



FIAP

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

DISRUPTIVE ARCHITECTURES: IOT, IOB & IA

03 – Arduino e Ecossistemas

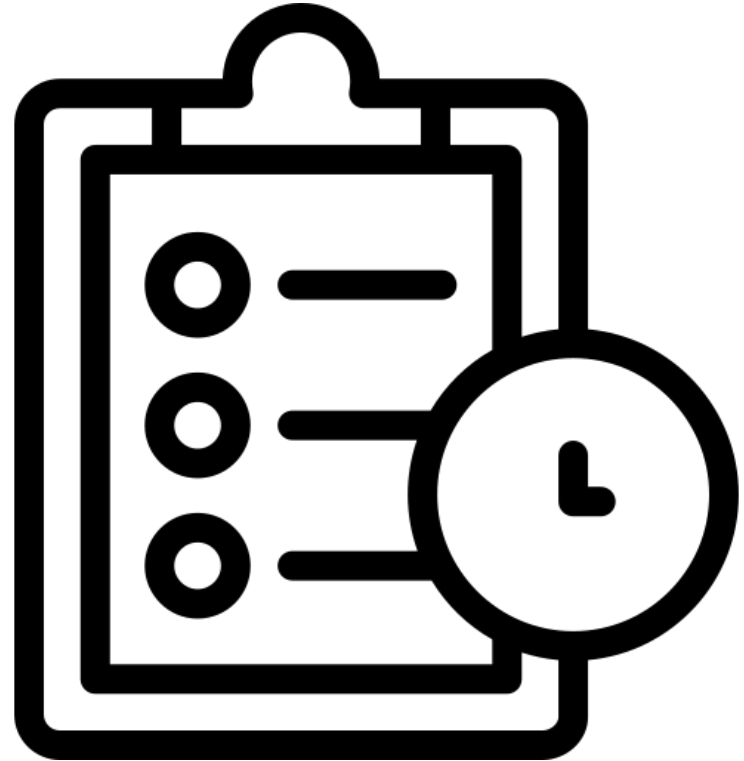


Prof. Airton Y. C. Toyofuku



profairton.toyofuku@fiap.com.br

- O que é o Arduino?
- Como surgiu o Arduino?
- Arquitetura de hardware do Arduino;
- Modelos de placas Arduino;
- Ambiente de desenvolvimento do Arduino;
- Laboratório;
- Exercício;



O que é Arduino?

Basicamente o Arduino é uma plataforma de prototipagem “Open Source” de eletrônica que foi desenvolvida para fins **educacionais**, para **projetistas amadores (Makers)** e facilitar o desenvolvimento de **provas de conceitos (POCs)**.

1

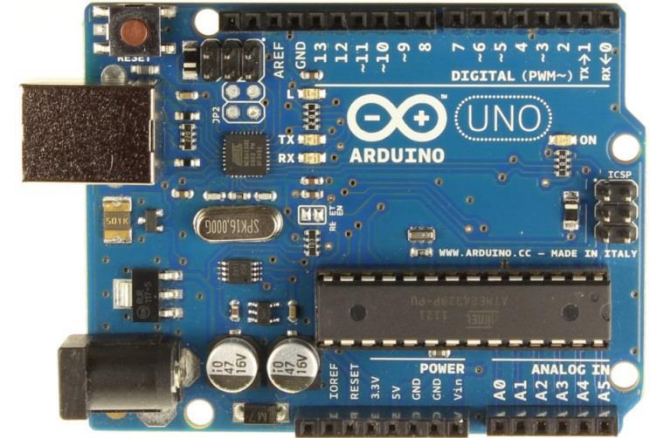
Usa
uControladores
AVR Atmega

2

Possui uma
arquitetura
Harvard de
instruções de 8
bits

3

Versão UNO:
ATmega328p |
16 MHz | 1kB
EEPROM | 2kB
SRAM | 32 kB



Fonte: <https://www.arduino.cc/>

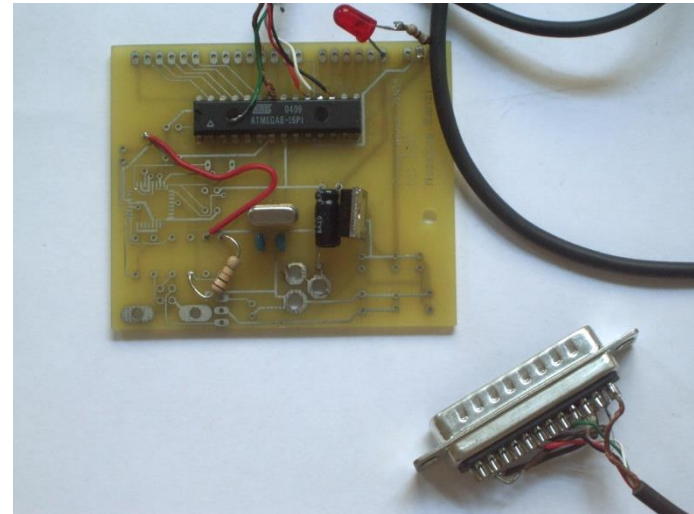
Como surgiu o Arduino?

O **Arduino** foi criado em 2005 por um grupo de 5 pesquisadores : Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis.

O objetivo era elaborar um dispositivo que fosse ao mesmo tempo barato, funcional e fácil de programar, sendo dessa forma acessível a estudantes e projetistas amadores.



Fonte: <https://linuxhint.com/who-invented-arduino/>



Primeiro protótipo
2005

Arquitetura de hardware do Arduino



Fonte: <https://www.arduino.cc/>

1. Conector USB para o cabo tipo AB
2. Botão de reset
3. Pinos de entrada e saída digital e PWM
4. LED verde de placa ligada
5. LED laranja conectado ao pin13
6. ATmega encarregado da comunicação com o computador
7. LED TX (transmissor) e RX (receptor) da comunicação serial
8. Porta ICSP para programação serial
9. Microcontrolador ATmega 328, cérebro do Arduino
10. Cristal de quartzo 16Mhz
11. Regulador de tensão
12. Conector Jack fêmea 2,1mm com centro positivo
13. Pinos de tensão e terra
14. Pinos de entrada analógica

Modelos de placas do Arduino



Microcontrolador	ATmega328P	ATmega32u4	Intel Curie	ATmega32u4
Tensão de operação	5V	5V	3.3V (5V tolerant I/O)	5V
Tensão de alimentação	7-12V	7-12V	7-12V	
Pinos I/O digital	14 (of which 6 provide PWM output)	20	14 (of which 4 provide PWM output)	
Pinos I/O PWM digital	6	7	4	
Pinos analógicos	6	12	6	
Corrente DC por pino I/O	20mA	40mA	20mA	
Corrente DC por pino I/O de 3,3V	50mA	50mA		

Modelos de placas do Arduino



Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader	32 KB (ATmega32u4) of which 4 KB used by bootloader	196 KB	32 KB of which 4 KB used by bootloader
SRAM	2 KB (ATmega328P)	2.5 KB (ATmega32u4)	24KB	2.5 KB
EEPROM	1 KB (ATmega328P)	1 KB (ATmega32u4)		1 KB
Clock Speed	16 MHz	16 MHz	32Mhz	16 MHz
Peso	25g	20g	34g	53g
Features			Bluetooth LE, 6-axis accelerometer/gyro	Analog joystick; Microphone; Light sensor; Temperature sensor ; three-axis accelerometer; Buzzer

Modelos de placas do Arduino



Microcontrolador	ATmega32U4	ATmega328
Tensão de operação	5V	5V
Tensão de alimentação	7-12V	
Pinos I/O digital	20	22
Pinos I/O PWM digital	7	6
Pinos analógicos	12	8
Corrente DC por pino I/O	20mA	40mA
Corrente DC por pino I/O de 3,3V	50mA	

Modelos de placas do Arduino



Flash Memory	32 KB (ATmega32U4) of which 4 KB used by bootloader	32 KB of which 2 KB used by bootloader
SRAM	2.5 KB (ATmega32U4)	2 KB
EEPROM	1 KB (ATmega32U4)	1 KB
Clock Speed	16 MHz	16 MHz
Peso	13g	7g
Comprimento	48 mm	45 mm
Largura	18 mm	18 mm

Modelos de placas do Arduino



Microcontrolador	ATmega2560	ATSAMD21G18, 32-Bit ARM Cortex M0+	AT91SAM3X8E
Tensão de operação	5V	3,3V	3,3V
Tensão de alimentação	7-12V		7-12V
Pinos I/O digital	54	20	54
Pinos I/O PWM digital	15	7	12
Pinos analógicos	16	6, 12-bit ADC channels	
Corrente DC por pino I/O	20mA	7mA	130 mA (juntos)
Corrente DC por pino I/O de 3,3V	50mA		800 mA

Modelos de placas do Arduino



Flash Memory	256 KB of which 8 KB used by bootloader	256 KB	512 KB
SRAM	8 KB	32 KB	96 KB
EEPROM	4 KB		
Clock Speed	16 MHz	48 MHz	84 MHz
Peso	37 g	12g	36g

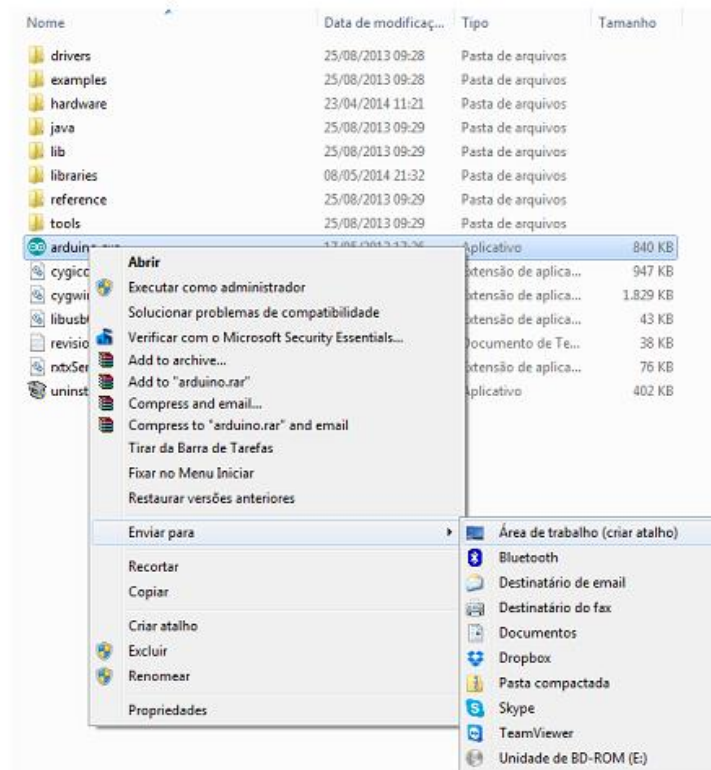
Ambiente de desenvolvimento do Arduino

Ambiente integrado de Desenvolvimento (IDE)
Pode ser gratuitamente baixado do site www.arduino.cc



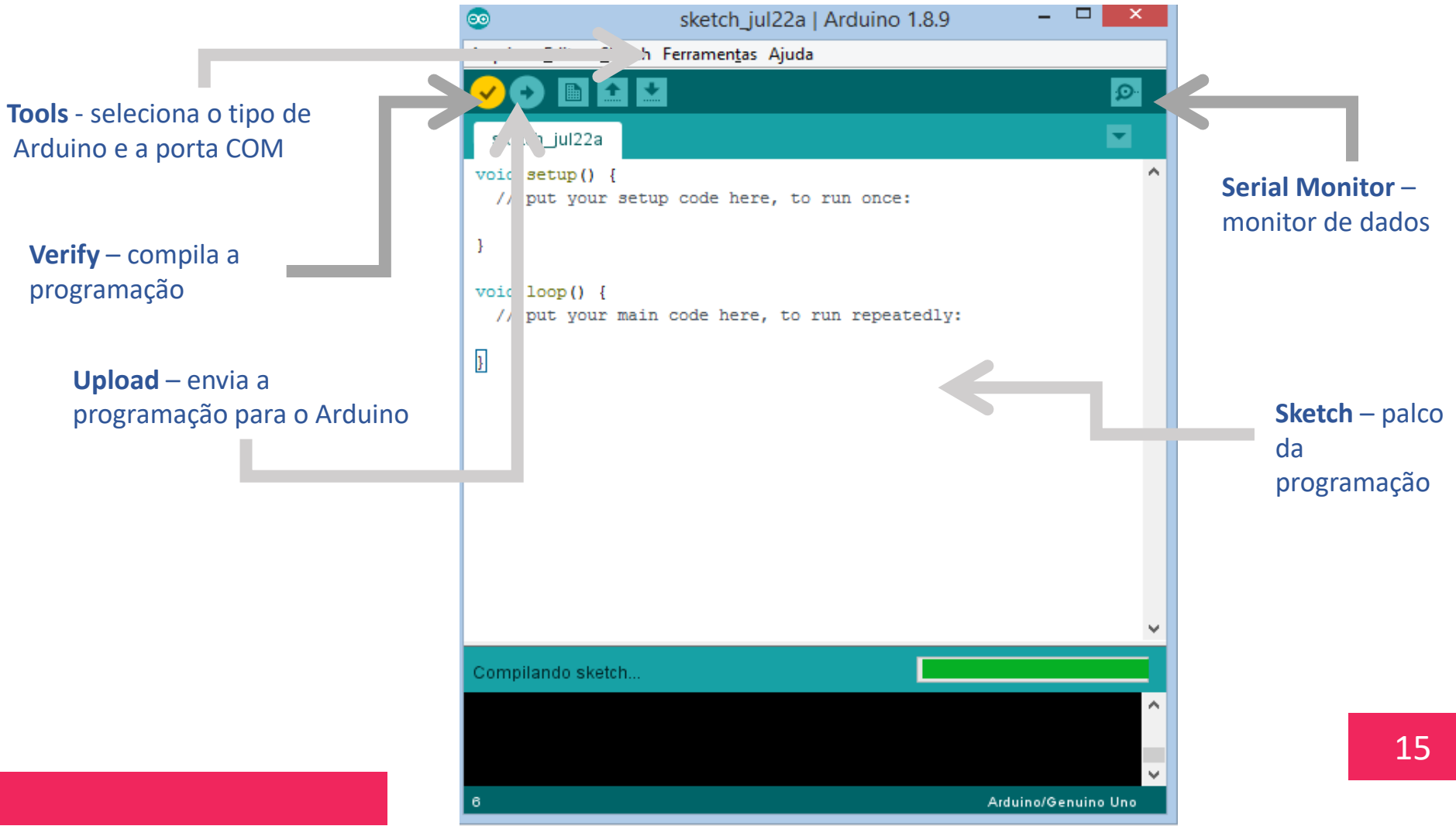
Ambiente de desenvolvimento do Arduino

Agora basta criar um atalho da IDE na área de trabalho e você já poderá programar sua placa!

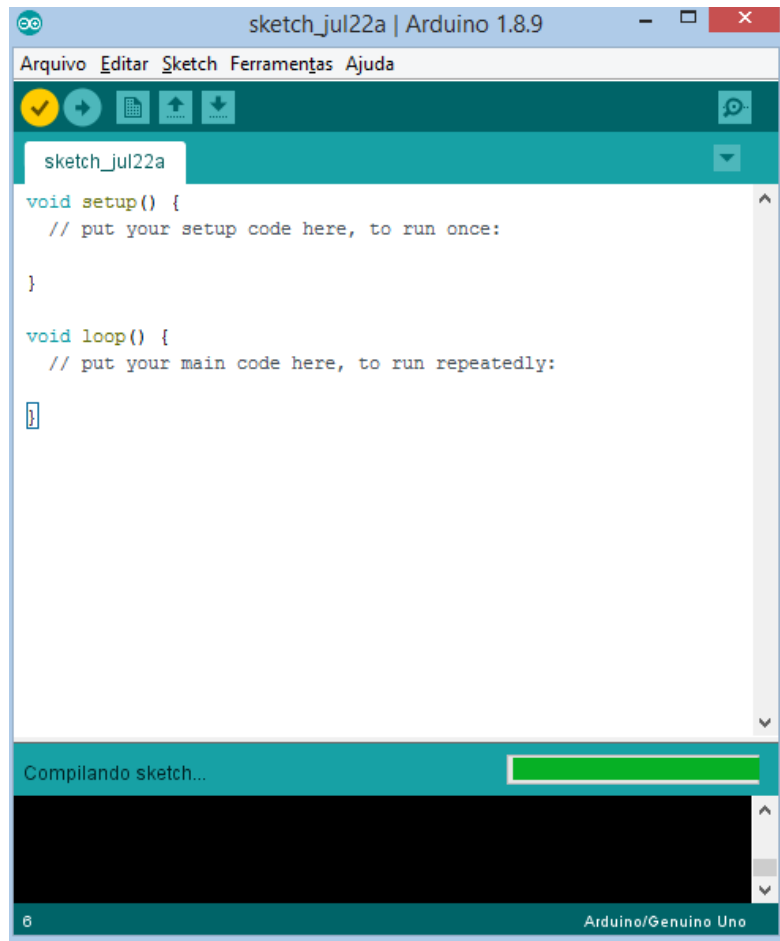


Fonte: Autor

Ambiente de desenvolvimento do Arduino



Ambiente de desenvolvimento do Arduino

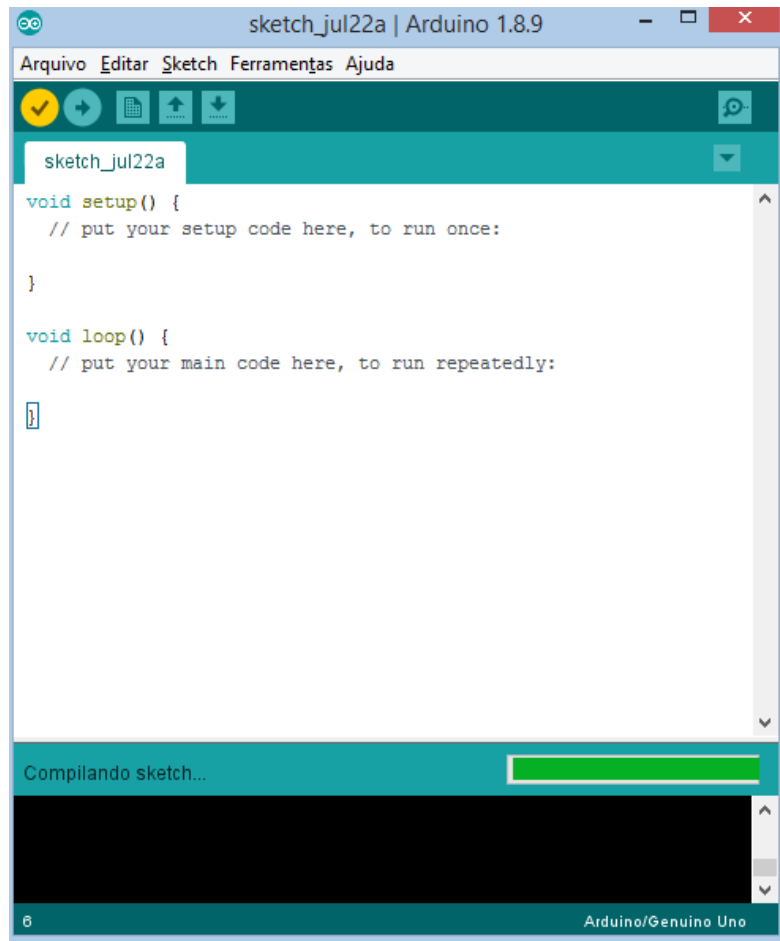


O IDE é muito simples e intuitivo. Um programa, que no Arduino é chamado de sketch, apresenta duas funções básicas: `setup()` e `loop()`.

A função **`setup()`** deverá conter o código que irá executar apenas uma vez, quando o sketch iniciar. Normalmente colocamos nesta função as definições iniciais do programa.

```
void setup() {  
    // initialize the LED pin as an output:  
    pinMode(ledPin, OUTPUT);  
    // initialize the pushbutton pin as an input:  
    pinMode(buttonPin, INPUT);  
}
```

Ambiente de desenvolvimento do Arduino

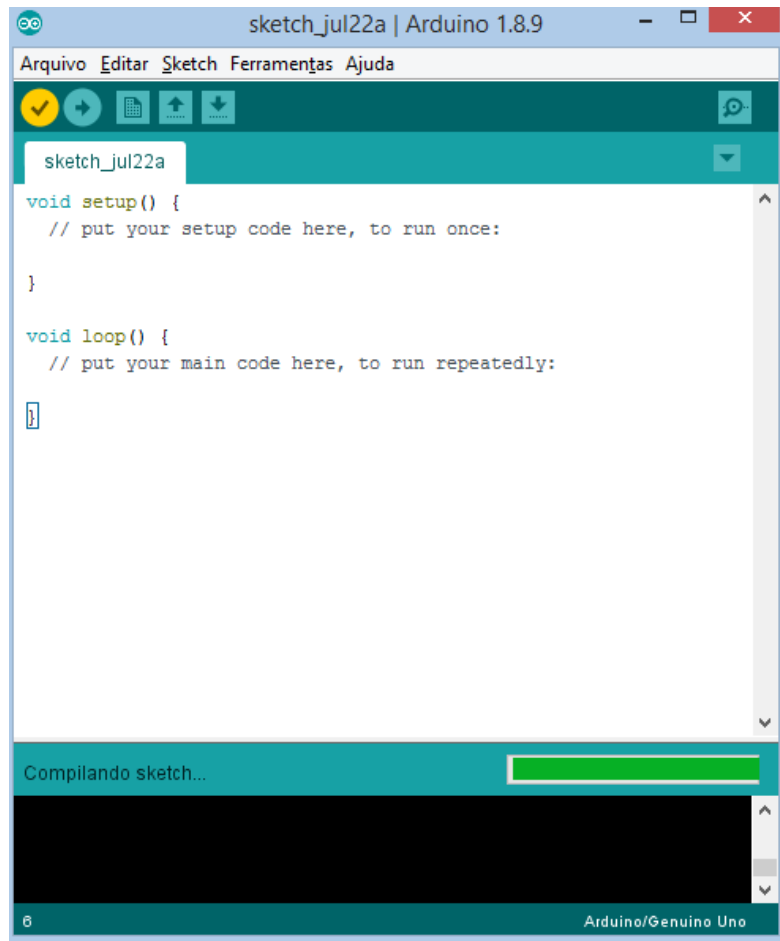


A função **loop()** irá executar continuamente as instruções que estão lá até que outro sketch seja carregado na memória “flash” do Arduino.

É importante notar que no Arduino é possível armazenar e executar um sketch por vez, desta forma, sempre quando transferimos um sketch esse irá substituir o programa que estava anteriormente carregado na memória.

```
void loop() {  
  // read the state of the pushbutton value:  
  buttonState = digitalRead(buttonPin);  
  
  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:  
  if (buttonState == HIGH) {  
    // turn LED on:  
    digitalWrite(ledPin, HIGH);  
  } else {  
    // turn LED off:  
    digitalWrite(ledPin, LOW);  
  }  
}
```

Ambiente de desenvolvimento do Arduino



Também observe que como o sketch fica armazenado na memória “flash”, que é permanente, mesmo quando desligamos o Arduino, o programa continua armazenado e irá entrar novamente em execução quando o Arduino for ligado novamente.

Note também que, nestas duas funções, a palavra reservada **void** indica que as funções não apresentam um valor de retorno, sendo usadas exclusivamente para realizar a execução de um conjunto de instruções.

Laboratório – Piscando LED

Agora que temos acesso ao **TinkerCad**, vamos montar o nosso primeiro circuito. **Um Pisca Led Simples.**

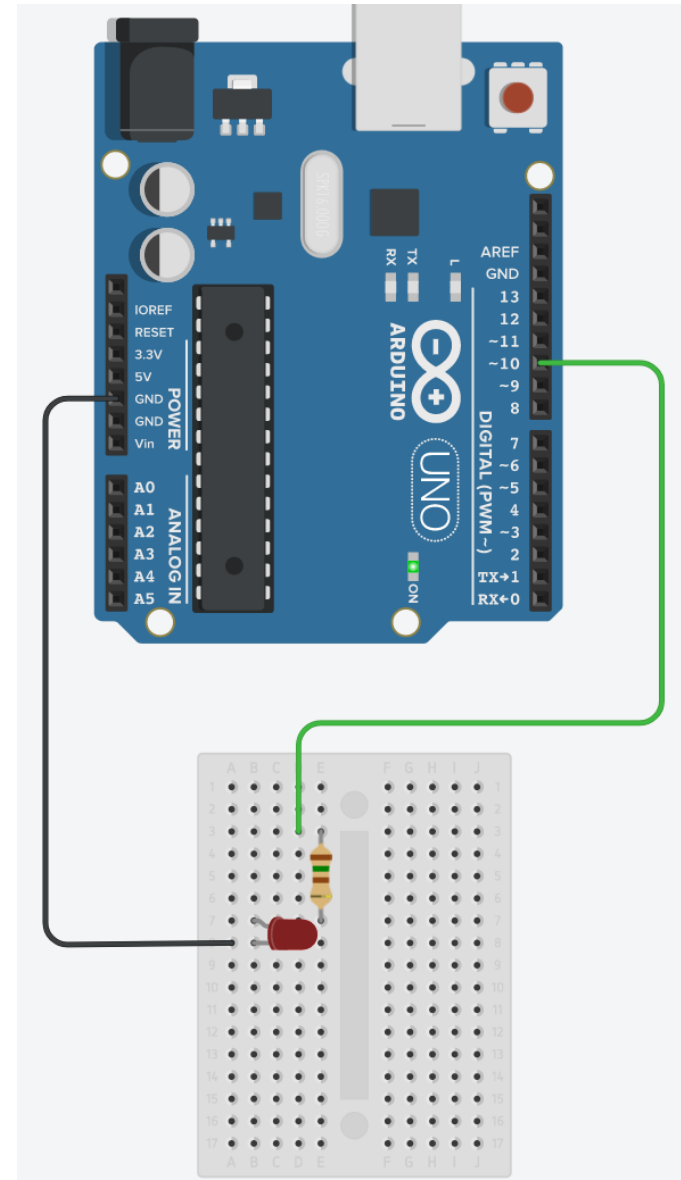
Nesse projeto vamos conhecer a interface de programação do Arduino e entender um poquinho como o hardware de prototipagem funciona.

Material necessário:

- 1 Arduino;
- 1 Resistor de 150;
- 1 Led (qualquer cor);
- 1 Protoboard;
- Jumpers cables.



Link: [Projeto 01 – LED Flasher](#)



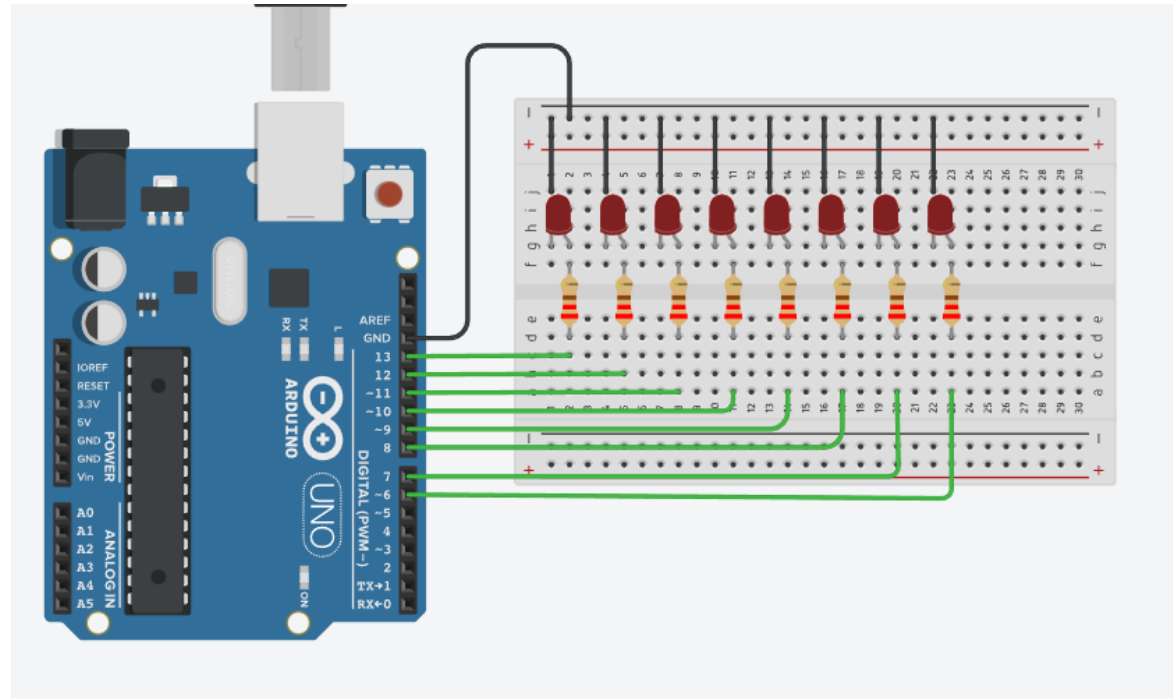
Exercício Desafio

Vamos aplicar o que vimos nessa aula e montar no **um contador em binário**.

Use como base o código que vimos no laboratório e faça com que os LEDs apresentem os números de **0 a 255, em binário, com intervalos de 1 segundo**.

Por onde você começaria a resolver esse exercício? Faça um esboço no papel com ideias e verifique se elas fazem sentido.

Pesquise na internet por problemas semelhantes e tente entender o racional para resolver esse exercício.



Material necessário:

- 1 Arduino;
- 8 Resistores de 220 Ohms;
- 8 Leds Vermelhos;
- 1 Protoboard;
- Jumpers cables.

Copyright © 2023 Prof. Airton Y. C. Toyofuku

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).

This presentation has been designed using images from Flaticon.com
Images from Monty Python's Flying Circle: BBC, 1969. Netflix, 2019
Imagens from Dragon Ball, Saint Seiya: Toei Animation