



UNIVERSIDADE  
FEDERAL DO CEARÁ



DEPARTAMENTO  
DE COMPUTAÇÃO

## **CK0176/177 - Informática Aplicada ao Ensino de Ciências: Linguagem LOGO e SCRATCH**

Prof. Maurício Moreira Neto

# Relembrando a linguagem LOGO...

- A linguagem de programação LOGO
  - Simples e Estruturada
  - Voltada para o ensino de programação
- Principal objetivo desta linguagem é a familiarização de conceitos lógicos e matemáticos para resoluções de problemas
- Foi desenvolvida pelo MIT meados dos anos 60

# Relembrando a linguagem LOGO...

- Vantagens do uso do LOGO
  - Acompanhamento gradual do raciocínio lógico do aluno
  - Desperta a curiosidade para as diversas possibilidades do uso da linguagem
  - Integralização do conhecimento
  - Diferentes soluções para um mesmo problemas
  - ...
- A linguagem envolve uma tartaruga gráfica
  - Um robô que está pronto para receber os comandos do usuário

# Relembrando a linguagem LOGO...

- O LOGO é uma linguagem interpretada e interativa...
- Mas o que é uma **linguagem interpretada**?!
  - Linguagem interpretada são aquelas que precisam de um programa especial chamado interpretador para que seus códigos seja executados. O programa analisa a instrução e, em seguida, a executa!

# Linguagem LOGO

- Vamos usar um ambiente online para programar usando a linguagem de programação LOGO
  - Acesse o link a seguir para entrar no ambiente online:
  - <https://www.calormen.com/jslogo/>

# Linguagem LOGO

Logo Interpreter [Tests](#) | [Source](#) | [Collaborate](#) |

↑

**Área de Execução do Código**



Exemplo de funções

} ↓

```
TO star
  repeat 5 [ fd 100 rt 144 ]
END
clearscreen
star

TO square :length
  repeat 4 [ fd :length rt 90 ]
END

TO randomcolor
  setcolor pick [ red orange yellow green blue violet ]
END
clearscreen
repeat 36 [ randomcolor square random 200 rt 10 ]

clearscreen window hideturtle
repeat 144 [
  setlabelheight reccount
  penup
  fd reccount * reccount / 30
  label "Logo
  bk reccount * reccount / 30
  pendown
  rt 10
  wait 5
]
showturtle

TO tree :size
  if :size < 5 [forward :size back :size stop]
  forward :size/3
  left 30 tree :size*2/3 right 30
  forward :size/6
  right 25 tree :size/2 left 25
  forward :size/3
  right 25 tree :size/2 left 25
```

Type your code here... ← **Área de codificação** ↗

Run  
Clear

[Reference](#) – the Logo language  
[Library](#) – your procedures  
[History](#) – everything you've done here  
[Examples](#) – fun things to try out  
[Extras](#) – helpful utilities  
[Links](#) – other Logo resources

# Criando Algoritmos

- Pronto, agora estamos pronto para criar os nossos algoritmos....
- **Mas calma... O que são algoritmos?**
  - Algoritmo é uma sequência finita de passos (ou instruções) que visam atingir a um objetivo bem definido. A lógica é utilizada para ordenar os passos do algoritmo.
  - **Ex:** Receita do bolo, Realizar alguma tarefa, ...

# Criando Algoritmos

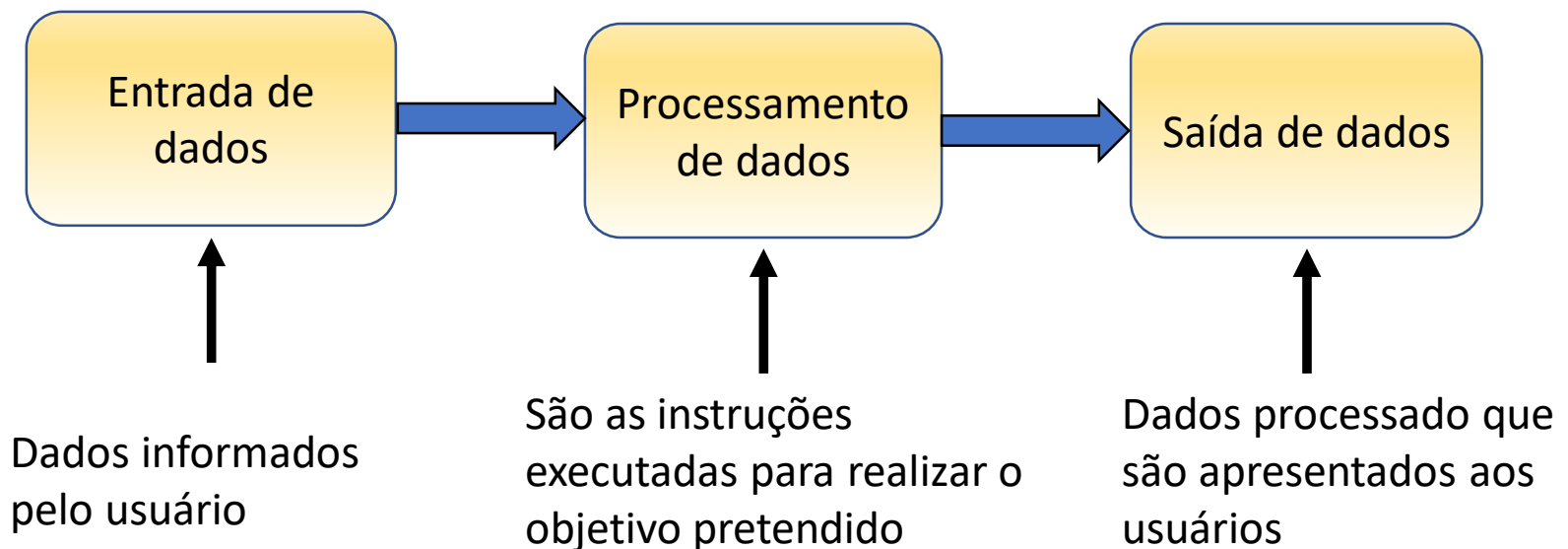
- **Exemplo de algoritmo:**

1. Aqueça o forno a 200°C
2. Unte uma forma redonda
3. Junte em uma panela manteiga, açúcar, 4 ovos, ...
4. Deite a massa na forma
5. Leve ao forno durante 40 minutos
6. Retire quando a massa inchar
7. ...



# Criando Algoritmos

- Um algoritmo é um procedimento computacional definido em 3 partes:



# Linguagem LOGO

- Vamos começar com um simples “Olá Mundo!”

```
print[Olá Mundo]
```

ou

```
show “Olá_Mundo”
```

Esta função mostra o dado de entrada na tela principal!

# Linguagem LOGO

- Também é possível realizar cálculos usando a função **print** junto com os operadores matemáticos
- Exemplo:

```
print 5 + 1
```

```
print 6/2
```

```
print 4*2
```

# Linguagem LOGO

É possível fazer a tartaruga se mover por meio dos comandos

## Exemplo:

```
forward 100
```

```
back 100
```

```
left 90
```

```
right 90
```

# Linguagem LOGO

- A linguagem LOGO também fornece estruturas de repetição
- Exemplo:

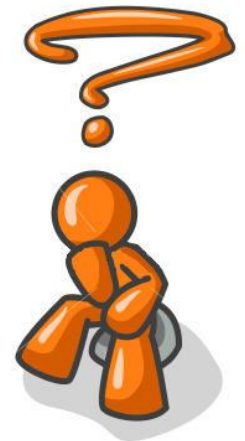
```
repeat 5 [print [Olá pessoal]]
```

```
Olá pessoal  
Olá pessoal  
Olá pessoal  
Olá pessoal  
Olá pessoal
```



# Linguagem LOGO

Com base nestes comandos aprendidos, como vocês seria um algoritmo para que a tartaruga fizesse um quadrado?



# Linguagem LOGO

- Algoritmo Quadrado
  1. Andar pra frente 100 passos
  2. Girar 90 graus para a direita
  3. Andar pra frente 100 passos
  4. Girar 90 graus para a direita
  5. Andar pra frente 100 passos
  6. Girar 90 graus para a direita
  7. Andar pra frente 100 passos
  8. Girar 90 graus para a direita

# Linguagem LOGO

- Algoritmo Quadrado

- 1. Andar pra frente 100 passos**
- 2. Girar 90 graus para a direita**
3. Andar pra frente 100 passos
4. Girar 90 graus para a direita
5. Andar pra frente 100 passos
6. Girar 90 graus para a direita
7. Andar pra frente 100 passos
8. Girar 90 graus para a direita



Podemos repetir isso 4 vezes!



# Linguagem LOGO

- Usando a linguagem LOGO, este algoritmo ficaria da seguinte maneira:

```
repeat 4 [forward 100 right 90]
```

- Também é possível fazer um procedimento e somente chamarmos quando necessário:

```
To quadrado  
  repeat 4 [forward -100 right 90]  
end  
quadrado
```

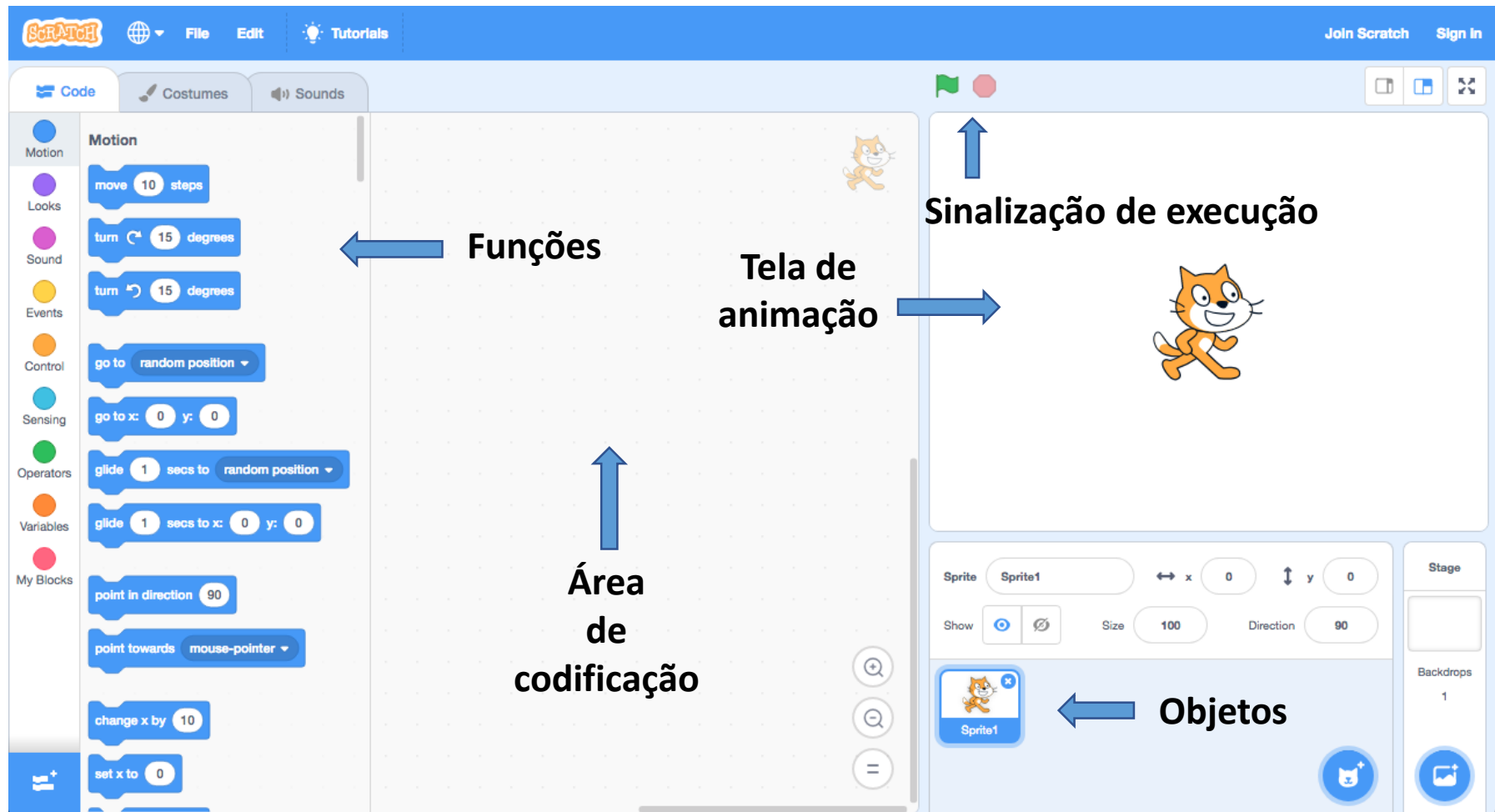
# Linguagem SCRATCH



# Linguagem SCRATCH

- O scratch, assim como o LOGO, é uma linguagem visual criado com intuito de facilitar o aprendizado de lógica e programação para iniciantes
- O scratch oferece uma interface intuitiva e de fácil compreensão, remetendo ao modelo construtivista
- O scratch também possui uma plataforma online
  - <https://scratch.mit.edu/projects/editor/?tutorial=getStarted>

# Linguagem SCRATCH



The image shows the Scratch programming environment interface. The left sidebar contains a palette of blocks categorized by function: Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, and My Blocks. The main workspace is a grid for coding. The right side features a stage for animation, a sprite control panel, and a stage control panel. Annotations with blue arrows point to specific areas:

- Funções**: Points to the block palette on the left.
- Área de codificação**: Points to the main workspace grid.
- Tela de animação**: Points to the stage area where the cat sprite is visible.
- Sinalização de execução**: Points to the green flag icon at the top right of the stage.
- Objetos**: Points to the sprite control panel at the bottom right.

# Linguagem SCRATCH

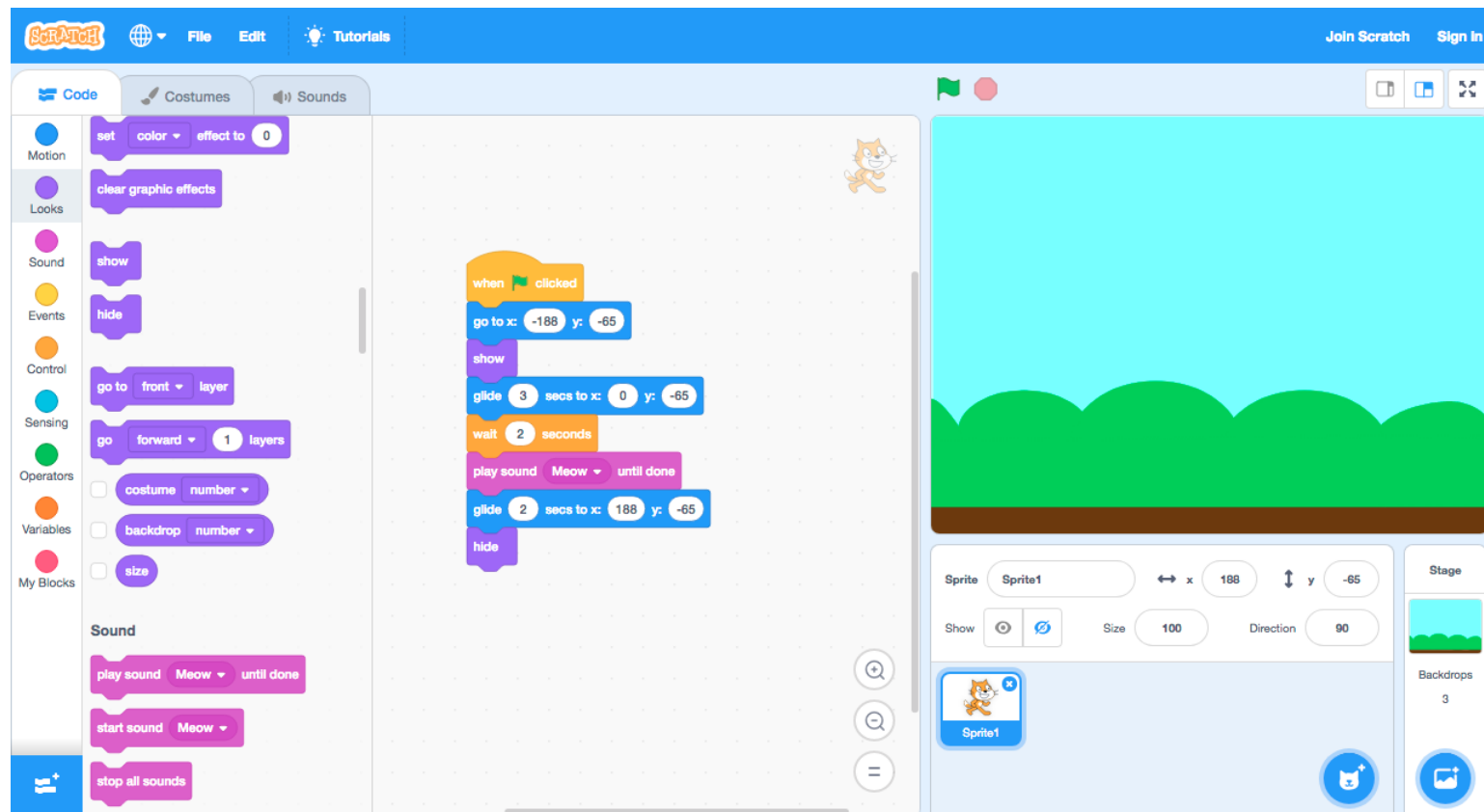
O SCRATCH oferece um conjunto de funções em que podem ser postos em sequencias lógicas para criar um algoritmo

Cada algoritmo é referente a um objeto que está disposta na tela

Gera-se a animação ao executarmos o algoritmo criado

# Linguagem SCRATCH

- Vamos fazer o primeiro programa no scratch!



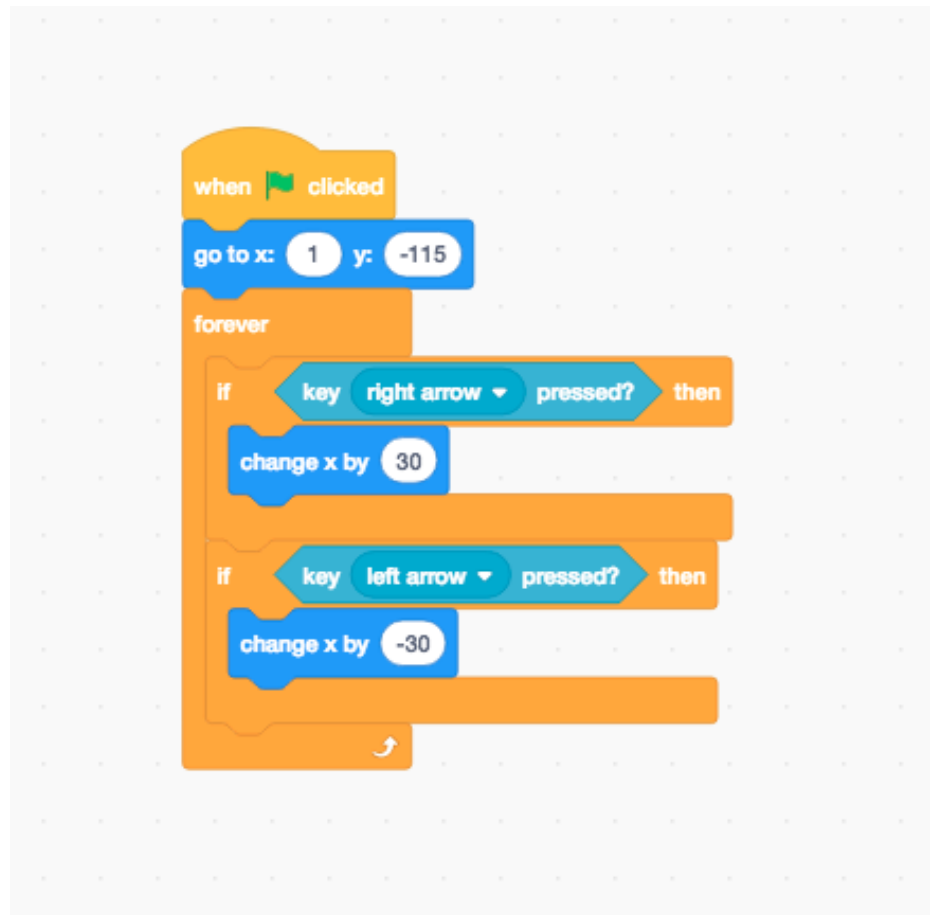
# A linguagem SCRATCH

## Vamos fazer o pequeno jogo usando o SCRATCH!

- O jogo é a captura de maçãs! As maçãs irão cair na parte superior do tela e uma cesta deve ser colocada na posição em que ela cair para contabilizar os pontos. Se a maçã cair no chão, o ponto não é contabilizado.
- Quais são os principais objetos desse jogo?
  - Maça
  - Cesta

# A linguagem SCRATCH

## Código da Cesta:



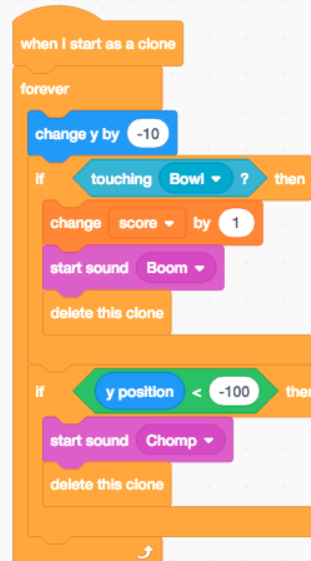


# A linguagem SCRATCH

## Código da Maça:



```
when clicked
  set score to 0
  show
  repeat 30
    go to x: pick random -210 to 210 y: 180
    create clone of myself
    wait pick random 0.1 to 1.5 seconds
  hide
```



```
when I start as a clone
  forever
    change y by -10
    if touching Bowl ? then
      change score by 1
      start sound Boom
      delete this clone
    if y position < -100 then
      start sound Chomp
      delete this clone
```

# Dúvidas?

