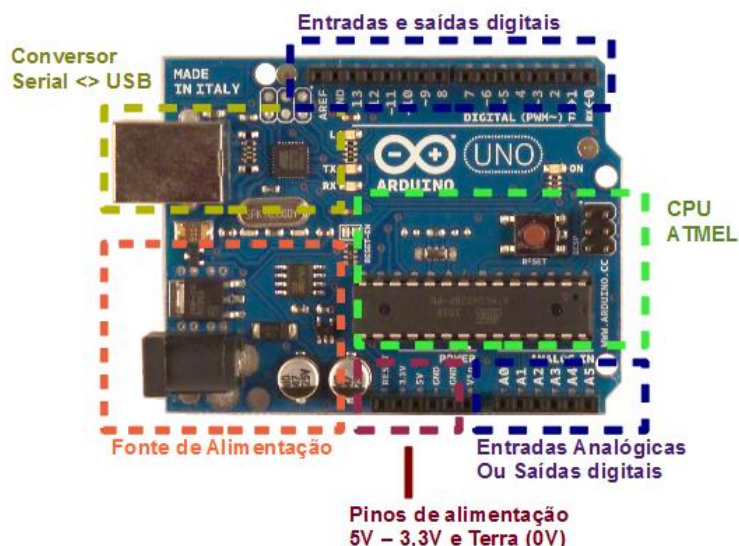


Introdução ao Arduíno

A parte prática da UC de Internet das Coisas irá consistir numa série de exercícios e desafios a realizar utilizando Arduíno (<https://www.arduino.cc/>). A Figura seguinte mostra os principais pinos existentes no dispositivo.



Links para tutoriais de Arduíno:

- <https://www.allaboutcircuits.com/tools/resistor-color-code-calculator/>
- http://www.audioacustica.com.br/exemplos/Valores_Resistores/Calculadora_Ohms_Resistor.html
- <https://www.arduino.cc/en/Tutorial/HomePage?from=Main.Tutorials>
- <https://thetempedia.com/tutorial-hub/arduino/>
- <https://thetempedia.com/tutorials/arduino-ide/>

Simulador Tinkercad

Os exercícios deverão ser resolvidos no simulador Tinkercad (<https://www.tinkercad.com>) para facilitar a partilha. Para isso, é necessário a criação de uma conta gratuita.

Em cada um dos exercícios, poderá basear-se num dos circuitos já construídos. Para isso, nas opções do circuito na página principal, escolha “Duplicar”. Em cima, onde aparece “Cópia de ...”, altere o nome do circuito.

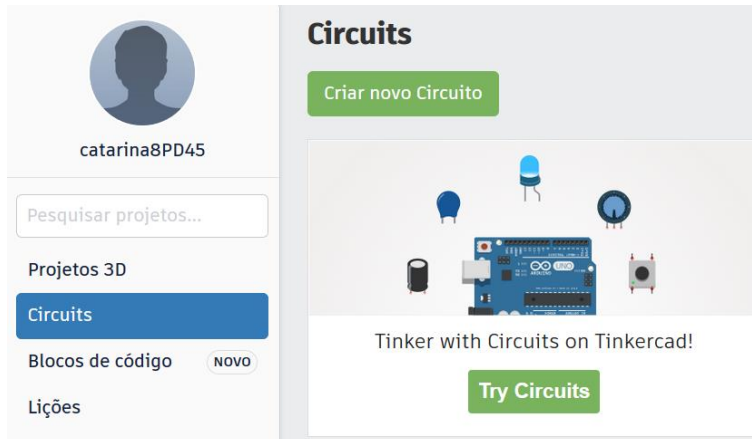
IDE Arduíno

O IDE pode ser obtido a partir de https://www.arduino.cc/download_handler.php e depois instalado. Será configurado na primeira utilização.

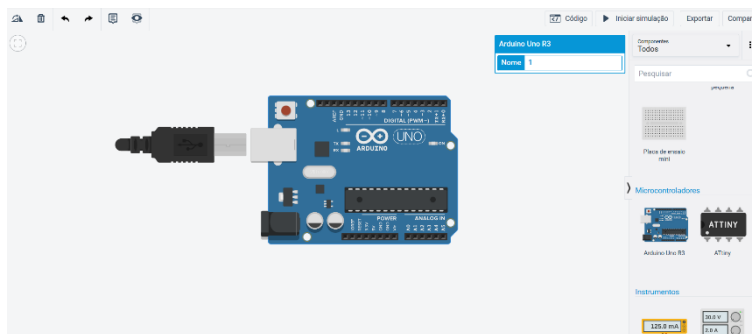
Exercícios – com o simulador TinkerCad

1. LEDs que piscam

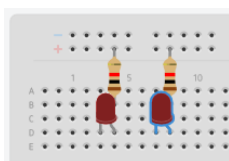
1. Circuits > Criar novo circuito



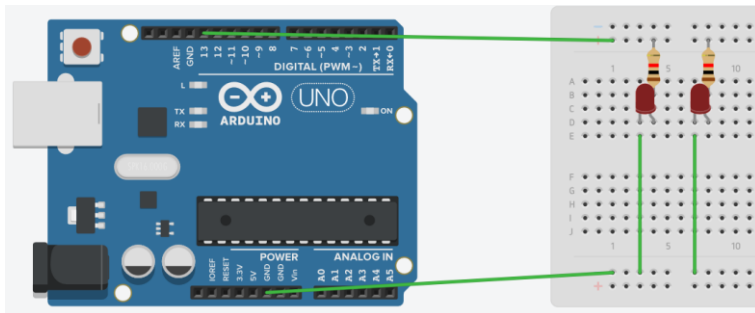
2. Em “Componentes” selecionar “Todos”
3. Escolher o “Arduino UNO R3” e arrastá-lo para o espaço de trabalho



4. Arraste os restantes componentes necessários:
 - 1 placa de ensaio
 - 2 resistores
 - 2 LEDs
5. Arraste os resistores para a placa de ensaio de forma a que uma das extremidades de cada resistor esteja na linha (+)
6. Os LEDs têm uma perna maior (“torta”), que é a que se deve ligar ao lado positivo. Posicione os LEDs de forma a que essa perna fique ligada à outra extremidade dos resistores



7. Crie as ligações como na figura abaixo



8. Abrir o “Código”
9. Onde está “Blocos” escolher “Texto”
10. Clicar em “Iniciar Simulação”. **O que acontece?**
11. Alterar valores do delay para 100. **O que acontece?**
12. Alterar o valor do último delay de volta para 1000. **O que acontece?**
13. Guarde o circuito: Clicar no símbolo do TinkerCad (canto superior esquerdo do browser).
 Nas opções do circuito (roldana) escolha “Propriedades”, altere o nome do projeto para “Primeiro teste” e grave as alterações

2. LEDs que piscam 2

Conhece a música “We Will Rock You” dos Queen (<https://youtu.be/-tJYN-eG1zk>)? Altere o código necessário para que os LEDs pisquem ao ritmo inicial da música.

3. LEDs que piscam 3

Faça as alterações necessárias ao circuito e código para que, em vez de os LEDs piscarem ao mesmo tempo, pisquem alternadamente durante um segundo cada.

4. LEDs que piscam 4

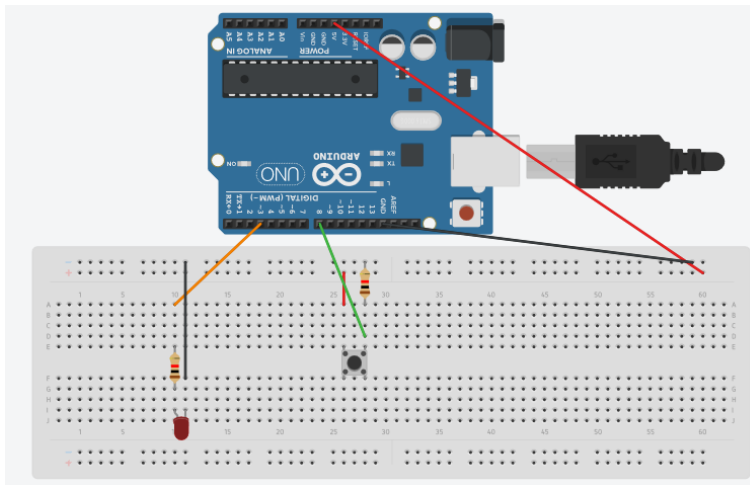
Com base no desafio 2, altere o código necessário para que os LEDs pisquem ao ritmo de “We Will Rock You”, em que um dos LEDs será o “tum” e outro será o “tá”.

5. Ligar e desligar LED com botão

1. Crie um novo circuito
2. Arraste os componentes necessários:

- 1 Placa de ensaio
- 1 Arduino UNO R3
- 1 LED
- 2 Resistores
- 1 Botão

3. Faça o seguinte setup do circuito:



4. Insira o seguinte código em linguagem C:

```

1  /* LED controlado por botão */
2
3  // set pinos
4  const int buttonPin = 8; // número do pino a que está ligado o botão
5  const int ledPin = 3; // número do pino a que está ligado o LED
6
7  // set variables
8  int estado = 0; // variável para leitura do botão
9  int guarda_estado = LOW; // variável para armazenar valores do botão
10
11 void setup() {
12   pinMode(ledPin, OUTPUT); // define o pino do Led como saída do Arduino
13   pinMode(buttonPin, INPUT); // define pino do pushbutton como entrada do Arduino
14 }
15
16 void loop() {
17   estado = digitalRead(buttonPin); // le o estado botão: ligado (HIGH) ou desligado (LOW)
18   if (estado == HIGH) { // verifica se o botão está pressionado
19     guarda_estado = !guarda_estado; // inverte valor do estado
20     delay(500); //espera o tempo de 500ms para evitar que haja várias vezes alterações
21   }
22
23   if (guarda_estado == HIGH) {
24     digitalWrite(ledPin, HIGH); // liga o led
25   } else {
26     digitalWrite(ledPin, LOW); // desliga o led
27   }
28 }

```

5. Inicie a simulação e teste o seu funcionamento

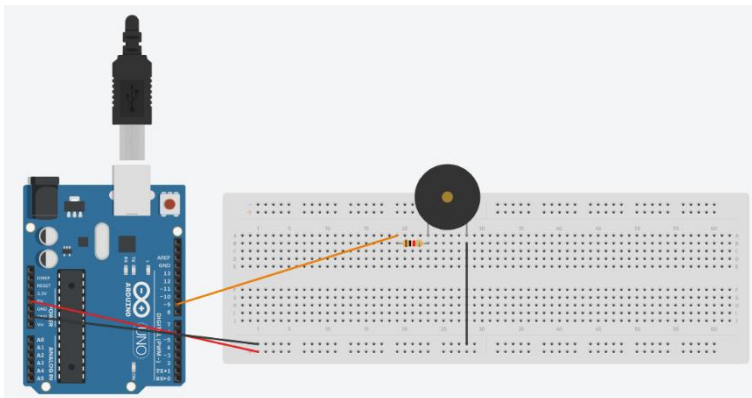
6. Utilização de Buzzer

Piezo speakers (buzzers) A "piezo buzzer" is basically a tiny speaker that you can connect directly to an Arduino. ... Connect one pin (it doesn't matter which one) to the Arduino's ground (Gnd) and the other end to digital pin 8. From the Arduino, you can make sounds with a buzzer by using tone.

Documentação: <https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/>

Buzzer emite som intermitentemente

1. Circuito:



2. Código:

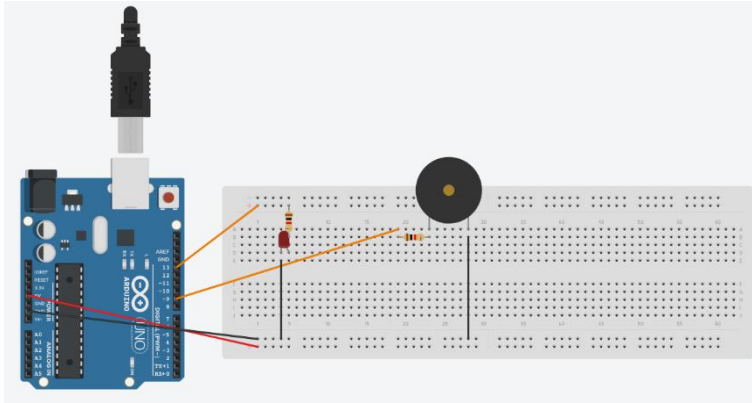
```
Texto [v] [Download] [Salvar] [Executar] 1 (Arduino Uno R3) [v]
1  const int buzzer = 9;
2
3  void setup() {
4      pinMode(buzzer, OUTPUT);
5  }
6
7  void loop() {
8      tone(buzzer, 1000); // Send 1KHz sound signal...
9      delay(1000);        // ...for 1 sec
10     noTone(buzzer);      // Stop sound...
11     delay(1000);        // ...for 1sec
12 }
```

3. Inicie a simulação e teste o seu funcionamento

7. Utilização de Buzzer e LED

Buzzer emite som intermitentemente, ao mesmo tempo que LED pisca

1. Circuito:



2. Código:

```
Texto 1 (Arduino Uno R3)
1  const int buzzer = 9;
2  //completar: definir o pino do led
3
4
5  void setup() {
6      pinMode(buzzer, OUTPUT);
7      // completar: definir o pino do led como sendo de output
8  }
9
10 void loop() {
11
12     tone(buzzer, 1000);
13     // completar: o que tem de acontecer aqui?
14     delay(1000);
15     noTone(buzzer);
16     // completar: o que tem de acontecer aqui?
17     delay(1000);
18
19 }
```

3. Complete os espaços assinalados com “// completar” com o código necessário
4. Inicie a simulação e teste o seu funcionamento

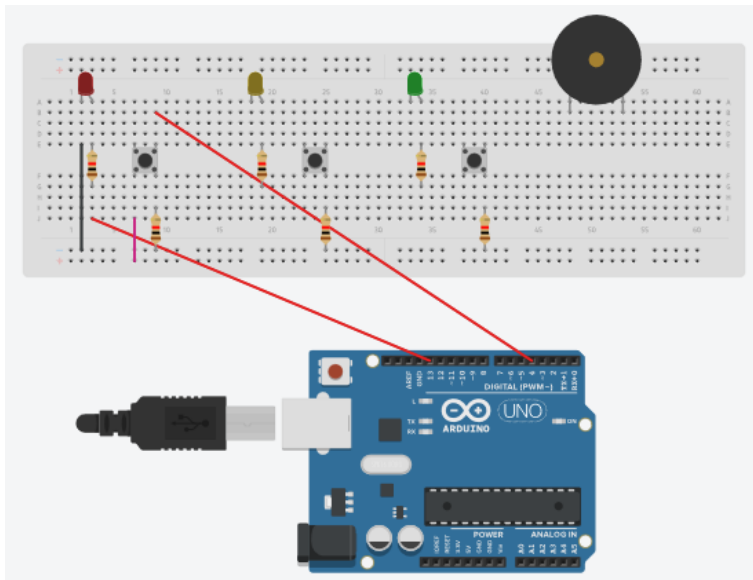
8. Piano que pisca

Objetivo: Criar um circuito onde cada botão acende um LED e toca uma nota musical diferente.

1. Componentes necessários:

- 1 Placa de ensaio
- 1 Arduino UNO R3
- 3 LEDs
- 6 Resistores
- 3 Botões
- 1 Buzzer

2. Circuito parcial (apenas contém as ligações para o primeiro LED e botão)



3. Complete o circuito

4. Código parcial (apenas contém o código para o primeiro LED e botão)

```

1 const int ledpinol = 13;
2 const int botaoA = 4;
3 int estado_botaoA = 0;
4 // completar: leds 2 e 3, botões B e C
5 const int som = 9;
6 int tom = 0;
7
8 void setup() {
9   pinMode(ledpinol, OUTPUT);
10  pinMode(botaoA, INPUT);
11  // completar: leds 2 e 3, botões B e C
12  pinMode(som, OUTPUT);
13 }
14
15 void loop() {
16   estado_botaoA = digitalRead(botaoA);
17   // completar: botões B e C
18
19   if(estado_botaoA && !estado_botaoB && !estado_botaoC) {
20     tom = 100;
21     digitalWrite(ledpinol, HIGH);
22   }
23   // completar: botões B e C (mudar tom), leds 2 e 3
24
25   if(tom > 0) {
26     digitalWrite(som, HIGH); // Liga buzzer
27     delayMicroseconds(tom); // Espera o tempo proporcional ao comprimento de onda da nota musical em milissegundos
28     digitalWrite(som, LOW); // Desliga buzzer
29     delayMicroseconds(tom); // Espera o tempo proporcional ao comprimento de onda da nota musical em milissegundos
30     tom = 0; // Reset do tom para zero, para sair do loop while e nao tocar o som constantemente
31     digitalWrite(ledpinol, LOW);
32     // completar: leds 2 e 3
33   }
34 }

```

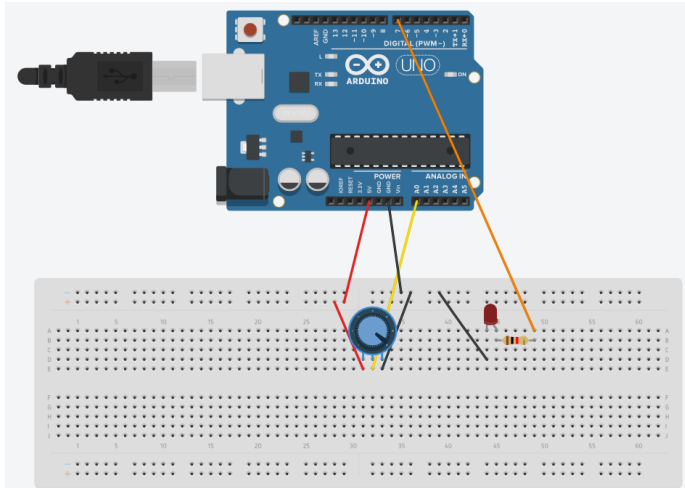
5. Complete os espaços assinalados com “// completar” com o código necessário

6. Inicie a simulação e teste o seu funcionamento

9. Input analógico + LED

Objetivo: usar o input analógico para regular a intensidade do LED

1. Circuito



2. Código

```

1  int led = 7;
2
3  void setup() {
4      Serial.begin(9600);
5      pinMode(led, OUTPUT);
6  }
7
8  void loop() {
9      int sensorValue = analogRead(A0);
10     int newValue = map(sensorValue, 0, 1023, 0, 255);
11     analogWrite(led, newValue);
12
13     // print the results to the Serial Monitor:
14     Serial.print("sensor = ");
15     Serial.print(sensorValue);
16     Serial.print("\t output = ");
17     Serial.println(newValue);
18
19     delay(1000);
20 }
    
```

3. Inicie a simulação e teste o seu funcionamento

10. Input analógico + LED + Buzzer

Usar o input analógico para controlar a intensidade do LED e a frequência a que um buzzer emite som.

11. Sensor de temperatura e LCD 16x2

Usar um sensor de temperatura para medir a temperatura atual e mostrar o resultado num visor LCD 16x2



Documentação:

- Sensor de temperatura:
 - <https://www.instructables.com/id/TMP36-Temperature-Sensor-Arduino-Tinkercad/>
 - <https://www.bc-robotics.com/tutorials/using-a-tmp36-temperature-sensor-with-arduino/>
 - Nota: o output direto do sensor de temperatura não é a temperatura (em °C). Para converter o output para temp(°C) é necessário usar a expressão existente no projeto "TMP36 Temperature Sensor With Arduino in Tinkercad" (admitindo que o sensor está ligado à saída analógica A0):

```
celsius = map((analogRead(A0) - 20) * 3.04), 0, 1023, -40, 125);
```
 - Função map:
<https://www.arduino.cc/reference/en/language/functions/math/map/>
- Visor LCD 16x2:
https://www.tinkercad.com/things/dGiMrTYFn9R-copy-of-16x2-lcd-display/edit!/?sharecode=mnqd_76wiTlxLtJsoFUPIPDmBtywokO1j6thV38e8KA

12. Sensor de luminosidade e LEDs

Usar um sensor de luminosidade (fotorresistor) para medir a luminosidade atual e, conforme o valor, acender um LED verde (luminosidade baixa - < 300), amarelo (média - < 600) ou vermelho (alta).



Documentação:

- Fotorresistor: <https://blog.tinkercad.com/2018/08/15/light-sensor-photoresistor-with-arduino-in-tinkercad-circuits>
- LED: <https://www.instructables.com/id/Blink-an-LED-With-Arduino-in-Tinkercad/>

13. Sensor de gás e Buzzer

Usar um sensor de gás para medir a concentração de gás e, conforme o valor, tocar um som diferente no buzzer (o valor pode ser definido, por exemplo, para metade do valor obtido pelo sensor de gás)

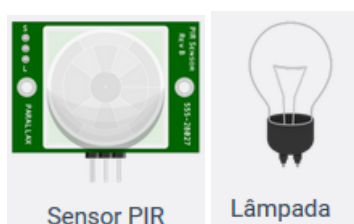


Documentação:

- Sensor de gás
 - <https://www.tinkercad.com/things/7S1QneIskY8-copy-of-gas-sensor/editel?sharecode=MjEuHGF7JDcU0CepZEmqa2t5FWN5YaavLPZCvmtCgqw>
- Buzzer
 - <https://www.tinkercad.com/things/d4URImkqGYu-piezo>
 - Função tone: <https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/>

14. Sensor de movimento e lâmpada

Usar um sensor PIR para detetar movimento e, sempre que for detetado, deve acender uma lâmpada.

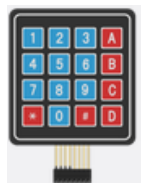


Documentação:

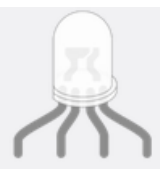
- Sensor PIR: https://www.tinkercad.com/things/kXM4u4GVFDd-copy-of-if-else-pir-sensor-leds/editel?sharecode=ItjLpEMR6_YsBznT8LwZs5viorl9SBRtG-Jcon6o454
- Lâmpada: https://www.tinkercad.com/things/3W0m9blpRrd-copy-of-turn-on-an-ac-lamp-using-arduino-5v-relay-and-motion/editel?sharecode=IV9HRpSHwx95U_9E2u7ogN-bufKNT_IceXOdAphD81o

15. Teclado 4x4 e LED RGB

Usar um teclado 4x4 para mudar a cor de um LED RGB conforme a tecla que for pressionada.



Teclado 4x4



LED RGB

As cores a definir no LED RGB podem ser encontradas na seguinte tabela:

| Tecla | Red | Green | Blue | Tecla | Red | Green | Blue |
|-------|-----|-------|------|-------|-----|-------|------|
| 1 | 0 | 0 | 0 | 7 | 192 | 192 | 192 |
| 2 | 255 | 255 | 255 | 8 | 128 | 128 | 128 |
| 3 | 255 | 0 | 0 | 9 | 128 | 0 | 0 |
| A | 0 | 255 | 0 | C | 128 | 128 | 0 |
| 4 | 0 | 0 | 255 | * | 0 | 128 | 0 |
| 5 | 255 | 255 | 0 | 0 | 128 | 0 | 128 |
| 6 | 0 | 255 | 255 | # | 0 | 128 | 128 |
| B | 255 | 0 | 255 | D | 0 | 0 | 128 |

Documentação:

- Teclado 4x4: <https://www.tinkercad.com/things/8iDg0ZuguN8-using-keypad-4x4-with-arduino>
- LED RGB: <https://www.tinkercad.com/things/a8KMGOddc8F-copy-of-rgb-led/editel?sharecode=D5X48FVf3IJGAr1Xjd6Dcp5mNcfSz2tjnAKeHPJjEE4>

16. Teclado 4x4

Usar um teclado 4x4 para inserir um código que deve ser mostrado no monitor série. Os códigos podem conter as letras A, B, C e D e os números de 0 a 9. Não há limite mínimo nem máximo para o número de caracteres. Os códigos começam com o caráter * e terminam com o caráter #. O código inserido deve ser mostrado no monitor série.

17. Teclado 4x4 e LCD



Usar um teclado 4x4 para inserir um código que deve ser mostrado no monitor série. Os códigos podem conter as letras A, B, C e D e os números de 0 a 9. Não há limite mínimo nem máximo para o número de caracteres. Os códigos começam com o caráter * e terminam com o caráter #. O código inserido deve ser mostrado no LCD.

18. Comando Infravermelho

Usar um comando e um sensor de infravermelhos. Fazer com que o botão carregado no comando apareça no monitor série.

19. Comando infravermelho e LEDs

Usar um comando e um sensor de infravermelhos. Fazer com que, dependendo da tecla carregada, acenda um LED de cor diferente:

- 1: verde
- 2: azul
- 3: amarelo
- Qualquer outra: vermelho

Para além disso, quando se acende o LED vermelho, deve também aparecer no monitor série a razão do erro (exemplo: “Erro – carregou tecla 4”)

Exercícios – com a placa Arduino

1. LEDs que piscam

1. Monte o mesmo circuito do exercício **1** da secção anterior utilizando o material existente no laboratório
2. Ligue o circuito ao seu computador
3. Abra o IDE e verifique se a placa foi reconhecida:
 - Ferramentas > Placa “Arduino/Genuíno Uno”
 - Porta > Serial Port - COM3 ou COM1
4. Selecione o exemplo Blink: Ficheiro > Exemplos > 01.Basics > Blink
5. Na nova janela aberta, clique em “Verificar” para verificar que não existem erros
6. Clique em “Envio” para ver o circuito a funcionar

2. Ligar e desligar LED com botão

Reproduza o exercício **5** da secção anterior com o hardware existe na sala e com o IDE do Arduino no seu computador.

3. Input analógico

Reproduza o exercício **9** da secção anterior com o hardware existe na sala e com o IDE do Arduino no seu computador.

Para além destes, obrigatórios, poderá reproduzir qualquer um dos exercícios da secção anterior recorrendo ao material existente no laboratório.