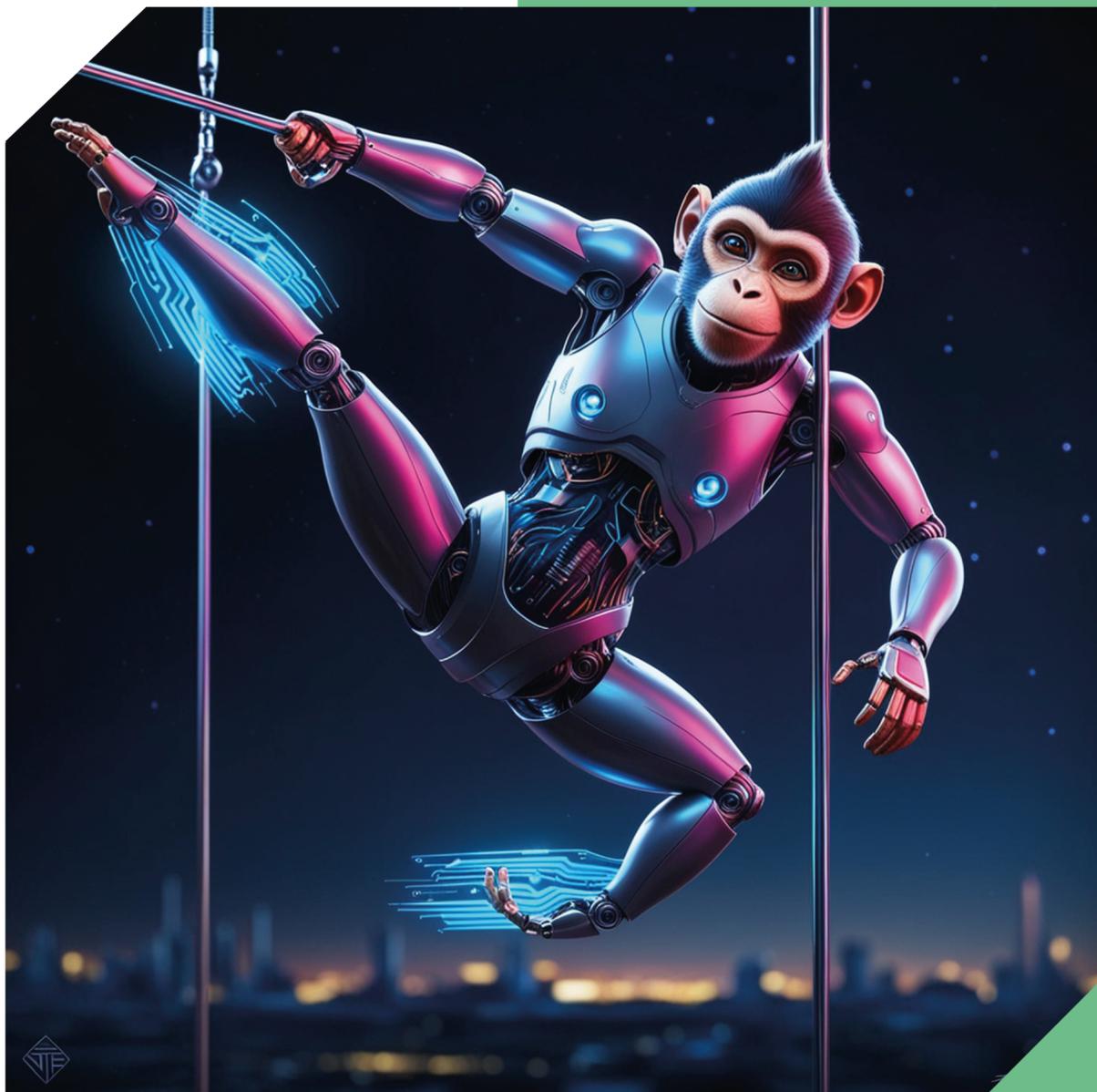


ROBÓTICA

AULA 03

Primeiros Passos Módulo 4



Robô macaco
trapezista - II

Diretoria de Tecnologia e Inovação

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

Andrea da Silva Castagini Padilha

Viviane Dziubate Pittner

Validação de Conteúdo

Darice Alessandra Deckmann Zanardini

Viviane Dziubate Pittner

Revisão Textual

Kellen Pricila dos Santos Cochinski

Projeto Gráfico e Diagramação

Edna do Rocio Becker

Apoio Técnico

Equipe UFMS

2025

SUMÁRIO

Introdução	2
Objetivos	3
Roteiro da aula	4
1. Contextualização	4
2. Programação	5
3. Feedback e finalização	15
Referências bibliográficas	15

Introdução

Na aula 2, demos os primeiros passos para criar nosso próprio macaco robô trapezista! Lembrem-se de como a Física entrou em ação para entendermos os movimentos do macaco? Agora chegou o momento da programação, ela nos permitirá dar vida ao nosso protótipo. Com o Arduino como nosso principal aliado, conseguimos construir um modelo básico e já observamos como os princípios da força, movimento e energia se aplicam a um sistema robótico. A partir desse ponto, podemos explorar ainda mais as possibilidades e aprimorar nosso macaco robô, adicionando novas funcionalidades e complexidade. Nesse primeiro momento, trabalharemos somente com a programação.

Figura 1 - Robô macaco trapezista



Fonte: Imagem produzida por IA, 2024.



Objetivos desta aula

- Programar e testar o protótipo do macaco trapezista montado na aula anterior.

Lista de materiais

- Protótipo de macaco trapezista montado;
- Mblock;
- Notebook.

Roteiro da aula

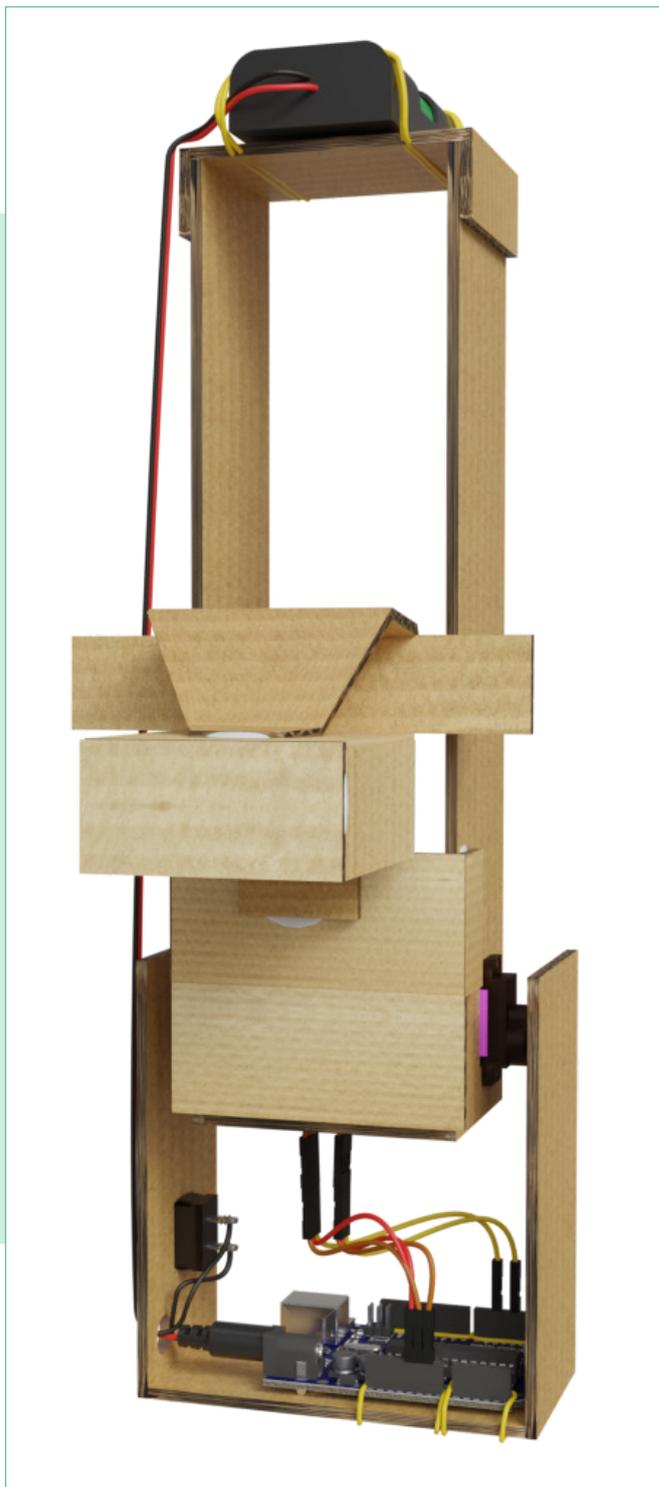
1. Contextualização

A aula anterior nos proporcionou uma imersão no fascinante mundo da robótica e suas aplicações no entretenimento e como ela está transformando o modo como interagimos com o mundo, especialmente em áreas como o circo e os parques temáticos. Exploramos exemplos como o RoboPole, que demonstra a interação entre robôs e acrobatas em performances circenses, e os animatrônicos hiperrealistas da Disney, que substituem animais em suas atrações. Essas inovações mostram como a robótica está expandindo os limites do que é possível no entretenimento, combinando tecnologia e criatividade.

Nosso projeto, o macaco robô trapezista, permitirá aplicar na prática os conceitos aprendidos, vamos para a programação do protótipo. Pegue o protótipo finalizado na aula anterior e mãos à obra!

Não se esqueça, como sinalizado anteriormente, este é o projeto inicial e você poderá ampliá-lo e melhorar após os testes.

Figura 2 - Protótipo de macaco robô trapezista



2. Programação

Nossa programação será a seguinte: o robô inicia com os braços na posição central e então realiza um movimento oscilatório contínuo, onde um braço se move para 0 graus enquanto o outro se move para 180 graus, e vice-versa. Esse movimento se repete 4 vezes e então há uma pausa de 3 segundos antes de começar novamente. Todos esses movimentos farão dele o robô trapezista.

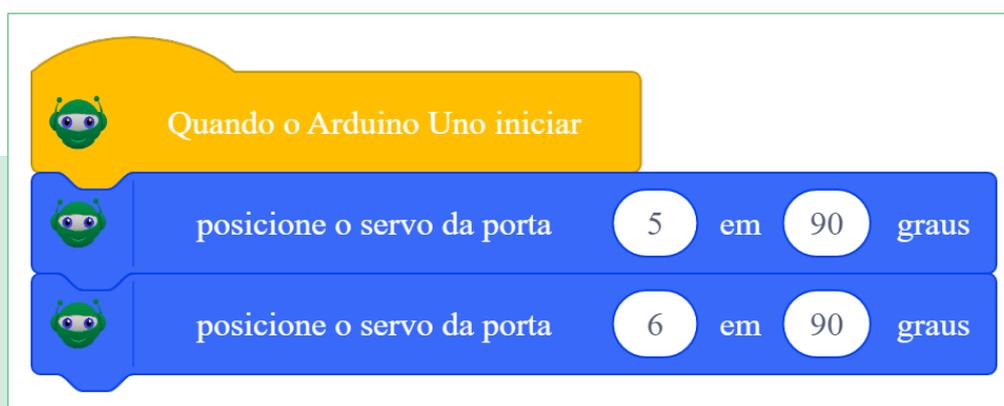
Para isso, comece sua programação pelo início, ou ponto de partida na programação por blocos - arraste o <**Quando o Arduino Uno iniciar**>.

Figura 3 – Bloco inicial



Na sequência, entre em portas e arraste o bloco <**posicione o servo da porta__em__graus**>, duplique, preencha os parâmetros com portas **5** e **6** e **90** graus. Com isso, ao iniciar o programa, os servos conectados às portas **5** e **6** do Arduino são posicionados em **90** graus. Isso significa que ambos os braços do trapezista estão em uma posição central.

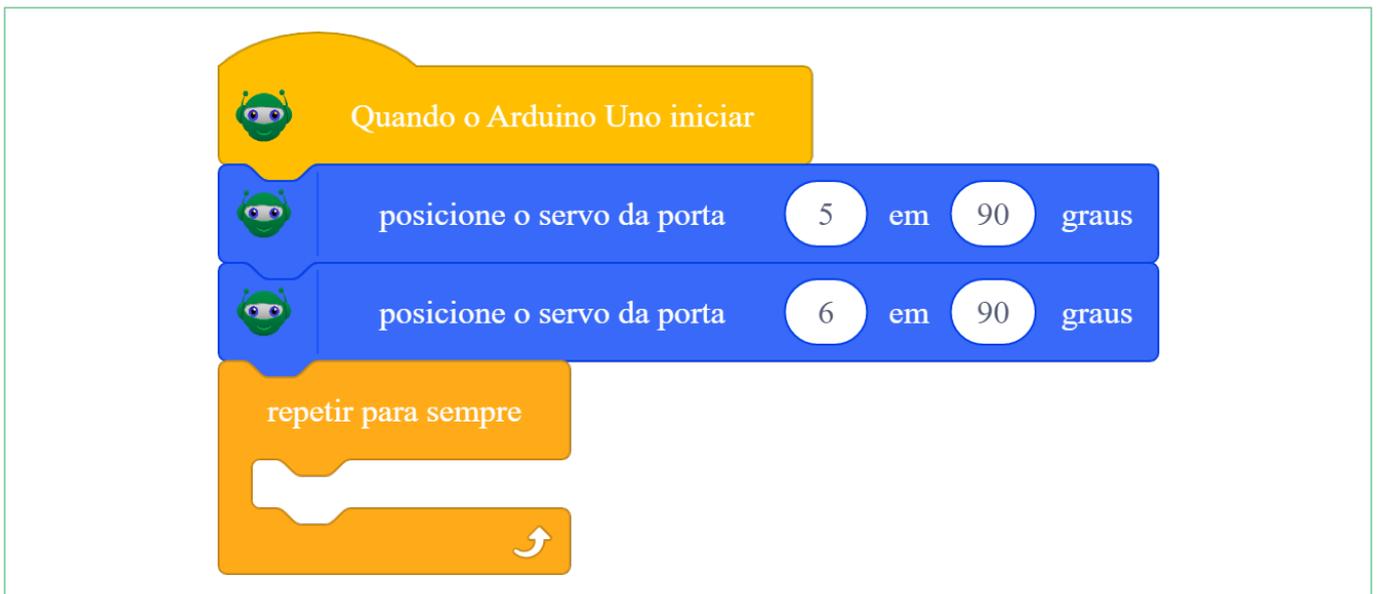
Figura 4 – Bloco posição dos servos



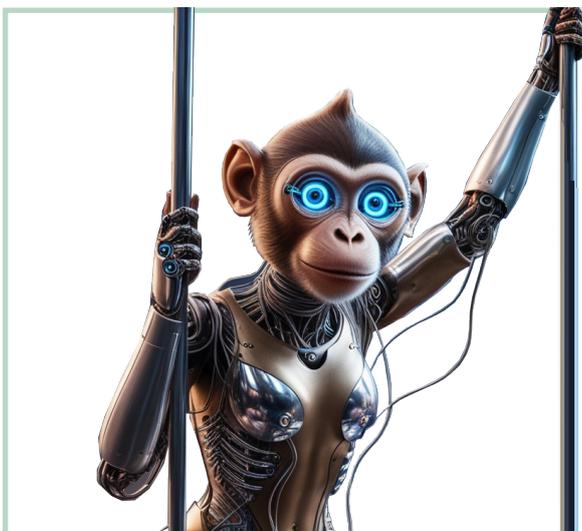
Robô macaco trapezista - II

Como queremos que os movimentos do robô macaco trapezista sejam realizados de forma constante, arraste o bloco <repetir para sempre> e tudo que será inserido nesse bloco ficará rodando em um loop infinito.

Figura 5 – Bloco posição dos servos e loop



Nesse caso, o loop será indicar o tempo e as variações de graus nas portas, assim, insira o bloco <esperar ___ segundos>, complete com **3** segundos, loop novamente <repetir> e nesse loop indique o valor **4** e as posições das portas **5** e **6** nos ângulos em graus – **0** e **180** respectivamente.



Lembre-se: a programação é uma forma de arte. Não tenha medo de experimentar diferentes combinações de blocos e ver o que acontece. Divirta-se criando movimentos incríveis para o seu robô trapezista!

Figura 6 – Sequência de ângulos e tempo dentro do loop



Robô macaco trapezista - II

Por último, insira os mesmos blocos, entretanto, mudamos a variação dos ângulos - porta 5 em 180 graus e porta 6 em 0 graus, depois novamente o tempo de espera em 0,45 segundos.

Figura 7 – Programação completa



Robô macaco trapezista - II

Assim, fechamos a programação do robô macaco trapezista. Faça o teste e divirta-se! Lembre-se de que você poderá alterar os ângulos e tempo de espera. Você pode ajustar o tempo de espera (0,45 segundos) para controlar a velocidade do movimento. Um tempo menor resultará em um movimento mais rápido e vice-versa. Para observar o funcionamento, coloque a estrutura de um lado do palito sobre uma mesa e debaixo de algo pesado que possa suportar o peso e os movimentos do protótipo - Figura 8.

Figura 8 – Robô em movimento com a estrutura assegurada por um peso



Antes de iniciarmos o desafio, outra informação interessante: o MBlock oferece a possibilidade de visualizar o código em C++ na lateral dos blocos. Ao clicar no ícone de setas laterais, ao lado do bloco laranja, você poderá alternar entre a programação por blocos e a programação em C++. Essa funcionalidade nos permitirá explorar uma nova linguagem de programação e aprofundar nossos conhecimentos em programação. Essa prática logo se tornará constante e aos poucos iremos programar também usando essa forma de linguagem, principalmente quando você estiver no Ensino Médio.

Figura 9 – Setas laterais

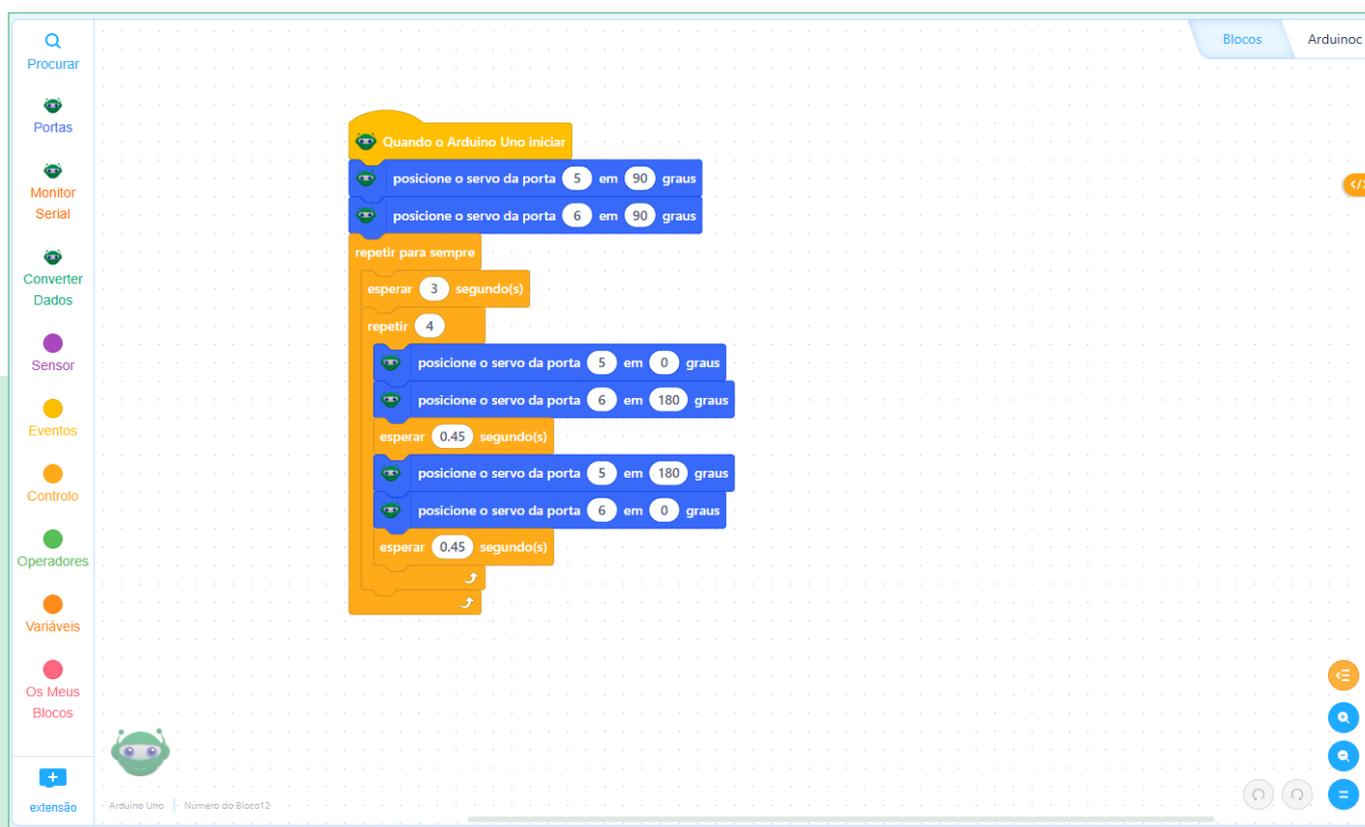
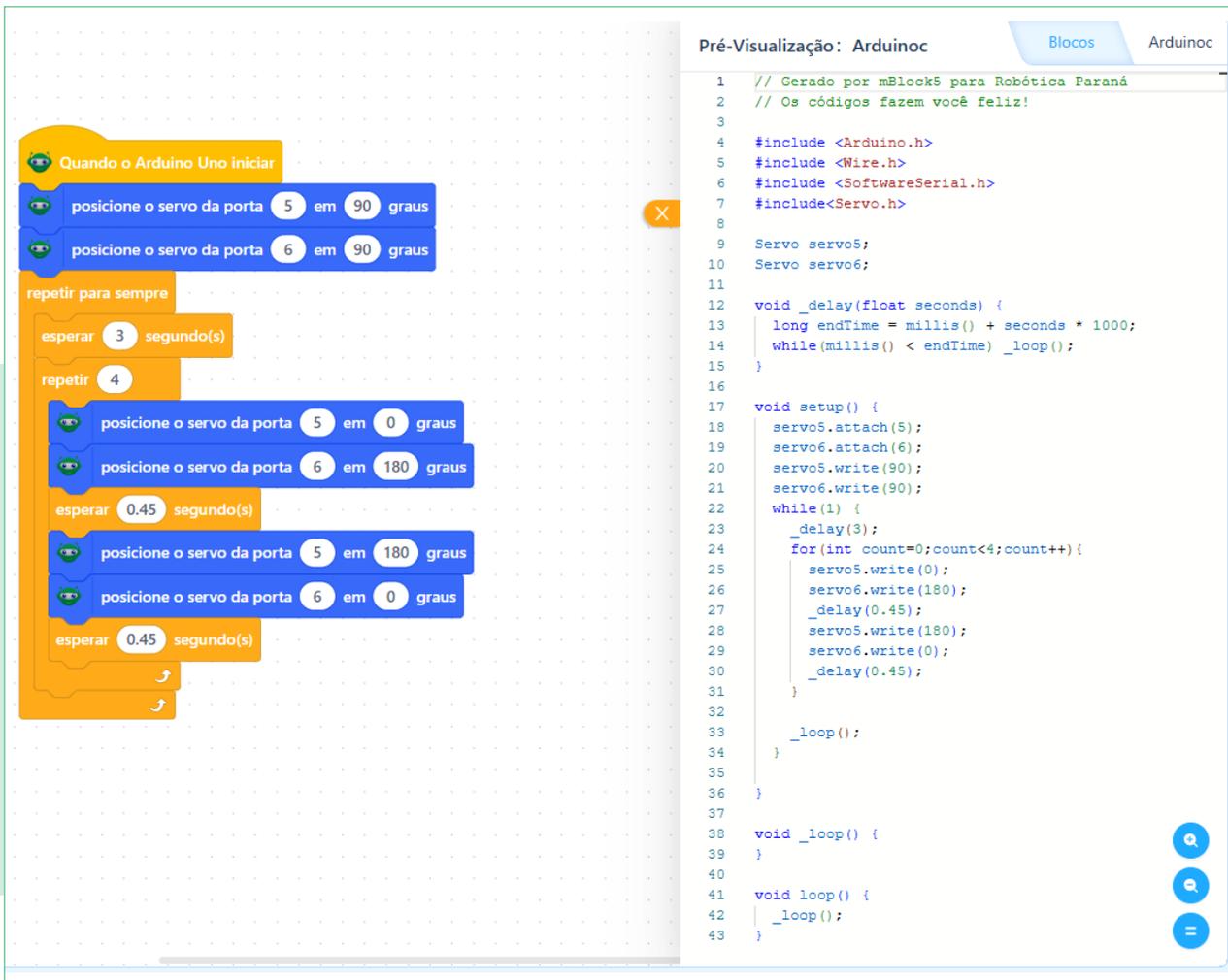


Figura 10 – Linguagem C++

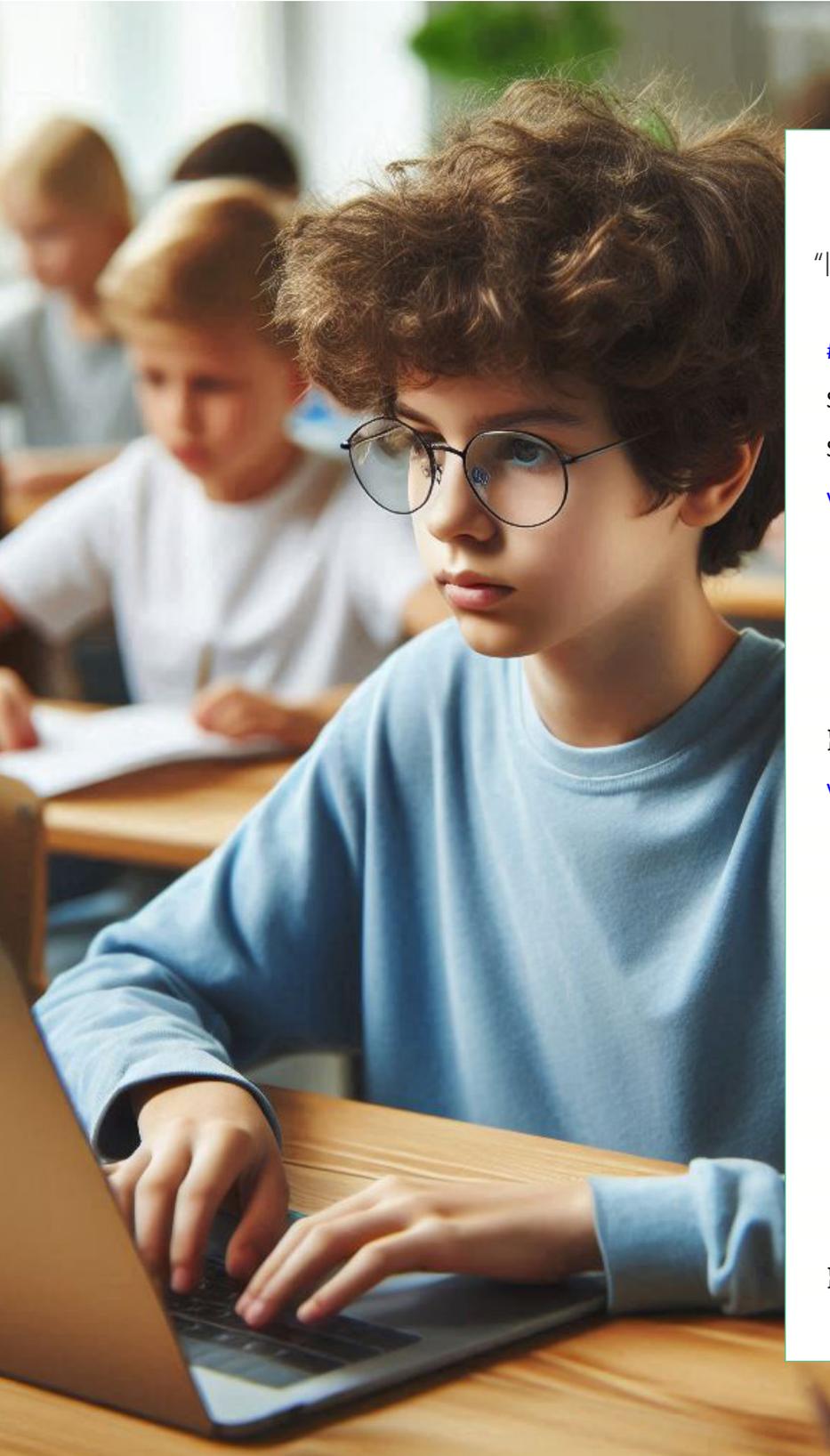


The image shows the Arduino IDE interface. On the left, a block-based program is visible, starting with 'Quando o Arduino Uno iniciar', followed by two 'posicione o servo da porta' blocks (5 em 90 graus and 6 em 90 graus). A 'repetir para sempre' loop contains an 'esperar 3 segundo(s)' block, a 'repetir 4' block, and a sequence of servo positioning and delay blocks: 'posicione o servo da porta 5 em 0 graus', 'posicione o servo da porta 6 em 180 graus', 'esperar 0.45 segundo(s)', 'posicione o servo da porta 5 em 180 graus', 'posicione o servo da porta 6 em 0 graus', and 'esperar 0.45 segundo(s)'. On the right, the C++ code is displayed, starting with comments and including headers for Arduino, Wire, SoftwareSerial, and Servo. It defines servo5 and servo6, a custom delay function, a setup function that attaches servos and writes initial positions, and a loop function that repeats the servo sequence four times.

```
1 // Gerado por mBlock5 para Robótica Paraná
2 // Os códigos fazem você feliz!
3
4 #include <Arduino.h>
5 #include <Wire.h>
6 #include <SoftwareSerial.h>
7 #include <Servo.h>
8
9 Servo servo5;
10 Servo servo6;
11
12 void _delay(float seconds) {
13     long endTime = millis() + seconds * 1000;
14     while(millis() < endTime) _loop();
15 }
16
17 void setup() {
18     servo5.attach(5);
19     servo6.attach(6);
20     servo5.write(90);
21     servo6.write(90);
22     while(1) {
23         _delay(3);
24         for(int count=0;count<4;count++){
25             servo5.write(0);
26             servo6.write(180);
27             _delay(0.45);
28             servo5.write(180);
29             servo6.write(0);
30             _delay(0.45);
31         }
32     }
33     _loop();
34 }
35
36 }
37
38 void _loop() {
39 }
40
41 void loop() {
42     _loop();
43 }
```

No primeiro módulo do nosso curso, quando introduzimos a programação por blocos, fizemos a seguinte analogia! Você se lembra? Imagine que você está construindo um Lego. Você pode usar os bloquinhos coloridos para montar o que quiser, sem precisar saber exatamente como cada peça funciona por dentro. Essa é a ideia da **programação por blocos**. Cada bloco representa uma ação ou um comando, e você os encaixa para criar um programa.

Com a linguagem C++, é como se você pegasse todas as peças do Lego e tivesse que escrever um manual de instruções detalhado para montá-las. Em vez de usar bloquinhos coloridos, você usa palavras e símbolos para dar comandos ao computador. Essa forma de programar é mais parecida com a linguagem que os programadores profissionais usam.



Essa programação, de forma “limpa”, ficaria da seguinte forma:

```
#include <Servo.h>
Servo servo5;
Servo servo6;
void setup() {
  servo5.attach(5);
  servo6.attach(6);
  servo5.write(90);
  servo6.write(90);
}
void loop() {
  delay(3000);
  for (int count = 0; count < 4; count++) {
    servo5.write(0);
    servo6.write(180);
    delay(450);
    servo5.write(180);
    servo6.write(0);
    delay(450);
  }
}
```

Qual a diferença entre as duas?

Característica	Programação por blocos (MBlock)	Linguagem C++
Visualização	Utiliza blocos coloridos que representam comandos.	Utiliza texto e símbolos para escrever o código.
Facilidade de uso	Mais fácil de aprender e usar, ideal para iniciantes.	Requer mais conhecimento de sintaxe e lógica de programação.
Flexibilidade	Menos flexível para criar programas complexos.	Mais flexível, permite criar programas mais complexos.
Similaridade com linguagens de programação	Se assemelha a linguagens de programação textuais, preparando para linguagens como C++.	É uma linguagem de programação textual, similar ao C++.

Por que aprender C++?

- Aprofundar conhecimentos: ao entender como os blocos se transformam em código, você compreende melhor como os programas funcionam.
- Preparar para o futuro: aprender C++ é um passo importante para quem deseja se tornar um programador.
- Criar projetos mais complexos: a linguagem C++ permite criar programas mais sofisticados e personalizados.

E aí, gostou de conhecer as duas formas de programar? Qual você acha mais interessante?

Desafios:

Que tal mudar os ângulos e intervalos para criar novos movimentos no robô macaco trapezista?

Que tal inserir um buzzer ou um alto-falante para tocar uma música e sincronizar os movimentos do robô com o ritmo?

E se...

O projeto robô macaco trapezista não funcionar?

Verifique todas as conexões: certifique-se de que os servos estão conectados corretamente às portas corretas do Arduino. Verifique se os jumpers estão bem encaixados e se não há fios soltos.

Certifique-se de que os servos não estejam danificados ou se estão com algum obstáculo que impeça o movimento.



3. Feedback e finalização

Parabéns por chegar até aqui! Construir um robô trapezista é um projeto desafiador e gratificante. Para finalizar seu projeto e obter o máximo de aprendizado, vamos explorar algumas possibilidades de feedback e aprimoramento:

a. **Funcionalidade:** o robô realiza os movimentos desejados de forma consistente? Os servos respondem aos comandos do programa?

b. **Estabilidade:** o robô mantém o equilíbrio durante os movimentos? A estrutura é resistente o suficiente?

c. **Precisão:** os movimentos são precisos e coordenados?

d. **Análise de dados:** analise os dados coletados para identificar pontos fortes e fracos do projeto.

e. **Ajustes no código:** modifique o código para melhorar o desempenho do robô, como aumentar a precisão dos movimentos ou reduzir o consumo de energia.

f. **Otimização da estrutura:** faça ajustes na estrutura física do robô para melhorar a estabilidade e a eficiência, mude os materiais.

Lembre-se: aprender e construir um robô é um processo contínuo. Não tenha medo de experimentar e fazer ajustes. O mais importante é se divertir e aprender com cada desafio!

g. Para finalizar, reflita se as seguintes situações ocorreram:

i. Colaboração e cooperação: você e os membros de sua equipe interagiram entre si, compartilhando ideias que promoveram a aprendizagem e o desenvolvimento deste projeto?

ii. Pensamento crítico e resolução de problemas: você conseguiu identificar os problemas, analisar informações e tomar decisões de modo a contribuir para o projeto desenvolvido?

h. Reúna todos os componentes utilizados nesta aula e os organize novamente, junto aos demais, no kit de Robótica.

REFERÊNCIAS

ARDUINO. **Documentação de Referência da Linguagem Arduino**. Disponível em:

<https://www.arduino.cc/reference/pt/>.

Acesso em: 26, nov. 2024.

MBlock. Disponível em: [mBlock Block-Based IDE- Coding for Beginners](#) . Acesso em

26, nov. 2024.

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL (UFMS)
FACULDADE DE COMPUTAÇÃO (FACOM)

PROFESSORES

- Amaury Antônio de Castro Junior
- Anderson Corrêa de Lima
- Glauder Guimarães Ghinozzi
- Graziela Santos de Araújo
- Said Sadique Adi

ESTUDANTES

- Arthur Henrique Andrade Farias - Ciência da Computação
- Anny Beatriz Silva Corrêa Aranda - Ciência da Computação
- Bruno Pereira Wesner da Silva - Engenharia de Computação
- Fernanda das Neves Merqueades Santos - Ciência da Computação
- Gabriel Pereira Falcão - Ciência da Computação
- Jenniffer Oliveira Checchia - Ciência da Computação
- Leonardo Vargas de Paula - Sistemas de Informação
- Marcos Gabriel da Silva Rocha - Engenharia de Computação
- Maria Paula do Nascimento Santos - Engenharia de Computação
- Nathanael Martins Wink - Ciência da Computação
- Victor Luiz Marques Saldanha Rodrigues - Ciência da Computação

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

EQUIPE ROBÓTICA PARANÁ

- Adilson Carlos Batista
- Ailton Lopes
- Andrea da Silva Castagini Padilha
- Cleiton Rosa
- Darice Alessandra Deckmann Zanardini
- Edna do Rocio Becker
- Kellen Pricila dos Santos Cochinski
- Marcelo Gasparin
- Michele Serpe Fernandes
- Michelle dos Santos
- Roberto Carlos Rodrigues
- Sandra Aguera Alcova Silva
- Viviane Dziubate Pittner

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/)

