



# Pensamento Computacional

Profa. Ana Grasielle Dionísio Corrêa

Profa. Valéria Farinazzo Martins

Faculdade de Computação e Informática/ Programa de Pós-Graduação em  
Distúrbios do Desenvolvimento

Universidade Presbiteriana Mackenzie

# Agenda

---

Introdução/ Motivação

---

Projetos e ferramentas existentes no mundo

---

Projetos e ferramentas existentes no Brasil

---

Estudo de Caso - Scratch

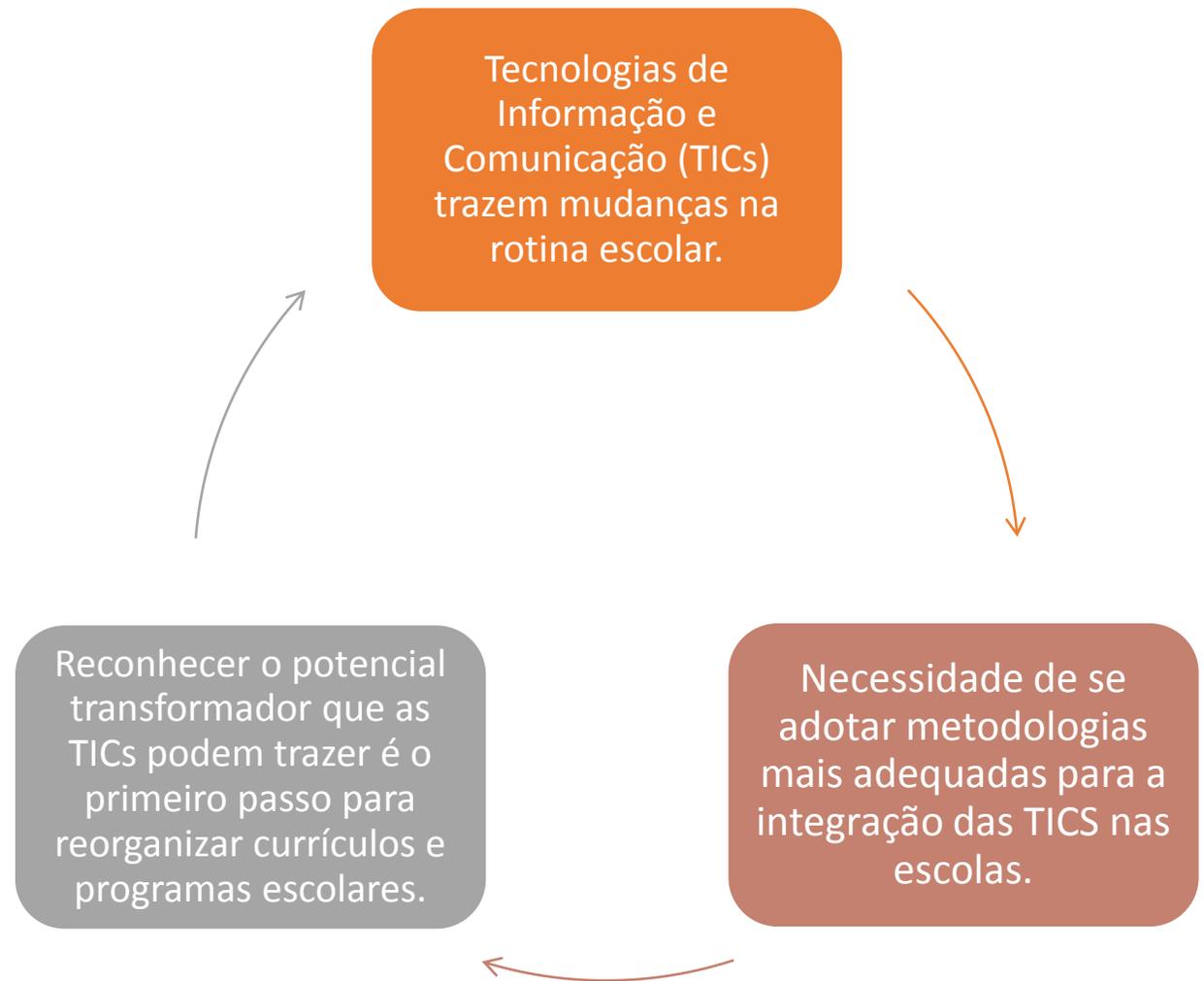
---

App Inventor

---

Conclusões

# Introdução/ Motivação



# Introdução/ Motivação

O objetivo é desenvolver habilidades cognitivas, sociais e profissionais.



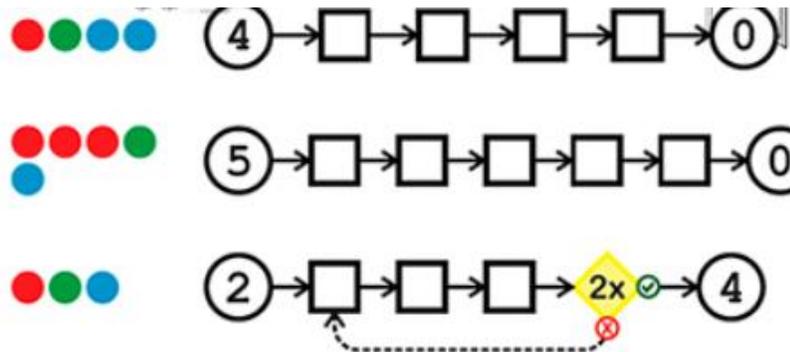
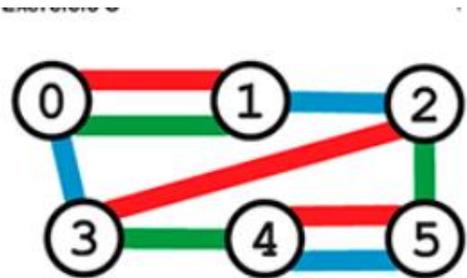
Recursos cognitivos importantes:

raciocínio lógico e a resolução de problemas.

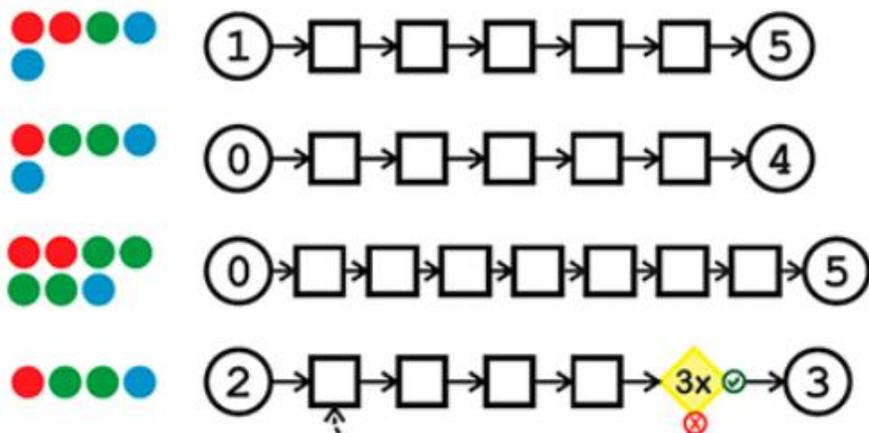
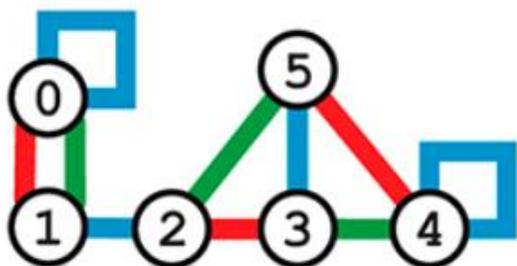


Pensamento computacional:

pensar nos problemas de forma analítica e desenvolver soluções em forma de algoritmos.



Exercício D



# Introdução/ Motivação

- Características do pensamento computacional:
  - formulação de problemas;
  - organização e análise lógica de dados;
  - representação por meio de abstrações;
  - soluções automatizadas por meio de algoritmos;
  - identificação análise e implementação de soluções;
  - generalização e transferência do processo de solução encontrado para resolução de outros problemas.

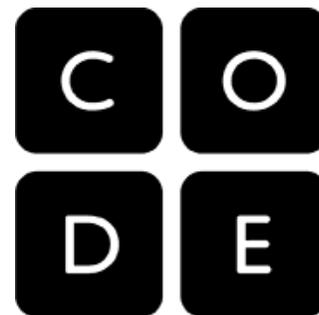
# Introdução/ Motivação

Desafio:

Como levar o pensamento computacional aos alunos de ensino fundamental e médio?



Uso de ferramentas que possam apoiar a aprendizagem de conceitos de lógica de programação de forma mais lúdica e adequada para essa faixa etária.

The Scratch logo, featuring the word "Scratch" in a stylized, orange, bubbly font.The Google CS[4]HS logo, featuring a yellow lightbulb icon followed by the text "Google CS[4]HS" in a grey sans-serif font.The CODE logo, consisting of four black squares arranged in a 2x2 grid, each containing a white letter: 'C' (top-left), 'O' (top-right), 'D' (bottom-left), and 'E' (bottom-right).The CS UNPLUGGED logo, featuring the letters "CS" in a red box, followed by a red line that loops around the word "UNPLUGGED" in a bold, black, sans-serif font. Below it, the text "Computer Science without a computer" is written in a smaller, grey font.The Scratch Jr logo, featuring the word "Scratch" in a bubbly, orange font with a blue outline, followed by "Jr" in a blue, rounded font inside an orange circle.The wonder workshop logo, featuring a stylized orange "w" icon above the word "wonder" in a lowercase, orange, sans-serif font, with "workshop" in a smaller, grey font below it.

Projetos Existentes no Mundo

# O Início

---

- A expressão *pensamento computacional* foi colocada em evidência na comunidade de [ciência da computação](#) como o resultado de um artigo escrito por [Jeannette Wing](#) em 2006.
- O artigo propõe que o pensamento computacional seja uma competência fundamental para qualquer pessoa e não somente para cientistas de computação.
- Ensinar a programar ainda não basta para o desenvolvimento de habilidades relacionadas ao processo de Pensamento Computacional:
  - Mudança nos hábitos e métodos de ensino, baseando-se no [Construcionismo](#).
  - O estudante passa a ser o protagonista de sua aprendizagem.



Jeannette Wing, professora e chefe do Departamento de Ciência da Computação na Universidade de Carnegie Mellon, Pittsburgh.

# Pensamento computacional no contexto do ensino fundamental

Mindstorming: Children, Computers and Powerful Ideas (1980):

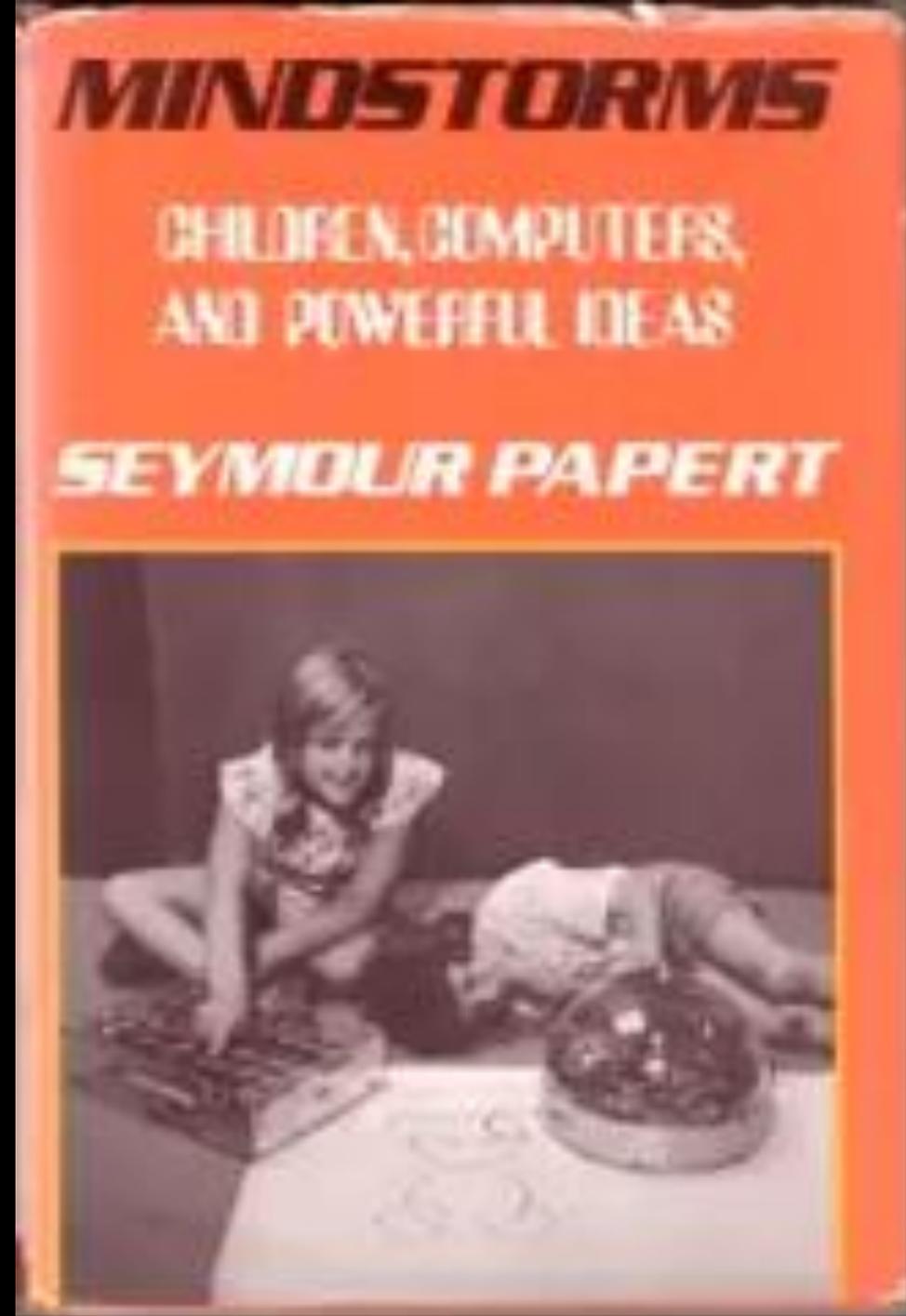
Seymour Papert

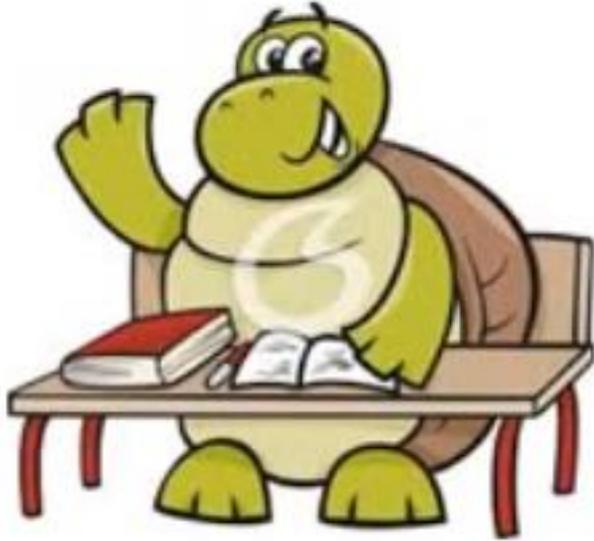
Criação da linguagem de programação logo (Seymour Papert):

“Low floor, high ceiling”

Melhorar a maneira como crianças pensam e resolvem problemas

Scratch, Game Maker, App Inventor, Arduino, etc.





# A Linguagem Logo

---

- A tartaruga é o cursor gráfico.
- O aluno programa a tartaruga para ir para frente ou para trás, girar para esquerda ou direita.
- O aluno pode somar, subtrair, dividir e multiplicar.
- O aluno desenha, apaga, erra e depura o erro.
- O aluno aprende a pensar.



forward 50



right 90



forward 50



right 90

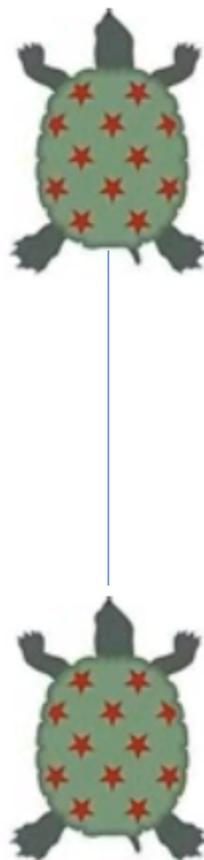
# A Linguagem Logo

**Vamos fazer a tartaruga se mover para criar um quadrado**



# A Linguagem Logo

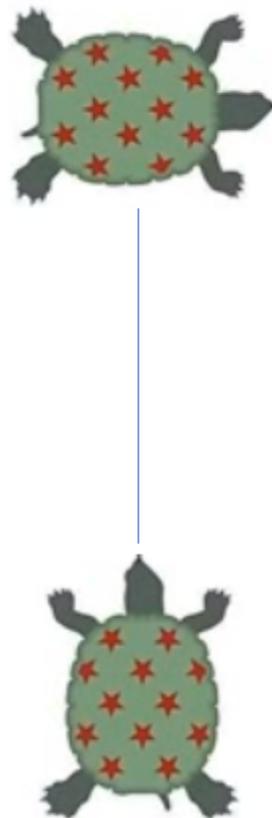
**PF 40 (para frente 40 passos)**



# A Linguagem Logo

**PF 40 (para frente 40 passos)**

**GD 90 (gire à direita 90 graus)**

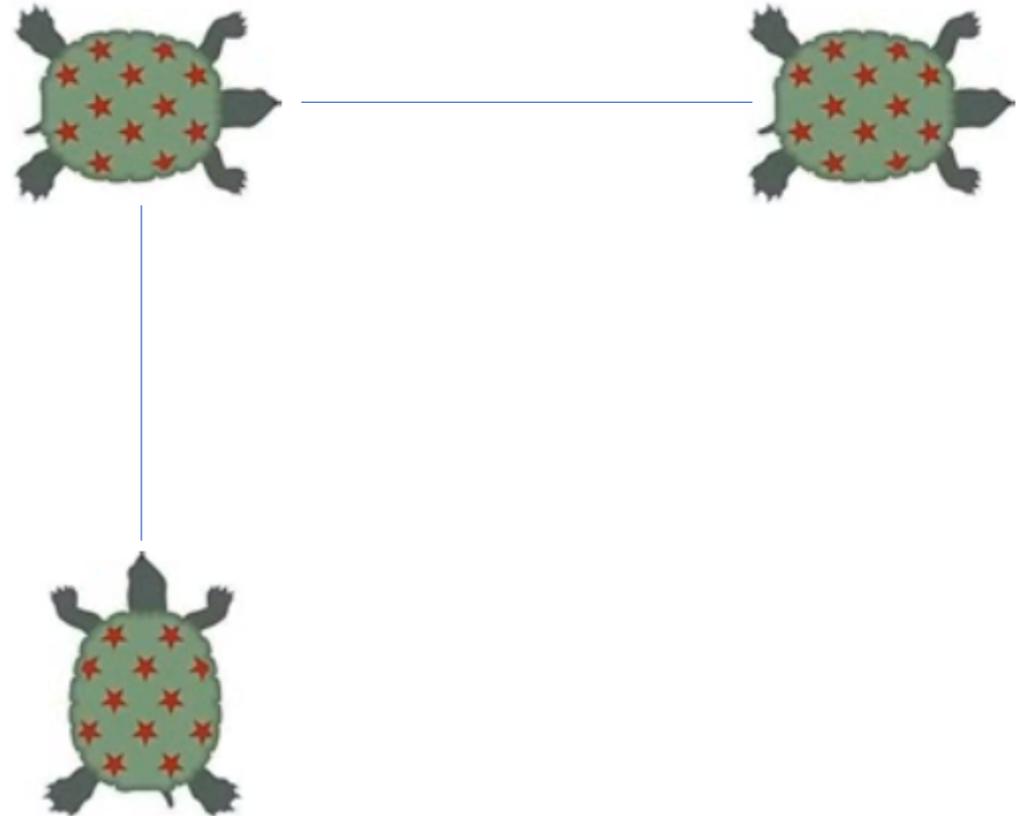


# A Linguagem Logo

**PF 40 (para frente 40 passos)**

**GD 90 (gire à direita 90 graus)**

**PF 40 (para frente 40 passos)**



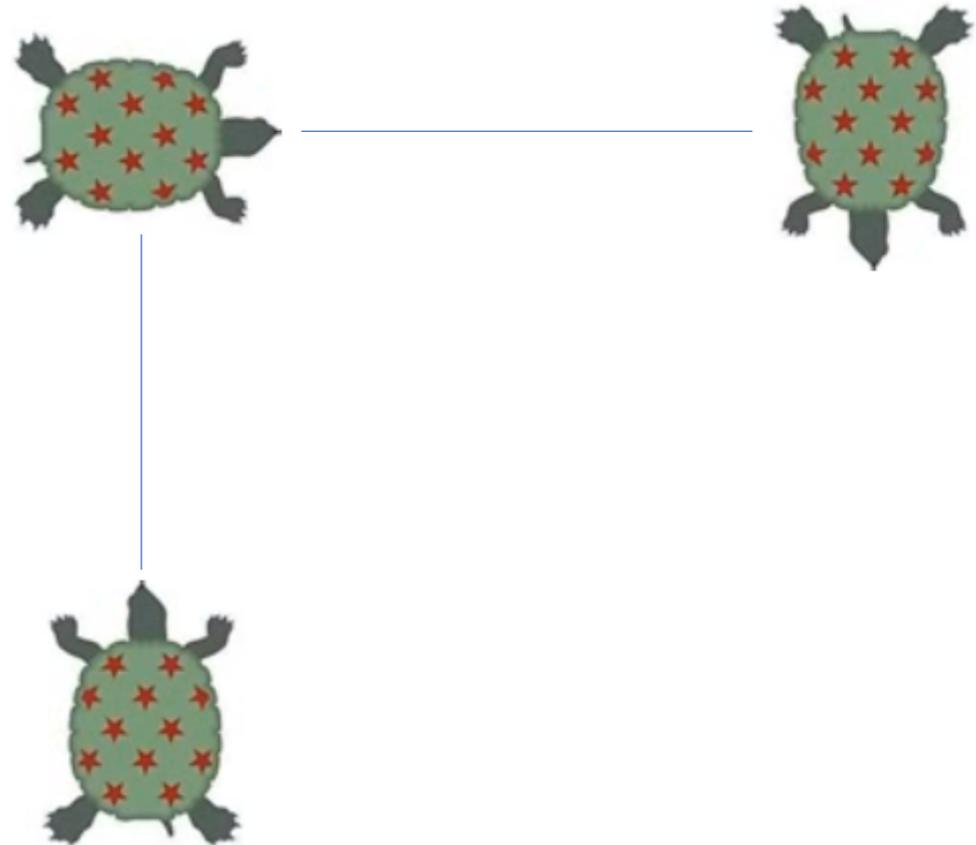
# A Linguagem Logo

**PF 40 (para frente 40 passos)**

**GD 90 (gire à direita 90 graus)**

**PF 40 (para frente 40 passos)**

**GD 90 (gire à direita 90 graus)**



# A Linguagem Logo

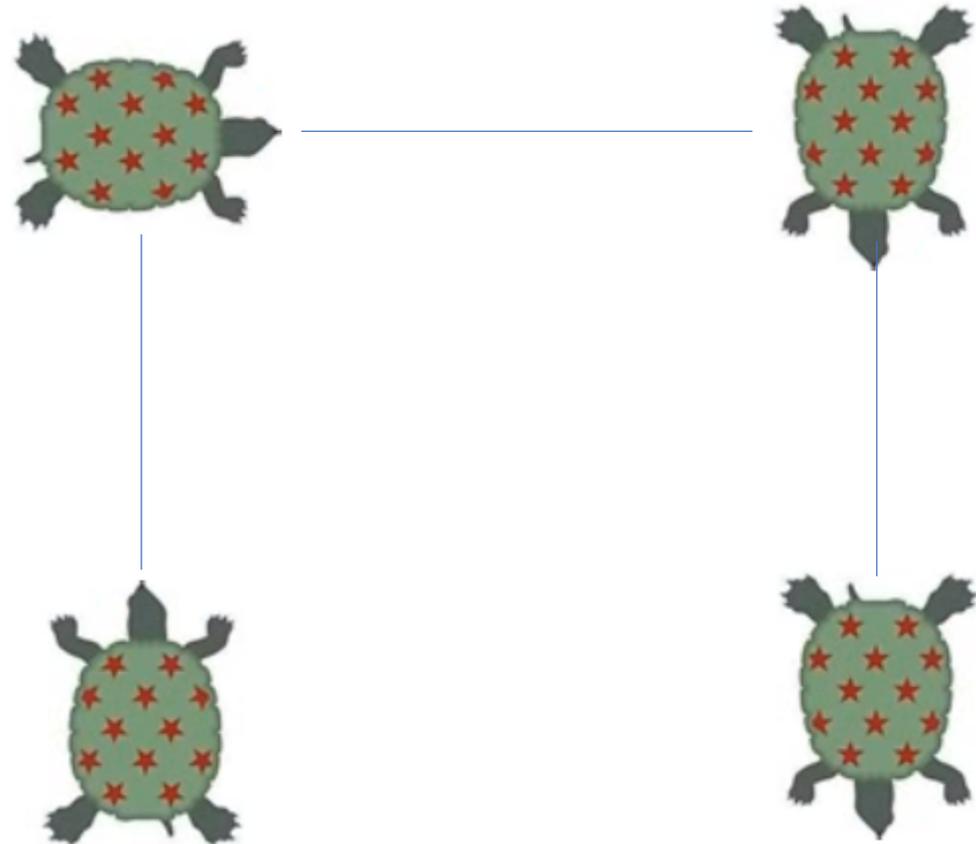
**PF 40 (para frente 40 passos)**

**GD 90 (gire à direita 90 graus)**

**PF 40 (para frente 40 passos)**

**GD 90 (gire à direita 90 graus)**

**PF 40 (para frente 40 passos)**



# A Linguagem Logo

**PF 40 (para frente 40 passos)**

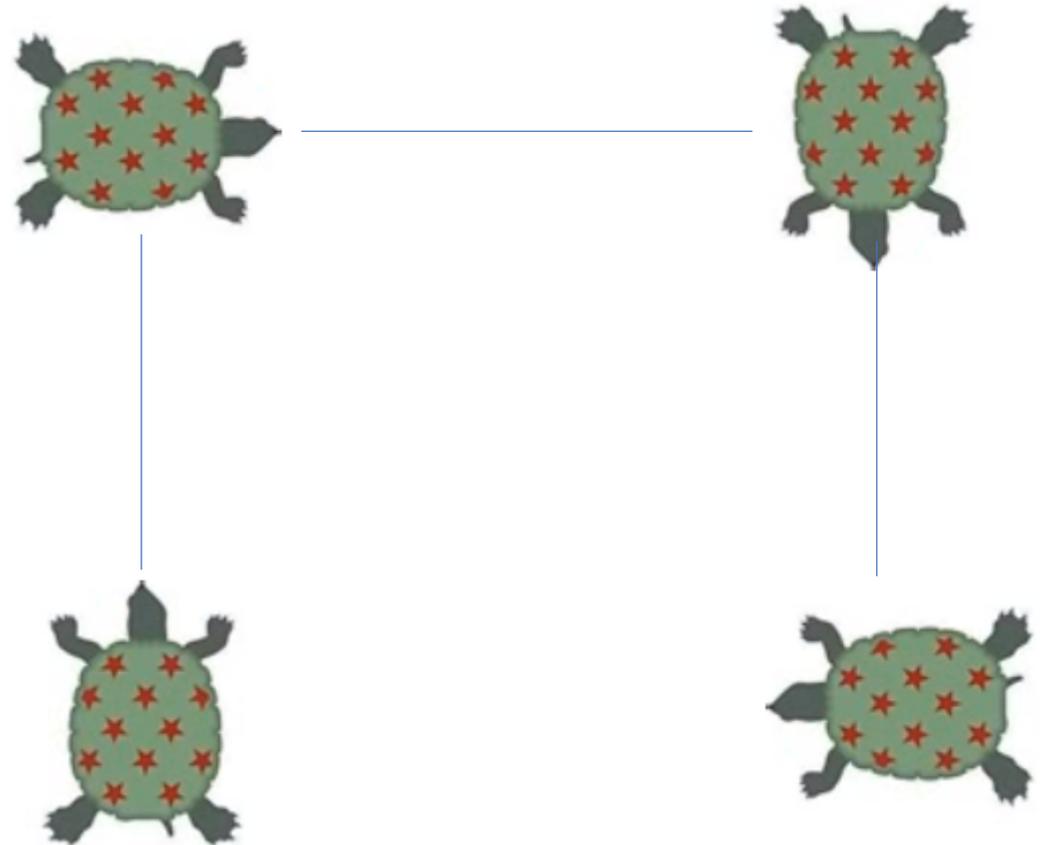
**GD 90 (gire à direita 90 graus)**

**PF 40 (para frente 40 passos)**

**GD 90 (gire à direita 90 graus)**

**PF 40 (para frente 40 passos)**

**GD 90 (gire à direita 90 graus)**



# A Linguagem Logo

**PF 40 (para frente 40 passos)**

**GD 90 (gire à direita 90 graus)**

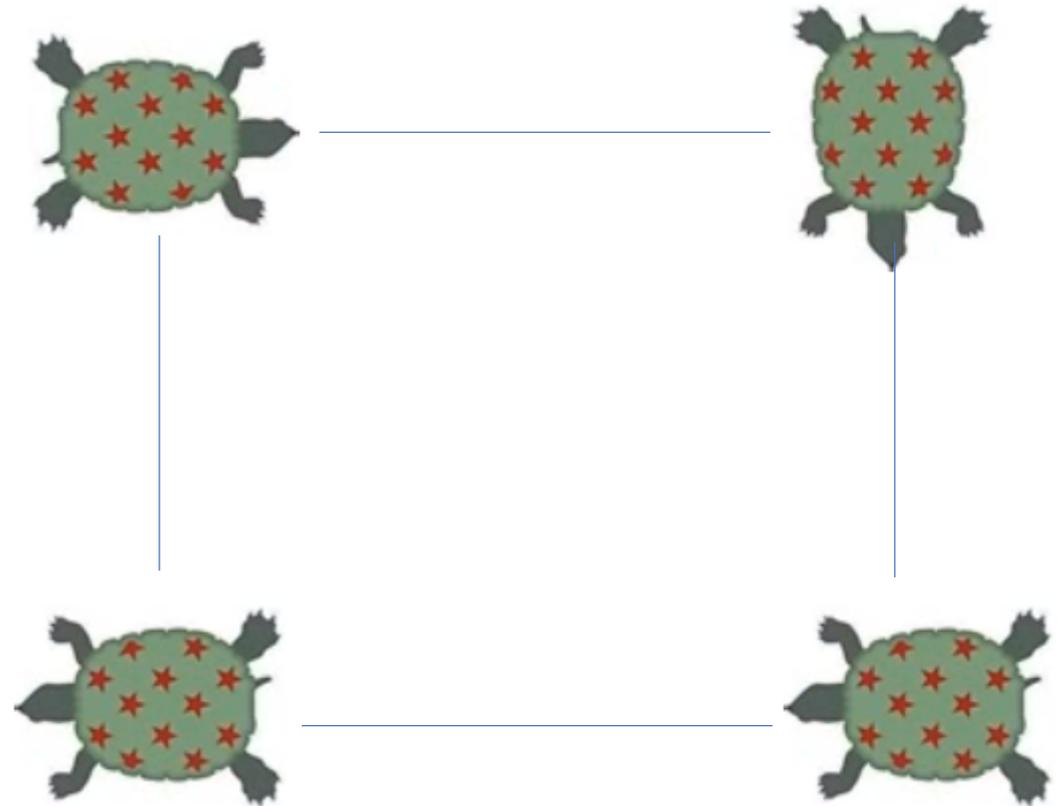
**PF 40 (para frente 40 passos)**

**GD 90 (gire à direita 90 graus)**

**PF 40 (para frente 40 passos)**

**GD 90 (gire à direita 90 graus)**

**PF 40 (para frente 40 passos)**



# A Linguagem Logo

Observem quantos comandos o aluno teve que pensar para fazer a tartaruga se mover e desenhar um quadrado:

**PF 40**

**GD 90**

**PF 40**

**GD 90**

**PF 40**

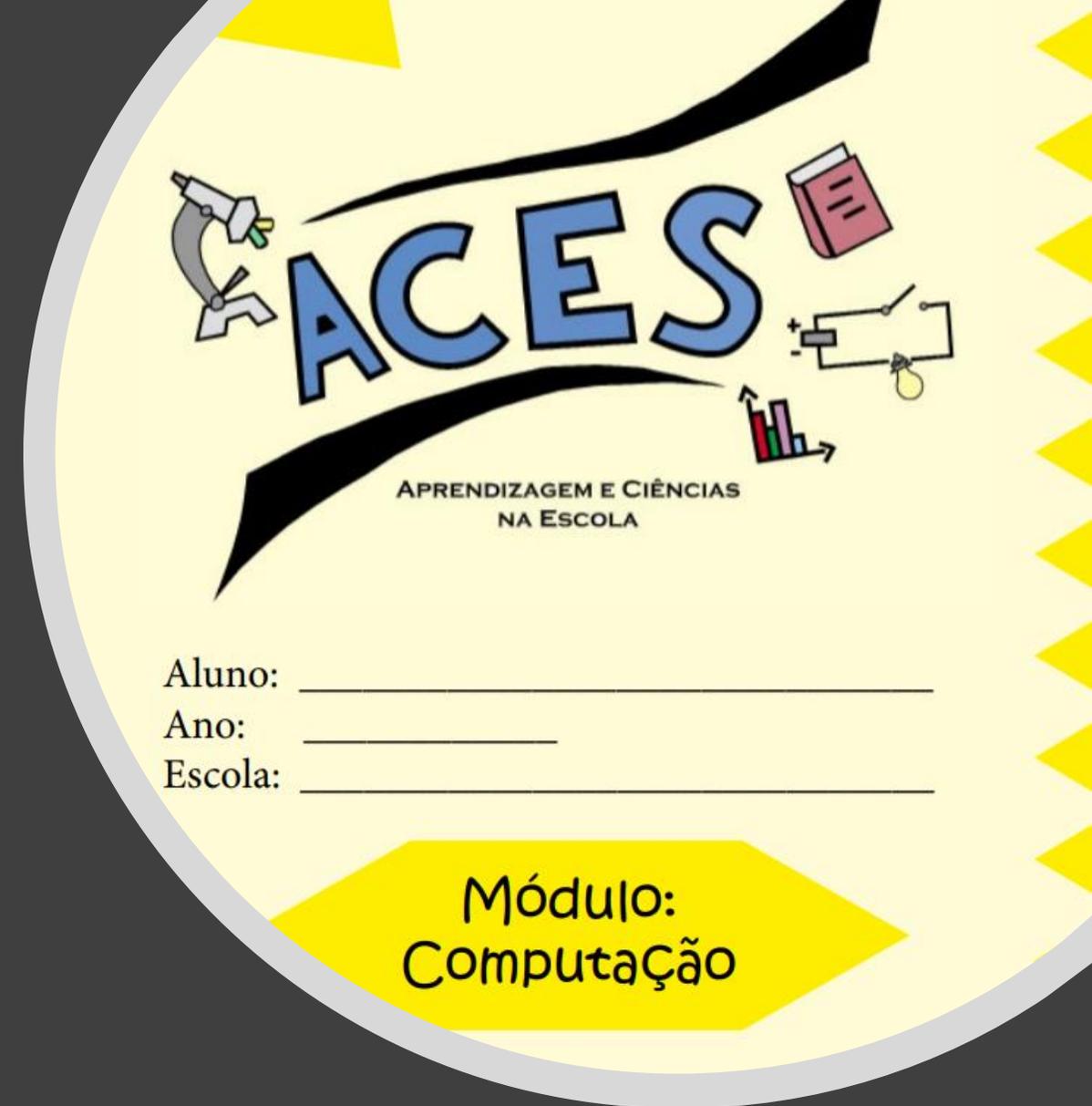
**GD 90**

**PF 40**

Esta é a descrição do percurso percorrido pelo pensamento para conceber um conceito, no caso, um quadrado.

# ACES – Aprendizagem e Ciência na Escola

- Programa ACES (UNICAMP):
  - Aprendizagem e Ciência na Escola
- Atividades desplugadas (off-line):
  - Despertar o interesse dos alunos
  - Fornecer suporte aos professores
- Computer Science Unplugged



Aluno: \_\_\_\_\_  
Ano: \_\_\_\_\_  
Escola: \_\_\_\_\_

Módulo:  
Computação

# 1ª Parte - Introdução à Computação

- Como você descreveria o que é computação? Ela existe sem um computador?
- Como você acha que um programa de computador funciona? Pense num programa que resolva o seguinte problema: Somar os números 8 e 6.
- Que matérias escolares você acha que se associam à computação? Que tipos de trabalhos podem se beneficiar da computação?



## O que é um algoritmo?

- Um algoritmo é uma sequência de passos ordenados necessários a um programa para que ele resolva determinado problema, como o de somar, subtrair, dividir, multiplicar números, procurar por palavras num texto, etc.

### Algoritmo

Como você pôde perceber a partir da discussão anterior, um algoritmo é como uma receita para o computador seguir. Por exemplo:

Calcular a seguinte conta:  $3 + 4 - 2 + 6 =$

1° Passo: Calcular  $3 + 4 = 7$

2° Passo: Calcular  $7 - 2 = 5$

3° Passo: Calcular  $5 + 6 = 11$

Tudo isso é um algoritmo.

- — mova um quadrado para a direita
- ← — mova um quadrado para a esquerda
- ↓ — mova um quadrado para baixo
- ↑ — mova um quadrado para cima
- ↻ — mude para a próxima cor
- ⚡ — pinte o quadrado

Imagem 1	Algoritmo	Código
 <p data-bbox="1355 644 1505 668">fonte: code.org</p>		

# Atividades desplugadas

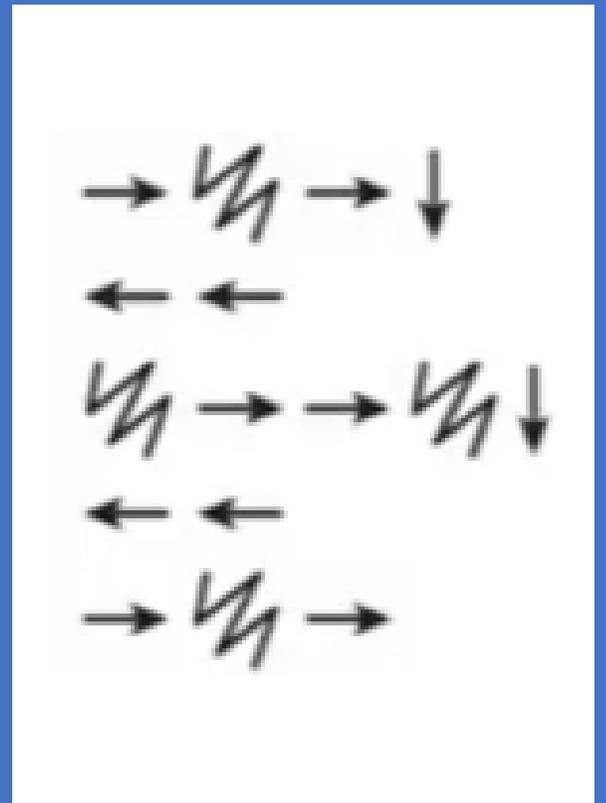
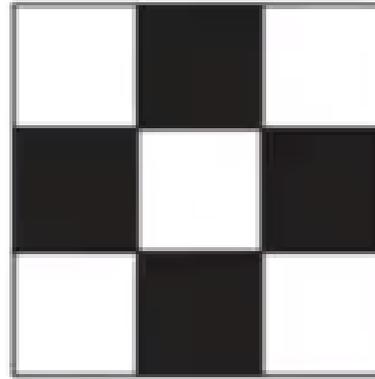
Criação de códigos e algoritmos através de desenhos e vice-versa.

# Atividades desplugadas

---

Algoritmo:

- Mova para a direita, pinte o quadrado, mova para a direita, mova para baixo.
- Mova para a esquerda, mova para a esquerda.
- Pinte o quadrado, mova para a direita, mova para a direita, pinte o quadrado, mova para baixo.
- Mova para a esquerda, mova para a esquerda.
- Mova para a direita, pinte o quadrado, mova para a direita.



# Atividades desplugadas

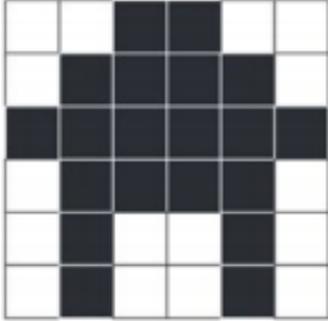
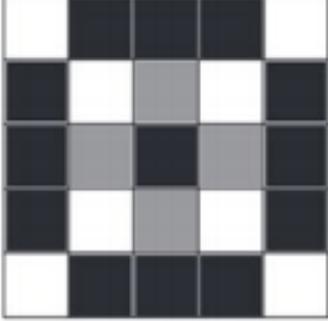
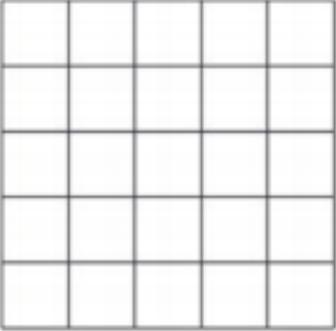
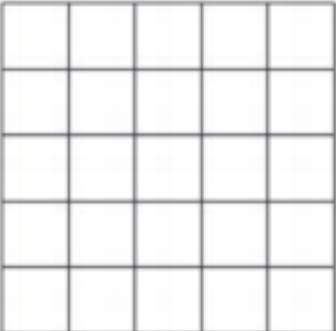
Imagem 3	Algoritmo	Código
 <p data-bbox="1090 625 1276 658">fonte: code.org</p>		

Imagem 4	Algoritmo	Código
 <p data-bbox="1116 1089 1416 1153">■ Cor 1   ■ Cor 2</p> <p data-bbox="1090 1189 1276 1222">fonte: code.org</p>		

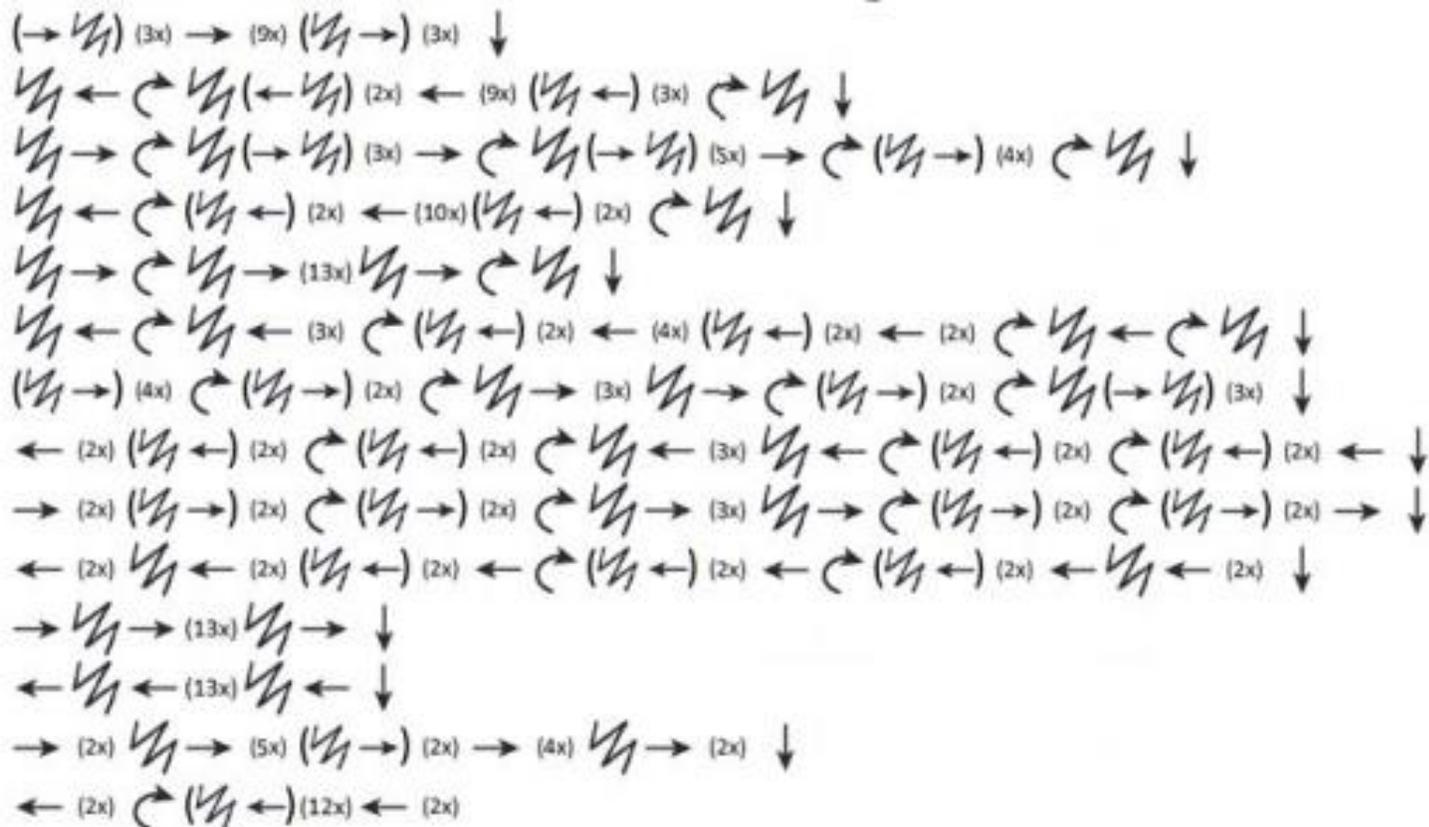
# Atividades desplugadas

<p>Crie um desenho para outro grupo</p> 	<p>Algoritmo</p>	<p>Código (forneça-o a outro grupo)</p>
<p>Código (fornecido pelo outro grupo)</p>	<p>Desenho obtido através do código fornecido</p> 	

Quando terminam os algoritmos e códigos das imagens, cada grupo cria um desenho e fornece a outro grupo o código para que esse desenho seja feito.

# Atividades desplugadas

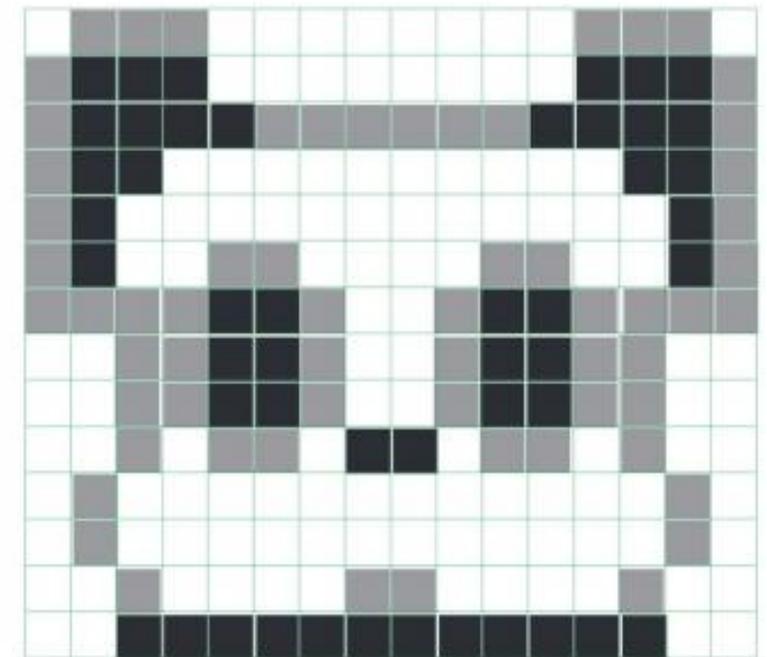
**Desafio:** criar o desenho do código abaixo.



# Atividades desplugadas

**Desafio:** criar o desenho do código abaixo.

(→ ↘) (3x) → (9x) (↘ →) (3x) ↓  
↘ ← ↻ ↘ (← ↘) (2x) ← (9x) (↘ ←) (3x) ↻ ↘ ↓  
↘ → ↻ ↘ (→ ↘) (3x) → ↻ ↘ (→ ↘) (5x) → ↻ (↘ →) (4x) ↻ ↘ ↓  
↘ ← ↻ (↘ ←) (2x) ← (10x) (↘ ←) (2x) ↻ ↘ ↓  
↘ → ↻ ↘ → (13x) ↘ → ↻ ↘ ↓  
↘ ← ↻ ↘ ← (3x) ↻ (↘ ←) (2x) ← (4x) (↘ ←) (2x) ← (2x) ↻ ↘ ← ↻ ↘ ↓  
(↘ →) (4x) ↻ (↘ →) (2x) ↻ ↘ → (3x) ↘ → ↻ (↘ →) (2x) ↻ ↘ (→ ↘) (3x) ↓  
← (2x) (↘ ←) (2x) ↻ (↘ ←) (2x) ↻ ↘ ← (3x) ↘ ← ↻ (↘ ←) (2x) ↻ (↘ ←) (2x) ← ↓  
→ (2x) (↘ →) (2x) ↻ (↘ →) (2x) ↻ ↘ → (3x) ↘ → ↻ (↘ →) (2x) ↻ (↘ →) (2x) → ↓  
← (2x) ↘ ← (2x) (↘ ←) (2x) ← ↻ (↘ ←) (2x) ← ↻ (↘ ←) (2x) ← ↘ ← (2x) ↓  
→ ↘ → (13x) ↘ → ↓  
← ↘ ← (13x) ↘ ← ↓  
→ (2x) ↘ → (5x) (↘ →) (2x) → (4x) ↘ → (2x) ↓  
← (2x) ↻ (↘ ←) (12x) ← (2x)



■ Cor 1    ■ Cor 2

fonte: code.org

# Atividades Online

The screenshot displays the Code.org 'Stage 1: Puzzle' interface. At the top, there's a teal header with 'C O D E' icons and 'Stage 1: Puzzle' text. Below this, a progress bar shows a yellow dot followed by several purple dots. To the right, it says 'Trophies 3 of 27' with a 'MORE' link. The main area is a 10x10 grid with a red bird, a green pig, and TNT. A 'Run Program' button is at the bottom left. The right side shows a 'Show Code' button and a block palette with 'move forward', 'turn right', and 'turn left' blocks. A trash icon is also visible.

Atividades online são sugeridas para complementar as atividades off-line

Permite que os alunos trabalhem os conceitos fora da sala de aula.

Permite colocar em prática os conceitos de algoritmo e sequência.

<https://studio.code.org/hoc/1>



## 2ª Parte - Condicional e Repetição

- Conceitos passados através de dinâmicas divertidas:
  - Dinâmica para o conceito de condicional
  - Dinâmica para o conceito de repetição
  - Dinâmica para a introdução de pseudocódigos.

Pseudocódigo - origem da palavra

**PSEUDO + CÓDIGO**

Pseudo: Falso, imitação.

Código: Conjunto de símbolos que têm um determinado valor e que, combinados, podem formar uma mensagem.

# Pseudocódigos

# Pseudocódigos

---

Para condições do tipo "Se": Se <Condição> então <Comandos> FimSe	Para condições do tipo "Se... Senão": Se <Condição> então <Comandos> Senão <Comandos2> FimSe	Para condições ligadas: Se <Condição> então <Comandos> Senão se <condição2> então <Comandos2> Senão <Comandos3> FimSe
Para repetições do tipo "Enquanto": Enquanto <Condição> faça <Comandos> FimEnquanto	Para repetições do tipo "Repita... até": Repita <Comandos> Até <Condição>	
Para repetições do tipo "Para": Para <Contagem> - <Início> até <Fim> faça <Comandos> FimPara		

- Para condições do tipo “Se”:

Se o dia for uma segunda-feira então

    Todos os alunos devem bater palmas uma vez.

FimSe

- Para condições do tipo “Se... Senão”:

Se o dia for uma segunda-feira então

    Todos os alunos devem bater palmas uma vez.

Senão

    Todos os alunos devem bater palmas duas vezes.

FimSe

# Pseudocódigos

---

# Pseudocódigos

---

- Para condições encadeadas:

Se o dia for uma segunda-feira então

    Todos os alunos devem bater palmas uma vez.

Senão se o dia for uma terça-feira então

    Todos os alunos devem bater palmas duas vezes.

Senão

    Todos os alunos devem bater palmas três vezes.

FimSe

- Para repetições do tipo “Enquanto”:

Enquanto o(a) professor(a) escreve na lousa faça

    Todos os alunos devem copiar em seus cadernos.

FimEnquanto

- Para repetições do tipo “Repita... até”:

Repita

    Todos os alunos ficam em silêncio

Até o professor fazer uma pergunta

- Para repetições do tipo “Para”:

Para dias = segunda-feira até sexta-feira faça

    Os alunos vão à escola

FimPara

# Pseudocódigos

---

# Pseudocódigos

- Conceitos de condicional:
  - Alunos divididos em dois grupos para que uma competição fosse criada.
  - As regras são escritas em pseudocódigo

“Se a carta for vermelha, o grupo 1 ganha 10 pontos, caso contrário, o grupo 2 ganha 10 pontos.”

**Se carta for vermelha então**

**Grupo 1 ganha 10 pontos**

**Senão**

**Grupo 2 ganha 10 pontos**

**FimSe**



= Ouros



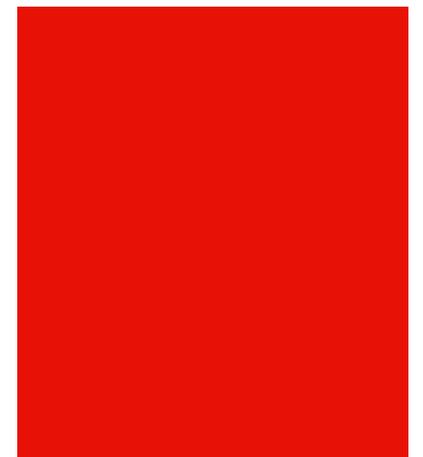
= Espadas



= Copas



= Paus



# Pseudocódigos

- Conceitos de condicional e repetição:
  - Alunos divididos em quatro grupos
  - Cada grupo representa um naipe do baralho

**Enquanto número de rodadas for menor do que 20  
então**

**Vire uma nova carta**

**FimEnquanto**

**Se o naipe da carta for Ouro então**

**Grupo Ouro deve bater palmas por 10  
segundos**

**FimSe**

**Se o grupo cumpriu uma tarefa então**

**O grupo ganha 10 pontos**

**Senão**

**O grupo perde 5 pontos**

**FimSe**



= Ouros



= Espadas



= Copas



= Paus

# Atividades Online

Fase 9: A Fazendeira 4

Você pode me ajudar a remover todos os quatro montes de terra? Dica: se puder, use um bloco de repetição.

Blocos Área de trabalho: 1 / 5 blocos

quando executar

avance

vire à esquerda

vire à direita

remova 1

preencha 1

repita 5 vezes

faça

▶ Executar

**Precisa de ajuda?** Veja os vídeos e as dicas

Introdução à Fazendeira

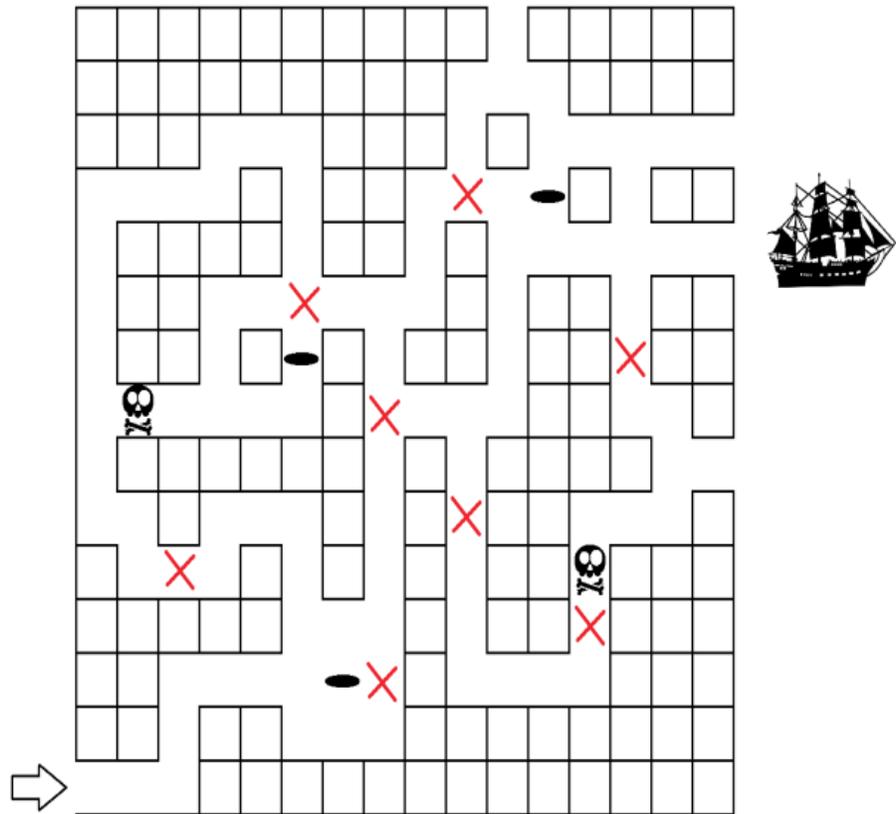
Bloco repita

Atividades online são sugeridas para complementar as atividades off-line

Permite que os alunos trabalhem os conceitos fora da sala de aula.

Permite colocar em prática os conceitos de algoritmo e sequência.

<https://studio.code.org/s/20-hour/stage/9/puzzle/1>



São piratas malvados, se encontrar um deles, você deve enfrentá-lo.



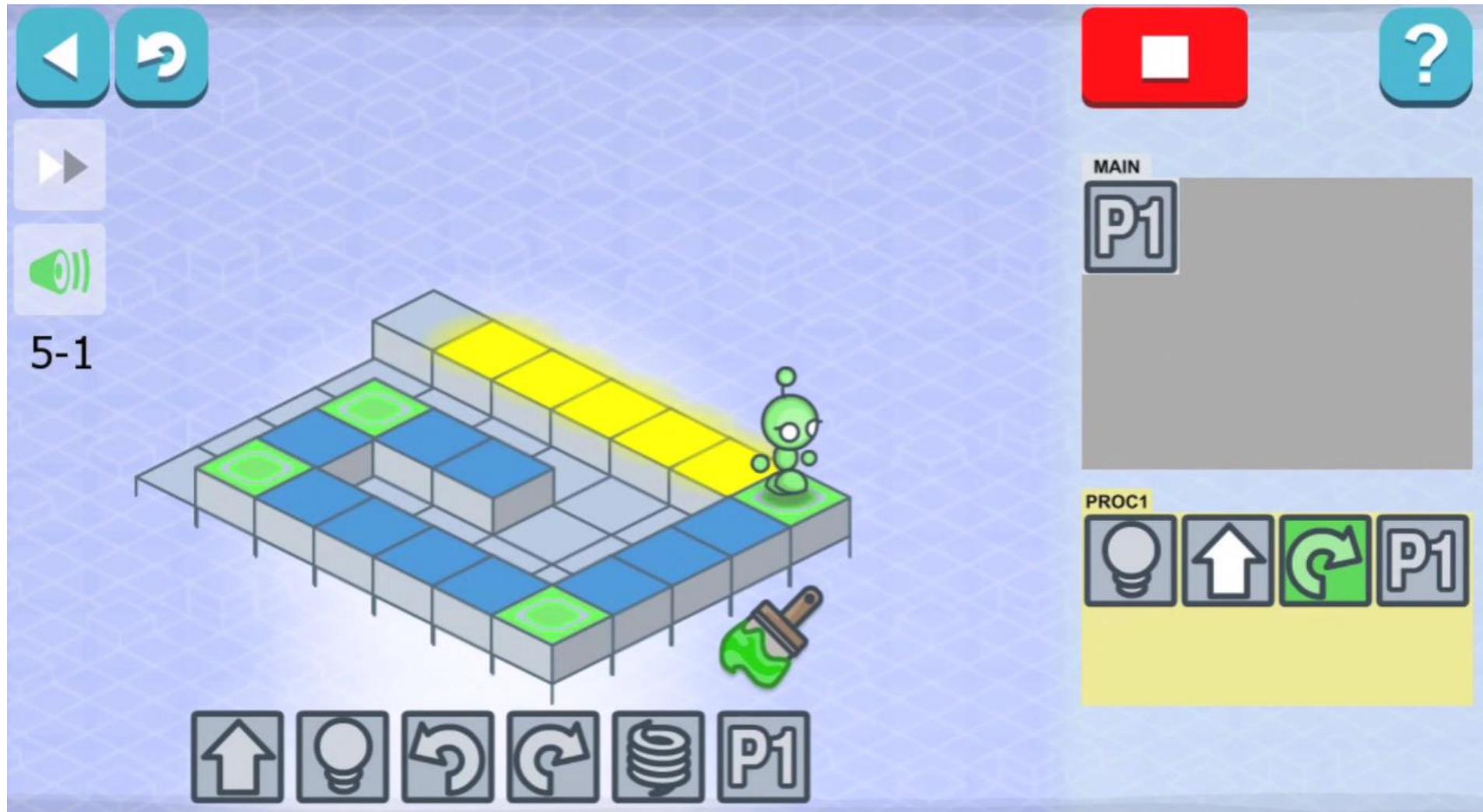
São tesouros, se encontrar um deles, você deve desenterrar e recuperá-lo.



São buracos fundos, se encontrar um deles, você deve mudar seu caminho.

## 3ª Parte - Labirinto de decisões

# Outras atividades online - Lightbot



# Outras atividades online - Lightboot



How does Lightbot teach 'real' programming?

The 'Hour of Code' is nationwide initiative by Computer Science Education Week [cseweek.org] and Code.org [code.org] to introduce millions of students to one hour of computer science and computer programming.



# Estudo de caso - Scratch

Criação de um projeto de extensão dentro  
do Programa Caminhando Juntos

# Objetivos

- Proporcionar aos alunos conhecimentos básicos de lógica de programação para a criação de animações e jogos interativos através da ferramenta Scratch

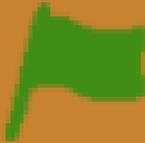


- Software para ensino de programação para crianças.
- Destinado a desenvolver a criatividade.
- Criar histórias, animações e jogos.
- Criado em 2007 pelo Media Lab do MIT.

Scratch

# Scratch

- O Scratch é mais acessível que outras linguagens de programação, por fazer uso de uma interface gráfica que permite que programas sejam criados através de blocos de montar, lembrando o brinquedo lego.

when  clicked

turn in direction

say

say sound

steps

when touching  ?

degrees

# Scratch

Competência  
artística:

Desenho artístico ou histórias

Competência  
matemática:

Noção de algoritmos; coordenadas e  
números aleatórios; variáveis e  
expressões booleanas

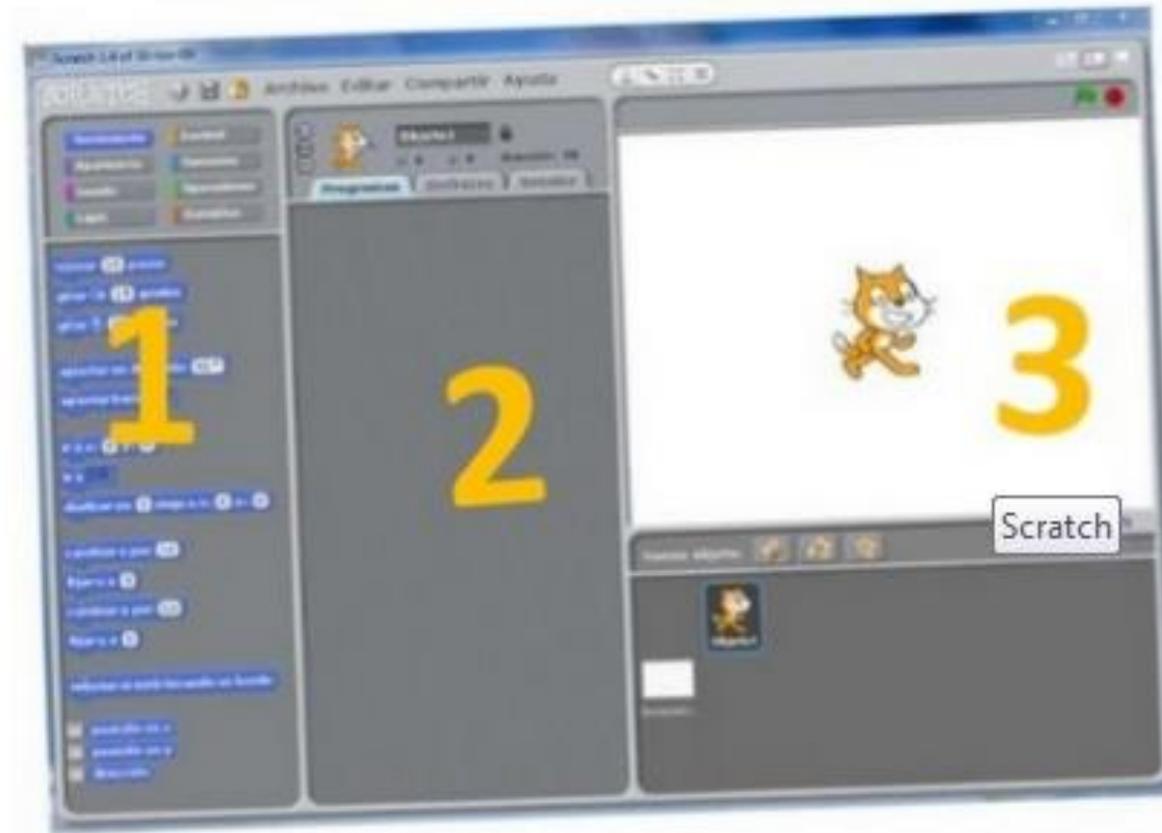
Competências  
digitais:

Introdução à programação

Competência  
para aprender a  
aprender  
autonomia e  
iniciativa pessoal:

Resolução de problemas, improvisar,  
ensaiar e corrigir erros

# Elementos do Ambiente Scratch



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

# Aula 01

---

Objetivo

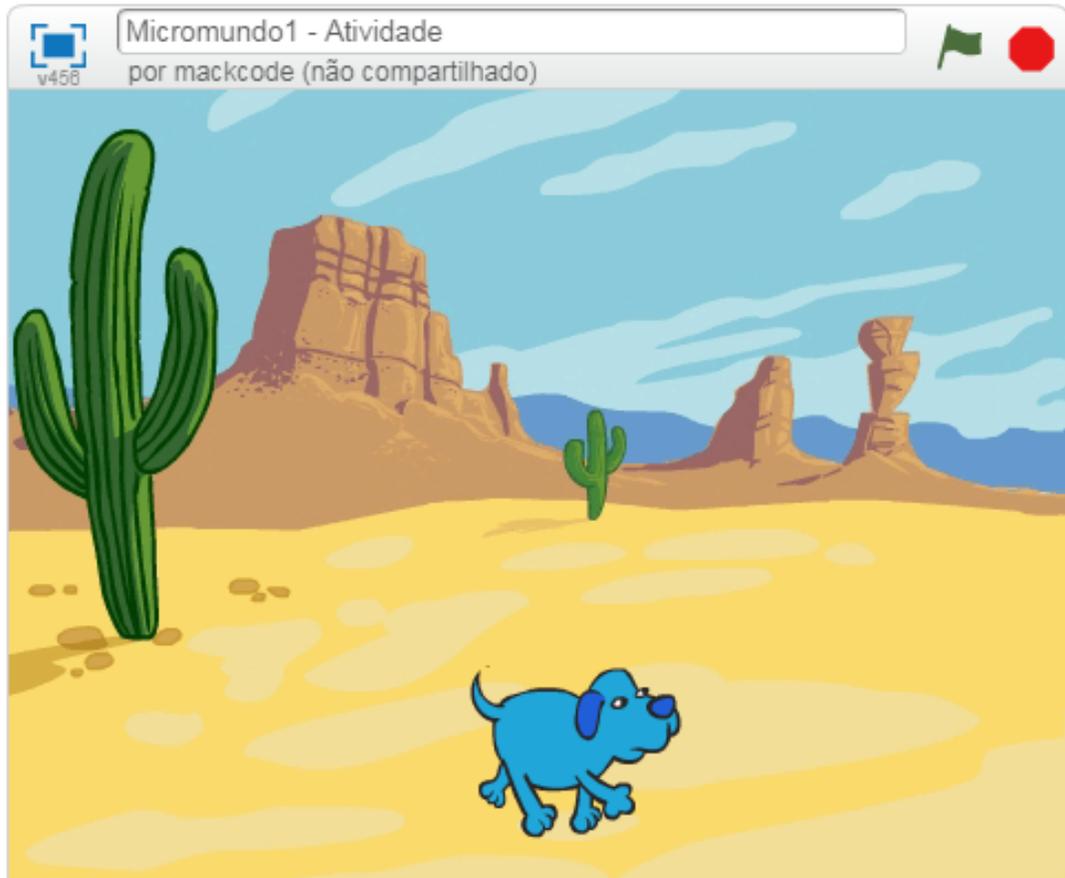
Criar uma animação simples

---

Competências matemáticas e digitais:

Noções de algoritmos, introdução à programação e resolução de problemas.

---



Atores

Novo ator: 🎭 ✂️ 📁 📷

Palco  
2 panos de fundo

Novo pano de fundo

📷 ✂️ 📁 📷

Dog2

Scripts Fantacias Sons

- Movimento
- Aparência
- Som
- Caneta
- Variáveis
- Eventos
- Controle
- Sensores
- Operadores
- Mais Blocos

mova 10 passos

gire 15 graus

gire 15 graus

aponte para a direção 90 graus

aponte para ponteiro do mouse

vá para x: 17 y: -115

vá para ponteiro do mouse

deslize por 1 seg até x: 17 y: -115

adicione 10 a x

mude x para 0

adicione 10 a y

mude y para 0

se tocar na borda, volte

próxima fantasia

quando clicar em

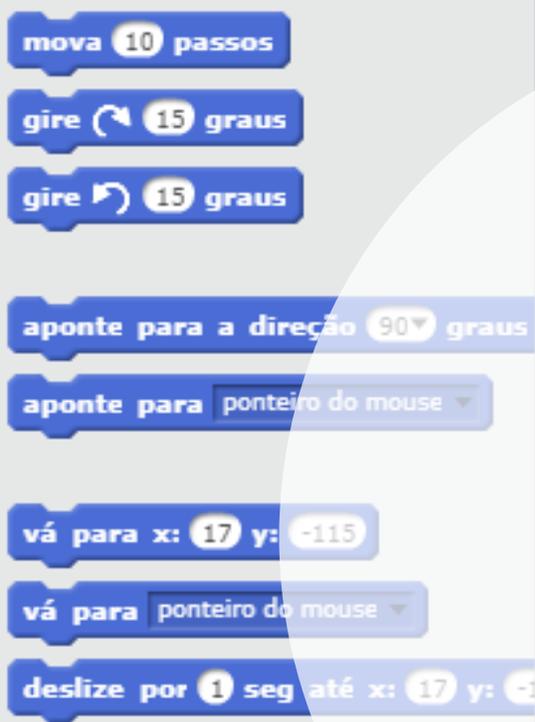
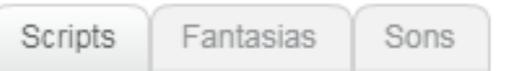
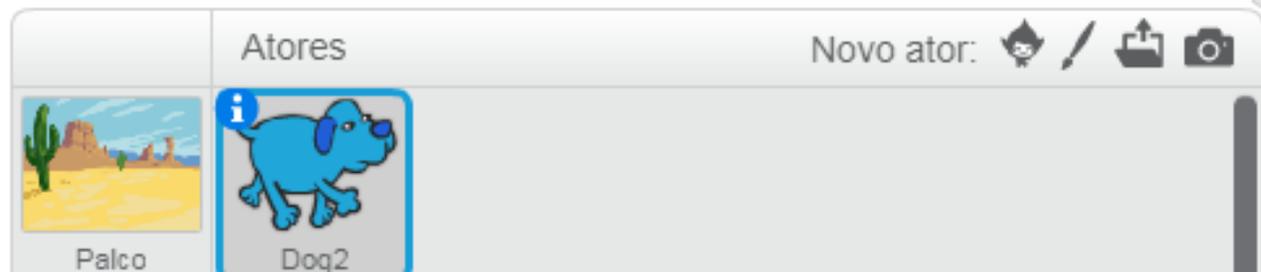
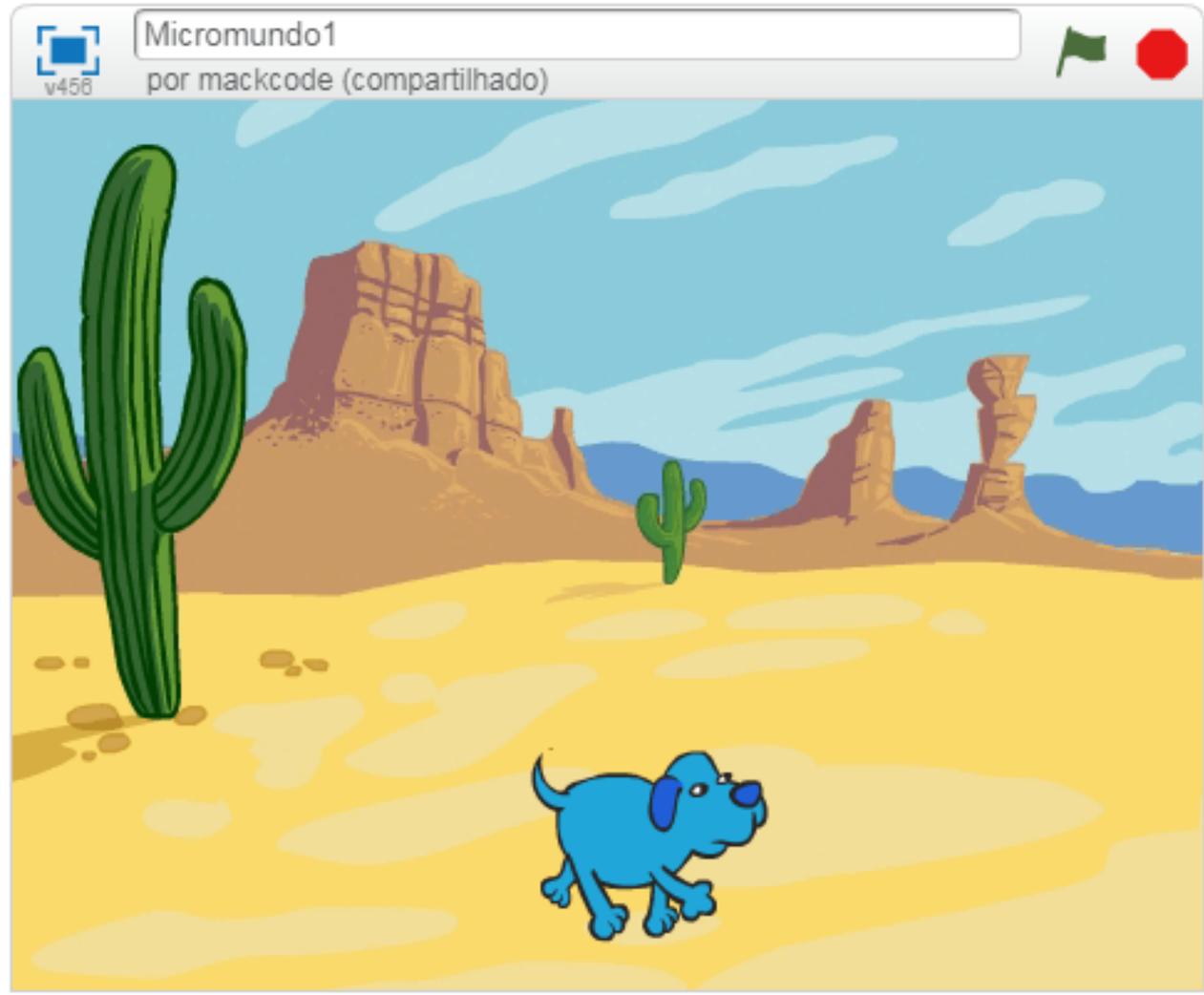
diga Hello! por 0.2 segundos

se tocar na borda, volte

sempre

mova 10 passos

# Aula 01



# Aula 01

# Aula 02

---

Objetivo

Criar uma animação com diálogos e sons.

---

Competências matemáticas e digitais:

Noções de algoritmos, introdução à programação e resolução de problemas.

---



Micromundo2  
por mackcode (compartilhado)

Atores

Novo ator: 🗑️ ✂️ 📁 📷

Palco  
2 panos de fundo

Novo pano de fu

Tera Cat1

X: 240 y: -180

Scripts Fantasia Sons

- Movimento
- Aparência
- Som
- Caneta
- Variáveis
- Eventos
- Controle
- Sensores
- Operadores
- Mais Blocos

```

mova 10 passos
gire 15 graus
gire 15 graus
aponte para a direção 90 graus
aponte para ponteiro do mouse
vá para x: -160 y: -100
vá para ponteiro do mouse
deslize por 1 seg até x: -160 y:
adicione 10 a x
mude x para 0
adicione 10 a y
mude y para 0
    
```

quando clicar em

mude para a fantasia Tera-a

mude para a fa

deslize por 1 seg até x: -80 y:

# Aula 02

vá para x: -180 y: -100

envie Oi

diga Oi gatinho! por 2 segundos



```
quando clicar em   
vá para x: 180 y: -106  
deslize por 1 seg até x: 30 y: -114  
  
quando receber Oi  
toque o som meow  
diga Miau! por 1 segundos
```



```
quando clicar em   
vá para x: -180 y: -100  
mude para a fantasia tera-a  
deslize por 1 seg até x: -80 y: -100  
mude para a fantasia tera-b  
diga Oi gatinho! por 2 segundos  
envie Oi a todos
```

# Aula 02

---

# Aula 03

---

Objetivo

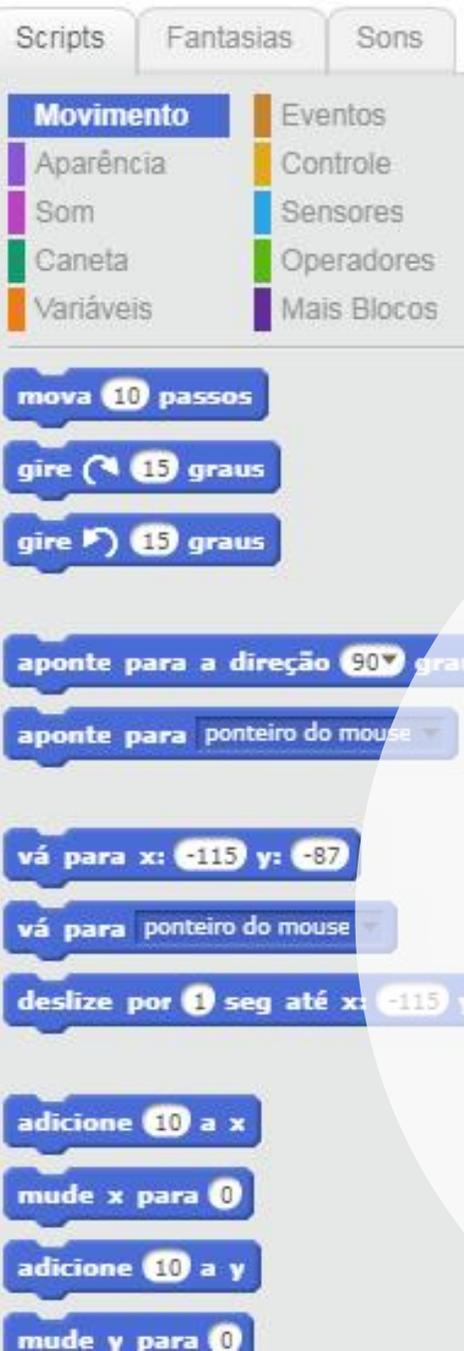
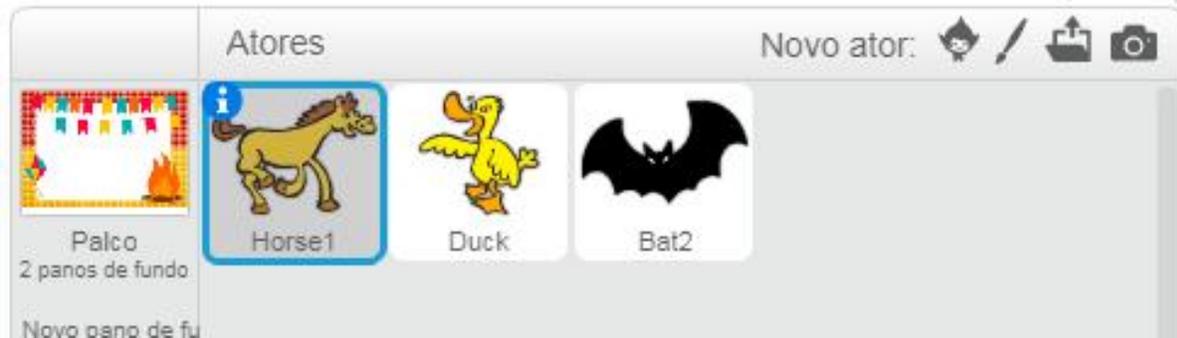
Criar uma animação sobre a Festa da Família.

---

Competências matemáticas e digitais:

Noções de algoritmos, introdução à programação improvisar, ensaiar e corrigir erros.

---



Aula 03

# Aula 04

Objetivo

Criar projeto sobre Ciências.

Projeto em grupo (duplas e/ou trios)

Competências matemáticas e digitais:

Noções de algoritmos, introdução à programação improvisar, ensaiar e corrigir erros.



# Exemplos

# Metodologia

Quatro encontros

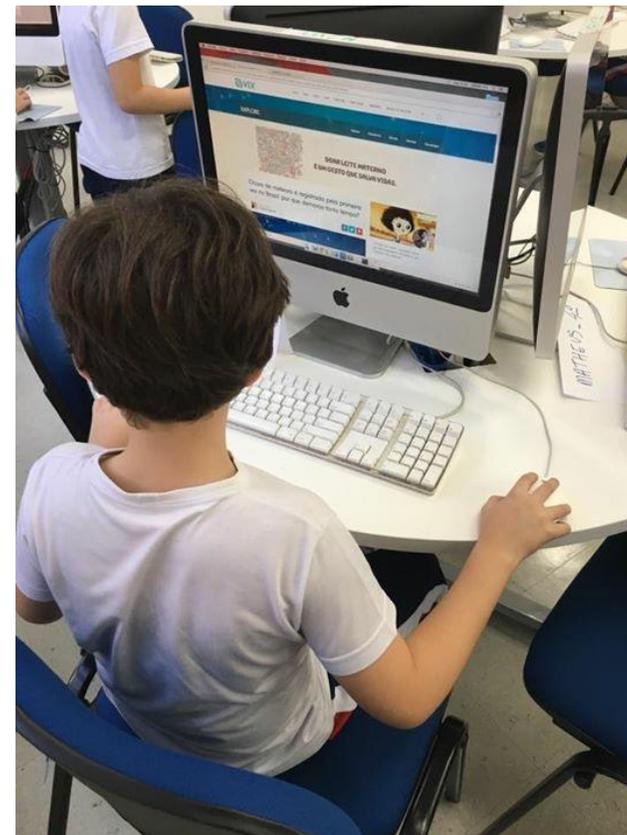
22 alunos:

- 04 alunos do 3º ano
- 06 alunos do 4º ano
- 07 alunos do 5º ano
- 05 alunos do 6º ano

Instrumentos de coleta de dados:

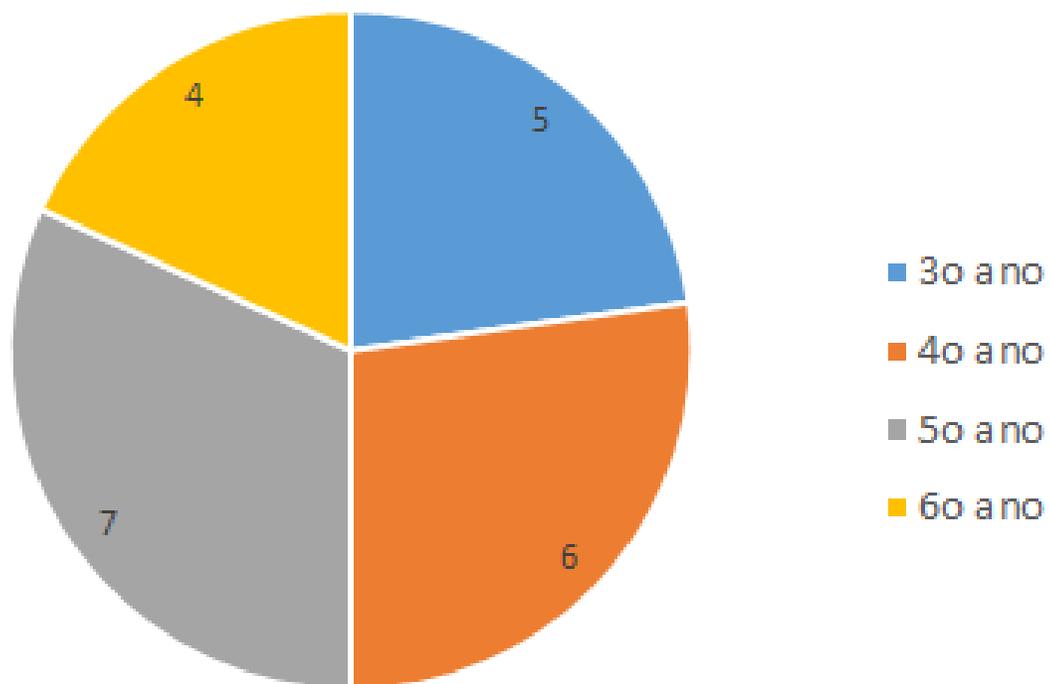
- questionários de perfil, expectativa, pré-teste, pós-teste e de satisfação

Preparação do ambiente de desenvolvimento



# O Ambiente

---

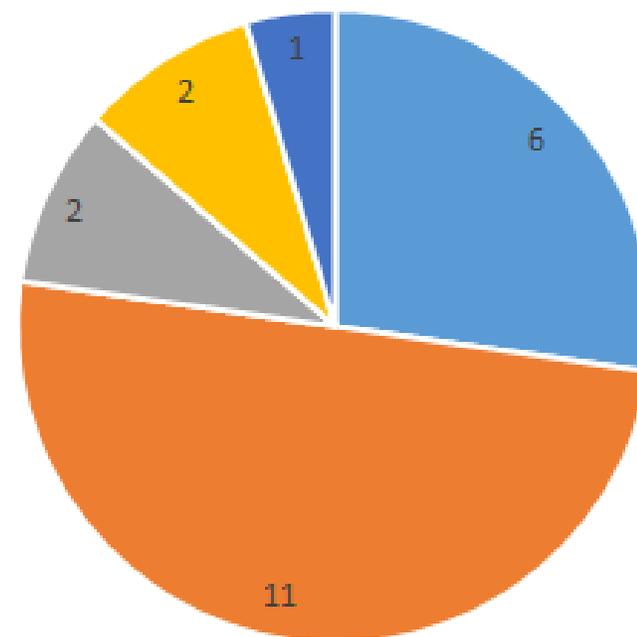


## Resultados – Perfil dos alunos

- 14 meninos e 7 meninas.
- Todos os alunos afirmaram ter tido contato com computadores e tablets;
- apenas 2 alunos afirmaram não ter contato com smartphones e videogames.
- A metade dos alunos respondeu usar computador de 3 a 5 vezes por semana.
- Apenas um dos alunos se manifestou sobre já ter feito curso de programação e já ter usado a ferramenta Scratch.

# Resultados - Expectativas

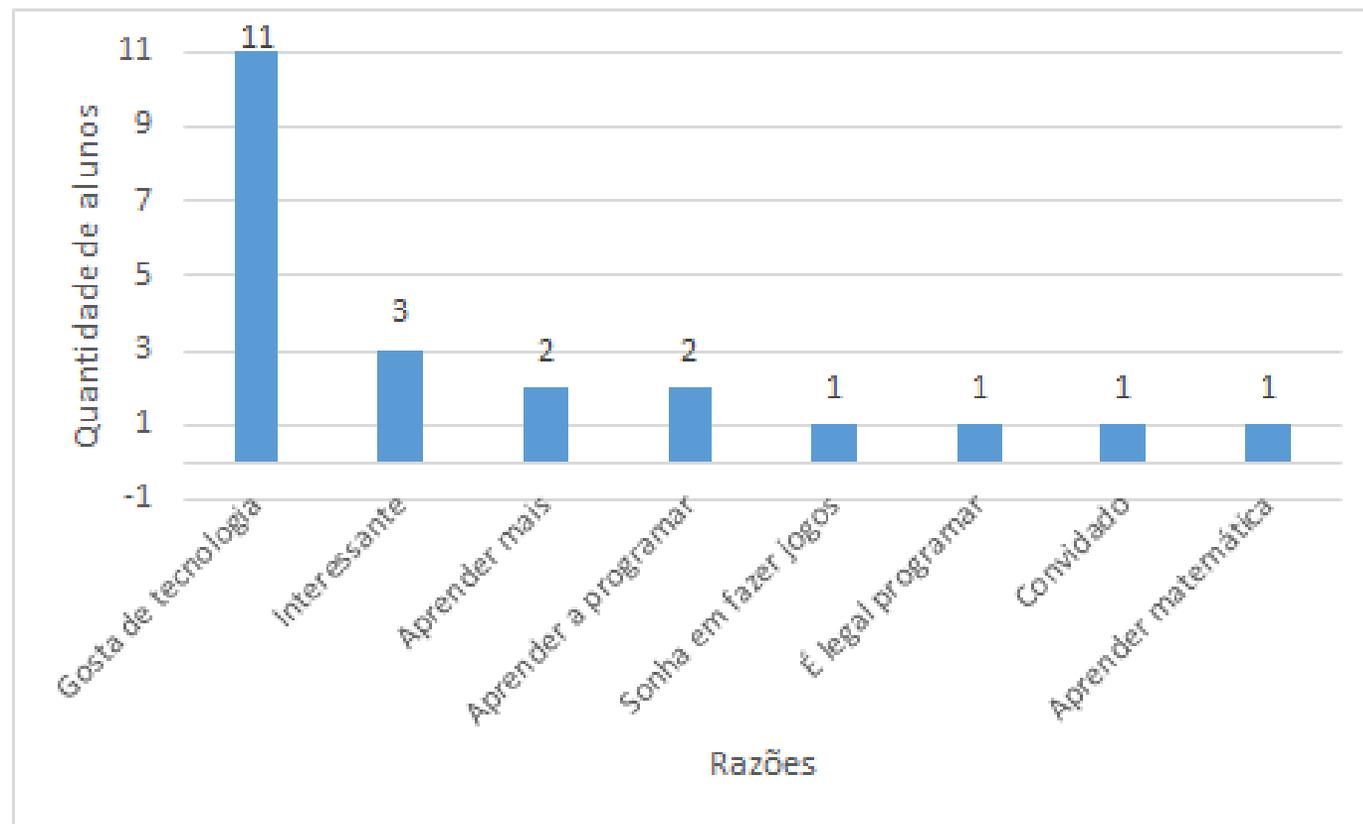
---



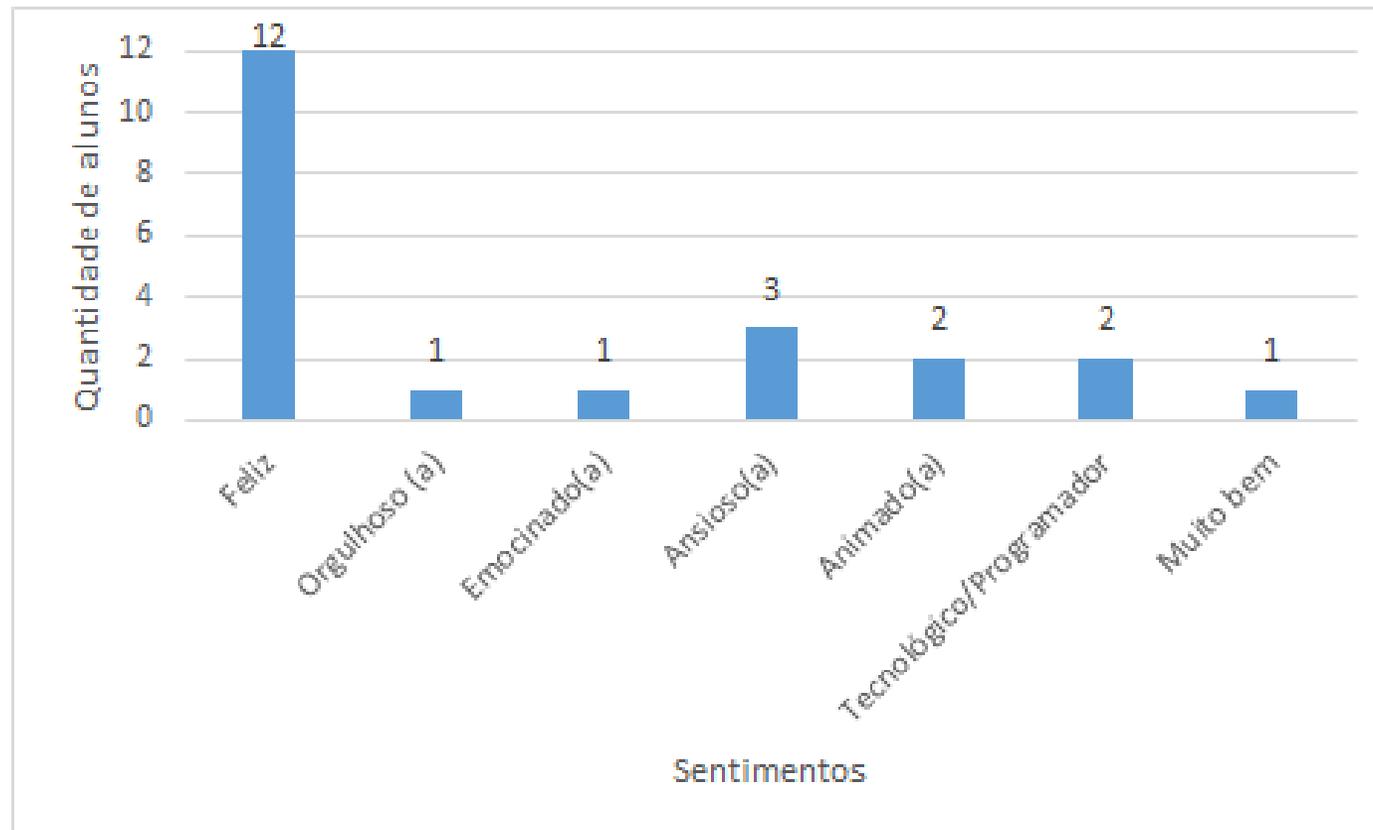
- Fazer Jogos
- Mais sobre tecnologia e informática
- Treinar o cérebro
- Aprender programar
- Nada respondido

# Resultados - Motivação

---



# Resultados – Sentimentos antes de iniciar o curso



## Resultados - Satisfação

---

A média das notas atribuídas pelos alunos sobre a ferramenta Scratch foi de 9,68, com desvio padrão de 0,56. Isso significa que as respostas foram bastante homogêneas.

---

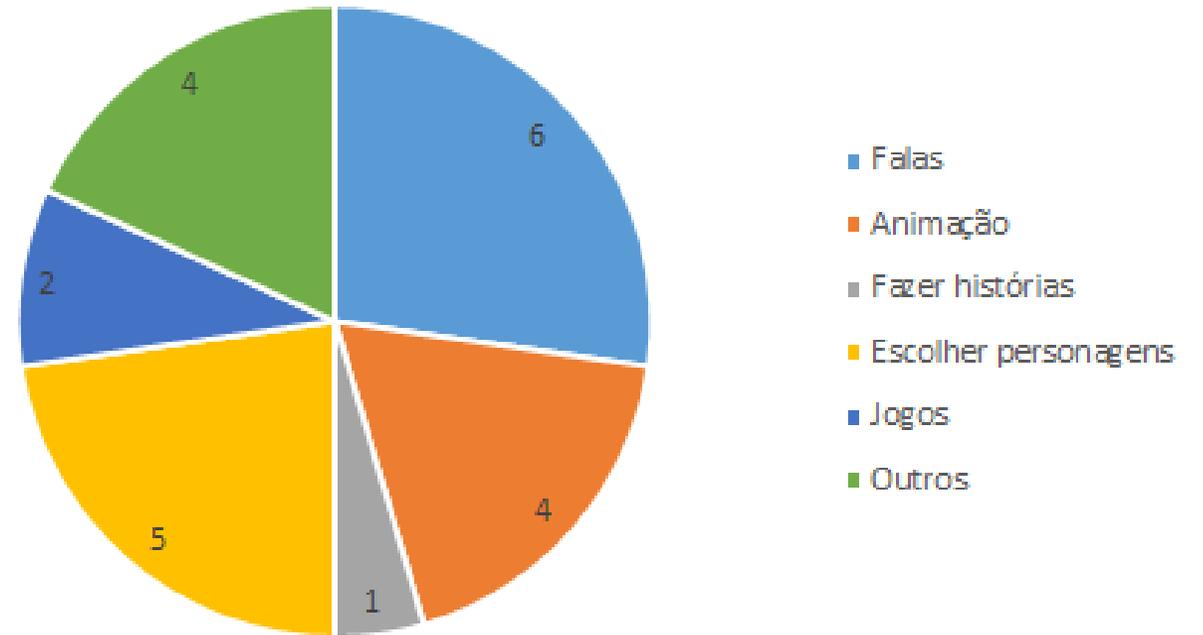
Todos os alunos responderam terem gostado de aprender Scratch.

---

Todos os alunos responderam que gostariam de continuar a utilizar Scratch em casa e na escola.

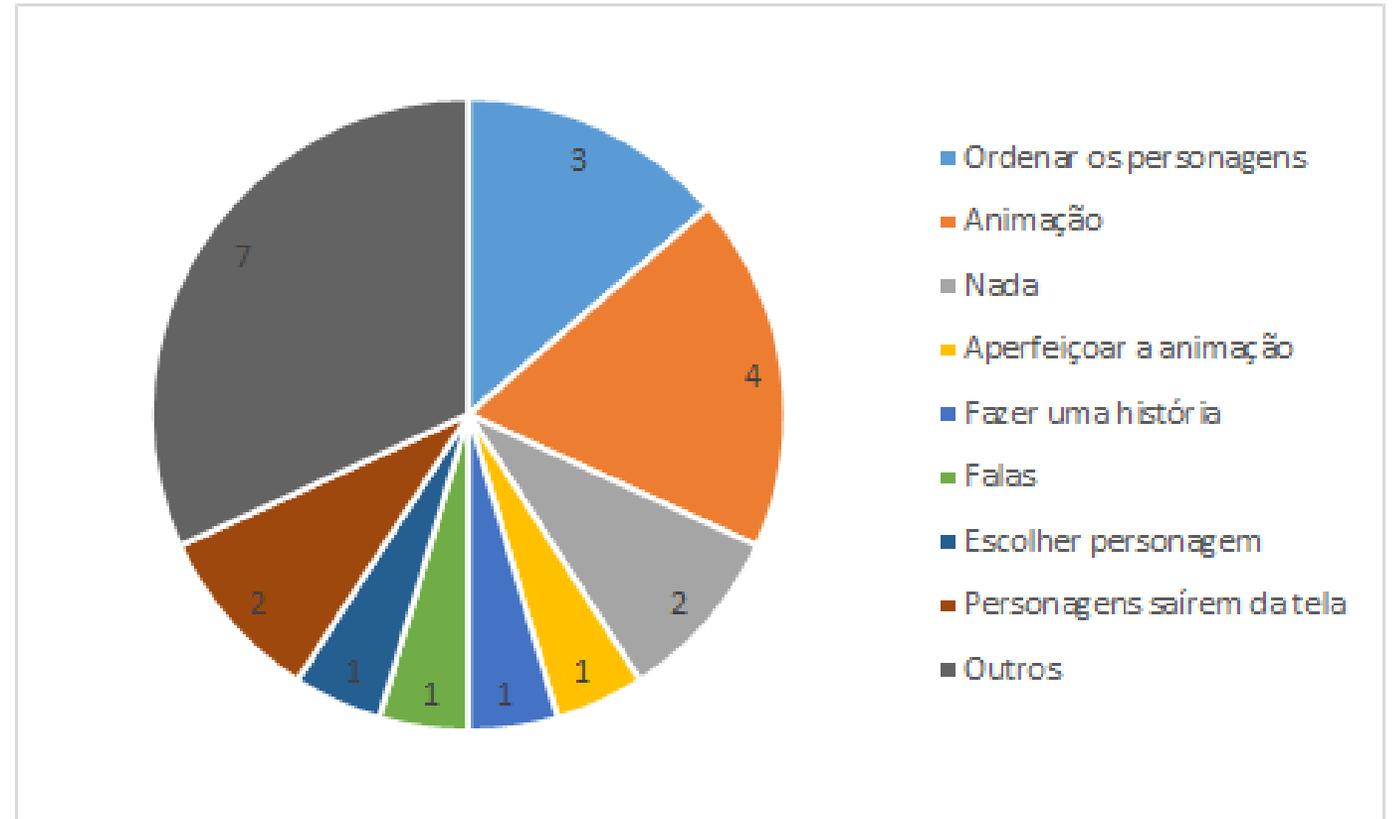
# Resultados – Satisfação – Facilidades com a ferramenta

---



# Resultados – Satisfação – Dificuldade com a ferramenta

---



## Discussões - Expectativas

---

Os alunos e pais tinham uma grande expectativa em relação a esse curso.

---

Os alunos apresentaram, através do questionário de satisfação, que o curso obteve um grande sucesso.

## Discussões – Impressão dos professores

---

O curso cumpriu seus objetivos iniciais e superou as expectativas dos professores.

---

O resultado final dos grupos, através das apresentações dos projetos finais, na área de Ciências, mostrou que os alunos dominaram os assuntos aprendidos durante o curso.

---

Eram visíveis a empolgação dos alunos em todas as aulas e o orgulho de fazerem um projeto que envolvia animações, sons e múltiplos personagens.

# Discussões – Dificuldades e Melhorias

---

Para trabalhar com este público, é importante ter auxiliares de ensino para levar o aluno ao banheiro e ficar com os mesmos no período pré e pós-aula.

---

Apesar das dificuldades de fazer na Universidade, os alunos se sentiram importantes e motivados ao frequentar as escolas dos "grandes".

---

Proxy, senhas da máquina, habilitação de plug-ins de áudio, teclado não ajustado para o layout correto entre outras configurações aumentaram as barreiras de utilização, dificultando o aprendizado e aumenta a demanda de suporte dos alunos.

# AppInventor - Apps

- Criada pelo MIT em 2010.
- O App Inventor permite que pessoas com conhecimentos básicos de programação criem aplicativos para Android.



**Build your project on  
your computer**



**Test it in real-time on  
your device**



# AppInventor - Apps

- Interface de programação em blocos



<http://appinventor.mit.edu/explore/>

# Conclusões

- Despertar o interesse pela Computação
- Exercitar o raciocínio lógico
- Promover o trabalho colaborativo



# Referências

---

<http://www.ufrgs.br/secom/ciencia/o-pensamento-computacional-no-ensino-fundamental/>

---

Brackmann, C. P. (2017). Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica.

---

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

<http://appinventor.mit.edu/explore/>

# Obrigada!

- Contatos
  - [anagras@gmail.com](mailto:anagras@gmail.com)
  - [valeria.farinazzo@mackenzie.br](mailto:valeria.farinazzo@mackenzie.br)

