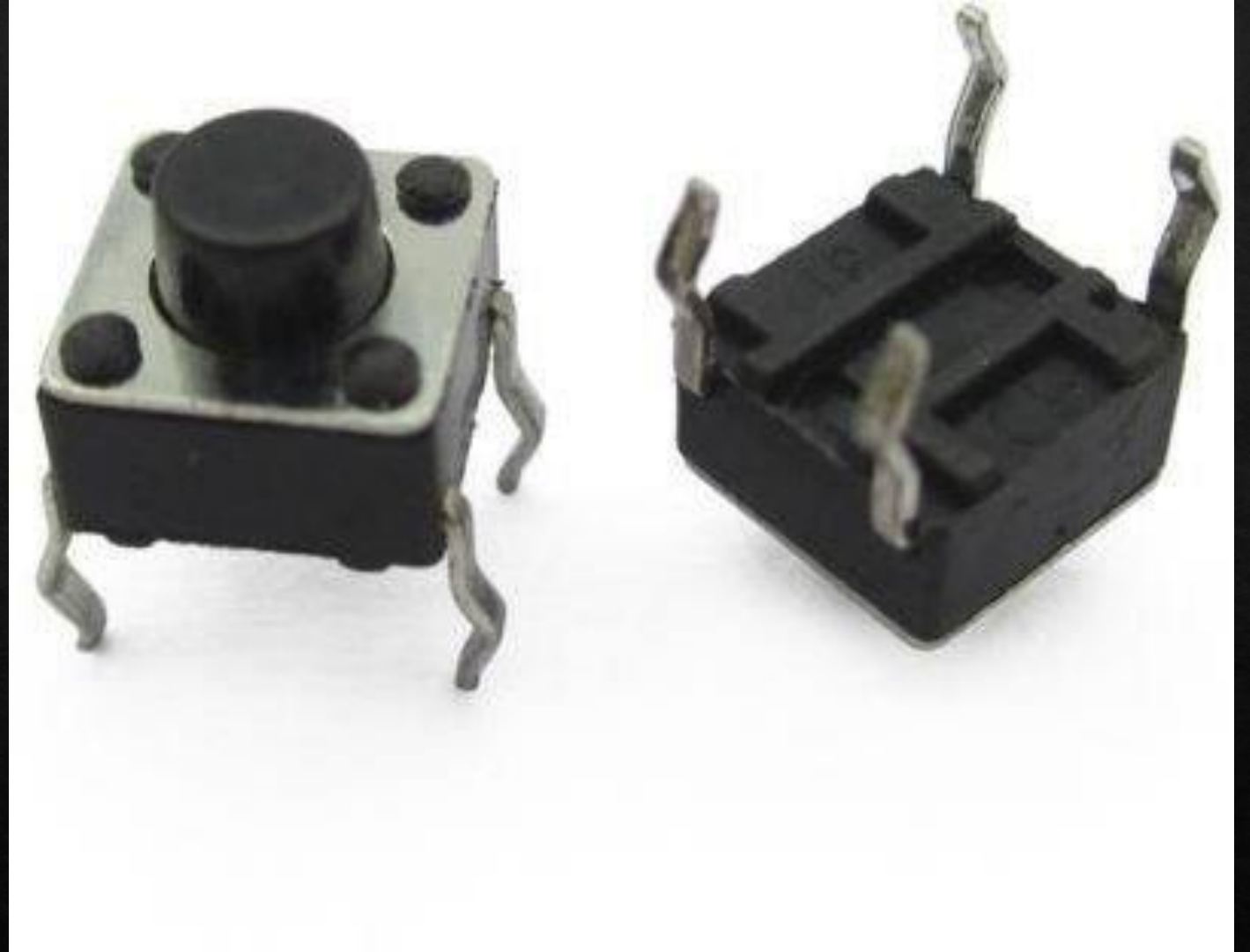
A close-up photograph of an Arduino Uno microcontroller board. The board is blue with various electronic components. A silver USB Type-A connector is plugged into the USB Type-B port on the left. A black component, possibly a USB-to-UART bridge, is plugged into the header pins on the right. The board features the Arduino logo and the text "ARDUINO UNO" and "MADE IN ITALY".

# Arduino – aula 04

# Tópicos a serem abordados

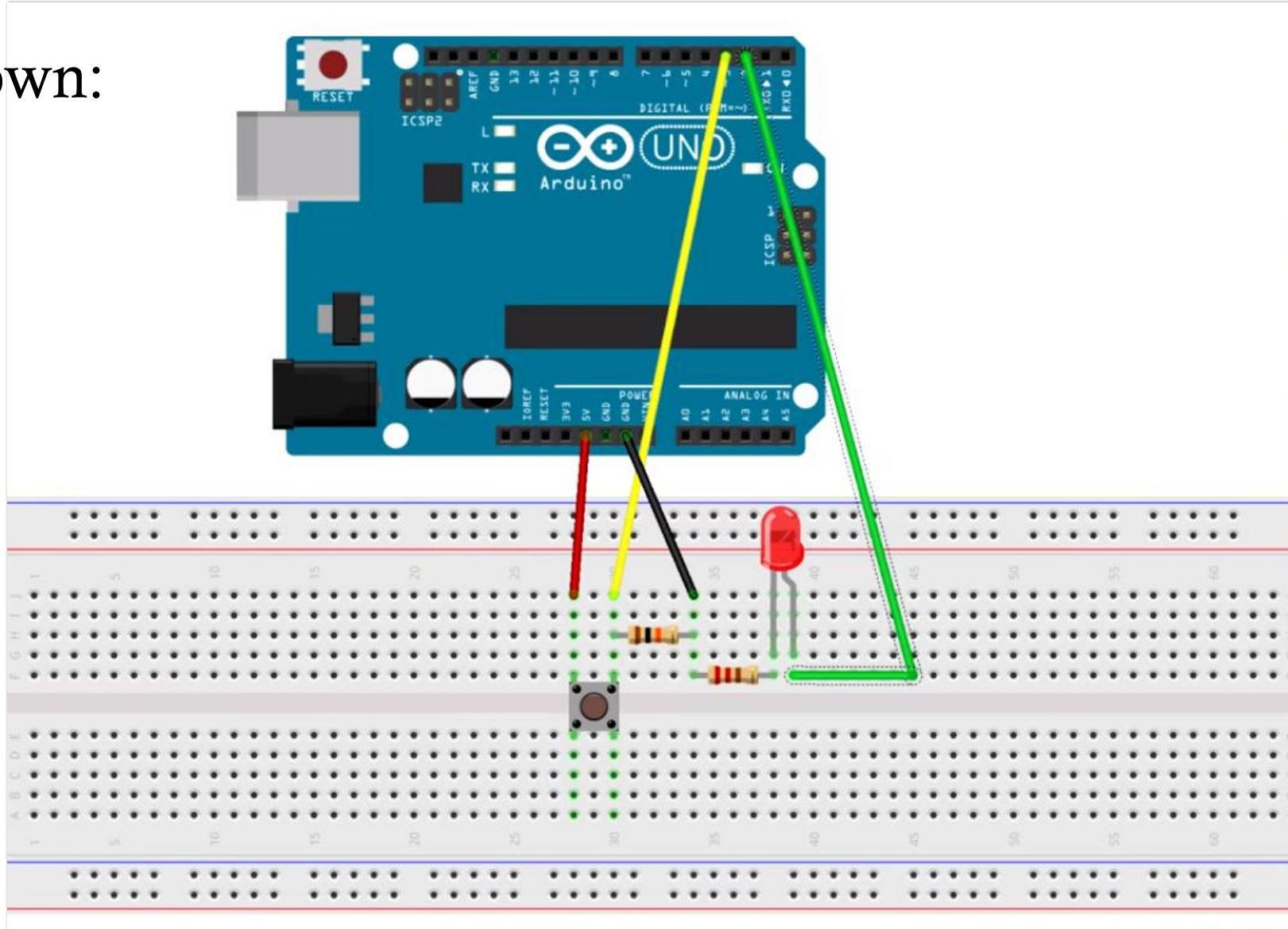
- ◊ Botão
- ◊ Efeito Bounce

# Push Button

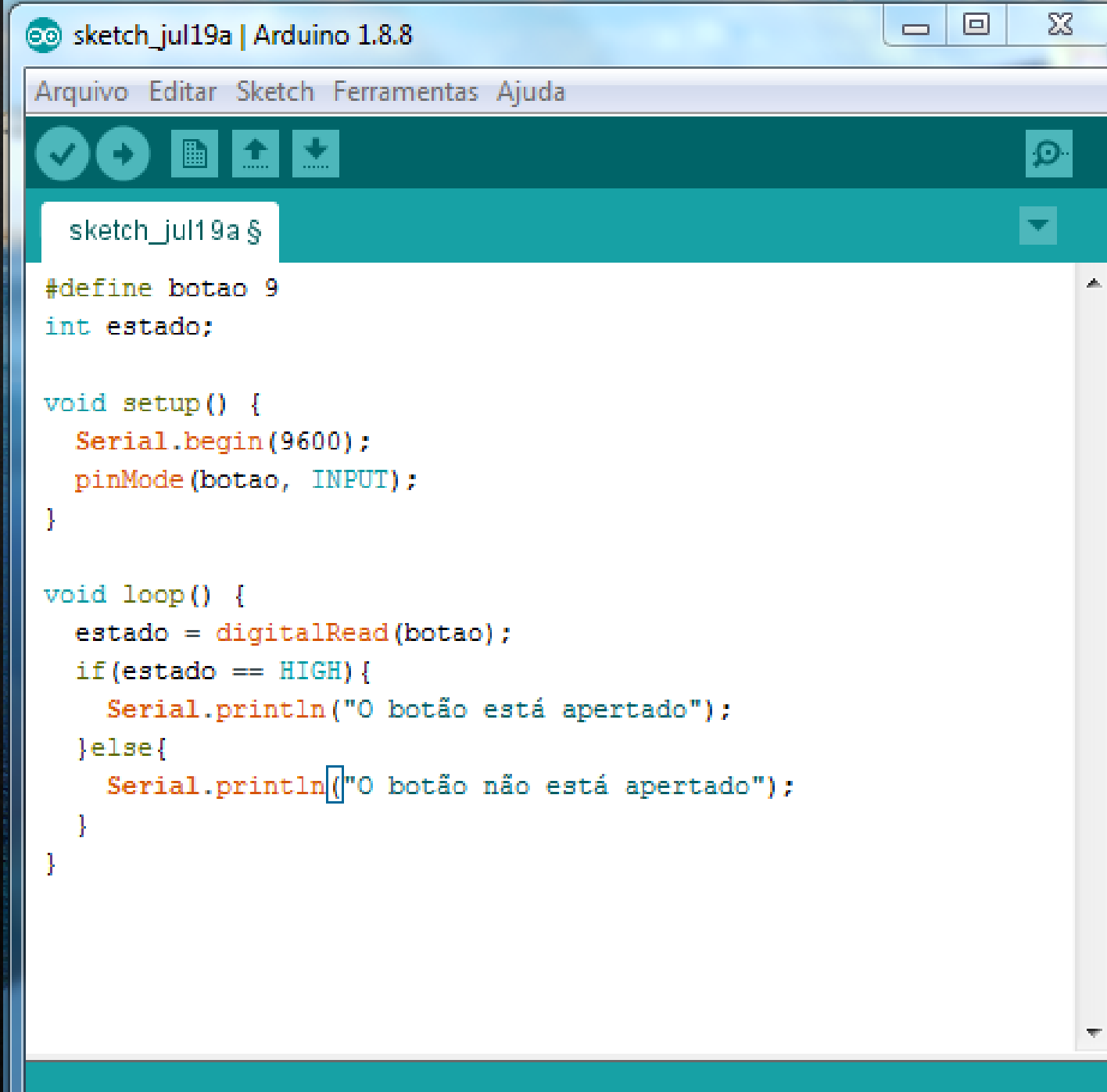




Pull down:



Sketch:

A screenshot of the Arduino IDE interface. The title bar at the top reads "sketch\_jul19a | Arduino 1.8.8". Below the title bar is a menu bar with "Arquivo", "Editar", "Sketch", "Ferramentas", and "Ajuda". Underneath the menu bar is a toolbar with icons for checking, running, saving, and other functions. The main text area shows the following C++ code:

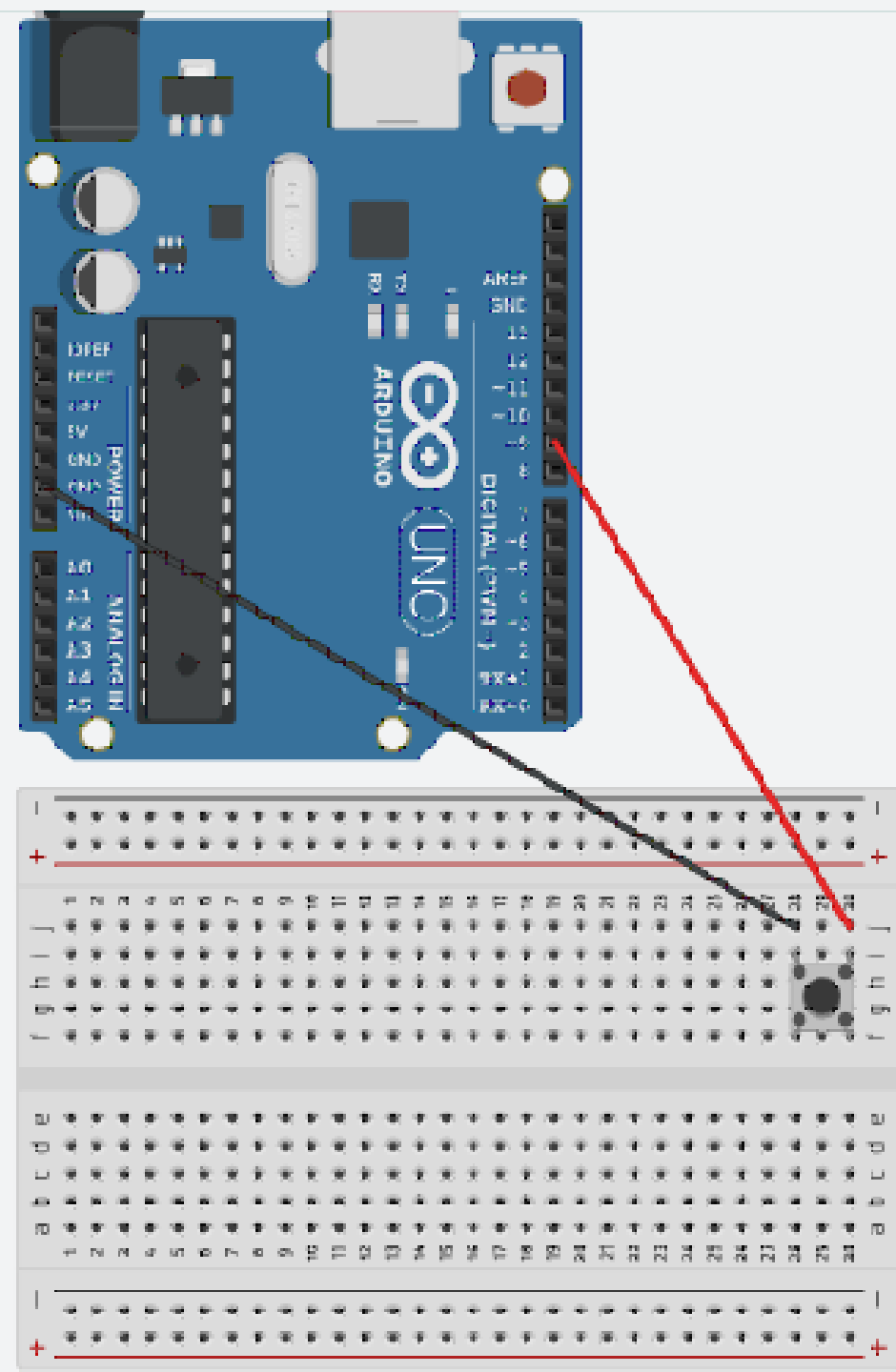
```
#define botao 9
int estado;

void setup() {
  Serial.begin(9600);
  pinMode(botao, INPUT);
}

void loop() {
  estado = digitalRead(botao);
  if(estado == HIGH){
    Serial.println("O botão está apertado");
  }else{
    Serial.println("O botão não está apertado");
  }
}
```

# Comando INPUT\_PULLUP

Se usarmos INPUT\_PULLUP, não é necessário resistor. O circuito ficaria como o da figura ao lado.





sketch\_oct22a\$

```
#define botao 9
int estado;

void setup() {
  Serial.begin(9600);
  pinMode(botao, INPUT_PULLUP);
}

void loop() {
  estado = digitalRead(botao);
  if(estado == LOW){
    Serial.println("O botão está apertado");
  }else{
    Serial.println("O botão não está apertado");
  }
}
```

# Resumo

- ◆ `Serial.begin(9600);` >>> inicializa a comunicação com o monitor serial
- ◆ `if() {}` >>> condicional simples
- ◆ `Serial.print();` >>> mostra uma mensagem no monitor serial
- ◆ `Serial.println();` >>> mostra uma mensagem no monitor serial e pula linha
- ◆ `#define` >>> cria uma definição