



# neeeil it >



Workshop introdutório de Raspberry Pico W

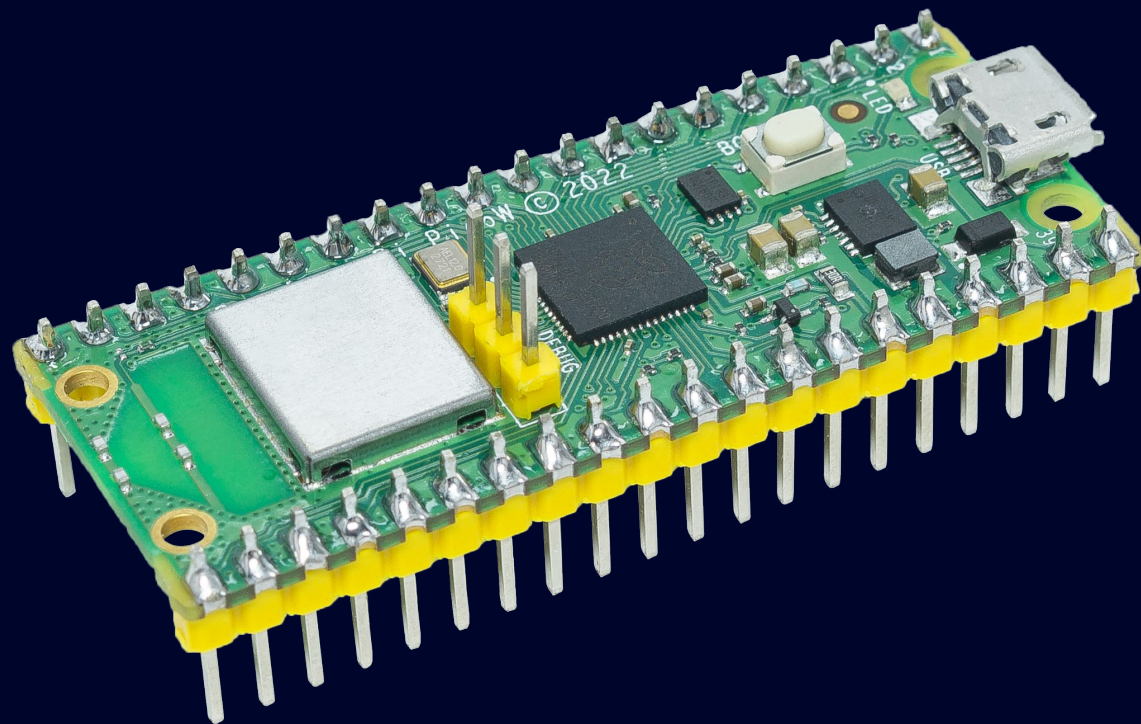


#NailTheFuture

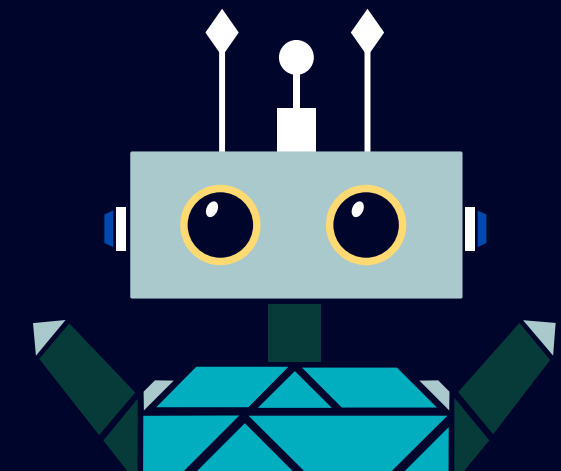


# Introdução

## Raspberry Pico W



- Microcontrolador de baixo custo.
- Microprocessador RP2040 integrado.
- Dois cores de processamento.
- Comunicação sem fios integrada.
- 26 pinos de uso geral.

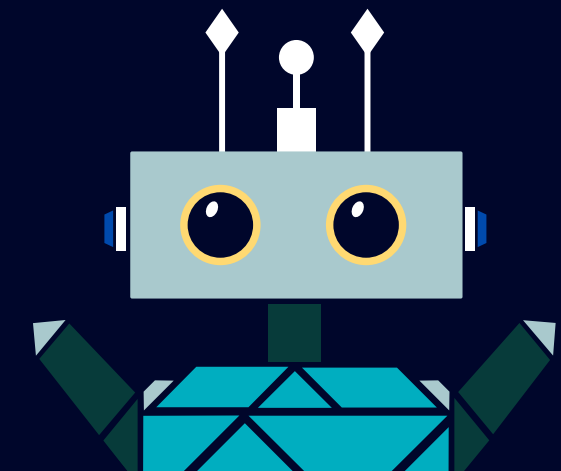


# Introdução

## Arduino IDE 2

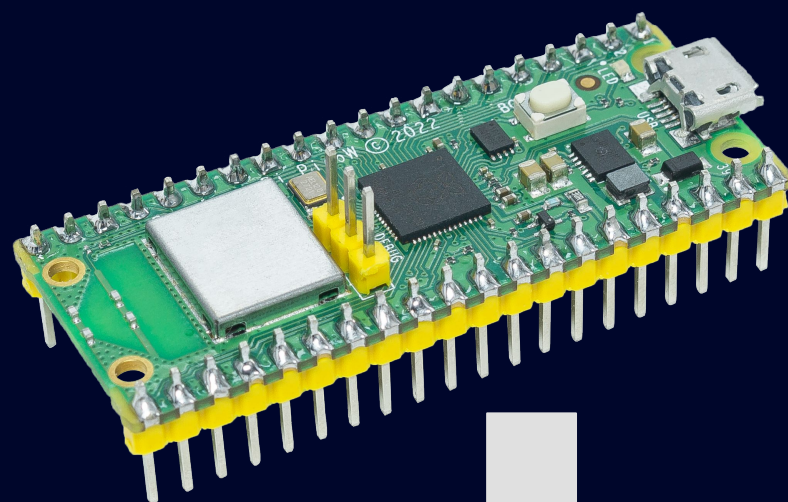


- Plataforma de desenvolvimento de programação de microcontroladores.
- Processo simples de escrever e carregar código para uma variedade de placas.



# Introdução

Pico W + Arduino IDE 2

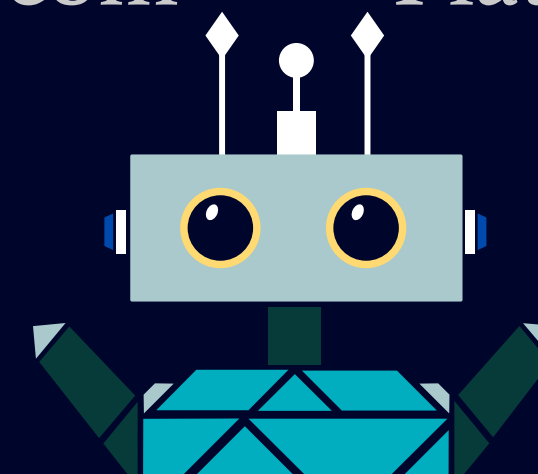


Vantagens:

- Familiaridade;
- Grande Comunidade e Recursos;
- Facilidade de Uso;
- Suporte a Bibliotecas;

Integração  
Existentes;

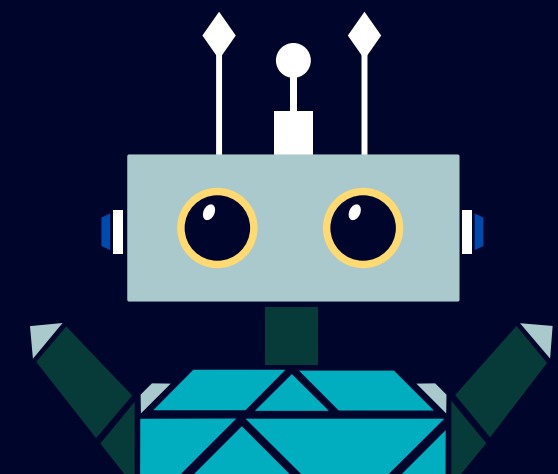
com Plataformas



# Inicialização do Arduino IDE 2



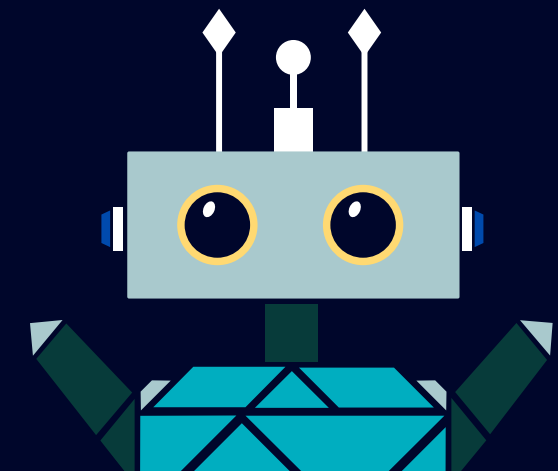
1. Instalar o Arduino IDE 2: <https://www.arduino.cc/en/software>



# Inicialização do Arduino IDE 2



1. Instalar o Arduino IDE 2: <https://www.arduino.cc/en/software>
2. Adicione o Suporte para a Raspberry Pi Pico:

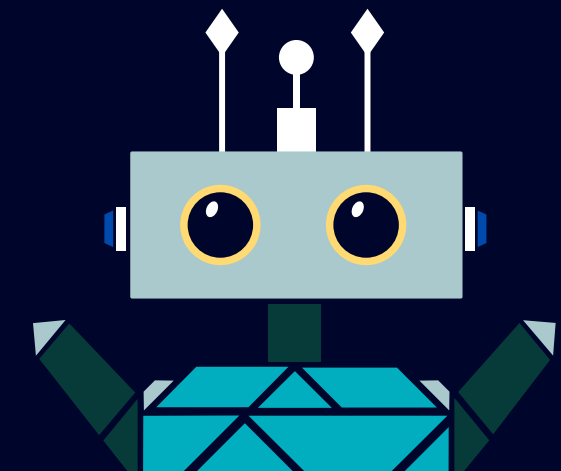




# Inicialização do Arduino IDE 2



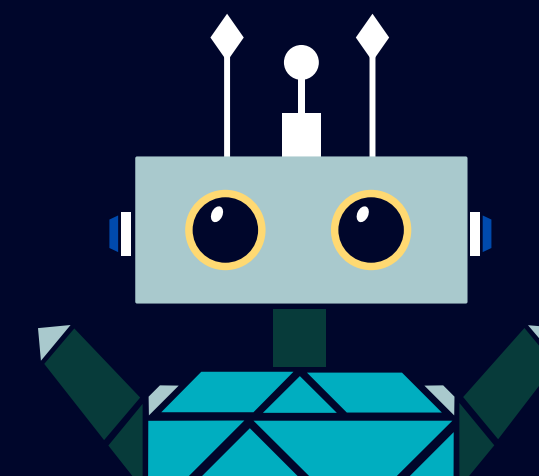
1. Instalar o Arduino IDE 2: <https://www.arduino.cc/en/software>
2. Adicione o Suporte para a Raspberry Pi Pico:
  - a. Abrir Arduino IDE 2: Clique em “File” -> “Preferences...”



# Inicialização do Arduino IDE 2



1. Instalar o Arduino IDE 2: <https://www.arduino.cc/en/software>
2. Adicione o Suporte para a Raspberry Pi Pico:
  - a. Abrir Arduino IDE 2: Clique em “File” -> “Preferences...”
  - b. Colar o link no campo “*URLs Adicionais de Gerenciamento de Placas*”:  
[https://github.com/earlephilhower/arduino-pico/releases/download/global/package\\_rp2040\\_index.json](https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json)

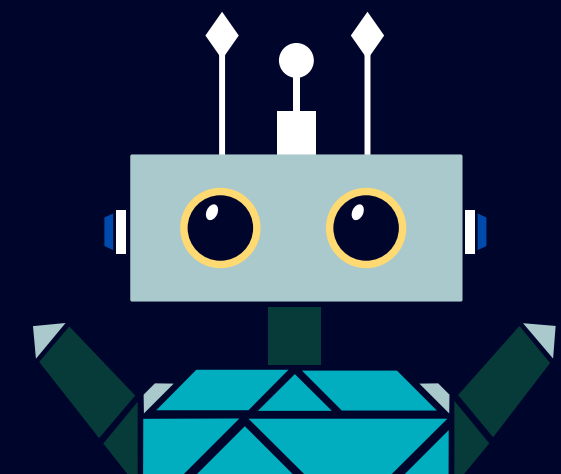




# Inicialização do Arduino IDE 2



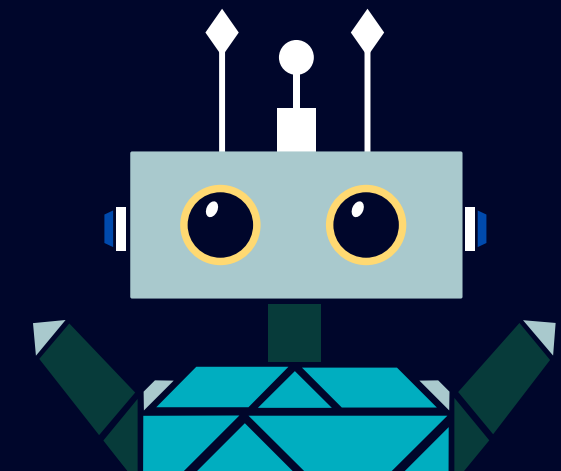
1. Instalar o Arduino IDE 2: <https://www.arduino.cc/en/software>
2. Adicione o Suporte para a Raspberry Pi Pico:
  - a. Abrir Arduino IDE 2: Clique em “File” -> “Preferences...”
  - b. Colar o link no campo “*URLs Adicionais de Gerenciamento de Placas*”:  
[https://github.com/earlephilhower/arduino-pico/releases/download/global/package\\_rp2040\\_index.json](https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json)
  - c. Clique em "OK" para fechar a janela de preferências.



# Inicialização do Arduino IDE 2



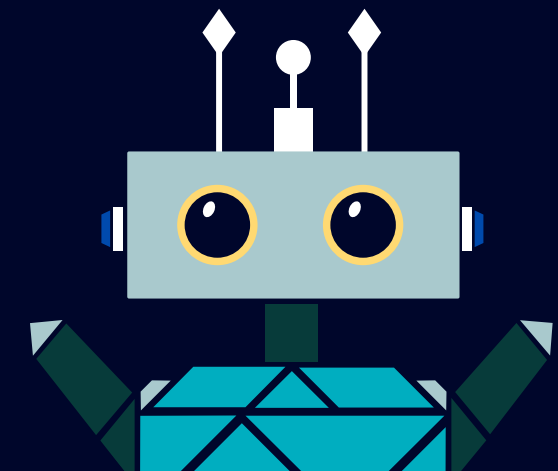
1. Instalar o Arduino IDE 2: <https://www.arduino.cc/en/software>
2. Adicionar o Suporte para a Raspberry Pi Pico.
3. Instalar a Placa Raspberry Pi Pico:



# Inicialização do Arduino IDE 2



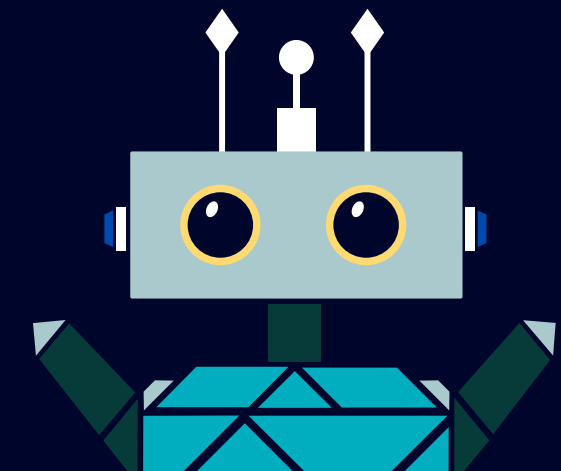
1. Instalar o Arduino IDE 2: <https://www.arduino.cc/en/software>
2. Adicionar o Suporte para a Raspberry Pi Pico
3. Instalar a Placa Raspberry Pi Pico:
  - a. Clique em "Ferramentas" -> "Placas" -> "Gerenciador de Placas..."



# Inicialização do Arduino IDE 2



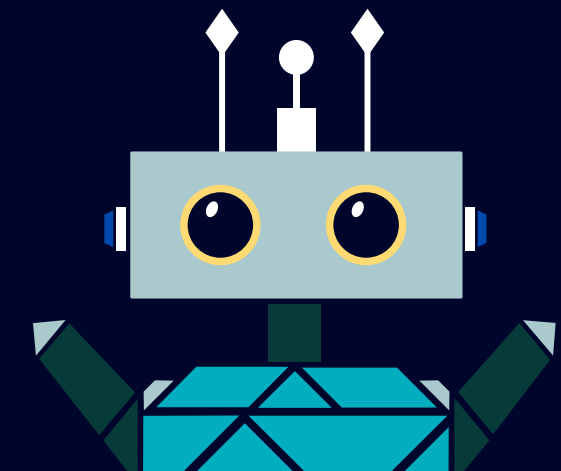
1. Instalar o Arduino IDE 2: <https://www.arduino.cc/en/software>
2. Adicionar o Suporte para a Raspberry Pi Pico.
3. Instalar a Placa Raspberry Pi Pico:
  - a. Clique em "Ferramentas" -> "Placas" -> "Gerenciador de Placas...".
  - b. Pesquise por "Pico" e instale a opção "Raspberry Pi Pico/RP2040".



# Inicialização do Arduino IDE 2



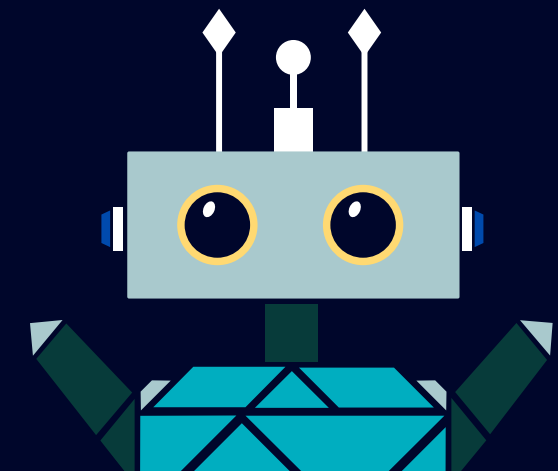
1. Instalar o Arduino IDE 2: <https://www.arduino.cc/en/software>
2. Adicionar o Suporte para a Raspberry Pi Pico.
3. Instalar a Placa Raspberry Pi Pico.
4. Selecionar a Placa e a Porta:



# Inicialização do Arduino IDE 2



1. Instalar o Arduino IDE 2: <https://www.arduino.cc/en/software>
2. Adicionar o Suporte para a Raspberry Pi Pico
3. Instalar a Placa Raspberry Pi Pico
4. Selecionar a Placa e a Porta:
  - a. Volte para o menu "Ferramentas" e selecione a placa "Raspberry Pi Pico W" em "Placa" -> "Raspberry Pi Pico/RP2040".

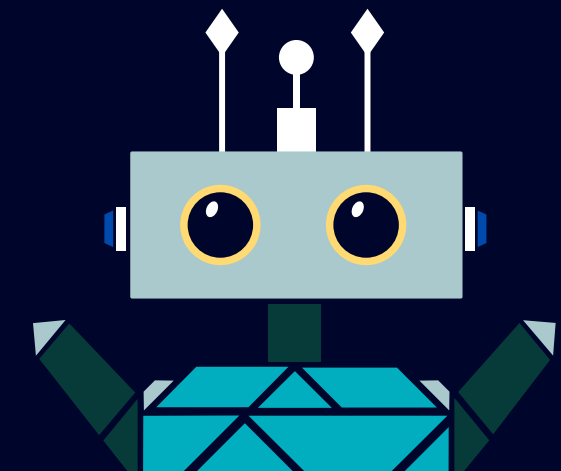




# Inicialização do Arduino IDE 2



1. Instalar o Arduino IDE 2: <https://www.arduino.cc/en/software>
2. Adicionar o Suporte para a Raspberry Pi Pico.
3. Instalar a Placa Raspberry Pi Pico.
4. Selecionar a Placa e a Porta.
5. Testar conexão:



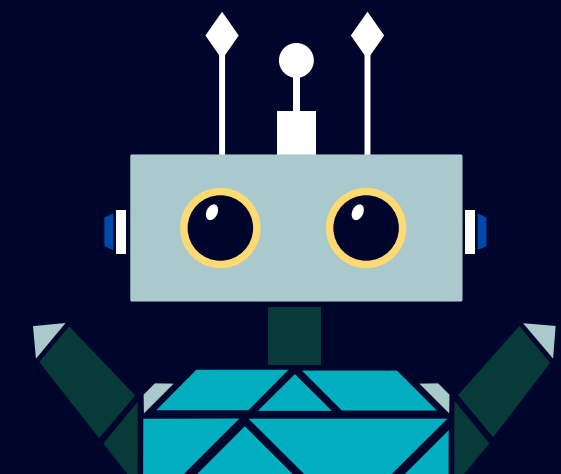
# Inicialização do Arduino IDE 2



4. Selecionar a Placa e a Porta.

5. Testar conexão:

- a. Carrega um exemplo de teste simples, como o "Blink", clicando em "Arquivo" -> "Exemplos" -> "01.Basics" -> "Blink".
- b. Carrega o código clicando no botão "Carregar" (ícone de seta para a direita).

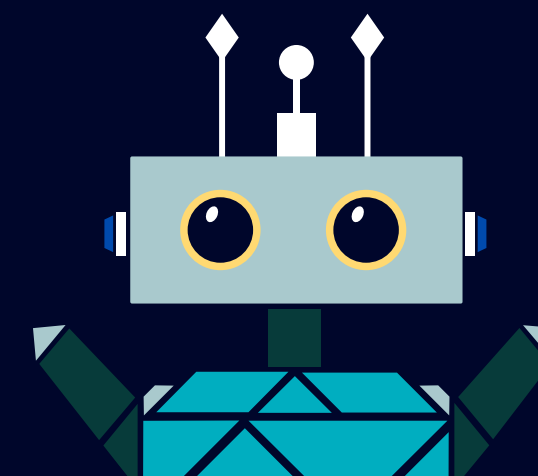


# Inicialização do Arduino IDE 2



Dicas:

- Caso não estejas a conseguir estabelecer ligação com a Pico, desconecta a placa do computador e, em seguida, conecta ao computador mantendo premido o botão “BOOTSEL” presente na pico.
- Caso exista múltiplas portas “COMx”, é provável que a pico seja a que tenha o número maior.



# Utilização do Arduino IDE e Raspberry Pi Pico W



- Definição de entradas e saídas:

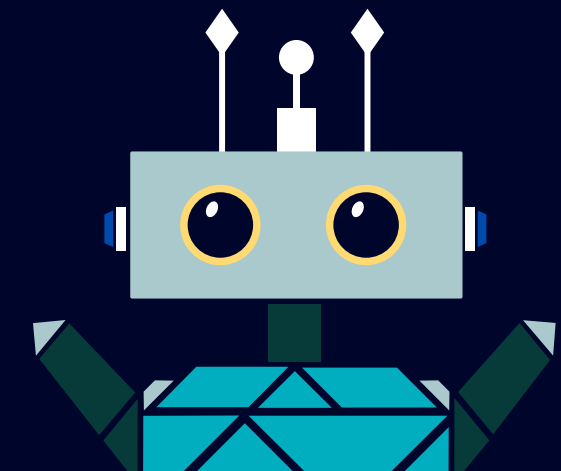
- `pinMode`:

Define se o pino vai ser usado como entrada ou como saída.

- Exemplo:

```
pinMode(pushButton, INPUT);
```

```
pinMode(LED_BUILTIN, OUTPUT);
```



# Utilização do Arduino IDE e Raspberry Pi Pico W



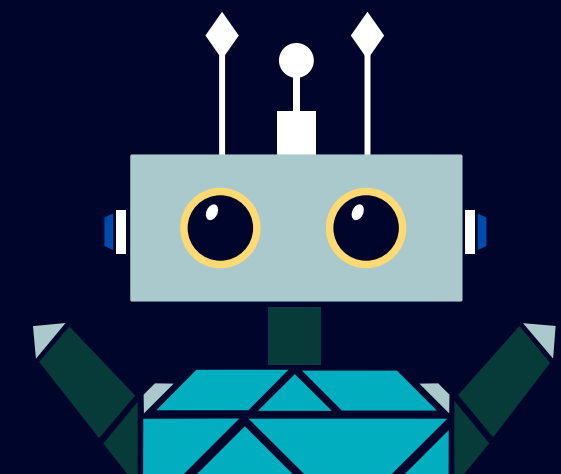
- Leitura de entradas:

- `digitalRead`:

Leitura de uma entrada digital. Retorna “HIGH” ou “LOW”.

- `analogRead`:

Leitura de uma entrada analógica. Retorna um número inteiro entre 0 e 4095.



# Utilização do Arduino IDE e Raspberry Pi Pico W



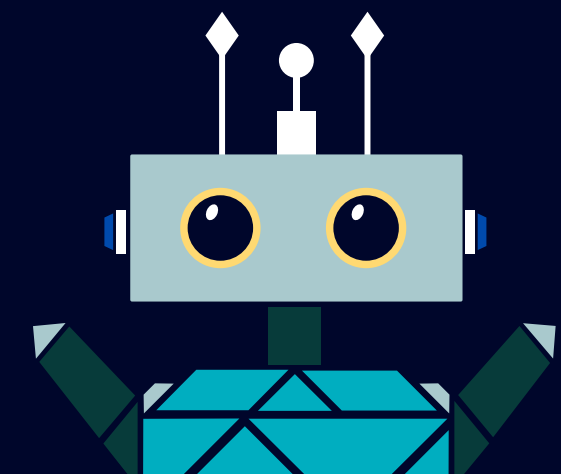
- Controlo de saídas

- `digitalWrite`:

Escrita de um valor digital numa saída.

- Exemplo:

```
digitalWrite(LED_BUILTIN, HIGH);
```





# Utilização do Arduino IDE e Raspberry Pi Pico W



- Controlo de saídas em PWM

- `analogWrite`:

Geração de uma onda quadrada com um duty cycle definido pelo utilizador. O duty cycle pode variar entre 0 (sempre desligado) a 255 (sempre ligado)

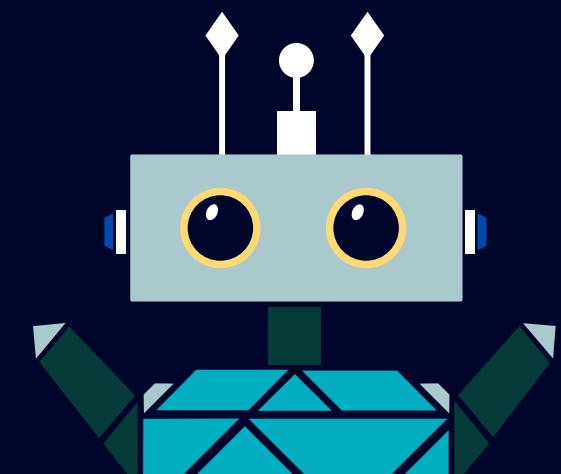
- Exemplo:

```
analogWrite(LED_BUILTIN, 128);
```

- `analogWriteFreq`:

Define a frequência do PWM usado.

Pode ir de 100Hz a 1MHz.



# Utilização do Arduino IDE e Raspberry Pi Pico W



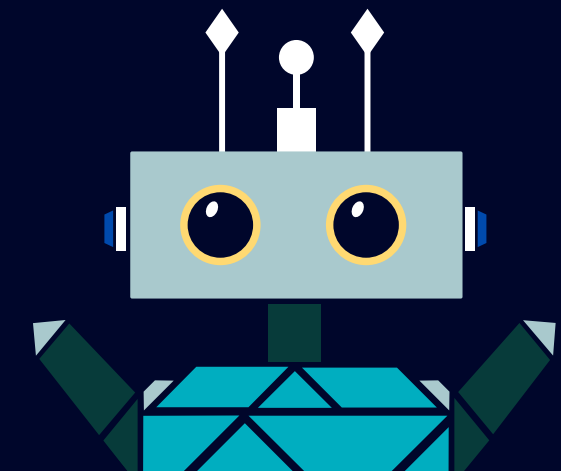
Funcionamento da Pico:

- **setup:**

Função que corre uma vez quando o programa inicializa.

- **loop:**

Função que corre está sempre a correr.

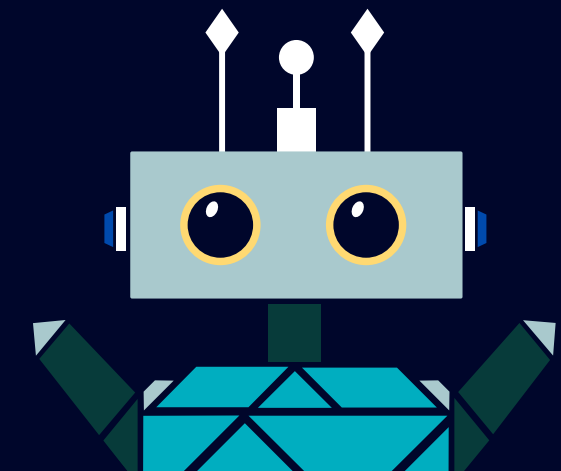


# Utilização do Arduino IDE e Raspberry Pi Pico W



Exemplo:

```
void setup() {  
  
    pinMode(LED_BUILTIN, OUTPUT);  
    pinMode(pushButton, INPUT);  
}  
  
void loop() {  
    if (digitalRead(pushButton))  
        digitalWrite(LED_BUILTIN, HIGH);  
    else  
        digitalWrite(LED_BUILTIN, LOW);  
    delay(100);  
}
```

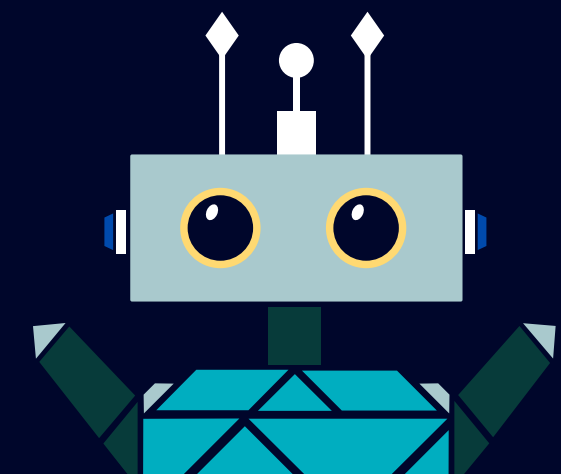


# Utilização do Arduino IDE e Raspberry Pi Pico W



Exemplo:

```
void setup() {  
    // initialize digital pin LED_BUILTIN as an output and the pushButton as input.  
    pinMode(LED_BUILTIN, OUTPUT);  
    pinMode(pushButton, INPUT);  
}  
// the loop function runs over and over again forever  
void loop() {  
    if (digitalRead(pushButton)          // read the state of the button  
        digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
    else  
        digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW  
    delay(100);                          // wait for a tenth of a second  
}
```



# Utilização do Arduino IDE e Raspberry Pi Pico W



Funcionamento das Interrupções:

- **attachInterrupt:**

Função que corre uma vez quando um evento acontece

- Exemplo:

```
attachInterrupt(digitalPinToInterrupt(pushButton), pressedISR, RISING)
```

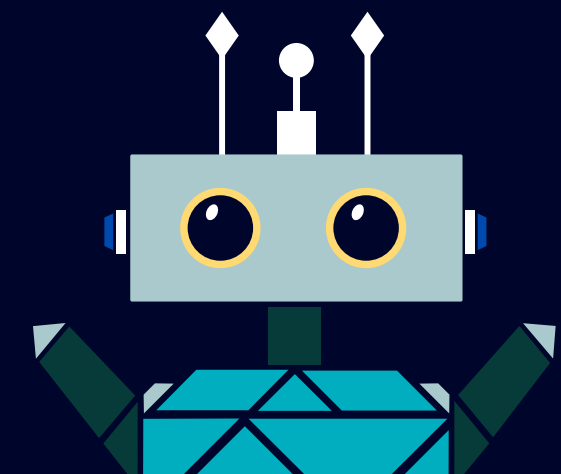
MODES:

**LOW** : pin is low,

**CHANGE**: pin changes value

**RISING**: pin goes from low to high,

**FALLING**: pin goes from high to low.



# Desafio



Cria um programa que controle a luminosidade do led presente na pico usando o botão “BOOTSEL” de maneira que cada clique (premir + largar) do botão altere em 25% a luminosidade do led. Caso a luminosidade seja 100% quando o botão é premido, o led desliga.

