

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

DISRUPTIVE ARCHITECTURES: IOT, IOB E GENERATIVE AI

Prof. Erick Toshio Yamamoto

proferick.yamamoto@fiap.com.br

2025

DISRUPTIVE ARCHITECTURES: IOT, IOB E GENERATIVE AI

- **Conceitos básicos de eletrônica**
- **Introdução aos Microcontroladores**
 - **Arduino UNO**
- **Pinos de Entrada e Saída (INPUT/OUTPUT – I/Os)**
 - **Apresentação do Arduino IDE**
- **Apresentação dos ambientes de simulação**

Material baseado e adaptado dos slides do prof. André Tritiack

ANTES DE COMEÇAR, VAMOS ENTENDER ALGUMAS COISAS:

1. Onde você usa IoT (Internet das Coisas)?
2. Quais exemplos de dispositivos “coisas” você conhece que estão conectados à internet dentro da sua casa?
3. Quais oportunidades de negócio a IoT/IA pode gerar.
Quais problemas ela pode ajudar a resolver?

O que esperar do futuro?

IoT Device Process



Connect IoT device to a network.



Devices collect data and send for centralized storage.



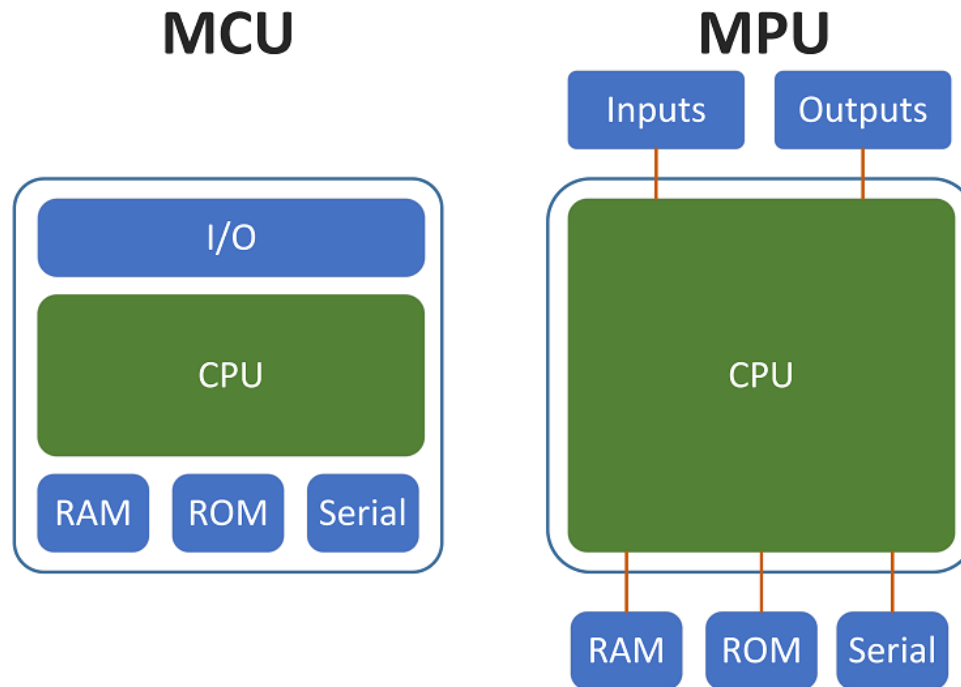
Analyze data locally or send to the cloud for processing.



Make data available.

<https://www.verytechnology.com/iot-whitepapers/what-are-iot-devices-learn-these-basics-before-building-or-deploying>

Microcontrolador x Microprocessador



<https://learn.microsoft.com/pt-br/azure/iot-develop/concepts-iot-device-types>

O que é Arduino?

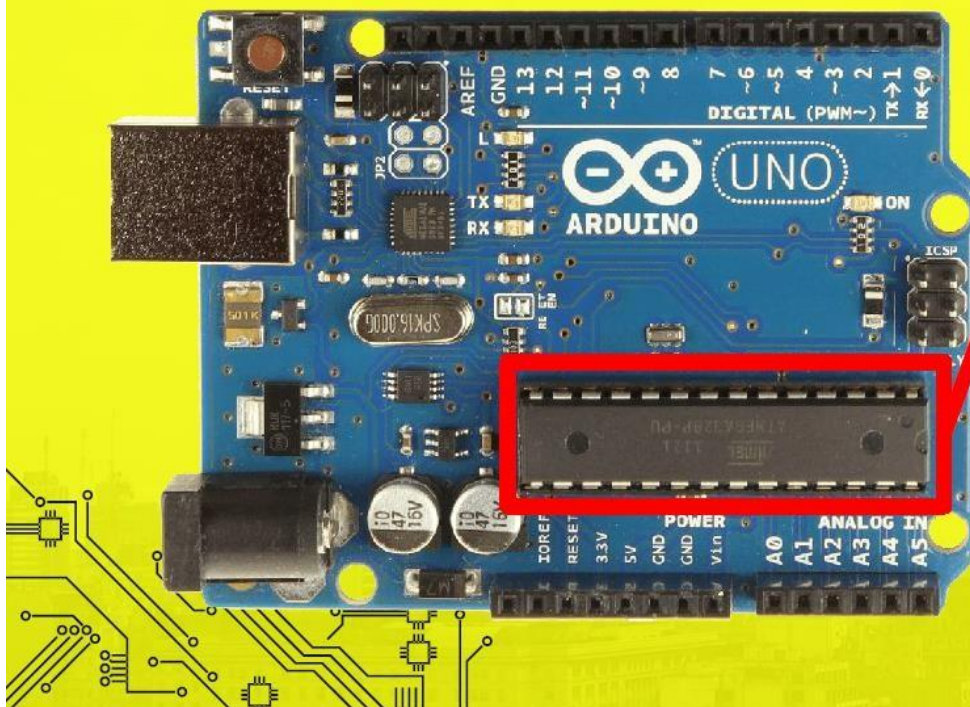
- Plataforma de prototipagem de hardware livre
- Possui microcontrolador AVR ATmega
- Arquitetura Harvard (modificada) de 8 bits (instruções)
- Versão UNO:
 - ◆ ATmega328p | 16 MHz | 1kB EEPROM | 2kB SRAM | 32 kB



Componentes

Microcontrolador

Processa os dados e executa a programação



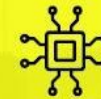
Componentes



Portas analógicas
(0 até 1023)

Aplicações

→ Leitura de sensores analógicos



Componentes



Portas digitais

(0 ou 1)

PWM~ (0 a 255)

Serial (RX e TX)

Aplicações

- Leitura de sensores digitais
- Acionamento de atuadores
- Comunicação



Componentes

Entrada jack

Serve como alimentação externa para a placa.



Componentes

Entrada USB



Utilizada para possibilitar a gravação do firmware no microcontrolador.

Também faz comunicação Serial (transferência de dados em série) e alimentação (5V).

Componentes

Vin



Serve como alimentação externa para a placa.

Caso for conectada alguma fonte de energia (entrada jack ou USB) a Vin passará a fornecer a mesma tensão da fonte. Portanto, NUNCA conecte outra fonte de energia no Vin quando o Arduino estiver ligado.

Componentes

GND

Pontos de aterramento ("terra")
da placa.



Componentes

5V



Fornece a tensão de 5V
(corrente contínua) para
circuitos externos.

Ideal para alimentação de
sensores e atuadores de baixa
tensão.

Componentes

3.3V



Fornece a tensão de 3.3V
(corrente contínua) para
circuitos externos.

Ideal para alimentação de
sensores e atuadores de baixa
tensão.

Componentes

Reset



Serve para reiniciar a placa

The background is a solid yellow color. In the top right corner, there is a dark blue semi-circular shape with a series of small, light blue dashes along its inner edge. To its right, there are black circuit board traces and components. In the bottom left corner, there are more black circuit board traces and components. In the bottom right corner, there is a faint, yellow-tinted silhouette of a city skyline with various skyscrapers.

Comandos Principais

para programação do firmware em Wiring

<https://www.arduino.cc/reference/pt/>

A dark blue, wavy border at the bottom of the slide, resembling a stylized horizon or a series of connected arches.



pinMode (**pin**, **type**)

→ Configura uma porta como **entrada** ou **saída** de corrente elétrica

- ◆ **pin**: número da porta
- ◆ **type**: INPUT (entrada) ou OUTPUT (saída)
- ◆ Exemplo: pinMode(13, OUTPUT);

<https://www.arduino.cc/reference/pt/>



digitalWrite(*pin*, *action*)

→ **Liga** ou **desliga** uma porta digital.

- ◆ **pin**: número da porta que deseja alterar o estado
- ◆ **action**: HIGH ou 1 (ligado), LOW ou 0 (desligado)
- ◆ Exemplo: digitalWrite(13, HIGH);

<https://www.arduino.cc/reference/pt/>





`delay(time)`

→ **Aguarda** alguns milissegundos antes de iniciar a próxima instrução (linha) do código.

- ◆ **time**: tempo em milissegundos
- ◆ Exemplo: `delay(1000);`

<https://www.arduino.cc/reference/pt/>

Copyright ©

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor Erick Toshio Yamamoto.