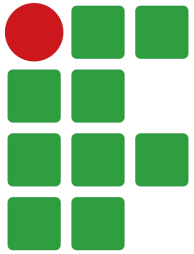
Eletricidade e Eletrônica

Curso Técnico em Informática para Internet

Lucas Sampaio Leite

Baseada na aula do professor Elton B. Torres

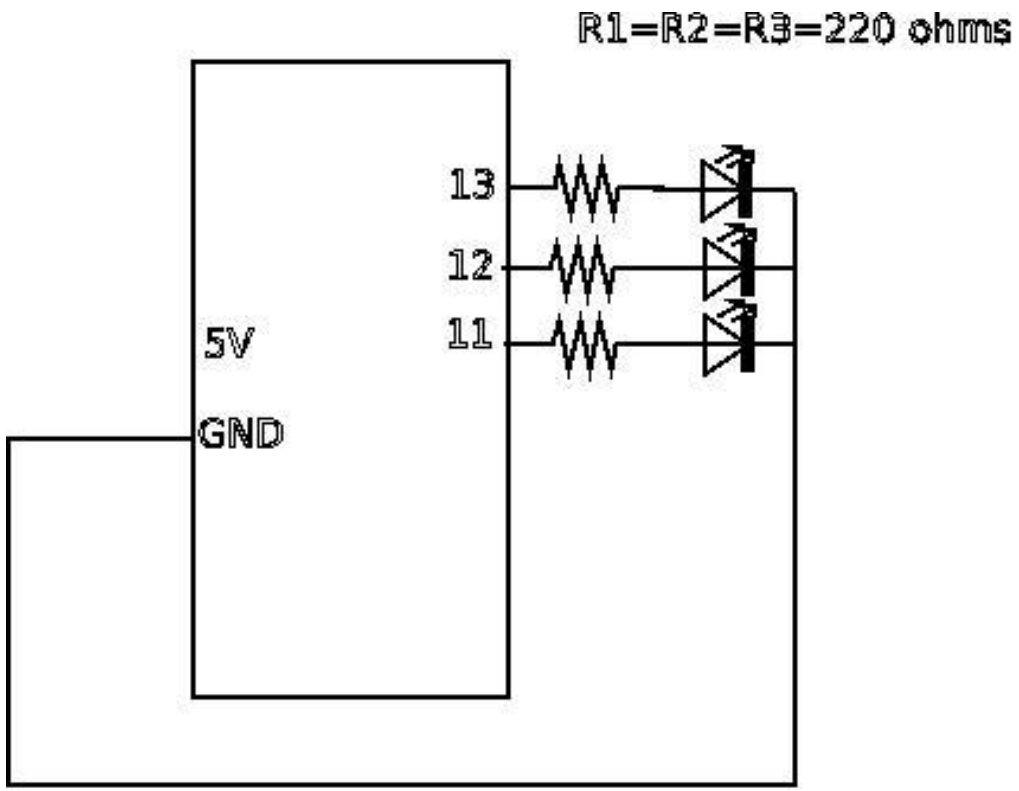


INSTITUTO FEDERAL
Pernambuco

Exercício

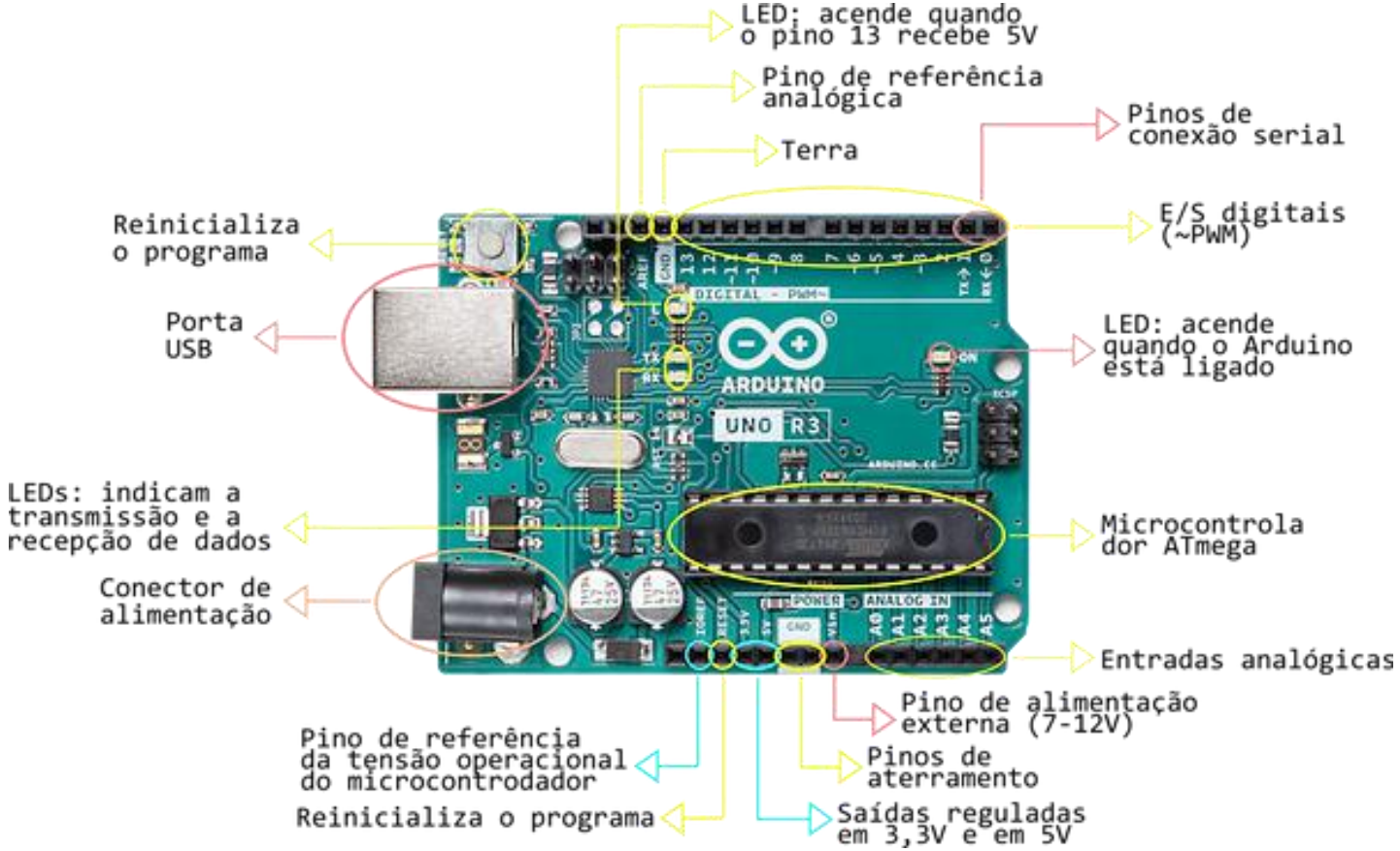
INSTITUTO FEDERAL
Pernambuco

- Construa um circuito e codifique para realizar a função de um semáforo de trânsito seguindo esquema abaixo.



Entradas e saídas digitais

- As entradas e saídas digitais (GPIO - General Purpose Input/Output) são os recursos mais básicos e importantes do Arduino. Elas permitem que a placa receba informações do ambiente e controle dispositivos externos.



Entradas e saídas digitais

INSTITUTO FEDERAL
Pernambuco

- Um pino digital pode assumir apenas dois estados lógicos:
  - LOW (0) → aproximadamente 0 V
  - HIGH (1) → aproximadamente 5 V (Arduino Uno) ou 3,3 V (algumas outras placas)
- Por isso, dizemos que ele trabalha com sinais digitais, diferentemente dos sinais analógicos, que podem assumir vários valores.

Saídas Digitais

INSTITUTO FEDERAL
Pernambuco

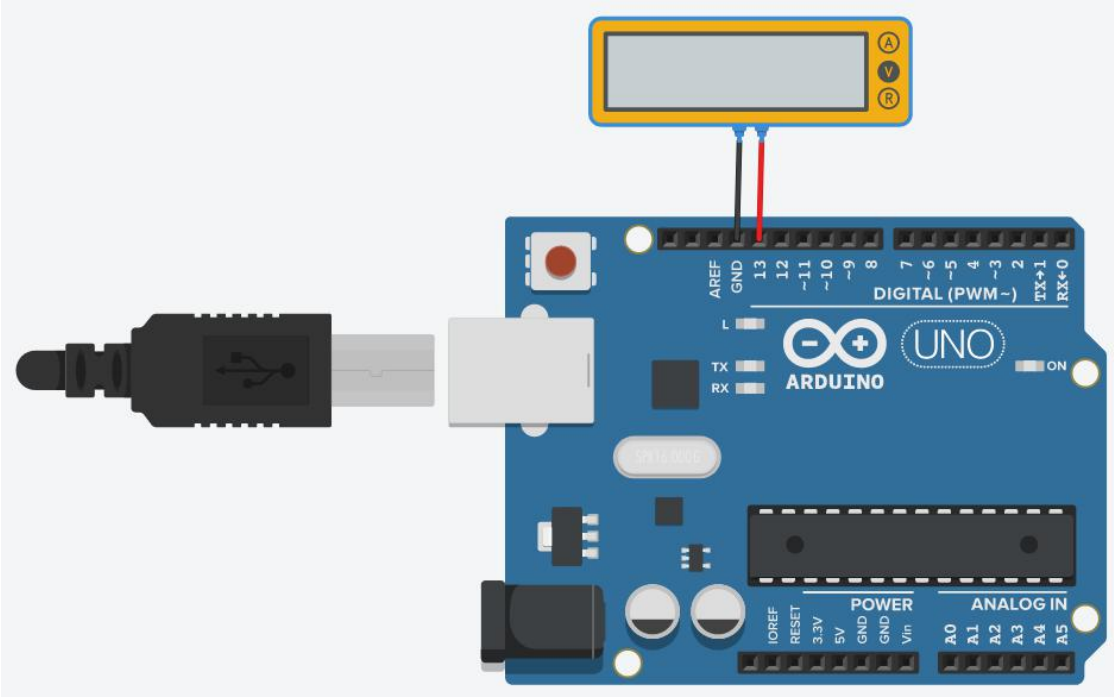
- Quando um pino é configurado como saída, o Arduino envia um sinal elétrico por esse pino, permitindo controlar dispositivos externos.
- Configuração:

```
const int led = 13;

void setup() {
    pinMode(led, OUTPUT);
}
```

- Leitura:

```
void loop() {
    digitalWrite(led, HIGH);
    delay(1000);

    digitalWrite(led, LOW);
    delay(1000);
}
```

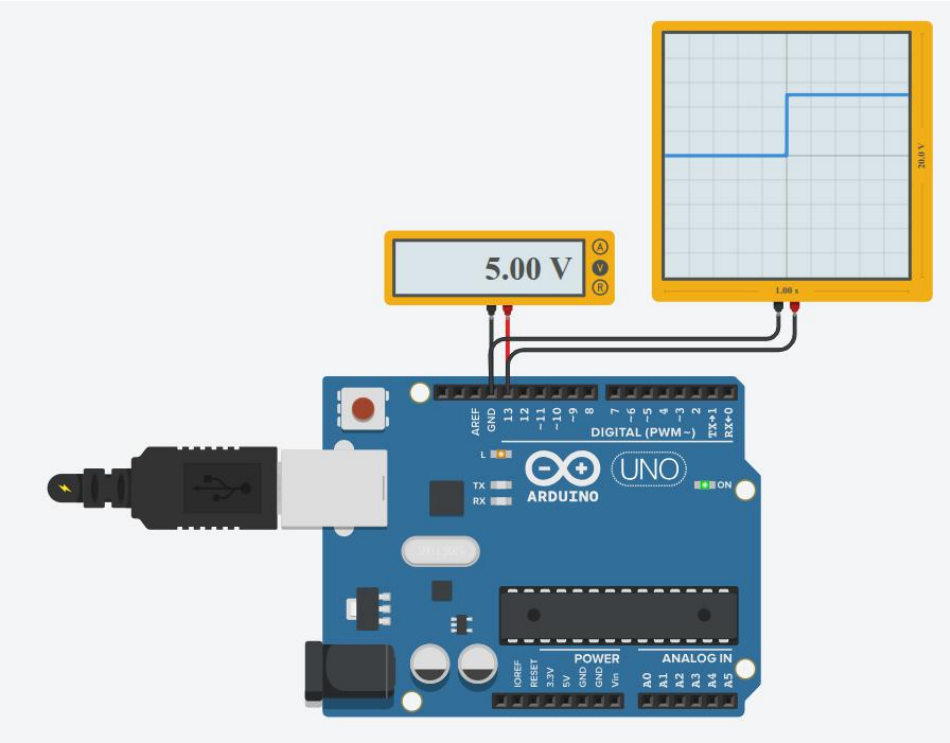
Prática: saída digital

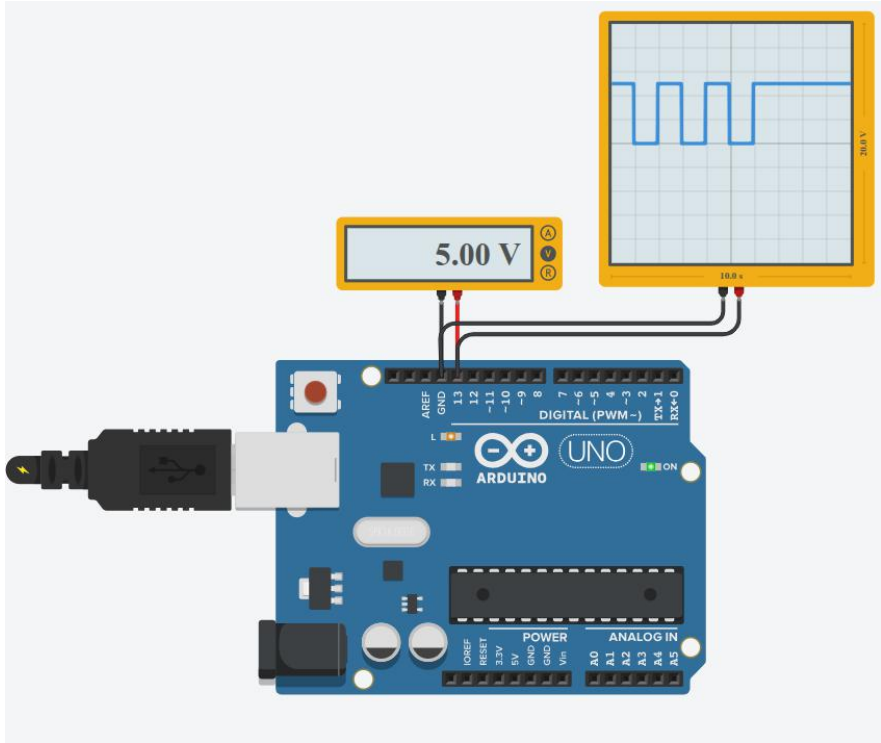


1. Monte o circuito acima no Tinkercad e execute o sketch blink
2. Adicione também um osciloscópio
3. Altere o valor do delay para 1ms.

Prática: saída digital

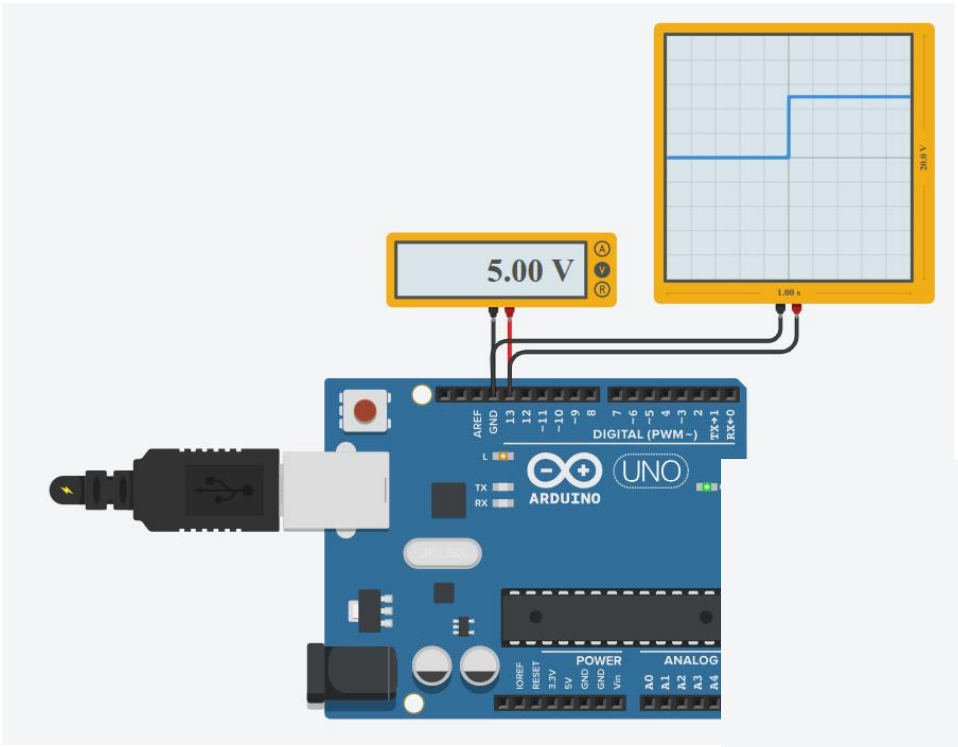
INSTITUTO FEDERAL
Pernambuco

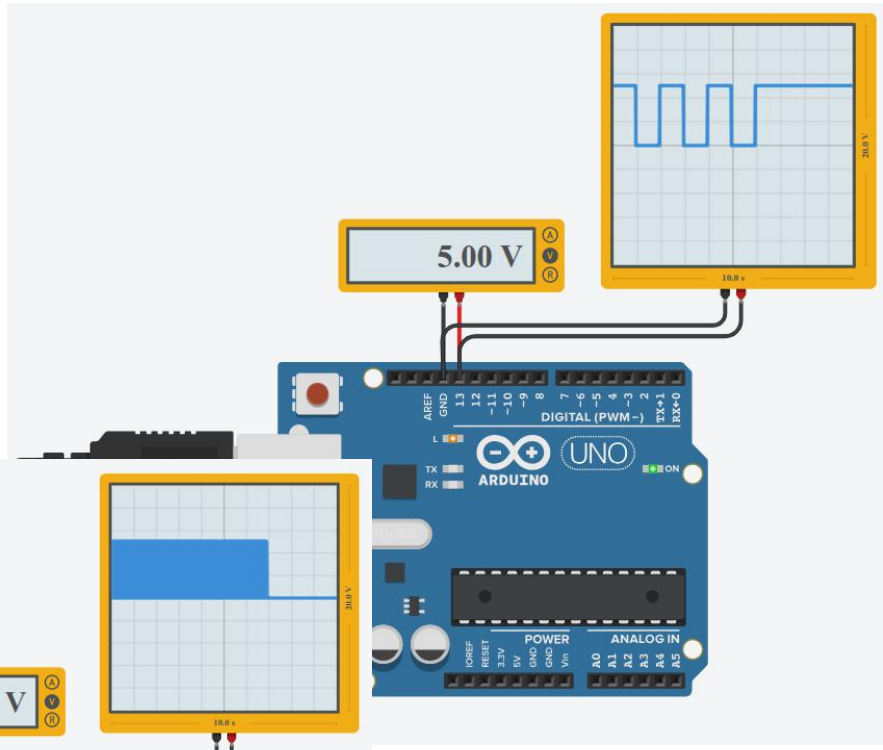


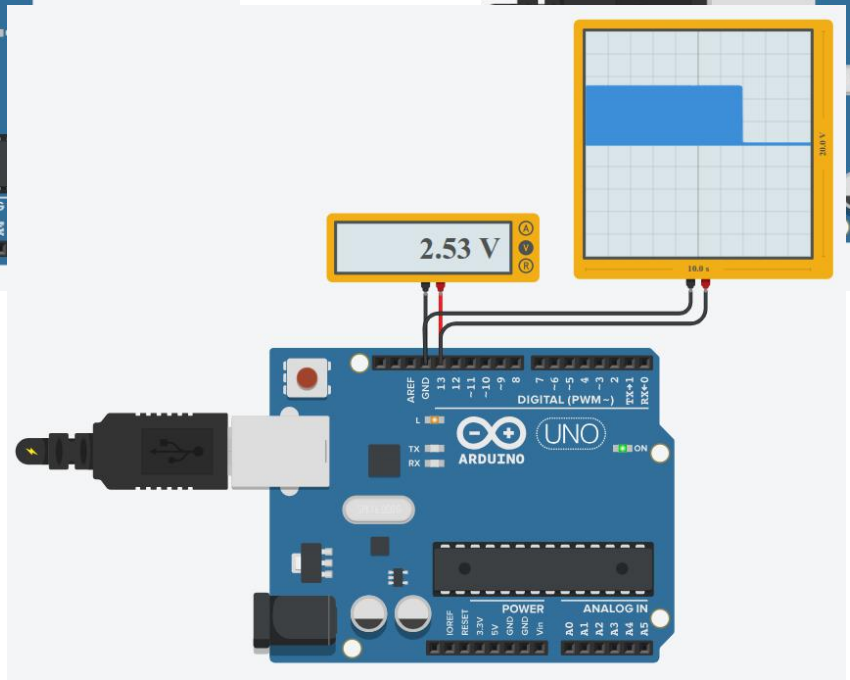


Prática: saída digital

INSTITUTO FEDERAL
Pernambuco



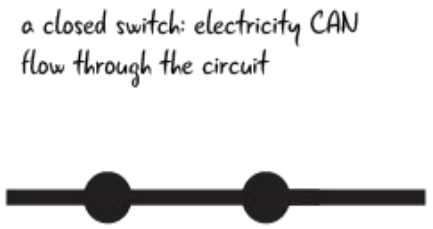


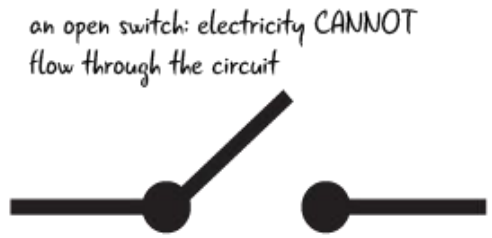


Revisando o push buttom

INSTITUTO FEDERAL
Pernambuco

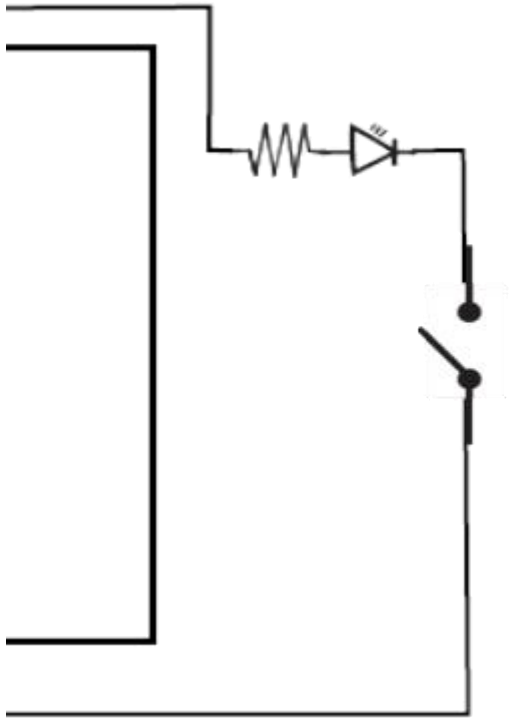
- Os push buttons (botões de pressão) são interruptores momentâneos.
- Eles fecham ou abrem um circuito apenas enquanto estão sendo pressionados.
- Ao soltar o botão, ele retorna automaticamente à posição original por meio de uma mola interna.

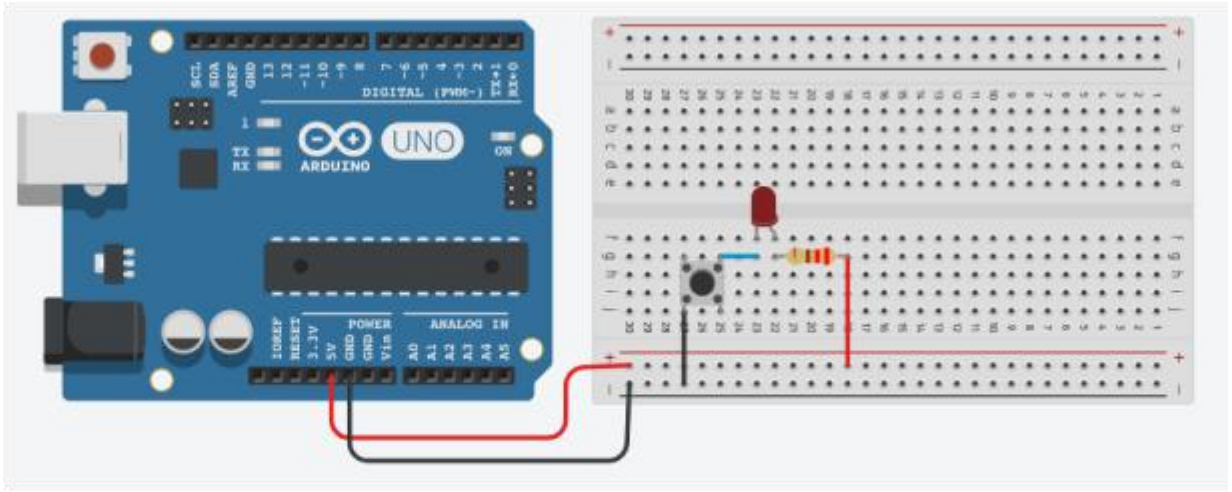




Revisando o push buttom

INSTITUTO FEDERAL
Pernambuco





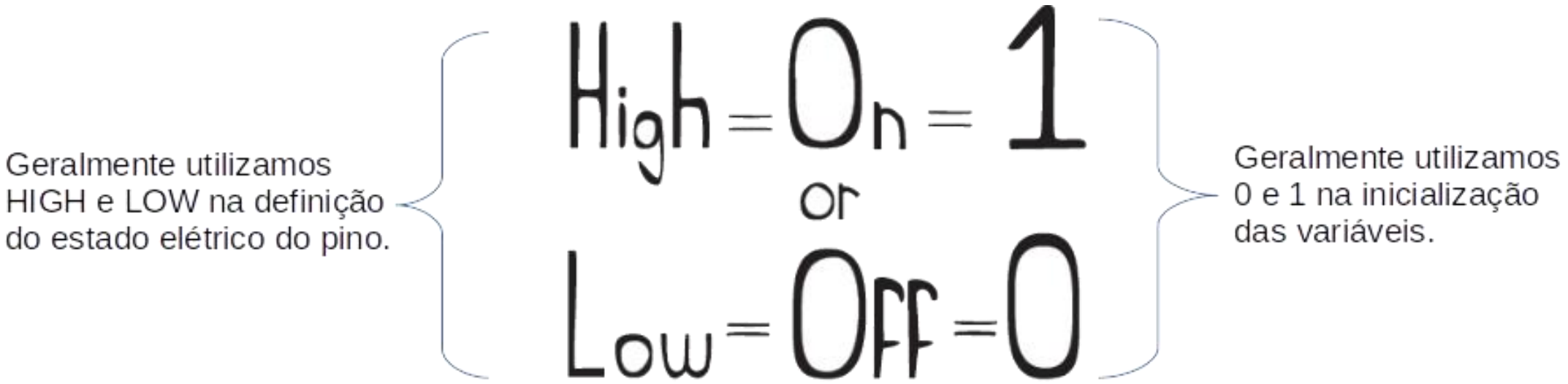
Entradas Digitais

INSTITUTO FEDERAL
Pernambuco

- A entrada digital permite que o Arduino receba informações do ambiente, identificando se um dispositivo está em um de dois estados possíveis.
- Por exemplo, ela pode ser utilizada para verificar se:
  - um botão foi pressionado;
  - uma chave está ligada ou desligada;
  - um sensor detectou ou não um objeto.

Entradas Digitais

- Uma entrada digital possui apenas dois estados lógicos possíveis:
  - LOW (0) – nível lógico baixo (aproximadamente 0 V);
  - HIGH (1) – nível lógico alto (aproximadamente 5 V no Arduino Uno).



Se a tensão de entrada for menor que 2.5V (0-desligada) se for maior de que 2.5V (1-ligada).

Entradas Digitais

INSTITUTO FEDERAL
Pernambuco

- Quando um pino é configurado como entrada, o Arduino lê o estado elétrico presente nesse pino.
- Configuração:

```
const int sensor = 2;

void setup() {

  pinMode(numPin, INPUT);

}
```

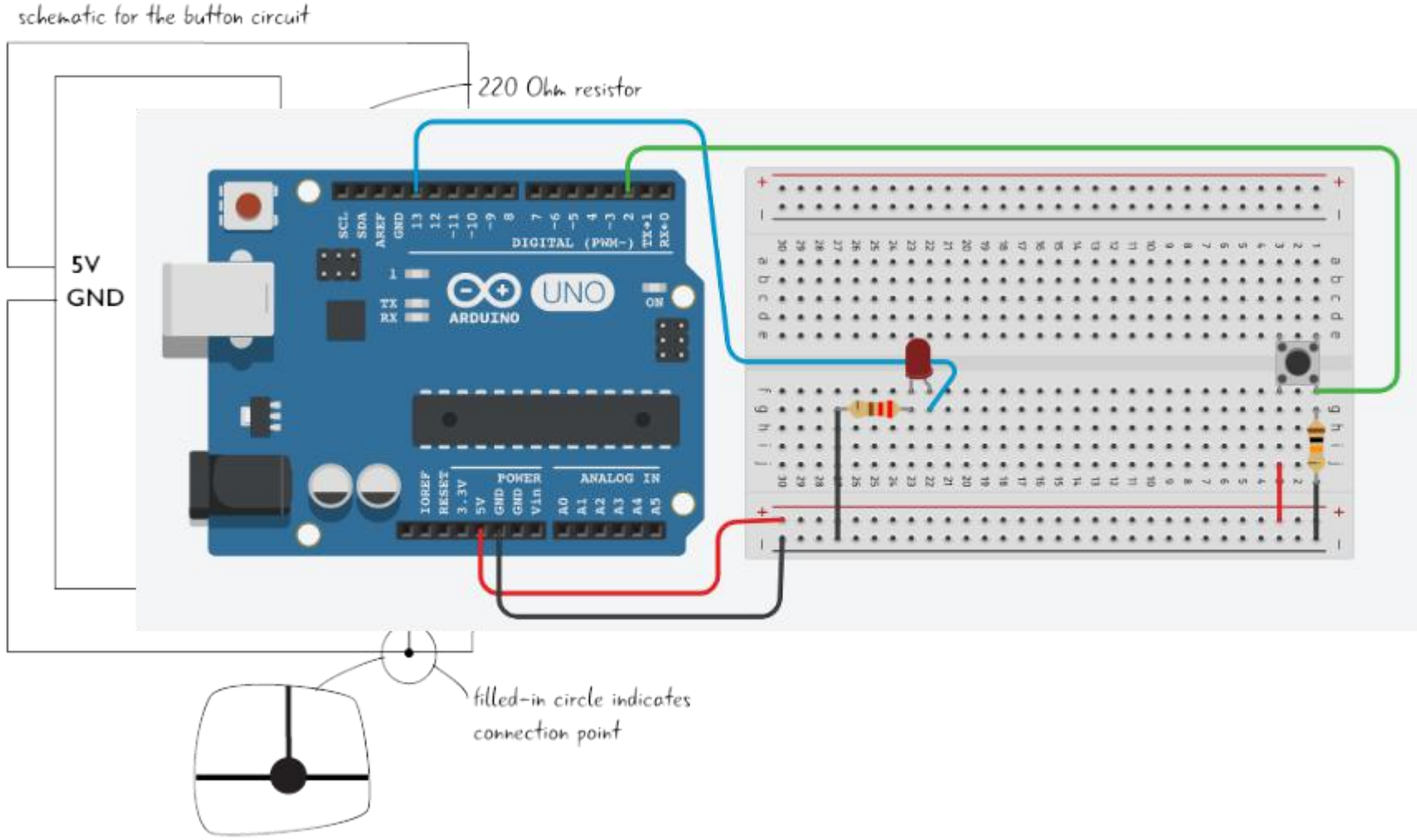
- Leitura:

```
void loop() {

  int estado = digitalRead(numPin);

}
```

O resultado será: HIGH ou LOW

Prática: acionando um led com um botão usando entrada digital

INSTITUTO FEDERAL
Pernambuco



Prática: acionando um led com um botão usando entrada digital

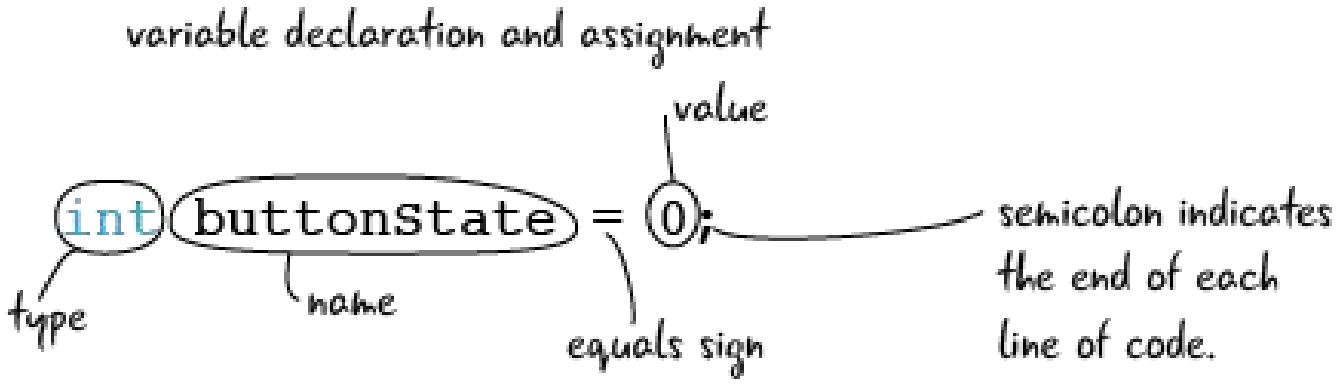
INSTITUTO FEDERAL
Pernambuco

```
// constantes não mudam. Elas são usadas para definir o número dos pinos:
const int buttonPin = 2;     // o número do pino do botão
const int ledPin =  13;      // o número do pino do LED

// variáveis mudam:
int buttonState = 0;         // variável para ler o status do botão

void setup() {
  // inicia o pino do LED como uma saída:
  pinMode(ledPin, OUTPUT);
  // inicia o pino do botão como uma entrada:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // lê o estado do valor do botão:
  buttonState = digitalRead(buttonPin);

  // verifica se o botão de pressão está pressionado. Se estiver
  // buttonState é igual a HIGH:
  if (buttonState == HIGH) {
    // acende o LED:
    digitalWrite(ledPin, HIGH);
  } else {
    // apaga o LED:
    digitalWrite(ledPin, LOW);
  }
}
```

Prática: acionando um led com um botão usando entrada digital

INSTITUTO FEDERAL
Pernambuco

- Analisando o sketch:
  - Variáveis: espaço de memória onde armazenamos um valor e acessamos através de um nome;
  - Declaração e atribuição: declarar uma variável é dar um nome. E atribuir um valor é associar um valor a essa variável;
  - Declarando variáveis: tipo de dado, nome da variável, sinal de atribuição e valor que queremos atribuir.



Prática: acionando um led com um botão usando entrada digital

INSTITUTO FEDERAL
Pernambuco

- Analisando o sketch:
  - Nomes das variáveis: não pode começar com número; não pode conter espaços e não pode ser uma palavra reservada da linguagem (ex.: int, delay). Deve existir apenas uma variável com determinado nome para cada sketch.
  - Valor da variável: valor a ser armazenado na variável. Ex.: valor 0 corresponde ao estado, ou valor de tensão no pino.
  - Tipo da variável: define que tipo de informação queremos salvar dentro da variável. Ex: int, float, string, character …
  - Qualificadores de variáveis - const: determina que o valor da variável não será mudado durante a execução do programa. É opcional, porém é uma boa prática de programação.

As funções setup() e loop() no Arduino

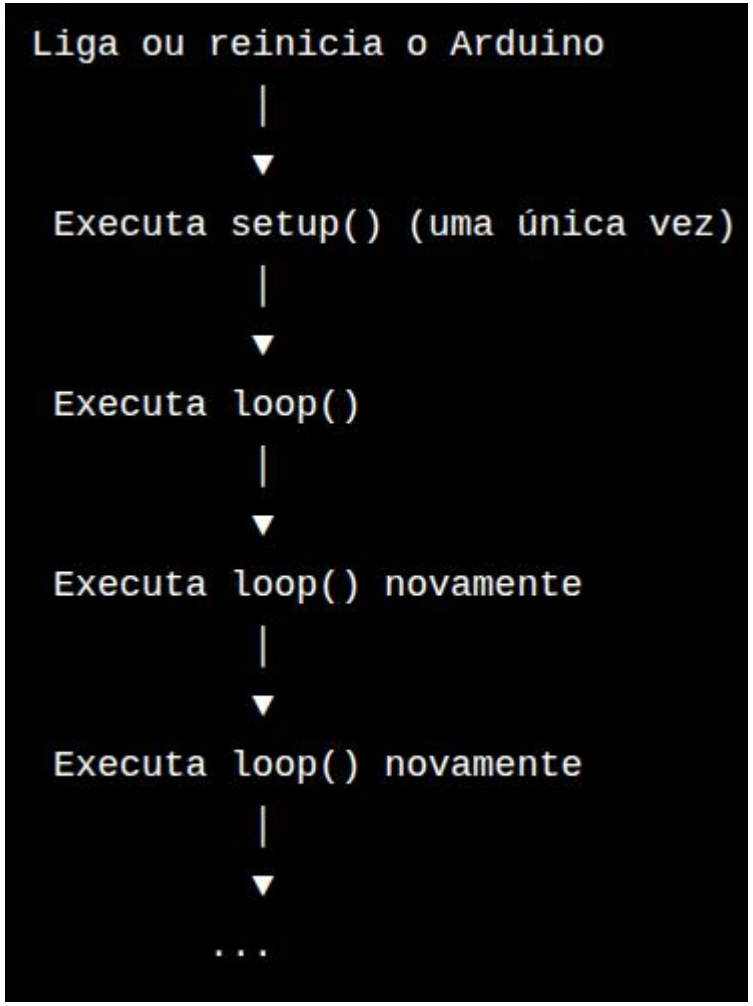
INSTITUTO FEDERAL
Pernambuco

- Todo programa em Arduino, chamado de sketch, possui duas funções principais: setup() e loop().
- Função setup():
  - A função setup() é executada uma única vez, logo após o Arduino ser ligado ou reiniciado.
  - Ela é utilizada para realizar as configurações iniciais do programa.
- Função loop():
  - Após executar o setup(), o Arduino entra na função loop().
  - Essa função é executada continuamente, repetindo-se enquanto a placa estiver ligada.
  - É nela que ficam as tarefas que o Arduino deve realizar repetidamente.

As funções setup() e loop() no Arduino

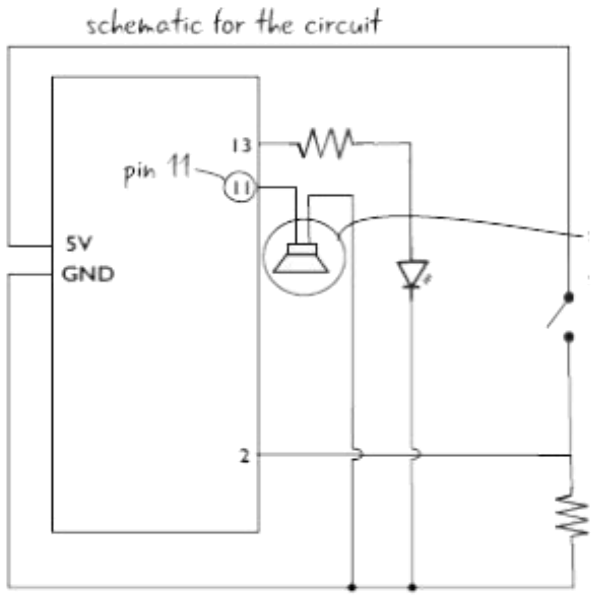
INSTITUTO FEDERAL
Pernambuco

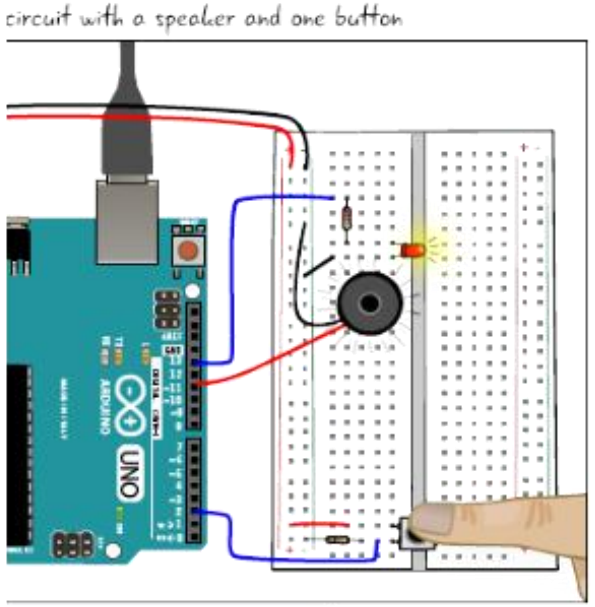
- O funcionamento do Arduino pode ser representado da seguinte forma:

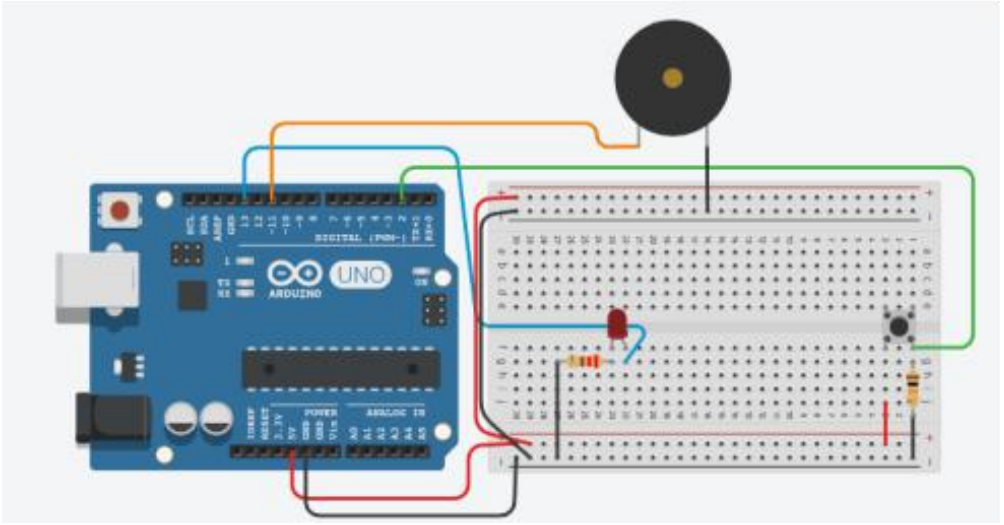


Prática: adicionando um alto-falante

INSTITUTO FEDERAL
Pernambuco







Prática: adicionando um alto-falante

INSTITUTO FEDERAL
Pernambuco

1)
```
const int speakerPin =  11;       // o número do pino do alto-falante
```

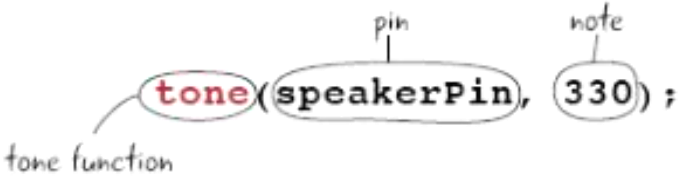
2)
```
// inicia o pino do alto-falante como uma saída:
pinMode(speakerPin, OUTPUT);
```

3)
```
tone(speakerPin, 330);
```

4)
```
noTone(speakerPin);
```

Prática: adiconando um alto-falante

- Funcionamento das funções:
  - tone() gera um som (nota musical) em um pino conectado a um buzzer;
  - noTone() interrompe a reprodução desse som.



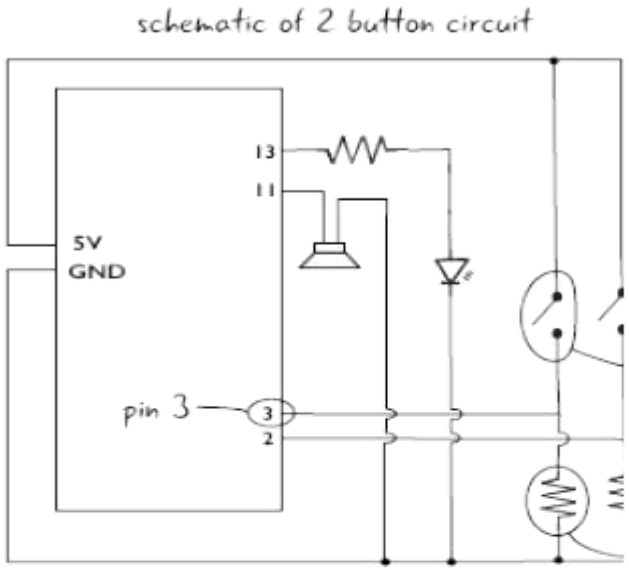
the values for a few notes

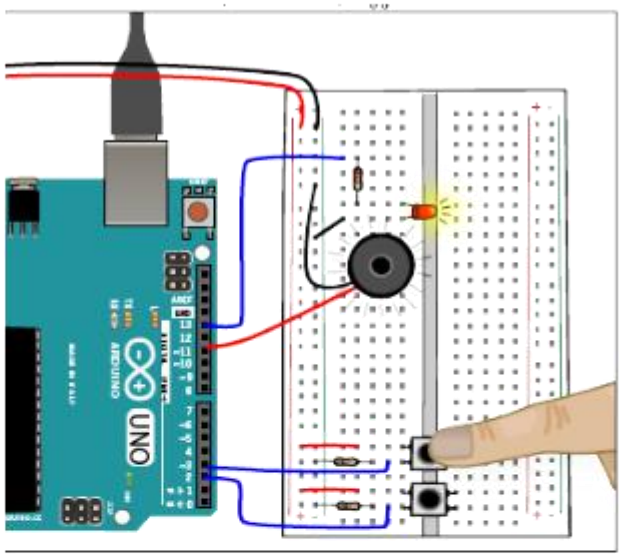
NOTE	FREQUENCY (hertz)
C (Dó)	262
D (Ré)	294
E (Mi)	330
F (Fá)	349
G (Sol)	392
A (Lá)	440
B (Si)	494
C (Dó)	523

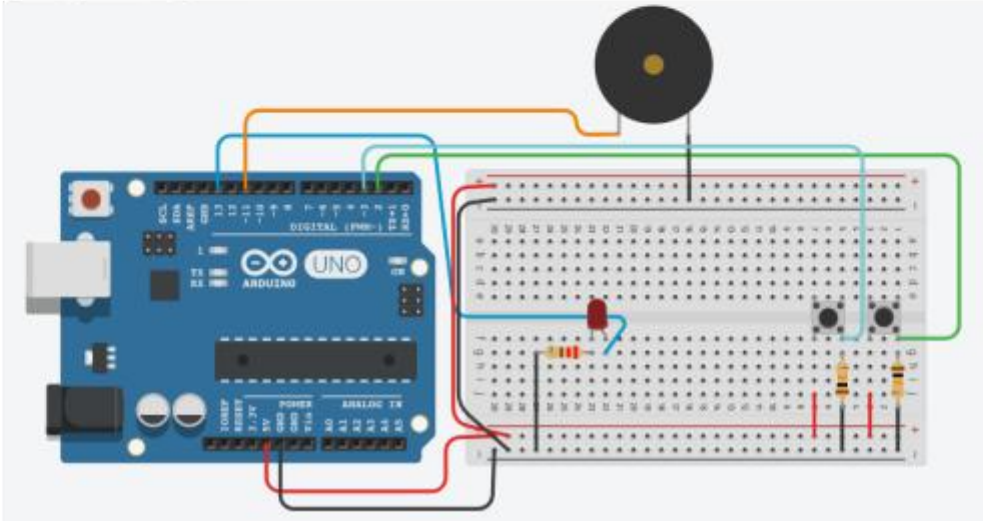
our first note

the sound an orchestra tunes to

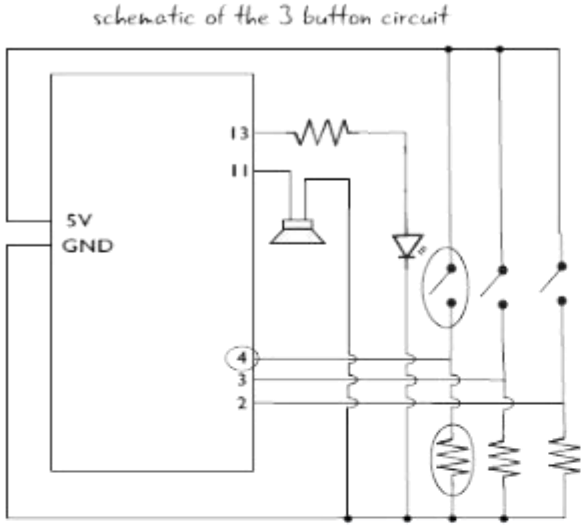
Prática: adicionando um segundo botão ao teclado

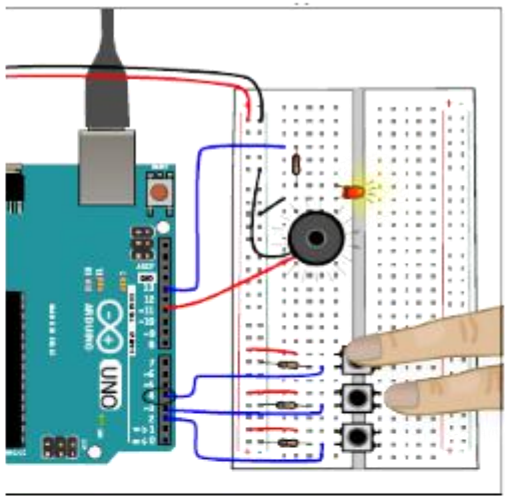


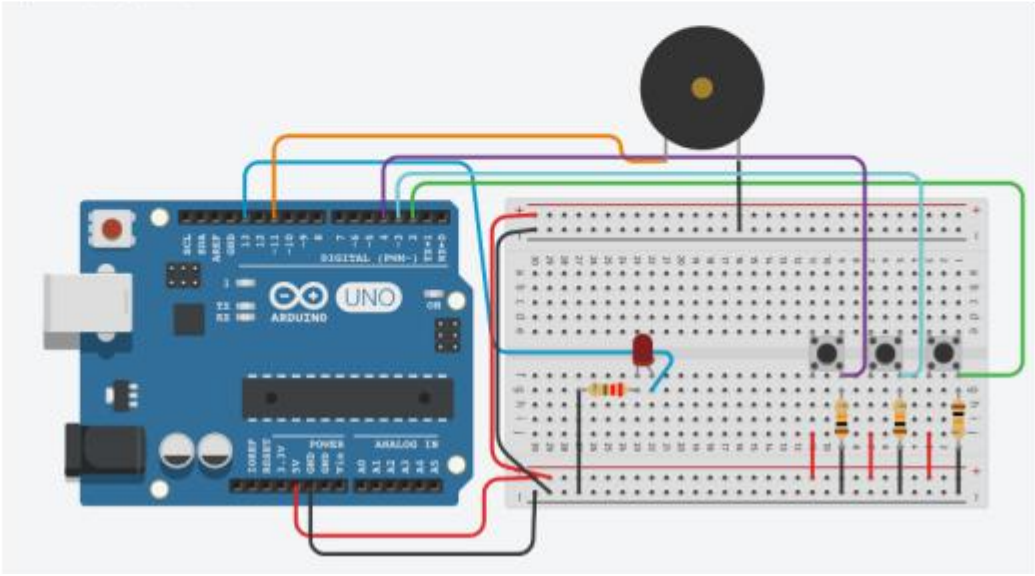




Prática: adicionando um terceiro botão ao teclado







Prática: adicionando um terceiro botão ao teclado

INSTITUTO FEDERAL
Pernambuco

```
// constantes não mudam. Elas são usadas para definir o número dos p
const int buttonPin = 2;     // o número do pino do botão 1
const int buttonPin2 = 3;     // o número do pino do botão 2
const int buttonPin3 = 4;     // o número do pino do botão 3
const int ledPin =  13;      // o número do pino do LED
const int speakerPin =  11;      // o número do pino do alto-falante

// variáveis mudam:
int buttonState = 0;         // variável para ler o status do botão1
int buttonState2 = 0;         // variável para ler o status do botão
int buttonState3 = 0;         // variável para ler o status do botão

void setup() {
  // inicia o pino do LED como uma saída:
  pinMode(ledPin, OUTPUT);
  // inicia os pinos dos botões como uma entrada:
  pinMode(buttonPin, INPUT);
  pinMode(buttonPin2, INPUT);
  pinMode(buttonPin3, INPUT);
  // inicia o pino do alto-falante como uma saída:
  pinMode(speakerPin, OUTPUT);
}

void loop() {
  // lê o estado do valor dos botões:
  buttonState = digitalRead(buttonPin);
  buttonState2 = digitalRead(buttonPin2);
  buttonState3 = digitalRead(buttonPin3);

  // verifica se o botão de pressão está pressionado. Se estiver
  // buttonState é igual a HIGH:
  if (buttonState == HIGH) {
    // acende o LED:
    digitalWrite(ledPin, HIGH);
    tone(speakerPin, 262);
  } else if (buttonState2 == HIGH) {
    digitalWrite(ledPin, HIGH);
    tone(speakerPin, 349);
  } else if (buttonState3 == HIGH) {
    digitalWrite(ledPin, HIGH);
    tone(speakerPin, 494);
  } else {
    noTone(speakerPin);
    // apaga o LED:
    digitalWrite(ledPin, LOW);
  }
}
```

Dúvidas

INSTITUTO FEDERAL
Pernambuco

Eletricidade e Eletrônica

Curso Técnico em Informática para Internet

Lucas Sampaio Leite