



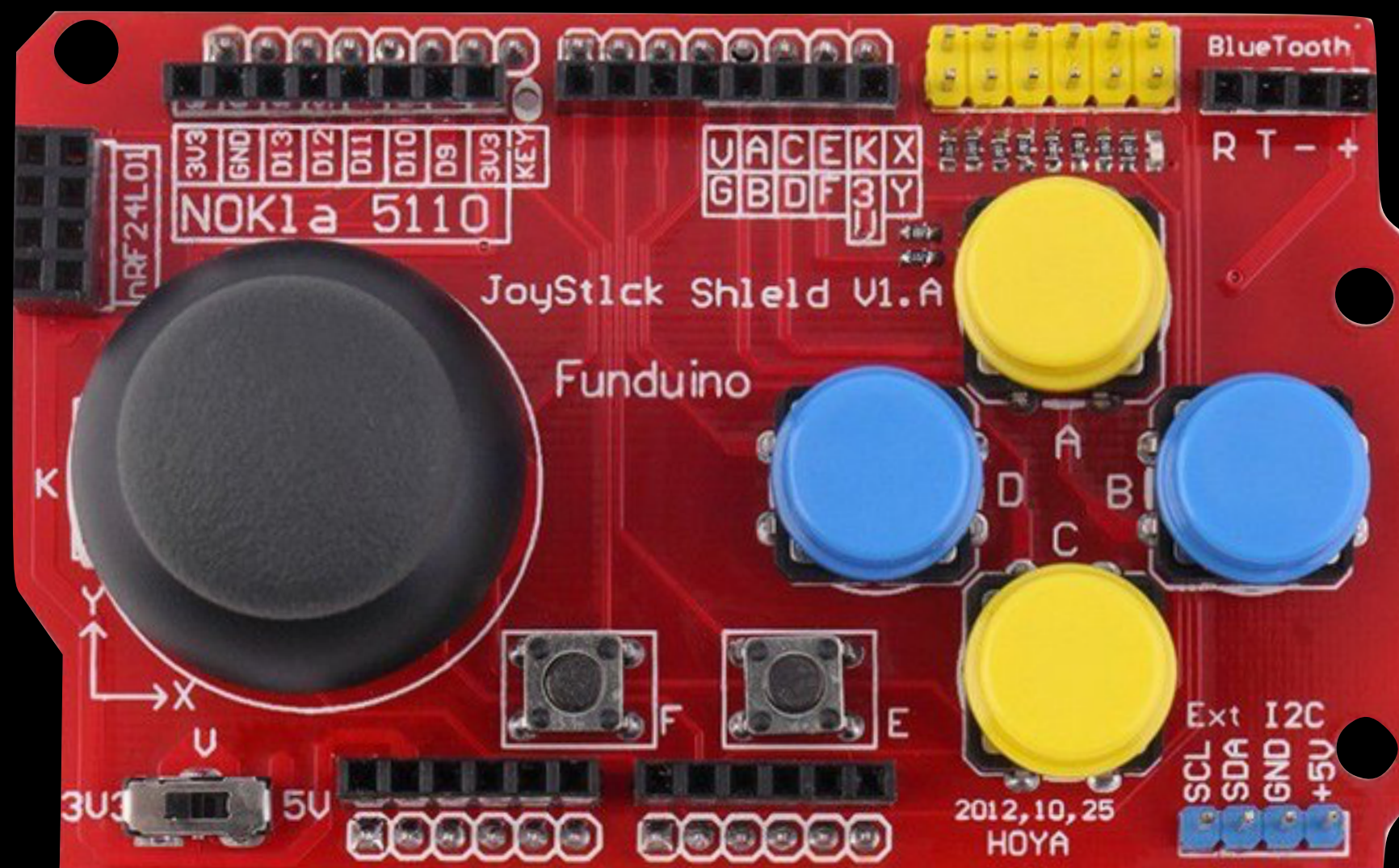
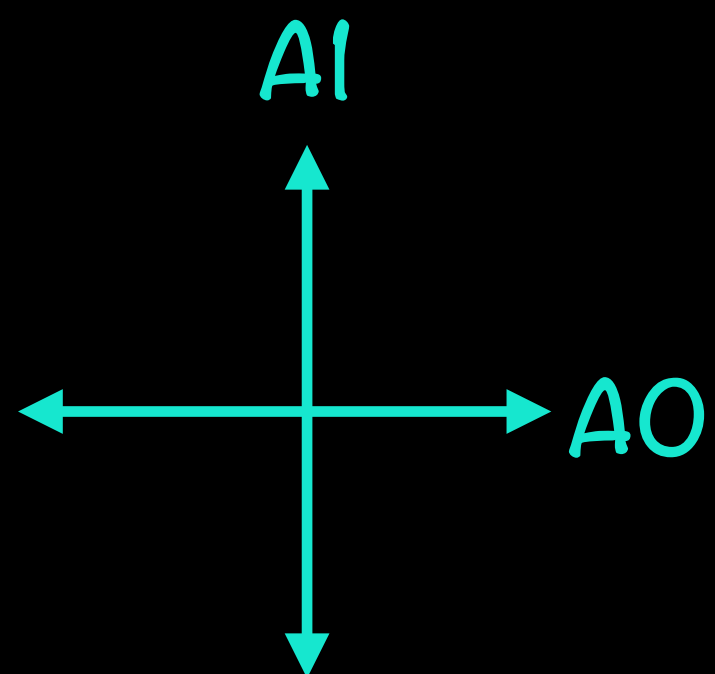
# Projeto 08

## Controle Analógico – Prática

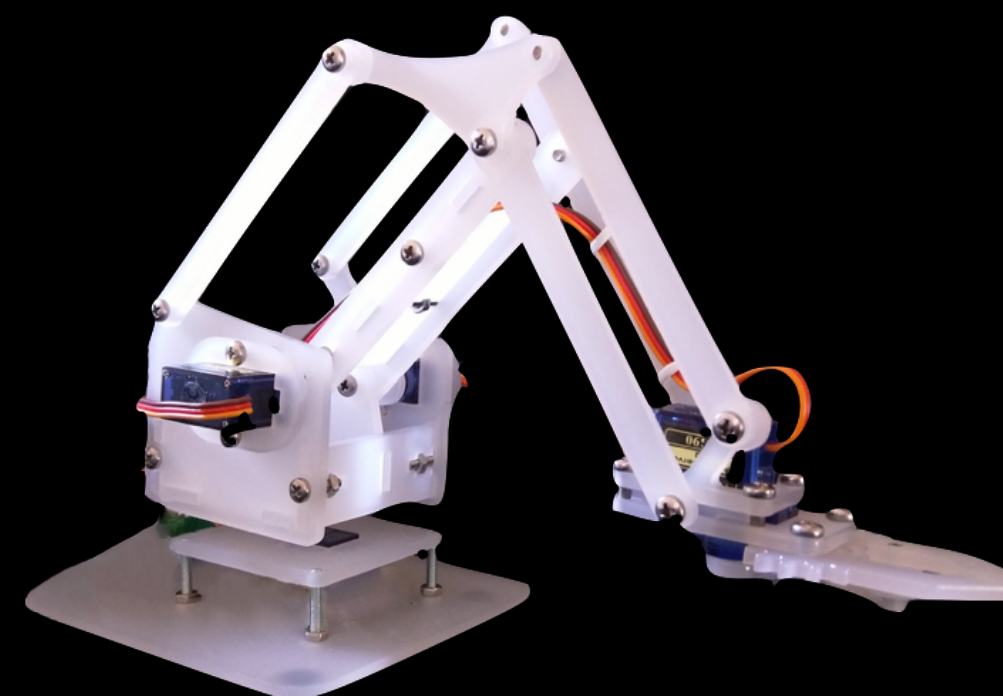
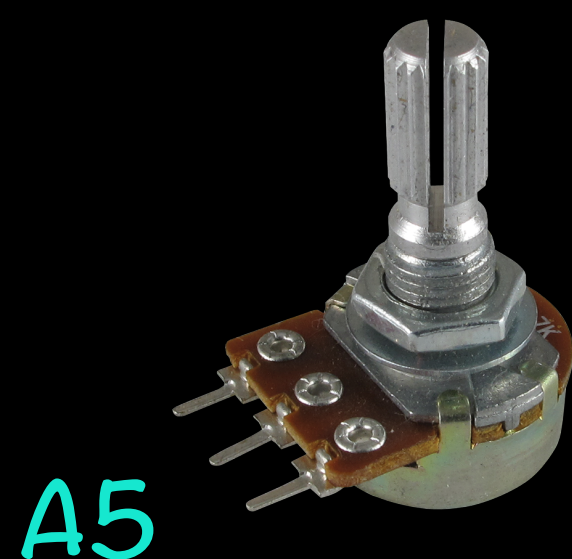
Jan K. S. – [janks@puc-rio.br](mailto:janks@puc-rio.br)

ENG1419 – Programação de Microcontroladores

# Testes Iniciais



2  
5 3  
4



base: 12  
ombro: 11  
cotovelo: 10  
garra: 9

Pinos Usados pelos Componentes

```
#include <EEPROM.h>
```

```
#include <Servo.h>
```

~~#include <meArm.h>~~

Teste da EEPROM e de Controle de Servo (Sem Usar meArm)



## Testes Iniciais

Crie uma variável global indicando **quantas vezes o Botão B (Direita) foi apertado**, imprimindo a contagem via serial.

↳ DICA: use a `GButton`.

Ao apertar o Botão B (Direita), **salve na EEPROM** a contagem no endereço 0. Ao iniciar o programa, carregue essa contagem da memória como o valor inicial.

Ao girar o potenciômetro, **varie o ângulo do servo da base** entre 0 e 180°.

↳ DICA: use a função `map`.

Crie uma variável global para o ângulo do servo do ombro. Se o Botão A (Cima) estiver apertado, **diminua essa variável gradualmente até 45**. Se o Botão C (Baixo) estiver apertado, **aumente essa variável gradualmente até 135**.

↳ DICA: use as funções `digitalRead` e `delay` dentro do loop principal. Verifique o valor imprimindo na serial.

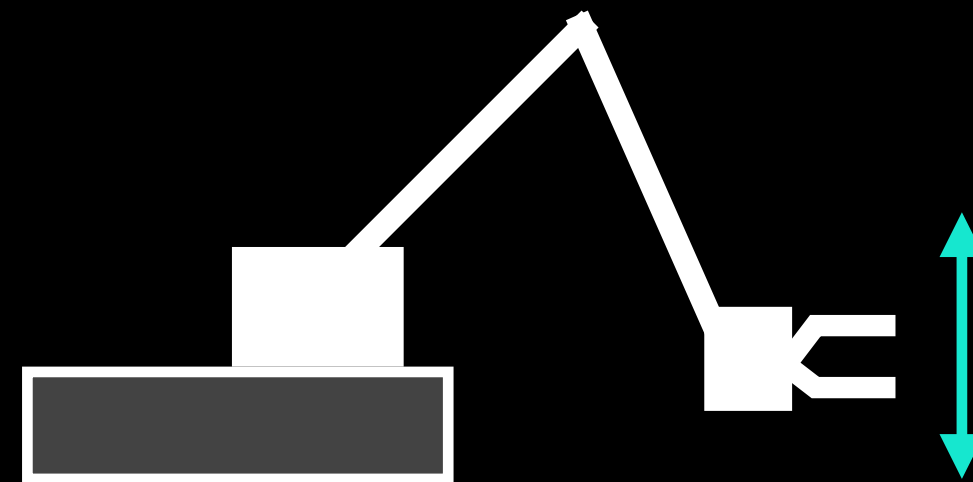
Associe o valor da variável acima ao **ângulo do servo do ombro**. Teste cuidadosamente o movimento.

# Implementação

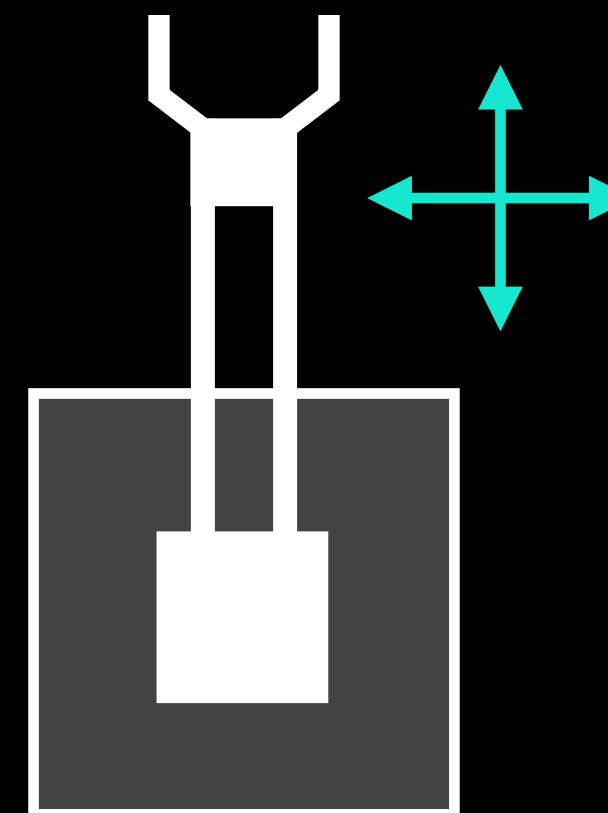
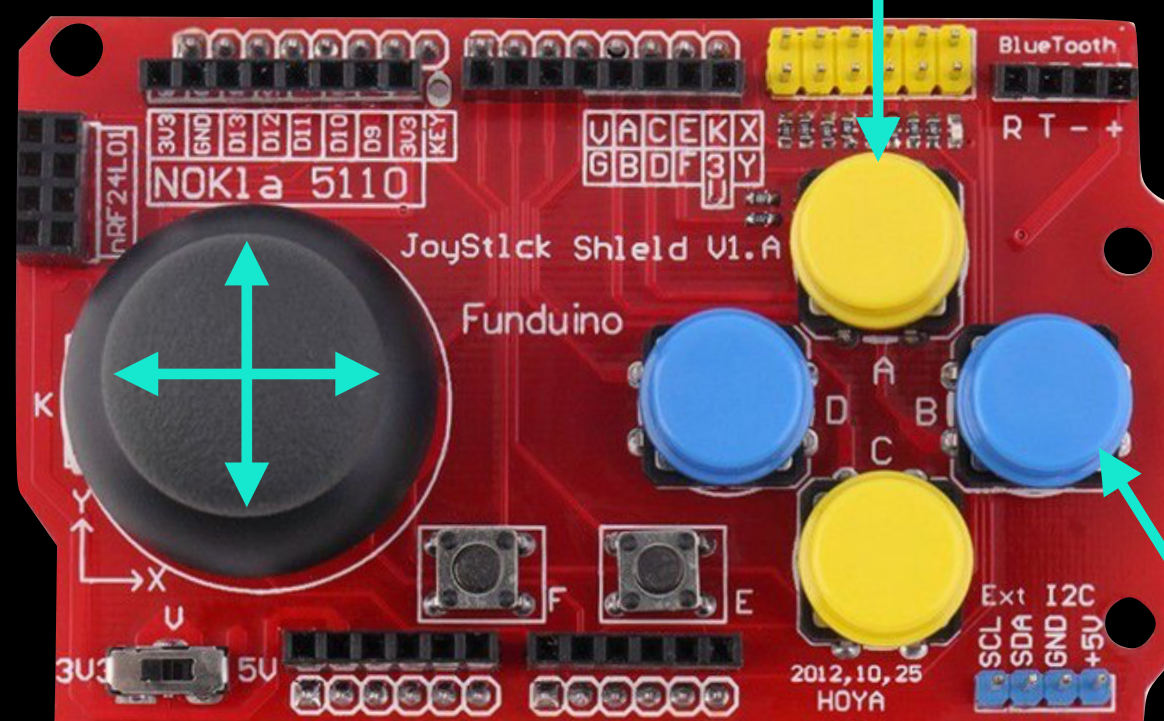




Controle do Braço Mecânico



abre/fecha garra

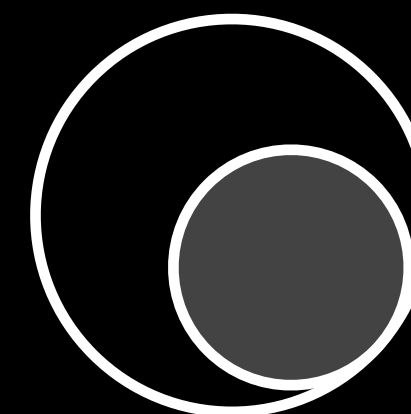
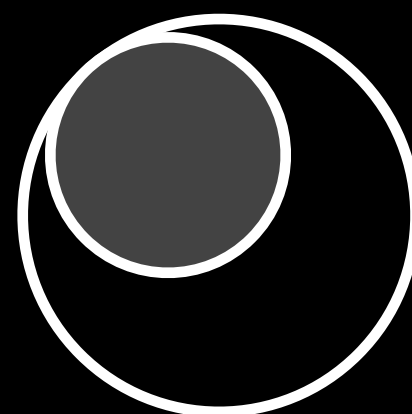
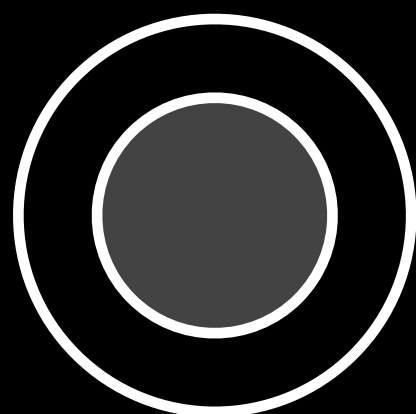
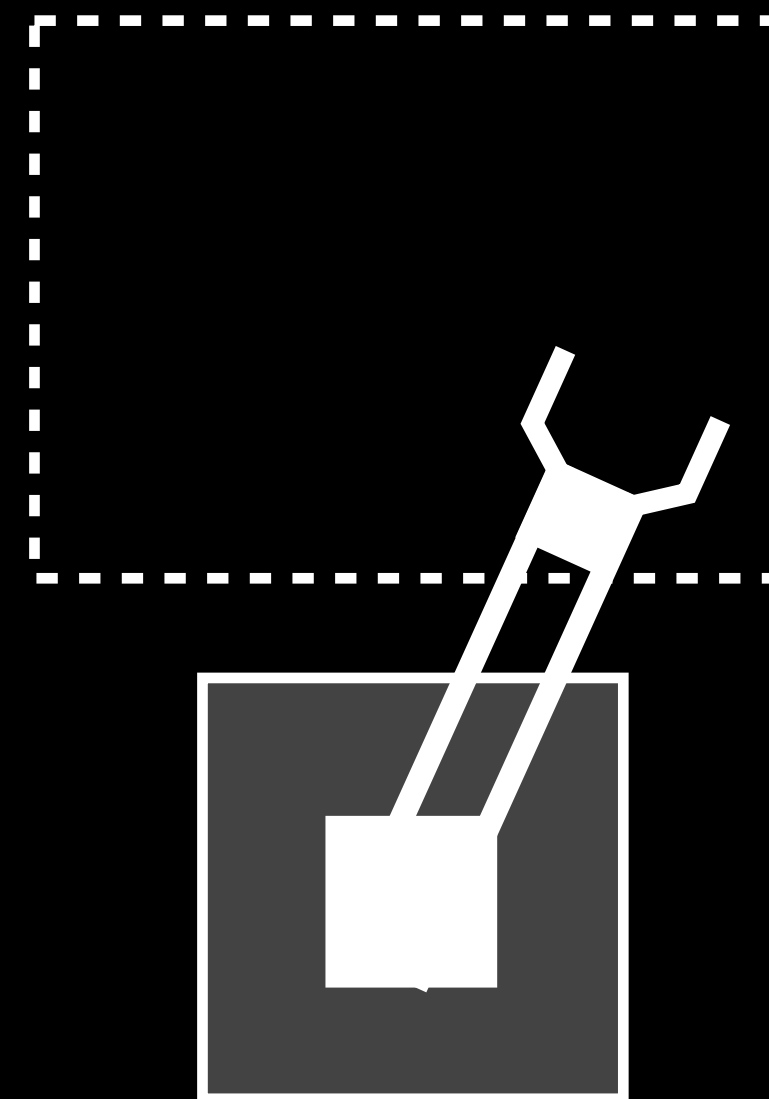
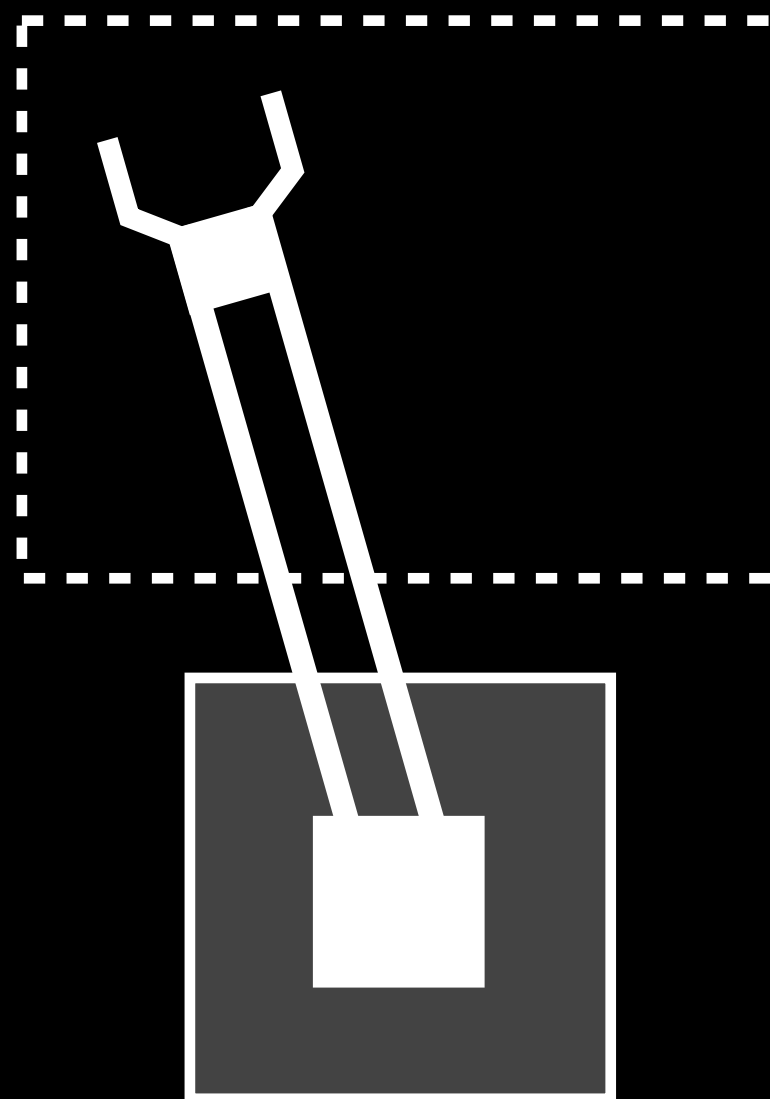
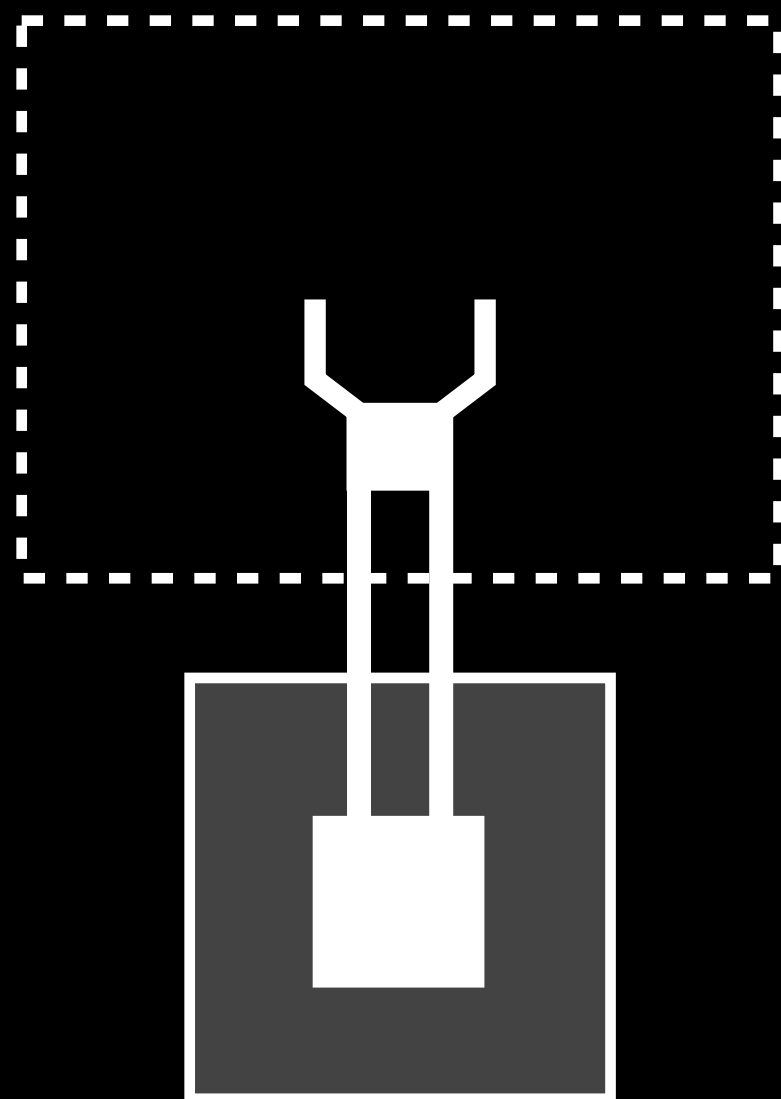


modo absoluto / relativo

Controle Analógico de 3 Coordenadas

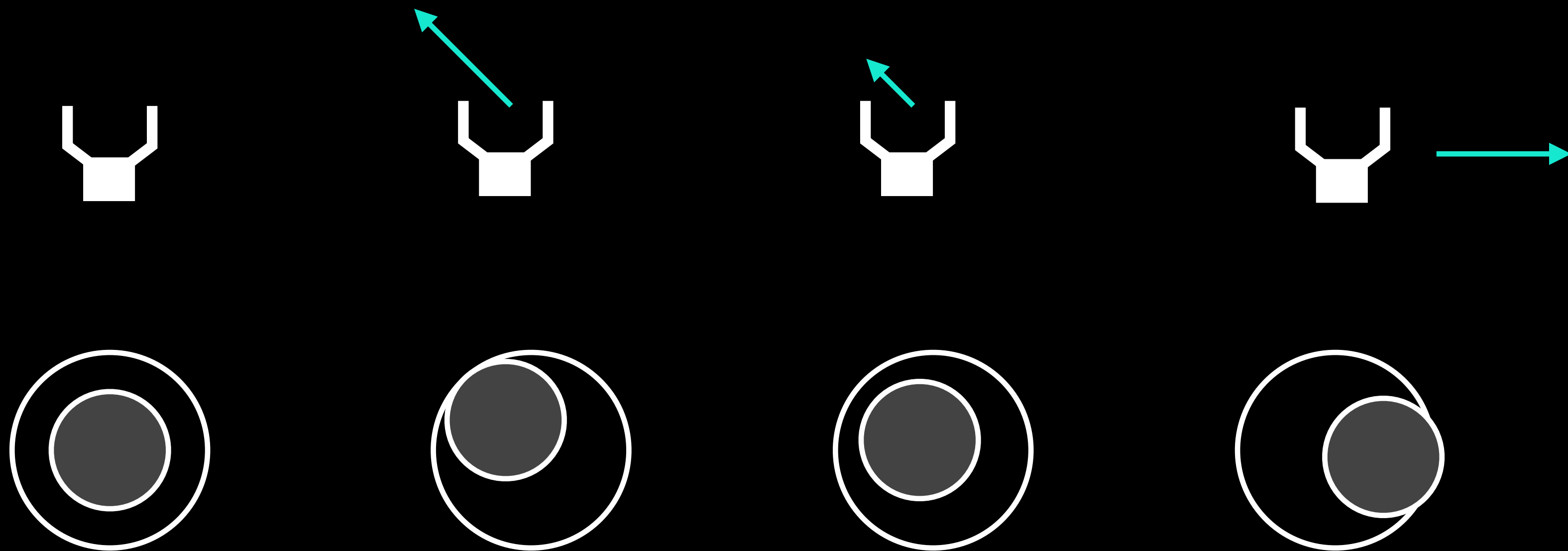


modo absoluto = posição



"Modo Absoluto": Ajuste da Posição

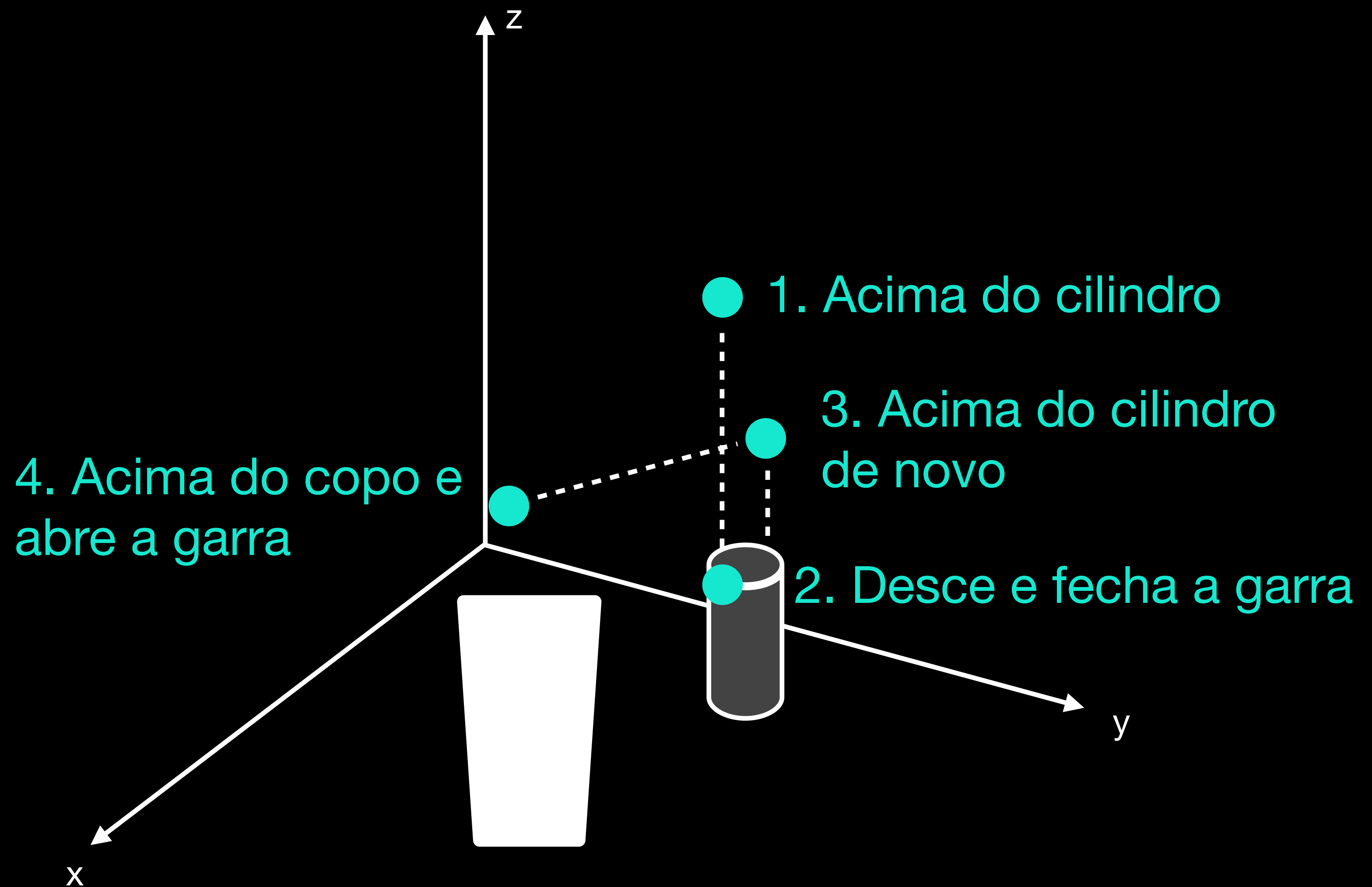
modo relativo = direção



"Modo Relativo": Ajuste da Direção de Movimento

## *Movimento relativo*

- 1. Usar variáveis globais para X e Y*
- 2. Mapear eixos do joystick para valores entre -10 e 10*
- 3. Usar valores como incremento de X e Y, com um delay de 50ms*
- 4. Garantir que X e Y não ultrapassem os limites do braço*
- 5. Imprimir variáveis pela Serial, verificando os valores*
- 6. Usar X e Y como coordenadas da garra, indo **diretamente** para a posição definida.*



Trajetória Desejada: Colocar o Pino dentro do Copo



## Implementação

Ao iniciar o programa, **mova suavemente** a garra para o ponto (0, 100, 0).

↳ DICA: use a biblioteca meArm.

Ao apertar o Botão A (Cima), **abra a garra**. Ao soltar, **feche a garra**.

Ao mexer no potenciômetro, **varie a altura do braço** entre -30 e 100.

Ao mexer no joystick, **mova a garra suavemente no plano XY no "modo absoluto"**, variando X entre -150 e 150 e Y entre 100 e 200.

Ao apertar o Botão B (Esquerda), **alterne o movimento do joystick entre "modo absoluto" e "modo relativo"**. Neste último caso, ajuste as variáveis globais X e Y seguindo o algoritmo indicado.

Associe as variáveis do item anterior à **posição do braço**, indo diretamente para as coordenadas.

Use o braço para **mover o cilindro para o copo**.



Aperfeiçoamento



08b\_implementacao.ino

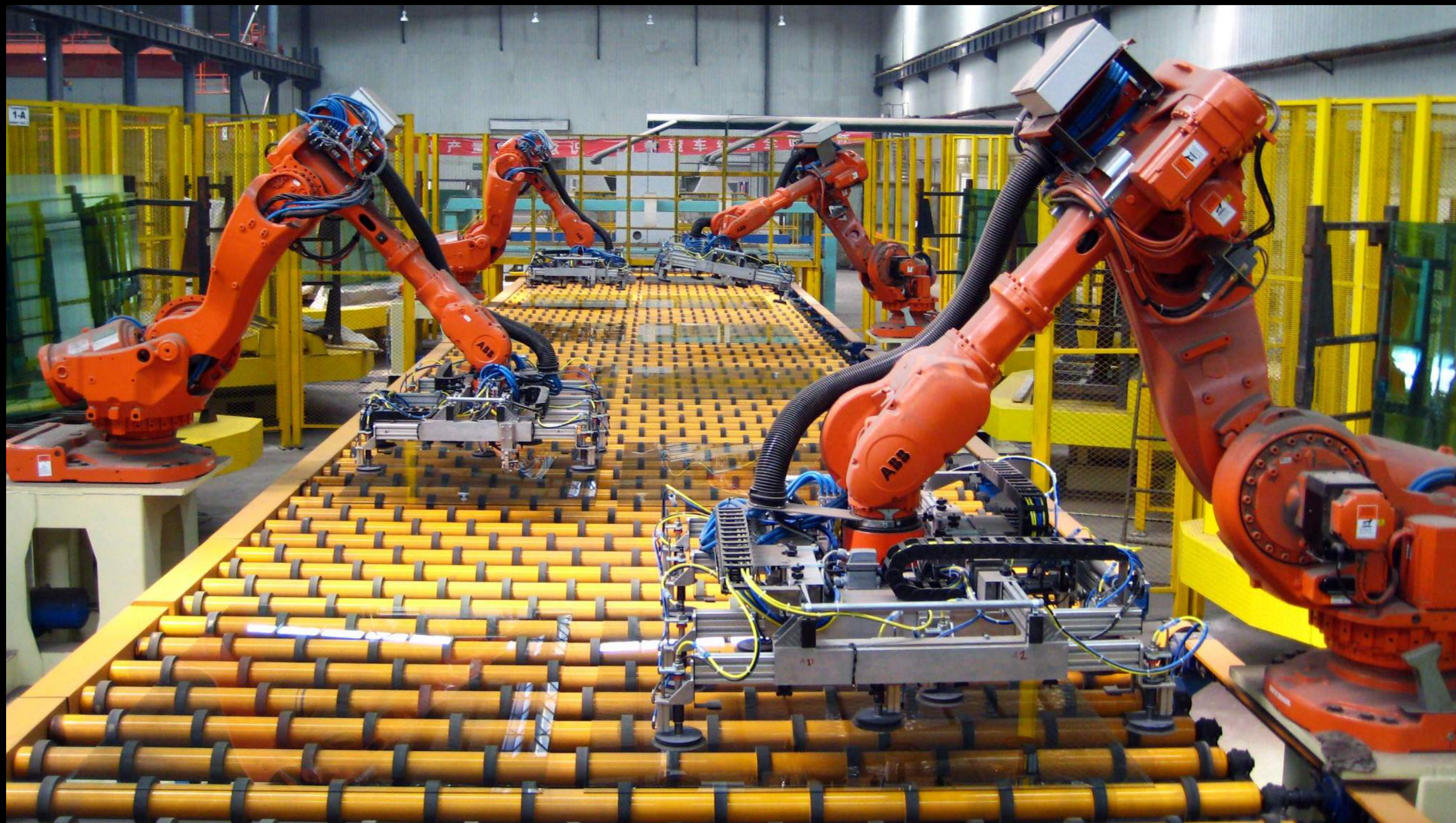
cópia  
-----▶



08c\_aperfeicoamento.ino

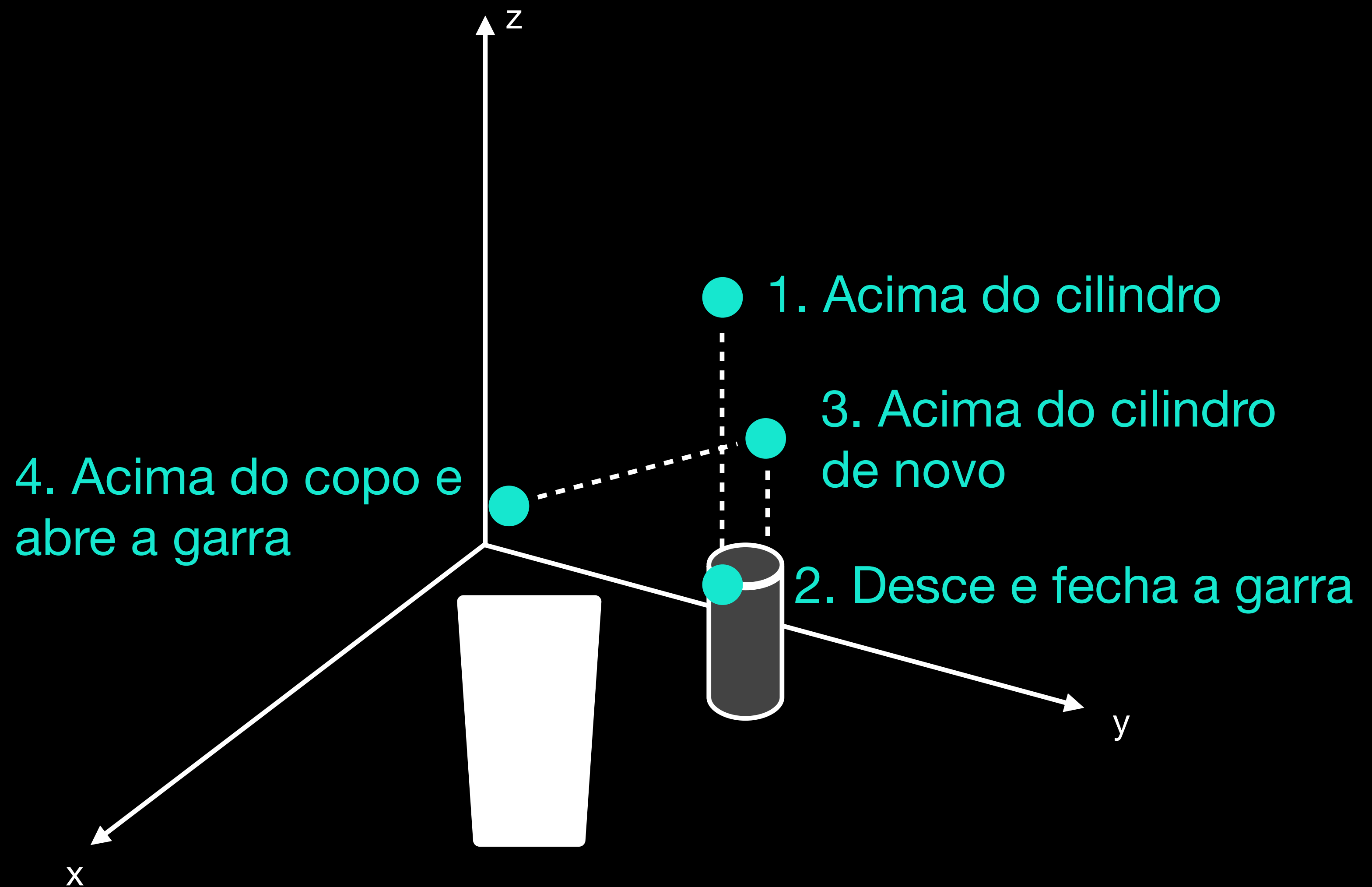
Cópia do Código da Implementação para o Aperfeiçoamento





Controle Automático do Braço





Trajeto Desejado: Colocar o Pino dentro do Copo

## Botão C (Baixo)

move o braço  
para a posição



salva coordenadas e garra na  
matriz (linhas de 0 a 3)

## Botão D (Esquerda)



move braço para as  
coordenadas salvas

Trajeto Desejado: Colocar o Pino dentro do Copo



```
float pontosSalvos[4][4];
```

	x	y	z	garra aberta/fechada
ponto 1				
ponto 2				
ponto 3				
ponto 4				

Armazenamento dos Pontos



## Aperfeiçoamento

Ao apertar o Botão C (Baixo), **salve as coordenadas e o estado da garra (aberto/fechado)** na matriz 4x4, variando o ponto atual de 0 a 4.

Ao apertar o Botão D, leia os dados salvos e **mova o braço suavemente** para cada uma das 4 posições, abrindo ou fechando a garra, com intervalos de 500 ms entre cada ponto.

Ao salvar o ponto, **guarde a matriz dentro da EEPROM**. Ao iniciar o programa, carregue a matriz da EEPROM.

Desafio Extra



08c\_aperfeicoamento.ino

cópia  
-----▶



08c\_desafio.ino

Cópia do Código do Aperfeiçoamento para o Desafio

```
float pontosSalvos[4][4];
```



E se eu quiser  
mais posições?

```
float pontosSalvos[1000][4];
```



Desperdiça muita  
memória e não identifica  
direito os dados dentro  
da matriz.

Problemas com a Solução Matricial



lista encadeada

estrutura

x: -35.2  
y: 104.6  
z: 56.4  
aberto: false

x: 75.3  
y: 167.9  
z: 81.7  
aberto: true

x: -119.6  
y: 199.0  
z: -27.1  
aberto: false

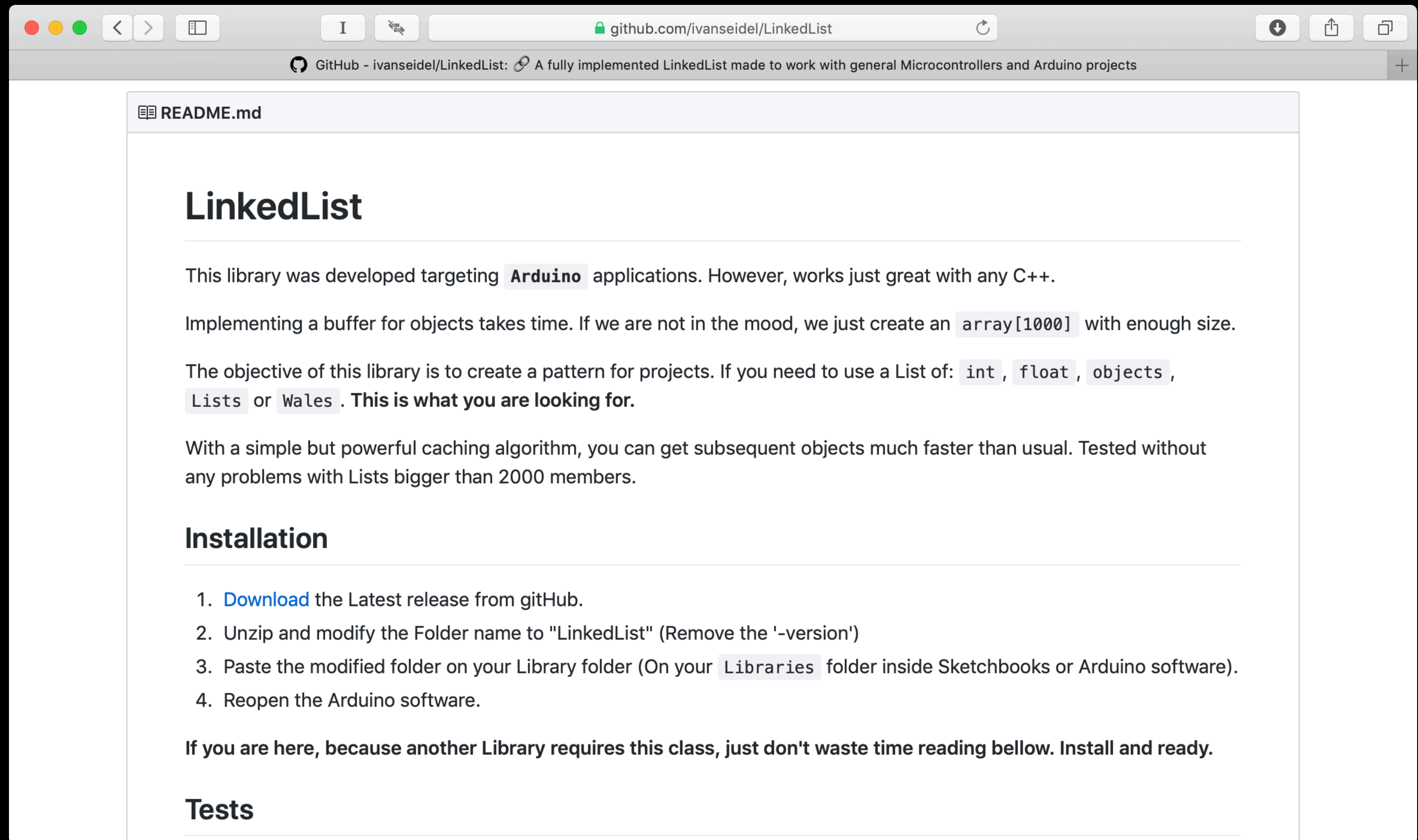
...

Solução Elegante com Estruturas de Dados e Lista Encadeada

```
struct Posicao {  
    float x;  
    float y;  
    float z;  
    bool garraAberta;  
}
```

...

```
Posicao novaPosicao;  
novaPosicao.x = 2.4;  
novaPosicao.y = 162.3;  
novaPosicao.z = -19.8;  
novaPosicao.garraAberta = true;
```



```
// criação de lista para um certo tipo de elemento
LinkedList<int> listaDeInteiros;
LinkedList<float> listaDeDecimais;
LinkedList<bool> listaDeBooleanos;
```

```
LinkedList<Posicao> listaDeEstruturas;
```

```
...
```

```
// adiciona elemento no final da lista
listaDeEstruturas.add(elemento);
```

```
// acessa elemento da lista
Posicao elemento = listaDeEstruturas.get(indice);
```

```
// remove todos os elementos
listaDeEstruturas.clear();
```

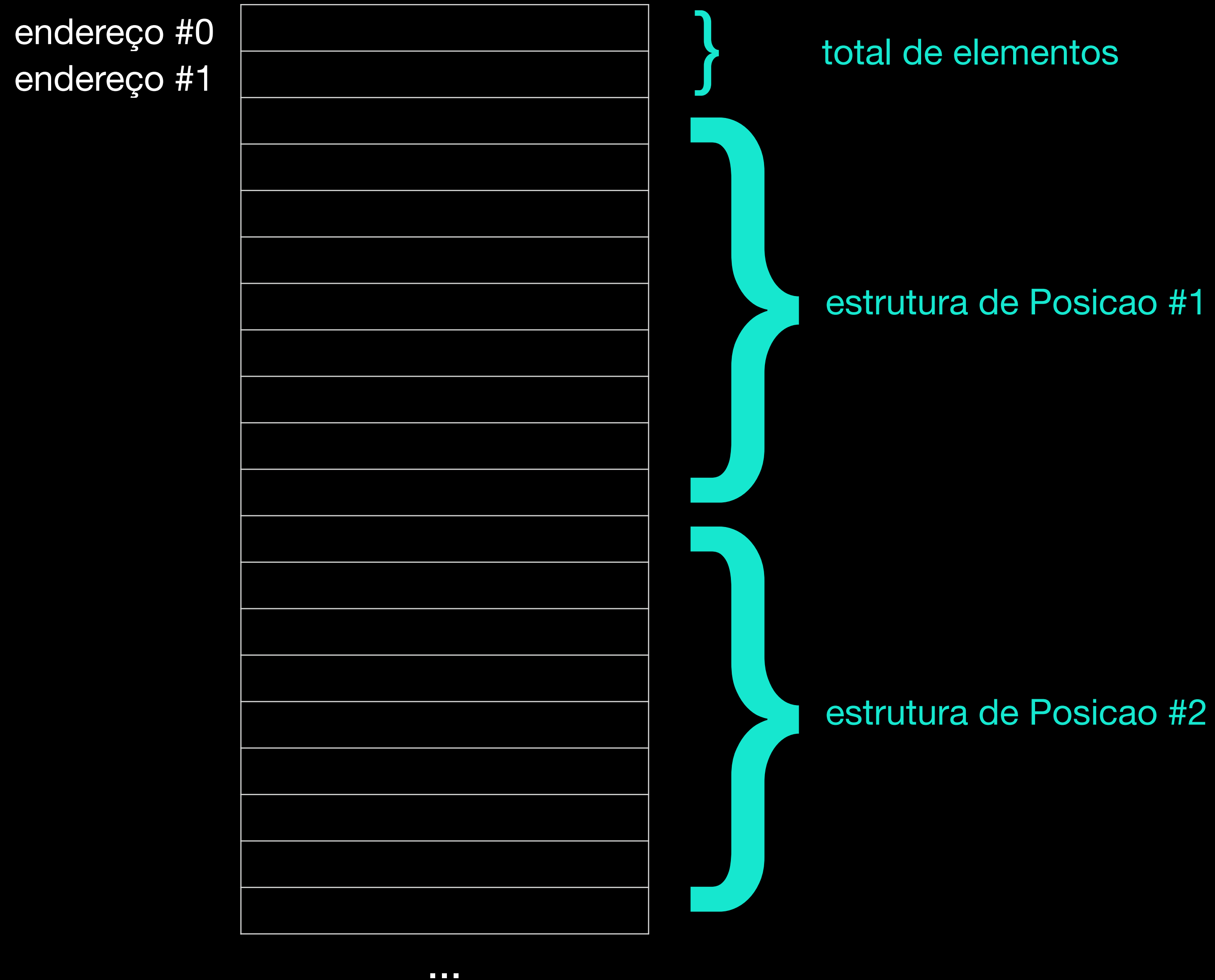
Exemplo de Uso da Biblioteca LinkedList

```
Posicao novaPosicao;  
novaPosicao.x = 2.4;  
novaPosicao.y = 162.3;  
novaPosicao.z = -19.8;  
novaPosicao.garraAberta = true;  
  
EEPROM.put(endereco, novaPosicao); // funciona!
```

```
LinkedList<Posicao> lista;  
lista.add(estrutura1);  
lista.add(estrutura2);
```

```
EEPROM.put(endereco, lista); // não funciona!
```







## Desafio Extra

Adicione a definição da estrutura e variável global de lista encadeada. Em seguida, modifique o código do Aperfeiçoamento para **salvar e percorrer as estruturas de posições salvas na lista** em vez de na matriz.

Modifique também o código do Aperfeiçoamento para **salvar o total de elementos e cada estrutura na memória**.

↳ DICA: usa a função `sizeof` para calcular quantos bytes a estrutura vai ocupar na memória.

Ao apertar o botão E (porta 6), **apague a lista de posições salvas** e atualize o total na EEPROM.



[janks.link/micro/projeto08.zip](https://janks.link/micro/projeto08.zip)

Material do Projeto 08