

FUNDAMENTOS DA PROGRAMAÇÃO



complexidade
de algoritmos

compilação

arrays

SQL

estruturas
de dados

ponteiros

algoritmos

internet

fundamentos da programação

web

linux

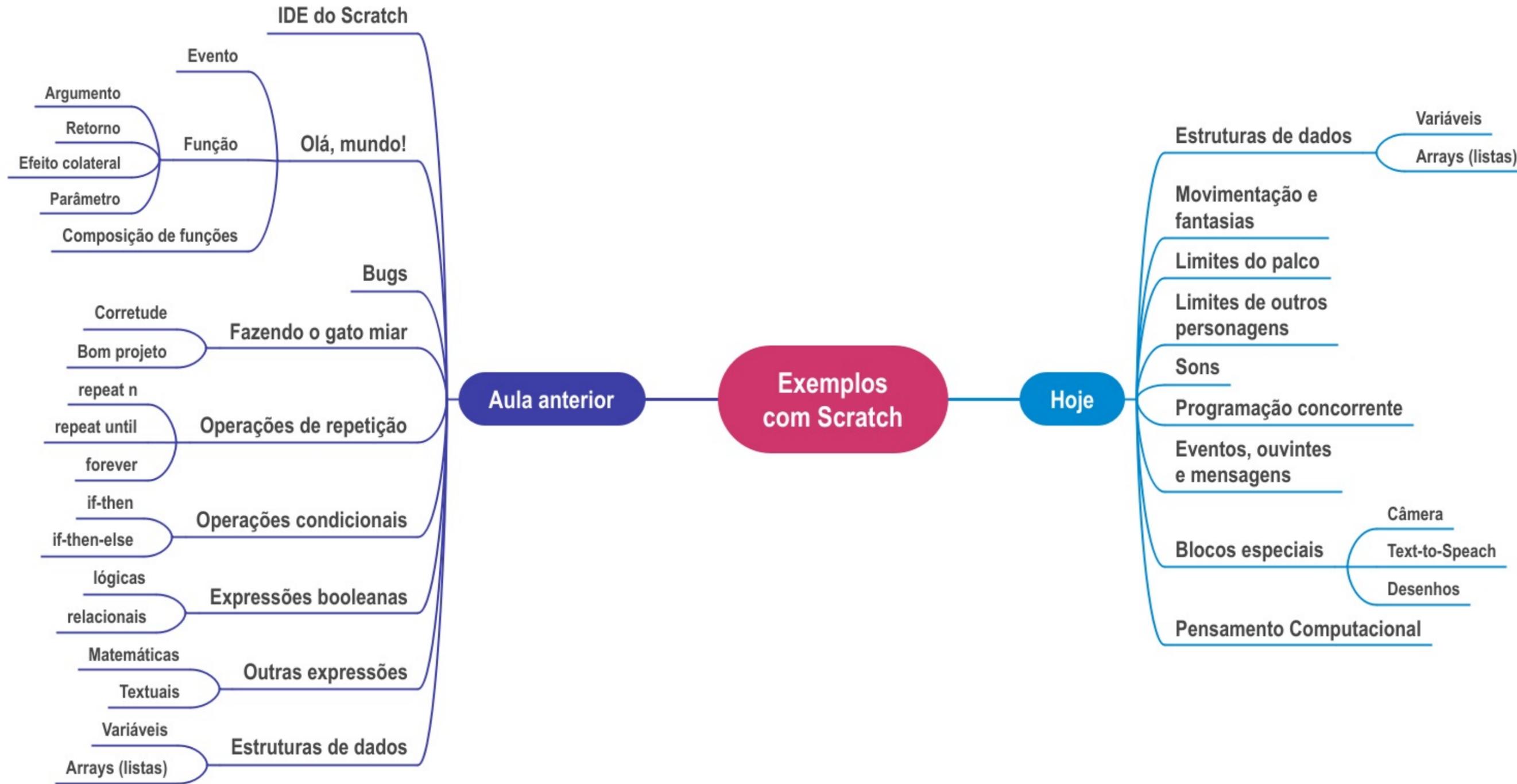
pensamento
computacional

memória

C

fundamentos da computação

Outros exemplos



Estruturas de dados: variáveis e arrays

Scratch fornece duas estruturas de dados:

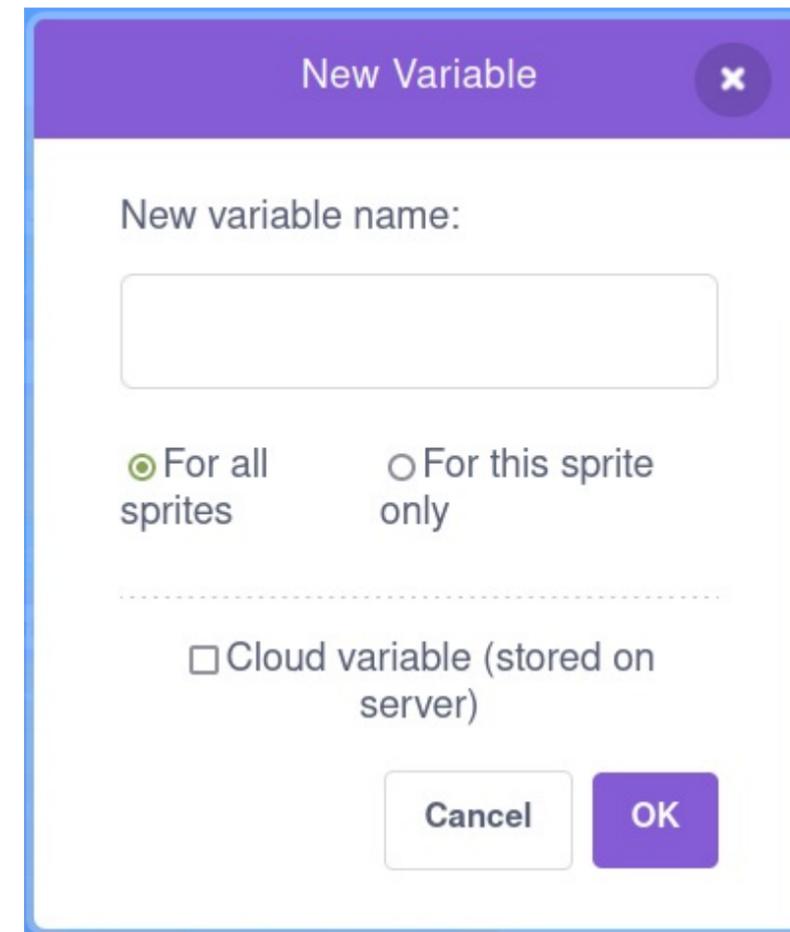
- **Variável:** É um **espaço na memória** para o **armazenamento de um único valor**. Pode corresponder a uma ou mais células de memória. Algumas linguagens obrigam o usuário a escolher o **tipo de dado** que será armazenado (inteiro, real, caractere, etc). No Scratch não é necessário escolher o tipo 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

The image shows a Scratch script for checking if a number is prime. It starts with a 'when clicked' block, followed by an 'ask' block with the text 'Informe um número inteiro positivo:'. The script then branches into two main paths. The first path checks if the answer is 1; if so, it says 'Não é primo.' for 2 seconds and stops. The second path checks if the answer is 2; if so, it says 'É primo.' for 2 seconds and stops. If the answer is neither 1 nor 2, it sets a variable named 'contador' to 2 and enters a 'repeat until' loop. The loop condition is 'contador > answer - 1'. Inside the loop, it checks if 'answer mod contador = 0'. If true, it says 'Não é primo.' for 2 seconds and stops. If false, it changes 'contador' by 1 and loops back. After the loop, it says 'É primo.' for 2 seconds.

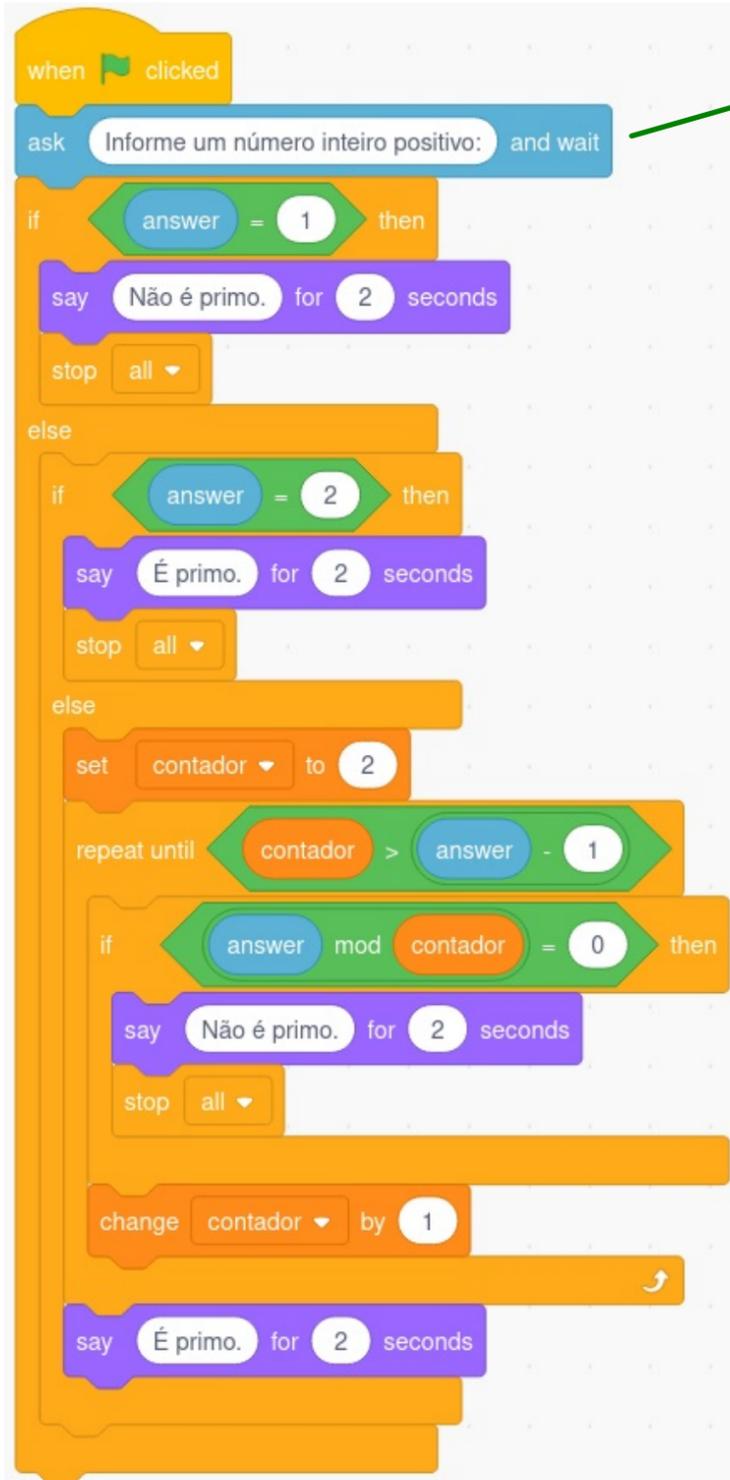


Ao criar variáveis no Scratch, você deve escolher se elas serão visíveis em todos os sprites (**global**) ou somente no sprite atual (**local**).

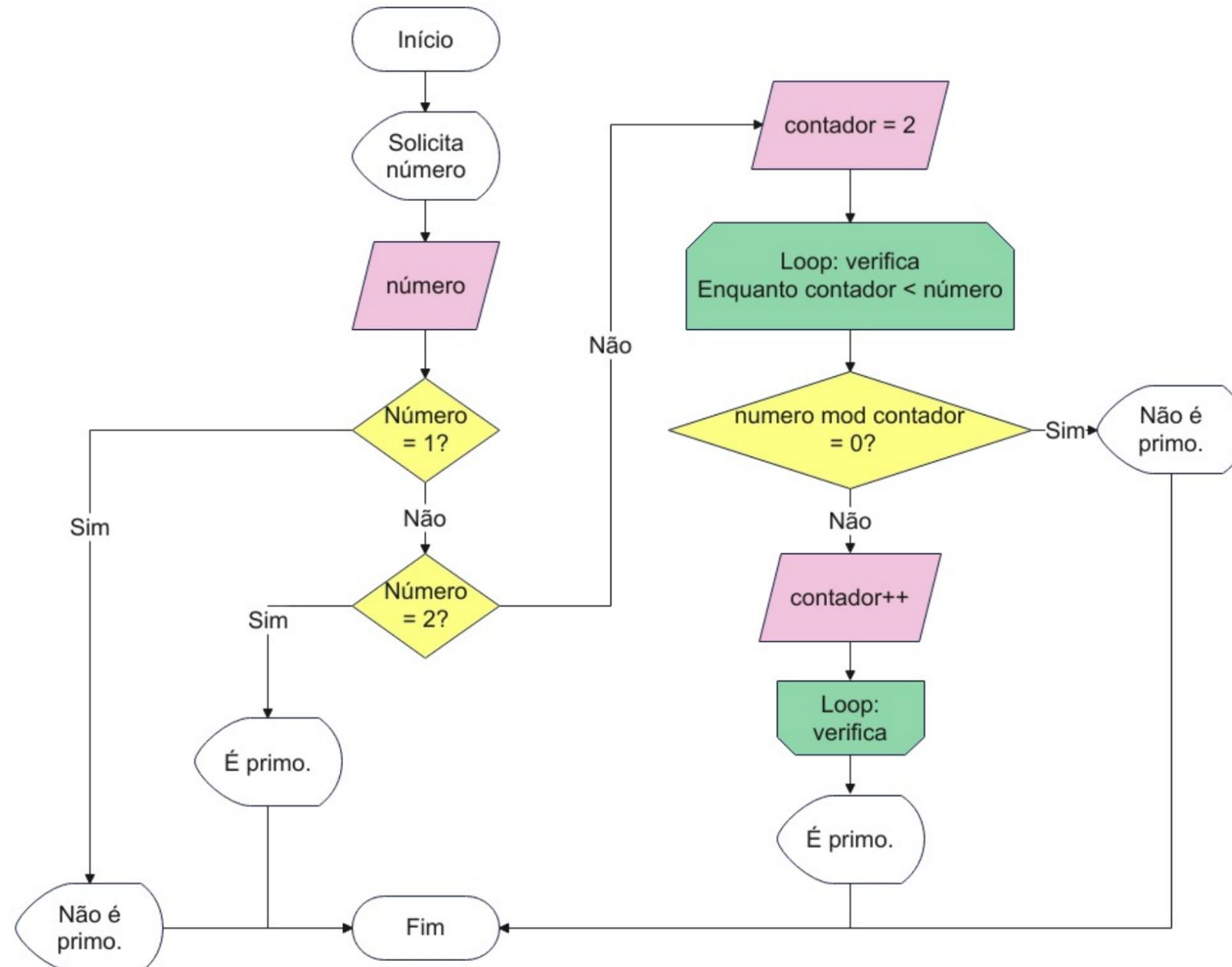


<https://scratch.mit.edu/projects/975855697/>

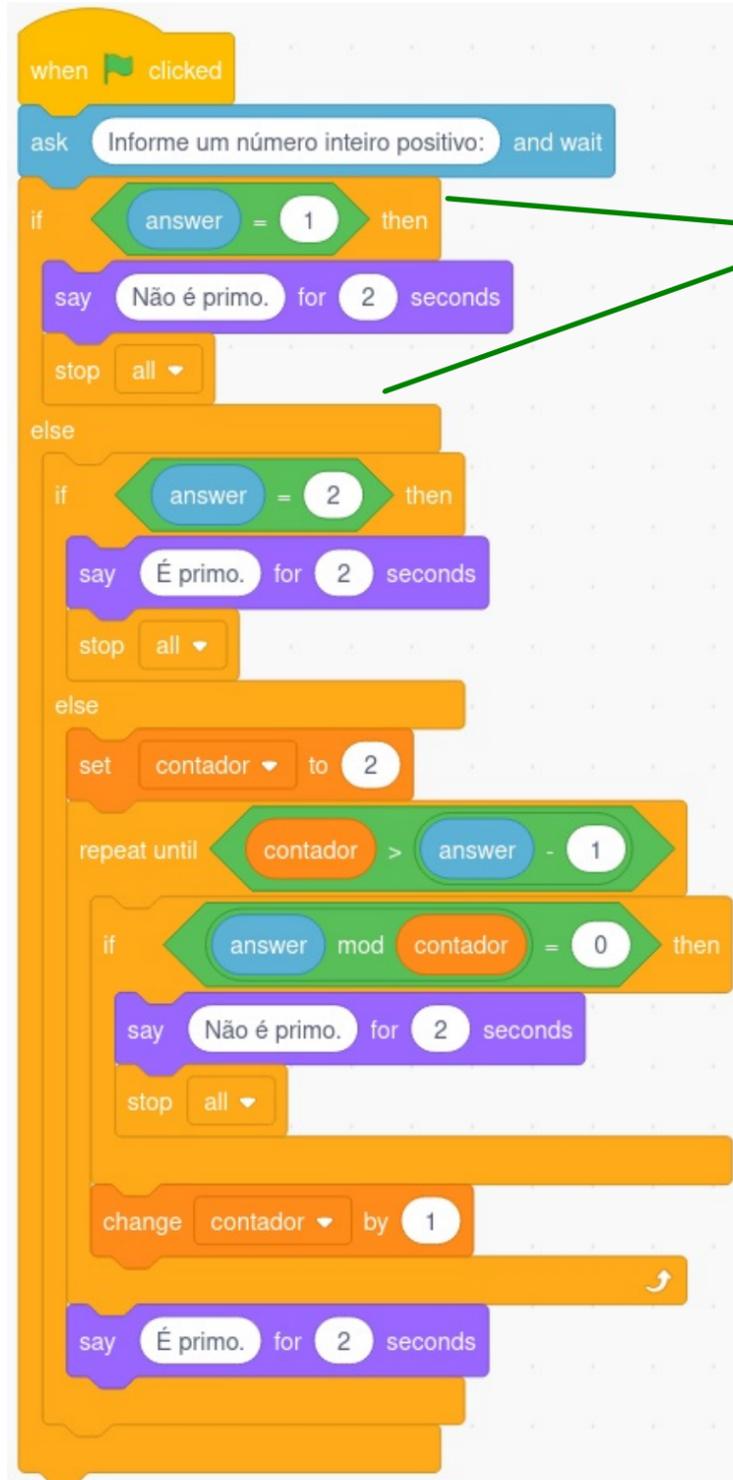
Variáveis: entendendo este código



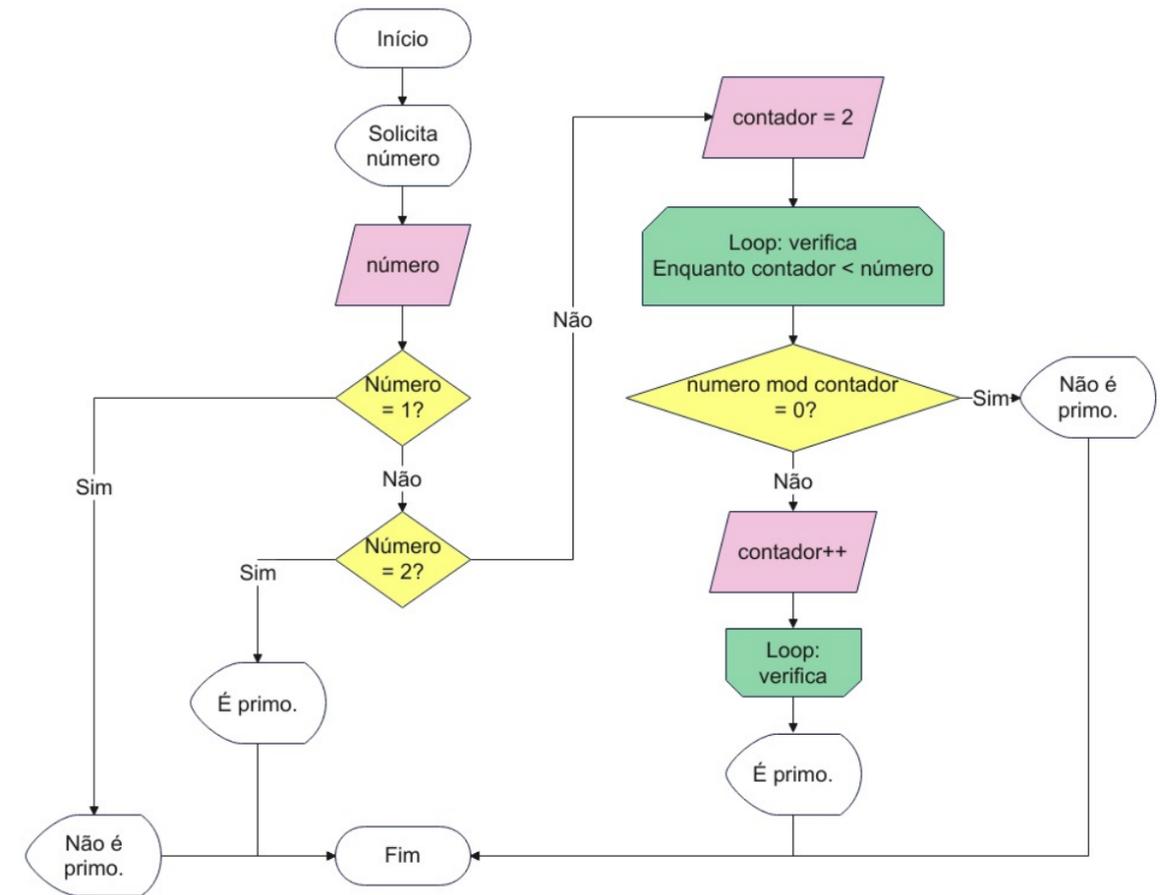
Solicita número.



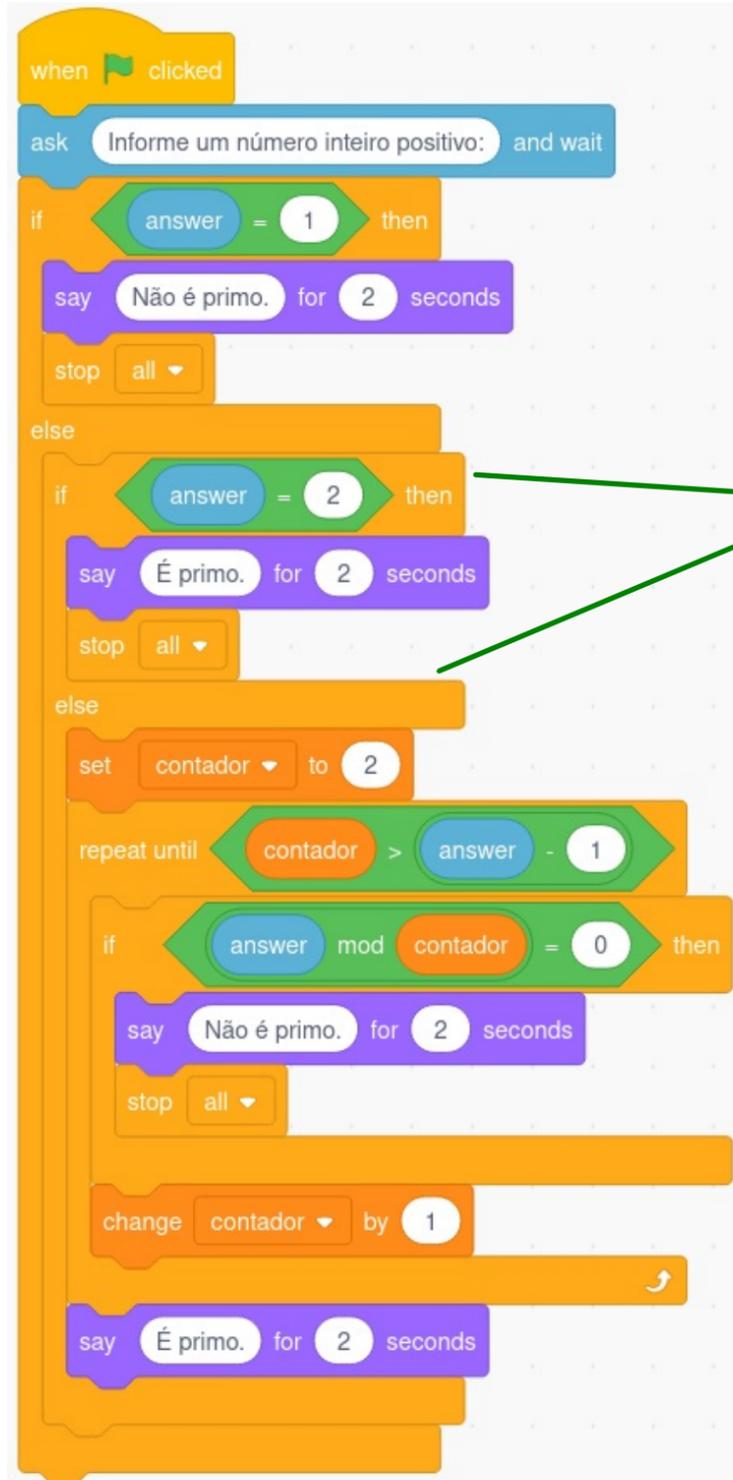
Variáveis: entendendo este código



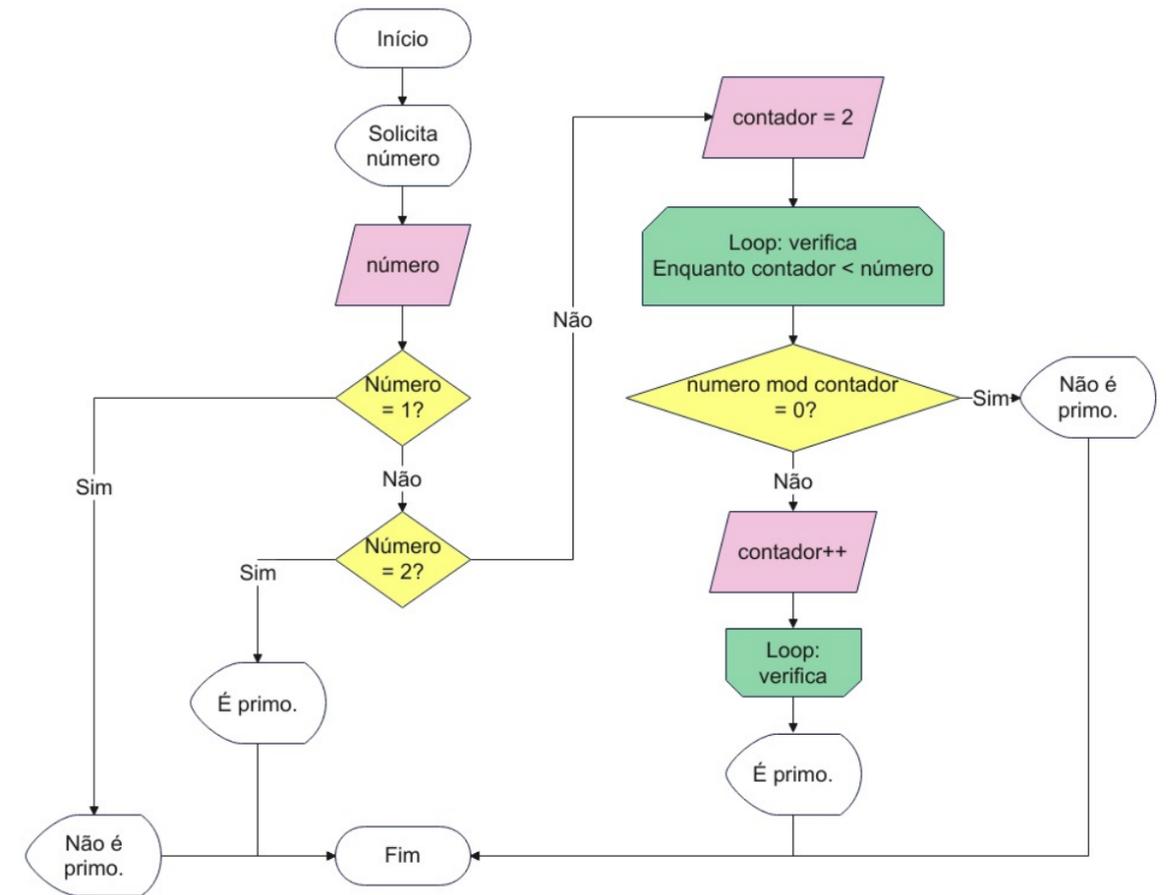
Verifica se o número é 1. Se for, diz que não é primo e termina o programa.



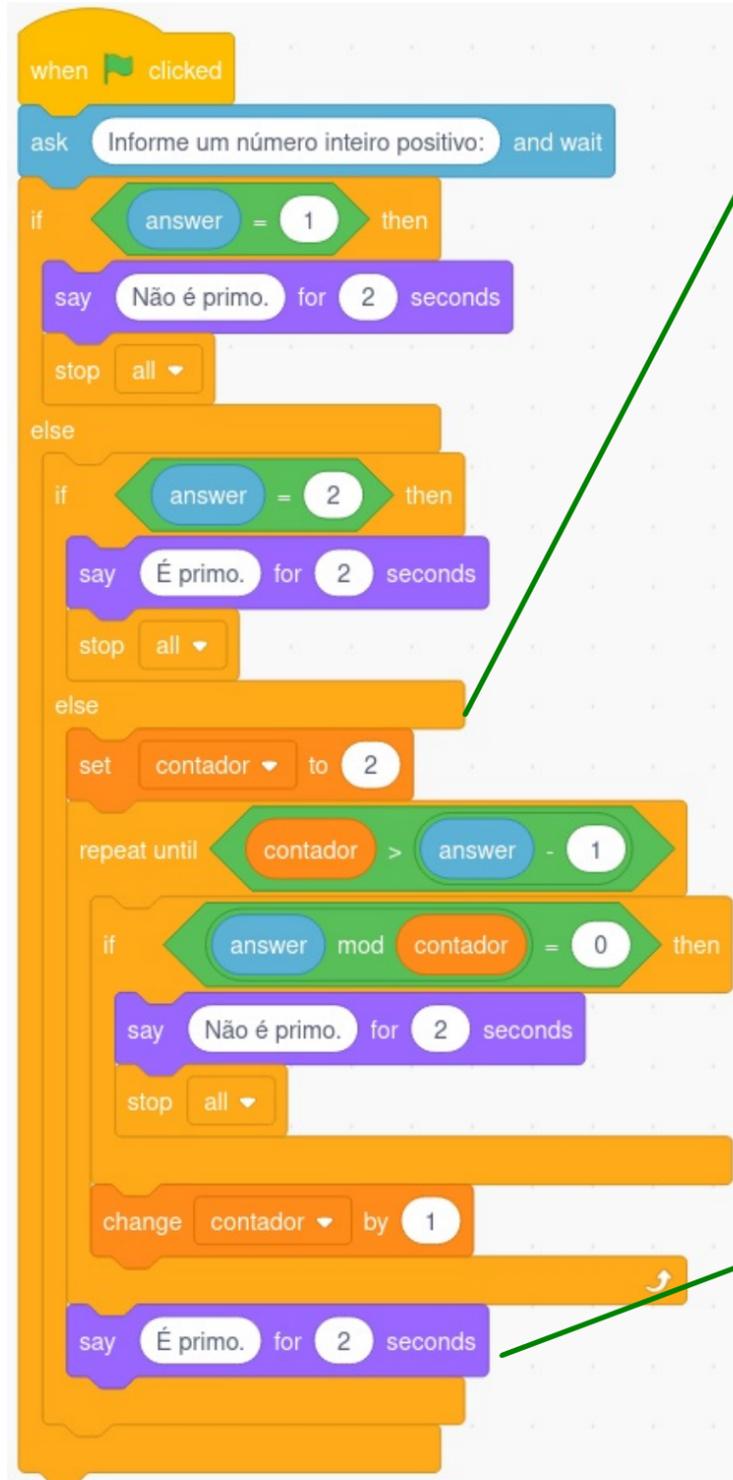
Variáveis: entendendo este código



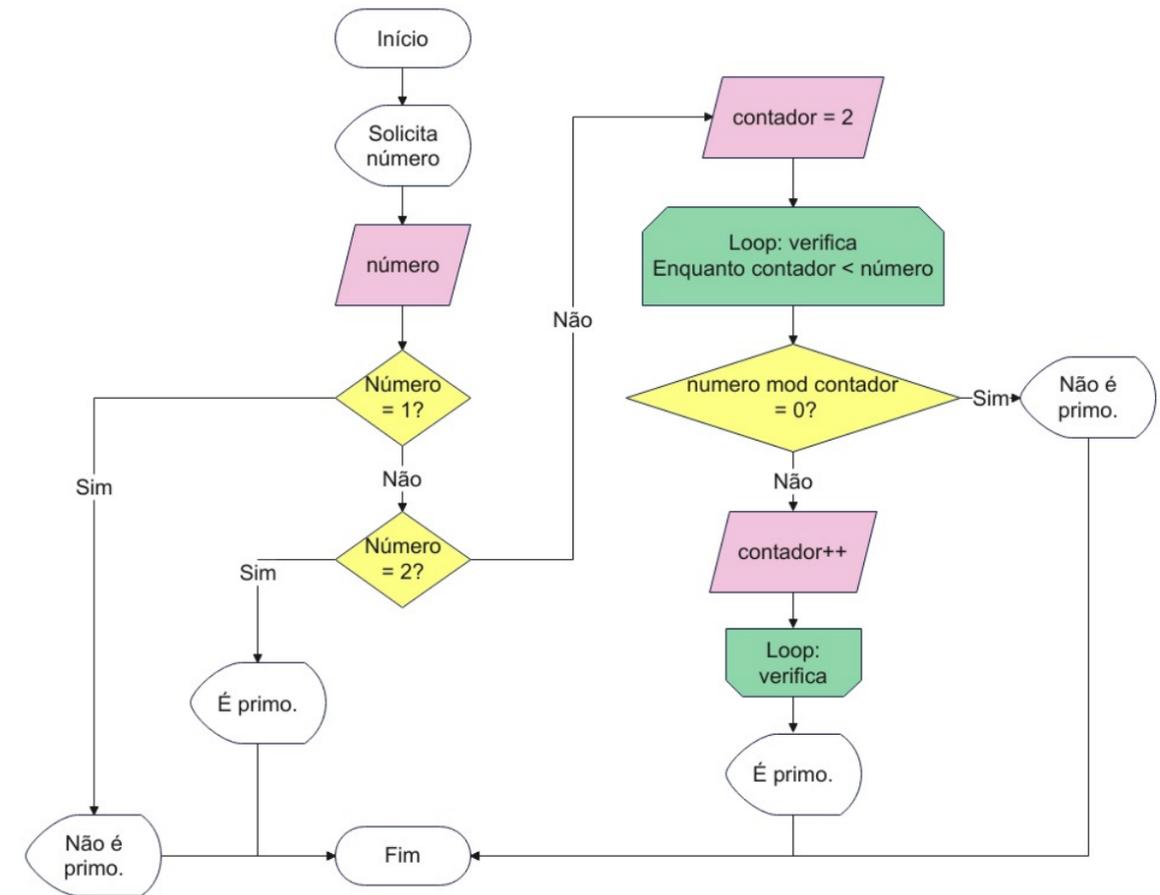
Verifica se o número é 2. Se for, diz que é primo e termina o programa.



Variáveis: entendendo este código



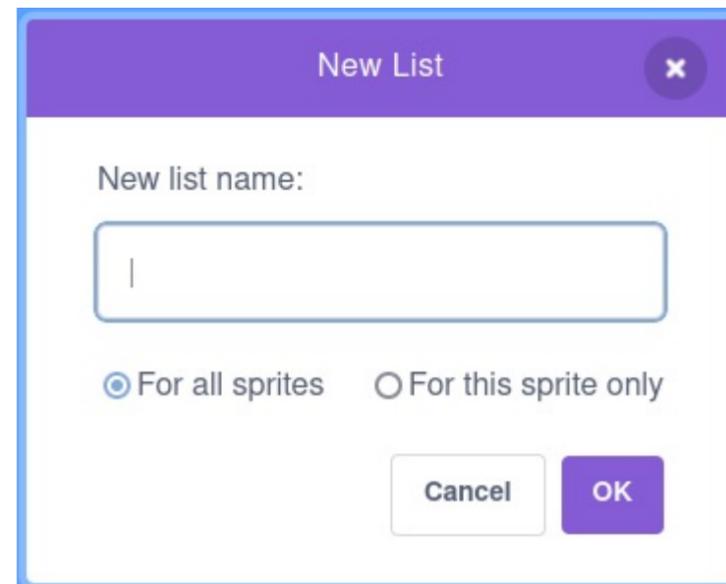
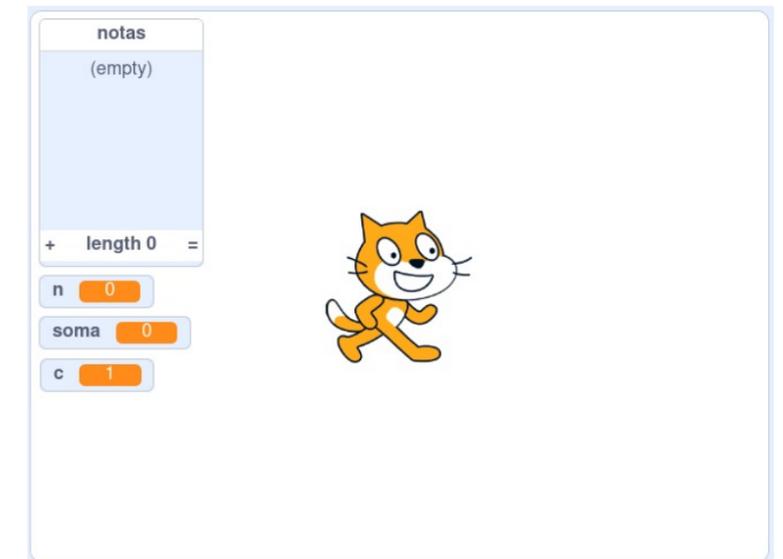
Se o número não for 1 ou 2, usamos a variável "contador" para verificar o resto da divisão do número informado pelo usuário por todos os números entre 2 e o antecessor do número informado. Se encontrarmos resto 0 então imprime que não é primo e termina. Caso contrário imprime que é primo e termina.



Arrays

```
when green flag clicked
  set n to 0
  set soma to 0
  set c to 1
  delete all of notas
  ask "Quantas notas serão informadas?" and wait
  set n to answer
  repeat n
    ask "Informe a nota:" and wait
    add answer to notas
  repeat until c > length of notas
    set soma to soma + item c of notas
    change c by 1
  say join "A média é:" round (soma / n) for 2 seconds
```

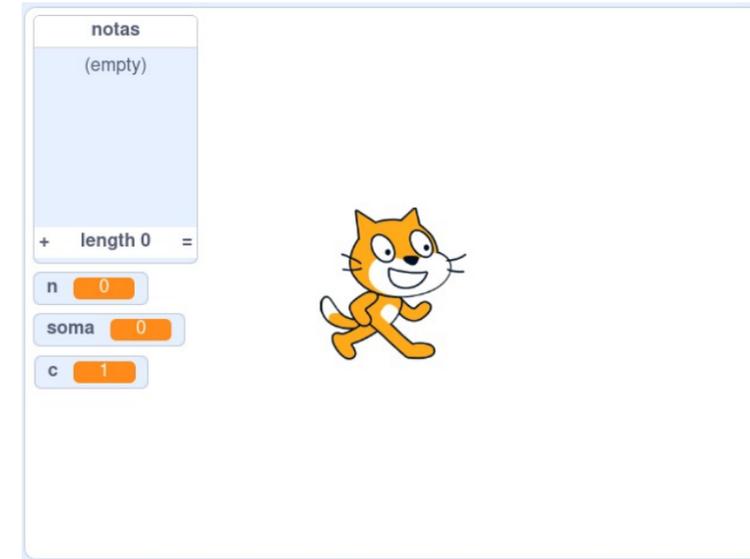
Ao criar arrays no Scratch, você deve escolher se eles serão visíveis em todos os sprites ou somente no sprite atual.



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

Arrays: entendendo este código

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.

```
when clicked
  set n to 0
  set soma to 0
  set c to 1
  delete all of notas
  ask "Quantas notas serão informadas?" and wait
  set n to answer
  repeat n
    ask "Informe a nota:" and wait
    add answer to notas
  repeat until c > length of notas
    set soma to soma + item c of notas
    change c by 1
  say join "A média é:" round (soma / n) for 2 seconds
```

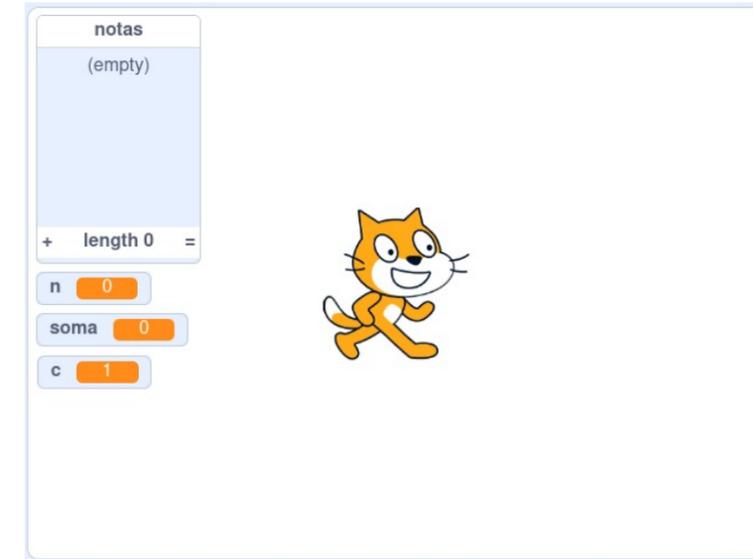
Arrays: entendendo este código

```
when clicked
  set n to 0
  set soma to 0
  set c to 1
  delete all of notas
  ask "Quantas notas serão informadas?" and wait
  set n to answer
  repeat n
    ask "Informe a nota:" and wait
    add answer to notas
  repeat until c > length of notas
    set soma to soma + item c of notas
    change c by 1
  say join "A média é:" round (soma / n) 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.

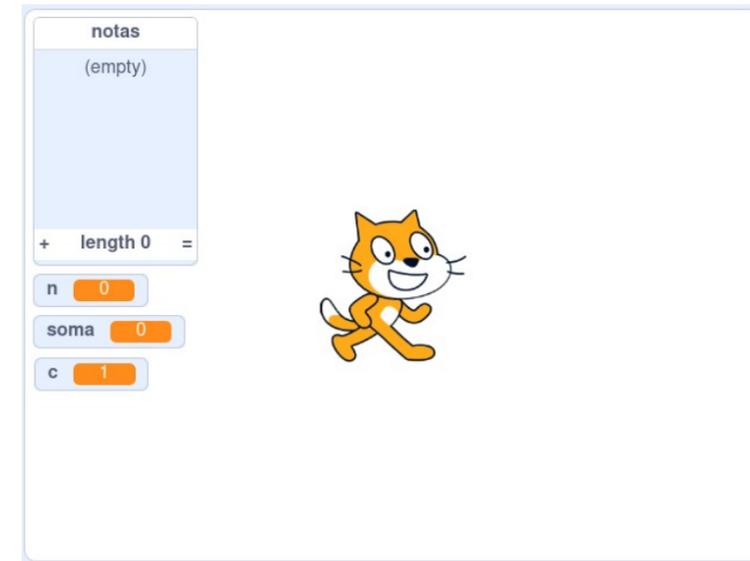
Note que após o usuário informar cada nota, estamos **incluindo** a nota informada ao array.

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.

The image shows a Scratch script with the following blocks:

- when green flag clicked
- set n to 0
- set soma to 0
- set c to 1
- delete all of notas
- ask "Quantas notas serão informadas?" and wait
- set n to answer
- repeat n times:
 - ask "Informe a nota:" and wait
 - add answer to notas
- repeat until c > length of notas:
 - set soma to soma + item c of notas
 - change c by 1
- say "A média é: round(soma / n) for 2 seconds



Arrays: entendendo este código

```
when clicked
  set n to 0
  set soma to 0
  set c to 1
  delete all of notas
  ask "Quantas notas serão informadas?" and wait
  set n to answer
  repeat n
    ask "Informe a nota:" and wait
    add answer to notas
  repeat until c > length of notas
    set soma to soma + item c of notas
    change c by 1
  say join "A média é:" round (soma / n) for 2 seconds
```

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!**

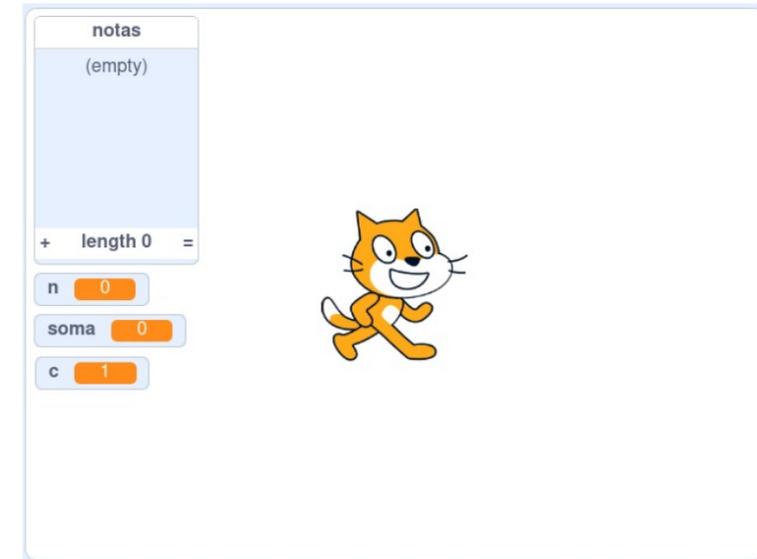
<https://programming-idioms.org>

Arrays: entendendo este código

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

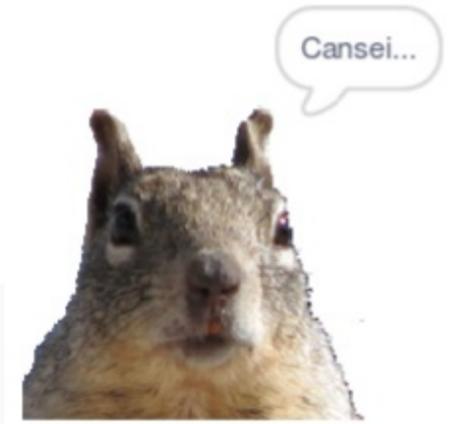
Obs.: note que a média está sendo informada com a função **round**, que arredonda para um número inteiro.

```
when clicked
  set n to 0
  set soma to 0
  set c to 1
  delete all of notas
  ask "Quantas notas serão informadas?" and wait
  set n to answer
  repeat n
    ask "Informe a nota:" and wait
    add answer to notas
  repeat until c > length of notas
    set soma to soma + item c of notas
    change c by 1
  say join "A média é:" round (soma / n) for 2 seconds
```



Ao usar variáveis, cuidado com situações especiais

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



```
when green flag clicked
  set contador to 2
  forever loop
    say contador
    set contador to contador * 456789
```

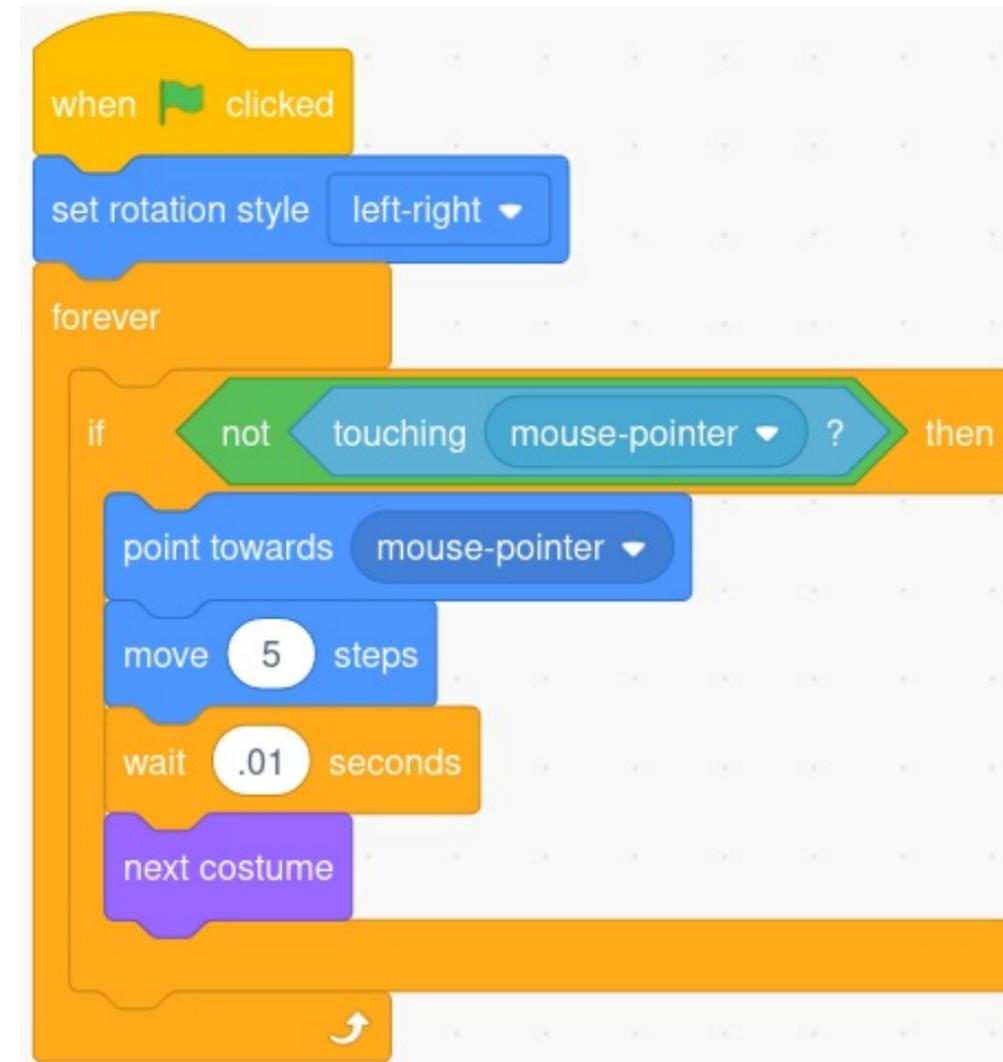
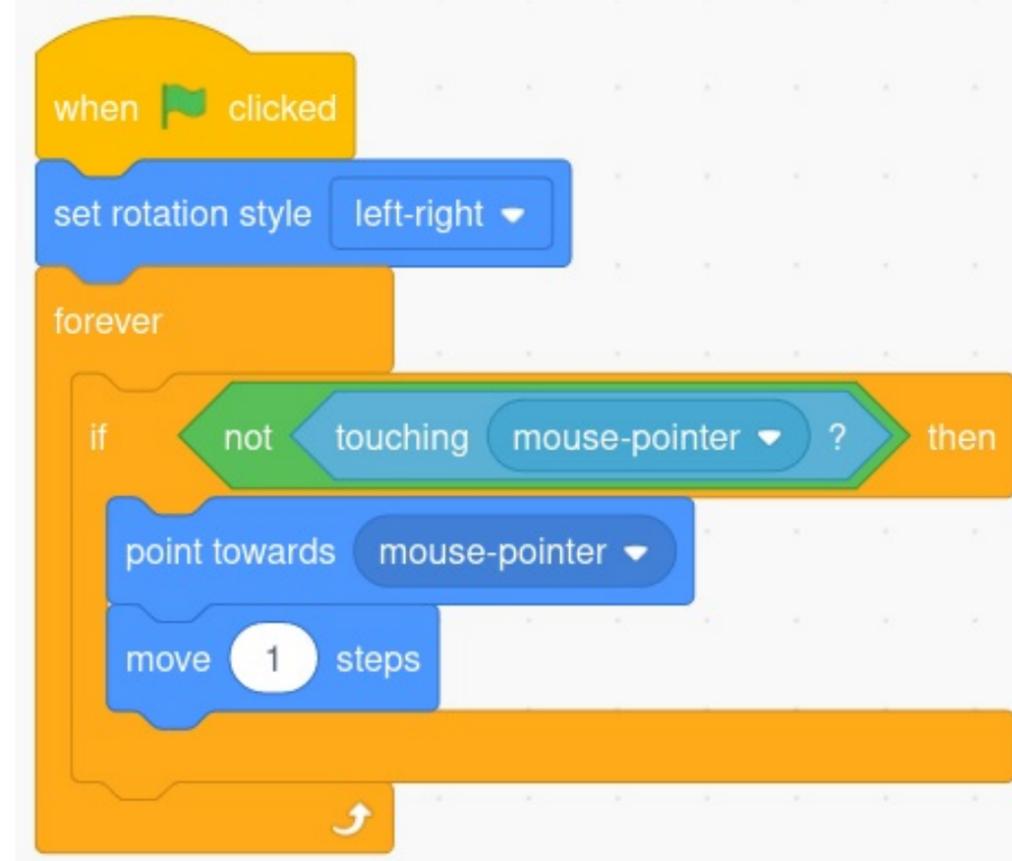
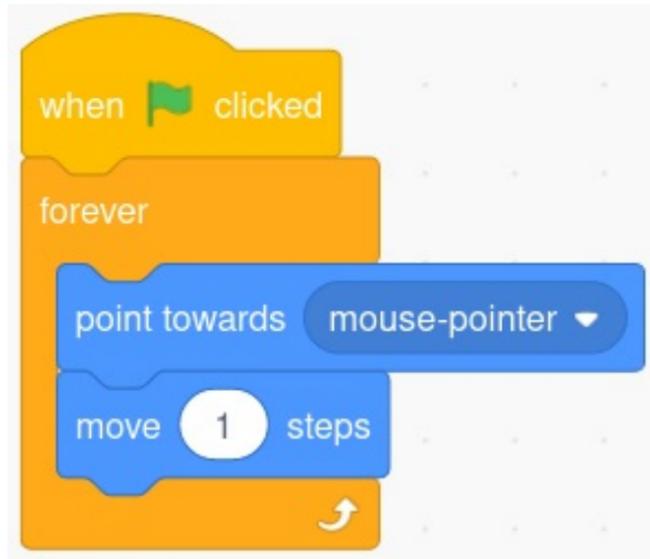
<https://scratch.mit.edu/projects/997034544/>

```
when green flag clicked
  set contador to 1
  forever loop
    say contador
    set contador to 16 * contador
    if contador = Infinity then
      say Cansei... for 2 seconds
    stop all
```

Você deve validar a entrada do usuário!

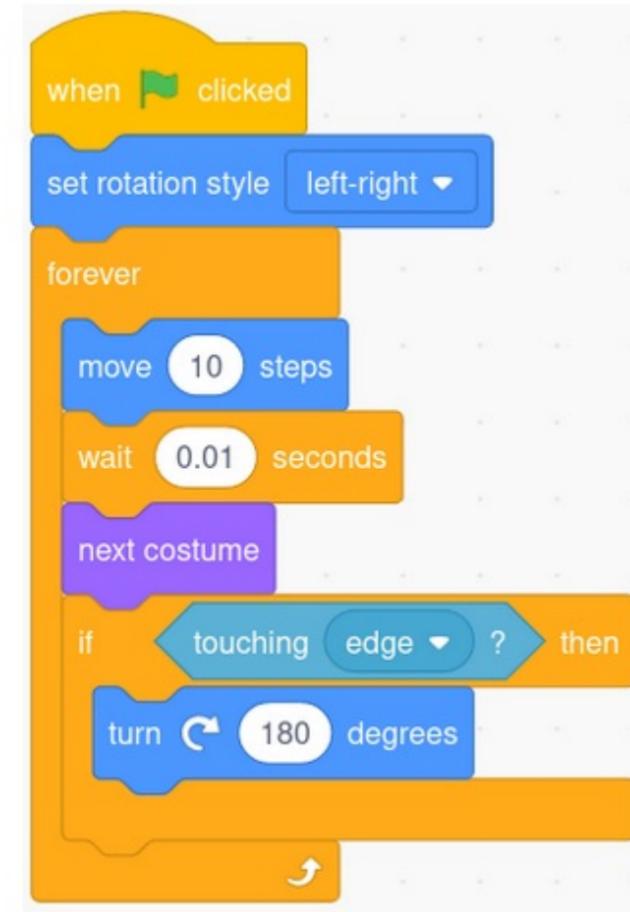
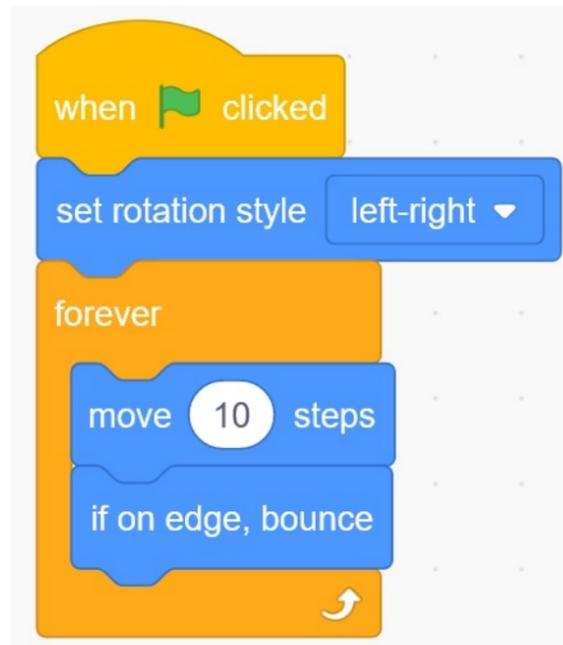
<https://scratch.mit.edu/projects/984134758/>

Movimentação, fantasias



<https://scratch.mit.edu/projects/807108713/>
<https://scratch.mit.edu/projects/807108956/>
<https://scratch.mit.edu/projects/997041209/>

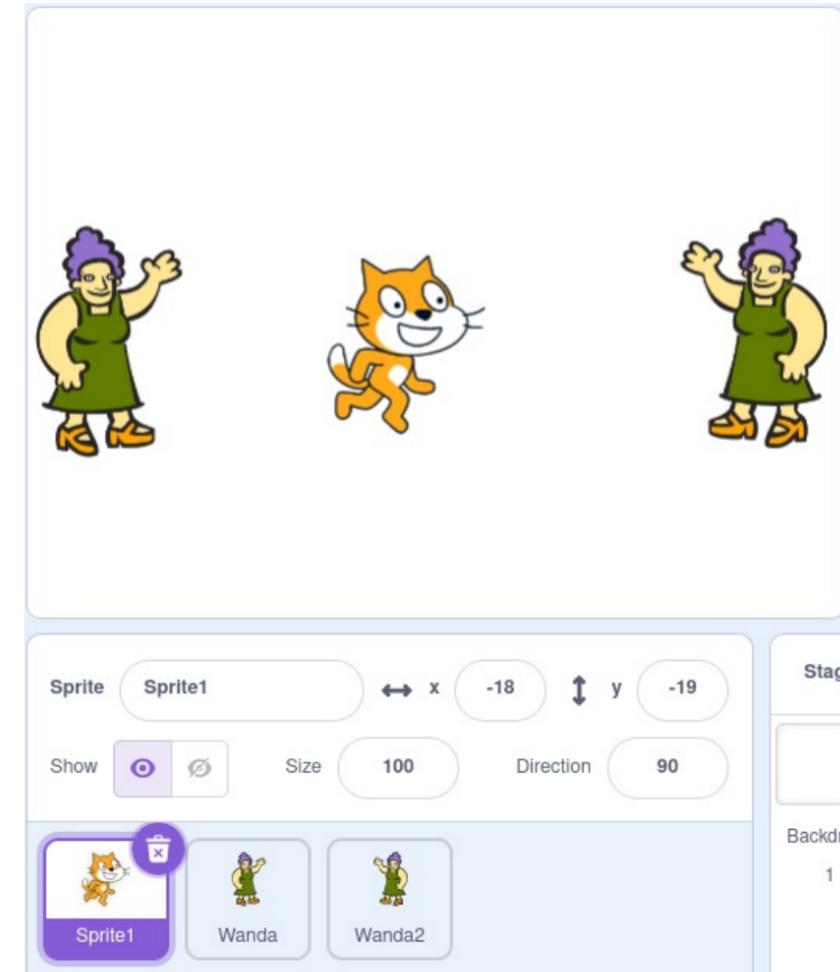
Limites do palco



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

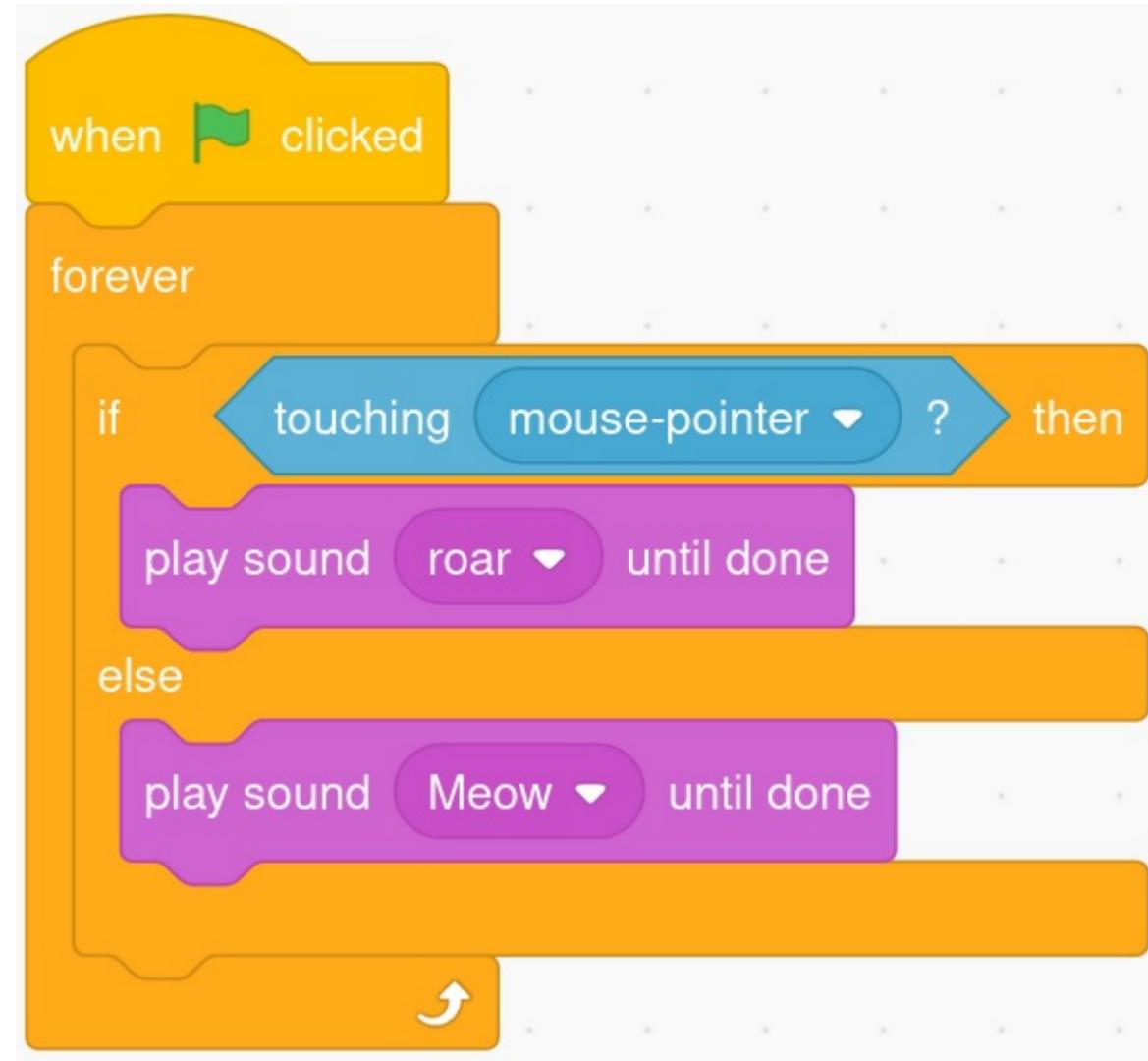
Outros personagens

```
when green flag clicked
  set rotation style to left-right
  forever loop
    if touching Wanda or touching Wanda2 then
      turn 180 degrees
      move 10 steps
      next costume
```



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

Sons

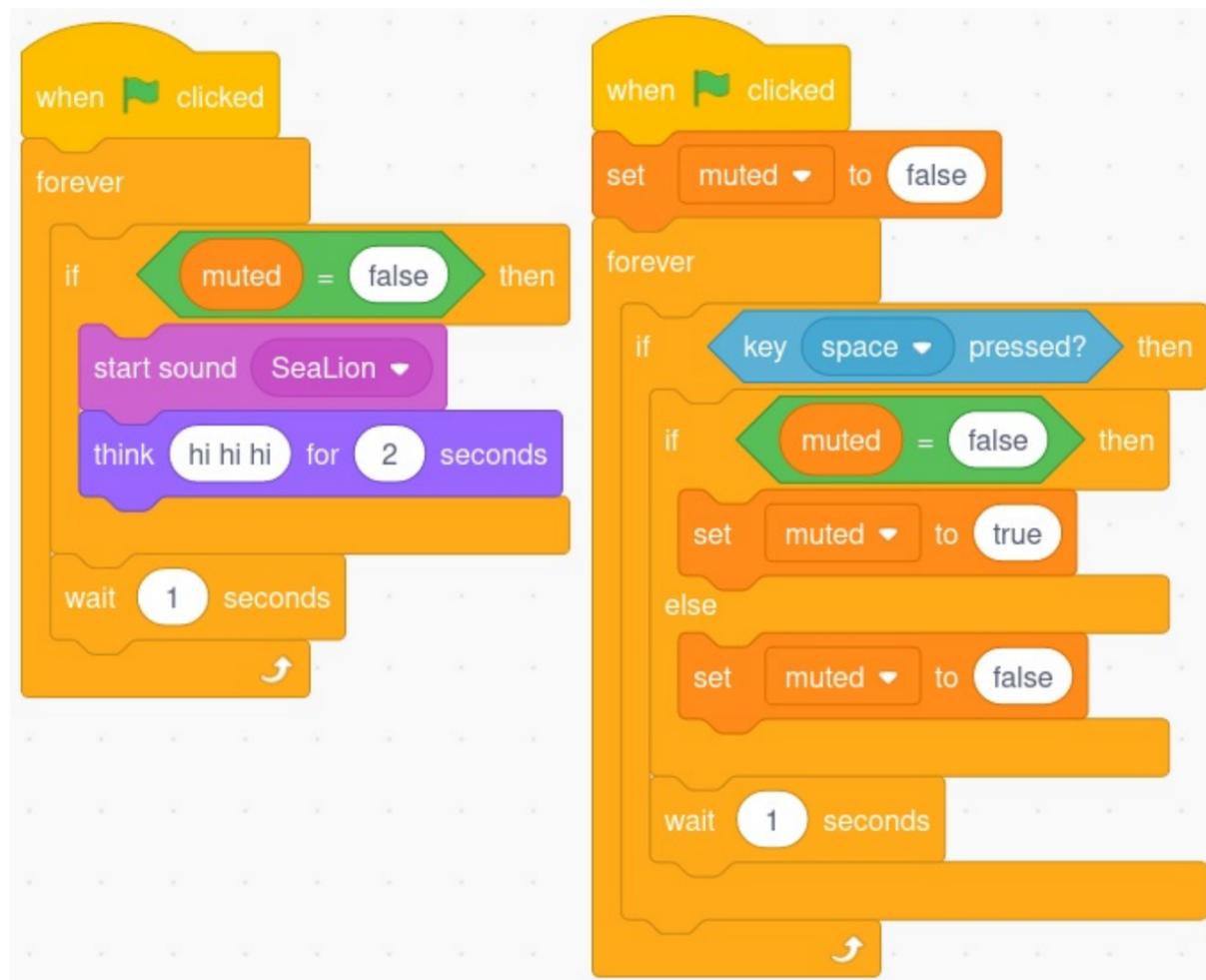


<https://scratch.mit.edu/projects/984136673/>

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:

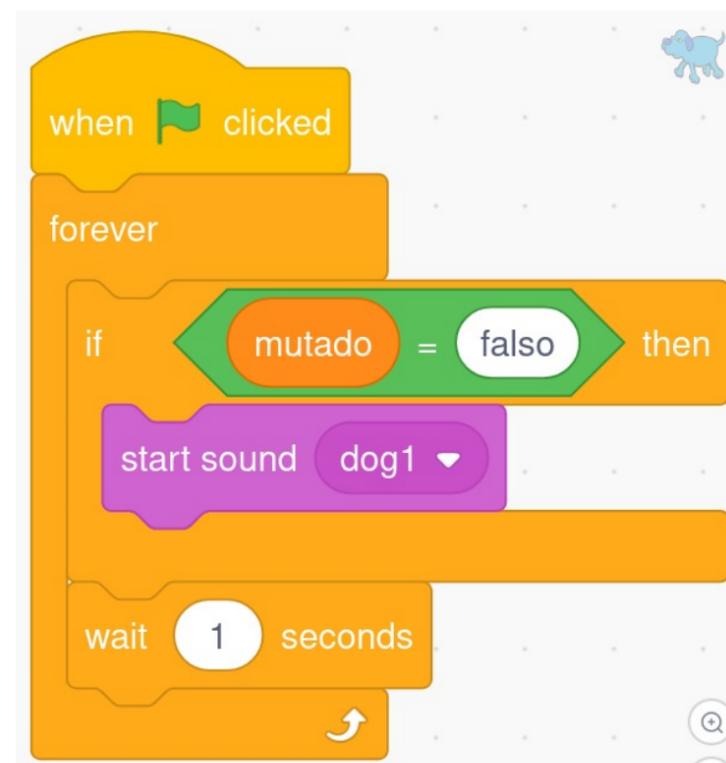
<https://scratch.mit.edu/projects/807110618>



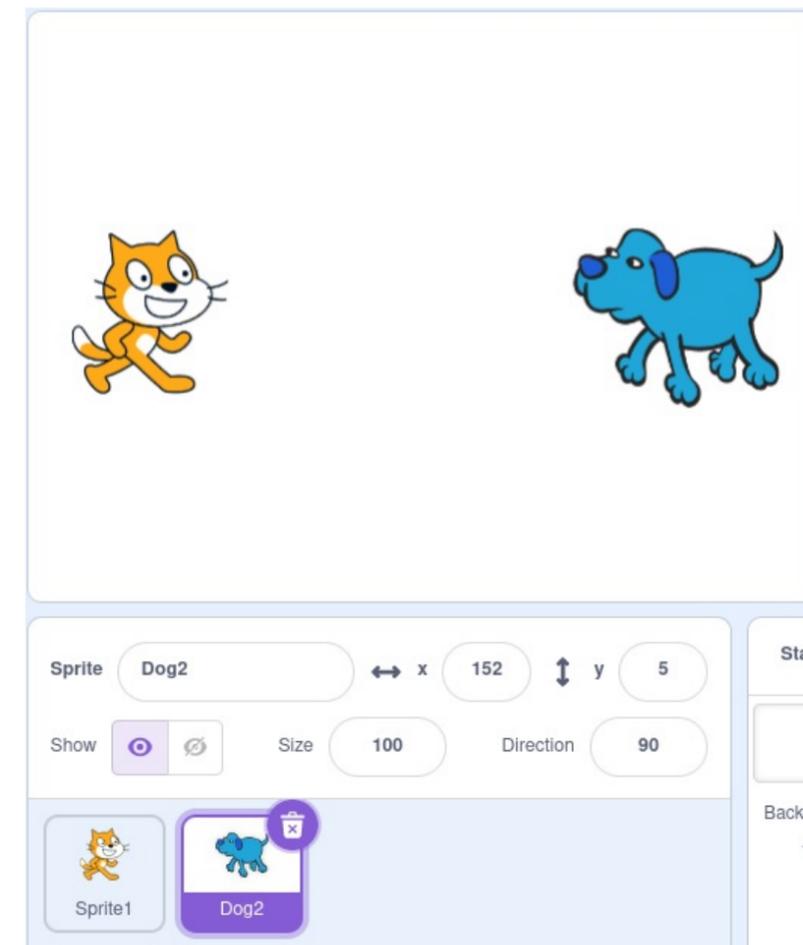
Programação Concorrente



```
when green flag clicked
  set mutado to falso
  forever loop
    if key space pressed? then
      if mutado = verdadeiro then
        set mutado to falso
      else
        set mutado to verdadeiro
    wait 1 seconds
```



```
when green flag clicked
  forever loop
    if mutado = falso then
      start sound dog1
    wait 1 seconds
```



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

Eventos, ouvintes, mensagens

```
when clicked
  forever
    if key space pressed? then
      say Marco! for 2 seconds
      broadcast event
```

```
when I receive event
  say Polo! for 2 seconds
```



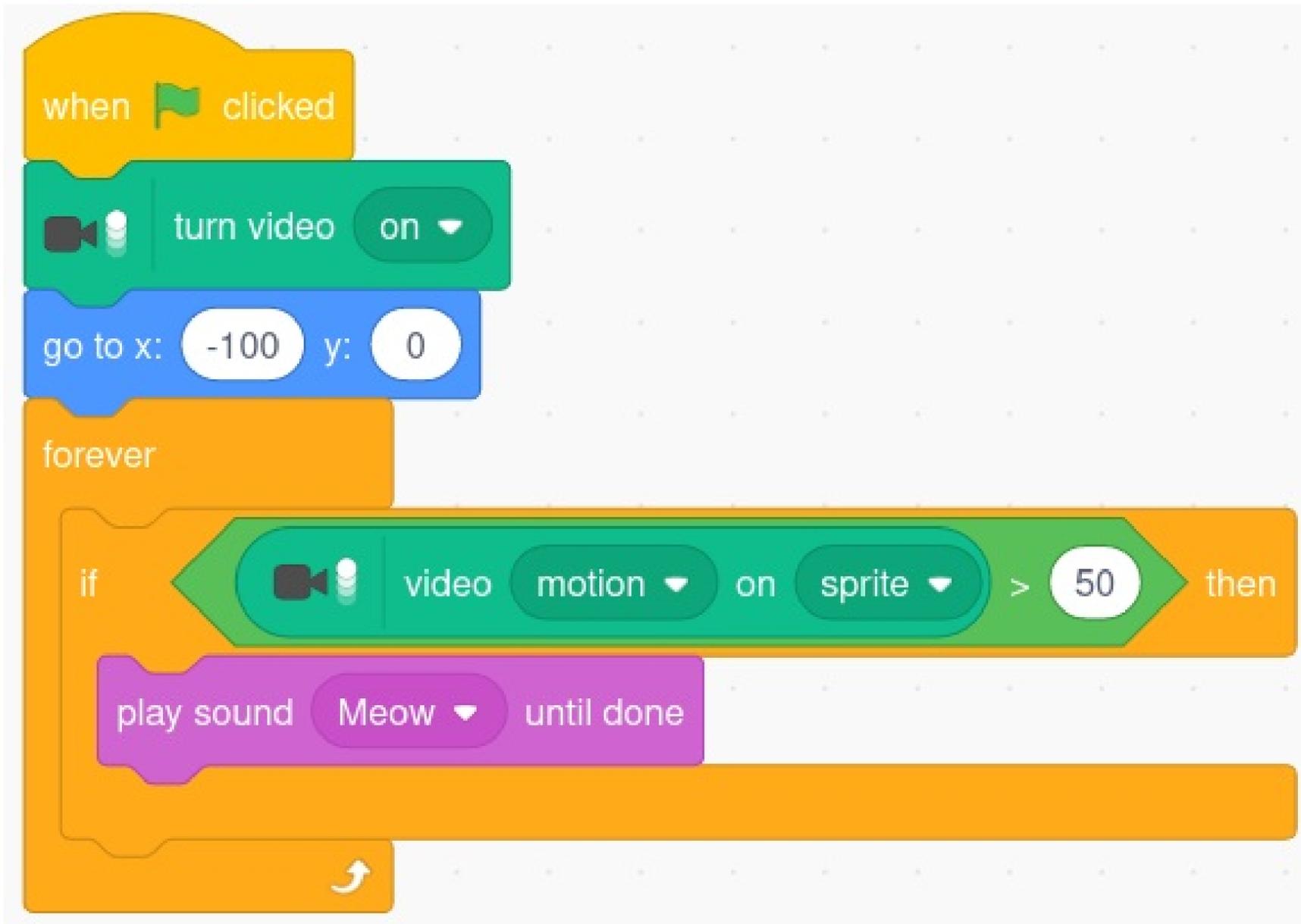
<https://scratch.mit.edu/projects/807110924/>

Blocos avançados

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

Blocos avançados: câmera

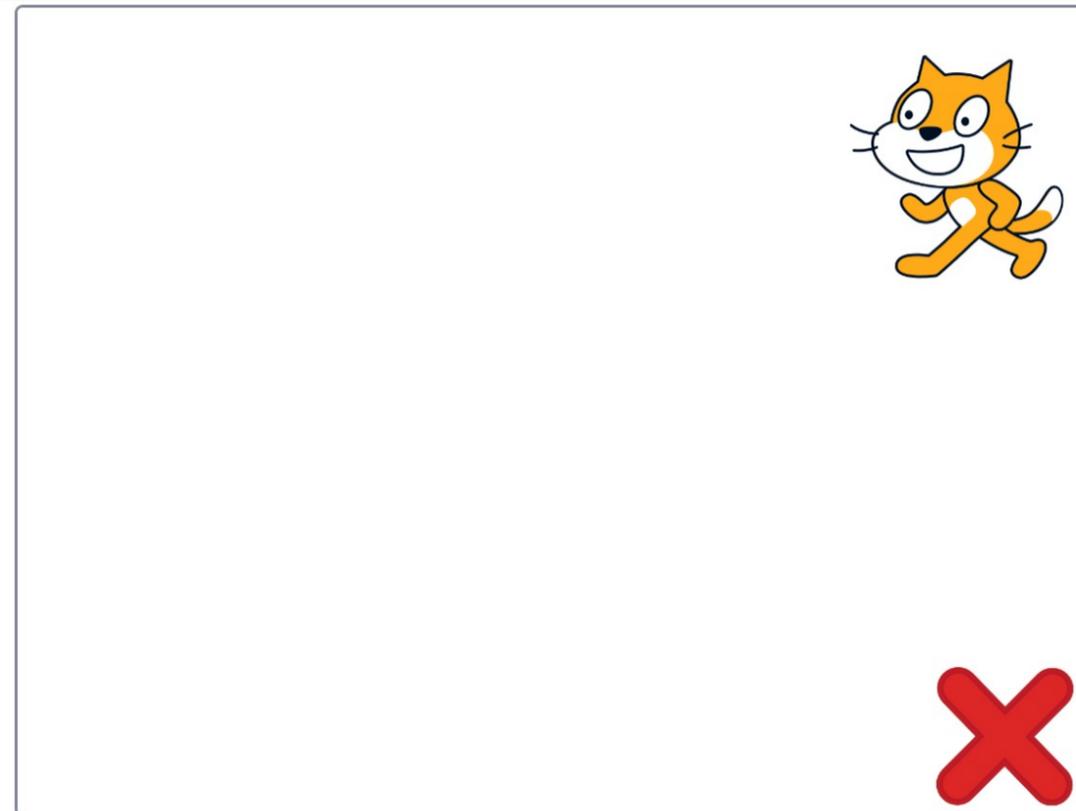
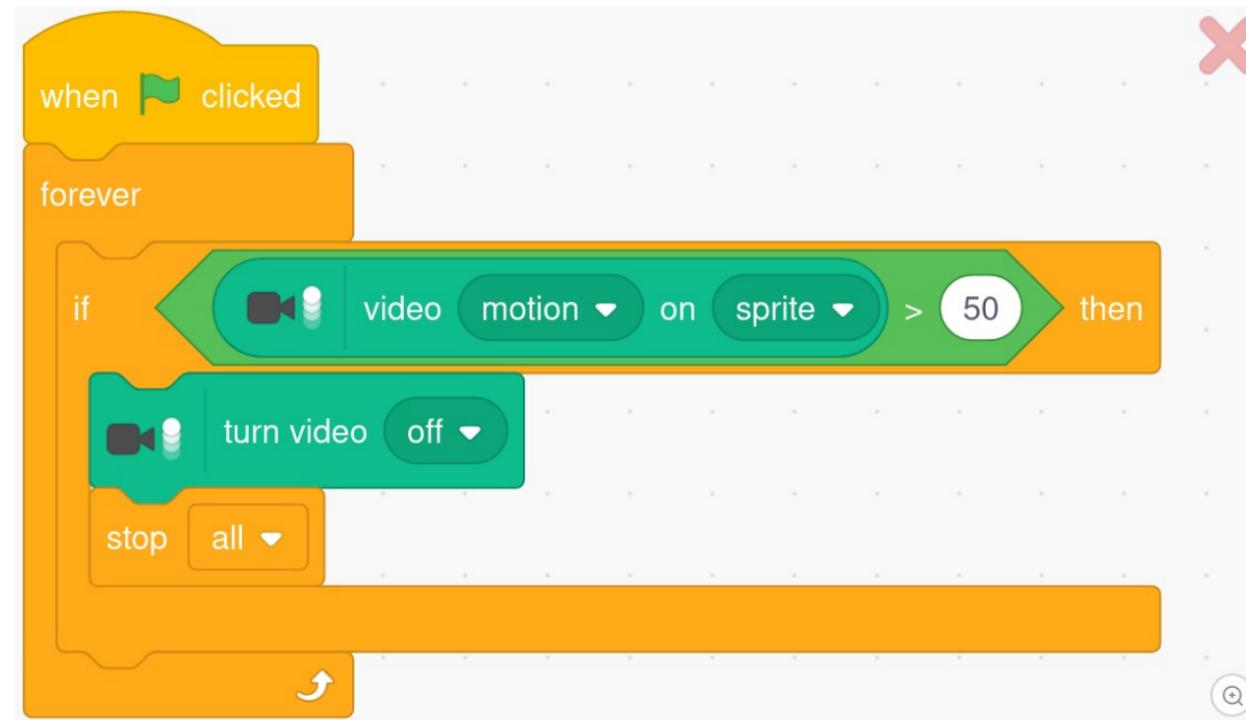
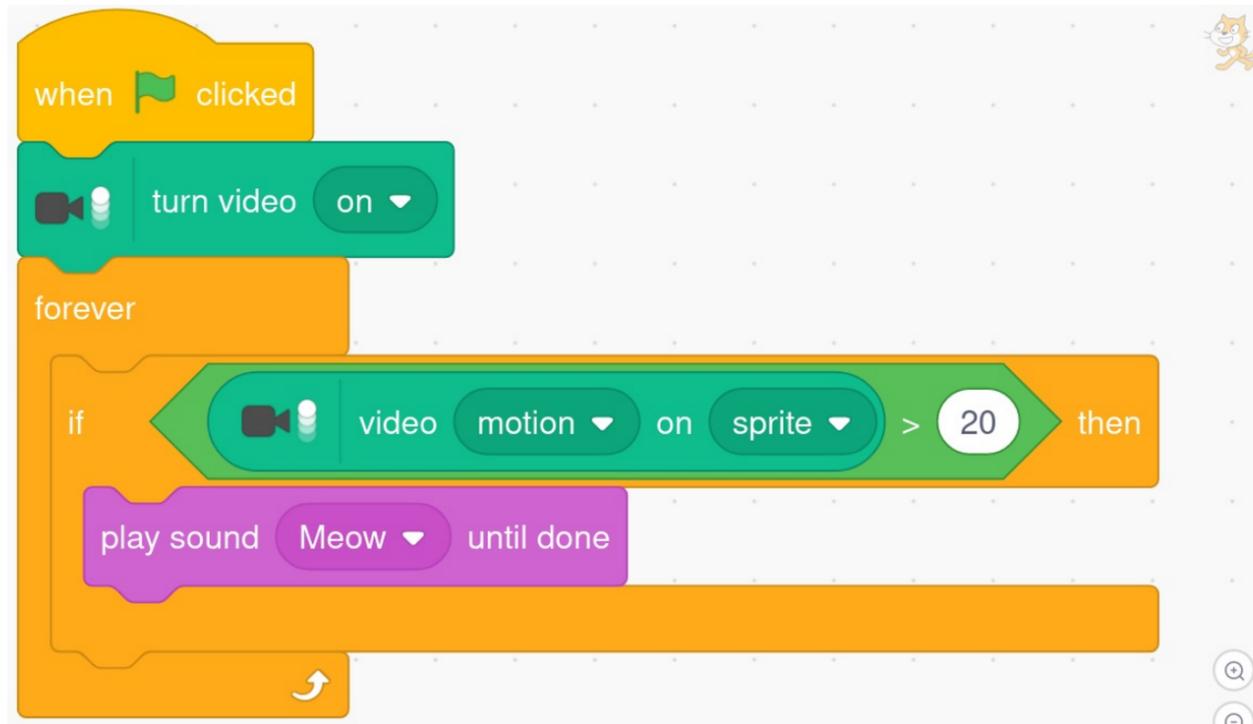
Miar condicionalmente com o movimento da câmera



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

Blocos avançados: câmera

Ligar e desligar



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

Blocos avançados: text-to-speech

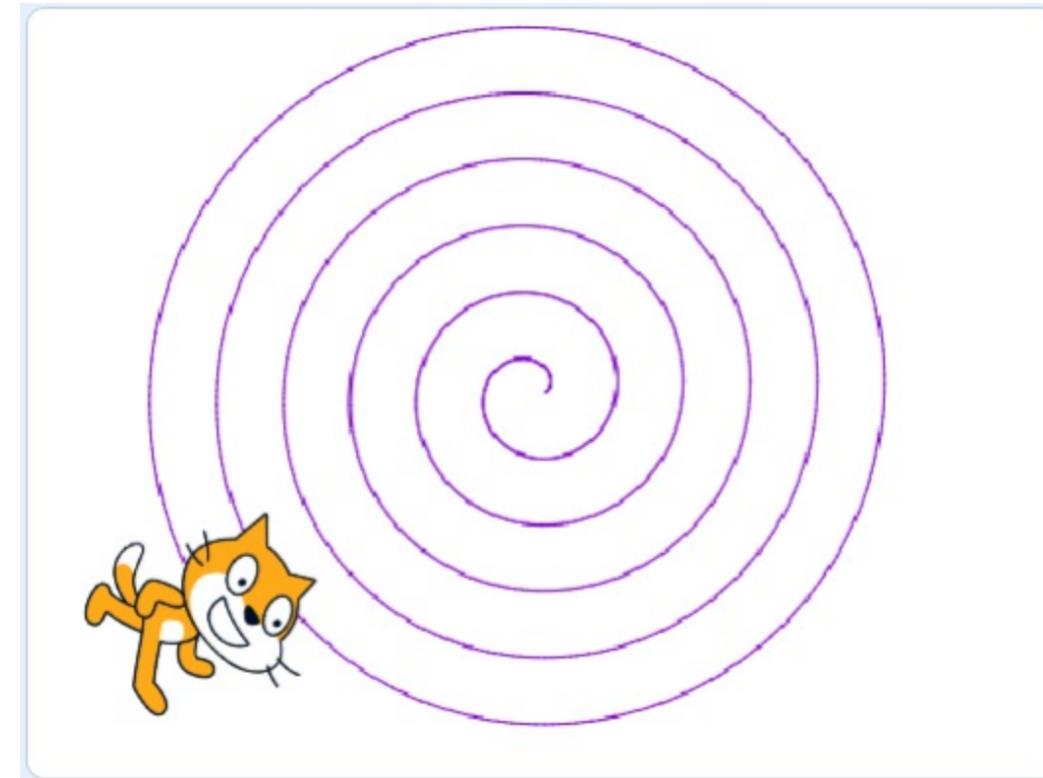
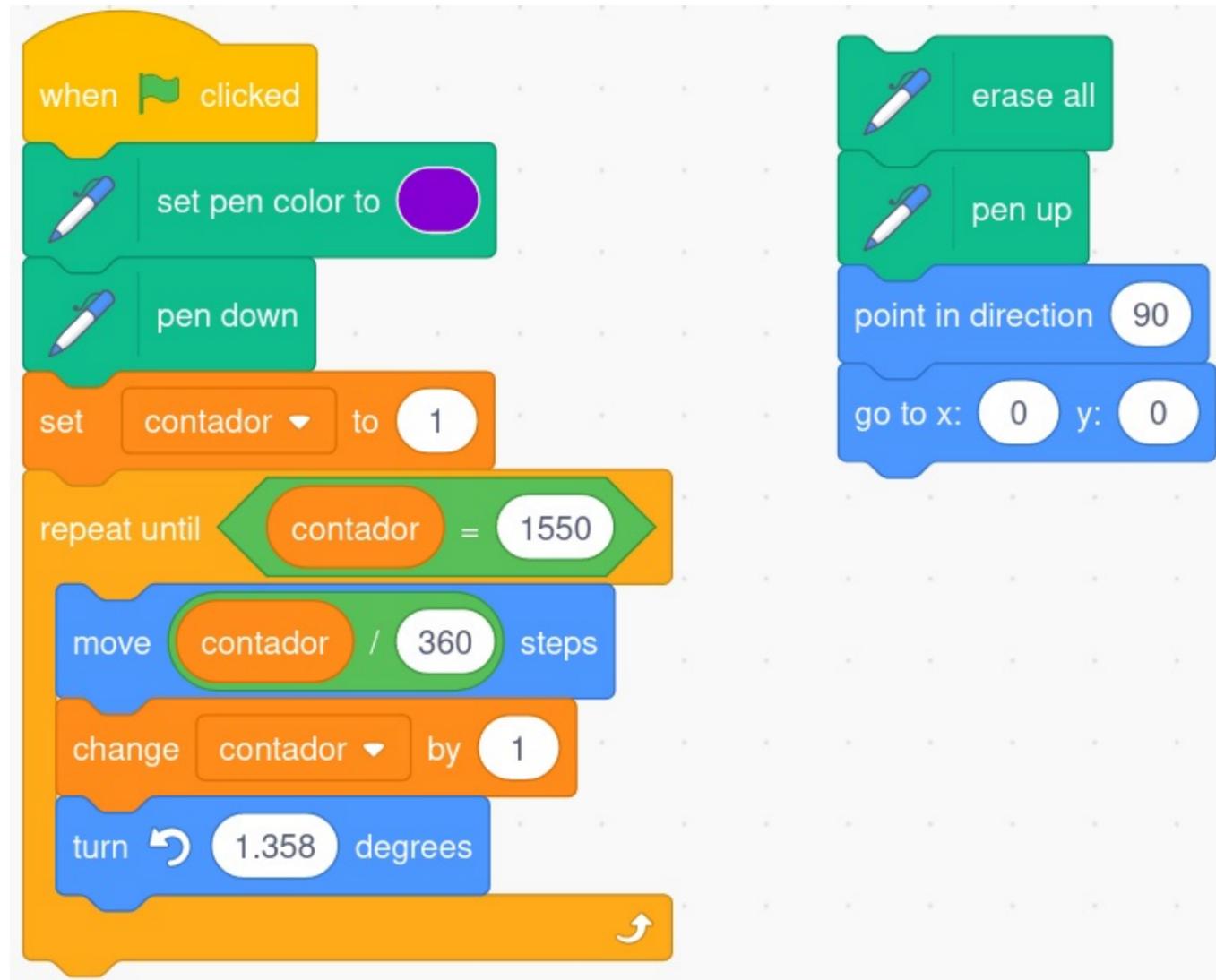
Seu programa pode falar!



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

Blocos avançados: desenhos

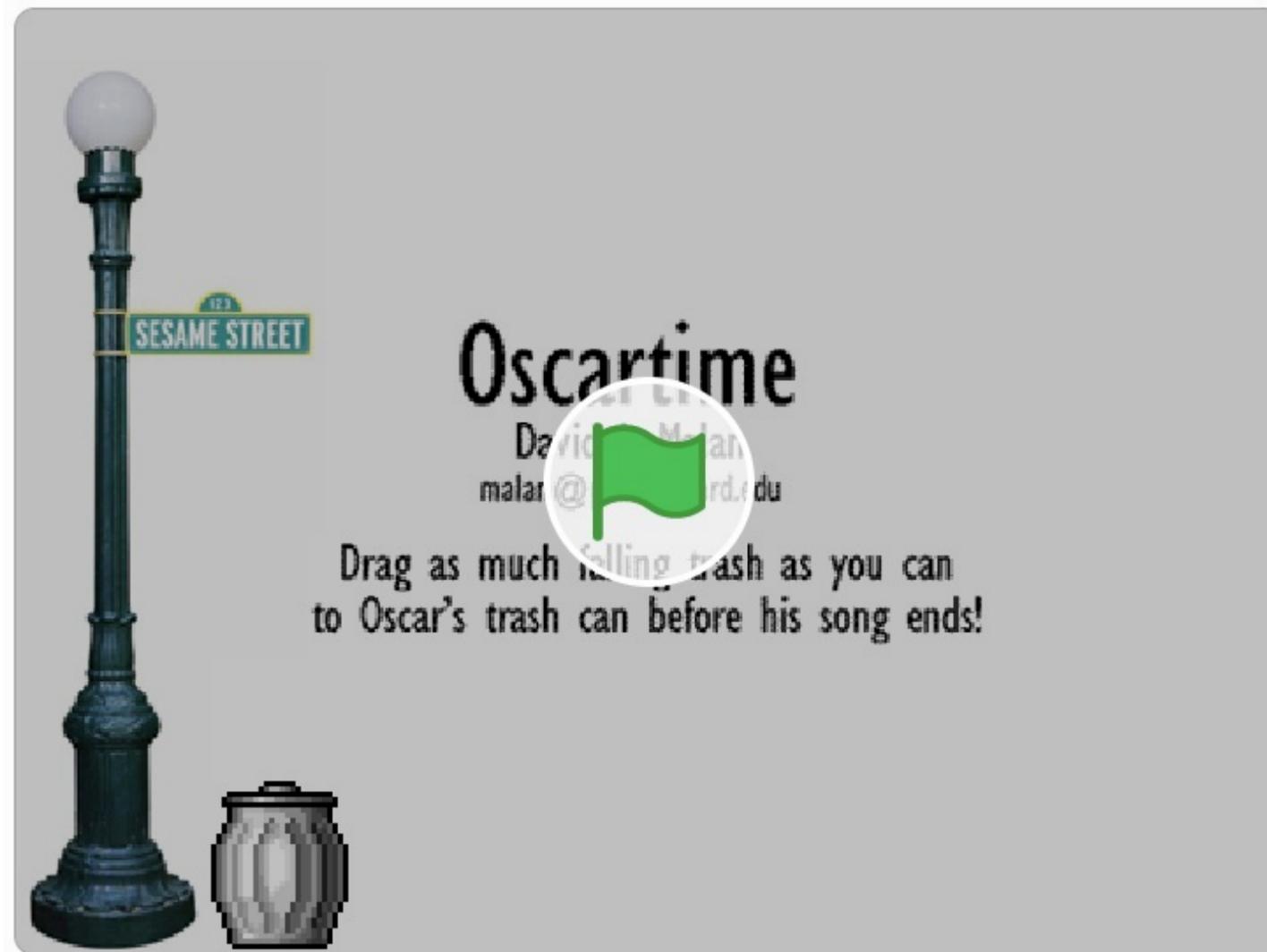
Seu programa pode desenhar!



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

Inspiração

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



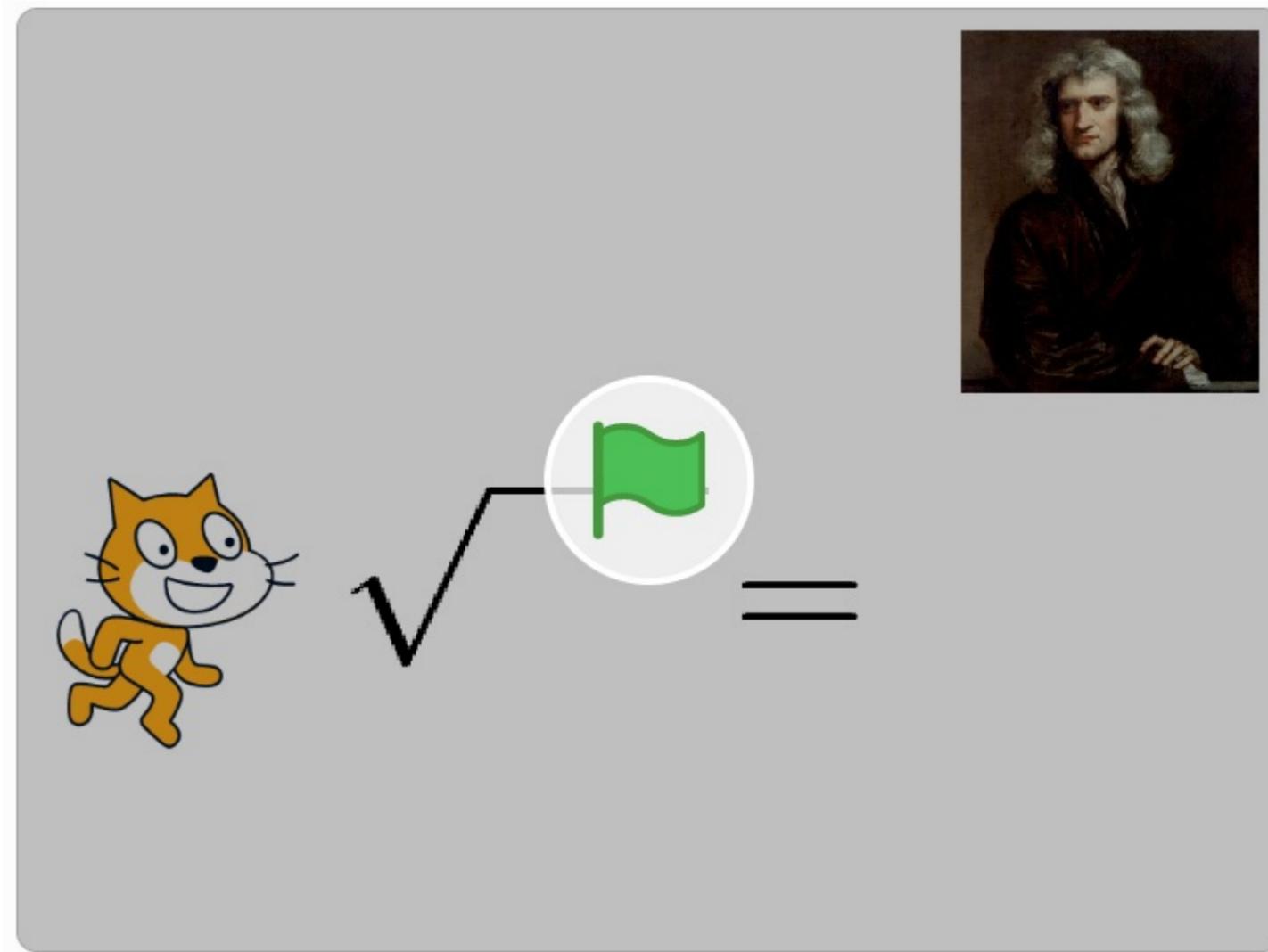
Inspiração

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/>



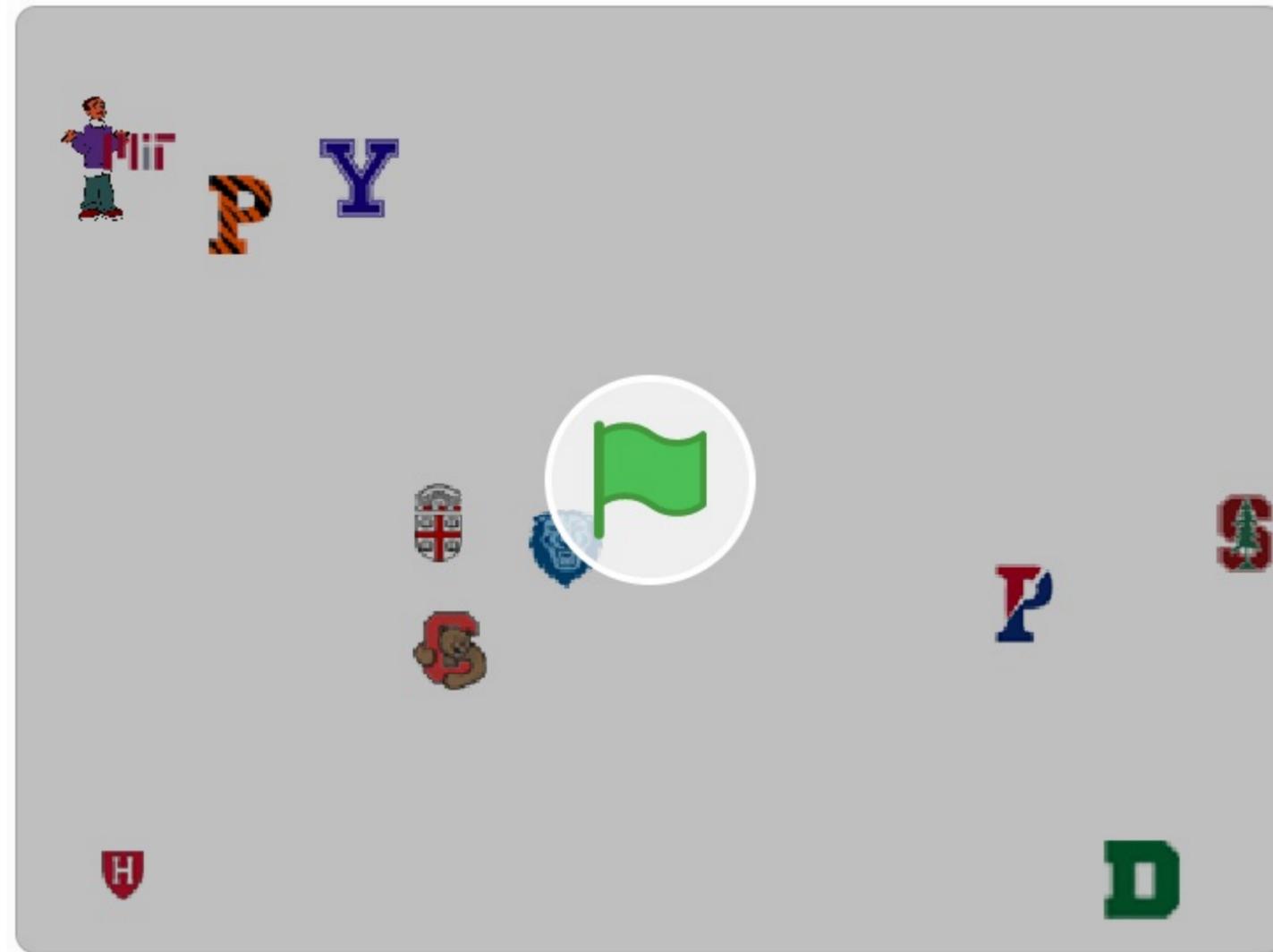
Inspiração

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



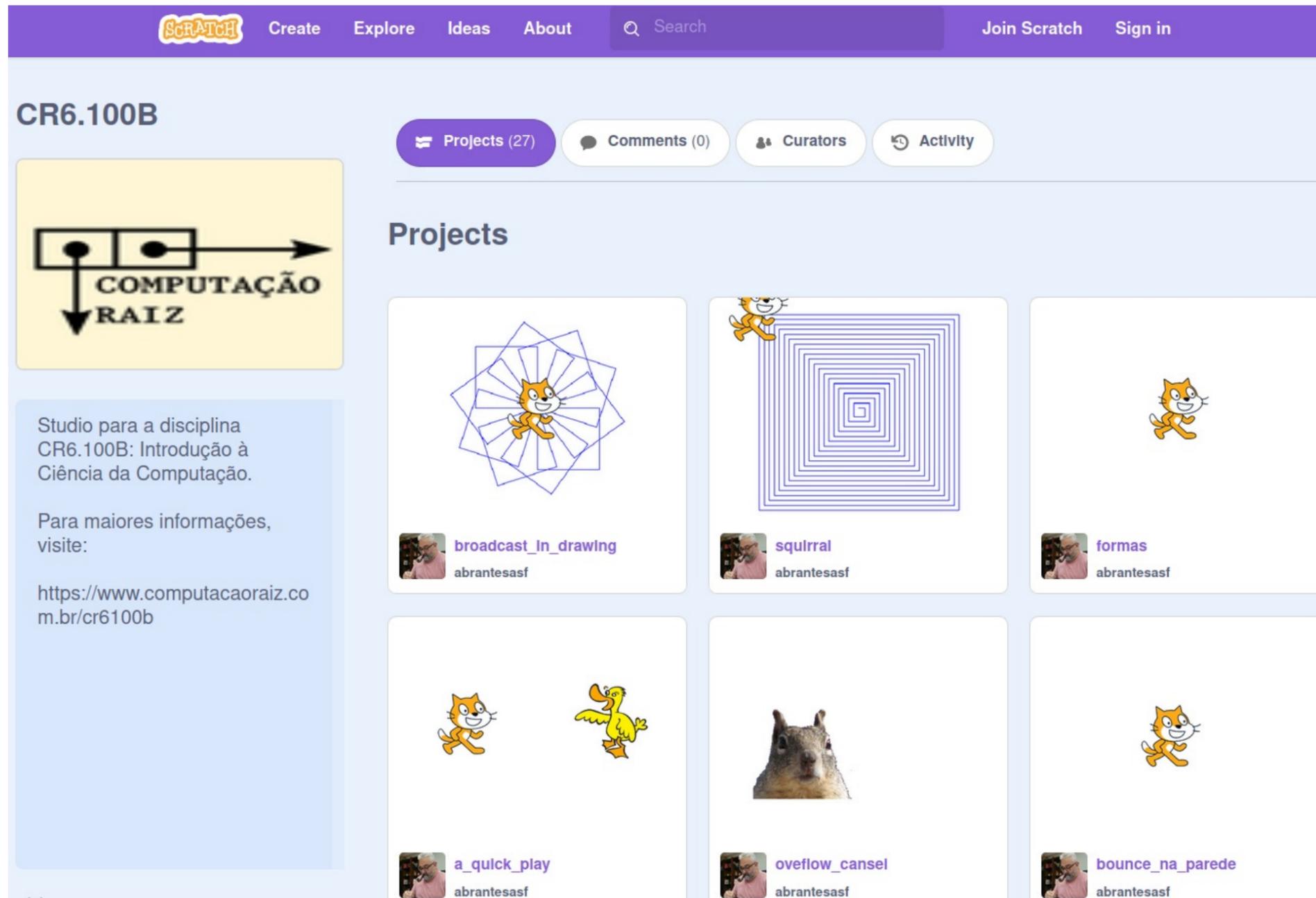
Inspiração

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



Inspiração

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



The image shows a screenshot of a Scratch Studio page. At the top, there is a purple navigation bar with the Scratch logo, links for 'Create', 'Explore', 'Ideas', 'About', a search bar, and 'Join Scratch' and 'Sign in' buttons. Below the navigation bar, the studio title 'CR6.100B' is displayed. To the right of the title are four buttons: 'Projects (27)', 'Comments (0)', 'Curators', and 'Activity'. On the left side, there is a yellow square containing a logo with a computer monitor, a mouse, and the text 'COMPUTAÇÃO RAIZ'. Below this logo, there is a text box with the following content: 'Studio para a disciplina CR6.100B: Introdução à Ciência da Computação. Para maiores informações, visite: <https://www.computacaoraiz.com.br/cr6100b>'. The main area of the studio is titled 'Projects' and displays a grid of six project thumbnails. Each thumbnail shows a different Scratch project, including a cat character in a geometric pattern, a cat character in a maze, a cat character, a cat character and a yellow bird, a squirrel, and a cat character. Each project thumbnail includes the project name and the creator's name, 'abranterasf'.

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-IXGoRV1V7u4>)

Em resumo

