**AULA 2 e 3 – Raspberry Pi Pico**

**REDES e IoT**

**MICROPYTHON e Componentes**

**O que é?**

E a versão leve do Python 3 que roda em placas como:

* ESP32
* ESP8266
* Raspberry Pi Pico.

**Uso:** Ideal para projetos de automação, sensores, motores, LEDs, IoT e sistemas embarcados.

**Vantagem:** Código enxuto, fácil de aprender, funciona direto no microcontrolador.

**Módulos/Bibliotecas Padrão**

**Micropython**

No MicroPython, **bibliotecas** são conjuntos de funções e classes prontas que facilitam o uso de recursos específicos do microcontrolador, como manipulação de tempo, matemática, controle de hardware (pinos, sensores), entre outros.

Vamos iniciar nessa aula com dois dos mais importantes Módulos/bibliotecas:

* Machine
* utime (ou time)

O Módulo/biblioteca **Machine** é uma das mais importantes no MicroPython. Ela permite acesso direto ao **hardware da placa**, como **pinos**:

* GPIO
* ADC (conversão analógica)
* PWM (modulação por largura de pulso)
* Comunicação:
  + I2C
  + SPI
  + UART

**Módulos/Bibliotecas Padrão**

**Micropython: Machine**

**GPIO (Pin, IN e OUT)**

* **O que é? :** São pinos digitais configuráveis como entrada (IN) ou saída (OUT).
* **Por que usar? :** Permite interagir com componentes eletrônicos, como botões, LEDs e sensores.

Exemplo:

#Importa o comando Pin originando-se do módulo Machine

from **machine** import **Pin**

led = Pin(15, Pin.OUT) #Define o LED e sua respectiva porta

led.value(1) #Liga o LED

**2. ADC (Conversor Analógico para Digital)**

* **O que é?:** Permite ler sinais analógicos (Ex: 0V a 3V) e converter esses valores em dados digitais (0 a 65535)
* **Por que usar?:** Serve para medir grandezas variáveis, como luz, temperatura ou posição, e transformá-las em números compreensíveis pelo código.

**3. PWM (Sinal de modulação por largura de pulso)**

* **O que é?:** Simula uma saída analógica usando um sinal digital que oscila rapidamente.
* **Por que usar?:** Ideal para controlar a intensidade de LEDs, velocidade de motores ou posição de servos.

**4. I2C – Comunicação Serial**

* **O que é?:** Um protocolo de comunicação que usa apenas dois fios para conectar vários dispositivos.
* **Por que usar?:** É ideal quando você precisa conectar vários dispositivos (como sensores e displays) e não quer usar muitos pinos.

**5. SPI – Comunicação SPI**

* **O que é?:** Um protocolo de comunicação que usa mais fios, mas é mais rápido que o I2C.
* **Por que usar?:** Se você precisa de maior velocidade para comunicação, como em cartões SD ou displays mais complexos.
  + Exemplo de uso: Cartões de memória, sensores rápidos e displays gráficos.

**6. UART – Comunicação Serial Padrão**

* **O que é?:** Comunicação serial entre dois dispositivos, como um microcontrolador e um módulo Bluetooth.
* **Por que usar?:** Simples e direto, ótimo para enviar e receber dados entre dois dispositivos, sem complicação.
  + Exemplo de uso: Comunicação com módulos como Bluetooth, GPS ou conexão serial com o computador.

**Módulos/Bibliotecas Padrão**

**Micropython: Utime**

A **biblioteca utime** permite o uso de funções relacionadas ao tempo, como:

1. **Atrasos (sleep):**

* O **que é?:** Usamos para pausar o programa por um tempo determinado.
* **Por que usar?:** Às vezes, você precisa que o microcontrolador faça algo, e depois de um tempo ele faz outra coisa. O atraso ajuda a dar esse "tempo de espera".

**2. Marcação de tempo (marcar o tempo) - ticks\_ms() + ticks\_diff()**

* **O que é?:** Serve para medir o tempo que passou entre duas ações.
* **Por que usar?:** Quando você quer saber quanto tempo levou para fazer algo, como medir o tempo de execução de uma tarefa.

**3. Data e hora local - localtime() e mktime()**

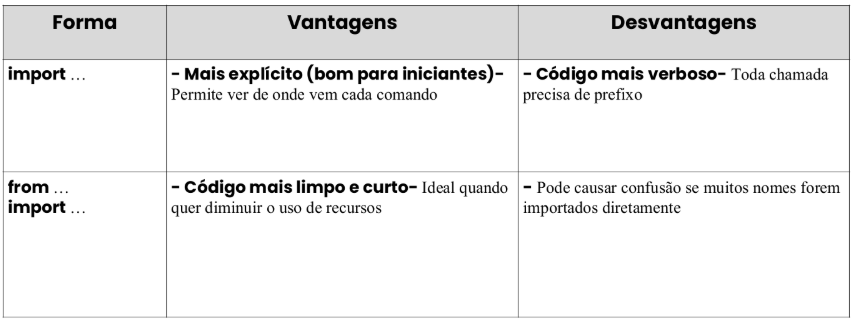
**O que é?:** Serve para trazer: Data (completo - dia, mês e ano) e hora de acordo com a sua localidade. (não é preciso no Wokwi)

**Por que usar?:** Para que, de acordo com essas informações poder configurar equipamentos para executar tarefas em dias específicos. (captar informações, ligar, desligar, etc.)

**COMANDOS e EXEMPLOS DAS BIBLIOTECAS NO PPT:** [**MICROPYTHON**](https://classroom.google.com/c/Njg4Mzg0ODAyNzg1/m/Nzc4Mzc4ODMyNjM3/details)

**Diferença de usar:**

**import e from, import**

****

**Qual é melhor?**

**Para aprendizado ou aulas iniciais:**

* **import … é melhor, pois ajuda o aluno a entender de onde vêm os comandos.**

**Para código mais limpo, curto e prático:**

* **from … import ... é melhor, especialmente em projetos maiores com muitos recursos.**

**COMPONENTES**

**Sensores e Atuadores**

**ATUADORES**

**O que são?**

* São dispositivos que executam uma ação física em resposta a um comando eletrônico.

**Para que servem?**

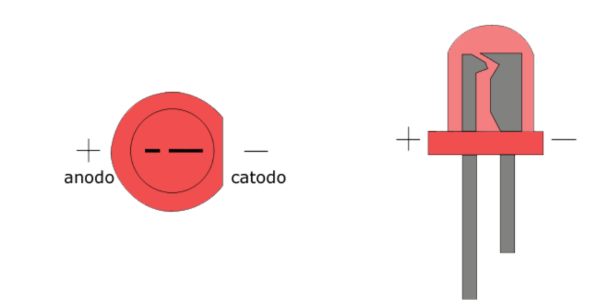
* São utilizados para transformar um sinal elétrico em movimento ou outra forma de energia.

**Exemplos:**

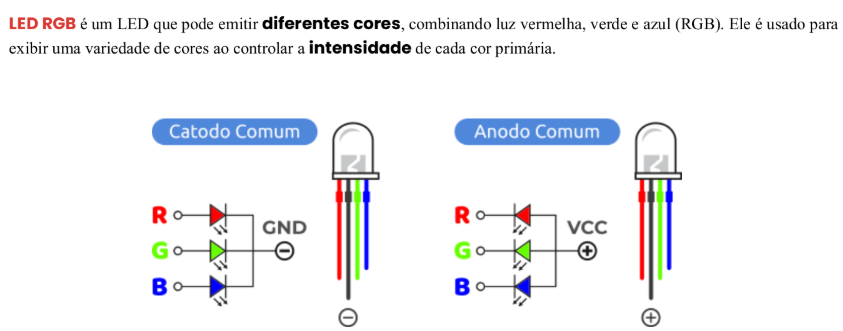
* LEDs, motores, displays, entre outros.

**LED:**

LED (Light Emitting Diode) é um dispositivo que emite luz quando uma corrente elétrica passa por ele. Ele é usado para indicar estados (ligado/desligado) ou até mesmo para fornecer feedback visual.



**LED RGB:**

****

**SENSORES**

**O que são?**

* São dispositivos que captam informações do ambiente físico e as transformam em sinais elétricos que podem ser lidos por um microcontrolador.

**Para que servem?**

* Permitem que o sistema detecte e interprete variáveis do ambiente, como luz, temperatura, umidade, movimento, toque e muito mais.

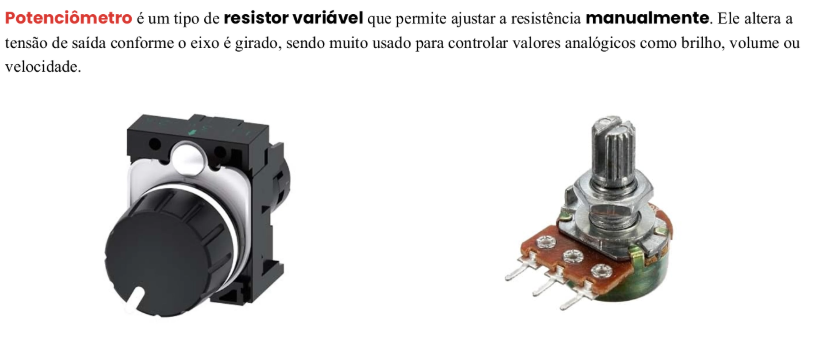
**Exemplos:**

* Potenciômetro, sensor de temperatura, LDR, sensor ultrassônico, botão, entre outros.

**Pushbutton (Botão):**



**Potenciômetro:**



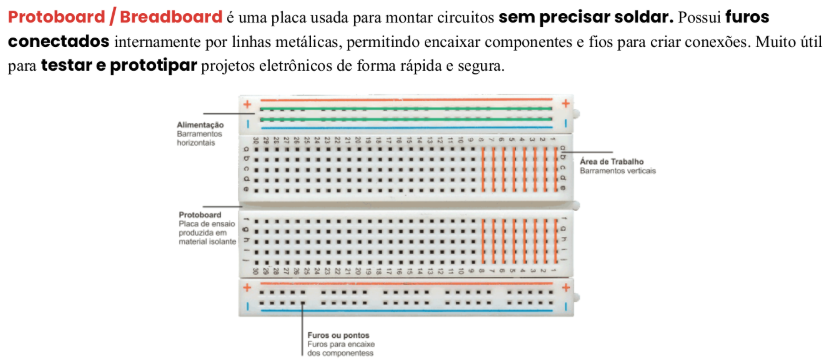
**Outros componentes**

Para o desenvolvimento de um projeto com Microcontroladores, além da placa, sensores e atuadores, precisamos de componentes que dão suporte a prototipação desses projetos.

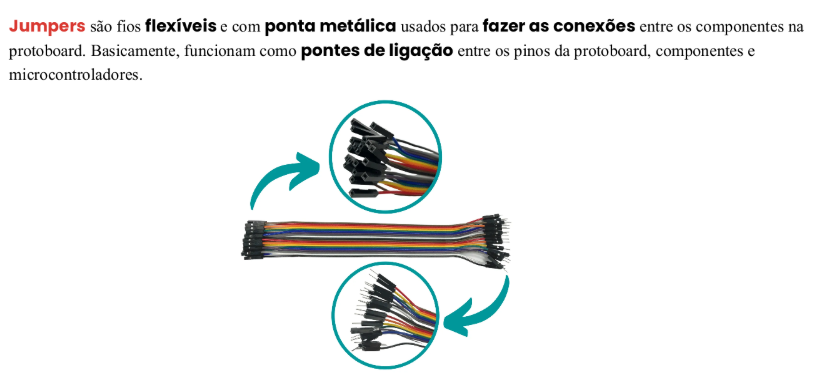
Os principais são:

* Protoboard (ou BreadBoard)
* Resistores
* Jumpers

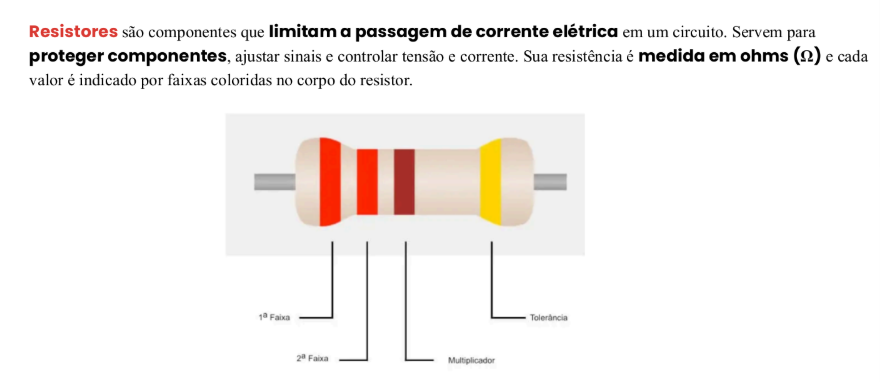
**Protoboard:**

****

**Jumpers**



**Resistores**

****

**Realização de atividades no Raspberry Pi Pico – LED RGB e Pushbutton**