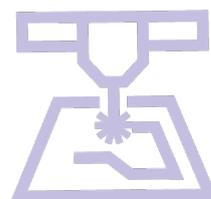
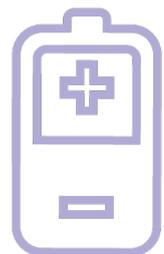
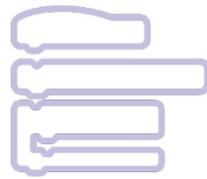


# ROBÓTICA

Primeiros Passos

Módulo 2



AULA 07

Criando jogos no  
mBlock [labirinto] II

**GOVERNADOR DO ESTADO DO PARANÁ**

Carlos Massa Ratinho Júnior

**SECRETÁRIO DE ESTADO DA EDUCAÇÃO**

Roni Miranda Vieira

**DIRETOR DE TECNOLOGIA E INOVAÇÃO**

Claudio Aparecido de Oliveira

**COORDENADOR DE TECNOLOGIAS EDUCACIONAIS**

Marcelo Gasparin

**Produção de Conteúdo**

Adilson Carlos Batista

Cleiton Rosa

**Validação de Conteúdo**

Cleiton Rosa

**Revisão Textual**

Orlando de Macedo Junior

**Normalização Bibliográfica**

Ricardo Hasper

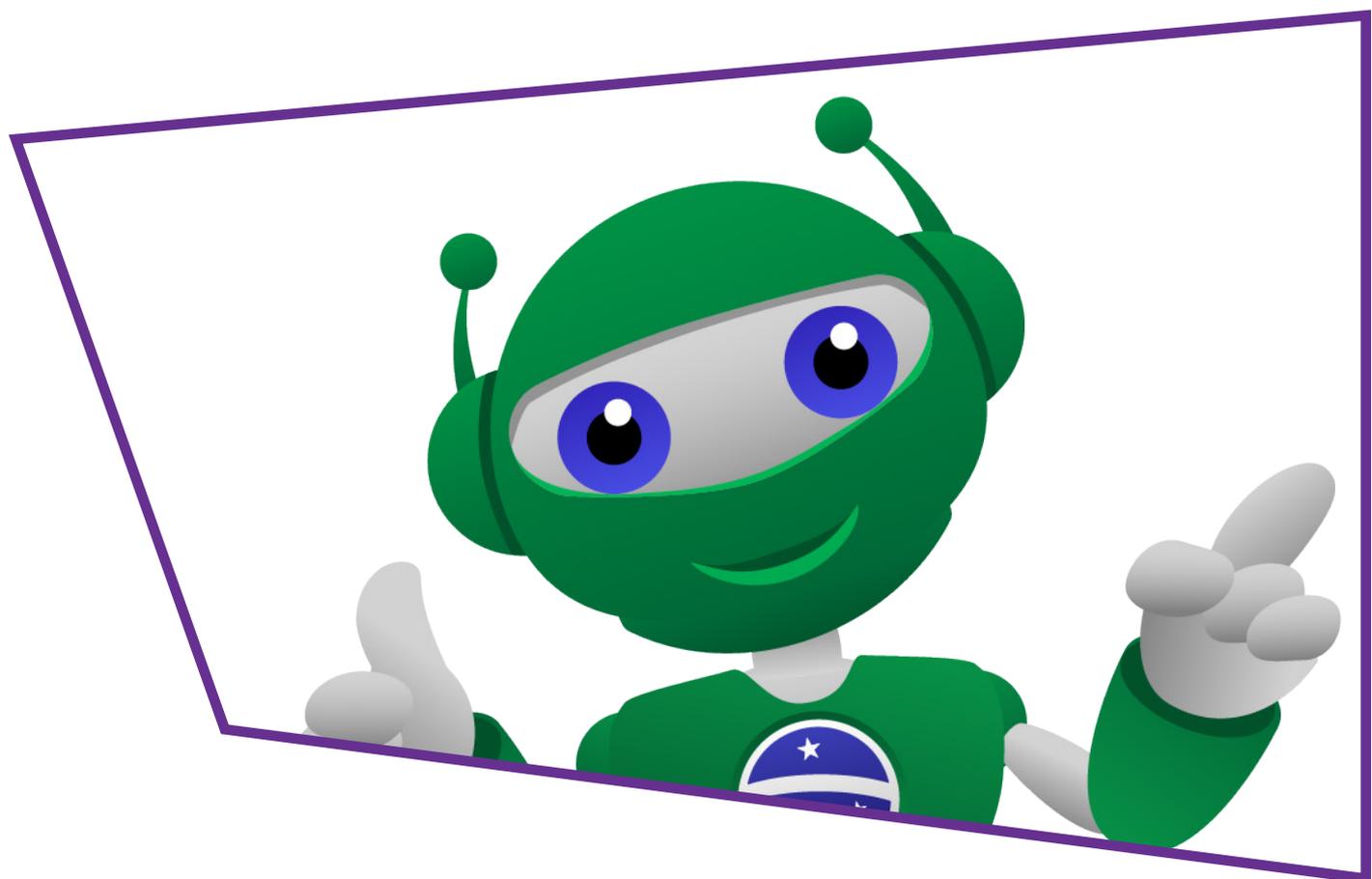
**Projeto Gráfico e Diagramação**

Edna do Rocio Becker

2023

# SUMÁRIO

<b>Introdução</b>	<b>2</b>
<b>Objetivos desta aula</b>	<b>2</b>
<b>Competências gerais previstas na BNCC</b>	<b>3</b>
<b>Habilidades do século XXI a serem desenvolvidas</b>	<b>4</b>
<b>Lista de materiais</b>	<b>4</b>
<b>Roteiro da aula</b>	<b>5</b>
1. Contextualização	5
2. Conteúdo	6
3. Feedback e finalização	26
<b>Referências</b>	<b>26</b>





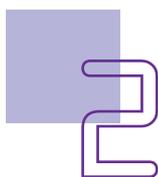
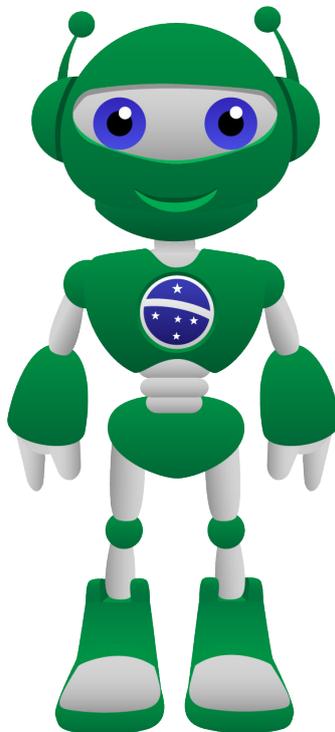
## Introdução

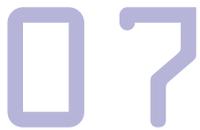
Na aula anterior, você conheceu o joystick shield e as funções desse componente e fez um labirinto para construir o jogo de mesmo nome, nesta aula, você será levado à programação do jogo a partir do software mBlock.



## Objetivos desta aula

- Compreender a linguagem de programação em blocos;
- Explorar recursos disponíveis pelo mBlock;
- Produzir um jogo utilizando o joystick shield e o Arduino como controle das ações no jogo.





# Criando jogos no mBlock [Labirinto] II



## Competências gerais previstas na BNCC

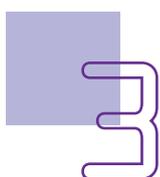
**[CG02]** - Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas)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



# Criando jogos no mBlock [labirinto] II



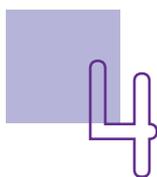
## Habilidades do século XXI a serem desenvolvidas

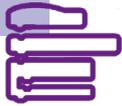
- Pensamento crítico;
- Afinidade digital;
- Resiliência;
- Resolução de problemas;
- Colaboração;
- Comunicação;
- Criatividade.



## Lista de materiais

- Computador;
- Software mBlock;
- 1 joystick shield;
- 1 placa Arduino.





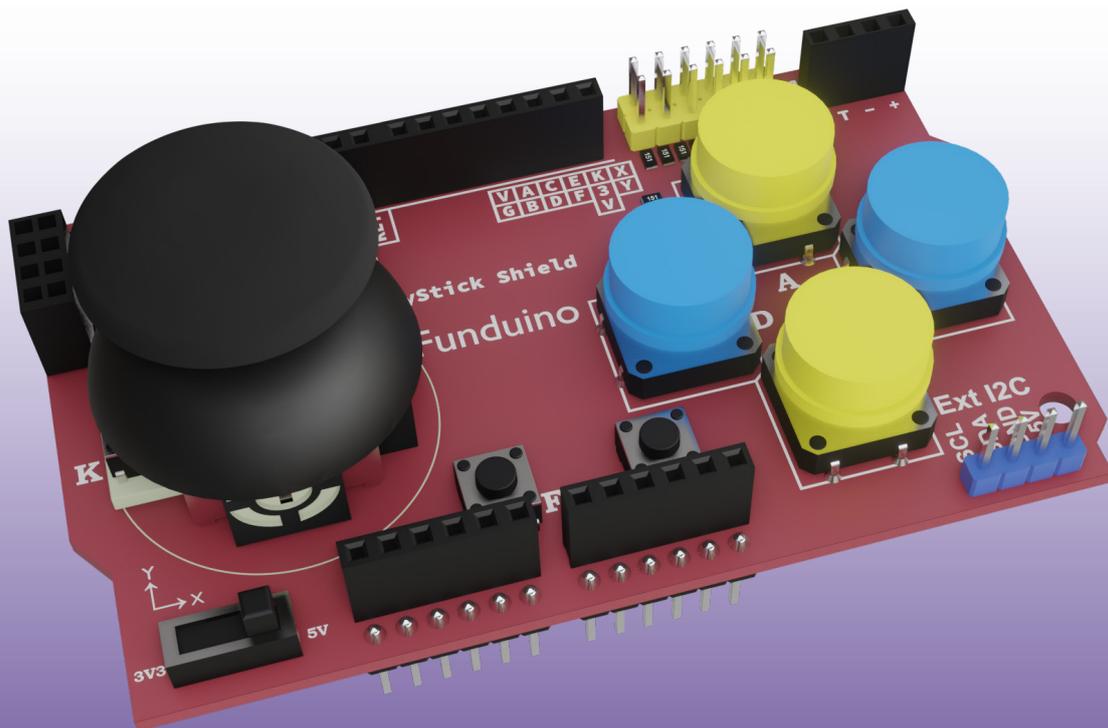
## Roteiro da aula

### 1. Contextualização

Muitas pessoas não gostam de utilizar o teclado para jogar devido à dificuldade encontrada em manipular as ações apenas apertando algumas teclas. E você, tem essa dificuldade?

Nesta aula, iremos continuar o trabalho com o joystick shield e você fará a programação do jogo labirinto para que a personagem possa ser controlada pelo joystick.

Figura 1 - Joystick shield



Fonte: SEED/ DTI/ CTE, 2023

# Criando jogos no mBlock [labirinto] II

## 2. Conteúdo

### 2.1 Retomando o joystick shield

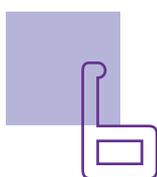
Como você pode ter percebido, quando acoplamos o joystick shield ao Arduino, os pinos do joystick são conectados a algumas portas do Arduino. Isso indica que a haste e os botões estarão ligados a portas específicas (analógicas ou digitais). Vamos conhecê-las?

Movimento horizontal (eixo X)	A0
Movimento vertical (eixo Y)	A1
Botão A	D2
Botão B	D3
Botão C	D4
Botão D	D5
Botão E	D6
Botão F	D7
Botão K	D8

### 2.2 Dando *start* à programação!

Para começarmos a pensar sobre o nosso jogo, será preciso utilizar o labirinto que você produziu na aula anterior. Entre no <mBlock>, clique em <FICHEIRO>, depois <importar do seu computador>, procure pelo nome do arquivo que você salvou na aula passada. Clique neste nome e o arquivo com o desenho se abrirá.

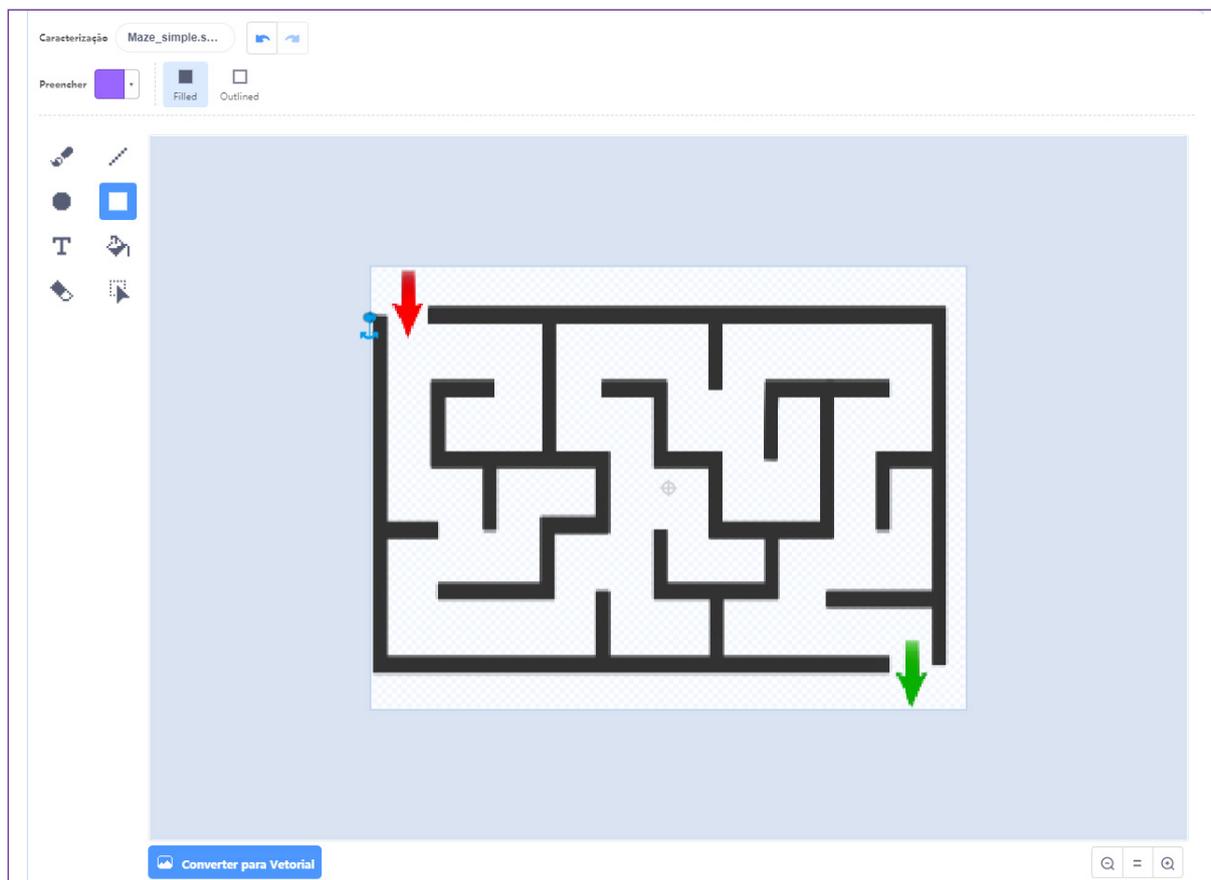
Caso não consiga abrir o arquivo, no mBlock mesmo, clique em <FUNDO>, e, na sequência, utilizando o QUADRADO, desenhe um labirinto de acordo com a sua imaginação, o importante é que o desenho apresente uma única entrada e uma única saída (figura 2). Será este labirinto que a personagem terá que percorrer em menos tempo possível para que o jogador ganhe o jogo, nesse sentido, quanto mais caminhos e obstáculos apresentar, mais difícil será o jogo.



# Criando jogos no mBlock [labirinto] II

Para desenhar a flecha indicativa de entrada e saída, você poderá utilizar o QUADRADO para fazer o corpo da flecha e a LINHA para fazer as pontas.

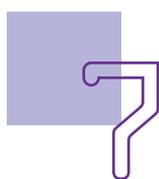
Figura 2 – Desenho do labirinto



Fonte: mBlock, 2022

Tendo essa referência de como será o jogo, poderemos refletir sobre o que precisamos fazer na programação e criar um pequeno roteiro, isso irá te ajudar no passo a passo da criação e mobilização de ideias.

Então, vamos lá! O que será preciso para esquematizar o jogo?



# Criando jogos no mBlock [labirinto] II

## Esquematisando:

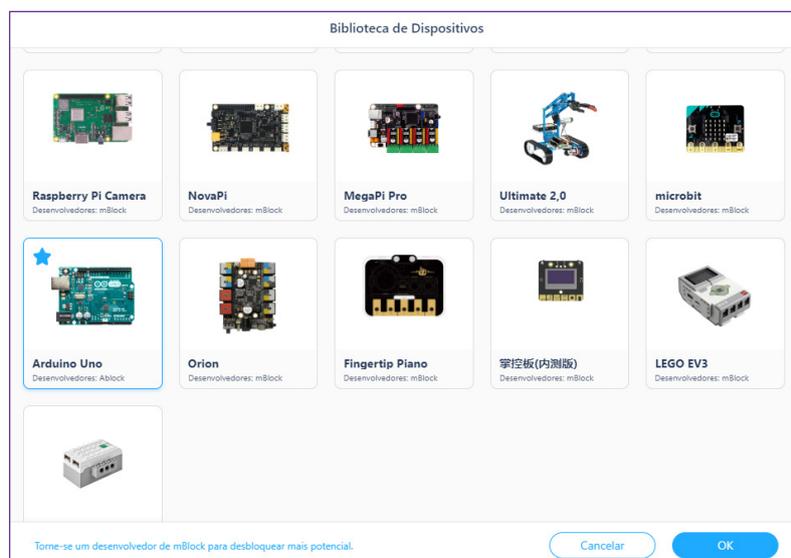
- Inserir o joystick para controlar a personagem;
- Programar o joystick para fazer os movimentos necessários;
- Criar uma personagem;
- Criar comandos para ela poder se locomover tanto para esquerda, quanto para a direita, mover para cima e para baixo.
- Criar um tempo para percorrer o labirinto.
- Criar um obstáculo para que a personagem não ultrapasse as linhas.

## 2.2 Inserindo o Dispositivo joystick shield:

Para inserir esse componente, será necessário, primeiramente, verificar se o Arduino Uno está selecionado para executar os comandos.

Entre em <DISPOSITIVOS> + <BIBLIOTECA DE DISPOSITIVO> + <SELECIONE ARDUINO UNO>, deixe o dispositivo selecionado, marcado com uma “estrelinha”. Assim, toda vez que você abrir o mBlock o Arduino já estará disponível para uso. Clique em <OK>.

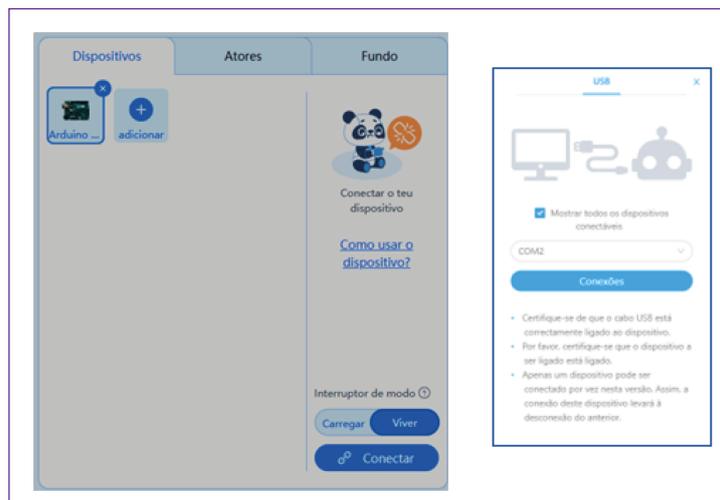
Figura 3 – Selecionando o Arduino



# Criando jogos no mBlock [labirinto] II

Com um cabo USB, faça a conexão do dispositivo joystick shield com o computador, deixe selecionado em <DISPOSITIVO> no interruptor de modo <VIVER> e, na sequência, clique em <CONECTAR>. Uma nova janela irá abrir para você confirmar as conexões, selecione <MOSTRAR TODOS OS DISPOSITIVOS CONECTÁVEIS>, clique em <CONEXÕES>. O dispositivo joystick shield estará apto para receber a programação.

Figura 4 - Ativando o dispositivo joystick shield



Após a conexão, você verá que o dispositivo abrirá diversos comandos para você trabalhar a programação deste componente, uns estarão na cor azul e outros na cor cinza, esses últimos estão nesta cor porque não serão utilizados, pois o mBlock se adapta de acordo com o dispositivo instalado.

ler pino digital 9

ler pino analógico (A) 0

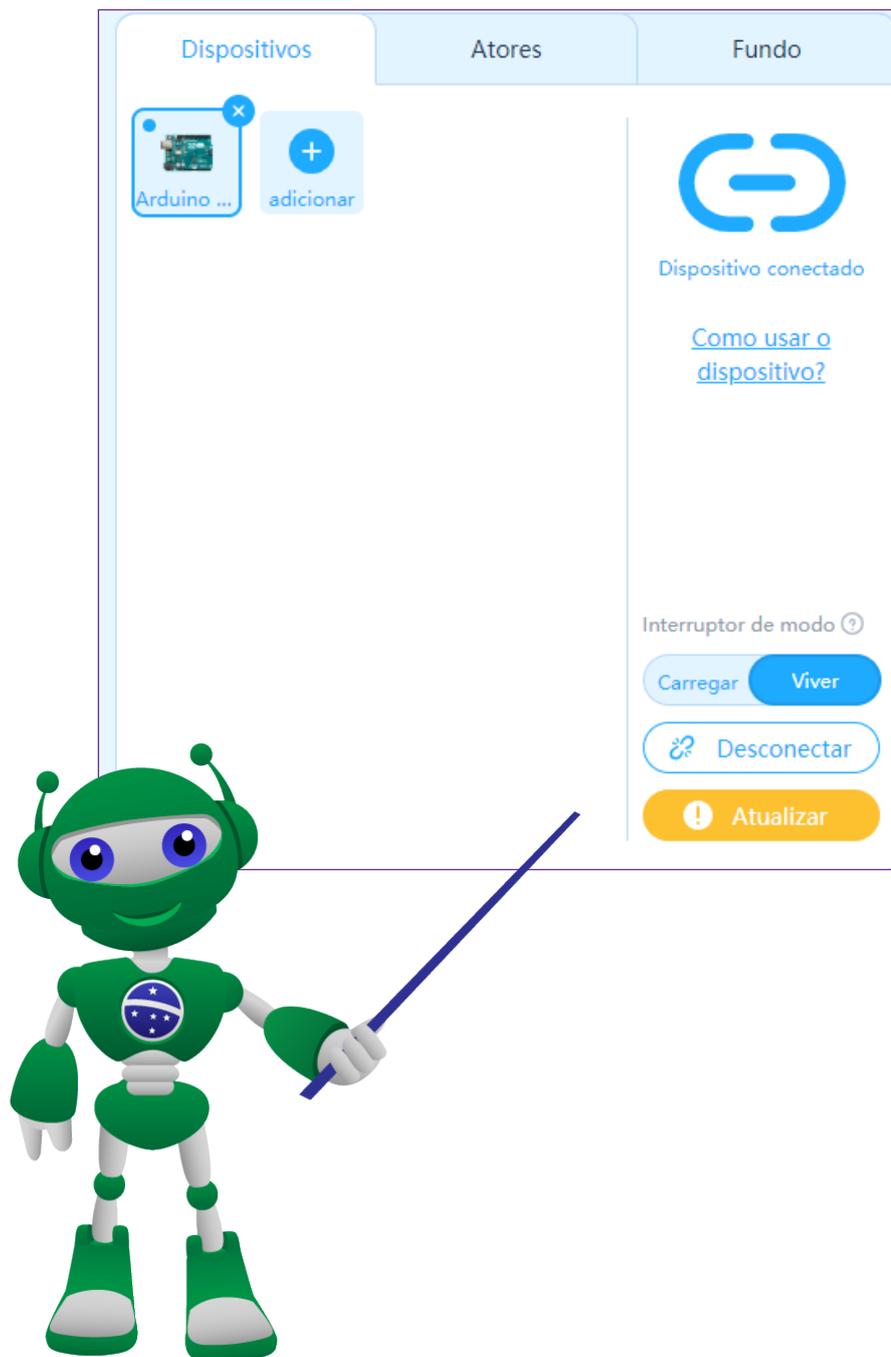
definir a saída do pino digital 9 como alto



# Criando jogos no mBlock [labirinto] II

**Atenção:** caso apareça em **Dispositivo** a palavra <atualizar>, clique nela para que o dispositivo seja atualizado. Selecionar <Mostrar todos os dispositivos conectáveis> e clicar em <conexões>.

Figura 5 - Atualização do dispositivo



## 2.3 Programando o joystick

### Dar o *start*

Antes de começarmos a programação, uma pergunta: você se lembra qual Bloco é utilizado para dar o início das ações? Se você pensou naquele que possui a bandeirinha verde, parabéns! Você já compreendeu que precisamos de um comando para darmos início nas ações de programação. Assim, o primeiro bloco será esse, então, clique em <EVENTO> e escolha o bloco <quando\_\_\_for clicado>.

Figura 6 - Bloco de início



Fonte: mBlock, 2022

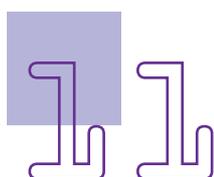
Para a programação do jogo, também se faz necessário compreender que as ações serão repetidas para sempre, tipo dar o *start*, caminhar, ir para frente ou para trás, assim, devemos abrir com o bloco de programação que define as ações como repetição - clique em <CONTROLO> e escolha <repetir para sempre>.

Figura 7 - Bloco controle de repetir para sempre



O que será repetido para sempre?

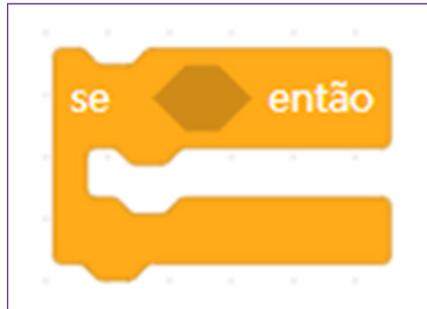
Todas as ações criadas para movimentar a personagem Panda. Nesse sentido, os comandos de programação do joystick que vier na sequência entrarão neste bloco de repetição. E toda vez que abrir uma nova ação, será necessário abrir novamente esse bloco de repetição. Lembre-se! Você está programando apenas o joystick ainda, função a função.



# Criando jogos no mBlock [labirinto] II

O primeiro comando é dar *start* ao jogo, para isso, será preciso escolher qual botão (push button) fará isso no joystick e indicá-lo no comando da programação. Para isso, será necessário criar uma condicional do tipo - “se apertar o botão X” então o jogo iniciará.

Figura 8 - Bloco controle - condicional

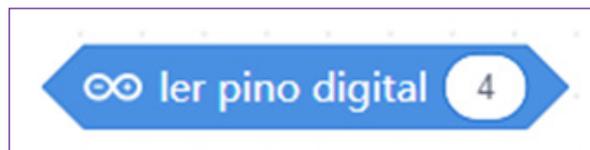


Ressignificando essa informação em linguagem técnica - se ler o valor do pino digital da porta X (4, por exemplo) ele dará o *start*. Esse valor ainda será definido.

Para fazer controle das portas analógicas e digitais do Arduino, precisamos acessar no mBlock o PIN.

Clique em <PIN>, arraste para dentro da condicional o bloco <ler o pino digital \_\_\_>.

Figura 9 - Bloco de pino



Antes disso, será preciso definir o valor, clique em <OPERADORES>, arraste o operador que indica <\_\_\_ = \_\_\_>.

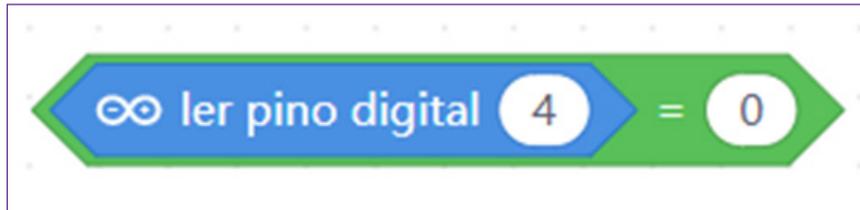
Figura 10 - Bloco de operadores



# Criando jogos no mBlock [labirinto] II

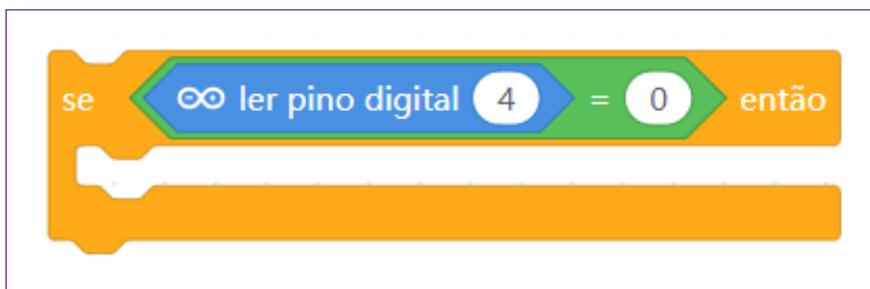
Veja que esse operador apresenta dois parâmetros <\_\_\_=\_\_\_>. No primeiro parâmetro, você deve indicar a porta/pino, no nosso caso, será o pino 4, então, arraste o bloco para dentro deste parâmetro e o segundo será o número 0, conforme figura 11:

Figura 11 - Bloco operador e de pinagem



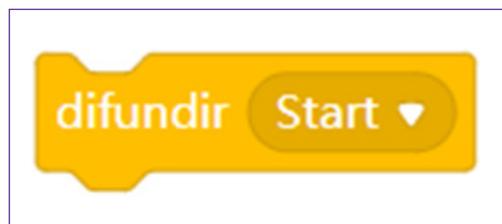
Se o valor lido no pino digital 4 for = 0, então faça o que está dentro desta condicional:

Figura 12 - Blocos agrupados



Mande a mensagem *start* para todos os atores/elementos da programação. Entre em <EVENTOS> arraste o bloco <difundir (espalhar) *Start*>.

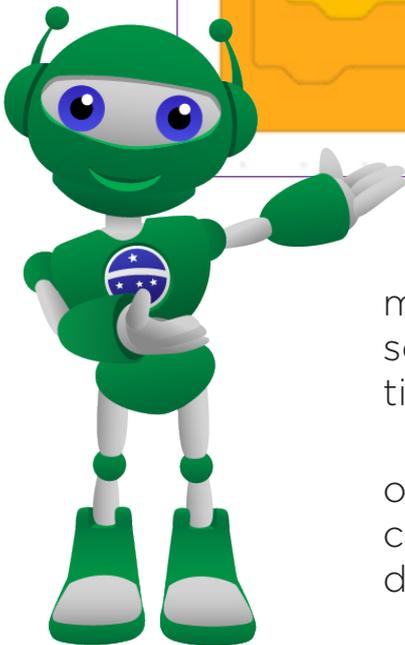
Figura 13 - Bloco de evento - difundir



# Criando jogos no mBlock [labirinto] II

Após este bloco, finalizamos a primeira sequência de comandos para o joystick - dar o *start*. A programação com todos os blocos ficará da seguinte maneira:

Figura 14 - Blocos de programação completa



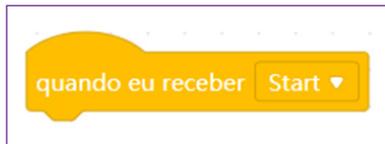
Lembre que o joystick não será utilizado somente para dar o *start* no jogo, mas também para ser usado para conduzir a personagem, nesse sentido, será preciso programar os movimentos.

Quando o joystick receber o *start* o que irá ocorrer? Resposta - produzir os movimentos de caminhar, mover para a direita, mover para esquerda, subir, descer.

# Criando jogos no mBlock [labirinto] II

São essas informações que serão programadas nos blocos, assim, iniciamos com a informação <quando eu receber *start*>.

Figura 15 - Blocos de programação completa



Como a personagem irá fazer uma fala de início, devemos pedir para esperar alguns segundos.

Figura 16 - Blocos de controle - tempo



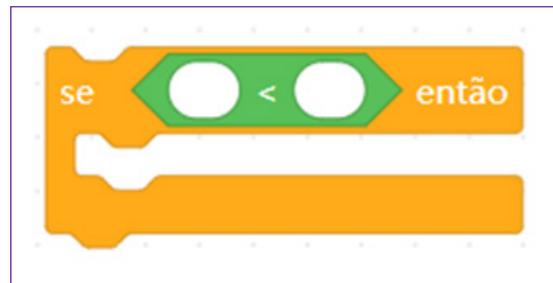
Na sequência, serão programadas as informações específicas de movimento, então, será preciso, novamente, trazer o bloco de <repetir para sempre> e inserir as outras programações dentro deste bloco.

Figura 17 - Blocos de controle - repetir



Para fazer as indicações das portas e valores dos pinos, será preciso abrir uma condicional para cada movimento que queremos para a personagem. Como você já fez isso antes, vamos trazer os dois blocos. Clique em <CONTROLO> e arraste o bloco de condicional <se \_\_\_\_ então> e, na sequência, clique em <OPERADORES> e arraste o operador <\_\_\_\_<\_\_\_\_> para inserirmos o primeiro movimento de esquerda.

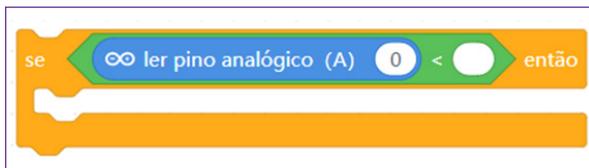
Figura 18 - Blocos de controle e operador integrados



# Criando jogos no mBlock [labirinto] II

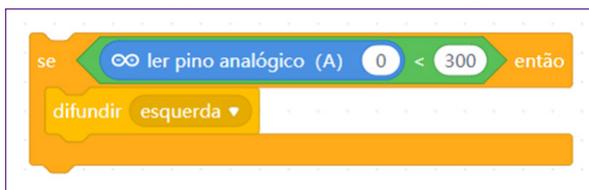
Para indicar qual pino analógico será utilizado, novamente, precisaremos de informações para preencher o primeiro parâmetro do operador. Clique em <PIN>, arraste o bloco <ler o pino analógico (A)\_\_\_\_>.

Figura 19 - Blocos de controle, operador e pin integrados



Com o bloco de indicação do pino inserido, vamos para os valores. Na porta Analógica (A) vamos manter o número 0 (zero) para controle do eixo X. Para indicar o movimento, inserir o número 300, ou seja, quando a haste do joystick se movimentar para um dos lados das coordenadas cartesianas, o valor será de 300, porém será preciso indicar os lados. Assim, insira os valores e clique em <EVENTOS>, arraste o bloco de programação <Difundir\_\_\_\_>, escolha para preencher a posição “esquerda”.

Figura 20 - Blocos de controle, operador, pin e eventos integrados



Seguindo esse mesmo princípio, deve-se fazer a programação do lado direito, porém com valor de 700.

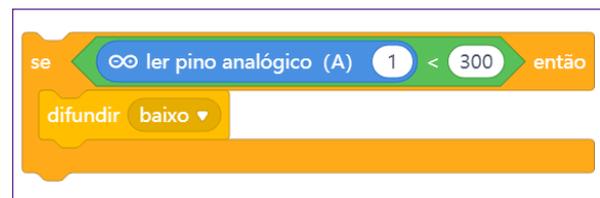
Figura 21 - Blocos de controle, operador, pin e evento integrados



Agora, vamos para o movimento de mover para cima e mover para baixo. Usa os mesmos blocos, porém altera o <difundir baixo>; <difundir cima> e a porta analógica para controle do eixo y.

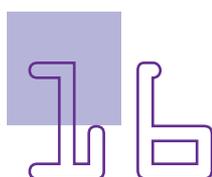
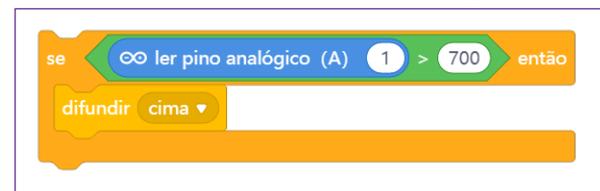
Mover para baixo:

Figura 22 - Blocos de controle, operador, pin e evento integrados



Mover para cima:

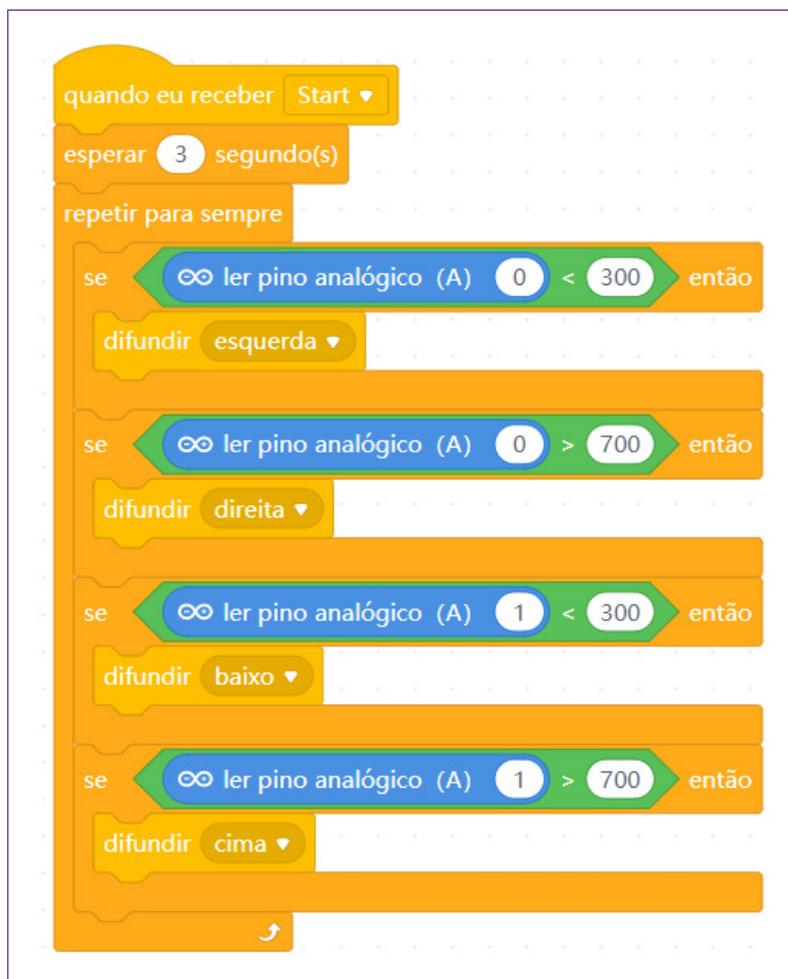
Figura 23 - Blocos de controle, operador, pin e evento integrados



# Criando jogos no mBlock [labirinto] II

Com todos os blocos prontos e valores estabelecidos, a programação completa deve ser:

Figura 24 – Blocos com a programação completa



## 2.4 Criar a personagem e a programação dela no jogo

Neste jogo, iremos utilizar o script Panda como personagem do jogo, como você aprendeu anteriormente, é possível inserir qualquer personagem da biblioteca de atores ou selecionar alguma que você criou ou inseriu – a mascote da Robótica, por exemplo.

Clique em <ATORES> e a personagem Panda estará selecionada, desta forma, é começar a programá-la de acordo com o planejado do roteiro.



# Criando jogos no mBlock [labirinto] II

Deixe a personagem com um tamanho ideal para percorrer o labirinto criado para o jogo, mova o Panda para o início do labirinto.

Figura 25 - Modificando o tamanho da personagem



Observe os valores que nós deixamos - Tamanho 30 e Direção 90. Em relação à posição no plano cartesiano, ao levar o Panda para o início do labirinto ficou com valor de X: - 212 e Y: 146. Isso significa, conforme estudado anteriormente, a 212 negativo do ponto zero (0) e 146 positivo.



# Criando jogos no mBlock [labirinto] II

Agora que a personagem está posicionada, será preciso programá-la de acordo com a programação e de acordo com os dados informados na programação do joystick.

Quando o Panda receber do joystick o comando de `<start>`, ele estará pronto. O que quero que ele faça no jogo?

Vamos programar seguindo o passo a passo:

- Fazer com que a personagem vá para o ponto de início;
- Fazer com que a personagem diga: “Preparar. Apontar. Já!”.
- Definir que a personagem se movimente para a esquerda, direita, cima e baixo sem atravessar as linhas criadas no labirinto;
- Definir um marcador de tempo para o trajeto da personagem;
- Fazer com que a personagem diga quando chegar ao final: “Cheguei!”, e pare a contagem do tempo.

Iniciamos a programação com o bloco que indica `<quando eu receber Start>`, porque foi esse bloco que programamos

no joystick. Entre em `<EVENTOS>`, arraste o bloco com essa informação:

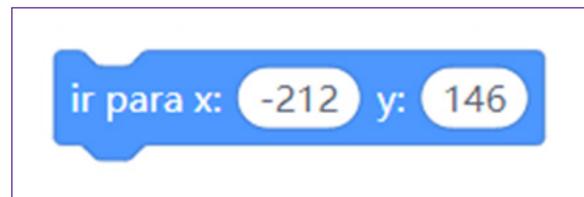
Figura 26 – Bloco de eventos



Na sequência, precisamos indicar a posição da personagem no início do labirinto - jogo, assim, clique em `<MOVIMENTO>`, arraste o bloco `<ir para x:____y:____>`.

Se você arrastou a personagem Panda para o início do labirinto, o bloco já irá marcar esse número. No caso da nossa programação os números ficaram da seguinte forma:

Figura 27 – Bloco de movimento



Ou seja, quando o jogador der o *start* no jogo, a personagem Panda irá para essa posição na tela, marcando o início do jogo.



# Criando jogos no mBlock [labirinto] II

Para ele falar “Preparar. Apontar. Já!” você deverá programar isso, porém antes, lembre-se que foi programado 3 segundos nestas falas no joystick.

Entre em <APARÊNCIA>, arraste o bloco <dizer\_\_\_\_\_> e complete o bloco com a fala <Preparar!> entre em <CONTROLE>, arraste o bloco que indica o tempo <esperar\_\_\_\_\_segundo(s)>, faça isso para as duas falas.

Figura 28 - Conjunto de blocos de fala e tempo



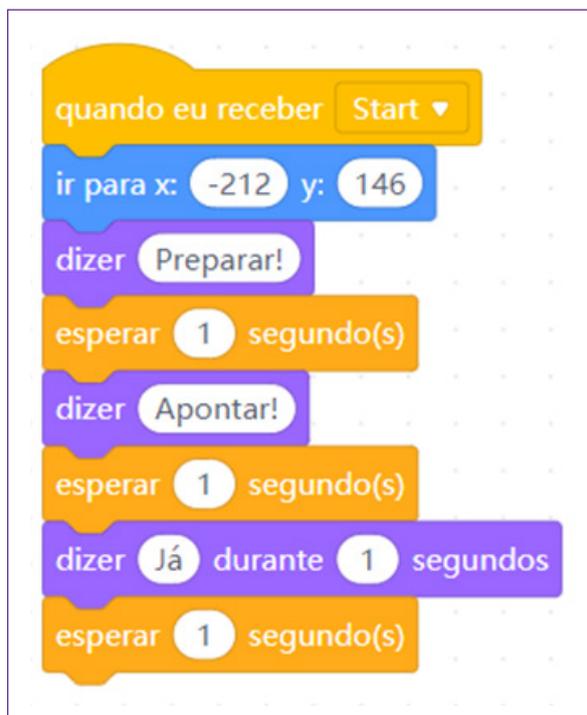
Agora, para a terceira fala entre em <APARÊNCIA>, arraste o bloco com o dizer e a marcação de tempo <dizer\_\_\_\_\_durante\_\_\_\_\_segundos>, na sequência insira o bloco normal de tempo como você fez nos anteriores.

Figura 29 - Blocos de fala e tempo



Após esse bloco, programamos a primeira parte, ficando da seguinte forma:

Figura 30 - Programação completa da primeira parte



Agora, vamos programar os movimentos, se mover o joystick para esquerda, o que acon-

# Criando jogos no mBlock [labirinto] II

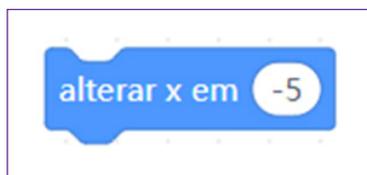
tece? Para dar esse início entra em <EVENTOS>, arraste o bloco que indica <quando eu receber esquerda>.

Figura 31 - Bloco de eventos



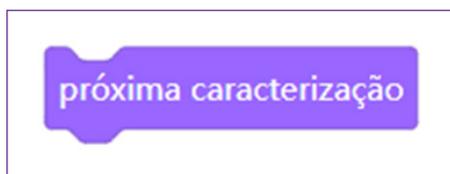
Na sequência, indique o valor para os movimentos em X - clique em <MOVIMENTO>, arraste o bloco <altera X em \_\_\_>. Esse número será a medida do movimento para a esquerda.

Figura 32 - Bloco de movimento



Como o Panda tem duas caracterizações de passos, é preciso indicar a cada movimento a troca de caracterização. Clique em <APARÊNCIA>, arraste o bloco <próxima caracterização>.

Figura 33 - Bloco de aparência - caracterização



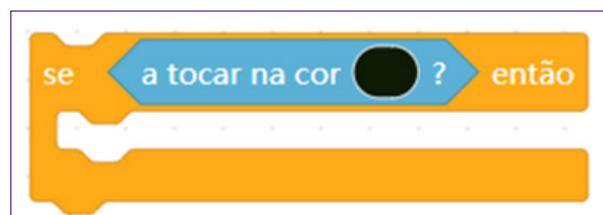
Como a personagem não pode atravessar a linha/muros do labirinto, será preciso programar isso. Assim, entre em <CONTROLO>, arraste o bloco de condicional que você já conhece <se \_\_\_\_\_ então>.

Figura 34 - Bloco de controle - condicional



Na sequência, insira qual é a condicional, agora será preciso ativar um sensor. Entre em <SENSORES>, arraste o sensor <a tocar na cor \_\_\_?>, escolha a cor de acordo com o que foi feito o labirinto, neste caso, cor preta.

Figura 35 - Blocos integrados

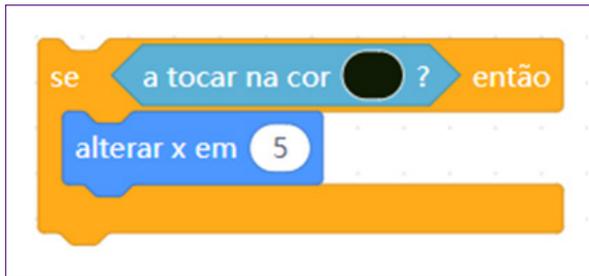


O que irá ocorrer, se a personagem tocar nesta cor? Ele volta um x número de valor, isso também deverá ser

# Criando jogos no mBlock [labirinto] II

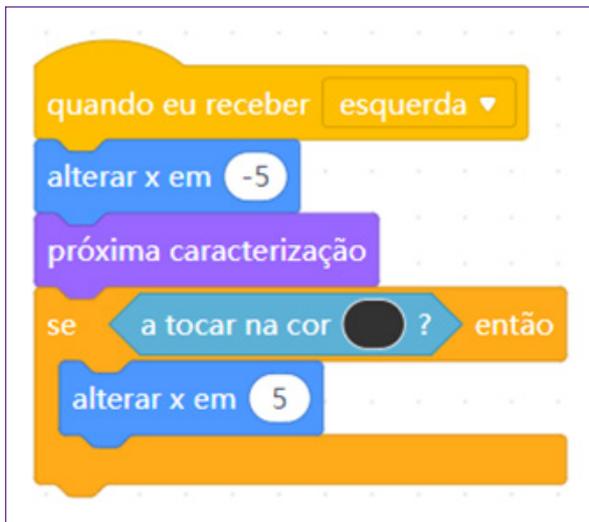
programado. Para isso, entre em MOVIMENTO, arraste o bloco “alterar x em\_\_” Indique o valor de 5, mesmo valor indicado anteriormente.

Figura 36 - Blocos integrados



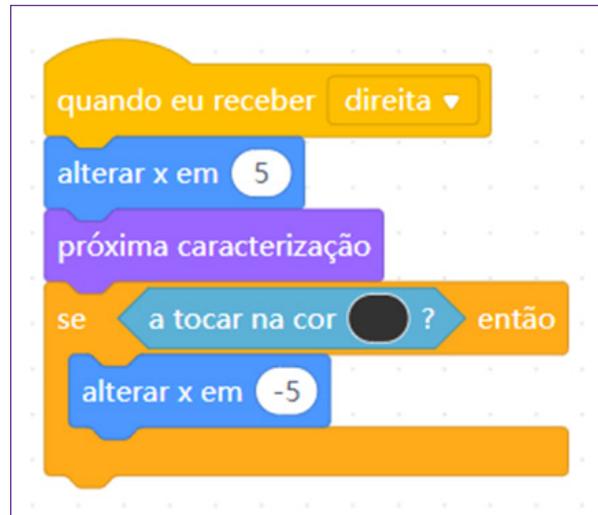
Assim, a programação completa da esquerda ficará da seguinte forma:

Figura 37 - Programação completa do movimento para esquerda



Você deverá fazer o mesmo para a programação da direita:

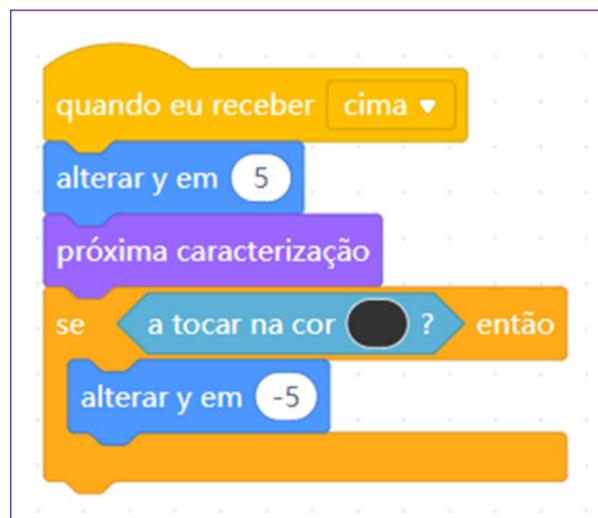
Figura 38 - Programação completa do movimento para direita



O mesmo para cima e para baixo, porém as mudanças deve ser em Y e não mais em X.

Para cima:

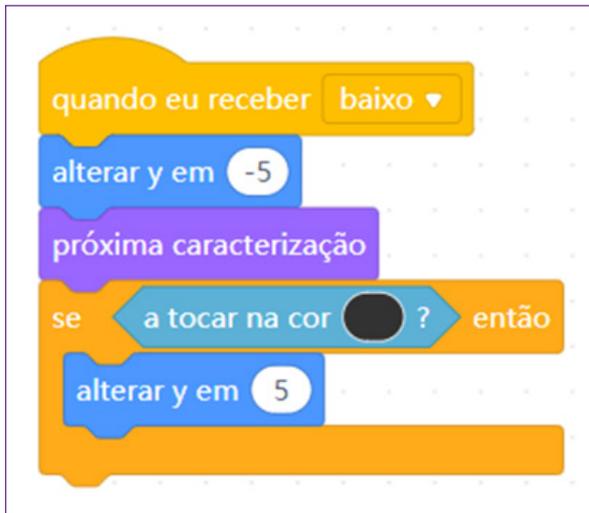
Figura 39 - Programação completa do movimento para cima



# Criando jogos no mBlock [labirinto] II

Para baixo:

Figura 40 - Programação completa do movimento para baixo



Agora, conforme roteiro, precisamos programar um cronômetro de tempo. Para isso, é preciso usar as informações disponíveis no jogo. Vamos lá!

Entre em <EVENTOS>, arraste <quando eu receber *start*>.

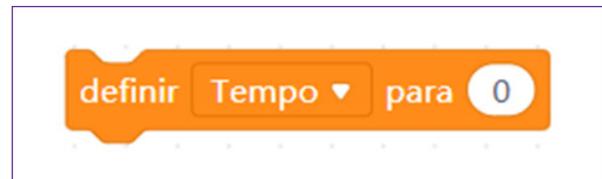
Figura 41 - Bloco de eventos



Na sequência, indique a definição de tempo. Para isso, você terá que criar uma variável chamada tempo. Entre em <VARIÁVEIS>, clique em <CRIAR NOVA

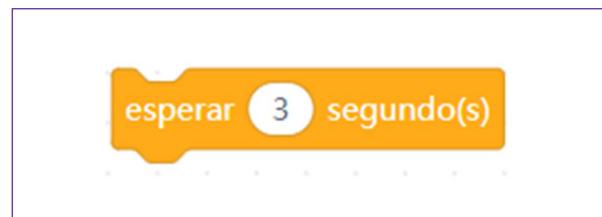
VARIÁVEL>, ao abrir a solicitação da variável insira a palavra <TEMPO>, e como você não irá usar outro ator, selecione <Apenas para este ator>, assim, sua nova variável estará pronta para uso. Arraste o bloco <definir Tempo para 0>

Figura 42 - Bloco de variável - Tempo



Após o marcador estar inserido, precisamos dar o tempo de início da fala da personagem de 3 segundos, conforme programado anteriormente. Entre em <CONTROLO>, arraste o bloco <esperar \_\_\_\_segundo(s)>, complete com o número 03.

Figura 43 - Bloco tempo

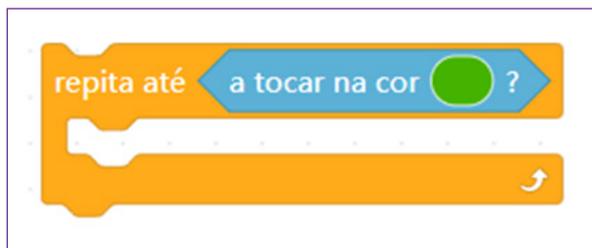


Para indicar o tempo final teremos que criar uma forma do tempo parar, assim, a cor verde será o nosso alvo, considerando que a saída está indicada com

# Criando jogos no mBlock [labirinto] II

a flecha verde, quando a personagem tocar nesta cor, o tempo irá parar. Como será uma ação constante, precisamos criar um loop - repetir para sempre. No <CONTROLO>, arraste o bloco <Repita até\_\_\_\_\_?> e entre em <SENSORES>, arraste para completar o sensor <a toca na cor\_\_\_\_?> selecione a cor verde.

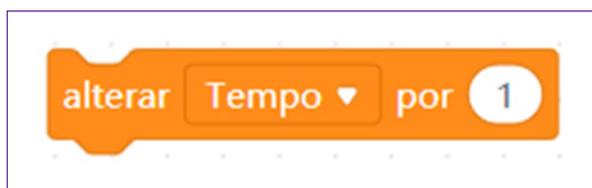
Figura 44 - Bloco de controle e sensor



Qual ação queremos que repita ao tocar na cor verde? Que o tempo seja alterado de 0 para 1 e se a personagem tocar no verde ela diga "Cheguei!".

Para isso, entre novamente em <VARIÁVEIS>, arraste o bloco de tempo <alterar o Tempo para 1>.

Figura 45 - Bloco de variável - Tempo



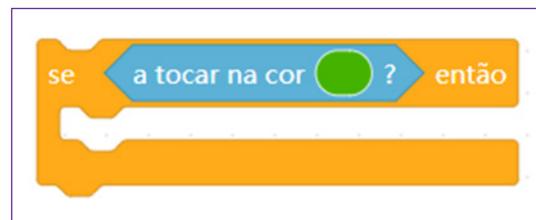
Na sequência, entre em <CONTROLO>, arraste novamente o bloco de tempo <esperar\_\_\_\_segundo(s)>, complete com o tempo de 1 segundo.

Figura 46 - Bloco de tempo



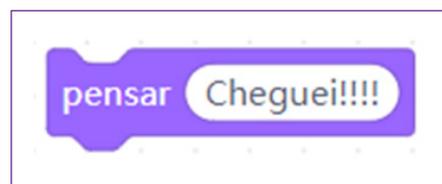
Arraste novamente a condicional <se\_\_\_\_então> e complete com o comando do sensor, clique em <SENSOR>, arraste o sensor <a tocar na cor\_\_\_\_?>, deixe novamente na cor verde.

Figura 47 - Blocos integrados



Para indicar a fala, clique em <APARÊNCIA>, arraste o bloco <pensar\_\_\_\_>, preencha com a palavra "Cheguei!".

Figura 48 - Bloco de fala



# Criando jogos no mBlock [labirinto] II

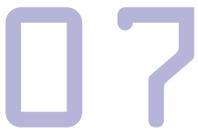
Após essa inserção deste bloco, finalizamos toda a programação de marcação de tempo, ficando da seguinte forma:

Figura 49 - Programação completa dos tempos



Fonte: mBlock, 2022

Com isso, conseguimos programar todo o jogo do labirinto. Divirta-se com essa programação. Na próxima aula iremos trabalhar com um jogo mais complexo! Antes disso, vamos para o feedback!



# Criando jogos no mBlock [Labirinto] II

## Desafio

Que tal deixar o jogo um pouco mais difícil? Insira na programação para que quando o panda tocar nas paredes do labirinto retornar ao início.

### 3. Feedback e Finalização

**a.** Confira e compartilhe com seus colegas o jogo programado, deixe eles jogar para ver se a programação está funcionando adequadamente;

**b.** Analise e troque informações com os colegas sobre como você criou sua programação;

**c.** Reveja se você aprendeu como trabalhar com o mBlock para fazer um jogo simples como esse do labirinto;

**d.** Reflita se as seguintes situações ocorreram:

**i.** Você e os seus colegas trocaram ideias no momento da criação e programação do jogo?

**ii.** Você teve problemas ao fazer a programação do jogo? Qual? Como você resolveu?



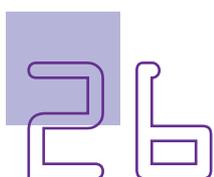
## Referências:

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](http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf). Acesso em: 08 mar. 2022.

MAKEBLOCK. mBlock. **Download mBlock**. Disponível em: <https://mblock.makeblock.com/en-us/download/>. Acesso em: 15 mar. 2022.

MAKEBLOCK. MBlock. **Make with Code**. Disponível em: <https://mblock.makeblock.com/en-us/>. Acesso em: 15 mar. 2022.

MAKEBLOCK. mBlock. **Programação em blocos**. Disponível em: <https://ide.mblock.cc/>. Acesso em: 15 mar. 2022.



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

**EQUIPE ROBÓTICA PARANÁ**

Adilson Carlos Batista  
Andrea da Silva Castagini Padilha  
Cleiton Rosa  
Darice Alessandra Deckmann Zanardini  
Edgar Cavalli Junior  
Edna do Rocio Becker  
José Feuser Meurer  
Marcelo Gasparin  
Michele Serpe Fernandes  
Michelle dos Santos  
Orlando de Macedo Junior  
Roberto Carlos Rodrigues

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 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/)



DTI - DIRETORIA DE TECNOLOGIA E INOVAÇÃO  
DETED - DEPARTAMENTO DE TECNOLOGIAS EDUCACIONAIS