



# Identificação de Características do Ambiente no *Software* LOGIKID

Tania Sanai Shimabukuro (171025717)  
Profº Drº Renê Pegoraro

# Sumário



- Introdução;
- Fundamentação teórica;
- Desenvolvimento do projeto;
- Contribuições;
- Conclusão;
- Trabalhos futuros.

# Introdução



- Robótica educacional;
- AEDROMO;
- Atividades com outros sistemas;
- LOGIKID (2015);
- Linguagem simples;
- Ambiente físico ou virtual;
- Compreensão da lógica de programação;

# Introdução



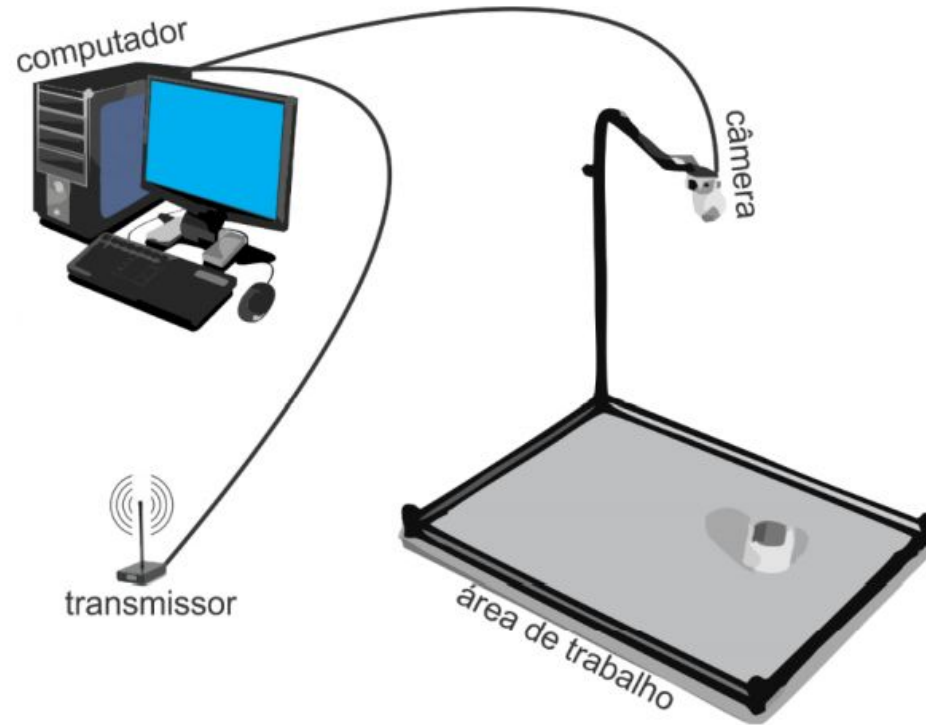
- Comunicação através de perguntas;
- Não percebe obstáculos e características;
- Atualizar o *software*;
- Expansão de possibilidades e limites;
- Utilização das novas particularidades da linguagem.

# Fundamentação teórica - AEDROMO



- Disponível no LISDI;
- Ambiente versátil, amigável e escalável;
- Pode ser programado por qualquer linguagem de programação que suporte *sockets*;
- Formado por: área de trabalho horizontal e plana, objetos passivos, dois (ou mais) robôs, uma câmera global do tipo *webcam* e um transmissor.

# Fundamentação teórica - AEDROMO



# Fundamentação teórica - LOGO

- Seymour Papert (1960);
- Objetivo indeterminado;
- Crianças a partir de 5 anos de idade;
- Exploração do ambiente;
- Melhora da motivação e interesse dos alunos.

The screenshot shows the Logo programming environment. At the top, there is a navigation bar with links for "Lições", "User programs", "Diversão", "Notícias", and "Sobre". To the right of these links are buttons for "Login/Registre-se", social media icons for Facebook and Twitter, and a language selector set to "English". Below the navigation bar is a progress indicator with four steps: "1. A Tat Logo", "2. Controle a Tat e o Lápis", "3. Mundo da Tat" (highlighted in orange), and "4. Onde Está a Tat?". The main interface is divided into two panels. The left panel, titled "lição3: Mundo da Tat", contains a list of seven tasks: "1. Onde Vive a Tat?", "2. Mundo da Tartaruga", "3. Limpar a Tela", "4. Mova no eixo x", "5. Em Casa", "6. Mudando x e y", and "7. Mudando as coordenadas x e y Juntas" (marked with a checkmark). Below this list is a section titled "8. A Cabeça da Tat" which contains a paragraph of text explaining the turtle's head movement: "Ao ver a tartaruga triangular, sabemos para onde sua cabeça aponta. Podemos também mandar sua cabeça girar 360 graus. Quando estamos começando um programa e dissemos à Tat `gd 90`, estamos movendo a cabeça da tartaruga em 90 graus para a direita, e `ge 90` vai movê-la 90 graus para a esquerda. Se quisermos que a cabeça aponte para algum lugar, podemos usar o comando `ar`." The right panel is a large gray area representing the turtle's world, with a small yellow and orange turtle icon in the center. Below this panel is a text input field with the placeholder "digite seu código aqui" and a "Faça" button. At the bottom right, there is a "commands history" section showing the following commands: "tat mudexy 50 50" and "arco 360 100".

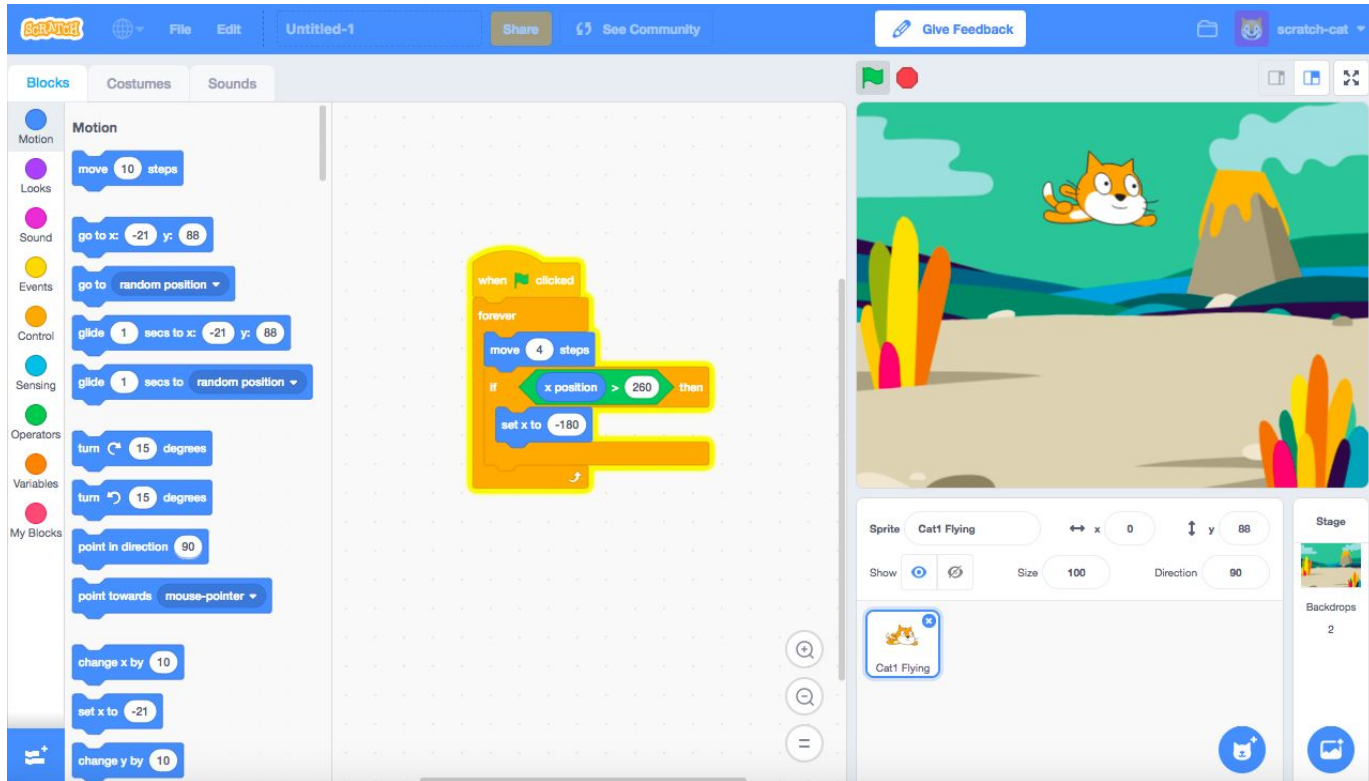
# Fundamentação teórica - SCRATCH



- Grupo Lifelong Kindergarten;
- Gratuito;
- Crianças a partir de 8 anos de idade;
- Auxílio no aprendizado de conceitos matemáticos e computacionais;
- Permite criação de história interativas, jogos, animações e compartilhamento das criações;
- Manipulação de blocos de comando.

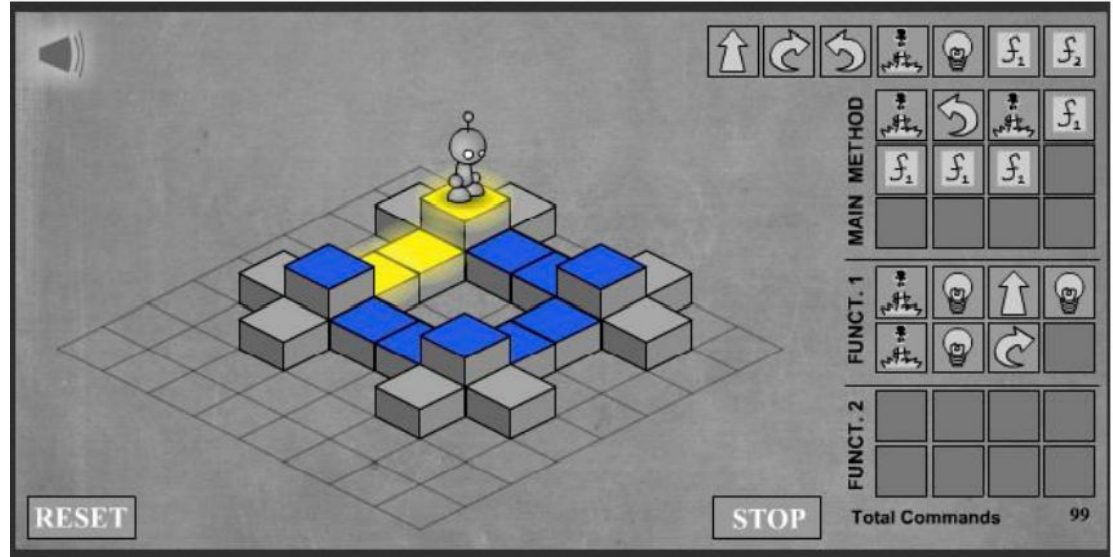


# Fundamentação teórica - SCRATCH



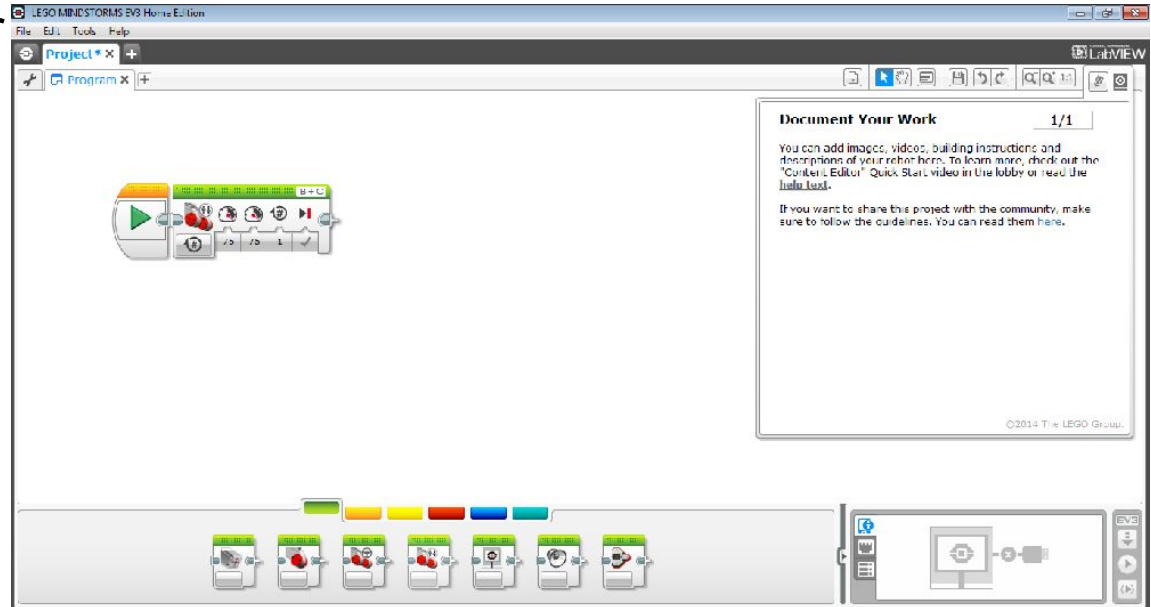
# Fundamentação teórica - LIGHTBOT

- Criado em 2008;
- O objetivo é mover o robô por um tabuleiro;
- *Script* com quantidade restrita ações.



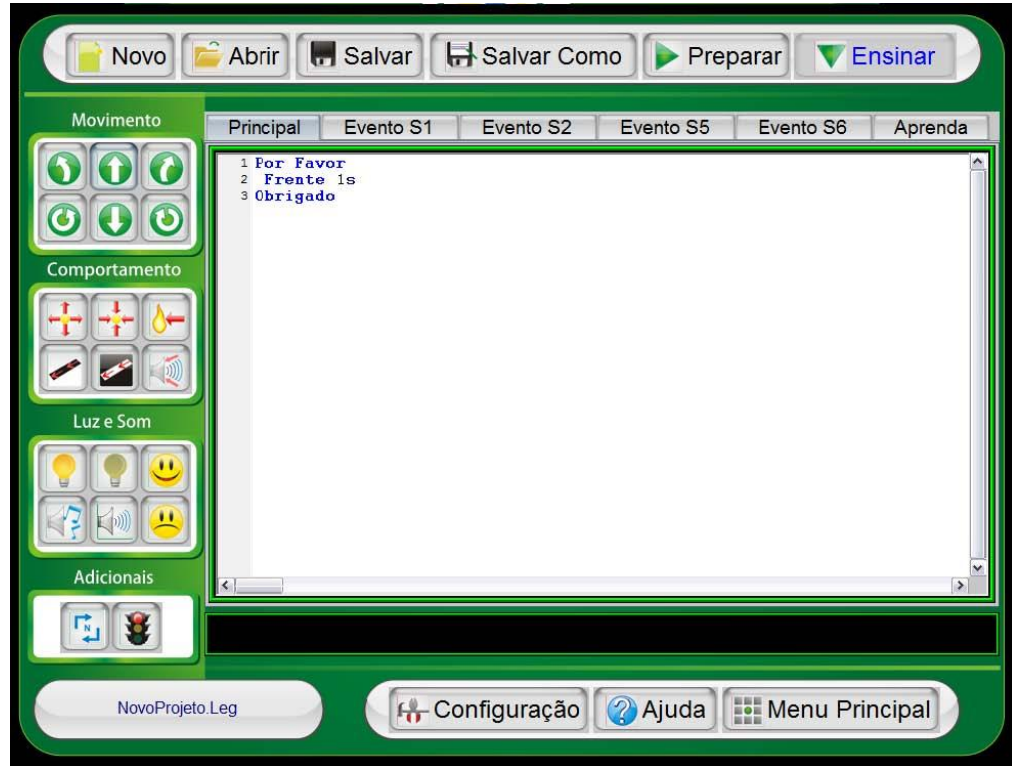
# Fundamentação teórica - LEGO MINDSTORMS

- Inspirado por Seymour Papert;
- Desenvolvimento de projetos de pequeno e médio porte;
- Blocos pré-programados.



# Fundamentação teórica - LEGAL

- PETE;
- Interface gráfica com o usuário e uma linguagem de programação com sintaxe em português;
- Utilização de "por favor" e "obrigado".



# Desenvolvimento do Projeto - Materiais

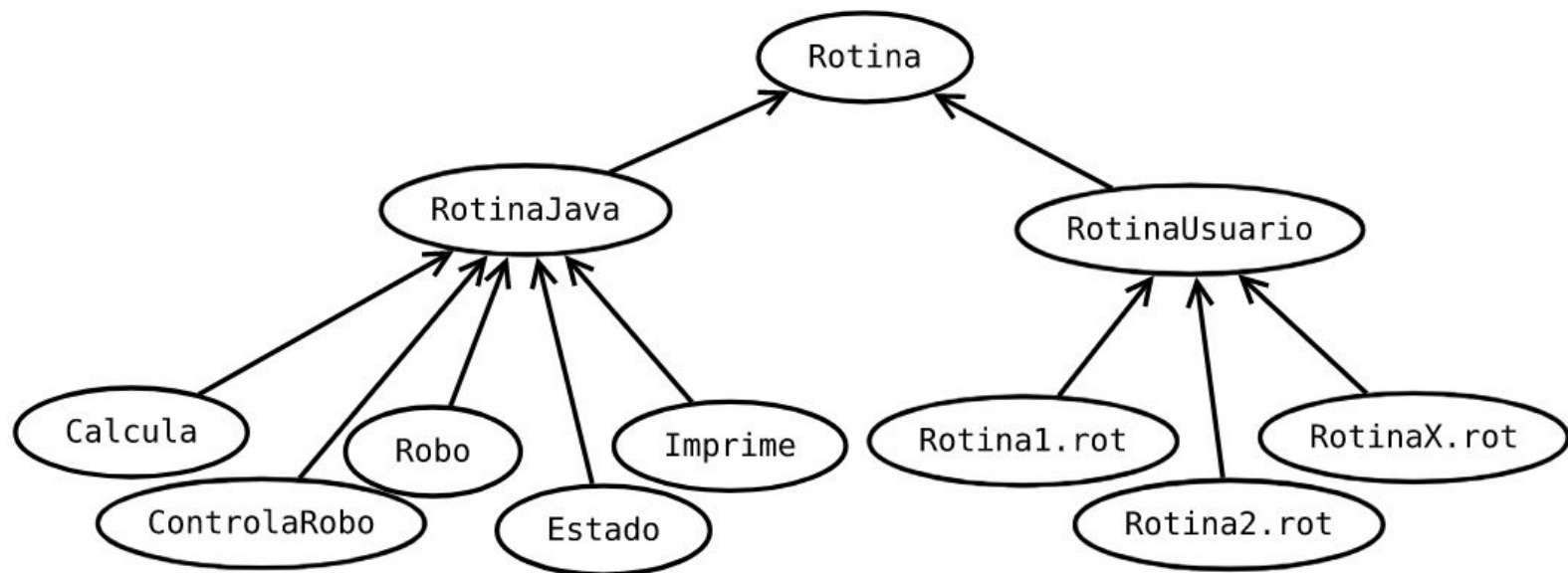
- AEDROMO;
- JAVA;
- Eclipse;
- Processing.



# Desenvolvimento do Projeto - Linguagem do LOGIKID

- Conexão do usuário com o robô;
- Classe *Rotina* é a base da linguagem desenvolvida;
- As classes *RotinaJava* e *RotinaUsuario* são descendentes da classe *Rotina*;

# Desenvolvimento do Projeto - Linguagem do LOGIKID



# Desenvolvimento do Projeto - Linguagem do LOGIKID

- O usuário que define a rotina principal e outras rotinas auxiliares através do ambiente LOGIKID;
- O programa do usuário trata-se de uma rotina que chama outras rotinas pré-definidas ou rotinas definidas pelo próprio usuário;
- Instruções em português;
- Possibilidade de estender e ampliar a linguagem;



# Desenvolvimento do Projeto - Linguagem do LOGIKID

```
1. deslocaRobo x y
2.      Estado ROBO1 xr yr ar
3.      se xr>200
4.          irPara x y
5.      fim
6. fim
```

# Desenvolvimento do Projeto - Linguagem do LOGIKID

- Funções com recursividade;
- Passagem de parâmetro por referência;
- Utilização do '&';
- Manipulação vetores e matrizes;
- Chamada de funções matemáticas com dois parâmetros;
- Exibição de textos a partir da execução da função *Imprime*;

# Desenvolvimento do Projeto - Linguagem do LOGIKID

```
1.  fibonacci & r n
2.      se n = 0
3.          Calcula r 0
4.      senao
5.          se n = 1
6.              Calcula r 1
7.          senao
8.              Calcula r1 0
9.              Calcula r2 0
10.             fibonacci r1 n-1
11.             fibonacci r2 n-2
12.             Calcula r r1+r2
13.             Imprime n
14.         fim
15.     fim
16. fim
```

# Desenvolvimento do Projeto - Linguagem do LOGIKID

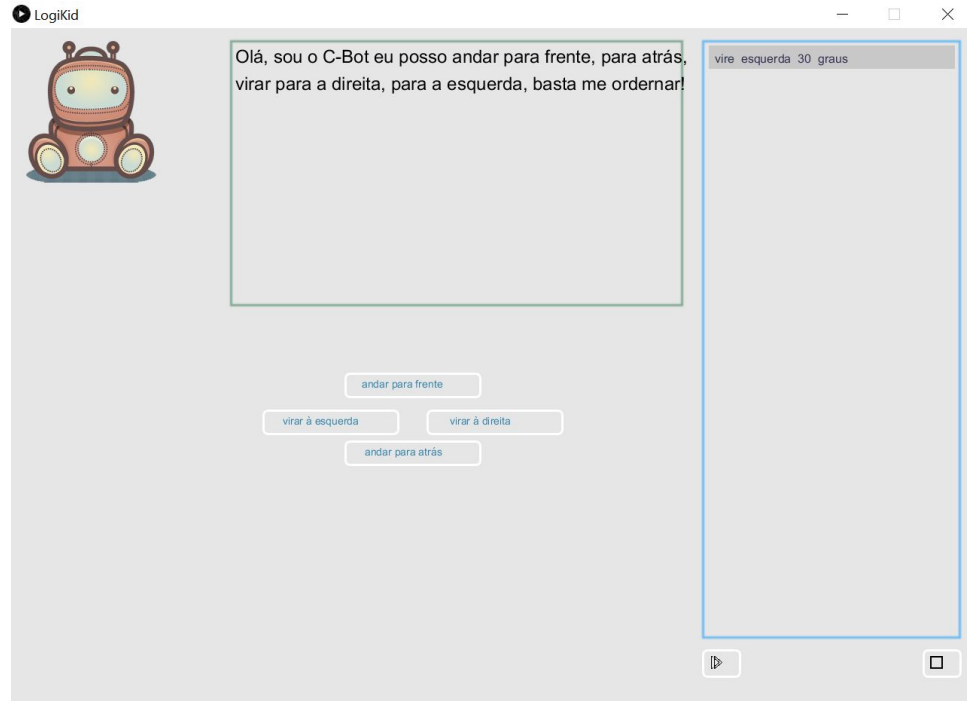
```
class Soma extends RotinaJava {  
  
    void executa(String linChamada[], Map<String, Object> locais) {  
        super.executa(linChamada, locais);  
        Float f;  
        try {  
            f = expressao.valor(linChamada[2]) + expressao.valor(linChamada[3]);  
            String dest = variavelDestino(linChamada[1]);  
            locais.put(dest, f);  
        } catch (Exception e) {  
            erro(e.getMessage() + "\n Argumento invalido.");  
        }  
    }  
}
```

# Desenvolvimento do Projeto - Linguagem do LOGIKID

```
1. testeSoma x y  
2.   Soma x y 10  
3. fim
```

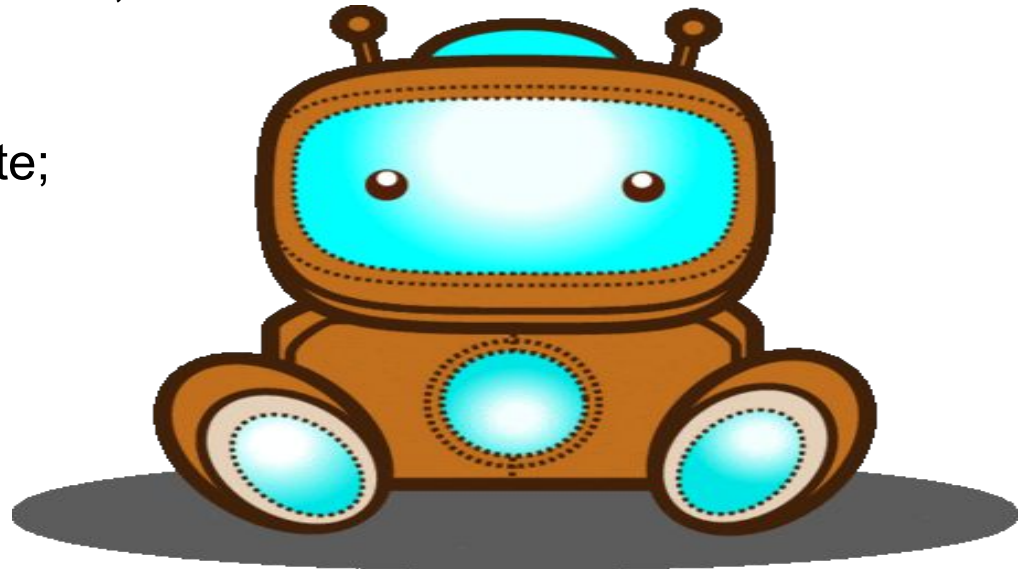
# Desenvolvimento do Projeto - Interface gráfica com o usuário

- Interface original;
- Três principais áreas:  
comunicação com o robô, seleção de instrução e programa.

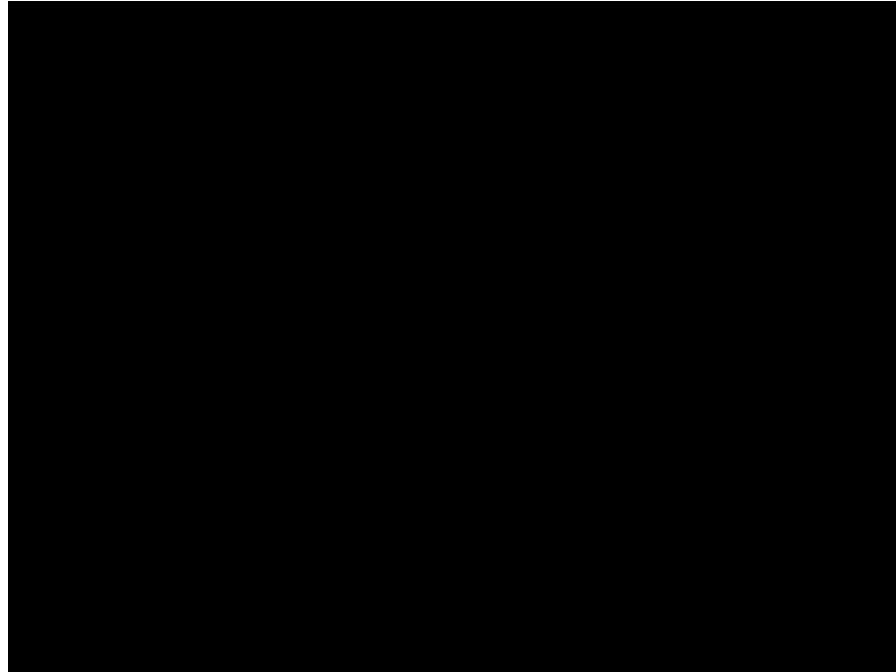


# Desenvolvimento do Projeto - Interface gráfica com o usuário

- Reestruturação do software;
- Estrutura remodelada;
- Modificação do mascote;
- Código modularizado.



# Desenvolvimento do Projeto - Interface gráfica com o usuário





# Contribuições



- Linguagem: passagem de parâmetros por referência, manuseio de matrizes e vetores, exibição de textos, capacidade de chamar funções matemáticas com dois parâmetros;
- Interface: exibição do comando sendo criado, seleção da linha do programa, exibição de uma pergunta por vez, possibilidade de voltar.

# Conclusão



- Baseado em pressupostos de softwares como LOGO, SCRATCH e LIGHTBOT;
- Intuito de possibilitar o desenvolvimento de um pensamento computacional;
- Códigos de programação que controlam um robô, além de indicar o equivalente desse código em "português estruturado".

# Trabalhos futuros



- Permitir edição e exclusão dos comandos criados;
- Inserir a opção de áudio para as perguntas existentes;
- Inserir características de gamificação.

# Referências



- LISBOA, C. O. LOGIKID: software para o ensino de lógica e programação com o AEDROMO. 2015. 54 f. Monografia (Trabalho de Conclusão de Curso) — Universidade Estadual Paulista “Júlio de Mesquita Filho” (UNESP), Bauru, 2015.
- LOPES, D. d. Q. A exploração de modelos e os níveis de abstração nas construções criativas com robótica educacional. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, 2008.
- SCRATCH. Acerca do scratch. 2020. Disponível em: <<https://scratch.mit.edu/about>>. Acesso em: 27 nov. 2020.