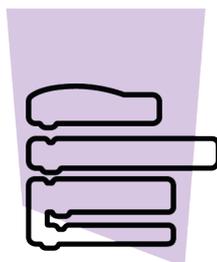
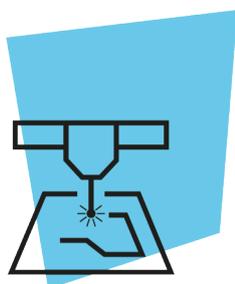
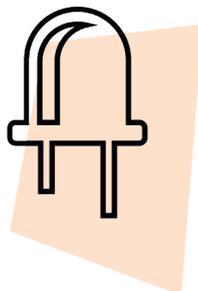
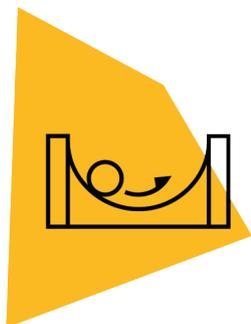
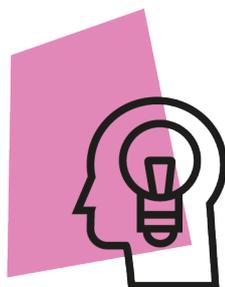
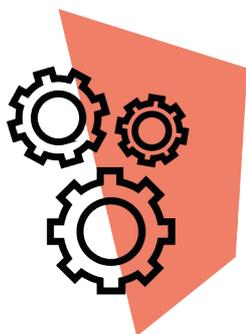


ROBÓTICA

Módulo 1



LED RGB

AULA **25**

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

Cleres Rozeli Cristófolli Grande

Darice Alessandra Deckmann Zanardini

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 - Compartilha Igual 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-Íris
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 24
Buzzer Passivo

Aula 25
LED RGB

Aula 26
Arco-Íris



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	8
3. Feedback e Finalização	19
Videotutorial	20



Introdução

Na **Aula 08 - LED e Resistor**, conhecemos o diodo emissor de luz, disponível em nosso kit de robótica nas cores amarelo, azul, branco, verde e vermelho. Agora, chegou o momento de conhecermos o componente eletrônico LED RGB, o qual permite a emissão de um espectro mais amplo de cores, com aplicações variadas.



Objetivos desta Aula

- Identificar as possibilidades de utilização do LED RGB;
- Compreender sobre circuitos e conceitos básicos de eletrônica;
- Abordar conceitos algorítmicos;
- Compreender conceitos da lógica booleana;
- Controlar o acendimento do LED RGB;
- Alternar as cores do LED RGB com Potenciômetro;
- Prototipar com Arduino;
- Programar por blocos ou código.



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.



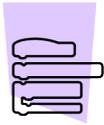
Habilidades do Século XXI a Serem Desenvolvidas

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



Lista de Materiais

- 01 Placa Protoboard;
- 01 Placa Arduino Uno R3;
- 01 Cabo USB;
- 09 Jumpers Macho-Macho;
- 01 Resistor 220 Ohms;
- 01 LED 5mm RGB alto brilho;
- 01 Potenciômetro linear;
- 01 Notebook;
- Software Arduino IDE ou mBlock.



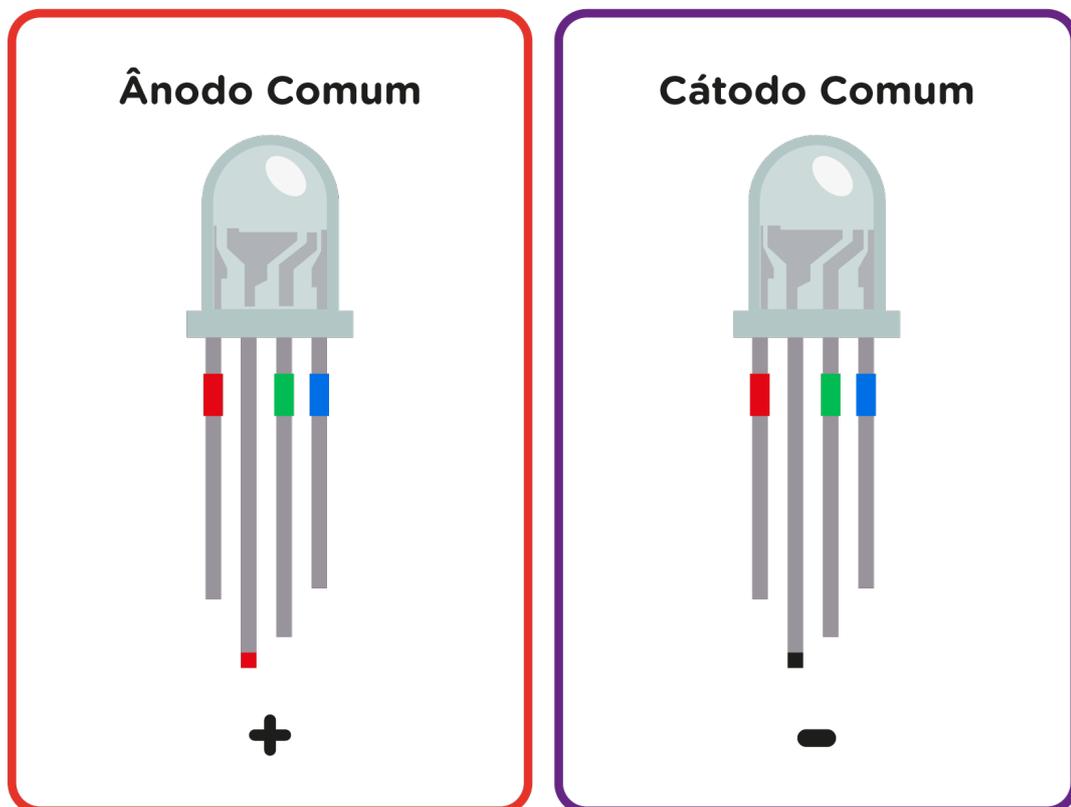
Roteiro da Aula

1. Contextualização (15min):

Você já observou algumas iluminações decorativas – como, por exemplo, jardins, piscinas, alto-falantes ou micro systems – em que um mesmo ponto de luz vai alternando entre cores? O que será que promove este efeito? E como controlar – ou definir – cada cor que será emitida?

Na **Aula 08 - LED e Resistor**, conhecemos a estrutura e o funcionamento de LEDs monocromáticos. O LED RGB é um outro tipo de diodo semiconductor que emite luz através da passagem de corrente elétrica. A diferença é que este modelo (RGB) é composto por três LEDs encapsulados em sua estrutura – cada pastilha de material semiconductor, no interior do LED RGB, é “responsável” por uma cor, conforme indicado na figura 1.

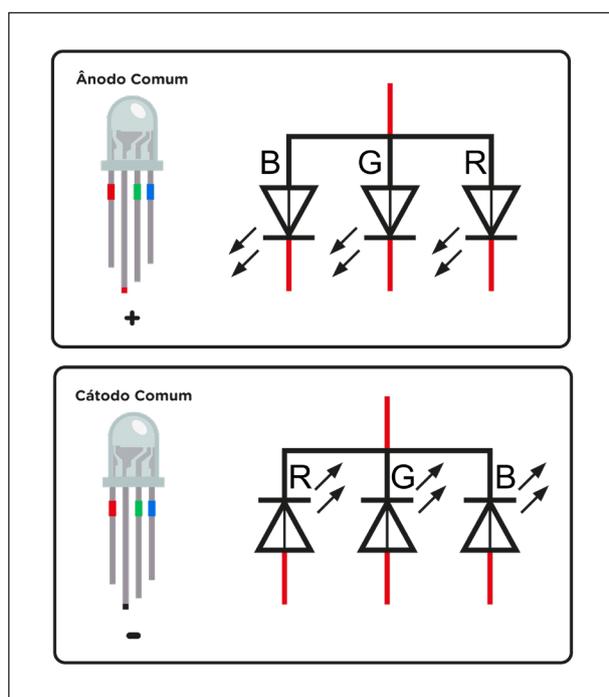
Figura 1 – Estrutura física do LED RGB modelos ânodo comum e cátodo comum



Fonte: SEED/DTI/CTE, 2022

Nesta estrutura física do LED RGB, como representado na figura 1, o terminal ou pino mais comprido, referente à pastilha interna maior, pode corresponder ao polo positivo (**ânodo comum**); em outros modelos, pode corresponder ao polo negativo (**cátodo comum**). A programação disponível nesta aula possui uma variável que contempla ambos os modelos de LED RGB - ânodo comum e cátodo comum. Portanto, a ligação deste LED RGB, no Arduino, considera que seu pino maior ânodo comum é conectado ao Vcc, enquanto os demais pinos são conectados ao GND, ou pode considerar que seu pino maior cátodo comum é conectado ao GND, enquanto os demais são conectados ao Vcc, conforme representado na figura 2

Figura 2 - Representação gráfica, para diagrama de circuito elétrico, do LED RGB cátodo comum



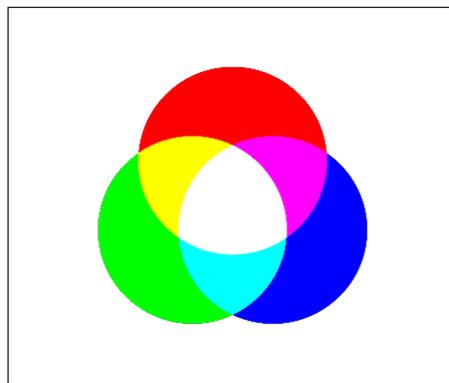
Fonte: SEED/DTI/CTE, 2022

Nas conexões, assim como para os LEDs monocromáticos, o LED RGB precisa de resistores para limitação do valor da corrente presente no circuito, sem risco de danos ao componente (confira, na **Aula 08 - LED e Resistor**, como calcular o valor adequado do Resistor a ser utilizado).

Os terminais ou pinos do LED RGB, referentes a cada cor, podem também ser conectados individualmente em uma fonte de energia, junto com o ânodo comum ou cátodo comum, conforme modelo do LED RGB, mas neste caso, priorizando a emissão de apenas uma das cores. No caso do LED RGB, o interessante é explorar a conexão de todos os pinos, simultaneamente às portas do Arduino, visando às potencialidades de uso deste componente eletrônico.

O LED RGB possibilita a emissão de um amplo espectro de luz visível através das cores primárias vermelho, verde e azul (por isso seu nome RGB, a partir do inglês *red*, *green* e *blue*).

Figura 3 - Sistema RGB



Fonte: WikiCommons



Para Saber Mais...

RGB é um sistema de cores utilizado em dispositivos eletrônicos, telas, monitores e design no qual, por meio da combinação do vermelho, do verde e do azul, possibilita a geração de variadas tonalidades (na **Aula 26 - Arco-Íris**, trabalharemos com a definição de cores por meio da regulação dos valores do RGB).

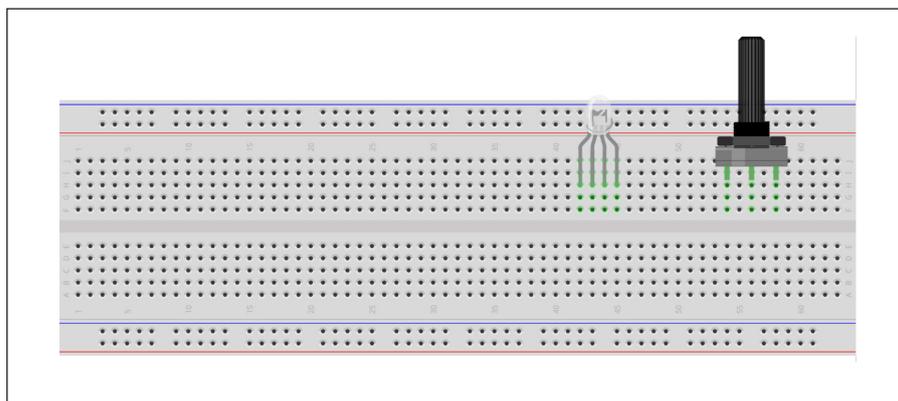
A programação do LED RGB envolve o controle individual de cor, quando seus pinos são conectados ao Arduino e utilizando a função `digitalWrite`, ou o controle de todas as cores a partir da delimitação de cada valor do RGB utilizando a função `analogWrite` (na **Aula 26 - Arco-Íris**, exploraremos os valores do RGB em um projeto com muita cor!).

Nesta aula, para alternar o LED RGB entre as cores vermelho, verde e azul, utilizaremos o Potenciômetro, componente para controle de corrente que conhecemos na **Aula 23 - Potenciômetro**.

2. Montagem e Programação (60min):

Vamos, primeiro, iniciar a montagem dos componentes eletrônicos. Encaixe na Protoboard o LED 5mm RGB alto brilho e o Potenciômetro, conforme indicado na figura 4.

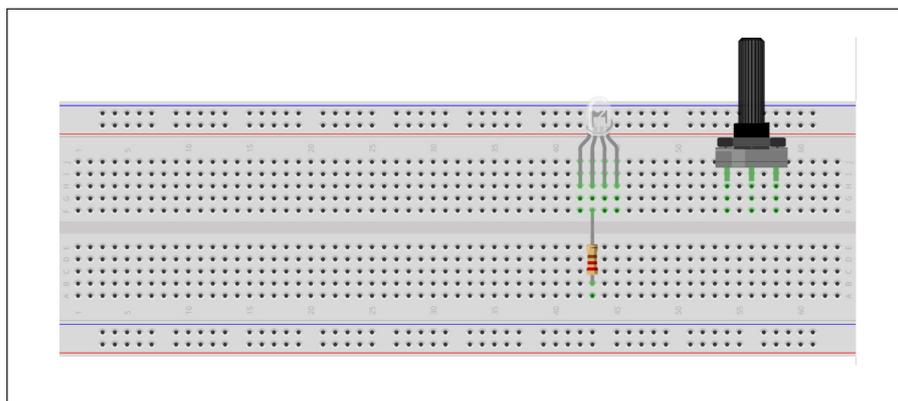
Figura 4 - Inserção do LED RGB e do Potenciômetro na Protoboard



Fonte: Fritzing

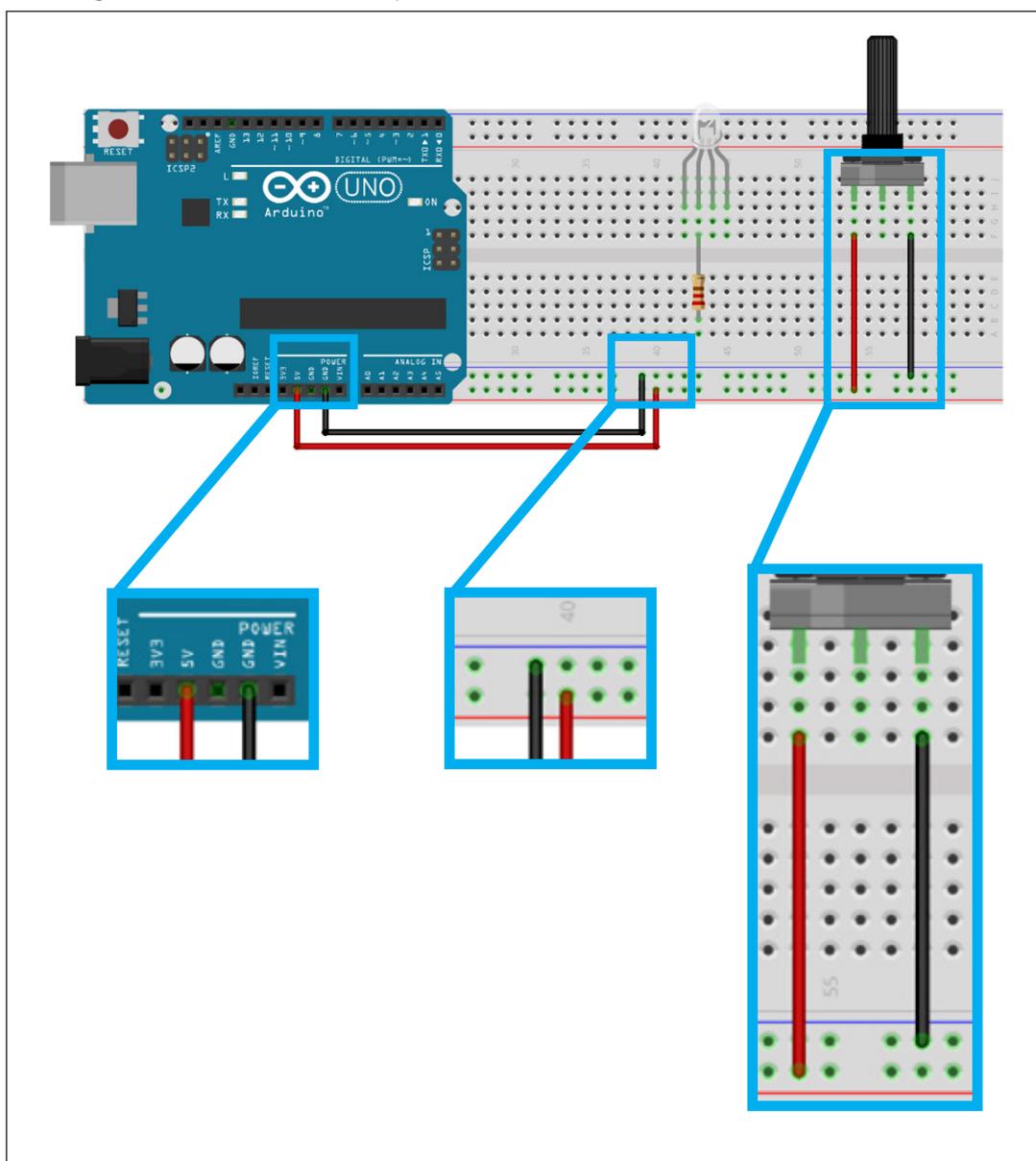
Insira o Resistor, conectando um de seus terminais no terminal negativo do LED RGB (conforme vimos anteriormente na estrutura do LED RGB é o pino ou terminal mais longo), e o outro terminal na parte inferior da Protoboard, conforme mostra a figura 5.

Figura 5 - Inserindo Resistor na Protoboard



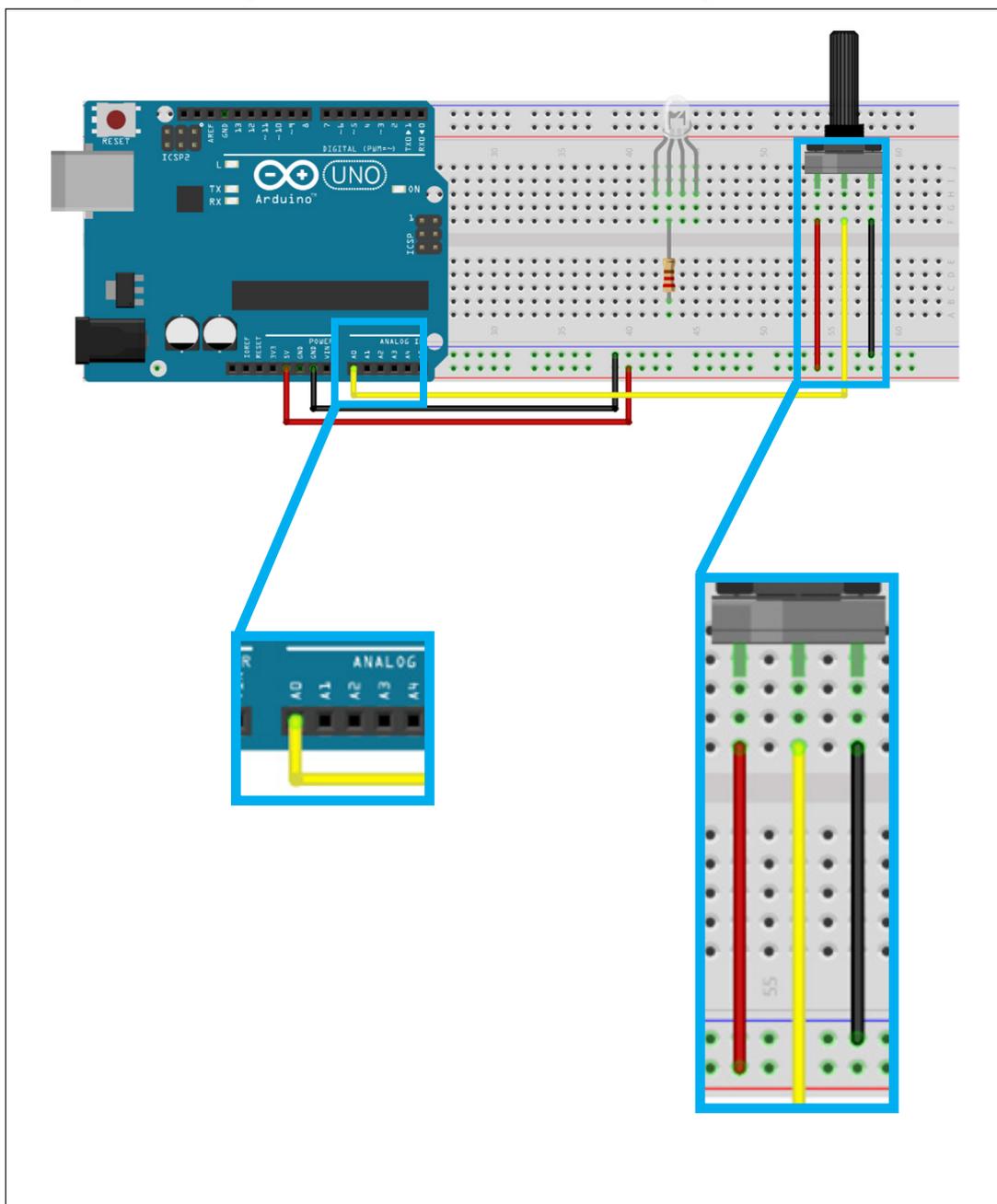
Vamos, agora, alimentar a placa Protoboard, conectando 2 Jumpers às portas GND e 5V do Arduino até as duas linhas inferiores da Protoboard (azul e vermelha, respectivamente). Para o Potenciômetro, conectaremos 2 Jumpers entre seus terminais extremos e as duas linhas inferiores da Protoboard (vermelha e azul), como mostra a figura 6.

Figura 6 - Alimentando a placa Protoboard, LED RGB e o Potenciômetro



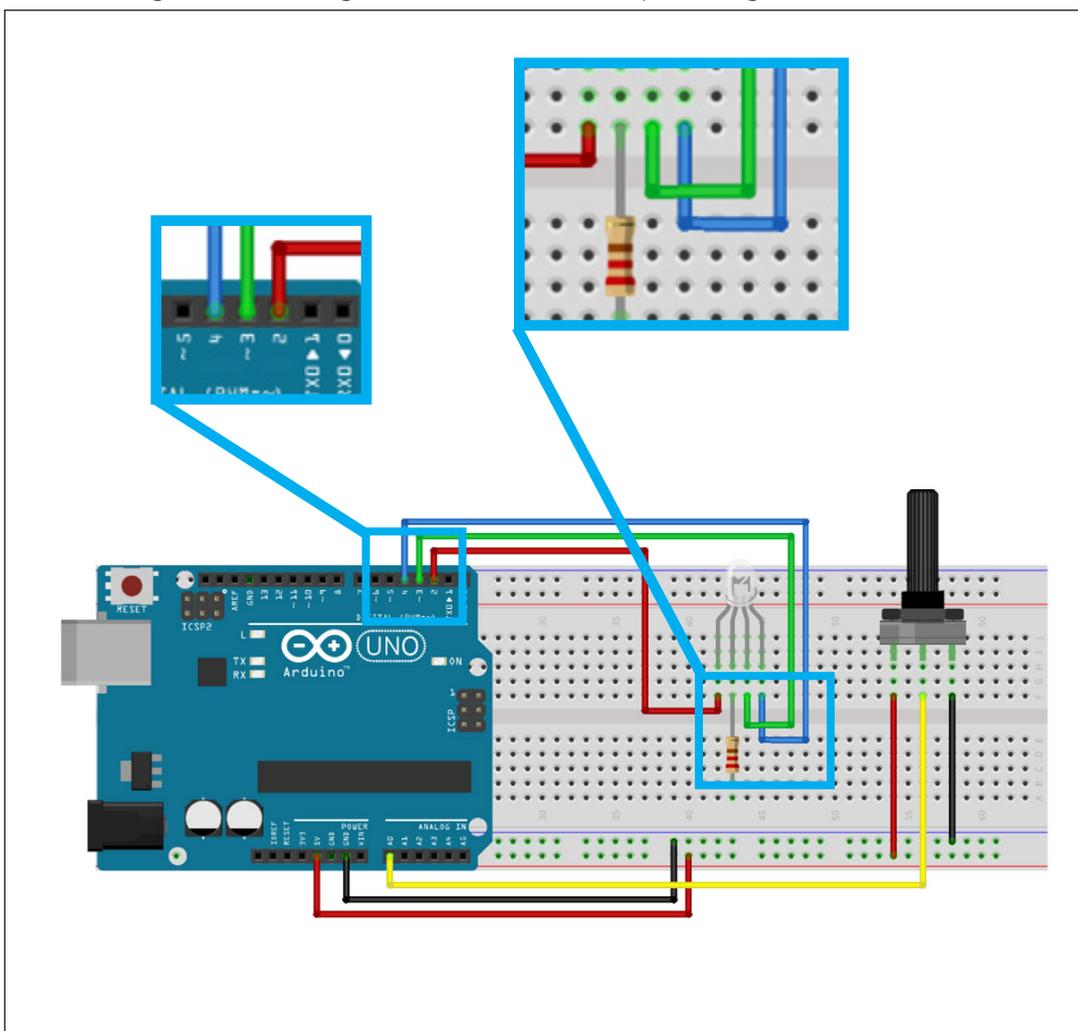
Interligue, com 1 jumper, o terminal central do Potenciômetro ao pino ou porta analógica A0 do Arduino, como indicado na figura 7.

Figura 7 - Interligando o Potenciômetro à porta analógica da placa Arduino



Utilizando mais 3 Jumpers, conecte o LED RGB às portas ou pinos digitais 2, 3 e 4 da placa Arduino, conforme a ordem representada pela figura 8.

Figura 8 - Interligando o LED RGB aos pinos digitais do Arduino



Por fim, utilizando mais 1 Jumper, conecte o terminal comum do LED RGB (aquele ao qual o Resistor já está interligado) à linha lateral da Protoboard, respeitando o tipo de LED RGB que você possui: LED RGB com ânodo comum interliga com a linha lateral vermelha da Protoboard (figura 9) e LED RGB com cátodo comum interliga com a linha lateral azul da Protoboard (figura 10).

Figura 9 - Interligando o LED RGB ânodo comum à linha vermelha da Protoboard

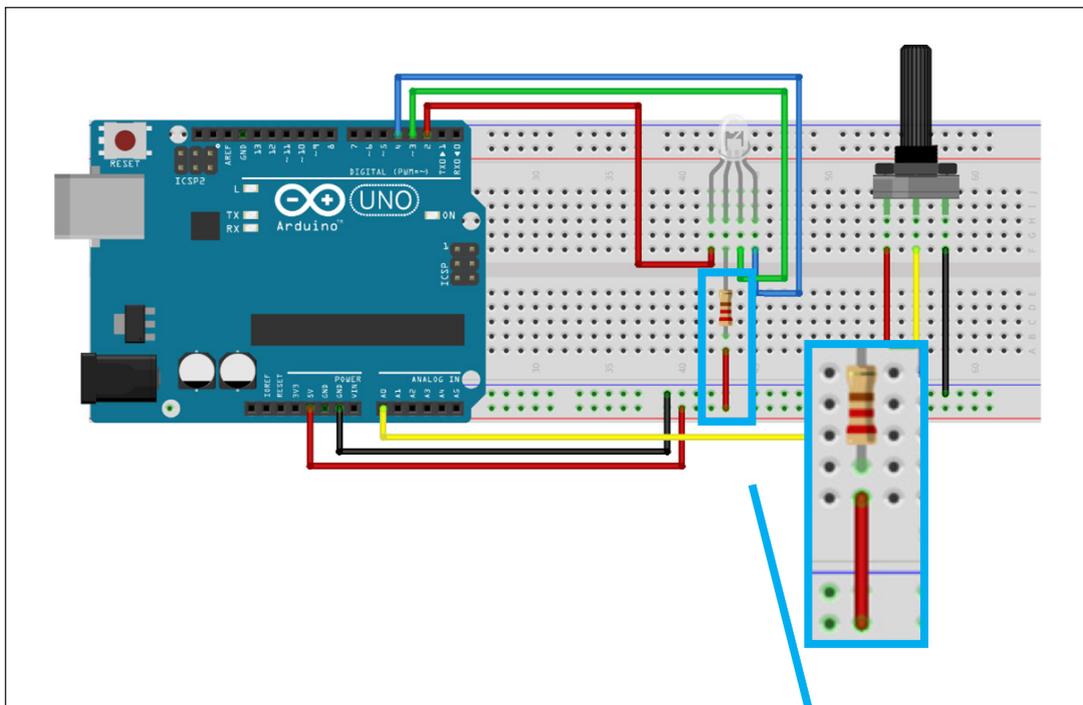
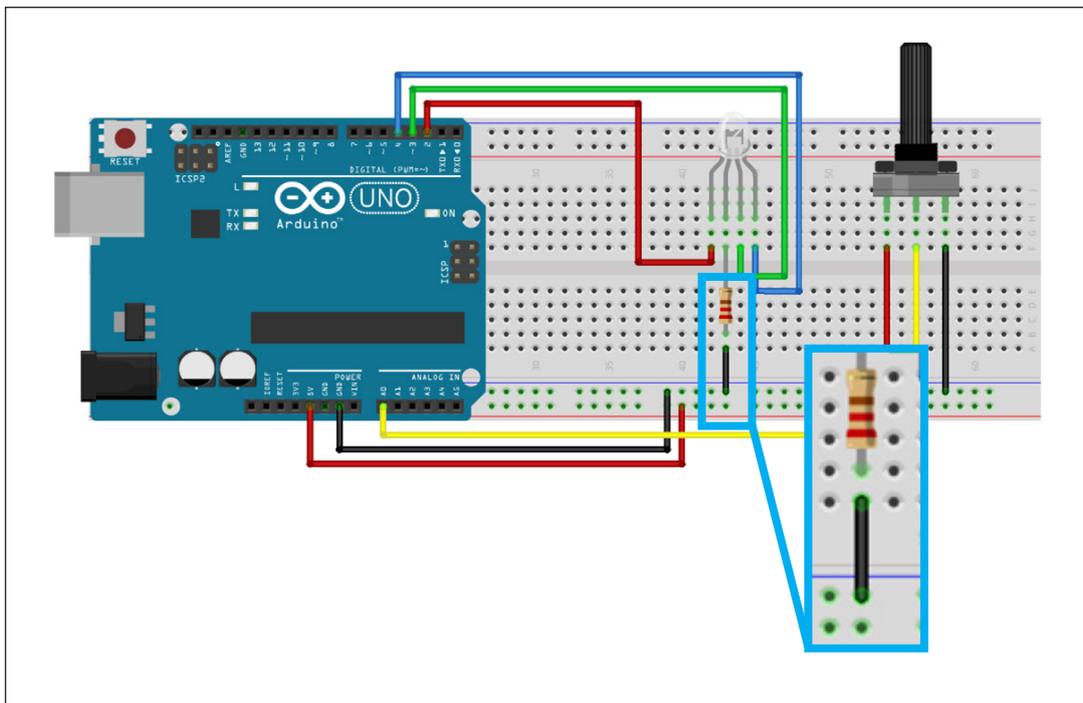


Figura 10 - Interligando o LED RGB cátodo comum à linha azul da Protoboard



Fonte: Fritzing



Agora, vamos programar!

Com os componentes eletrônicos montados, vamos programar, por codificação e por blocos, o acendimento do LED RGB e seu controle pelo Potenciômetro.

i. Linguagem de programação por código

Para iniciar a programação, conecte a placa Arduino ao computador, através 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, lendo com atenção os comentários da programação e atentando-se à variável que define se a programação será aplicada ao modelo de LED RGB ânodo comum ou cátodo comum, conforme apresentado no quadro 01:

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

```
/*
*****
*/
/* Aula 25 - LED RGB */
/* Programação: Alternando as cores do LED RGB através */
/* do sensor potenciômetro. */
/*
*/
/* ATENÇÃO! */
/* Nas linhas 13 ou 14 deste código, você deverá informar*/
/* se o LED RGB utilizado é ânodo comum ou cátodo */
/* comum para o correto funcionamento deste protótipo */
/* descomentando a linha correspondente ao seu LED RGB. */
*****
*/

// char modelo[] = "Ânodo Comum";
// char modelo[] = "Cátodo Comum";

/* Define o pino analógico A0 para o potenciômetro. */
int Pin_pot = A0;
/* Variável que armazenará os dados do potenciômetro. */
int Pot = 0;
/* Define o pino digital 2 para o LED Vermelho. */
int Led_R = 2;
/* Define o pino digital 3 para o LED Verde. */
int Led_G = 3;
/* Define o pino digital 4 para o LED Azul. */
int Led_B = 4;
/* Variáveis para controle dos LEDs. */
int ligar;
int desligar;
```

```
void setup() {
  /* Define o pino Led_R como SAÍDA. */
  pinMode(Led_R, OUTPUT);
  /* Define o pino Led_G como SAÍDA. */
  pinMode(Led_G, OUTPUT);
  /* Define o pino Led_B como SAÍDA. */
  pinMode(Led_B, OUTPUT);

  /* Rotina que verifica o tipo de LED RGB utilizado e */
  /* adequa os parâmetros para ligar e desligar o LED. */
  if (modelo[0] != 'C') {
    ligar = 0;
    desligar = 1;
  } else {
    ligar = 1;
    desligar = 0;
  }
  /* Inicia com os LEDs desligados. */
  digitalWrite(Led_R, desligar);
  digitalWrite(Led_G, desligar);
  digitalWrite(Led_B, desligar);
}
void loop() {
  /* Armazena os dados do potenciômetro na variável Pot. */
  Pot = analogRead(Pin_pot);
  /* Se Pot estiver entre 0 e 341, faça... */
  if (Pot >= 0 && Pot <= 341) {
    /* Ligue o LED Vermelho. */
    digitalWrite(Led_R, ligar);
    /* Senão... */
  } else {
    /* Desligue o LED Vermelho. */
    digitalWrite(Led_R, desligar);
  }
  /* Se Pot estiver entre 342 e 682, faça... */
  if (Pot >= 342 && Pot <= 682) {
    /* Ligue o LED Verde. */
    digitalWrite(Led_G, ligar);
    /* Senão... */
  } else {
    /* Desligue o LED Verde. */
    digitalWrite(Led_G, desligar);
  }
}
```

```

/* Se Pot estiver entre 686 e 1023, faça... */
if (Pot >= 683 && Pot <= 1023) {
  /* Ligue o LED Azul. */
  digitalWrite(Led_B, ligar);
  /* Senão... */
} else {
  /* Desligue o LED Azul. */
  digitalWrite(Led_B, desligar);
}
}

```

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 o Arduino. Para tal, pressione o botão **Upload** (botão com uma seta apontando para a direita).

Após a transferência do programa para o Arduino, as cores do LED RGB (vermelha, verde e azul) podem ser alternadas através do giro do eixo do Potenciômetro.

ii. Linguagem de programação por blocos

Outra forma de programar o controle do LED RGB, através do Potenciômetro, é por meio da linguagem de programação que utiliza blocos de funções prontas, os quais representam comandos de programação. Para isso, 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.

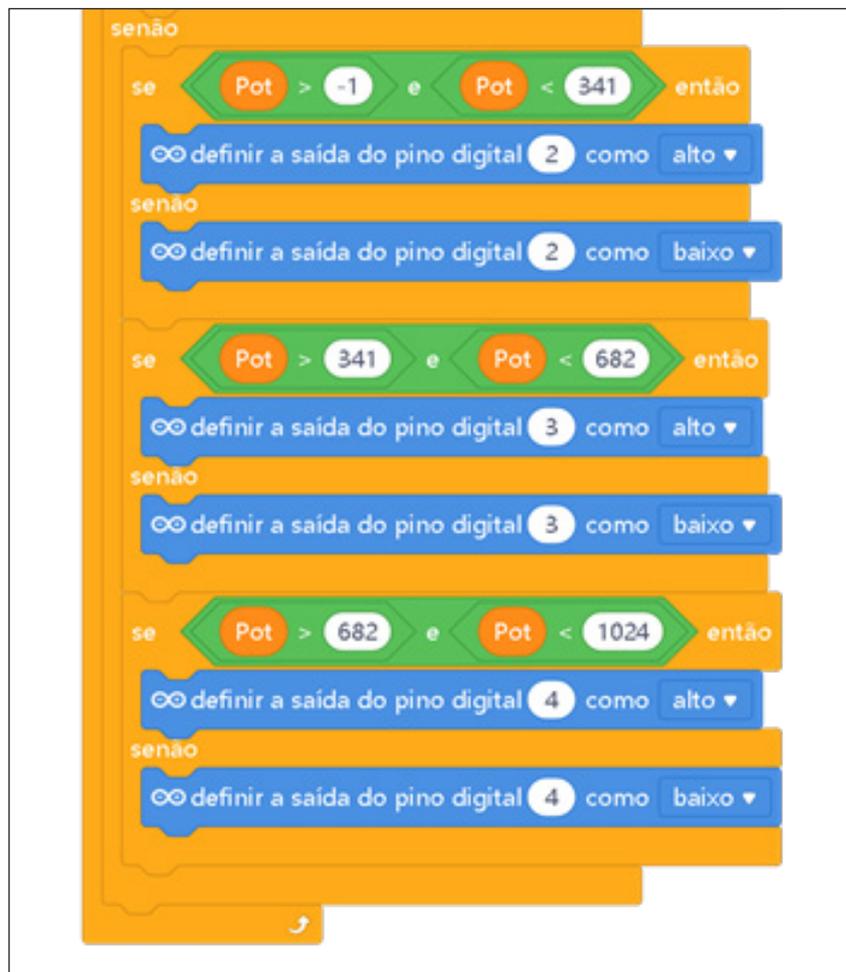
Nesta programação, utilizaremos variáveis que auxiliarão na estrutura do nosso programa (para recordar como criar uma variável, consulte a **Aula 05 - Softwares Arduino IDE e mBlock**).

Monte os blocos, arrastando e soltando, de acordo com a programação de funcionamento para controlar o LED RGB através do Potenciômetro, como mostra a figura 11.

Atenção! No segundo bloco informe o modelo do LED RGB utilizado no seu protótipo (0 para ânodo comum ou 1 para cátodo comum).

Figura 11 - Programação em blocos para controle do LED RGB





Fonte: mBlock

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 irá verificar se não há erros na estrutura do programa e, então, compilará para enviar o programa à placa Arduino.

Com a transferência do código para o dispositivo Arduino Uno, inicia-se o funcionamento do projeto: agora, as cores do LED RGB (vermelha, verde e azul) podem ser alternadas através do giro do eixo do Potenciômetro.



Desafios:

I. Que tal modificar essa programação e criar um efeito de transição entre as cores? Ao iniciar o giro do eixo do Potenciômetro, o LED RGB gradativamente acende, em uma dada cor, até seu valor máximo; na sequência, diminui até apagar, e então, inicia o acendimento gradativo da próxima cor – e assim, sucessivamente, passando pelas três cores até o final do giro do eixo.

II. Que tal alterar a programação para controlar, através do Potenciômetro, diferentes efeitos com o LED, observando e testando os resultados obtidos?



E se... ?

i. Os pinos do LED RGB estiverem com o mesmo comprimento, como identificar o ânodo ou cátodo comum?

1. Observe o interior da capsula do LED RGB: a maior pastilha interna de material semicondutor corresponderá ao ânodo ou cátodo comum, enquanto as menores a cada uma das cores RGB.

ii. Caso o projeto não funcione, se atente a alguns dos possíveis erros:

1. Verifique se os jumpers estão na mesma coluna dos terminais dos componentes, fazendo assim a conexão;

2. Verifique se os jumpers estão ligados nos pinos corretos no Arduino;

3. Verifique se o LED não está conectado de modo invertido;

4. Teste a integridade do LED RGB, inserindo o terminal negativo na porta GND e, para a conferência de cada cor, repita o teste inserindo cada terminal positivo diretamente na porta 13, com o Arduino desligado. Ao ligá-lo, o LED deverá piscar 3 vezes.

5. Verifique se a programação está adequada a cada porta digital e ao modelo do LED RGB – ânodo comum ou cátodo comum.

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 aos requisitos para controlar o acendimento do LED RGB e alternar cores com o giro do eixo do Potenciômetro.
- 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.



AULA

25



Videotutorial

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



<https://rebrand.ly/a25robotica>

Acesse, também, pelo QRCode:



20

Robótica

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/)

