

Learning Unit	
Subject	Programação
Title	Programação em C++ de Arduino
Author	Eduardo Leite
School	FORAVE – Associação para a Educação Tecnológica do Vale do Ave
Description of the unit	Esta unidade tem como objetivo aprender as bases de programação em C++ usando Arduino.
Contents	Programação em linguagem C++ <ul style="list-style-type: none"> - Algoritmia - Usar o IDE da Arduino; - Led's e Resistências; - Servomotor; - Buzzer; - LCD.
Learning Outcomes / Skills	Aprender a criar, planejar, resolver problemas interligando artefactos tangíveis como o Arduino, construindo algo com uma finalidade, proporcionando também a articulação com conteúdos das diferentes áreas do saber. Aprender técnicas de programação do controlador Arduino. Desenvolver programação específica para utilização dos sensores, atuadores e motores associados ao Arduino. Aprender a fazer a ligação dos componentes de comunicação ao Arduino.
Target students/class	Ensino secundário (15-17 anos)
Prerequisites	Os alunos devem ser capazes de: <ul style="list-style-type: none"> - Realizar os esquemáticos eletrónicos - Realizar fluxogramas - Realizar a montagem de equipamentos eletrónicos - Identificar as diversas expressões, comandos e operadores em linguagem de C++ - Utilizar os ambientes de simulação Tinkercad
Time expected	5 horas
Interdisciplinary links	TIC
Methodology	<ul style="list-style-type: none"> - Exposição dos conteúdos; - Resolução de exercícios e problemas; - Resolução de fichas de trabalho e trabalho em grupos.
Human Resources (internal and/or external)	Professor de Matemática

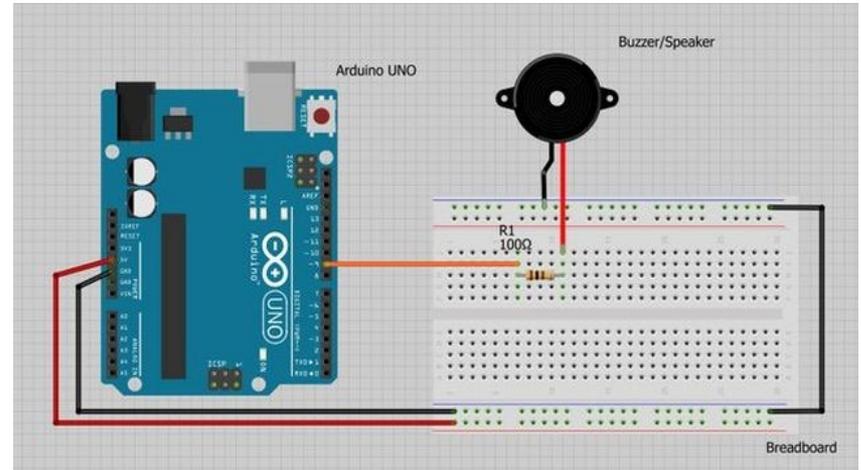


Learning Unit	
Resources	<ul style="list-style-type: none"> - Fichas de trabalho; - Arduíno - Leds, resistências, LCDs, servomotor. - Computadores com IDE Arduíno
Lesson Plan	<p><u>1.ª Aula</u></p> <p>Sumário:</p> <ul style="list-style-type: none"> - Conceitos de eletrónica básica <p>Introdução dos conceitos teóricos relacionados com o tema da lição. Depois de introduzir os conceitos, e analisar o exemplo resolvido, realizar a montagem de um Led que é acionado por um botão. Esta montagem será realizada em cenária de simulação e em cenário real de laboratório. Esclarecimento de dúvidas.</p> <p style="text-align: center;"><u>2.ª Aula:</u></p> <p>Sumário:</p> <ul style="list-style-type: none"> - Algoritmia. - Fluxogramas. - Pseudocódigo. <p>Introdução dos conceitos teóricos relacionados com o tema da lição. Depois de introduzir os conceitos, e analisar o exemplo resolvido, realizar o programa “Hello World” em linguagem C++.</p> <p>Nesta lição será também solicitado ao formando que execute a montagem do LED na placa de Arduíno e realize a sua programação na linguagem C++.</p> <p>Esclarecimento de dúvidas</p> <p style="text-align: center;"><u>3.ª Aula:</u></p> <p>Sumário:</p> <ul style="list-style-type: none"> - Programação em C++ (expressões, comandos e operadores) <p>Introdução dos conceitos teóricos relacionados com o tema da lição. Depois de introduzir os conceitos, e analisar o exemplo resolvido, realizar os exercícios da ficha de trabalho nº1, em pares.</p> <p>Esclarecimento de dúvidas.</p> <p style="text-align: center;"><u>4.ª Aula:</u></p> <p>Sumário:</p> <ul style="list-style-type: none"> - Motor passo-a-passo <p>Introdução dos conceitos teóricos relacionados com o tema da lição. Depois de introduzir os conceitos, e analisar o exemplo resolvido, realizar os exercícios da ficha de trabalho nº2, em pares.</p> <p>Esclarecimento de dúvidas.</p> <p style="text-align: center;"><u>5.ª Aula:</u></p>

Learning Unit	
	<p>Sumário:</p> <ul style="list-style-type: none"> - Buzzer - LCD <p>O professor introduz os conceitos teóricos relacionados com expressões, comandos e operadores. Depois de introduzir os conceitos, e analisar o exemplo resolvido, o professor propõe a resolução da ficha de trabalho nº3, em pares. Esclarecimento de dúvidas.</p>
21st Century Skills	<p>Pensamento crítico: os alunos serão capazes de analisar dados durante experiências práticas e comunicar as suas conclusões.</p> <p>Colaboração: os alunos serão capazes de colaborar nos seus grupos e com os restantes grupos, ajudarem-se mutuamente a compreender os conteúdos e as atividades experimentais.</p> <p>Comunicação: Os alunos devem ser capazes de partilhar conclusões e dúvidas com os seus colegas e professores.</p> <p>Literacia tecnológica: os alunos serão capazes de utilizar diferentes ferramentas tecnológicas para realizar as tarefas.</p>
Assessment	<p>Avaliação Formativa:</p> <ul style="list-style-type: none"> - Assiduidade; - Pontualidade; - Comportamento: - Atenção e participação na aula; - Realização de fichas de trabalho (Grelhas de observação direta).
Remarks	--



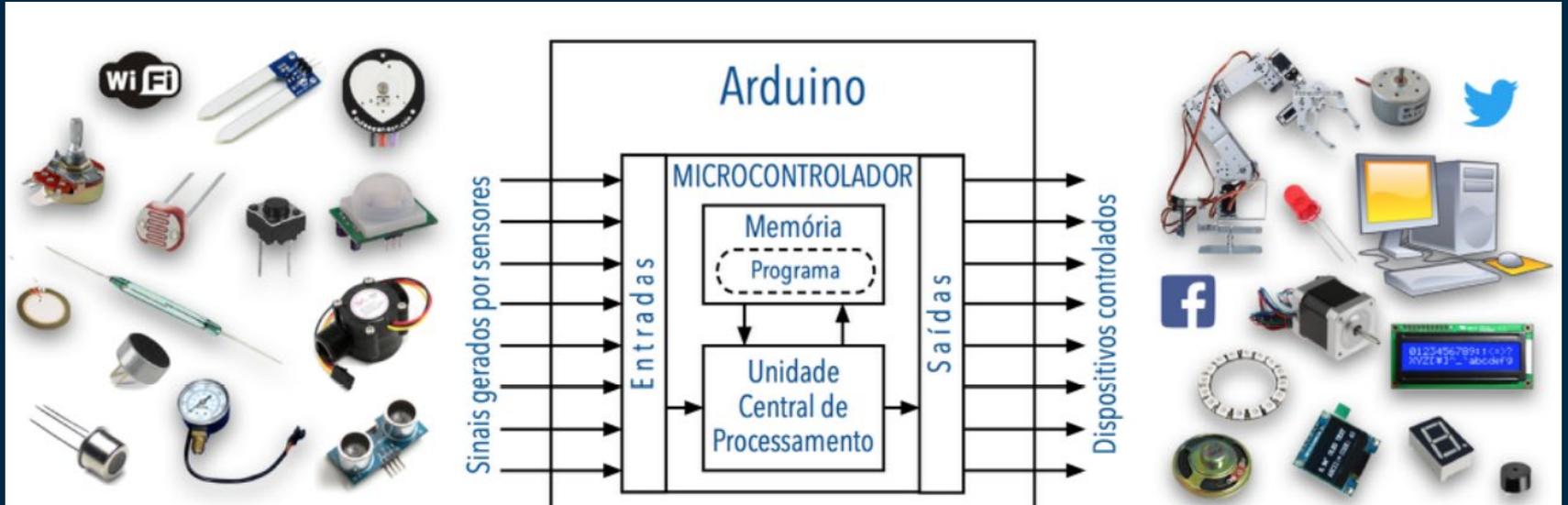
Programação



The background is a dark blue gradient. It features several vertical white lines of varying lengths. Scattered throughout are small squares in light blue, orange, and pink. The text is centered and reads "PROGRAMAÇÃO ARDUÍNO DO BUZZER".

PROGRAMAÇÃO ARDUÍNO DO BUZZER

ARDUÍNO

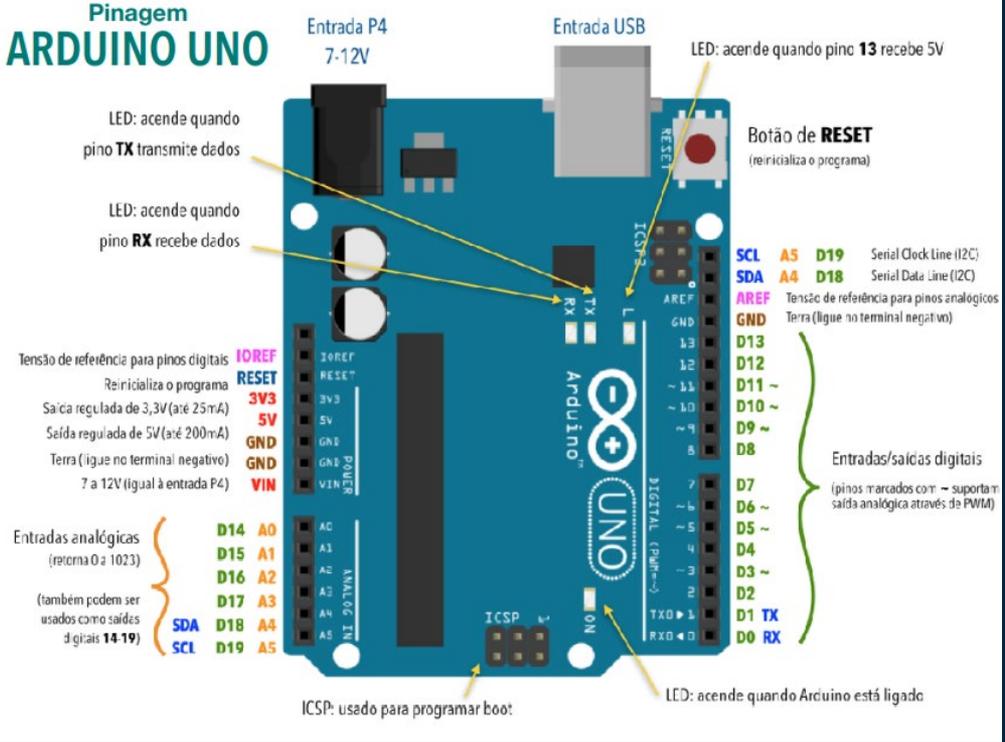


CONCEITO



- Linguagem de alto nível (baseada na Linguagem C)
- Programa é compilado (traduzido para a linguagem máquina do Arduino)
- A transferência é feita através da saída USB do computador que também fornece energia para a placa.

Pinagem ARDUINO UNO



LED: acende quando pino TX transmite dados

LED: acende quando pino RX recebe dados

Tensão de referência para pinos digitais
Reinicializa o programa

3V3 Saída regulada de 3,3V (até 25mA)

5V Saída regulada de 5V (até 200mA)

Terra (ligue no terminal negativo) 7 a 12V (igual à entrada P4)

Entradas analógicas (retorna 0 a 1023)

(também podem ser usados como saídas digitais 14-19)

- IOREF
- RESET
- 3V3
- 5V
- GND
- GND
- VIN
- AC
- A0
- A1
- A2
- A3
- A4
- A5
- SDA
- SCL
- D14 A0
- D15 A1
- D16 A2
- D17 A3
- D18 A4
- D19 A5

- SCL A5 D19 Serial Clock Line (I2C)
- SDA A4 D18 Serial Data Line (I2C)
- AREF Tensão de referência para pinos analógicos Terra (ligue no terminal negativo)
- GND
- D13
- D12
- D11 ~
- D10 ~
- D9 ~
- D8

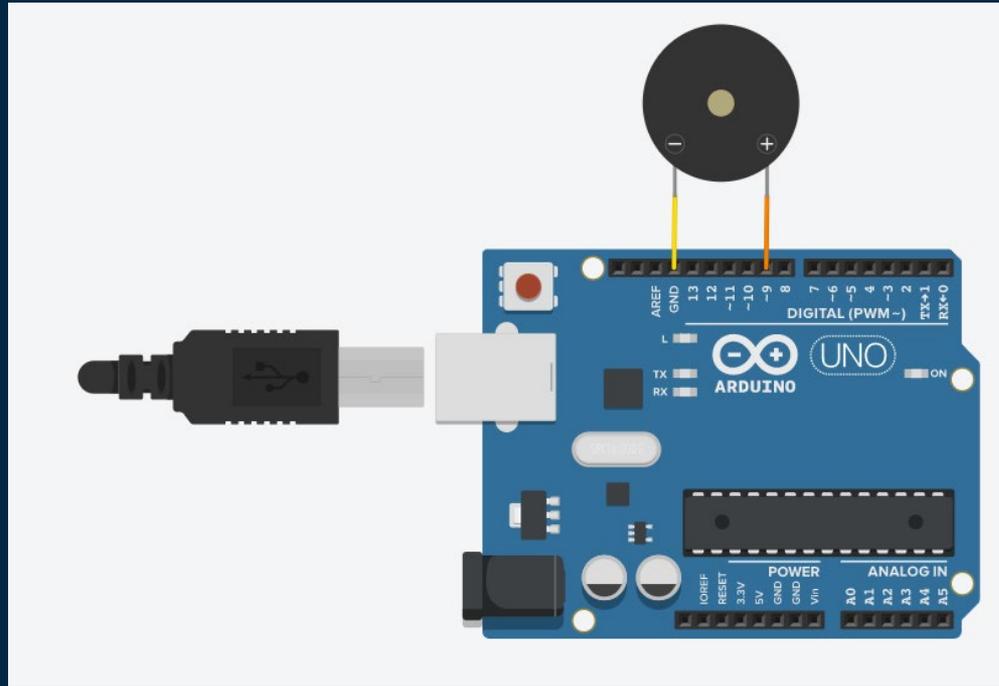
Entradas/saídas digitais

(pinos marcados com ~ suportam saída analógica através de PWM)

ICSP: usado para programar boot

LED: acende quando Arduino está ligado

Diagrama elétrico



Vamos programar!!!

www.tinkercad.com



Programação

```
int pinBuzzer = 8; // o pino onde o buzzer está conectado
```



Programação

```
void setup() {  
  pinMode(pinBuzzer, OUTPUT); // configura o pino do buzzer  
  como saída  
}
```



Programação

```
void loop() {  
  tone(pinBuzzer, 1000); // faz o buzzer tocar em uma  
  frequência de 1000 Hz  
  delay(1000); // aguarda 1 segundo
```



Programação

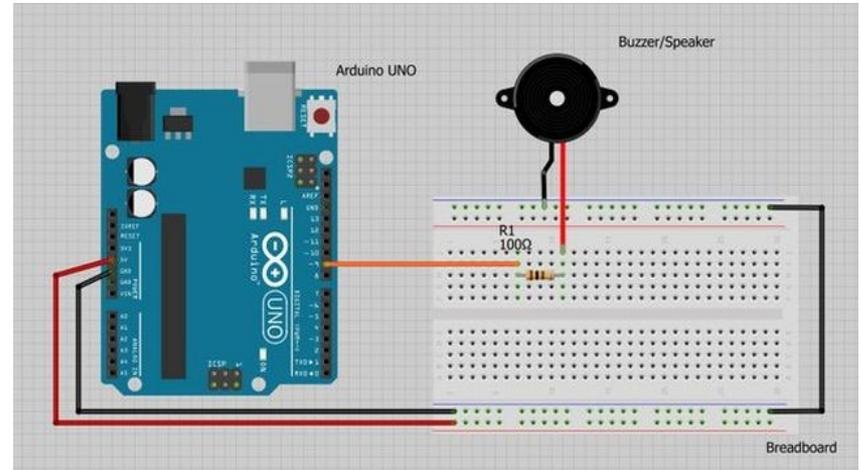
```
noTone(pinBuzzer); // para o som do buzzer  
delay(1000); // aguarda mais 1 segundo  
}
```



Programação

```
void loop()  
{  
  // Aciona o buzzer na frequencia relativa ao Dó em Hz  
  tone(buzzer,261);  
  // Espera um tempo para Desativar  
  delay(200);  
  //Desativa o buzzer  
  noTone(buzzer);  
  // Aciona o buzzer na frequencia relativa ao Ré em Hz  
  tone(buzzer,293);  
  delay(200);  
  noTone(buzzer);  
  // Aciona o buzzer na frequencia relativa ao Mi em Hz  
  tone(buzzer,329);  
  delay(200);  
  noTone(buzzer);  
  // Aciona o buzzer na frequencia relativa ao Fá em Hz  
  tone(buzzer,349);  
  delay(200);  
  noTone(buzzer);  
  // Aciona o buzzer na frequencia relativa ao Sol em Hz  
  tone(buzzer,392);  
  delay(200);  
  noTone(buzzer);  
}
```

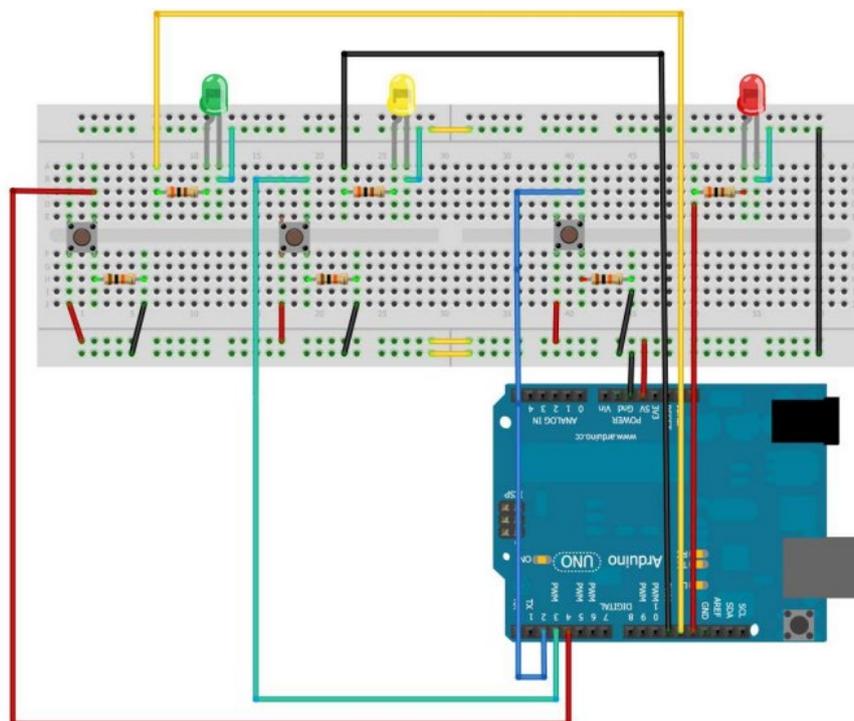
Programação



Learning Unit	Programação em C++ de Arduino
Author	Eduardo Leite
School	FORAVE –Associação para a Educação Tecnológica do Vale do Ave
Date	

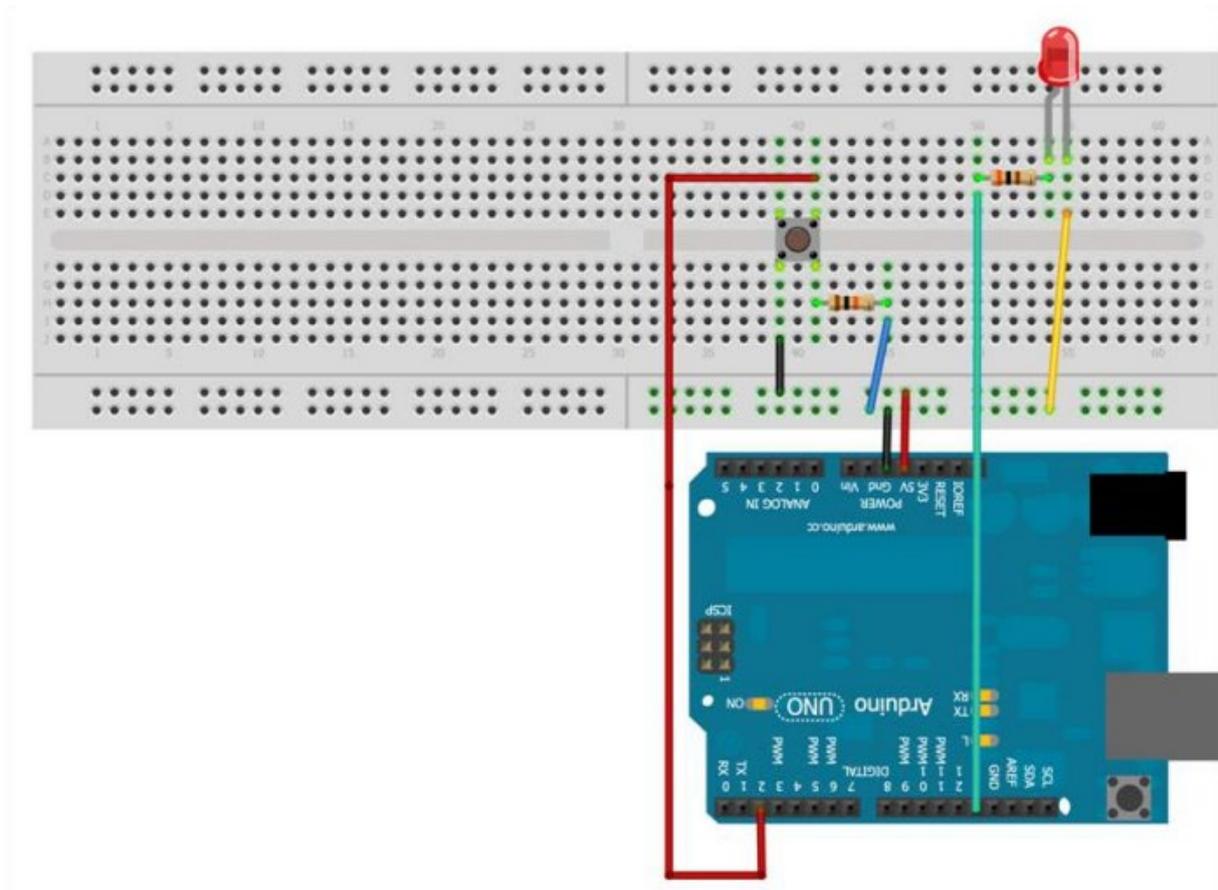
Nota: Aceda à plataforma Tinkercad <https://www.tinkercad.com> e crie um projeto novo para cada exercício.

1. Execute o circuito na plataforma acima mencionada, da maneira que se segue abaixo. Através dos botões que se encontram no exercício execute um programa de forma que cada botão acenda e apague um LED.



Worksheet No. 1

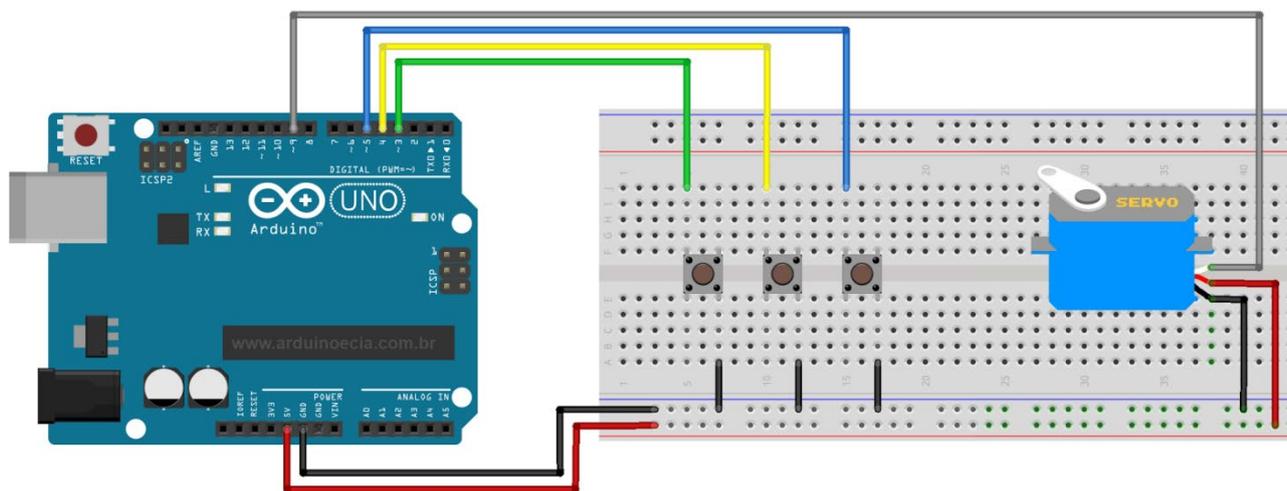
2. Execute o circuito na plataforma acima mencionada, da maneira que se segue abaixo. Note que o circuito será muito semelhante ao primeiro, no entanto, será utilizada a estrutura “while” em vez do “if”.



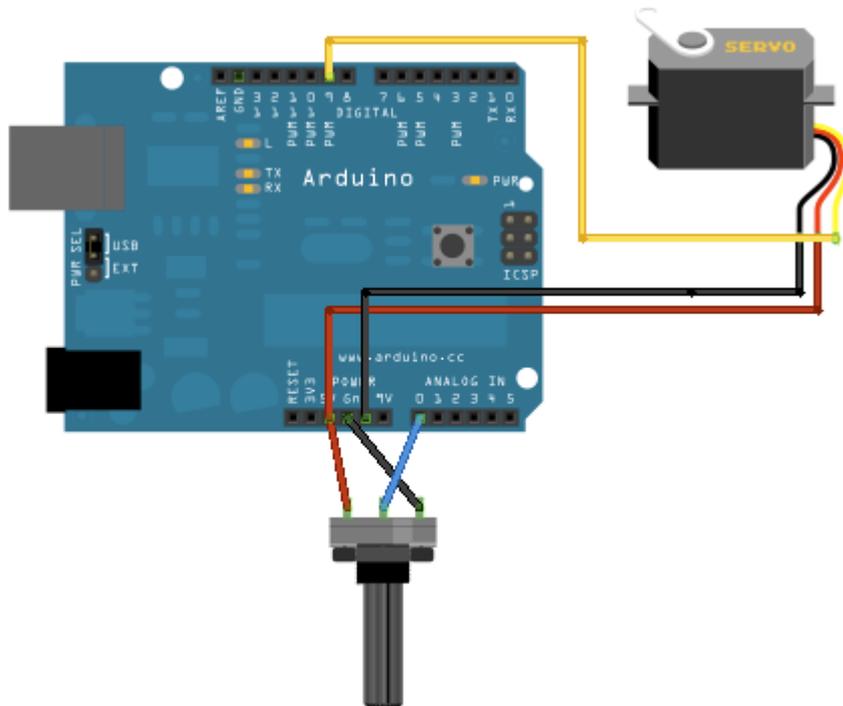
Learning Unit	Programação em C++ de Arduino
Author	Eduardo Leite
School	FORAVE –Associação para a Educação Tecnológica do Vale do Ave
Date	

Nota: Acesse a plataforma Tinkercad <https://www.tinkercad.com> e crie um projeto novo para cada exercício.

1. Execute o circuito na plataforma acima mencionada, da maneira que se segue abaixo. Através dos botões que se encontram no exercício execute um programa de forma que cada botão posicione o servo em três posições diferentes.



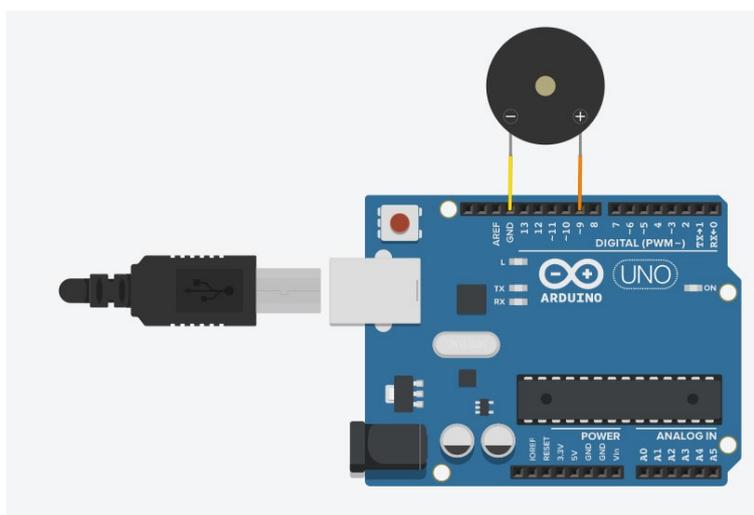
2. Execute o circuito na plataforma acima mencionada, da maneira que se segue abaixo. Através do potenciômetro que se encontra no exercício execute um programa de forma que a variação do potenciômetro varie a posição do servo em 180°.



Learning Unit	Programação em C++ de Arduino
Author	Eduardo Leite
School	FORAVE –Associação para a Educação Tecnológica do Vale do Ave
Date	

Nota: Aceda à plataforma Tinkercad <https://www.tinkercad.com> e crie um projeto novo para cada exercício.

1. Execute o circuito na plataforma acima mencionada, da maneira que se segue abaixo. Através do buzzer que se encontram no exercício execute um programa de forma a reproduzir um som.



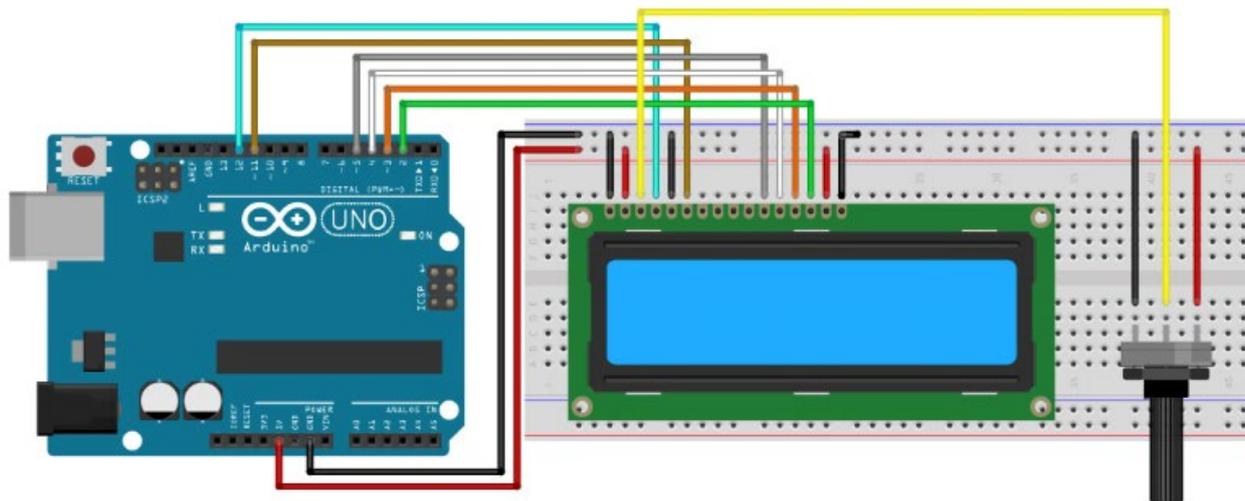
2. Através do circuito criado na alínea 1, crie um programa que, através do buzzer reproduza as notas Dó, Ré, Mi, Fá e Sol.

Nota	Frequência [Hz]
Dó	261
Ré	293
Mi	329
Fá	349
Sol	392



3. Execute o circuito na plataforma acima mencionada, da maneira que se segue abaixo. Através do LCD que se encontram no exercício execute um programa de forma a reproduzir mensagem no LCD.





```
#include <LiquidCrystal.h>
//Define os pinos que serão utilizados para ligação ao display
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup()
{
  //Define o número de colunas e linhas do LCD
  lcd.begin(16, 2);
}
void loop()
{
  //Limpa a tela
  lcd.clear();
  //Posiciona o cursor na coluna 3, linha 0;
  lcd.setCursor(3, 0);
  //Envia o texto entre aspas para o LCD
  lcd.print("Hello World");
  lcd.setCursor(3, 1);
  lcd.print(" LCD 16x2");
  delay(5000);
  for (int posicao = 0; posicao < 3; posicao++)
  {
    lcd.scrollDisplayLeft();
    delay(300);
  }
  for (int posicao = 0; posicao < 6; posicao++)
  {
    lcd.scrollDisplayRight();
    delay(300);
  }
}
```