopes**
 - **Demais: grupo.**
 - **Infraestrutura**
 - **GitHub actions e Heroku: Valesca Moura**
 - **Integração front-back: Valesca Moura**
 - **Integração ia-máquina de regras: Gabriela e Tiago Lacerda**



DEMO

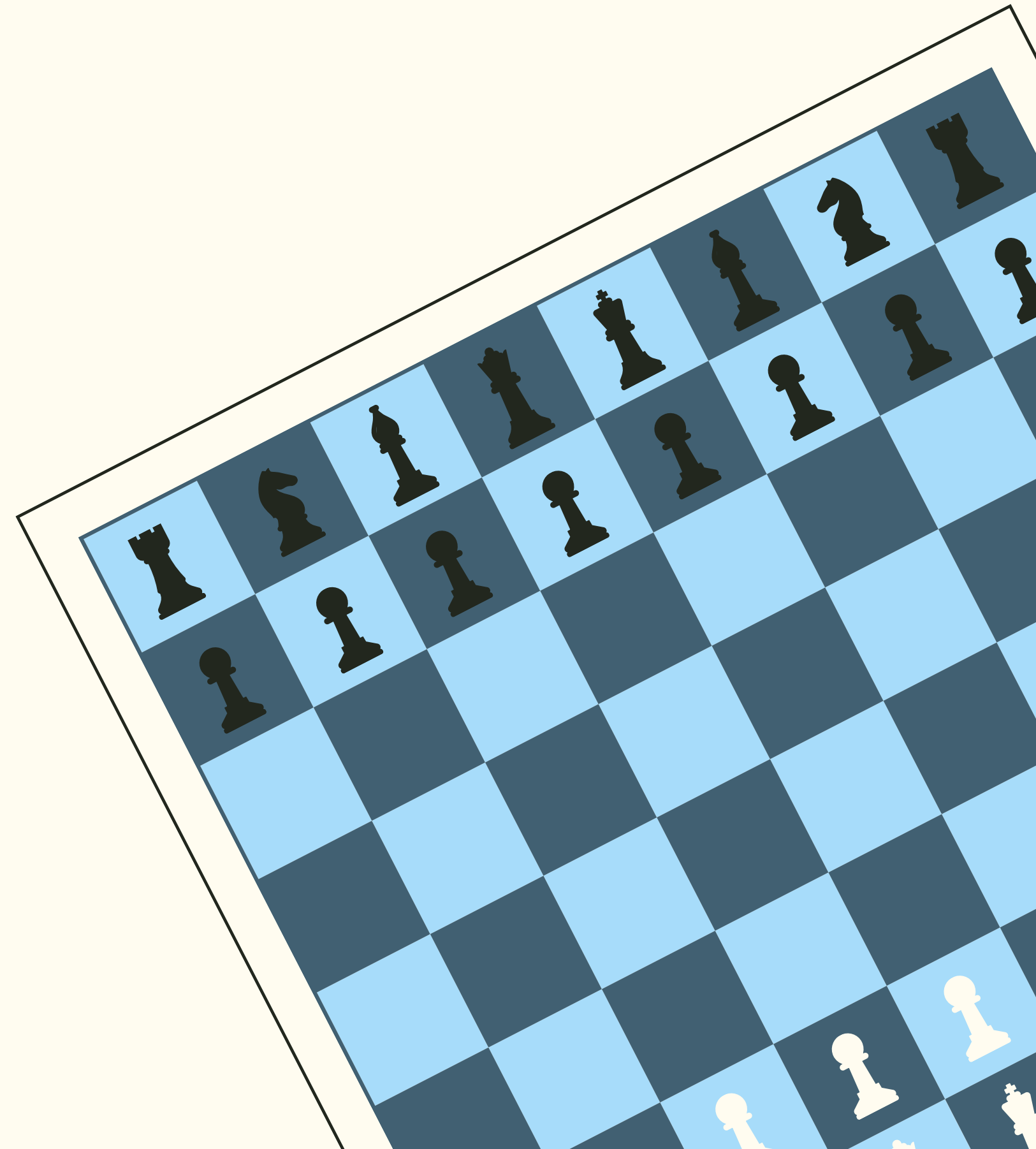
REPOSITÓRIO GITHUB



[HTTPS://GITHUB.COM/V
ALESCAMOURA/CHES](https://github.com/valecamoura/chess)

CI → TESTES

DEPLOY



Obrigado
pela atenção!

SAVE

