

**ARDUINO**

**Arduino**

**Gabriel Felipe**

**[akiradev.netlify.app](https://akiradev.netlify.app)**

# Introdução


- **Arduino** é uma empresa de **hardware** e **software** de código aberto, comunidade de projetos e usuários que projeta e fabrica **microcontroladores** de **placa única** e kits de microcontroladores para a construção de dispositivos digitais.
- Seus produtos de hardware são licenciados sob uma licença **CC-BY-SA**, enquanto o software é licenciado sob a **GNU Lesser General Public License** (LGPL) ou a **GNU General Public License** (GPL), permitindo a fabricação de placas Arduino e distribuição de software por qualquer pessoa.
- As placas Arduino estão disponíveis comercialmente no site oficial ou por meio de distribuidores autorizados.

# Design

- Os designs da placa Arduino usam uma variedade de microprocessadores e controladores. As placas são equipadas com conjuntos de pinos de **Input/Output (I/O)** digitais e analógicos que podem ter interface com várias placas de expansão ('shields') ou **breadboards** (para prototipagem) e outros circuitos.
- As placas apresentam interfaces de comunicação seriais, incluindo **Universal Serial Bus (USB)** em alguns modelos, que também são usados para carregar programas.
- Os microcontroladores podem ser programados usando as linguagens de programação C e C++, usando uma API padrão também conhecida como "**linguagem Arduino**".

# Arduino IDE

- Além de usar cadeias de ferramentas de compilador tradicionais, o projeto Arduino fornece um Integrated Development Environment (**IDE**) e uma ferramenta de linha de comando desenvolvida em Go.

A screenshot of the Arduino IDE window titled "Blink | Arduino 1.8.5". The window has a dark teal header bar with icons for file operations and a search icon. Below the header, the code editor shows the "Blink" example code. The code is in C++ and includes comments explaining the setup and loop functions. The status bar at the bottom indicates "32" and "Arduino/Genuino Uno on COM1".

```
Blink | Arduino 1.8.5

Blink $

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {$$
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

32 Arduino/Genuino Uno on COM1
```

# O Projeto Arduino

- O projeto Arduino começou em 2005 como uma ferramenta para estudantes do Interaction Design Institute Ivrea, Itália, com o objetivo de fornecer uma maneira fácil e de baixo custo para iniciantes e profissionais criarem dispositivos que interajam com seu ambiente usando sensores e atuadores.
- Exemplos comuns de tais dispositivos destinados a amadores iniciantes incluem robôs simples, termostatos e detectores de movimento.
- O nome Arduino vem de um bar em Ivrea, Itália, onde alguns dos fundadores do projeto costumavam se reunir. O bar foi nomeado em homenagem a Arduin de Ivrea, que foi o margrave da Marcha de Ivrea e Rei da Itália de 1002 a 1014.

# Hardware

- Arduino é um hardware de código aberto. Os designs de referência de hardware são distribuídos sob uma licença Creative Commons Attribution Share-Alike 2.5 e estão disponíveis no site do Arduino.
- Arquivos de layout e produção para algumas versões do hardware também estão disponíveis.
- Embora os designs de hardware e software estejam disponíveis gratuitamente sob licenças copyleft, os desenvolvedores solicitaram que o nome Arduino seja exclusivo para o produto oficial e não seja usado para trabalhos derivados sem permissão.
- O documento oficial da política sobre o uso do nome Arduino enfatiza que o projeto está aberto para incorporar o trabalho de outros no produto oficial.

# Hardware

- A maioria das placas Arduino consiste em um microcontrolador AVR Atmel de 8 bits (ATmega8, ATmega168, ATmega328, ATmega1280 ou ATmega2560) com quantidades variáveis de memória flash, pinos e recursos.
- Os microcontroladores Arduino são pré-programados com um bootloader que simplifica o upload de programas para a memória flash on-chip.
- O bootloader padrão do Arduino Uno é o Optiboot bootloader.
- As placas Arduino atuais são programadas via Universal Serial Bus (USB), implementado usando chips adaptadores USB para serial, como o FTDI FT232.



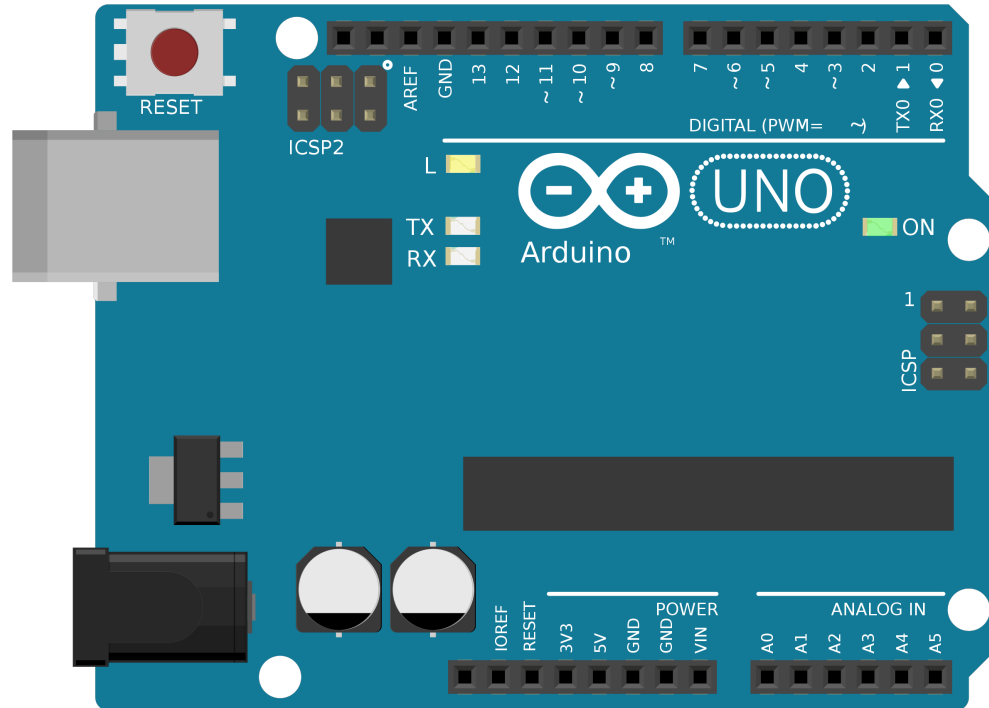
# Hardware

- A placa Arduino expõe a maioria dos pinos de I/O do microcontrolador para uso por outros circuitos.
- O Diecimila, Duemilanove e o Uno atual fornecem 14 pinos de I/O digitais, seis dos quais podem produzir sinais modulados por largura de pulso (PWD) e seis entradas analógicas, que também podem ser usadas como seis pinos de I/O digitais.
- Esses pinos estão na parte superior da placa, por meio de conectores fêmeas de 0.1 pol. (2.54 mm).

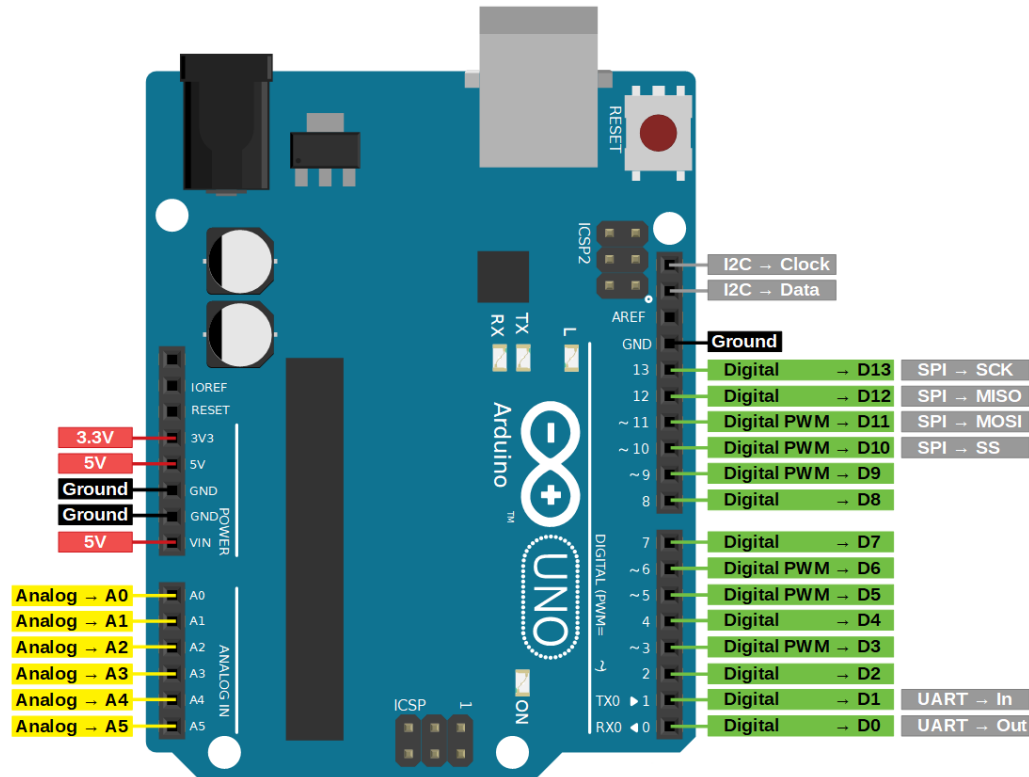
# Arduino UNO

- Nesta apresentação, falaremos sobre o que está na placa Arduino UNO e o que ela pode fazer.
- UNO não é a única placa da família Arduino. Existem outras placas como Arduino Diecimila, Arduino Mini, Arduino Mega e Arduino Nano.
- No entanto, a placa Arduino UNO se tornou mais popular do que outras placas da família porque possui uma documentação muito mais detalhada.
- Isso levou ao aumento da adoção da prototipagem eletrônica, criando uma vasta comunidade de geeks eletrônicos e amadores.

# Arduino UNO



# Arduino UNO



# Arduino UNO

- Componentes da placa Arduino UNO:

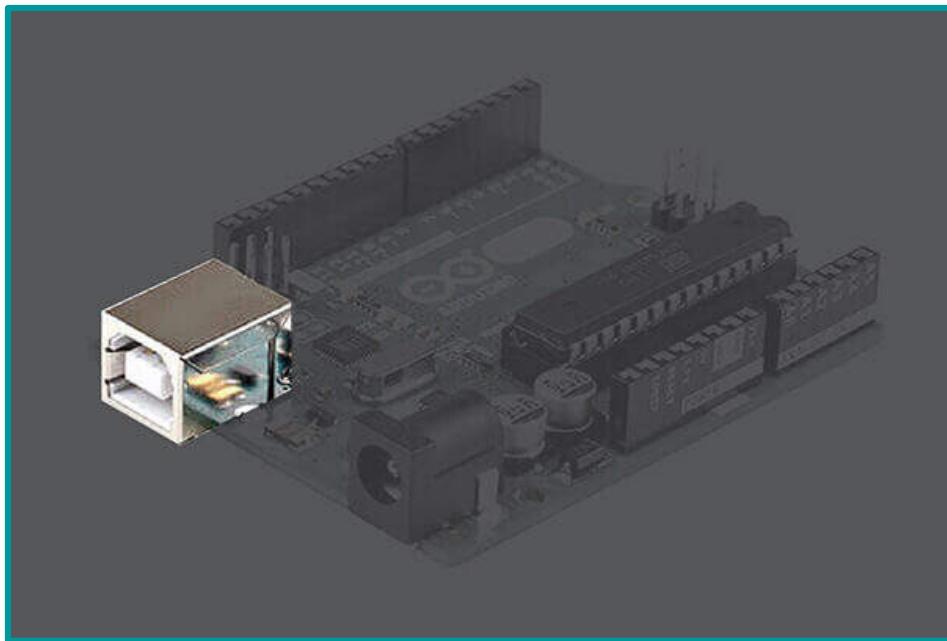


# Arduino UNO

- Os principais componentes da placa Arduino UNO são os seguintes:
  - Conector USB
  - Porta de alimentação
  - Microcontrolador
  - Pinos de entrada analógica
  - Pinos digitais
  - Botão de reset
  - Crystal oscillator
  - Chip de interface USB
  - LEDs TX RX

# Arduino UNO

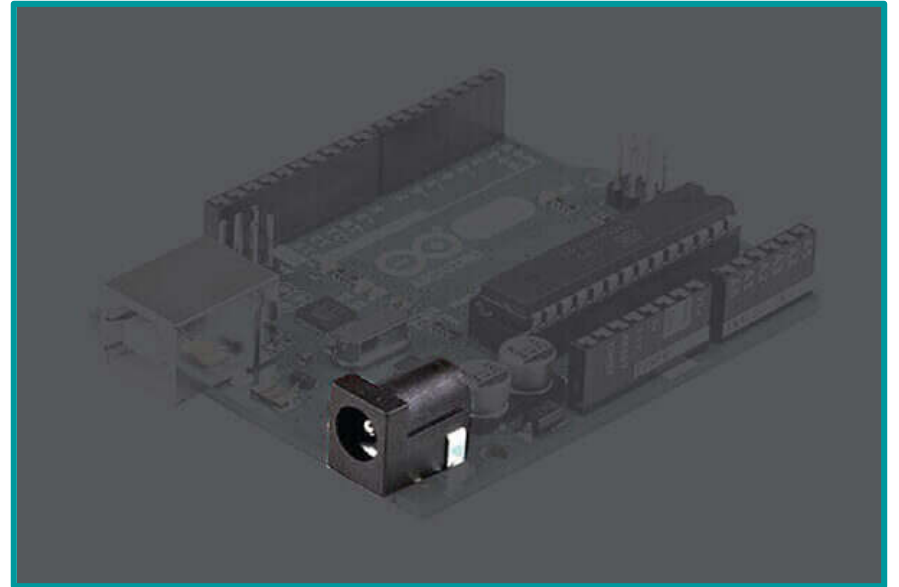
- Esta é uma porta USB de impressora, ela é usada para carregar um programa do IDE do Arduino para a placa do Arduino. A placa também pode ser alimentada por esta porta.



# Arduino UNO

- A placa Arduino pode ser alimentada por um adaptador AC-DC ou bateria. A fonte de alimentação pode ser conectada conectando-se um plugue positivo central de 2.1 mm ao conector de alimentação da placa.

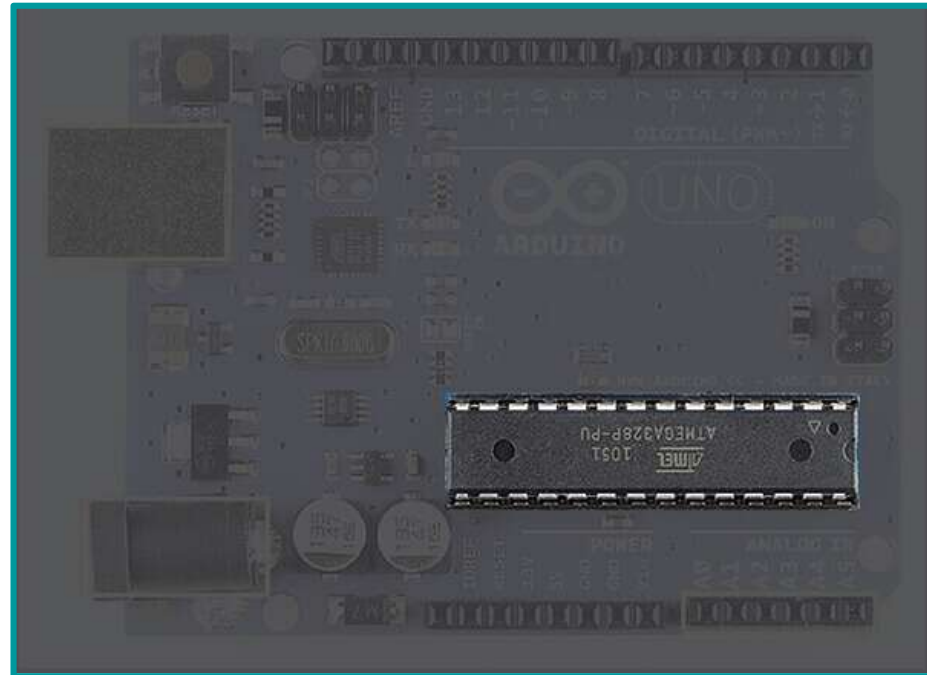
A placa Arduino UNO opera com uma tensão de 5 volts, mas pode suportar uma tensão máxima de 20 volts. Se a placa for fornecida com uma tensão mais alta, existe um regulador de tensão (ele fica entre a porta de alimentação e o conector USB) que protege a placa de queimar.





# Arduino UNO

- Microcontrolador:

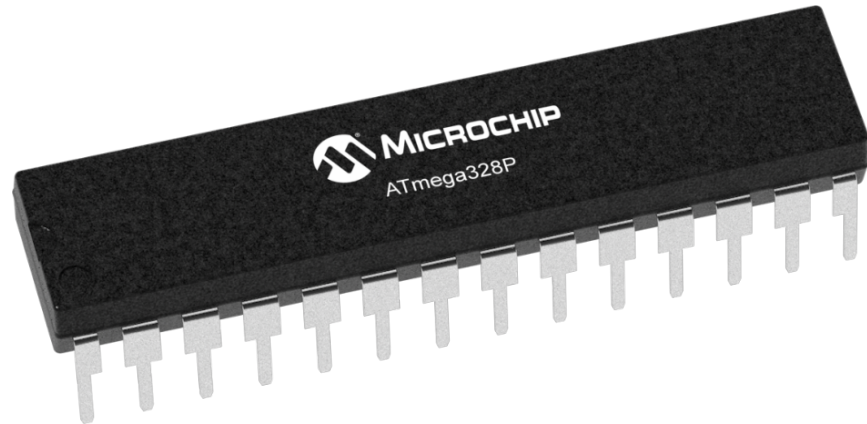


# Arduino UNO

- O microcontrolador é o chip retangular preto mais proeminente com 28 pinos. Pense nele como o cérebro do seu Arduino.
- O microcontrolador usado na placa UNO é o Atmega328P da Atmel (um dos principais fabricantes de microcontroladores).
- O Atmega328P possui os seguintes componentes:
  - Memória flash de 32 KB. O programa carregado do Arduino IDE é armazenado aqui.
  - RAM de 2 KB. Esta é uma memória de tempo de execução.
  - CPU: controla tudo o que acontece dentro do dispositivo. Ele busca as instruções do programa na memória flash e as executa com a ajuda da RAM.
  - Electrically Erasable Programmable Read Only Memory (EEPROM) de 1 KB. Este é um tipo de memória não volátil e mantém os dados mesmo após a reinicialização e reinicialização do dispositivo.

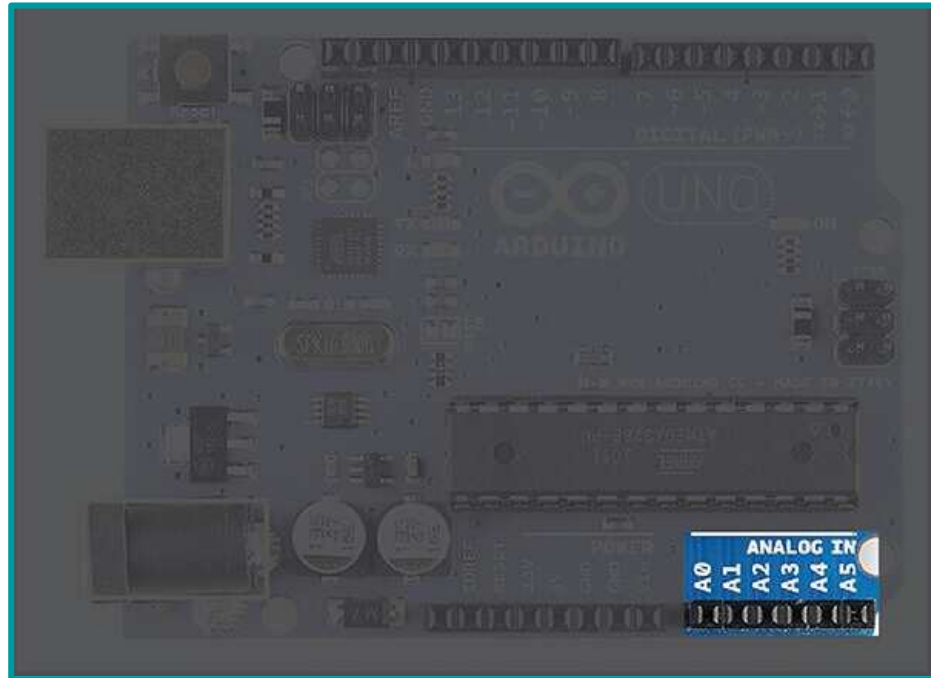
# Arduino UNO

- Atmega328P é pré-programado com bootloader. Isso permite que você carregue diretamente um novo programa Arduino no dispositivo, sem usar nenhum programador de hardware externo, tornando a placa Arduino UNO fácil de usar.



# Arduino UNO

- Pinos de entrada analógica:

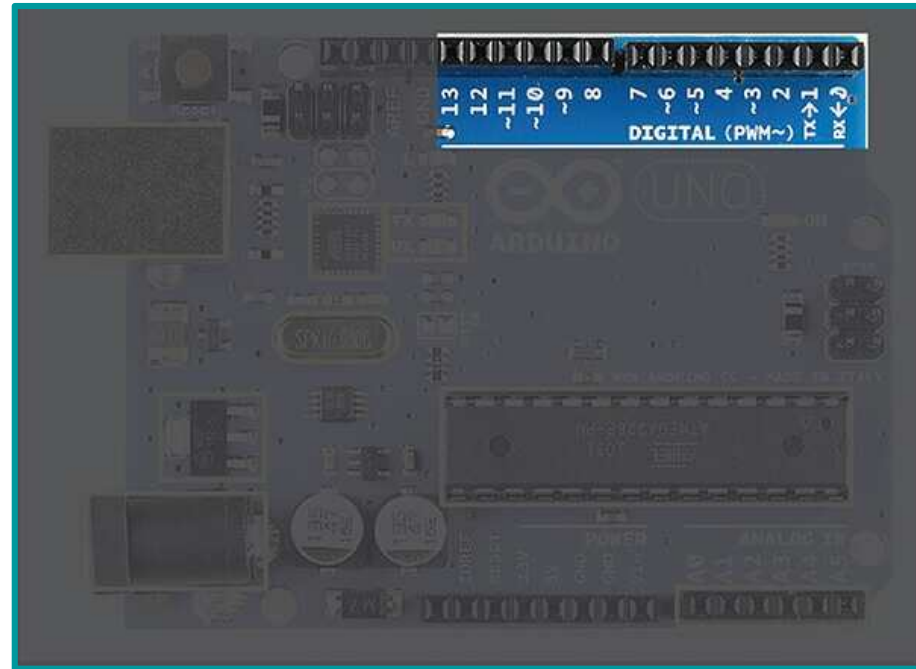


# Arduino UNO

- A placa Arduino UNO tem 6 pinos de entrada analógica, identificados como “Analógico 0 a 5”.
- Esses pinos podem ler o sinal de um sensor analógico como um sensor de temperatura e convertê-lo em um valor digital para que o sistema entenda.
- Esses pinos medem apenas a tensão e não a corrente, porque têm uma resistência interna muito alta. Conseqüentemente, apenas uma pequena quantidade de corrente flui através desses pinos.
- Embora esses pinos sejam rotulados como analógicos e sejam entradas analógicas por padrão, eles também podem ser usados para entrada ou saída digital.

# Arduino UNO

- Pinos digitais:



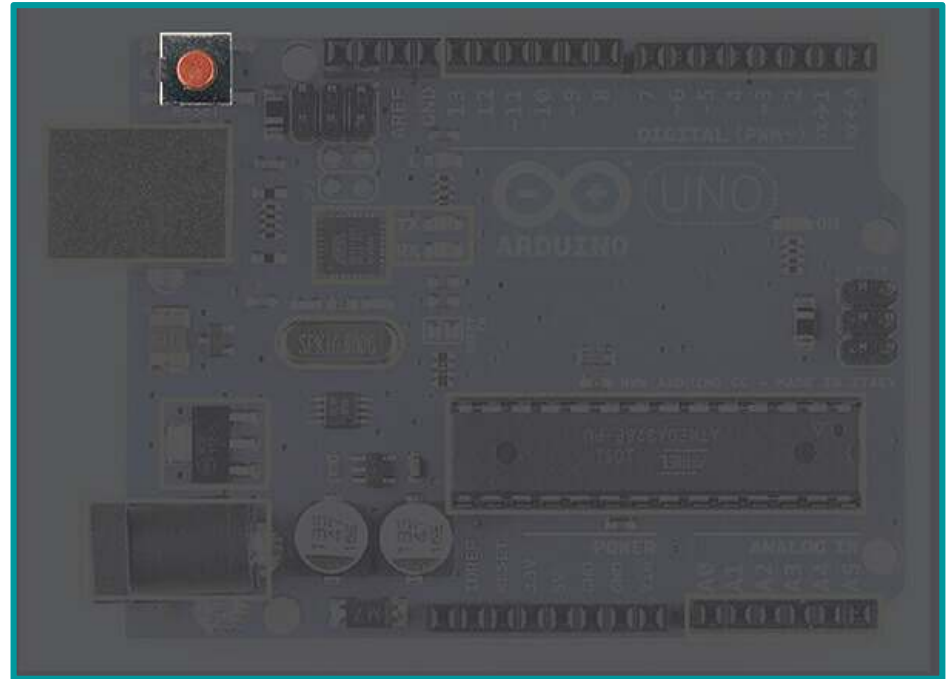
# Arduino UNO

- Você pode encontrar esses pinos identificados como "Digital 0 a 13". Esses pinos podem ser usados como pinos de entrada ou saída. Quando usados como saída, esses pinos atuam como uma fonte de alimentação para os componentes conectados a eles. Quando usados como pinos de entrada, eles leem os sinais do componente conectado a eles.
- Quando os pinos digitais são usados como pinos de saída, eles fornecem 40 miliamperes de corrente a 5 volts, o que é mais do que suficiente para acender um LED.
- Alguns dos pinos digitais são identificados com o símbolo de til (~) próximo aos números dos pinos (números dos pinos 3, 5, 6, 9, 10 e 11). Esses pinos atuam como pinos digitais normais, mas também podem ser usados para Modulação por Largura de Pulso (PWM), que simula a saída analógica como o esmaecimento de um LED.

# Arduino UNO

## ■ Botão de reset:

Quando este botão é apertado, ele envia um pulso lógico para o pino de reinicialização do microcontrolador e executa o programa novamente desde o início. Isso pode ser muito útil se o seu código não se repetir, mas você deseja testá-lo várias vezes.

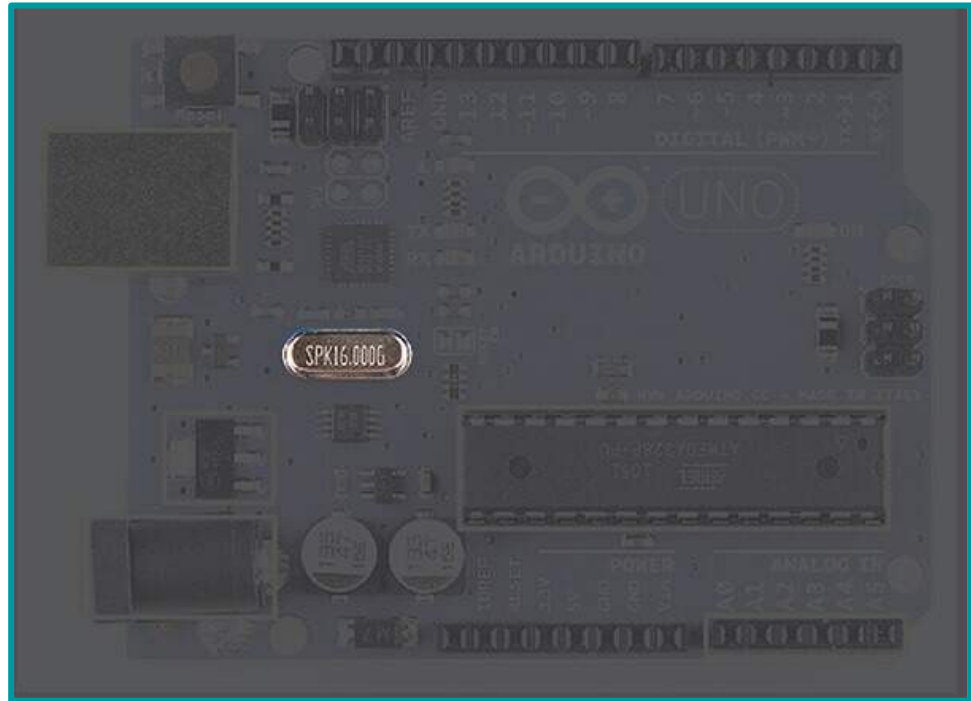




# Arduino UNO

## ■ Crystal oscillator:

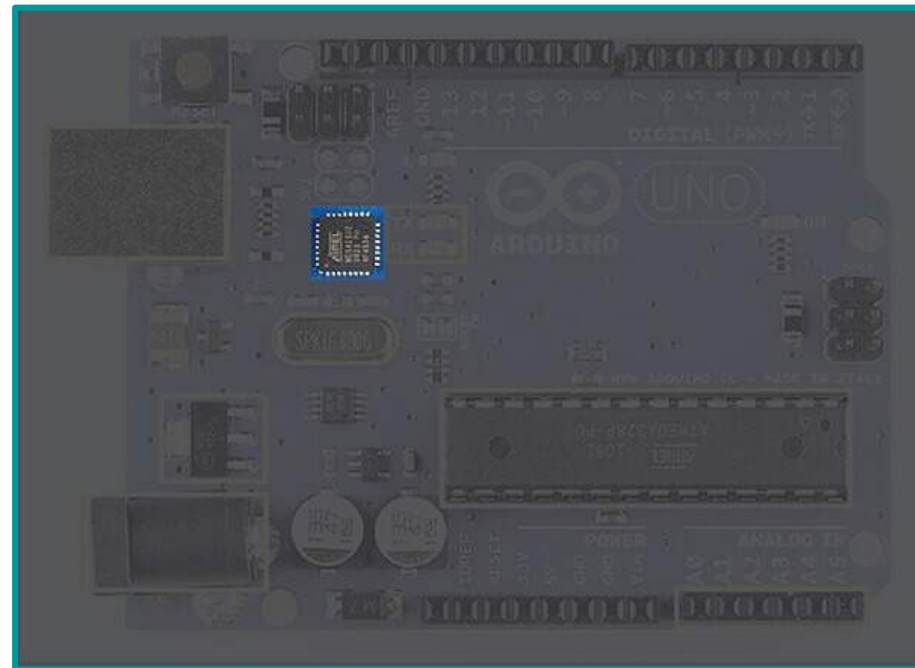
Este é um oscilador de cristal de quartzo que bate 16 milhões de vezes por segundo. Em cada tick, o microcontrolador executa uma operação, por exemplo, adição, subtração, etc.



# Arduino UNO

- Chip de interface USB:

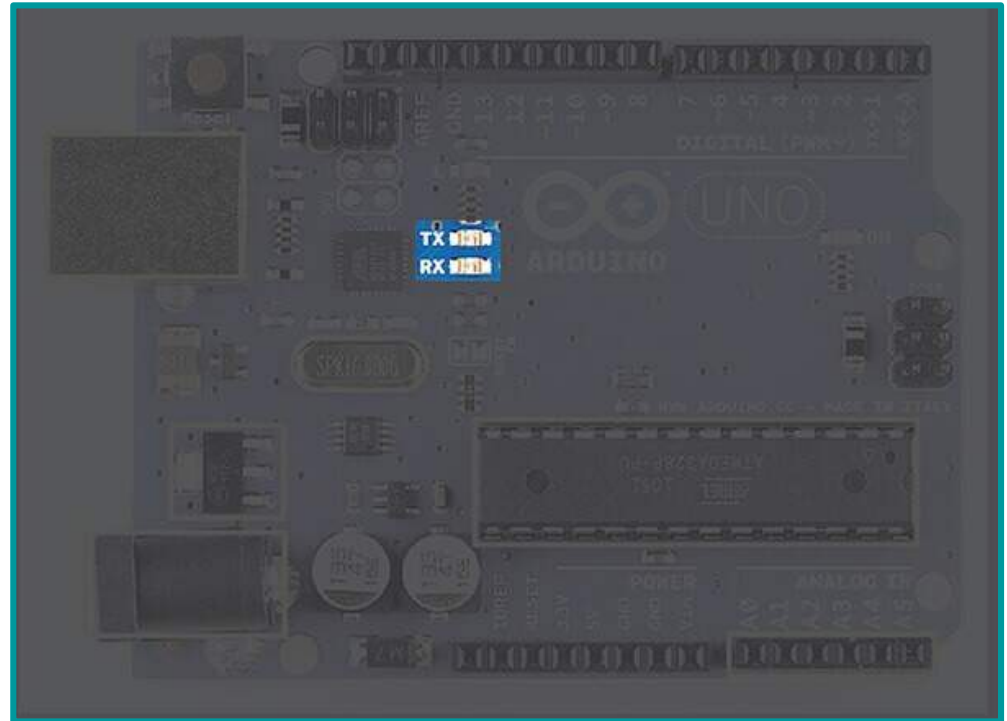
Pense nele como um tradutor de sinais. Ele converte sinais no nível USB para um nível que uma placa Arduino UNO entende.



# Arduino UNO

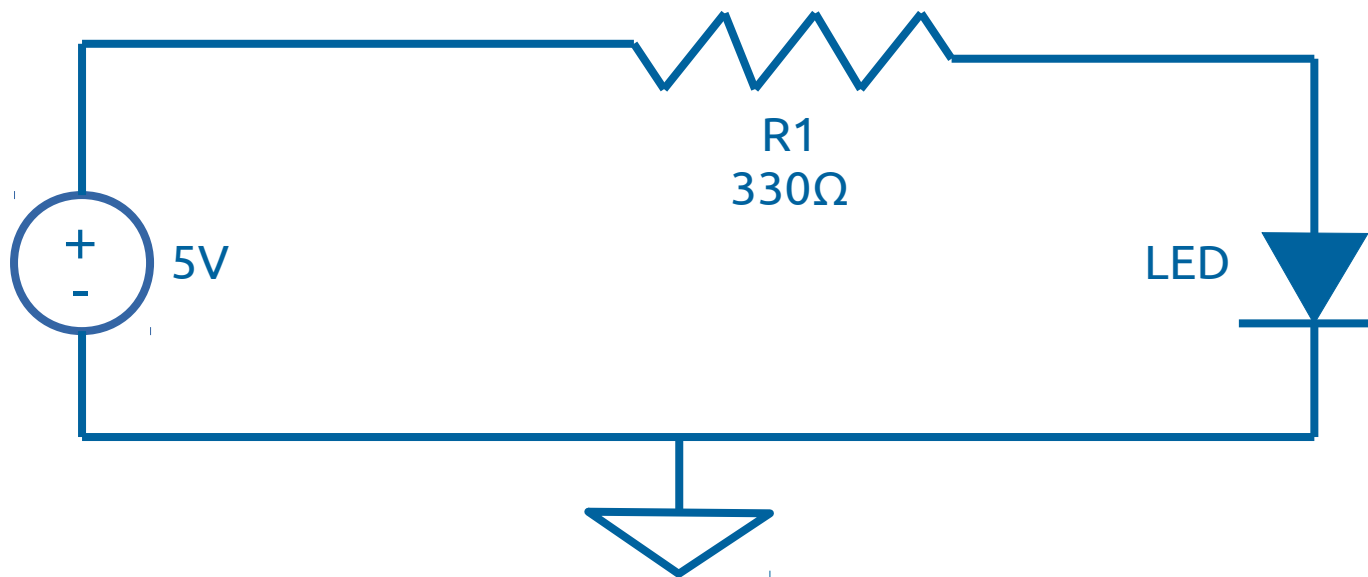
## ■ TX – RX LEDs:

TX significa transmitir e RX significa receber. Estes são LEDs indicadores que piscam sempre que a placa UNO está transmitindo ou recebendo dados.

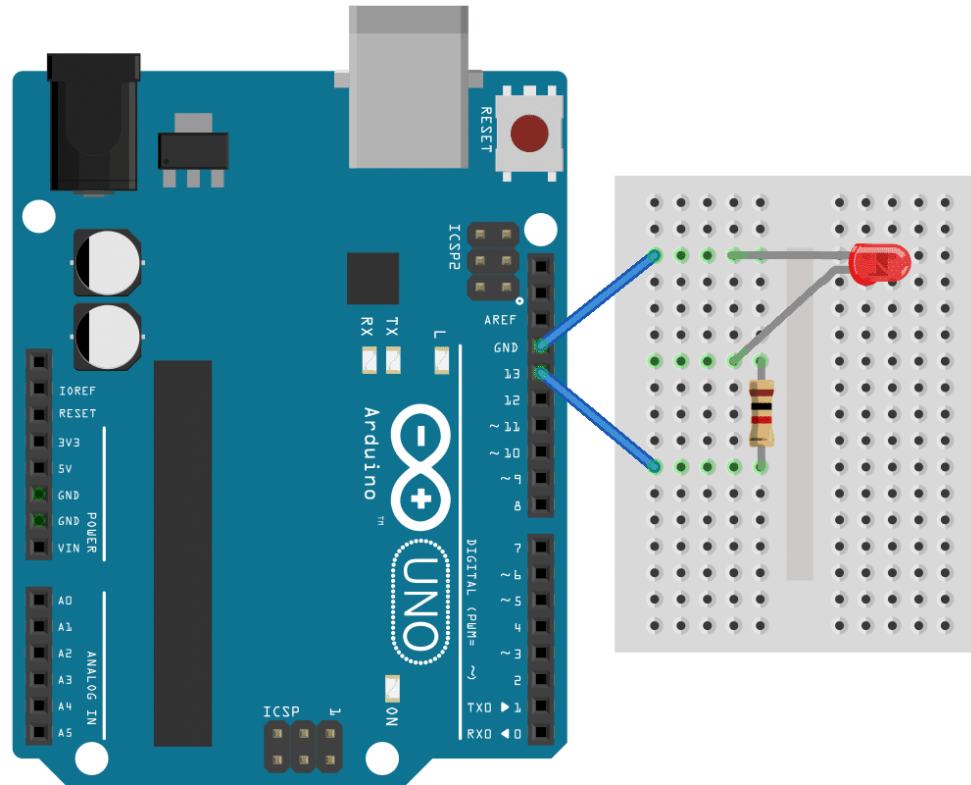


# Blinking LED

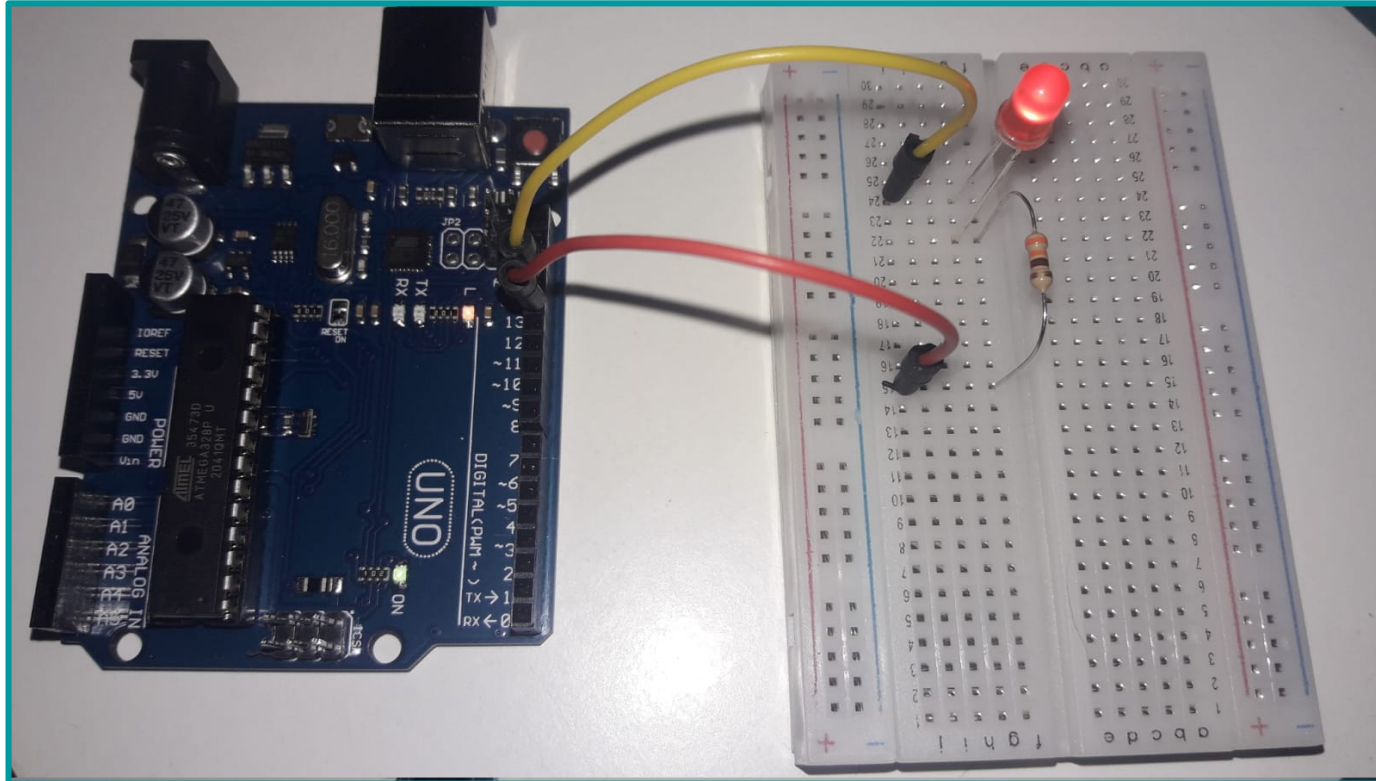
- Agora que sabemos o básico sobre o Arduino, vamos fazer um LED piscar, teremos como base o seguinte circuito:



# Blinking LED



# Blinking LED

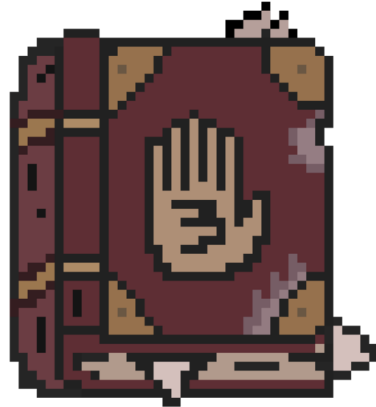


# Blinking LED

## ■ Código-fonte:

```
void setup() {  
  pinMode(13,OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13,HIGH);  
  delay(500);  
  digitalWrite(13,LOW);  
  delay(500);  
}
```

# Documentação



<https://www.arduino.cc/en/main/docs>



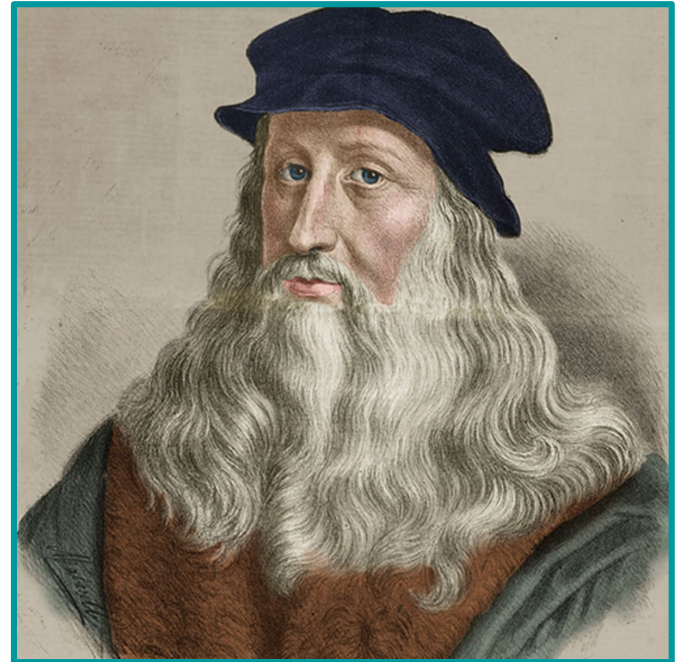
# Bons Estudos!

“O prazer mais nobre é a alegria de compreender.”

— **Leonardo da Vinci**

“Aprender nunca esgota a mente.”

— **Leonardo da Vinci**



# Referências

- <https://en.wikipedia.org/wiki/Arduino>
- <https://www.hackerearth.com/blog/developers/a-tour-of-the-arduino-uno-board/>
- <https://www.youtube.com/playlist?list=PLGs0VKk2DiYw-L-RibttcvK-WBZm8WLEP>