

```

'PROGRAMA DE ADQUISICION Y TRANSMISION'
""

"Definicion de variables y librerias"
import serial #Serial para GPS y comunicacion
import pynmea2 #pynmea2 para GPS

#Importar Librerias
import pigpio
import DHT22
import Adafruit_BMP.BMP085 as BMP085
import time
import smbus
from time import sleep

#Inicio GPIO
pi= pigpio.pi()

#Configuracion de sensores
dht22 = DHT22.sensor(pi,4)
dht22.trigger()
bmp = BMP085.BMP085()

#Tiempo de muestreo
sleepTime = 20

"Definicion de Funciones"
def readDHT22(): #Medicion con DHT22
    dht22.trigger()
    humidity = dht22.humidity()
    temp = dht22.temperature()
    return (humidity, temp)
def readBMP180(): #Medicion con BMP180
    temper = bmp.read_temperature()
    pressure = bmp.read_pressure()
    return (temper, pressure)

def sensores(): #Devuelve datos cada minuto
    i=0
    promhum=0
    promtem=0
    for i in range (3):
        sleep(sleepTime-0.3333)
        humidity, temp = readDHT22()
        temper, pressure = readBMP180()
        tmed=(temp+temper)/2
        promhum=promhum+humidity
        promtem=promtem+tmed
        i=i+1
    meteoros=['%.2f C'%(promtem/3), '%.2f %'%(promhum/3),
              '%.2f hPa'%(pressure/100)]
    return meteoros

```

```

def enviar(): #Enviar informacion inalambricamente
    #Abrir puerto serial ttyUSB0
    serialport = serial.Serial('/dev/ttyUSB0',115200, timeout=0.5)
    #Escribir en el puerto
    serialport.write(str(total))
    #Leer el puerto serial
    x=serialport.read(101)
    print(str(x)) #Desplegar informacion
    serialport.close()#cerrar puerto

```

"INICIO DEL PROGRAMA PRINCIPAL"

#Proceso Iterativo

```

while True:
    serialPort=serial.Serial('/dev/ttyAMA0',9600)
    datoGPS=serialPort.readline()
    if datoGPS.find('GGA')>0:
        #Vector Meteorologico
        meteo2=sensores()
        #Funcion para obtener Posicion Geografica
        msg=pynmea2.parse(datoGPS)
        #Vector de Posicion
        posicion=[time.strftime('%x'), '%s'%msg.timestamp, '%04.7f'
                    %(msg.longitude), '%03.7f'%(msg.latitude), '%s m'
                    % msg.altitude]
        #Vector de Informacion
        total=posicion+meteo2
        serialPort.close()
        enviar()

```

#VOLVER AL CICLO WHILE

#FIN DE LA TRANSMISION