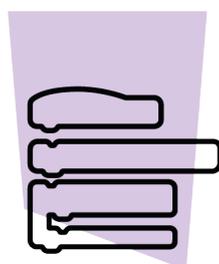
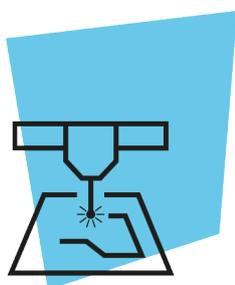
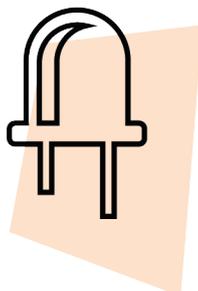
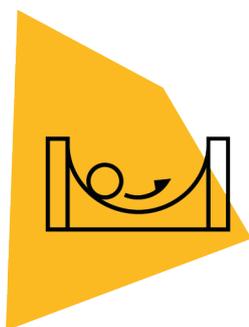
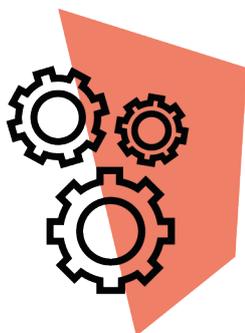


ROBÓTICA

Módulo 1



Robô Sumô [Estrutura]

AULA 38

GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Renato Feder

DIRETOR DE TECNOLOGIA E INOVAÇÃO

Andre Gustavo Souza Garbosa

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Cleiton Rosa

Michelle Santos

Simone Sinara de Souza

Revisão Textual

Adilson Carlos Batista

Projeto Gráfico e Diagramação

Edna do Rocio Becker

2021



Este trabalho está licenciado com uma Licença Creative Commons
Atribuição NãoComercial - CompartilhaIgual 4.0 Internacional

Aula 01	Por Que Robótica?
Aula 02	Tensão, Corrente e Resistência
Aula 03	Kit de Robótica
Aula 04	Arduino Uno R3
Aula 05	Softwares Arduino IDE e mBlock
Aula 06	Portas Digitais
Aula 07	Circuito Elétrico
Aula 08	LED e Resistor
Aula 09	Semáforo [Carros]
Aula 10	Semáforo [Cruzamento Carros]
Aula 11	Semáforo [Pedestres]
Aula 12	Semáforo [Cruzamento Carros + Pedestres]
Aula 13	Push Button
Aula 14	Feedbacks + Inventário I
Aula 15	Semáforo [Carros + Pedestres com Botão]
Aula 16	Display 7 Segmentos
Aula 17	Fonte DC + Plug P4
Aula 18	Portas PWM
Aula 19	LED Fade-In
Aula 20	LED Fade-Out
Aula 21	Super Máquina 80's
Aula 22	Super Máquina 2008
Aula 23	Potenciômetro
Aula 24	Buzzer Passivo
Aula 25	LED RGB
Aula 26	Arco-Iris
Aula 27	Sensor LDR
Aula 28	Feedbacks + Inventário II
Aula 29	Sensor de Temperatura
Aula 30	Sensor de Obstáculo IR
Aula 31	Controle Motor DC
Aula 32	Kit Chassi 2WD Robô
Aula 33	Seguidor de Linha
Aula 34	Sensor de Distância
Aula 35	Sensor de Estacionamento
Aula 36	Display LCD 16x2
Aula 37	Trena Digital
Aula 38	Robô Sumô [Estrutura]
Aula 39	Robô Sumô [Programação + Treinamento I]
Aula 40	Robô Sumô [Programação + Treinamento II]
Aula 41	Disputa de Sumôs
Aula 42	Feedbacks + Inventário III

Aula 37
Trena Digital

Aula 38
Robô Sumô
[Estrutura]

Aula 39
Robô Sumô
[Programação + Treinamento I]

Sumário

Introdução	2
Objetivos desta aula	2
Competências gerais previstas na BNCC	3
Habilidades do século XXI a serem desenvolvidas	4
Lista de materiais	4
Roteiro da aula	5
1. Contextualização	5
2. Montagem e programação	5
3. Feedback e finalização	21
Videotutorial	22



Introdução

A palavra robô pode ser definida como um agente composto por dispositivos eletromecânicos, capazes de desempenhar tarefas de forma autônoma ou pré-programada, para mover, manipular objetos e realizar diversos trabalhos considerados perigosos à espécie humana.

Nesta aula, realizaremos a montagem e a programação de um robô com finalidade de atuar numa competição denominada “Sumô de Robôs”, onde o objetivo é empurrar o adversário para fora da arena plana.



Objetivos desta aula

- Compreender o que é um robô, sua construção e programação com um Arduino;
- Apresentar uma aplicação prática da combinação de sensores e motores shield e DC;
- Aprender a programar sensores de obstáculo e distância para delimitar a movimentação do robô sumô.



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.



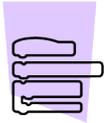
Habilidades do século XXI a serem desenvolvidas

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



Lista de materiais

- 1 placa Arduino Uno R3;
- 1 cabo USB;
- 1 sensor de distância ultrassônico HC-SR04;
- 2 módulos sensor de obstáculo infravermelho IR;
- 1 motor shield L293D driver ponte H para Arduino;
- 1 kit chassi 2WD robô para Arduino;
- 10 jumpers fêmea-fêmea;
- 1 fonte 9V ou baterias adequadas (6 a 12V);
- 1 notebook;
- Software mBlock ou Arduino IDE.



Roteiro da aula

1. Contextualização (15min)

Você já teve a oportunidade de assistir, em jogos olímpicos ou campeonatos esportivos, uma luta de sumô? Que detalhes nesta luta te chamaram a atenção?

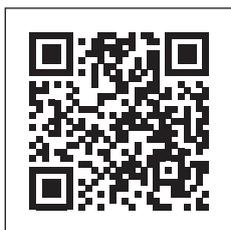
O sumô é uma modalidade de luta de origem japonesa, na qual dois lutadores, vestidos com apenas uma faixa de tecido grosso (*mawashi*), enrolada em volta da cintura, se enfrentam em um ringue sem cordas (*dohyo*), com o objetivo de forçar o oponente a sair do círculo ou tocar o chão com qualquer parte do corpo que não seja a sola do pé.

Os robôs sumô, geralmente, são dotados de rodas ou esteiras e sensores (ultrassônicos) que localizam o adversário e detectam a borda da arena (infravermelhos), evitando sair dessa; possuem também uma lâmina angulada posicionada à frente do robô, usada para levantar o adversário e empurrá-lo com mais facilidade para fora da arena.

No combate robô sumô existem duas categorias: robôs operados por controle remoto e robôs autônomos. Nesta aula, apresentaremos um protótipo de robô autônomo, constituído por componentes eletrônicos que permitem a simulação dos movimentos do robô sumô, com a programação na linguagem escrita e em blocos.

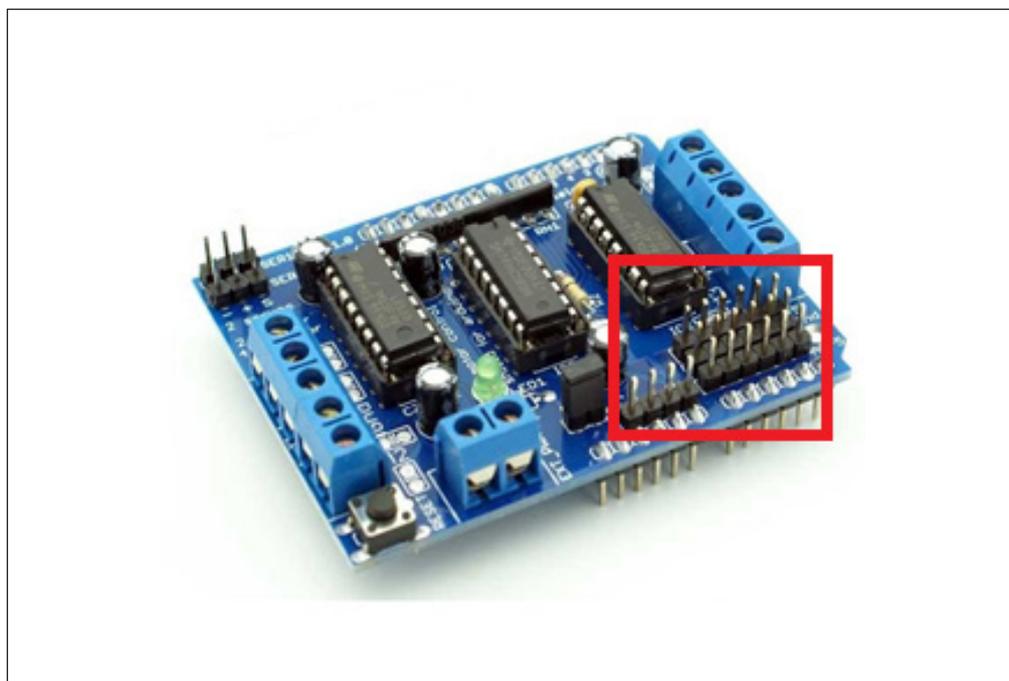
2. Montagem e Programação (60min)

Para esta aula, você utilizará o kit chassi montado na **Aula 32 - Kit Chassi 2WD Robô**. Antes da montagem, será necessário soldar uma barra de pinos para que seja possível conectarmos, com o uso de jumpers, o sensor ultrassônico ao robô. Para realizar este procedimento, aconselhamos acompanhar o videotutorial: “[Como soldar a barra de pinos na Ponte H L293D Driver para Arduino](#)”, disponível em: <https://youtu.be/OAEO5c8RANA>. A figura 1 apresenta o local da soldagem de pinos na placa motor shield.



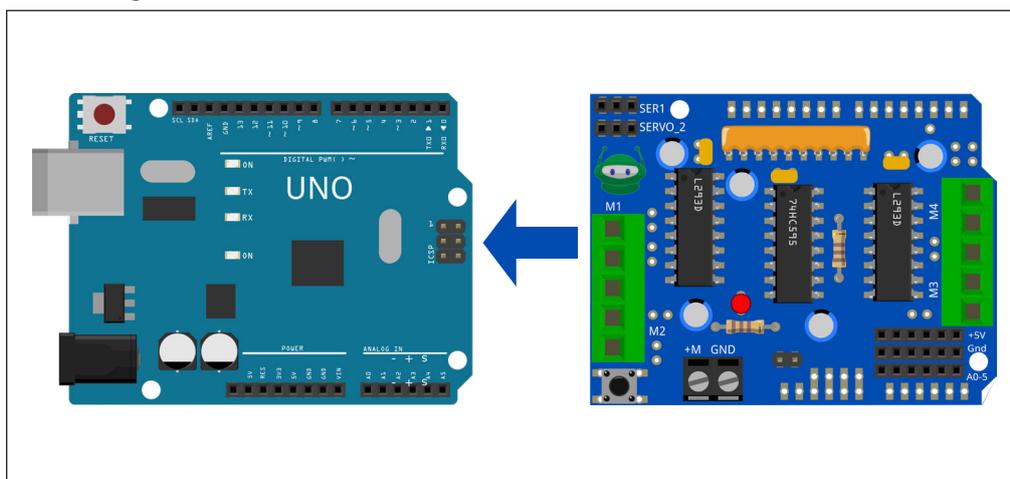
[Como soldar a barra de pinos na Ponte H L293D Driver para Arduino](#)

Figura 1 - Pinos soldados na motor shield



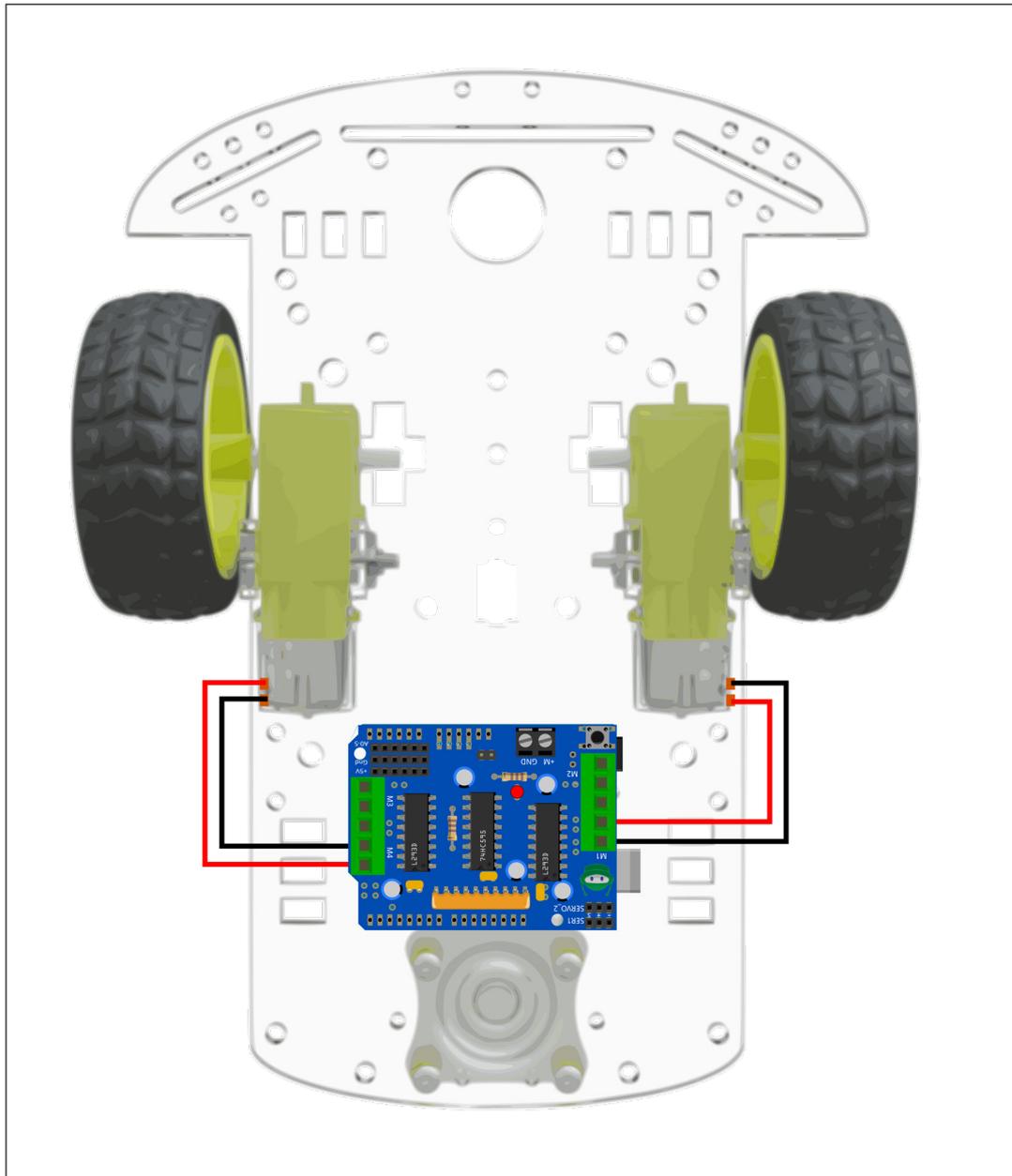
Após realizar a soldagem, vamos iniciar a montagem do robô, encaixe a placa motor shield L293D driver sobre o Arduino Uno (figura 2).

Figura 2 - Conexões entre motor shield, Arduino e motores DC



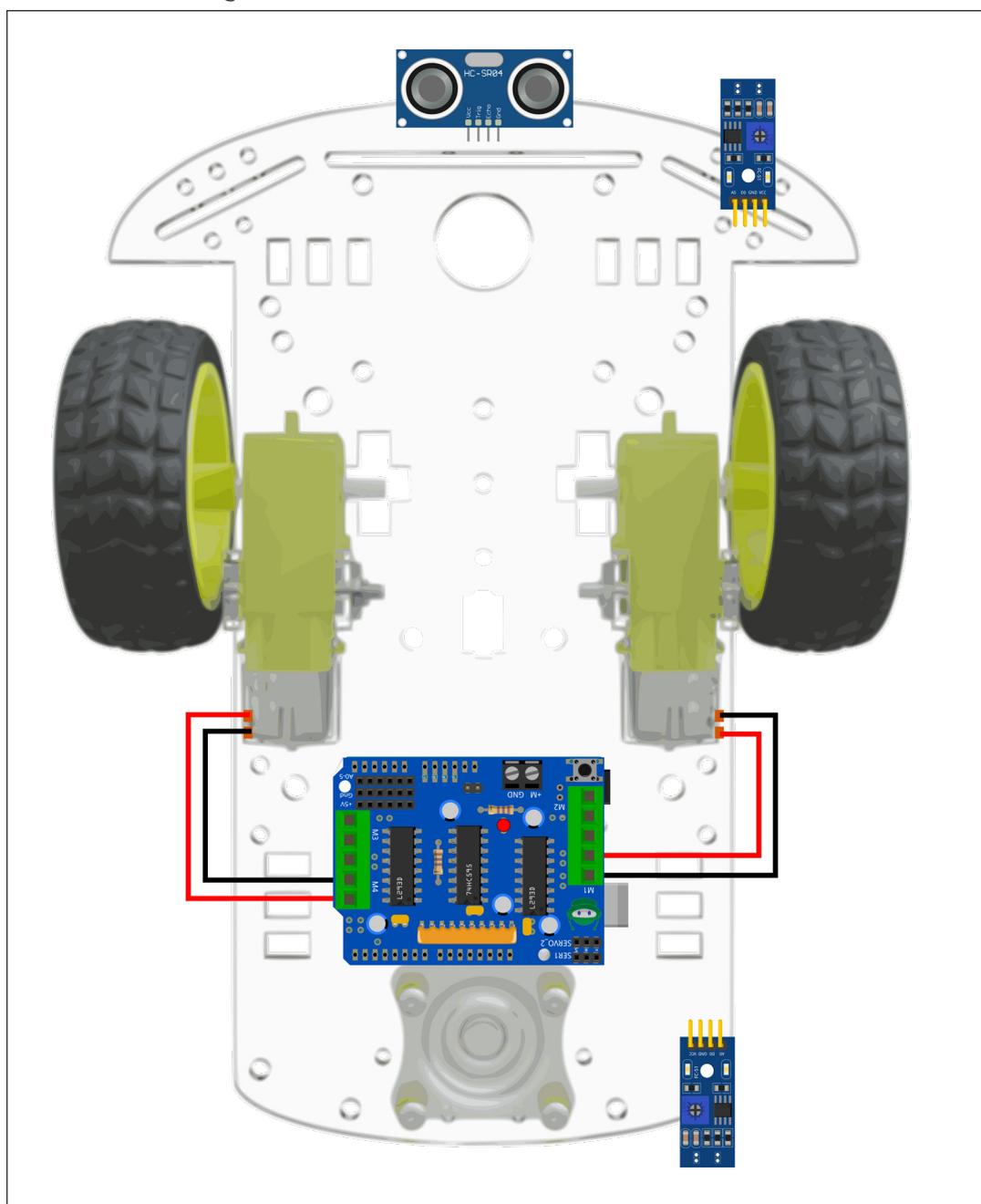
Em seguida, posicione o conjunto (Arduino + motor shield) sobre o chassi e fixe os terminais dos motores direito e esquerdo nos bornes M1 e M4 da placa motor shield, respectivamente. Não se preocupe com a ordem dos fios, pois durante os testes, podemos alterá-los caso o(s) motor(es) não esteja(m) girando no sentido correto.

Figura 3 - Fixando os motores na motor shield



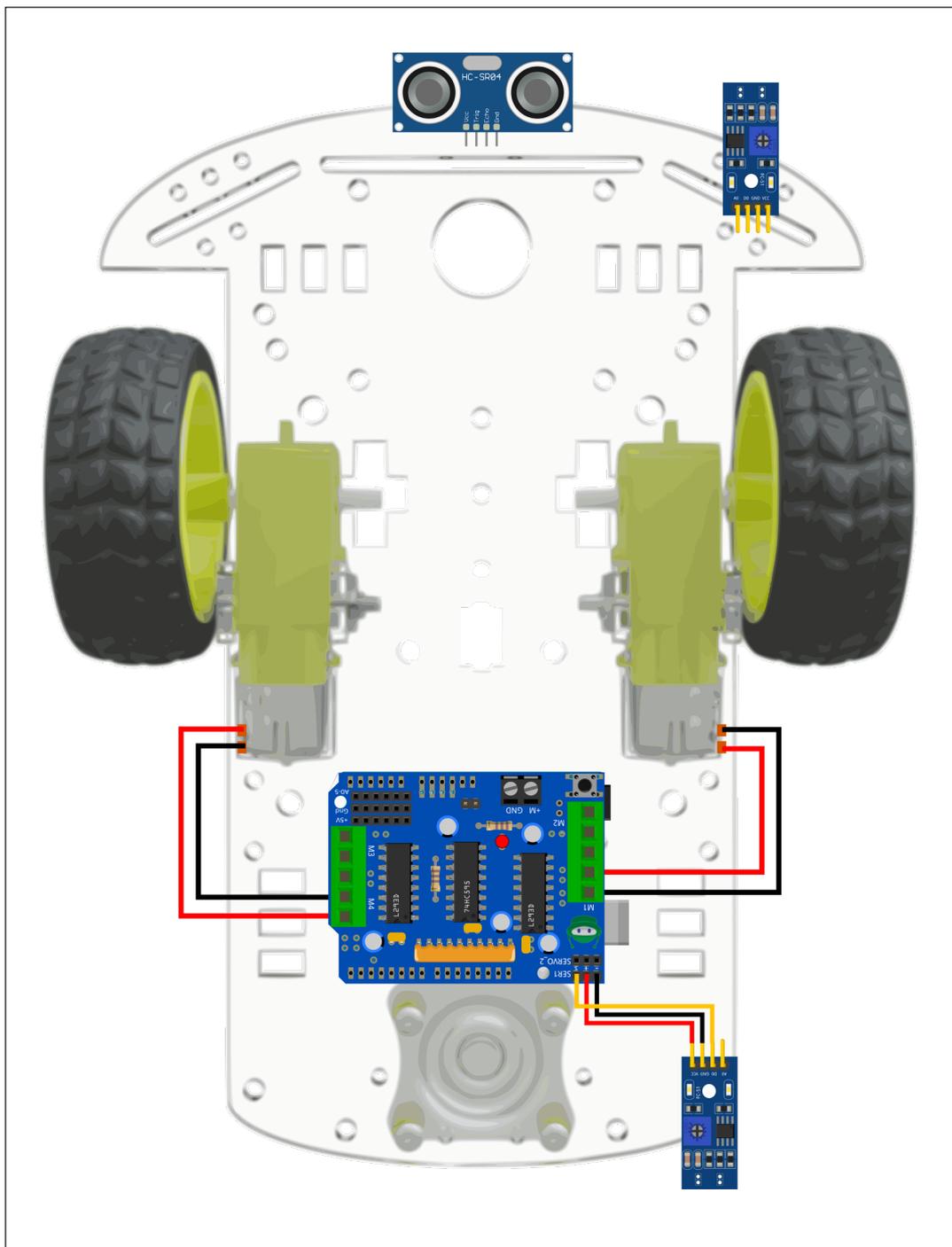
Com o auxílio de parafusos, fita adesiva ou cola quente, fixe os sensores sobre o chassi respeitando as posições - o sensor ultrassônico na parte frontal do robô e os sensores de obstáculos IR posicionados na parte frontal e traseira (figura 4).

Figura 4 - Fixando os sensores no chassi do robô.



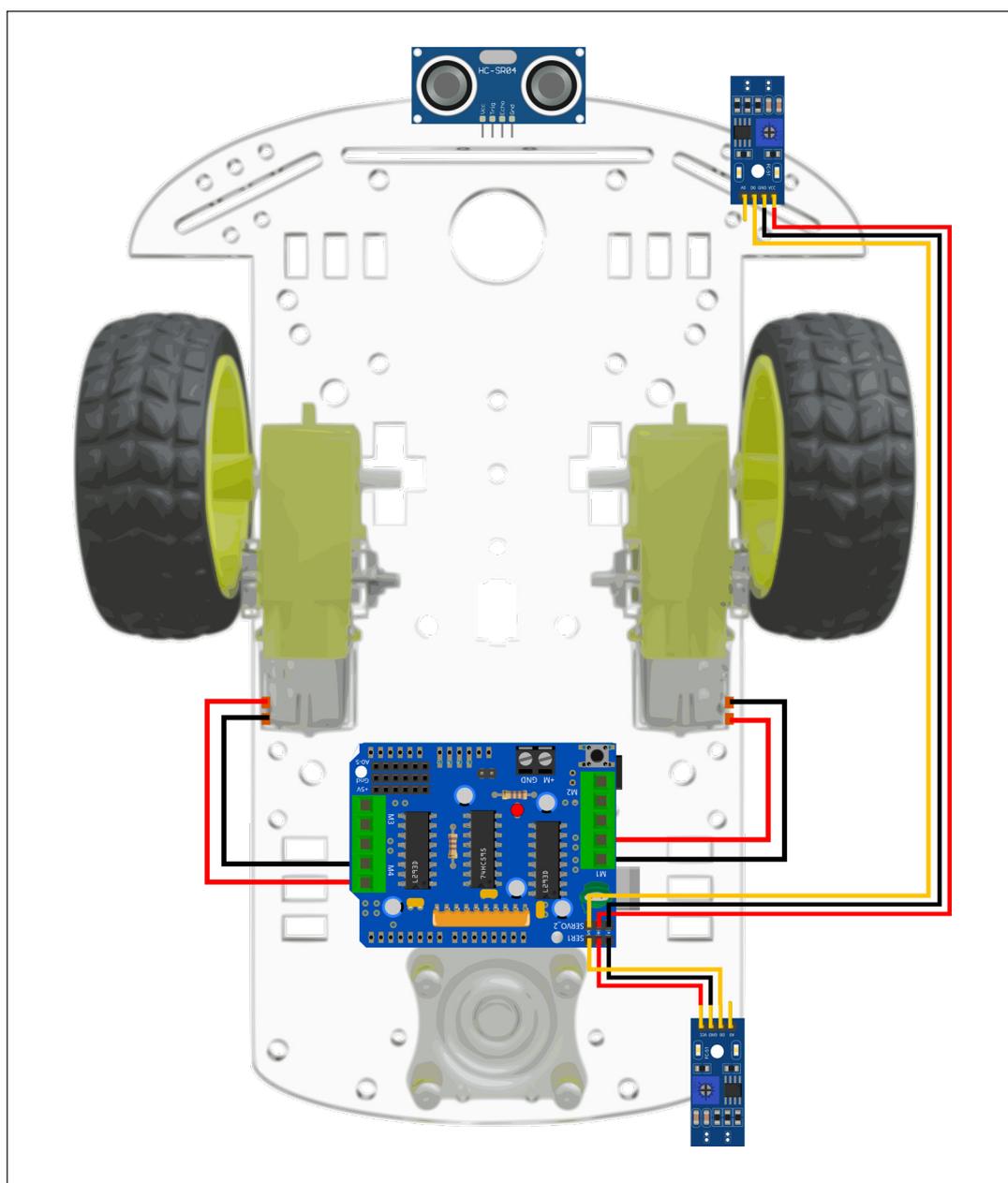
Com o auxílio de 3 jumpers fêmea-fêmea, conecte os pinos **VCC**, **GND** e **DO** do sensor de obstáculos IR traseiro aos pinos **+**, **-** e **S** do conector SER1 (servo 1) da placa motor shield (figura 5).

Figura 5 - Conectando módulo sensor de obstáculos IR traseiro à placa motor shield



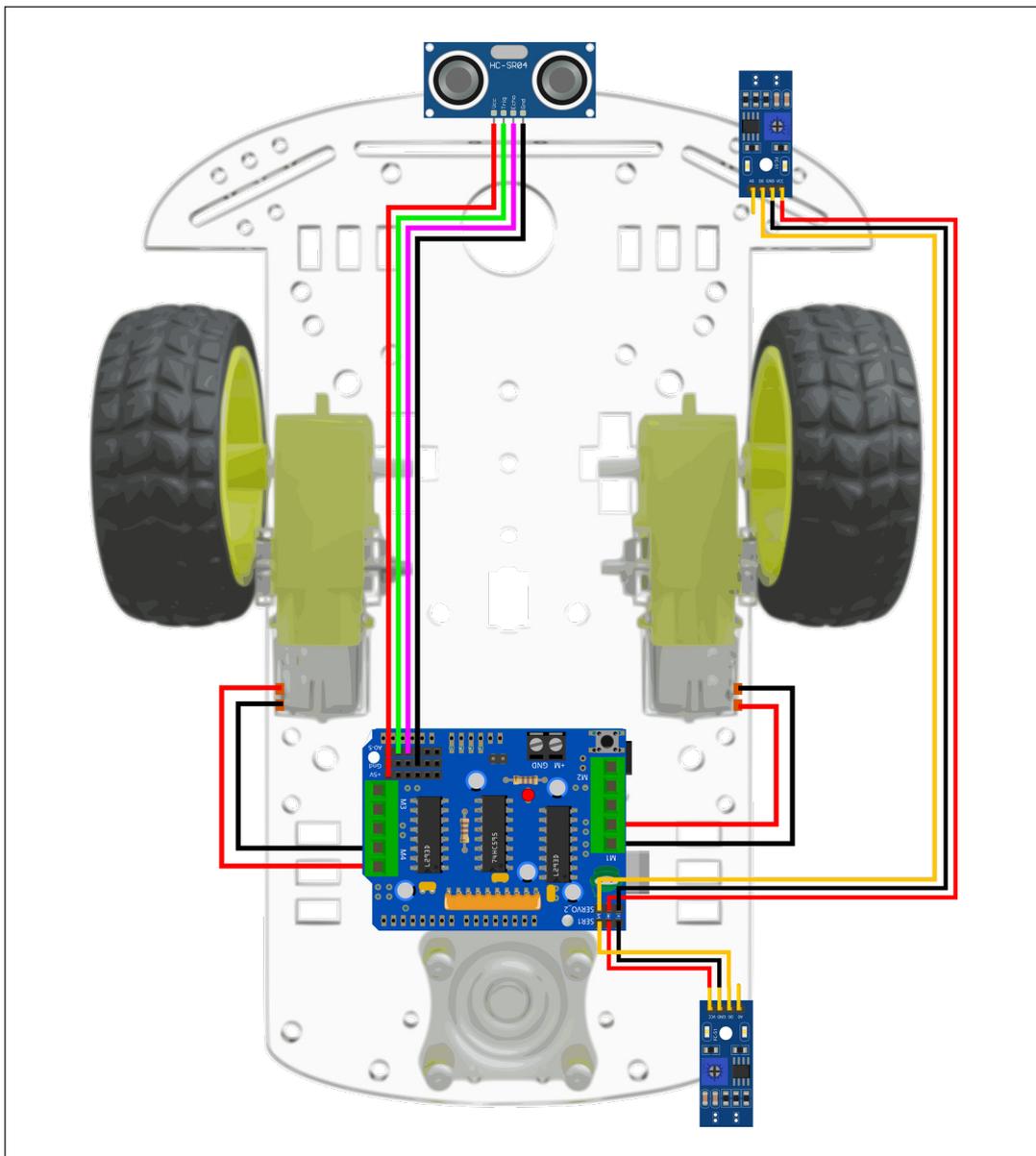
Utilizando mais 3 jumpers fêmea-fêmea, conecte os pinos **VCC**, **GND** e **DO** do sensor de obstáculos IR dianteiro aos pinos **+**, **-** e **S** do conector **SERVO_2** (servo 2) da placa motor shield (figura 6).

Figura 6 - Conectando o módulo sensor de obstáculos IR dianteiro à placa motor shield



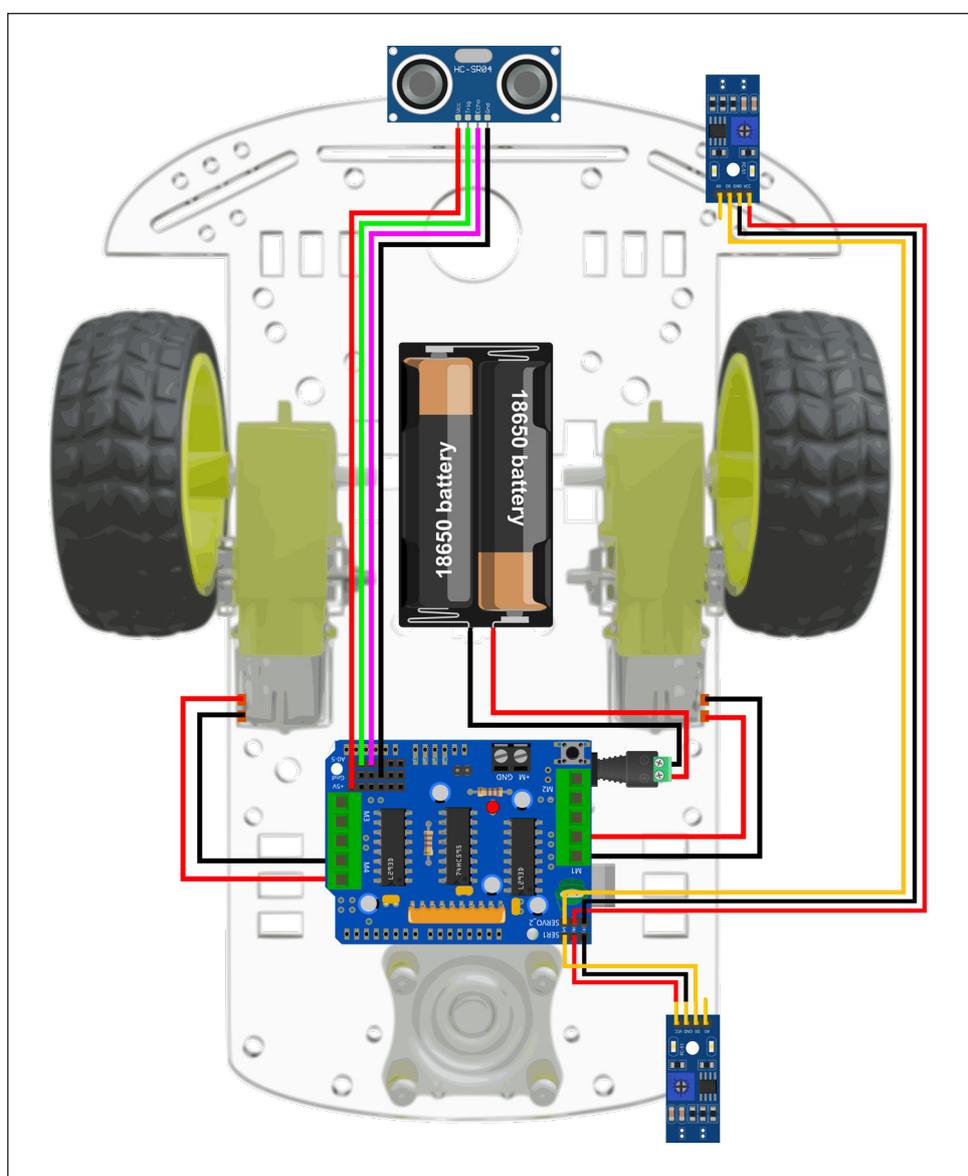
Utilizando 4 jumpers fêmea-fêmea vamos conectar o sensor ultrassônico à placa motor shield, utilizando as barras de pinos soldadas na placa. Faça as seguintes conexões: o pino **VCC** do módulo sensor ultrassônico interligue a um dos pinos positivos **+5V** (linha de pinos soldados à placa indicado por **+5V**), o pino **GND** do módulo ultrassônico interligue a um dos pinos negativos **GND** (linha de pinos soldados à placa indicado por **GND**), o pino **Echo** do módulo sensor ultrassônico ao pino **A3** (linha de pinos soldada à placa indicada por **A0-5**) e o pino **Trig** do módulo sensor ultrassônico ao pino **A4** (linha de pinos soldada à placa indicada por **A0-5**) (figura 7).

Figura 7 - Conectando sensor ultrassônico à placa motor shield



Por fim, procederemos à conexão da fonte de alimentação na motor shield. É crucial estar atento à fonte de energia utilizada, uma vez que os motores demandam um consumo significativo durante o funcionamento. Portanto, é essencial empregar uma fonte capaz de fornecer uma tensão de 6 a 12 Volts. Como exemplo, utilizou-se um estojo com duas pilhas 18650 (figura 8), no entanto, você tem a liberdade de escolher qualquer outra fonte que satisfaça os requisitos mínimos de operação do seu projeto.

Figura 8 - Conectando a fonte de alimentação do robô à motor shield



Com os componentes eletrônicos montados, vamos programar nosso protótipo, por codificação ou por blocos.

i. Linguagem de programação por codificação

Para essa programação, serão necessárias as bibliotecas **Adafruit Motor Shield Library**, elaborada pela Adafruit, e **Ultrasonic**, desenvolvida por Erick Simões, instaladas no software Arduino IDE. Na **Aula 05 - Softwares Arduino IDE e mBlock** você encontrará orientações sobre como realizar a instalação de bibliotecas, mas se você estiver utilizando a versão online do Arduino IDE, basta informar no sketch o uso das bibliotecas com os comandos `#include <AFMotor.h>` e `#include <Ultrasonic.h>`.

Uma vez instaladas as bibliotecas, inicie a programação conectando a placa Arduino Uno ao computador, com o uso de um cabo USB, para que ocorra a comunicação entre a placa microcontroladora e o software Arduino IDE.

No software IDE, escreva ou copie e cole o código-fonte de programação (quadro 1).

Quadro 1 - Código-fonte da programação na linguagem do Arduino (Wiring)

```
/*
*****
/* Aula 38 - Robô Sumô
/* Programação do robô sumô. O robô conta com dois
/* sensores de obstáculo Infravermelho que o permite
/* permanecer dentro do dojô e um sensor de distância
/* ultrassônico que o ajudará a detectar o adversário.
/* Links para obtenção da biblioteca by Erick Simões
/* http://librarymanager/All#minimalist#Ultrasonic
/* https://github.com/ErickSimoes/Ultrasonic
*****
/* Inclui a Biblioteca AFMotor para controle do motor.
#include <AFMotor.h>
/* Inclui a Biblioteca do sensor ultrassônico.
#include <Ultrasonic.h>
/* Definição dos pinos para os sensores.
#define pino_Trig A4
#define pino_Echo A3
#define pino_IR_traseiro 10
#define pino_IR_dianteiro 9
/* Definindo as variáveis utilizadas no código.
int distancia;
int sensor_dianteiro;
int sensor_traseiro;
```

```
/* Define a distância para ataque. (cm). */
int distancia_ataque = 15;
/* Define as velocidades do robô. (0-255). */
int vel_padrao = 150;
int vel_ataque = 255;

/* Cria o objeto Sensor para realizar a leitura. */
Ultrasonic ultrassonico(pino_Trig, pino_Echo);
/* Cria os objetos de controle para os motores */
/* conectados nas saídas M1 e M4 da Motor Shield. */
AF_DCMotor motor1(1);
AF_DCMotor motor4(4);

void setup() {
  /* Configura os pinos dos sensores IR como entradas */
  pinMode(pino_IR_dianteiro, INPUT);
  pinMode(pino_IR_traseiro, INPUT);
  /* Aguarda 3 segundos para iniciar a disputa. */
  delay(3000);
  /* Seta a velocidade dos motores. */
  motor1.setSpeed(vel_padrao);
  motor4.setSpeed(vel_padrao);
}

void loop() {
  /* Chama a função criada para a leitura dos sensores. */
  ler_sensores();
  /* Se o sensor dianteiro detectar a linha, faça... */
  if (sensor_dianteiro == 1) {
    Pare();
    Re();
    Pare();
    Manobre(250);
    Pare();
  }
  /* Se o sensor traseiro detectar a linha, faça... */
  if (sensor_traseiro == 1) {
    Pare();
    motor1.setSpeed(vel_ataque);
    motor4.setSpeed(vel_ataque);
    Ataque();
    delay(500);
    motor1.setSpeed(vel_padrao);
    motor4.setSpeed(vel_padrao);
  }
}
```

```

/* Se o sensor ultrassônico detectar o adversário, */
/* faça... */
if (distancia <= distancia_ataque) {
/* Enquanto o sensor dianteiro não chegar até a */
/* linha, faça... */
while (sensor_dianteiro == 0) {
motor1.setSpeed(vel_ataque);
motor4.setSpeed(vel_ataque);
Ataque();
/* Atualiza as informações dos sensores. */
ler_sensores();
}
motor1.setSpeed(vel_padrao);
motor4.setSpeed(vel_padrao);
}
/* Chama a função que move o robô para frente. */
Frente();
}

/* Funções criadas para operação do robô sumô. */

/* Função fâra as leituras dos sensores. */
void ler_sensores() {
distancia = ultrassonico.read();
sensor_dianteiro = digitalRead(pino_IR_dianteiro);
sensor_traseiro = digitalRead(pino_IR_traseiro);
}

/* Função mover robô para frente. */
void Frente() {
motor1.run(FORWARD);
motor4.run(FORWARD);
delay(1);
}

/* Função mover robô para trás. */
void Re() {
motor1.run(BACKWARD);
motor4.run(BACKWARD);
delay(500);
}

/* Função parar o robô. */
void Pare() {
motor1.run(RELEASE);
motor4.run(RELEASE);
delay(1);
}

```

```
/* Função ataque do robô. */
void Ataque() {
  motor1.run(FORWARD);
  motor4.run(FORWARD);
  delay(1);
}

/* Função parar manobrar o robô. */
void Manobre(int tempo) {
  if (random(1, 3) == 1) {
    motor1.run(FORWARD);
    motor4.run(BACKWARD);
    delay(1);
  } else {
    motor1.run(FORWARD);
    motor4.run(BACKWARD);
  }
  delay(tempo);
}
```

Com o código-fonte inserido no Arduino IDE, compile o programa pressionando o botão **Verify** (botão com sinal de tique) para verificar se não há erros de sintaxe. Estando o código correto, o próximo passo é realizar a transferência do programa para a placa de Arduino. Para isso, pressione o botão **Upload** (botão com uma seta apontando para a direita).

Após a transferência do programa para o Arduino, inicia-se o funcionamento do robô. Ao energizá-lo, o robô aguarda 3 segundos para começar a batalha. Após, verificará os estados dos sensores, caso o adversário não esteja ao seu alcance (menos de 15 cm de distância), ele permanecerá se movendo para frente em busca de seu adversário, com velocidade moderada. Ao detectar a presença do adversário, o robô entrará no modo de ataque, ou seja, ativará os motores com a velocidade máxima, empurrando para fora do dojô (ringue do sumô). Os sensores IR farão o monitoramento da área do dojô, caso o robô atinja o limite das bordas, os motores serão ativados para reposicioná-lo ao centro.

ii. Linguagem de programação por blocos

Outra forma de programar o robô sumô é por meio da linguagem de programação que utiliza blocos de funções prontas, os quais representam comandos de programação. Vamos utilizar o software mBlock.

Para conectar o mBlock ao Arduino, você deve clicar no ícone **Adicionar**, localizado no campo **Dispositivos**, e selecionar o Arduino, na biblioteca de dispositivos do mBlock, clicando, na sequência, no botão **OK**.

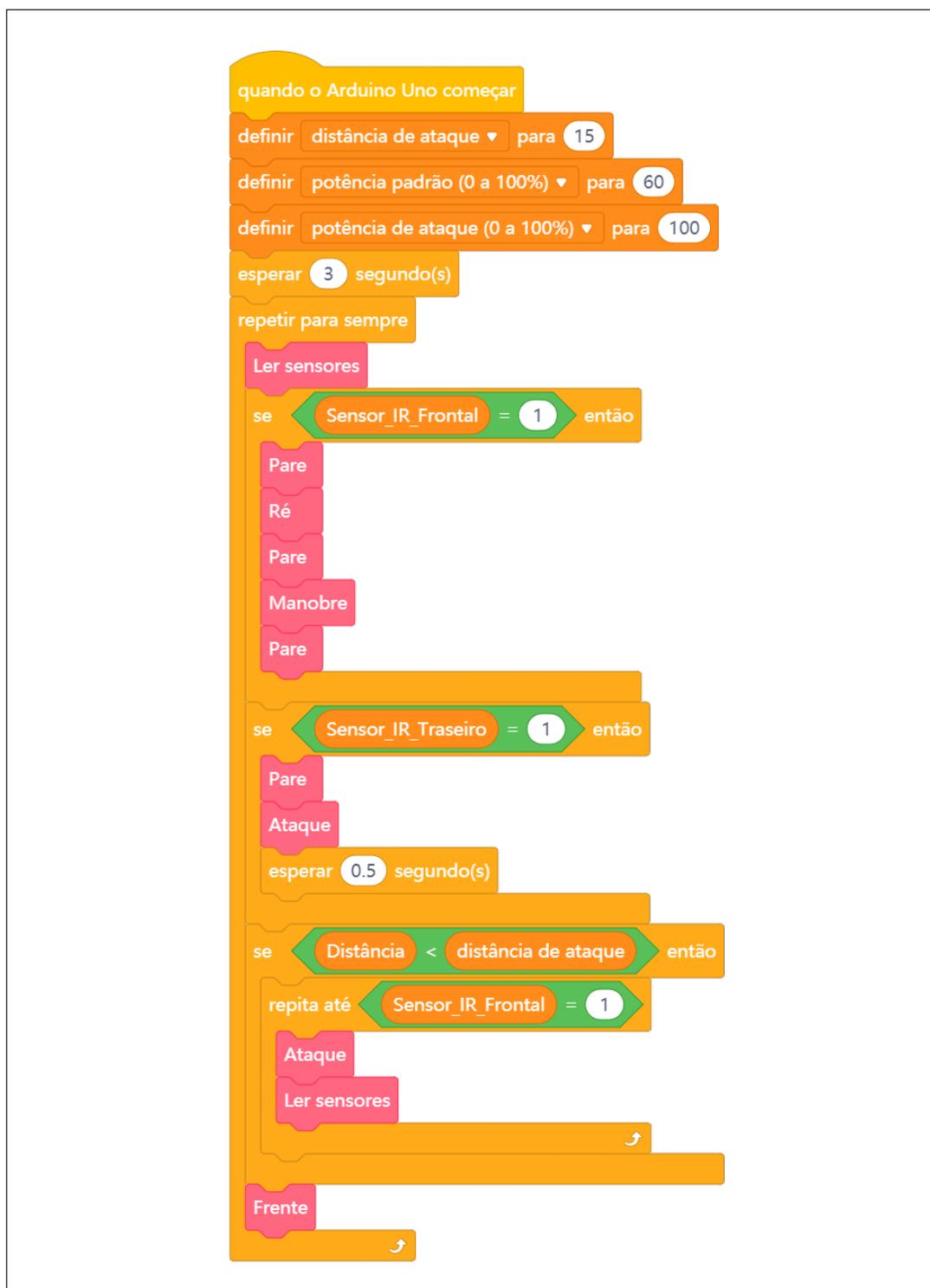
Uma vez selecionado, o Arduino Uno é visualizado no campo **Dispositivos** do mBlock e já é possível iniciar a programação em blocos.

Nessa programação, utilizaremos variáveis e criaremos blocos que auxiliarão na estrutura do nosso programa e a extensão **Motor Shield L293D**. Para recordar como criar variáveis, blocos e instalar uma extensão no mBlock, consulte a **Aula 05 - Softwares Arduino IDE e mBlock**.

Uma vez realizada a instalação da extensão necessária, monte os blocos, arrastando e soltando, de acordo com a programação dos movimentos do robô sumô (figura 9).



Figura 9 – Programação em blocos para funcionamento do robô sumô



```
definiir Ler sensores
definiir Sensor_IR_Frontal para ler pino digital 9
definiir Sensor_IR_Traseiro para ler pino digital 10
definiir Distância para ler pin trigonométrico do sensor de ultrassom 18 pin de eco 17

definiir Frente
  Move forward motor M1 at power potência padrão (0 a 100%) %
  Move forward motor M4 at power potência padrão (0 a 100%) %
  esperar 0.001 segundo(s)

definiir Ré
  Move backward motor M1 at power potência padrão (0 a 100%) %
  Move backward motor M4 at power potência padrão (0 a 100%) %
  esperar 0.5 segundo(s)

definiir Pare
  Stop motor M1
  Stop motor M4
  esperar 0.001 segundo(s)

definiir Ataque
  Move forward motor M1 at power potência de ataque (0 a 100%) %
  Move forward motor M4 at power potência de ataque (0 a 100%) %
  esperar 0.001 segundo(s)

definiir Manobre
  se escolher aleatório entre 1 e 2 = 1 então
    Move forward motor M1 at power potência padrão (0 a 100%) %
    Move backward motor M4 at power potência padrão (0 a 100%) %
  senão
    Move backward motor M1 at power potência padrão (0 a 100%) %
    Move forward motor M4 at power potência padrão (0 a 100%) %
  esperar 0.25 segundo(s)
```

Assim que os blocos estiverem montados, clique no botão **Conectar** para iniciar a comunicação entre o software mBlock com a placa de Arduino Uno. Ao clicar sobre o botão **Conectar**, aparecerá um *tooltip* solicitando a confirmação da conexão entre os dois dispositivos.

Uma vez realizada a conexão entre os dispositivos, será ativado, na interface do mBlock, o botão **Upload**. Ao clicar neste botão, o software verificará se não há erros na estrutura do programa e, então, compilará para enviar o programa à placa Arduino.

Após a transferência do programa para o Arduino, inicia-se o funcionamento do robô. Ao energizá-lo, o robô aguarda 3 segundos para começar a batalha. Após, verificará os estados dos sensores, caso o adversário não esteja ao seu alcance (menos de 15 cm de distância), ele permanecerá se movendo para frente em busca de seu adversário, com velocidade moderada. Ao detectar a presença do adversário, o robô entrará no modo de ataque, ou seja, ativará os motores com a velocidade máxima, empurrando para fora do dojô (ringue do sumô). Os sensores IR farão o monitoramento da área do dojô, caso o robô atinja o limite das bordas, os motores serão ativados para reposicioná-lo ao centro.



Desafios

i. Que tal alterar o modelo de decisão do robô ao se deparar com um obstáculo? Uma simples mudança no código, ao invés do robô ir para cima do adversário, ele simplesmente se afasta dele. O Sensor Ultrassônico é uma ótima opção para ajudar nesse desafio. Pode-se, também, elaborar um sistema que transporta uma determinada peça dentro de um ambiente fabril e consiga desviar de pessoas, paredes, obstáculos, dentre outros.



E se...

i. O projeto não funcionar, se atente a alguns dos possíveis erros:

Confira se a motor shield foi conectada corretamente sobre a placa Arduino.

Verifique se os motores estão conectados nas mesmas portas da motor shield (M1 e M4) que foram utilizadas na programação.

Ateste se os motores estão conectados as mesmas portas da motor shield (M2 e M3) que foram utilizadas na programação.

Verifique se os sensores estão conectados corretamente aos pinos da motor shield que foram utilizadas na programação.

Certifique de que a fonte de alimentação externa utilizada tenha carga suficiente para manter o robô em funcionamento, se necessário faça a substituição.

3. Feedback e Finalização (15min)

a. Confira, compartilhando seu projeto com os demais colegas, se o objetivo foi alcançado.

b. Analise seu projeto desenvolvido, de modo a atender os requisitos para funcionamento do robô sumô. .

c. 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?

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



Videotutorial

Com o intuito de auxiliar na montagem e na programação desta aula, apresentamos um videotutorial, disponível em:



<https://rebrand.ly/a38robotica>

Acesse, também, pelo QRCode:



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

EQUIPE ROBÓTICA PARANÁ

Andrea da Silva Castagini Padilha
Cleiton Rosa
Darice Alessandra Deckmann Zanardini
Edgar Cavalli Júnior
Edna do Rocio Becker
José Feuser Meurer
Marcelo Gasparin
Michele Serpe Fernandes
Michelle dos Santos
Orlando de Macedo Júnior
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/)

