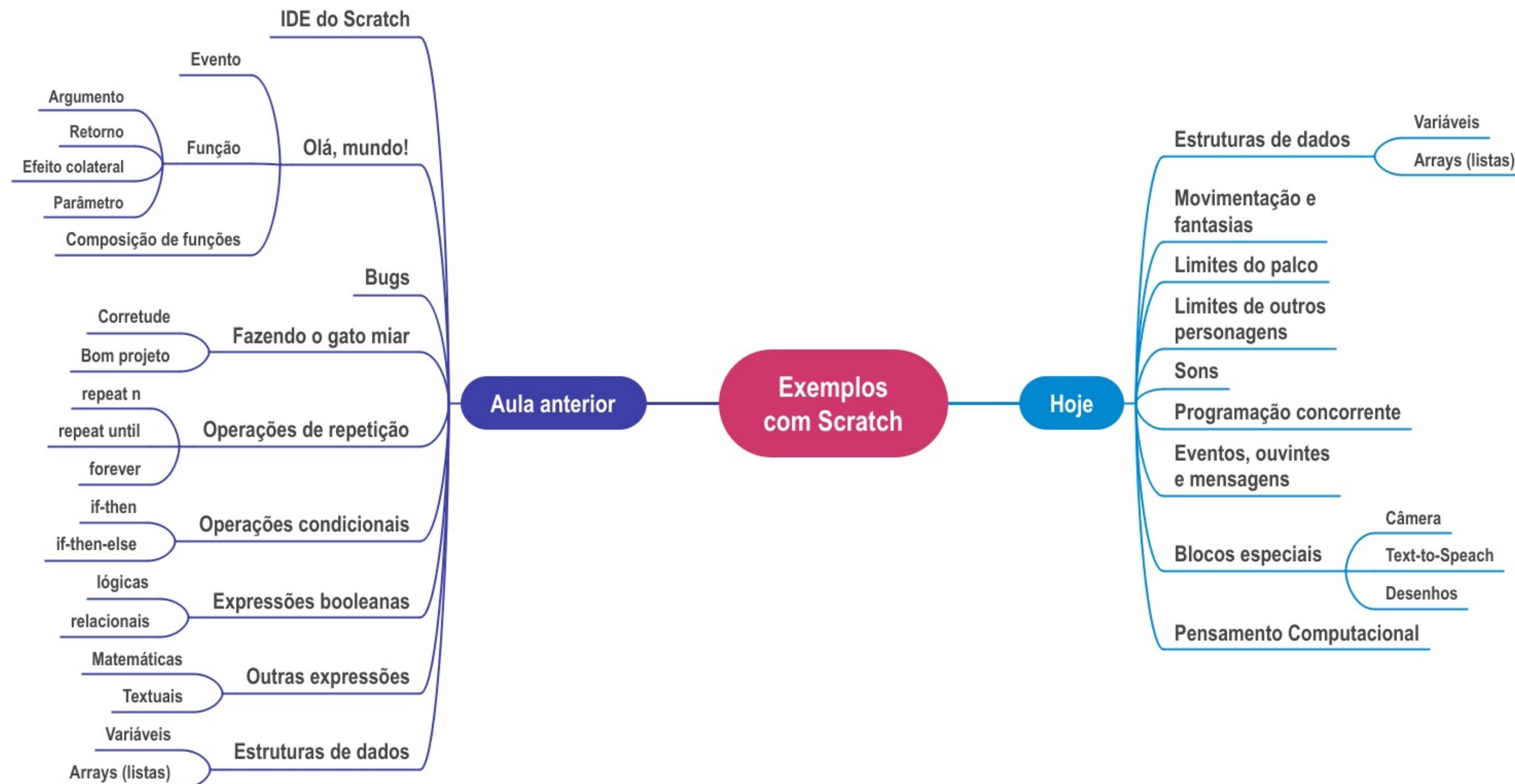


# FUNDAMENTOS DA PROGRAMAÇÃO



# Outros exemplos

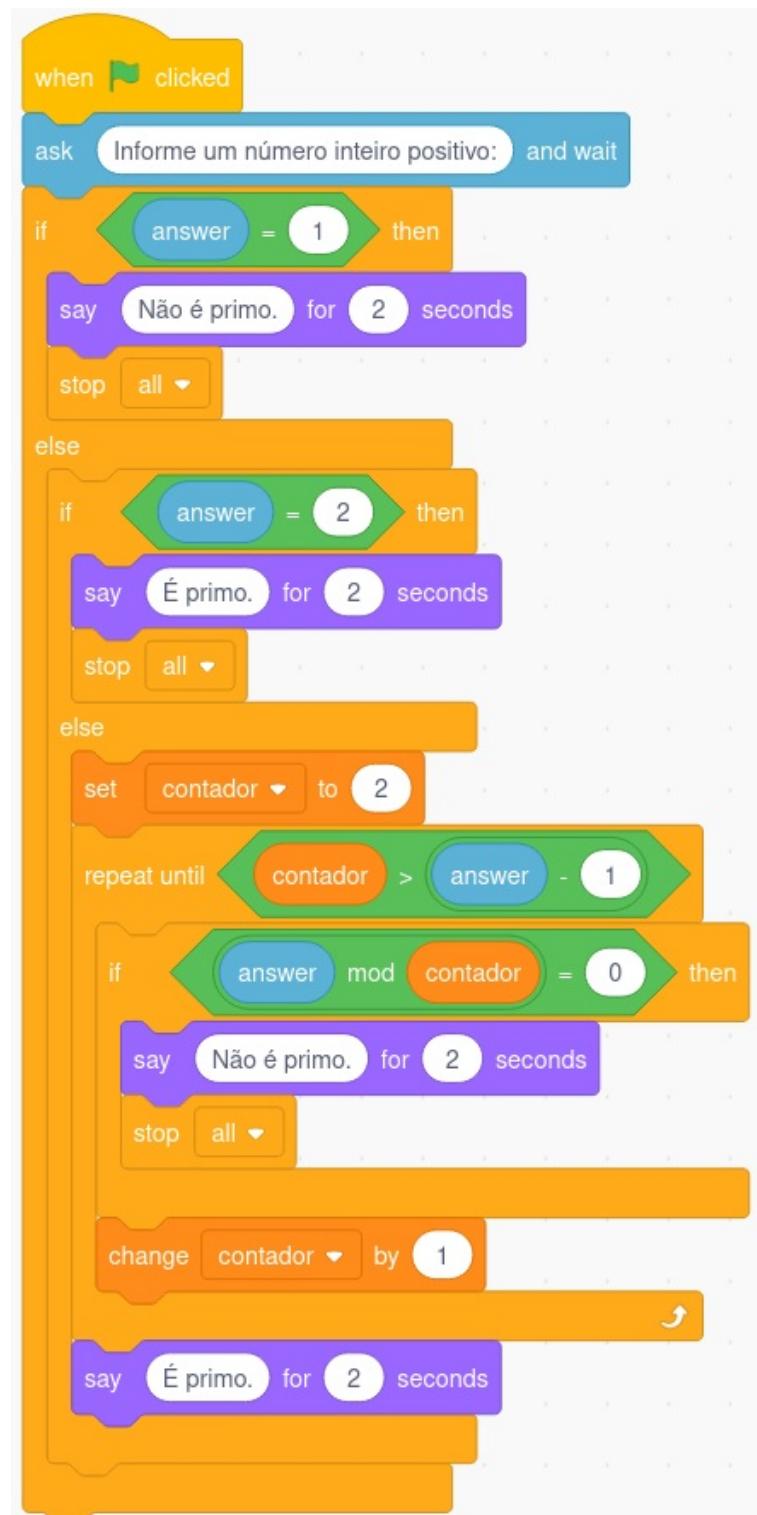


# Estruturas de dados: variáveis e arrays

Scratch fornece duas estruturas de dados:

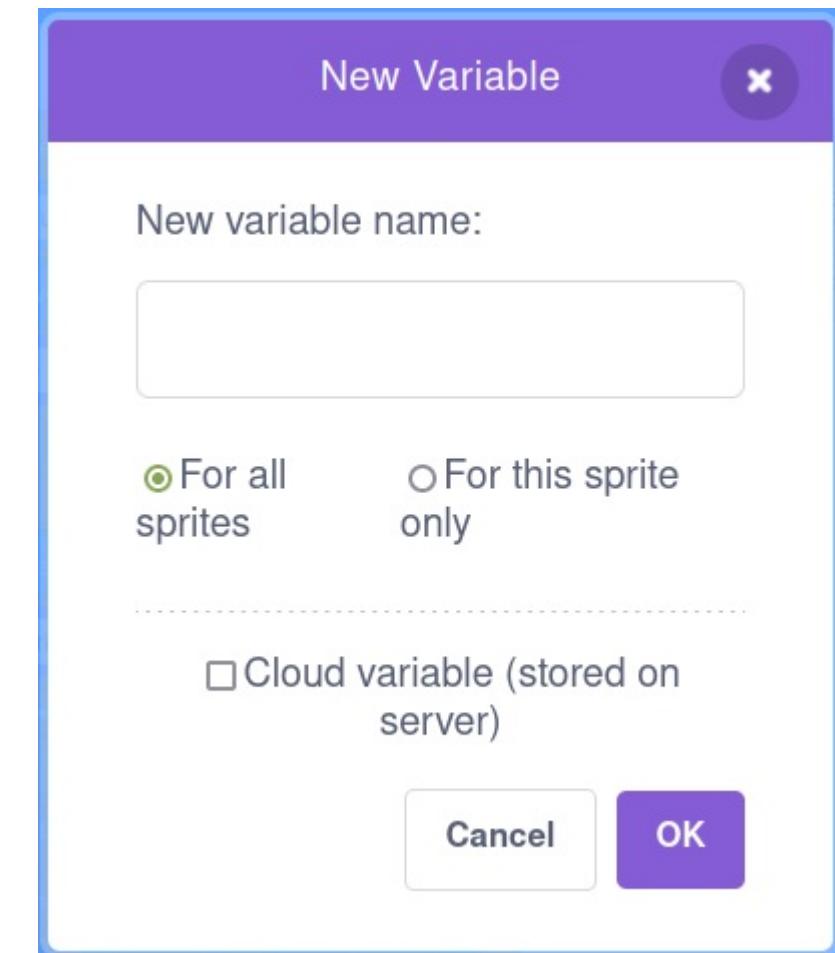
- **Variável:** É um **espaço na memória** para o **armazenamento de um único valor**. Pode corresponder a uma ou mais células de memória. Algumas linguagens obrigam o usuário a escolher o **tipo de dado** que será armazenado (inteiro, real, caractere, etc). No Scratch não é necessário escolher o tipo de dado a ser armazenado. Existem 2 tipos no Scratch:
  - variável **global** (visível em todos os sprites)
  - variável **local** (visível somente em 1 sprite)
- **Array:** Também chamado de **lista** ou **vetor**, é uma estrutura de dados que **armazena mais de um valor ao mesmo tempo**. Utilizado em situações nas quais precisamos trabalhar com muitos dados semelhantes. Também em 2 tipos:
  - array **global** (visível em todos os sprites)
  - array **local** (visível somente em 1 sprite)

# Variáveis



answer  
contador

**Ao criar variáveis no Scratch, você deve escolher se elas serão visíveis em todos os sprites (**global**) ou somente no sprite atual (**local**).**

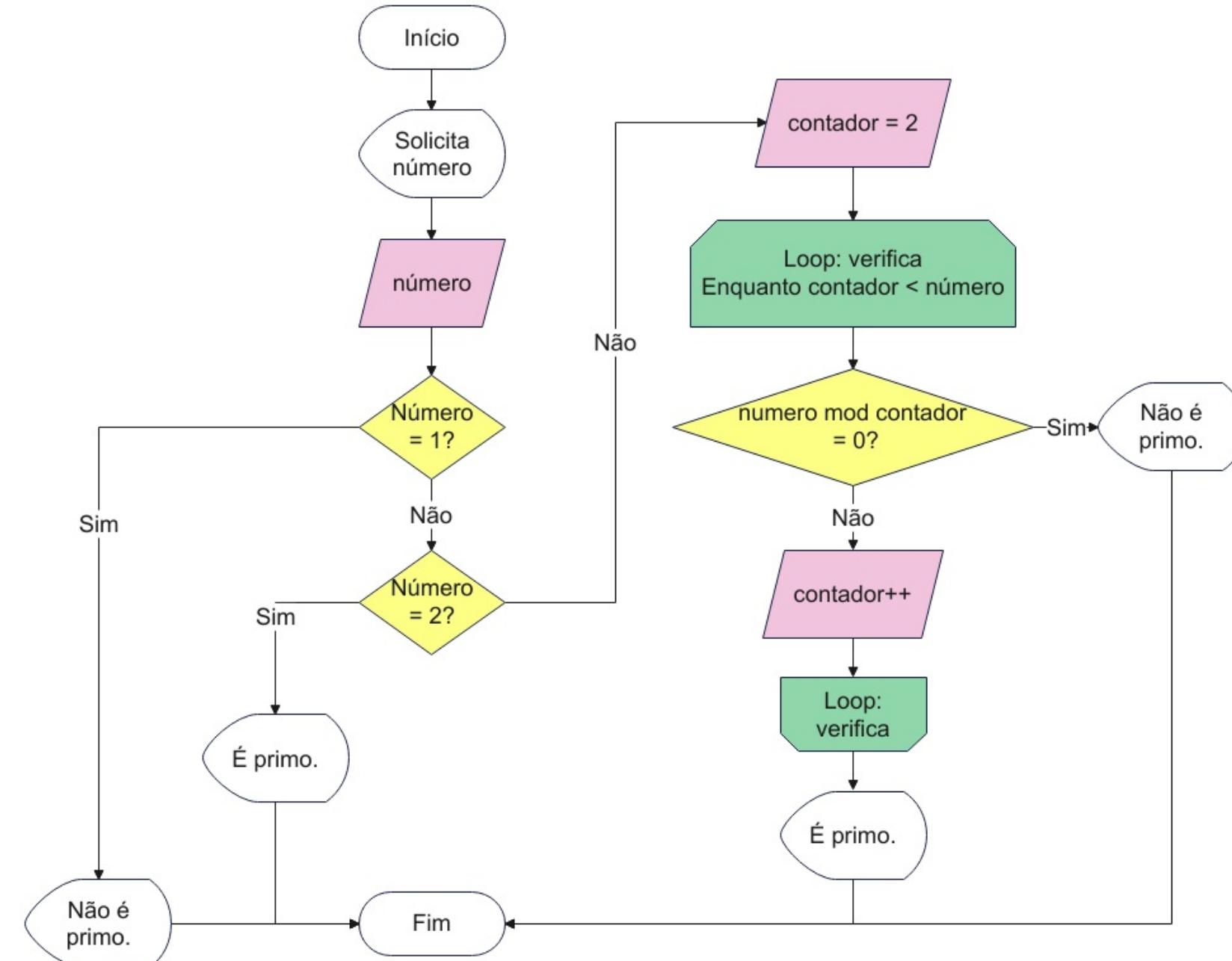


<https://scratch.mit.edu/projects/975855697/>

# Variáveis: entendendo este código

```
when green flag clicked
  ask [Informe um número inteiro positivo:] and wait
  if [answer = 1] then
    say [Não é primo.] for [2] seconds
    stop [all]
  else
    if [answer = 2] then
      say [É primo.] for [2] seconds
      stop [all]
    else
      set [contador] to [2]
      repeat until [contador > answer - 1]
        if [answer mod contador = 0] then
          say [Não é primo.] for [2] seconds
          stop [all]
        change [contador] by [-1]
      end
      say [É primo.] for [2] seconds
  end
end
```

Solicita número.

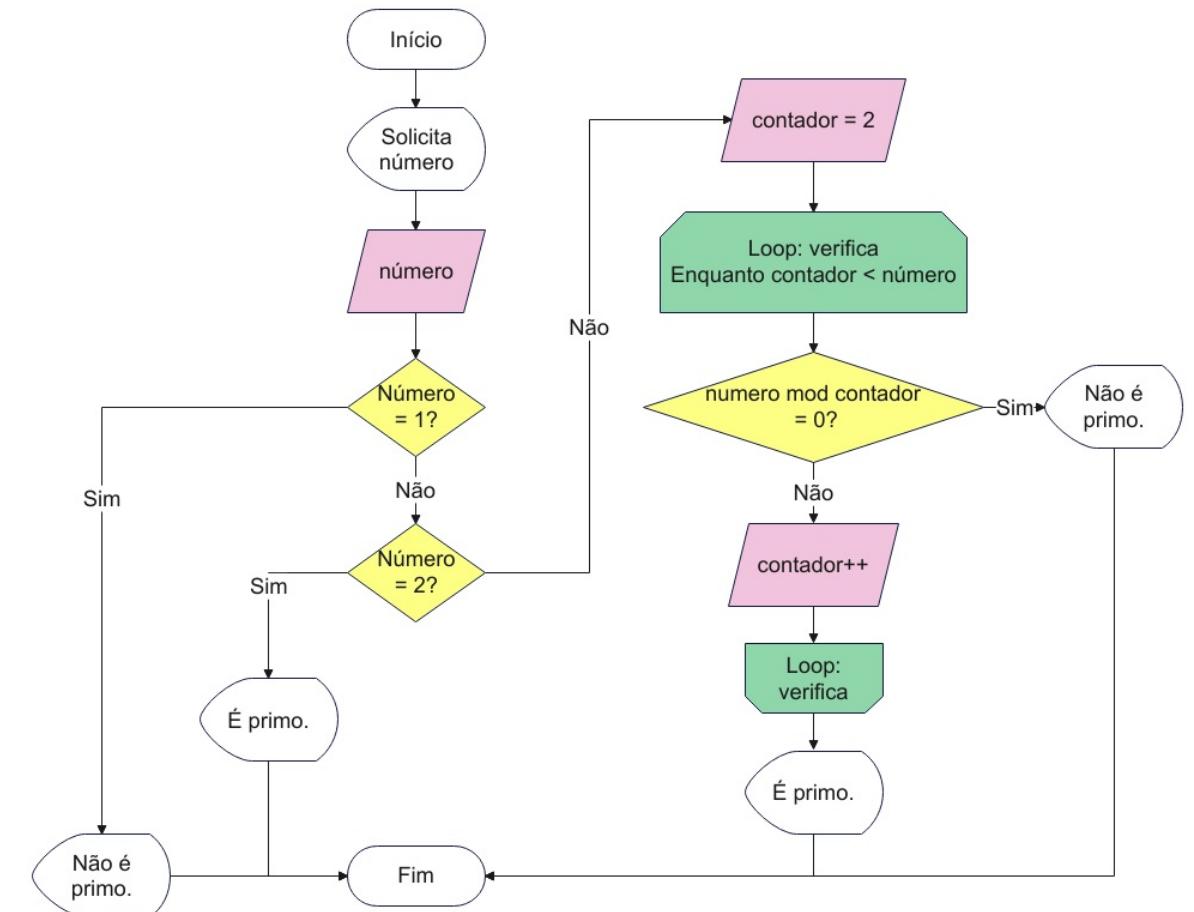


# Variáveis: entendendo este código

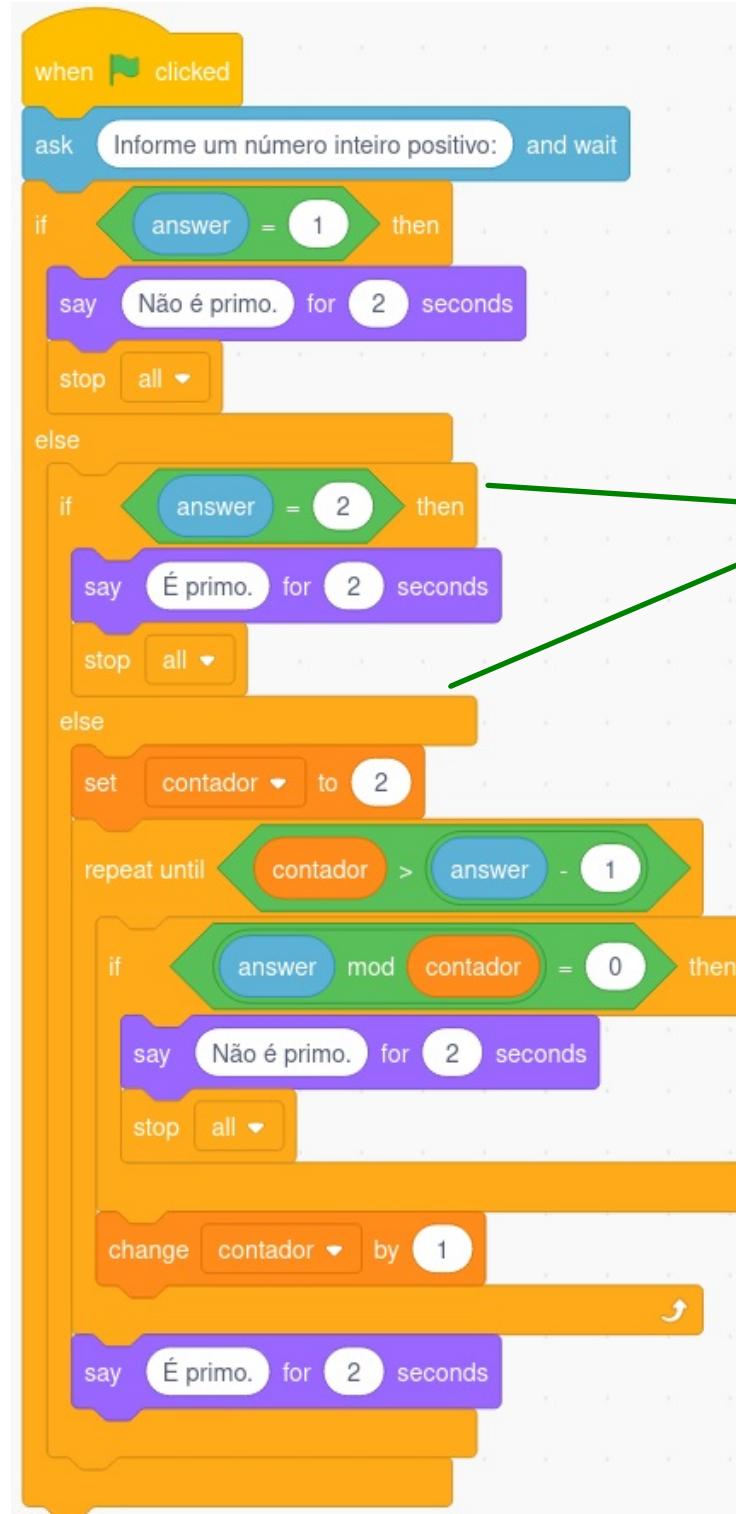
A Scratch script starting with "when green flag clicked". It asks for a number and checks if it's 1 or 2. If 1 or 2, it says it's prime. Otherwise, it sets a counter to 2 and loops until the counter is greater than the number. Inside the loop, it checks if the number is divisible by the counter. If yes, it says it's not prime and stops. If no, it increments the counter and continues the loop. Finally, it says the number is prime.

```
when green flag clicked
ask [Informe um número inteiro positivo:] and wait
if [answer] = [1] then
  say [Não é primo.] for [2] seconds
  stop [all v.]
else
  if [answer] = [2] then
    say [É primo.] for [2] seconds
    stop [all v.]
  else
    set [contador v.] to [2]
    repeat until [contador > answer - 1]
      if [answer mod contador = 0] then
        say [Não é primo.] for [2] seconds
        stop [all v.]
      end
      change [contador v.] by [1]
    end
    say [É primo.] for [2] seconds
  end
end
```

Verifica se o número é 1. Se for, diz que não é primo e termina o programa.

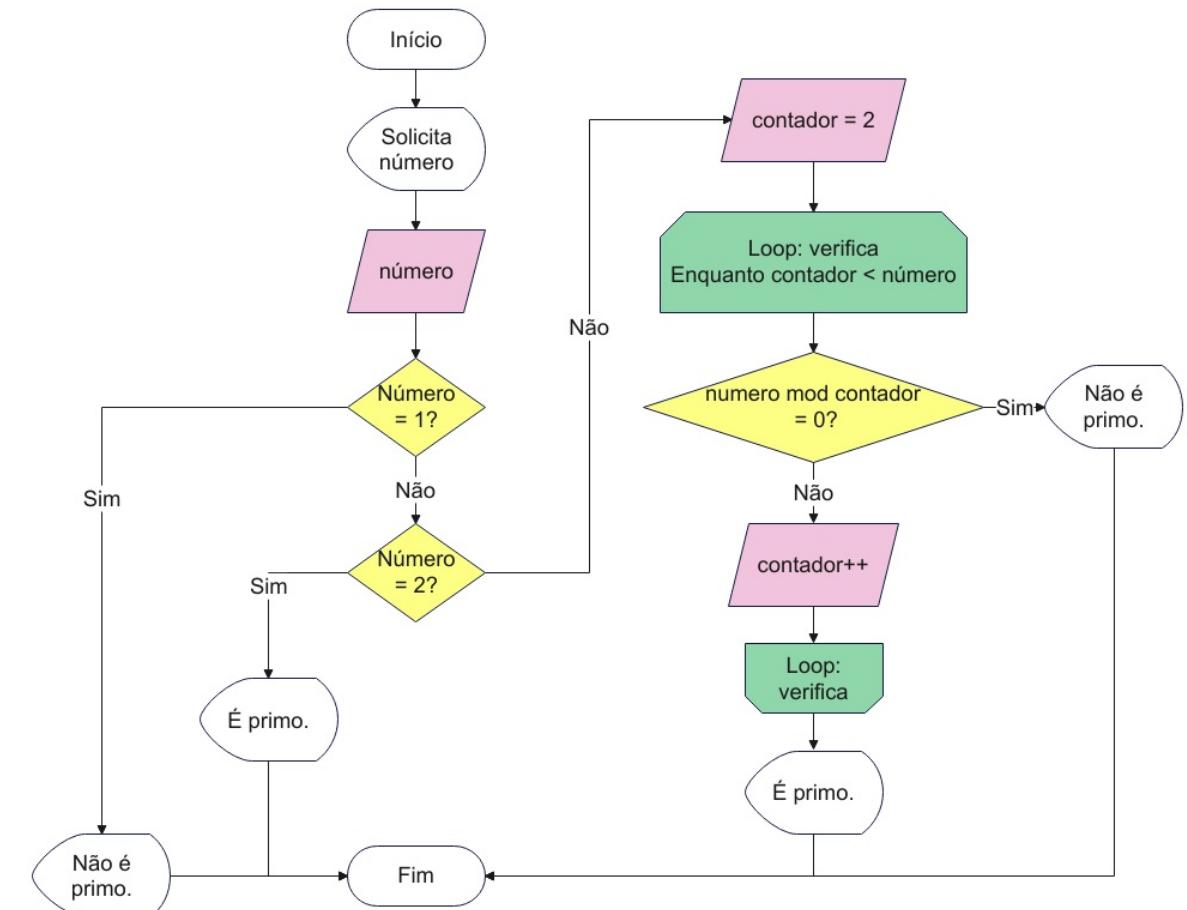


# Variáveis: entendendo este código

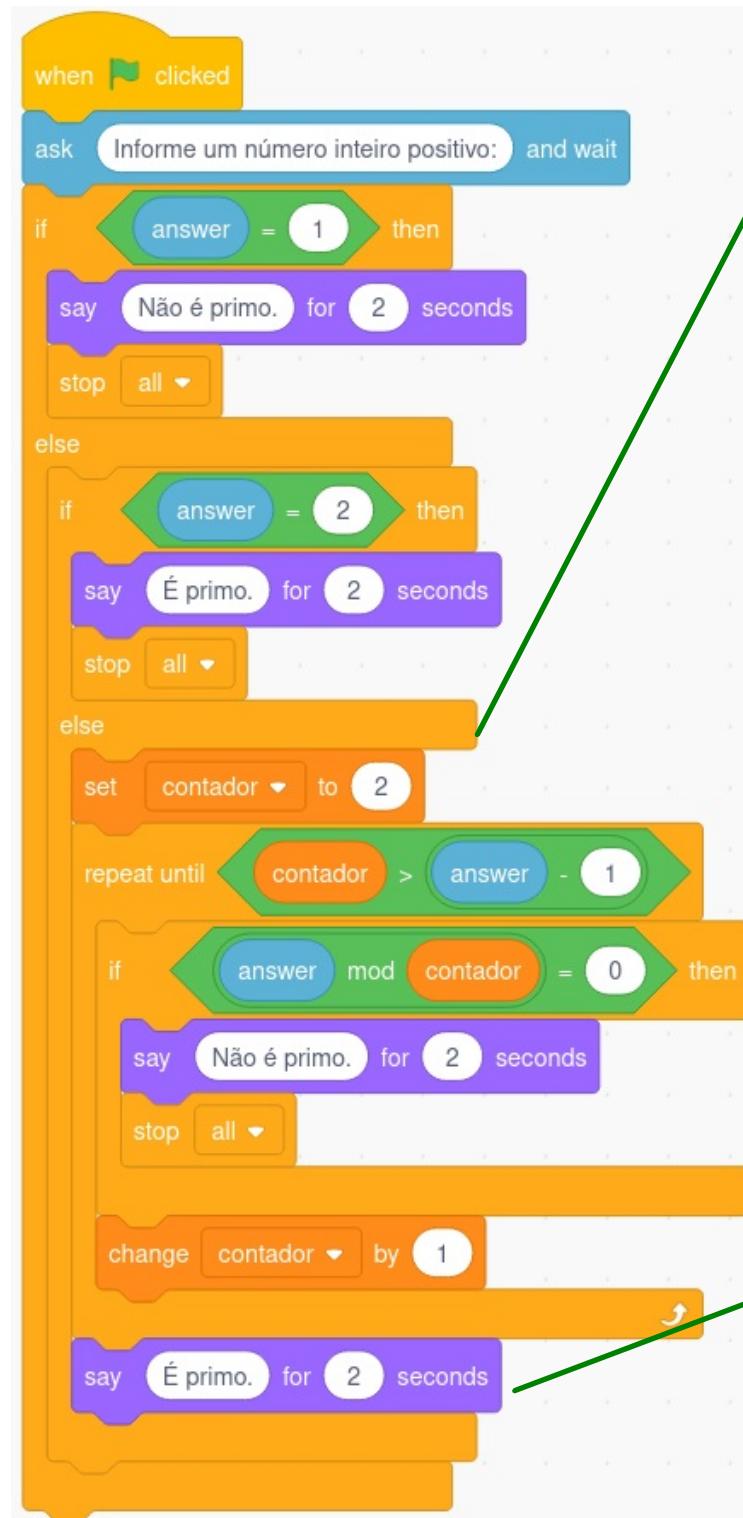


A Scratch script starting with "when green flag clicked". It asks for a number, then checks if it's 1 or 2. If 1, it says "Não é primo." for 2 seconds. If 2, it says "É primo." for 2 seconds. Otherwise, it sets a counter to 2 and enters a loop until the counter is greater than the answer minus 1. Inside the loop, it checks if the answer is divisible by the counter. If yes, it says "Não é primo." for 2 seconds and stops. If no, it increments the counter by 1 and loops back. Finally, it says "É primo." for 2 seconds.

Verifica se o número é 2. Se for, diz que é primo e termina o programa.



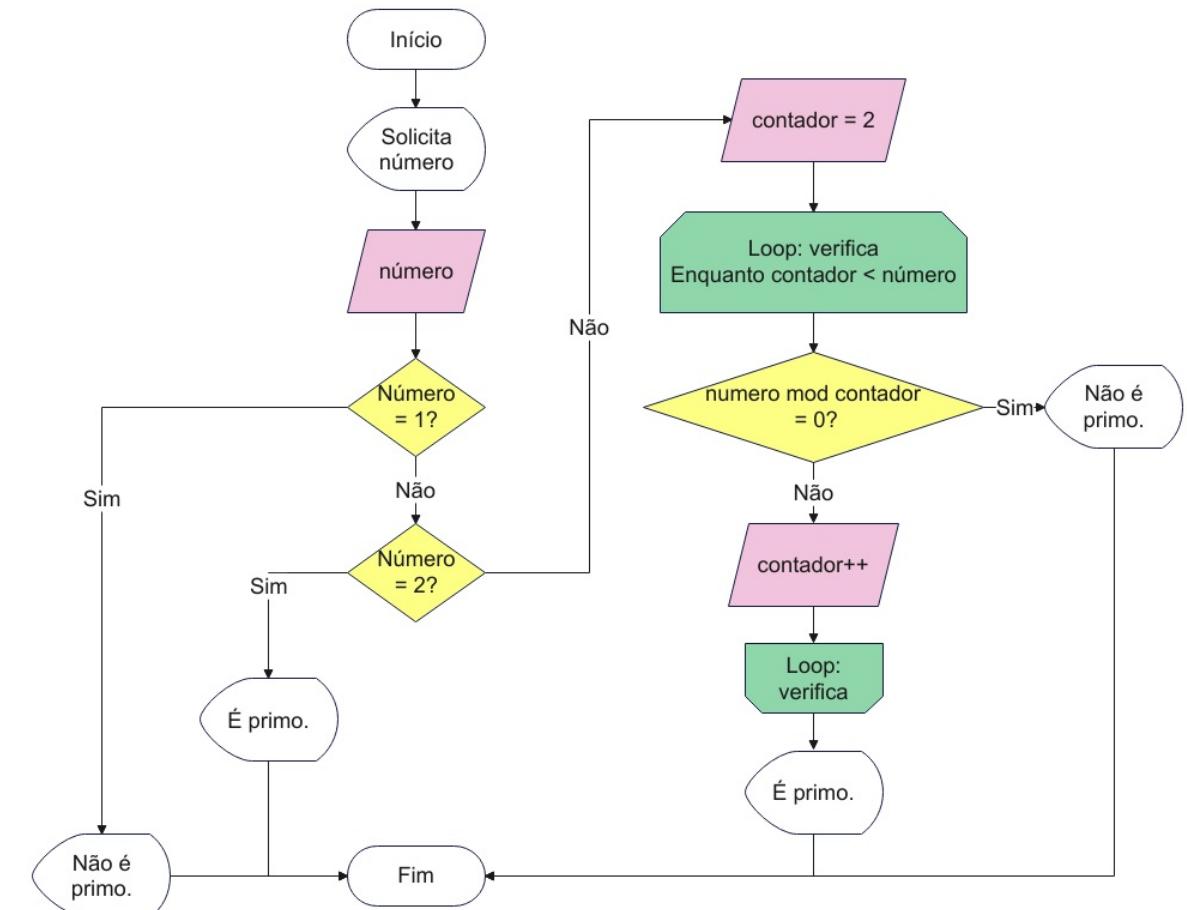
# Variáveis: entendendo este código



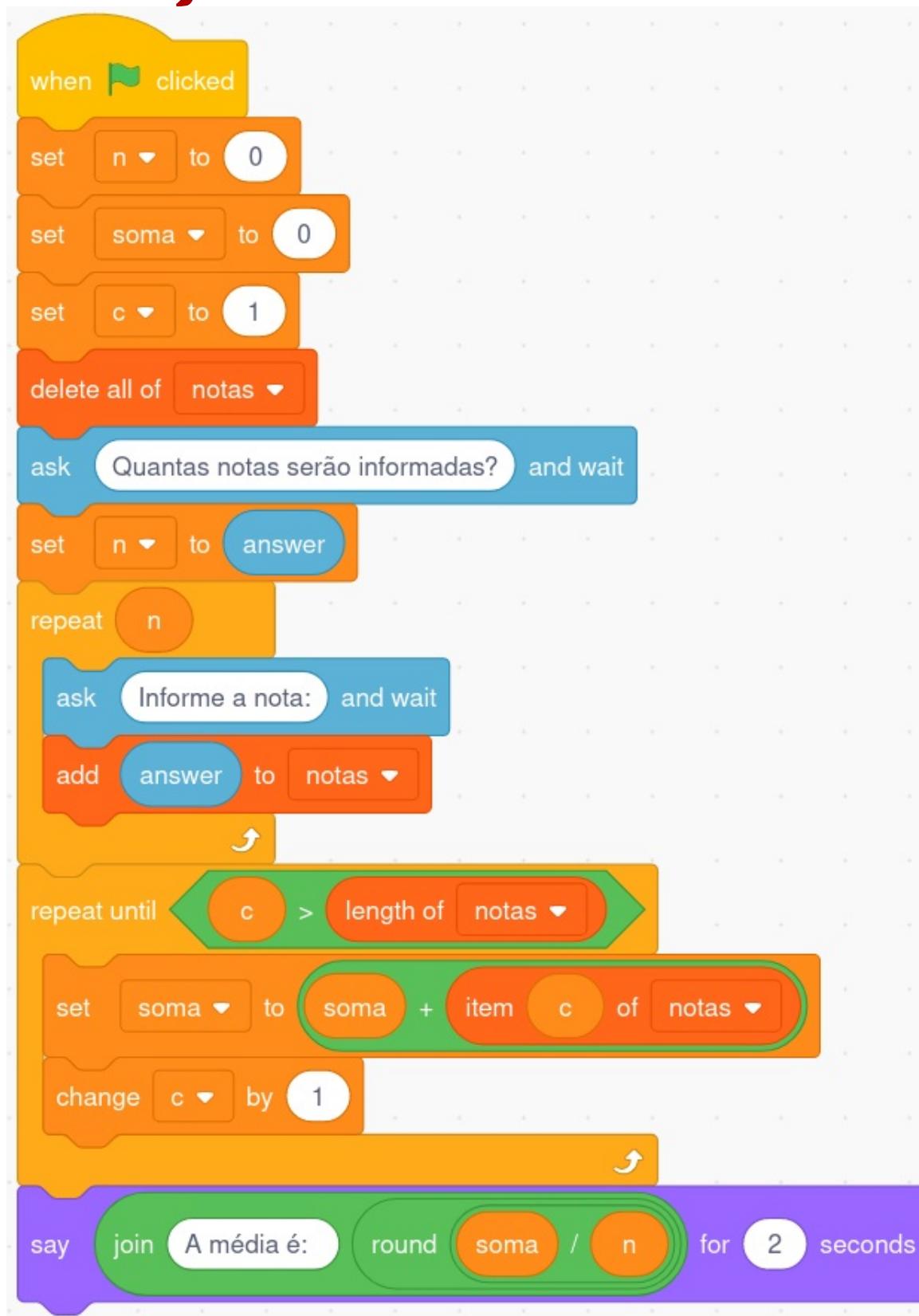
A Scratch script starting with "when green flag clicked". It asks for a number, then checks if it's 1 or 2. If not, it initializes a counter to 2 and loops until the counter is greater than the user's number. Inside the loop, it checks if the user's number is divisible by the counter. If yes, it says "Não é primo." and stops. If no, it increments the counter and continues the loop. Finally, it says "É primo." and stops.

```
when green flag clicked
ask [Informe um número inteiro positivo:] and wait
if [answer = 1] then
  say [Não é primo.] for [2] seconds
  stop [all v]
else
  if [answer = 2] then
    say [É primo.] for [2] seconds
    stop [all v]
  else
    set [contador v] to [2]
    repeat until [contador > answer - 1]
      if [answer mod contador = 0] then
        say [Não é primo.] for [2] seconds
        stop [all v]
      end
      change [contador v] by [1]
    end
    say [É primo.] for [2] seconds
  end
end
```

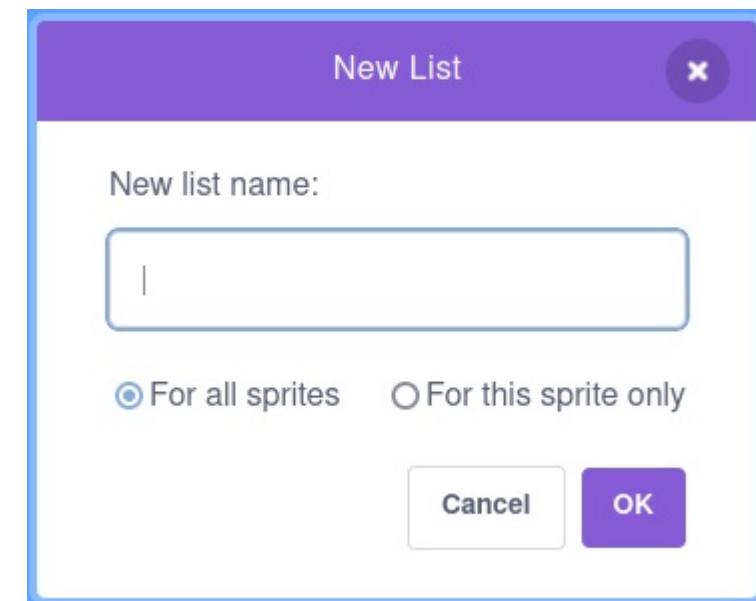
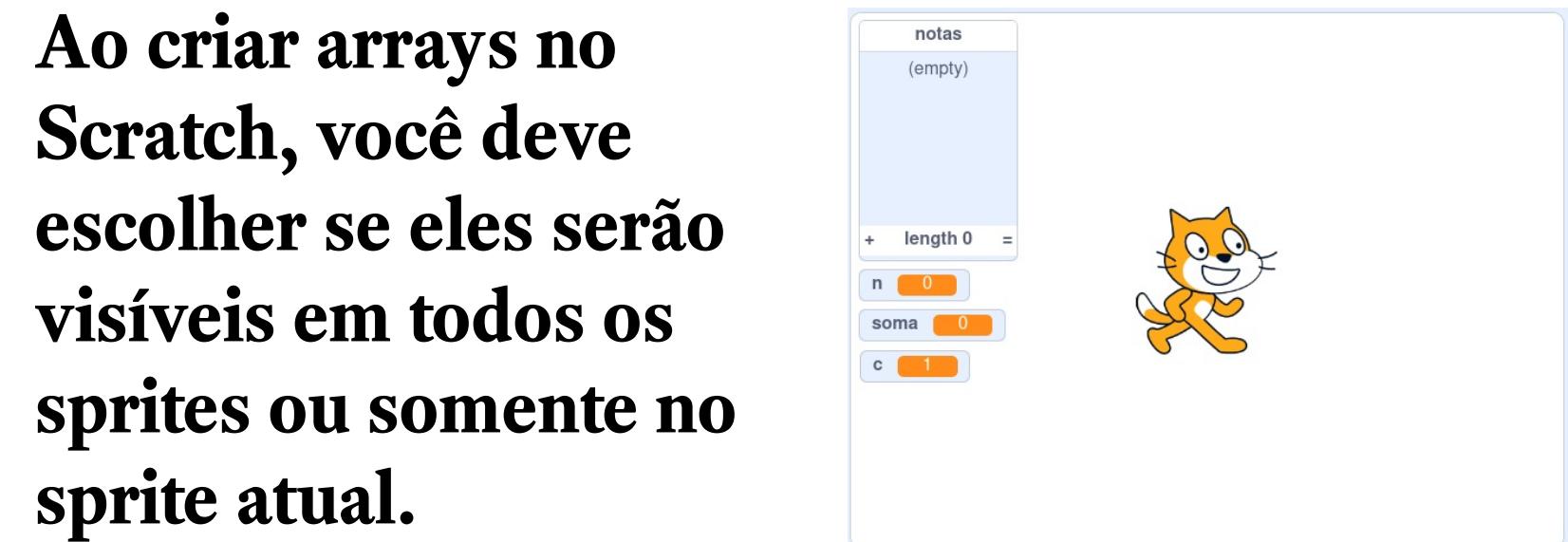
**Se o número não for 1 ou 2, usamos a variável "contador" para verificar o resto da divisão do número informado pelo usuário por todos os números entre 2 e o antecessor do número informado. Se encontrarmos resto 0 então imprime que não é primo e termina. Caso contrário imprime que é primo e termina.**



# Arrays

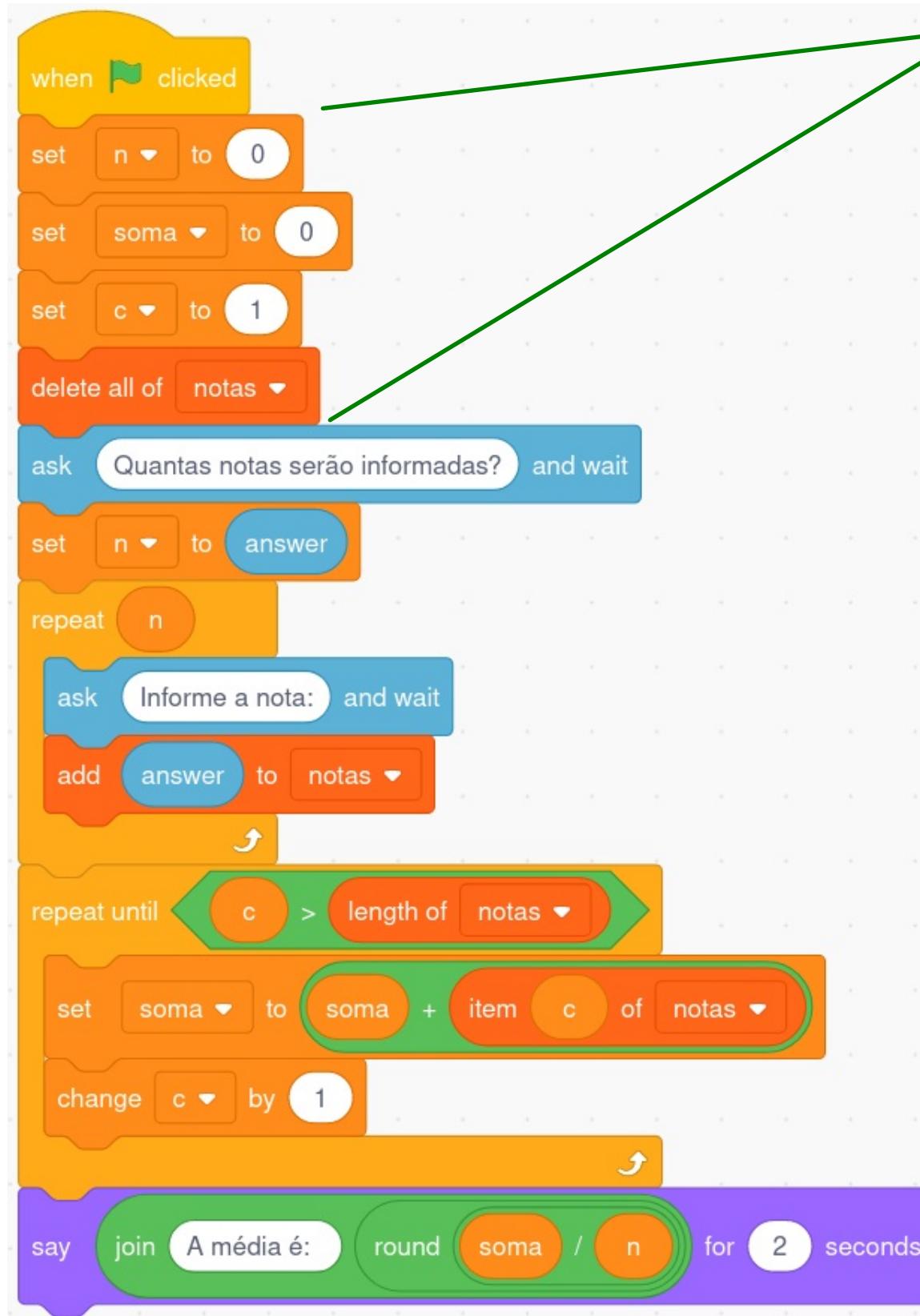


**Ao criar arrays no Scratch, você deve escolher se eles serão visíveis em todos os sprites ou somente no sprite atual.**



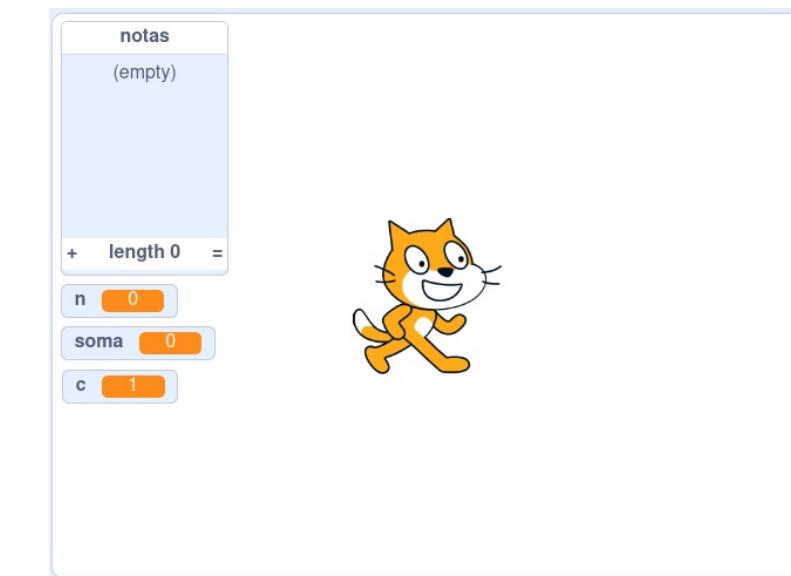
<https://scratch.mit.edu/projects/975857248/>

# Arrays: entendendo este código



A Scratch script starting with a green flag click. It initializes variables n (0), soma (0), and c (1). It then removes all items from the 'notas' array. An ask block prompts for the number of notes (n). A repeat loop begins, asking for each note (c) and adding it to the 'notas' array. The loop continues until c is greater than the length of the 'notas' array. Finally, it calculates the average (soma / n) and says "A média é: " followed by the result.

Depois de **declarar** as variáveis necessárias, **atribuímos** os valores iniciais necessários.



## Termos importantes:

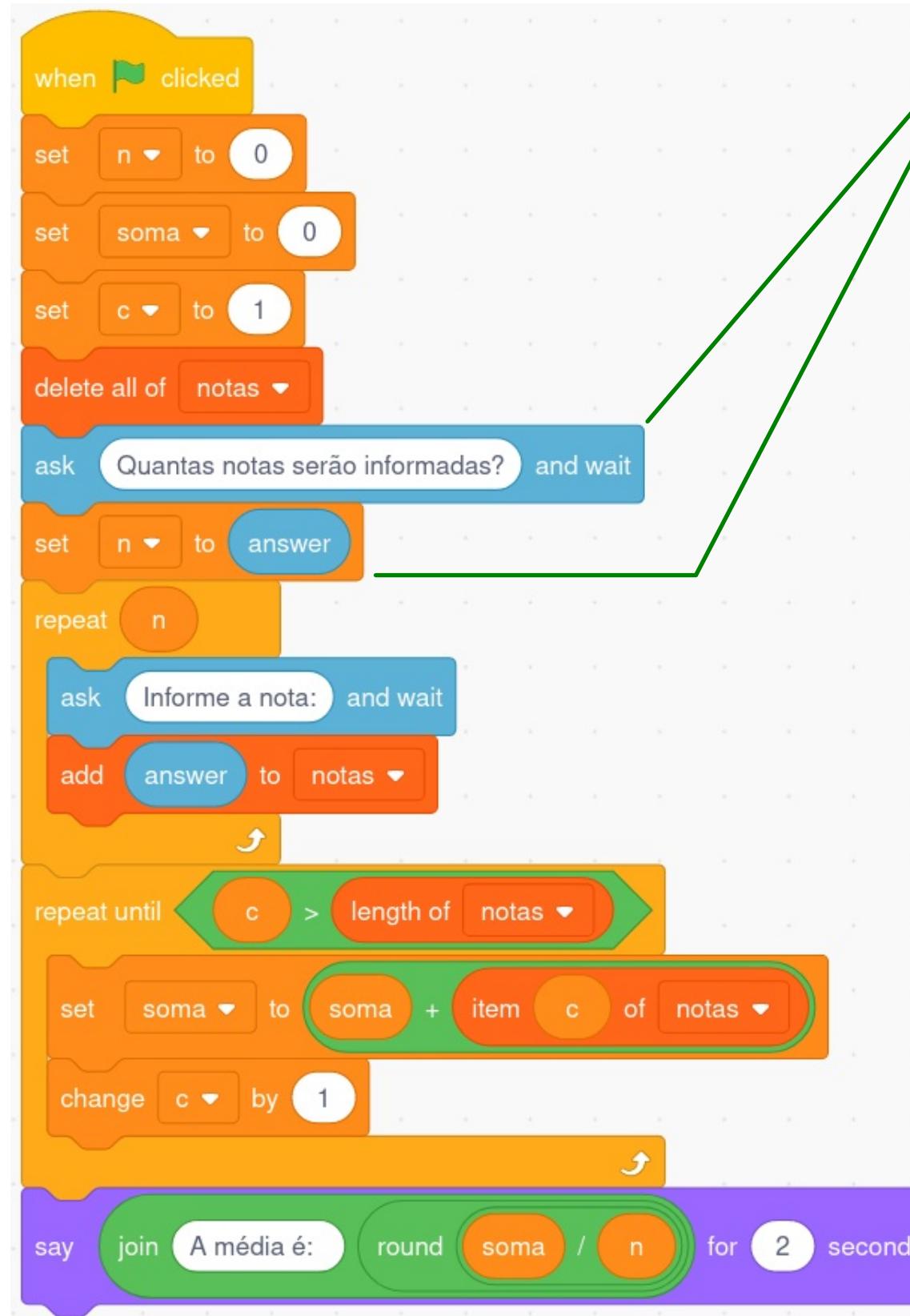
**Declarar:**

significa "criar" a variável, informar ao programa que ela existe.

**Atribuir:**

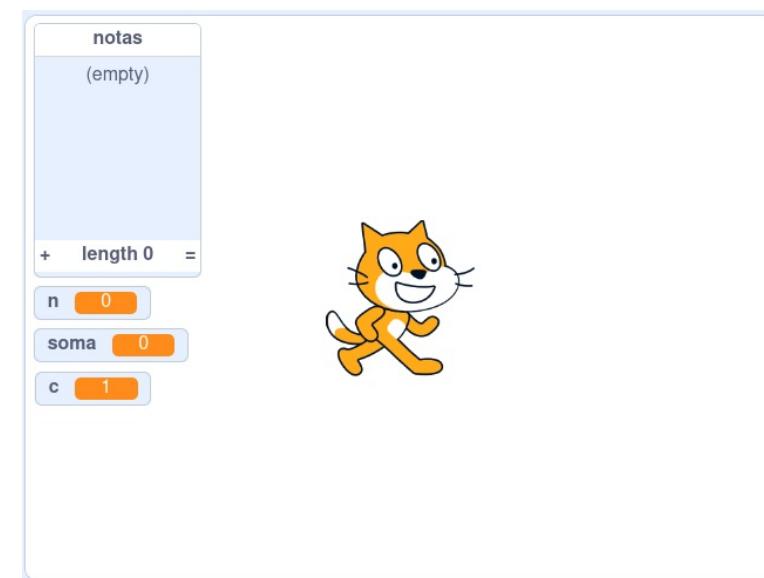
significa dar um valor (ou trocar o valor) à variável.

# Arrays: entendendo este código



A Scratch script starting with a green flag click. It initializes variables n (0), soma (0), and c (1). It then deletes all items from the 'notas' array. An ask block prompts the user for the number of notes (n). A repeat loop begins, asking for each note and adding it to the 'notas' array. The loop continues until the value of c is greater than the length of the 'notas' array. Finally, it calculates the average (soma / n) and says "A média é: [average]" for 2 seconds.

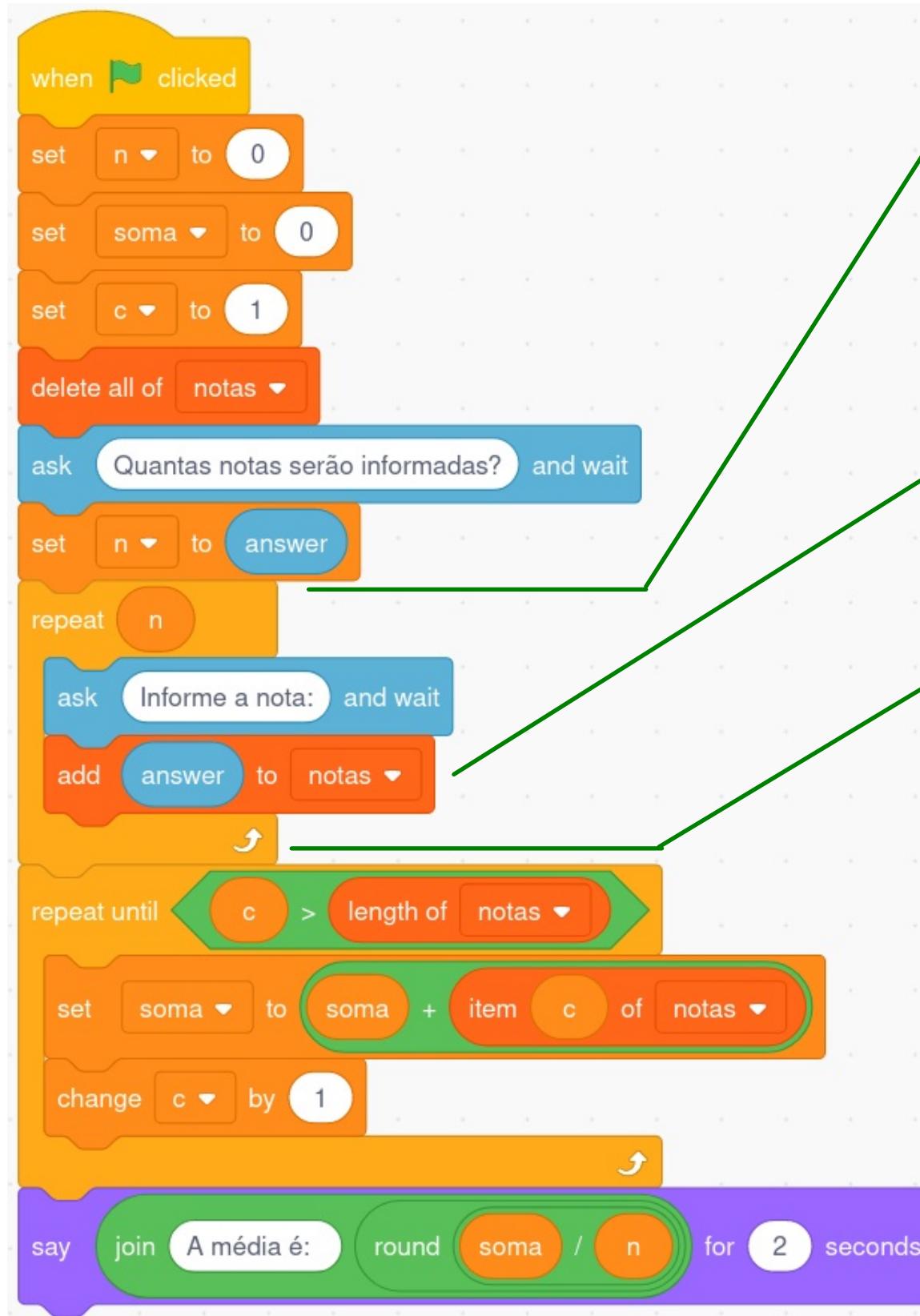
Perguntamos ao usuário quantas notas ele irá informar.



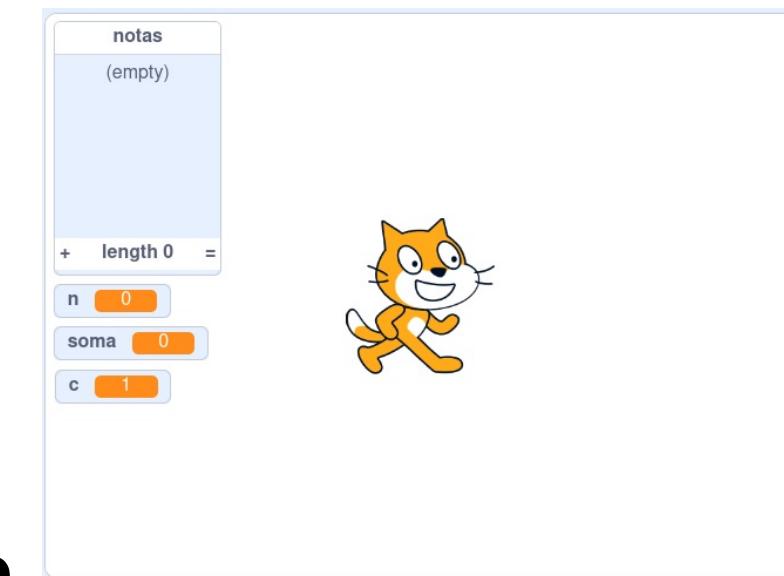
Isso é importante para solicitarmos que o usuário informe a quantidade correta de notas.

Nós atribuímos esse valor à variável **n**.

# Arrays: entendendo este código



Agora que sabemos n,  
basta um loop para  
solicitar cada nota  
individual ao usuário.

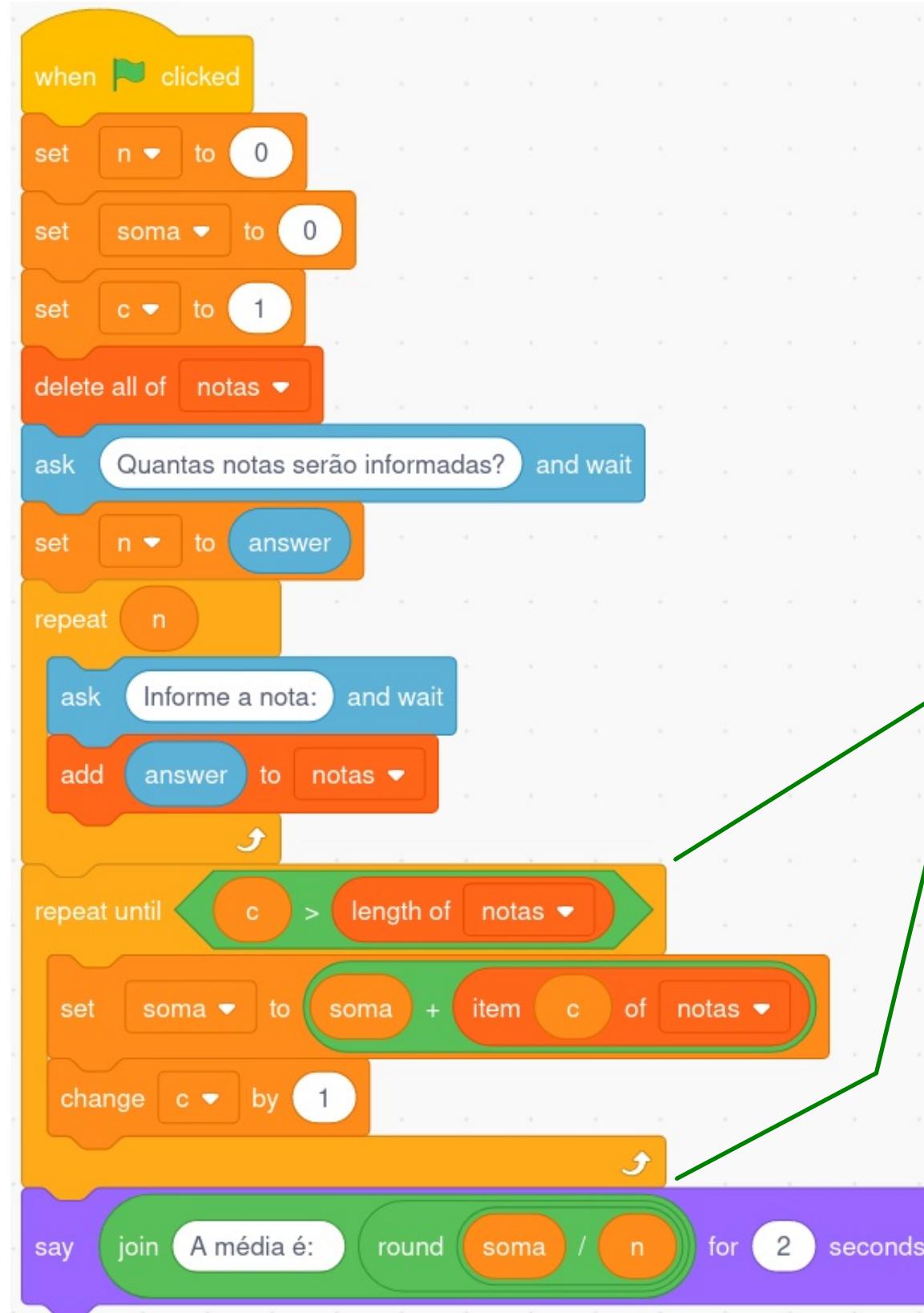


Note que após o usuário  
informar cada nota, estamos **includindo** a nota  
informada ao array.

Atenção: o limite de notas no array  
é dado pelo limite **n** de repetições,  
não pela capacidade do array (que  
poderia armazenar mais notas).

No Scratch não precisamos fixar um  
tamanho para os arrays.

# Arrays: entendendo este código



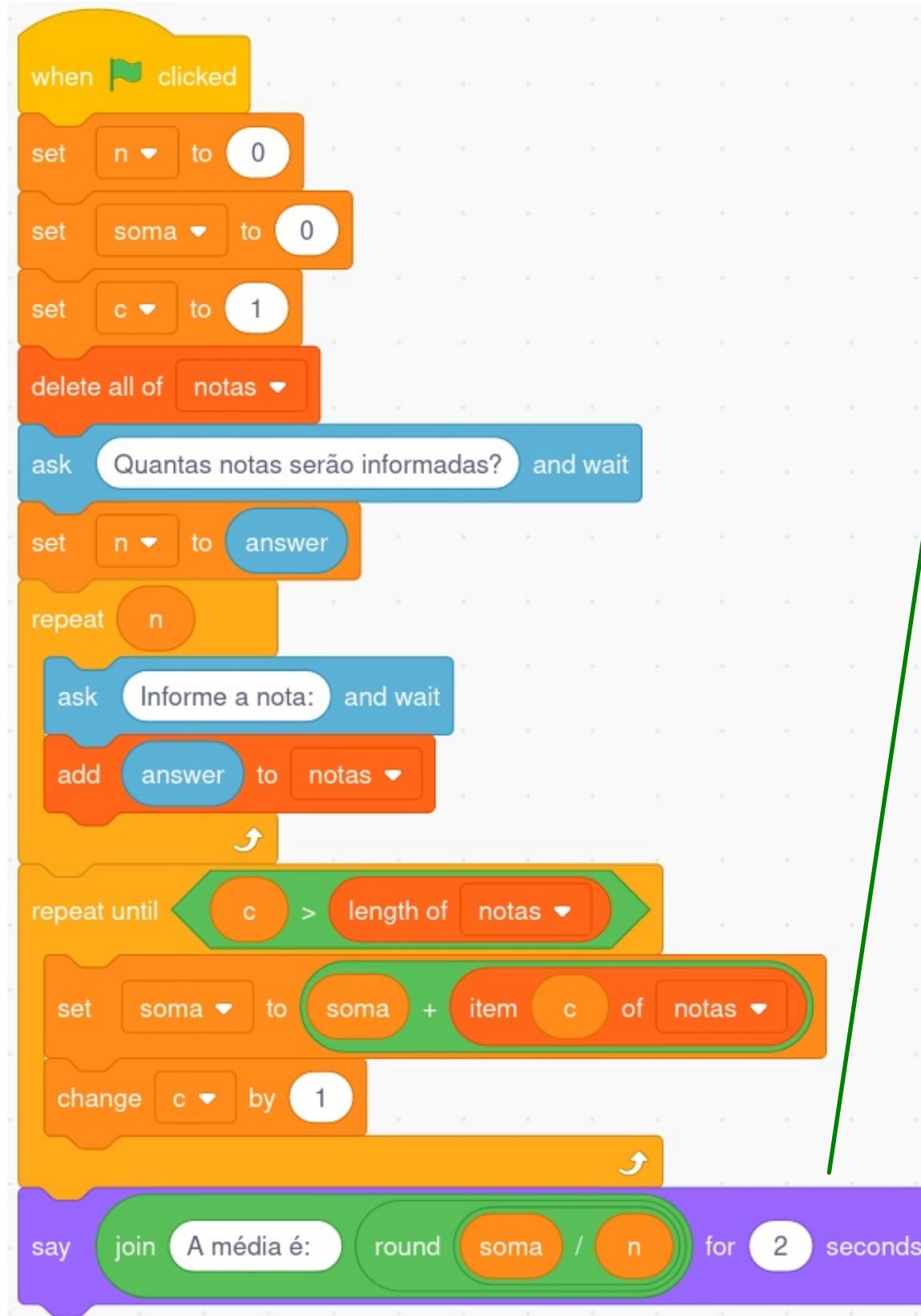
Agora temos que percorrer o array e fazer a soma de todas as notas do array à variável "soma", que criamos apenas para acumular esse resultado.

Isso é um "**idiom**" (idioma) comum na programação:  
percorrer um array acumulando uma soma:  
 $soma = soma + valor$

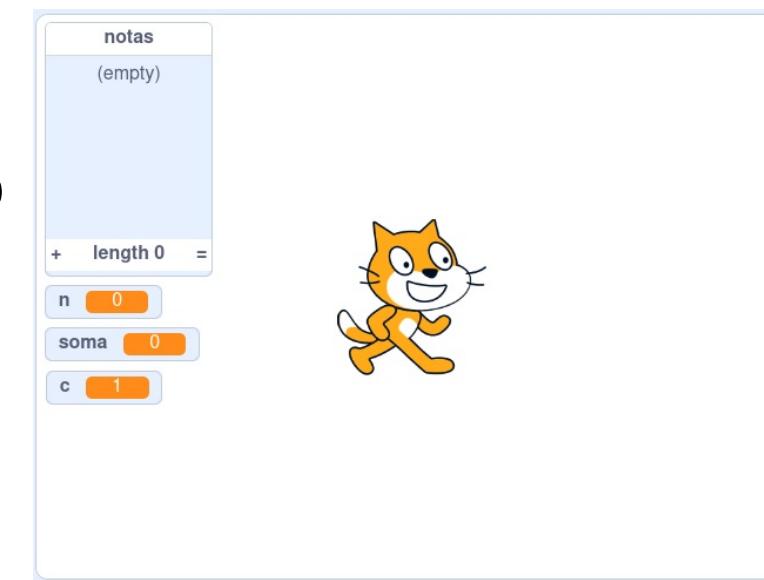
Para entender o que ocorre, faça sempre um **teste de mesa**!

<https://programming-idioms.org>

# Arrays: entendendo este código



Por fim, imprimimos a média das notas fazendo uma divisão da soma pelo número de notas.



Obs.: note que a média está sendo informada com a função **round**, que arredonda para um número inteiro.

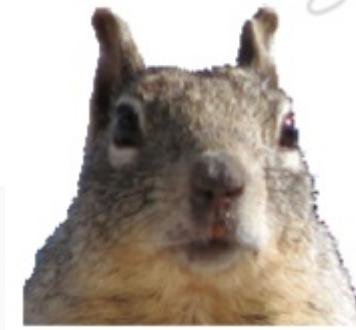
# Ao usar variáveis, cuidado com situações especiais

Cansei...

Alguns problemas, como **overflow**, podem ocorrer:



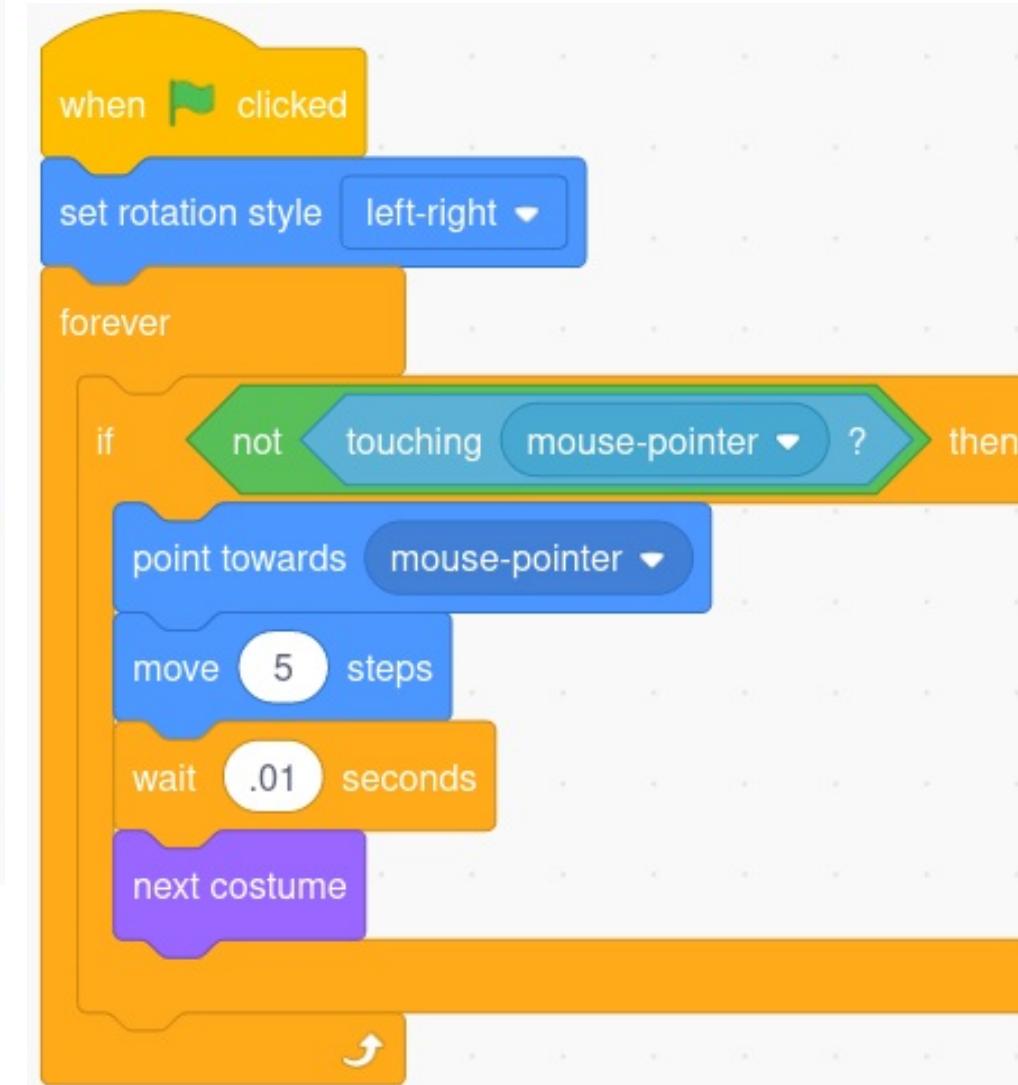
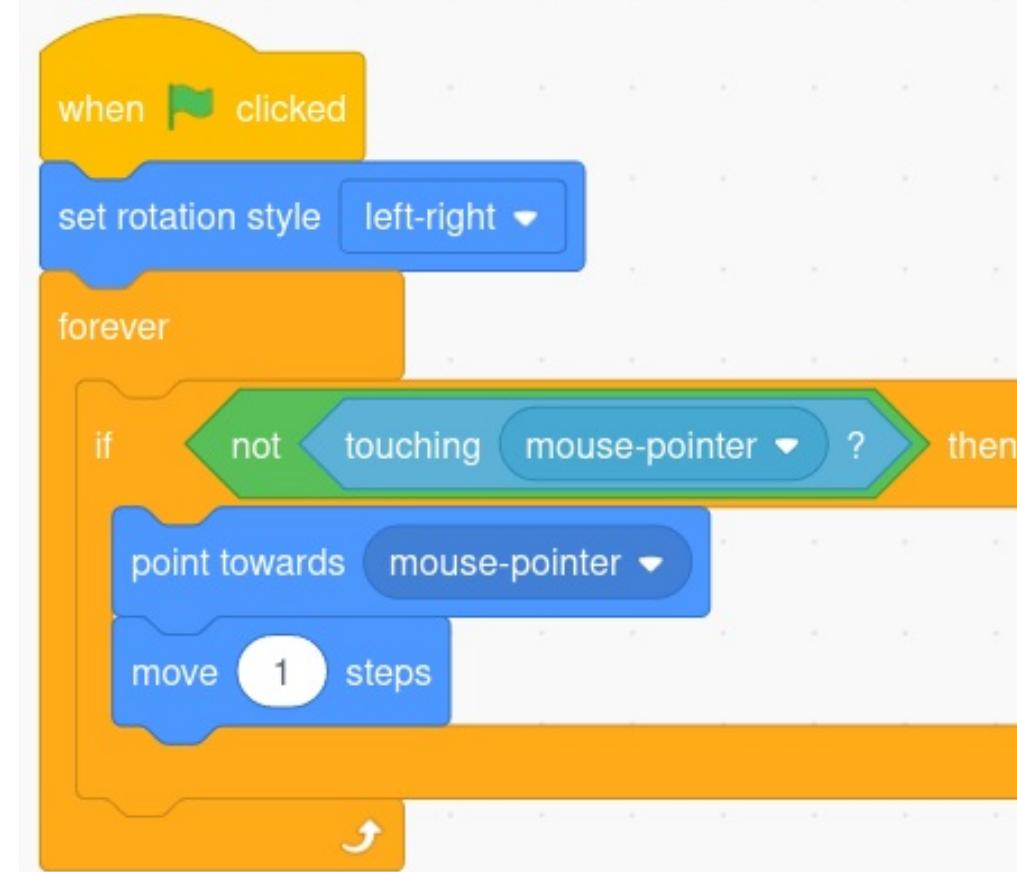
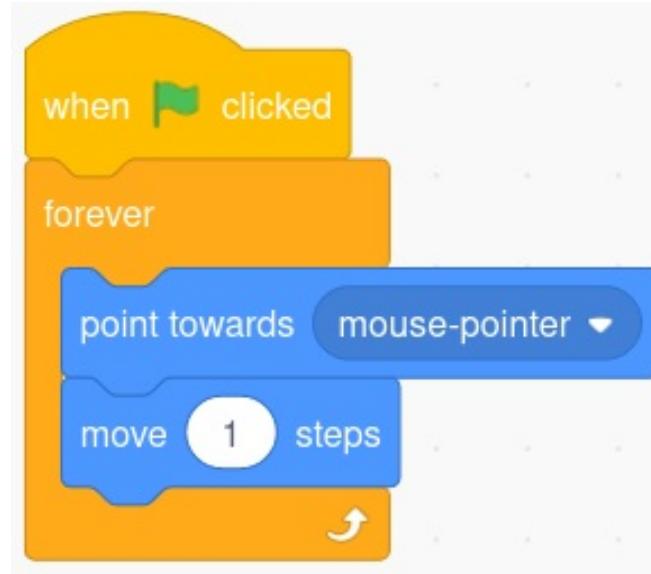
<https://scratch.mit.edu/projects/997034544/>



**Você deve validar a  
entrada do usuário!**

<https://scratch.mit.edu/projects/984134758/>

# Movimentação, fantasias

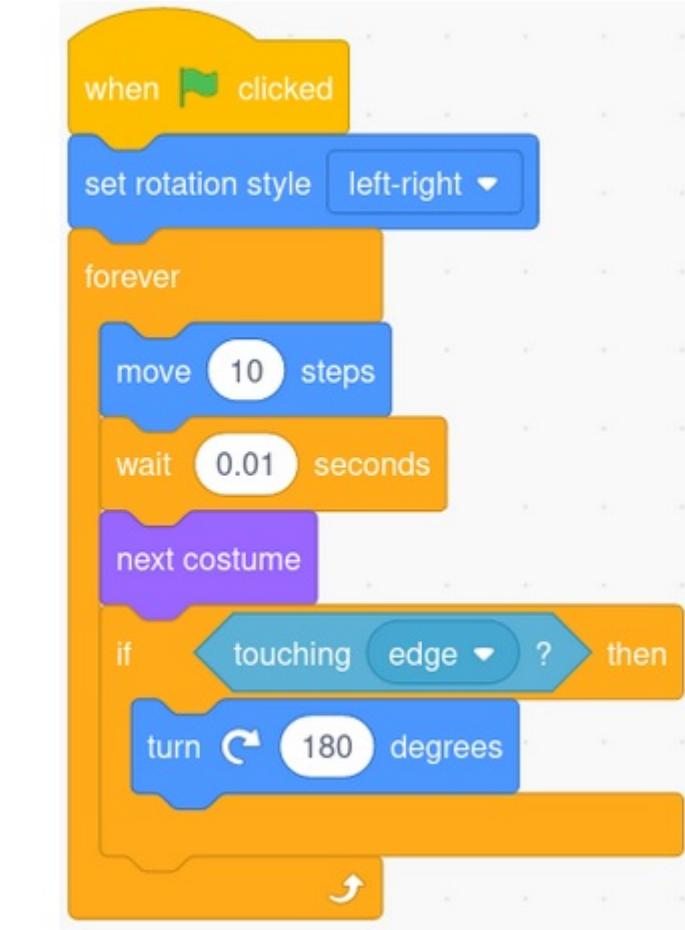
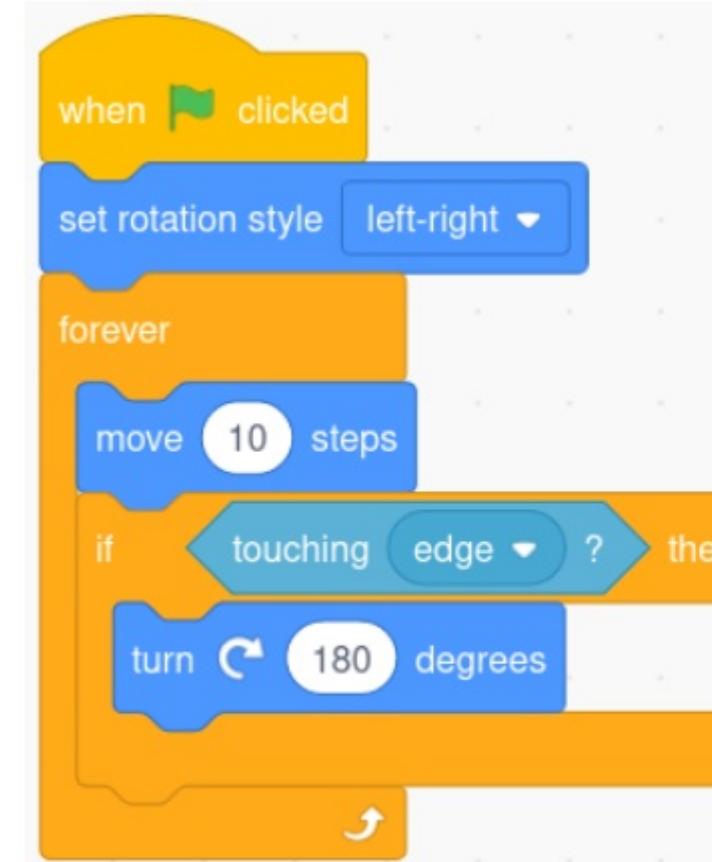
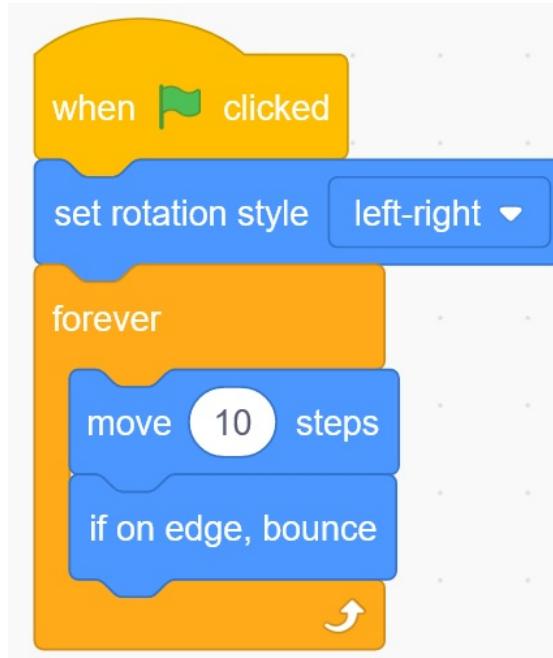


<https://scratch.mit.edu/projects/807108713/>

<https://scratch.mit.edu/projects/807108956/>

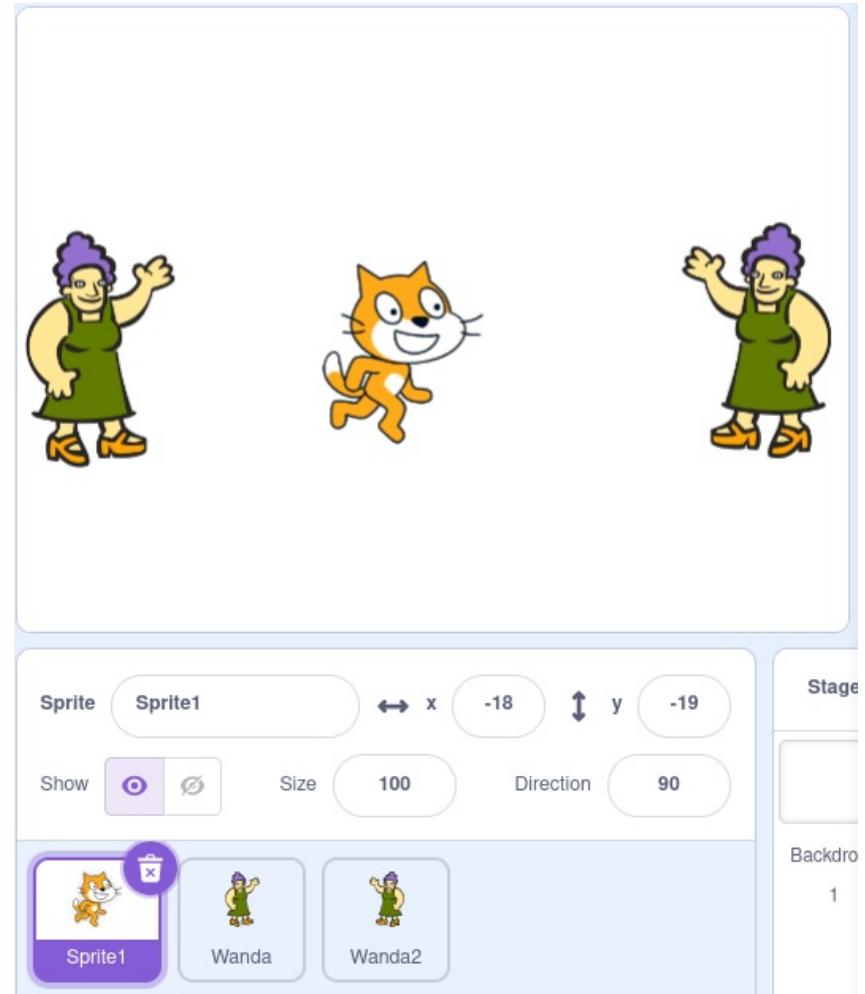
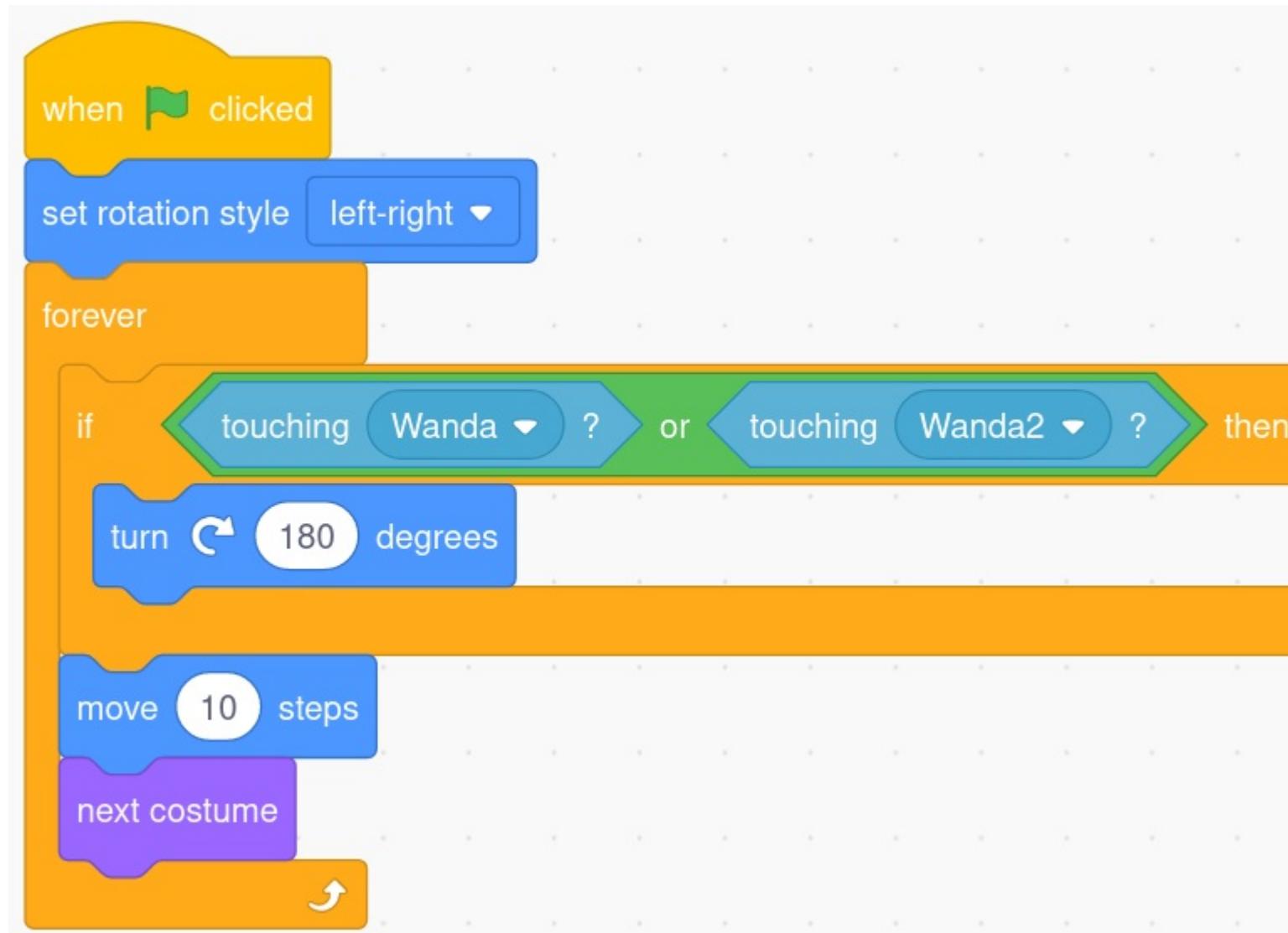
<https://scratch.mit.edu/projects/997041209/>

# Limits do palco



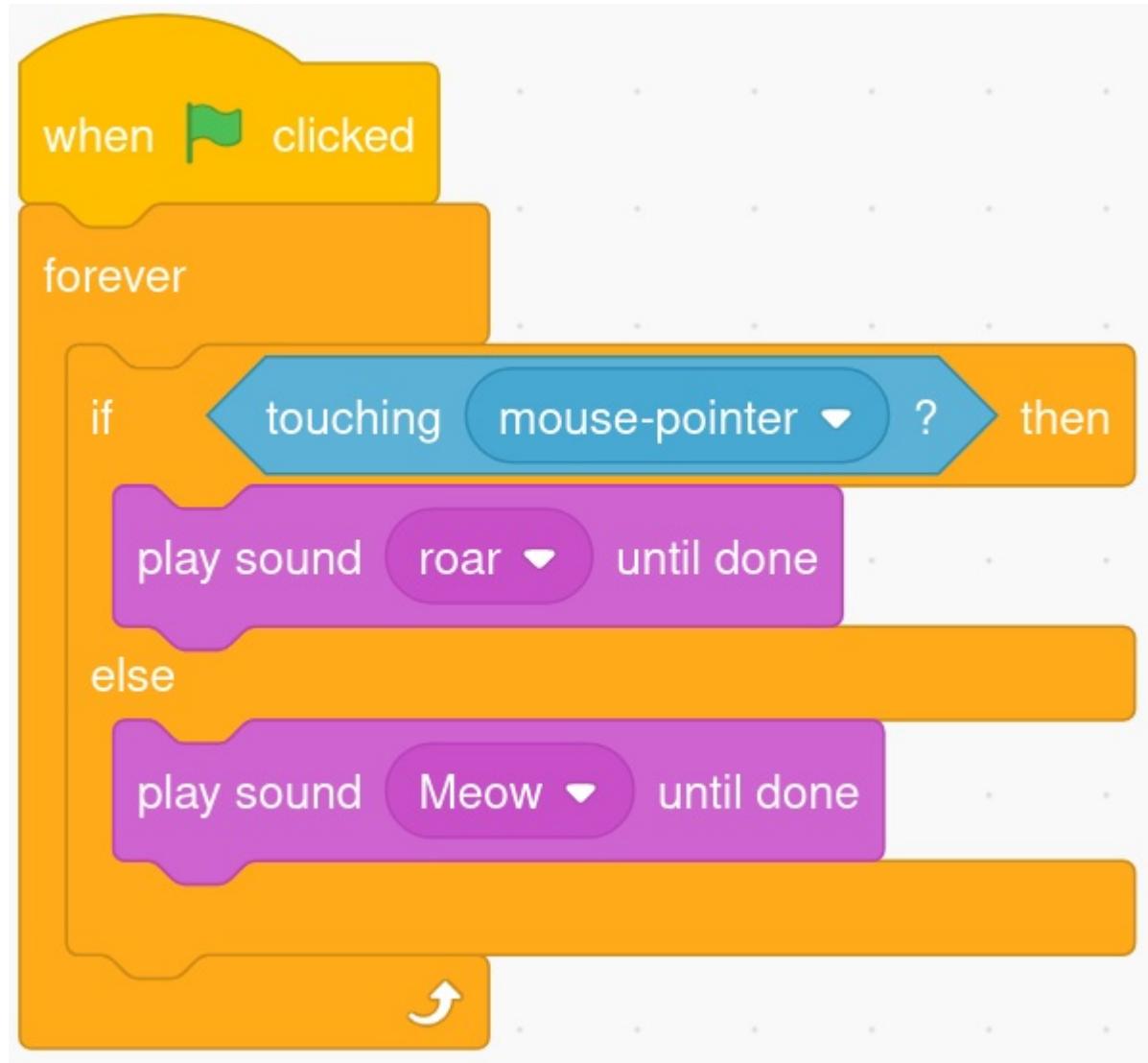
<https://scratch.mit.edu/projects/981880176/>  
<https://scratch.mit.edu/projects/807109974/>  
<https://scratch.mit.edu/projects/807110248/>

# Outros personagens



<https://scratch.mit.edu/projects/984138623>

# Sons

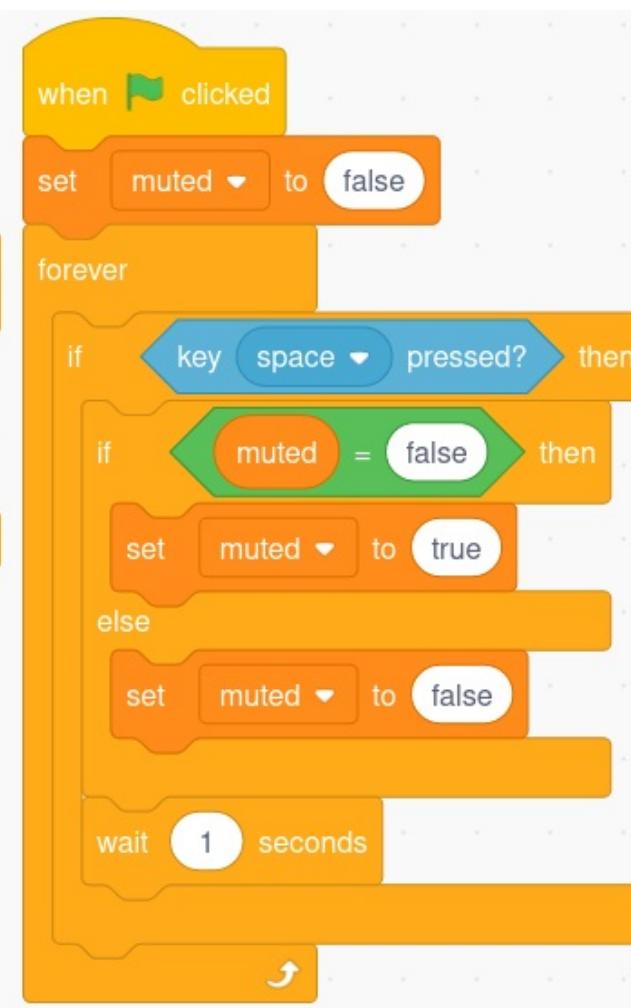
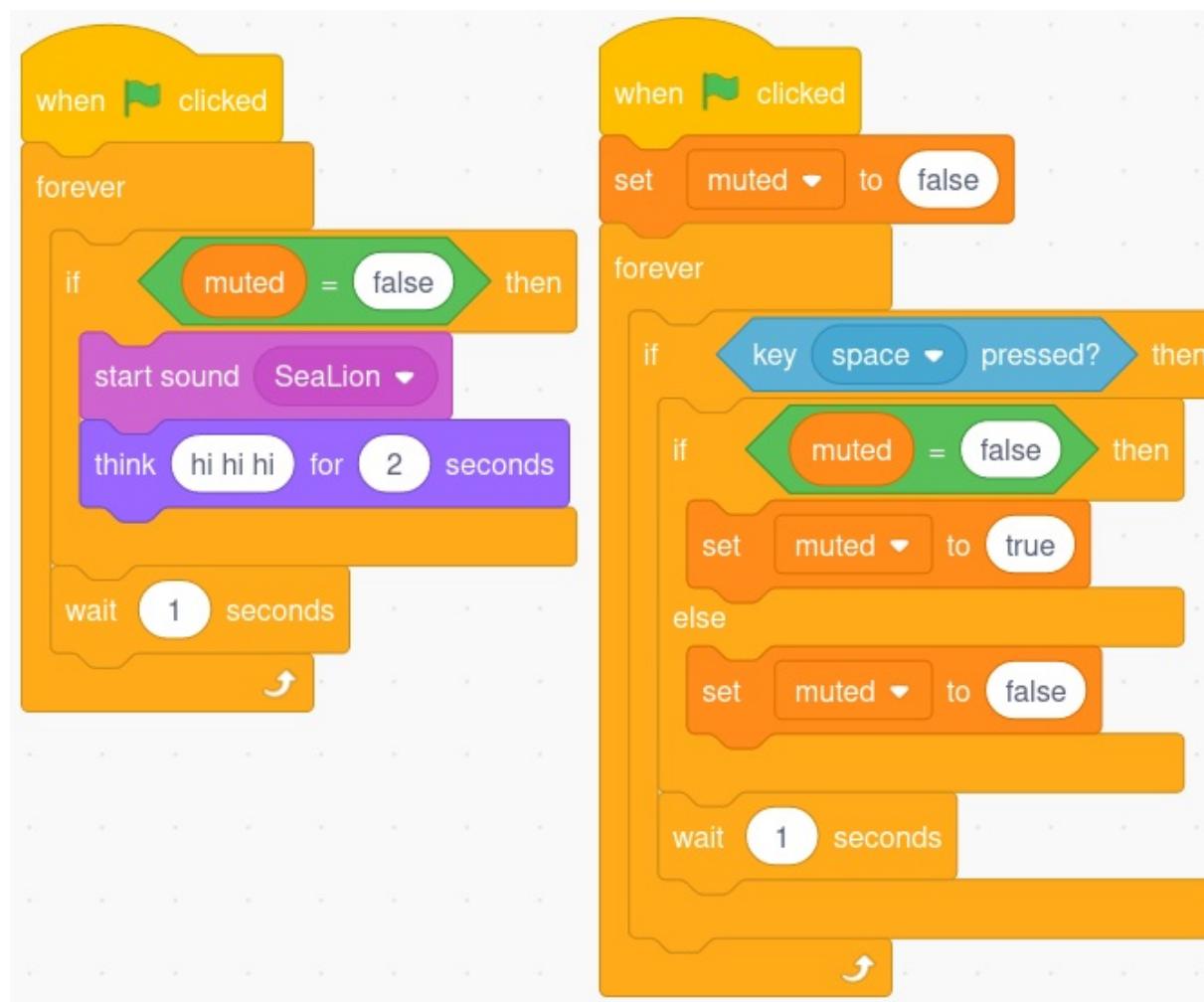


<https://scratch.mit.edu/projects/984136673/>

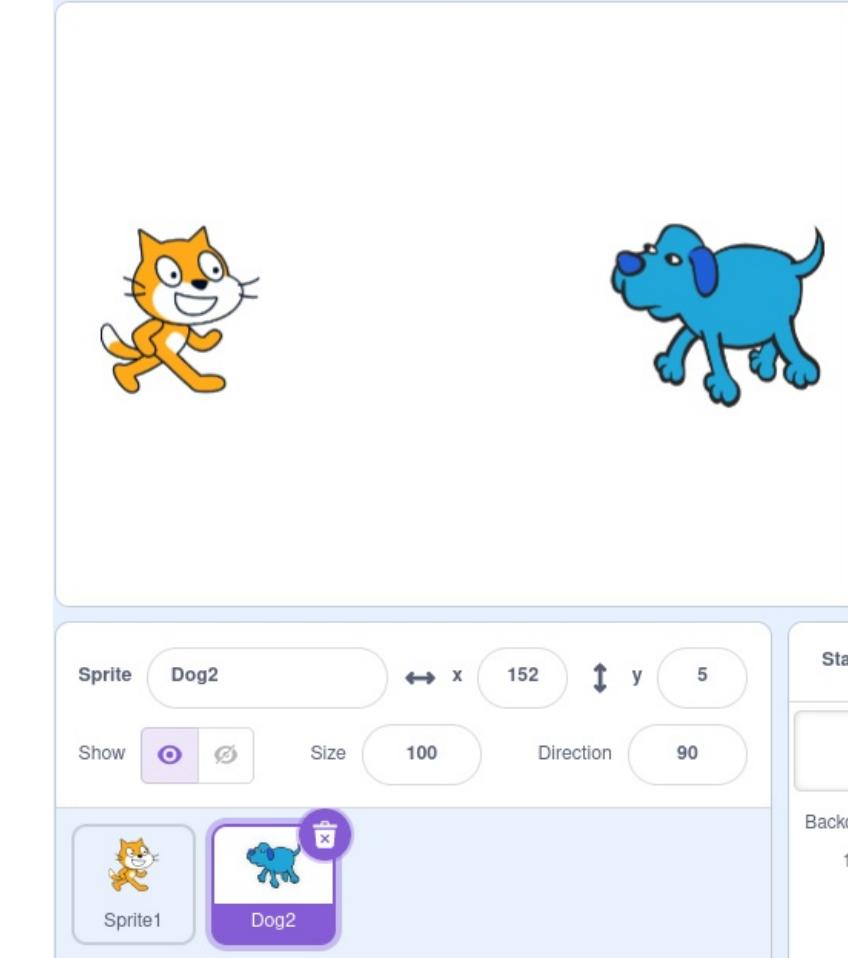
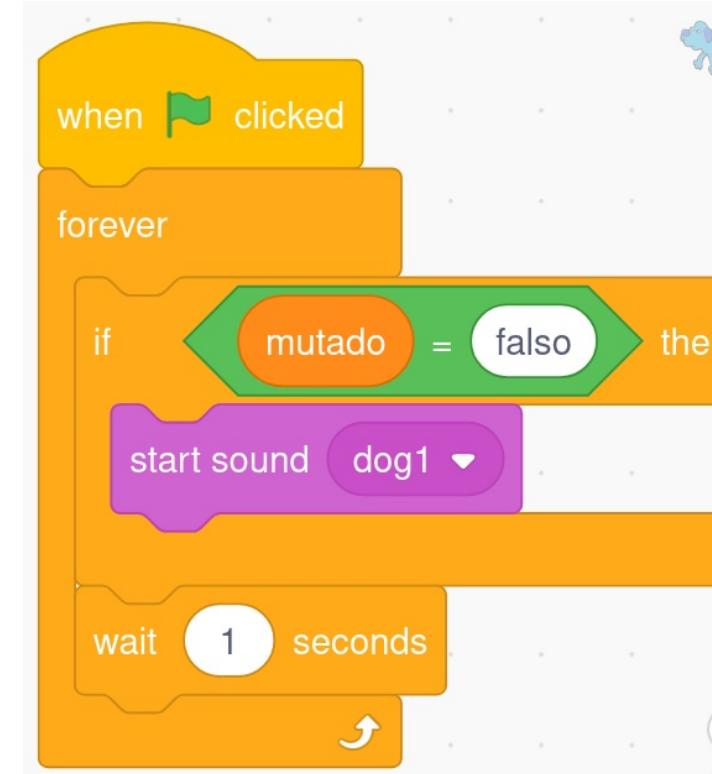
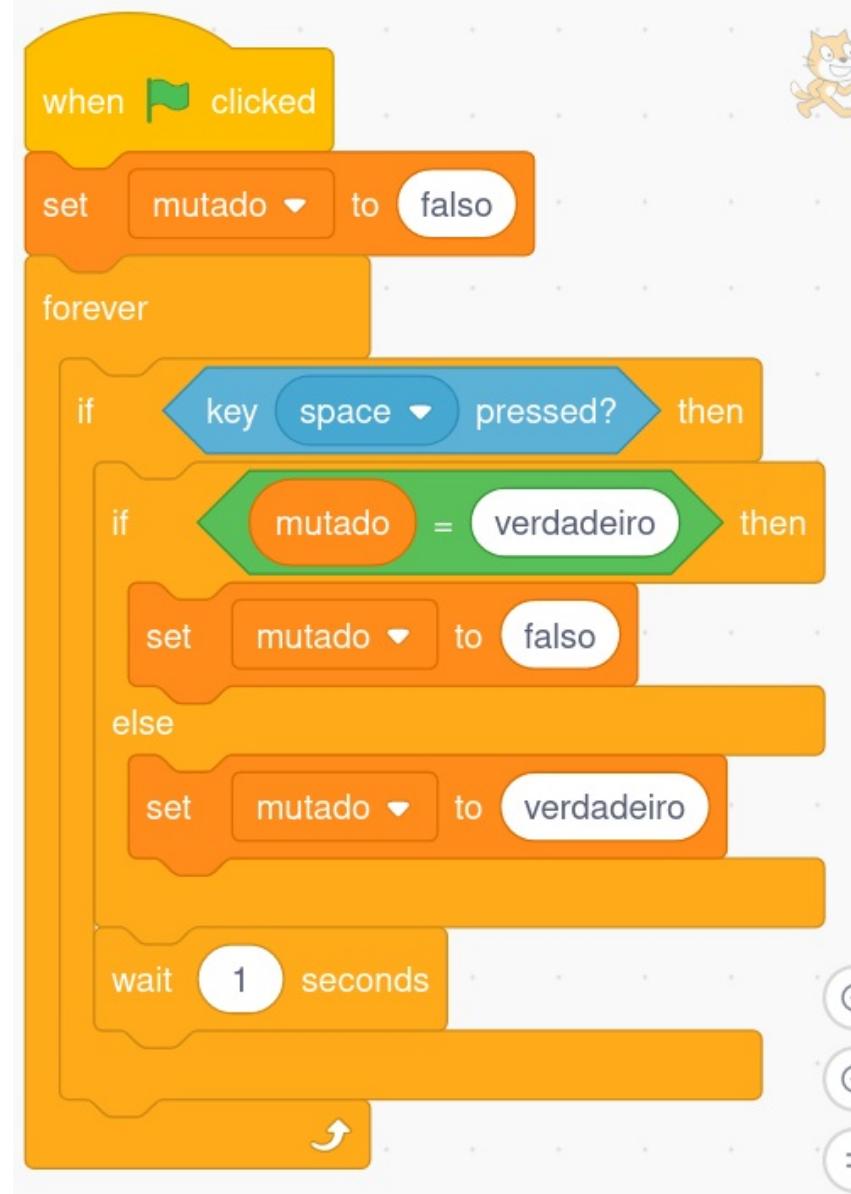
# Programação Concorrente

**Mais de um programa pode rodar ao mesmo tempo e em diversos sprites!** Podem escutar e responder a eventos de forma independente. No Scratch e no Snap! é muito fácil. Em outras linguagens é difícil... Exemplos:

<https://scratch.mit.edu/projects/807110618>

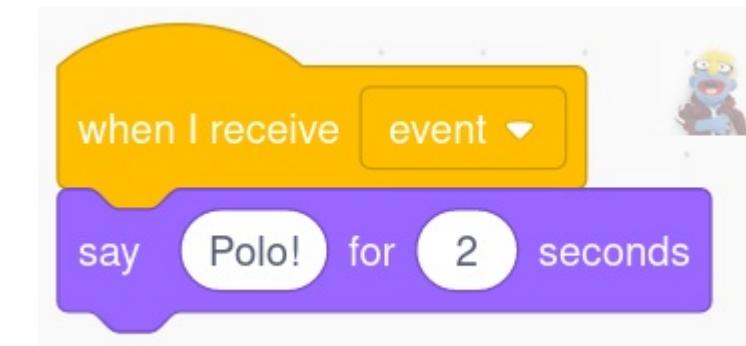
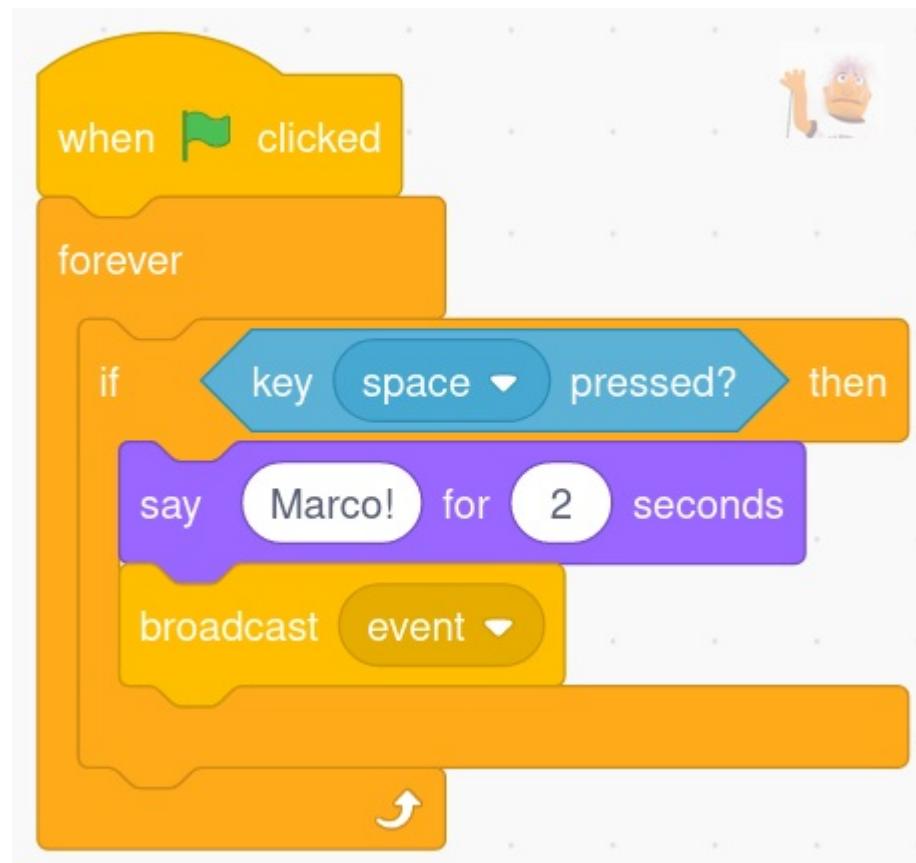


# Programação Concorrente



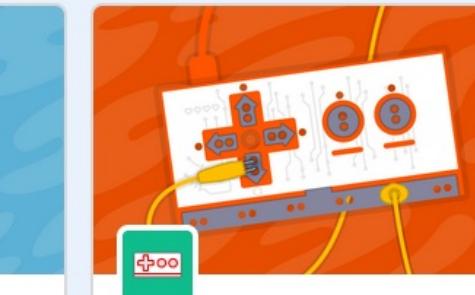
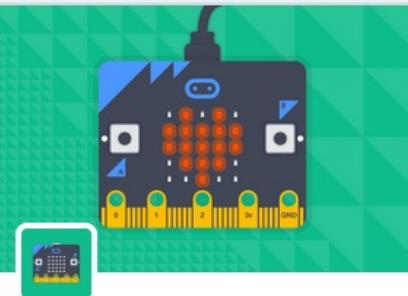
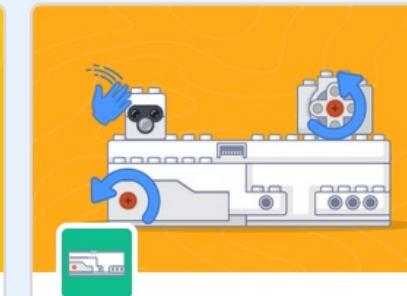
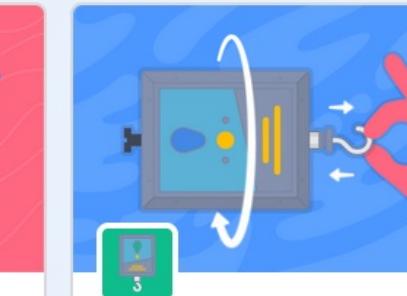
<https://scratch.mit.edu/projects/984139520/>

# Eventos, ouvintes, mensagens



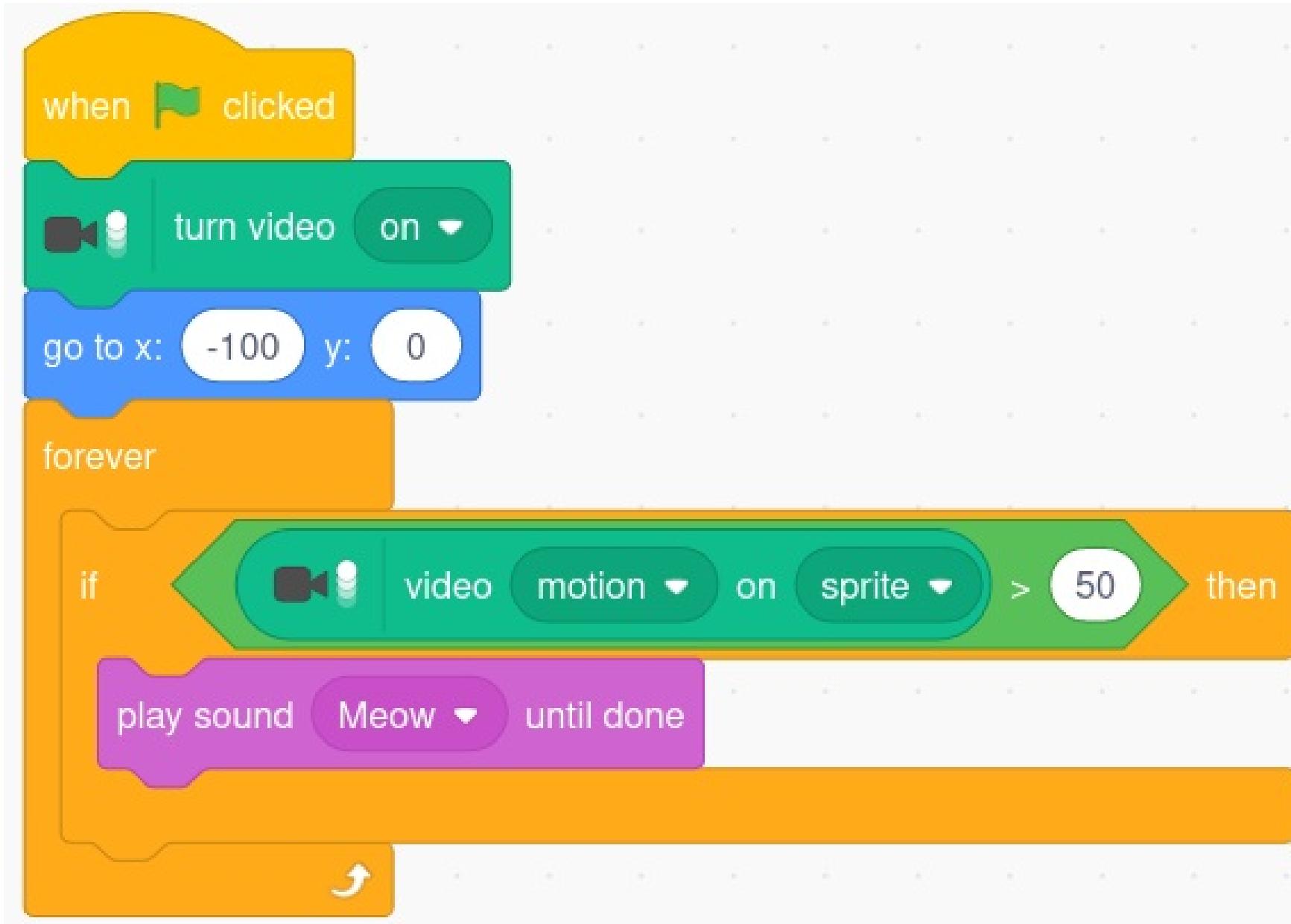
<https://scratch.mit.edu/projects/807110924/>

# Blocos avançados

 <b>Music</b> Play instruments and drums.	 <b>Pen</b> Draw with your sprites.	 <b>Video Sensing</b> Sense motion with the camera.	 <b>Text to Speech</b> Make your projects talk. Requires  Collaboration with Amazon Web Services	 <b>Translate</b> Translate text into many languages. Requires  Collaboration with Google	 <b>Makey Makey</b> Make anything into a key. Collaboration with JoyLabz
 <b>micro:bit</b> Connect your projects with the world. Requires   Collaboration with <b>micro:bit</b>	 <b>LEGO MINDSTORMS EV3</b> Build interactive robots and more. Requires   Collaboration with <b>LEGO</b>	 <b>LEGO BOOST</b> Bring robotic creations to life. Requires   Collaboration with <b>LEGO</b>	 <b>LEGO Education WeDo 2.0</b> Build with motors and sensors. Requires   Collaboration with <b>LEGO</b>	 <b>Go Direct Force &amp; Acceleration</b> Sense push, pull, motion, and spin. Requires   Collaboration with <b>Vernier</b>	

# Blocos avançados: câmera

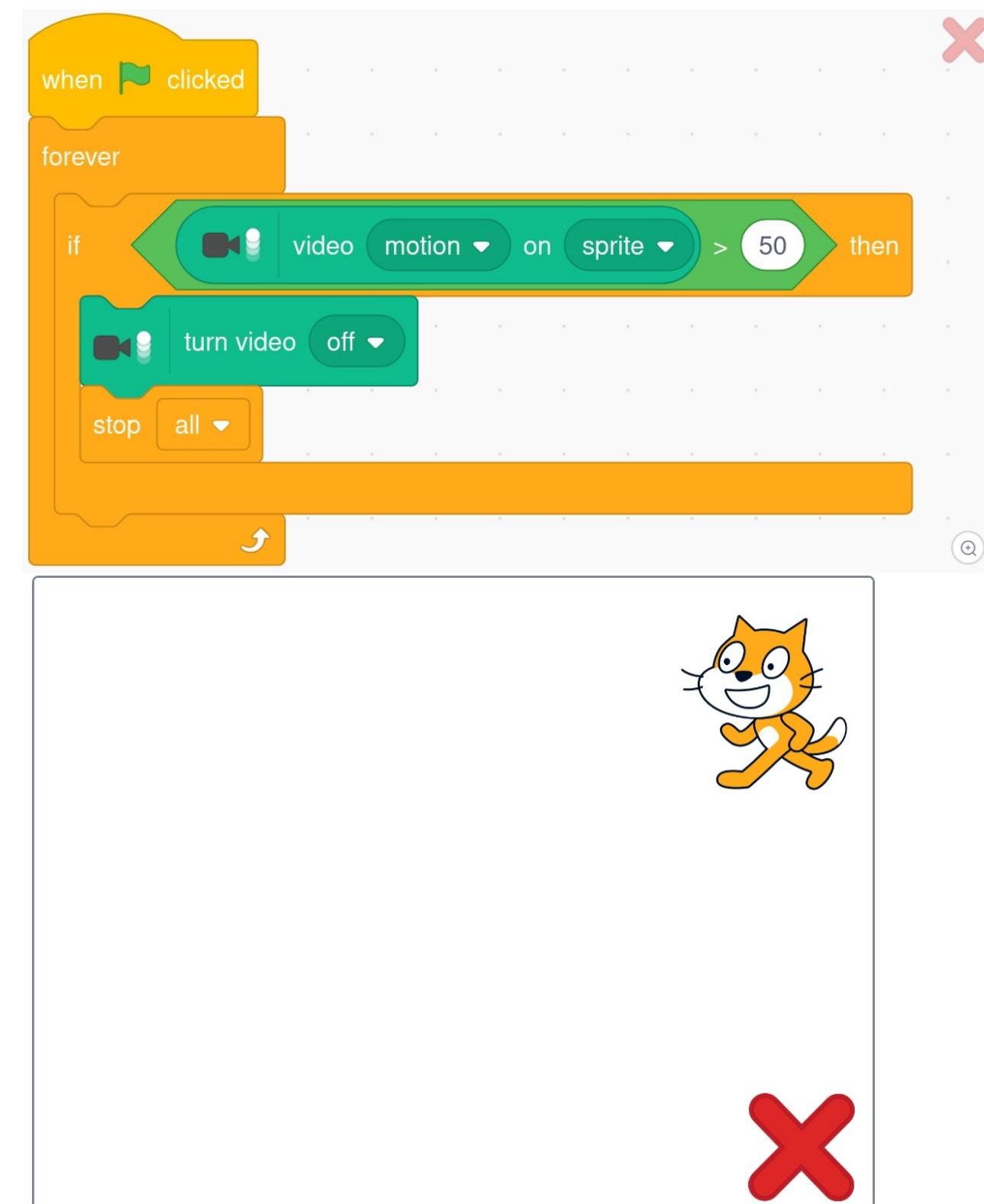
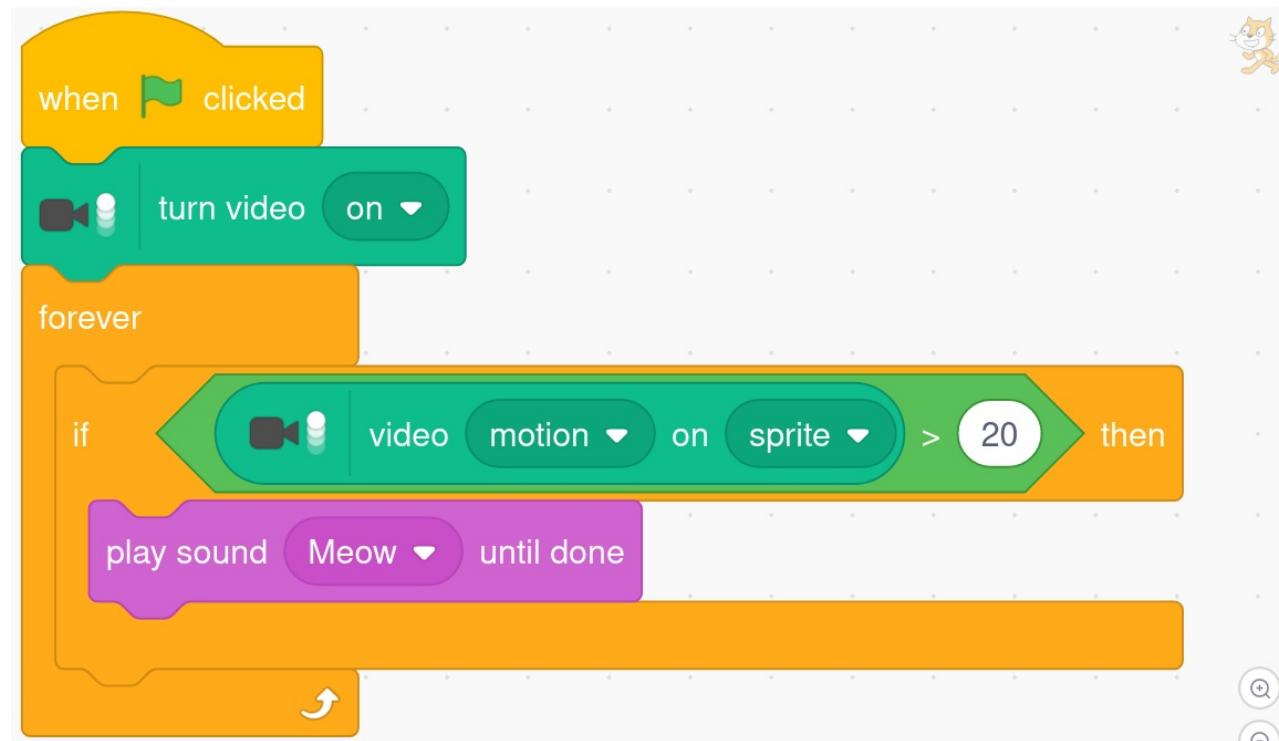
## Miar condicionalmente com o movimento da câmera



<https://scratch.mit.edu/projects/997064863/>

# Blocos avançados: câmera

## Ligar e desligar



<https://scratch.mit.edu/projects/984141301/>

# Blocos avançados: text-to-speech

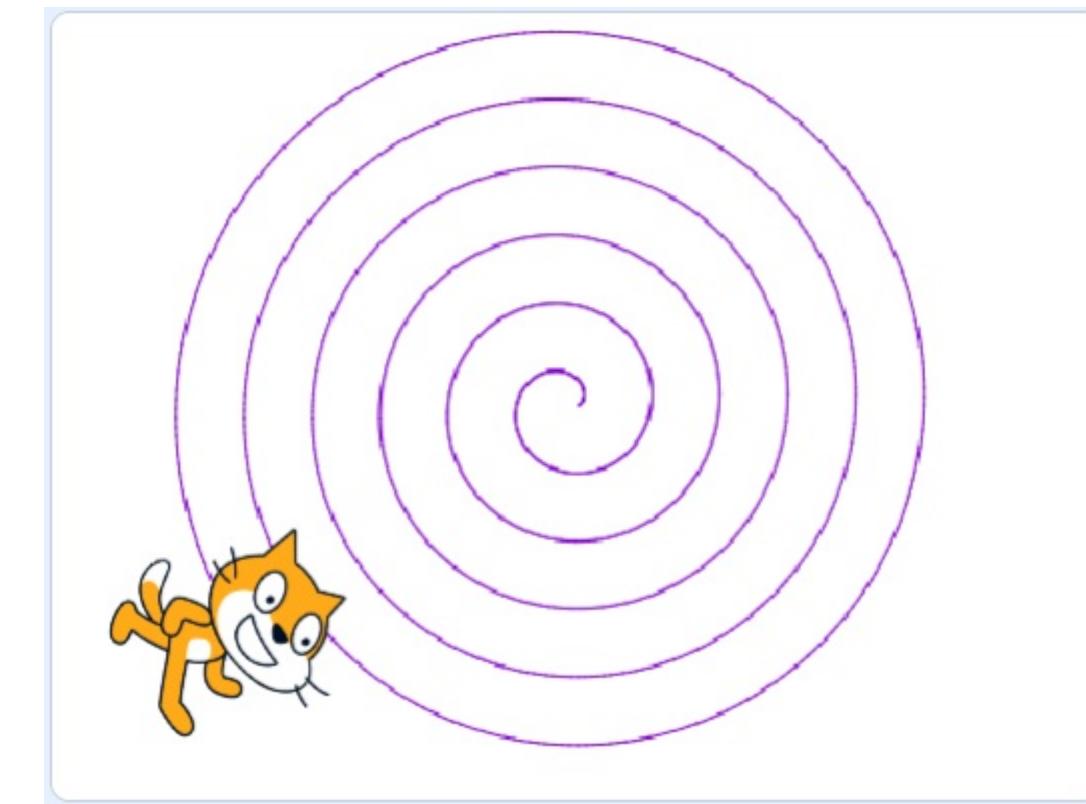
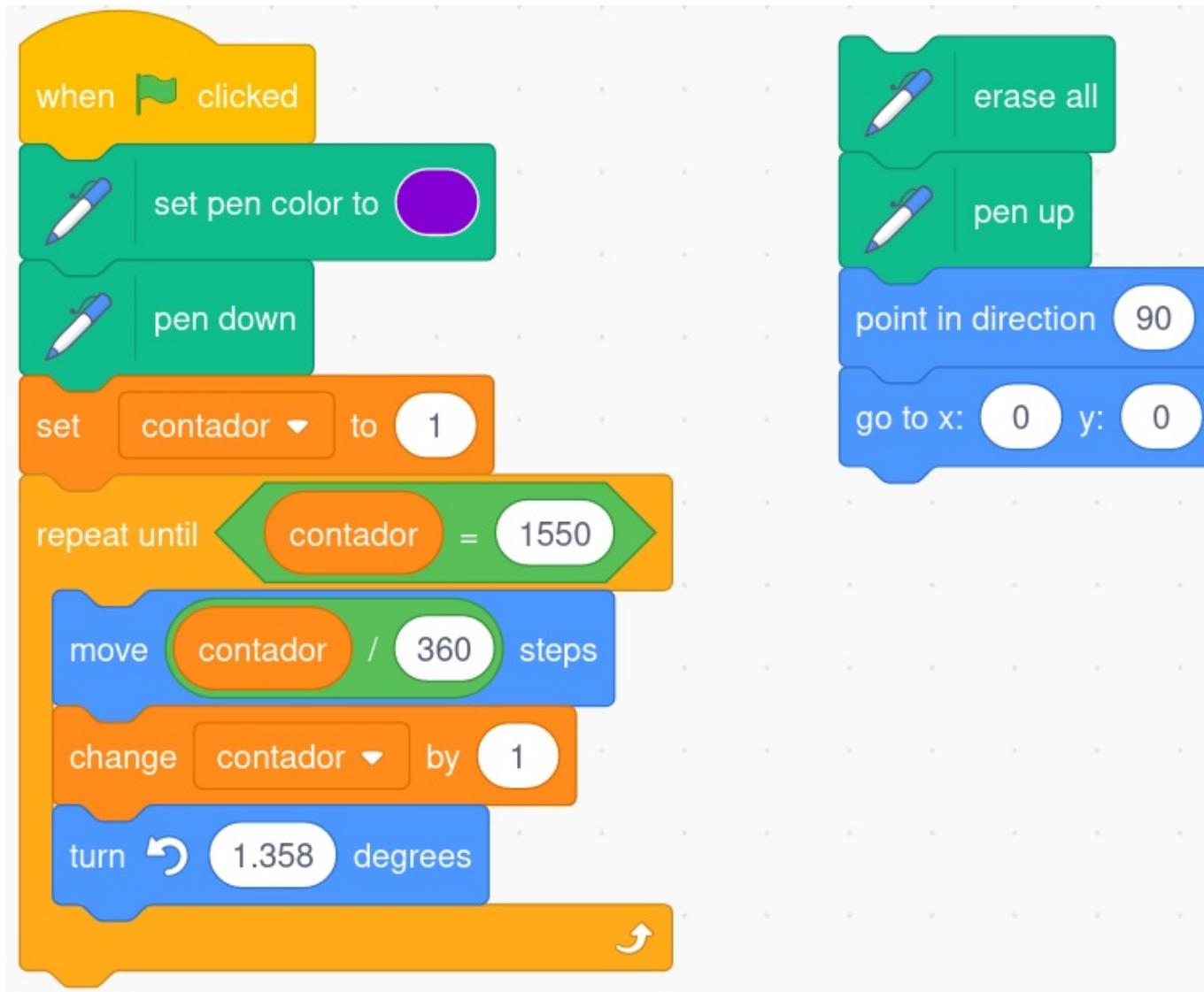
Seu programa pode falar!



<https://scratch.mit.edu/projects/980099415/>

# Blocos avançados: desenhos

Seu programa pode desenhar!



<https://scratch.mit.edu/projects/984283257/>

# Inspiração

Oscar Time: <https://scratch.mit.edu/projects/807112621/>



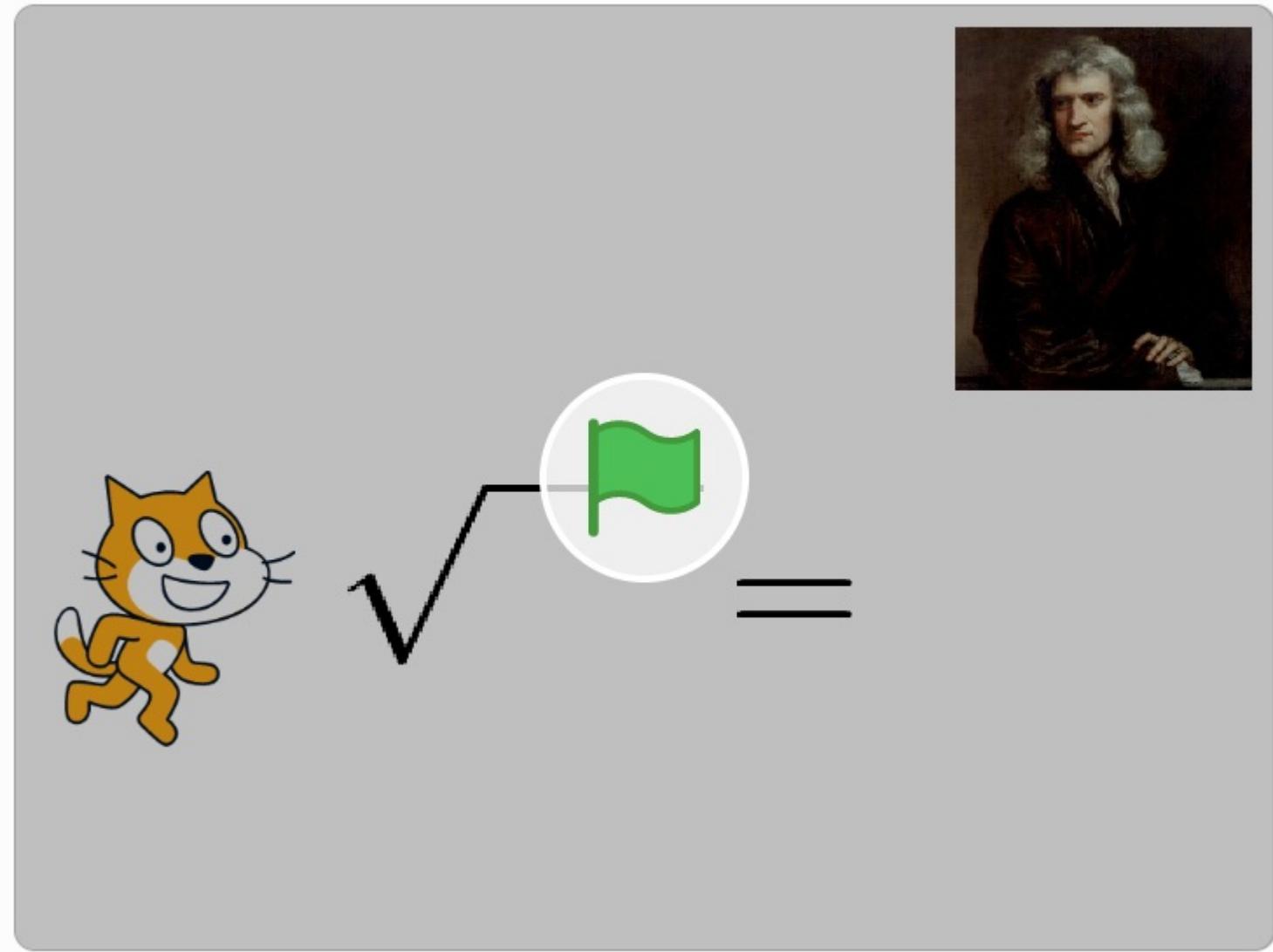
# Inspiração

Gingerbread Tales: <https://scratch.mit.edu/projects/807112827/>



# Inspiração

Raiz por Newton: <https://scratch.mit.edu/projects/809581863/>



# Inspiração

Defenda a FAESA: <https://scratch.mit.edu/projects/356031284/>



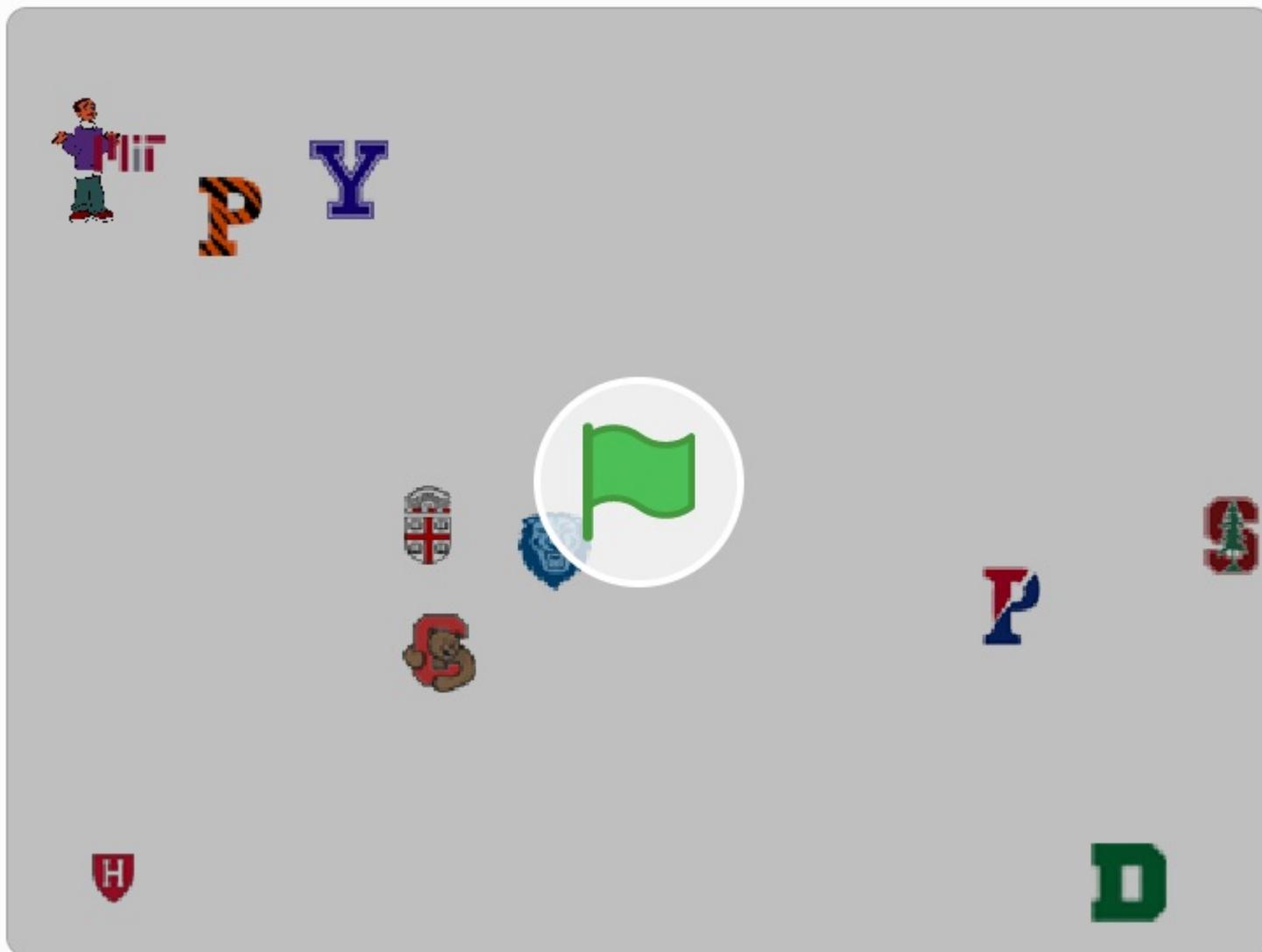
# Inspiração

Whack-a-mole: <https://scratch.mit.edu/projects/807165430/>



# Inspiração

Ivy's Hardest Game: <https://scratch.mit.edu/projects/807192580/>



# Inspiração

Studio do Computação Raiz: <https://scratch.mit.edu/studios/34743563>

The screenshot shows the Scratch website interface with a purple header bar. The header includes the Scratch logo, navigation links for Create, Explore, Ideas, About, a search bar, and buttons for Join Scratch and Sign in.

The main content area displays a studio titled "CR6.100B". On the left, there is a yellow box containing a logo with two black dots and a horizontal arrow pointing right, labeled "COMPUTAÇÃO RAIZ" below it. To the right of the logo is a blue box containing text about the studio:

Studio para a disciplina  
CR6.100B: Introdução à  
Ciência da Computação.  
  
Para maiores informações,  
visite:  
  
<https://www.computacaoraiz.com.br/cr6100b>

At the top of the main content area, there are four buttons: Projects (27), Comments (0), Curators, and Activity. Below these buttons, the word "Projects" is displayed in bold. The page then lists six Scratch projects arranged in a grid:

- broadcast\_in\_drawing** by abrantesasf: A project showing a cat sprite running through a complex geometric pattern of overlapping blue lines.
- squirrel** by abrantesasf: A project showing a squirrel sprite running through a series of concentric blue squares.
- formas** by abrantesasf: A project showing a cat sprite running across a plain white background.
- a\_quick\_play** by abrantesasf: A project showing a cat sprite and a yellow bird-like sprite running side-by-side.
- overflow\_cansel** by abrantesasf: A project showing a squirrel sprite running towards the viewer, with its head and body appearing to overflow from the bottom of the screen.
- bounce\_na\_parede** by abrantesasf: A project showing a cat sprite running towards the right side of the screen, with its body appearing to bounce off the right edge.

# Como criar seu primeiro programa?

## PENSAMENTO COMPUTACIONAL!

- Decomposição
- Reconhecimento de padrões
- Abstração
- Estruturas de dados
- Algoritmos

### Lembre-se:

- Não tente fazer tudo de uma vez, **quebre o programa em pequenos pedaços**
- Ache a solução para esses pequenos pedaços
- Programe a solução para esses pequenos pedaços
- Continue até terminar



Adaptado de Charlie Harris, no Unsplash  
(<https://unsplash.com/photos/a-person-typing-on-a-keyboard-lXGoRVIV7u4>)

# Em resumo

