

Programação de Computadores - CI-208

Prof. Murilo da Silva - DINF/UFPR

Aula 02

ESTRUTURA GERAL DE UM ALGORITMO

- **Entrada:** como os ingredientes de uma receita, a entrada de um algoritmo é um dado.
- **Processamento:** são os passos executados sobre a entrada (assim como os passos de uma receita).
- **Saída:** é o resultado do processamento sobre a entrada (se a sua receita fosse de bolo, a saída seria o próprio bolo).

Os Dados

DIFERENTES DISPOSITIVOS PERMITEM DIFERENTES INTERAÇÕES
COM UM COMPUTADOR!

Todas essas interações se baseiam em troca de dados (entrada e saída)

Porém, para que um computador compreenda o que é recebido, é necessário determinar uma **representação geral e unificada** de dados

Números

No **sistema decimal** temos dez símbolos numéricos disponíveis: **0...9**

Após contarmos até o nove, iniciamos uma composição de números que resultam em dezenas, centenas, milhares...

No **sistema binário**, executamos o mesmo processo, porém considerando apenas dois símbolos numéricos disponíveis: **0...1**

Números

Assim, do mesmo jeito que 10 representa $(9 + 1)$ no sistema decimal; 10 representa $(1 + 1)$ no sistema binário

$$0 = 0$$

$$1 = 1$$

$$10 = 2$$

$$11 = 3$$

$$100 = 4$$

$$101 = 5$$

$$110 = 6$$

$$111 = 7$$

$$1000 = 8$$

$$1001 = 9$$

$$1010 = 10$$

$$1011 = 11$$

$$1100 = 12$$

$$1101 = 13$$

$$1110 = 14$$

$$1111 = 15$$

Letras e Símbolos

Em geral, podemos usar uma codificação **ASCII Ext. (8 bits)** para representar letras e símbolos usuais. Entretanto, existem outros padrões, como o Unicode – atualmente, o mais usado é UTF-8.

BANG MA ASCII																														
0	16	▶	32	48	0	64	@	80	P	96	`	112	p	128	Ç	144	É	160	Á	176	█	192	┌	208	┐	224	α	240	≡	
1	17	◀	33	!	49	1	65	A	81	Q	97	a	113	q	129	ü	145	æ	161	í	177	▒	193	└	209	┘	225	β	241	±
2	18	↑	34	50	2	66	B	82	R	98	b	114	r	130	é	146	Æ	162	ó	178	▓	194	┐	210	┘	226	Γ	242	≥	
3	19	!!	35	#	51	3	67	C	83	S	99	c	115	s	131	á	147	ø	163	û	179	█	195	└	211	┘	227	Π	243	≤
4	20	¶	36	\$	52	4	68	D	84	T	100	d	116	t	132	ä	148	ö	164	n	180	█	196	└	212	┘	228	Σ	244	∫
5	21	§	37	%	53	5	69	E	85	U	101	e	117	u	133	à	149	÷	165	ñ	181	█	197	└	213	┘	229	σ	245	∫
6	22	—	38	&	54	6	70	F	86	V	102	f	118	v	134	á	150	û	166	*	182	█	198	└	214	┘	230	μ	246	≈
7	23	‡	39	'	55	7	71	G	87	W	103	g	119	w	135	ç	151	ü	167	°	183	█	199	└	215	┘	231	τ	247	∞
8	24	↑	40	(	56	8	72	H	88	X	104	h	120	x	136	ë	152	ÿ	168	¿	184	█	200	└	216	┘	232	Φ	248	√
9	25	↓	41	)	57	9	73	I	89	Y	105	i	121	y	137	ë	153	Ö	169	—	185	█	201	└	217	┘	233	Θ	249	·
10	26	*	42	*	58	10	74	J	90	Z	106	j	122	z	138	ë	154	Ü	170	—	186	█	202	└	218	┘	234	Ω	250	°
11	27	—	43	+	59	11	75	K	91	[	107	k	123	{	139	ÿ	155	ç	171	½	187	█	203	└	219	┘	235	δ	251	√
12	28	└	44	<	60	12	76	L	92	\	108	l	124		140	í	156	£	172	¾	188	█	204	└	220	┘	236	∞	252	n
13	29	↔	45	=	61	13	77	M	93	]	109	m	125	}	141	í	157	¥	173	¡	189	█	205	└	221	┘	237	φ	253	²
14	30	▲	46	>	62	14	78	N	94	^	110	n	126	~	142	Ä	158	£	174	«	190	█	206	└	222	┘	238	ε	254	■
15	31	▼	47	/	63	15	79	O	95	_	111	o	127	ó	143	Ä	159	f	175	»	191	█	207	└	223	┘	239	∩	255	

Cores e Imagens

Imagens podem ser representadas através da **associação de bits ou conjunto de bits a cores** específicas. A associação mais simples é a B&W



PRETO: 0

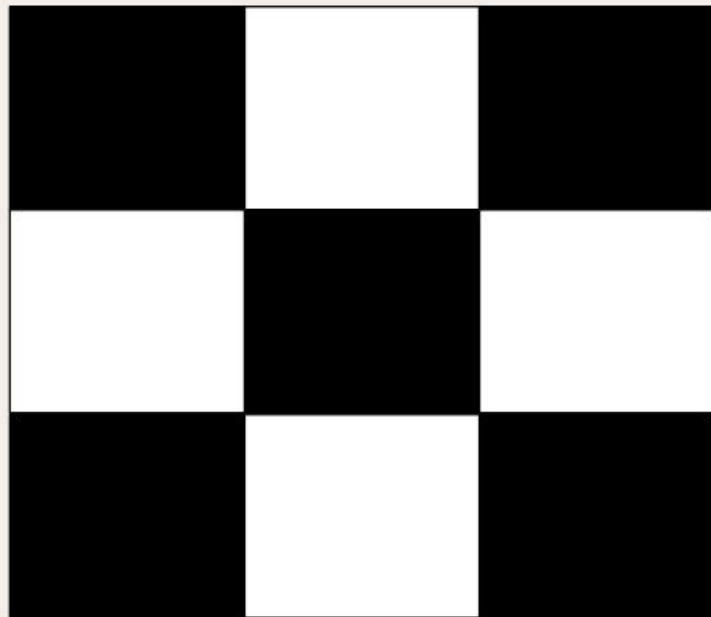


BRANCO: 1

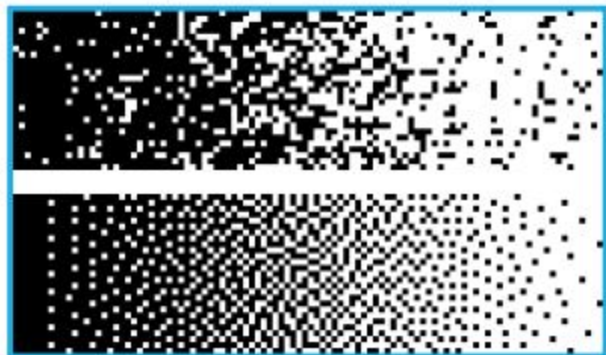
Cores e Imagens

0	1	0
1	0	1
0	1	0

Matriz de pixels



Imagem



A imagem é uma MATRIZ de pontos !

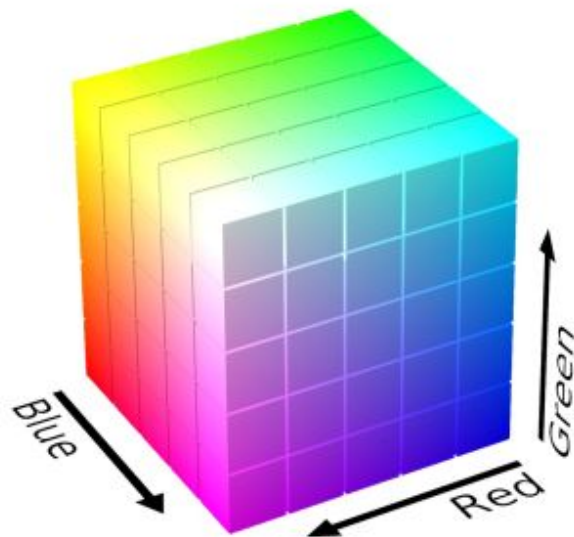
Imagens

- **E as cores?** → Tons de cinza para cada ponto

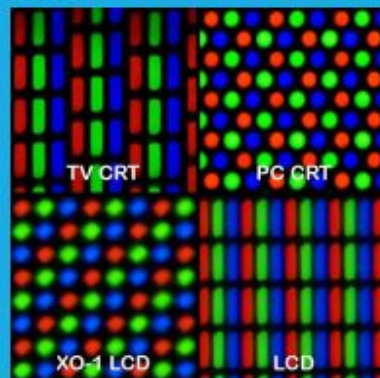
1 bit		2 bits		3 bits	
0		00		000	
1		01		001	
		10		010	
		11		011	
				100	
				101	
				110	
				111	

Imagens

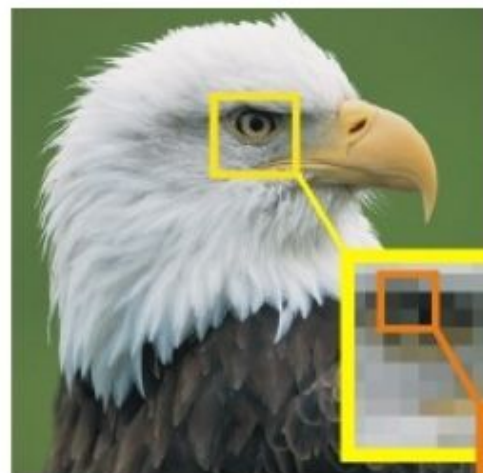
- Mas, e as cores? → Sistema **RGB**



Não é a tãa que a TV
vista BEM de perto é
assim:



Imagens



Um pixel

Imagem: matriz de pixels

Pixel:

- ponto da imagem
- 24 bits

8 bits para VERMELHO
8 bits para VERDE
8 bits para AZUL

1001
0110
0011
1101
0111
0011

Os Dados

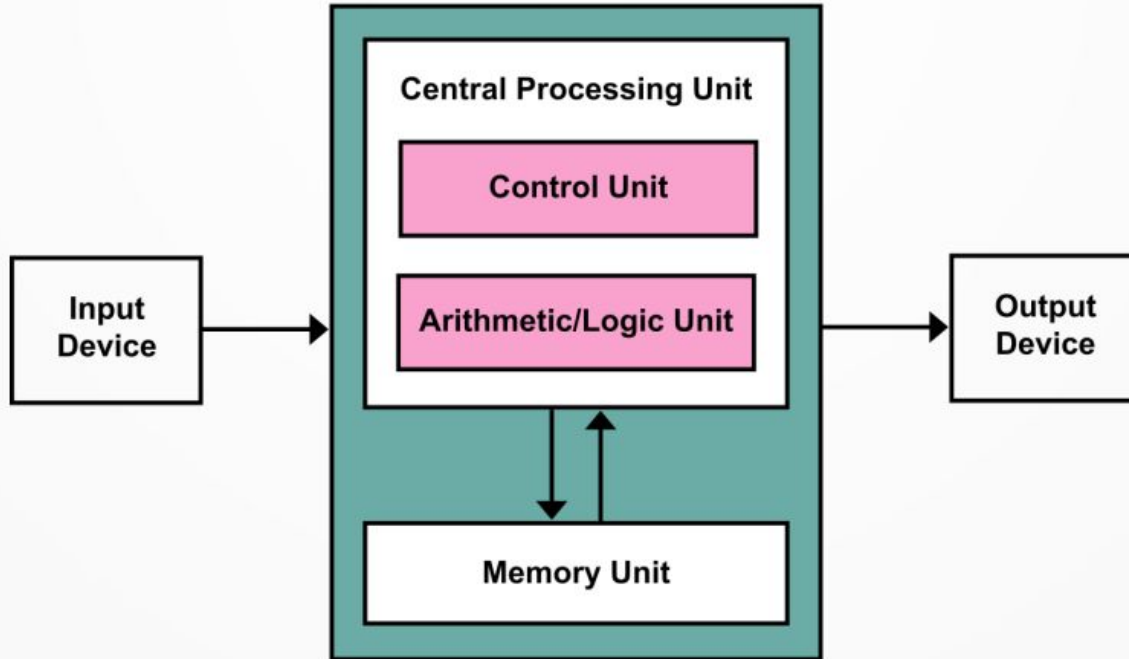
Os dados de um computador são persistidos em unidades de armazenamento, como discos magnéticos, memórias *flash*, e unidades de estado sólido em geral



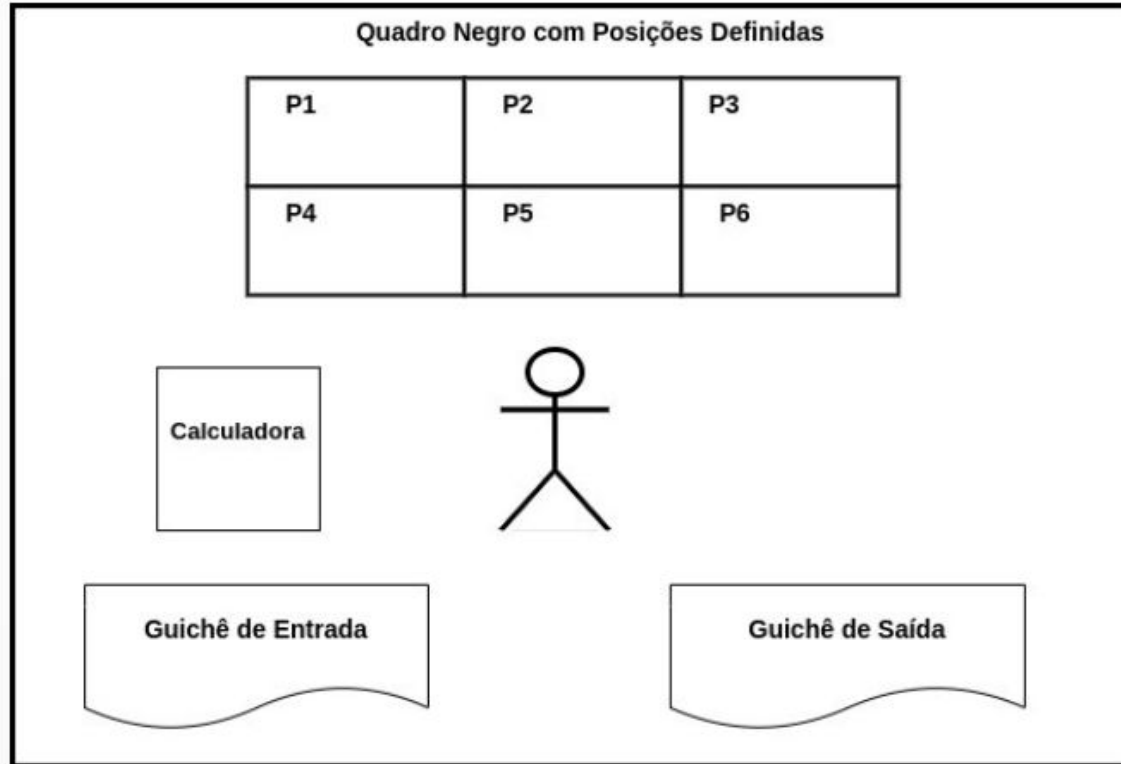
Processa-se dados através da execução de instruções:

- Representadas por bits
- Definidas em um algoritmo

Modelo von Neumann



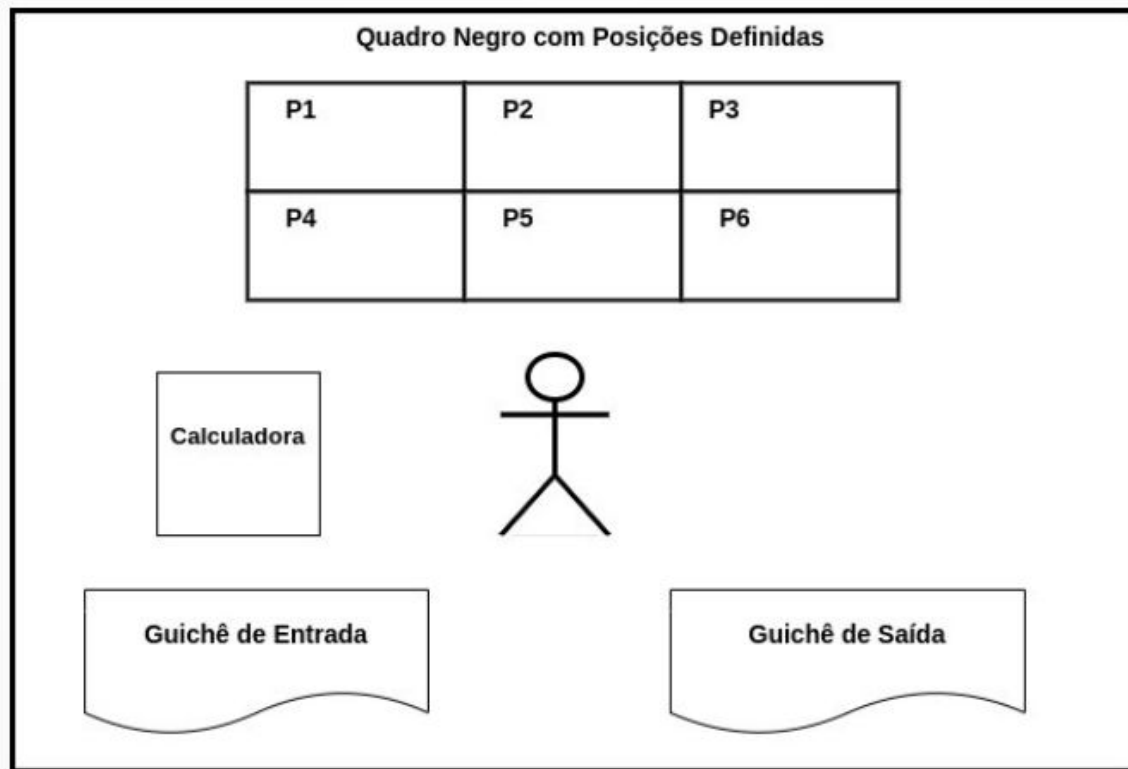
Versão lúdica do Modelo de von Neumann



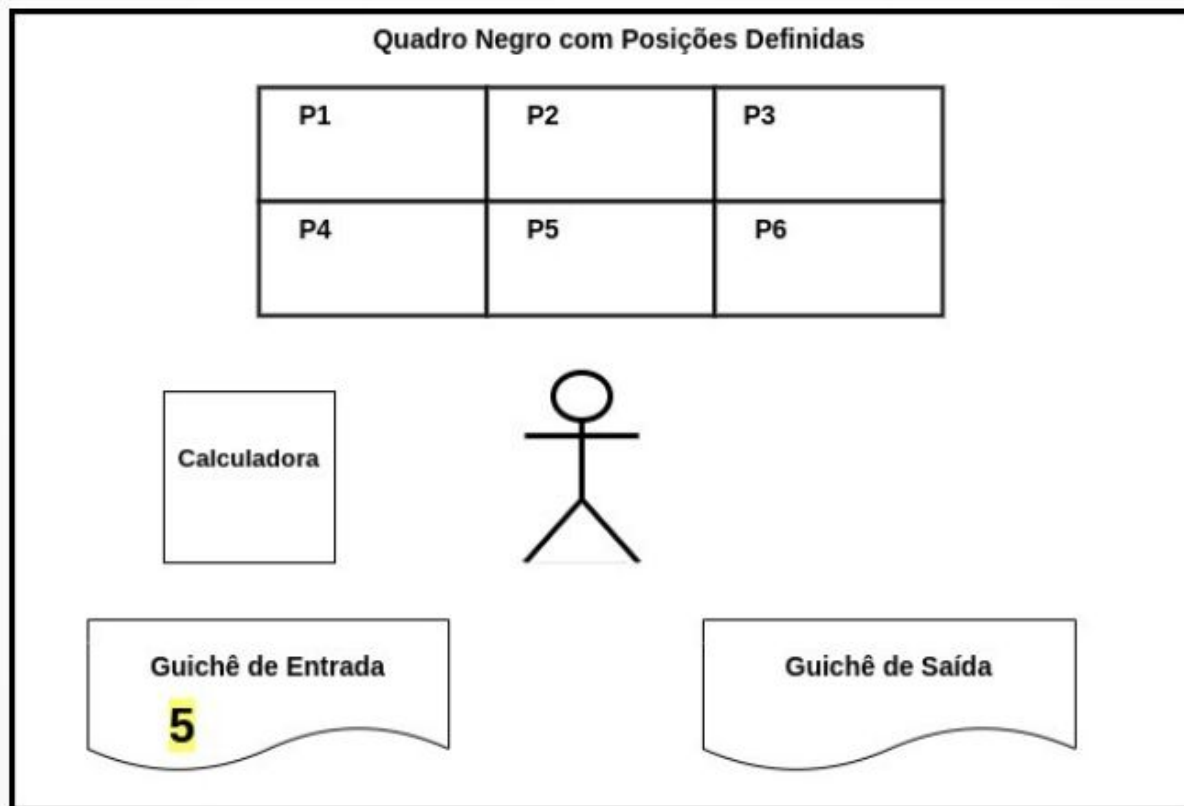
- Algoritmo 1:

- Leia um número do guichê de entrada e escreva na posição P1 do quadro negro
- Leia mais um número do guichê de entrada e escreva na posição P2 do quadro negro
- Calcule a soma do número na posição P1 com o número na posição P2 e escreva o resultado na posição P3
- Agora produza um relatório no guichê de saída informando o resultado da soma

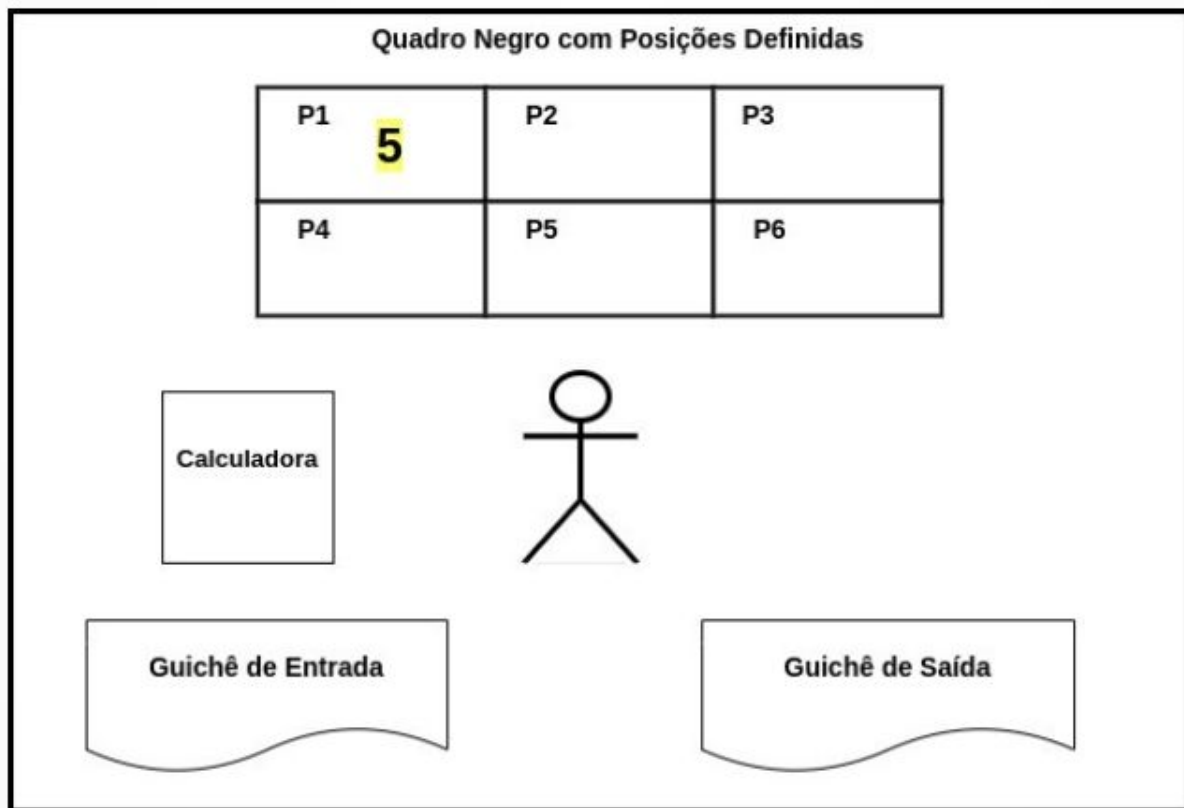
- Leia um número do guichê de entrada e escreva na posição P1 do quadro negro



- Leia um número do guichê de entrada e escreva na posição P1 do quadro negro



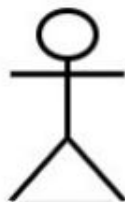
- Leia um número do guichê de entrada e escreva na posição P1 do quadro negro



Quadro Negro com Posições Definidas

P1 5	P2	P3
P4	P5	P6

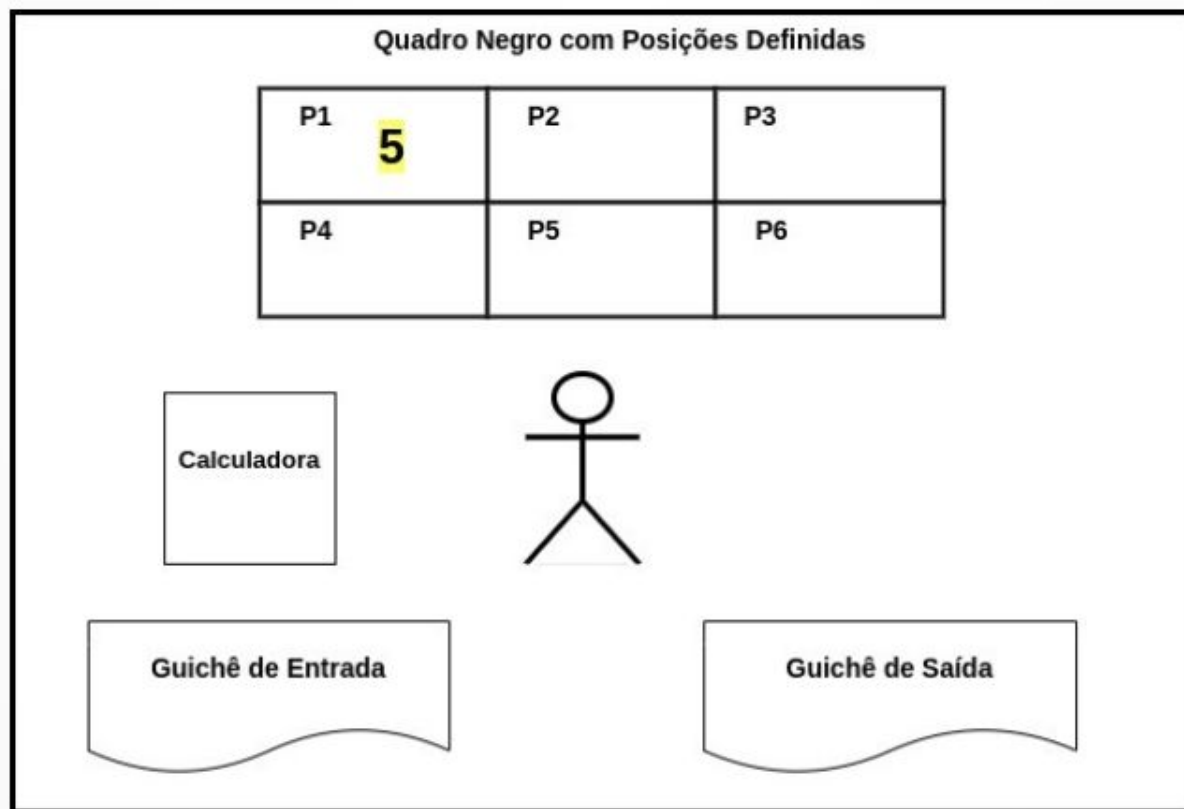
Calculadora



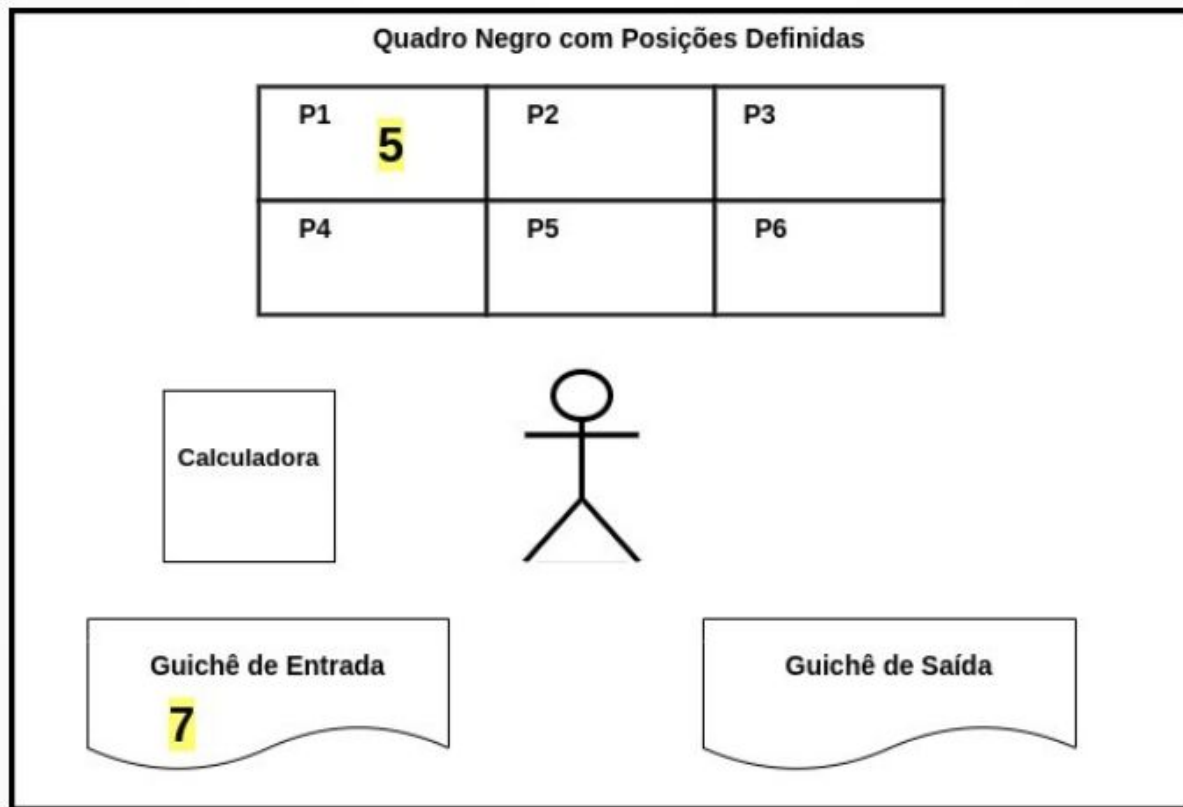
Guichê de Entrada

Guichê de Saída

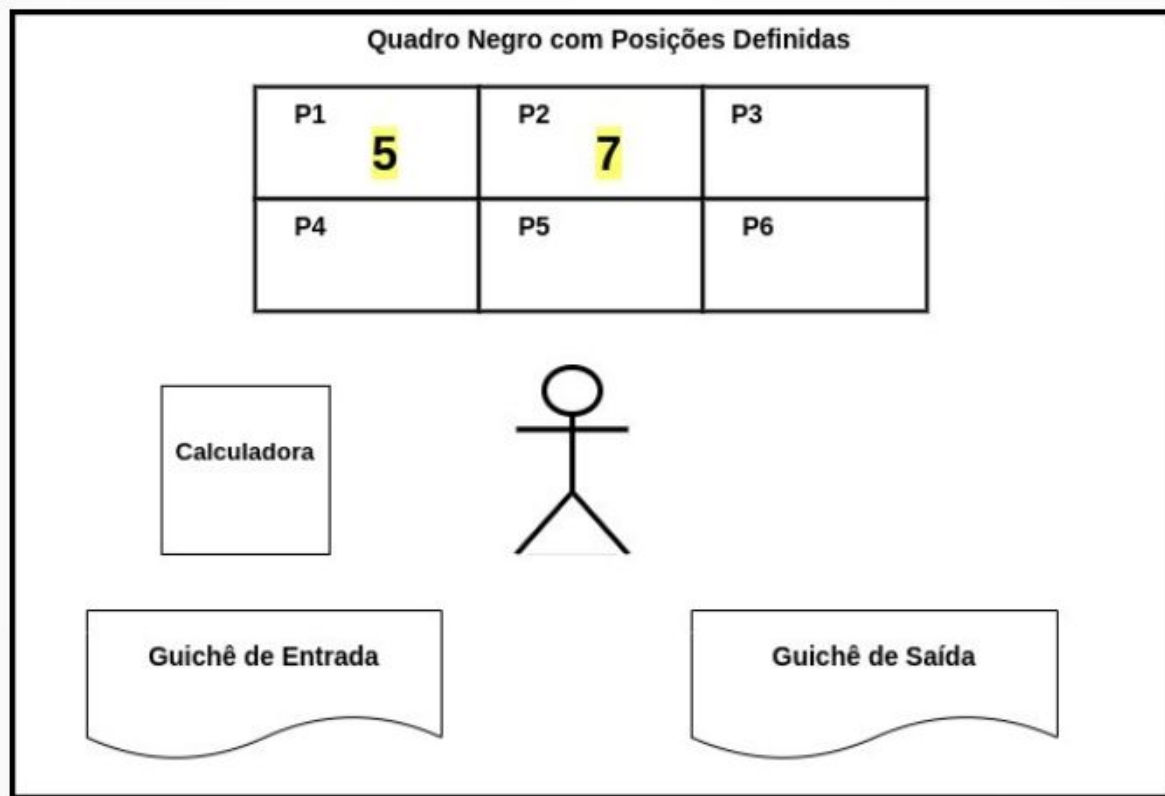
- Leia mais um número do guichê de entrada e escreva na posição P2 do quadro negro



- Leia mais um número do guichê de entrada e escreva na posição P2 do quadro negro



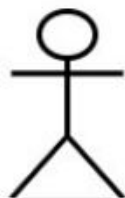
- Leia mais um número do guichê de entrada e escreva na posição P2 do quadro negro



Quadro Negro com Posições Definidas

P1 5	P2 7	P3
P4	P5	P6

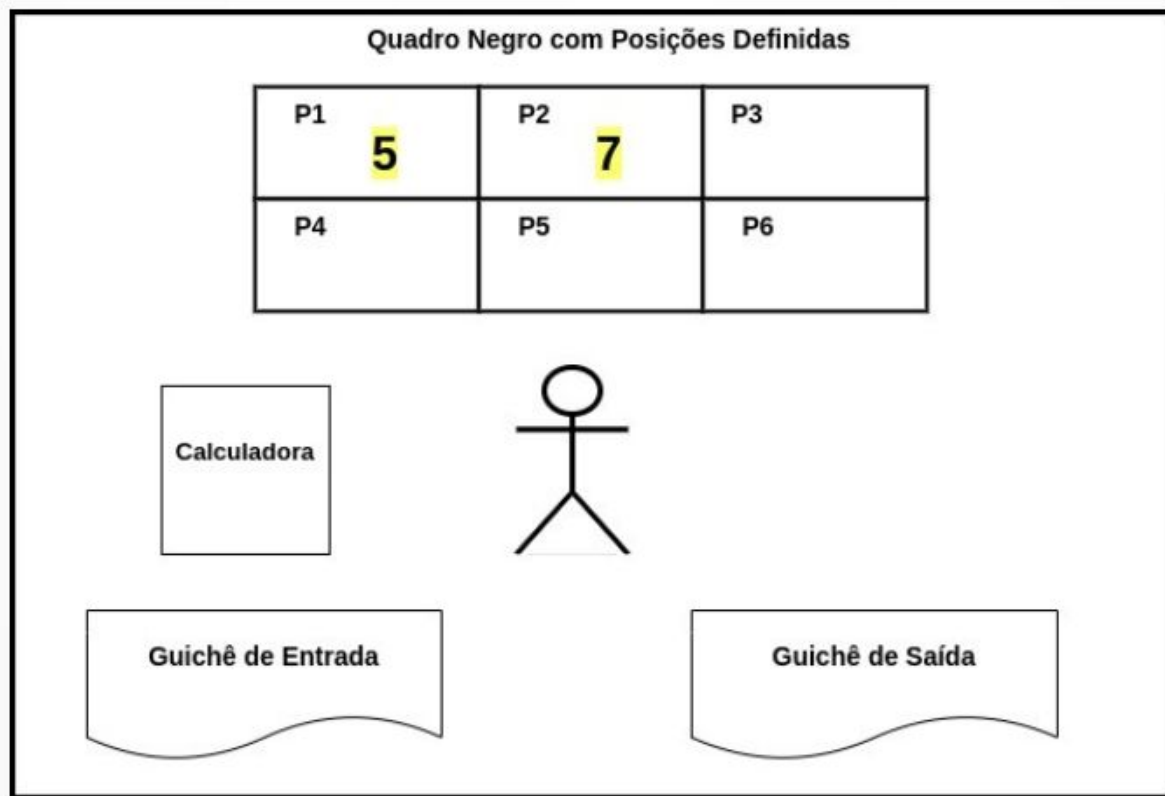
Calculadora



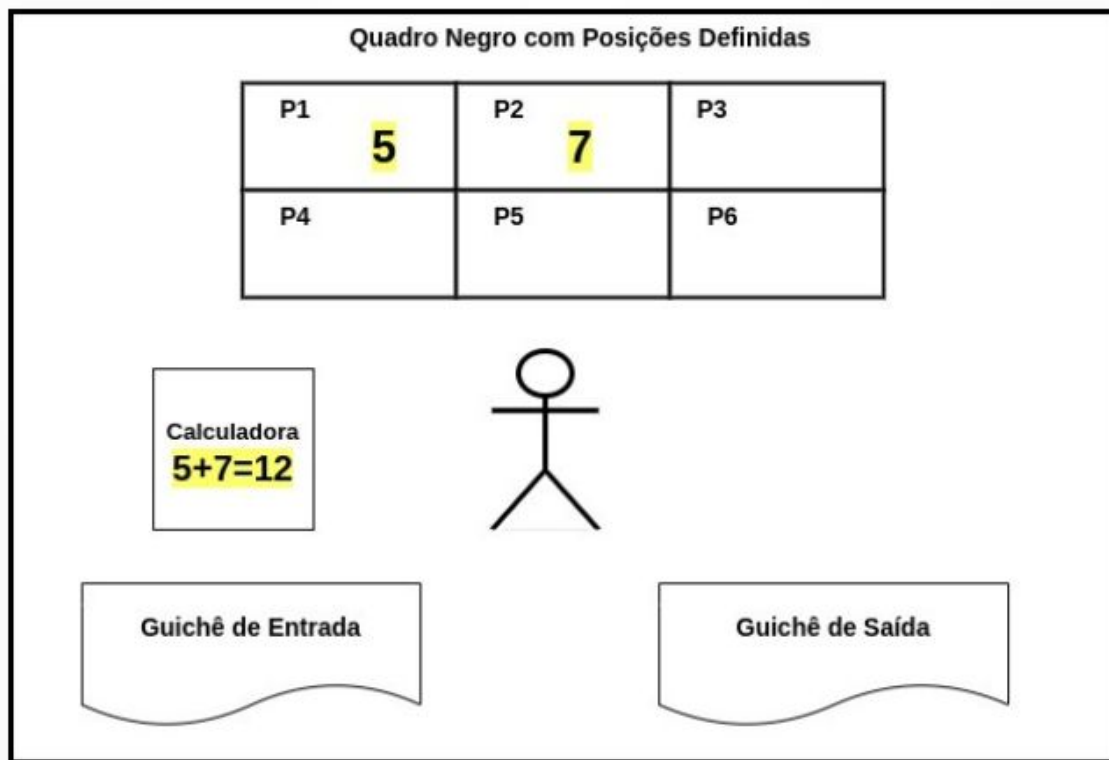
Guichê de Entrada

Guichê de Saída

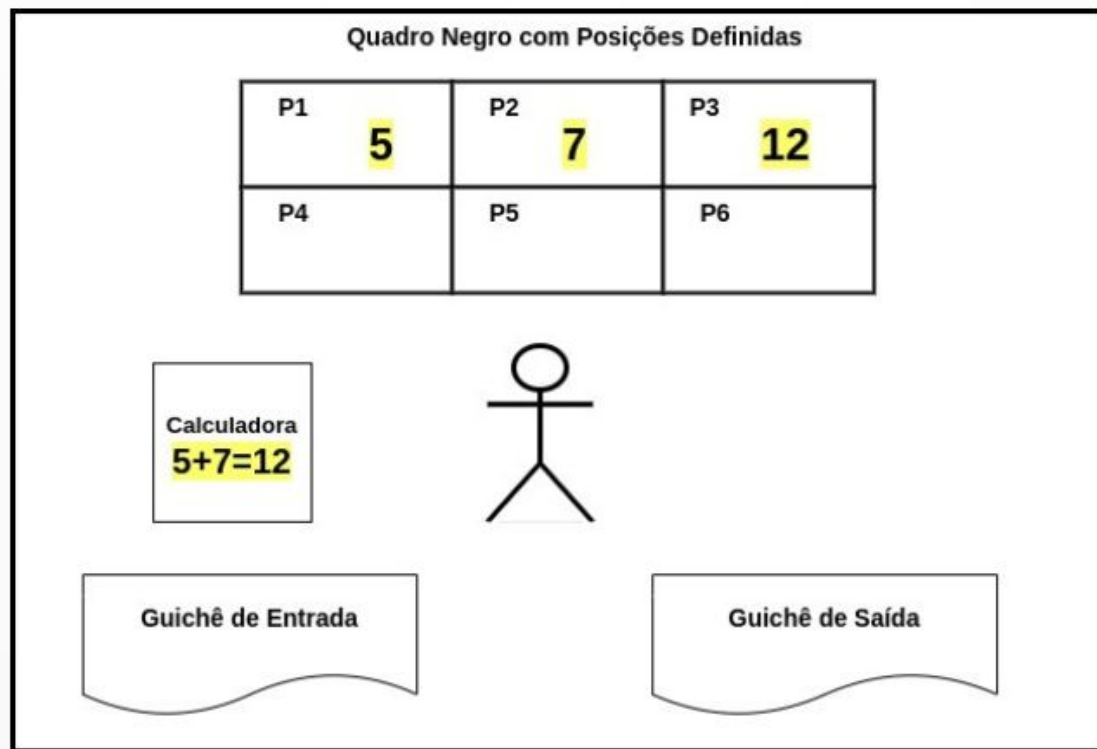
- Calcule a soma do número na posição P1 com o número na posição P2 e escreva o resultado na posição P3



- Calcule a soma do número na posição P1 com o número na posição P2 e escreva o resultado na posição P3



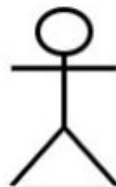
- Calcule a soma do número na posição P1 com o número na posição P2 e escreva o resultado na posição P3



Quadro Negro com Posições Definidas

P1 5	P2 7	P3 12
P4	P5	P6

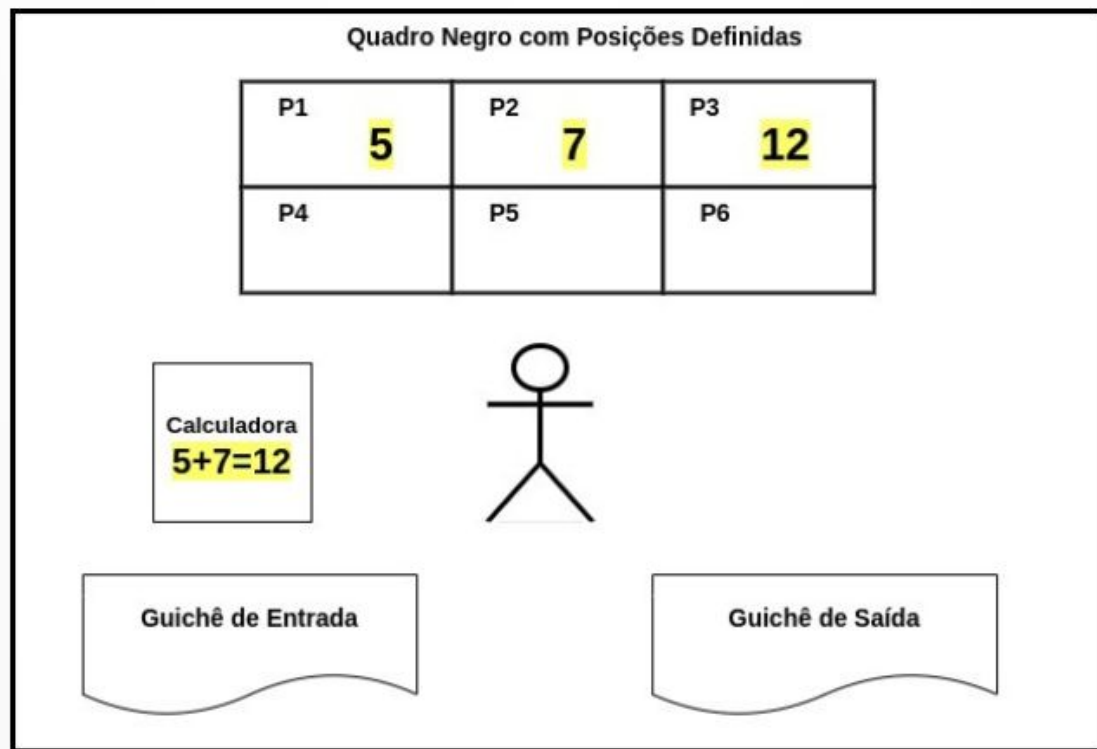
Calculadora
5+7=12



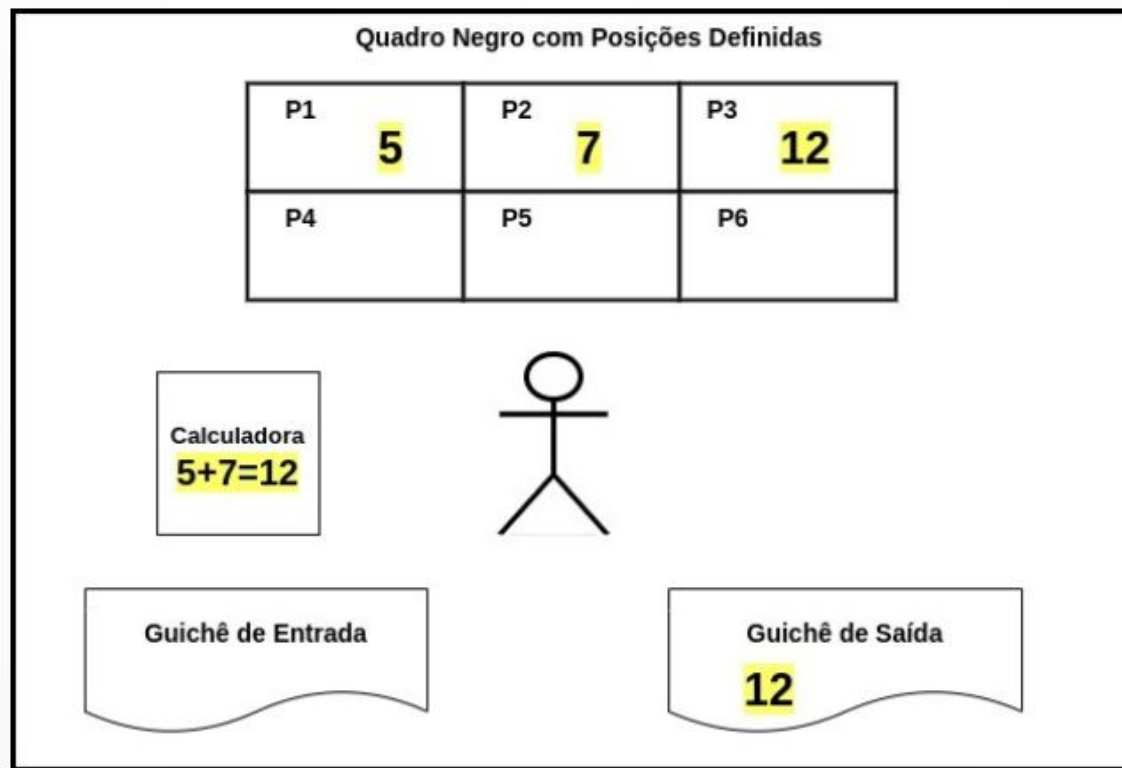
Guichê de Entrada

Guichê de Saída

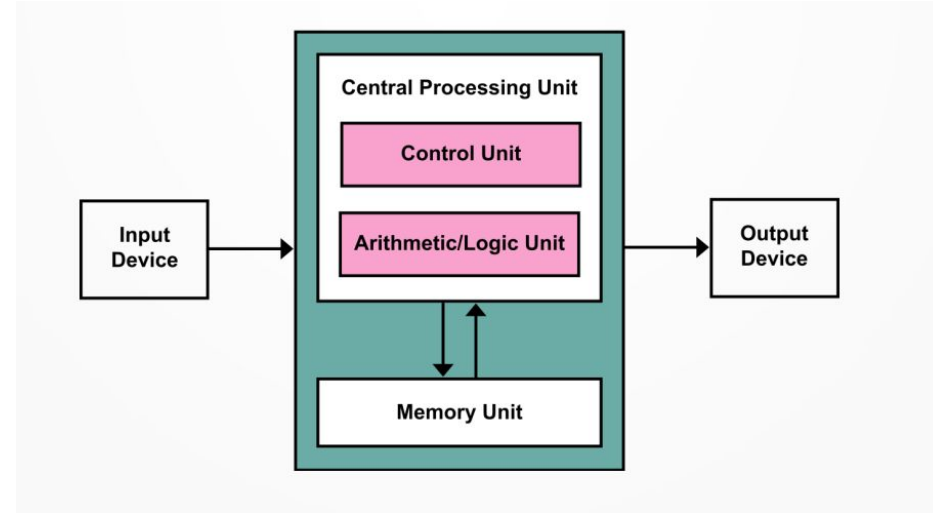
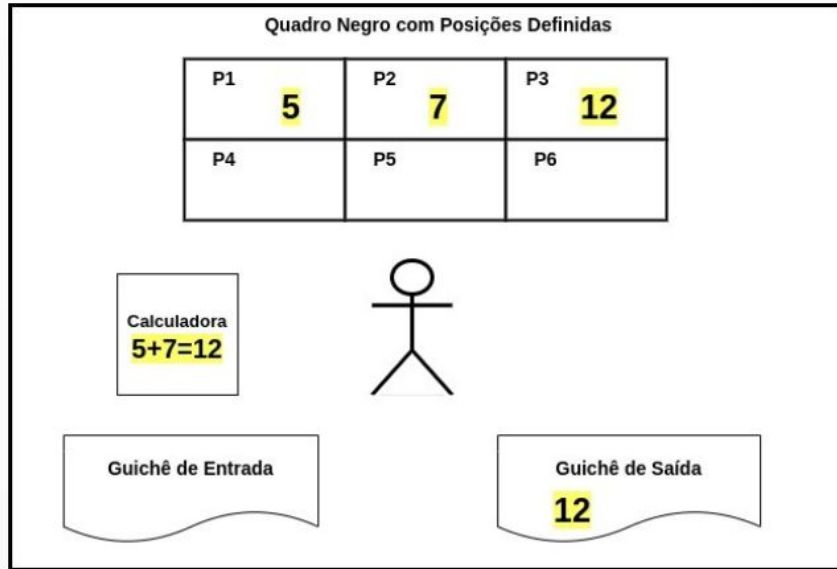
- Agora produza um relatório no guichê de saída informando o resultado da soma



- Agora produza um relatório no guichê de saída informando o resultado da soma



Lembrando: na realidade isso é uma ilustração do que realmente ocorre



- Depois que termina a execução de um programa *limpa tudo antes de começar o seguinte*
 - Leia um número do guichê de entrada e escreva na posição P1 do quadro negro
 - Leia mais um número do guichê de entrada e escreva na posição P2 do quadro negro
 - Subtraia o número na posição P2 do número na posição P1 e escreva o resultado na posição P3
 - Agora produza um relatório no guichê de saída informando o resultado da subtração

Quadro Negro com Posições Definidas

P1 10	P2 4	P3 6
P4	P5	P6

Calculadora
10-4=6



Guichê de Entrada

Guichê de Saída

6

Vamos melhorar a forma de escrever os algoritmos: mais ágil!

- Algoritmo:
 - **Leia(P1)** {ao invés de “leia um número inteiro do guichê de entrada e armazene na posição P1”}
 - **Leia(P2)** {ao invés de “leia um número inteiro do guichê de entrada e armazene na posição P2”}
 - **P3 ← P1+P2** {ao invés de “faça a soma dos valores nas posições P1 e P2 na calculadora e armazene o resultado na posição P3”}
 - **Escreva(P3)** {ao invés de “informe o resultado da soma que está na posição P3 no guichê de saída”}

Veja como ficou o Algoritmo 1:

- Leia(P1)
- Leia(P2)
- $P3 \leftarrow P1 + P2$
- Escreva(P3)

Exercício: re-escreva de forma sintética o Algoritmo 2

- Leia um número do guichê de entrada e escreva na posição P1 do quadro negro
- Leia mais um número do guichê de entrada e escreva na posição P2 do quadro negro
- Subtraia o número na posição P2 do número na posição P1 e escreva o resultado na posição P3
- Agora produza um relatório no guichê de saída informando o resultado da subtração

Resultado do Exercício

Resultado do Exercício: Algoritmo 2

- Leia(P1)
- Leia(P2)
- $P3 \leftarrow P1 - P2$
- Escreva(P3)

Próximo Exercício

- Escreva um algoritmo para ler 2 números inteiros do guichê de entrada, somá-los e depois ler um 3º número inteiro, subtrai-lo da soma, e finalmente escrever o resultado no guichê de saída.

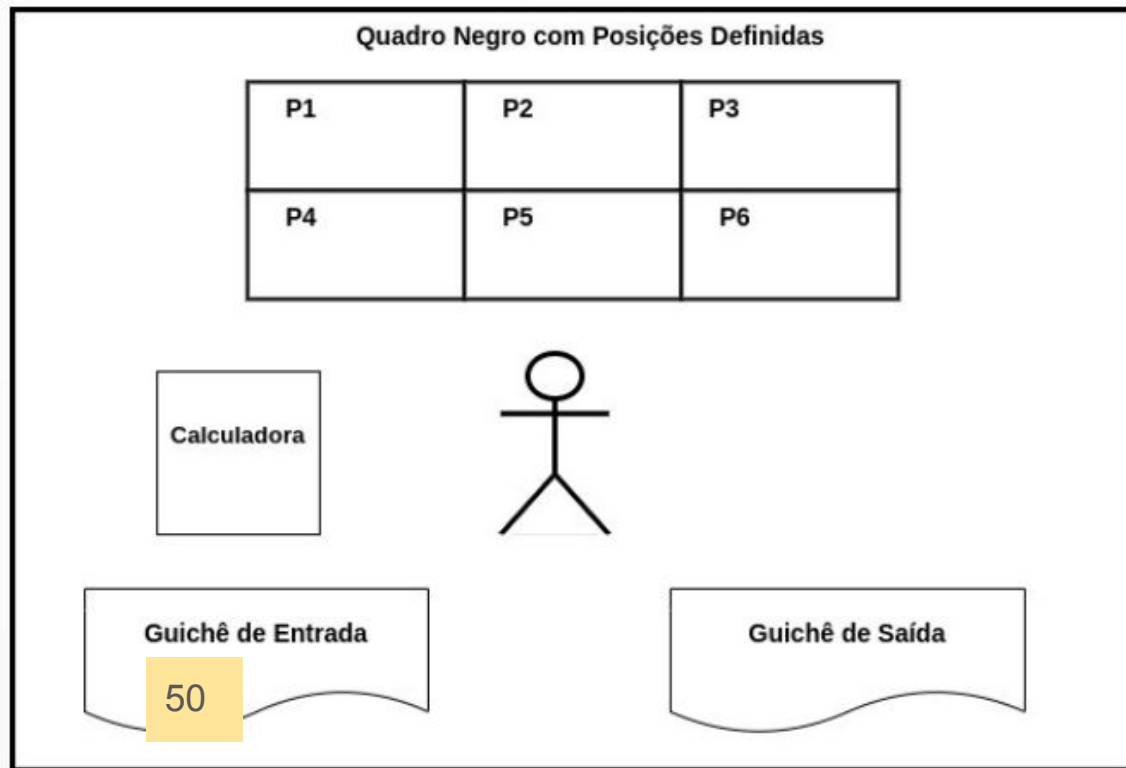
Algoritmo 3: Solução

- Leia(P1)
- Leia(P2)
- $P3 \leftarrow P1 + P2$
- Leia(P4)
- $P5 \leftarrow P3 - P4$
- Escreva(P5)

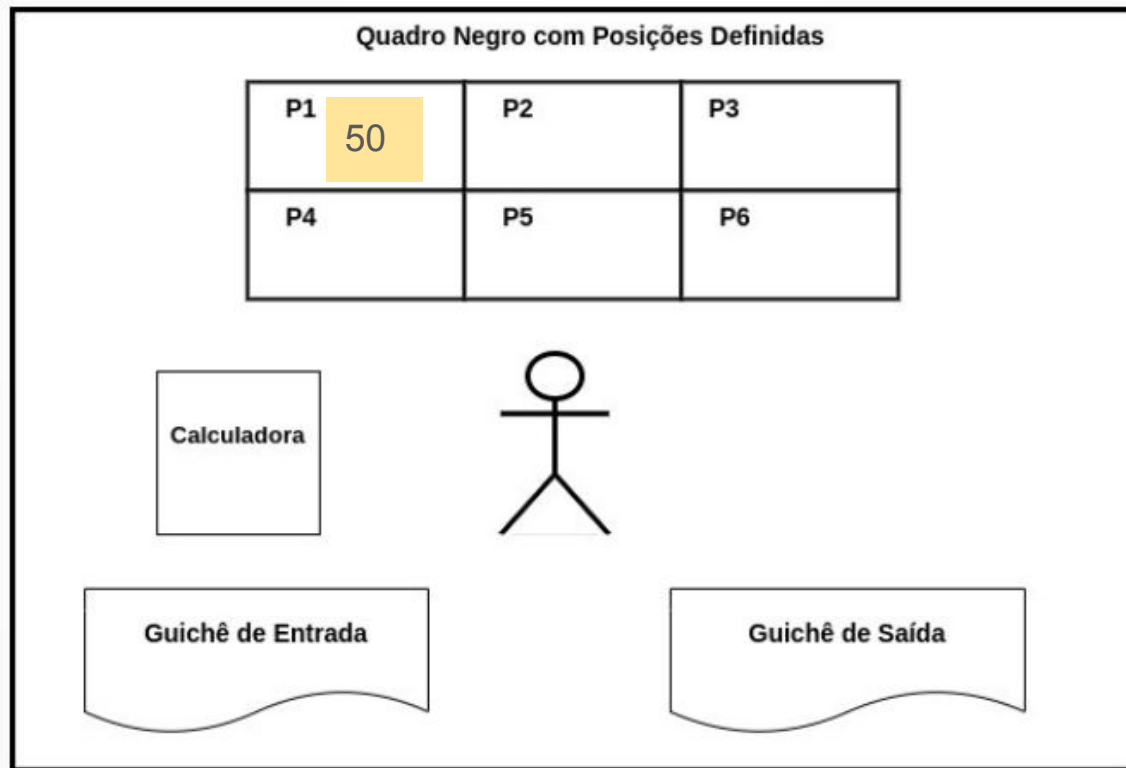
Algoritmo 4

- Leia(P1)
- Leia(P2)
- se $P1 > P2$ então Escreva(P1), senão Escreva(P2)

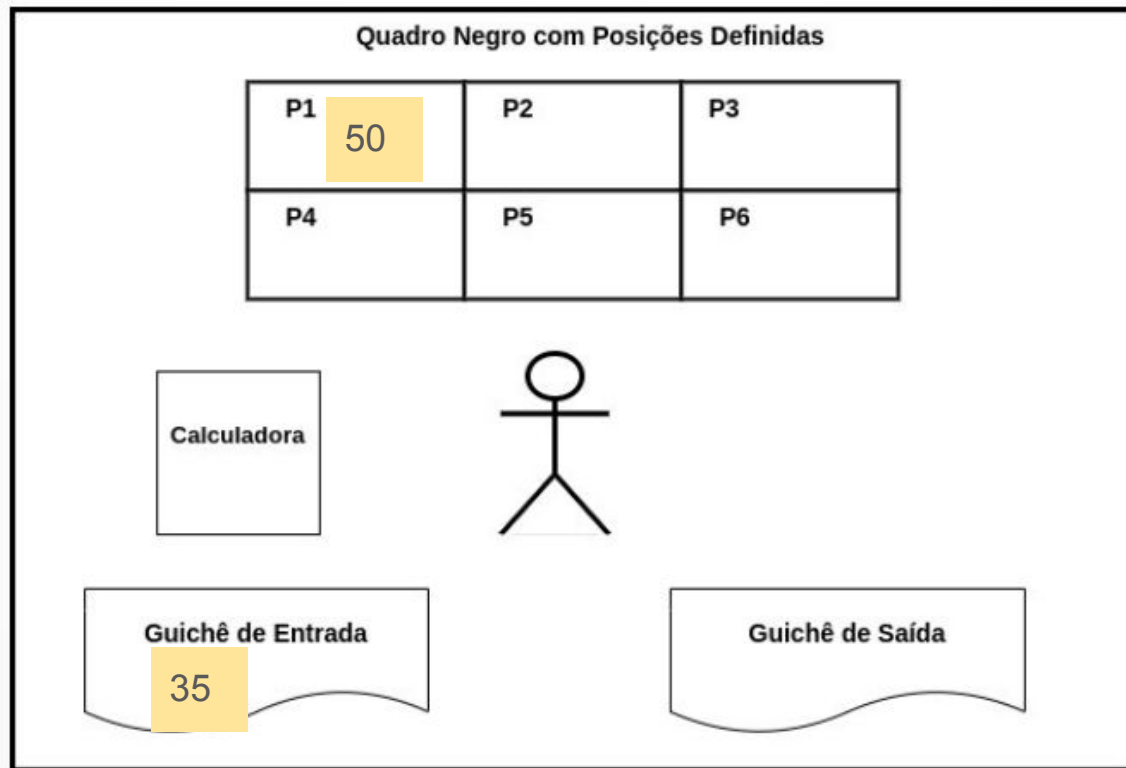
- Leia(P1)



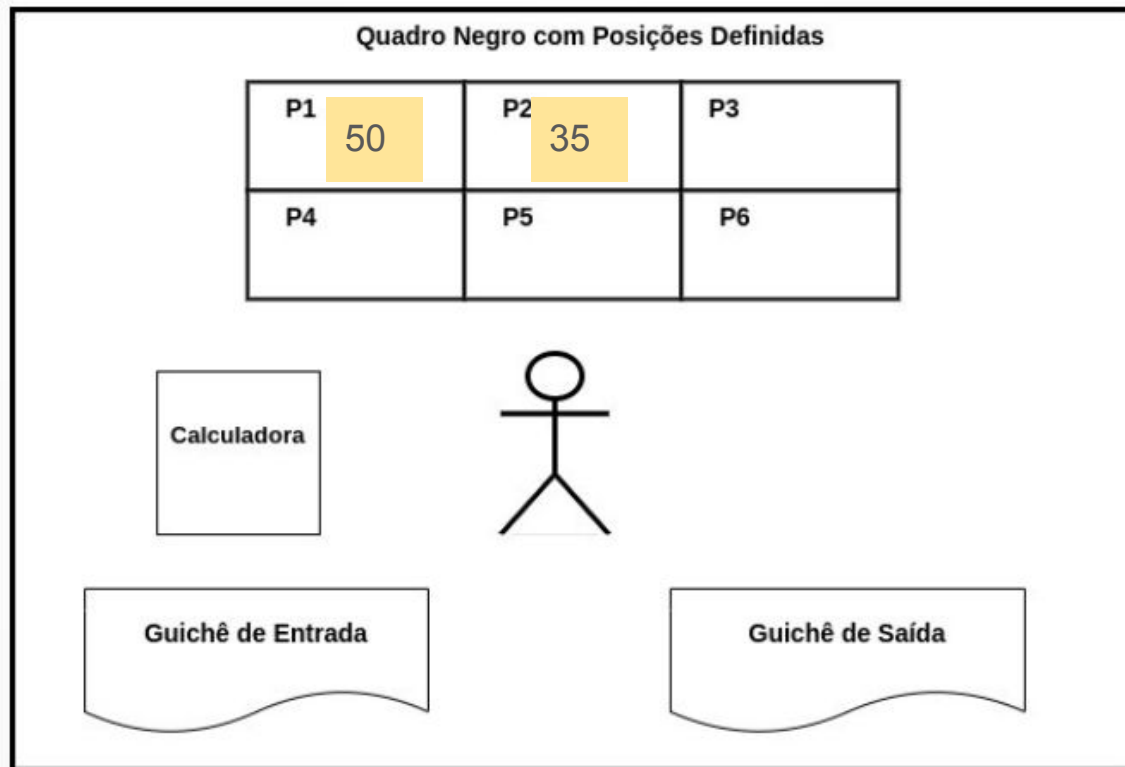
- Leia(P1)



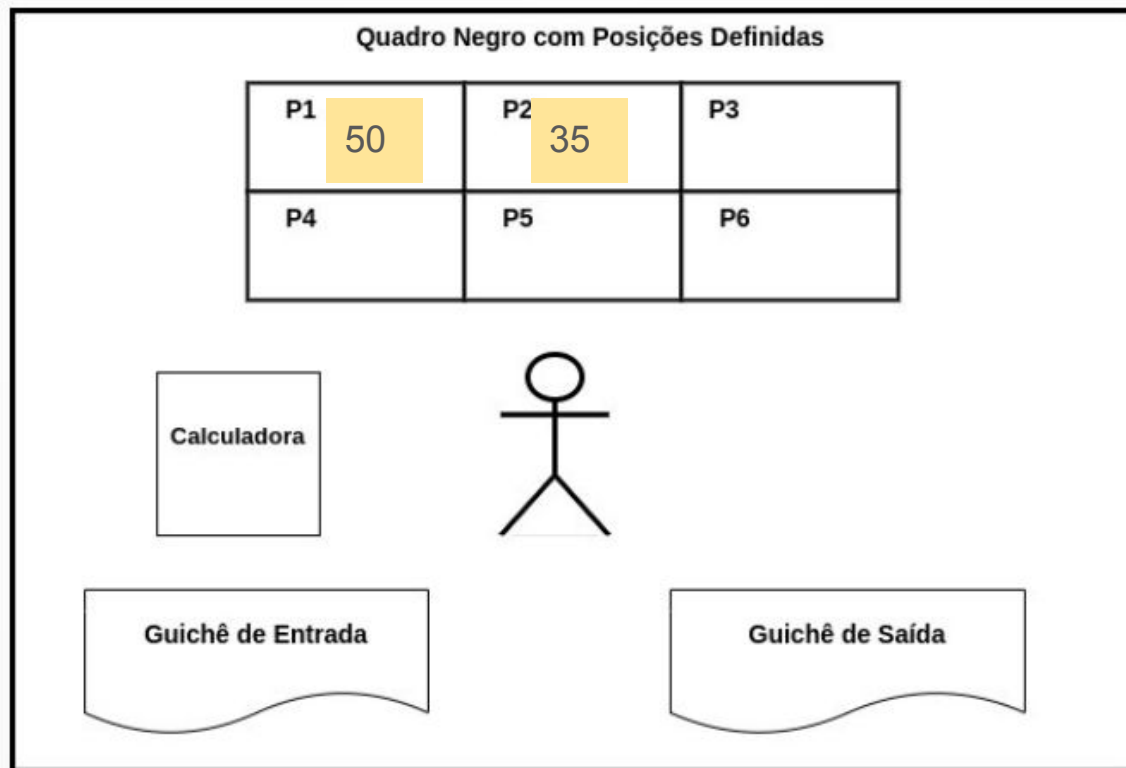
- Leia(P2)



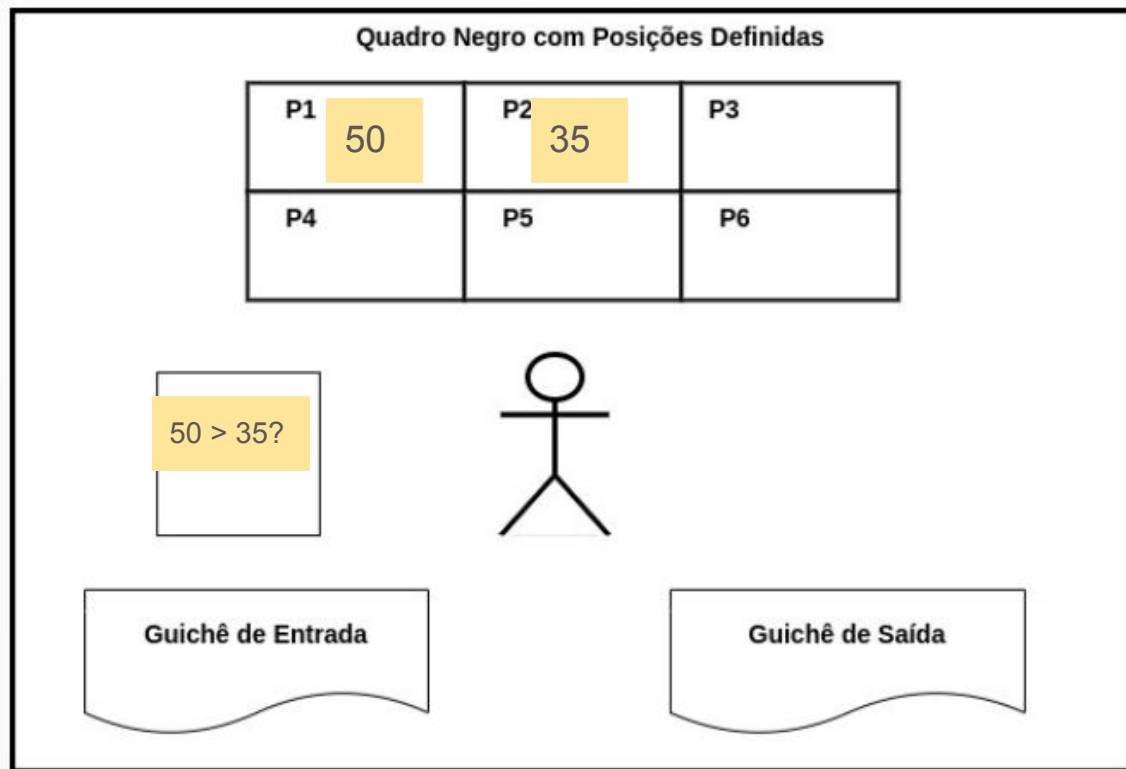
- Leia(P2)



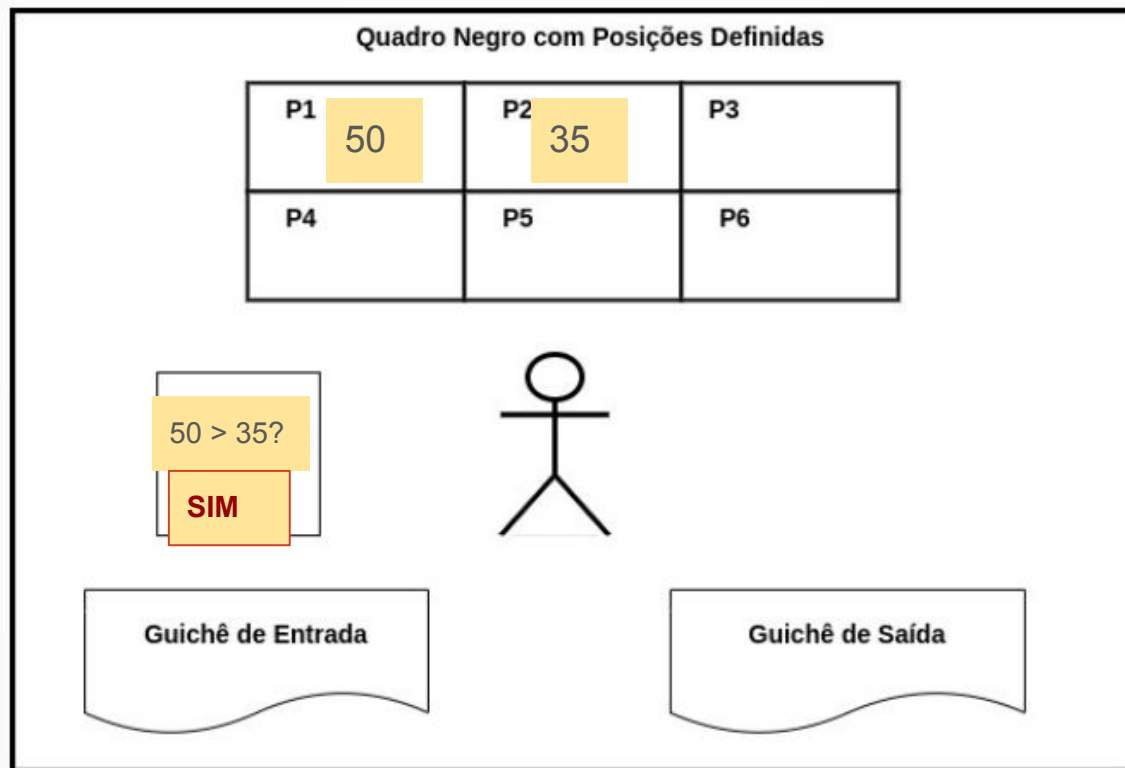
- se $P1 > P2$ então Escreva(P1), senão Escreva(P2)



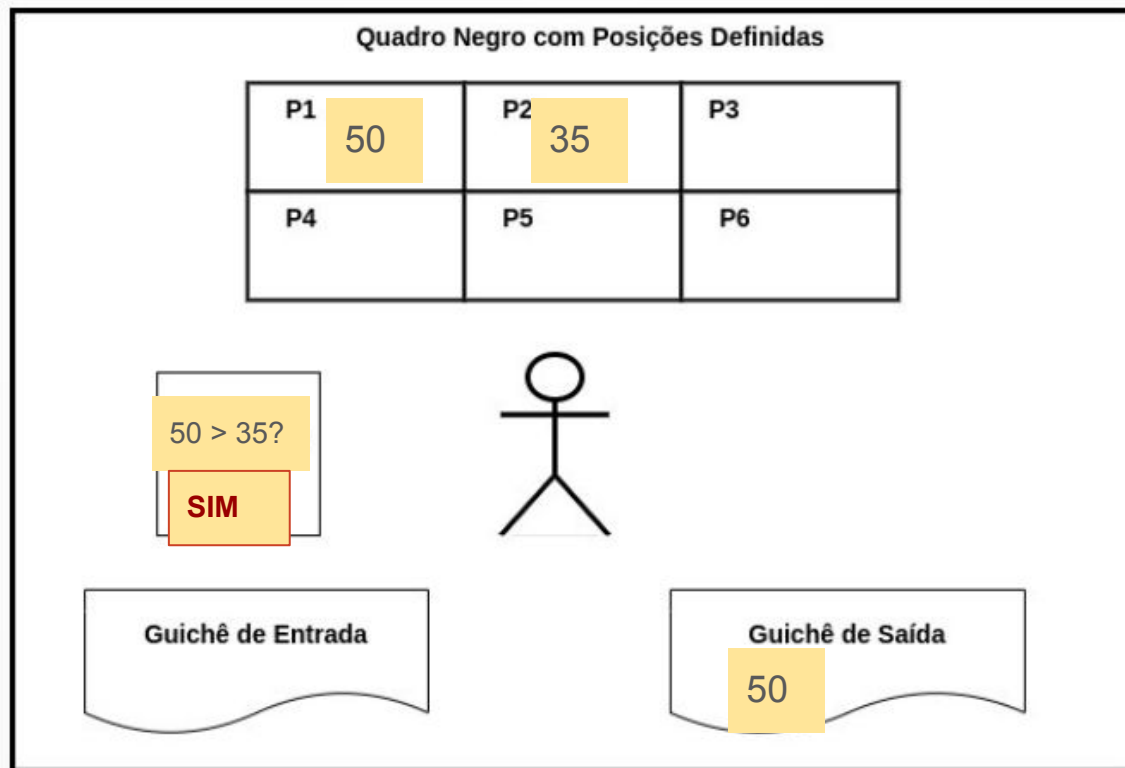
- se $P1 > P2$ então Escreva(P1), senão Escreva(P2)



- se $P1 > P2$ então Escreva(P1), senão Escreva(P2)



- se $P1 > P2$ então Escreva(P1), senão Escreva(P2)



Exercício

Escreva um algoritmo para ler um número da entrada e escrever na saída se o número lido é “par” ou “impar”.

Dica: a calculadora tem uma função *mod* que retorna o resto da divisão de dois números naturais

Solução: Algoritmo 5

- Leia(P1)
- $P2 \leftarrow P1 \bmod 2$
- se $P2 = 0$ então Escreva("Par"); senão Escreva("Ímpar")