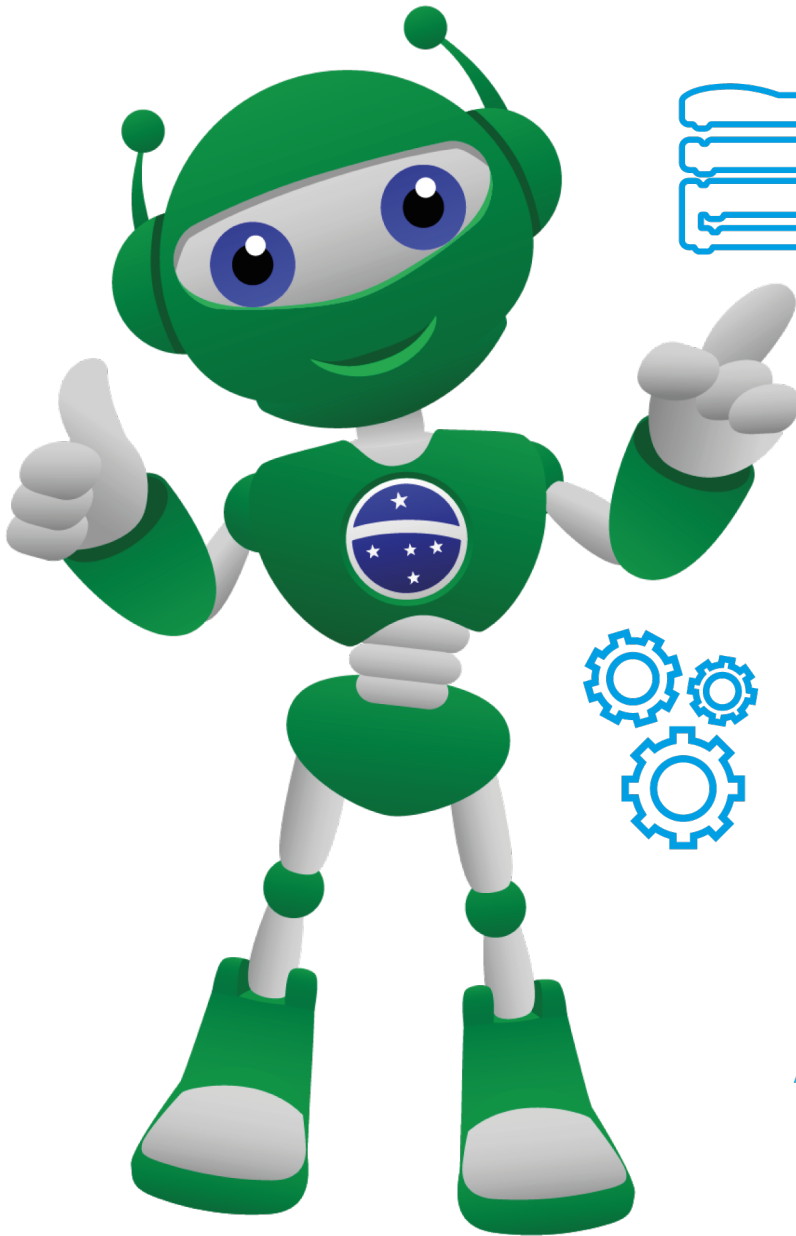


ROBÓTICA

Primeiros Passos

Módulo 1



AULA 24

Criando Animação no mBlock

Diretoria de Tecnologia e Inovação

GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Renato Feder

DIRETOR DE TECNOLOGIA E INOVAÇÃO

André Gustavo Souza Garbosa

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Michelle dos Santos

Validação de Conteúdo

Revisão Textual

Adilson Carlos Batista

Leitura Crítica e Normalização Bibliográfica

Ricardo Hasper

Projeto Gráfico e Diagramação

Edna do Rocio Becker

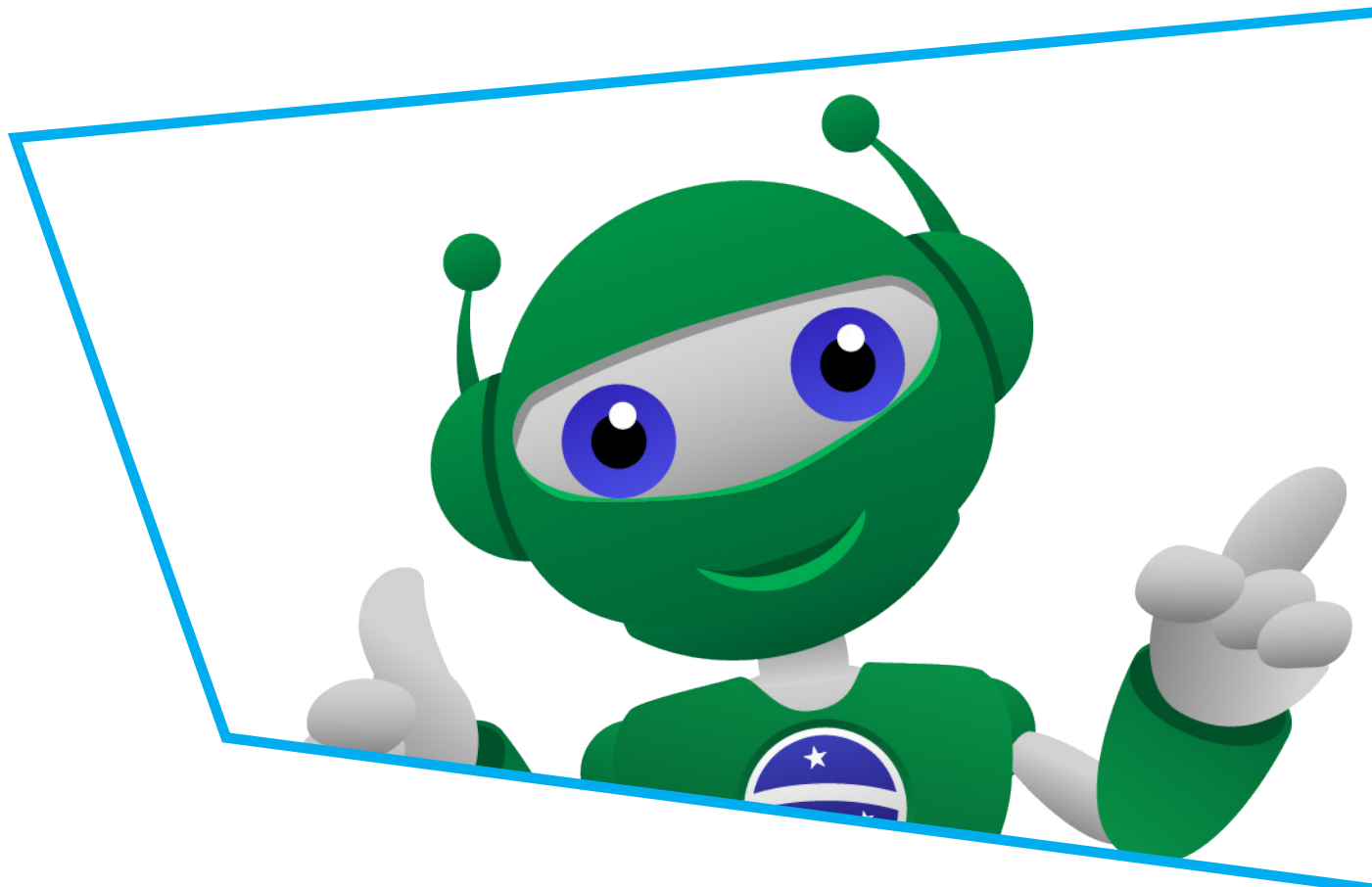
Ilustração

Jocelin Vianna (Educaplay)

2022

SUMÁRIO

Introdução	2
Objetivos desta Aula	2
Competências Gerais Previstas na BNCC	3
Habilidades do Século XXI a Serem Desenvolvidas	4
Roteiro da aula	4
1. Contextualização	5
2. Conteúdo	5
3. Feedback e Finalização	26
Referências	27





Introdução

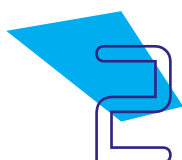
Vamos “dar asas” à imaginação? Dar vida à imaginação é poder materializar as ideias, como por exemplo, dar movimento às imagens virtuais trazendo mais próximas da realidade.

Nesta aula, aplicaremos as funcionalidades da linguagem de programação mBlock para desenvolver uma animação com a mascote da Robótica.



Objetivos desta Aula

- Compreender a linguagem de programação mBlock;
- Desenvolver uma animação através do software mBlock;
- Estimular o desenvolvimento de habilidades para criação de novas programações em novos projetos.





Competências Gerais Previstas na BNCC

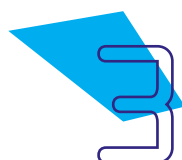
[CG02] - Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas) com base nos conhecimentos das diferentes áreas.

[CG04] - Utilizar diferentes linguagens – verbal (oral ou visual-motora, como Libras, e escrita), corporal, visual, sonora e digital –, bem como conhecimentos das linguagens artística, matemática e científica, para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e produzir sentidos que levem ao entendimento mútuo.

[CG05] - Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.

[CG09] - Exercitar a empatia, o diálogo, a resolução de conflitos e a cooperação, fazendo-se respeitar e promovendo o respeito ao outro e aos direitos humanos, com acolhimento e valorização da diversidade de indivíduos e de grupos sociais, seus saberes, identidades, culturas e potencialidades, sem preconceitos de qualquer natureza.

[CG10] - Agir pessoal e coletivamente com autonomia, responsabilidade, flexibilidade, resiliência e determinação, tomando decisões com base em princípios éticos, democráticos, inclusivos, sustentáveis e solidários.



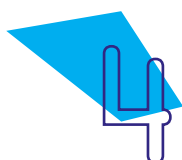
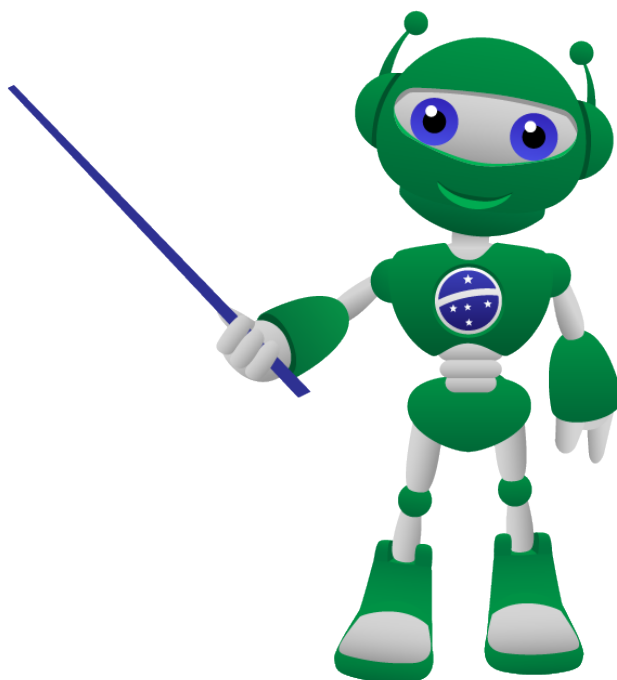


Habilidades do Século XXI a Serem Desenvolvidas

- Pensamento crítico;
- Afinidade digital;
- Resiliência;
- Resolução de problemas;
- Colaboração;
- Comunicação;
- Criatividade.

Lista de Materiais

- Notebook;
- Software mBlock.





Roteiro da Aula

1. Contextualização:



Nossa mascote **BIT** tem adicionado mais alegria em nossas aulas de Robótica, não é mesmo? Que tal dimensionar esta interação e dar vida a ele através de uma animação?


Na aula anterior, **Aula 24 - Software mBlock**, conhecemos a linguagem de programação e como montar uma programação juntando os blocos. Com isso, estamos aprendendo a programar de forma divertida, explorando o software mBlock, que permite a criação de animações, jogos e histórias. Então, que tal colocar a “mão na massa” e criarmos uma animação?

2. Conteúdo:

Acesse primeiro a plataforma de programação do mBlock.



<https://ide.mblock.cc/>

Vamos começar escolhendo a personagem que iremos construir a animação. Perceba que o Panda aparecerá como personagem padrão do mBlock. Por isso, primeiro vamos excluí-lo. Clique no X no ícone que seleciona este ator  na área dos atores.

A seguir, vamos acrescentar a imagem da mascote da Robótica **BIT** que será nosso ator, buscando nos arquivos do computador (figura 1).

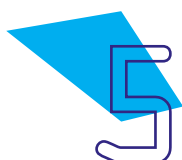
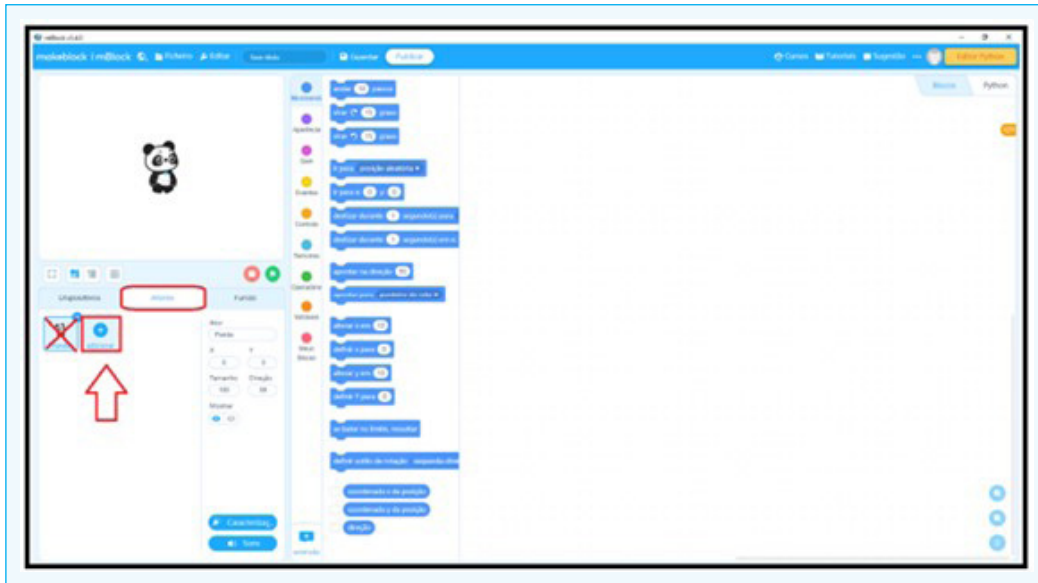


Figura 1 - Seleção do ator do computador

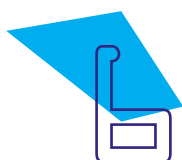
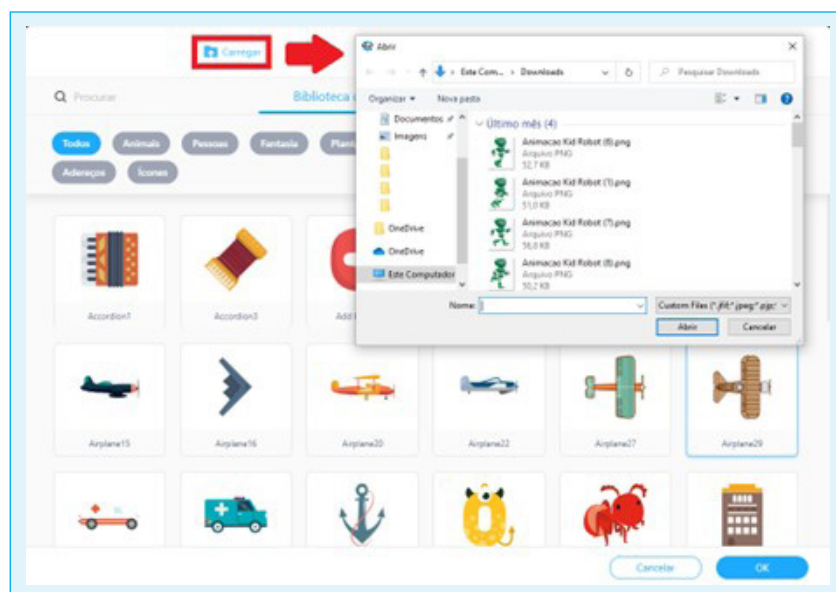


Fonte: Makeblock, 2022

Para adicionar a personagem do seu computador, primeiro, faça o download das imagens da mascote em seu computador através do link https://drive.google.com/drive/folders/17kD_y9ObfyqC8Mn8dQm0mxdG3-pkVgBH?usp=sharing.

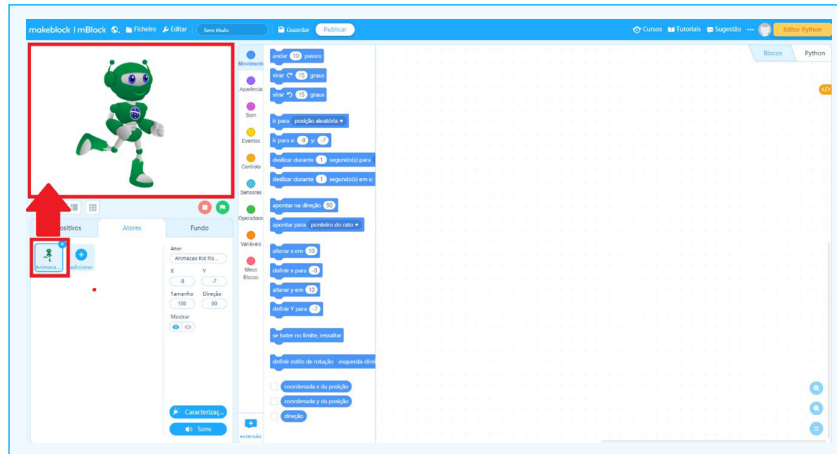
Após baixar as imagens em seu computador, na área de atores do mBlock, clique em **Adicionar** e selecione o ícone **Carregar** (figura 2).

Figura 2 - Inclusão da mascote da Robótica



Selecione a figura da mascote da Robótica do arquivo do computador (figura 3).

Figura 3 - Visualização do personagem



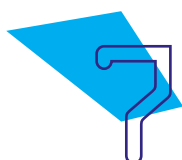
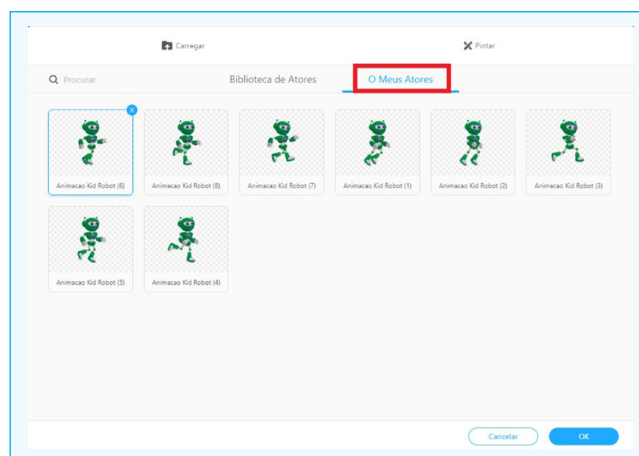
Você percebeu que existem mais de uma imagem da mascote B1T?

Nesta animação, o B1T estará em movimento, e cada movimento exigirá uma posição (figura 4).

As variedades de figuras da mascote da Robótica B1T, com posições diferenciadas, são chamadas de fantasias, que servirão para simular o B1T se deslocando, alternando as imagens. Mas para produzir este efeito é preciso criar um script com o bloco que intercale as fantasias.

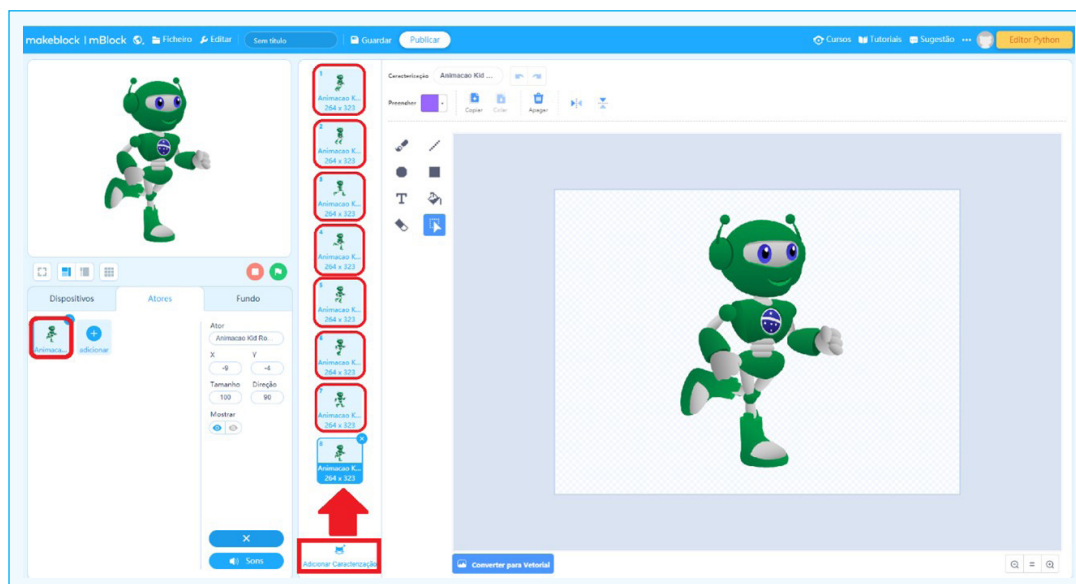
Assim, inclua as outras 7 figuras com diferentes posições, observe que elas estão identificadas pela numeração da fantasia, conforme mostrado na figura 4.

Figura 4 - Variedades de Fantasias



Entenda, a ideia é que seja apenas uma personagem na animação, mas com efeito de deslocamento. Então, clique em **Adicionar Caracterização** na parte inferior da área, conforme mostrado na figura 5.

Figura 5 - Adição de caracterização




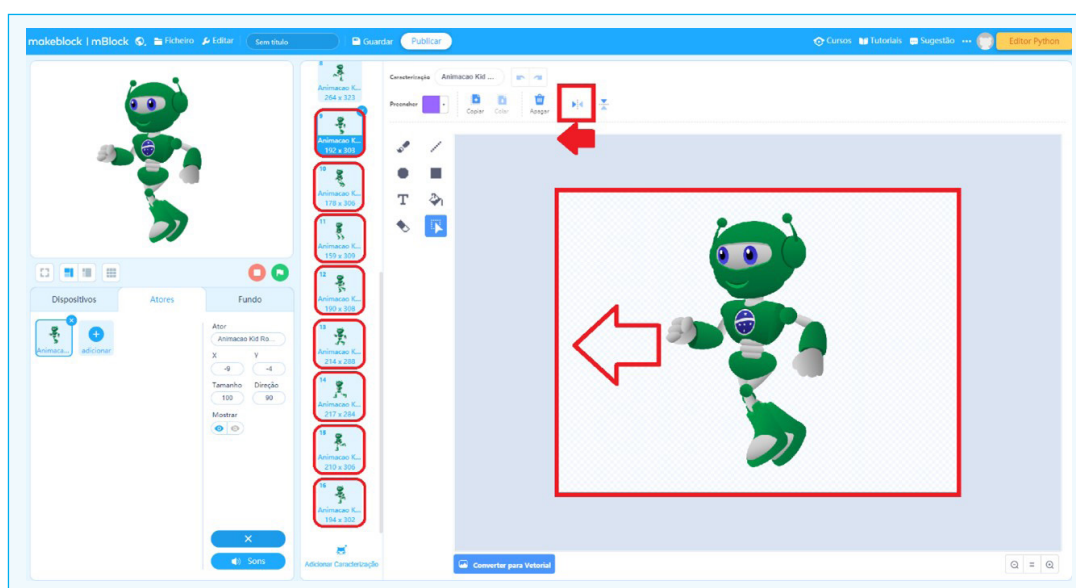
Depois que adicionamos as 8 fantasias, vamos inserir cada fantasia novamente, mas neste caso, iremos invertê-las mudando a direção delas, pois faremos a mascote voltar o percurso percorrido na animação. Clique no ícone inverter na horizontal  para girar o ator, conforme apresentado na figura 6.

Figura 6 - Adição do personagem na direção oposta

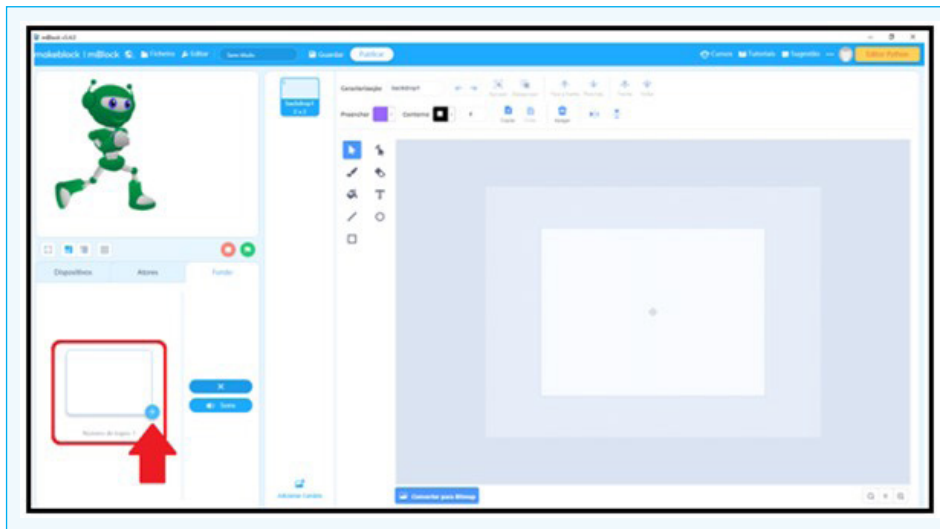


Dessa forma, serão, no total, 16 fantasias da mascote BIT, sendo as fantasias de 1 a 8 no sentido ida e as fantasias de 9 a 16 de retorno (sentido contrário).

Para dar mais realidade ao nosso projeto, vamos definir um ambiente onde nosso ator irá se deslocar.

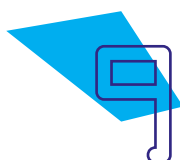
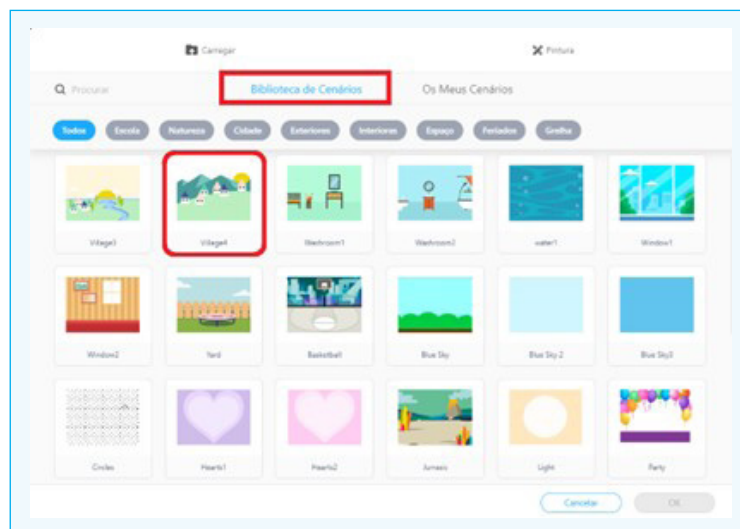
Na área fundo, clique em **+ número de trajés 1**  para adicionar um cenário, conforme figura 7.

Figura 7 - Definição de cenário



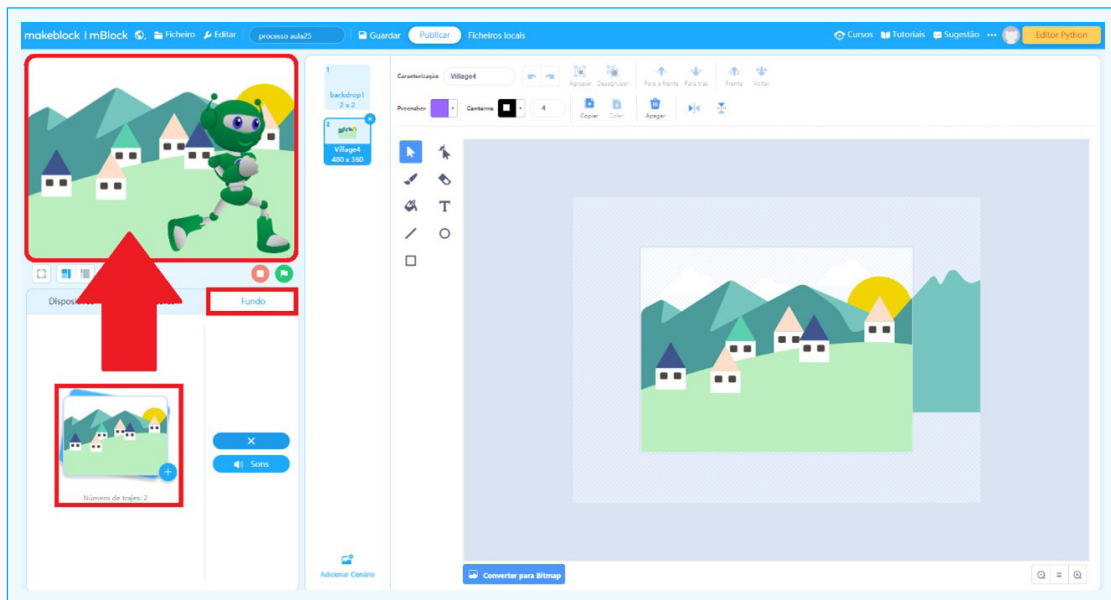
8. Abrirá uma caixa com opções de Cenário, conforme mostrado na figura

Figura 8 - Biblioteca de Cenários



Escolhemos a opção Village4. Mas você pode personalizar, segundo sua preferência. Clique sobre a opção escolhida e será possível ver o fundo da mascote BIT com o cenário selecionado, conforme figura 9.

Figura 9 – Visualização do cenário



Chegou o momento de dar ação a nossa personagem! Para isso, clicando na área dos atores e no campo de BLOCOS, selecionaremos os comandos para construção da programação.

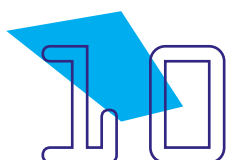
Para dar início aos comandos, adicionaremos o bloco **Evento** quando for clicado. Este bloco, geralmente, usamos para iniciar a programação, pois sempre que for clicado na bandeira, começa a executar os comandos da programação.

Agora, vamos criar blocos com as características específicas que usaremos em nossa animação. Então, vamos definir uma Variável para a Fantasia.

Clique no bloco **Variáveis** e depois em **Criar uma Variável**. Abre uma caixa, onde criaremos um bloco tipo FANTASIA. Depois de dar o nome, clique em **Está bem** para confirmar.



Agora, que aprendeu como criar uma variável, faremos o mesmo com outras funções.

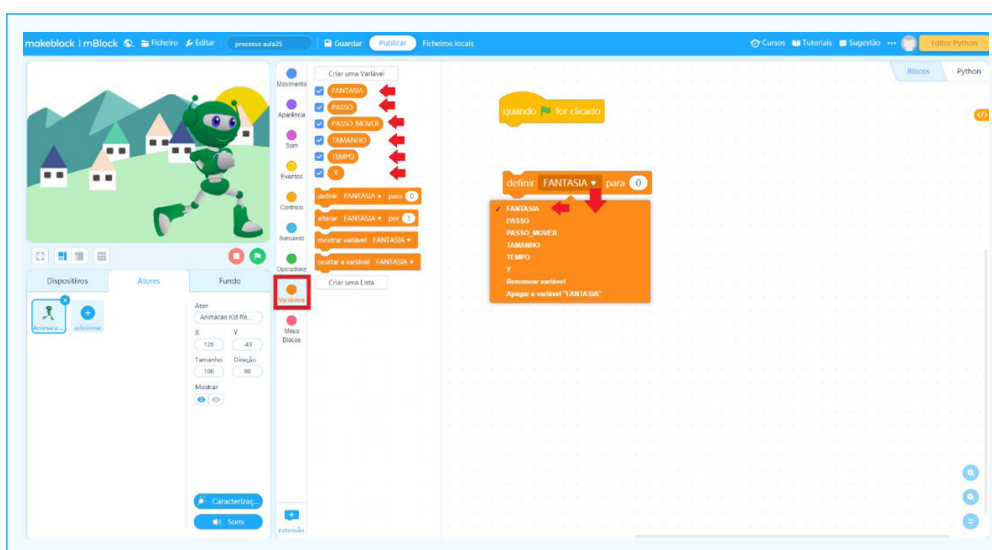


Criaremos outras variáveis nomeadas de: PASSO, PASSO-MOVER, TAMANHO, TEMPO e Y.

Depois de criadas as variáveis, definiremos um valor para cada uma delas através do bloco: **definir fantasia para** definir fantasia para 0.

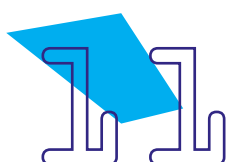
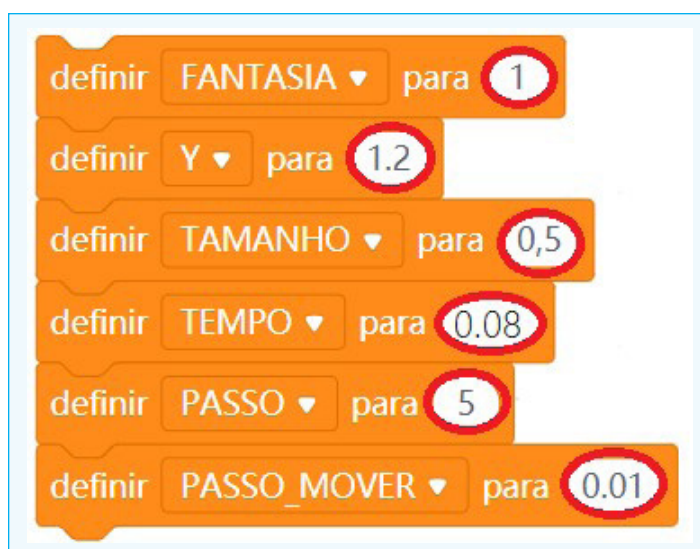
Clique na seta do bloco e selecione as demais variáveis que criou para defini-las, conforme mostrado na figura 10.

Figura 10 - Criação dos blocos variáveis



Agora, preencha cada bloco de variável com os valores representados na figura 11.

Figura 11 - Determinando valores às variáveis

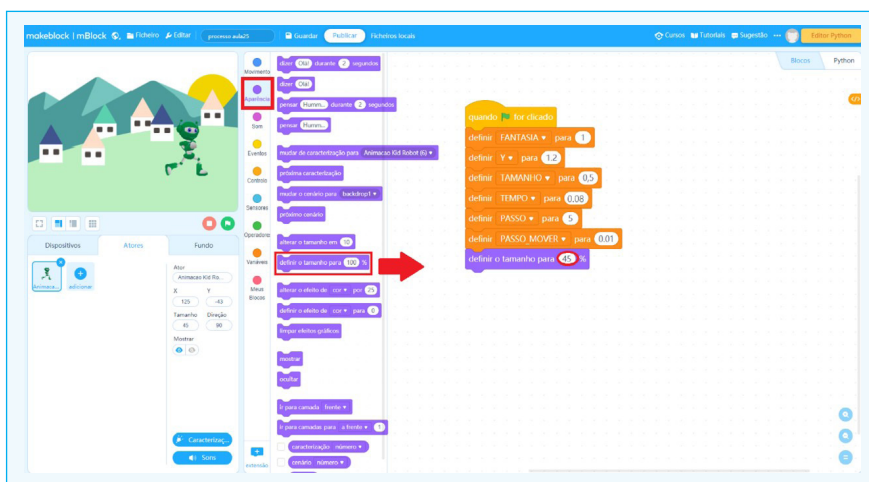


Estes comandos determinarão um valor definido quanto à fantasia, à direção do deslocamento vertical (Y), ao tamanho da personagem, conforme o deslocamento, velocidade a cada passo na horizontal e vertical, padronizando, assim, estes valores, quando utilizados na programação.

Agora, vamos adicionar um bloco do tipo aparência: **definir o tamanho para 100%** `definir o tamanho para 100 %`, que definirá o tamanho da mascote BIT.

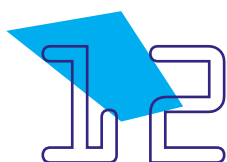
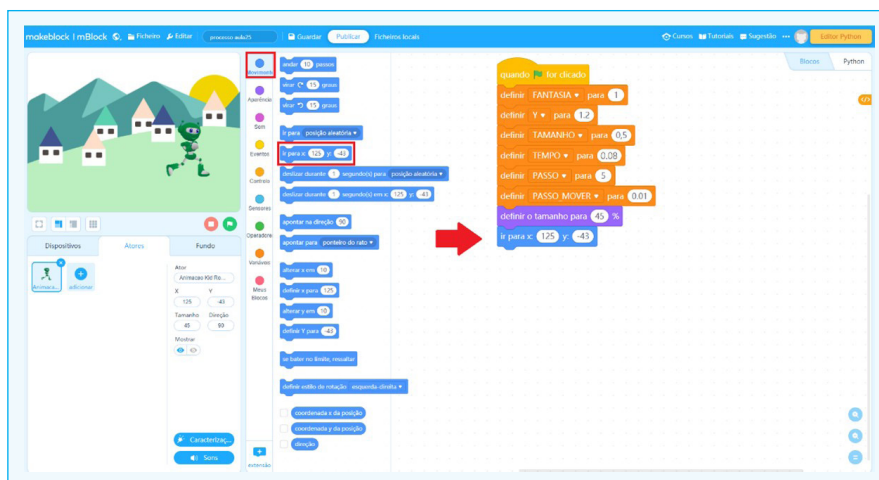
Neste comando, estamos definindo **o tamanho** da fantasia em 45%, conforme mostrado na figura 12.

Figura 12 - Definição do tamanho da fantasia



O próximo passo, vamos adicionar bloco de Movimento: **ir para: x e y** `ir para x: -182 y: -102`, para definir a posição inicial da mascote dentro do cenário, sendo que X representa a posição horizontal e Y a vertical (figura 13).

Figura 13 - Definição do sentido de deslocamento



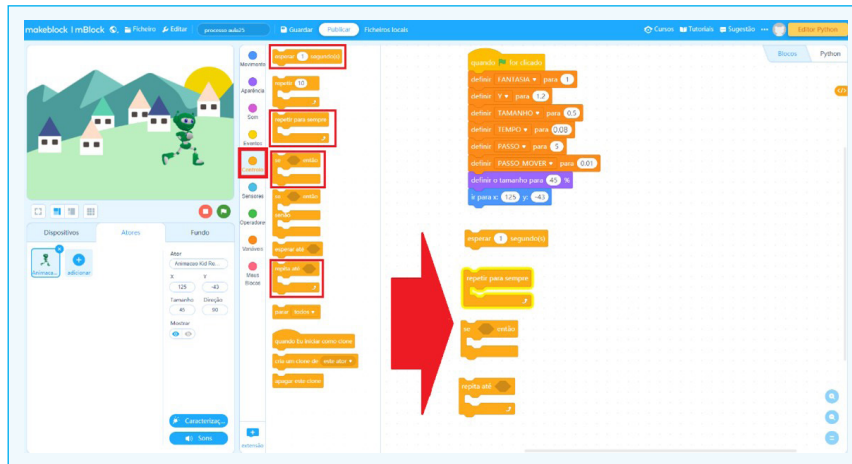
Criando animação no mBlock

Vamos adicionar os seguintes blocos de Controle: **esperar 1 segundo**

, repetir **para sempre**, **se então**, **repita até**.

Eles são capazes de gerenciar vários comandos como repetir ou colocar uma condição (figura 14).

Figura 14 - Selecionando blocos de controle



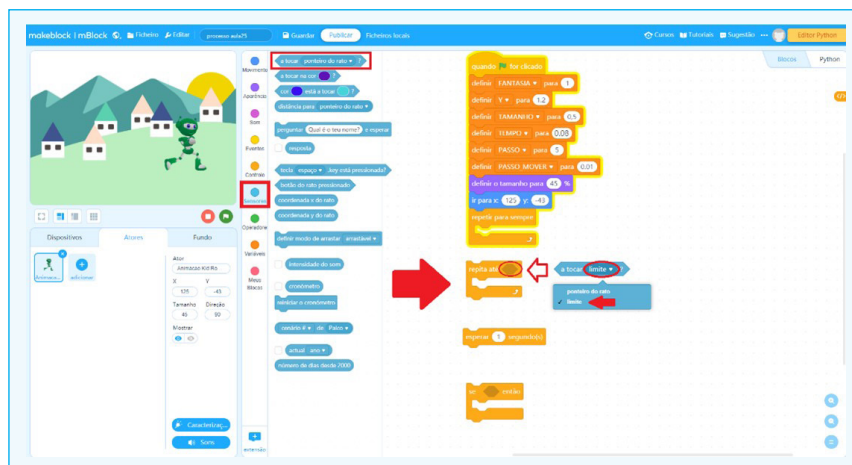
Agora, vamos definir o limite até onde a mascote percorrerá até retornar. Determinaremos que o limite de deslocamento será a tela, isto é, quando tocá-la, inverterá a posição.

Para isso, vamos adicionar o bloco de sensor: **a tocar ponteiro do rato**

, o encaixando no bloco de controle: **repetir até**.

Agora, clique na seta dentro do bloco e altere para Limite, conforme mostrado na figura 15.

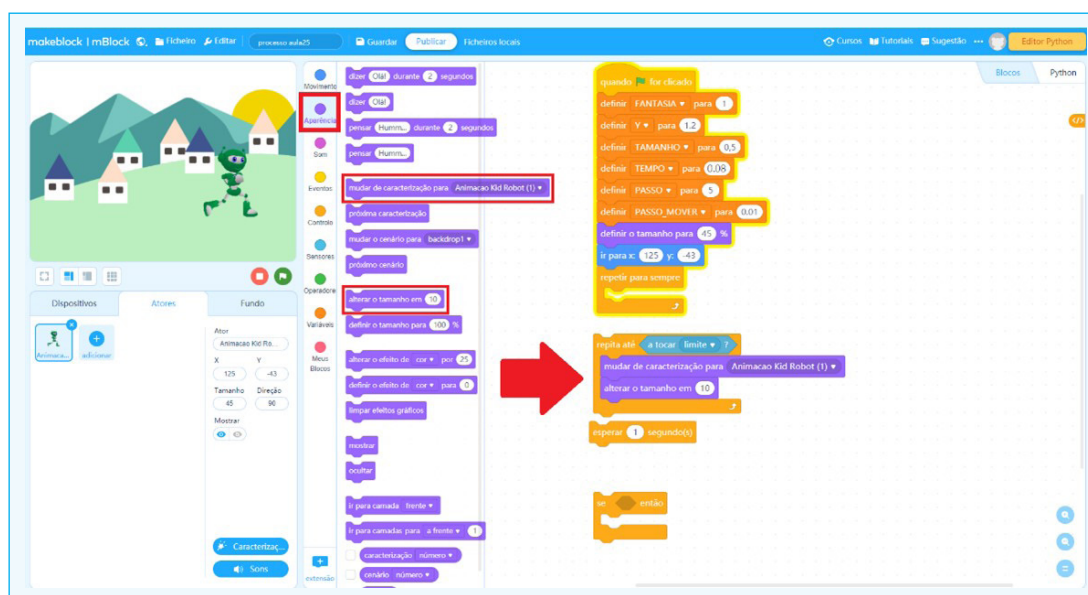
Figura 15 - Definindo o limite de deslocamento



Determinaremos, agora, a aparência da mascote, definindo a alteração das fantasias, sendo que na ida irá começar com a fantasia 1 e vai alternando até chegar na 8, e o valor padrão do tamanho que irá diminuir conforme se desloca.

Então, adicione os seguintes blocos de Aparência: **mudar de caracterização para (1)** **mudar de caracterização para 1** e **alterar o tamanho em 10** **alterar o tamanho em 10** (figura 16).

Figura 16 - Definição da aparência.



A cada passo a mascote deverá mudar a fantasia e diminuir o tamanho de forma constante, para dar a impressão de distanciamento conforme vai se deslocando.

Para isso, vamos definir os comandos para alternância das fantasias e o tamanho, adicionando as duas variáveis que criamos de **fantasia** **fantasia** e **tamanho** **tamanho**.

Encaixe o bloco **fantasia** **fantasia** no espaço do bloco de caracterização.

No bloco de tamanho, iremos adicionar antes um bloco operador: , conforme figura 17.

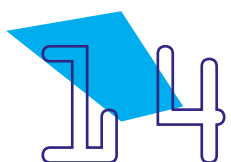
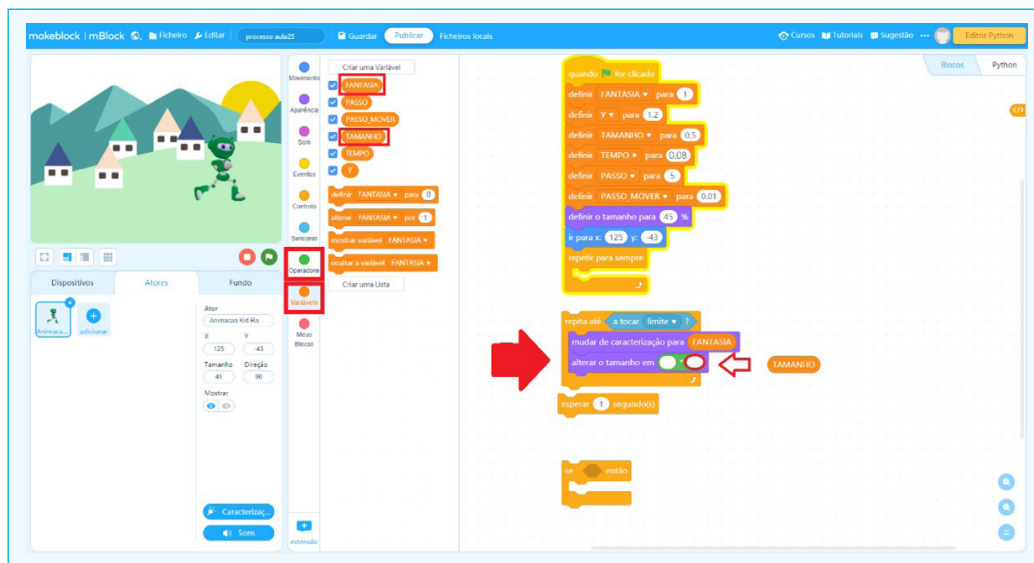


Figura 17 - Definição de mudança de fantasia




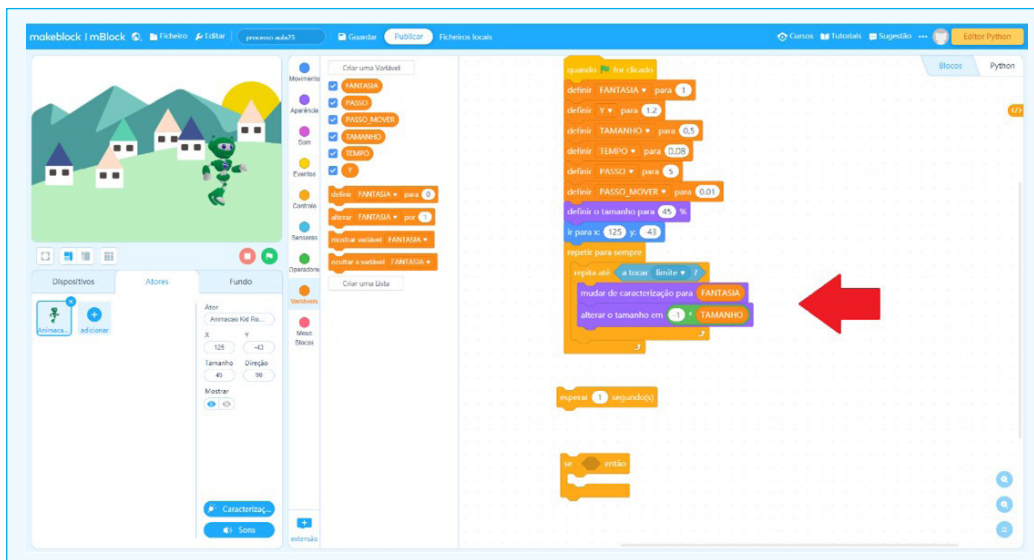
Preencha no primeiro espaço do bloco operador o valor -1 . No segundo espaço, encaixe o bloco de variáveis **Tamanho** tamanho, conforme mostrado na figura 18. Estamos definindo com estes comandos que o tamanho da mascote diminuirá $-1 *$ multiplicando o valor da variável que foi definido 0,5 a cada mudança de fantasia. Encaixe na sequência da programação.

Figura 18 - Diminuição da fantasia



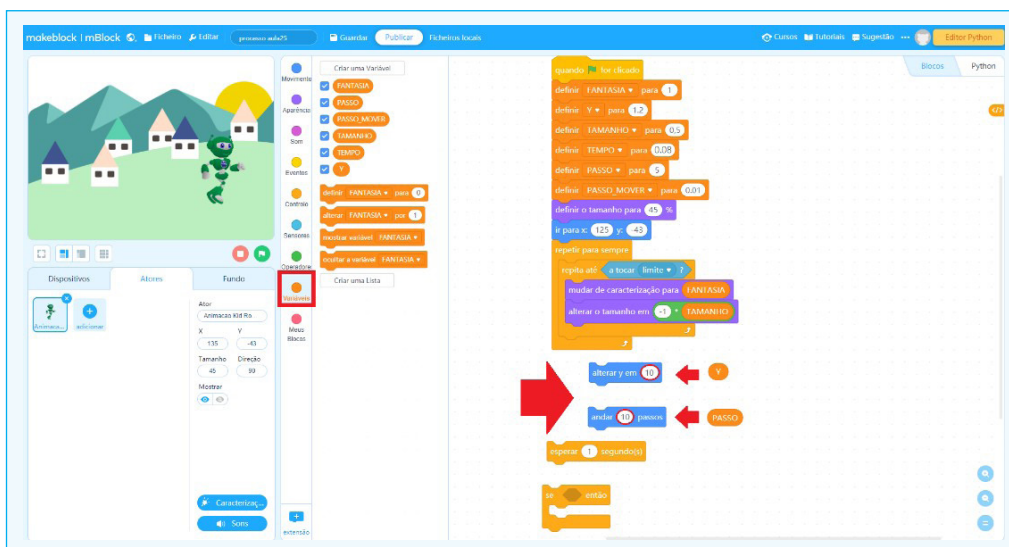
No próximo passo, vamos adicionar os blocos de movimento: **alterar x em 10** e **andar 10 passos** (figura 19). São responsáveis por definir os passos que a mascote se moverá na vertical (Y) e na horizontal (X), conforme valor da variável que criamos para Y. Depois ao andar o tamanho do passo definido pela variável que criamos - **passo**.

Figura 19 - Definição do movimento vertical



Encaixe as variáveis nos espaços de cada bloco de movimento **y** e **passo**, conforme mostrado na figura 20.

Figura 20 - Definindo os padrões de movimentos



Encaixe na sequência da programação, conforme figura 21.

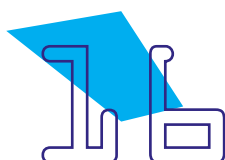
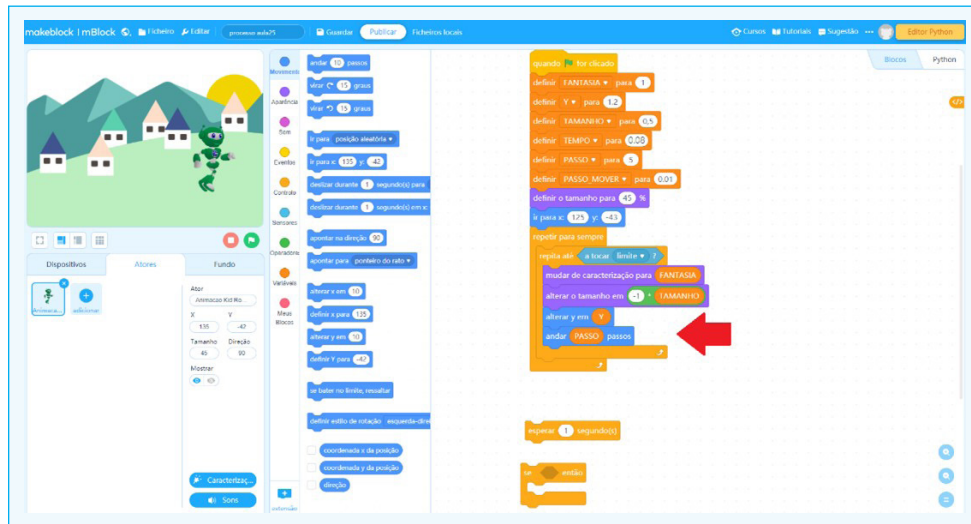


Figura 21 - Encaixe da programação do deslocamento




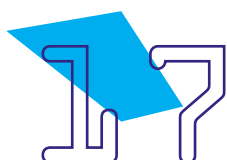
Agora, vamos adicionar os seguintes blocos de variáveis: **alterar passo por 1** (alterar passo por 1), **passo_mover** (passo_mover) (figura 22), definindo que a cada passo na horizontal (X) também se desloca na vertical (Y).

A variável **passo** é quando o deslocamento é na horizontal (X) e **passo_mover** é a variável que determina a diminuição dos passos na horizontal (desaceleração).

Figura 22. Definição da variável Passo.

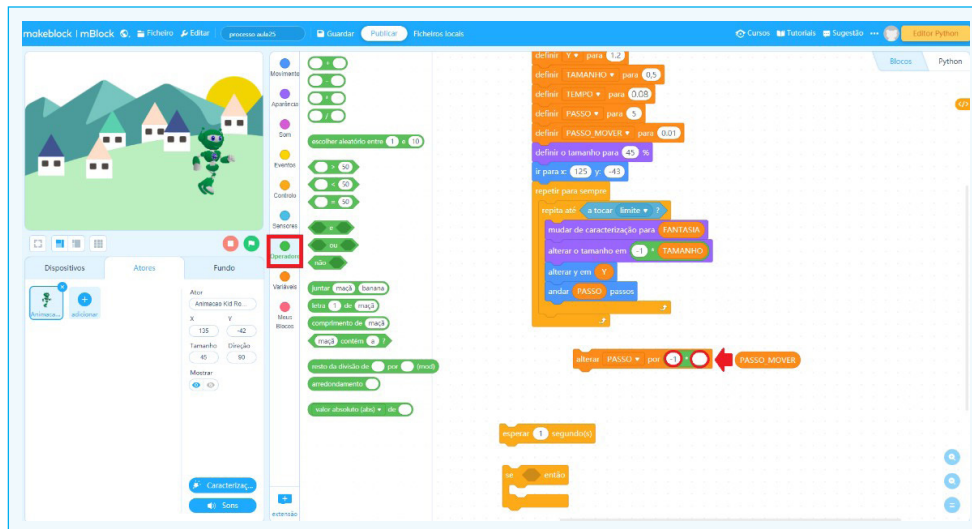


Vamos adicionar um bloco operador  para condicionar um valor para diminuir a velocidade do passo. Preencha o primeiro espaço com -1 encaixe a variável **passo_mover** no segundo espaço deste bloco (figura 23).



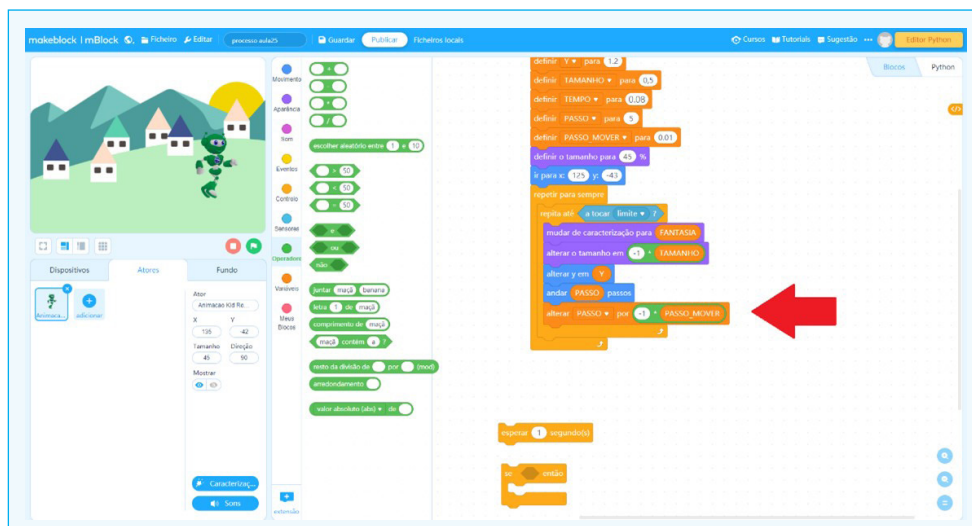
Este comando está dizendo que a cada passo a mascote BIT irá diminuir para facilitar a visualização da mascote em seu distanciamento.

Figura 23 - Adição operador definindo cada passo

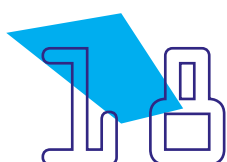


Após acoplados os blocos, encaixe estes comandos na sequência da programação (figura 24).

Figura 24 - Encaixe do comando de passo na programação



Vamos, agora, utilizar o bloco de controle: **esperar 1 segundo** `esperar 1 segundo(s)` que separamos. Selecione a variável **tempo** `tempo`, encaixando no espaço disponível do bloco (figura 25). Depois o encaixe na sequência da programação.

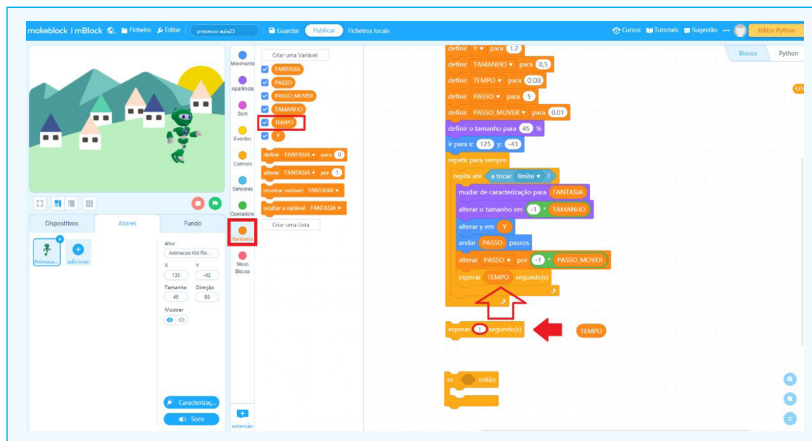


Este comando padroniza o tempo de 0.08 para se mover e trocar de fantasia.

Lembrete

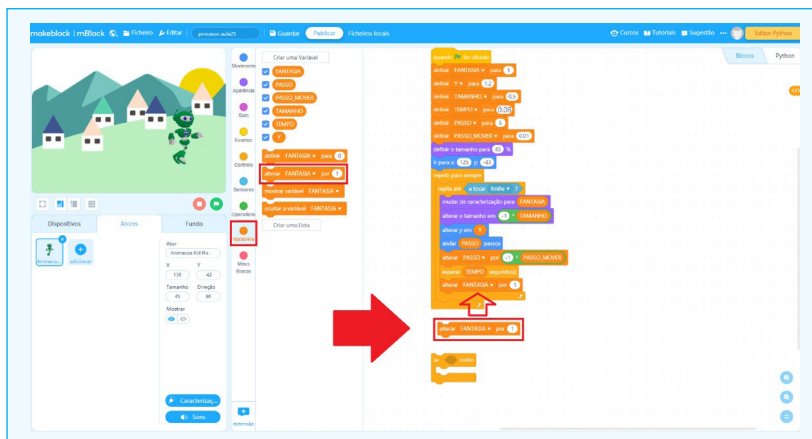
Quando se trabalha com milissegundos, é preciso adotar o padrão americano de ponto (.) ao invés de vírgula (,) para as casas decimais, pois o software mBlock utiliza este sistema de pontuação para os números.

Figura 25 - Definição de tempo de locomoção e troca de fantasia



Vamos adicionar um bloco de variáveis: **alterar fantasia por 1** alterar fantasia por 1, que alterará as fantasias uma em sequência da outra, dando a sensação de movimento (figura 26).

Figura 26 - Definição da fantasia inicial de ida





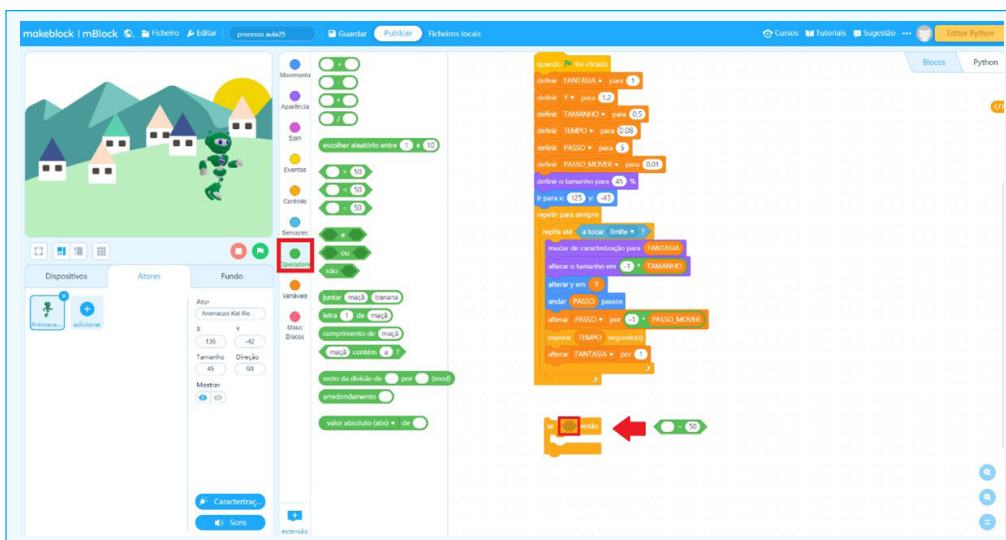
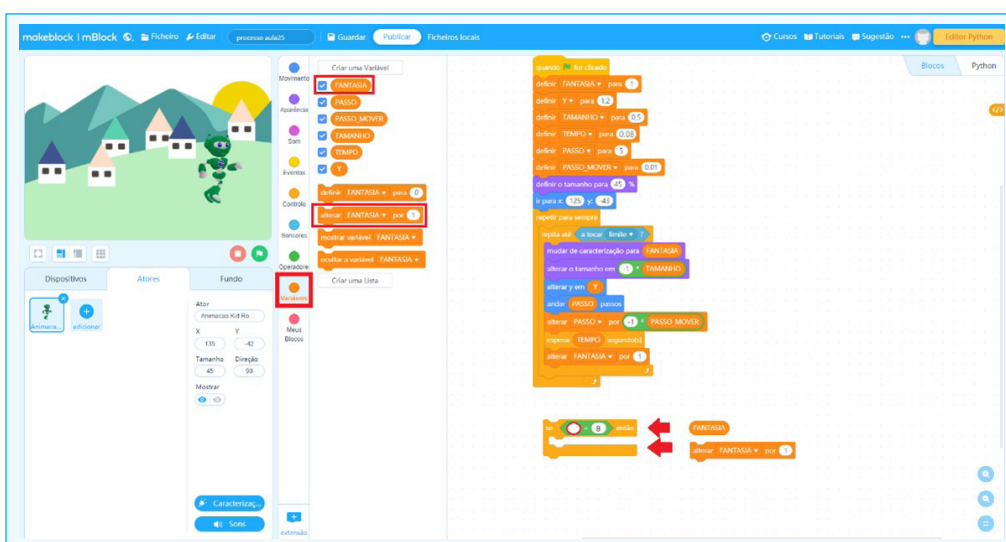
Utilizaremos, agora, o bloco de controle: **se então** , e selecionaremos o bloco operador  para encaixar no espaço que delimitará as fantasias de ida até 8, conforme figura 27.

Figura 27 - Delimitação das caracterizações de ida



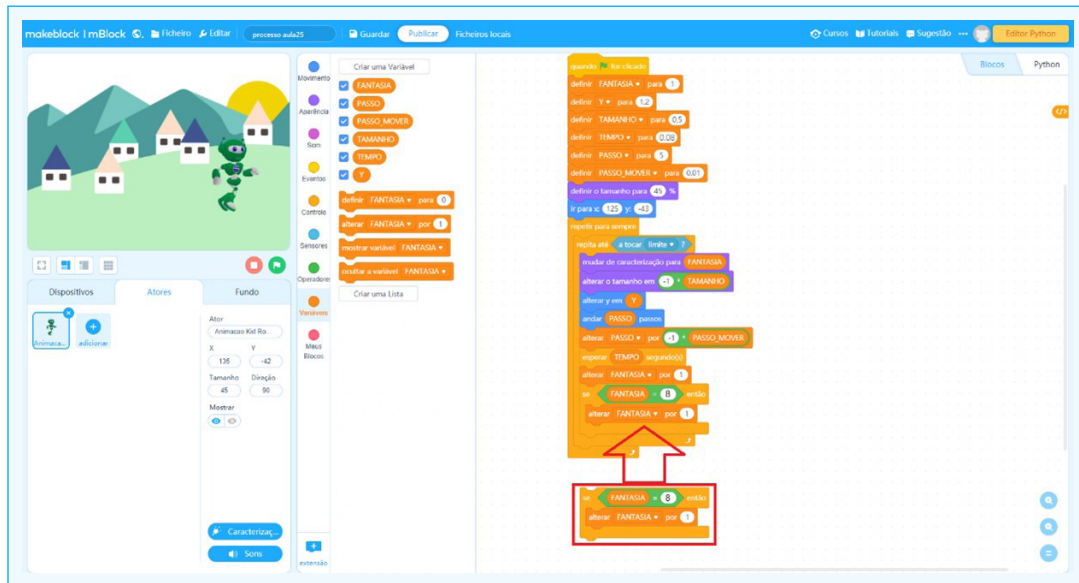
Perceba que este bloco operador tem dois espaços , onde um preencheremos com a variável **fantasia** e o valor 8 (figura 28).

Figura 28 - Definição da relação entre as fantasias



Quando a mascote BIT chegar à fantasia 8, voltará à fantasia 1, continuamente, até chegar o limite da tela. Encaixe estes comandos na sequência da programação (figura29).

Figura 29 - Definição de limite da fantasia de ida

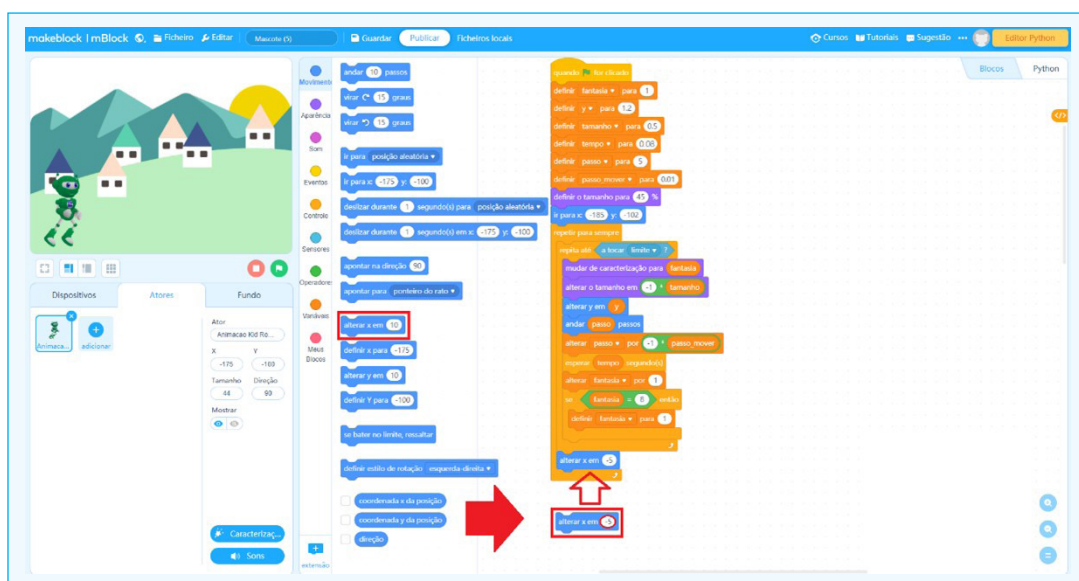


Até aqui, todos estes comandos indicam que a mascote BIT está indo, ou seja, se desloca da esquerda até tocar o limite da tela direita.

Agora, faremos os mesmos comandos para quando ela tocar o limite da tela, ela volte, mas com algumas especificidades.

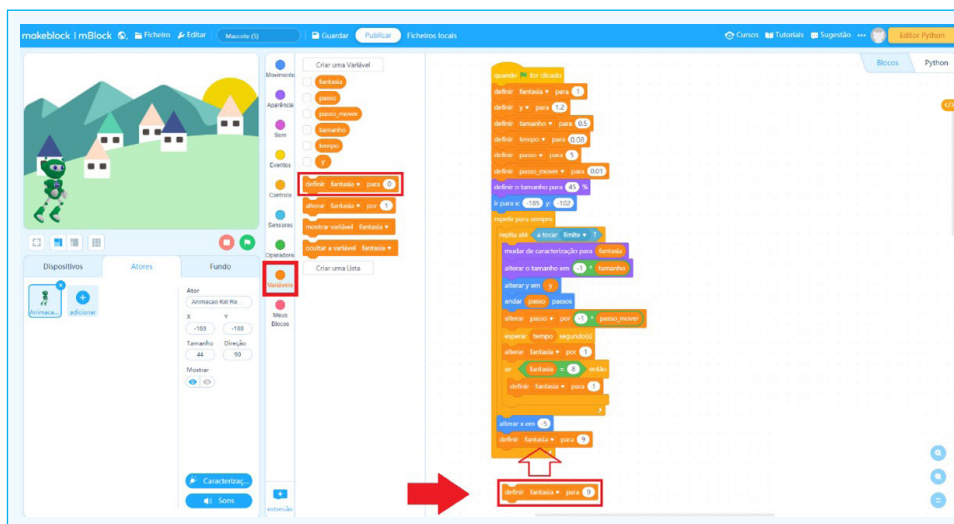
Para tanto, vamos adicionar o bloco de movimento: **alterar x em -5** que fará a mascote BIT desencostar da tela para que inicie seu retorno (figura 30).

Figura 30 - Desencostar da tela



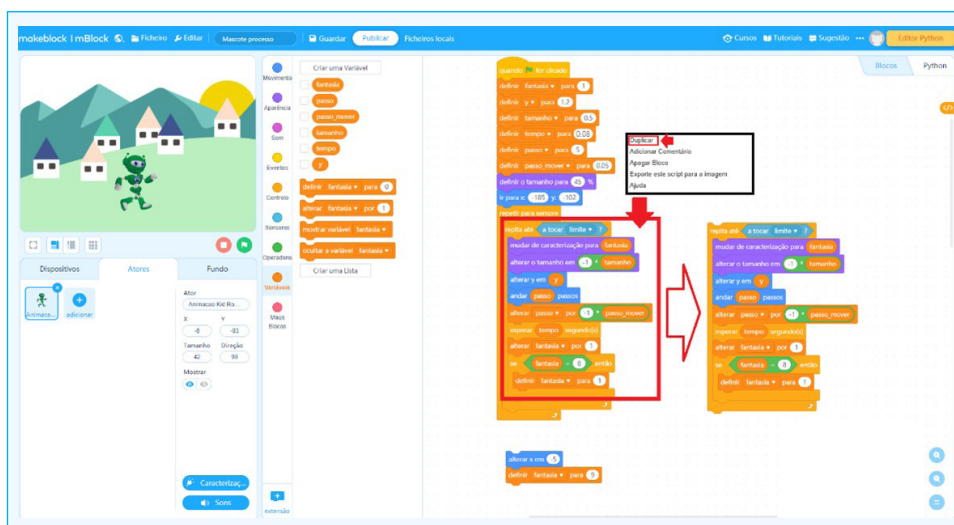
Vamos definir a fantasia de retorno, que deverá alternar as caracterizações do BIT, agora, está virado no sentido oposto a partir da fantasia 9. Então, adicione o bloco de variáveis: **definir fantasia para** `definir fantasia para` e preencha no espaço o valor 9, conforme figura 31.

Figura 31 - Definindo a fantasia de retorno



Como a animação da mascote BIT, nesta aula, é correr nos dois sentidos continuamente, a programação de retorno é semelhante à da ida. Para facilitar, copiaremos um conjunto de comandos de ida. Duplique os seguintes comandos, clicando com o botão direito em cima dos blocos, selecione a opção **duplicar**, conforme mostrado na figura 32.

Figura 32 - Duplicação do trecho da programação



Criando animação no mBlock

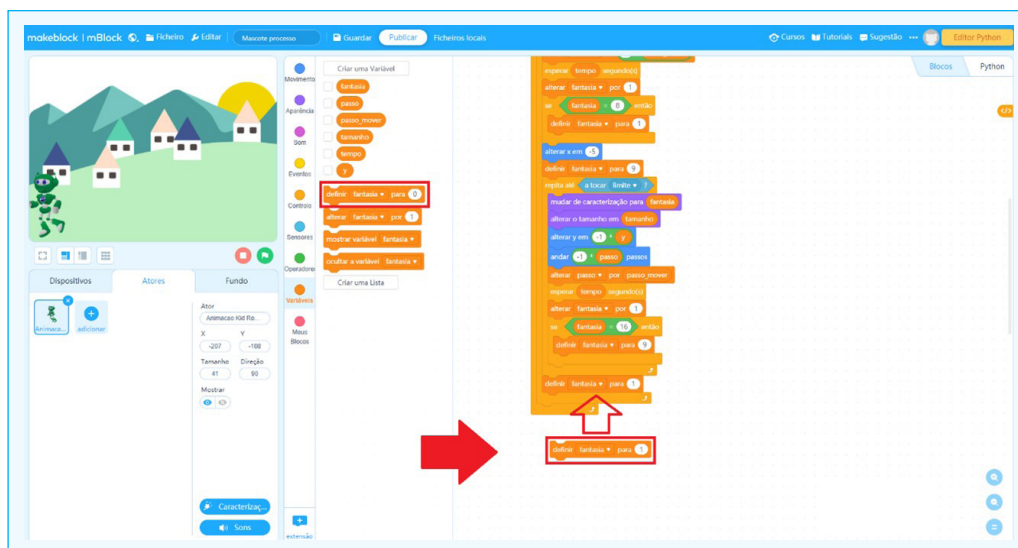
Contudo, vamos alterar alguns dados dentro deste conjunto de blocos. Percebam que o tamanho da mascote não irá diminuir, ao contrário, ele deverá aumentar, para causar a impressão de proximidade a cada passo que retornar (figura 33).

Figura 33 - Comandos de deslocamento da mascote indo e voltando



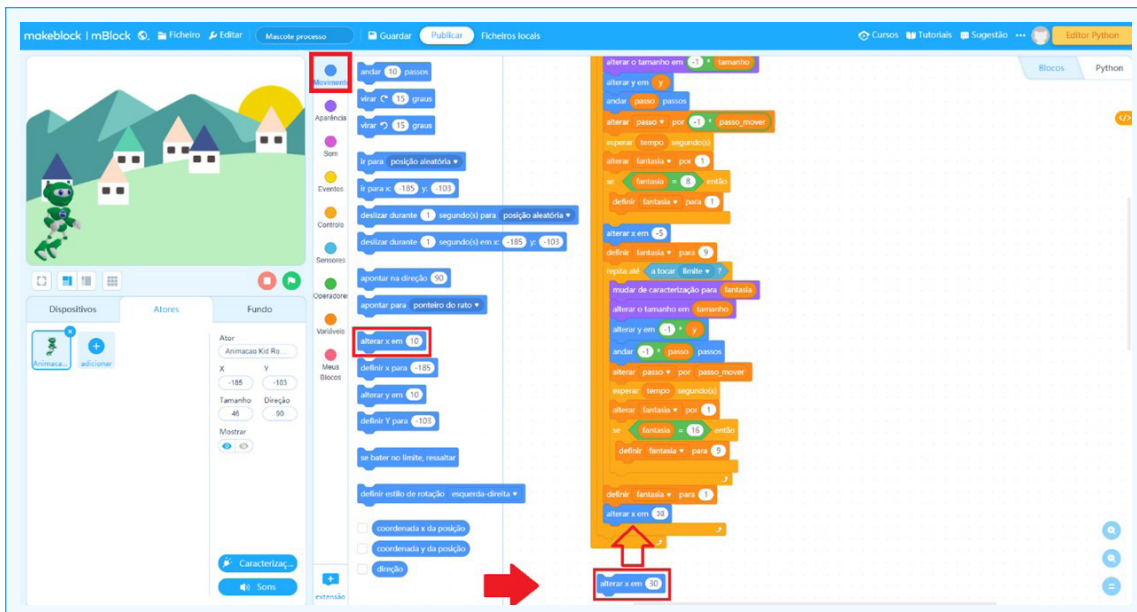
Para que a animação seja contínua, no retorno quando chegar no limite da tela, deverá recomeçar o processo e mudar a direção da mascote BIT. Para isso, definiremos que ela precisa mudar para a fantasia 1, adicionando o bloco: **definir fantasia para** definir fantasia para 1 (figura 34).

Figura 34 - Alteração da fantasia para reinício



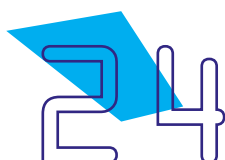
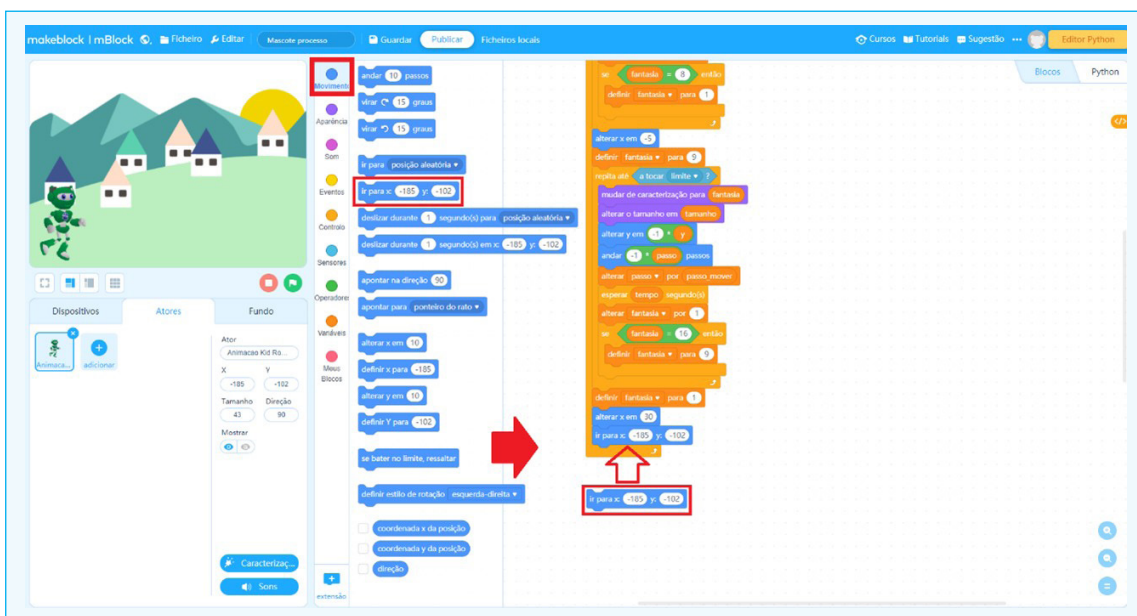
Adicione o bloco: **alterar x em 10** `alterar x em 10` e altere o valor para 30, agora, para o B1T desencostar da tela do lado direito, conforme figura 35.

Figura 35 - Mascote desencostar da tela à direita



E para finalizar, vamos definir a posição da mascote na horizontal (X) do reinício, adicionando o bloco de movimento: **ir para x e y** `ir para x: 0 y: 0` (figura 36).

Figura 36 - Definição da posição do reinício



Pronto, sua animação está concluída.

Agora, cada vez que clicar na bandeira verde, irá começar sua animação. Você verá a mascote BIT correndo e voltando até os limites da tela (figura 37).

Figura 37 - Visualização da animação



Fonte: Makeblock, 2022

Que tal criar outros projetos como jogos e histórias? Para dar luz a sua imaginação, vamos aprimorar esta animação!



Desafio

- Que tal dar mais ações à mascote e adicionar um diálogo nesta animação?
- Que tal colocar som e animar mais ainda sua história?



3. Feedback e Finalização

- a. Confira e mostre ao seu colega de classe a sua criação da animação;
- b. Analise e troque informações se os projetos desenvolvidos pelos seus colegas seguiram as instruções e estão funcionando de acordo com o projetado;
- c. Reveja se você entendeu ao funcionamento da programação;
- d. Reflita se as seguintes situações ocorreram:
 - i. Você e os seus colegas trocaram ideias no momento da criação e montagem da animação?
 - ii. Você teve problemas na montagem desta animação? Quais? Quais foram as soluções encontradas?





Referências

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**. Brasília, 2018. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf. Acesso em: 18 maio 2022.

MAKEBLOCK. mBlock. **Programação em blocos**. Disponível em: <https://ide.mblock.cc/>. Acesso em: 23 maio 2022.

SANTANA, Bianca Leite; ARAUJO, Luís Gustavo de Jesus; BITTENCOURT, Roberto Almeida. **Computação & Sociedade**. Livro do Professor - 8º ano. Projeto Computação Fundamental. Versão 1.0. Disponível em: https://ia802803.us.archive.org/7/items/computacao_e_sociedade_livro_do_professor/computacao_e_sociedade_livro_professor_8_ano_V1.pdf. Acesso em: 10 maio 2022.



DIRETORIA DE TECNOLOGIAS E INOVAÇÃO (DTI)
COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)

EQUIPE ROBÓTICA PARANÁ

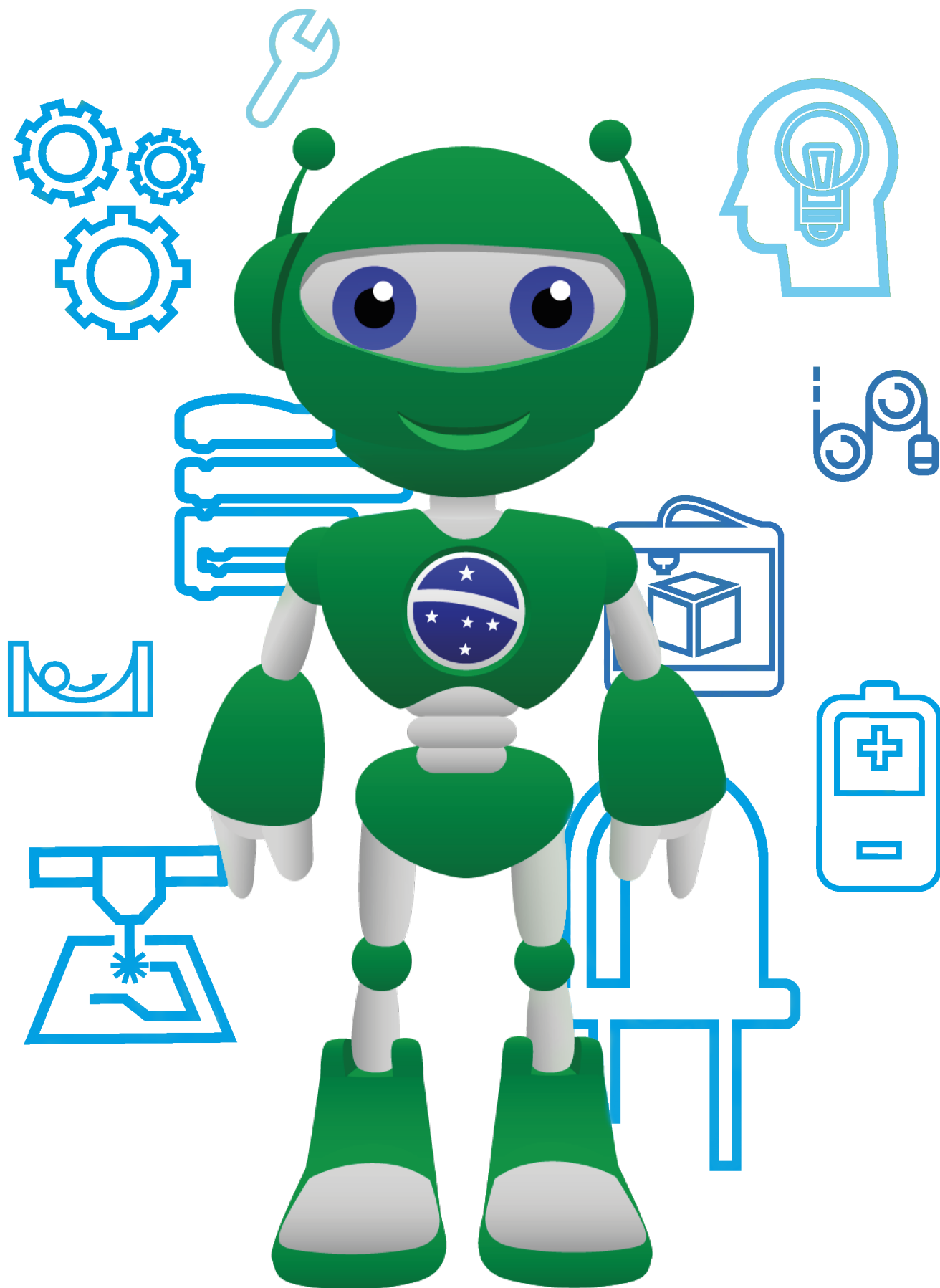
Adilson Carlos Batista
Cleiton Rosa
Darice Alessandra Deckmann Zanardini
Edna do Rocio Becker
Marcelo Gasparin
Michelle dos Santos
Ricardo Hasper
Roberto Carlos Rodrigues
Simone Sinara de Souza

Os materiais, aulas e projetos da “Robótica Paraná”, foram produzidos pela Coordenação de Tecnologias Educacionais (CTE), da Diretoria de Tecnologia e Inovação (DTI), da Secretaria de Estado da Educação e do Esporte do Paraná (Seed), com o objetivo de subsidiar as práticas docentes com os estudantes por meio da Robótica.

Este material foi produzido para uso didático-pedagógico exclusivo em sala de aula.



Este trabalho está licenciado com uma Licença
Creative Commons – CC BY-NC-SA
[Atribuição - NãoComercial - Compartilha Igual 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)



Diretoria de Tecnologia e Inovação