



Software de cálculo de posição e velocidade

Diagrama de casos de uso

Geral

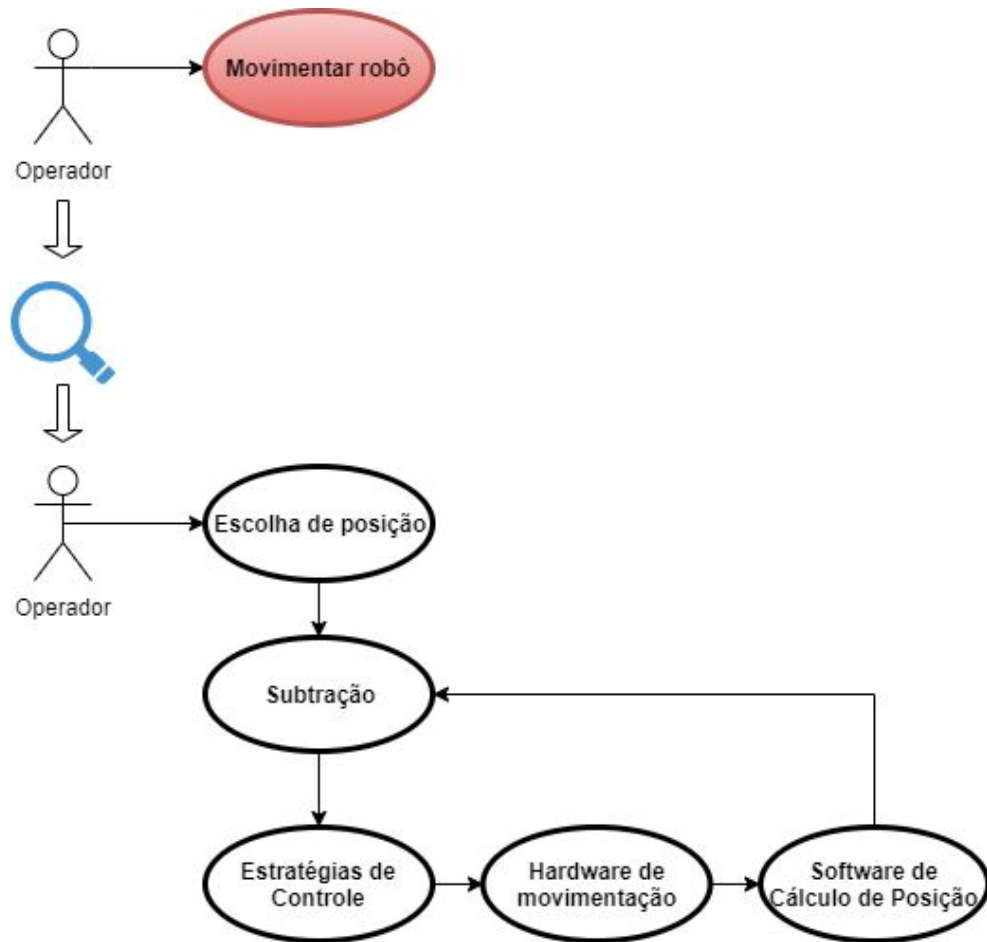
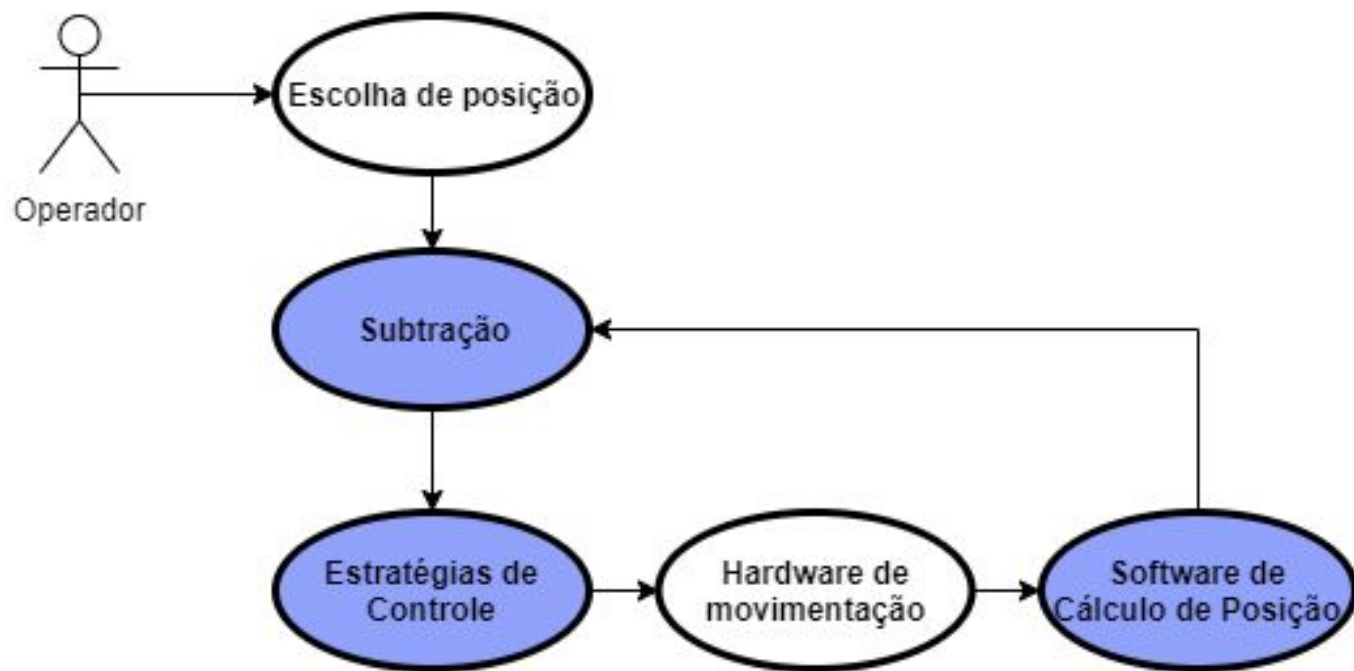


Diagrama de casos de uso

Específico



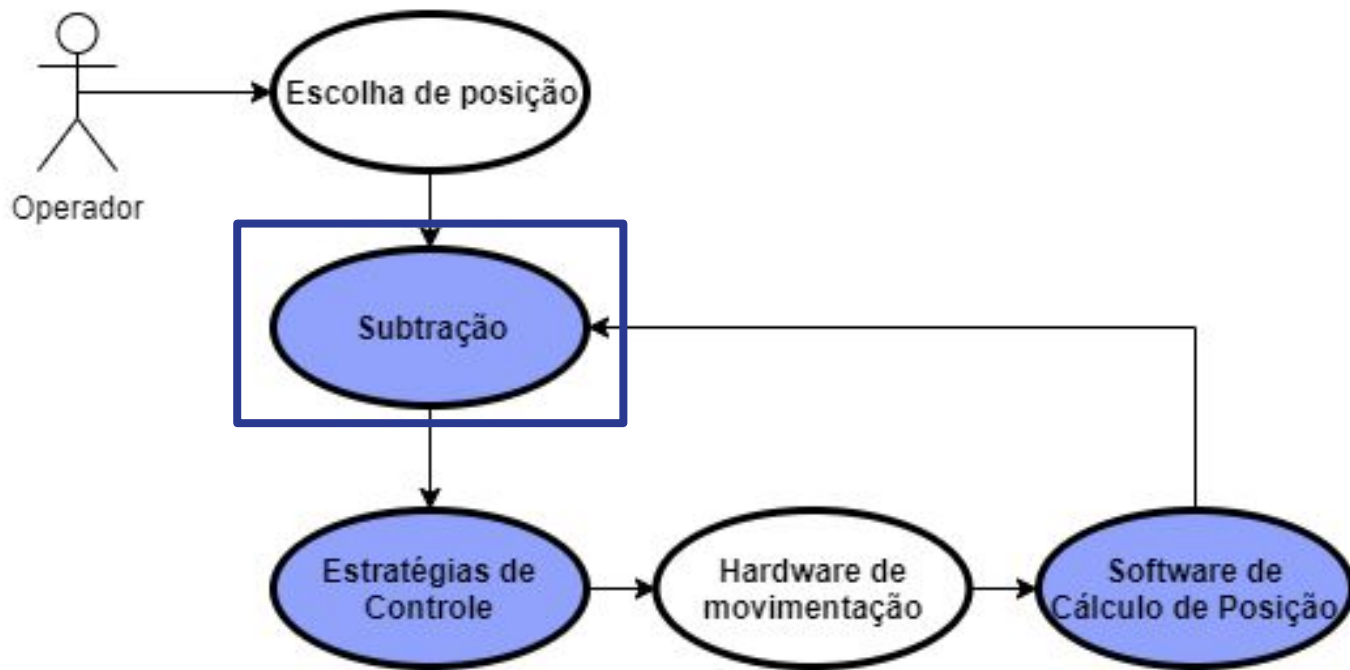


Diagrama de casos de uso - Subtração

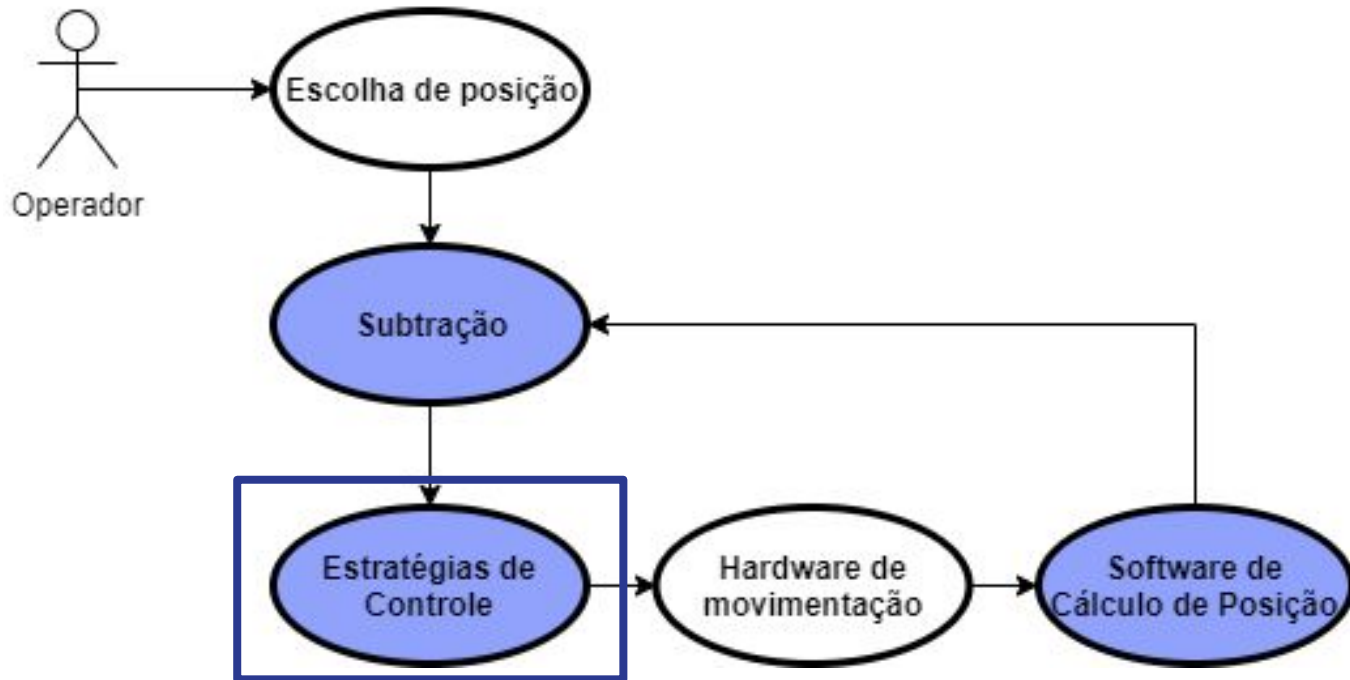


Diagrama de casos de uso - Estratégias de Controle

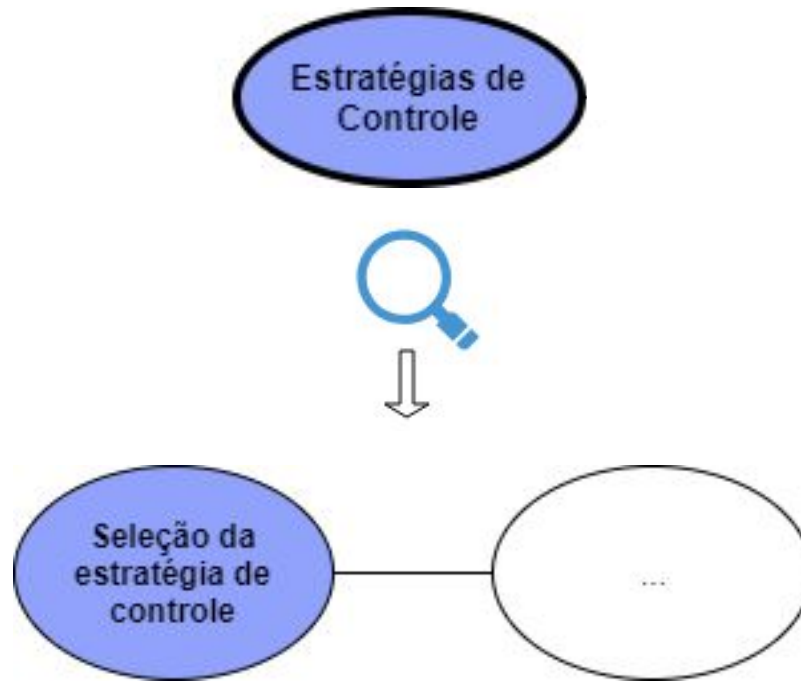


Diagrama de casos de uso - Estratégias de controle em detalhe

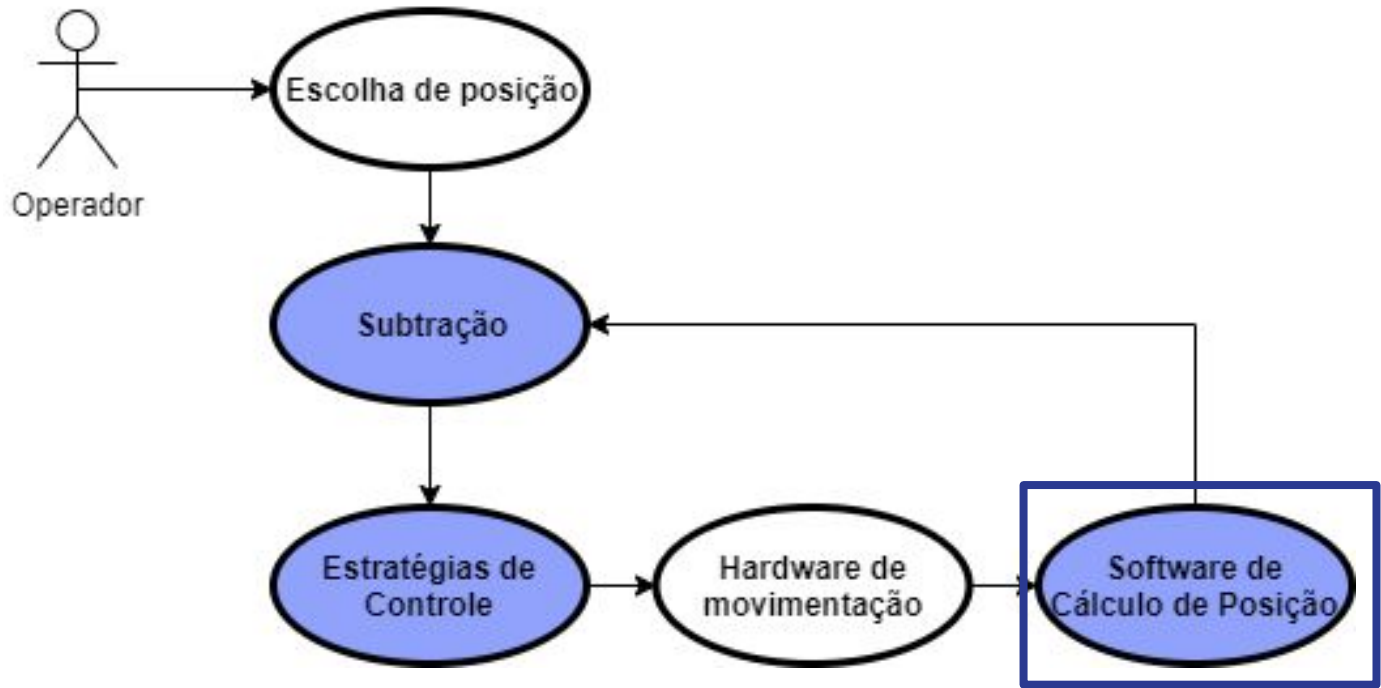


Diagrama de casos de uso - Cálculo de posição e velocidade

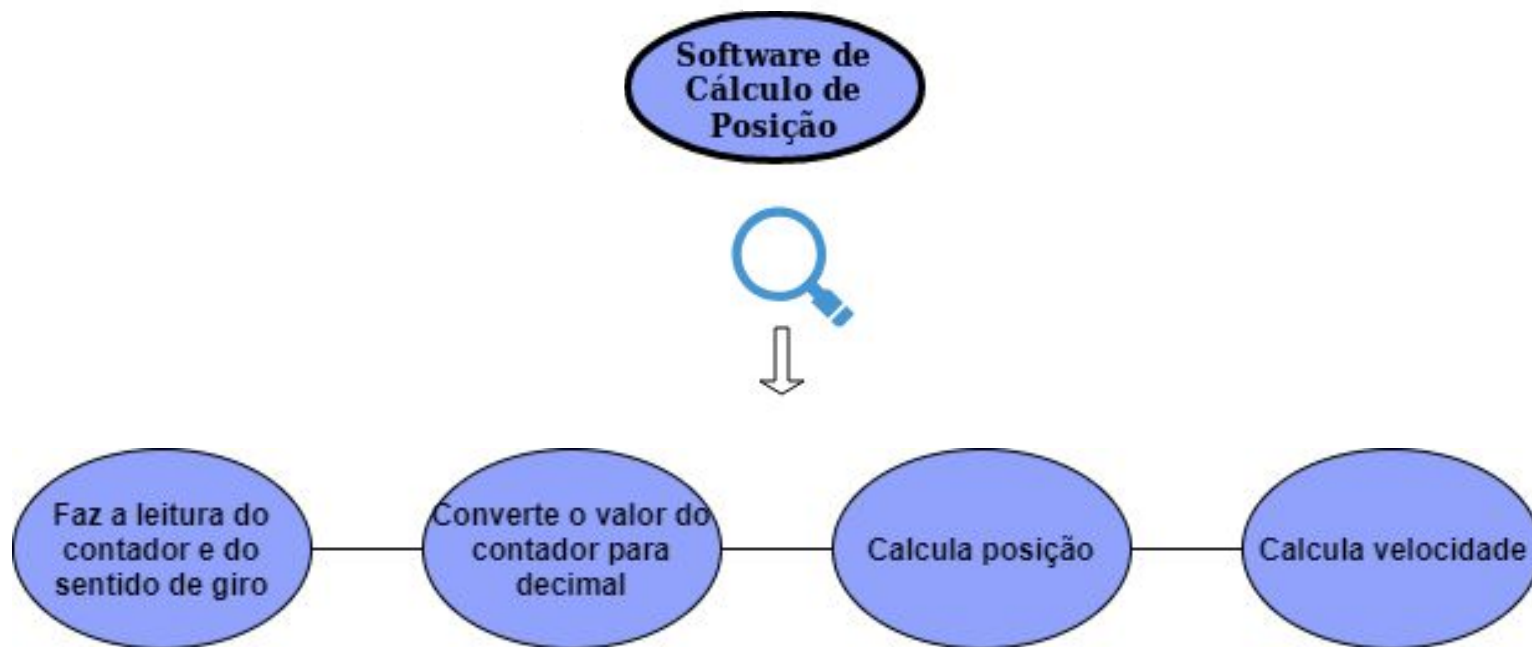


Diagrama de casos de uso - Cálculo de posição e velocidade em detalhe

Diagrama de classe

CalculaPosVel

- contagem : int
- sentido: int
- posicao: double
- velocidade: double

- + getSentido(): int
- + getPosicao(): double
- + getVelocidade(): double

contagem

Valor da contagem do contador

sentido

1 - Horário

-1 - Anti-horário

posicao

Valor da posição em mm

velocidade

Valor da velocidade em mm/s

getPosicao()

Atualiza e retorna o valor de posicao

getVelocidade()

Atualiza e retorna o valor de velocidade

CalculaPosVel	
<ul style="list-style-type: none"> - contagem: int - contagemAnterior: int - pinos: int* - sentidoPin: int - sentido: int - posicao: double - posicaoAnterior: double - velocidade: double - tempoAtual: double - tempoAnterior: double 	
<ul style="list-style-type: none"> + getSentido(): int + getPosicao(): double + getVelocidade(): double 	

Diagrama de sequência

Geral

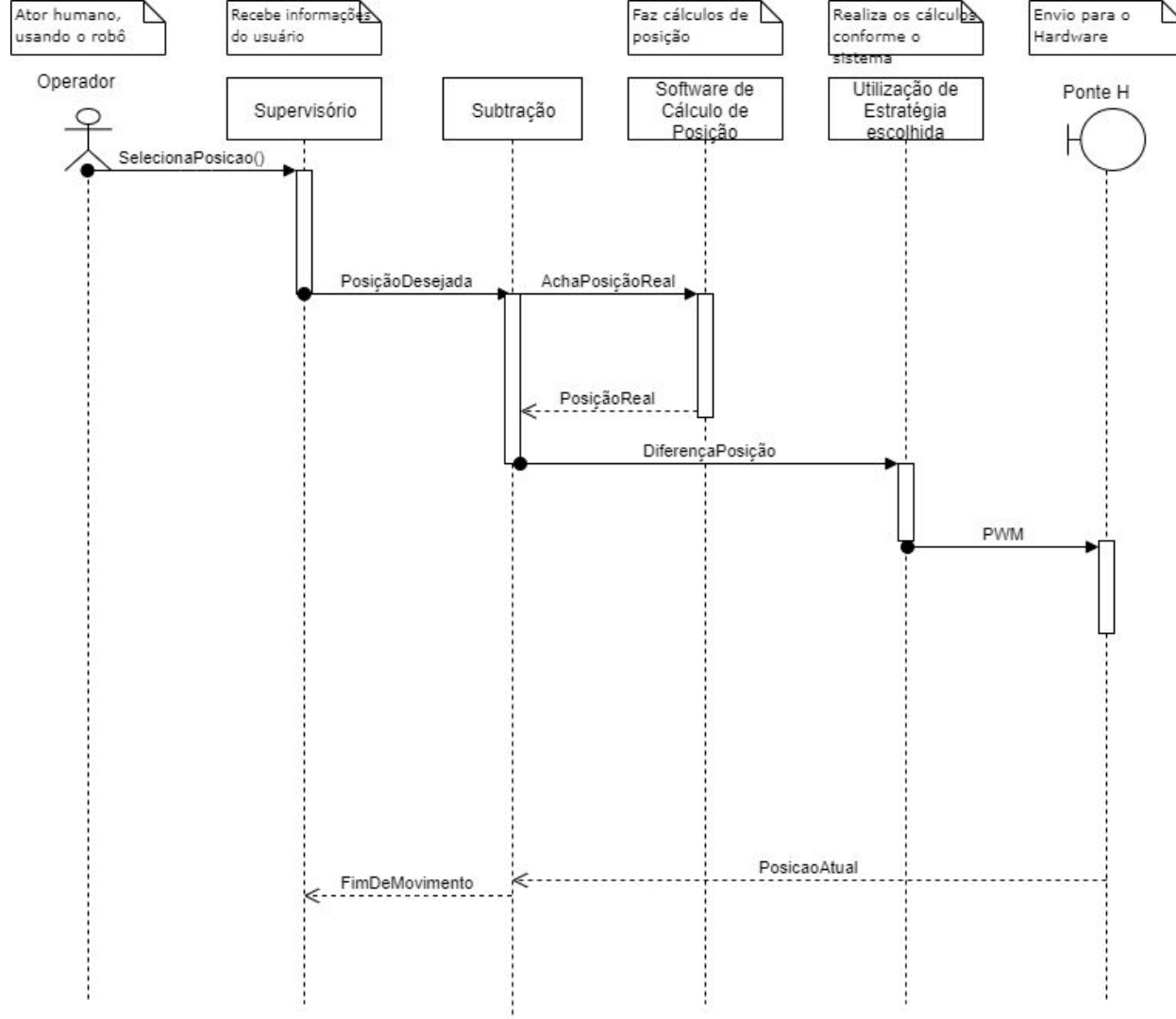


Diagrama de sequência

Específico

Ator humano, usando o robô

Recebe informações do usuário

Faz cálculos de posição

Realiza os cálculos conforme o sistema

Envio para o Hardware

Operador

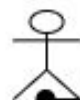
Supervisório

Subtração

Software de Cálculo de Posição

Utilização de Estratégia escolhida

Ponte H



SelecionaPosicao()

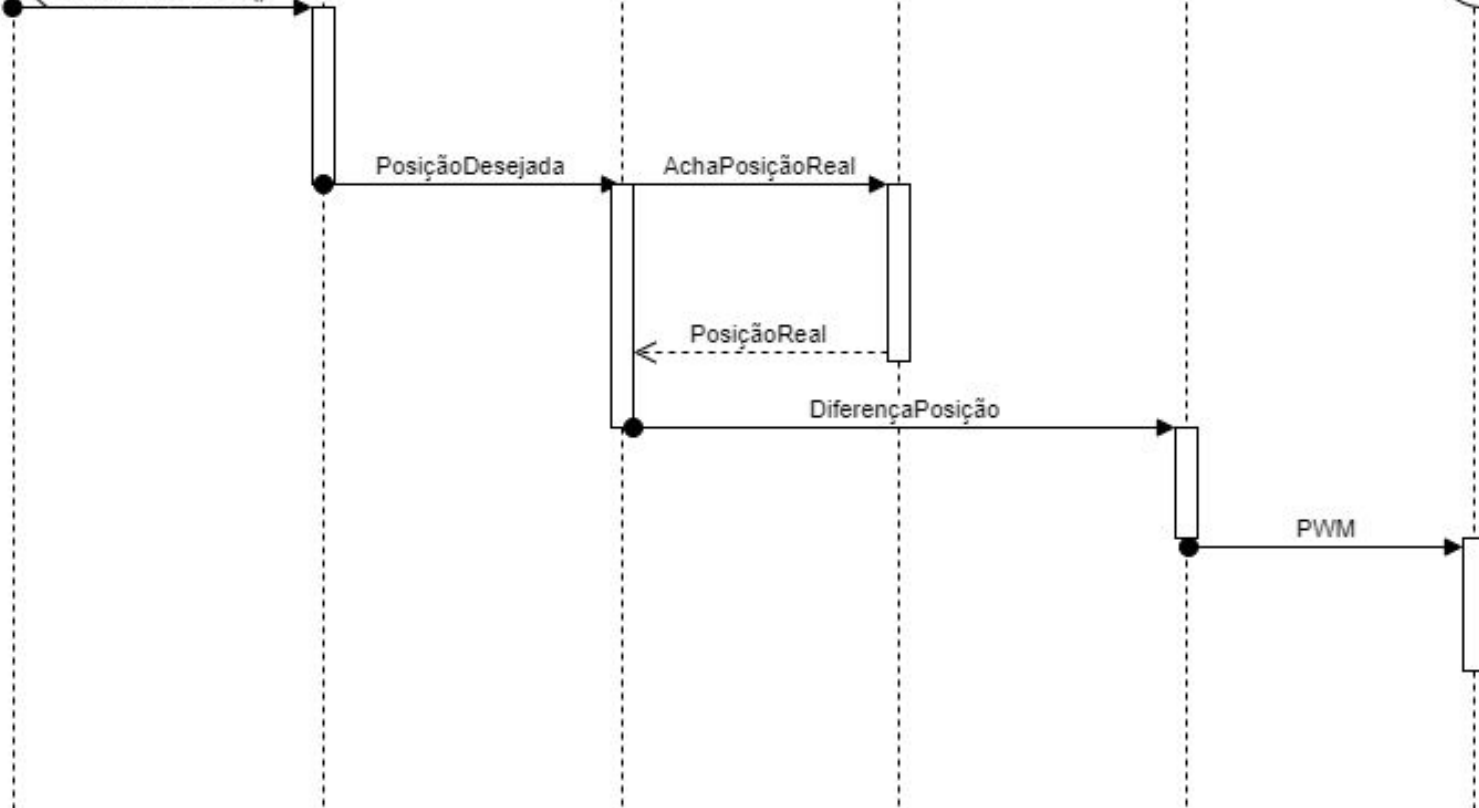
PosiçãoDesejada

AchaPosiçãoReal

PosiçãoReal

DiferençaPosição

PWM



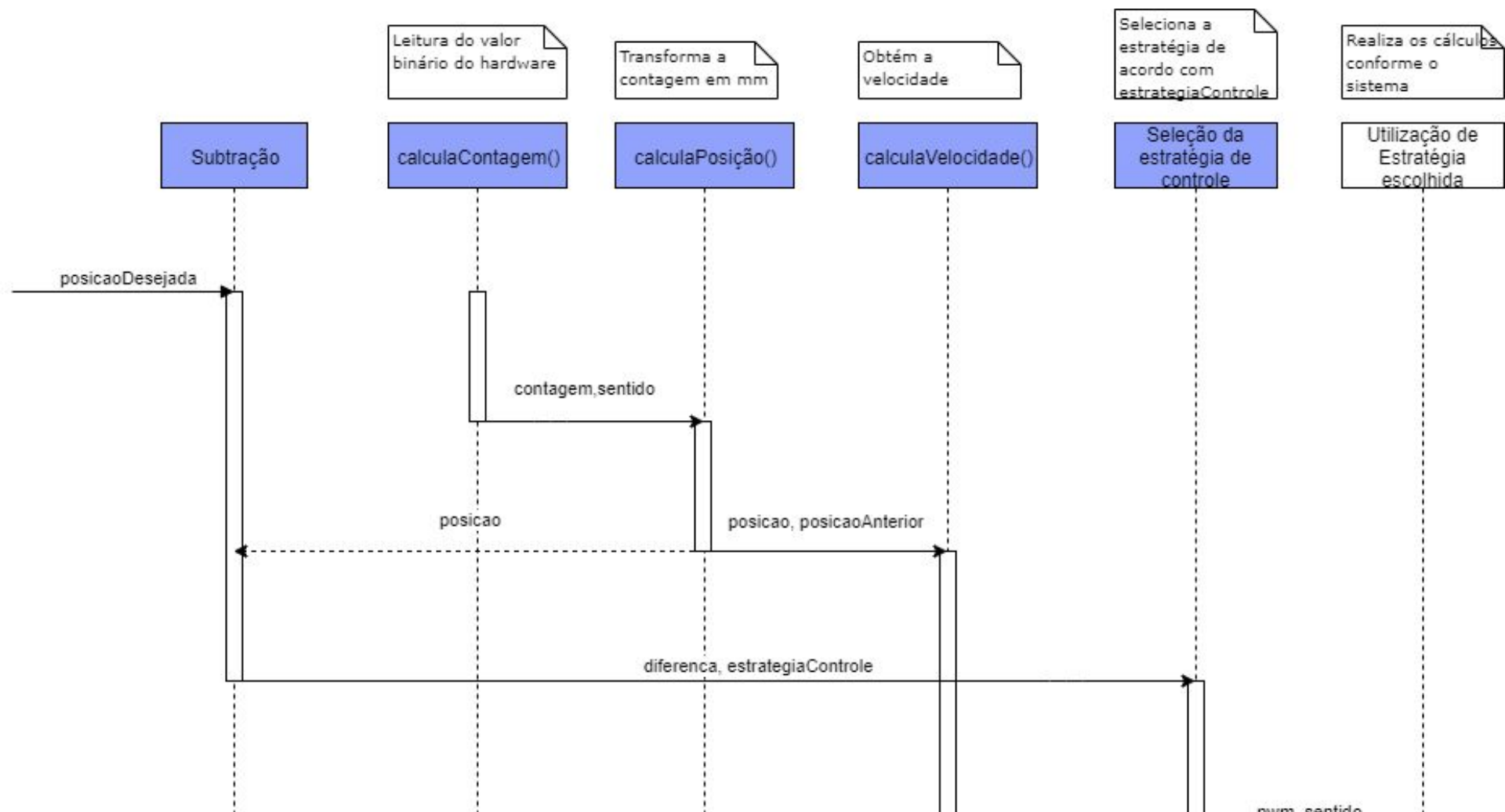


Diagrama de atividades

Geral

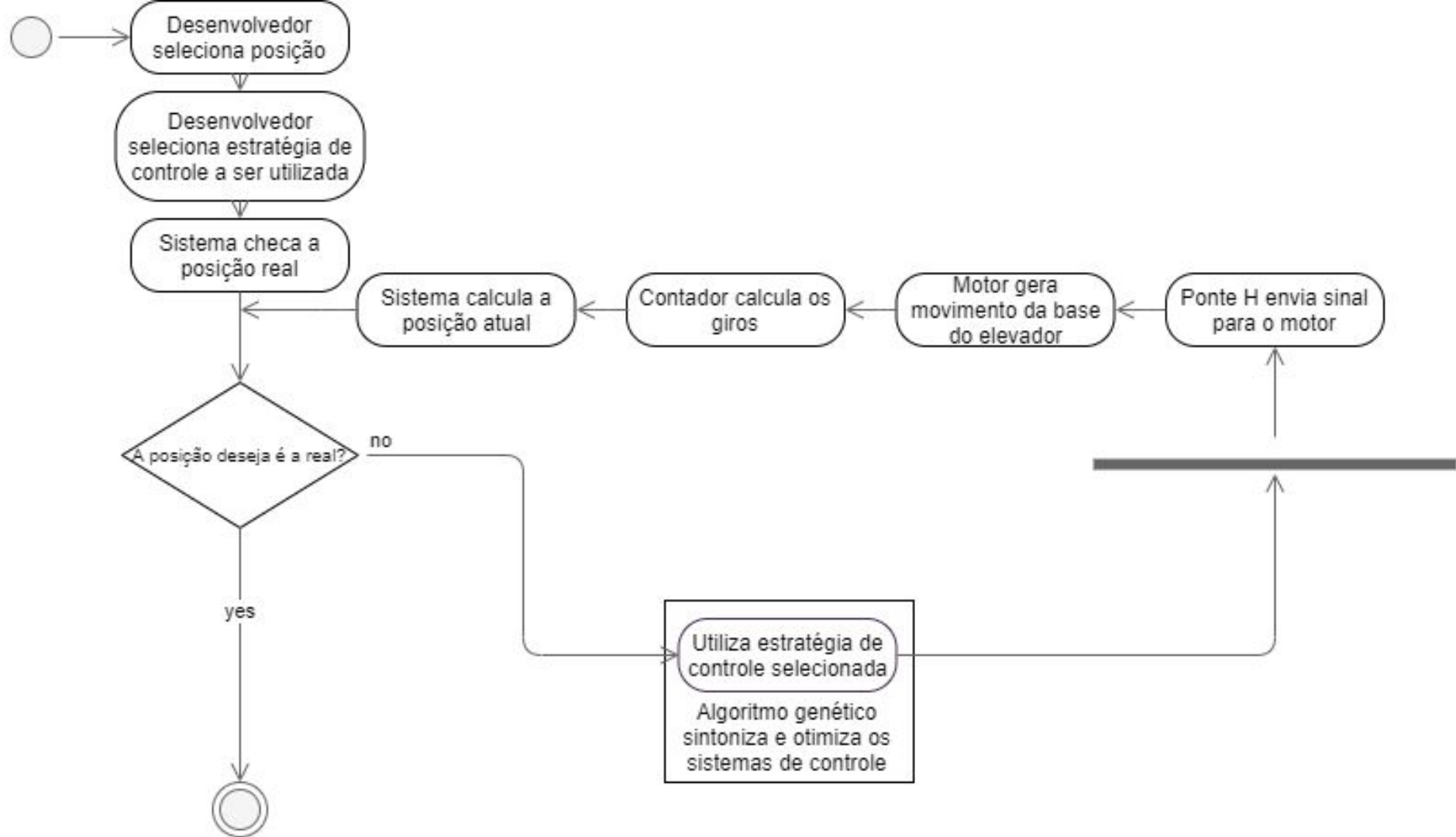
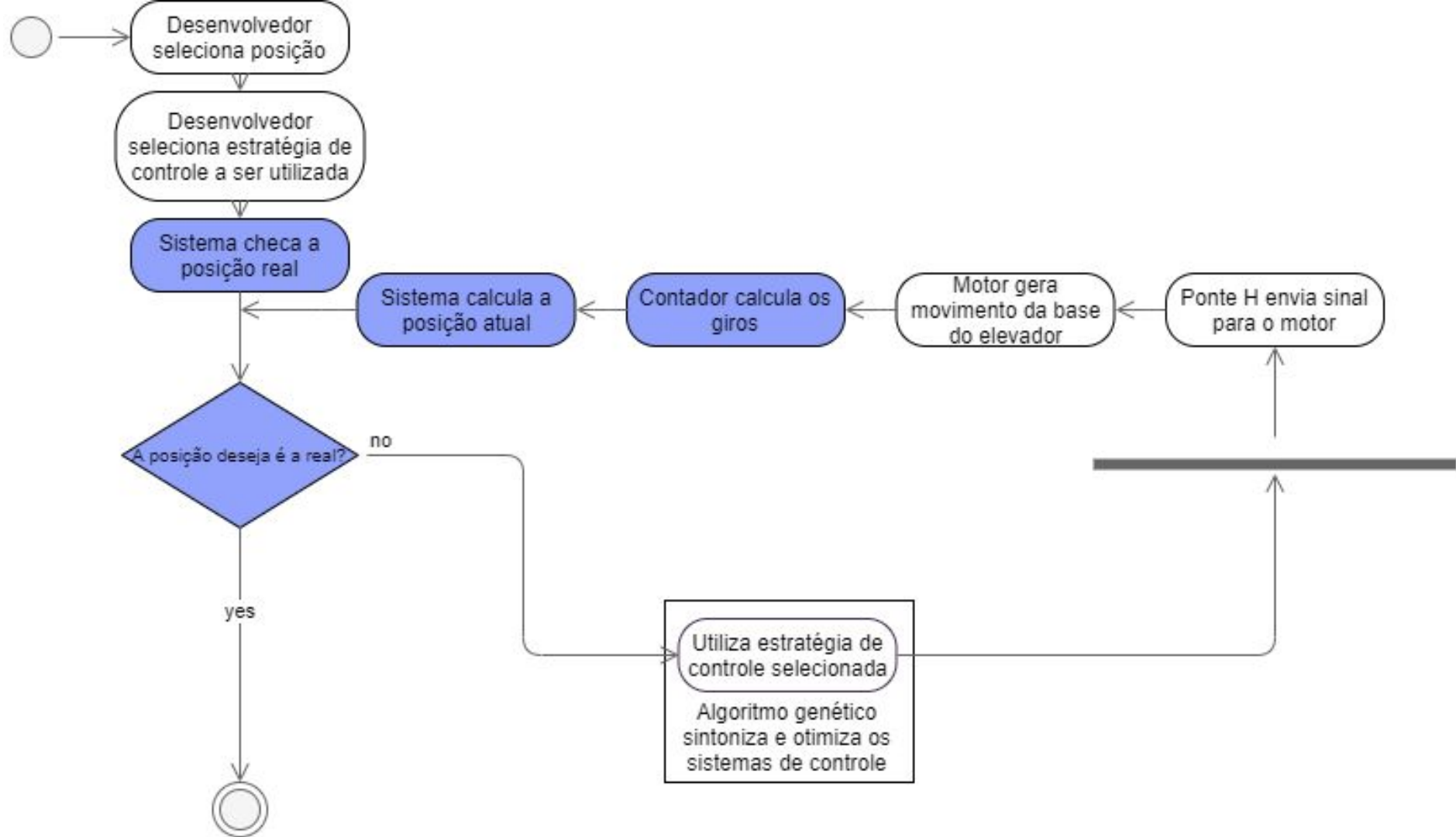
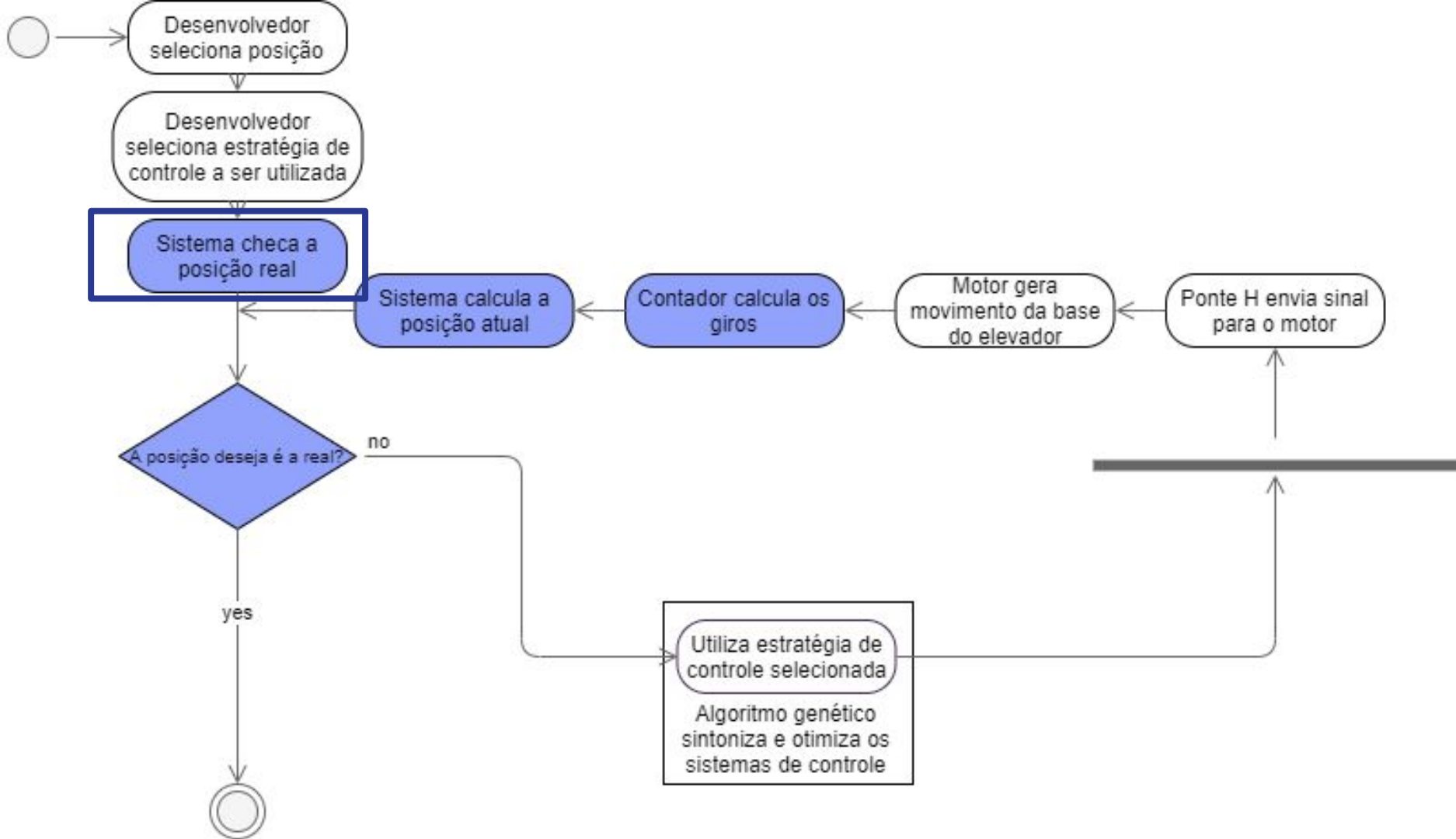


Diagrama de atividades

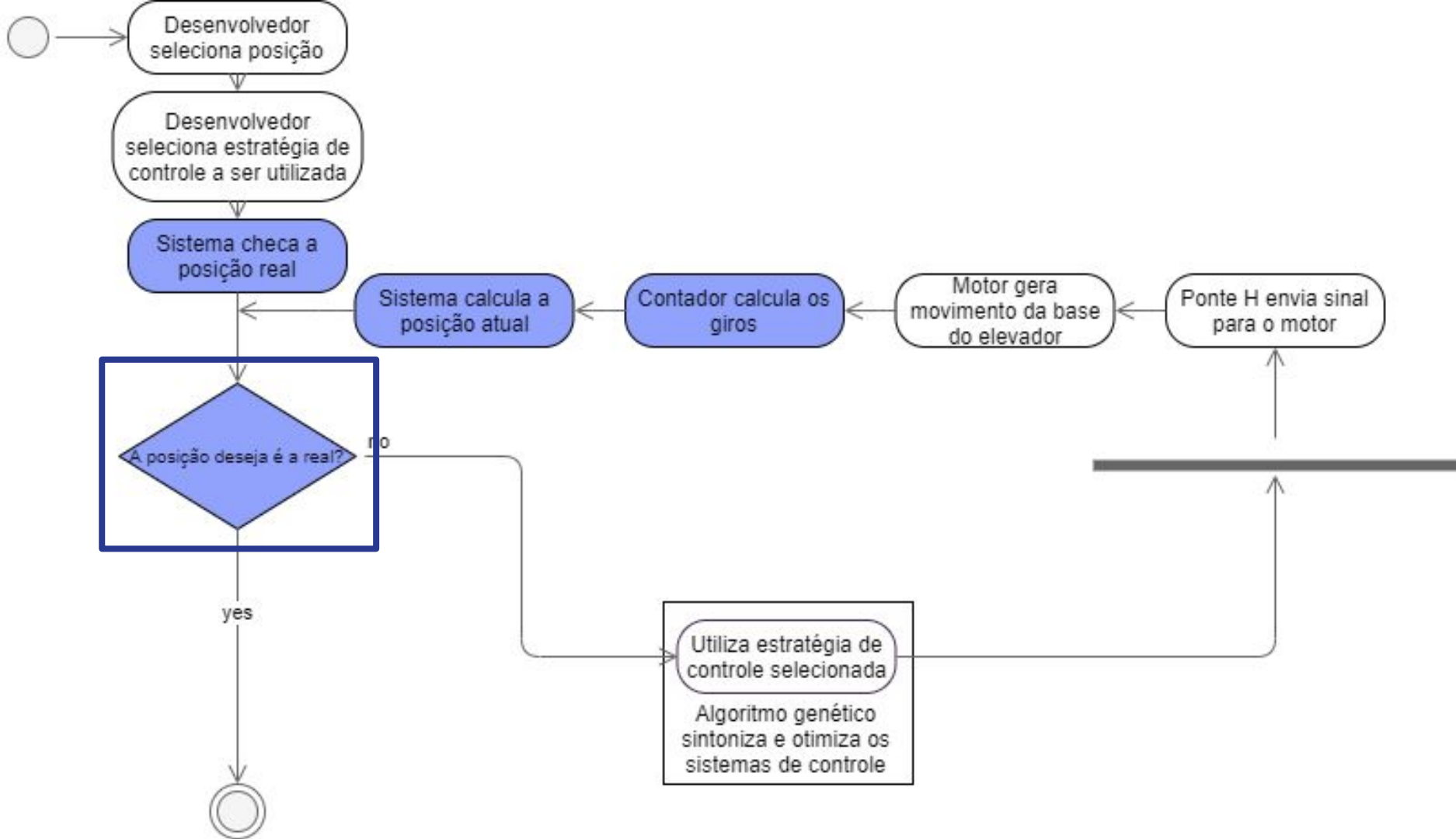
Específico





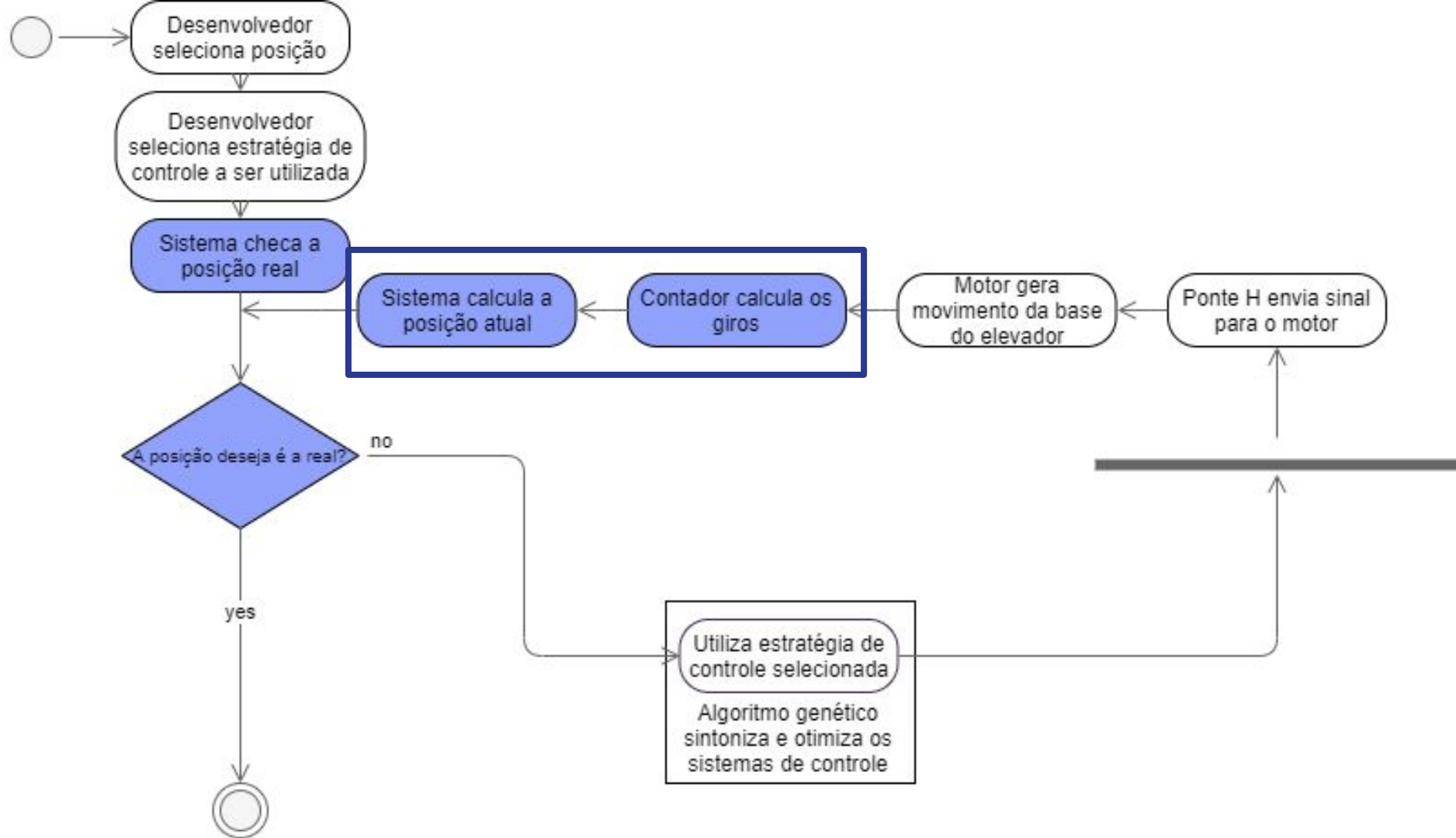
Sistema checa a posição real

```
// Sistema checa a posição real  
double diferenca = PosicaoDesejada - contador.getPosicao();
```

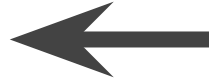



A posição desejada é a real?

```
// A posição real é a desejada?  
if (diferenca  $\neq$  0)  
{  
    int sentido;  
    double PWM;  
    selecionaEstrategia(estrategiaControle, &PWM, &sentido);  
    envia(PWM, sentido);  
}
```



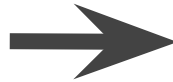
Sistema calcula a
posição atual



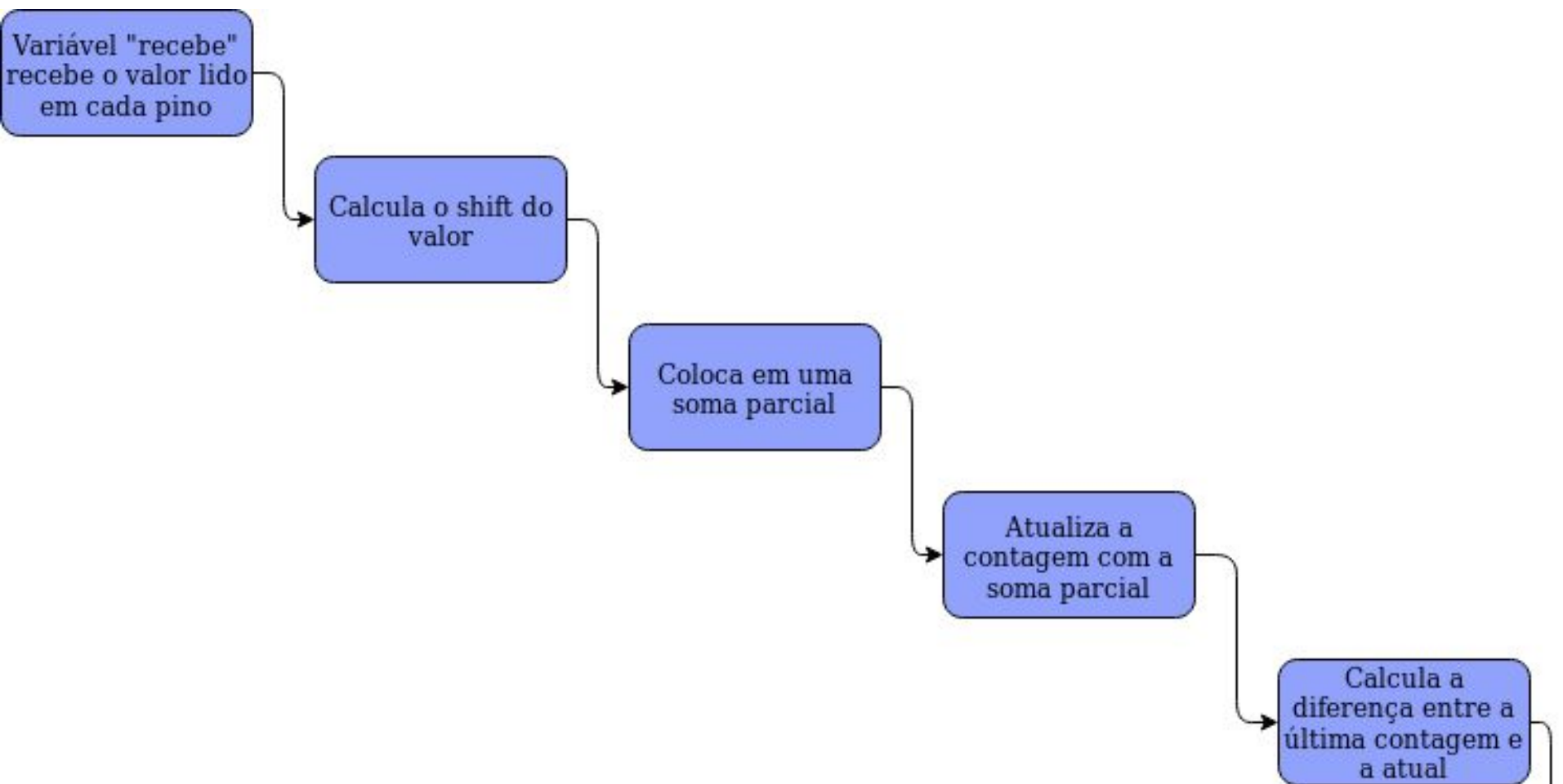
Contador calcula
giros

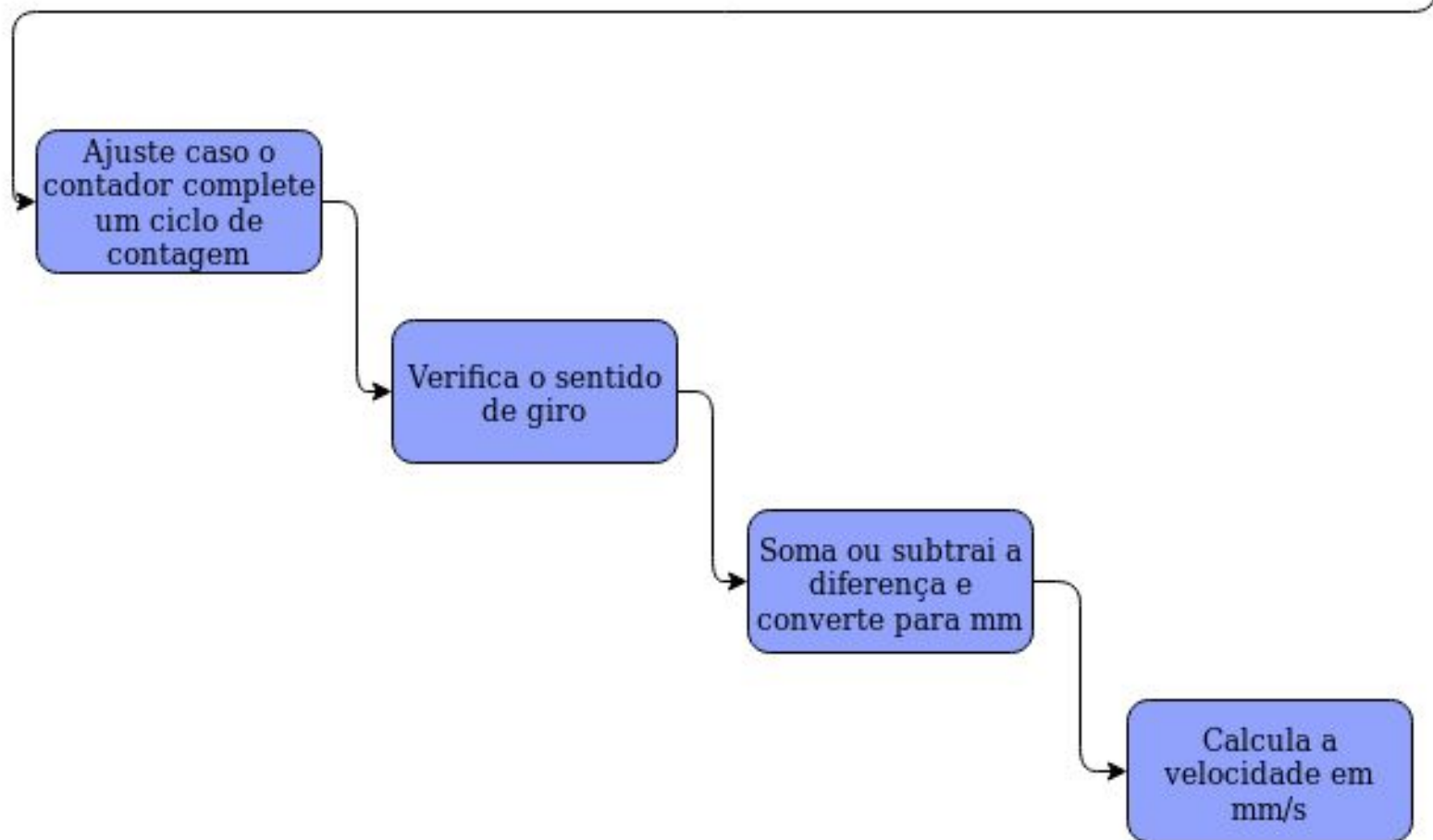
```
// Sistema calcula a posição atual  
double PosicaoReal = contador.getPosicao();
```

```
double Contador::getPosicao()  
{  
    calculaPosicao();  
    return posicao;  
}
```

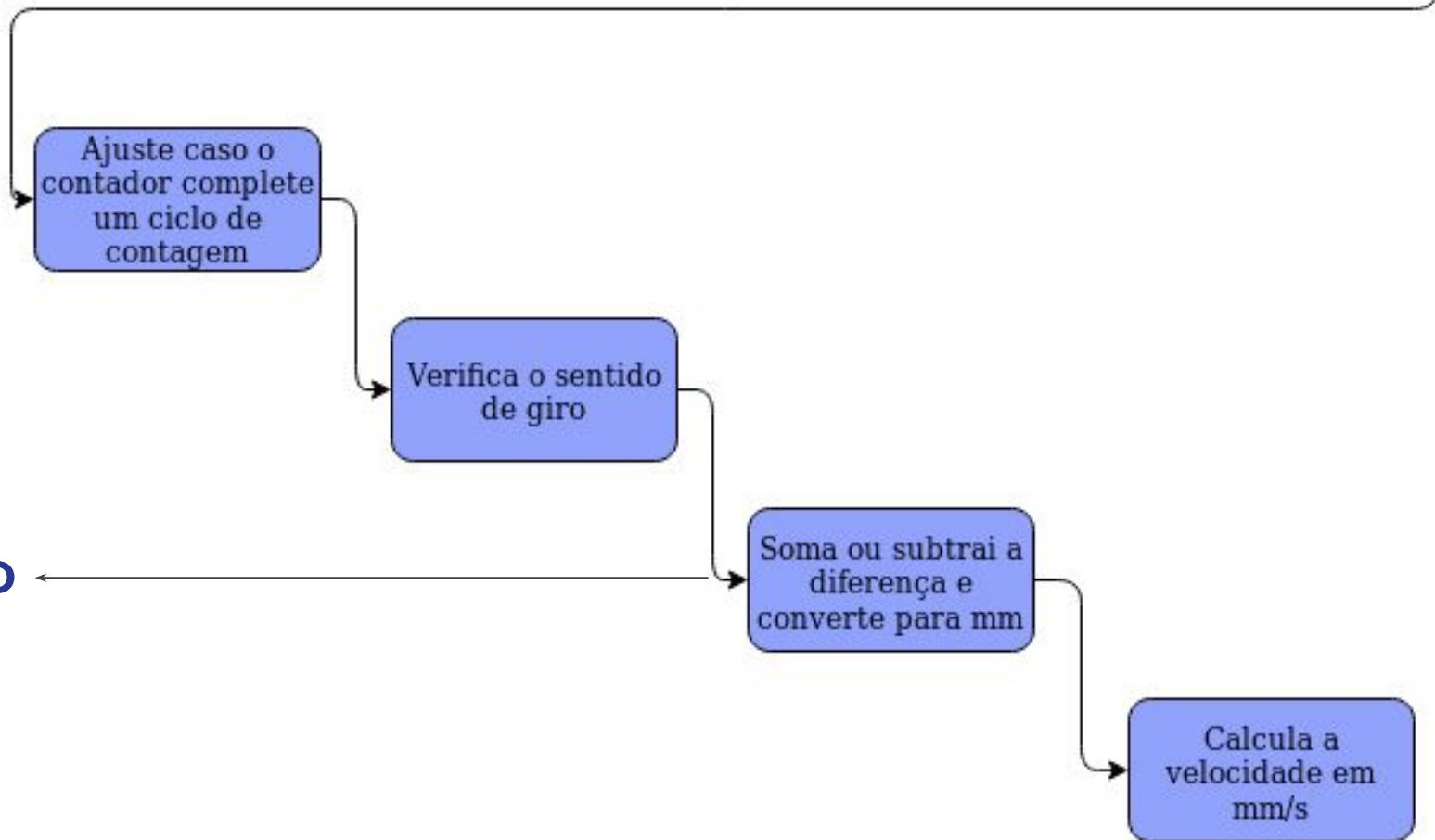


```
void Contador::calculaPosicao()  
{  
    // Contador calcula giros  
    calculaContagem();  
}
```



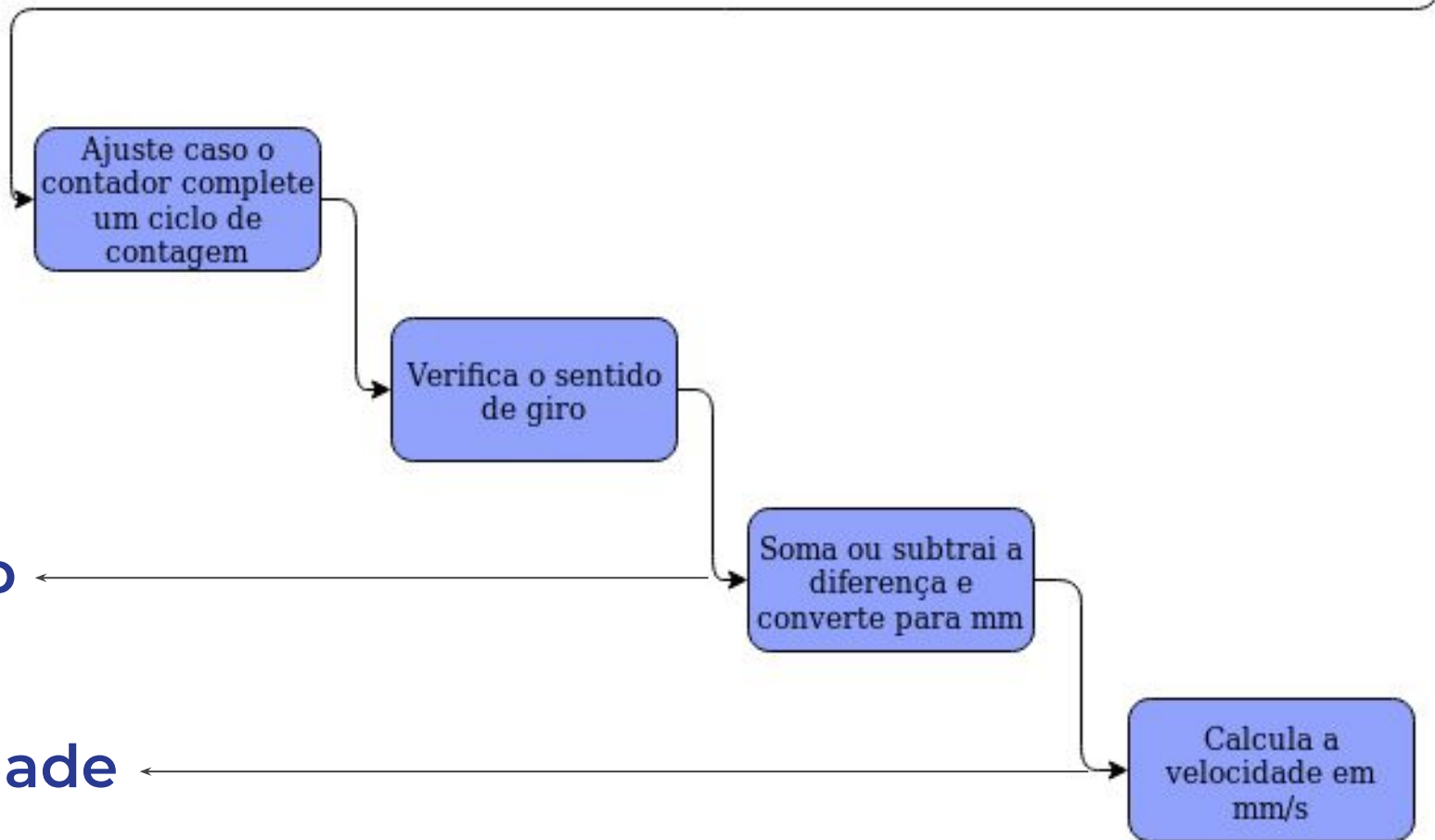


Posição



Posição

Velocidade





Testes

testes.txt



Testes

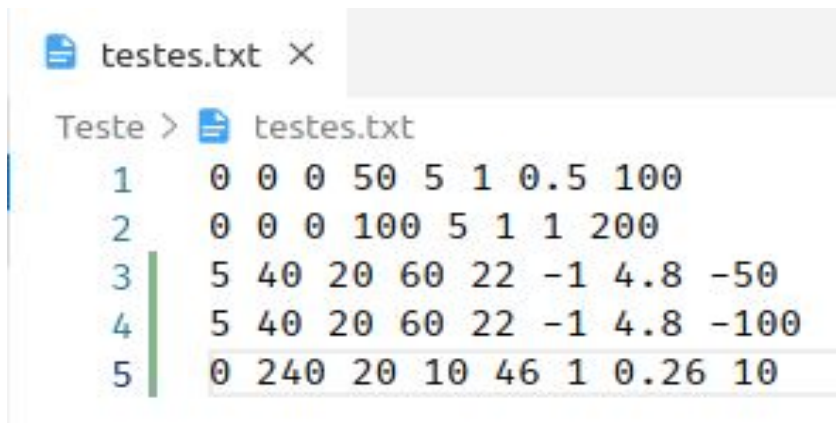


resultados.txt

testes.txt

Como testar?

- Posição inicial
- Contagem anterior
- Tempo anterior
- Contagem atual
- Tempo atual
- Sentido
- Posição correta
- Velocidade correta



A screenshot of a code editor window titled 'testes.txt'. The editor shows a list of 5 test cases, each with 9 numerical values. The first two lines are highlighted in light blue, and the last two lines are highlighted in light orange. A vertical green line is positioned to the left of the third line.

1	0	0	0	50	5	1	0.5	100
2	0	0	0	100	5	1	1	200
3	5	40	20	60	22	-1	4.8	-50
4	5	40	20	60	22	-1	4.8	-100
5	0	240	20	10	46	1	0.26	10

resultados.txt

 resultados.txt X

Teste >  resultados.txt

1	Posicao : 0.500000 mm Correto	Velocidade : 100.000000 mm/s Correto
2	Posicao : 1.000000 mm Correto	Velocidade : 200.000000 mm/s Correto
3	Posicao : 4.800000 mm Correto	Velocidade : -100.000000 mm/s Errado
4	Posicao : 4.800000 mm Correto	Velocidade : -100.000000 mm/s Correto
5	Posicao : 0.260000 mm Correto	Velocidade : 10.000000 mm/s Correto

Obrigado

Código: <https://github.com/JhonataQuerobim/EncoderSIAI>

Carlos Gomes - 727335

Jhonata Querobim - 727342