



LETRAMENTO DIGITAL

L

D

Curso:
Especialização em Letramento Digital

Introdução à Lógica de Programação

Pós-Graduação Lato Sensu em Letramento Digital

Profa. Dra. Elloá B. Guedes

ebgcosta@uea.edu.br

www.elloaguedes.com

Profa. Dra. Elloá B. Guedes

Formação:

- **Doutorado em Ciência da Computação – UFCG (2013)**
- **Mestrado em Ciência da Computação – UFCG (2011)**
- **Graduação em Ciência da Computação – UFCG (2009)**

Como a disciplina está organizada?

Aula 01 - A importância da programação e os primeiros fundamentos

- Manhã – Motivação e contextualização, conhecendo o Scratch
- Tarde – Conhecendo os blocos
- 10 horas/aula

Aula 02 - Conhecendo e praticando comandos

- Manhã – Comandos de entrada e saída
- Tarde – Desvio Condicional
- 10 horas/aula

Como a disciplina está organizada?

Aula 03 - Melhorando as habilidades

- Manhã – Comandos de Repetição
- Tarde – Integrando conceitos
- 10 horas/aula

Como está organizada nossa aula nesta manhã?

Aula 1/Manhã: A importância da programação e os primeiros fundamentos

- 8h as 9h30min – Aula
- 9h30min as 10h- DLI - Atividade 01
- 10h as 10h10min - Intervalo
- 10h10min as 10h30min – Interatividade
- 10h30min as 12h – Aula

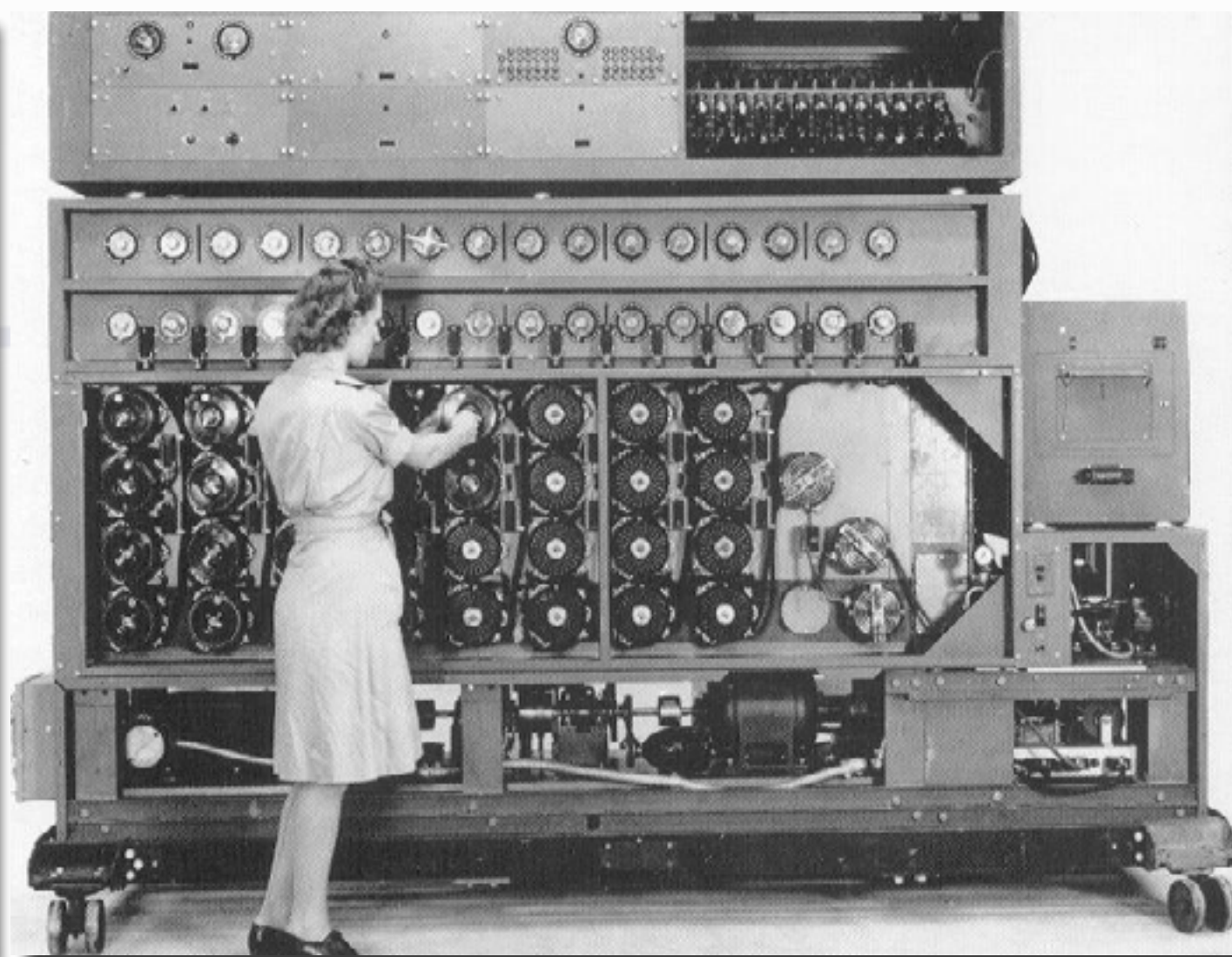
Objetivos

1. Motivar e contextualizar a importância do ensino de programação na formação dos estudantes
2. Conhecer o ambiente de programação Scratch e seus componentes



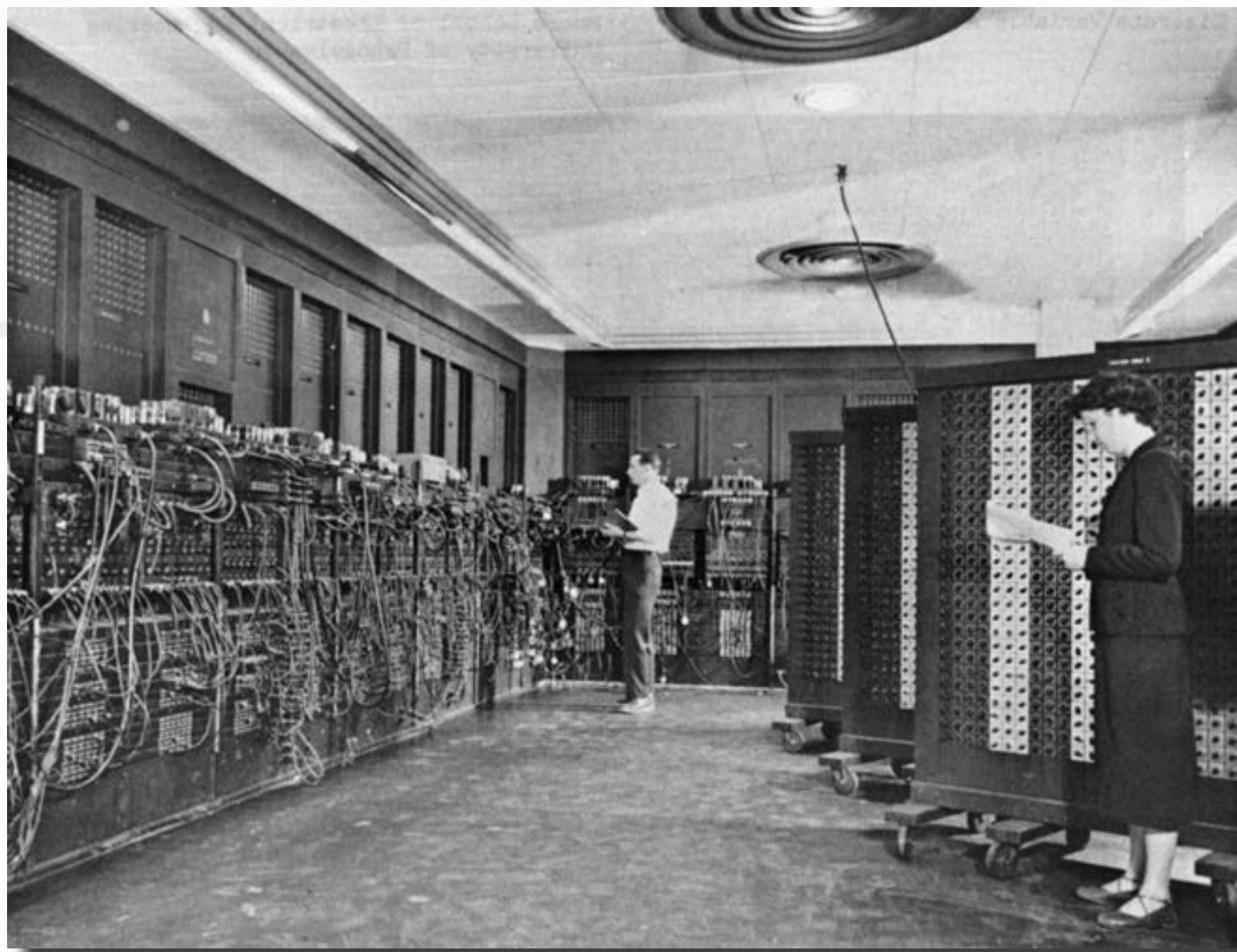
Contextualização

- Alan Turing, matemático britânico
- 1936: Noção de Computação como conhecemos hoje



Contextualização

- ENIAC, 1943: Electronic Numerical Integrator and Calculator
- Pesava 27 toneladas e ocupava 167m²



- **Precisava de pessoas para operar cabos e válvulas**
- **Uso de cartões perfurados**
- **Resultado era mostrado em lâmpadas**
- **Funcionou por dez anos**
- **Peças em museus de todo o mundo**

Contextualização

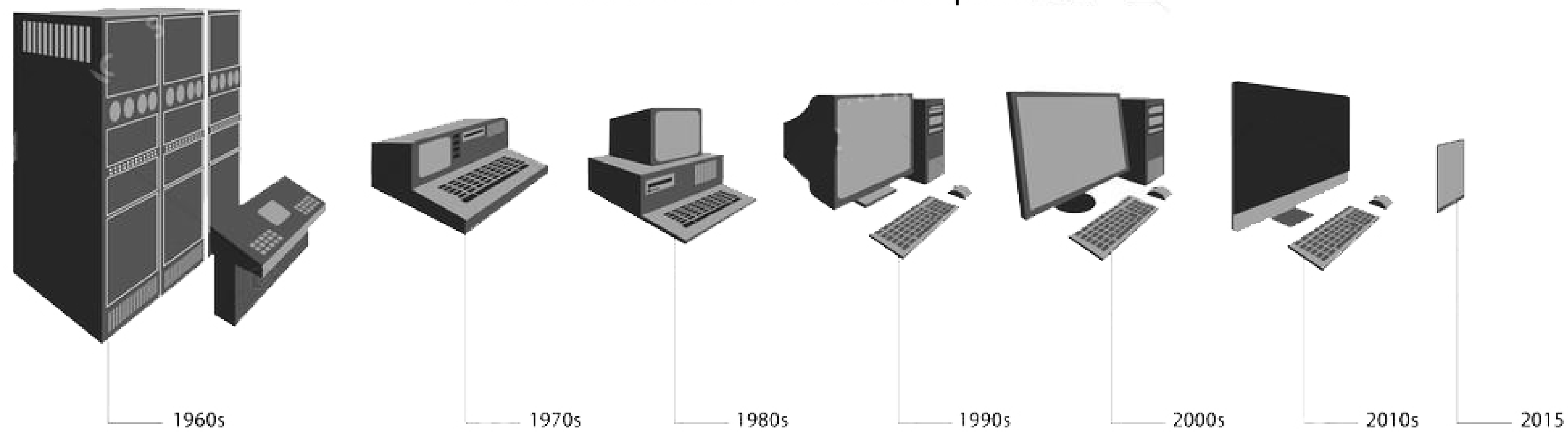
- Computadores só possuíam um programa
- Eram muitíssimo grandes e caros
- Demandavam muitas pessoas para funcionar
- Consumiam muita energia
- Precisavam de muita manutenção



- Grace Hooper
- Bug

Contextualização

Evolution of computers



- **Evolução de várias áreas**
- **Química: bateria**
- **Elétrica: componentes e circuitos**
- **Produção: fabricação**
- **Computação: sistemas operacionais, programas**
- **Muitas outras**

Contextualização

- Com melhorias, computadores foram diminuindo de tamanho e aumentando suas capacidades
- Passaram a estar disponíveis nas universidades e grandes empresas



- Preço tornou-se mais acessível
- Melhorias realizadas aproximam o computador das pessoas comuns



- **Propaganda de 1980 (40 anos atrás)**
- **Valor aproximado: 67 mil reais**
- **Menor poder de processamento que seu celular atualmente**

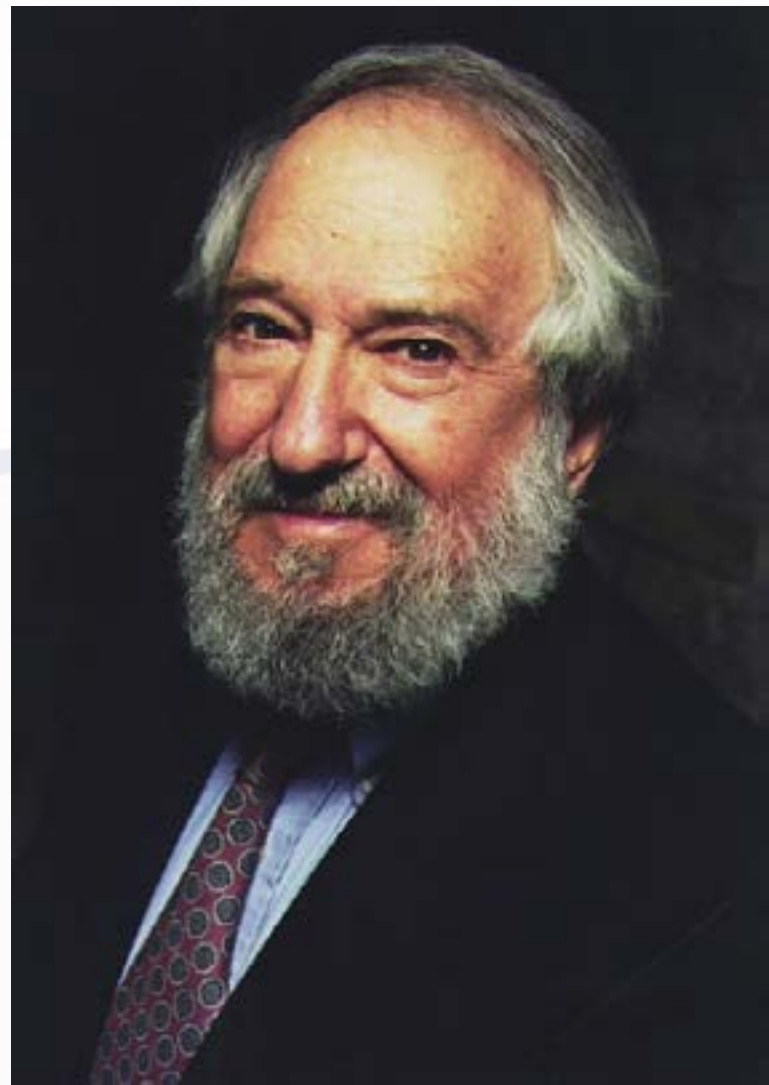
Contextualização

- Dispositivos de entrada: teclado, microfone, câmera, etc.
- Dispositivos de saída: monitor, caixas de som, impressora, etc.
- Sistema operacional, ex.: Windows
- Programas instalados
 - Pacote Office
 - Calculadora
 - Jogos
 - Muitos outros

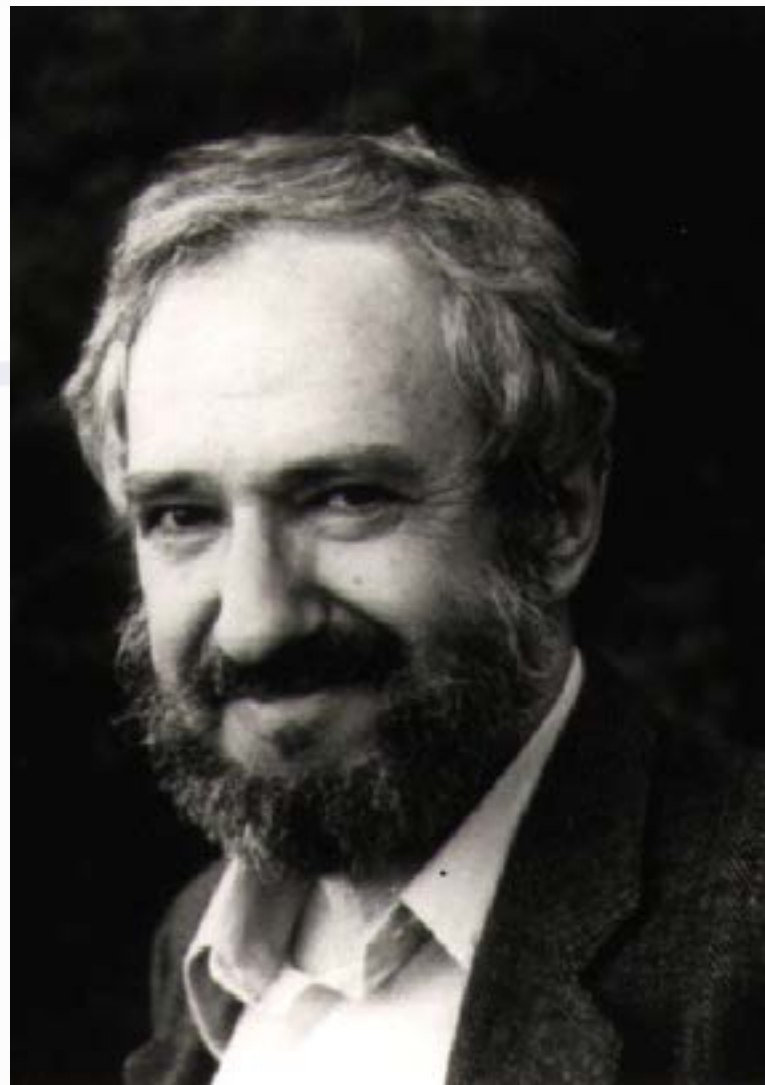


Contextualização

- Linguagem de programação Logo, 1967
- Criaram uma linguagem de programação voltada para crianças, jovens e adultos
- *Massachusetts Institute of Technology*, Estados Unidos



Wally Feurzeig



Seymour Papert

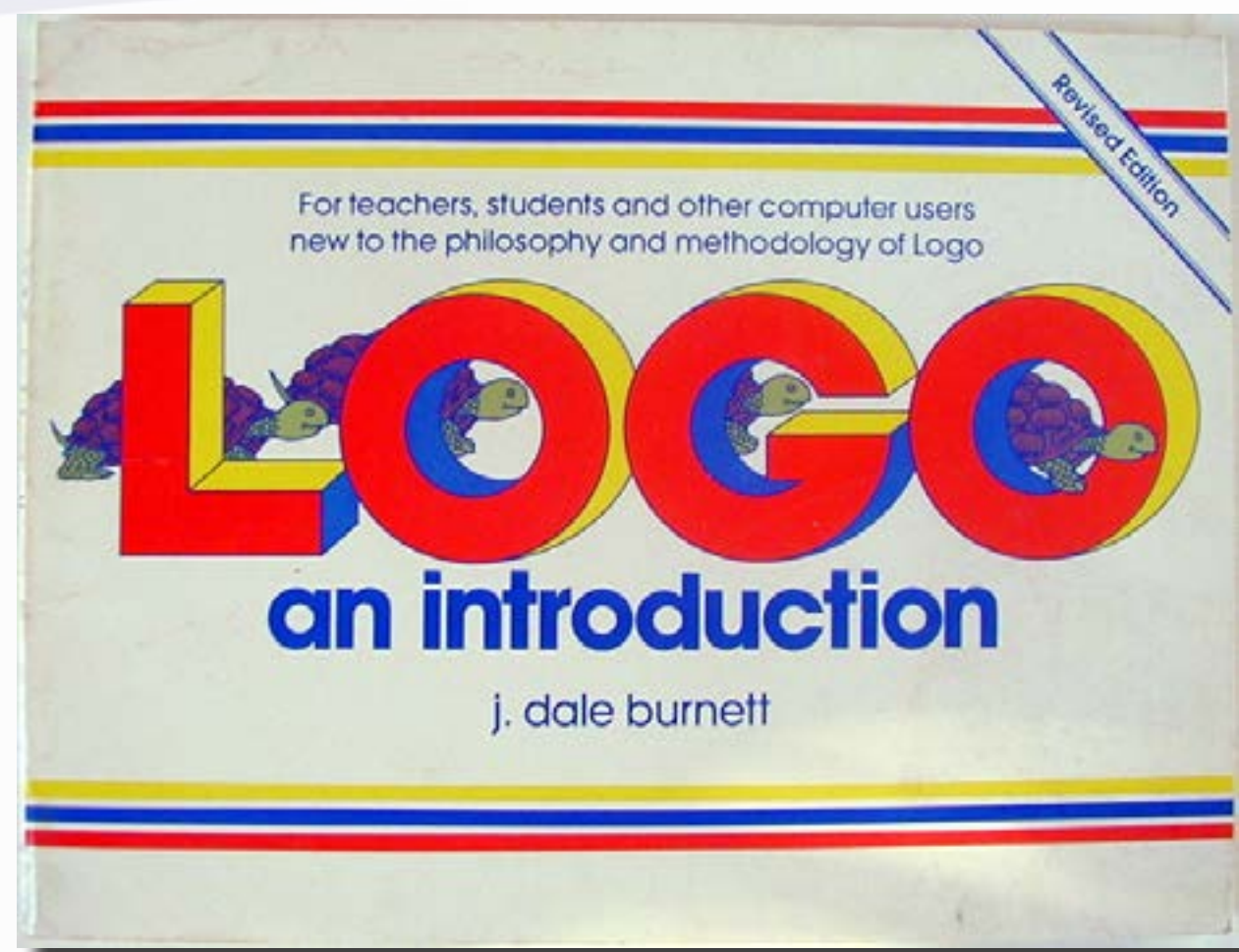


Cynthia Solomon

- O que é uma linguagem de programação?
- É a maneira que construímos programas de computador
- Os computadores não entendem a nossa linguagem, ela é complexa, ambígua
- Com uma linguagem de programação usamos comandos para dizer ao computador o que deve ser feito

Logo: Linguagem de Programação para Crianças

- Baseada na filosofia construtivista
- Logos, do grego: pensamento, ciência, raciocínio
- Anos 70: Começa a ser utilizada fora do laboratório
- Objetivo: criar desenhos ou programas



- O aluno é construtor de seus próprios conhecimentos, passos sucessivos
- O professor deve permitir a reflexão ao aluno, sem autoritarismo
- O aluno é estimulador a tentar e a aprender com seus próprios erros, refletindo sobre novas formas de resolver os problemas
- Observem a tartaruginha!!

Logo: Linguagem de Programação para Crianças

- Inspirada em um robô que havia no laboratório
- Cursor: Tartaruga com um lápis



Comandos:

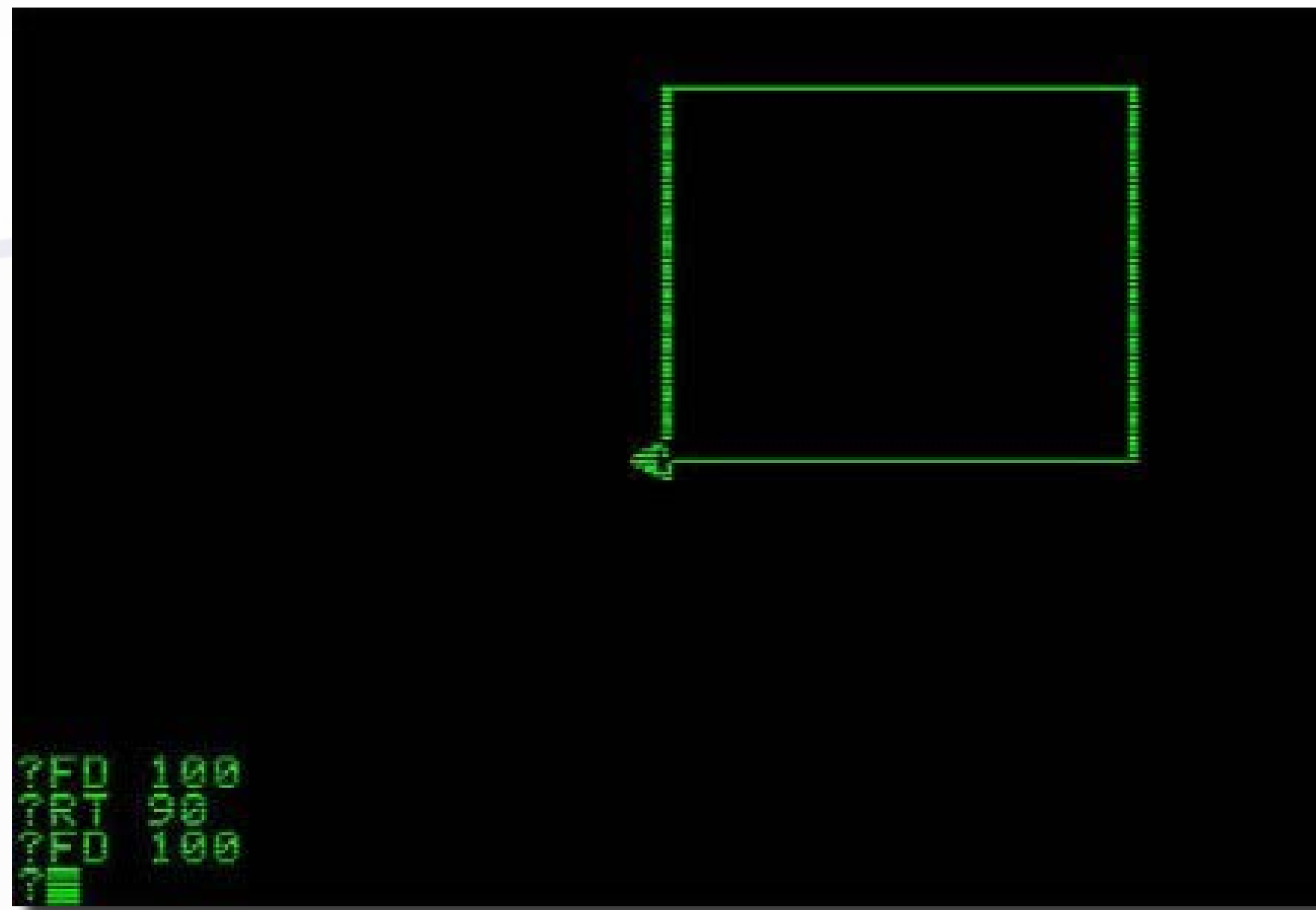
- Andar x
 - x é o número de passos
- Girar y
 - y é a quantidade de graus
- Abaixar ou levantar o lápis



- Essa linguagem de programação possui os comandos andar, girar, abaixar e levantar o lápis
- Este robô era caríssimo e só estava disponível em laboratório
- Poderia ser perigoso para crianças, pois possuía peças pequenas e risco de choque elétrico
- Poderia demandar supervisão constante por um adulto

Logo: Linguagem de Programação para Crianças

- Gráficos ainda eram limitados
- Tartaruga era representada por um triângulo
 - Metáfora para o robô anterior
- Criança era estimulada a construir figuras

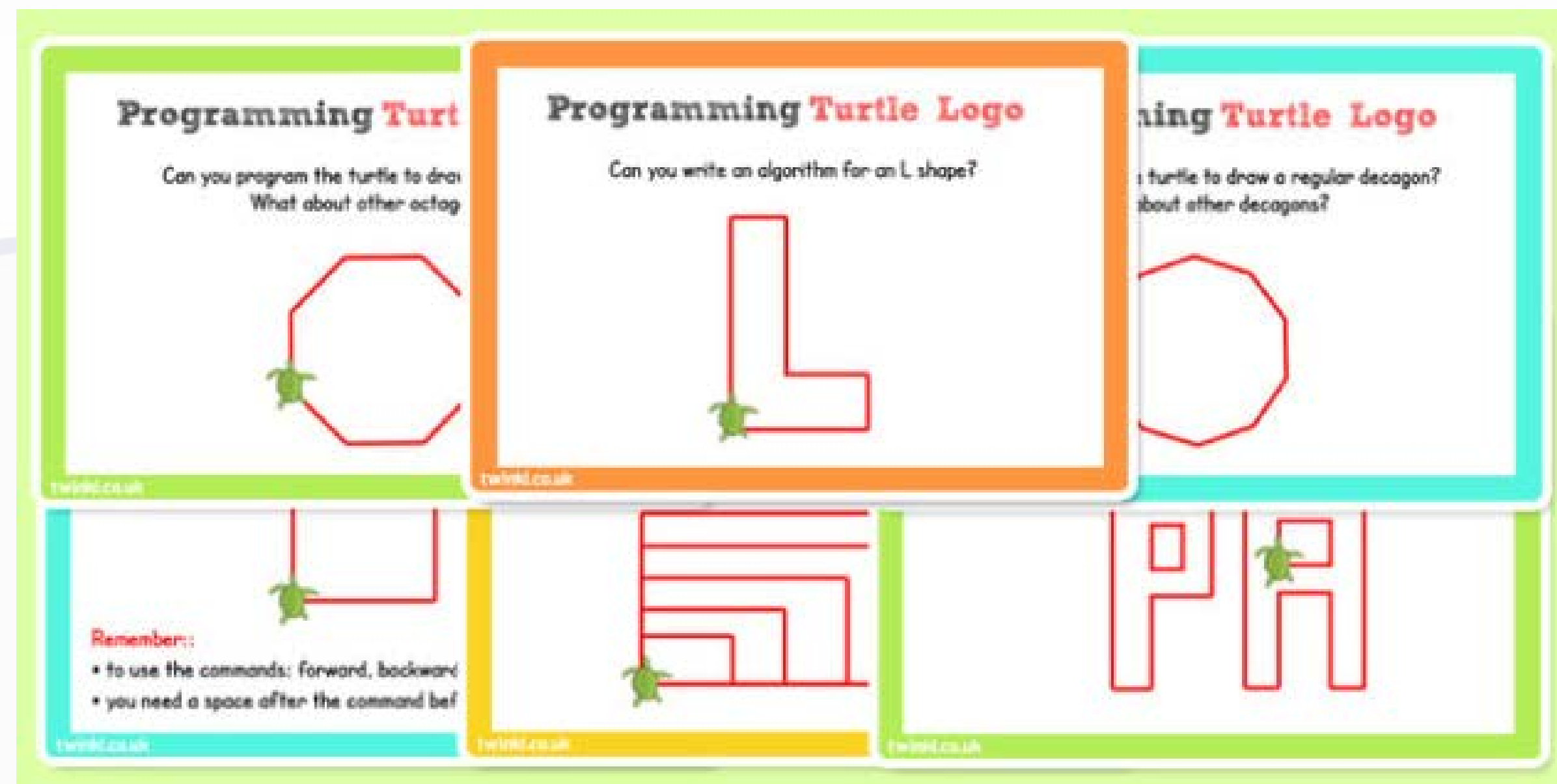


Logo: Linguagem de Programação para Crianças

- Exemplo: “Use a tartaruga e desenhe um quadrado”

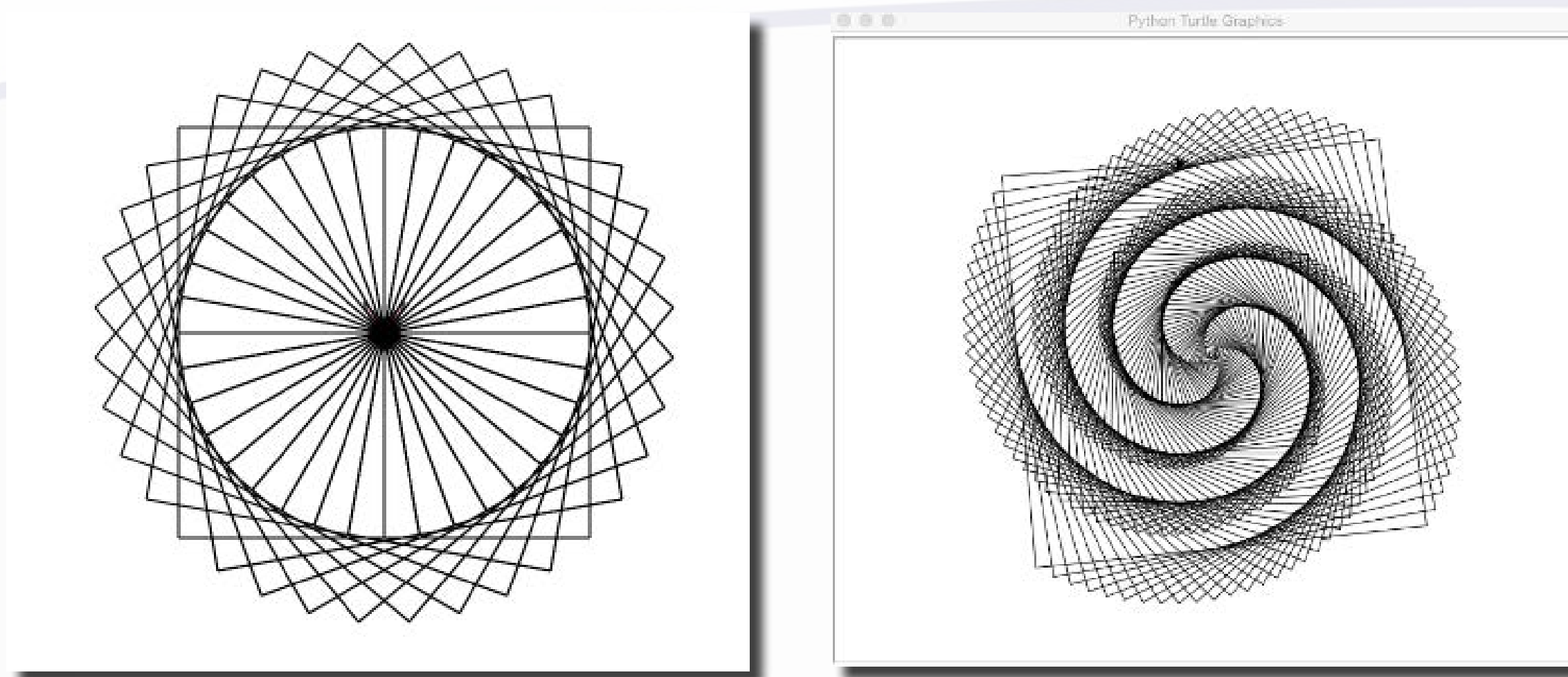
Logo: Linguagem de Programação para Crianças

- A linguagem possuía comandos simples
- Atividades com dificuldade crescente
- Oportunidade de trabalhar conceitos da geometria, por exemplo



Logo: Linguagem de Programação para Crianças

- A partir dos comandos simples, era possível criar figuras complexas
- Recebeu melhorias: uso de diversas cores, espessura da caneta
- Tida por especialistas de todo o mundo como o melhor e mais



Logo: Linguagem de Programação para Crianças

As principais características da linguagem LOGO são:

- Amigabilidade
- Modularidade
- Interatividade
- Flexibilidade
- Capacidade



- **Amigabilidade:** comandos são simples e fáceis de entender
- **Modularidade:** você pode criar seus próprios comandos. Por exemplo, poderíamos fazer um módulo chamado “quadrado” a partir da sequência de comandos que desenharam um quadrado
- **Interatividade:** o aluno constrói os comandos e logo em seguida executa, já podendo ver o resultado na tela
- **Flexibilidade:** pode ser usada com crianças e também com adultos na universidade
- **Capacidade:** Possui ferramentas necessárias para criar programas complexos, a exemplo de fractais

Crianças e Computadores

- A partir deste pontapé, muitas iniciativas para aproximar as crianças dos computadores foram criadas
- Muita ênfase no mercado de jogos
- Introdução de Robótica Educacional nas escolas
- Década de 90: pouca ênfase no ensino de programação para crianças no geral



- Mercado de jogos fazia com que as crianças passassem mais tempo no computador
- Jogos educacionais e que estimulavam o raciocínio: Xadrez, The Incredible Machine, Carmen Sandiego
- Outros jogos eram apenas lúdicos
- Lego introduziu uma metodologia de ensino de Robótica Educacional, que envolve material didático e kits de robótica, mas ainda é muito cara e pouco abrangente nos dias atuais
- Kits de Robótica Educacional da Lego fomentam competições de Robótica no mundo inteiro
- Na prática, poucos progressos para estimular as crianças a programarem e criarem conteúdos

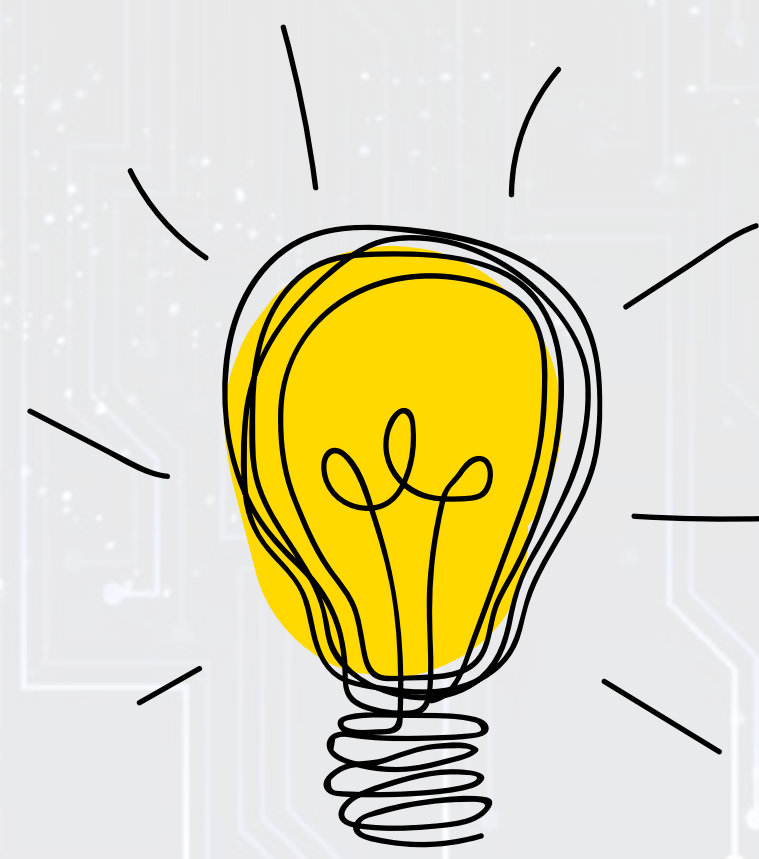
Vamos pensar um pouco:

“Por que é importante ensinar crianças a programar?”



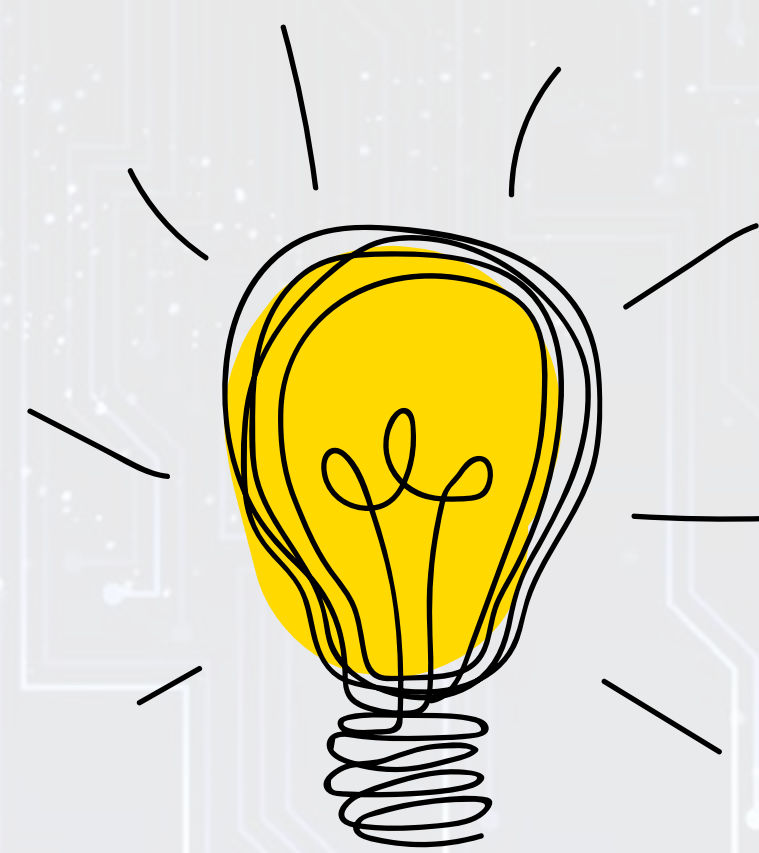
- Programar significar usar a linguagem de programação para dizer ao computador o que deve ser feito
- E estimular as crianças a programarem
- Não é um conteúdo exigido no ENEM
- Poucas pessoas seguem carreira na área de Ciências Exatas
- Nós não aprendemos a programar quando crianças e estamos bem hoje

Clareando as ideias!



- A escola prepara para a vida!
- Cada vez mais interagimos com dispositivos computacionais
- A maior fonte de informação hoje é a internet
- Programar ensina a pensar (Steve Jobs)
- Eles são nativos digitais

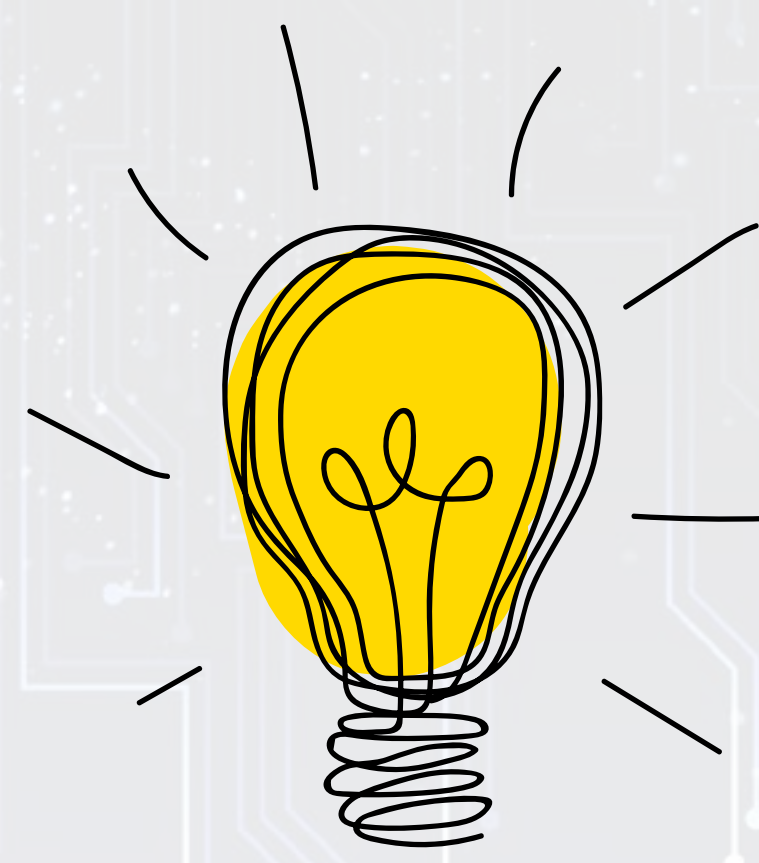
Clareando as ideias!



- Em termos acadêmicos:
- Matemática:
 - Facilita a visualização de conceitos abstratos
 - Aplicação de conceitos na vida prática
 - Torna a matemática divertida

- **Conceitos como ângulos, distância, funções, etc.**
- **O aluno está vendo ali na prática e utilizando o conceito**
- **Pode corrigir suas impressões e ir aprendendo, construindo noções mais sólidas**

Clareando as ideias!

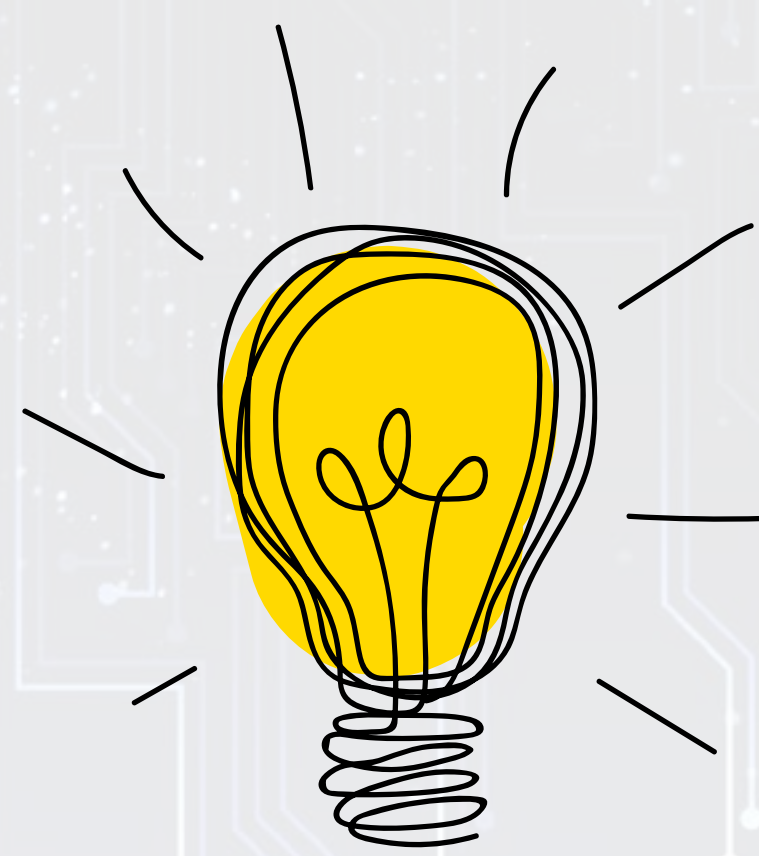


- Em termos acadêmicos:
- Matemática
- Escrita:
 - Aprende sobre ser conciso e a planejar a escrita
- Confiança:
 - Ganham confiança à medida que resolvem os problemas propostos

- **Programação não pode ter ambiguidades, ajuda a elaborar o que está sendo escrita**
- **Trabalha a precisão no que precisa ser comunicado, afinal o computador irá fazer exatamente aquilo para o qual foi programado**

Clareando as ideias!

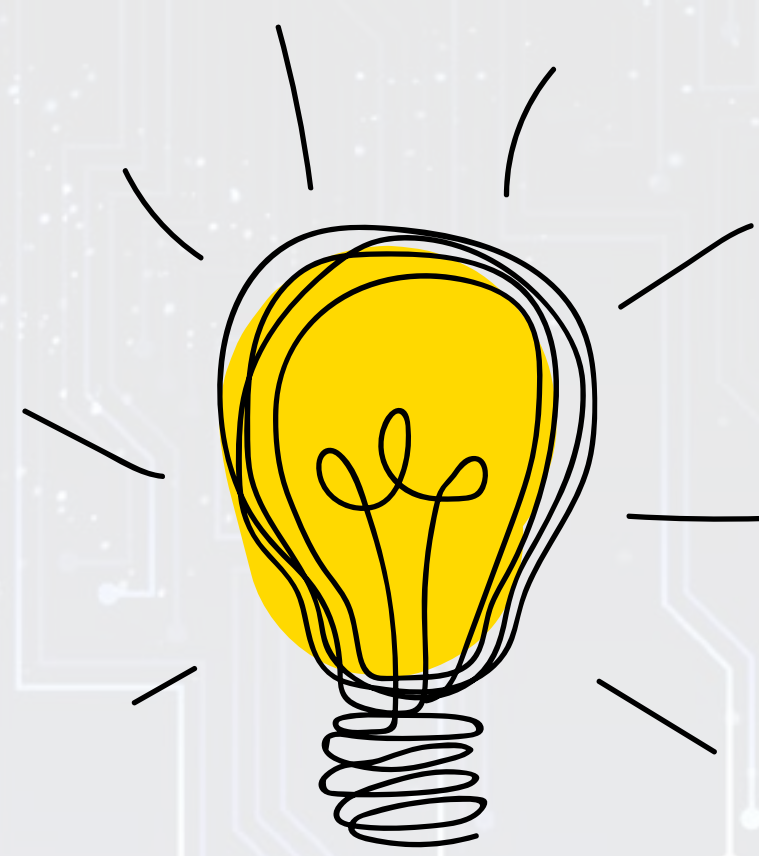
- Outras habilidades
- Foco e organização
- Resiliência
- Comunicação



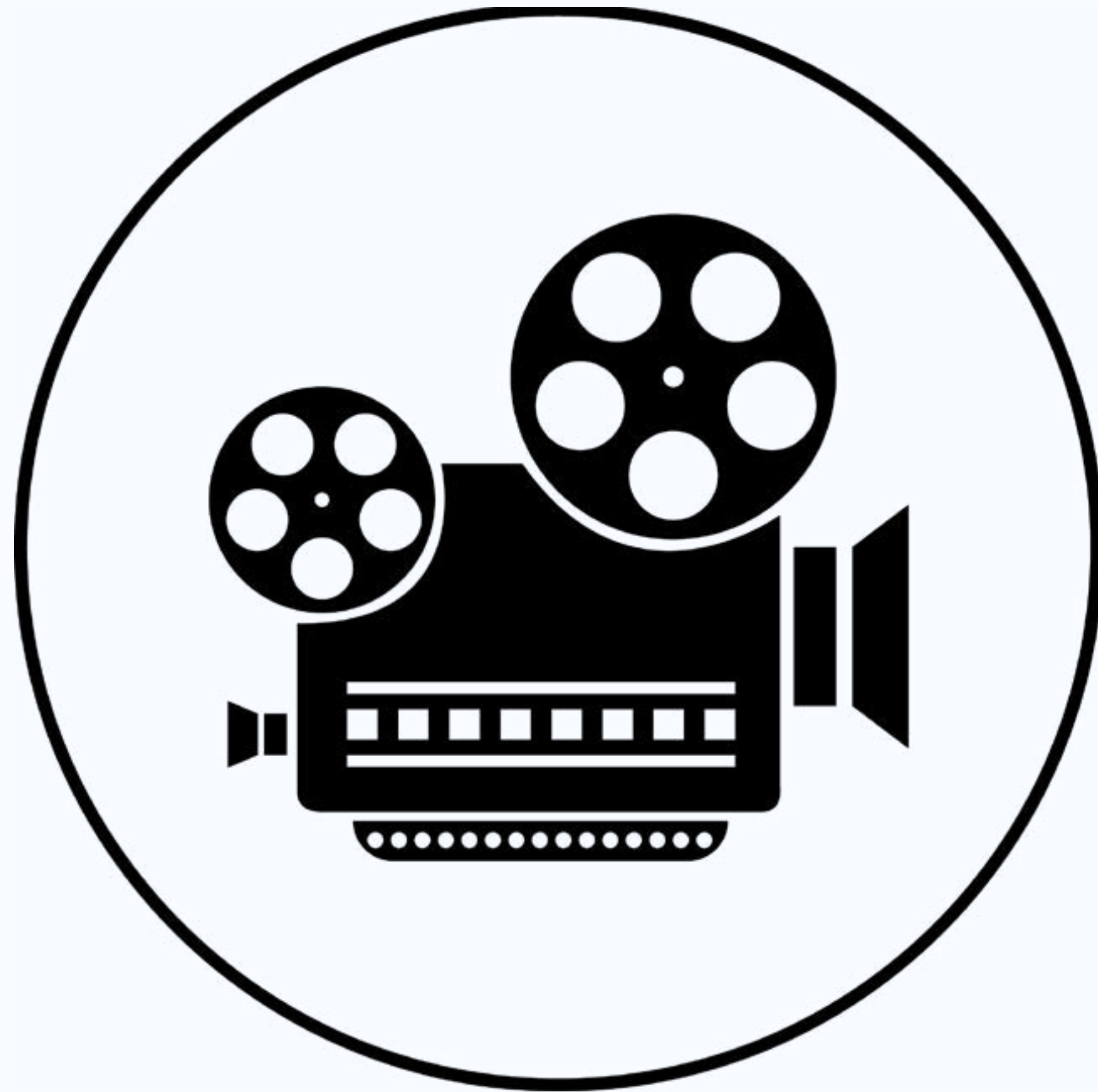
- À medida que o programa torna-se mais complexo, precisam se concentrar e organizar o que vai sendo produzido
- Programa macarrônico ou gelatinoso
- Resiliência tem a ver com persistir quando as coisas não vão bem: ter tolerância à frustração
- O código nem sempre vai funcionar, o aluno vai precisar perseverar para fazer dar certo!
- Considera o fortalecimento da comunicação verbal, pois vai precisar expressar ideias, dúvidas, e também a comunicação escrita

Clareando as ideias!

- Pensando no amanhã:
- Empoderamento
- Preparação para futuras carreiras
- Estímulo à criatividade
- Liberdade



- **As crianças se sentem capazes**
- **Muitas manifestações de tolerância e apoio entre eles**
- **Quem prosseguir nessa área, terá vantagens em ter aprendido a programar desde cedo**
- **Quem tem mais afinidade com matemática, mais ênfase neste aspecto**
- **E se gostar de geografia? Pode criar uma animação, um jogo**
- **Se gosta de carros, faz algo com carros**
- **Se gosta de borboletas, faz algo com borboletas**



Code Stars

Metodologia STEAM

O QUE É ESSE TAL DE



CIÊNCIA
(Science)

TECNOLOGIA
(Technology)

ENGENHARIA
(Engineering)

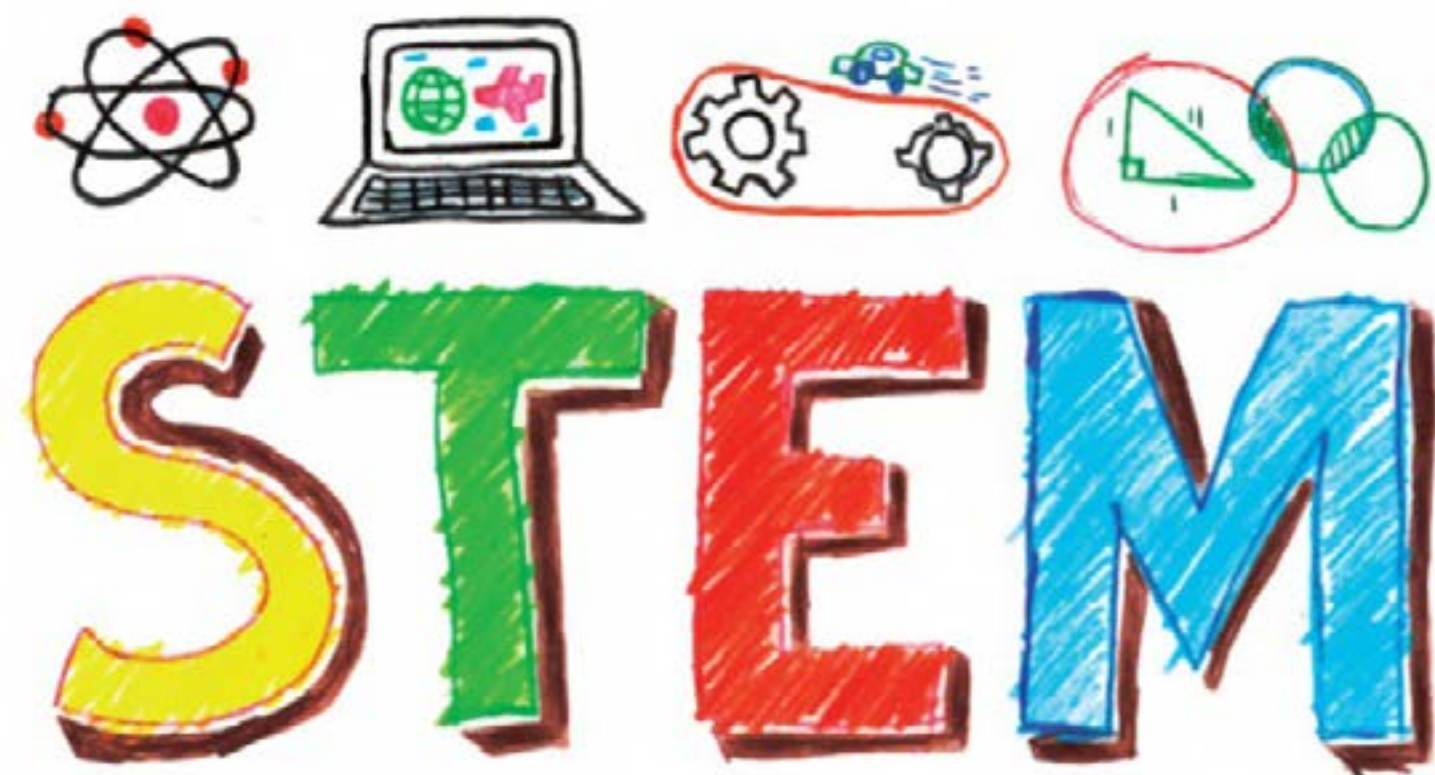
ARTES
(Arts)

MATEMÁTICA
(Mathematics)

- Explicar os elementos da sigla, oriundos da língua inglesa
- É uma metodologia para preparar alunos e cidadãos do futuro

Metodologia STEAM

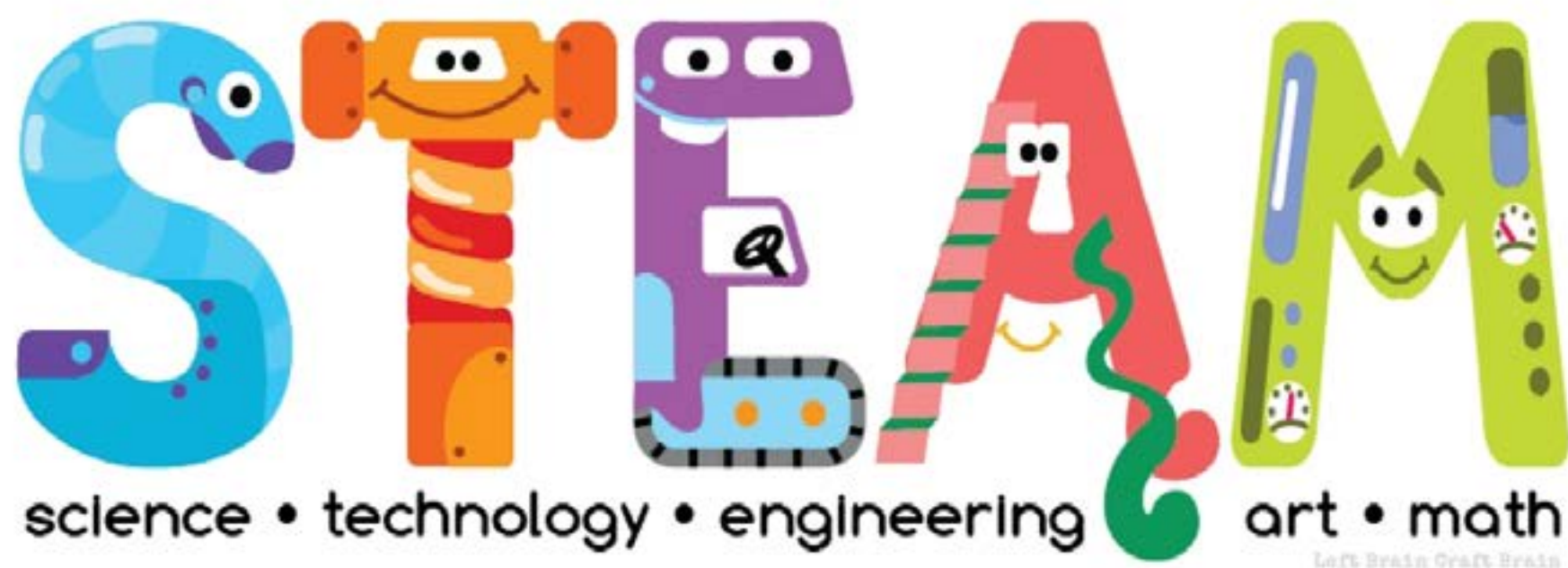
- Abordagem educacional
- Pode ser aplicada em diversos níveis escolares
- Baseada em projetos
- Formar pessoas com diversos conhecimentos e valores
- Ajuda a interpretar melhor o mundo



- Os conhecimentos não precisam mais ficar em compartimentos distintos
- Os grandes problemas do mundo hoje em dia dificilmente dizem respeito a uma só área. Exemplo: a barragem de Brumadinho é apenas uma questão da Engenharia? Envolve economia, pois trata do dinheiro da cidade, meio ambiente, como pensar em sustentabilidade, dentre outros.

Metodologia STEAM

- Projetos mão na massa para praticar algum conceito visto em sala
- Utilize problemas reais
- Integre e aplique os diferentes assuntos
- Encoraje o questionamento e a imaginação
- Investigar, Descobrir, Conectar, Criar e Refletir

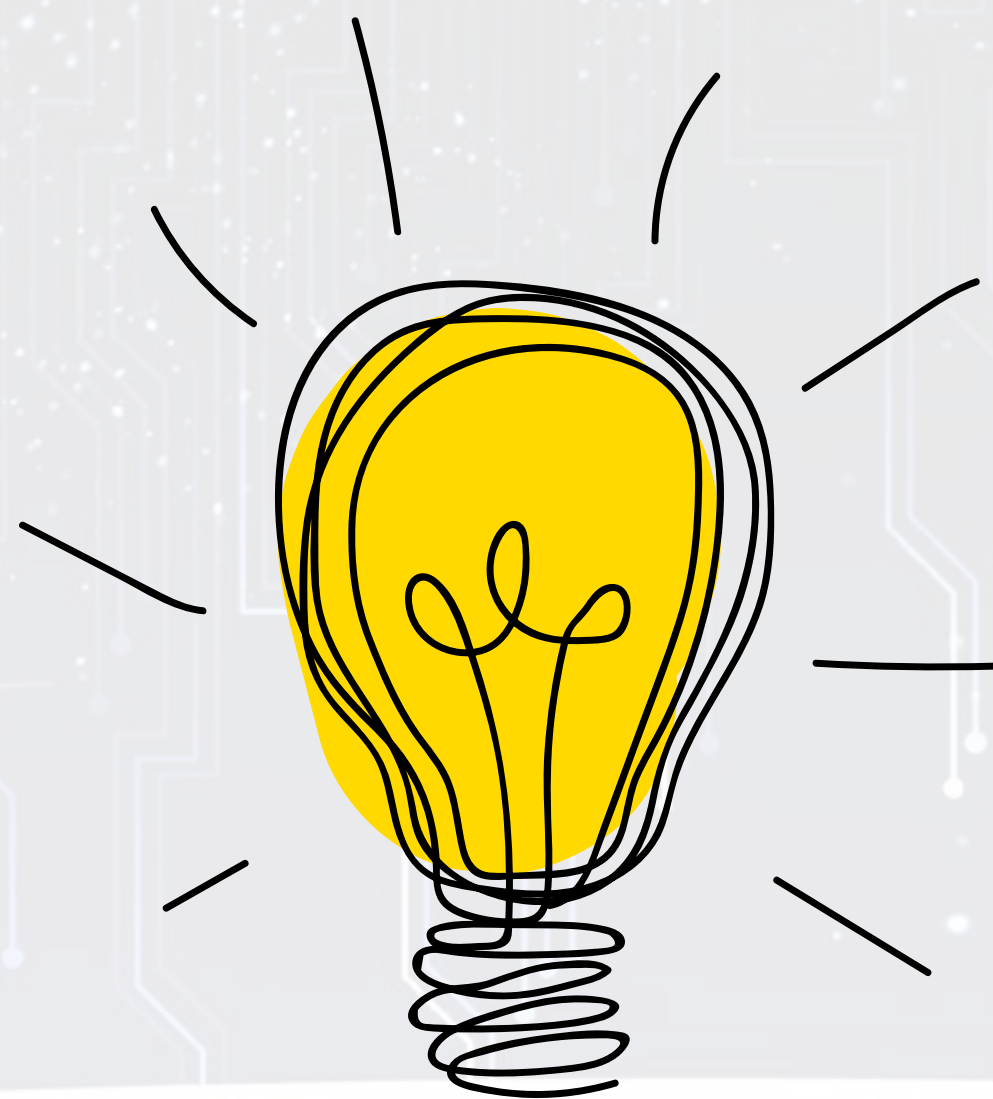


- Quem não lembra de ter plantado um pezinho de feijão na escola?
- Por exemplo: o que aconteceria se cavássemos um buraco de um lado a outro da terra? (Ensino de Ciências)
- Como usar a programação (Tecnologia) para criar músicas (arte)?
- Estimular a curiosidade para verificar se as crianças realmente estão entendendo
- Como construir um filme com massinhas de modelar que mostre a tabuada da multiplicação por 3?

Metodologia STEAM

Muitas ideias de projeto:

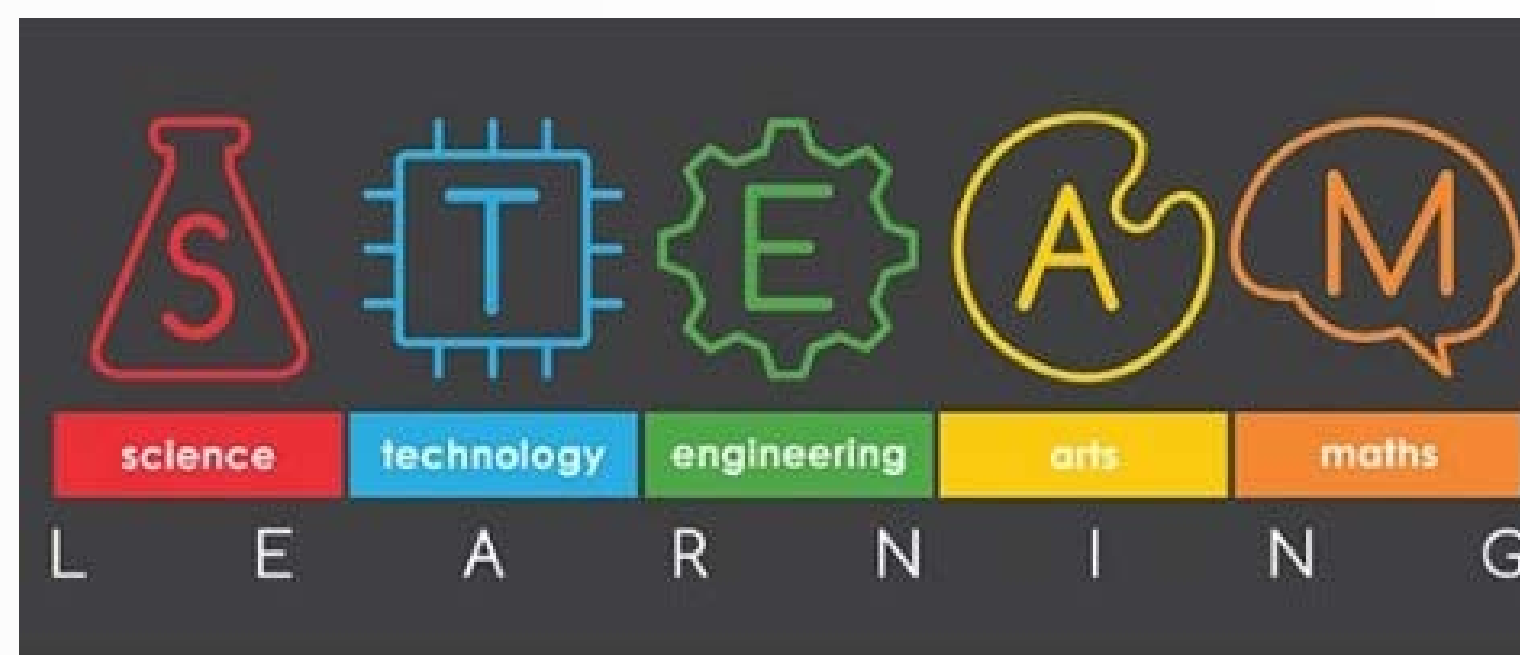
- Observar o crescimento de diferentes sementes
- Desafio do ovo de galinha
- Fabricar massa de modelar que muda de cor com o calor
- Fazer um carrinho robô controlado com programação



- **Ovo: Consistem em criar um dispositivo que amorteeça a queda de um ovo de galinha de determinada altura (altura, peso, velocidade, gravidade, força)**
- **Misturar ingredientes, ler quantidades, avaliar o resultado, entender o calor**
- **As ideias precisam ser compatíveis com o nível escolar da criança**

Programação no STEAM

- O papel da programação no STEAM é essencial
- Permitirá a criação de animações que ilustrem as ideias do projeto
- Controlar algum robô construído
- Exemplificar os conceitos matemáticos trabalhados
- Construir simulações



- Tecnologia é um dos componentes do STEAM
- A programação aparece dentro desse T que é tecnologia
- Pode vir acompanhado de Robótica também
- Muitas vezes, construir ou comprar dispositivos pode ser caro ou inviável
- Com programação, os alunos podem transmitir suas ideias
- Robô: robô resgate ou robô seguidor de linha
- Conceitos matemáticos: figuras geométricas
- Simulações: relacionadas à velocidade, gravidade, lançamento de objetos

Sintetizando o que vimos...

- Uma linguagem de programação é a maneira que utilizamos para programar um computador
- Programar um computador significa escrever as instruções do que ele deve executar
- LOGO foi a primeira linguagem de programação voltada para crianças
- Há diversos aspectos positivos em incluir programação na formação escolar
- A programação favorece o desenvolvimento de projetos STEAM



Reforçando Conceitos: Linguagem de Programação

- Nossa linguagem é muito rica: significados, palavras, ambiguidades, intenções, etc.
- Seria muito difícil construir uma máquina (computador) que capturasse todas essas sutilezas da comunicação
- Computadores não entendem a nossa linguagem
- Precisamos das linguagens de programação para dizer ao computador o que deve ser feito



Exemplo: “Ele sentou na cadeira e quebrou o braço” – o braço da cadeira ou o próprio braço?

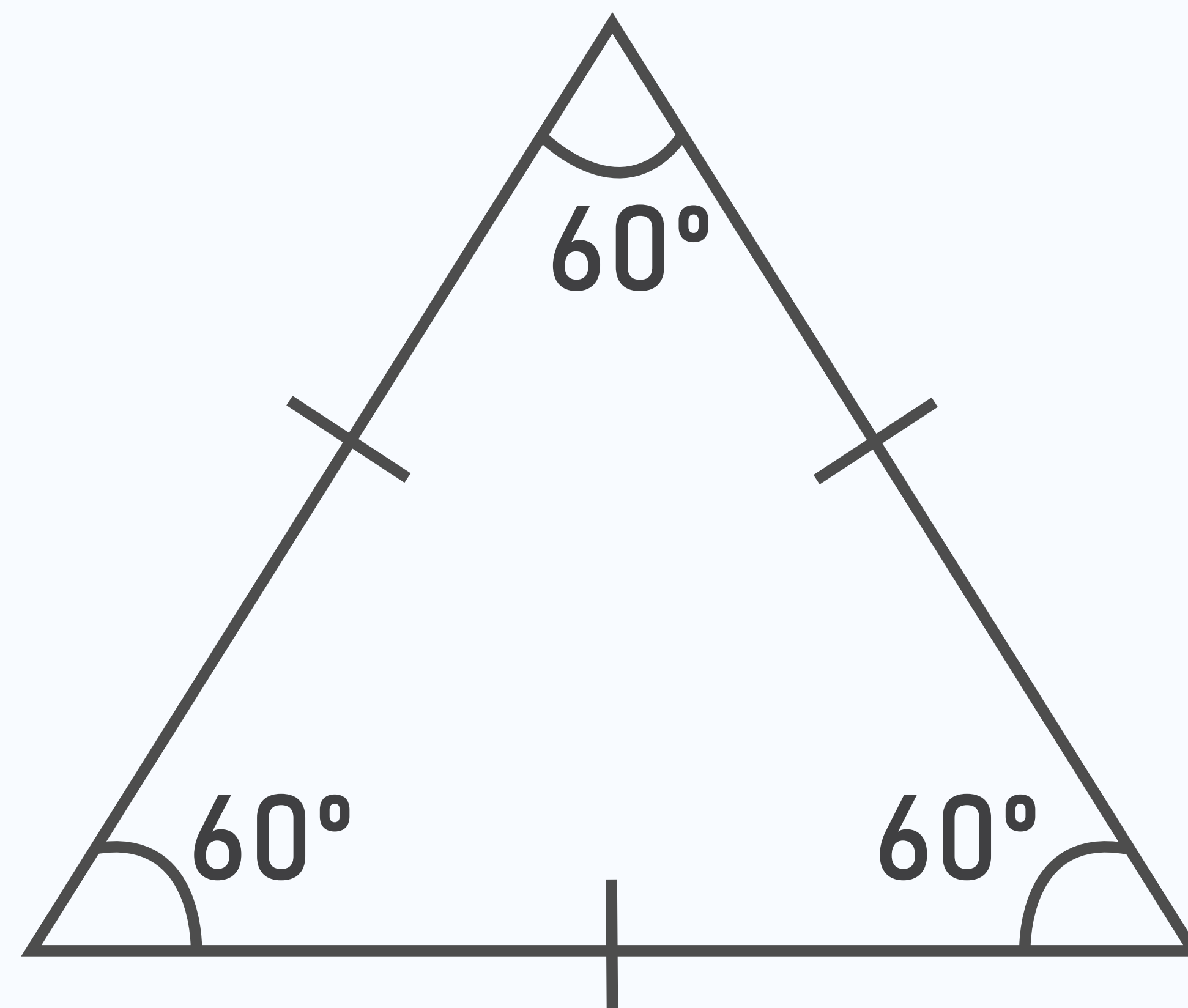
Reforçando conceitos: Linguagem de Programação

- A linguagem de programação traz consigo comandos básicos
- Para programar, utilizaremos esses comandos básicos
- Podemos criar comandos complexos a partir de vários comandos básicos



Exemplo com a Linguagem LOGO

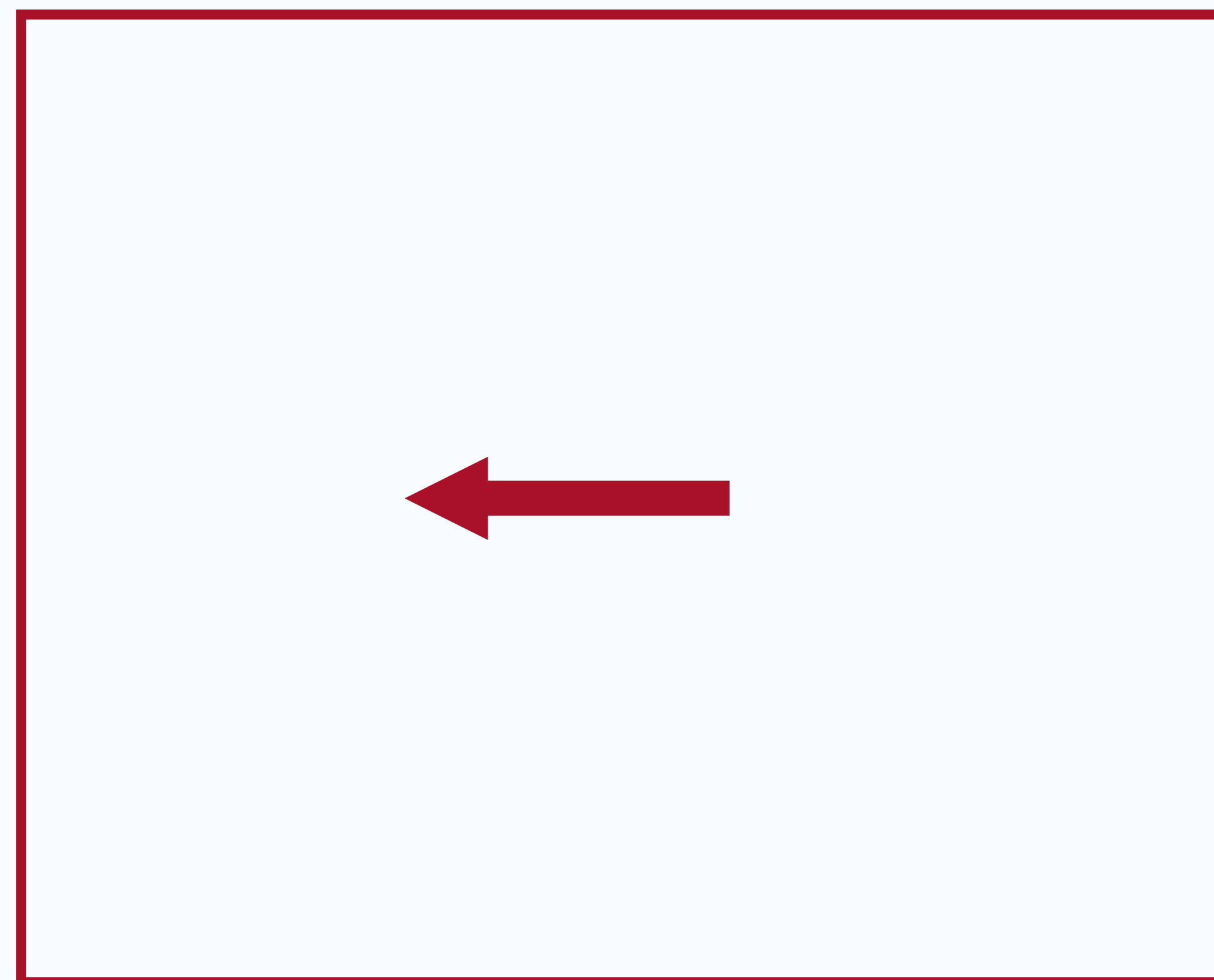
- Vamos ver um exemplo da Linguagem LOGO
- Os comandos mais simples são “andar” e “girar”
- Vamos construir um triângulo com esses dois comandos?
- Triângulo equilátero:
 - 3 lados iguais
 - Ângulos internos iguais a 60 graus



Exemplo com a Linguagem LOGO

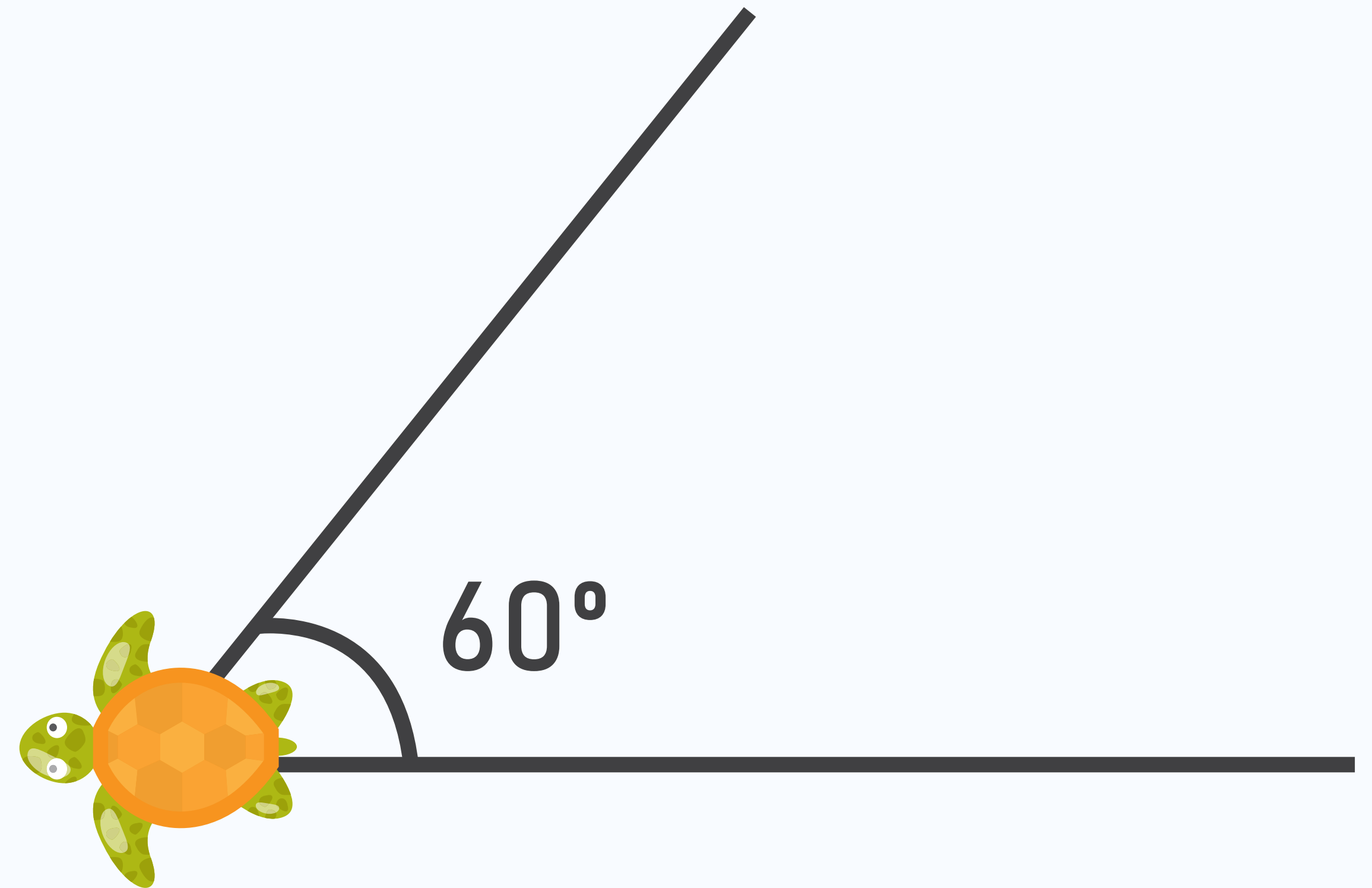
Primeiramente a tartaruga deve andar 10 passos

- Comando: **forward 10**



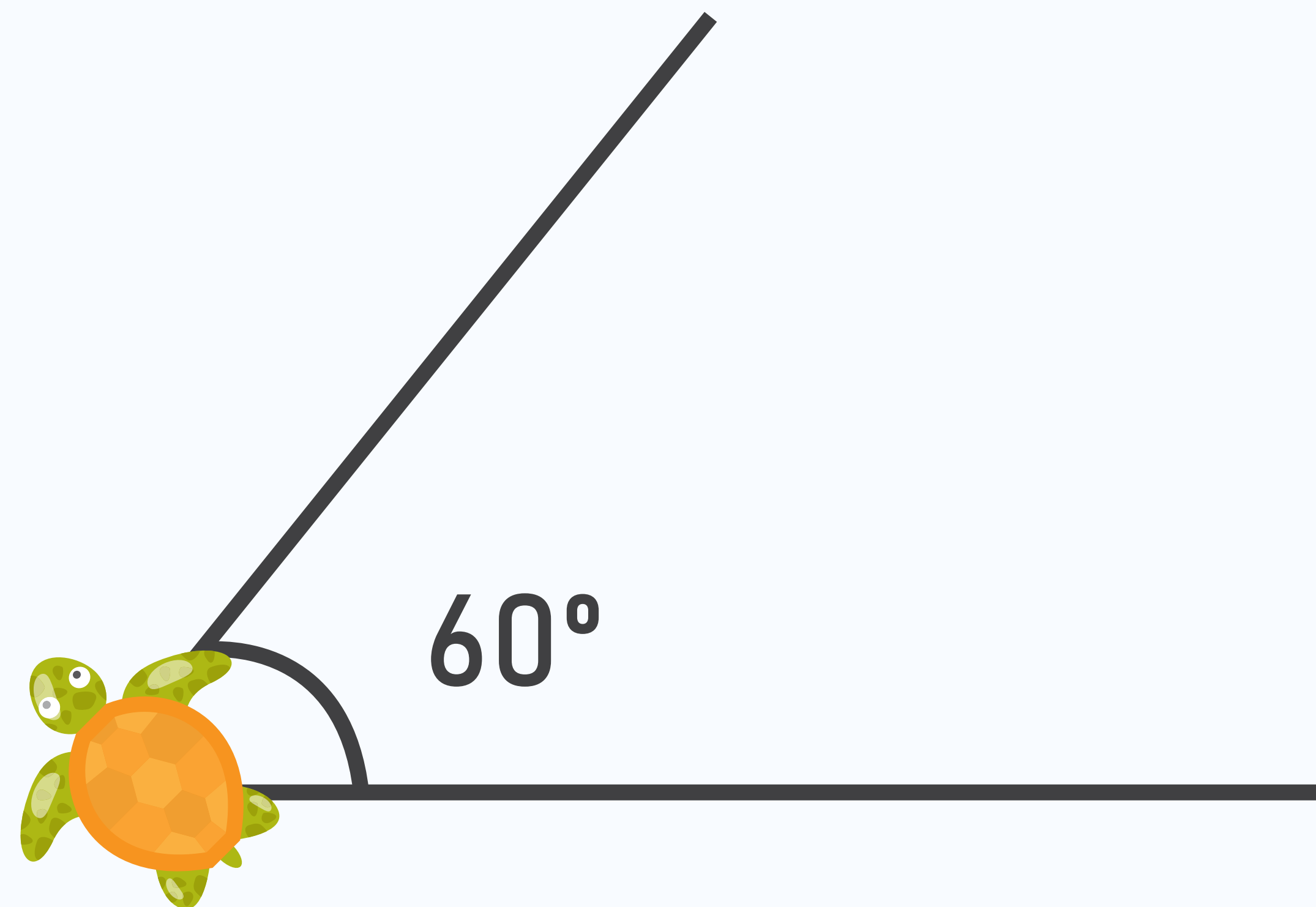
Exemplo com a Linguagem LOGO

- Em seguida, devemos desenhar o outro lado do triângulo
- Sabemos que o ângulo interno é de 60 graus



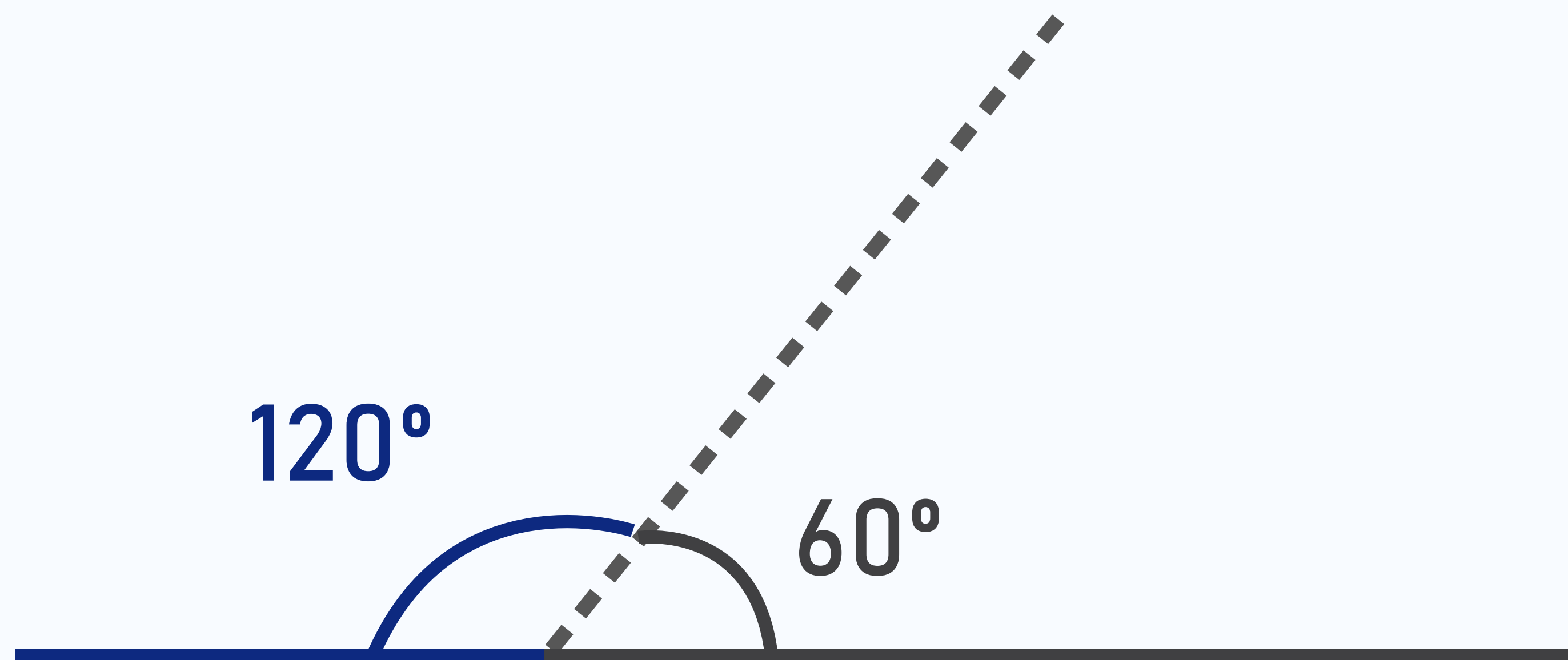
Exemplo com a Linguagem LOGO

- O que acontece se rotacionarmos a tartaruga 60 graus?
- É o efeito desejado?



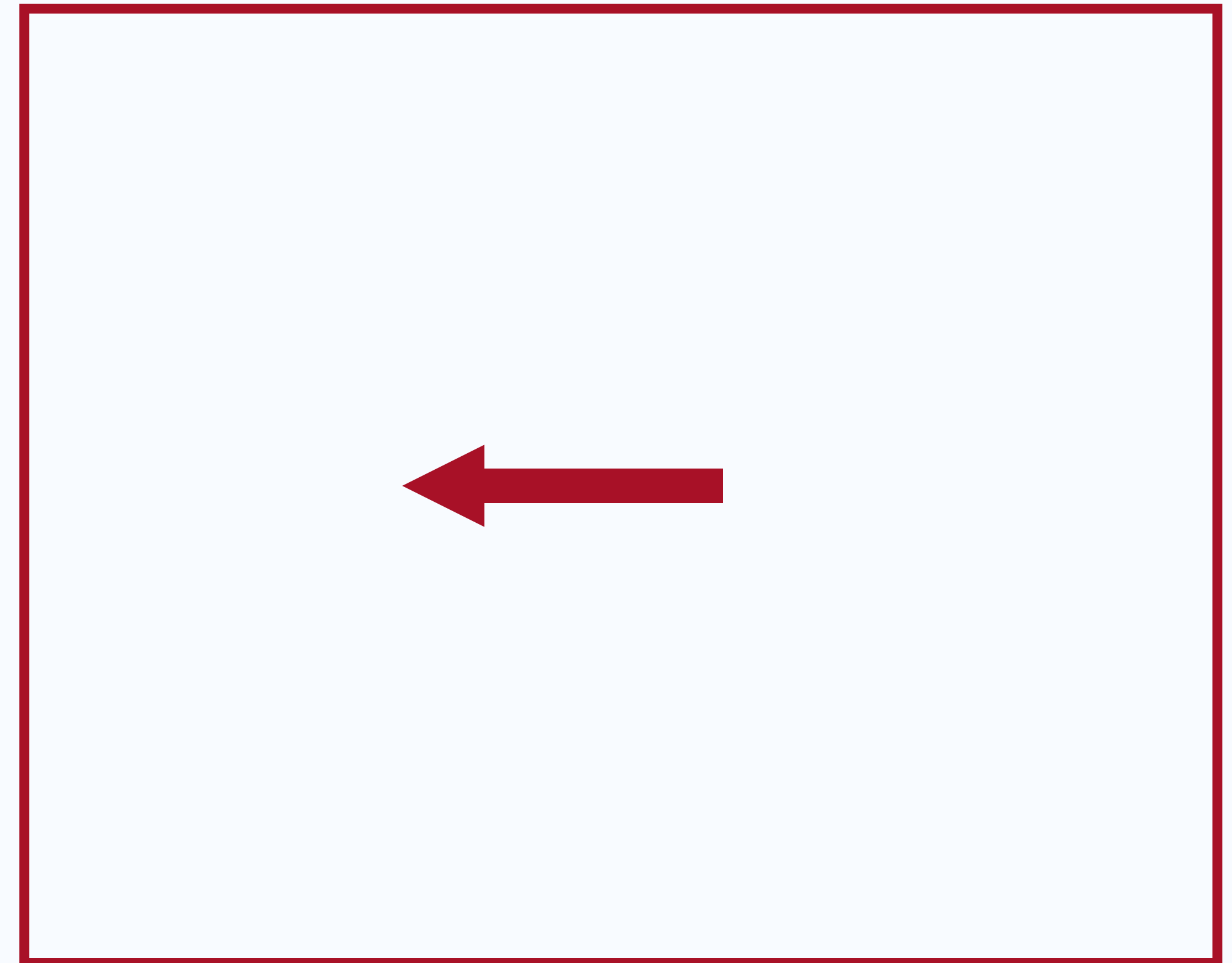
Exemplo com a Linguagem LOGO

- Precisamos olhar atentamente a relação entre os ângulos internos ao triângulo e externo
- A tentativa e erro do estudante vai guiando o processo



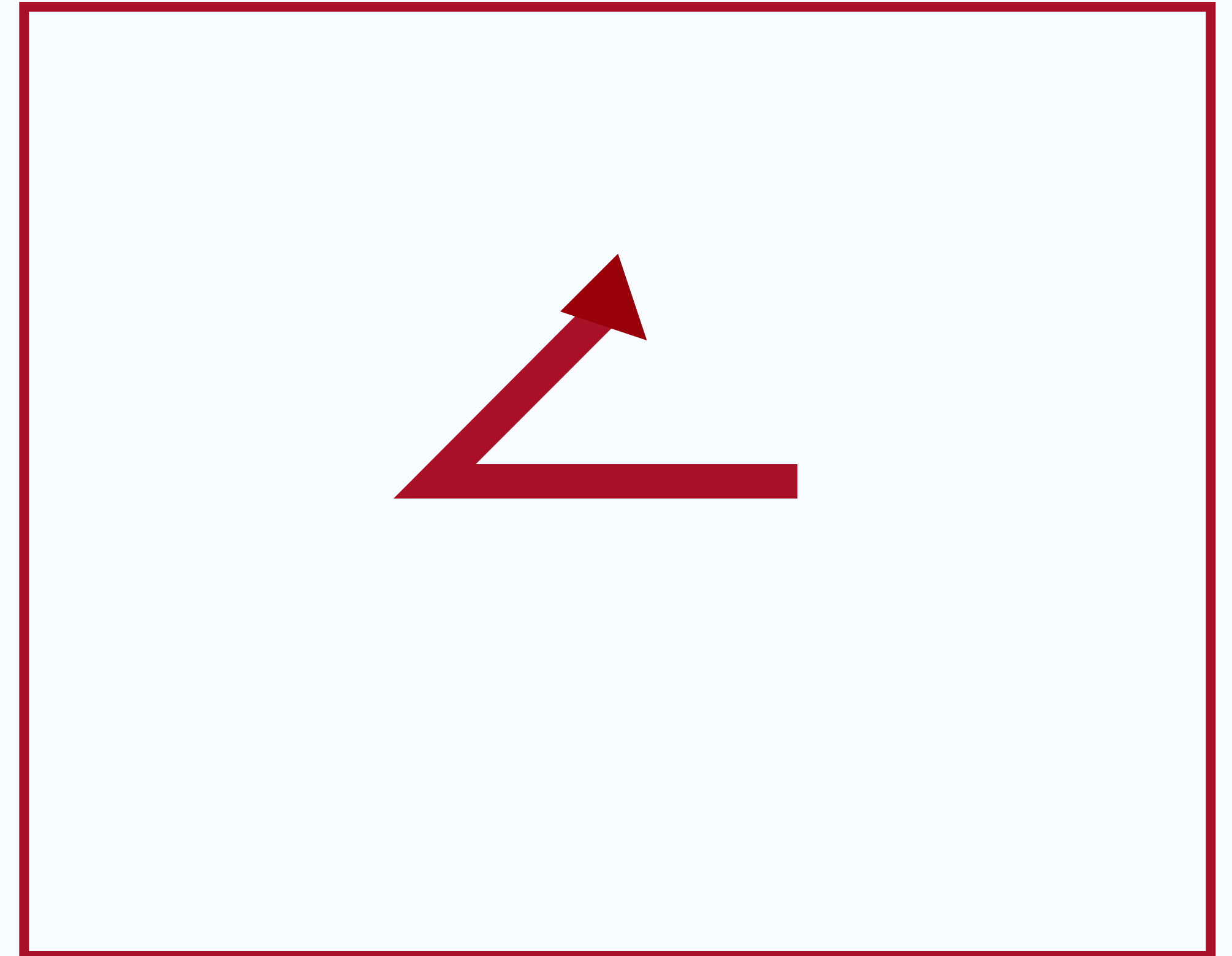
Exemplo com a Linguagem LOGO

- Comando: `right 120`



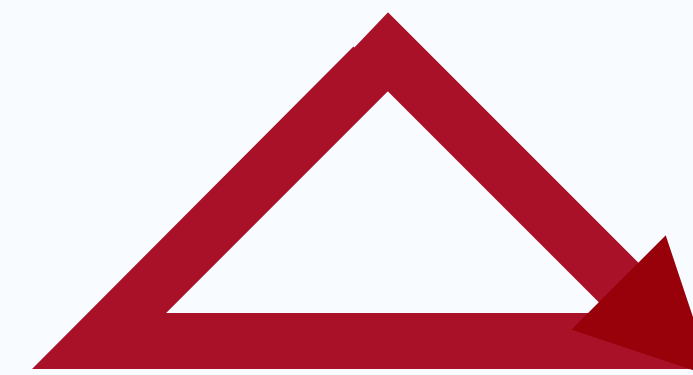
Exemplo com a Linguagem LOGO

- Comando:
 - forward 10
 - right 120



Exemplo com a Linguagem LOGO

- Por fim: `forward 10`

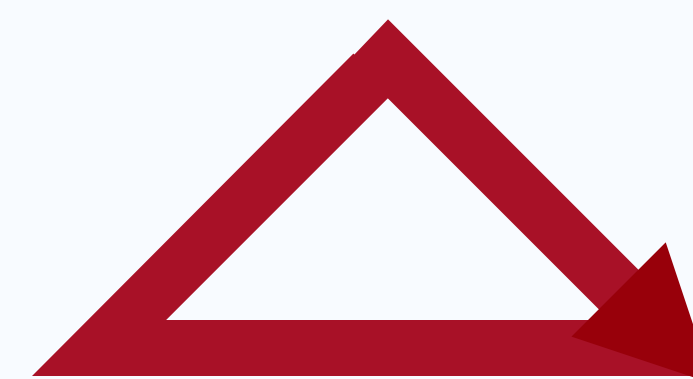


Exemplo com a Linguagem LOGO

- Comandos utilizados:
 - forward 10
 - right 120
 - forward 10
 - right 120
 - forward 10

Resultado da tartaruga

- Esses comandos poderiam ser agrupados nesta ordem
- Comando **triangulo**



Mão na massa!



Objetivos da manhã parcialmente cumpridos!

- Programação
- Linguagem de Programação
- Programação para Crianças



Atividade 01 – Praticando a Programação com LOGO

Entrega

Mesmo dia – Manhã

Quantidade de alunos
por equipe

No máximo 3

Duração da Atividade

1h

Descrição da Atividade

Consulte o Roteiro de Aprendizagem 01

Linguagem LOGO

- Primeira linguagem de programação voltada para crianças
- Possui um escopo limitado
- Pouca personalização, pouco atrativa
- Não dialoga com tecnologias dos dias atuais
 - Sons, imagens, etc.
- Não há interatividade



- Como uma linguagem de programação, possui elementos que são essenciais: os comandos, as repetições
- Trabalhamos a ideia de executar o código produzido
- Não consigo trabalhar conceitos de história ou português, por exemplo
- Crianças gostam de experimentar, de deixar sua marca individual naquilo que experimentam
- Pensem na possibilidade de usar sons, por exemplo
- O usuário não pode interagir com teclado e mouse, por exemplo

Linguagens de Programação Profissionais

- Objetivo: exibir a palavra "Olá!" na tela do computador

```
print("Olá!")
```

Linguagem Python

```
std::cout <<  
"Olá!" << std::  
endl;
```

C++

```
System.out.print-  
ln("Olá!");
```

Java

- Muitos aspectos técnicos
- Palavras em língua inglesa
- Pontuação excessiva
- Sintaxe e semânticas complexas
- Rotineiras na formação de Cientistas e Engenheiros da Computação

Novos desenvolvimentos

- 2003, MIT, Estados Unidos
- Projeto “Jardim de Infância ao longo da vida”
- Desenvolvimento de uma nova linguagem de programação para crianças: Scratch



Quais habilidades seriam importantes uma criança adquirir nos primeiros anos da escola e levarem para toda vida?

Scratch

- Linguagem de programação para crianças
- É uma linguagem de programação gráfica
- Traduzida para mais de 70 idiomas
- Utilizada no ensino de programação em todo o mundo
- Em seu site oficial, há mais de 46 milhões de projetos compartilhados



SCRATCH

- No Logo, as instruções dadas à tartaruginha eram feitas sob a forma de comandos textuais, ou seja, era preciso escrever as instruções a que ela deveria obedecer
- No Scratch, essas instruções são passadas de maneira visual, numa ideia similar ao encaixe de brinquedos de bloquinho

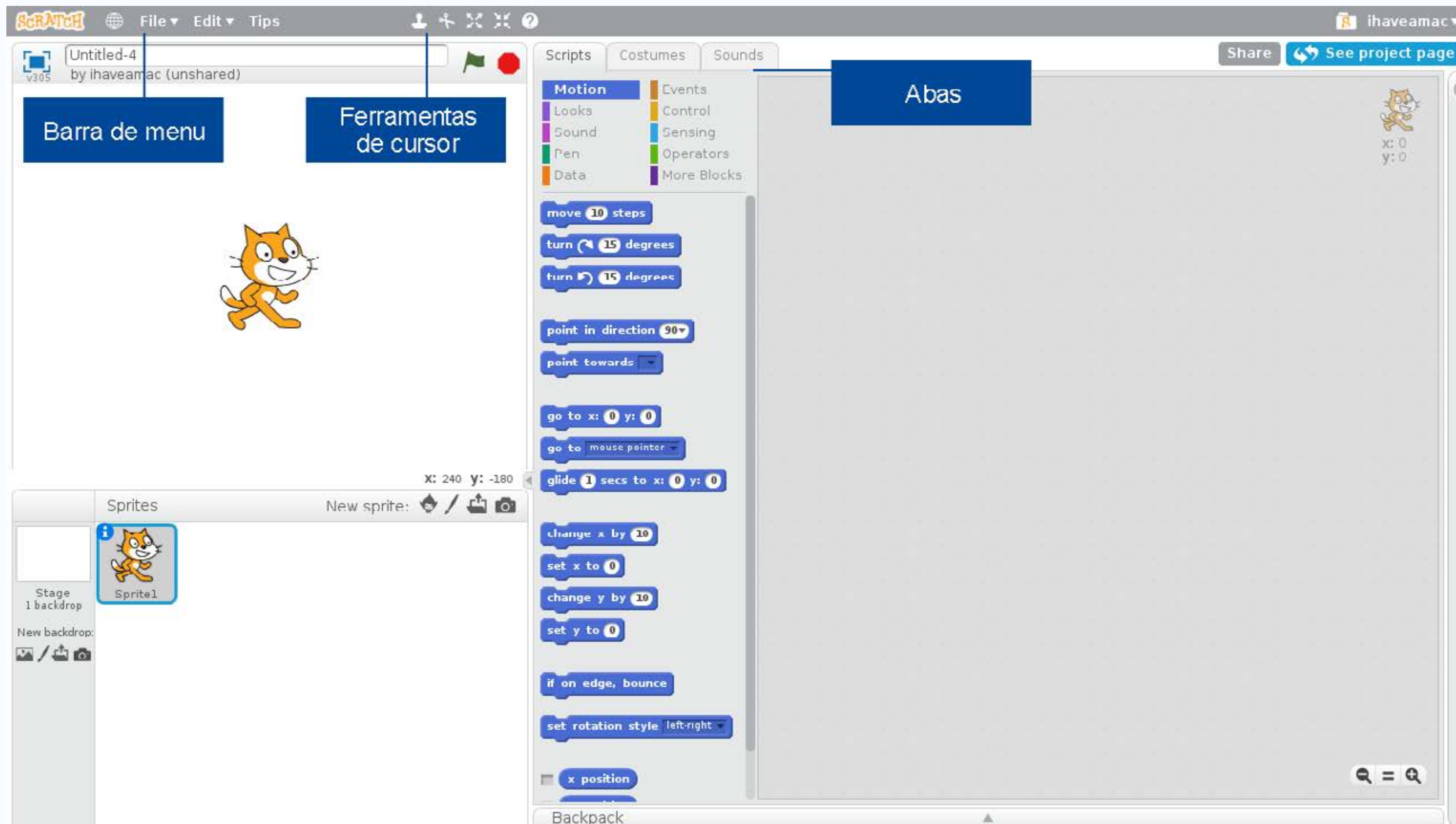
Scratch

Vantagens:

- Não é necessário digitar longos comandos complexos
- Grátis
- Criar, visualizar ou adaptar projetos
- Compartilhar projetos



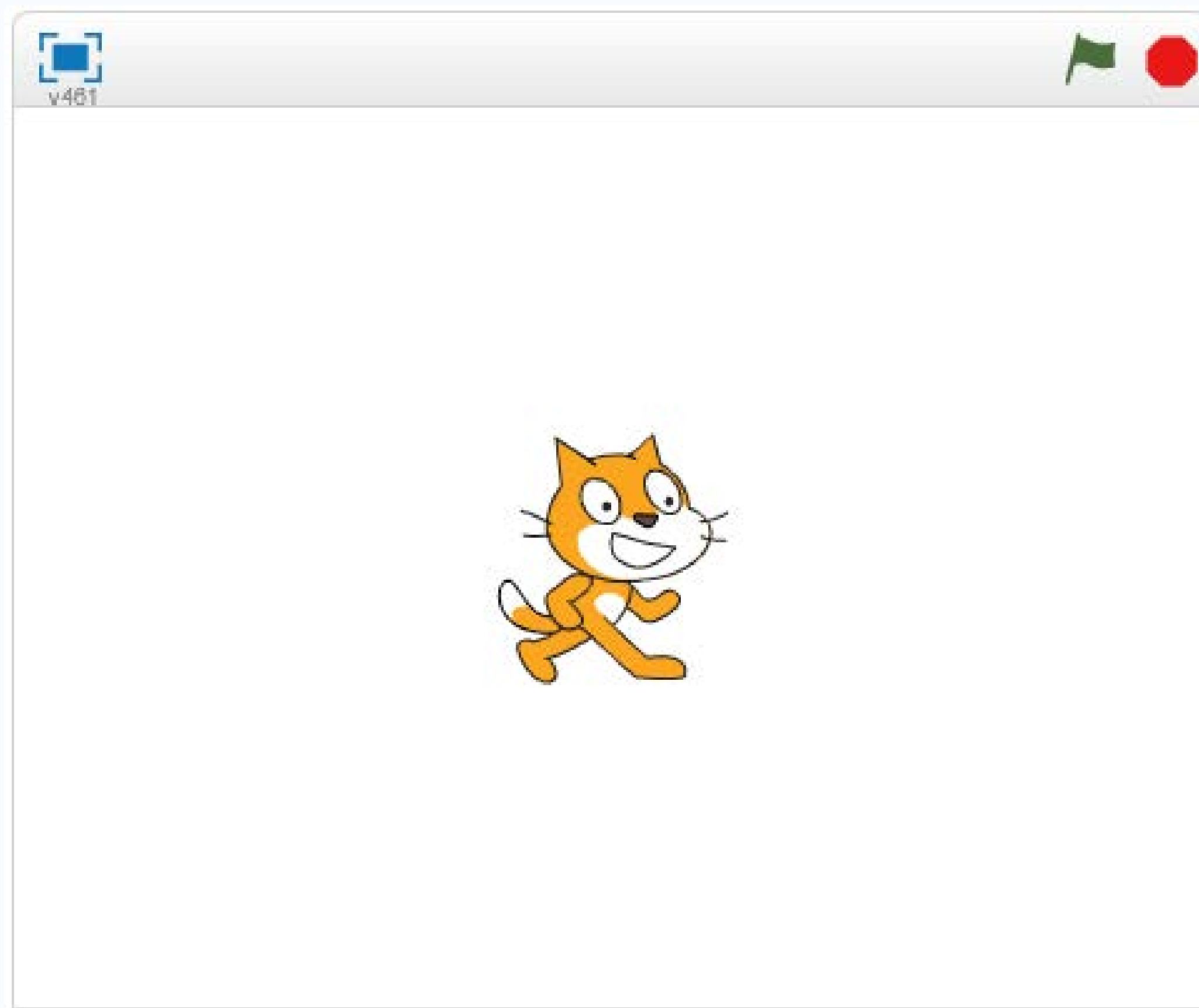
Scratch: Conhecendo o Ambiente de Programação



- Na Barra de Menu, vamos encontrar as opções tradicionais de salvar, abrir, fechar o programa, copiar e colar. Nesse globo ao lado, vocês têm a opção de escolher o idioma. Se o seu não estiver em português, pode clicar para ver esta opção.
- As ferramentas de cursor auxiliam a aumentar ou diminuir o zoom, colar e cortar objetos.
- Temos aqui outros elementos: o palco, a lista de sprites, a aba scripts e a área de scripts

Scratch: Conhecendo o Ambiente de Programação

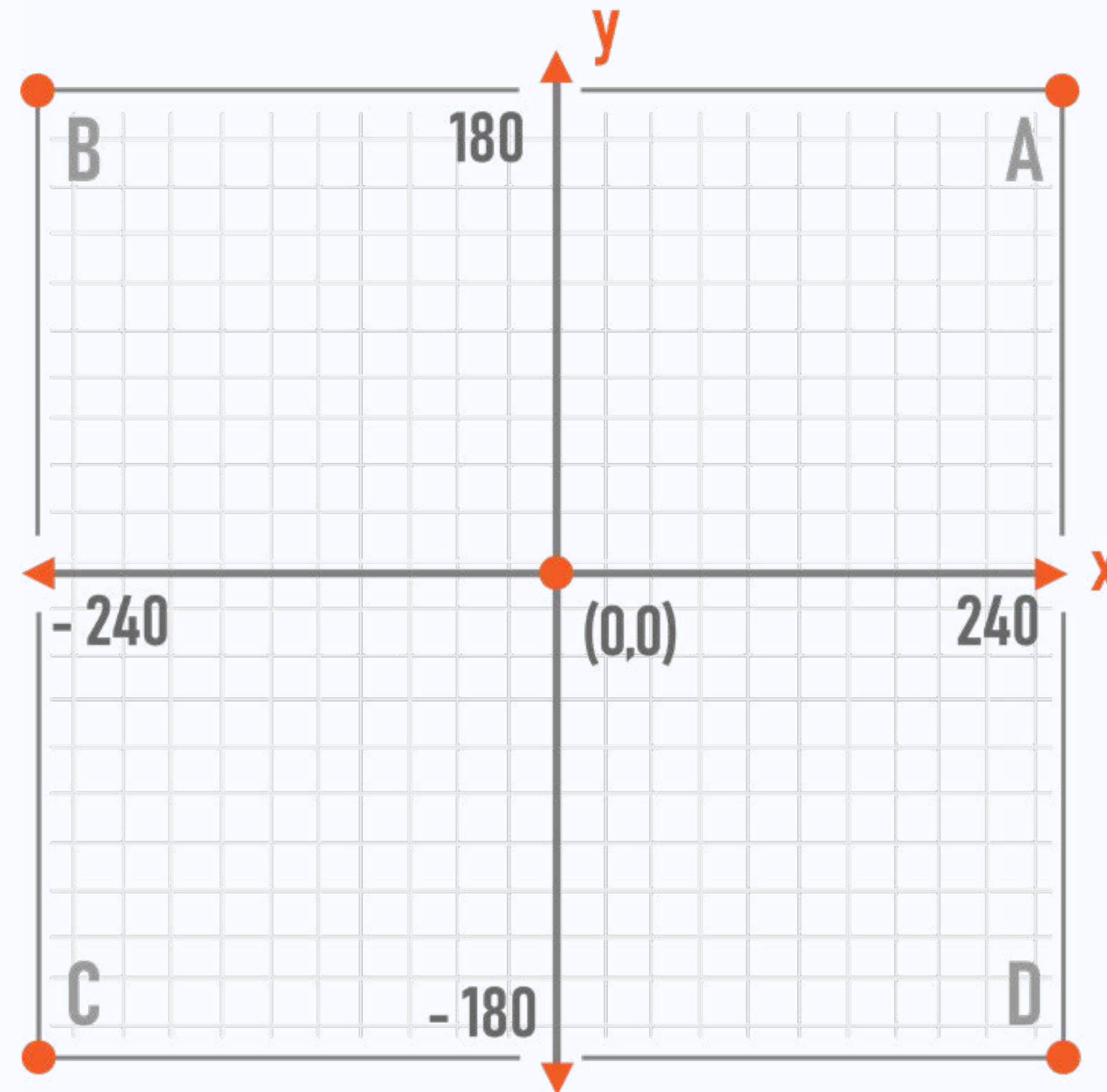
Palco: É o local onde os sprites se movem



- Palco em tela cheia
- Bandeira verde: inicia o programa
- Bandeira vermelha: encerra o programa

Scratch: Conhecendo o Ambiente de Programação

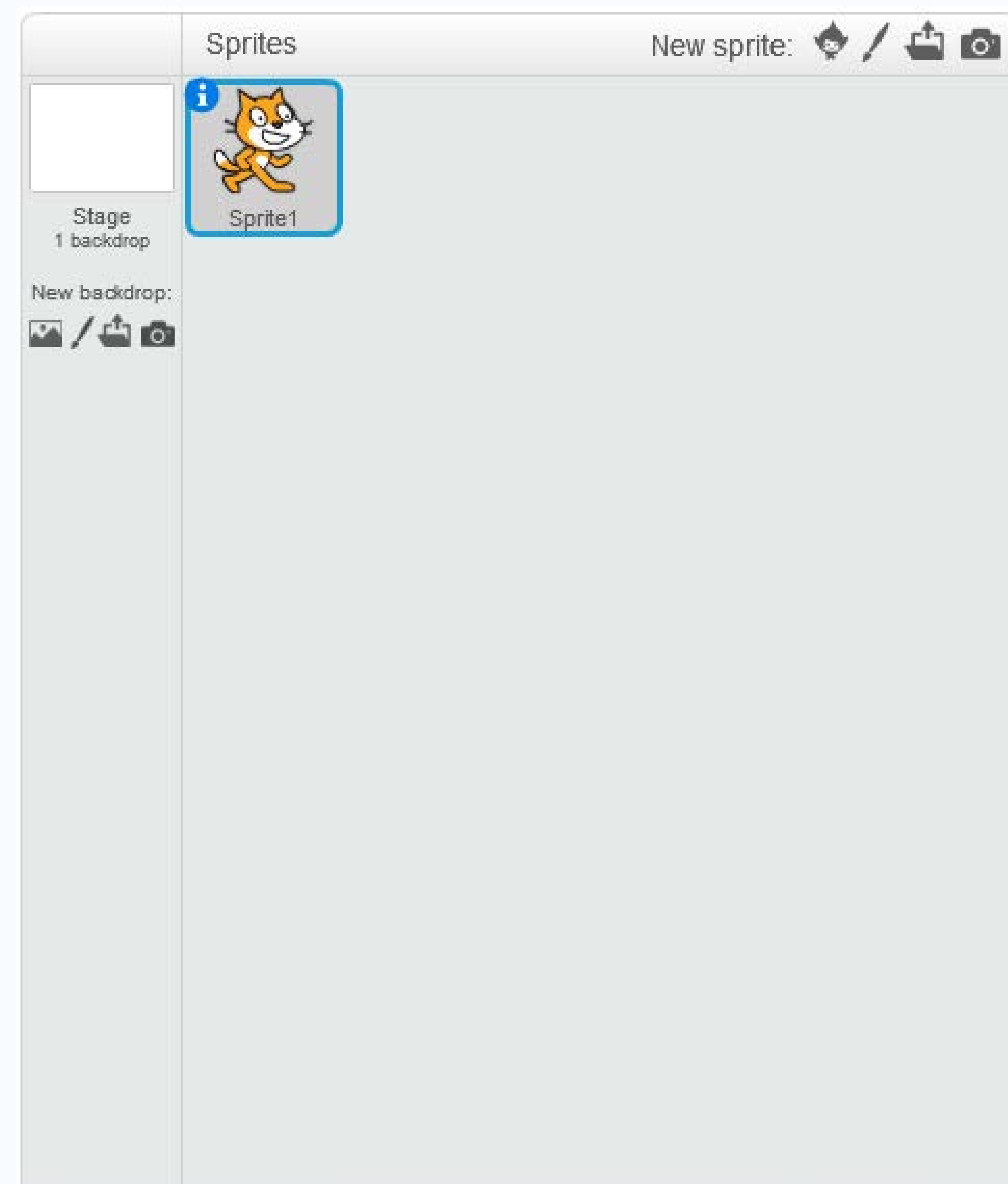
Palco



- 480 passos de largura
- 360 passos de altura
- O centro é a coordenada (0,0)

Scratch: Conhecendo o Ambiente de Programação

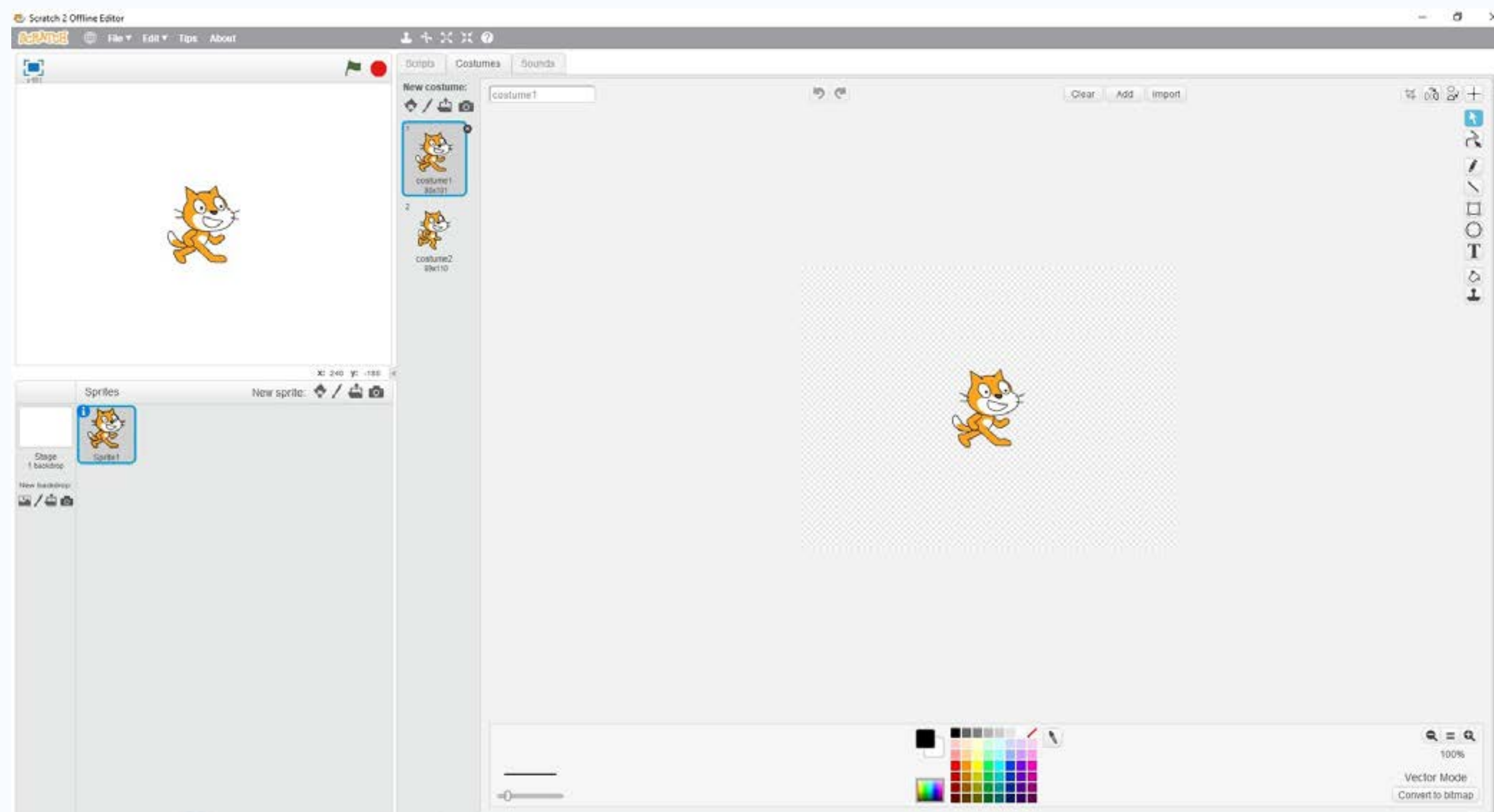
Lista de Sprites: exibe os nomes e as miniaturas de todos os sprites do seu projeto



- O Sprite é como se fossem nossos personagens. Um mesmo personagem pode ter várias fantasias. No nosso caso, poderemos adicionar fantasias ao nosso Sprite.
- Botão escolher sprite do acervo (biblioteca com vários sprites, bolas, meninos, meninas, animais, etc.)
- Pintar novo sprite
- Usar Sprite a partir do arquivo (imagem oriunda da internet)
- Ou até mesmo capturar um novo Sprite a partir da câmera
- Inicialmente, só temos o Sprite do gatinho e o palco está em branco

Scratch: Conhecendo o Ambiente de Programação

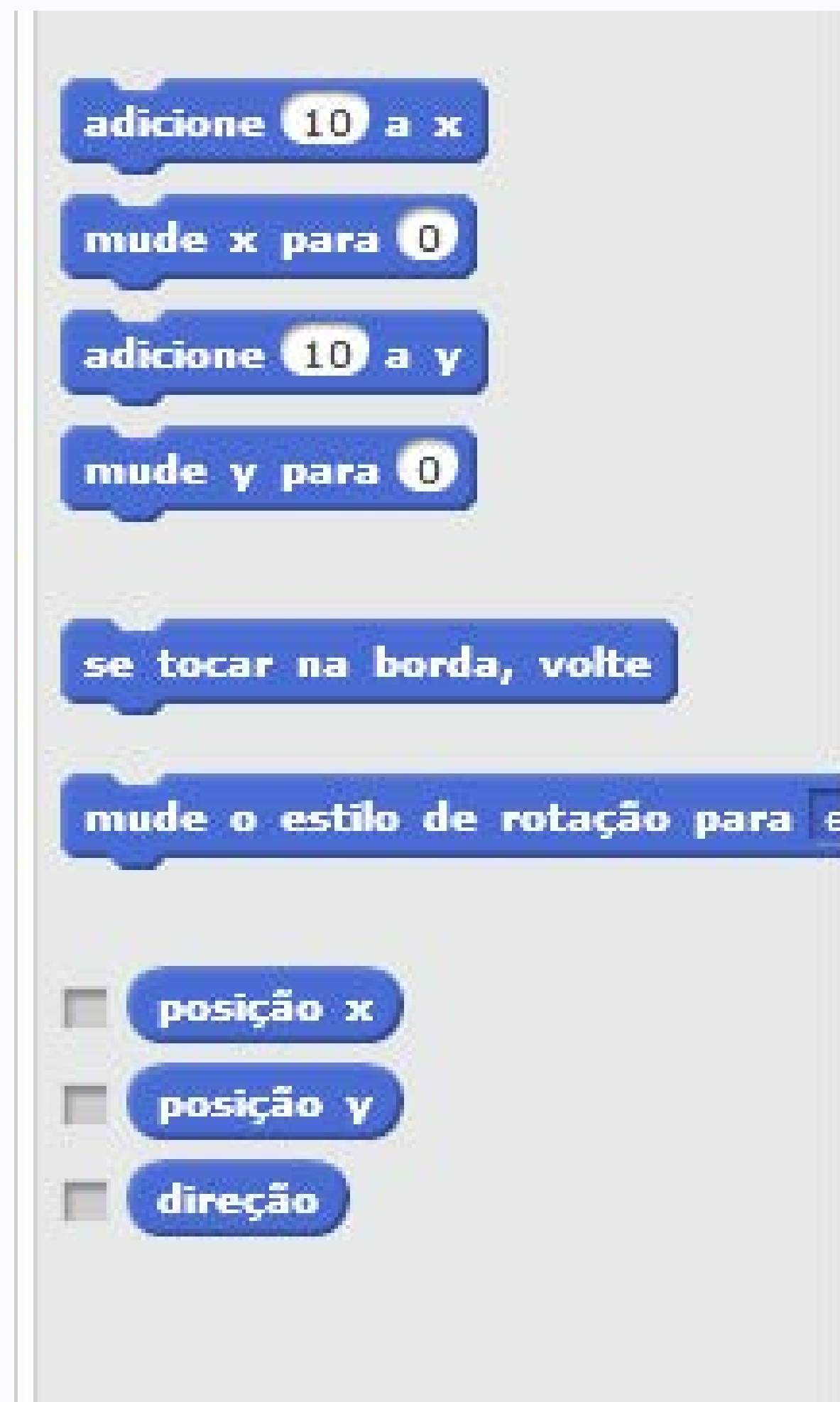
- Cada Sprite é único!
- Um Sprite tem suas próprias fantasias, scripts e sons
- Dê duplo clique no Sprite para ver as propriedades



Scratch: Conhecendo o Ambiente de Programação

Blocos: comandos de programação

- Organizados em dez categorias
- Mais de cem blocos
- Cada categoria possui sua respectiva cor



- Nem todos os blocos aparecem inicialmente, alguns ficam disponíveis sob certas condições que vamos aprendendo
- Ao clicar em um bloco, vemos o que ele faz
- Aparência: Diga oi por 2 segundos

Scratch: Conhecendo o Ambiente de Programação

Blocos: comandos de programação

- Alguns blocos exigem uma ou mais entradas
- Dê duplo clique para editar os valores

mova 10 passos

pense Hmm... por 2 segundos

Scratch: Conhecendo o Ambiente de Programação

Área de Scripts: Programação dos sprites

- Iremos definir o que os sprites devem fazer
- Arraste os blocos para esta área e combine-os

mova 20 passos

pense Hello! por 2 segundos

gire 45 graus

toque o som miau ▼

- Adicionar outro Sprite e mostrar a programação individual de ambos
- Introduzir bloco de controle

Scratch: Conhecendo o Ambiente de Programação

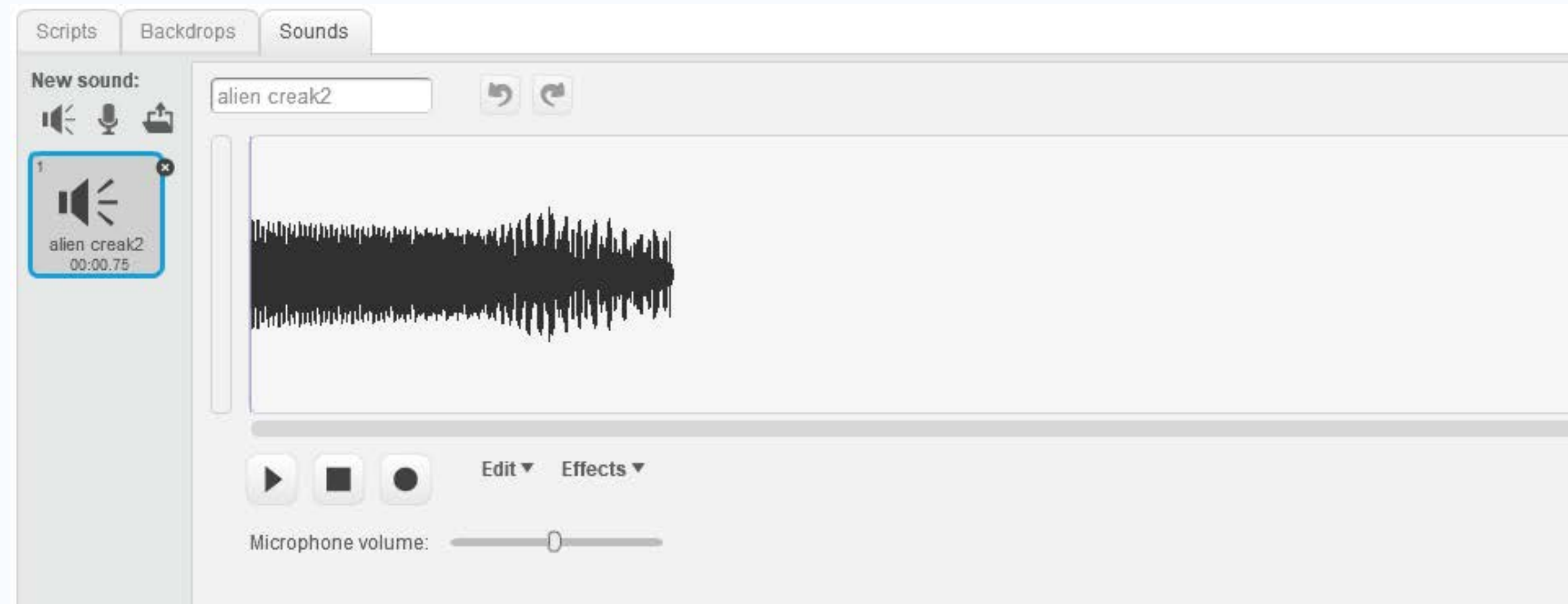
Aba Fantasias: mudar a aparência dos sprites



Scratch: Conhecendo o Ambiente de Programação

Aba Sons: Usar ou gravar novos sons para os sprites

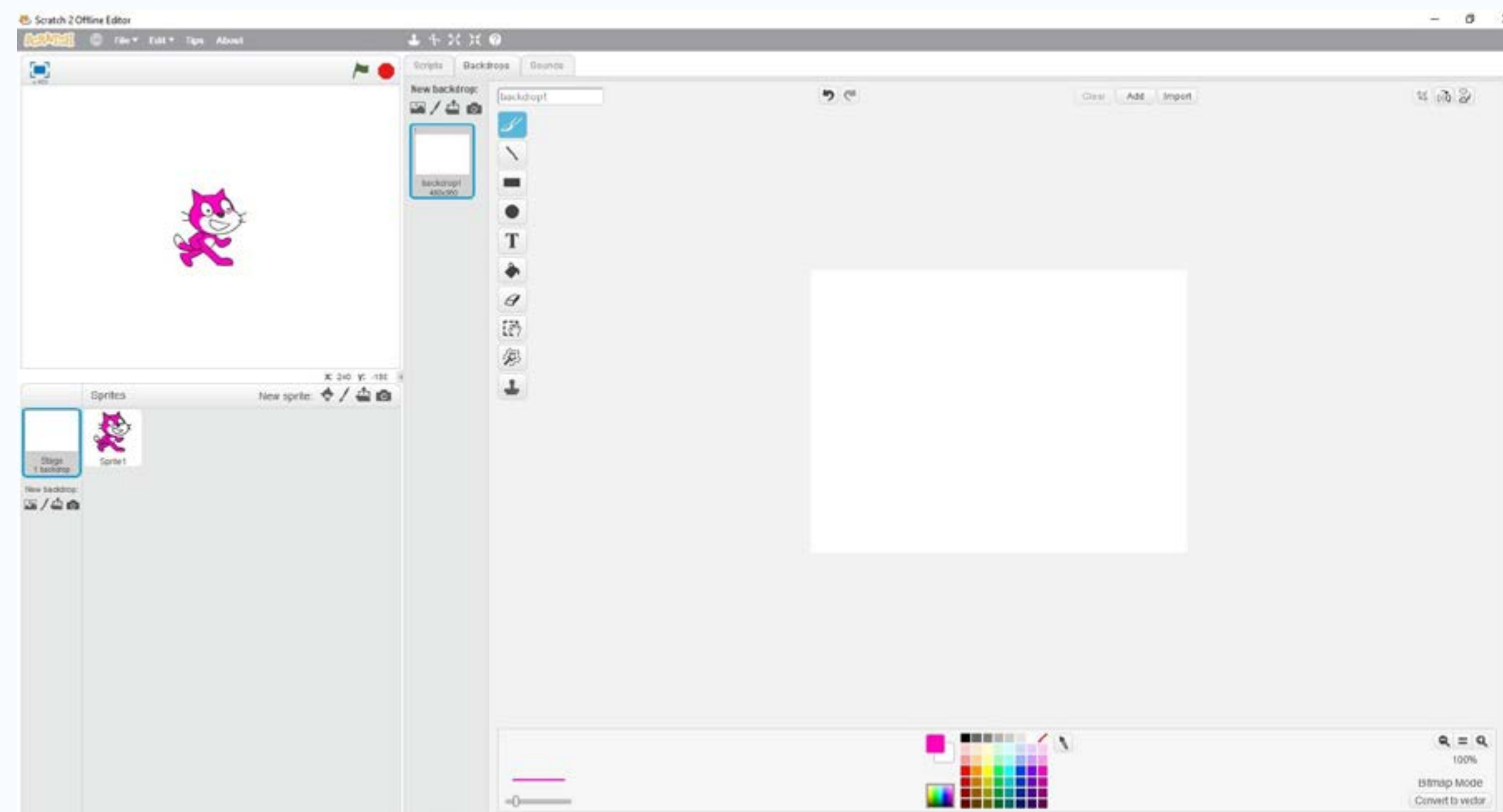
- Biblioteca de sons
- Captura com o Microfone
- Um Sprite pode ter mais de um som associado



Scratch: Conhecendo o Ambiente de Programação

Aba Plano de Fundo: Muda o background da sua aplicação

- Primeiro pano de fundo pode ser as instruções do jogo
- Depois uma cena em que o jogo corre
- Scripts também controlam a mudança no pano de fundo



- Possibilidade de criar imagens ou importa-las da internet
- Biblioteca com vários panos de fundo
- Alguns blocos disponíveis para fazer a troca de fundo

Scratch: Conhecendo o Ambiente de Programação

- Informações sobre o Sprite



- Clique no i ao lado do sprite
- Nome do Sprite
- Arrastar o Sprite e mostrar o efeito
- Indica a posição atual, o estilo de rotação e a visibilidade, isto é, se está visível ou não
- Exemplificar se tocar na borda (bloco movimento)
- Exemplificar pode ser arrastado no modo apresentação

Scratch: Conhecendo o Ambiente de Programação

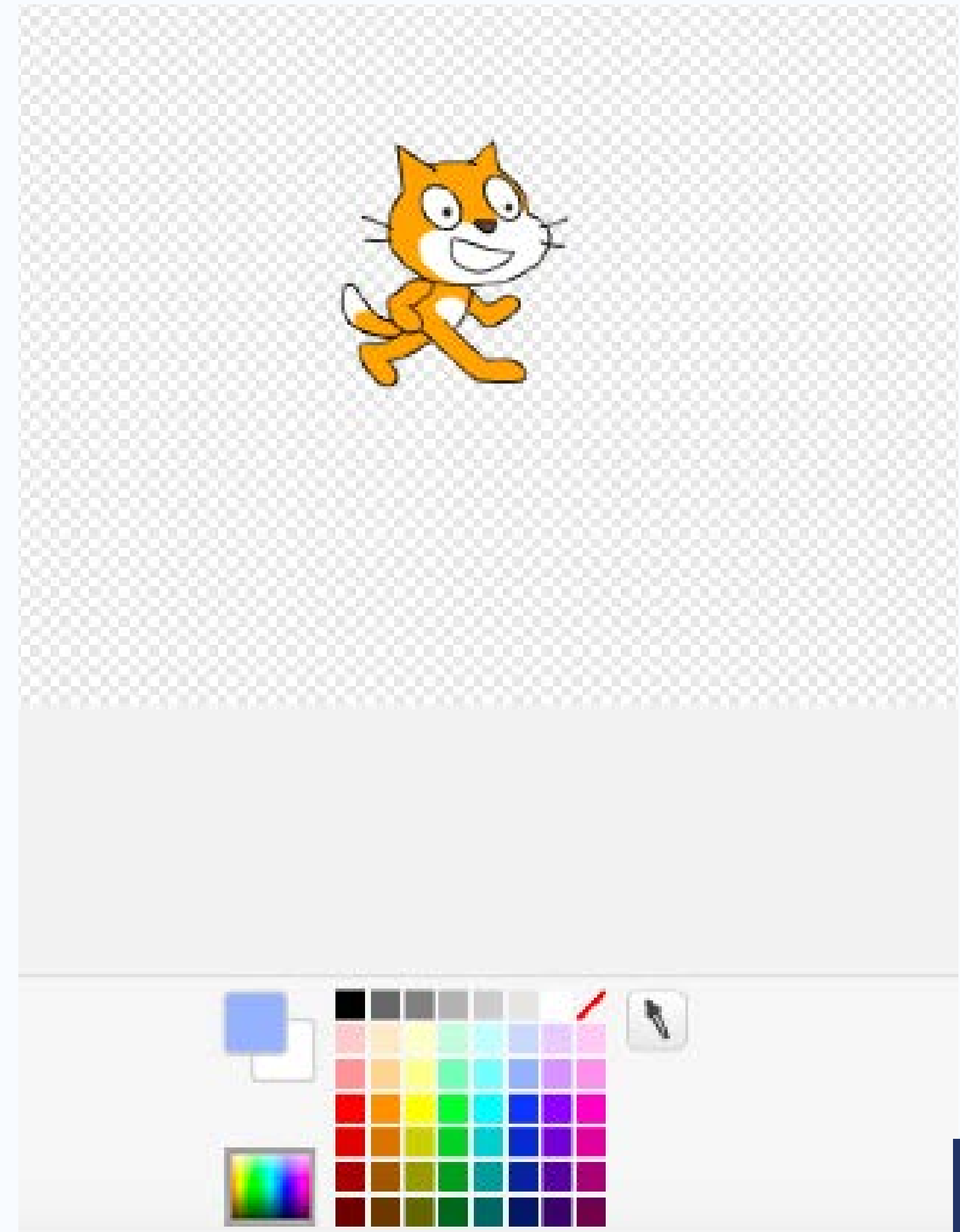
- Barra de Ferramentas



- A barra de ferramentas da nossa versão possui alguns elementos diferentes, pois temos o scratch off-line
- Modo turbo: acelera a execução dos passos
- Modo palco pequeno: favorece a concentração nos scripts

Scratch: Conhecendo o Ambiente

- Paint Editor: Definir a cor transparente



importante definir a cor transparente, pois dois sprites podem interagir e não queremos um sobrepondo o outro

Objetivos da manhã cumpridos!

1. Motivar e contextualizar a importância do ensino de programação na formação dos estudantes
2. Conhecer o ambiente de programação Scratch e seus componentes



Introdução à Lógica de Programação

**Pós-Graduação Lato Sensu em
Letramento Digital**

Profa. Dra. Elloá B. Guedes

ebgcosta@uea.edu.br

www.elloaguedes.com

Como a disciplina está organizada?

Aula 01 - A importância da programação e os primeiros fundamentos

- Manhã – Motivação e contextualização, conhecendo o Scratch
- Tarde – Conhecendo os blocos
- 10 horas/aula

Aula 02 - Conhecendo e praticando comandos

- Manhã – Comandos de entrada e saída
- Tarde – Desvio Condicional
- 10 horas/aula

Aula 03 - Melhorando as habilidades

- Manhã – Comandos de Repetição
- Tarde – Integrando conceitos
- 10 horas/aula

Como está organizada nossa aula nesta tarde?

Aula/Tarde - Conhecendo os Blocos

- 13h as 14h30min – Aula
- 14h30min as 14h40h - Intervalo
- 14h40min as 15h40min – Aula
- 15h40min as 15h50min – Instruções para o roteiro 2
- 15h50min as 16h – Instruções para o projeto prático

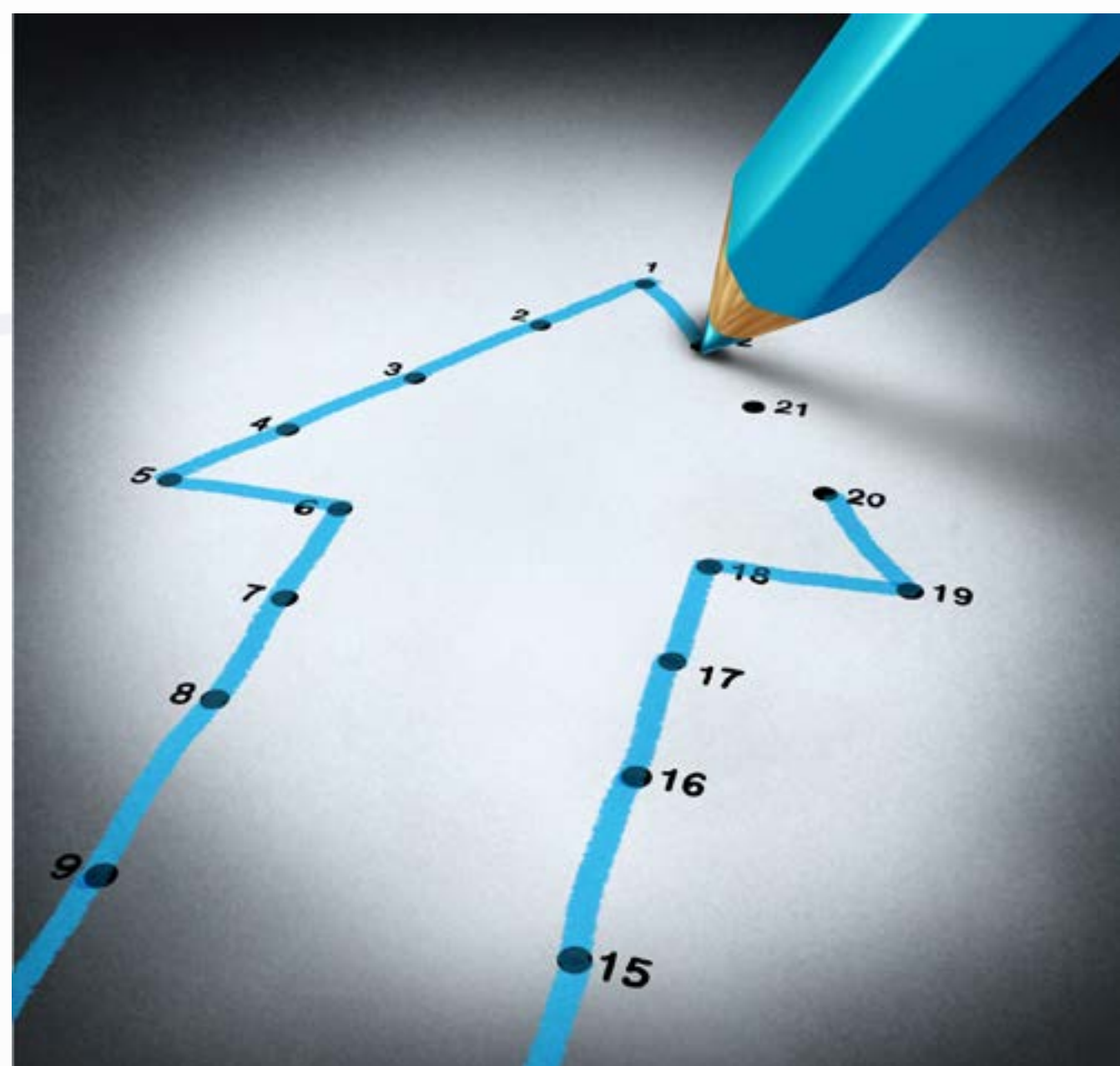
Objetivos

1. Conhecer os blocos do Scratch
2. Adentrar nos conceitos de programação, entendendo algoritmos, valores e expressões



Objetivos

- Linguagem de programação nos permite construir programas
- Scratch é uma linguagem de programação moderna e voltada para crianças



Objetivos

- Vamos usar o Scratch para construir **algoritmos**
- Conceito: sequência de passos que visa resolver um objetivo



- Algoritmos no dia a dia: receita de bolo ou maneira de trocar o pneu
- Precisamos ter em mente o objetivo
- Todos os programas de computadores são conjuntos de algoritmos
- Quando estamos praticando, esquecemos até o nome deste conceito
- A construção dos algoritmos no Scratch é feita por meio da combinação dos blocos
- Precisamos conhecer os blocos!

Conhecendo os Blocos

- Na interface do Scratch, localize a aba dos Blocos
- Há dez tipos de blocos disponíveis
- Quatro grandes grupos de blocos



Visão Geral dos Blocos: Comando e Controle

- Possuem saliências na parte de baixo
- Alguns podem ter saliências na parte de cima
- Úteis para empilhar os blocos, criando os algoritmos

mova 10 passos

toque o som miau ▼

Diga Hello!

gire ↻ 15 graus

Diga Hello!

mova 10 passos

toque o som miau ▼

gire ↻ 15 graus

- No Scratch, este tipo de blocos correspondem às ações dos sprites
- Posso combinar várias dessas ações

Tarefa

- Faça o gatinho andar, dizer oi por 2 segundos e miar!



- Lembrar que esses bloquinhos juntos atendem a um objetivo e, portanto, constituem um algoritmo

Visão geral dos blocos:

- Blocos de evento dependem de um evento
- Se “tal coisa acontece”, então o sprite deve executar uma ação em consequência
- Não possuem saliência na parte de cima
- O evento que estou esperando é o gatilho para as ações



- Esses eventos que servem como gatilho comumente depende de ações do usuário: apertar uma determinada tecla, clicar com o mouse.
- Ator é o mesmo que sprite

Tarefa

- Quando clicar no gatinho, ele mia!



- **Blocos de evento são úteis para interação com o usuário!**
- **Em um jogo, por exemplo, o usuário irá usar mouse e teclado**
- **Numa animação, pode servir para passar para uma próxima etapa**
- **Apresentar o bloco de controle da bandeirinha verde**

Visão geral dos blocos: Blocos de Função

- Blocos de função não possuem saliência
- Devolvem um valor
- Possuem formato arredondado ou pontiagudo
- Não fazem nada sozinhos, mas ajudam outros blocos

Mouse pressionado

posição x do mouse

posição y do mouse

- Observe que as extremidades dos bloquinhos são diferentes
- São relacionadas com o valor que devolve
- Bloquinhos arredondados devolvem texto ou número
- Bloquinhos pontiagudos devolvem verdadeiro ou falso

Visão Geral dos Blocos: Blocos de Controle

- Ajudam a controlar uma ação do Sprite
- Podem repetir blocos de comandos por uma certa quantidade de vezes ou para sempre

espere até



repita 10 vezes



- Nós vamos explorar mais esses bloquinhos nas aulas seguintes
- Estamos interessados em refinar as ações dos nossos atores

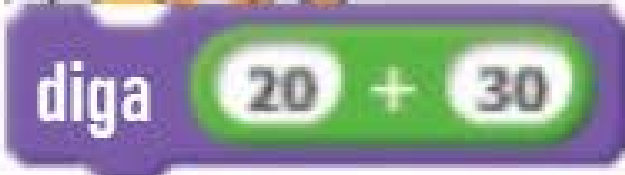
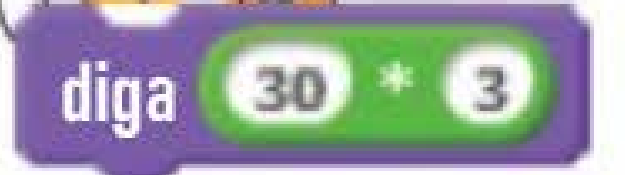
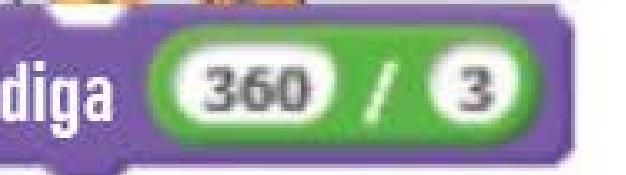
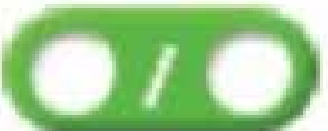
Tarefa

- Fazer o gatinho andar de um lado para outro no palco quando clicar na bandeirinha verde



Visão Geral dos Blocos: Operadores

- O Scratch possui módulos para realização de algumas operações matemáticas

Exemplo				
Operador				
				
	Adição	Subtração	Multiplicação	Divisão

Tipos de dados

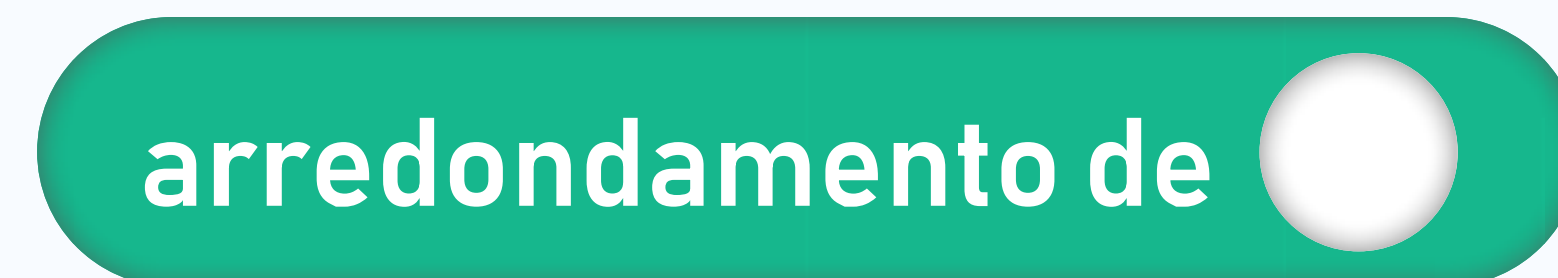
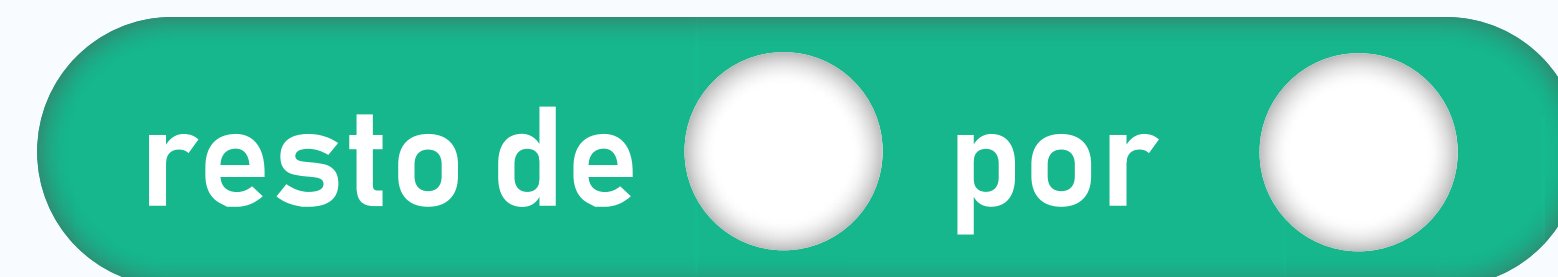
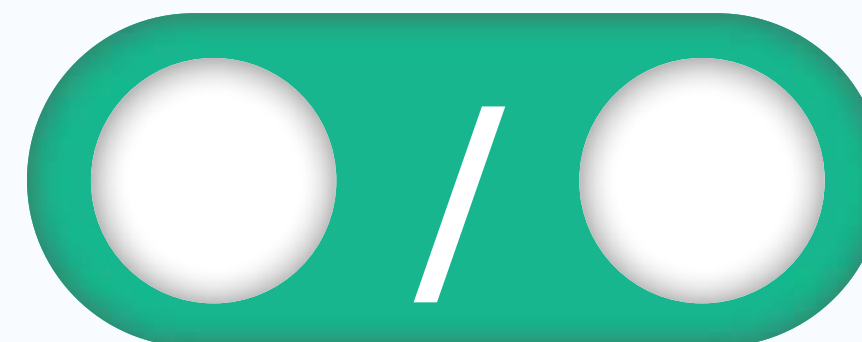
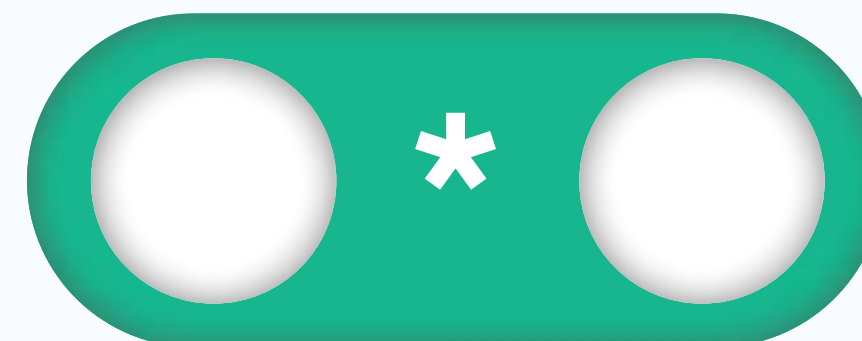
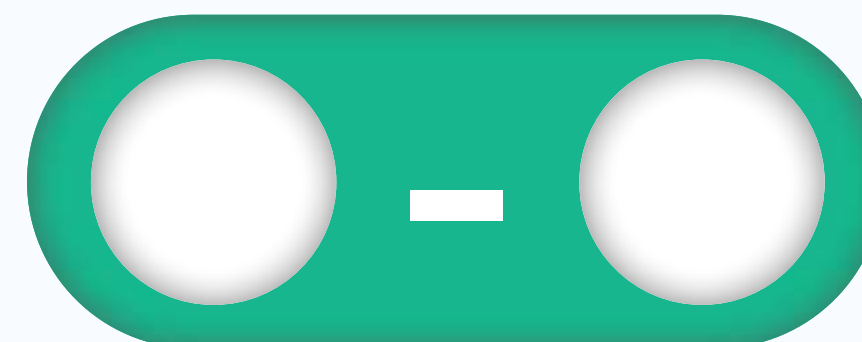
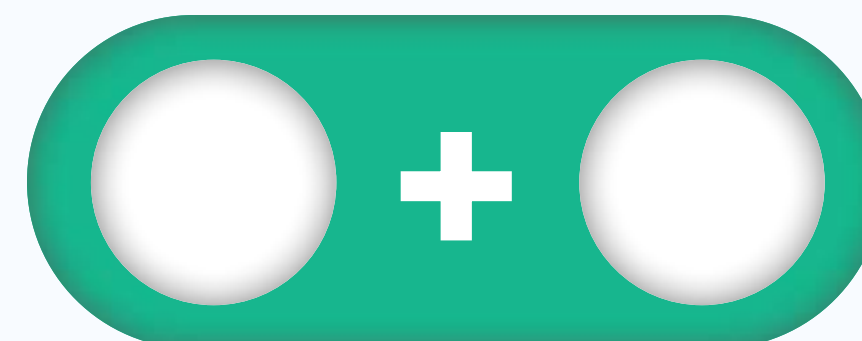
Os tipos de dados considerados no Scratch são:

- Numéricos
 - Números inteiros
 - Números reais
- Textual
 - Chamados de strings
- Lógico
 - Chamados de booleanos

Visão Geral dos Blocos: Operadores

- Compatíveis com os tipos de dados numéricos

Operações:



- **Mostrar detalhadamente as operações e seus resultados**
- **Falar que a operação do resto é também chamada de mod**
- **Exemplificar zero dividido por zero**
- **Cerca de 14 funções matemáticas**

Visão Geral dos Blocos: Operadores

- Compatível com o tipo de dados numérico
- Útil para jogos e simulações
- Gera valores aleatórios

número aleatório entre 0 e 1.0

número aleatório entre 1 e 10

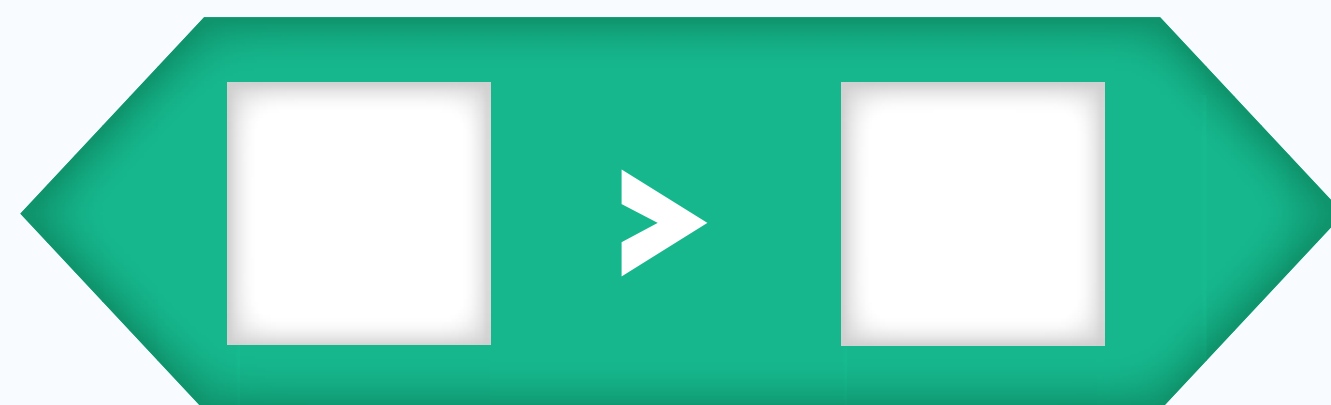
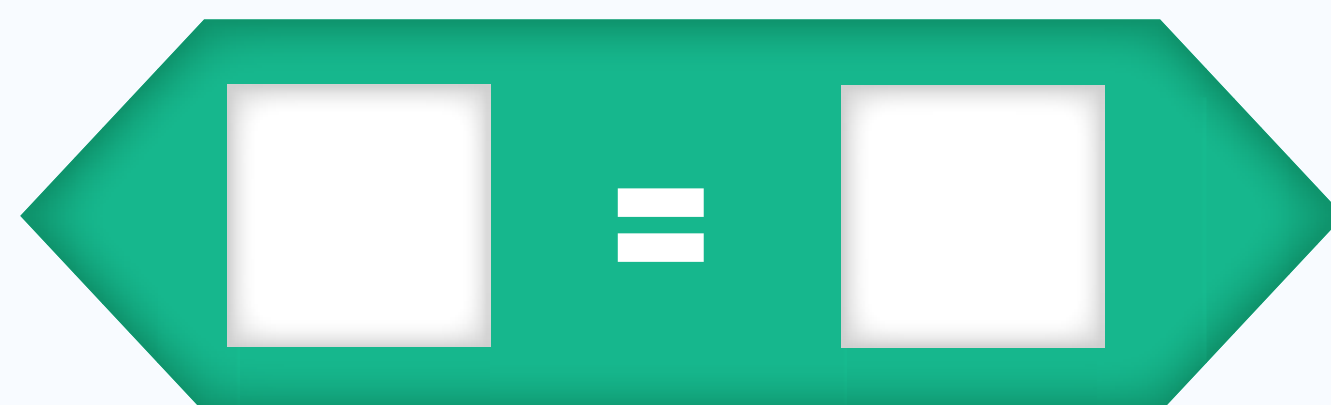
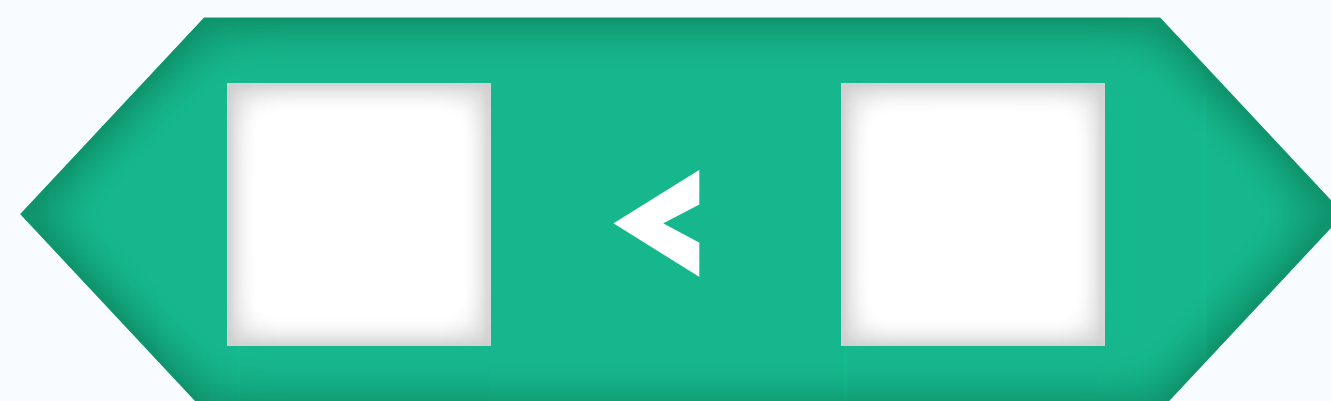
Exemplificar, ressaltando a diferença entre as duas situações

Tipos de dados: Dados Booleanos

- O tipo de dados lógico (booleano) representa os valores verdadeiro e falso
- **true** corresponde a verdadeiro
- **false** corresponde a falso

Visão Geral dos Blocos: Operadores

- Compatíveis com os tipos de dados booleanos



Comparações



Operações lógicas

Introduzir valores lógicos true e false
Mostrar tabela verdade

Visão Geral dos Blocos: Operadores

- Compatíveis com os tipos de dados textuais

junte **olá** com **mundo**

letra **3** de **abacaxi**

tamanho de **paralelepipedo**

Úteis para montar frases aos usuários, tornando o jogo ou animação mais personalizado

Introdução à Lógica de Programação

Pós-Graduação Lato Sensu em Letramento Digital

Profa. Dra. Elloá B. Guedes

ebgcosta@uea.edu.br

www.elloaguedes.com

Objetivos

1. Conhecer os blocos do Scratch
2. Adentrar nos conceitos de programação, entendendo algoritmos, valores e expressões



Conhecendo os blocos: Blocos de movimento

- Blocos de comandos → Blocos de movimento
- • Denotados pela cor azul escura
- • Dizem respeito ao movimento do sprite



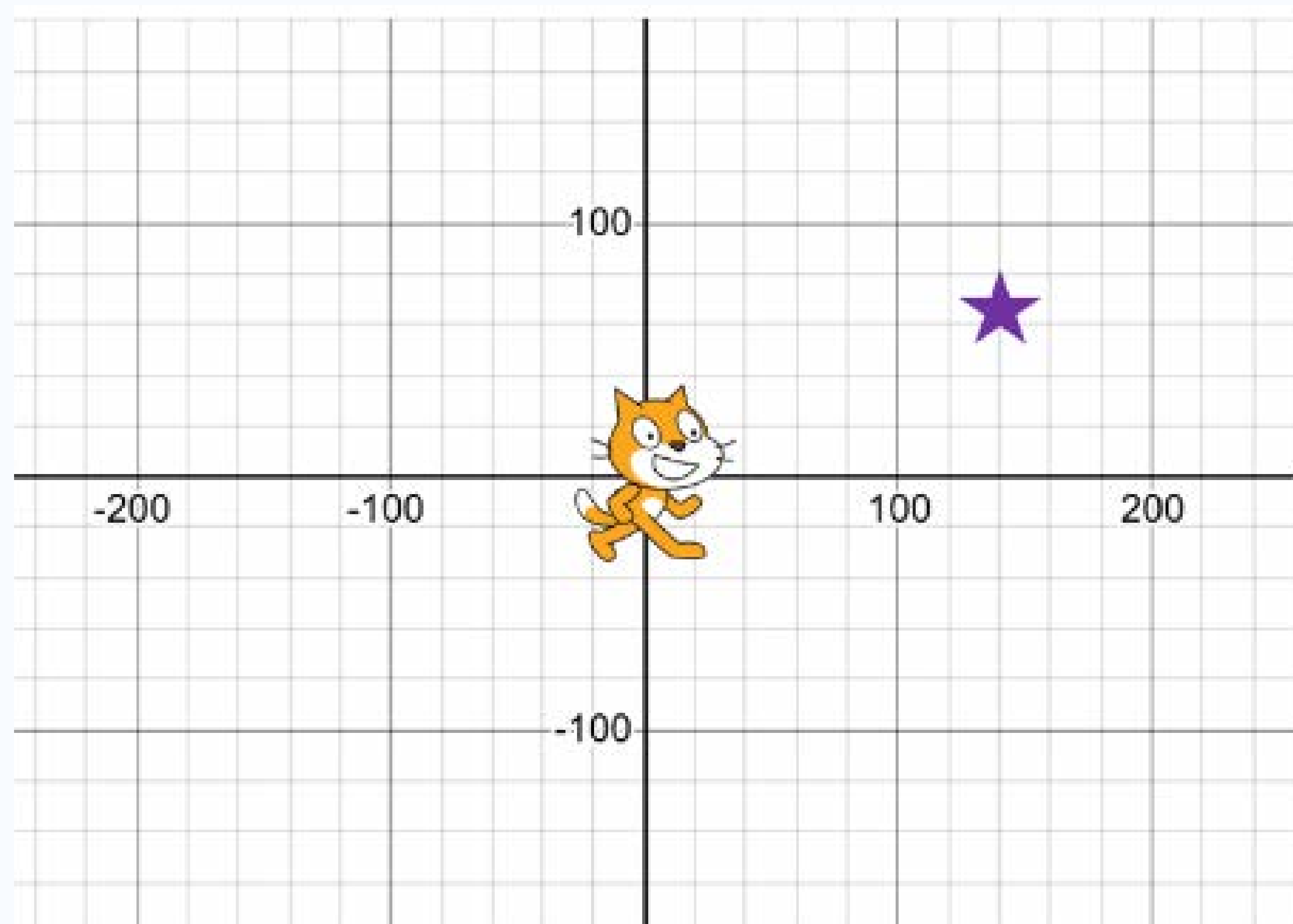
- Úteis para mover os sprites na tela
- A partir daí, colocar o gato no centro da tela, na posição original e fazê-lo andar 30 passos

Tarefa

- Ao ser clicado, o gatinho deve ir para a posição ($x = 120$, $y = 60$) no palco



Blocos de Movimento



- Vamos lembrar o sistema de coordenadas do palco
- Solução 1: via movimento absoluto (vá para x, y)
- Solução 2: deslize por 1 segundo para
- Solução 3: mude x para 120, mude y para 60
- Este é um exemplo de movimento absoluto: o problema é resolvido indicando diretamente as coordenadas a alcançar

Tarefa

- Coloque no palco dois sprites: um relativo a um meio de transporte e outro a uma estrela
- Os dois sprites devem estar em posições escolhidas aleatoriamente por você
- Faça o meio de transporte atingir a estrela



Blocos de Movimento

- Nesta solução, utilizamos movimentos relativos
- De onde está, mova mais dez passos
- De onde está, gire 15 graus à direita
- E assim por diante!

mova 10 passos

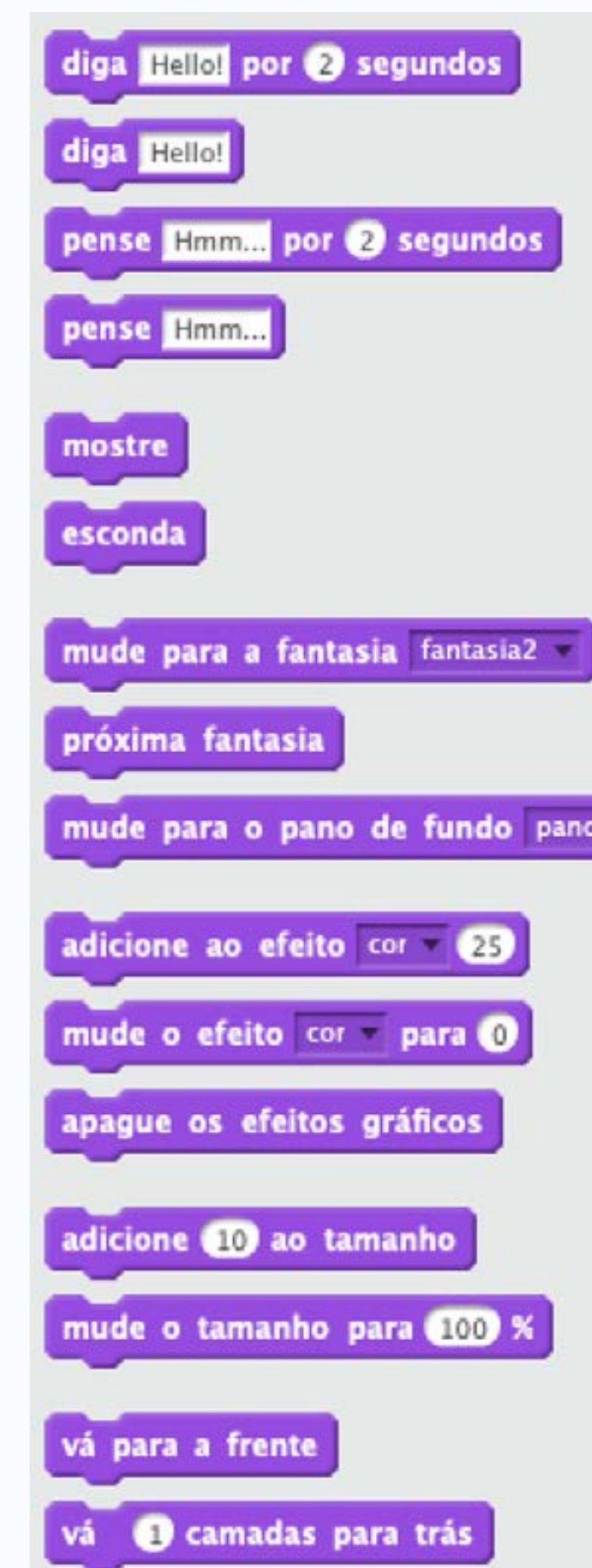
aponte para star 2 ▼

gire ↻ 15 graus

- Muitas vezes, o usuário irá interagir com o sprite, alterando sua posição
- Nem sempre saberemos exatamente onde o sprite está, mas iremos querer que ele vá sutilmente até outra posição

Blocos de Aparência

- Blocos de comandos → Blocos de aparência
- Denotados pela cor roxa
- Dizem respeito à aparência do personagem
 - Balões de fala
 - Fantasia do personagem
 - Tamanho do personagem



Blocos de Aparência

1  clear graphic effects	2  set color effect to 20	3  set fisheye effect to 50
4  set whirl effect to 200	5  set pixelate effect to 30	6  set mosaic effect to 10
7  set ghost effect to 70	8  set brightness effect to 80	1 clear 2 color effect 3 fisheye effect 4 whirl effect 5 pixelate effect 6 mosaic effect 7 ghost effect 8 brightness effect

Tarefa

- Faça o gato pensar “Oxe”, andar 20 passos, crescer 10% em seu tamanho, falar “Olá, pessoal!” e ficar cor de rosa



Tarefa

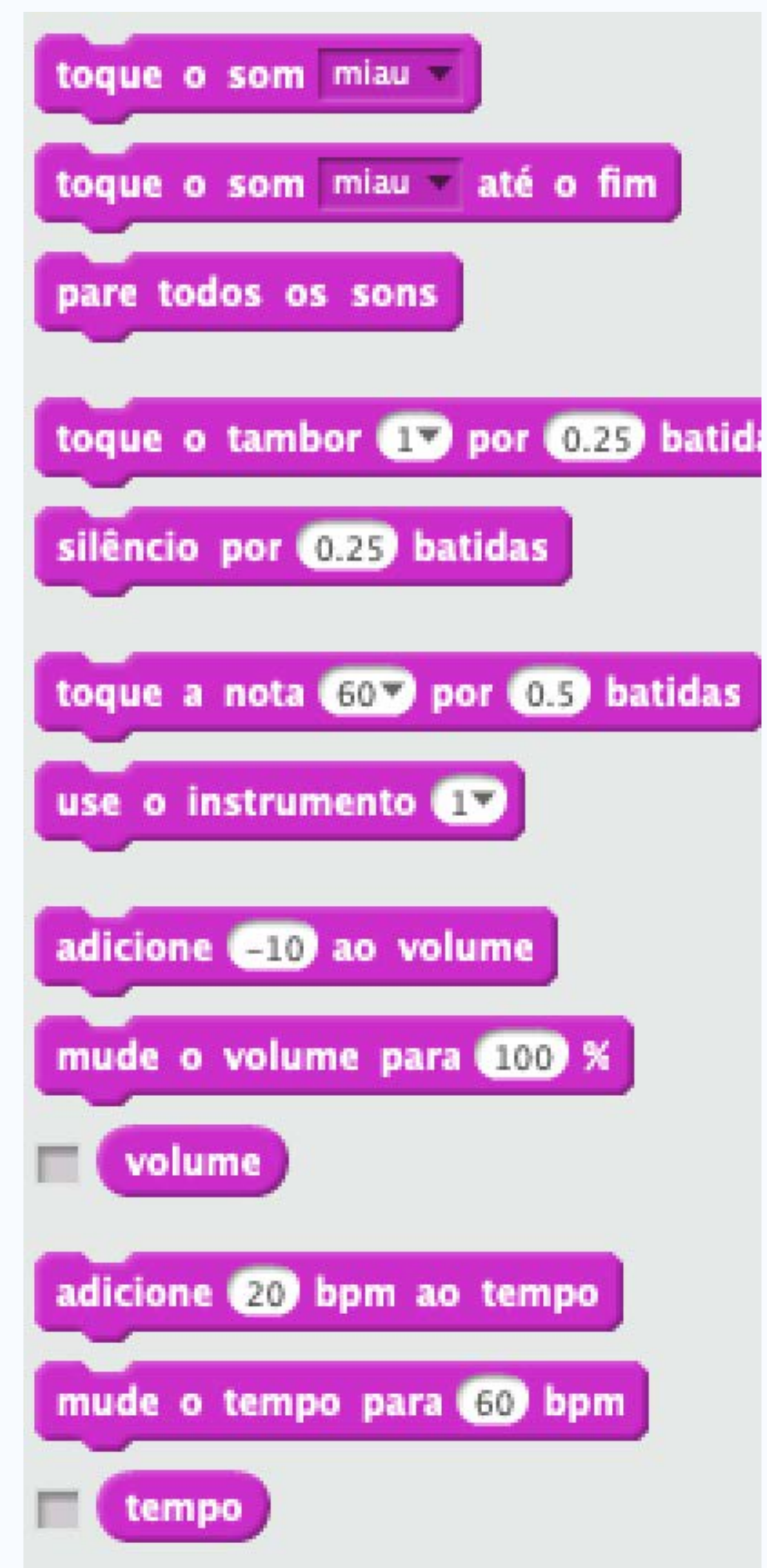
- Simule um semáforo de trânsito, que mude de vermelho para amarelo e então para verde, repetindo continuamente esse processo



- Procurar na galeria dos sprites e mostrar que não há semáforo
- Desenhar sprite e suas fantasias
- Animar e colocar Pensar por x segundos mas sem texto, simulando a espera

Conhecendo os blocos

- Blocos de comandos → Blocos de som
- Denotados pela cor rosa
- Efeitos sonoros para jogos
- Fundo musical para animações



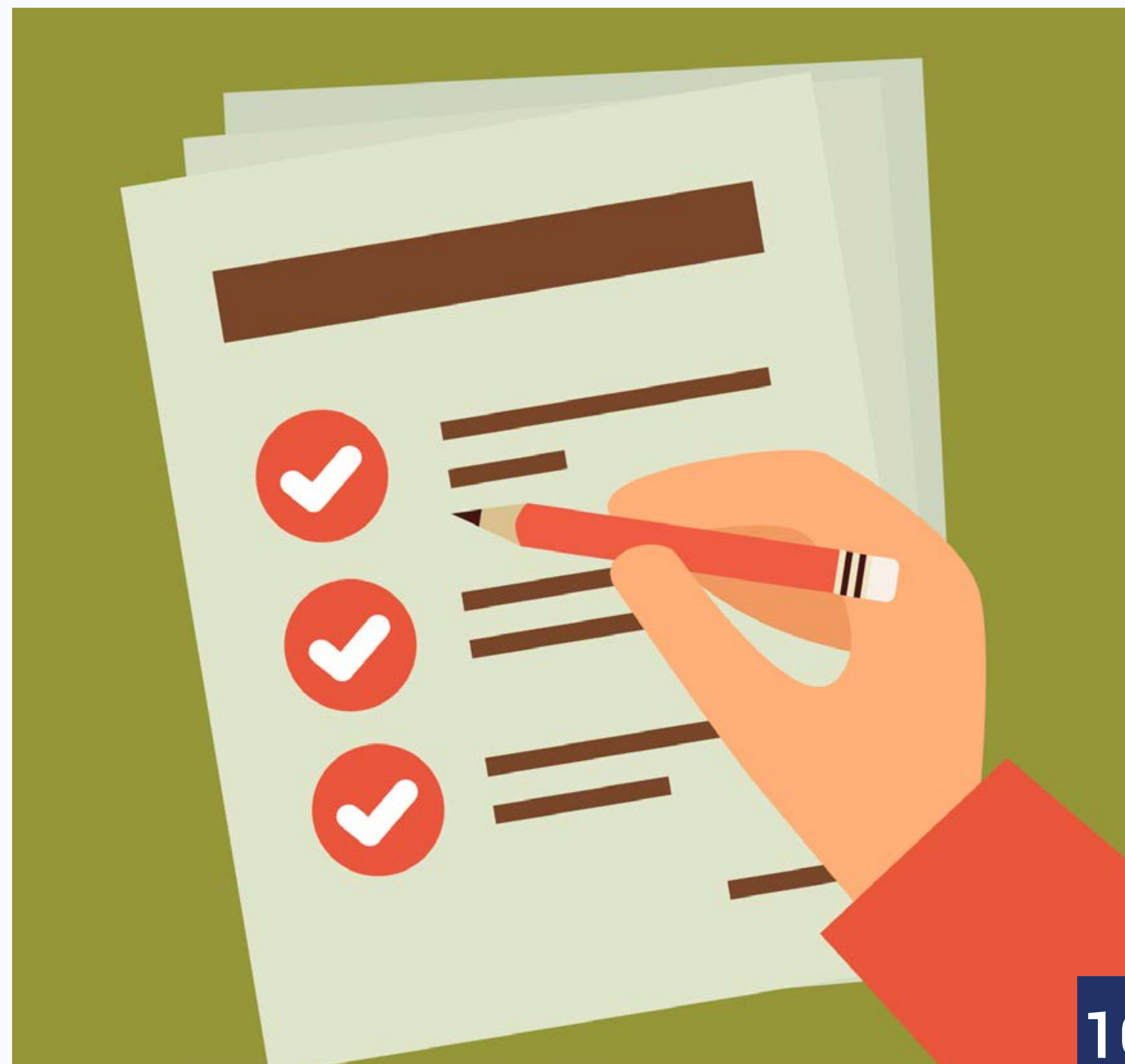
Tarefa

Complemente a tarefa anterior, fazendo o gatinho miar!



Tarefa

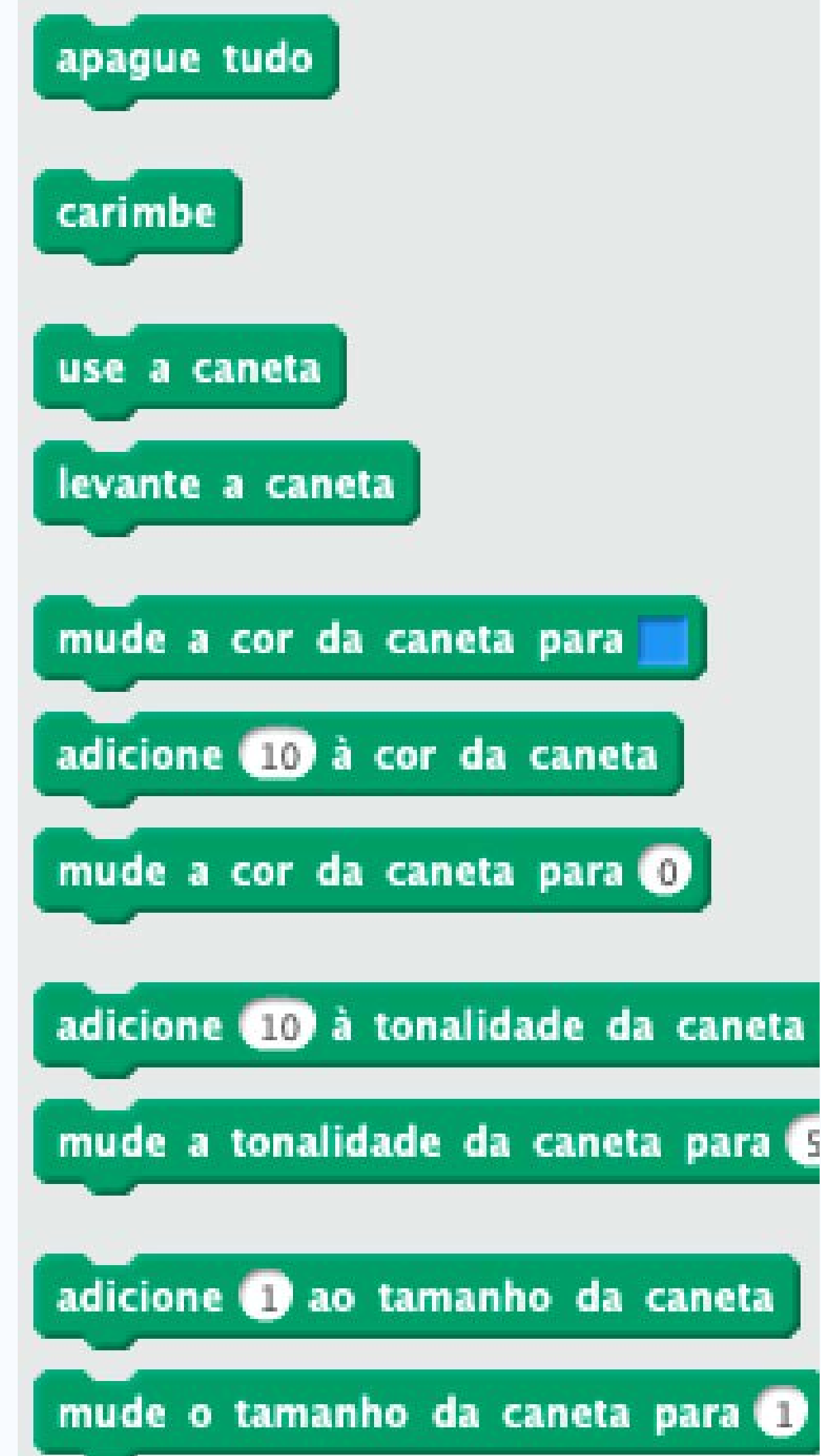
- Imagine que o gato e o cachorro estão conversando, visando discutir qual dos dois é o mais bonito!
- O gato fala primeiro “O mais bonito sou eu!” e depois mia
- Depois o cachorro fala “Não, o mais bonito sou eu!” e então late
- Seguem discutindo eternamente!



- Usar stage1 como pano de fundo
- Arrumar posição do cachorro

Conhecendo os blocos

- Blocos de comando → Blocos de caneta
- Denotados pela cor verde escuro
- Servem para que o sprite possa desenhar na tela, atuando como a tartaruguinha do Logo



- Note que aqui temos mais possibilidades de personalização, pois as cores da caneta podem ser mudadas, a espessura do traço e muito mais!
- Vamos exemplificar com o gatinho fazendo um quadrado com a largura da caneta aumentando

Tarefa

- Use o sprite do besouro como se fosse a tartaruga
- Quando a bandeira verde for clicada, coloque-o na posição (0,0), direção 90 graus e limpe o palco
- Quando o besouro for clicado, você deve desenhar um círculo na tela, na cor rosa e com espessura igual a 2



Tarefa

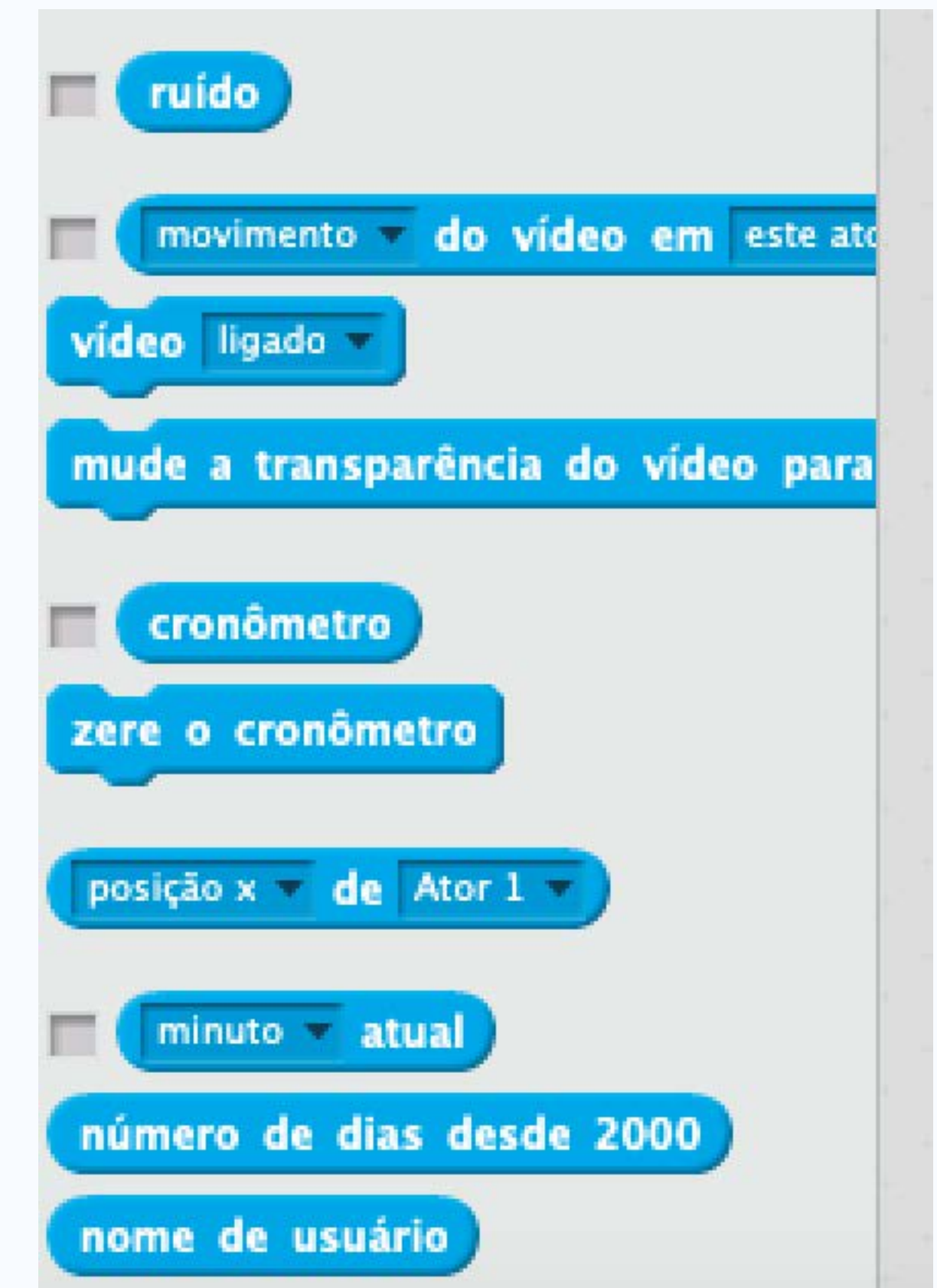
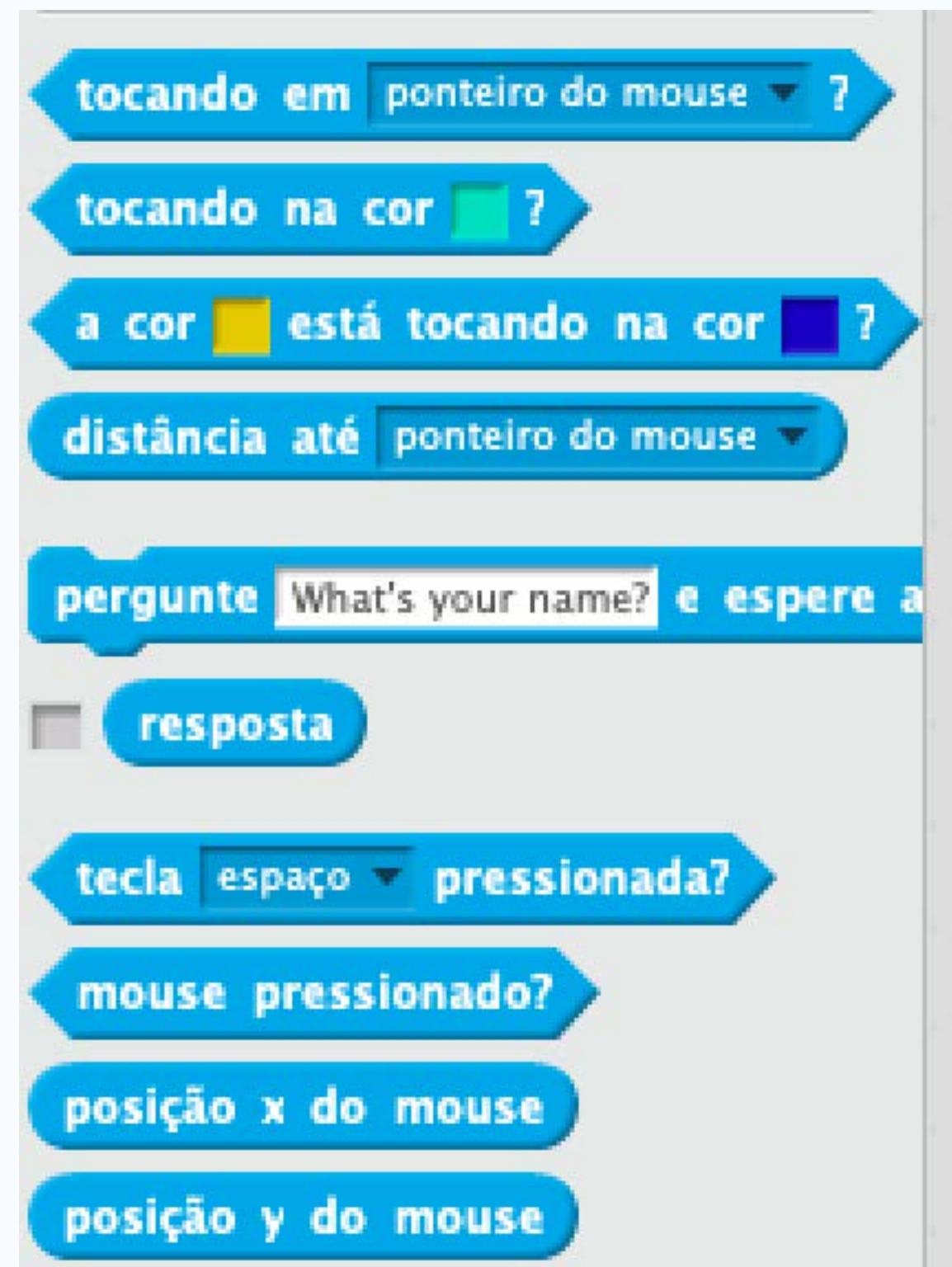
- Construa um algoritmo que faça o gato ir para onde o cursor do mouse está sempre que a barra de espaço for pressionada



- O gatilho da ação é “a barra de espaço é pressionada”
- Preciso saber a posição do mouse (coordenadas x e y)
- Preciso mandar o gato para lá
- Blocos de função mostrados sabem a posição do mouse

Blocos sensores

- Blocos sensores: cor azul clara
- Atenção ao formato
- Detecção de informações



- Muito dos algoritmos que estamos fazendo até o momento não são muito interativos
- Precisamos de uma maneira de ler informações do usuário

Tarefa

- Faça o gatinho perguntar o nome do usuário
- O gatinho deve cumprimentar o usuário "Olá, nome do usuário" e miar de felicidade
- Em seguida, deve correr atrás do mouse pelo palco
- Por onde correr, o gato deve deixar um rastro riscado no chão
- Ao apertar a barra de espaço, deve parar e miar novamente



- **Note os diferentes papéis dos blocos sensores nesta resposta: capturar a resposta, a posição do mouse e se a tecla de espaço foi ou não pressionada**
- **Note também que fizemos uma entrada de dados, isto é, usamos os blocos sensores para capturar uma informação dos usuários para dentro do nosso programa**
- **Também fizemos saída de dados, isto é, imprimos várias vezes na tela mensagens para o usuário**
- **Combinamos blocos de diferentes tipos para compor nosso objetivo!**

Atividade 02 – Operadores e expressões

Entrega

Mesmo dia – Tarde

Quantidade de alunos
por equipe

No máximo 3

Duração da Atividade

30 min

Descrição da Atividade

Consulte o Roteiro de Aprendizagem 02