

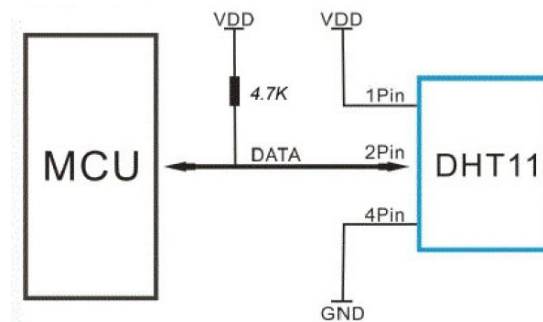
Clase III - Módulo B

¿Cómo agregar nuevos
módulos a mi proyecto?

Nuevo sensor: DHT11

1) Busco información, conexasionado, tipo de comunicación. **Idealmente ir al Datasheet.** Por ejemplo:

<https://datasheetpdf.com/datasheet-pdf/785592/DHT11.html>

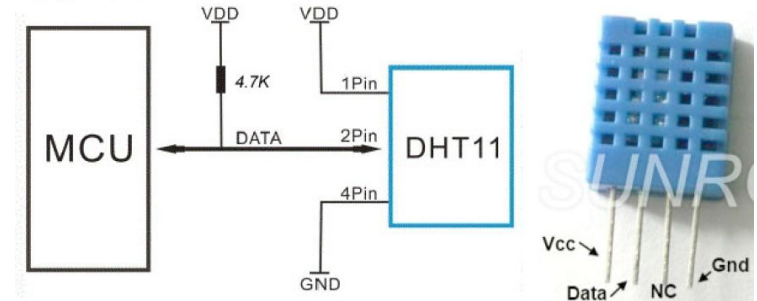
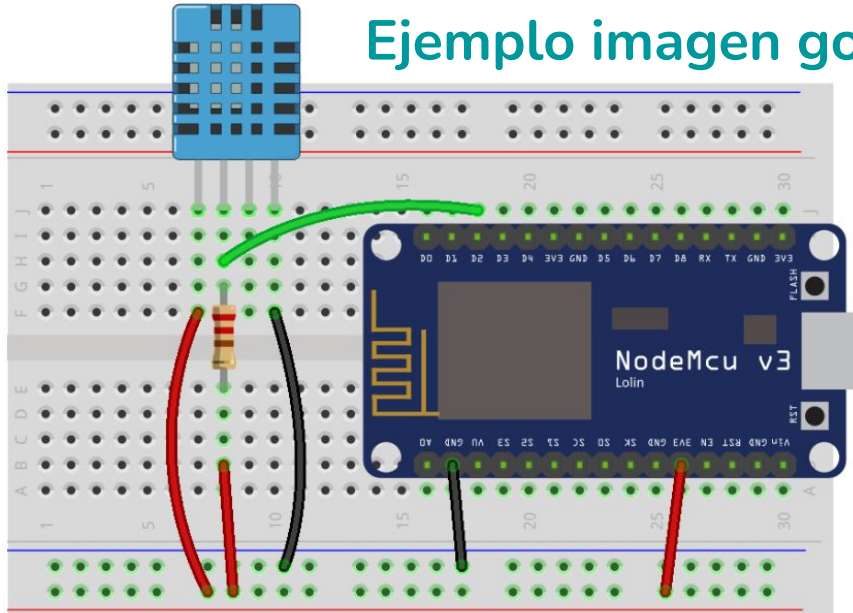


Item	Condition	Min	Typical	Max	Unit
Power supply	DC	3	5	5.5	V
Current supply	Measuring	0.5		2.5	mA
	Stand-by	100	Null	150	uA
	Average	0.2	Null	1	mA

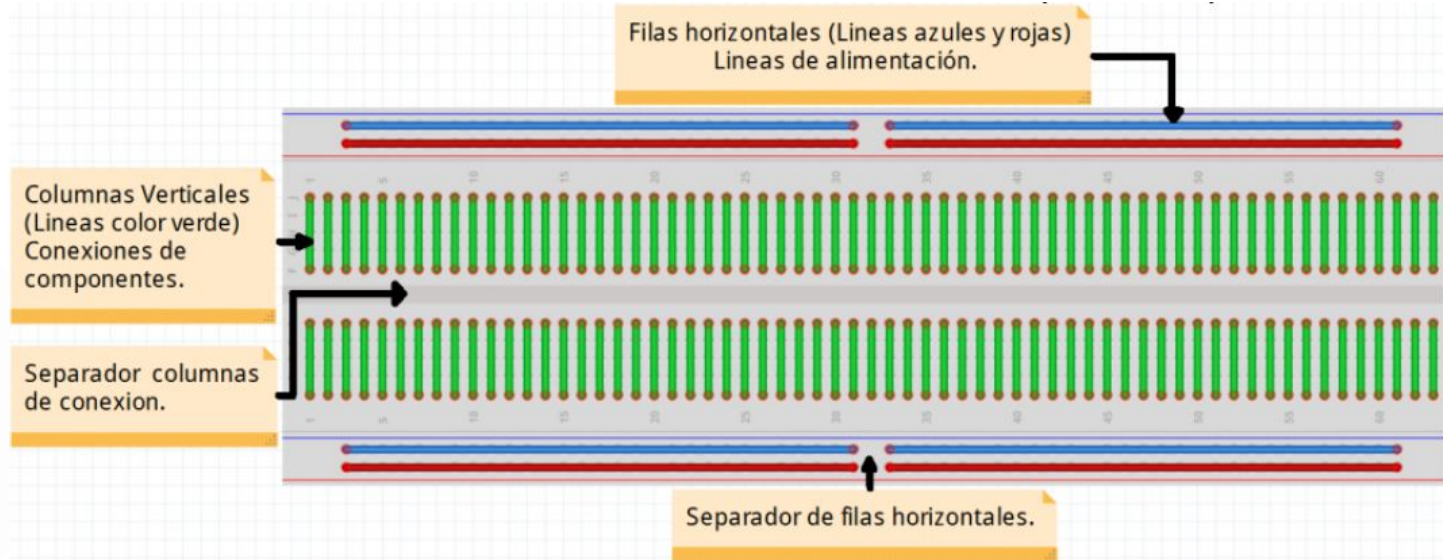
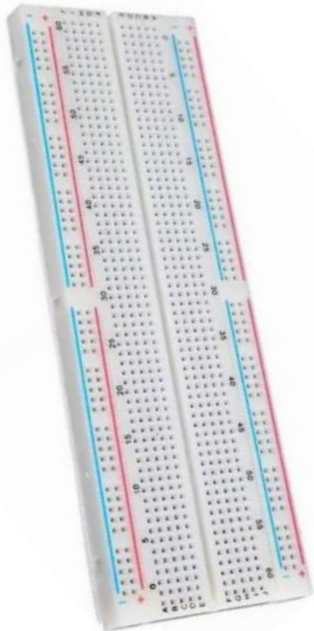
Nuevo sensor: DHT11

2) Lo conecto a mi microcontrolador. Elijo un pin digital adecuado.

Ejemplo imagen google: pin digital D2

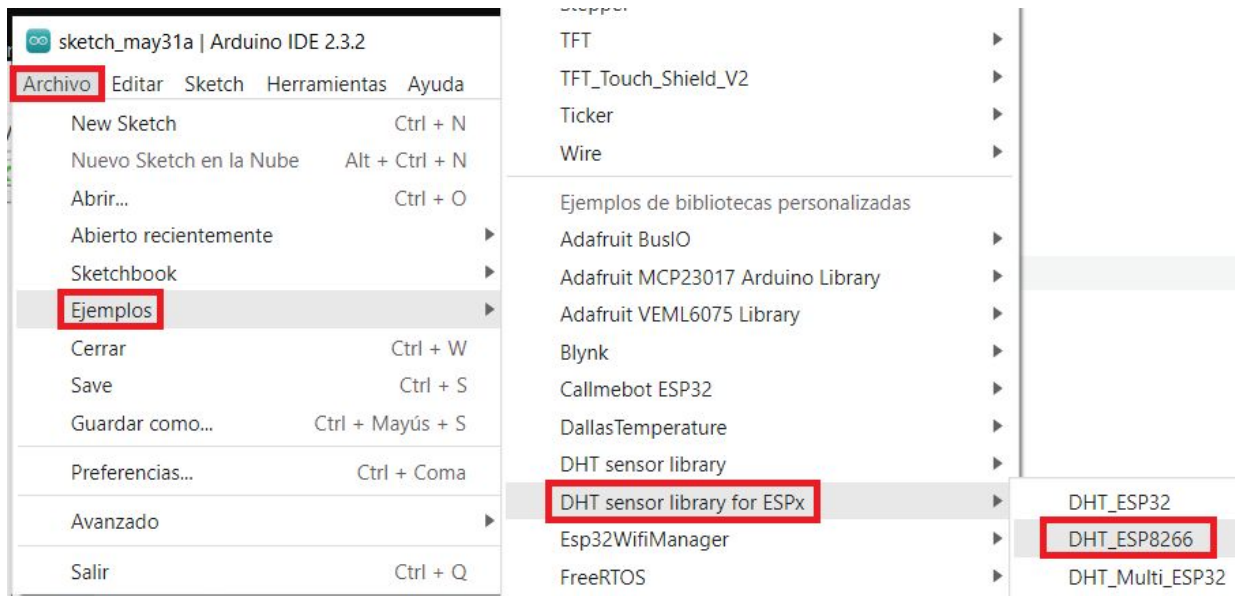


Nota: protoboard



Nuevo sensor: DHT11

3) Busco código en ejemplos de Arduino.



Nuevo sensor: DHT11

- 4) Análisis código fuente: **LIBRERÍAS** y **DEFINICIONES**
Instalo librerías necesarias.



The screenshot shows the Arduino IDE interface. At the top, the board is set to "NodeMCU 0.9 (ESP-12...)". On the left, the "GESTOR DE BIBLIOTECAS" (Library Manager) is open, with "DHTesp" entered in the search bar. Below the search bar, the filters are set to "Tipo: Todo" and "Tema: Todo". The search results show "DHT sensor library for ESPx de beegee_tokyo" with version "1.19" marked as "instalado". On the right, the code editor shows the file "DHT_ESP8266.ino" with the following code:

```
1  #include "DHTesp.h" // Click here to get the library:
2
3  #ifdef ESP32
4  #pragma message(THIS EXAMPLE IS FOR ESP8266 ONLY!)
5  #error Select ESP8266 board.
6  #endif
7
8  DHTesp dht;
```


Nuevo sensor: DHT11

5) Análisis código fuente: **SETUP**.

Adapto a mi conexionado y tipo de sensor.

Placa entrenadora TR3BOL: pin digital D7

```
void setup()
{
  Serial.begin(115200);
  Serial.println();
  Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)\tHeatIndex (C)\t(F)");
  String thisBoard= ARDUINO_BOARD;
  Serial.println(thisBoard);

  // Autodetect is not working reliable, don't use the following line
  // dht.setup(17);
  // use this instead:
  dht.setup(17, DHTesp::DHT22); // Connect DHT sensor to GPIO 17 → dht.setup(D7, DHTesp::DHT11);
}
```

Nuevo sensor: DHT11

Analizo código fuente: **LOOP**. Interpreto el código.
Adapto a mi aplicación.

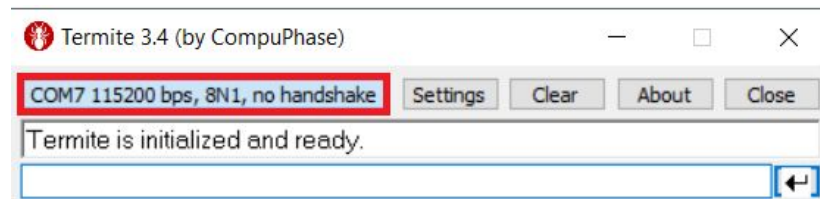
```
void loop(){  
  delay(dht.getMinimumSamplingPeriod());  
  float humidity = dht.getHumidity();  
  float temperature = dht.getTemperature();  
  Serial.print(dht.getStatusString());  
  Serial.print("\t");  
  Serial.print(humidity, 1);  
  Serial.print("\t\t");  
  Serial.print(temperature, 1);  
  Serial.print("\t\t");  
  Serial.print(dht.toFahrenheit(temperature), 1);  
  Serial.print("\t\t");  
  Serial.print(dht.computeHeatIndex(temperature, humidity, false), 1);  
  Serial.print("\t\t");  
  Serial.println(dht.computeHeatIndex(dht.toFahrenheit(temperature), humidity, true), 1);  
  delay(2000);  
}
```


Nuevo sensor: DHT11

6) Compilo el programa.

¿Errores de compilación? LEO e INTERPRETO.

Error típico



```
Salida
. Variables and constants in RAM (global, static), used 28592 / 80192 bytes (35%)
| SEGMENT BYTES DESCRIPTION
| DATA 1504 initialized variables
| RODATA 984 constants
| BSS 26104 zeroed variables
. Instruction RAM (IRAM_ATTR, ICACHE_RAM_ATTR), used 59815 / 65536 bytes (91%)
| SEGMENT BYTES DESCRIPTION
| ICACHE 32768 reserved space for flash instruction cache
| IROM 27047 code in IROM
. Code in flash (default, ICACHE_FLASH_ATTR), used 236276 / 1048576 bytes (22%)
| SEGMENT BYTES DESCRIPTION
| IROM 236276 code in flash

A fatal esptool.py error occurred: could not open port 'COM7': PermissionError(13,
'Acceso denegado.', None, 5)esptool.py v3.0 Serial port COM7
```

Nuevo sensor: DHT11

Compilo el programa. ¿Más errores de compilación?
LEO e INTERPRETO.

Error típico 2

DHT_ESP8266.ino

```
19 void loop()
20 {
21     float humidity = dht.getHumidity();
22     Serial.print("DHT11 ")
23     Serial.print(dht.getStatusString());
24     Serial.print("\t");
25     Serial.println(humidity, 0);
26     delay(2000);
27 }
```

Salida

```
C:\Users\Win10\AppData\Local\Temp\.arduinoIDE-un
C:\Users\Win10\AppData\Local\Temp\.arduinoIDE-un
22 | Serial.print("DHT11 ")
   |                      ^
   |                      ;
23 | Serial.print(dht.getStatusString());
   | ~~~~~
exit status 1
Compilation error: expected ';' before 'Serial'
```

Nuevo sensor: DHT11

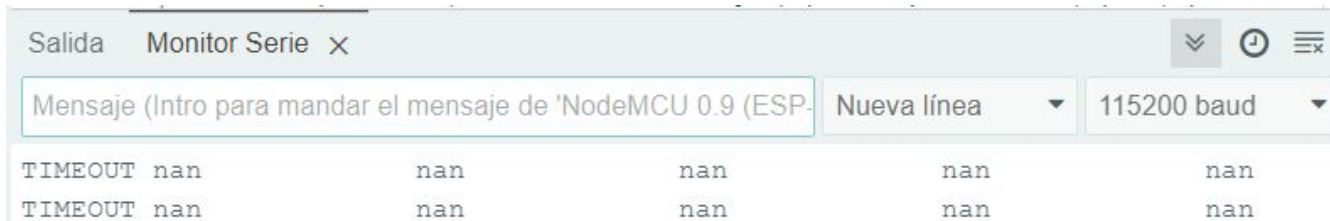
7) Cargo el programa, verifico funcionamiento.

Ejemplo: ¿usa la UART? ¿a qué velocidad? ¿qué debería mostrar?

Mal baudrate



Mal conexionado o sensor



Nuevo sensor: DHT11

8) Verifico desempeño, tiempos, información

Conexión y programación exitosa

Salida Monitor Serie x					
Mensaje (Intro para mandar el mensaje de 'NodeMCU 0.9 (ESP-)				Nueva línea ▾	115200 baud ▾
OK	51.0	17.0	62.6	16.1	61.0
OK	50.0	17.0	62.6	16.1	60.9

```
Serial.print(humidity, 1);
Serial.print("\t\t");
Serial.print(temperature, 1);
Serial.print("\t\t");
Serial.print(dht.toFahrenheit(temperature), 1);
Serial.print("\t\t");
Serial.print(dht.computeHeatIndex(temperature, humidity, false), 1);
Serial.print("\t\t");
Serial.println(dht.computeHeatIndex(dht.toFahrenheit(temperature), humidity, true), 1);
```

Nuevo sensor: DHT11

9) Adapto código a mi aplicación.

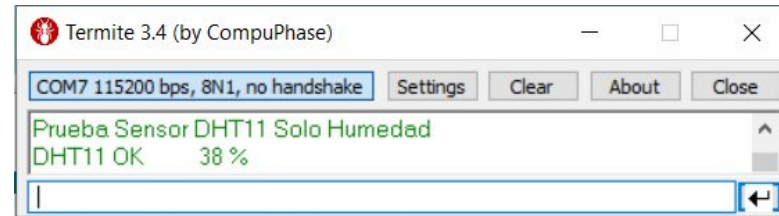
Ejemplo sólo Humedad → elimino instrucciones de T°

```

20 void loop(){
21     delay(dht.getMinimumSamplingPeriod());
22     float humidity = dht.getHumidity();
23     float temperature = dht.getTemperature();
24     Serial.print(dht.getStatusString());
25     Serial.print("\t");
26     Serial.print(humidity, 1);
27     Serial.print("\t\t");
28     Serial.print(temperature, 1);
29     Serial.print("\t\t");
30     Serial.print(dht.toFahrenheit(temperature), 1);
31     Serial.print("\t\t");
32     Serial.print(dht.computeHeatIndex(temperature, humidity, false), 1);
33     Serial.print("\t\t");
34     Serial.println(dht.computeHeatIndex(dht.toFahrenheit(temperature), humidity, true), 1);
35     delay(2000);
36 }
    
```

Nuevo sensor: DHT11

10) Lo encapsulo en un archivo externo



DHT_ESP8266.ino miDHT11.h ...

```
1  #include "miDHT11.h"
2  void setup() {
3      delay(400);
4      Serial.begin(115200);
5      Serial.println("\nDHT11 Solo Humedad");
6      iniciaDHT11();
7  }
8  void loop(){
9      float hum = leeDHT11();
10     Serial.print(hum, 0);
11     Serial.println(" %");
12     delay(2000);
13 }
```

DHT_ESP8266.ino miDHT11.h

```
1  #include "DHTesp.h" // Click here to get the libra
2  #define PIN_DHT11 D2
3  #ifdef ESP32
4  #pragma message(THIS EXAMPLE IS FOR ESP8266 ONLY!)
5  #error Select ESP8266 board.
6  #endif
7  DHTesp dht;
8  void iniciaDHT11(){
9      dht.setup(PIN_DHT11, DHTesp::DHT11);
10 }
11 float leeDHT11(){
12     float humidity = dht.getHumidity();
13     Serial.print("DHT11 ");
14     Serial.print(dht.getStatusString());
15     Serial.print("\t");
16     return humidity;
17 }
```


Nuevo sensor: DHT11

Prueba de funcionamiento final

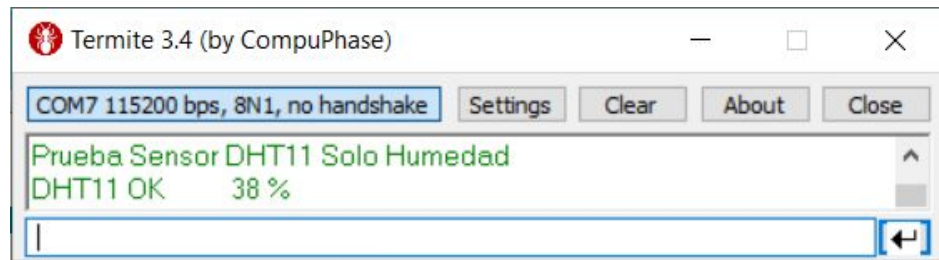
DHT_ESP8266.ino

miDHT11.h

...

```

1  #include "miDHT11.h"
2  void setup() {
3      delay(400);
4      Serial.begin(115200);
5      Serial.println("\nDHT11 Solo Humedad");
6      iniciaDHT11();
7  }
8  void loop(){
9      float hum = leeDHT11();
10     Serial.print(hum, 0);
11     Serial.println(" %");
12     delay(2000);
13 }
    
```



Ejercicio Final Integrador

Ejercicio Final Integrador

Crear un **sistema de monitoreo** de Humedad (extraída del sensor DHT11) y de Temperatura (extraída de la sonda DS18B20).

La humedad se expresa en porcentaje, con una precisión de $\pm 2\%$. La T° se expresa en $^\circ\text{C}$ con una precisión de $\pm 0.1^\circ\text{C}$. Ambas variables se deben visualizar tanto en el Display LCD16x02 como en una interfaz gráfica en Blynk.

- **Alarma 1:** incendio, simulada con el pulsador SW1
- **Alarma 2:** exceso de T° por encima de 30°C .

Ejercicio Final Integrador

Ante cualquier evento de alarma, se debe hacer sonar el Buzzer de forma intermitente cada 200 ms, y mostrar advertencia en Display y en Blynk.

La alarma 2 se recupera cuando desciende la T° por debajo de 30°C .

La alarma 1 se recupera mediante el pulsador SW2 de la placa, un switch en Blynk y/o recibiendo por puerto serie una letra Z.

¡Utilizar buenas prácticas de programación, interrupciones, timers, modularizar en distintos archivos.h y utilizar las recomendaciones explicadas en el taller!

¡Personaliza el Display y la interfaz gráfica a tu gusto! :)

Próximos pasos

- **1 mes** más disponibles los **VIDEOS** en la plataforma de Zoom para ver o descargar (27/08).
- **1 mes** más disponible el grupo de **CONSULTAS** (27/08).
- **1 mes** para la entrega de los **EJERCICIOS** (27/08)
- Encuesta de **FEEDBACK**
- Muestra de PROYECTOS y ENTREGA DE CERTIFICADOS (A coordinar con quienes realicen la entrega)

#CICLO DE MASTERCLASS GRATUITA

Segundo o tercer JUEVES de cada MES

#1 ¡MÁS ALLÁ DEL FENÓMENO! TODO LO QUE NECESITAS SABER SOBRE IOT

#2 LA PROMESA DEL IOT ES REAL

#3 LAS 4 VÍAS RÁPIDAS PARA UNA RENTABILIDAD SEGURA DEL IOT -> 15/08

#4 GENERACIÓN IOT: DESAFÍOS, FUERZA DE TRABAJO, EMPLEABILIDAD

#5 LA GESTIÓN DEL CAMBIO: LA LLEGADA DEL IOT A TU EMPRESA

#6 ERRORES COMUNES Y CÓMO ENFRENTARLOS

#7 SEGURIDAD EN EL IOT. ESTÁNDARES Y TECNOLOGÍA

TERCERA Y ÚLTIMA
EDICIÓN DEL AÑO.
(Septiembre)

Propuesta de taller
IoT AVANZADO.
(Octubre/Noviembre)

 **TR3BOL**
ARGENTINA

MI PRIMER
DESARROLLO II  **T**
3ra Edición

Aprendé cómo implementar
herramientas **gratuitas y Open Source**
para crear tu **primer desarrollo IIOT**

 IIoT

 7 de Septiembre
 10:00hs - 13:00 hs
 3 módulos virtuales

¡Incluye placa entrenadora y programable
TR3BOL con sensores de humedad y
temperatura, Display y mucho más!



¿Cómo ha sido tu experiencia
en este taller?