

DAMAS

TAMG

O PROJETO

APRESENTAÇÃO 3

Desenvolvimento de Jogo de Damas
Disciplina: Engenharia de Software II
Prof.: Troy Kohwalter

Equipe: TAMG

T – Thiago Freitas

A – Alan Gomes

M – Matheus Pimentel

G – Gabriel Ferreira

O que Vamos Abordar?

01

Sprints Finais

Tarefas Desenvolvidas

02

Monitoramento

Monitoramento e
Controle

03

Testes

Testes realizados, como
foram realizados e
exemplos

04

Demonstração

Demonstração do
produto



SISTEMA DE VERSIONAMENTO



VERSIONAMENTO

Ramificação

Mudança da estratégia de ramificação para **Caótica**

Neste caso, o ramo principal (main) recebe todas as modificações e correções



MONITORAMENTO E CONTROLE

04 - 11 NOV

SPRINT



SPRINT

The image shows a Trello board with three columns. The first column, 'SPRINT (4 a 11 NOV)', contains 8 cards with an orange progress bar on the first card. The second column, 'DONE', contains 8 cards with a green progress bar on each. The third column, 'UNDONE AND BUGS', is empty. Each card has a title and a description. The cards in the 'SPRINT' column are: 'BUG - Erro ao selecionar células na extremidade (erro ao calcular as possibilidades)', 'Procedimento de movimento do jogador', 'Métodos de análise de jogadas Obrigatorias', 'Método de pulo (comer peça)', 'definir dama', 'Método de verificação de vitória/perda', 'Método de Pulo (pulo para trás)', and 'Método para verificar quando a peça vira dama'. The cards in the 'DONE' column are: 'BUG - Erro ao selecionar células na extremidade (erro ao calcular as possibilidades)', 'Procedimento de movimento do jogador', 'Métodos de análise de jogadas Obrigatorias', 'Método de pulo (comer peça)', 'definir dama', 'Método de verificação de vitória/perda', 'Método de Pulo (pulo para trás)', and 'Método para verificar quando a peça vira dama'. The 'UNDONE AND BUGS' column has a button to add a card.

SPRINT (4 a 11 NOV)	DONE	UNDONE AND BUGS
BUG - Erro ao selecionar células na extremidade (erro ao calcular as possibilidades)	BUG - Erro ao selecionar células na extremidade (erro ao calcular as possibilidades)	+ Adicionar um cartão
Procedimento de movimento do jogador	Procedimento de movimento do jogador	
Métodos de análise de jogadas Obrigatorias	Métodos de análise de jogadas Obrigatorias	
Método de pulo (comer peça)	Método de pulo (comer peça)	
definir dama	definir dama	
Método de verificação de vitória/perda	Método de verificação de vitória/perda	
Método de Pulo (pulo para trás)	Método de Pulo (pulo para trás)	
Método para verificar quando a peça vira dama	Método para verificar quando a peça vira dama	
Testes	Testes	
+ Adicionar outro cartão	+ Adicionar outro cartão	

11 - 18 OUT

SPRINT



SPRINT

The screenshot displays a Trello board for a sprint cycle. The board is organized into three main columns:

- SPRINT (11 a 18 NOV)**: This column contains 8 cards representing tasks to be completed during the sprint. The tasks are:
 - Definir jogadas obrigatórias (para damas)
 - Jogadas possíveis para damas quando há peças para comer
 - Jogadas possíveis dama com e sem obrigatorias
 - pulo de damas
 - Encadeamento
 - função que testa se existem jogadas possíveis (se não o jogador perde)
 - troca de turno
 - Testes
- DONE**: This column contains 6 cards, which are identical to the ones in the 'SPRINT' column, indicating tasks that have been completed. Each card in this column has a green progress bar at the top.
- UNDONE AND BUGS**: This column is currently empty, with a button at the bottom to '+ Adicionar um cartão'.

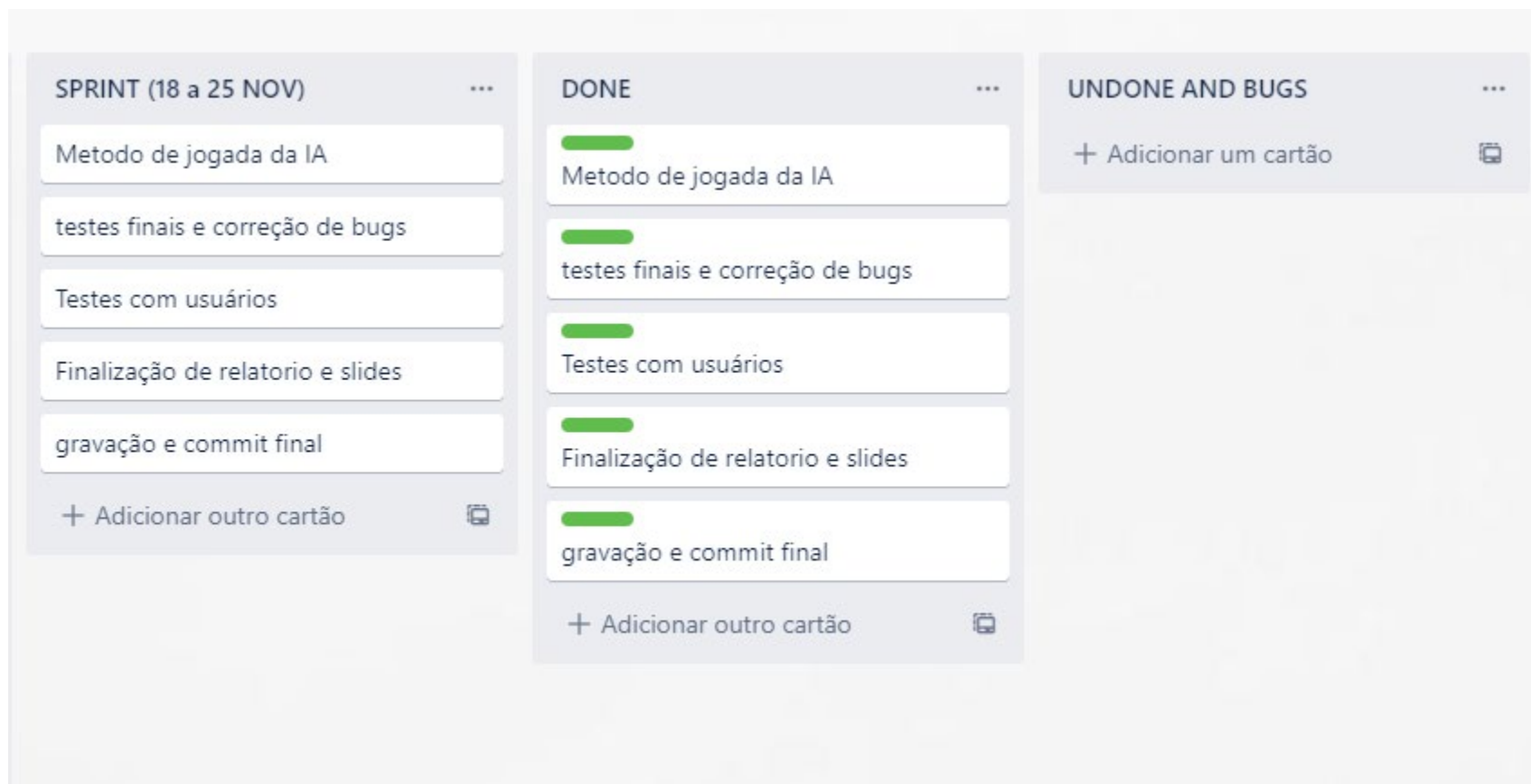
Each card in the 'SPRINT' and 'DONE' columns has a '+ Adicionar outro cartão' button at the bottom right.

18 OUT - 25 NOV

SPRINT

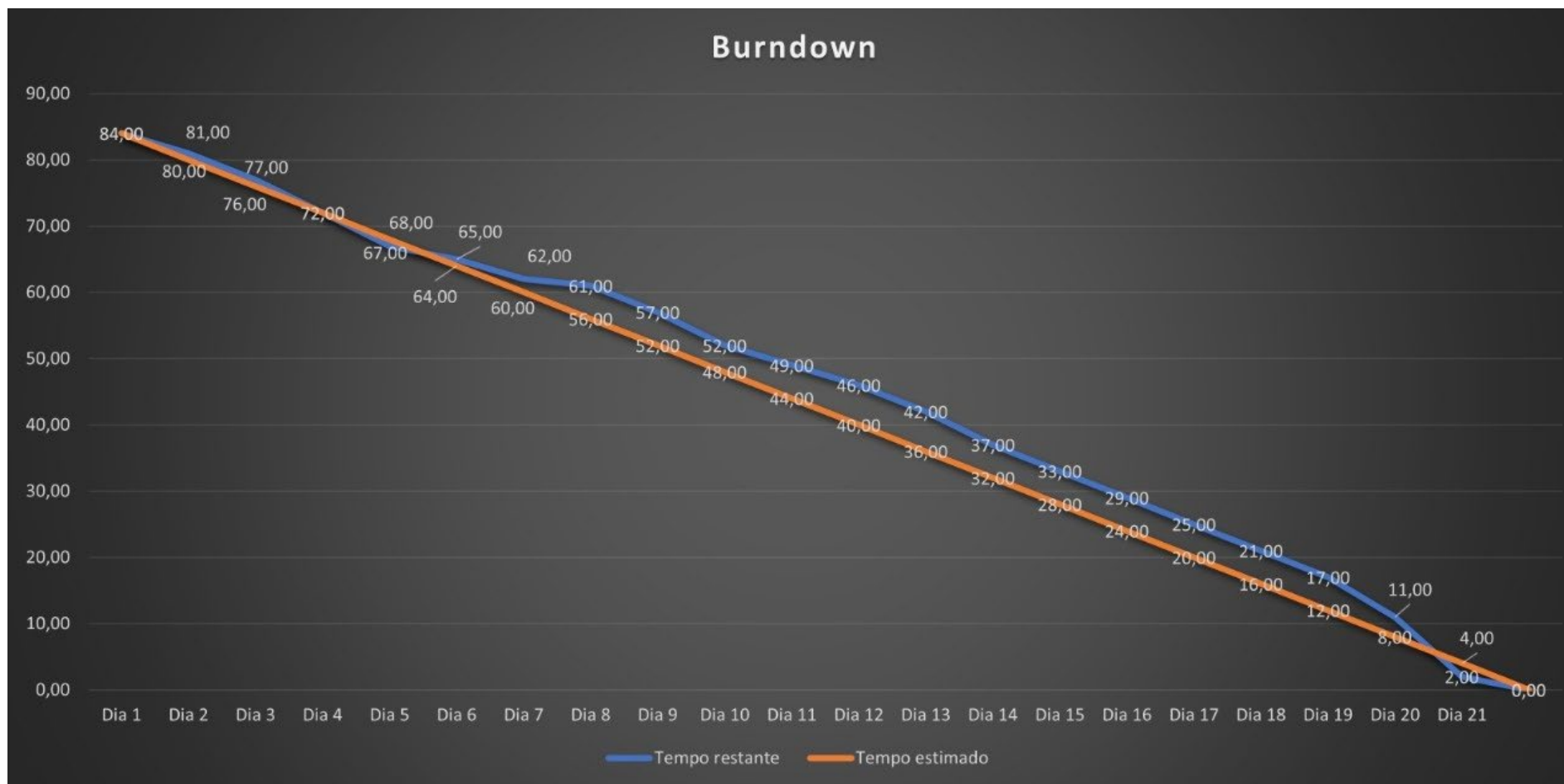


SPRINT



MONITORAMENTO E CONTROLE

BURNDOWN



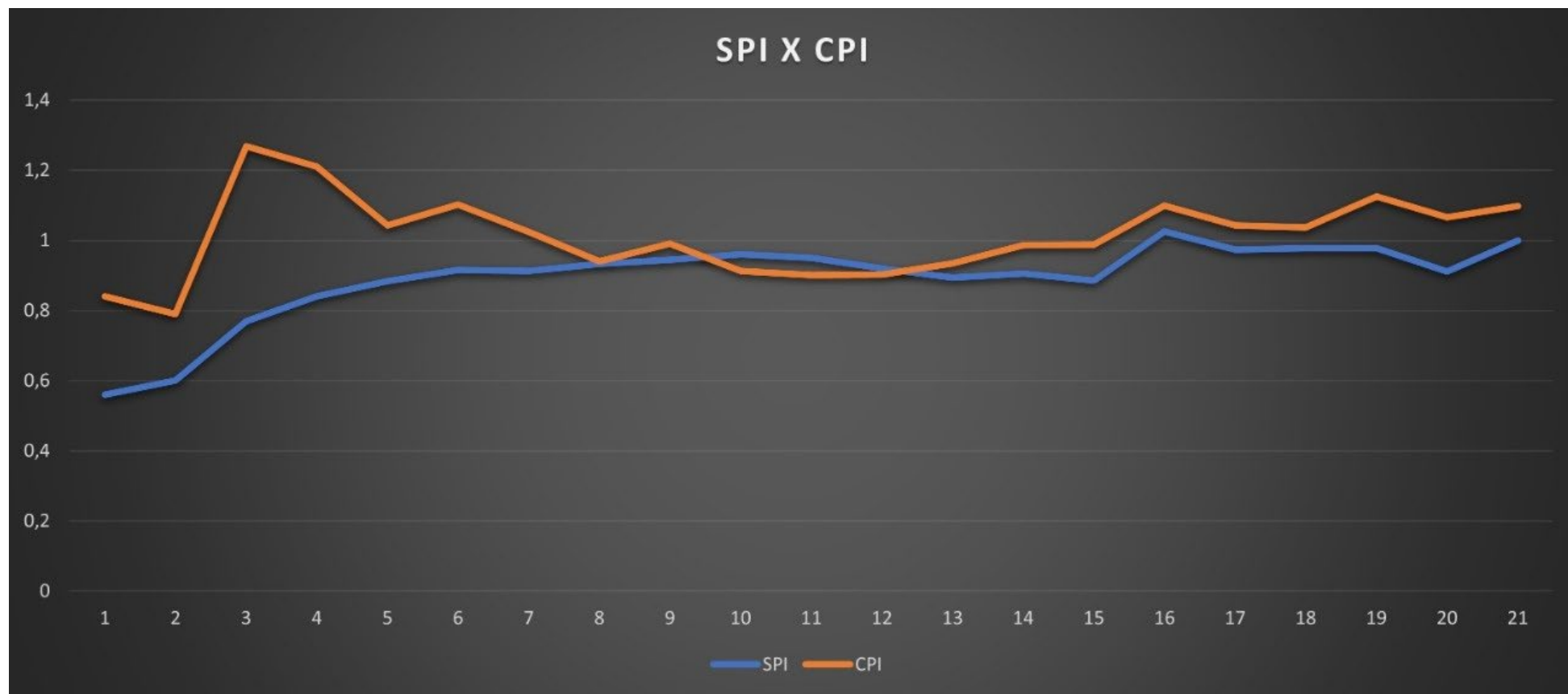
MONITORAMENTO E CONTROLE

PLANILHA DE ANÁLISE

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Atividades	Total de horas	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12
2	Correção de bug ao selecionar células na extremidade (erro ao calcular possibilidades)	3,00	3,00											
3	Procedimento de movimento do jogador	4,00		4,00										
4	Método de Análise de Jogadas Obrigatórias	5,00			5,00									
5	Método de pulo (comer peça)	5,00				5,00								
6	Método de verificação de vitória/perda	2,00					2,00							
7	Método de Pulo (pulo para trás)	3,00						3,00						
8	Método para verificar quando a peça vira dama	1,00							1,00					
9	Definir jogadas obrigatórias (para damas)	4,00								4,00				
10	Jogadas possíveis para damas quando há peças para comer	5,00									5,00			
11	Jogadas possíveis da dama (obrigatórias ou não)	3,00										3,00		
12	Pulo de damas	3,00											3,00	
13	Encadeamento	4,00												4,00
14	Função que testa se existem jogadas possíveis (se não o jogador perde)	2,00												
15	Troca de turno	3,00												
16	Método de jogada da IA	20,00												
17	Testes finais e correção de bugs	6,00												
18	Testes com usuários reais	4,00												
19	Finalização de relatório e slides	5,00												
20	Gravação e commit final	2,00												
21	Tempo restante	84,00	81,00	77,00	72,00	67,00	65,00	62,00	61,00	57,00	52,00	49,00	46,00	42,00
22	Tempo estimado	84,00	80,00	76,00	72,00	68,00	64,00	60,00	56,00	52,00	48,00	44,00	40,00	36,00
23														
24	Soma de horas		3,00	4,00	5,00	5,00	2,00	3,00	1,00	4,00	5,00	3,00	3,00	4,00
25														
26														
27	Custo total	R\$ 5.040,00												
28	Valor planejado		R\$ 180,00	R\$ 420,00	R\$ 720,00	R\$ 1.020,00	R\$ 1.140,00	R\$ 1.320,00	R\$ 1.380,00	R\$ 1.620,00	R\$ 1.920,00	R\$ 2.100,00	R\$ 2.280,00	R\$ 2.520,00
29	Percentual planejado de conclusão		4%	8%	14%	20%	23%	26%	27%	32%	38%	42%	45%	50%
30	Valor agregado		R\$ 100,80	R\$ 252,00	R\$ 554,40	R\$ 856,80	R\$ 1.008,00	R\$ 1.209,60	R\$ 1.260,00	R\$ 1.512,00	R\$ 1.814,40	R\$ 2.016,00	R\$ 2.167,20	R\$ 2.318,40
31	Percentual real de conclusão		2%	5%	11%	17%	20%	24%	25%	30%	36%	40%	43%	46%
32	Custo real		R\$ 120,00	R\$ 319,00	R\$ 437,00	R\$ 708,00	R\$ 967,00	R\$ 1.098,00	R\$ 1.230,00	R\$ 1.609,00	R\$ 1.830,00	R\$ 2.207,00	R\$ 2.403,00	R\$ 2.567,00
33														
34	SPI		0,56	0,6	0,77	0,84	0,884210526	0,916363636	0,913043478	0,933333333	0,945	0,96	0,950526316	0,92
35	CPI		0,84	0,789968652	1,268649886	1,210169492	1,042399173	1,101639344	1,024390244	0,939714108	0,99147541	0,913457182	0,901872659	0,903155434
36														
37														

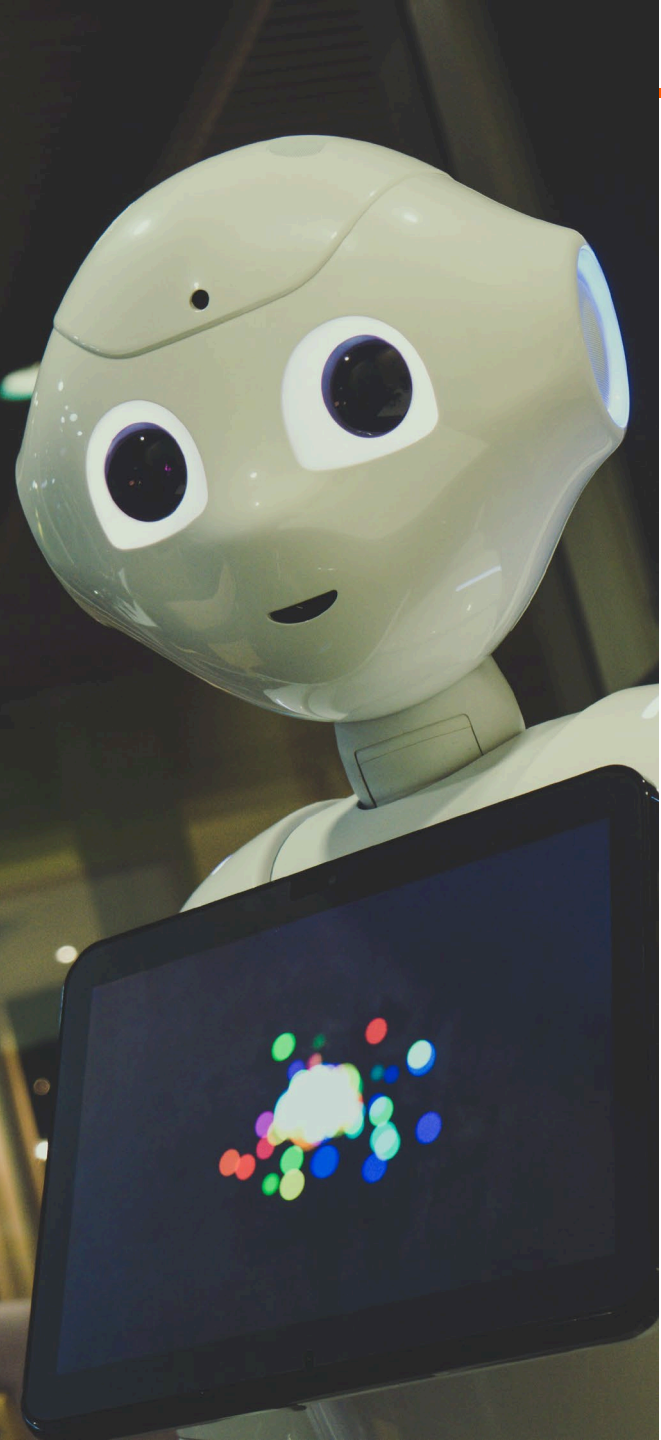
MONITORAMENTO E CONTROLE

SPI x CPI





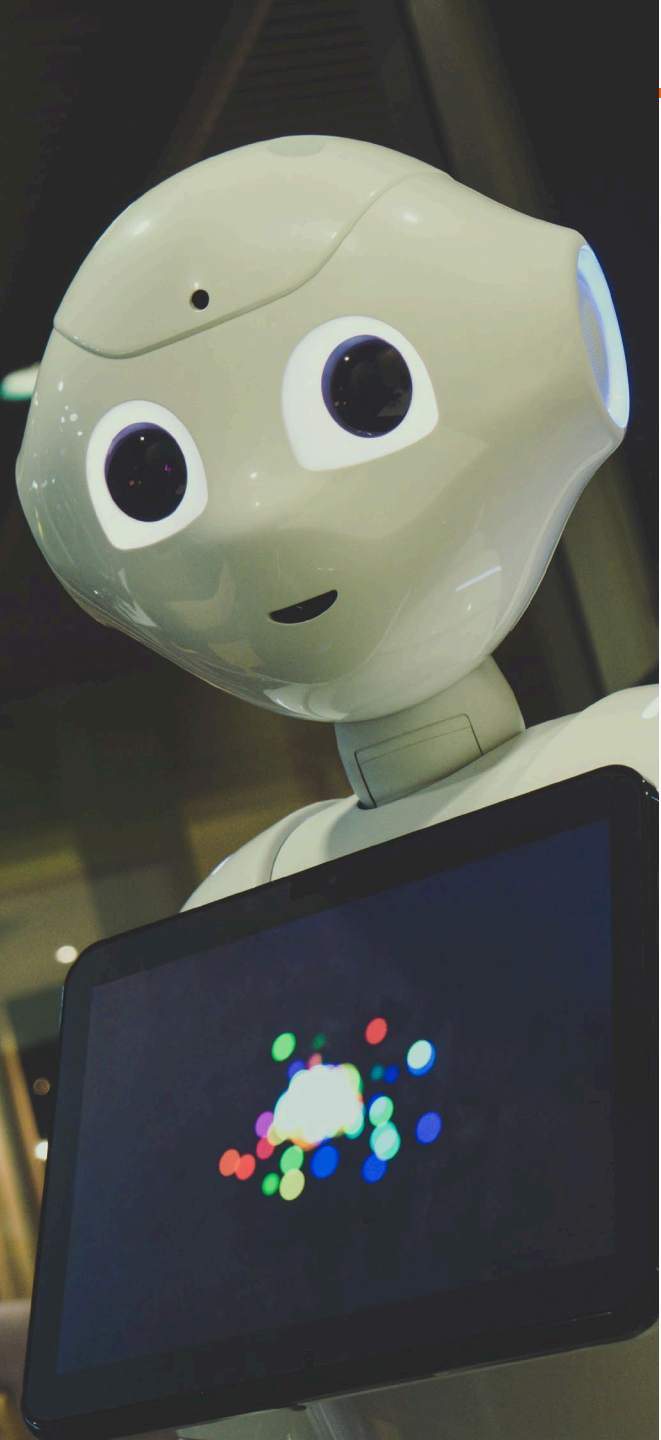
SOBRE A IA



IA

Características

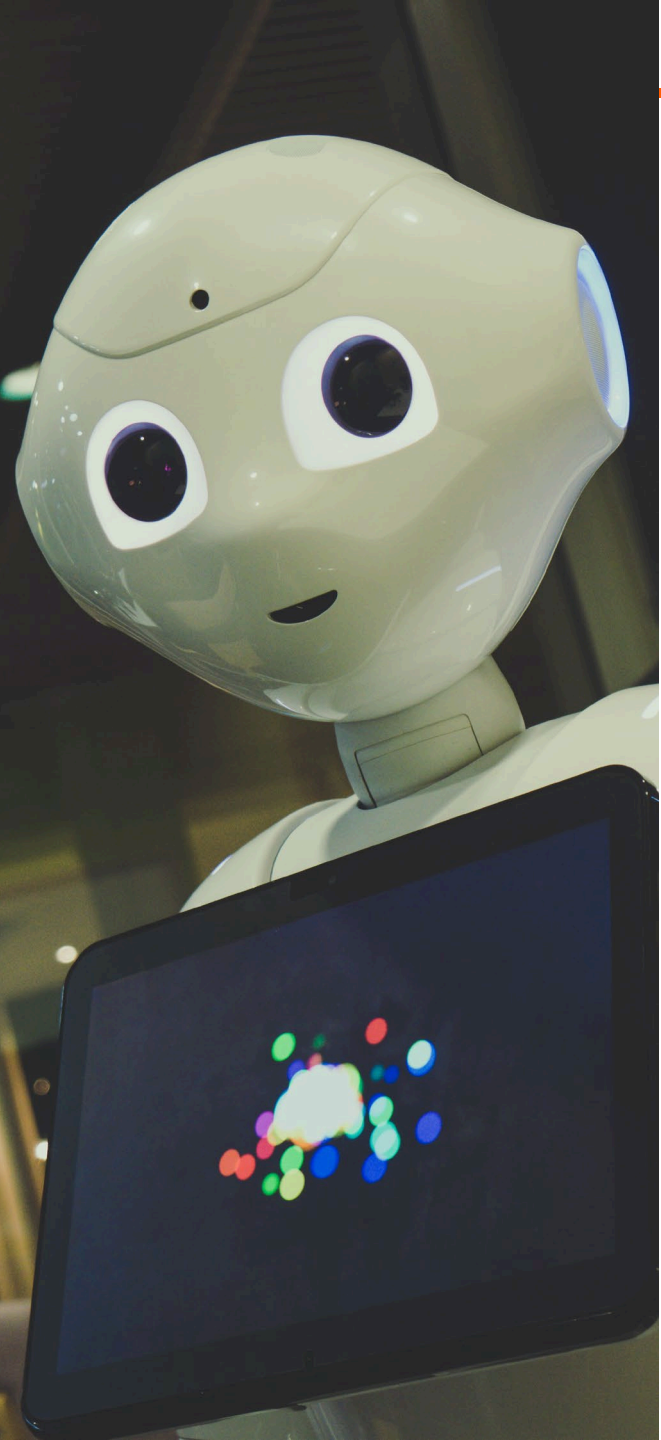
A IA funciona dentro de um procedimento de turno da IA na classe jogo, trabalhando com busca e listas para definir as suas possibilidades de jogada, a seleção e o movimento.



IA

Capacidades

- Detecta se há peças para comer, se houver vai fazê-lo
- Detecta se há encadeamento, se houver, fará
- Detecta quais peças possuem jogadas e seleciona entre elas
- Detecta pra quais casas as peças podem se mover
- Entende e explora (ainda que de forma rudimentar) as damas
- Consegue “comer” peças com damas



IA

Limitações

- A montagem das listas de possibilidades respeitam as regras do jogo, após montada a lista a IA toma decisões arbitrárias
- Não foi implementada Heurística para as decisões de jogadas da IA
- A IA é simples



TESTES

Matriz de teste

```
#tabuleiro de jogo
self.tabuleiro = [['x', '-', 'x', '-', 'x', '-', 'x', '-'],
                  ['- ', 'x', '-', 'x', '-', 'x', '-', 'x'],
                  ['x', '-', 'x', '-', 'x', '-', 'x', '-'],
                  ['- ', '-', '-', '-', '-', '-', '-', '-'],
                  ['- ', '-', '-', '-', '-', '-', '-', '-'],
                  ['- ', 'o', '-', 'o', '-', 'o', '-', 'o'],
                  ['o', '-', 'o', '-', 'o', '-', 'o', '-'],
                  ['- ', 'o', '-', 'o', '-', 'o', '-', 'o']]
```

```
#uso em testes
self.tabuleirot = [['x', '-', 'x', '-', 'o', '-', 'x', '-'],
                   ['- ', 'x', '-', 'x', '-', '-', '-', 'x'],
                   ['x', '-', 'x', '-', '-', '-', 'x', '-'],
                   ['- ', '-', '-', '-', '-', '-', '-', '-'],
                   ['- ', '-', '-', '-', '-', '-', 'x', '-'],
                   ['- ', 'o', '-', '-', '-', '-', '-', 'o'],
                   ['o', '-', 'o', '-', 'x', '-', 'o', '-'],
                   ['- ', 'o', '-', 'o', '-', 'o', '-', 'o']]
```


Uso de prints para verificação

```
#print(self.lista_obrigatorias)
self.lista_possibilidades = self.verificarjogadas()
print("obrigatorias: ", self.lista_obrigatorias)
#definir peça
peca = ''
# Se houver, Recuperar casa selecionada na matriz de jogo
if(self.casa_selecionada):
    for i in range(8):
        for j in range(8):
            if (self.tabuleiodes[i][j].x == self.selecionada.x) and (
                self.tabuleiodes[i][j].y == self.selecionada.y):
                peca = (i, j)
#verificar se ha clique no mouse
if(self.mouse.is_button_pressed(1)):
    print("entrou no loop")
    #verifica se há célula selecionada
    if(self.casa_selecionada):
        #print("achou casa selecionada")
        #se a peça na casa selecionada pertence ao jogador
        if (peca) and (self.tabuleiro[peca[0]][peca[1]] == 'o' or self.tabuleiro[peca[0]][peca[1]] == '0'):
```




DEMONSTRAÇÃO

DEMONSTRAÇÃO NO VIDEO



OBRIGADO