

Extracto de la memoria del proyecto CanSat

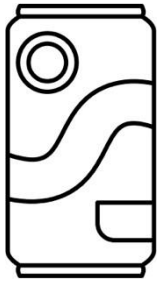


INDICE DE CONTENIDOS

>>Introducción al proyecto
- Descripción del proyecto
- Requisitos y condiciones
- Misión secundaria
- Diagrama de Gantt
>> Lista de materiales
>> Diseño de la carcasa
- Explicación del diseño
- Medidas
- Diseño final
>>Circuito
- Conexiones
- Esquema eléctrico
- Foto del circuito en la vida real
>>Programas
- BMP 280
- DHT11
- Lector SD
- Programa con todo junto
>>Construcción del paracaídas
- Cálculos y medidas

INTRODUCCIÓN AL DESAFÍO

DESCRIPCIÓN DEL PROYECTO



CanSat es una iniciativa de la Agencia Espacial Europea que desafía a estudiantes de toda Europa a construir y lanzar un mini satélite del tamaño de una lata de refresco. El desafío para los estudiantes es adaptar todos los subsistemas principales que se encuentran en un satélite, como la energía, los sensores y un sistema de comunicación, en el volumen y la forma de una lata de refrescos. Luego, el CanSat es lanzado por un cohete hasta una altitud de aproximadamente un kilómetro, o se deja caer desde una plataforma, un dron o un globo cautivo. Entonces comienza su misión. Durante la caída se realiza un experimento científico y / o una demostración tecnológica, lograr un aterrizaje seguro y analizar los datos recopilados.

REQUISITOS Y CONDICIONES

- 1- Una de las condiciones del proyecto es que se debe limitar el satélite a las dimensiones de una lata :115mm de largo y 65mm de diámetro.
- 2- Otro punto a cumplir es que el proyecto presente una serie de subsistemas que nos permitan recopilar datos sobre la temperatura y la presión.
- 3- La última condición es diseñar un sistema de aterrizaje efectivo para el satélite.

NUESTRA MISIÓN SECUNDARIA

A parte de cumplir los requisitos anteriores, también es necesario que el satélite cumpla con una misión secundaria particular. Nosotros hemos decidido que la nuestra sea medir, a parte de la temperatura y la presión, la humedad del ambiente.



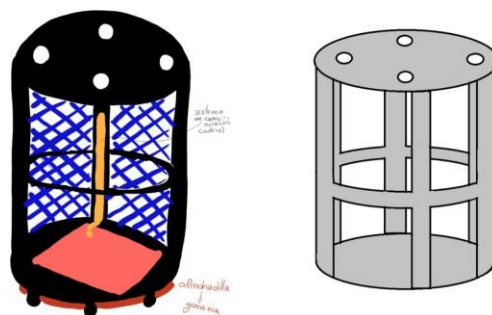
LISTA DE MATERIALES

COMPONENTE	PRECIO	ENLACE
Arduino NANO	16,195€	https://www.e-ika.com/arduino-nano-con-chip-ft32-y-cable-usb
Sistema de comunicación (HC12)	24,14€	https://tienda.bricogeek.com/lora/1122-ttgo-lora32-esp32-con-oled-900-mhz.html?search_query=lora+arduino&results=14
Sensor de presión y temperatura (BMP280)		
Sensor de humedad (DHT11)		
Lector de Tarjeta SD	1,56€	https://www.e-ika.com/modulo-sd-de-lectura-escritura
Placa de pruebas	2,15€	https://www.e-ika.com/placa-de-pruebas-de-5x7-cm-5uds
cables	2,62€	https://www.e-ika.com/cables-24awg-de-20cm-100uds
Materiales del paracaídas: bolsa de basura, cinta americana, cuerda		

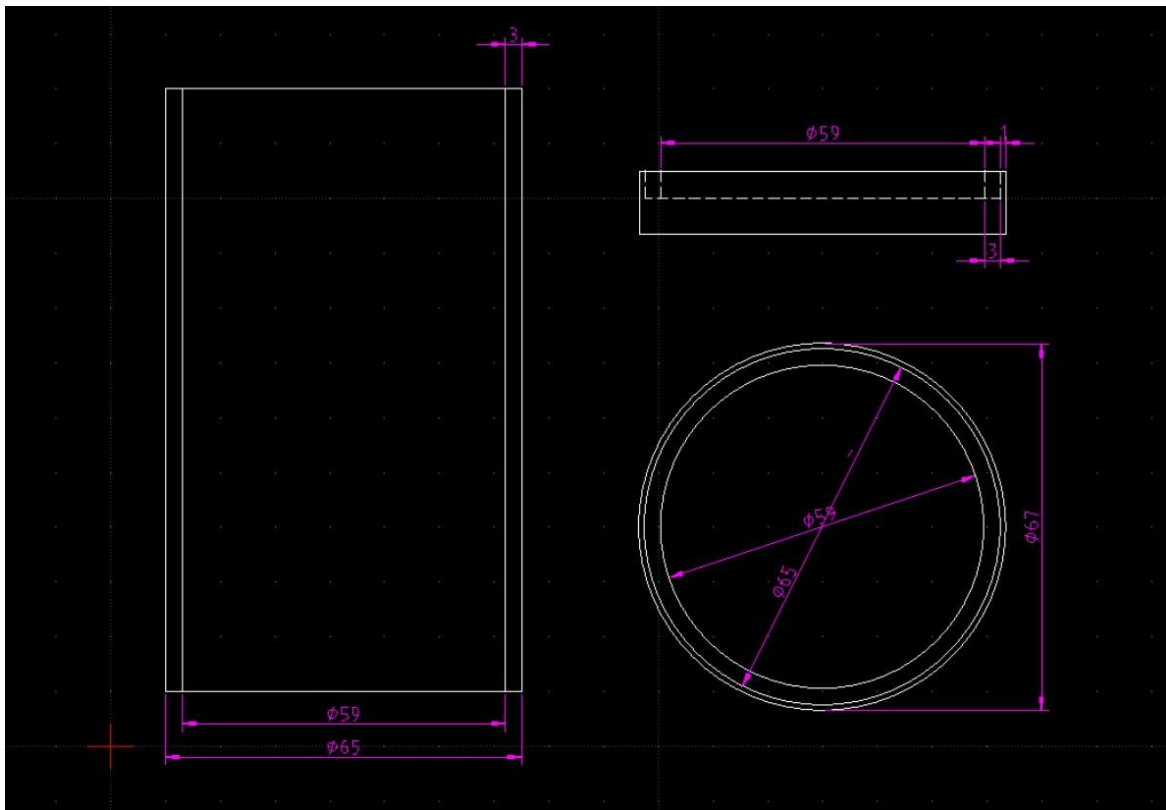
DISEÑO DE LA CARCASA

EXPLICACIÓN DEL DISEÑO

La carcasa es un elemento muy importante del proyecto. Para realizar su diseño hemos optado por tener en cuenta dos aspectos: el peso y la consistencia de la estructura. Para mantener el equilibrio entre estas dos cualidades hemos creado una carcasa que tenga una forma resistente a los golpes y, a su vez, que no pese excesivamente. Lo hemos conseguido insertando una serie de agujeros y huecos.

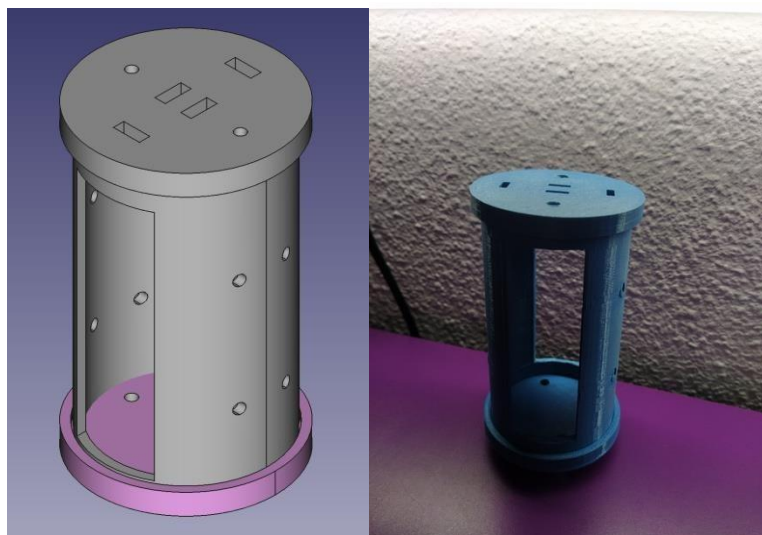


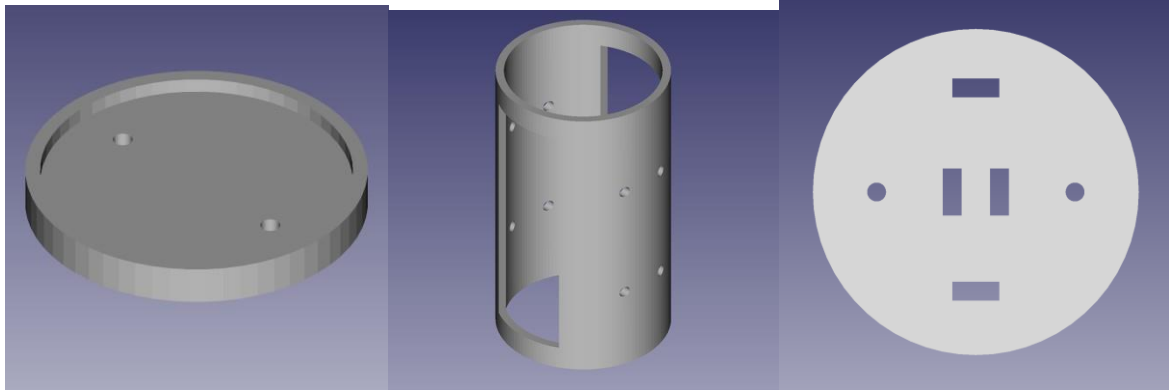
MEDIDAS (REALIZADAS MEDIANTE LIBRECAD)



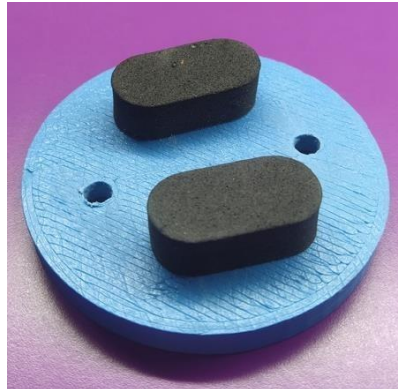
DISEÑO FINAL

El diseño final fue realizado mediante el Freecad. Para desarrollarlo hemos dividido la carcasa en tres piezas: un cilindro con una serie de agujeros en sus paredes y dos tapas desmontables en cada extremo del mismo.





Bajo la tapa inferior hemos puesto unos tacos para que funcionen de amortiguación.



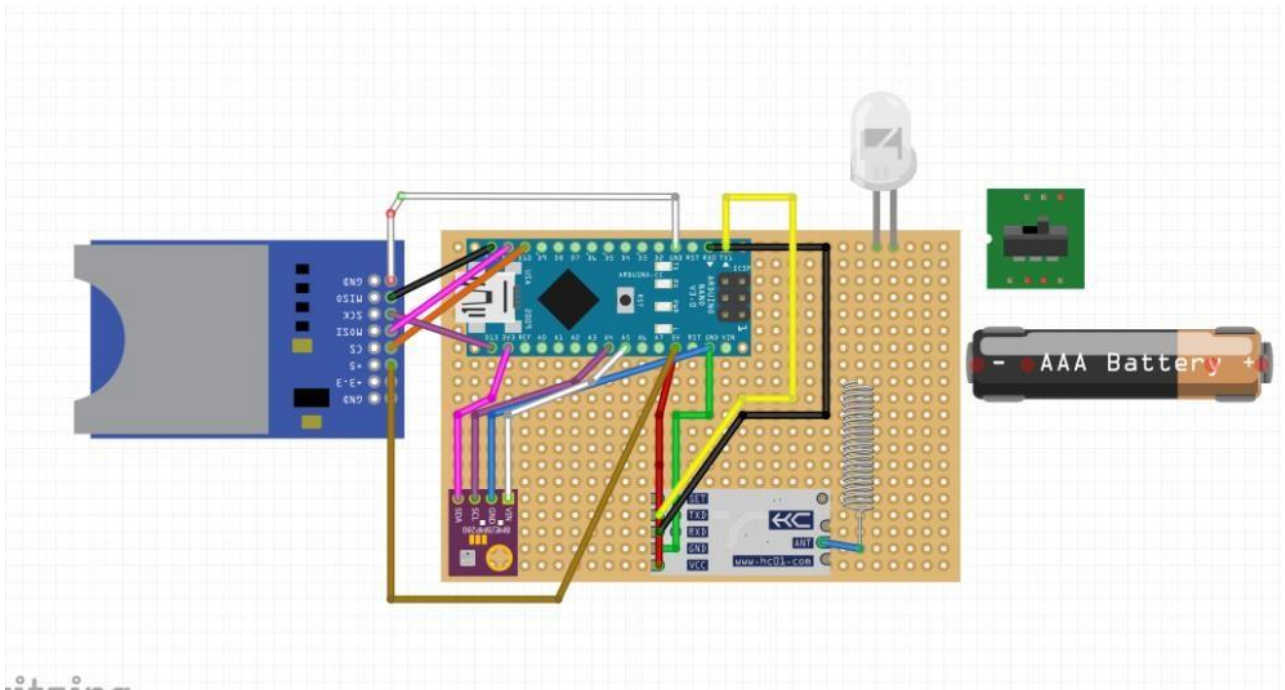
CIRCUITO

CONEXIONES (PINES)

DISPOSITIVO	PIN DE ARDUINO NANO (D=digital; A=analógico; *=pines PWM (3, 5, 6, 9, 10, 11))
Sensor de presión y temperatura BMP280	SCL → A5 SDA → A4
Lector tarjeta microSD	CS → D10* SCK → D13 MISO → D12 MOSI → D11*
Comunicación por radiofrecuencia	TXD → D0 (RX0) RXD → D1 (TX0)
Control de un servomotor	Cable naranja → D9*
Sensor solar ultravioleta	A0
Sensor humedad DHT22	D4
Conexión a un LED	D12
TOTAL	9 pines digitales y 3 analógicos

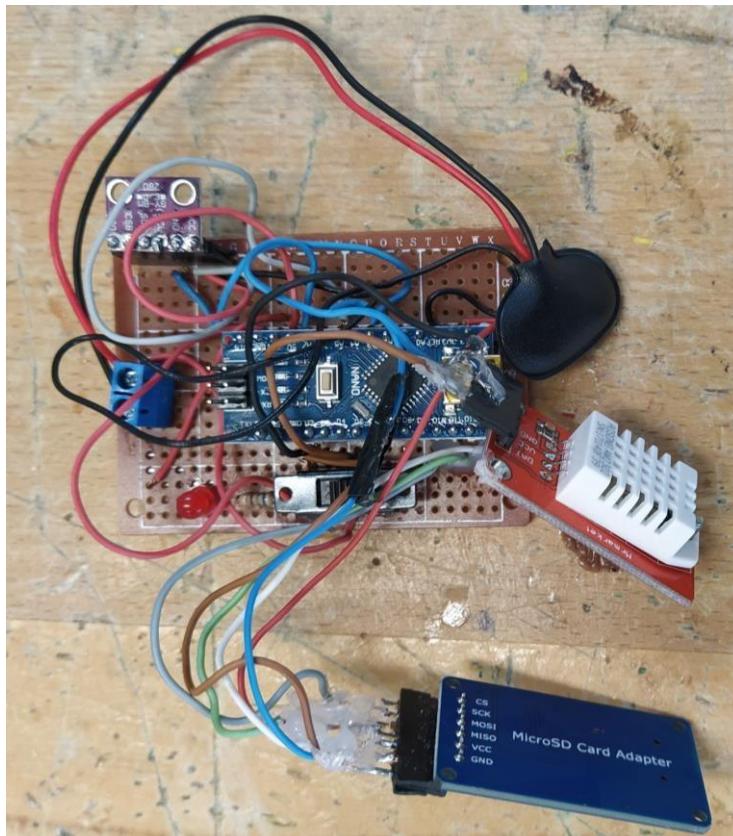
ESQUEMA ELÉCTRICO

EMISOR-SATÉLITE



(no sé cómo hacer las conexiones eléctricas de la pila y el LED)

FOTO DEL CIRCUITO EN LA VIDA REAL



PROGRAMAS

A continuación, se exponen los programas usados o estudiados para este proyecto:

BMP 280

```
#include <Wire.h>    // incluye libreria de bus I2C

#include <Adafruit_Sensor.h> // incluye librerias para sensor BMP280

#include <Adafruit_BMP280.h>

Adafruit_BMP280 bmp;    // crea objeto con nombre bmp

float TEMPERATURA;    // variable para almacenar valor de temperatura

float PRESION;        // variable para almacenar valor de presion atmosferica

void setup() {

    Serial.begin(9600);    // inicializa comunicacion serie a 9600 bps

    Serial.println("Iniciando:");    // texto de inicio

    if ( !bmp.begin() ) {    // si falla la comunicacion con el sensor
mostrar

        Serial.println("BMP280 no encontrado !"); // texto y detener flujo del
programa

        while (1);    // mediante bucle infinito

    }

}

void loop() {

    TEMPERATURA = bmp.readTemperature();    // almacena en variable el valor de
temperatura

    PRESION = bmp.readPressure()/100;    // almacena en variable el valor de
presion dividido

        // por 100 para covertirlo a hectopascales

    Serial.print("Temperatura: ");    // muestra texto

    Serial.print(TEMPERATURA);    // muestra valor de la variable

    Serial.print(" C ");    // muestra letra C indicando grados centigrados

    Serial.print("Presion: ");    // muestra texto

    Serial.print(PRESION);    // muestra valor de la variable

    Serial.println(" hPa");    // muestra texto hPa indicando hectopascales
```



```

    delay(5000);          // demora de 5 segundos entre lecturas
}

```

DHT11

```

#include <DHT.h>          // importa la Librerias DHT

#include <DHT_U.h>

int Pin_SENSOR = 2;      // pin DATA de DHT11 a pin digital 2
int TEMPERATURA;
int HUMEDAD;

DHT dht(Pin_SENSOR, DHT11); // creacion del objeto, cambiar segundo
                             // parametro
                             // por DHT22 si se utiliza en lugar del DHT11

void setup(){
    Serial.begin(9600);    // inicializacion de monitor serial
    dht.begin();           // inicializacion de sensor
}

void loop(){
    TEMPERATURA = dht.readTemperature(); // obtencion de valor de temperatura
    HUMEDAD = dht.readHumidity();        // obtencion de valor de humedad
    Serial.print("Temperatura: ");       // escritura en monitor serial de los
    // valores
    Serial.print(TEMPERATURA);
    Serial.print(" Humedad: ");
    Serial.println(HUMEDAD);
    delay(100);
}

```

Lector SD

```

#include <SPI.h>          // incluye libreria interfaz SPI
#include <SD.h>           // incluye libreria para tarjetas SD
#define SSpin 10         // Slave Select en pin digital 10
File archivo;            // objeto archivo del tipo File

```

```

void setup() {
    Serial.begin(9600);          // inicializa monitor serie a 9600 bps
    Serial.println("Inicializando tarjeta ..."); // texto en ventana de monitor
    if (!SD.begin(SSpin)) {      // inicializacion de tarjeta SD
        Serial.println("fallo en inicializacion !");// si falla se muestra texto
        correspondiente y
        return;                  // se sale del setup() para finalizar el programa
    }

    Serial.println("inicializacion correcta"); // texto de inicializacion
    correcta

    archivo = SD.open("prueba.txt", FILE_WRITE); // apertura para
    lectura/escritura de archivo prueba.txt

    if (archivo) {
        archivo.println("Probando 1, 2, 3"); // escritura de una linea de texto
        en archivo

        Serial.println("Escribiendo en archivo prueba.txt..."); // texto en
        monitor serie

        archivo.close();          // cierre del archivo

        Serial.println("escritura correcta"); // texto de escritura correcta en
        monitor serie
    } else {
        Serial.println("error en apertura de prueba.txt"); // texto de falla en
        apertura de archivo
    }

    archivo = SD.open("prueba.txt"); // apertura de archivo prueba.txt
    if (archivo) {
        Serial.println("Contenido de prueba.txt:"); // texto en monitor serie
        while (archivo.available()) { // mientras exista contenido en el archivo
            Serial.write(archivo.read()); // lectura de a un caracter por vez
        }

        archivo.close();          // cierre de archivo
    } else {
        Serial.println("error en la apertura de prueba.txt");// texto de falla en
        apertura de archivo
    }
}

```

```

    }
}

void loop() {    // funcion loop() obligatoria de declarar pero no utilizada
    // nada por aqui
}

```

PROGRAMA CON TODO JUNTO

No hemos conseguido que este aspecto funcione.

```

#include <SPI.h>    // incluye libreria interfaz SPI
#include <SD.h>     // incluye libreria para tarjetas SD
#define SSpin 10   // Slave Select en pin digital 10
File archivo;     // objeto archivo del tipo File/*
#include <Wire.h>   // incluye libreria de bus I2C
#include <Adafruit_Sensor.h> // incluye librerias para sensor BMP280
#include <Adafruit_BMP280.h>

Adafruit_BMP280 bmp;    // crea objeto con nombre bmp

float TEMPERATURA;     // variable para almacenar valor de temperatura
float PRESION, P0;     // variables para almacenar valor de presion atmosferica
// y presion actual como referencia para altitud
/*
#include <DHT.h>    // importa la Librerias DHT
#include <DHT_U.h>

int Pin_SENSOR = 2;    // pin DATA de DHT11 a pin digital 2
int TEMPERATURA;
int HUMEDAD;

DHT dht(Pin_SENSOR, DHT11); // creacion del objeto, cambiar segundo
parametro

    // por DHT22 si se utiliza en lugar del DHT11
void setup() {
    Serial.begin(9600);    // inicializa monitor serie a 9600 bps
    Serial.println("Inicializando tarjeta ..."); // texto en ventana de monitor
    if (!SD.begin(SSpin)) {    // inicializacion de tarjeta SD

```

```

    Serial.println("fallo en inicializacion !");// si falla se muestra texto
correspondiente y

    return;          // se sale del setup() para finalizar el programa
}

Serial.println("inicializacion correcta"); // texto de inicializacion
correcta

archivo = SD.open("prueba.txt", FILE_WRITE); // apertura para
lectura/escritura de archivo prueba.txt

if (archivo) {

    archivo.println("Probando 1, 2, 3"); // escritura de una linea de texto
en archivo

    Serial.println("Escribiendo en archivo prueba.txt..."); // texto en
monitor serie

    archivo.close();          // cierre del archivo

    Serial.println("escritura correcta"); // texto de escritura correcta en
monitor serie

} else { }

archivo = SD.open("prueba.txt");    // apertura de archivo prueba.txt

if (archivo) {

    Serial.println("Contenido de prueba.txt:"); // texto en monitor serie

    while (archivo.available()) {    // mientras exista contenido en el archivo

        Serial.write(archivo.read());    // lectura de a un caracter por vez

    }

    archivo.close();          // cierre de archivo

} else {

    Serial.println("error en la apertura de prueba.txt");// texto de falla en
apertura de archivo

}

    Serial.println("error en apertura de prueba.txt"); // texto de falla en
apertura de archivo

    Serial.begin(9600);          // inicializa comunicacion serie a 9600 bps

    Serial.println("Iniciando:");    // texto de inicio

    if ( !bmp.begin() ) {          // si falla la comunicacion con el sensor
mostrar

```

```

    Serial.println("BMP280 no encontrado !"); // texto y detener flujo del
programa

    while (1);          // mediante bucle infinito
}

P0 = bmp.readPressure()/100;    // almacena en P0 el valor actual de
presion

Serial.begin(9600);    // inicializacion de monitor serial

dht.begin();          // inicializacion de sensor

// put your setup code here, to run once:
}

void loop() {

    TEMPERATURA = bmp.readTemperature();    // almacena en variable el valor de
temperatura

    PRESION = bmp.readPressure()/100;    // almacena en variable el valor de
presion dividido

        // por 100 para covertirlo a hectopascales

    Serial.print("Temperatura: ");    // muestra texto
    Serial.print(TEMPERATURA);        // muestra valor de la variable
    Serial.print(" C ");              // muestra letra C indicando grados centigrados
    Serial.print("Presion: ");        // muestra texto
    Serial.print(PRESION);            // muestra valor de la variable
    Serial.println(" hPa");           // muestra texto hPa indicando hectopascales
    Serial.print("Altitud aprox: ");  // muestra texto

    Serial.print(bmp.readAltitude(P0)); // muestra valor de altitud con
referencia a P0

    Serial.println(" m");              // muestra letra m indicando metros
    delay(5000);                      // demora de 5 segundos entre lecturas

    TEMPERATURA = dht.readTemperature(); // obtencion de valor de temperatura
    HUMEDAD = dht.readHumidity();       // obtencion de valor de humedad

    Serial.print("Temperatura: "); // escritura en monitor serial de los
valores

    Serial.print(TEMPERATURA);

    Serial.print(" Humedad: ");

```

```
Serial.println(HUMEDAD);  
delay(100);  
// put your main code here, to run repeatedly:  
}
```