

Guia de Introdução ao Scratch

O **Scratch** é uma linguagem de programação visual criada para facilitar o aprendizado de programação, principalmente por crianças e iniciantes. Em vez de digitar códigos, o usuário encaixa blocos coloridos para criar jogos, animações, histórias interativas, músicas e projetos educativos.

O Scratch foi desenvolvido pelo grupo **Lifelong Kindergarten**, do **MIT Media Lab**, nos Estados Unidos, liderado por **Mitchel Resnick**. Seu desenvolvimento começou por volta de **2003**, e a primeira versão pública, o **Scratch 1.0**, foi lançada em **2007**.

Depois surgiram novas versões:

- **Scratch 2.0**: lançado em 2013;
- **Scratch 3.0**: lançado em 2019, com melhor funcionamento em computadores, tablets e celulares.

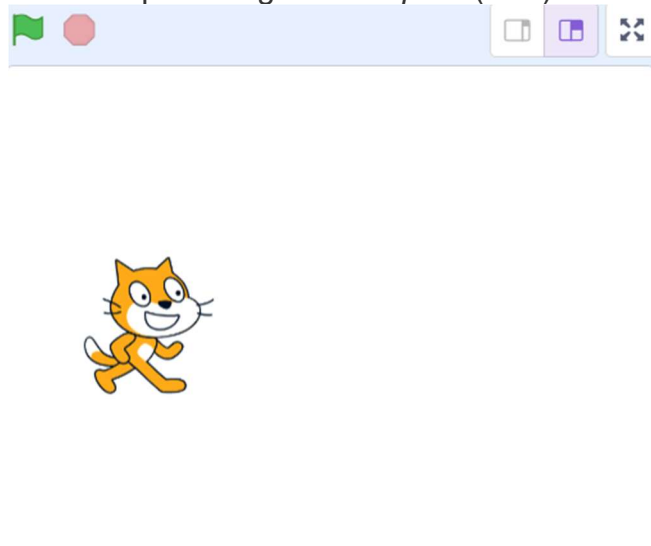
O nome “Scratch” foi inspirado na técnica dos DJs chamada **scratching**, pois a ferramenta permite misturar imagens, sons, movimentos e comandos para criar novos projetos.

O Scratch ajuda a desenvolver o raciocínio lógico, a criatividade, a resolução de problemas e o pensamento computacional.

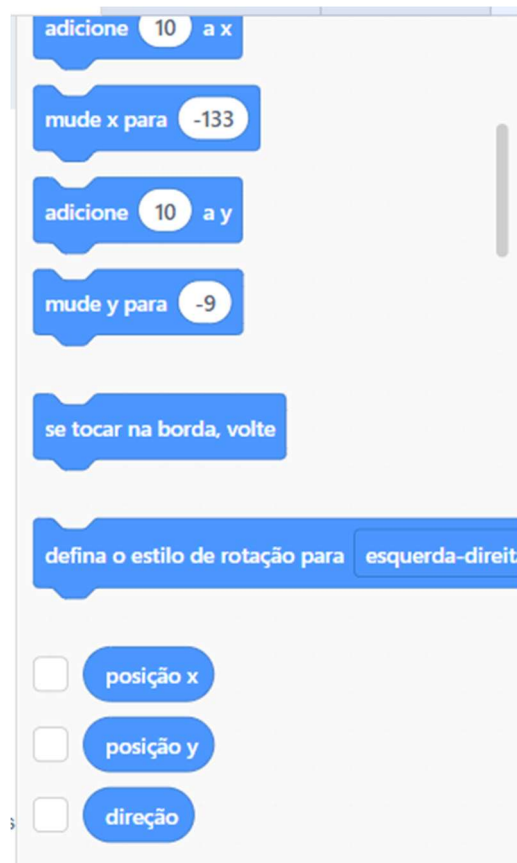
1. INTERFACE

O ambiente do Scratch é dividido em:

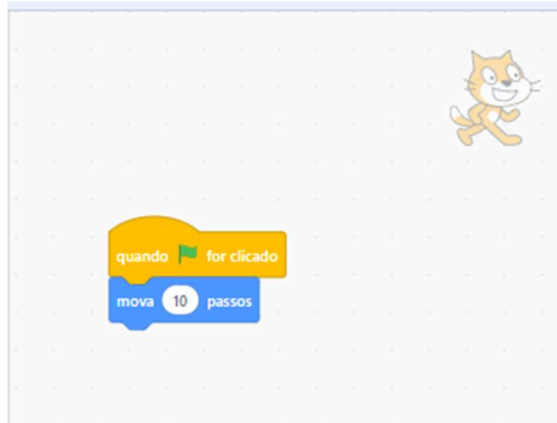
- **Palco**: Onde a ação acontece. O personagem é o *Sprite* (Ator).



- **Área de Blocos:** O menu de instruções.



- **Área de Programação:** Onde você monta seu código.



2. CATEGORIA DE BLOCOS

Categoria	Função
 Movimento	Controle posição e deslocamento
 Aparência	Controla fala e visuais.
 Som	Controla o som

 Eventos	Controla os eventos (gatilhos das ações)
 Controle	Loop's (qtde de vezes que deve ser executado), condicionais (condições que devem ser executadas)
 Sensores	Permite que o programa perceba ações, contatos, distâncias, teclas pressionadas e outras informações do ambiente.
 Operadores	Realizar cálculos, comparações e operações lógicas dentro do programa.
 Variáveis	Armazenam informações que podem mudar durante a execução do programa.

3. PRIMEIRO PROGRAMA O ATOR QUE ANDA

Monte o seguinte código:

```
Quando bandeira verde clicada
sempre:
  Mover 10 passos
  Se tocar na borda, volte
```

1 - Vamos ver este primeiro programa passo à passo.



2 - Temos que resolver o problema nosso ator não volta.



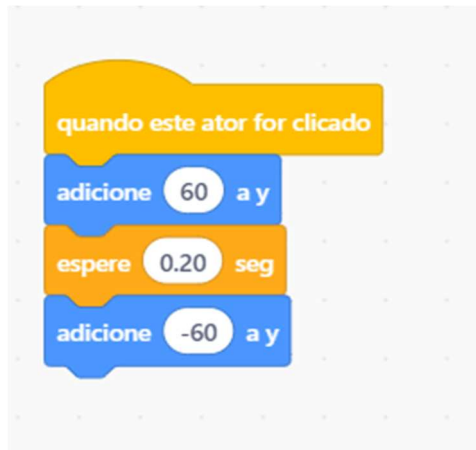
3 – Agora outro problema , ele volta de cabeça para baixo.



Melhorando um pouco nosso dando sensação de “movimento”



Vamos trabalhar com posicionamento, vamos simular que quando clicamos no “ator” ele dara um salto. Vamos adicionar o codigo abaixo na “Area de Programação”



Agora vamos trabalhar com a ideia de variáveis. Mas afinal o que uma variável ?

No Scratch, uma variável funciona exatamente como uma "caixa" na memória do computador onde você pode guardar um valor (um número ou um texto) para usar depois.

Aqui está uma definição simples e prática:

Imagine que você tem uma caixa com uma etiqueta. Você pode colocar algo dentro dessa caixa, ler o que está nela ou trocar o conteúdo quando quiser. No Scratch, essa "caixa" é a variável, e a "etiqueta" é o nome que você dá a ela (como pontos, vidas, velocidade ou nome_do_jogador).

Por que usamos variáveis?

Sem variáveis, o seu programa não teria "memória". Com elas, o Scratch consegue:

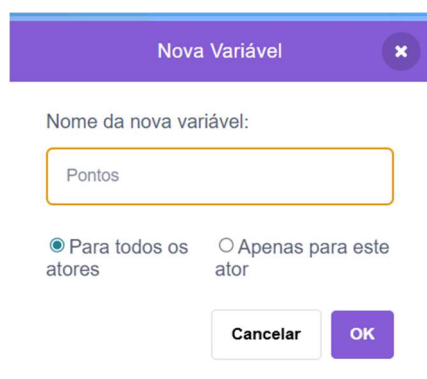
Contar: Armazenar quantos pontos o jogador fez.

Controlar o jogo: Saber quantas vidas restam antes de dar "Game Over".

Guardar informações: Memorizar o nome do jogador ou o nível atual em que ele está.

Fazer cálculos: Aumentar a velocidade de um objeto conforme o tempo passa.

Primeiro vamos criar s variáveis que vamos usar.



Vai aparecer esta tela

Variáveis

Criar uma Variável

☒ Pontos

mude Pontos para 0

adicione 1 a Pontos

mostre a variável Pontos

esconda a variável Pontos

Criar uma Lista

Meus Blocos

Criar um Bloco

Vamos criar a segunda variavel

Nova Variável

Nome da nova variável:

Velocidade

☒ Para todos os atores ☐ Apenas para este ator

Cancelar OK

Variáveis

Criar uma Variável

☒ Pontos

☐ Velocidade

mude Pontos para 0

adicione 1 a Pontos

mostre a variável Pontos

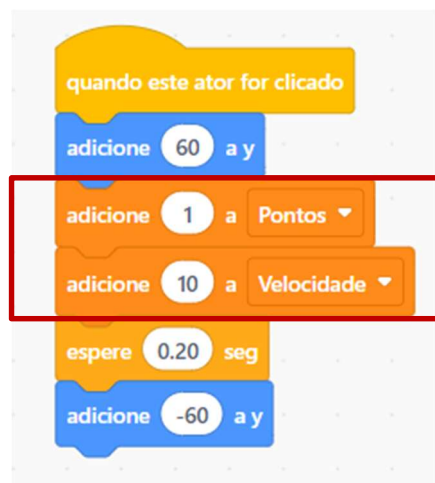
esconda a variável Pontos

Criar uma Lista

Meus Blocos

Criar um Bloco

Bem agora hora de programar.



E se quisermos trabalhar com números aleatórios?

Então vamos programar.

