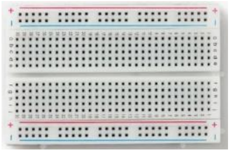
Componentes envolvidos:

Led Vermelho

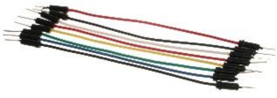
Breadboard



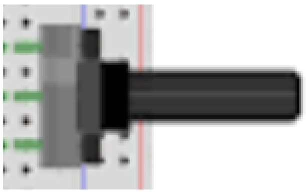
Resistência

Arduino

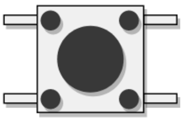
Cabos



Potenciómetro

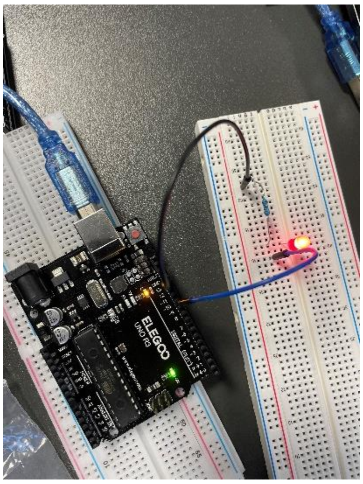


Botão

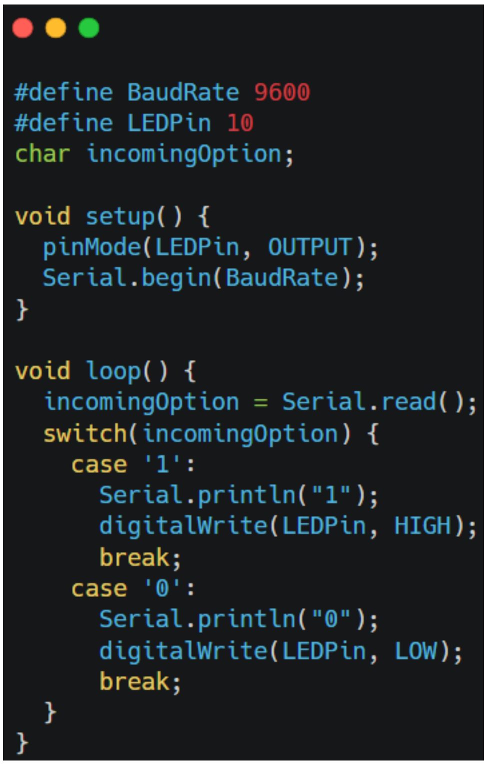


Exercício 1

Montagem do circuito e respetivos testes:



Código Utilizado:



```
#define BaudRate 9600
#define LEDPin 10
char incomingOption;

void setup() {
  pinMode(LEDPin, OUTPUT);
  Serial.begin(BaudRate);
}

void loop() {
  incomingOption = Serial.read();
  switch(incomingOption) {
    case '1':
      Serial.println("1");
      digitalWrite(LEDPin, HIGH);
      break;
    case '0':
      Serial.println("0");
      digitalWrite(LEDPin, LOW);
      break;
  }
}
```

Neste exercício programamos o LED de modo que ao inserir o valor 1 (ligar LED) ou 0 (desligar LED).

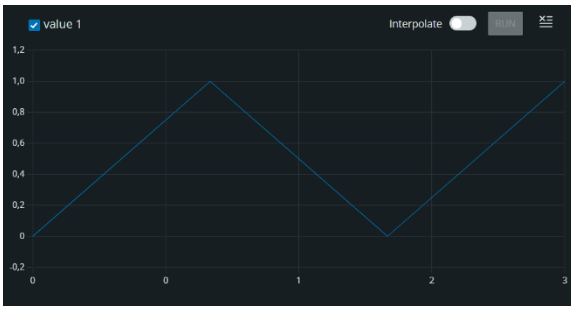
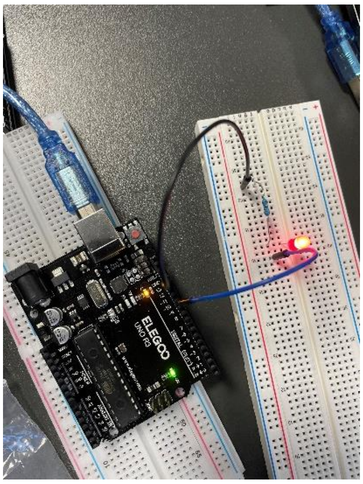


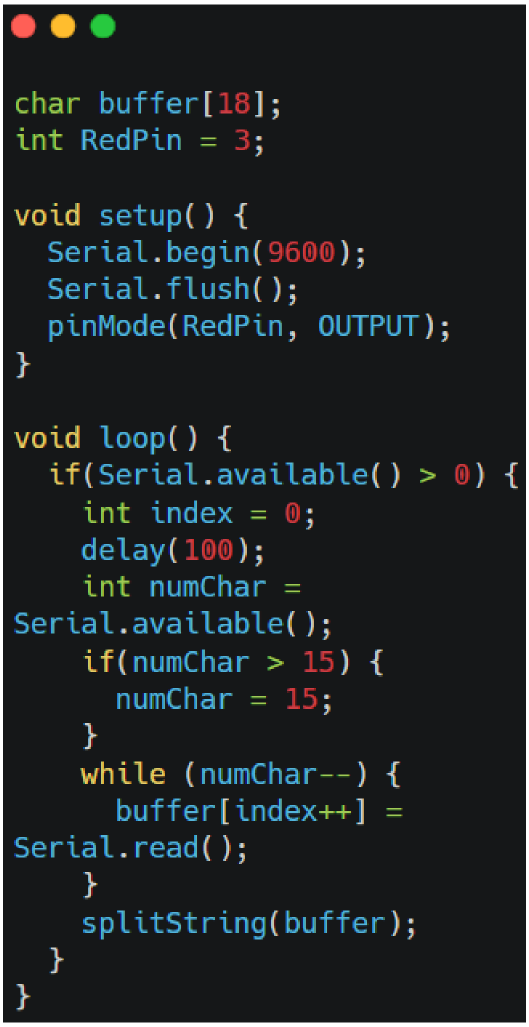
Figura 1 - Serial Plotter

Exercício 2

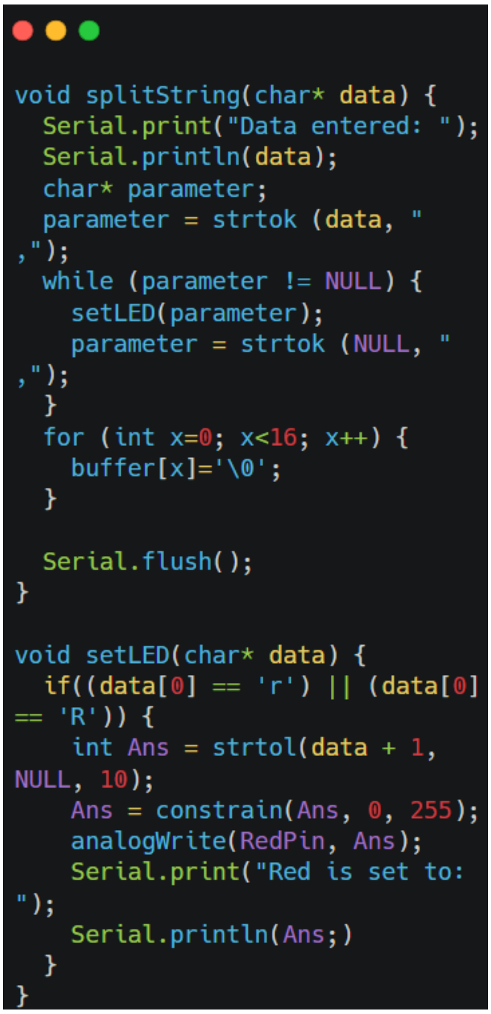
Montagem do circuito e respetivos testes:



Código Utilizado:



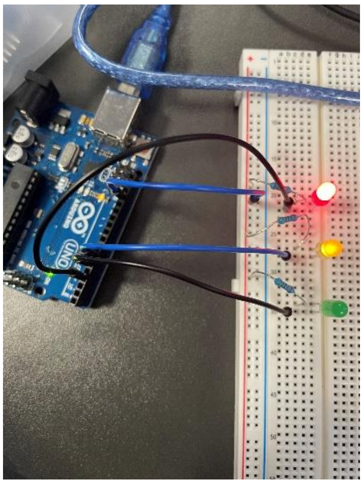
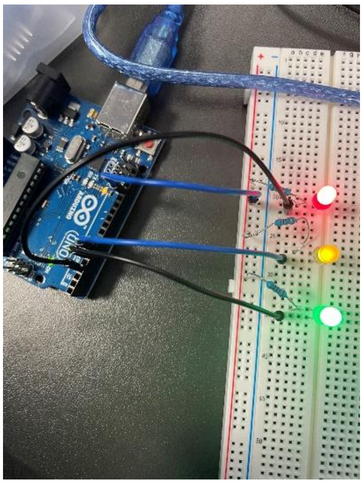
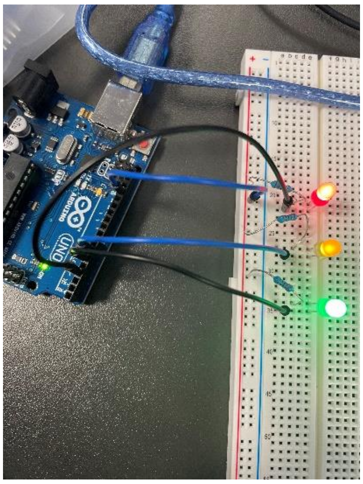
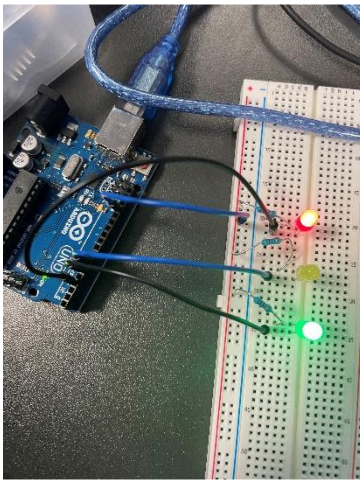
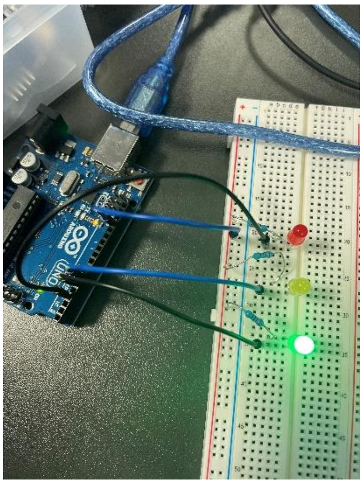
```
char buffer[18];
int RedPin = 3;

void setup() {
  Serial.begin(9600);
  Serial.flush();
  pinMode(RedPin, OUTPUT);
}

void loop() {
  if(Serial.available() > 0) {
    int index = 0;
    delay(100);
    int numChar =
Serial.available();
    if(numChar > 15) {
      numChar = 15;
    }
    while (numChar--) {
      buffer[index++] =
Serial.read();
    }
    splitString(buffer);
  }
}
```



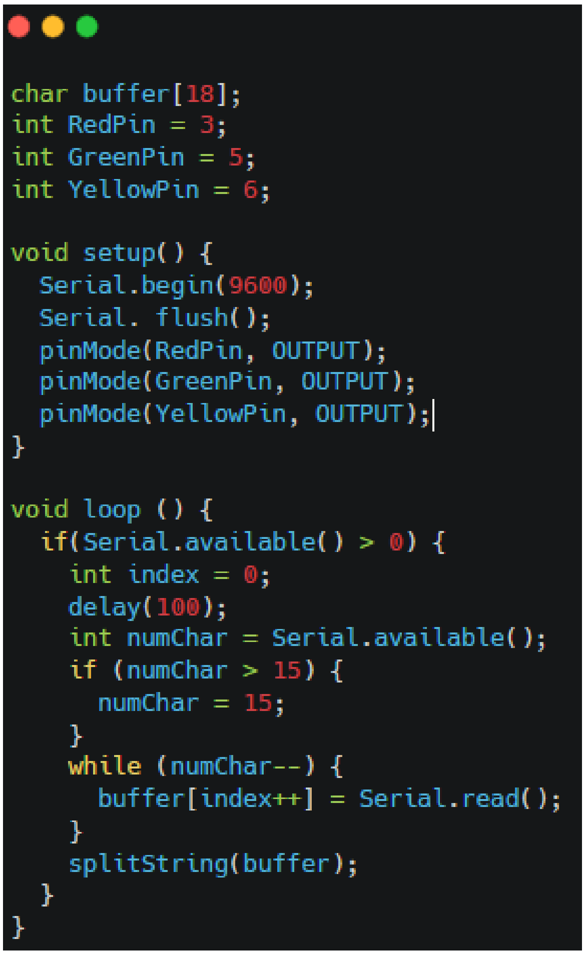
```
void splitString(char* data) {
  Serial.print("Data entered: ");
  Serial.println(data);
  char* parameter;
  parameter = strtok (data, "
,");
  while (parameter != NULL) {
    setLED(parameter);
    parameter = strtok (NULL, "
,");
  }
  for (int x=0; x<16; x++) {
    buffer[x]='\0';
  }

  Serial.flush();
}

void setLED(char* data) {
  if((data[0] == 'r') || (data[0]
== 'R')) {
    int Ans = strtol(data + 1,
NULL, 10);
    Ans = constrain(Ans, 0, 255);
    analogWrite(RedPin, Ans);
    Serial.print("Red is set to:
");
    Serial.println(Ans;)
  }
}
```

Exercício 3

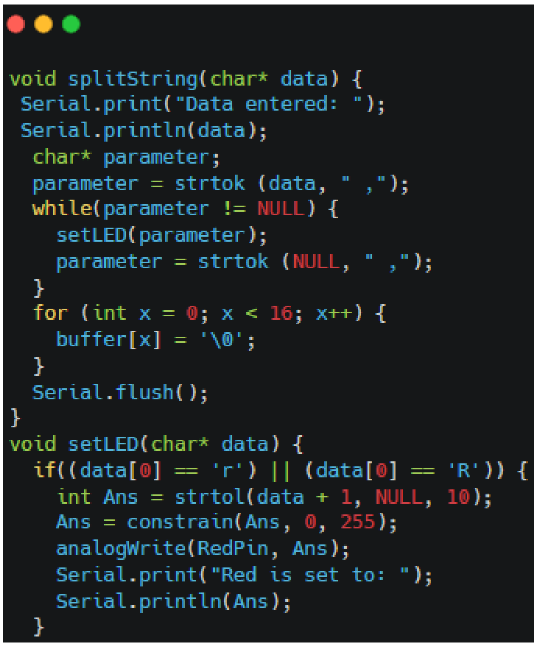
Montagem do circuito e respetivos testes:

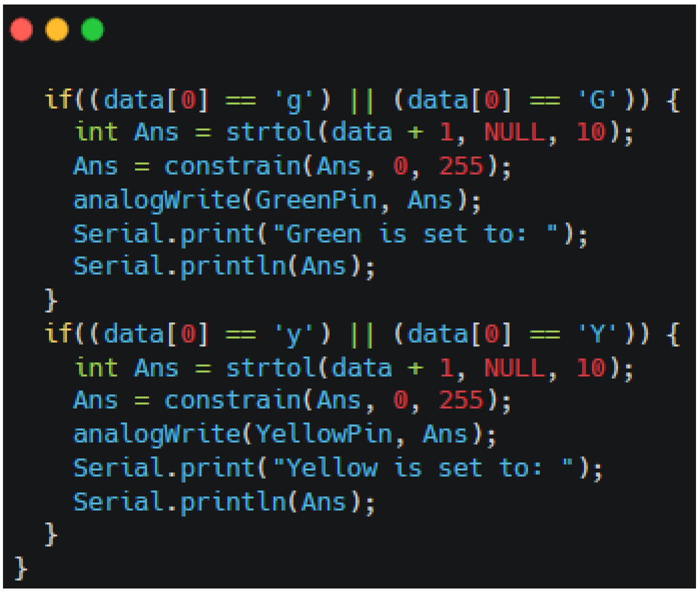
Código Utilizado:



```
char buffer[18];
int RedPin = 3;
int GreenPin = 5;
int YellowPin = 6;

void setup() {
  Serial.begin(9600);
  Serial. flush();
  pinMode(RedPin, OUTPUT);
  pinMode(GreenPin, OUTPUT);
  pinMode(YellowPin, OUTPUT);
}

void loop () {
  if(Serial.available() > 0) {
    int index = 0;
    delay(100);
    int numChar = Serial.available();
    if (numChar > 15) {
      numChar = 15;
    }
    while (numChar--) {
      buffer[index++] = Serial.read();
    }
    splitString(buffer);
  }
}
```



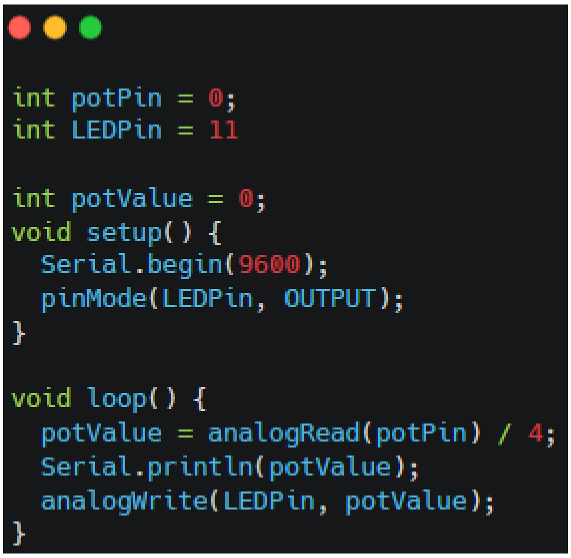
```
void splitString(char* data) {
 Serial.print("Data entered: ");
 Serial.println(data);
  char* parameter;
  parameter = strtok (data, " ,");
  while(parameter != NULL) {
    setLED(parameter);
    parameter = strtok (NULL, " ,");
  }
  for (int x = 0; x < 16; x++) {
    buffer[x] = '\0';
  }
  Serial.flush();
}
void setLED(char* data) {
  if((data[0] == 'r') || (data[0] == 'R')) {
    int Ans = strtol(data + 1, NULL, 10);
    Ans = constrain(Ans, 0, 255);
    analogWrite(RedPin, Ans);
    Serial.print("Red is set to: ");
    Serial.println(Ans);
  }
```



```
  if((data[0] == 'g') || (data[0] == 'G')) {
    int Ans = strtol(data + 1, NULL, 10);
    Ans = constrain(Ans, 0, 255);
    analogWrite(GreenPin, Ans);
    Serial.print("Green is set to: ");
    Serial.println(Ans);
  }
  if((data[0] == 'y') || (data[0] == 'Y')) {
    int Ans = strtol(data + 1, NULL, 10);
    Ans = constrain(Ans, 0, 255);
    analogWrite(YellowPin, Ans);
    Serial.print("Yellow is set to: ");
    Serial.println(Ans);
  }
}
```

Exercício 4

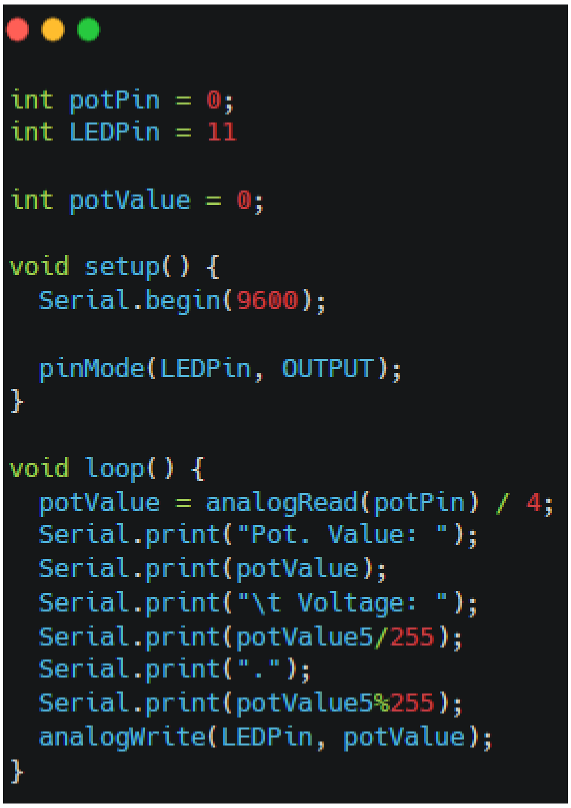
Código Utilizado:



```
int potPin = 0;
int LEDPin = 11

int potValue = 0;
void setup() {
  Serial.begin(9600);
  pinMode(LEDPin, OUTPUT);
}

void loop() {
  potValue = analogRead(potPin) / 4;
  Serial.println(potValue);
  analogWrite(LEDPin, potValue);
}
```

Exercício 5

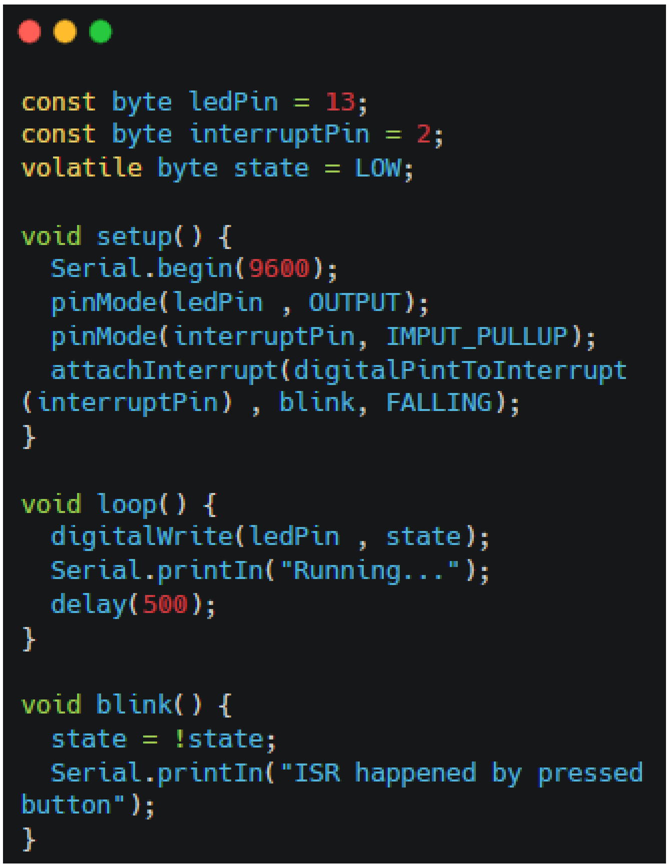
Código Utilizado:



```
int potPin = 0;
int LEDPin = 11

int potValue = 0;

void setup() {
  Serial.begin(9600);

  pinMode(LEDPin, OUTPUT);
}

void loop() {
  potValue = analogRead(potPin) / 4;
  Serial.print("Pot. Value: ");
  Serial.print(potValue);
  Serial.print("\t Voltage: ");
  Serial.print(potValue5/255);
  Serial.print(".");
  Serial.print(potValue5%255);
  analogWrite(LEDPin, potValue);
}
```

Exercício 6

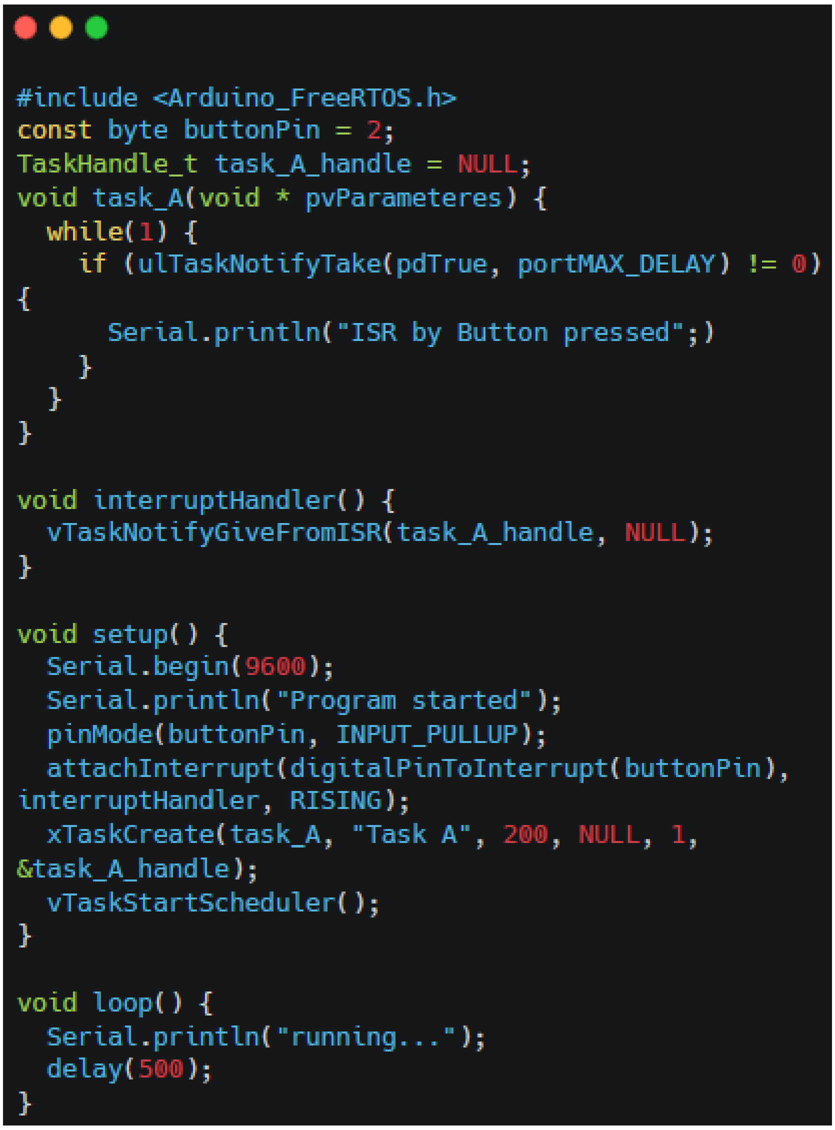
Código Utilizado:



```
const byte ledPin = 13;
const byte interruptPin = 2;
volatile byte state = LOW;

void setup() {
  Serial.begin(9600);
  pinMode(ledPin , OUTPUT);
  pinMode(interruptPin, IMPUT_PULLUP);
  attachInterrupt(digitalPintToInterrupt
(interruptPin) , blink, FALLING);
}

void loop() {
  digitalWrite(ledPin , state);
  Serial.printIn("Running...");
  delay(500);
}

void blink() {
  state = !state;
  Serial.printIn("ISR happened by pressed
button");
}
```

Exercício 7

Código Utilizado:



```
#include <Arduino_FreeRTOS.h>
const byte buttonPin = 2;
TaskHandle_t task_A_handle = NULL;
void task_A(void * pvParameteres) {
  while(1) {
    if (ulTaskNotifyTake(pdTrue, portMAX_DELAY) != 0)
{
      Serial.println("ISR by Button pressed";)
    }
  }
}

void interruptHandler() {
  vTaskNotifyGiveFromISR(task_A_handle, NULL);
}

void setup() {
  Serial.begin(9600);
  Serial.println("Program started");
  pinMode(buttonPin, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(buttonPin),
interruptHandler, RISING);
  xTaskCreate(task_A, "Task A", 200, NULL, 1,
&task_A_handle);
  vTaskStartScheduler();
}

void loop() {
  Serial.println("running...");
  delay(500);
}
```