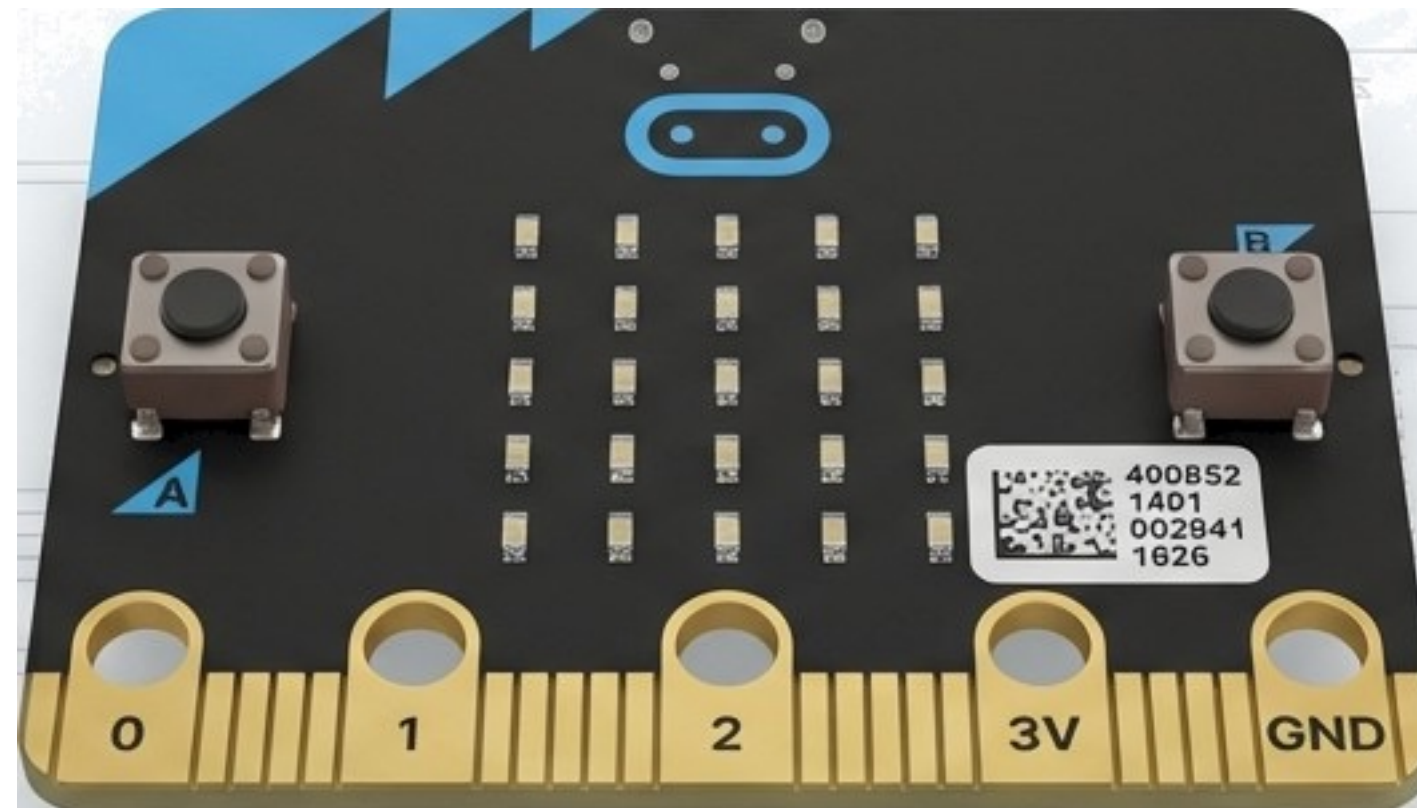


# Da Ideia ao Objeto Inteligente

Um Workshop Criativo com o micro:bit v2



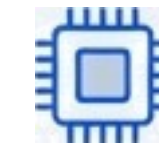
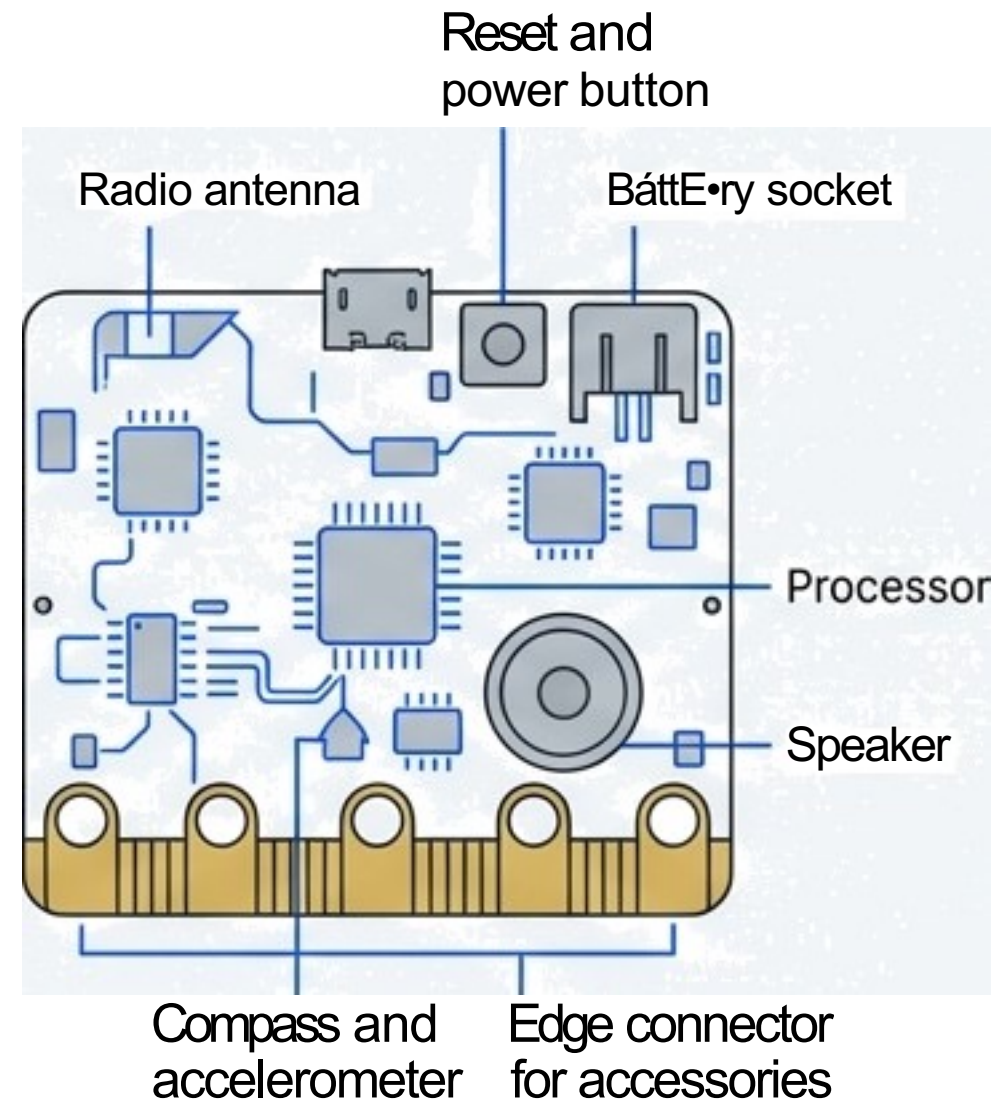
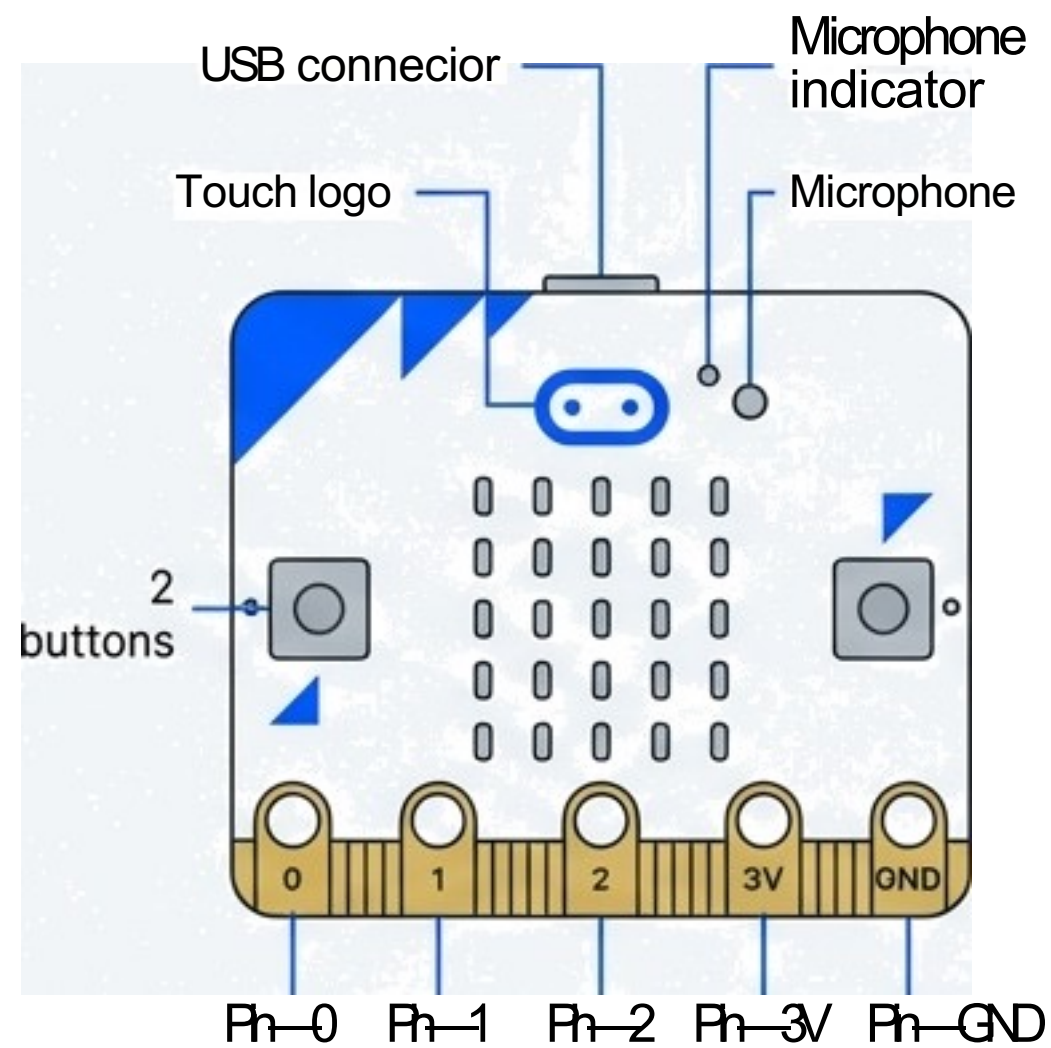
# Onde o Código Ganha Vida

A programação não existe apenas em ecrãs.  
E se o vosso código pudesse sentir, reagir  
e interagir com o mundo real?



*“Este pequeno dispositivo é um cérebro à espera dos vossos comandos. Hoje, vão dar-lhe vida.”*

# A Ferramenta: O Cérebro Compacto



- Processador: 0 núcleo inteligente.



- Matriz de 25 LEDs: O seu ecrá para comunicar.



- Sensores Integrados: Acelerómetro, bússola, luz, temperatura e microfone.



- Botões e Logo Tátil: Entradas diretas para os vossos comandos.



- Rádio & Bluetooth: A sua voz para falar com outros dispositivos.

# A Vossa Missão: Desbloquear o Potencial

Nos próximos 45 minutos, não vão apenas seguir instruções. Vão desbloquear três capacidades fundamentais que transformam este dispositivo numa ferramenta inteligente.



PERCEÇÃO



COMUNICAÇÃO

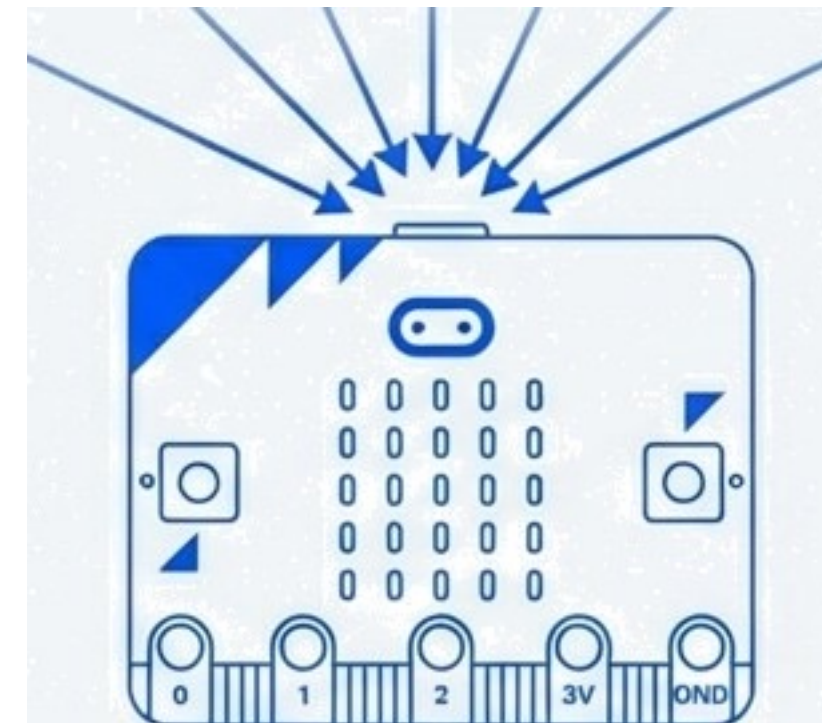


LÓGICA & MEMÓRIA



# A Chave da Percepção

O primeiro passo é dar ao vosso micro:bit a capacidade de sentir o ambiente que o rodeia. Vão ensiná-lo a 'ver' a luz, a 'sentir' o toque e a 'perceber' o movimento.



# Desafio 1: Sentir e Reagir (⋄ 15 min)

---

## Objetivo

Explorar os sensores e botões do micro:bit e programar respostas visuais imediatas.

## -õ O que vão aprender

- \* Ler valores de sensores internos (luz, temperatura).
- \* Utilizar eventos (botões, gestos como 'agitar').
- \* Mostrar informação no ecrã de LEDs.

## Requisitos Mínimos

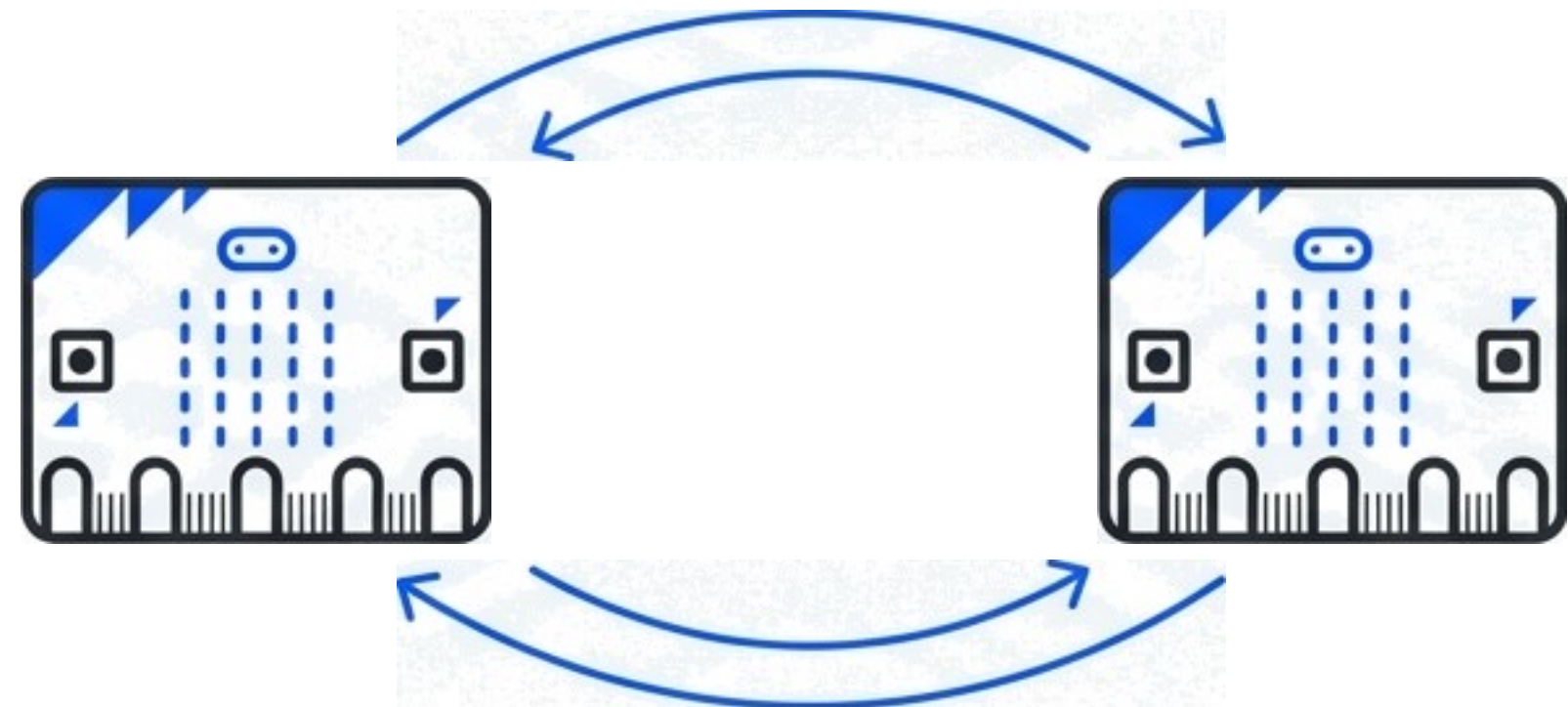
1. Ao iniciar + mostrar um ícone à vossa escolha.
2. Botão A premido + mostrar a temperatura atual.
3. Botão B premido e mostrar o nível de luz ambiente.
4. Ao agitar -+ mostrar um ícone de aviso.

**Extensão (Opcional):** Ir mais além. Classificar a temperatura (frio/bom/quente) ou mostrar texto a deslizar.



# A Chave da Comunicação

Um dispositivo inteligente não está isolado. Agora, vão dar ao micro:bit uma voz, permitindo que comunique sem fios com os seus pares, trocando mensagens e coordenando ações.



# Desafio 2: Comunicação por Rádio(<>15 min)

---

## Objetivo

Criar uma rede de comunicação sem fios entre os micro:bits do vosso grupo.

## O que se aprende

- \* Configurar e utilizar a comunicação por rádio.
- \* Enviar e receber diferentes tipos de mensagens.
- \* Programar reações a eventos externos (mensagens recebidas).

**Extensão (Opcional):** Criar códigos secretos, enviar dados de sensores ou reagir de forma diferente a mensagens distintas.

## Requisitos Mínimos

1. Definir um 'grupo de rádio' comum para a vossa equipa.
2. Ao premir um botão, enviar uma mensagem (número ou texto).
3. Ao receber uma mensagem, reagir mostrando um ícone ou texto.



# A Chave da Lógica & Memória

A inteligência requer memória. Nesta fase, vão dar ao micro:bit a capacidade de se lembrar de informação, de a atualizar e de tomar decisões com base nesses dados. Vão criar uma variável.

variável \_\_\_\_\_

0

2

# Desafio 3: Contador de Passos

10-15 min)



## Objetivo

Utilizar o acelerómetro e variáveis para contar movimentos repetidos e guardar o total.



## O que vão aprender

- Utilizar gestos do acelerómetro ('shake').
- Criar e manipular variáveis para armazenar dados.
- Atualizar e mostrar valores em tempo real.



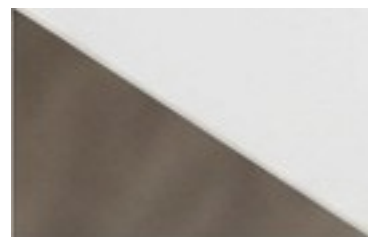
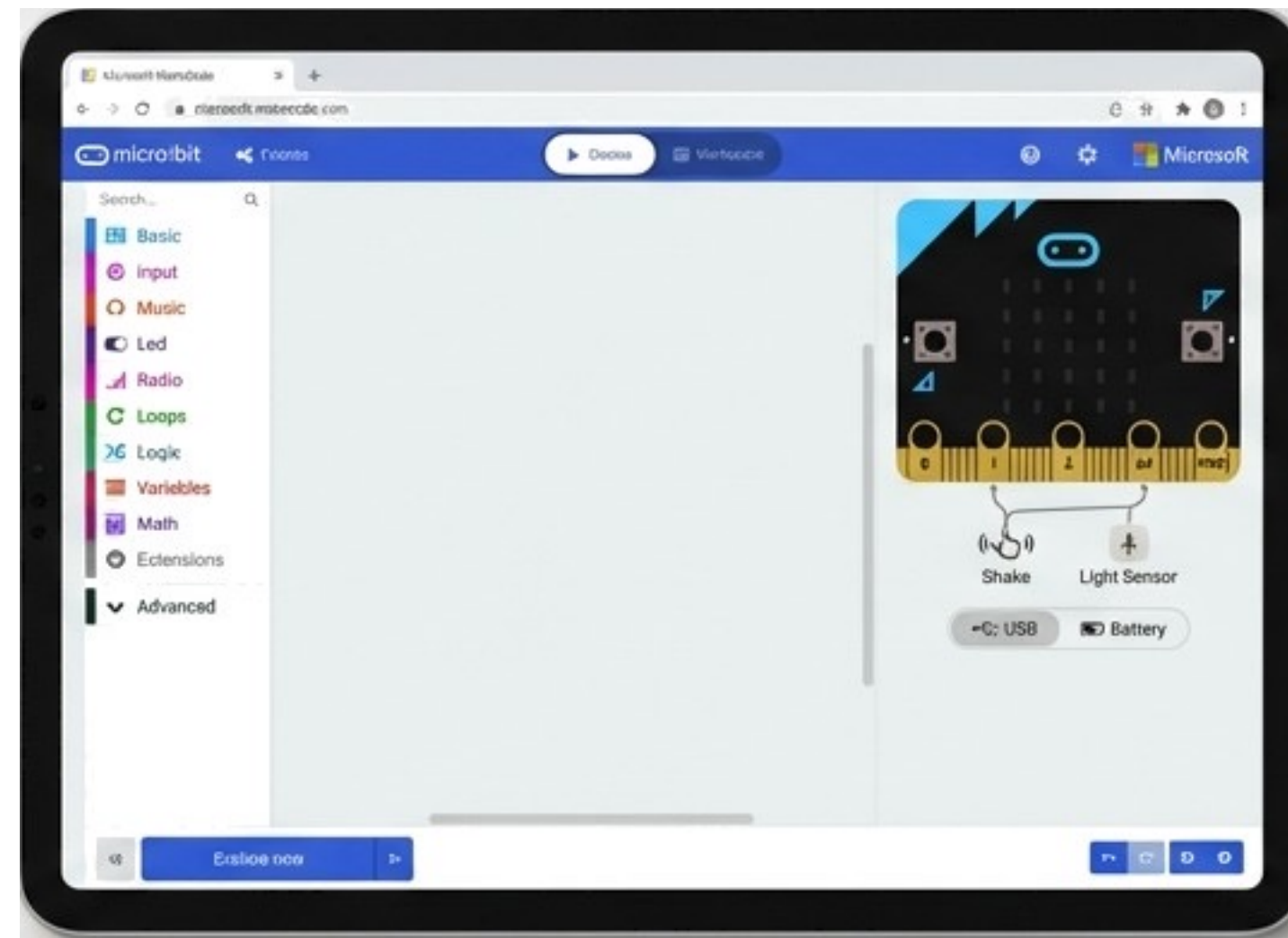
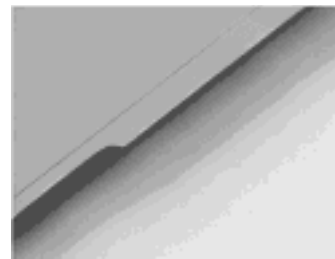
## Requisitos Mínimos

1. Criar uma variável chamada 'passos'.
2. Incrementar a variável sempre que a placa for agitada.
3. Mostrar o valor total de 'passos' no ecrã.

Extensão (Opcional): Adicionar uma função de reset com um botão, emitir um som a cada 10 passos ou enviar o total por rádio para outro micro:bit.

# O Vosso Atelier Criativo

Todas estas capacidades são desbloqueadas através de código. A vossa ferramenta para esculpir o comportamento do micro:bit é o Microsoft MakeCode, MakeCode, um editor visual e poderoso que corre diretamente no browser.



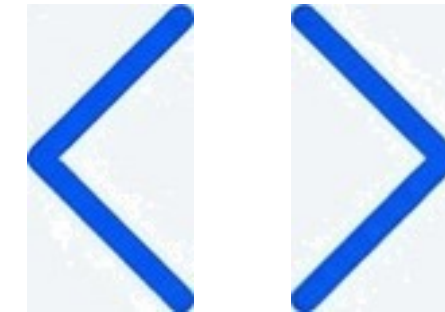
# Dois Caminhos para a Criação

O MakeCode oferece duas formas de programar. Não há uma escolha errada; apenas a que se adequa melhor ao vosso estilo de pensamento. A escolha é vossa.



## Blocos

Ideal para lógica visual, prototipagem rápida e para quem está a começar.

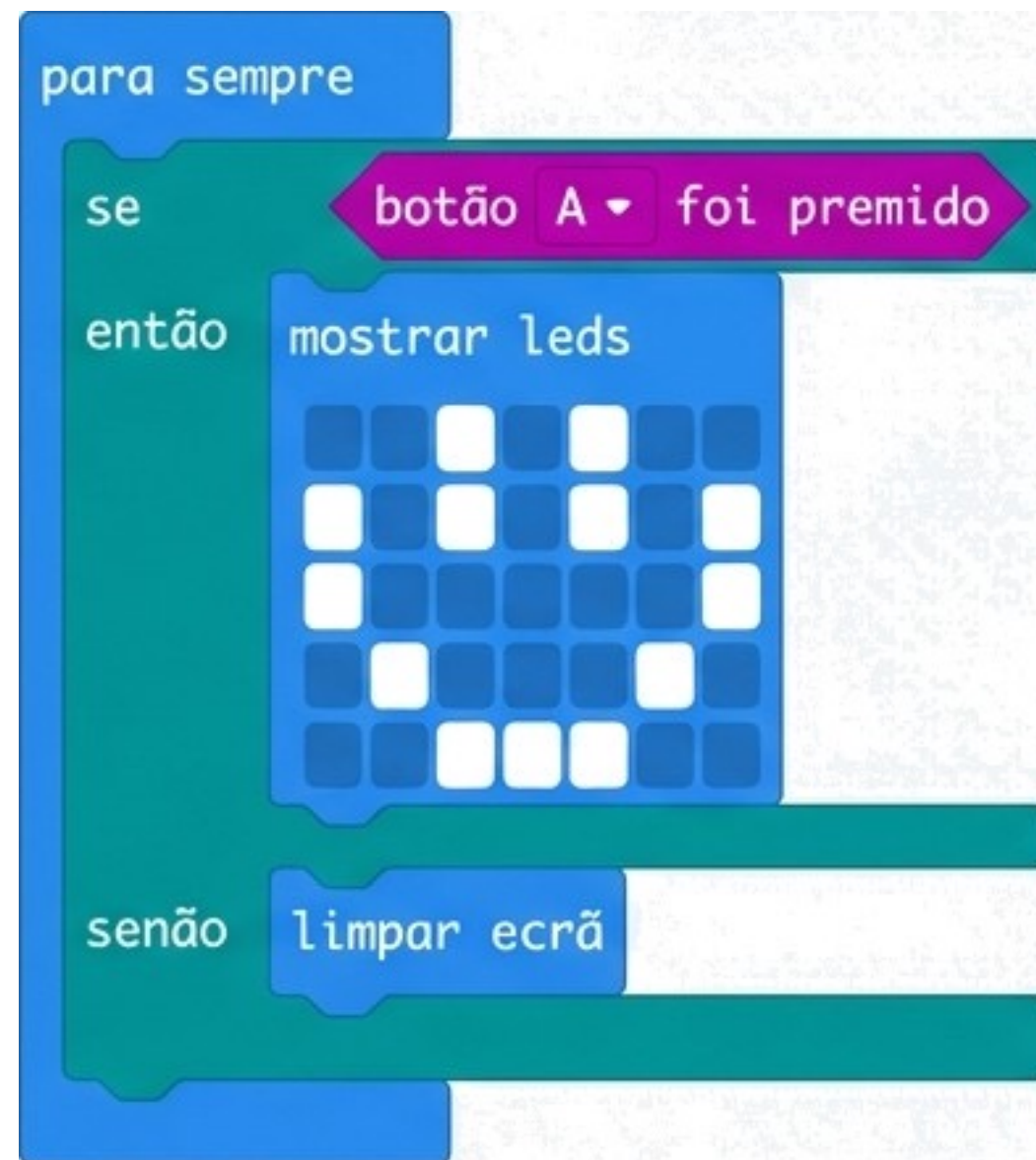


## Python

Ideal para controlo textual, estruturas de dados complexas e para quem já tem experiência em programação.

# Blocos vs. Python: Uma Comparação Rápida

## Blocos



## Python

```
from microbit import *  
  
while True:  
    if button_a.is_pressed():  
        display.show(Image.HAPPY)  
    else:  
        display.clear()
```

Podem alternar entre as duas vistas a qualquer momento no editor.

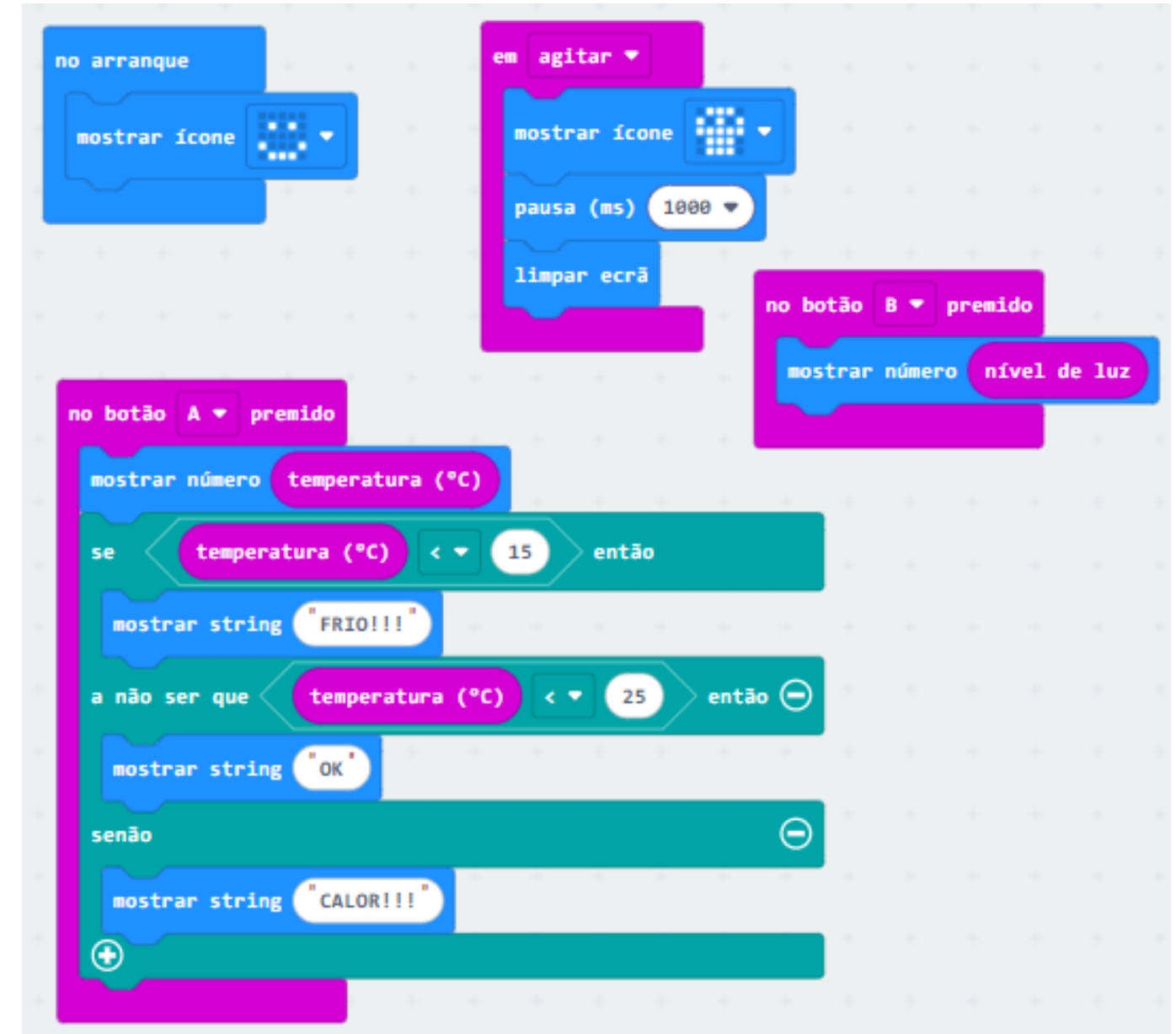
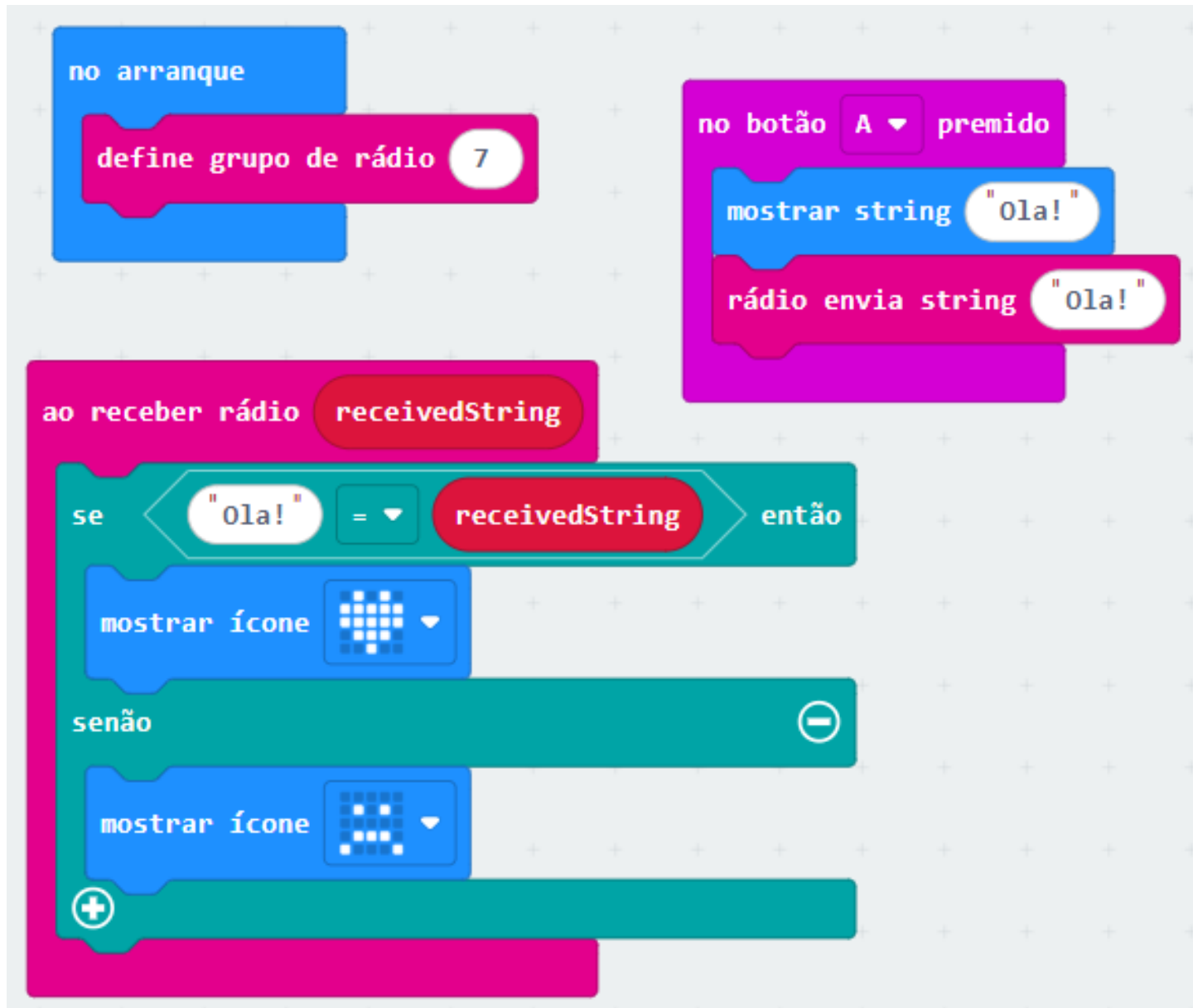
Inter Regular

O cérebro está pronto.  
As chaves estão nas vossas

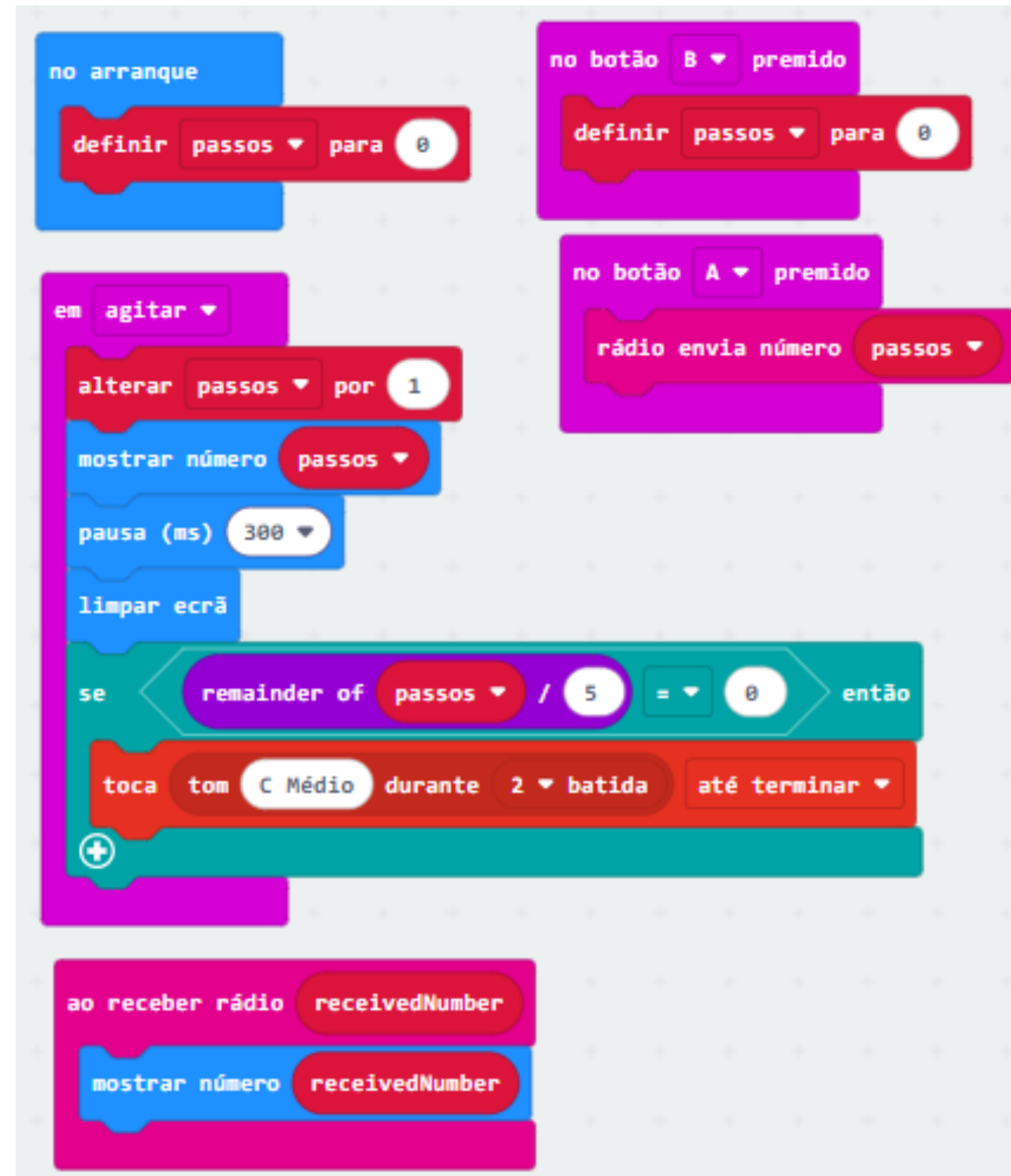
Trabalhem em grupo. **mãos.** Vão além do pedido.

**Surpreendam-nos.**

# Programas - BLOCOS



# Programas - BLOCOS



# Programas - PYTHON

```
def on_button_pressed_a():
    basic.show_string("Ola!")
    radio.send_string("Ola!")
input.on_button_pressed(Button.A, on_button_pressed_a)

def on_received_string(receivedString):
    if "Ola!" == receivedString:
        basic.show_icon(IconNames.HEART)
    else:
        basic.show_icon(IconNames.SAD)
radio.on_received_string(on_received_string)

radio.set_group(7)
```

```
def on_button_pressed_a():
    basic.show_number(input.temperature())
    if input.temperature() < 15:
        basic.show_string("FRIO!!!")
    elif input.temperature() < 25:
        basic.show_string("OK")
    else:
        basic.show_string("CALOR!!!")
input.on_button_pressed(Button.A, on_button_pressed_a)

def on_gesture_shake():
    basic.show_icon(IconNames.SKULL)
    basic.pause(1000)
    basic.clear_screen()
input.on_gesture(Gesture.SHAKE, on_gesture_shake)

def on_button_pressed_b():
    basic.show_number(input.light_level())
input.on_button_pressed(Button.B, on_button_pressed_b)

basic.show_icon(IconNames.HAPPY)
```

# Programas - PYTHON

```
def on_received_number(receivedNumber)
:
    basic.show_number(receivedNumber)
radio.on_received_number(on_received_n
umber)
```

```
def on_button_pressed_a():
    radio.send_number(passos)
input.on_button_pressed(Button.A, on_b
utton_pressed_a)
```

```
def on_gesture_shake():
    global passos
    passos += 1
    basic.show_number(passos)
    basic.pause(300)
    basic.clear_screen()
```

```
    if passos % 5 == 0:
        music.play(music.tone_playable
(262, music.beat(BeatFraction.DOUBLE))
,
                music.PlaybackMode.UNTIL_D
ONE)
input.on_gesture(Gesture.SHAKE, on_ges
ture_shake)
```

```
def on_button_pressed_b():
    global passos
    passos = 0
input.on_button_pressed(Button.B, on_b
utton_pressed_b)
```

```
passos = 0
```

```
passos = 0
```