

## Ajustando o nosso código

### Transcrição

Os ajustes que devemos fazer no nosso código são bem simples. Vamos começar pelo **controle.py**. Não vamos mais executar as funções `setup_motor()` e `le_tecla()` dentro do arquivo, pois vamos chamá-las dentro do próprio Flask. Logo podemos comentar as chamadas das funções:

```
# controle.py

# funções omitidas

# Comentar as duas linhas abaixo!
#setup_motor()
#le_tecla()
```

Salvamos o arquivo e agora modificaremos o **distancia.py**. Aqui também não executaremos as funções dentro do próprio arquivo, ou seja, temos que comentar (ou remover) as chamadas das funções `setup_sensor()` e `roda_medicao()`:

```
# distancia.py

# funções omitidas

# Comentar as duas linhas abaixo!
#setup_sensor()
#roda_medicao()
```

A distância também não será mais impressa na tela, logo vamos remover essa impressão também, dentro da função `roda_medicao()`:

```
# distancia.py

# código anterior omitido

def roda_medicao():
    global distancia_cm
    distancia_cm = 0
    while True:
        time.sleep(2)
        GPIO.output(TRIG, GPIO.HIGH)
        time.sleep(0.000010)
        GPIO.output(TRIG, GPIO.LOW)
        while GPIO.input(ECHO) == 0:
            pulso_inicial = time.time()
        while GPIO.input(ECHO) == 1:
            pulso_final = time.time()
        trigger = pulso_final - pulso_inicial
        distancia_cm = 34300 * (trigger/2)
        distancia_cm = round(distancia_cm,0)
```

```
# Comentando a impressão da distância
#print (distancia_cm, 'cm ',end="\r")
```

```
#setup_sensor()
#roda_medicao()
```

Mas como teremos acesso à distância? Criaremos uma função que a retorne para nós, a `get_distancia()` :

```
def get_distancia():
    return distancia_cm
```

Salvamos o arquivo e assim concluímos as alterações para os arquivos rodarem com o Flask! No próximo vídeo faremos os ajustes finais e rodaremos o nosso projeto!