

## Mão na massa: Ajustando o código

### Ajustando controle.py

Dentro do diretório **pibot**, abra o arquivo **controle.py**. Nele, remova ou comente as chamadas das funções `setup_motor()` e `le_tecla()`.

### Ajustando distancia.py

Agora abra o arquivo **distancia.py** e remova (ou comente) também as chamadas das funções `setup_sensor()` e `roda_medicao()`. Além disso, adicione uma nova função, chamada `get_distancia()`:

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

Como a distância não será mais impressa no console, remova a linha que a imprime, dentro da função `roda_medicao`:

```
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")
```

### Ajustando mjpg-streamer

O **pibot.sh** está configurado para executar o **mjpg-streamer** dentro da pasta **mjpg-streamer**, mas o seu código fica em **mjpg-streamer/mjpg-streamer-experimental**, então vamos movê-lo:

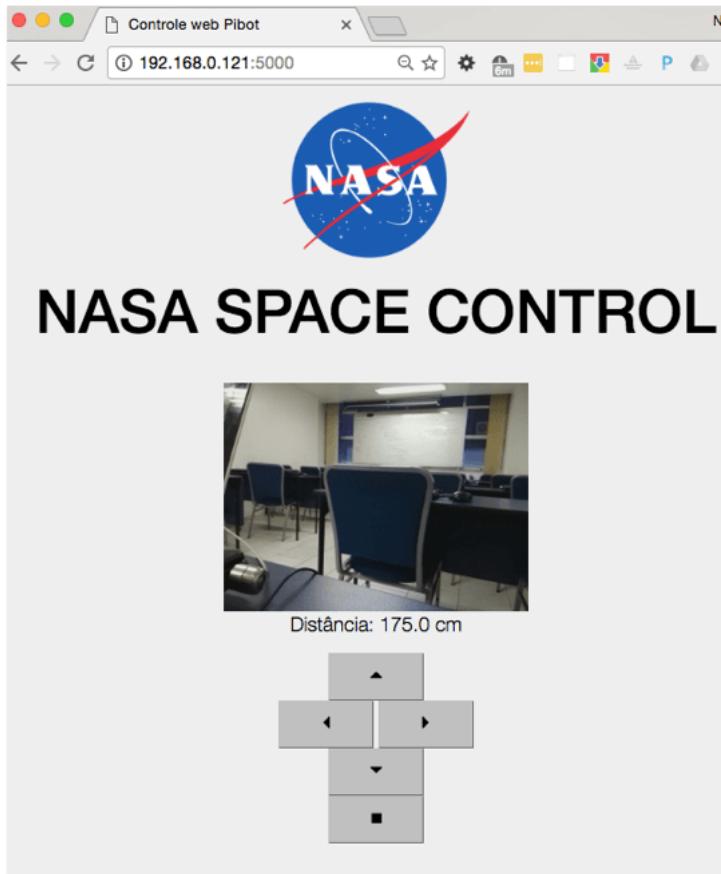
```
cd ~/mjpg-streamer/mjpg-streamer-experimental/
mv * ../
```

### Rodando o projeto

Agora, execute o script **pibot.sh**, ele executará o **mjpg-streamer** e o **pibot-web.py**:

```
cd ~/pibot  
./pibot.sh
```

Agora abra a página na seguinte URL <http://IP-DO-RASPBERRY-PI:5000> (<http://IP-DO-RASPBERRY-PI:5000>). Será exibida uma página semelhante a essa:



Uma página plenamente funcional, com acesso ao *streaming* da câmera, aos controles do carrinho e à distância do objeto à frente do mesmo. Tudo isso concentrado em apenas um lugar.