

Scratch: Ambiente Visual para Executar Algoritmos

Uma nova forma de visualizar a lógica que você já conhece

Eduardo Ogasawara

eduardo.ogasawara@cefet-rj.br

<https://eic.cefet-rj.br/~eogasawara>

💡 CONCEITO CENTRAL

Por Que Vamos Usar Scratch

Não É Nova Linguagem

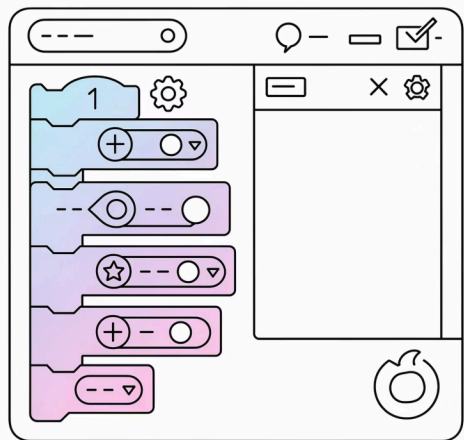
Scratch não é outra programação. É apenas outra forma de visualizar.

Mesma Lógica

A lógica de programação permanece idêntica ao que você já aprendeu.

Forma Diferente

Só muda a maneira de mostrar o algoritmo funcionando.



Partes Essenciais do Scratch

Explore os componentes que dão vida aos seus projetos no Scratch (<https://scratch.mit.edu/projects/editor/?tutorial=getStarted>)

O Palco

Onde vemos o resultado visual das suas criações, a "saída" do algoritmo.

O Personagem

O ator principal que executa as ações e comandos do seu algoritmo.

Os Blocos

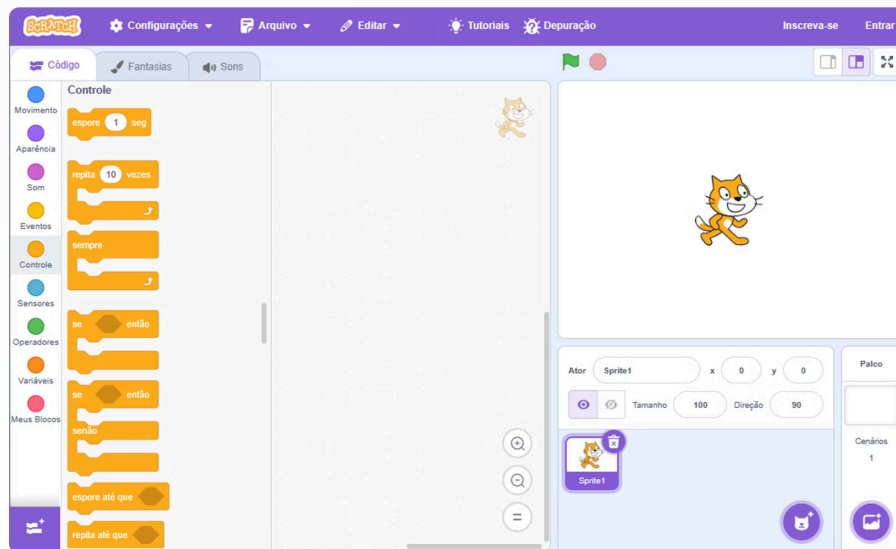
As peças coloridas que representam os comandos e instruções disponíveis.

Área do Código

O espaço onde você monta seu algoritmo, organizando os blocos em sequência.

Bandeira Verde

O botão universal para iniciar a execução do seu projeto.



O Que Você Já Sabe



Variáveis

Você já domina o conceito de criar e usar variáveis para armazenar dados.



Decisões

Estruturas condicionais e lógica de decisão já fazem parte do seu conhecimento.



Execução Passo a Passo

Você entende como algoritmos são executados sequencialmente.

Agora vamos **enxergar** tudo isso acontecendo visualmente

O Palco e o Personagem

O personagem executa as ações do algoritmo. Ele representa o código em movimento, enquanto o palco mostra o resultado visual.

Nada acontece sem comandos.



Blocos São Comandos



Cada Bloco = Uma Instrução

Cada bloco colorido representa um comando específico do algoritmo.



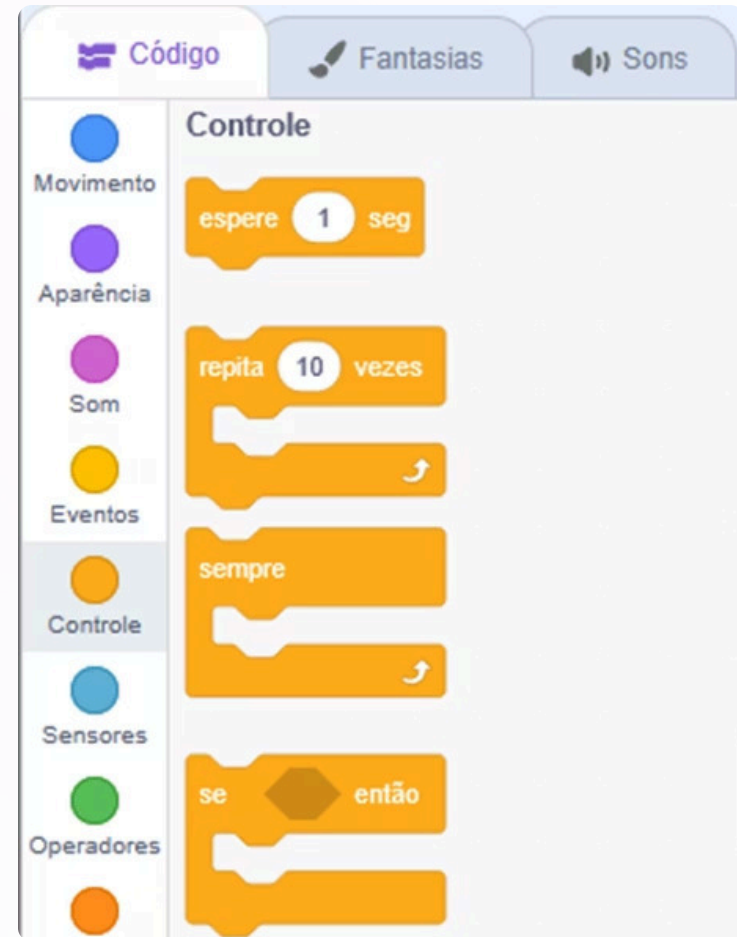
Execução de Cima para Baixo

Os blocos são processados sequencialmente, na ordem em que aparecem.



A Ordem Importa

Assim como no Português, a sequência dos comandos determina o resultado.

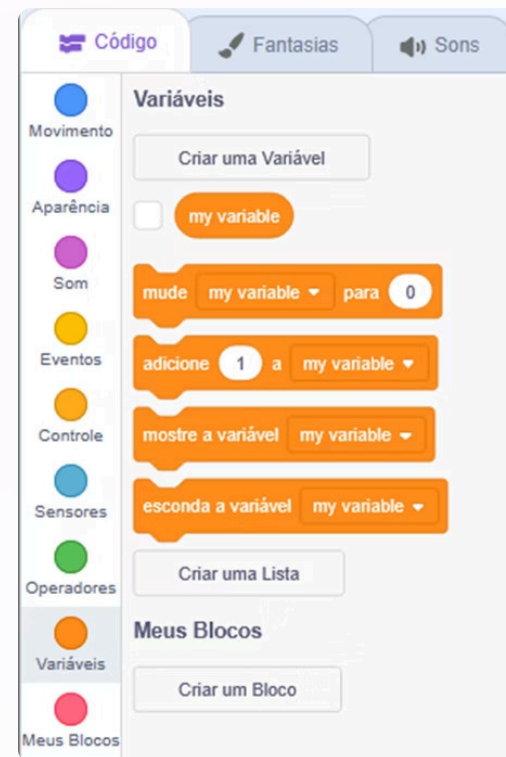


Variáveis no Scratch

Caixinhas de Valores

Variáveis continuam sendo recipientes que guardam informações.

- Têm um nome identificador
- Armazenam valores
- Podem mudar durante execução
- Funcionam igual ao Portugol



📄 A única diferença é a representação visual — a lógica permanece idêntica

Executar É Clicar na Bandeira

1

Clique na Bandeira Verde

Este é o gatilho que inicia a execução do algoritmo.

2

O Algoritmo Começa

Todos os blocos conectados à bandeira são ativados.

3

Blocos Executam em Ordem

Cada instrução é processada sequencialmente.

4

Igual ao Computador

O processo replica exatamente como a máquina executa código.



No exemplo, quando a bandeira verde é clicada, o gato diz "Olá" por 2 segundos.

O Que NÃO Vamos Aprender Agora

✗ Animação

Não é o foco criar animações ou movimentos complexos.

✗ Efeitos Visuais

Não vamos explorar efeitos gráficos ou visuais especiais.

✗ Design

Não é aula de design de personagens ou cenários.

Scratch aqui é só para pensar e visualizar lógica

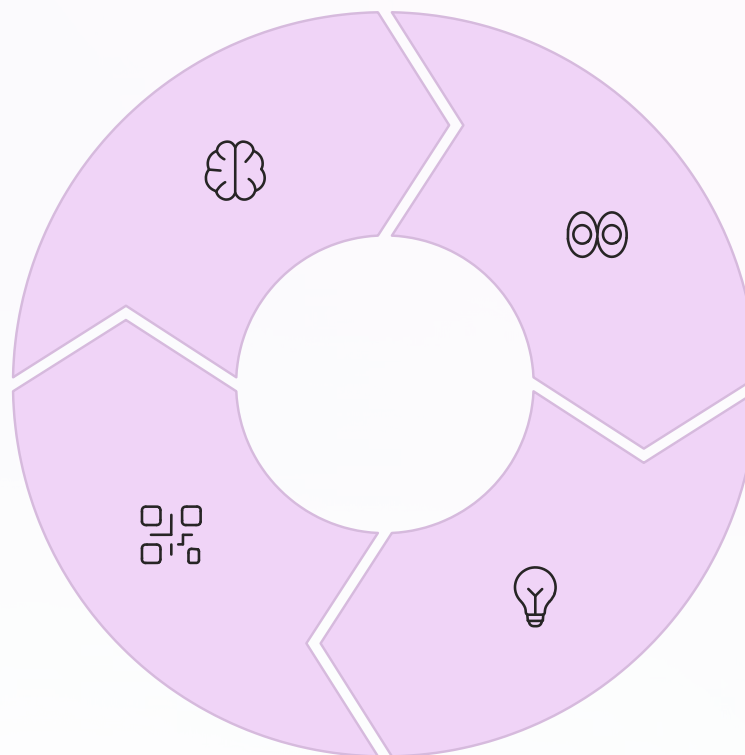
Ideia Central

O Algoritmo Já Existe

A lógica está na sua mente

Isso É Programação

Essência do pensamento
computacional



Scratch Mostra Funcionando

Visualização do processo

Entende Um, Entende Outro

A lógica é universal

Comparação Visual: Blocos e Comandos

No Scratch

Blocos coloridos empilhados verticalmente representam a sequência de comandos.



No Portugol

Linhas de código escritas sequencialmente executam as mesmas instruções.

```
inicio  
  escreva("Olá")  
  x <- 5  
fim
```

Mesma lógica, representações diferentes. O pensamento algorítmico permanece idêntico.

Exemplo Prático: Variáveis em Ação



1

Estado Inicial

Variável **idade** recebe valor 12

2

Processamento

Comando incrementa **idade** em 1

3

Estado Final

Variável **idade** agora contém 13

A caixinha tem nome, guarda valor e pode ser atualizada — exatamente como você já conhece.

Recapitulando



Ferramenta Visual

Scratch é uma forma de ver algoritmos funcionando, não uma nova linguagem.



Mesma Lógica

Variáveis, decisões e execução sequencial — tudo que você já sabe.



Foco no Pensamento

O objetivo é pensar algoritmicamente, não criar animações ou design.

Se você entende a lógica, você entende qualquer representação dela



Referências

1

WING, Jeannette M. Computational thinking. Communications of the ACM, New York, v. 49, n. 3, p. 33–35, 2006.

2

PAPERT, Seymour. Mindstorms: children, computers, and powerful ideas. New York: Basic Books, 1980.

3

PÓLYA, George. How to solve it: a new aspect of mathematical method. 2. ed. Princeton: Princeton University Press, 1957.

4

CAMPOS, A. F. G. A.; CAMPOS, E. A. V. Fundamentos da programação de computadores: algoritmos, Pascal, C/C++ (padrão ANSI) e Java. 3. ed. São Paulo: Pearson, 2012.