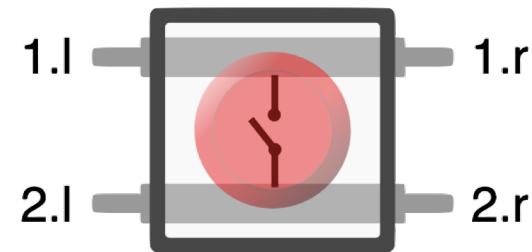
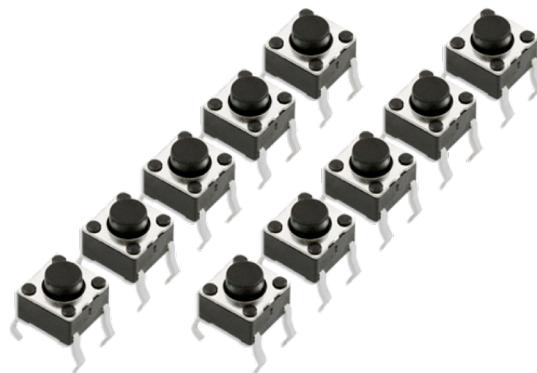


DevTitans

Wokwi: Push Button

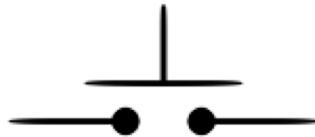
Push Buttons (Botão de Pressão)

- São chaves tácteis (*push buttons*) ou botões de pressão
- É um dos componentes eletrônicos mais utilizados em projetos



Normalmente aberta / fechada

Normally-Open
Pushbutton Switch



Normally-Closed
Pushbutton Switch



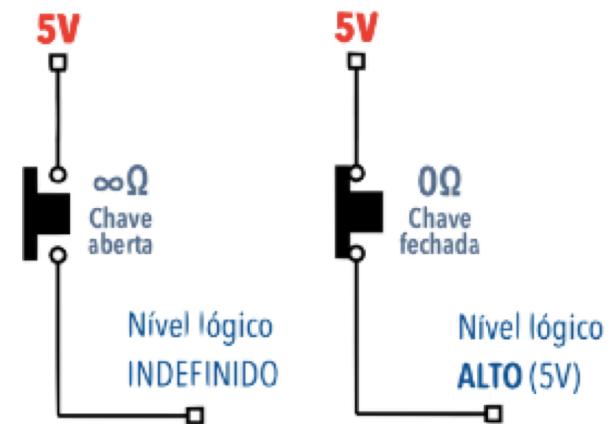
O push button normalmente **aberto** é o mais comum



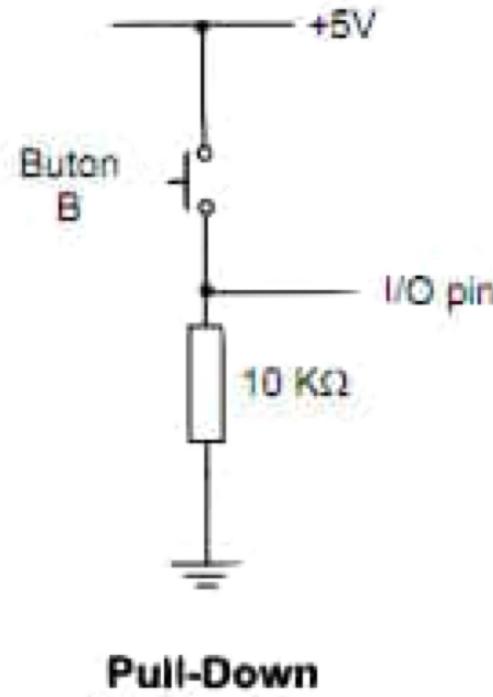
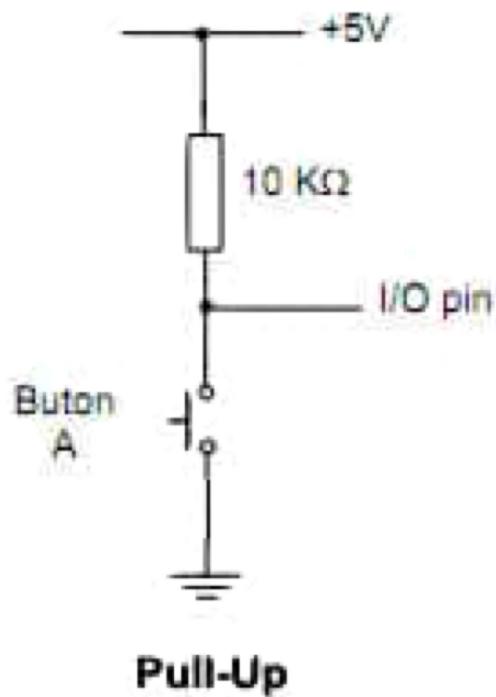
Resistores pull-up e pull-down

Estabilidade com Resistores

- Os resistores pull-up e pull-down garantem **um nível lógico estável** quando por exemplo uma tecla não está pressionada
- Geralmente utiliza-se um resistor de $10K\Omega$ para este propósito
- O circuito com push button pode ser feita de 3 formas:
 - Resistor Pull-down
 - Resistor Pull-up e
 - Resistor Pull-up interno (Arduino)

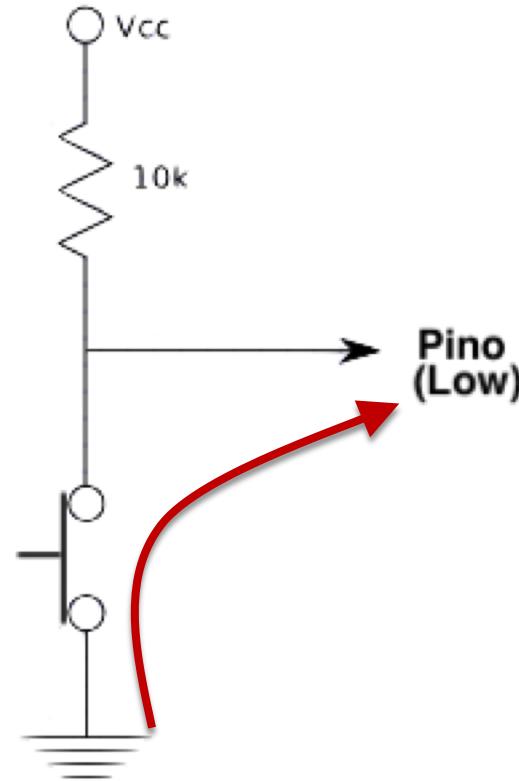
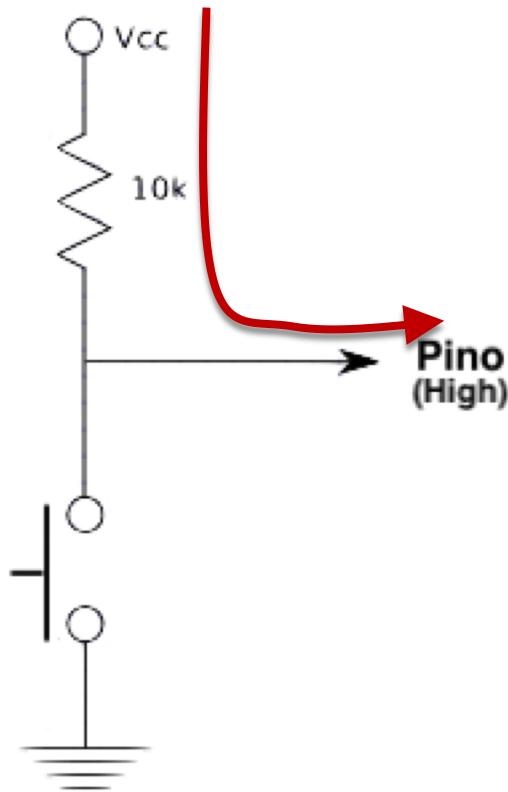


Circuitos com resistores pull-up e pull-down



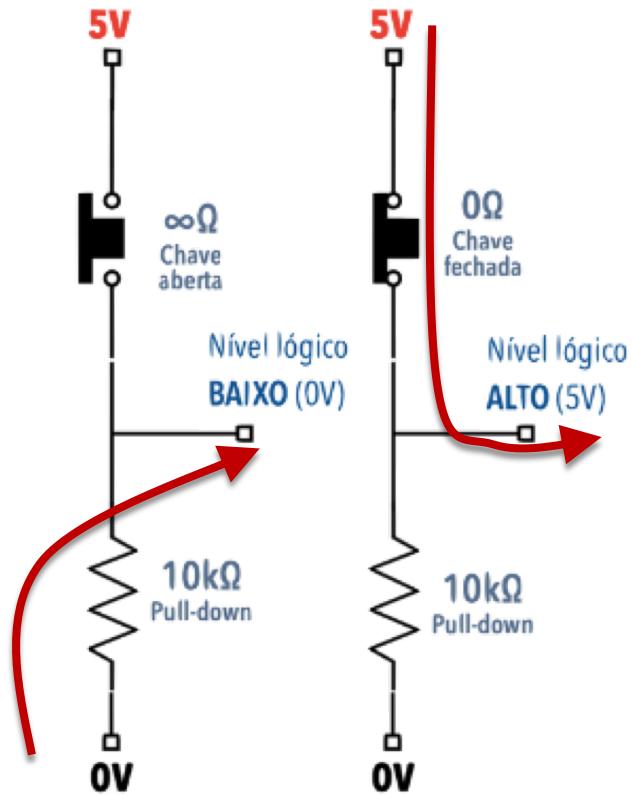
A única diferença é que, quando a chave está aberta (não acionada), o nível lógico **no pino** será **HIGH**, se for pull-up, e **LOW**, se for pull-down

Circuitos com resistor pull-up



Quando a chave está aberta, realmente o circuito **puxa para cima** (**HIGH**) no pino lógico

Circuitos com resistor pull-down

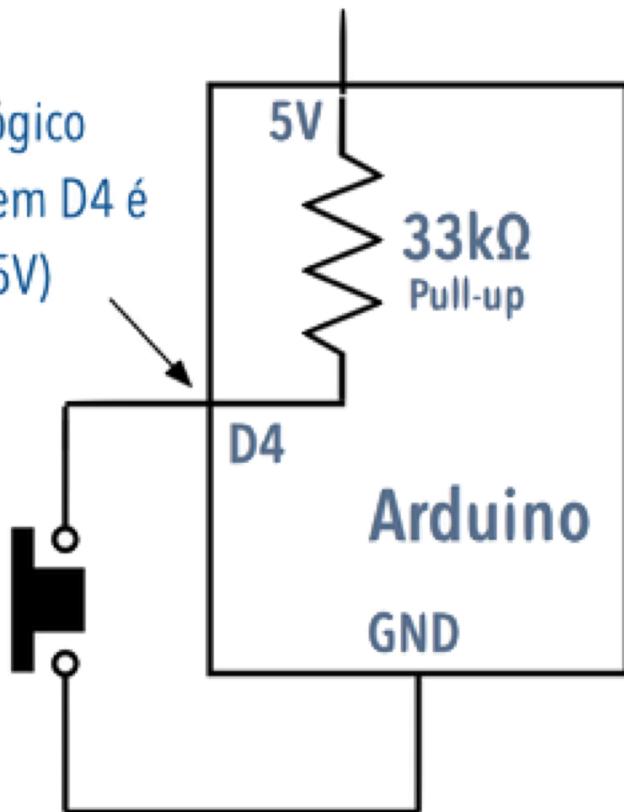


Quando a chave está aberta, realmente o circuito **puxa para baixo (LOW)** no pino lógico

Circuitos com resistor pull-up (interno)

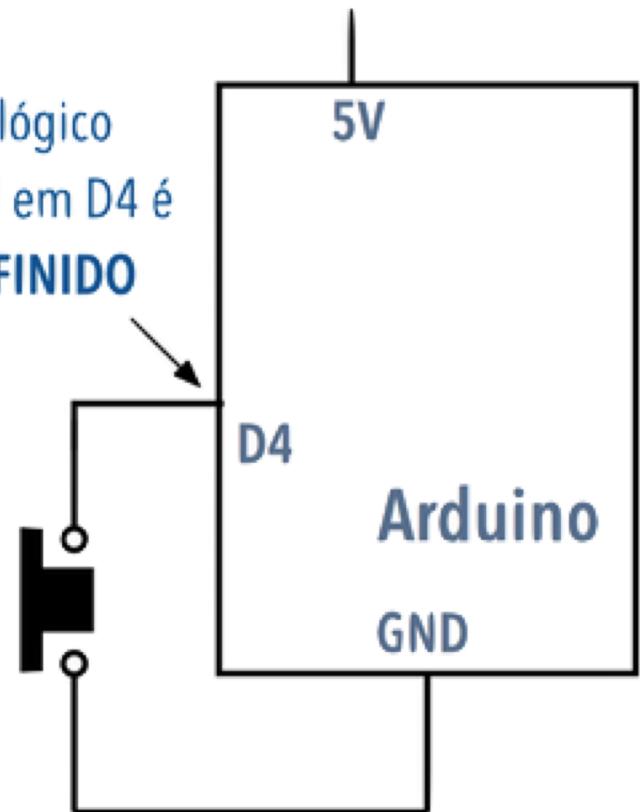
pinMode
INPUT PULLUP

Nível lógico
inicial em D4 é
ALTO (5V)



pinMode
INPUT

Nível lógico
inicial em D4 é
INDEFINIDO

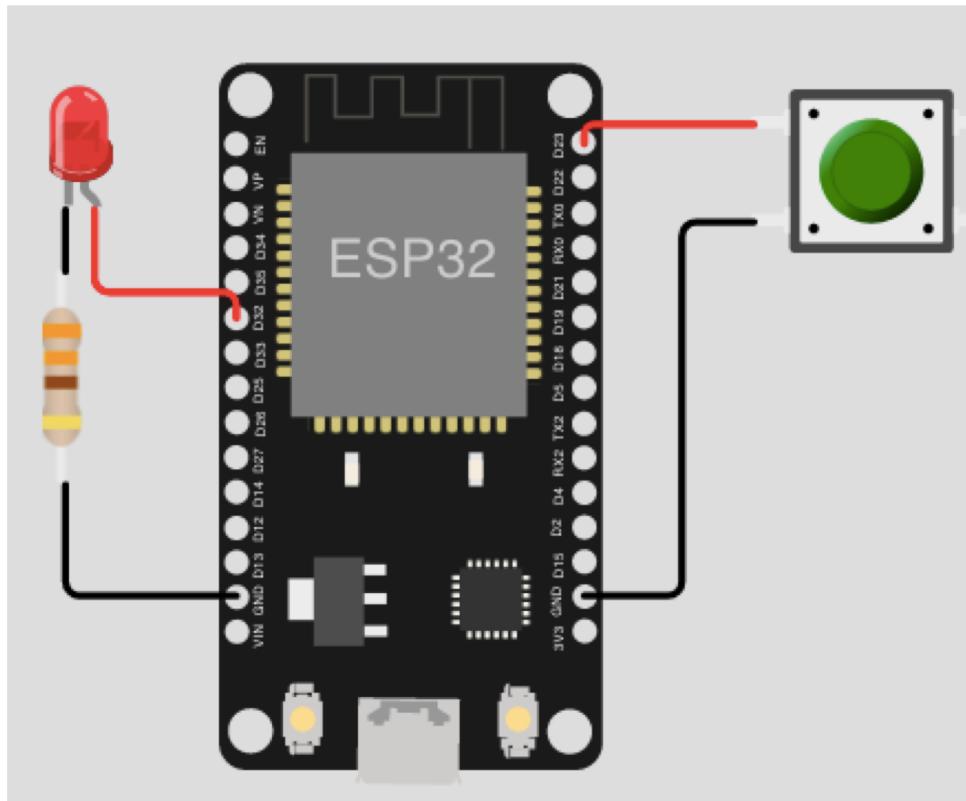




Resistor pull-up interno

Círculo

Construa o círculo abaixo



Anodo (+) do LED (Pino D32)
Catodo (-) do LED (Term. Resistor)
Term. Resistor (Pino GND2)
Contato1 (D23)
Contato 2 (GND1)

Codificação



The image shows the WOKWI online simulation environment. At the top, there's a navigation bar with the WOKWI logo, a 'SAVE' button, a 'SHARE' button, and a heart icon. Below the navigation bar, there are tabs for 'sketch.ino', 'diagram.json', and 'Library Manager'. The 'sketch.ino' tab is active and contains the following Arduino-style code:

```
1 // LED + PushButton com Pull-Up Implicito
2
3 const int ledPin = 32;
4 const int Botao = 23;
5 int estadoBotao;
6
7 void setup(){
8     pinMode(ledPin, OUTPUT);
9     pinMode(Botao, INPUT_PULLUP);
10 }
11
12 void loop(){
13     estadoBotao = digitalRead(Botao);
14
15     if (estadoBotao == HIGH)
16         digitalWrite(ledPin, HIGH);
17     else if (estadoBotao == LOW)
18         digitalWrite(ledPin, LOW);
19 }
20
```

Codificação

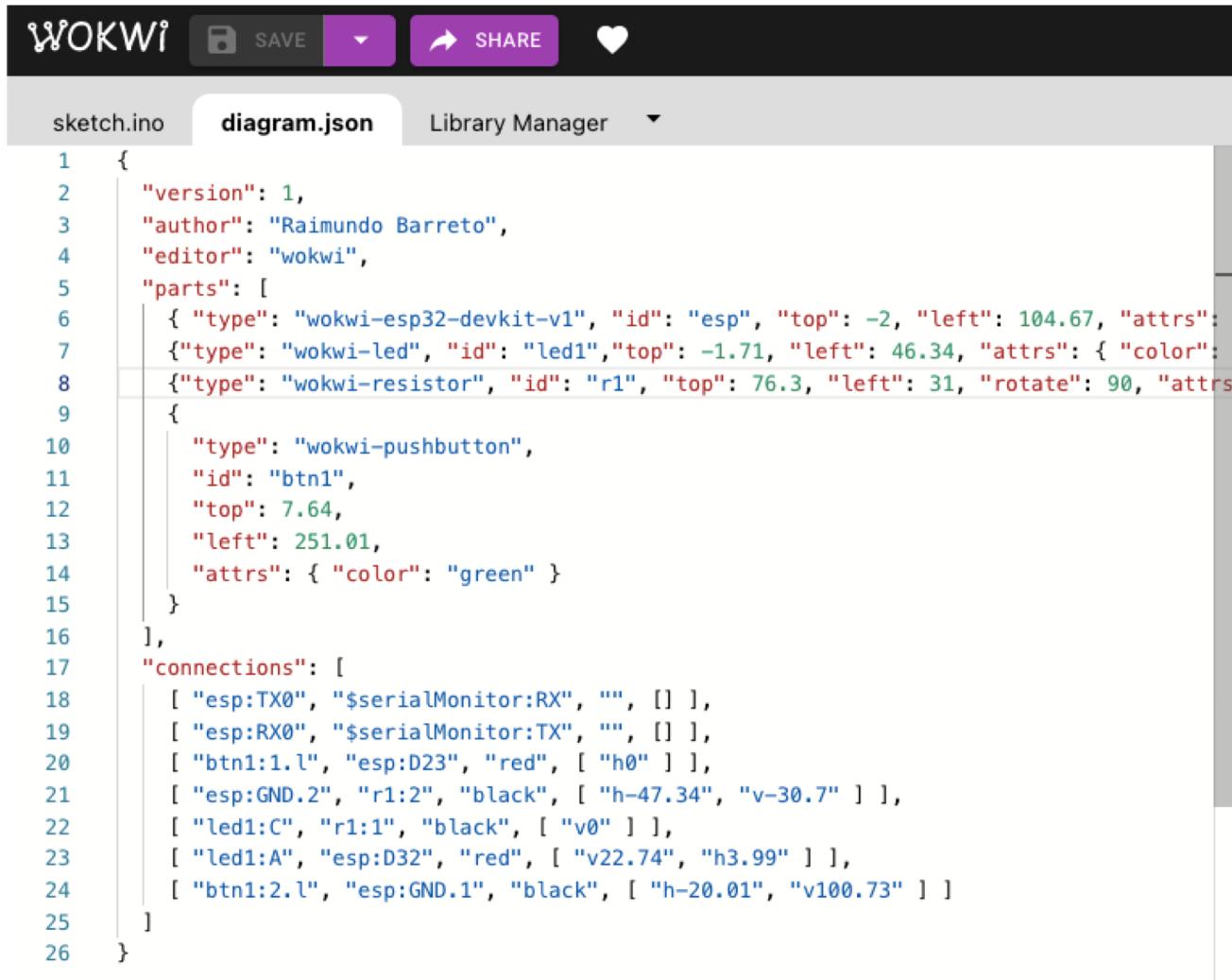
```
const int ledPin = 32;
const int Botao = 23;
int estadoBotao;

void setup(){
    pinMode(ledPin, OUTPUT);
    pinMode(Botao, INPUT_PULLUP);
}

void loop(){
    estadoBotao = digitalRead(Botao);

    if (estadoBotao == HIGH)
        digitalWrite(ledPin, HIGH);
    else if (estadoBotao == LOW)
        digitalWrite(ledPin, LOW);
}
```

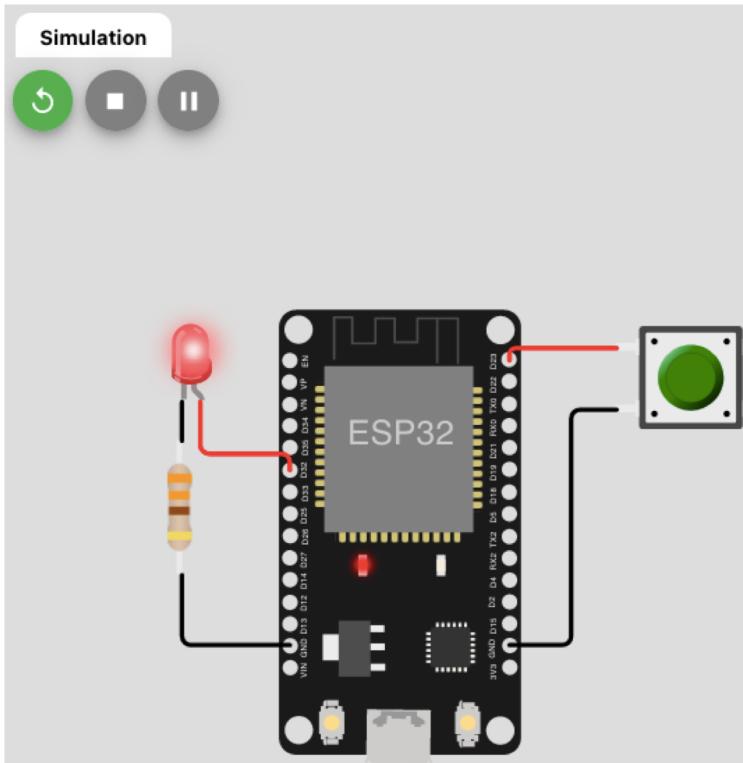
JSON



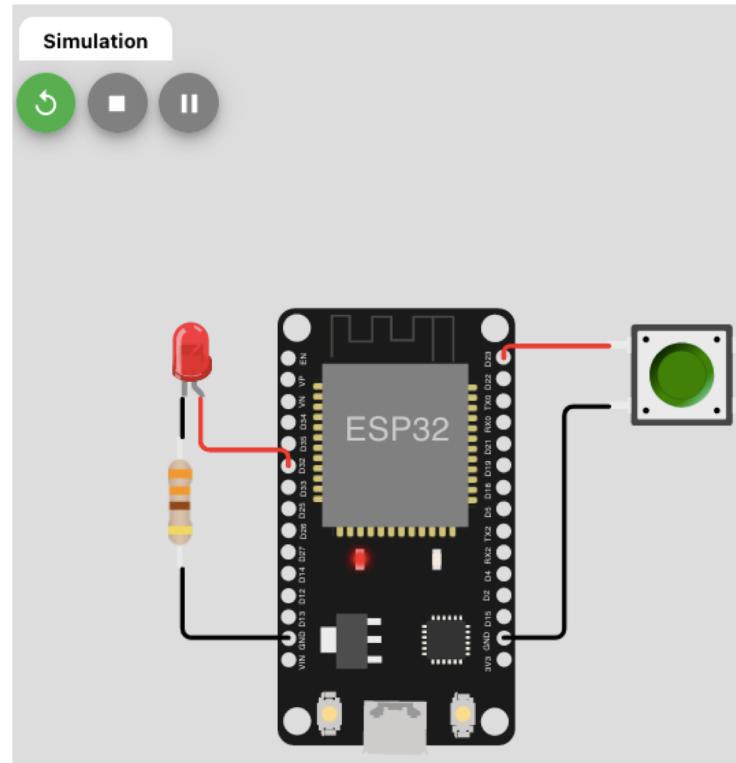
The screenshot shows the Wokwi editor interface with the title bar "WOKWI" and buttons for "SAVE", "SHARE", and a heart icon. Below the title bar, there are tabs for "sketch.ino" (selected), "diagram.json" (highlighted in purple), and "Library Manager". The "diagram.json" tab contains the following JSON code:

```
1  {
2    "version": 1,
3    "author": "Raimundo Barreto",
4    "editor": "wokwi",
5    "parts": [
6      { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -2, "left": 104.67, "attrs": {} },
7      { "type": "wokwi-led", "id": "led1", "top": -1.71, "left": 46.34, "attrs": { "color": "red" } },
8      { "type": "wokwi-resistor", "id": "r1", "top": 76.3, "left": 31, "rotate": 90, "attrs": {} },
9      {
10        "type": "wokwi-pushbutton",
11        "id": "btn1",
12        "top": 7.64,
13        "left": 251.01,
14        "attrs": { "color": "green" }
15      }
16    ],
17    "connections": [
18      [ "esp:TX0", "$serialMonitor:RX", "", [] ],
19      [ "esp:RX0", "$serialMonitor:TX", "", [] ],
20      [ "btn1:1.l", "esp:D23", "red", [ "h0" ] ],
21      [ "esp:GND.2", "r1:2", "black", [ "h-47.34", "v-30.7" ] ],
22      [ "led1:C", "r1:1", "black", [ "v0" ] ],
23      [ "led1:A", "esp:D32", "red", [ "v22.74", "h3.99" ] ],
24      [ "btn1:2.l", "esp:GND.1", "black", [ "h-20.01", "v100.73" ] ]
25    ]
26  }
```

Simulação



Botão não pressionado

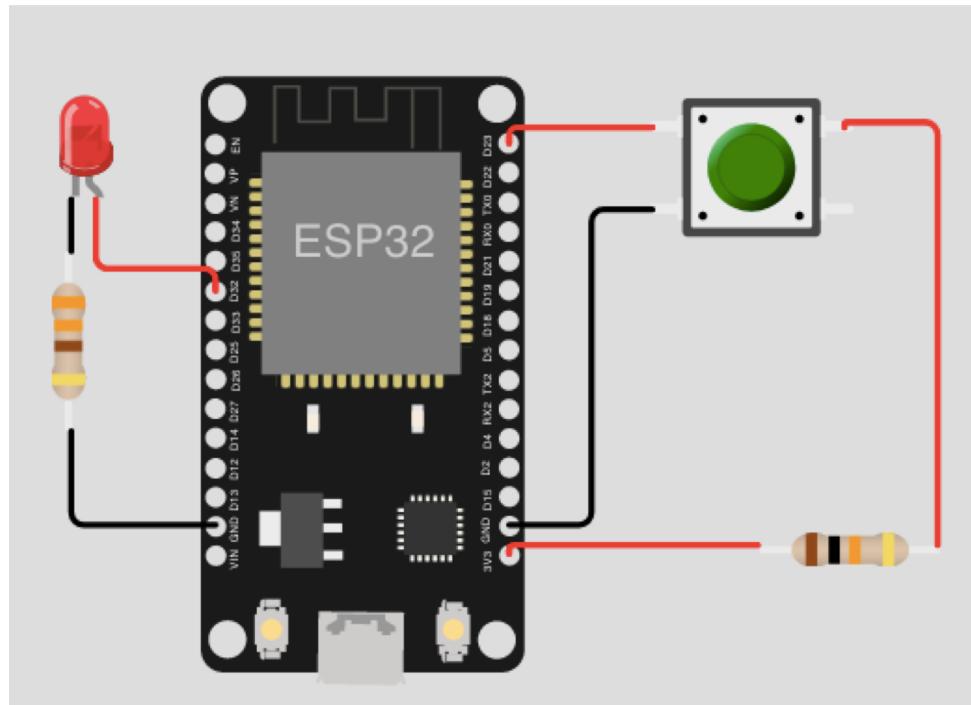


Botão pressionado

Resistor pull-up externo

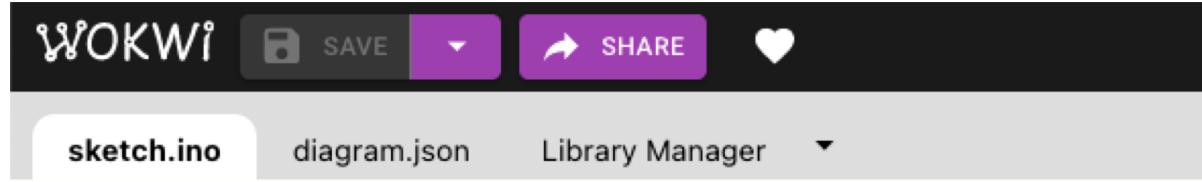
Círculo

Construa o círculo abaixo



- Anodo (+) do LED (Pino D32)
- Catodo (-) do LED (Term. Resistor)
- Term. Resistor (Pino GND2)
- Contato1 (D23 e Resistor 10K)
- Resistor 10K (Pino 3V3)
- Contato 2 (GND1)

Codificação



The image shows the WOKWi IDE interface. At the top, there is a dark header bar with the WOKWi logo, a save icon, a dropdown menu, a share icon, and a heart icon. Below the header, there is a navigation bar with tabs: "sketch.ino" (which is selected and highlighted in blue), "diagram.json", and "Library Manager". The main area contains the Arduino-style code for a sketch named "sketch.ino". The code is numbered from 1 to 20 on the left side.

```
1 // LED + PushButton com Pull-Up Implicito
2
3 const int ledPin = 32;
4 const int Botao = 23;
5 int estadoBotao;
6
7 void setup(){
8     pinMode(ledPin, OUTPUT);
9     pinMode(Botao, INPUT);
10 }
11
12 void loop(){
13     estadoBotao = digitalRead(Botao);
14
15     if (estadoBotao == HIGH)
16         digitalWrite(ledPin, HIGH);
17     else if (estadoBotao == LOW)
18         digitalWrite(ledPin, LOW);
19 }
20
```

Codificação

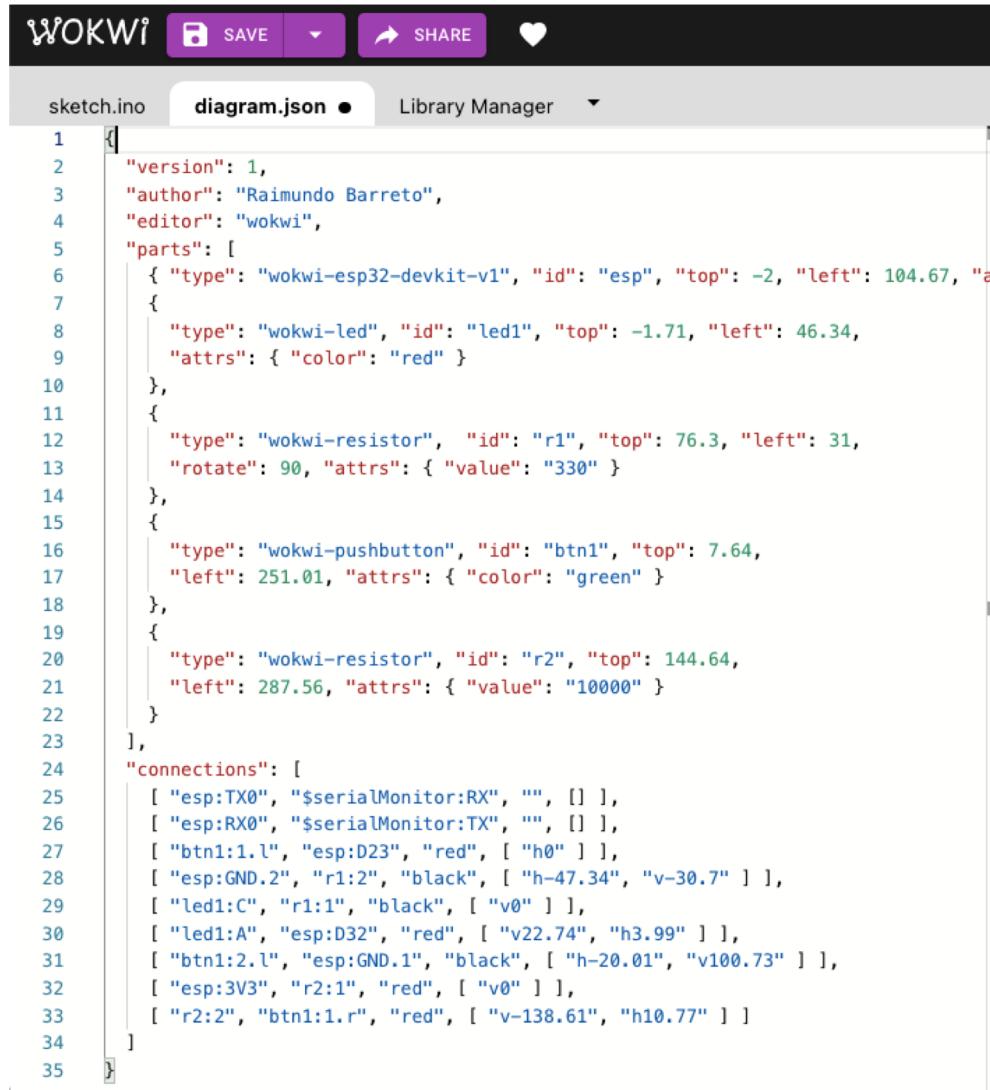
```
const int ledPin = 32;
const int Botao = 23;
int estadoBotao;

void setup(){
    pinMode(ledPin, OUTPUT);
    pinMode(Botao, INPUT);
}

void loop(){
    estadoBotao = digitalRead(Botao);

    if (estadoBotao == HIGH)
        digitalWrite(ledPin, HIGH);
    else if (estadoBotao == LOW)
        digitalWrite(ledPin, LOW);
}
```

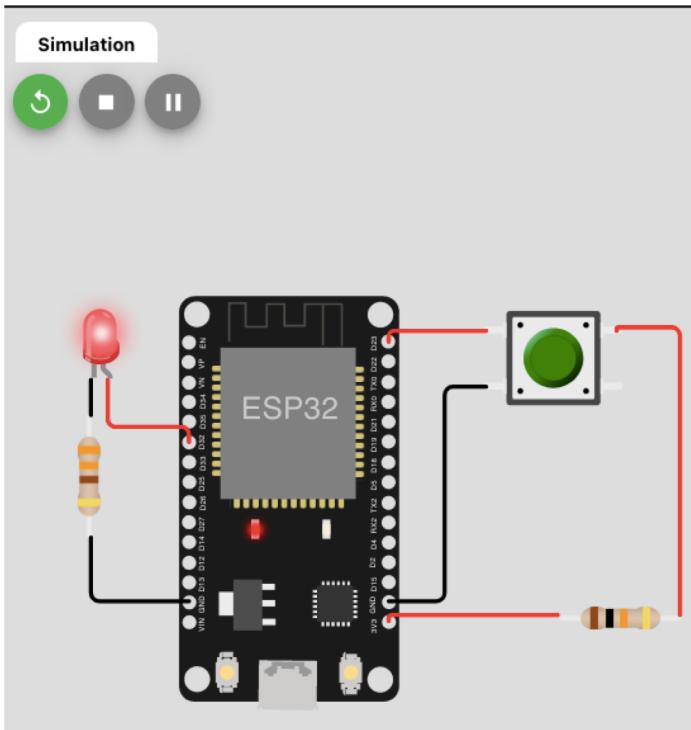
JSON



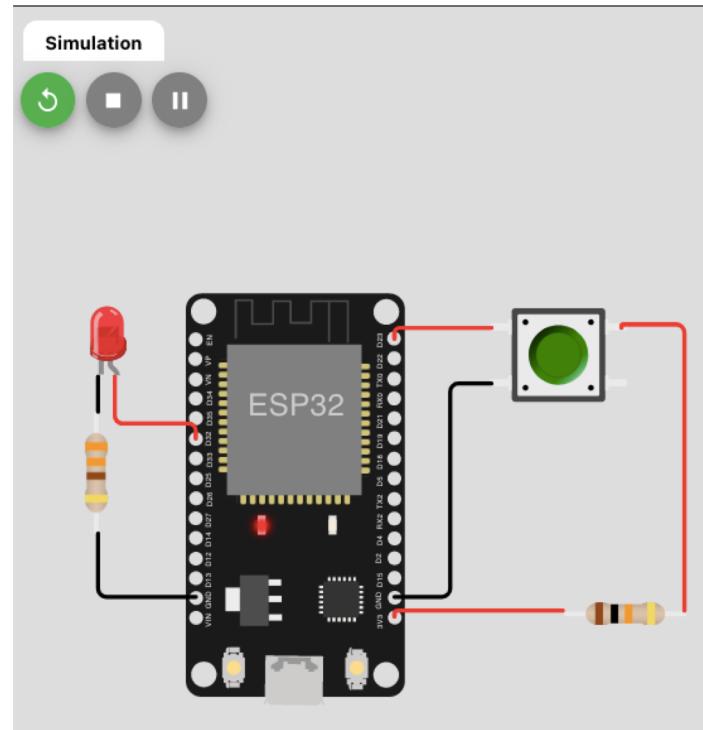
The screenshot shows the Wokwi web interface with a blue header bar. In the top left, it says "WOKWI". To its right are "SAVE" and "SHARE" buttons, and a heart icon. Below the header, there's a navigation bar with tabs: "sketch.ino" (which is currently selected), "diagram.json" (highlighted with a black border), and "Library Manager". The main area contains a JSON code editor with syntax highlighting. The JSON object describes a circuit diagram with components like an ESP32 DevKit v1, a red LED, a resistor, and a green pushbutton, along with their connections.

```
1  {
2    "version": 1,
3    "author": "Raimundo Barreto",
4    "editor": "wokwi",
5    "parts": [
6      { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -2, "left": 104.67, "at": 0, "w": 100, "h": 100 },
7      { "type": "wokwi-led", "id": "led1", "top": -1.71, "left": 46.34, "w": 10, "h": 10, "color": "red" },
8      { "type": "wokwi-resistor", "id": "r1", "top": 76.3, "left": 31, "w": 10, "h": 10, "value": "330", "rotate": 90 },
9      { "type": "wokwi-pushbutton", "id": "btn1", "top": 7.64, "left": 251.01, "w": 10, "h": 10, "color": "green" },
10     { "type": "wokwi-resistor", "id": "r2", "top": 144.64, "left": 287.56, "w": 10, "h": 10, "value": "10000" },
11   ],
12   "connections": [
13     [ "esp:TX0", "$serialMonitor:RX", "", [] ],
14     [ "esp:RX0", "$serialMonitor:TX", "", [] ],
15     [ "btn1:1.l", "esp:D23", "red", [ "h0" ] ],
16     [ "esp:GND.2", "r1:2", "black", [ "h-47.34", "v-30.7" ] ],
17     [ "led1:C", "r1:1", "black", [ "v0" ] ],
18     [ "led1:A", "esp:D32", "red", [ "v22.74", "h3.99" ] ],
19     [ "btn1:2.l", "esp:GND.1", "black", [ "h-20.01", "v100.73" ] ],
20     [ "esp:3V3", "r2:1", "red", [ "v0" ] ],
21     [ "r2:2", "btn1:1.r", "red", [ "v-138.61", "h10.77" ] ]
22   ]
23 }
24 }
```

Simulação



Botão não pressionado



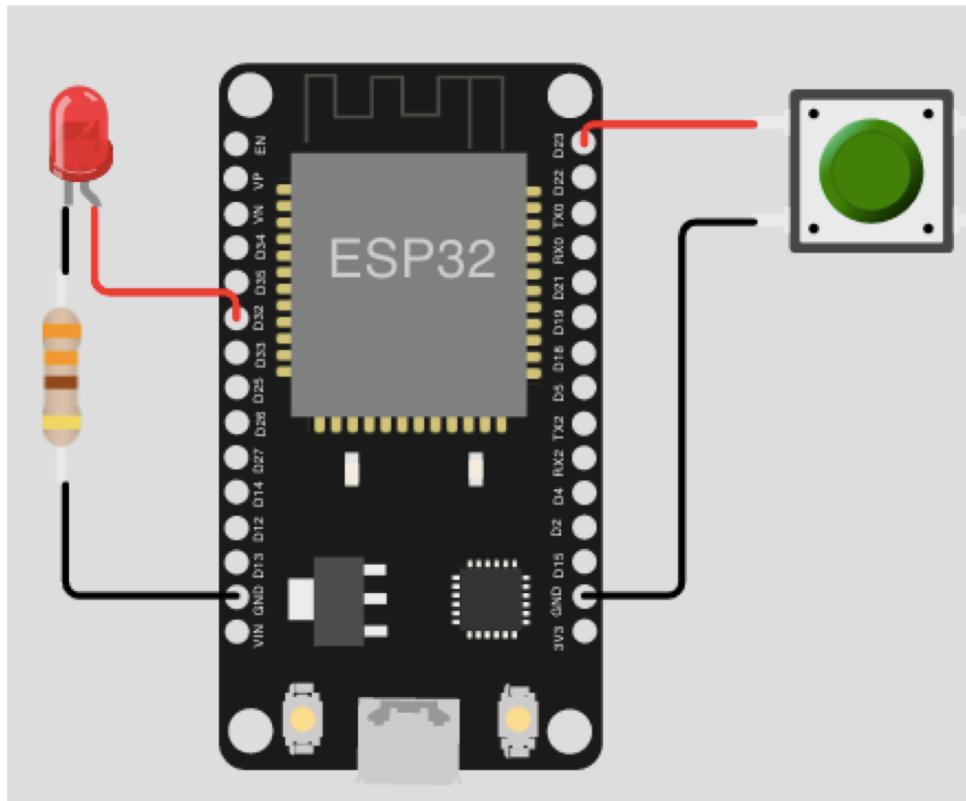
Botão pressionado

Chave liga-desliga com push button

Com resistor pull-up interno

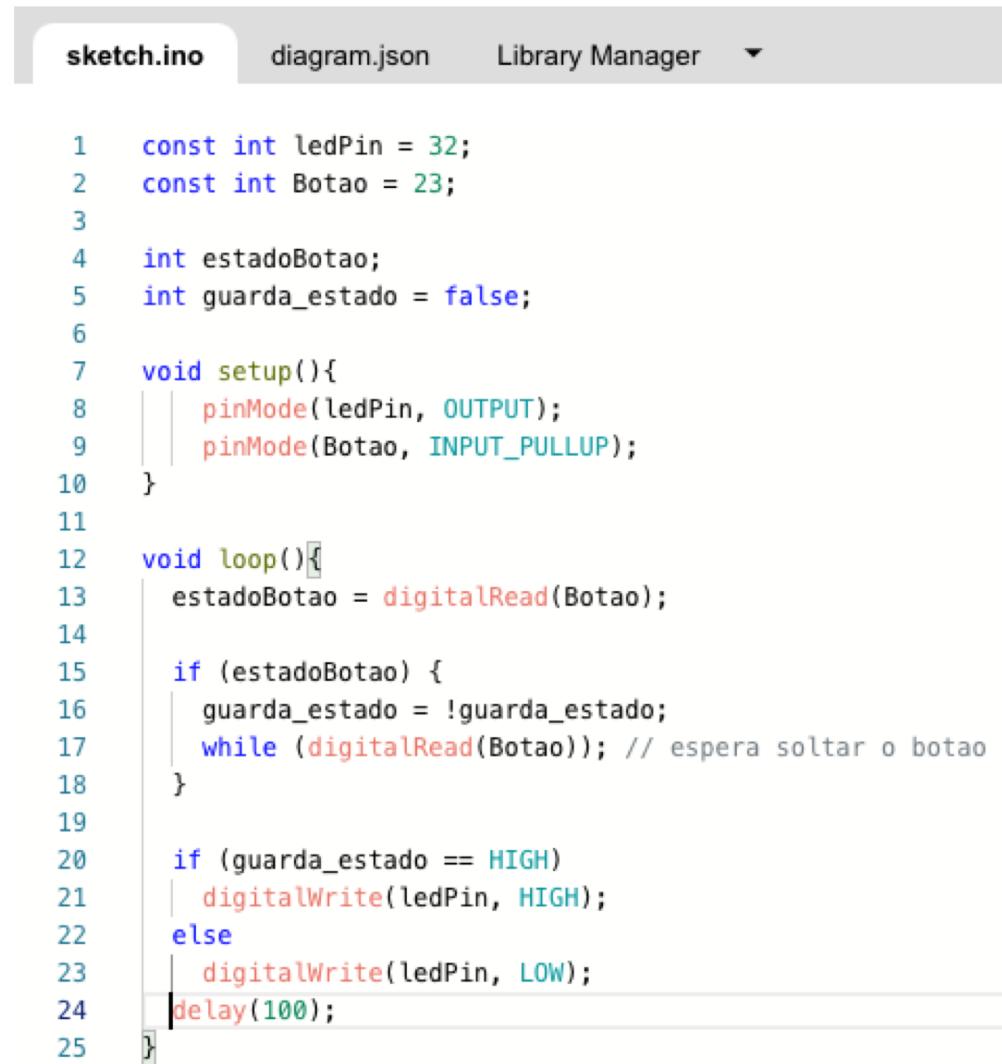
Círculo

Construa o círculo abaixo



Anodo (+) do LED (Pino D32)
Catodo (-) do LED (Term. Resistor)
Term. Resistor (Pino GND2)
Contato1 (D23)
Contato 2 (GND1)

Codificação



The screenshot shows the Arduino IDE interface with a sketch named "sketch.ino". The code is written in C++ and controls an LED connected to pin 32 and a push button connected to pin 23. The program initializes pins, reads the button state, and toggles the LED based on the button being pressed or released.

```
1  const int ledPin = 32;
2  const int Botao = 23;
3
4  int estadoBotao;
5  int guarda_estado = false;
6
7  void setup(){
8      pinMode(ledPin, OUTPUT);
9      pinMode(Botao, INPUT_PULLUP);
10 }
11
12 void loop(){
13     estadoBotao = digitalRead(Botao);
14
15     if (estadoBotao) {
16         guarda_estado = !guarda_estado;
17         while (digitalRead(Botao)); // espera soltar o botao
18     }
19
20     if (guarda_estado == HIGH)
21         digitalWrite(ledPin, HIGH);
22     else
23         digitalWrite(ledPin, LOW);
24     delay(100);
25 }
```

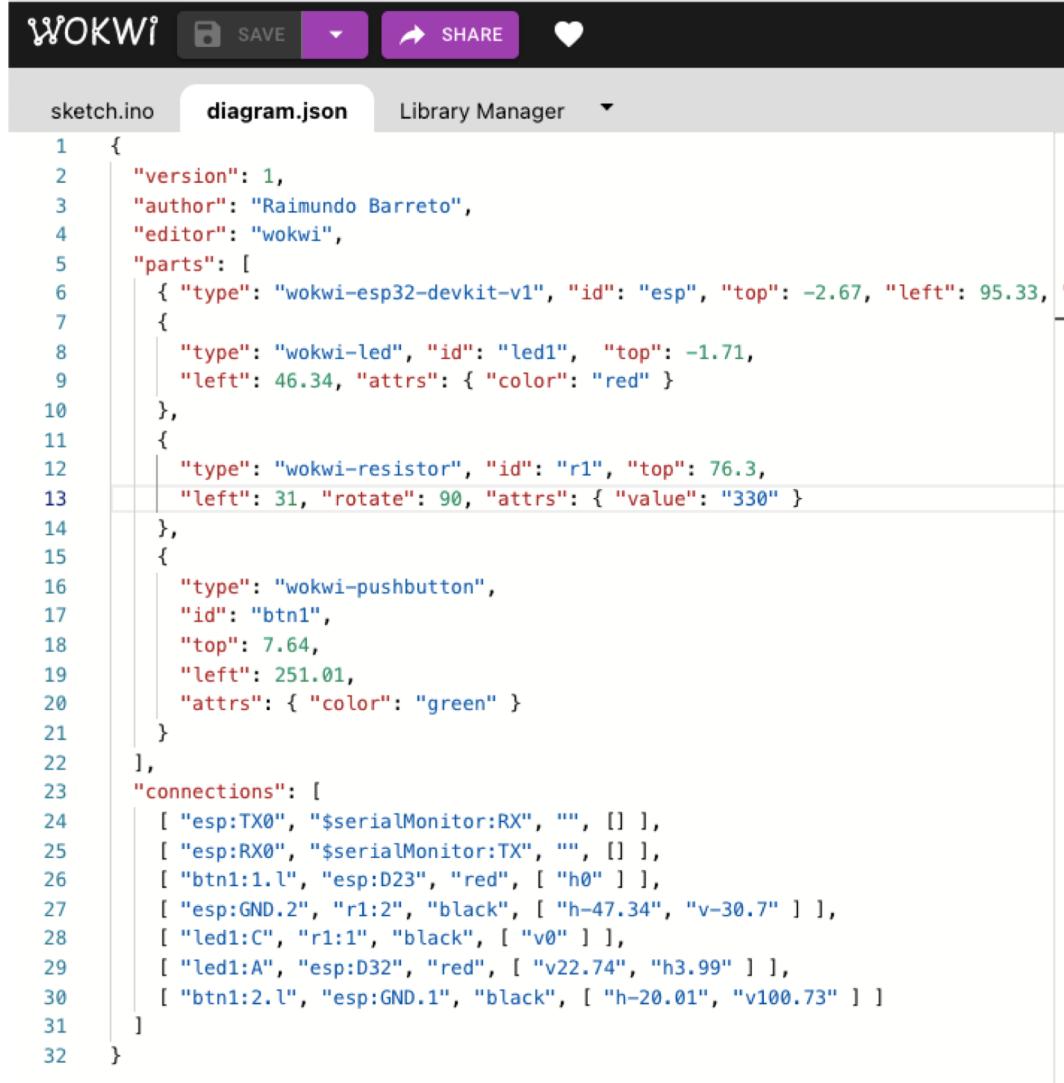
Codificação

```
const int ledPin = 32; const int Botao = 23;
int guarda_estado = false;

void setup() {
    pinMode(ledPin, OUTPUT);
    pinMode(Botao, INPUT_PULLUP);
}

void loop() {
    if (digitalRead(Botao)) {
        guarda_estado = !guarda_estado;
        while (digitalRead(Botao)); // espera soltar o botao
    }
    if (guarda_estado == HIGH)    digitalWrite(ledPin, HIGH);
    else digitalWrite(ledPin, LOW);
    delay(100);
}
```

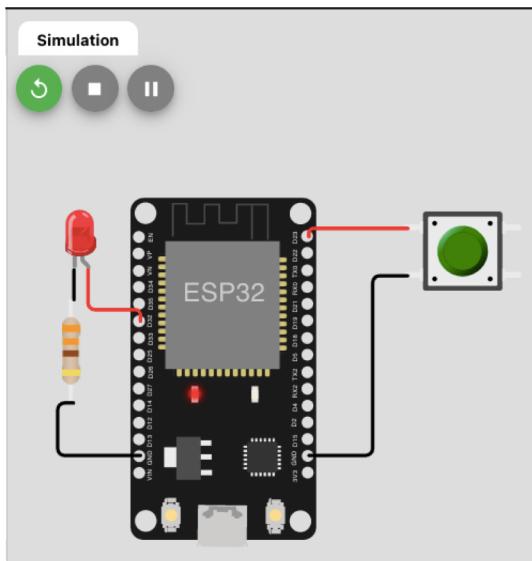
JSON



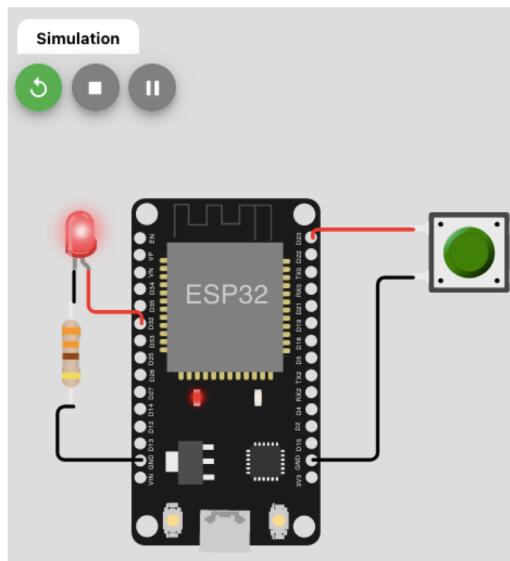
The screenshot shows the Wokwi editor interface with a dark theme. At the top, there are buttons for 'SAVE' and 'SHARE'. Below the header, there are tabs for 'sketch.ino' (which is currently selected), 'diagram.json', and 'Library Manager'. The main area displays the contents of the 'diagram.json' file.

```
1  {
2    "version": 1,
3    "author": "Raimundo Barreto",
4    "editor": "wokwi",
5    "parts": [
6      {
7        "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -2.67, "left": 95.33, "w": 100, "h": 100, "angle": 0, "pinouts": [
8          {
9            "type": "wokwi-led", "id": "led1", "top": -1.71, "left": 46.34, "w": 10, "h": 10, "color": "red", "pin": "D23", "pin_x": 46.34, "pin_y": -1.71, "pin_w": 10, "pin_h": 5, "pin_dx": 5, "pin_dy": 0
10         },
11         {
12           "type": "wokwi-resistor", "id": "r1", "top": 76.3, "left": 31, "w": 10, "h": 10, "value": "330", "pin": "GND", "pin_x": 31, "pin_y": 76.3, "pin_w": 10, "pin_h": 5, "pin_dx": 5, "pin_dy": 0
13         }
14       ],
15       "connections": [
16         [
17           "esp:TX0", "$serialMonitor:RX", "", [], [
18             {
19               "x": 95.33, "y": -2.67, "x2": 46.34, "y2": -1.71
20             }
21           ]
22         ],
23         [
24           "esp:RX0", "$serialMonitor:TX", "", [], [
25             {
26               "x": 95.33, "y": -2.67, "x2": 31, "y2": 76.3
27             }
28           ]
29         ],
30         [
31           "esp:D23", "red", [ "h0" ], [
32             {
33               "x": 46.34, "y": -1.71, "x2": 46.34, "y2": -47.34
34             }
35           ]
36         ],
37         [
38           "esp:D32", "red", [ "v22.74", "h3.99" ], [
39             {
40               "x": 31, "y": 76.3, "x2": 31, "y2": 22.74
41             }
42           ]
43         ],
44         [
45           "esp:GND.1", "black", [ "h-20.01", "v100.73" ], [
46             {
47               "x": 31, "y": 76.3, "x2": 20.99, "y2": 100.73
48             }
49           ]
50         ],
51         [
52           "esp:D23", "red", [ "h-47.34", "v-30.7" ], [
53             {
54               "x": 46.34, "y": -1.71, "x2": 46.34, "y2": -47.34
55             }
56           ]
57         ],
58         [
59           "esp:D32", "red", [ "v22.74", "h3.99" ], [
60             {
61               "x": 31, "y": 76.3, "x2": 31, "y2": 22.74
62             }
63           ]
64         ],
65         [
66           "esp:GND.1", "black", [ "h-20.01", "v100.73" ], [
67             {
68               "x": 31, "y": 76.3, "x2": 20.99, "y2": 100.73
69             }
70           ]
71         ]
72       ]
73     }
74   }
```

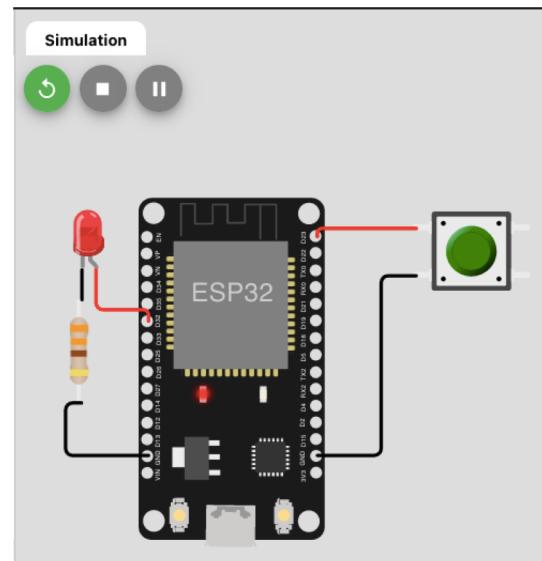
Simulação



Botão não pressionado



Pressionou e soltou



Pressionou e soltou