

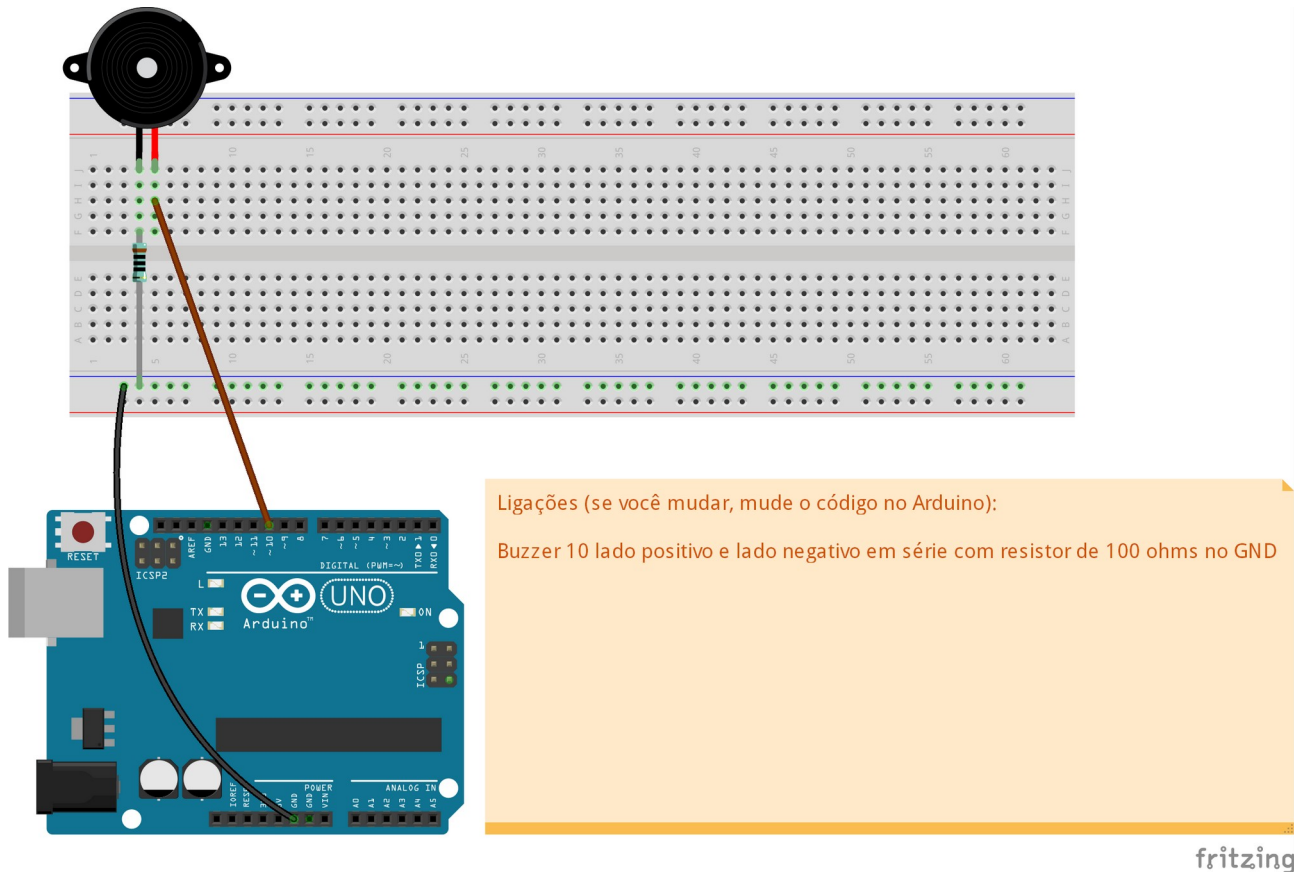
“Jogo” Simon.

Vamos fazer um brinquedo que foi sucesso há alguns anos, o Simon.

Resumidamente ele sorteia luzes em uma sequência aleatória; depois, você tem que repetir a sequência. Se você acertar a sequência ele toca a música de ‘acerto’ e, se você errar, a de ‘erro’.

Vamos fazer passo a passo.

Primeiro, vamos ligar e testar o *buzzer*:



Código - Simon_1_buzzer

```
/*
Nesta primeira etapa, vamos ligar o buzzer.
Ele está conectado no pino 10 (lado positivo) e no GND (lado negativo) em série com um
resistor de 100 ohms.
*/
#define pinoBuzzer 10 // se você ligou em outro pino, ajuste aqui

void setup() {
  pinMode(pinoBuzzer, OUTPUT); // é uma saída
  somPerdeu(); //executa a função 'somPerdeu'
  delay(800); // aguarda 800 ms
  somGanhou(); //executa a função 'somGanhou'
}

void loop() {

}

void somPerdeu(){
  //do código da SparkFun
  tone(pinoBuzzer, 130, 250); //E6
  delay(275);
  tone(pinoBuzzer, 73, 250); //G6
}
```

```

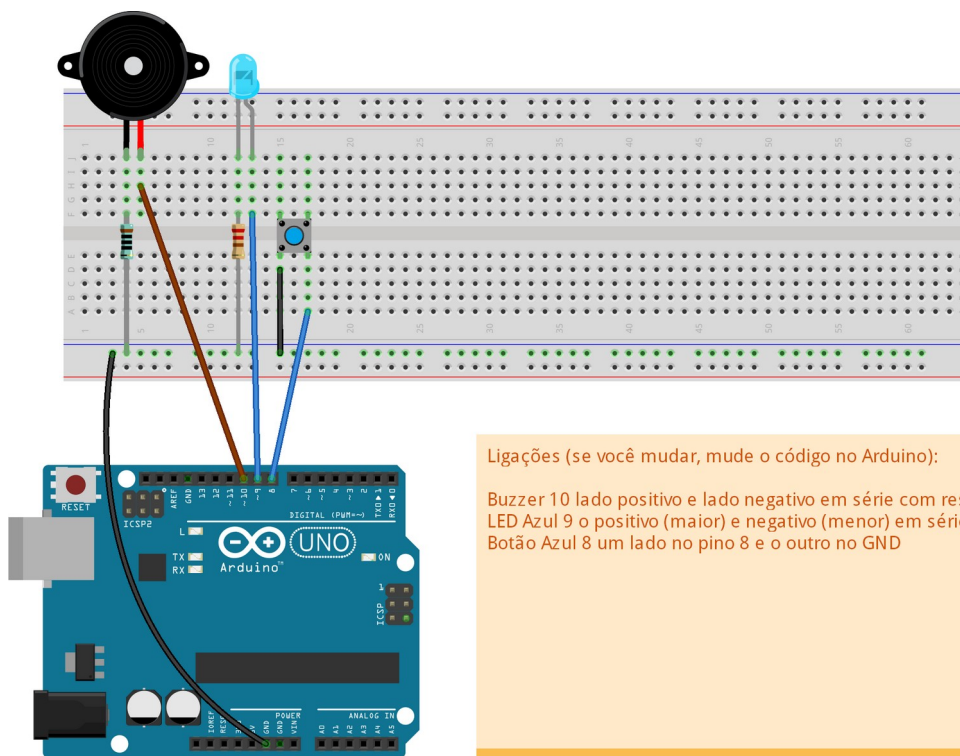
delay(275);
tone(pinoBuzzer, 65, 150);    //E7
delay(175);
tone(pinoBuzzer, 98, 500);    //C7
delay(500);
}

void somGanhou(){
  //do código da SparkFun
  tone(pinoBuzzer, 1318, 150); //E6
  delay(175);
  tone(pinoBuzzer, 1567, 150); //G6
  delay(175);
  tone(pinoBuzzer, 2637, 150); //E7
  delay(175);
  tone(pinoBuzzer, 2093, 150); //C7
  delay(175);
  tone(pinoBuzzer, 2349, 150); //D7
  delay(175);
  tone(pinoBuzzer, 3135, 500); //G7
  delay(500);
}

```

Ligue o buzzer e teste! Se tudo estiver ok, vá em ferente.

Na segunda etapa vamos ligar um LED e seu botão correspondente (o azul).



Ligações (se você mudar, mude o código no Arduino):

Buzzer 10 lado positivo e lado negativo em série com resistor de 100 ohms no GND
 LED Azul 9 o positivo (maior) e negativo (menor) em série com resistor de 220 ohms no GND
 Botão Azul 8 um lado no pino 8 e o outro no GND

fritzing

Próximo código - Simon_2_ledBotaoAzul

```

/*
Nesta segunda etapa, deixamos ligado o buzzer, que está conectado no pino 10 (lado
positivo) e no GND (lado negativo) em série com um resistor de 100 ohms.
Ligamos um LED azul, conectado com a parte positiva no pino 9 e a negativa em série com
um resistor de 220 ohms no GND.
Também vamos ligar o botão/ interruptor azul, com um lado no pino 8 e o outro no GND
*/
#define pinoBuzzer 10 // se você ligou em outro pino, ajuste aqui
#define LEDAzul 9 // parte positiva no pino 9 e a negativa em série com um resistor de
220 ohms no GND

```

```

#define botaoAzul 8 // um lado no pino 8 e o outro no GND

void setup() {
  pinMode(pinoBuzzer, OUTPUT); // é uma saída
  pinMode(LEDazul, OUTPUT); // é uma saída
  pinMode(botaoAzul, INPUT_PULLUP); // entrada que estará alta se não estiver pressionado
o botão
  somPerdeu(); //executa a função 'somPerdeu'
  delay(800); // aguarda 800 ms
  somGanhou(); //executa a função 'somGanhou'
}

void loop() {
  if(digitalRead(botaoAzul) == LOW){
    // se o botão foi pressionado
    digitalWrite(LEDazul, HIGH); // acende o LED azul
    tone(pinoBuzzer, 494, 200); // som de 494 Hz
  }
  else{
    // se o botão não estiver pressionado
    digitalWrite(LEDazul, LOW); // apaga o LED azul
    noTone(pinoBuzzer); // sem som
  }
}

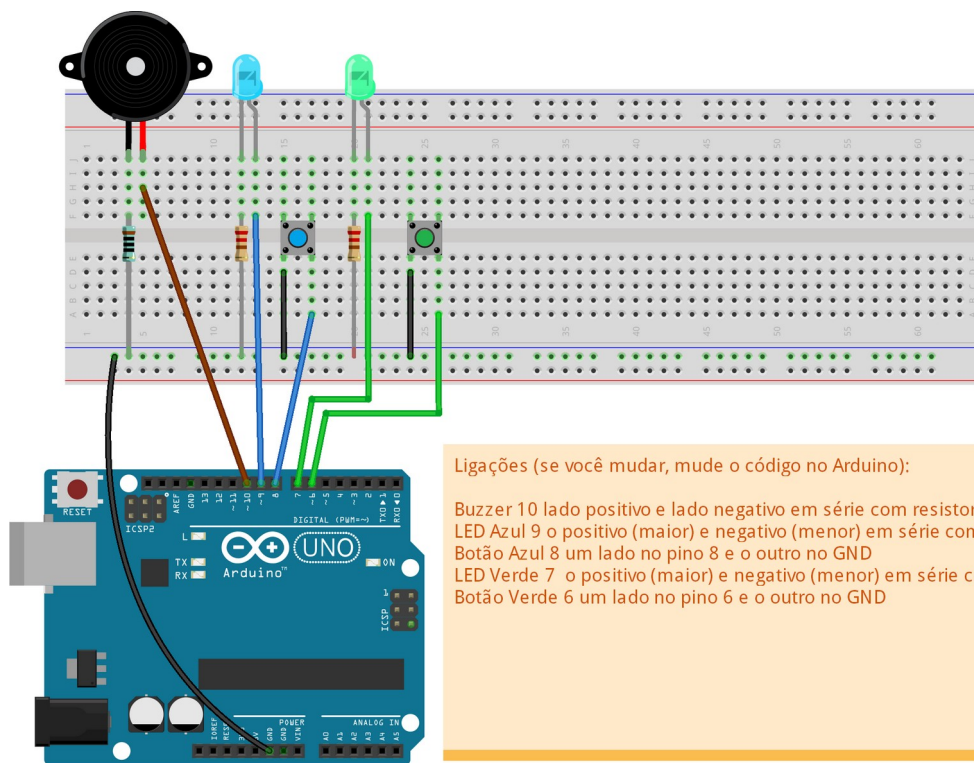
void somPerdeu(){
  //do código da SparkFun
  tone(pinoBuzzer, 130, 250); //E6
  delay(275);
  tone(pinoBuzzer, 73, 250); //G6
  delay(275);
  tone(pinoBuzzer, 65, 150); //E7
  delay(175);
  tone(pinoBuzzer, 98, 500); //C7
  delay(500);
}

void somGanhou(){
  //do código da SparkFun
  tone(pinoBuzzer, 1318, 150); //E6
  delay(175);
  tone(pinoBuzzer, 1567, 150); //G6
  delay(175);
  tone(pinoBuzzer, 2637, 150); //E7
  delay(175);
  tone(pinoBuzzer, 2093, 150); //C7
  delay(175);
  tone(pinoBuzzer, 2349, 150); //D7
  delay(175);
  tone(pinoBuzzer, 3135, 500); //G7
  delay(500);
}

```

Faça as ligações, envie o código para a plaquinha e teste. Se tudo deu certo, prossiga; caso contrário revise as ligações e o código.

Agora vamos colocar mais um LED (o verde) e seu botão:



Ligações (se você mudar, mude o código no Arduino):

Buzzer 10 lado positivo e lado negativo em série com resistor de 100 ohms no GND
 LED Azul 9 o positivo (maior) e negativo (menor) em série com resistor de 220 ohms no GND
 Botão Azul 8 um lado no pino 8 e o outro no GND
 LED Verde 7 o positivo (maior) e negativo (menor) em série com resistor de 220 ohms no GND
 Botão Verde 6 um lado no pino 6 e o outro no GND

fritzing

Perceba a semelhança das ligações. O led está ligado com seu lado positivo (o terminal maior) em um pino do Arduino. O lado negativo (o terminal menor) está ligado no GND. O que muda de um led para outro é o pino do Arduino. O mesmo ocorre em relação ao interruptor, o qual tem um lado ligado em um pino do Arduino e o outro no GND.

O código agora ficou assim - Simon_3_ledBotaoVerde:

```
/*
Nesta terceira etapa, deixamos ligado o buzzer, que está conectado no pino 10 (lado
positivo) e no GND (lado negativo) em série com um resistor de 100 ohms.
Deixamos ligado o LED azul, conectado com a parte positiva no pino 9 e a negativa em
série com um resistor de 220 ohms no GND.
Deixamos ligado o botão/ interruptor azul, com um lado no pino 8 e o outro no GND.
Vamos ligar um LED verde, conectado com a parte positiva no pino 7 e a negativa em série
com um resistor de 220 ohms no GND.
Vamos ligar um botão/ interruptor verde, com um lado no pino 6 e o outro no GND.
*/
#define pinoBuzzer 10 // se você ligou em outro pino, ajuste aqui
#define LEDAzul 9 // parte positiva no pino 9 e a negativa em série com um resistor de
220 ohms no GND
#define botaoAzul 8 // um lado no pino 8 e o outro no GND
#define LEDVerde 7 // parte positiva no pino 7 e a negativa em série com um resistor de
220 ohms no GND
#define botaoVerde 6 // um lado no pino 6 e o outro no GND

void setup() {
  pinMode(pinoBuzzer, OUTPUT); // é uma saída
  pinMode(LEDAzul, OUTPUT); // é uma saída
  pinMode(botaoAzul, INPUT_PULLUP); // entrada que estará alta se não estiver pressionado
o botão
  pinMode(LEDVerde, OUTPUT); // é uma saída
  pinMode(botaoVerde, INPUT_PULLUP); // entrada que estará alta se não estiver
pressionado o botão
  somPerdeu(); //executa a função 'somPerdeu'
  delay(800); // aguarda 800 ms
  somGanhou(); //executa a função 'somGanhou'
}

void loop() {
  if(digitalRead(botaoAzul) == LOW){
```

```

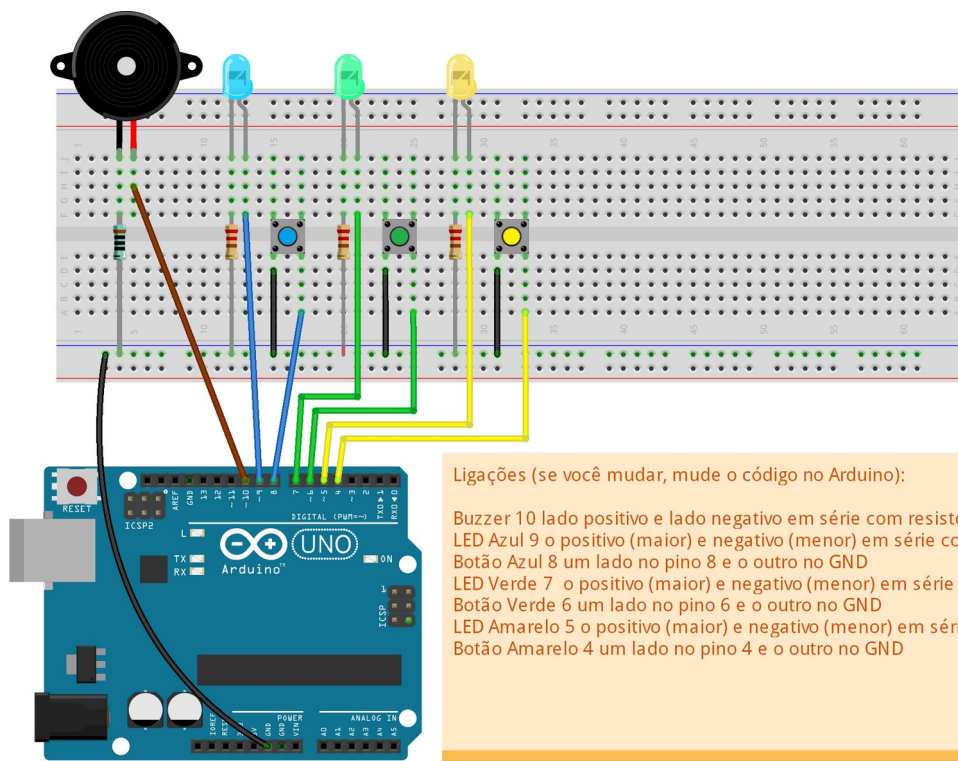
    // se o botão foi pressionado
    digitalWrite(LED Azul, HIGH); // acende o LED azul
    tone(pinoBuzzer, 494, 200); // som de 494 Hz
}
else{
    // se o botão não estiver pressionado
    digitalWrite(LED Azul, LOW); // apaga o LED azul
    noTone(pinoBuzzer); // sem som
}
if(digitalRead(botaoVerde) == LOW){
    // se o botão foi pressionado
    digitalWrite(LED Verde, HIGH); // acende o LED verde
    tone(pinoBuzzer, 392, 200); // som de 392 Hz
}
else{
    // se o botão não estiver pressionado
    digitalWrite(LED Verde, LOW); // apaga o LED verde
    noTone(pinoBuzzer); // sem som
}
}

void somPerdeu(){
    //do código da SparkFun
    tone(pinoBuzzer, 130, 250); //E6
    delay(275);
    tone(pinoBuzzer, 73, 250); //G6
    delay(275);
    tone(pinoBuzzer, 65, 150); //E7
    delay(175);
    tone(pinoBuzzer, 98, 500); //C7
    delay(500);
}

void somGanhou(){
    //do código da SparkFun
    tone(pinoBuzzer, 1318, 150); //E6
    delay(175);
    tone(pinoBuzzer, 1567, 150); //G6
    delay(175);
    tone(pinoBuzzer, 2637, 150); //E7
    delay(175);
    tone(pinoBuzzer, 2093, 150); //C7
    delay(175);
    tone(pinoBuzzer, 2349, 150); //D7
    delay(175);
    tone(pinoBuzzer, 3135, 500); //G7
    delay(500);
}

```

Novamente, faça as ligações e teste com o código. Se tudo deu certo prossiga para o próximo LED, o amarelo (e seu interruptor):



Ligações (se você mudar, mude o código no Arduino):

Buzzer 10 lado positivo e lado negativo em série com resistor de 100 ohms no GND
 LED Azul 9 o positivo (maior) e negativo (menor) em série com resistor de 220 ohms no GND
 Botão Azul 8 um lado no pino 8 e o outro no GND
 LED Verde 7 o positivo (maior) e negativo (menor) em série com resistor de 220 ohms no GND
 Botão Verde 6 um lado no pino 6 e o outro no GND
 LED Amarelo 5 o positivo (maior) e negativo (menor) em série com resistor de 220 ohms no GND
 Botão Amarelo 4 um lado no pino 4 e o outro no GND

fritzing

Próximo código – Simon_4_ledBotaoAmarelo

```

/*
Nesta quarta etapa, deixamos ligado o buzzer, que está conectado no pino 10 (lado
positivo) e no GND (lado negativo) em série com um resistor de 100 ohms.
Deixamos ligado o LED azul, conectado com a parte positiva no pino 9 e a negativa em
série com um resistor de 220 ohms no GND.
Deixamos ligado o botão/ interruptor azul, com um lado no pino 8 e o outro no GND.
Deixamos ligado o LED verde, conectado com a parte positiva no pino 7 e a negativa em
série com um resistor de 220 ohms no GND.
Deixamos ligado o botão/ interruptor verde, com um lado no pino 6 e o outro no GND.
Vamos ligar um LED amarelo, conectado com a parte positiva no pino 5 e a negativa em
série com um resistor de 220 ohms no GND.
Vamos ligar um botão/ interruptor amarelo, com um lado no pino 4 e o outro no GND.
*/
#define pinoBuzzer 10 // se você ligou em outro pino, ajuste aqui
#define LEDAzul 9 // parte positiva no pino 9 e a negativa em série com um resistor de
220 ohms no GND
#define botaoAzul 8 // um lado no pino 8 e o outro no GND
#define LEDVerde 7 // parte positiva no pino 7 e a negativa em série com um resistor de
220 ohms no GND
#define botaoVerde 6 // um lado no pino 6 e o outro no GND
#define LEDAmarelo 5 // parte positiva no pino 5 e a negativa em série com um resistor de
220 ohms no GND
#define botaoAmarelo 4 // um lado no pino 4 e o outro no GND

void setup() {
  pinMode(pinoBuzzer, OUTPUT); // é uma saída
  pinMode(LEDAzul, OUTPUT); // é uma saída
  pinMode(botaoAzul, INPUT_PULLUP); // entrada que estará alta se não estiver pressionado
  o botão
  pinMode(LEDVerde, OUTPUT); // é uma saída
  pinMode(botaoVerde, INPUT_PULLUP); // entrada que estará alta se não estiver
  pressionado o botão
  pinMode(LEDAmarelo, OUTPUT); // é uma saída
  pinMode(botaoAmarelo, INPUT_PULLUP); // entrada que estará alta se não estiver
  pressionado o botão
  somPerdeu(); //executa a função 'somPerdeu'
  delay(800); // aguarda 800 ms
  somGanhou(); //executa a função 'somGanhou'
}

```

```

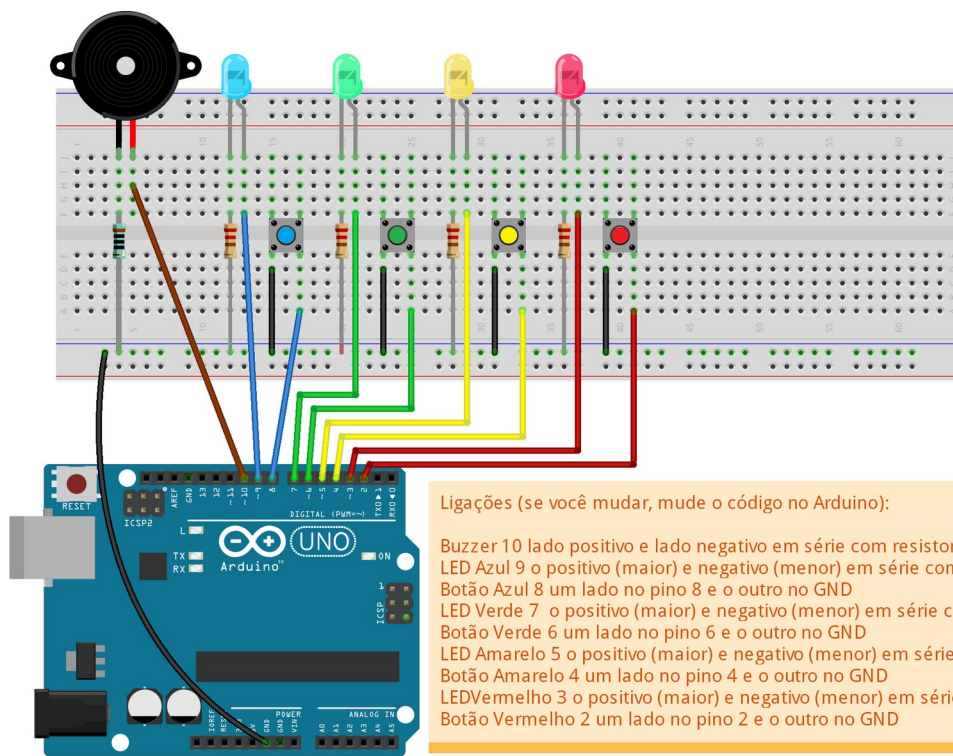
void loop() {
  if(digitalRead(botaoAzul) == LOW){
    // se o botão foi pressionado
    digitalWrite(LED Azul, HIGH); // acende o LED azul
    tone(pinoBuzzer, 494, 200); // som de 494 Hz
  }
  else{
    // se o botão não estiver pressionado
    digitalWrite(LED Azul, LOW); // apaga o LED azul
    noTone(pinoBuzzer); // sem som
  }
  if(digitalRead(botaoVerde) == LOW){
    // se o botão foi pressionado
    digitalWrite(LED Verde, HIGH); // acende o LED verde
    tone(pinoBuzzer, 392, 200); // som de 392 Hz
  }
  else{
    // se o botão não estiver pressionado
    digitalWrite(LED Verde, LOW); // apaga o LED verde
    noTone(pinoBuzzer); // sem som
  }
  if(digitalRead(botaoAmarelo) == LOW){
    // se o botão foi pressionado
    digitalWrite(LED Amarelo, HIGH); // acende o LED amarelo
    tone(pinoBuzzer, 330, 200); // som de 330 Hz
  }
  else{
    // se o botão não estiver pressionado
    digitalWrite(LED Amarelo, LOW); // apaga o LED amarelo
    noTone(pinoBuzzer); // sem som
  }
}

void somPerdeu(){
  //do código da SparkFun
  tone(pinoBuzzer, 130, 250); //E6
  delay(275);
  tone(pinoBuzzer, 73, 250); //G6
  delay(275);
  tone(pinoBuzzer, 65, 150); //E7
  delay(175);
  tone(pinoBuzzer, 98, 500); //C7
  delay(500);
}

void somGanhou(){
  //do código da SparkFun
  tone(pinoBuzzer, 1318, 150); //E6
  delay(175);
  tone(pinoBuzzer, 1567, 150); //G6
  delay(175);
  tone(pinoBuzzer, 2637, 150); //E7
  delay(175);
  tone(pinoBuzzer, 2093, 150); //C7
  delay(175);
  tone(pinoBuzzer, 2349, 150); //D7
  delay(175);
  tone(pinoBuzzer, 3135, 500); //G7
  delay(500);
}

```

Teste e prossiga para o LED vermelho se tudo estiver ok.



Ligações (se você mudar, mude o código no Arduino):

Buzzer 10 lado positivo e lado negativo em série com resistor de 100 ohms no GND
 LED Azul 9 o positivo (maior) e negativo (menor) em série com resistor de 220 ohms no GND
 Botão Azul 8 um lado no pino 8 e o outro no GND
 LED Verde 7 o positivo (maior) e negativo (menor) em série com resistor de 220 ohms no GND
 Botão Verde 6 um lado no pino 6 e o outro no GND
 LED Amarelo 5 o positivo (maior) e negativo (menor) em série com resistor de 220 ohms no GND
 Botão Amarelo 4 um lado no pino 4 e o outro no GND
 LED Vermelho 3 o positivo (maior) e negativo (menor) em série com resistor de 220 ohms no GND
 Botão Vermelho 2 um lado no pino 2 e o outro no GND

fritzing

Note como as ligações são feitas.

Próximo código – Simon_5_ledBotaoVermelho

```
/*
Nesta quarta etapa, deixamos ligado o buzzer, que está conectado no pino 10 (lado
positivo) e no GND (lado negativo) em série com um resistor de 100 ohms.
Deixamos ligado o LED azul, conectado com a parte positiva no pino 9 e a negativa em
série com um resistor de 220 ohms no GND.
Deixamos ligado o botão/ interruptor azul, com um lado no pino 8 e o outro no GND.
Deixamos ligado o LED verde, conectado com a parte positiva no pino 7 e a negativa em
série com um resistor de 220 ohms no GND.
Deixamos ligado o botão/ interruptor verde, com um lado no pino 6 e o outro no GND.
Deixamos ligado o LED amarelo, conectado com a parte positiva no pino 5 e a negativa em
série com um resistor de 220 ohms no GND.
Deixamos ligado o botão/ interruptor amarelo, com um lado no pino 4 e o outro no GND.
Vamos ligar um LED vermelho, conectado com a parte positiva no pino 3 e a negativa em
série com um resistor de 220 ohms no GND.
Vamos ligar um botão/ interruptor vermelho, com um lado no pino 2 e o outro no GND.
*/
#define pinoBuzzer 10 // se você ligou em outro pino, ajuste aqui
#define LEDAzul 9 // parte positiva no pino 9 e a negativa em série com um resistor de
220 ohms no GND
#define botaoAzul 8 // um lado no pino 8 e o outro no GND
#define LEDVerde 7 // parte positiva no pino 7 e a negativa em série com um resistor de
220 ohms no GND
#define botaoVerde 6 // um lado no pino 6 e o outro no GND
#define LEDAmarelo 5 // parte positiva no pino 5 e a negativa em série com um resistor de
220 ohms no GND
#define botaoAmarelo 4 // um lado no pino 4 e o outro no GND
#define LEDVermelho 3 // parte positiva no pino 3 e a negativa em série com um resistor
de 220 ohms no GND
#define botaoVermelho 2 // um lado no pino 2 e o outro no GND

void setup() {
  pinMode(pinoBuzzer, OUTPUT); // é uma saída
  pinMode(LEDAzul, OUTPUT); // é uma saída
  pinMode(botaoAzul, INPUT_PULLUP); // entrada que estará alta se não estiver pressionado
  o botão
  pinMode(LEDVerde, OUTPUT); // é uma saída
```



```

    pinMode(botaoVerde, INPUT_PULLUP); // entrada que estará alta se não estiver
pressionado o botão
    pinMode(LEDAmarelo, OUTPUT); // é uma saída
    pinMode(botaoAmarelo, INPUT_PULLUP); // entrada que estará alta se não estiver
pressionado o botão
    pinMode(LEDVermelho, OUTPUT); // é uma saída
    pinMode(botaoVermelho, INPUT_PULLUP); // entrada que estará alta se não estiver
pressionado o botão
    somPerdeu(); //executa a função 'somPerdeu'
    delay(800); // aguarda 800 ms
    somGanhou(); //executa a função 'somGanhou'
}

```

```

void loop() {
    if(digitalRead(botaoAzul) == LOW){
        // se o botão foi pressionado
        digitalWrite(LED Azul, HIGH); // acende o LED azul
        tone(pinoBuzzer, 494, 200); // som de 494 Hz
    }
    else{
        // se o botão não estiver pressionado
        digitalWrite(LED Azul, LOW); // apaga o LED azul
        noTone(pinoBuzzer); // sem som
    }
    if(digitalRead(botaoVerde) == LOW){
        // se o botão foi pressionado
        digitalWrite(LED Verde, HIGH); // acende o LED verde
        tone(pinoBuzzer, 392, 200); // som de 392 Hz
    }
    else{
        // se o botão não estiver pressionado
        digitalWrite(LED Verde, LOW); // apaga o LED verde
        noTone(pinoBuzzer); // sem som
    }
    if(digitalRead(botaoAmarelo) == LOW){
        // se o botão foi pressionado
        digitalWrite(LED Amarelo, HIGH); // acende o LED amarelo
        tone(pinoBuzzer, 330, 200); // som de 330 Hz
    }
    else{
        // se o botão não estiver pressionado
        digitalWrite(LED Amarelo, LOW); // apaga o LED vermelho
        noTone(pinoBuzzer); // sem som
    }
    if(digitalRead(botaoVermelho) == LOW){
        // se o botão foi pressionado
        digitalWrite(LED Vermelho, HIGH); // acende o LED vermelho
        tone(pinoBuzzer, 330, 200); // som de 330 Hz
    }
    else{
        // se o botão não estiver pressionado
        digitalWrite(LED Vermelho, LOW); // apaga o LED amarelo
        noTone(pinoBuzzer); // sem som
    }
}

```

```

void somPerdeu(){
    //do código da SparkFun
    tone(pinoBuzzer, 130, 250); //E6
    delay(275);
    tone(pinoBuzzer, 73, 250); //G6
    delay(275);
    tone(pinoBuzzer, 65, 150); //E7
    delay(175);
    tone(pinoBuzzer, 98, 500); //C7
    delay(500);
}

```

```

void somGanhou(){

```

```

//do código da SparkFun
tone(pinoBuzzer, 1318, 150); //E6
delay(175);
tone(pinoBuzzer, 1567, 150); //G6
delay(175);
tone(pinoBuzzer, 2637, 150); //E7
delay(175);
tone(pinoBuzzer, 2093, 150); //C7
delay(175);
tone(pinoBuzzer, 2349, 150); //D7
delay(175);
tone(pinoBuzzer, 3135, 500); //G7
delay(500);
}

```

Até aqui o que fizemos foi realizar a ligação de todo o *hardware* necessário (as peças) e testá-lo. Sabemos que tudo está funcionando, então podemos prosseguir.

Não vamos mudar nada nas ligações, somente no código - Simon_6_IniciodoCodigo:

```

/*
Seu hardware está ok, agora vamos programar.
Agora quando os sons de ganhou e perdeu tocarem, os leds acompanham.
*/
#define pinoBuzzer 10 // se você ligou em outro pino, ajuste aqui
#define LEDAzul 9 // parte positiva no pino 9 e a negativa em série com um resistor de
220 ohms no GND
#define botaoAzul 8 // um lado no pino 8 e o outro no GND
#define LEDVerde 7 // parte positiva no pino 7 e a negativa em série com um resistor de
220 ohms no GND
#define botaoVerde 6 // um lado no pino 6 e o outro no GND
#define LEDAmarelo 5 // parte positiva no pino 5 e a negativa em série com um resistor de
220 ohms no GND
#define botaoAmarelo 4 // um lado no pino 4 e o outro no GND
#define LEDVermelho 3 // parte positiva no pino 3 e a negativa em série com um resistor
de 220 ohms no GND
#define botaoVermelho 2 // um lado no pino 2 e o outro no GND

void setup() {
  pinMode(pinoBuzzer, OUTPUT); // é uma saída
  pinMode(LEDAzul, OUTPUT); // é uma saída
  pinMode(botaoAzul, INPUT_PULLUP); // entrada que estará alta se não estiver pressionado
o botão
  pinMode(LEDVerde, OUTPUT); // é uma saída
  pinMode(botaoVerde, INPUT_PULLUP); // entrada que estará alta se não estiver
pressionado o botão
  pinMode(LEDAmarelo, OUTPUT); // é uma saída
  pinMode(botaoAmarelo, INPUT_PULLUP); // entrada que estará alta se não estiver
pressionado o botão
  pinMode(LEDVermelho, OUTPUT); // é uma saída
  pinMode(botaoVermelho, INPUT_PULLUP); // entrada que estará alta se não estiver
pressionado o botão
  somPerdeu(); //executa a função 'somPerdeu'
  delay(800); // aguarda 800 ms
  somGanhou(); //executa a função 'somGanhou'
}

void loop() {
  if(digitalRead(botaoAzul) == LOW){
    // se o botão foi pressionado
    digitalWrite(LEDAzul, HIGH); // acende o LED azul
    tone(pinoBuzzer, 494, 200); // som de 494 Hz
  }
  else{
    // se o botão não estiver pressionado
    digitalWrite(LEDAzul, LOW); // apaga o LED azul
    noTone(pinoBuzzer); // sem som
  }
}

```

```

}
if(digitalRead(botaoVerde) == LOW){
    // se o botão foi pressionado
    digitalWrite(LEDVerde, HIGH); // acende o LED verde
    tone(pinoBuzzer, 392, 200); // som de 392 Hz
}
else{
    // se o botão não estiver pressionado
    digitalWrite(LEDVerde, LOW); // apaga o LED verde
    noTone(pinoBuzzer); // sem som
}
if(digitalRead(botaoAmarelo) == LOW){
    // se o botão foi pressionado
    digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
    tone(pinoBuzzer, 330, 200); // som de 330 Hz
}
else{
    // se o botão não estiver pressionado
    digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
    noTone(pinoBuzzer); // sem som
}
if(digitalRead(botaoVermelho) == LOW){
    // se o botão foi pressionado
    digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
    tone(pinoBuzzer, 330, 200); // som de 330 Hz
}
else{
    // se o botão não estiver pressionado
    digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
    noTone(pinoBuzzer); // sem som
}
}

void somPerdeu(){
    //do código da SparkFun
    digitalWrite(LED Azul, HIGH); // acende o LED azul
    tone(pinoBuzzer, 130, 250); //E6
    delay(275);
    digitalWrite(LED Azul, LOW); // apaga o LED azul
    digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
    tone(pinoBuzzer, 73, 250); //G6
    delay(275);
    digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
    digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
    tone(pinoBuzzer, 65, 150); //E7
    delay(175);
    digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
    digitalWrite(LEDVerde, HIGH); // acende o LED verde
    tone(pinoBuzzer, 98, 500); //C7
    delay(500);
    digitalWrite(LEDVerde, LOW); // apaga o LED verde
}

void somGanhou(){
    //do código da SparkFun
    digitalWrite(LED Azul, HIGH); // acende o LED azul
    tone(pinoBuzzer, 1318, 150); //E6
    delay(175);
    digitalWrite(LED Azul, LOW); // apaga o LED azul
    digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
    tone(pinoBuzzer, 1567, 150); //G6
    delay(175);
    digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
    digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
    tone(pinoBuzzer, 2637, 150); //E7
    delay(175);
    digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
    digitalWrite(LEDVerde, HIGH); // acende o LED verde
    tone(pinoBuzzer, 2093, 150); //C7

```

```

delay(175);
digitalWrite(LEDazul, HIGH); // acende o LED azul
digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
tone(pinoBuzzer, 2349, 150); //D7
delay(175);
tone(pinoBuzzer, 3135, 500); //G7
digitalWrite(LEDazul, LOW); // apaga o LED azul
digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
digitalWrite(LEDVerde, LOW); // apaga o LED verde
delay(500);
}

```

Teste !

Depois de testar, vamos prosseguir colocando a geração de um número aleatório, que produzirá o acendimento dos LEDs em sequências diferentes - Simon_7_CodigoComAleatorio:

```

/*
O código gera um número aleatório entre zero e três.
- se for zero executa a sequência para o LED vermelho;
- se for um executa a sequência para o LED amarelo;
- se for dois executa a sequência para o LED verde;
- se for três executa a sequência para o LED azul;

Após gerar o número ele lê qual botão foi pressionado e compara com o número aleatório:
- se forem iguais, toca o som de ganhou
- se forem diferentes ou um botão o botão não foi pressionado no tempo correto, toca o som de perdeu

O código só testa um botão e reinicia. O próximo passo é fazer uma sequência..., mas isto é o próximo programa.
*/
#define pinoBuzzer 10 // se você ligou em outro pino, ajuste aqui
#define LEDazul 9 // parte positiva no pino 9 e a negativa em série com um resistor de 220 ohms no GND
#define botaoAzul 8 // um lado no pino 8 e o outro no GND
#define LEDVerde 7 // parte positiva no pino 7 e a negativa em série com um resistor de 220 ohms no GND
#define botaoVerde 6 // um lado no pino 6 e o outro no GND
#define LEDAmarelo 5 // parte positiva no pino 5 e a negativa em série com um resistor de 220 ohms no GND
#define botaoAmarelo 4 // um lado no pino 4 e o outro no GND
#define LEDVermelho 3 // parte positiva no pino 3 e a negativa em série com um resistor de 220 ohms no GND
#define botaoVermelho 2 // um lado no pino 2 e o outro no GND

int sorteioAleatorio = 0; // variável para controlar o número aleatório
int velocidade = 500; // usa 500 ms como tempo para controle da velocidade de exibição
int leitura = 4; // lê o retorno do botão pressionado; se nenhum for pressionado o valor é 4

void setup() {
  pinMode(pinoBuzzer, OUTPUT); // é uma saída
  pinMode(LEDazul, OUTPUT); // é uma saída
  pinMode(botaoAzul, INPUT_PULLUP); // entrada que estará alta se não estiver pressionado o botão
  pinMode(LEDVerde, OUTPUT); // é uma saída
  pinMode(botaoVerde, INPUT_PULLUP); // entrada que estará alta se não estiver pressionado o botão
  pinMode(LEDAmarelo, OUTPUT); // é uma saída
  pinMode(botaoAmarelo, INPUT_PULLUP); // entrada que estará alta se não estiver pressionado o botão
  pinMode(LEDVermelho, OUTPUT); // é uma saída
  pinMode(botaoVermelho, INPUT_PULLUP); // entrada que estará alta se não estiver pressionado o botão
}

```

```
    randomSeed(A0); // inicializa o gerador de números aleatórios com uma 'semente'
    qualquer - ruído na porta A0
}
```

```
void loop() {
    sorteioAleatorio = random(4); // gera um número aleatório entre 0 e 3
    switch (sorteioAleatorio){
        case 0:{
            digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
            tone(pinoBuzzer, 262, 200); // som de 262 Hz
            delay(velocidade);
            digitalWrite(LEDVermelho, LOW); // apaga o LED vermelho
            noTone(pinoBuzzer); // sem som
            break;
        }
        case 1:{
            digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
            tone(pinoBuzzer, 330, 200); // som de 330 Hz
            delay(velocidade);
            digitalWrite(LEDAmarelo, LOW); // apaga o LED amarelo
            noTone(pinoBuzzer); // sem som
            break;
        }
        case 2:{
            digitalWrite(LEDVerde, HIGH); // acende o LED verde
            tone(pinoBuzzer, 392, 200); // som de 392 Hz
            delay(velocidade);
            digitalWrite(LEDVerde, LOW); // apaga o LED verde
            noTone(pinoBuzzer); // sem som
            break;
        }
        case 3:{
            digitalWrite(LEDAzul, HIGH); // acende o LED azul
            tone(pinoBuzzer, 494, 200); // som de 494 Hz
            delay(velocidade);
            digitalWrite(LEDAzul, LOW); // apaga o LED azul
            noTone(pinoBuzzer); // sem som
            break;
        }
    }
    delay(400); // aguarda 400ms após a apresentação do led
    leitura = qualBotao(); // lê o botão pressionado
    if (sorteioAleatorio == leitura){ // se for igual ao sorteado
        somGanhou(); // toca o som de ganhou
    }
    else{ // se não for o mesmo botão ou não for pressionado um botão
        somPerdeu(); // toca o som de perdeu
    }
    delay(1000);
}
```

```
void somPerdeu(){
    //do código da SparkFun
    digitalWrite(LEDAzul, HIGH); // acende o LED azul
    tone(pinoBuzzer, 130, 250); //E6
    delay(275);
    digitalWrite(LEDAzul, LOW); // apaga o LED azul
    digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
    tone(pinoBuzzer, 73, 250); //G6
    delay(275);
    digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
    digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
    tone(pinoBuzzer, 65, 150); //E7
    delay(175);
    digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
    digitalWrite(LEDVerde, HIGH); // acende o LED verde
    tone(pinoBuzzer, 98, 500); //C7
    delay(500);
    digitalWrite(LEDVerde, LOW); // apaga o LED verde
}
```

```

}

void somGanhou(){
    //do código da SparkFun
    digitalWrite(LEDAzul, HIGH); // acende o LED azul
    tone(pinoBuzzer, 1318, 150); //E6
    delay(175);
    digitalWrite(LEDAzul, LOW); // apaga o LED azul
    digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
    tone(pinoBuzzer, 1567, 150); //G6
    delay(175);
    digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
    digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
    tone(pinoBuzzer, 2637, 150); //E7
    delay(175);
    digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
    digitalWrite(LEDVerde, HIGH); // acende o LED verde
    tone(pinoBuzzer, 2093, 150); //C7
    delay(175);
    digitalWrite(LEDAzul, HIGH); // acende o LED azul
    digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
    digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
    tone(pinoBuzzer, 2349, 150); //D7
    delay(175);
    tone(pinoBuzzer, 3135, 500); //G7
    digitalWrite(LEDAzul, LOW); // apaga o LED azul
    digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
    digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
    digitalWrite(LEDVerde, LOW); // apaga o LED verde
    delay(500);
}

int qualBotao() {
    // deixe o botão pressionado alguns instantes
    if (digitalRead(botaoVermelho) == LOW) {
        return 0; // se pressionou o botão vermelho, retorna 0
    } else if (digitalRead(botaoAmarelo) == LOW) {
        return 1; // se pressionou o botão amarelo, retorna 1
    } else if (digitalRead(botaoVerde) == LOW) {
        return 2; // se pressionou o botão verde, retorna 2
    } else if (digitalRead(botaoAzul) == LOW) {
        return 3; // se pressionou o botão azul, retorna 3
    } else {
        return 4; // nenhum botão válido pressionado
    }
}

```

Vamos melhorar - Simon_8_CodigoComSequenciaAleatoria:

```

/*
Agora o código gera uma sequência aleatória com 16 possibilidades e depois o jogador tem
que repetir a sequência.
A sequência gerada está sendo exibida.
Aguarda 3 segundos pela ação do jogador.
A sequência é mostrada passo a passo - o usual é mostrar a sequência toda, mas isto
faremos no próximo código.
*/
#define pinoBuzzer 10 // se você ligou em outro pino, ajuste aqui
#define LEDAzul 9 // parte positiva no pino 9 e a negativa em série com um resistor de
220 ohms no GND
#define botaoAzul 8 // um lado no pino 8 e o outro no GND
#define LEDVerde 7 // parte positiva no pino 7 e a negativa em série com um resistor de
220 ohms no GND
#define botaoVerde 6 // um lado no pino 6 e o outro no GND
#define LEDAmarelo 5 // parte positiva no pino 5 e a negativa em série com um resistor de
220 ohms no GND
#define botaoAmarelo 4 // um lado no pino 4 e o outro no GND

```



```

#define LEDVermelho 3 // parte positiva no pino 3 e a negativa em série com um resistor
de 220 ohms no GND
#define botaoVermelho 2 // um lado no pino 2 e o outro no GND
#define tamanhoSequencia 16 // a sequência terá 16 valores, se quiser mais aumente o
número

int sorteioAleatorio = 0; // variável para controlar o número aleatório
int velocidade = 500; // usa 500 ms como tempo para controle da velocidade de exibição
int leitura = 4; // lê o retorno do botão pressionado; se nenhum for pressionado o valor
é 4
int sequencia[tamanhoSequencia]; // a sequência terá tamanhoSequencia valores, se quiser
mais aumente o número
long int tempoReacao; // para medir o tempo de resposta do jogador

void setup() {
    pinMode(pinoBuzzer, OUTPUT); // é uma saída
    pinMode(LEDAzul, OUTPUT); // é uma saída
    pinMode(botaoAzul, INPUT_PULLUP); // entrada que estará alta se não estiver pressionado
o botão
    pinMode(LEDVerde, OUTPUT); // é uma saída
    pinMode(botaoVerde, INPUT_PULLUP); // entrada que estará alta se não estiver
pressionado o botão
    pinMode(LEDAmarelo, OUTPUT); // é uma saída
    pinMode(botaoAmarelo, INPUT_PULLUP); // entrada que estará alta se não estiver
pressionado o botão
    pinMode(LEDVermelho, OUTPUT); // é uma saída
    pinMode(botaoVermelho, INPUT_PULLUP); // entrada que estará alta se não estiver
pressionado o botão
    randomSeed(A0); // inicializa o gerador de números aleatórios com uma 'semente'
qualquer - ruído na porta A0
}

void loop() {
    sorteia();
    delay(1000);
    if (jogar()){ // se retornou 1
        somGanhou(); // toca o som de ganhou
    }
    else{ // se não for o mesmo botão ou não for pressionado um botão
        somPerdeu(); // toca o som de perdeu
    }
    delay(1000); // aguarda 1 segundo antes de recomeçar
}

void somPerdeu(){
    //do código da SparkFun
    digitalWrite(LEDAzul, HIGH); // acende o LED azul
    tone(pinoBuzzer, 130, 250); //E6
    delay(275);
    digitalWrite(LEDAzul, LOW); // apaga o LED azul
    digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
    tone(pinoBuzzer, 73, 250); //G6
    delay(275);
    digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
    digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
    tone(pinoBuzzer, 65, 150); //E7
    delay(175);
    digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
    digitalWrite(LEDVerde, HIGH); // acende o LED verde
    tone(pinoBuzzer, 98, 500); //C7
    delay(500);
    digitalWrite(LEDVerde, LOW); // apaga o LED verde
}

void somGanhou(){
    //do código da SparkFun
    digitalWrite(LEDAzul, HIGH); // acende o LED azul
    tone(pinoBuzzer, 1318, 150); //E6
    delay(175);

```

```

digitalWrite(LEDAzul, LOW); // apaga o LED azul
digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
tone(pinoBuzzer, 1567, 150); //G6
delay(175);
digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
tone(pinoBuzzer, 2637, 150); //E7
delay(175);
digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
digitalWrite(LEDVerde, HIGH); // acende o LED verde
tone(pinoBuzzer, 2093, 150); //C7
delay(175);
digitalWrite(LEDAzul, HIGH); // acende o LED azul
digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
tone(pinoBuzzer, 2349, 150); //D7
delay(175);
tone(pinoBuzzer, 3135, 500); //G7
digitalWrite(LEDAzul, LOW); // apaga o LED azul
digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
digitalWrite(LEDVerde, LOW); // apaga o LED verde
delay(500);
}

int qualBotao() {
    // deixe o botão pressionado alguns instantes
    if (digitalRead(botaoVermelho) == LOW) {
        return 0; // se pressionou o botão vermelho, retorna 0
    } else if (digitalRead(botaoAmarelo) == LOW) {
        return 1; // se pressionou o botão amarelo, retorna 1
    } else if (digitalRead(botaoVerde) == LOW) {
        return 2; // se pressionou o botão verde, retorna 2
    } else if (digitalRead(botaoAzul) == LOW) {
        return 3; // se pressionou o botão azul, retorna 3
    } else {
        return 4; // nenhum botão válido pressionado
    }
}

void sorteia(){
    // gera a sequência
    for (int posicao = 0; posicao < tamanhoSequencia; posicao++){
        sequencia[posicao] = random(4); // gera um número aleatório entre 0 e 3 e coloca na
        posição da sequência
    }
    digitalWrite(LEDAzul, HIGH); // acende o LED azul
    digitalWrite(LEDVermelho, HIGH); // acende o LED amarelo
    digitalWrite(LEDAmarelo, HIGH); // acende o LED vermelho
    digitalWrite(LEDVerde, HIGH); // acende o LED verde
    tone(pinoBuzzer, 1318, 150); //E6
    delay(175);
    tone(pinoBuzzer, 1567, 150); //G6
    delay(175);
    tone(pinoBuzzer, 2637, 150); //E7
    delay(175);
    tone(pinoBuzzer, 2093, 150); //C7
    delay(175);
    tone(pinoBuzzer, 2349, 150); //D7
    delay(175);
    tone(pinoBuzzer, 3135, 500); //G7
    delay(500);
    digitalWrite(LEDAzul, LOW); // apaga o LED azul
    digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
    digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
    digitalWrite(LEDVerde, LOW); // apaga o LED verde
}

int jogar(){

```

```

// aguarda o jogador repetir a sequência
for (int posicao = 0; posicao < tamanhoSequencia; posicao++){
switch (sequencia[posicao]){
    case 0:{
        digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
        tone(pinoBuzzer, 262, 200); // som de 262 Hz
        delay(velocidade);
        digitalWrite(LEDVermelho, LOW); // apaga o LED vermelho
        noTone(pinoBuzzer); // sem som
        break;
    }
    case 1:{
        digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
        tone(pinoBuzzer, 330, 200); // som de 330 Hz
        delay(velocidade);
        digitalWrite(LEDAmarelo, LOW); // apaga o LED amarelo
        noTone(pinoBuzzer); // sem som
        break;
    }
    case 2:{
        digitalWrite(LEDVerde, HIGH); // acende o LED verde
        tone(pinoBuzzer, 392, 200); // som de 392 Hz
        delay(velocidade);
        digitalWrite(LEDVerde, LOW); // apaga o LED verde
        noTone(pinoBuzzer); // sem som
        break;
    }
    case 3:{
        digitalWrite(LEDAzul, HIGH); // acende o LED azul
        tone(pinoBuzzer, 494, 200); // som de 494 Hz
        delay(velocidade);
        digitalWrite(LEDAzul, LOW); // apaga o LED azul
        noTone(pinoBuzzer); // sem som
        break;
    }
}
tempoReacao = millis();
while( millis() - tempoReacao < 3000){ // espera até 3 segundos
    leitura = qualBotao();
    if (leitura < 4){
        break;
    }
}
if (leitura == sequencia[posicao]){
    delay(400);
}
else{
    return 0;
}
}
return 1;
}
}

```

Agora você já tem algo funcionando. Teste !

O brinquedo Simon não era assim. Ele mostrava a sequência e depois você tinha que repetir (ele ia aumentando a sequência. Vamos fazer ele mostrar uma sequência completa para o jogador ter que repetir. Coloque o seguinte código - Simon_9_SequenciaTotal:

```

/*
Agora o código gera uma sequência aleatória com 16 possibilidades e depois o jogador tem
que repetir a sequência.
A sequência gerada está sendo exibida.
Aguarda 3 segundos pela ação do jogador.
A sequência é mostrada passo a passo - o usual é mostrar a sequência toda, mas isto
faremos no próximo código.
*/

```

```

#define pinoBuzzer 10 // se você ligou em outro pino, ajuste aqui
#define LEDAzul 9 // parte positiva no pino 9 e a negativa em série com um resistor de
220 ohms no GND
#define botaoAzul 8 // um lado no pino 8 e o outro no GND
#define LEDVerde 7 // parte positiva no pino 7 e a negativa em série com um resistor de
220 ohms no GND
#define botaoVerde 6 // um lado no pino 6 e o outro no GND
#define LEDAmarelo 5 // parte positiva no pino 5 e a negativa em série com um resistor de
220 ohms no GND
#define botaoAmarelo 4 // um lado no pino 4 e o outro no GND
#define LEDVermelho 3 // parte positiva no pino 3 e a negativa em série com um resistor
de 220 ohms no GND
#define botaoVermelho 2 // um lado no pino 2 e o outro no GND
#define tamanhoSequencia 16 // a sequência terá 16 valores, se quiser mais aumente o
número

int sorteioAleatorio = 0; // variável para controlar o número aleatório
int velocidade = 500; // usa 500 ms como tempo para controle da velocidade de exibição
int leitura = 4; // lê o retorno do botão pressionado; se nenhum for pressionado o valor
é 4
int sequencia[tamanhoSequencia]; // a sequência terá tamanhoSequencia valores, se quiser
mais aumente o número
long int tempoReacao; // para medir o tempo de resposta do jogador
bool resultado = true; // saber se ganhou

void setup() {
    pinMode(pinoBuzzer, OUTPUT); // é uma saída
    pinMode(LEDAzul, OUTPUT); // é uma saída
    pinMode(botaoAzul, INPUT_PULLUP); // entrada que estará alta se não estiver pressionado
o botão
    pinMode(LEDVerde, OUTPUT); // é uma saída
    pinMode(botaoVerde, INPUT_PULLUP); // entrada que estará alta se não estiver
pressionado o botão
    pinMode(LEDAmarelo, OUTPUT); // é uma saída
    pinMode(botaoAmarelo, INPUT_PULLUP); // entrada que estará alta se não estiver
pressionado o botão
    pinMode(LEDVermelho, OUTPUT); // é uma saída
    pinMode(botaoVermelho, INPUT_PULLUP); // entrada que estará alta se não estiver
pressionado o botão
    randomSeed(A0); // inicializa o gerador de números aleatórios com uma 'semente'
qualquer - ruído na porta A0
}

void loop() {
    sorteia();
    resultado = true;
    jogar();
    if (resultado){
        somGanhou();
    }
    else{
        somPerdeu();
    }
    delay(1000); // aguarda 1 segundo antes de recomeçar
}

void somPerdeu(){
    //do código da SparkFun
    digitalWrite(LEDAzul, HIGH); // acende o LED azul
    tone(pinoBuzzer, 130, 250); //E6
    delay(275);
    digitalWrite(LEDAzul, LOW); // apaga o LED azul
    digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
    tone(pinoBuzzer, 73, 250); //G6
    delay(275);
    digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
    digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
    tone(pinoBuzzer, 65, 150); //E7
    delay(175);
}

```

```

digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
digitalWrite(LEDVerde, HIGH); // acende o LED verde
tone(pinoBuzzer, 98, 500);    //C7
delay(500);
digitalWrite(LEDVerde, LOW); // apaga o LED verde
}

void somGanhou(){
    //do código da SparkFun
    digitalWrite(LEDAzul, HIGH); // acende o LED azul
    tone(pinoBuzzer, 1318, 150);    //E6
    delay(175);
    digitalWrite(LEDAzul, LOW); // apaga o LED azul
    digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
    tone(pinoBuzzer, 1567, 150);    //G6
    delay(175);
    digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
    digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
    tone(pinoBuzzer, 2637, 150);    //E7
    delay(175);
    digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
    digitalWrite(LEDVerde, HIGH); // acende o LED verde
    tone(pinoBuzzer, 2093, 150);    //C7
    delay(175);
    digitalWrite(LEDAzul, HIGH); // acende o LED azul
    digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
    digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
    tone(pinoBuzzer, 2349, 150);    //D7
    delay(175);
    tone(pinoBuzzer, 3135, 500);    //G7
    digitalWrite(LEDAzul, LOW); // apaga o LED azul
    digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
    digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
    digitalWrite(LEDVerde, LOW); // apaga o LED verde
    delay(500);
}

int qualBotao() {
    // deixe o botão pressionado alguns instantes
    if (digitalRead(botaoVermelho) == LOW) {
        return 0; // se pressionou o botão vermelho, retorna 0
    } else if (digitalRead(botaoAmarelo) == LOW) {
        return 1; // se pressionou o botão amarelo, retorna 1
    } else if (digitalRead(botaoVerde) == LOW) {
        return 2; // se pressionou o botão verde, retorna 2
    } else if (digitalRead(botaoAzul) == LOW) {
        return 3; // se pressionou o botão azul, retorna 3
    } else {
        return 4; // nenhum botão válido pressionado
    }
}

void sorteia(){
    // gera a sequência
    for (int posicao = 0; posicao < tamanhoSequencia; posicao++){
        sequencia[posicao] = random(4); // gera um número aleatório entre 0 e 3 e coloca na
        posição da sequência
    }
    digitalWrite(LEDAzul, HIGH); // acende o LED azul
    digitalWrite(LEDVermelho, HIGH); // acende o LED amarelo
    digitalWrite(LEDAmarelo, HIGH); // acende o LED vermelho
    digitalWrite(LEDVerde, HIGH); // acende o LED verde
    tone(pinoBuzzer, 1318, 150);    //E6
    delay(175);
    tone(pinoBuzzer, 1567, 150);    //G6
    delay(175);
    tone(pinoBuzzer, 2637, 150);    //E7
    delay(175);
    tone(pinoBuzzer, 2093, 150);    //C7
}

```

```

delay(175);
tone(pinoBuzzer, 2349, 150);    //D7
delay(175);
tone(pinoBuzzer, 3135, 500);    //G7
delay(500);
digitalWrite(LEDAzul, LOW); // apaga o LED azul
digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
digitalWrite(LEDVerde, LOW); // apaga o LED verde
delay(1000); // aguarda 1 segundo
// exhibe o sorteio
for (int posicao = 0; posicao < tamanhoSequencia; posicao++){
    mostrarJogada(sequencia[posicao]);
}
}

int mostrarJogada(int qual){
    switch (qual){
        case 0:{
            digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
            tone(pinoBuzzer, 262, 200); // som de 262 Hz
            delay(velocidade);
            digitalWrite(LEDVermelho, LOW); // apaga o LED vermelho
            noTone(pinoBuzzer); // sem som
            break;
        }
        case 1:{
            digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
            tone(pinoBuzzer, 330, 200); // som de 330 Hz
            delay(velocidade);
            digitalWrite(LEDAmarelo, LOW); // apaga o LED amarelo
            noTone(pinoBuzzer); // sem som
            break;
        }
        case 2:{
            digitalWrite(LEDVerde, HIGH); // acende o LED verde
            tone(pinoBuzzer, 392, 200); // som de 392 Hz
            delay(velocidade);
            digitalWrite(LEDVerde, LOW); // apaga o LED verde
            noTone(pinoBuzzer); // sem som
            break;
        }
        case 3:{
            digitalWrite(LEDAzul, HIGH); // acende o LED azul
            tone(pinoBuzzer, 494, 200); // som de 494 Hz
            delay(velocidade);
            digitalWrite(LEDAzul, LOW); // apaga o LED azul
            noTone(pinoBuzzer); // sem som
            break;
        }
    }
}

void jogar(){
    // jogar
    for (int posicao = 0; posicao < tamanhoSequencia; posicao++){
        leitura = 4;
        tempoReacao = millis();
        do {
            leitura = qualBotao();
        } while(leitura == 4 && ((millis() - tempoReacao) < 3000)); // espera até 3 segundos
        pela tecla
        mostrarJogada(leitura);
        if (leitura != sequencia[posicao]){
            resultado = false;
            break;
        }
        delay(400);
    }
}

```



```
}
```

E então?

Perceba que agora fomos ao outro extremo: o Simon mostra toda a sequência e você tem que lembrar dela e repetir sem enganos. Fica bem difícil.

Como poderíamos melhorar? Que tal se o Simon fosse mostrando a sequência com incrementos de um LED? Por exemplo, primeiro mostra só um LED, depois uma sequência de dois LEDs, se você acertar ele mostra uma sequência com três, com quatro, etc, até terminar. Não seria mais tranquilo?

O código que segue permite uma jogada mais tranquila e o tempo que o jogador tem para apertar um botão foi transformado em uma diretiva no início do programa, você pode aumentar (o valor do código é de 3 segundos).

Código final - Simon_10_funcionamentoSuave:

```
/*
Agora o código gera uma sequência aleatória com 10 possibilidades e depois o jogador tem
que repetir a sequência.
A sequência gerada está sendo exibida.
Aguarda 3 segundos pela ação do jogador.
*/
#define pinoBuzzer 10 // se você ligou em outro pino, ajuste aqui
#define LEDAzul 9 // parte positiva no pino 9 e a negativa em série com um resistor de
220 ohms no GND
#define botaoAzul 8 // um lado no pino 8 e o outro no GND
#define LEDVerde 7 // parte positiva no pino 7 e a negativa em série com um resistor de
220 ohms no GND
#define botaoVerde 6 // um lado no pino 6 e o outro no GND
#define LEDAmarelo 5 // parte positiva no pino 5 e a negativa em série com um resistor de
220 ohms no GND
#define botaoAmarelo 4 // um lado no pino 4 e o outro no GND
#define LEDVermelho 3 // parte positiva no pino 3 e a negativa em série com um resistor
de 220 ohms no GND
#define botaoVermelho 2 // um lado no pino 2 e o outro no GND
#define tamanhoSequencia 10 // a sequência terá 10 valores

int sorteioAleatorio = 0; // variável para controlar o número aleatório
int velocidade = 500; // usa 500 ms como tempo para controle da velocidade de exibição
int leitura = 4; // lê o retorno do botão pressionado; se nenhum for pressionado o valor
é 4
int sequencia[tamanhoSequencia]; // a sequência terá tamanhoSequencia valores, se quiser
mais aumente o número
long int tempoReacao; // para medir o tempo de resposta do jogador
bool resultado = true; // saber se ganhou
int tempoLimite = 3000; // quantos segundos o jogador tem para pressionar o botão?

void somPerdeu(){
    //do código da SparkFun
    digitalWrite(LEDazul, HIGH); // acende o LED azul
    tone(pinoBuzzer, 130, 250); //E6
    delay(275);
    digitalWrite(LEDazul, LOW); // apaga o LED azul
    digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
    tone(pinoBuzzer, 73, 250); //G6
    delay(275);
    digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
    digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
    tone(pinoBuzzer, 65, 150); //E7
    delay(175);
    digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
    digitalWrite(LEDVerde, HIGH); // acende o LED verde
    tone(pinoBuzzer, 98, 500); //C7
    delay(500);
    digitalWrite(LEDVerde, LOW); // apaga o LED verde
```

```

}

void somGanhou(){
    //do código da SparkFun
    digitalWrite(LEDAzul, HIGH); // acende o LED azul
    tone(pinoBuzzer, 1318, 150); //E6
    delay(175);
    digitalWrite(LEDAzul, LOW); // apaga o LED azul
    digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
    tone(pinoBuzzer, 1567, 150); //G6
    delay(175);
    digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
    digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
    tone(pinoBuzzer, 2637, 150); //E7
    delay(175);
    digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
    digitalWrite(LEDVerde, HIGH); // acende o LED verde
    tone(pinoBuzzer, 2093, 150); //C7
    delay(175);
    digitalWrite(LEDAzul, HIGH); // acende o LED azul
    digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
    digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
    tone(pinoBuzzer, 2349, 150); //D7
    delay(175);
    tone(pinoBuzzer, 3135, 500); //G7
    digitalWrite(LEDAzul, LOW); // apaga o LED azul
    digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
    digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
    digitalWrite(LEDVerde, LOW); // apaga o LED verde
    delay(500);
}

int qualBotao() {
    // deixe o botão pressionado alguns instantes
    if (digitalRead(botaoVermelho) == LOW) {
        return 0; // se pressionou o botão vermelho, retorna 0
    } else if (digitalRead(botaoAmarelo) == LOW) {
        return 1; // se pressionou o botão amarelo, retorna 1
    } else if (digitalRead(botaoVerde) == LOW) {
        return 2; // se pressionou o botão verde, retorna 2
    } else if (digitalRead(botaoAzul) == LOW) {
        return 3; // se pressionou o botão azul, retorna 3
    } else {
        return 4; // nenhum botão válido pressionado
    }
}

void sorteia(){
    // gera a sequência
    for (int posicao = 0; posicao < tamanhoSequencia; posicao++){
        sequencia[posicao] = random(4); // gera um número aleatório entre 0 e 3 e coloca na
        posição da sequência
    }
    digitalWrite(LEDAzul, HIGH); // acende o LED azul
    digitalWrite(LEDVermelho, HIGH); // acende o LED amarelo
    digitalWrite(LEDAmarelo, HIGH); // acende o LED vermelho
    digitalWrite(LEDVerde, HIGH); // acende o LED verde
    tone(pinoBuzzer, 1318, 150); //E6
    delay(175);
    tone(pinoBuzzer, 1567, 150); //G6
    delay(175);
    tone(pinoBuzzer, 2637, 150); //E7
    delay(175);
    tone(pinoBuzzer, 2093, 150); //C7
    delay(175);
    tone(pinoBuzzer, 2349, 150); //D7
    delay(175);
    tone(pinoBuzzer, 3135, 500); //G7
    delay(500);
}

```

```

digitalWrite(LEDAzul, LOW); // apaga o LED azul
digitalWrite(LEDVermelho, LOW); // apaga o LED amarelo
digitalWrite(LEDAmarelo, LOW); // apaga o LED vermelho
digitalWrite(LEDVerde, LOW); // apaga o LED verde
delay(1000); // aguarda 1 segundo
// exibe o sorteio
//for (int posicao = 0; posicao < tamanhoSequencia; posicao++){
//  mostrarJogada(sequencia[posicao]);
//}
}

int mostrarJogada(int qual){
  switch (qual){
    case 0:{
      digitalWrite(LEDVermelho, HIGH); // acende o LED vermelho
      tone(pinoBuzzer, 262, 200); // som de 262 Hz
      delay(velocidade);
      digitalWrite(LEDVermelho, LOW); // apaga o LED vermelho
      noTone(pinoBuzzer); // sem som
      break;
    }
    case 1:{
      digitalWrite(LEDAmarelo, HIGH); // acende o LED amarelo
      tone(pinoBuzzer, 330, 200); // som de 330 Hz
      delay(velocidade);
      digitalWrite(LEDAmarelo, LOW); // apaga o LED amarelo
      noTone(pinoBuzzer); // sem som
      break;
    }
    case 2:{
      digitalWrite(LEDVerde, HIGH); // acende o LED verde
      tone(pinoBuzzer, 392, 200); // som de 392 Hz
      delay(velocidade);
      digitalWrite(LEDVerde, LOW); // apaga o LED verde
      noTone(pinoBuzzer); // sem som
      break;
    }
    case 3:{
      digitalWrite(LEDAzul, HIGH); // acende o LED azul
      tone(pinoBuzzer, 494, 200); // som de 494 Hz
      delay(velocidade);
      digitalWrite(LEDAzul, LOW); // apaga o LED azul
      noTone(pinoBuzzer); // sem som
      break;
    }
  }
}

void jogar(int ateOnde){
  // jogar
  for (int posicao = 0; posicao < ateOnde; posicao++){
    mostrarJogada(sequencia[posicao]);
  }
  for (int posicao = 0; posicao < ateOnde; posicao++){
    //mostrarJogada(sequencia[posicao]);
    leitura = 4;
    tempoReacao = millis();
    do {
      leitura = qualBotao();
    } while(leitura == 4 && ((millis() - tempoReacao) < tempoLimite)); // espera até 3
segundos (tempoLimite) pela tecla
    mostrarJogada(leitura);
    if (leitura != sequencia[posicao]){
      resultado = false;
      break;
    }
    delay(400);
  }
}
}

```

```

void setup() {
    pinMode(pinoBuzzer, OUTPUT); // é uma saída
    pinMode(LEDAzul, OUTPUT); // é uma saída
    pinMode(botaoAzul, INPUT_PULLUP); // entrada que estará alta se não estiver pressionado
o botão
    pinMode(LEDVerde, OUTPUT); // é uma saída
    pinMode(botaoVerde, INPUT_PULLUP); // entrada que estará alta se não estiver
pressionado o botão
    pinMode(LEDAmarelo, OUTPUT); // é uma saída
    pinMode(botaoAmarelo, INPUT_PULLUP); // entrada que estará alta se não estiver
pressionado o botão
    pinMode(LEDVermelho, OUTPUT); // é uma saída
    pinMode(botaoVermelho, INPUT_PULLUP); // entrada que estará alta se não estiver
pressionado o botão
    randomSeed(A0); // inicializa o gerador de números aleatórios com uma 'semente'
qualquer - ruído na porta A0
}

void loop() {
    sorteia();
    for(int seq = 0; seq < tamanhoSequencia; seq++){
        resultado = true;
        jogar(seq);
        if (!resultado){
            somPerdeu();
            break;
        }
    }
    if (resultado){
        somGanhou();
    }
    delay(1000); // aguarda 1 segundo antes de recomeçar
}

```

Boa diversão.