Diretoria de Tecnologia e Inovação

ROBÓTICA

Módulo 1

Arco-Íris AULA 26

GOVERNO
DO ESTADO DO PARANÁ
SECRETARIA DE EDUCAÇÃO
E DO ESPORTE

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

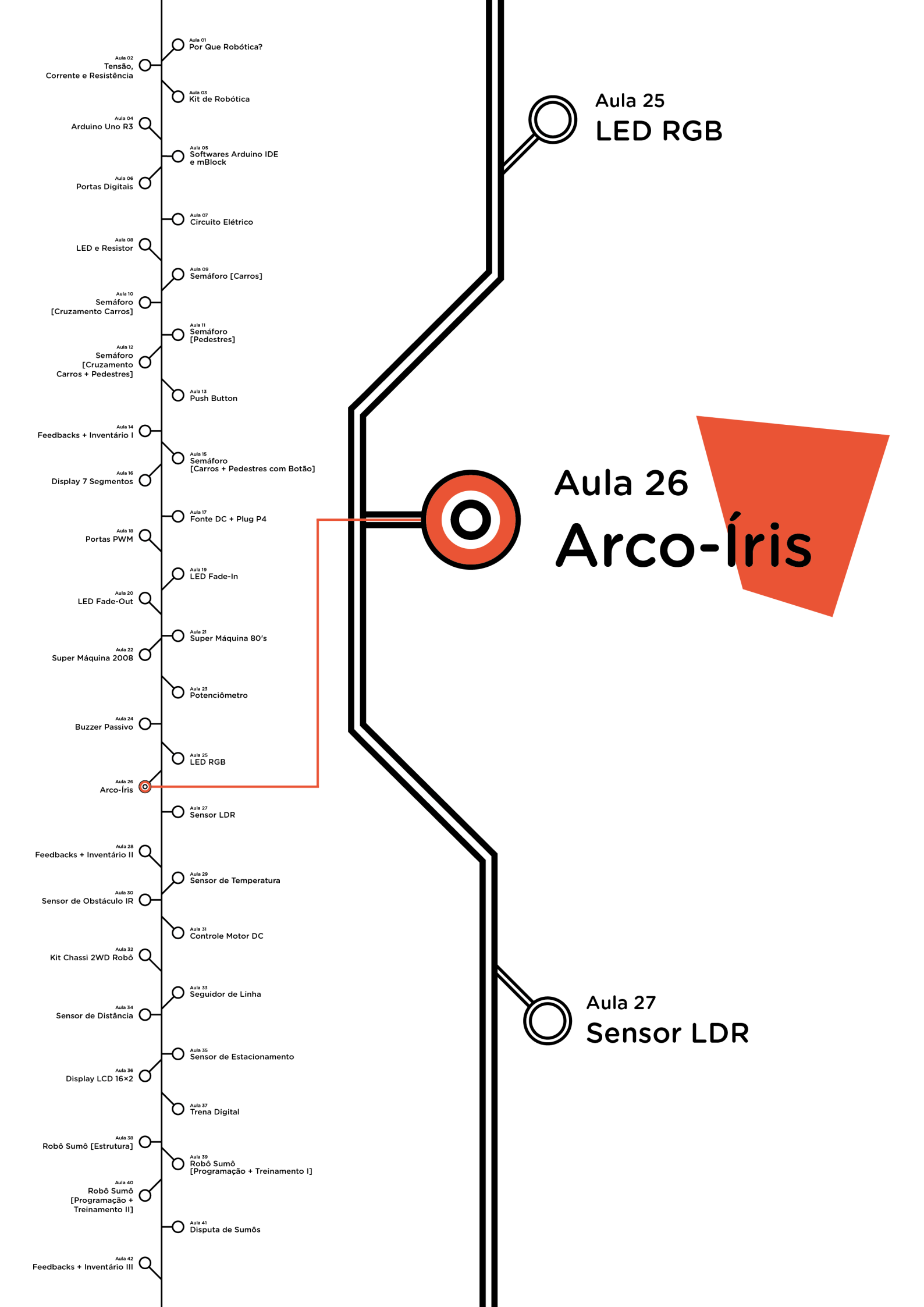
COORDENADOR DE TECNOLOGIAS EDUCACIONAIS
Marcelo Gasparin

Produção de Conteúdo
Cleiton Rosa
Darice Alessandra Deckmann Zanardini

Revisão Textual
Adilson Carlos Batista

Projeto Gráfico e Diagramação
Edna do Rocio Becker

2021


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 16×2
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 25
LED RGB
Aula 26
Arco-Íris
Aula 27
Sensor LDR

Sumário

AULA

26 ARCO-ÍRIS

Introdução

Na **Aula 25 - LED RGB**, conhecemos este componente eletrônico composto por três diodos emissores de luz de cores diferentes (vermelho, verde e azul). Agora, chegou o momento de trabalharmos com recursos que permitem a geração de um amplo espectro de cores em um projeto com muita cor!

Objetivos desta Aula

- Conhecer fenômenos óticos;
- Alterar valores do LED RGB;
- Alternar as cores do LED RGB com Potenciômetro;
- Abordar conceitos algorítmicos;
- Compreender conceitos da lógica booleana;
- Abordar o uso da função map();
- Utilizar a função analogWrite();
- 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.

AULA

26

Habilidades do Século XXI a Serem Desenvolvidas

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

Lista de Materiais

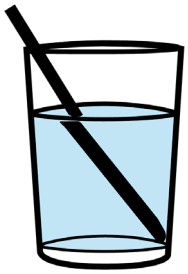
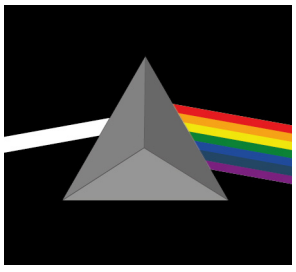
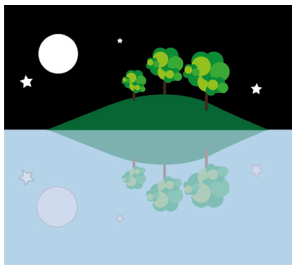
- 01 Placa Protoboard;
- 01 Placa Arduino Uno R3;
- 01 Cabo USB;
- 09 Jumpers macho-macho;
- 03 Resistores 220 Ohms;
- 01 LED 5mm RGB Alto Brilho;
- 01 Potenciômetro Linear;
- 01 Notebook;
- Software Arduino IDE ou mBlock.

Roteiro da Aula

1. Contextualização (15min):

Você já olhou para o céu, em um dia com sol, logo após ter chovido? Se sim, você, provavelmente, deve ter avistado um arco-íris! E você já observou quando a luz incide sobre um prisma de vidro ou um cristal transparente? Nestes fenômenos, podemos enxergar um arco-íris também!

O arco-íris é um fenômeno físico gerado através da decomposição da luz branca (ou luz policromática, pois é a mistura de todas as cores) do sol em outras cores.

Fenômenos Físicos da Ótica		
Refração da Luz	**Dispersão (ou Decomposição) da Luz**	**Reflexão da Luz**
A onda, ao passar de um meio para outro, sofre alteração na direção de sua propagação.	A onda, ao se refratar, se decompõe em cores.	Após incidir sobre uma superfície, a onda, ou parte dela, pode retornar.
		

AULA 26

Para Saber Mais...

Disco de Newton

O astrônomo, filósofo, físico e matemático inglês Isaac Newton (1643 - 1727), ao observar o comportamento da luz solar atravessando um prisma de vidro, percebeu que, em movimento inverso, a luz branca é a soma de todas as cores.

Utilizando um disco pintado com as cores do arco-íris e movimentos circulares rápidos, você pode confirmar esta afirmação de Newton e ver que quando nosso olho percebe a junção de todas as cores, tem-se o efeito da luz branca, conforme mostra a figura 1.

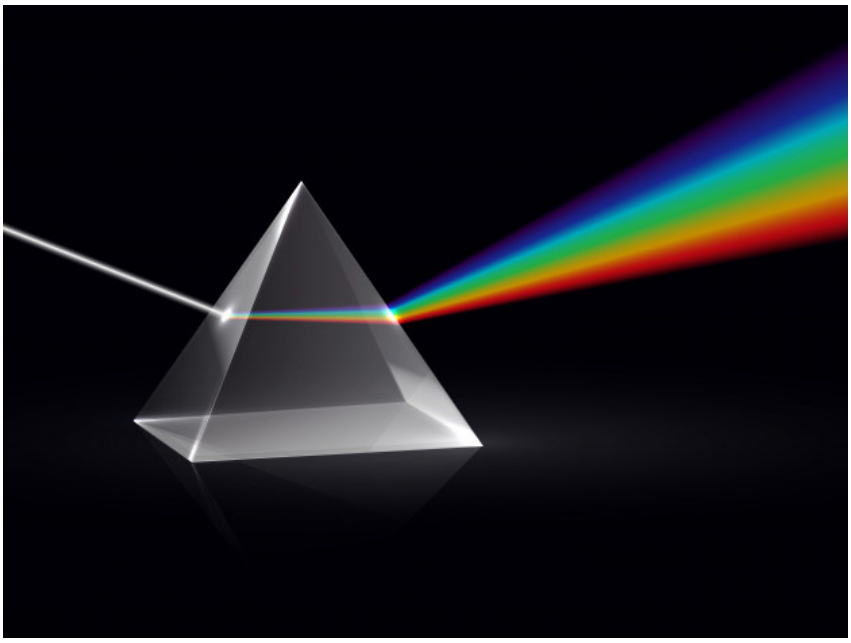
Figura 1 – Exemplo de montagem do Disco de Newton

Confira, nesta atividade do Manual do Mundo, a proposta de construção de um "Disco do Newton" para você fazer em casa: Azul + Verde + Vermelho = Branco?

Azul + Verde + Vermelho = Branco?

Conforme o ângulo do espectador em relação ao sol, tal percepção pode ser mais nítida em função da refração e dispersão da luz. Então, quando a luz do sol passa pelas gotas de água presentes na atmosfera, a luz branca é decomposta em cores - as quais os físicos denominam “espectro visível” - e o arco-íris é gerado. O mesmo efeito arco-íris ocorre, como podemos ver na figura 2, quando a luz passa por um cristal ou por um prisma de vidro.

Figura 2 - Dispersão da luz branca em um prisma



Fonte: Freepik

Para Saber Mais...

Confira a animação do fenômeno de dispersão (ou decomposição) da luz

AULA 26

Essa é uma animação esquemática de um feixe de luz contínuo sendo dispersado por um prisma. O feixe de luz branca (policromático) representa muitos comprimentos de onda da luz visível, dos quais sete são mostrados conforme viajam através do vácuo com velocidades iguais. O prisma faz com que a luz desacelere, desviando seu caminho pelo processo de refração. Este efeito ocorre mais fortemente nos comprimentos de onda mais curtos (cor violeta) do que nos comprimentos de onda mais longos (cor vermelha). Ao sair do prisma, cada componente retorna à mesma velocidade original e é refratado novamente.

No vácuo (mostrado em preto), a luz de qualquer comprimento de onda viaja em uma velocidade constante, porém quando ela passa para um meio mais denso, como o cristal, o vidro ou a água, sua velocidade se altera: comprimentos de onda mais curtos, como o violeta, passam mais devagar do que a luz de comprimentos de onda mais longos, como o vermelho.

A luz branca (ou policromática), representada aqui pelo feixe branco, é, na verdade, composta de luz de várias frequências (cores) viajando juntas. Essas frequências de luz visível são, como veremos na tabela abaixo, parte do **espectro visível** - e apenas uma pequena parte de todo o espectro eletromagnético.

Conforme a luz branca entra em um meio (neste caso, o prisma), cada um de seus comprimentos de onda que a compõem viajará em uma velocidade diferente no novo meio, e essa mudança na velocidade é o que determinará a direção da luz, que é o fenômeno que chamamos de **refração**.

Como a luz de diferentes comprimentos de onda mudará de direção em ângulos diferentes, tem-se aqui a divisão da luz branca em suas cores espectrais compostas,

representadas aqui por ondas coloridas, que chamamos de **dispersão**.

Uma vez que as frequências básicas são separadas nesta animação, podemos ver facilmente a diferença em suas velocidades. O vermelho, com comprimento de onda longo, passa quase sem qualquer alteração, enquanto o violeta, com comprimento de onda mais curto, é deixado para trás por todas as outras cores. No entanto, essa diferença de velocidade não se mantém no vácuo, e isso pode ser visto em como toda a luz que sai do prisma viajará novamente na velocidade constante da luz no vácuo.

Toda a gama de cores do arco-íris é simplificada em sete cores (vermelho, alaranjado, amarelo, verde, azul, anil e violeta) por serem as mais perceptíveis pela nossa visão. Como podemos ver no quadro 1, cada cor possui uma frequência e comprimento de onda, as quais nosso cérebro "interpreta" por meio do olho - por isso que a percepção do arco-íris varia conforme o espectador e seu ângulo de observação.

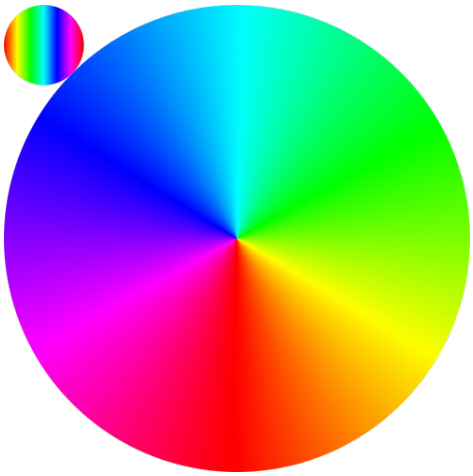
Quadro 1 - Frequência e comprimento das cores do arco-íris mais perceptíveis pela visão

Cor	Comprimento de onda	Frequência
Vermelho	6222 – 7800	4,82 – 3,84
Alaranjado	5970 – 6220	5,03 – 4,82
Amarelo	5770 – 5970	5,20 – 5,03
Verde	4920 – 5770	6,10 – 5,20
Azul	4550 – 4920	6,59 – 6,10
Anil	4500 – 4550	6,65 – 6,59
Violeta	3900 – 4500	7,69 – 6,65

AULA 26

A programação do LED RGB possibilita a determinação de cada cor e intensidade do brilho, definindo, assim, um espectro amplo de tonalidades. São 16.777.216 de cores que podem ser emitidas através das combinações de cada cor RGB, as quais utilizam, na programação, valores de 0 a 255. Dizemos que o sistema RGB é um sistema aditivo de cores por esta possibilidade – gerar milhões de cores a partir da soma das cores primárias vermelho, verde e azul, conforme representado na figura 3.

Figura 3 - Espectro de luz visível a partir do RGB

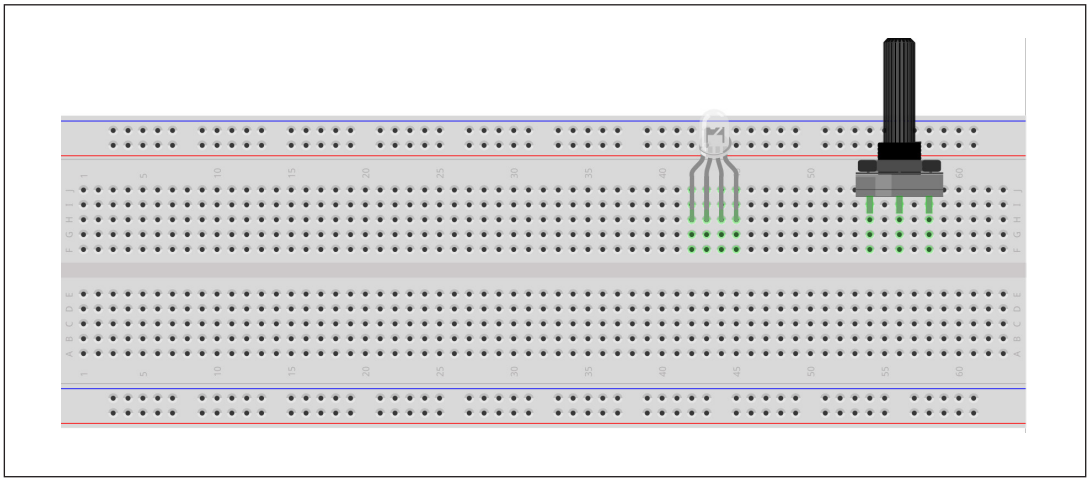


Neste projeto Arco-Íris, programaremos o LED RGB para simularmos as cores do arco-íris. Para controle do LED RGB, utilizaremos as portas PWM, portas digitais que, no Arduino, simulam saídas analógicas e possibilitam, através da função **analogWrite**, o controle de valores variados (vimos sobre este tema na **Aula 18 – Portas PWM**), e a transição entre as cores do arco-íris será realizada pelo Potenciômetro, componente eletrônico 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.

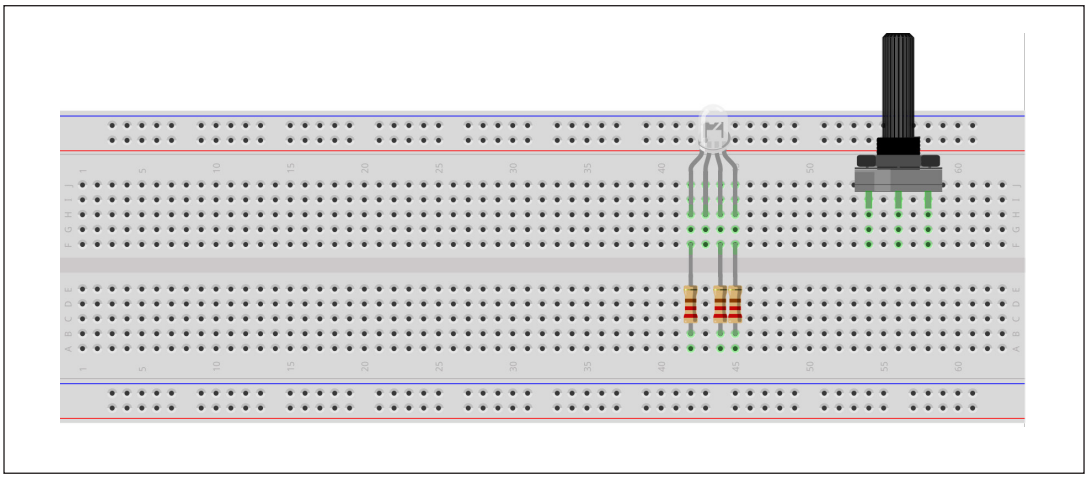
Figura 4 - Inserção do LED RGB e do Potenciômetro na Protoboard



Fonte: Fritzing

Insira os três resistores, conectando um de seus terminais aos terminais positivos do LED RGB, e o outro terminal na parte inferior da Protoboard, conforme mostra a figura 5.

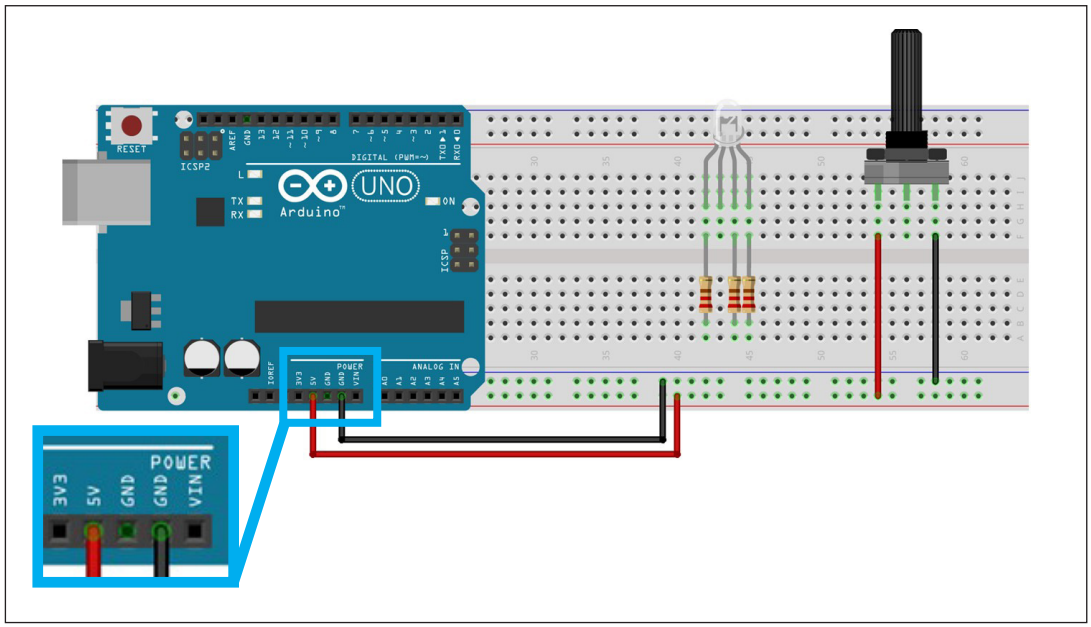
Figura 5 – Inserindo os Resistores na Protoboard



Fonte: Fritzing

Vamos agora alimentar a placa Protoboard, conectando dois jumpers às portas GND e 5V do Arduino até as duas linhas inferiores da Protoboard (azul e vermelha, respectivamente). Para o Potenciômetro, conectaremos dois jumpers entre seus terminais extremos e as duas linhas inferiores da Protoboard (vermelha e azul), como mostra a figura 06.

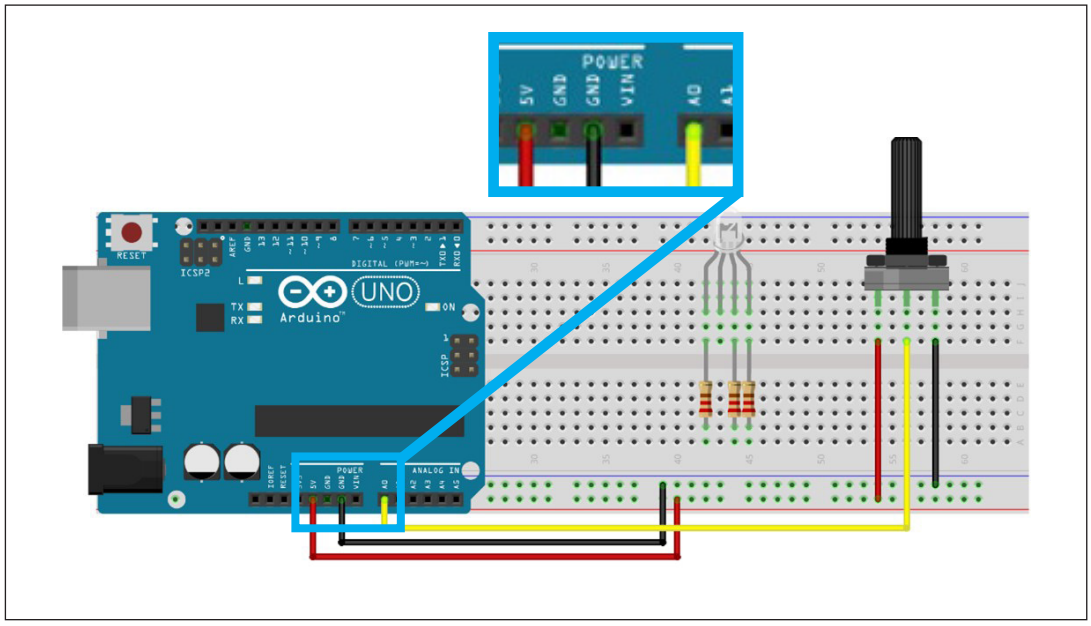
Figura 6 – Alimentando a placa protoboard, LED RGB e o potenciômetro



Fonte: Fritzing

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

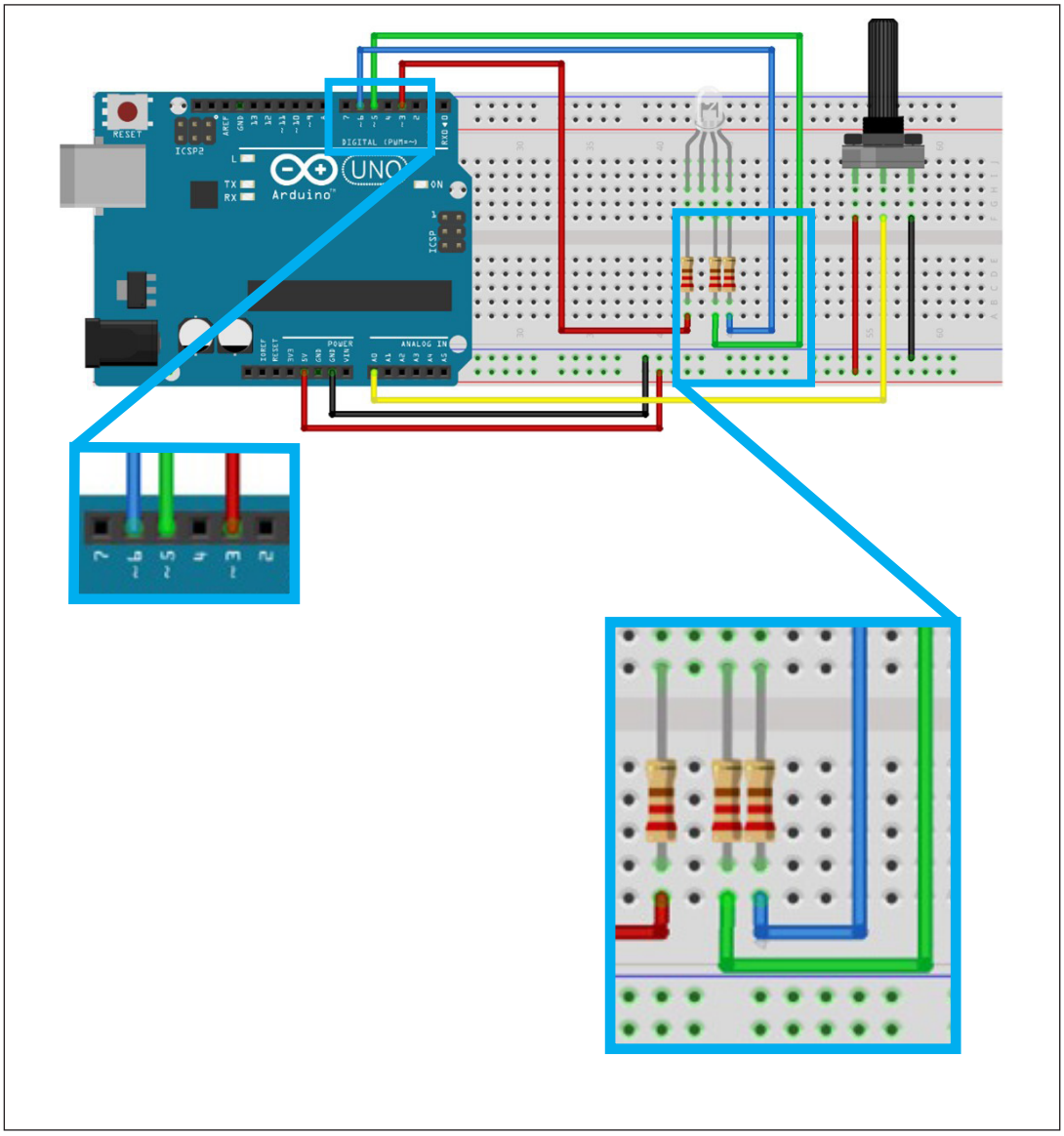
Figura 7 – Interligando o Potenciômetro à porta analógica da placa Arduino



Fonte: Fritzing

Utilizando mais três Jumpers, conecte o LED RGB às portas ou pinos digitais com recurso PWM 3, 5 e 6 da placa Arduino, conforme a ordem representada pela figura 08.

Figura 08 – Interligando os LEDs às portas digitais do Arduino



Fonte: Fritzing

Por fim, utilizando mais um Jumper, conecte o terminal comum do LED RGB (aquele sem Resistor) 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 de ânodo comum à linha vermelha da Protoboard

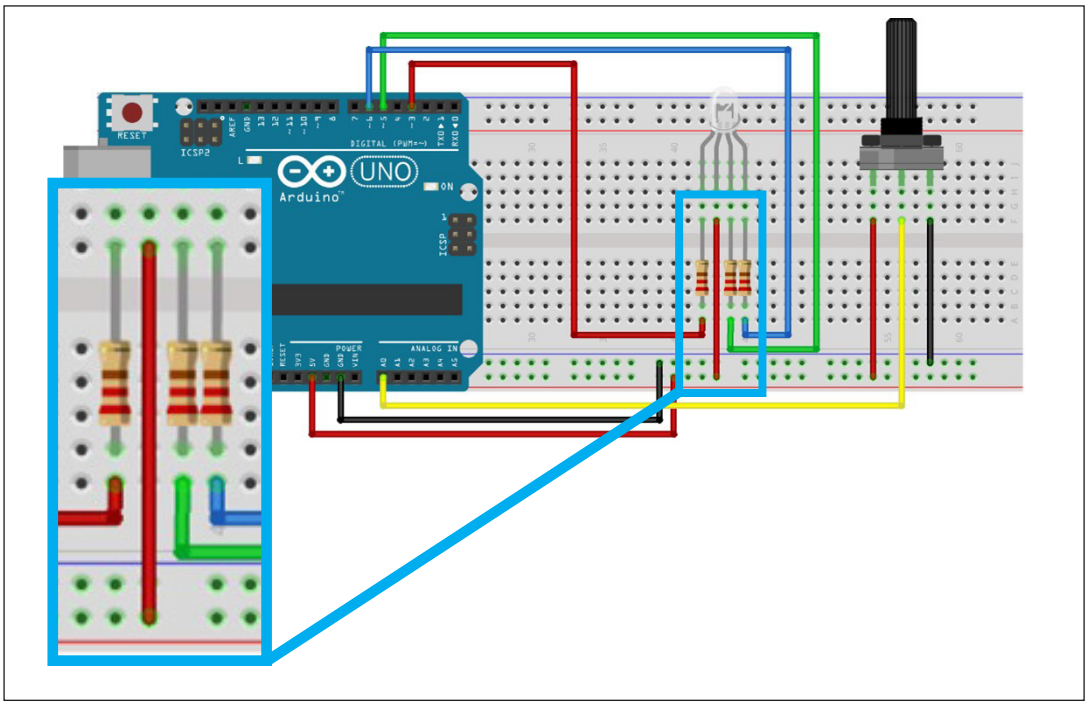
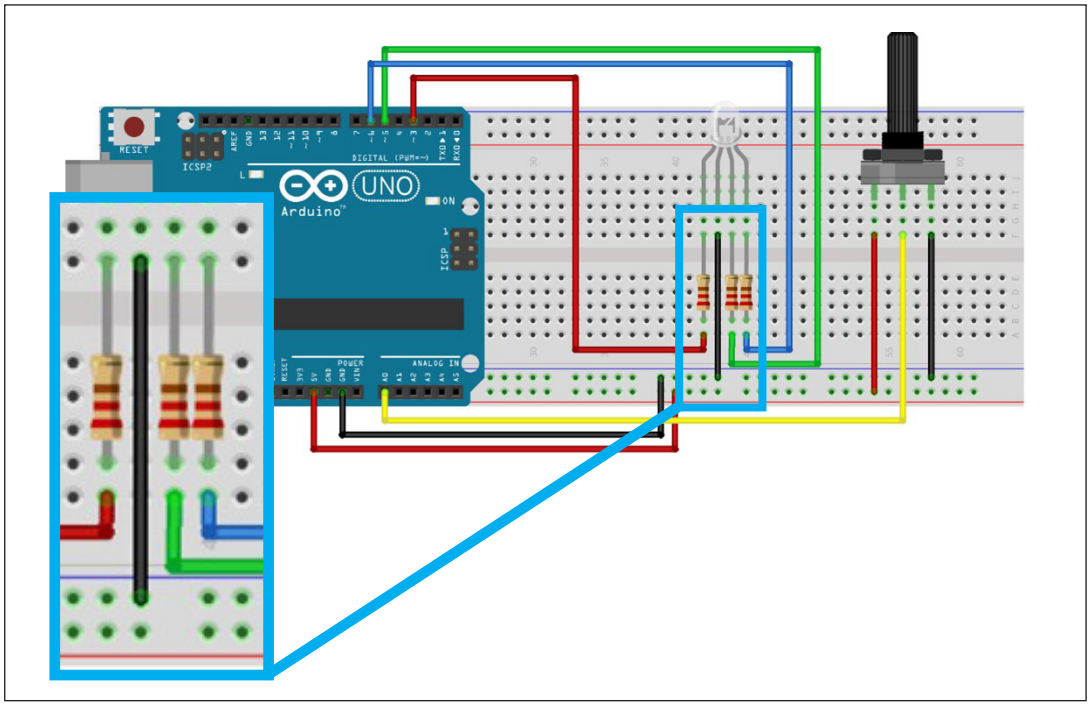


Figura 10 – Interligando o LED RGB de cátodo comum à linha azul da Protoboard



Agora, vamos programar!

Com os componentes eletrônicos montados, vamos programar, por codificação e por blocos, o projeto Arco-Íris.

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 2:

Quadro 02 – Código-fonte da programação na linguagem do Arduino (Wiring)

```
/*************************************************************/
/* Aula 26 - Arco Íris                                       */
/* Programação: Gerando no LED RGB, por meio das portas      */
/* digitais PWM, o espectro do arco-íris, controlado         */
/* pelo sensor potenciômetro.                                */
/*                                                           */
/*                    ATENÇÃO!                               */
/* Nas linhas 14 ou 15 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";

/* Variável que armazena os dados do potenciômetro.          */
int Pot = 0;
/* Define a porta analógica A0 como pino do potenciômetro.   */
int Pin_pot = A0;
/* Define a porta digital 3 (PWM) no controle do LED Vermelho. */
int Led_R = 3;
/* Define a porta digital 5 (PWM) no controle do LED Verde.  */
int Led_G = 5;
/* Define a porta digital 6 (PWM) no controle do LED Azul.   */
int Led_B = 6;
```

```
void setup() {
  /* Define o pino do LED Vermelho (Red) como SAÍDA.          */
  pinMode(Led_R, OUTPUT);
  /* Define o pino do LED Verde (Green) como SAÍDA.           */
  pinMode(Led_G, OUTPUT);
  /* Define o pino do LED Azul (Blue) como SAÍDA.             */
  pinMode(Led_B, OUTPUT);
}

void loop() {
  /* Mapeia o valor lido no potenciômetro (0 à 1023) para     */
  /* sete valores (1 à 7).                                    */
  Pot = map(analogRead(Pin_pot), 0, 1023, 1, 7);
/* Se o modelo do LED é cátodo comum, faça...                 */
  if (modelo[0] == 'C') {
    /* Se o valor de Pot for igual a 1, acenda VERMELHO       */
    if (Pot == 1) {
      /* Enviando o sinal PWM 255 no LED Vermelho.            */
      analogWrite(Led_R, 255);
      /* Enviando o sinal PWM 0 no LED Verde.                 */
      analogWrite(Led_G, 0);
      /* Enviando o sinal PWM 0 no LED Azul.                  */
      analogWrite(Led_B, 0);
    }
    /* Se o valor de Pot for igual a 2, acenda LARANJA        */
    if (Pot == 2) {
      /* Enviando o sinal PWM 255 no LED Vermelho.            */
      analogWrite(Led_R, 255);
      /* Enviando o sinal PWM 100 no LED Verde.               */
      analogWrite(Led_G, 100);
      /* Enviando o sinal PWM 0 no LED Azul.                  */
      analogWrite(Led_B, 0);
    }
    /* Se o valor de Pot for igual a 3, acenda AMARELO        */
    if (Pot == 3) {
      /* Enviando o sinal PWM 255 no LED Vermelho.            */
      analogWrite(Led_R, 255);
      /* Enviando o sinal PWM 255 no LED Verde.               */
      analogWrite(Led_G, 255);
      /* Enviando o sinal PWM 0 no LED Azul.                  */
      analogWrite(Led_B, 0);
    }
```

```
    /* Se o valor de Pot for igual a 4, acenda VERDE          */
    if (Pot == 4) {
      /* Enviando o sinal PWM 0 no LED Vermelho.             */
      analogWrite(Led_R, 0);
      /* Enviando o sinal PWM 255 no LED Verde.              */
      analogWrite(Led_G, 255);
      /* Enviando o sinal PWM 0 no LED Azul.                 */
      analogWrite(Led_B, 0);
    }
    /* Se o valor de Pot for igual a 5, acenda AZUL           */
    if (Pot == 5) {
      /* Enviando o sinal PWM 0 no LED Vermelho.             */
      analogWrite(Led_R, 0);
      /* Enviando o sinal PWM 0 no LED Verde.                */
      analogWrite(Led_G, 0);
      /* Enviando o sinal PWM 255 no LED Azul.               */
      analogWrite(Led_B, 255);
    }
    /* Se o valor de Pot for igual a 6, acenda ANIL           */
    if (Pot == 6) {
      /* Enviando o sinal PWM 75 no LED Vermelho.            */
      analogWrite(Led_R, 75);
      /* Enviando o sinal PWM 0 no LED Verde.                */
      analogWrite(Led_G, 0);
      /* Enviando o sinal PWM 130 no LED Azul.               */
      analogWrite(Led_B, 130);
    }
    /* Se o valor de Pot for igual a 7, acenda VIOLETA        */
    if (Pot == 7) {
      /* Enviando o sinal PWM 255 no LED Vermelho.           */
      analogWrite(Led_R, 255);
      /* Enviando o sinal PWM 0 no LED Verde.                */
      analogWrite(Led_G, 0);
      /* Enviando o sinal PWM 255 no LED Azul.               */
      analogWrite(Led_B, 255);
    }
  } else { /* Senão ... (LED com ânodo comum, faça...)       */
    /* Se o valor de Pot for igual a 1, acenda VERMELHO      */
    if (Pot == 1) {
      /* Enviando o sinal PWM 0 no LED Vermelho.             */
      analogWrite(Led_R, 0);
      /* Enviando o sinal PWM 255 no LED Verde.              */
      analogWrite(Led_G, 255);
      /* Enviando o sinal PWM 255 no LED Azul.               */
      analogWrite(Led_B, 255);
    }
```

```
  /* Se o valor de Pot for igual a 2, acenda LARANJA       */
  if (Pot == 2) {
    /* Enviando o sinal PWM 0 no LED Vermelho.            */
    analogWrite(Led_R, 0);
    /* Enviando o sinal PWM 155 no LED Verde.             */
    analogWrite(Led_G, 155);
    /* Enviando o sinal PWM 255 no LED Azul.              */
    analogWrite(Led_B, 255);
  }
  /* Se o valor de Pot for igual a 3, acenda AMARELO       */
  if (Pot == 3) {
    /* Enviando o sinal PWM 0 no LED Vermelho.            */
    analogWrite(Led_R, 0);
    /* Enviando o sinal PWM 0 no LED Verde.               */
    analogWrite(Led_G, 0);
    /* Enviando o sinal PWM 255 no LED Azul.              */
    analogWrite(Led_B, 255);
  }
  /* Se o valor de Pot for igual a 4, acenda VERDE         */
  if (Pot == 4) {
    /* Enviando o sinal PWM 255 no LED Vermelho.          */
    analogWrite(Led_R, 255);
    /* Enviando o sinal PWM 0 no LED Verde.               */
    analogWrite(Led_G, 0);
    /* Enviando o sinal PWM 255 no LED Azul.              */
    analogWrite(Led_B, 255);
  }
  /* Se o valor de Pot for igual a 5, acenda AZUL          */
  if (Pot == 5) {
    /* Enviando o sinal PWM 255 no LED Vermelho.          */
    analogWrite(Led_R, 255);
    /* Enviando o sinal PWM 255 no LED Verde.             */
    analogWrite(Led_G, 255);
    /* Enviando o sinal PWM 0 no LED Azul.                */
    analogWrite(Led_B, 0);
  }
```

```
      /* Se o valor de Pot for igual a 6, acenda ANIL          */
      if (Pot == 6) {
        /* Enviando o sinal PWM 180 no LED Vermelho.          */
        analogWrite(Led_R, 180);
        /* Enviando o sinal PWM 255 no LED Verde.             */
        analogWrite(Led_G, 255);
        /* Enviando o sinal PWM 125 no LED Azul.              */
        analogWrite(Led_B, 125);
      }
      /* Se o valor de Pot for igual a 7, acenda VIOLETA       */
      if (Pot == 7) {
        /* Enviando o sinal PWM 0 no LED Vermelho.            */
        analogWrite(Led_R, 0);
        /* Enviando o sinal PWM 255 no LED Verde.             */
        analogWrite(Led_G, 255);
        /* Enviando o sinal PWM 0 no LED Azul.                */
        analogWrite(Led_B, 0);
      }
    }
  }
```

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, ao girar o eixo do Potenciômetro, o LED alternará sua cor, passando pelas cores do arco-íris.

i. 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.

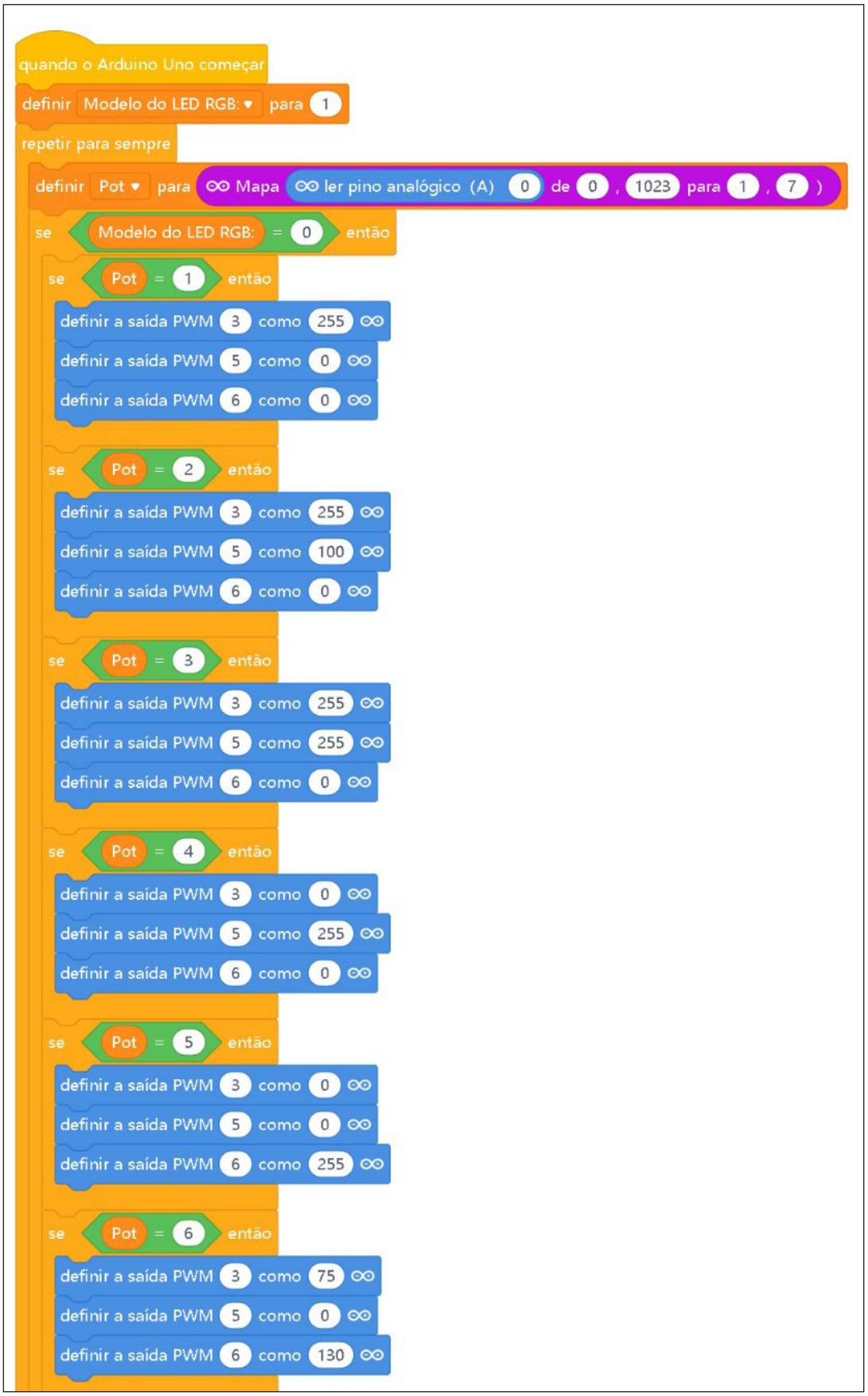
AULA 26

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**).

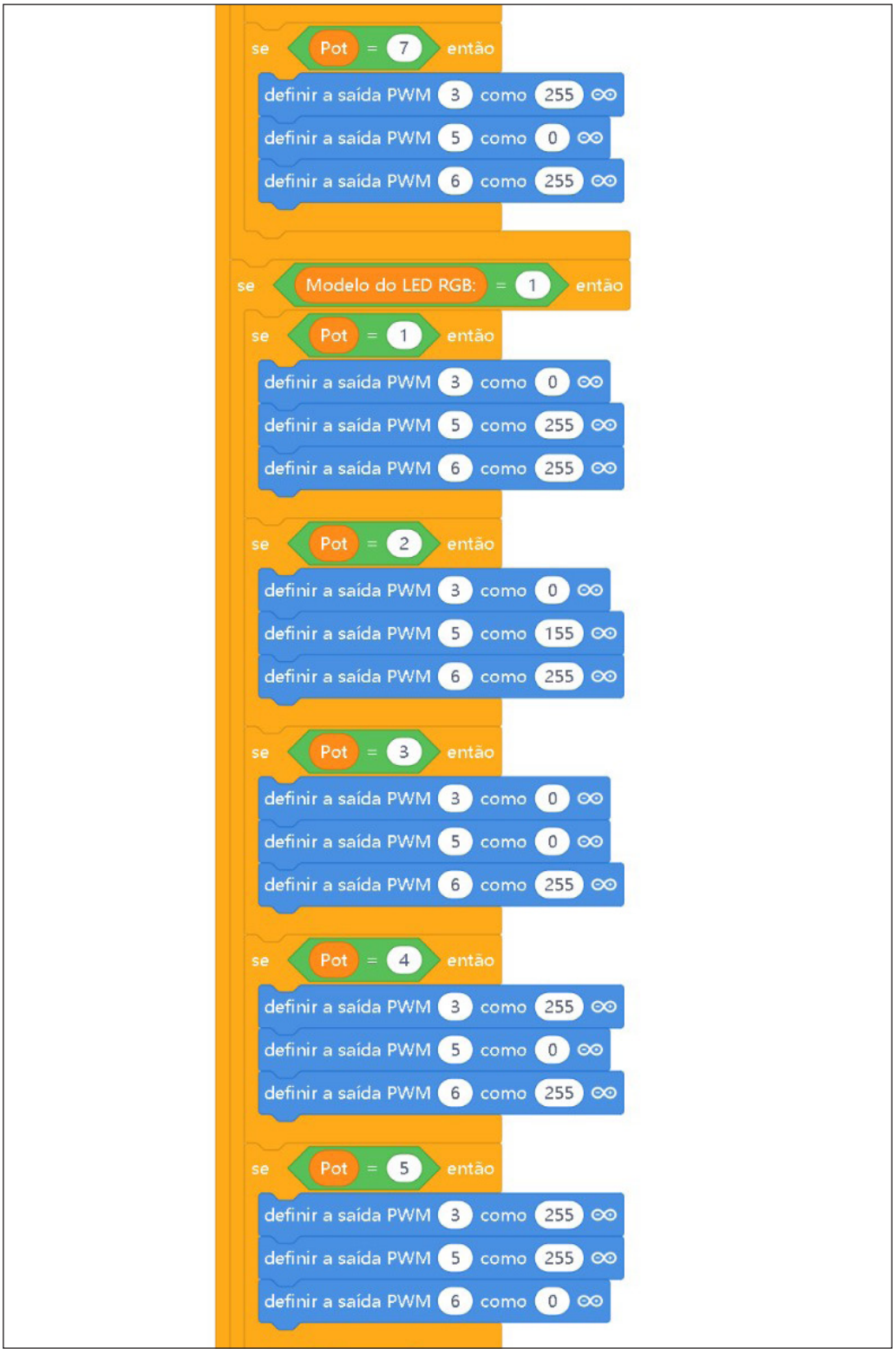
Monte os blocos, arrastando e soltando, de acordo com a programação de funcionamento para alternar as cores do arco-íris através do potenciômetro, como mostra a figura 9.

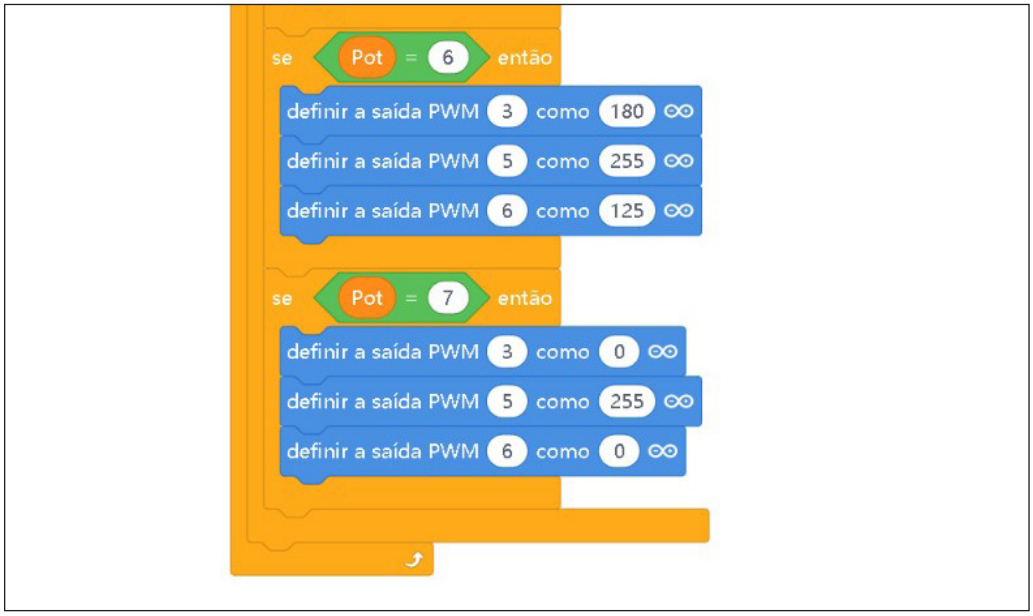
Atenção! No segundo bloco, informe o modelo do LED RGB utilizado no seu protótipo (0 para cátodo comum ou 1 para ânodo comum).

Figura 11 - Programação em blocos para controle do LED RGB



AULA 26


se Pot = 7 então
definir a saída PWM 3 como 255
definir a saída PWM 5 como 0
definir a saída PWM 6 como 255
se Modelo do LED RGB: = 1 então
se Pot = 1 então
definir a saída PWM 3 como 0
definir a saída PWM 5 como 255
definir a saída PWM 6 como 255
se Pot = 2 então
definir a saída PWM 3 como 0
definir a saída PWM 5 como 155
definir a saída PWM 6 como 255
se Pot = 3 então
definir a saída PWM 3 como 0
definir a saída PWM 5 como 0
definir a saída PWM 6 como 255
se Pot = 4 então
definir a saída PWM 3 como 255
definir a saída PWM 5 como 0
definir a saída PWM 6 como 255
se Pot = 5 então
definir a saída PWM 3 como 255
definir a saída PWM 5 como 255
definir a saída PWM 6 como 0



Fonte: mBlock, 2022.

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, ou seja, as cores do arco-íris podem ser alternadas através do giro do eixo do Potenciômetro.

Desafios:

I. Que tal criar novas cores? Utilizando a função **analogWrite** com o parâmetro que varia de 0 a 255, experimente produzir novas cores.

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. O projeto não funcionar, 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 os requisitos para controlar o acendimento do LED RGB de modo a alternar as cores com o giro do eixo do Potenciômetro, passando pelo espectro de um arco-íris.

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/a26robotica

Acesse, também, pelo QRCode:

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.