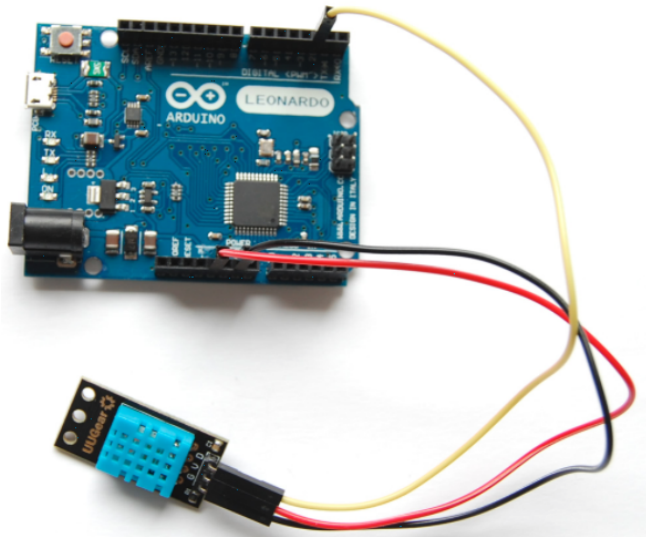


Temperatura y humedad

Vamos a utilizar dos sensores para medir estas variables: **DHT12** y su hermano pequeño **DHT11**. Ya explicamos estos sensores en el capítulo de sensores <https://libros.catedu.es/books/programa-arduino-mediante-codigo/page/sensores>



¿Cómo se conecta?

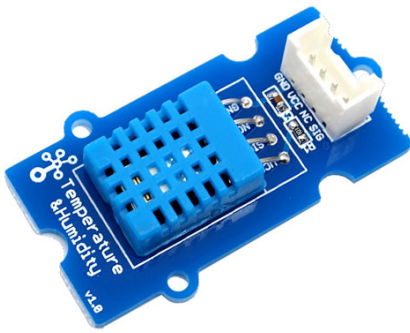
Para ello tendrás que realizar las conexiones necesarias para poder obtener los datos desde Arduino. Se trata de un sensor digital que utiliza 3 pines: Alimentación +5V, tierra/GND (-) y pin de datos (out) por donde se envían los datos de humedad y temperatura.

A veces el sensor viene sobre 4 pines, en este caso uno de ellos no se conecta.

Como las medidas de humedad y temperatura van por un solo pin, la información se transmite como un tren de pulsos en serie, por lo tanto, necesitamos un programa que "extraiga" eso dos datos de forma diferenciada. Para ello vamos a usar una librería referenciada por **DHT11.h**. A través del monitor serie del IDE de Arduino podremos ver las medidas obtenidas.

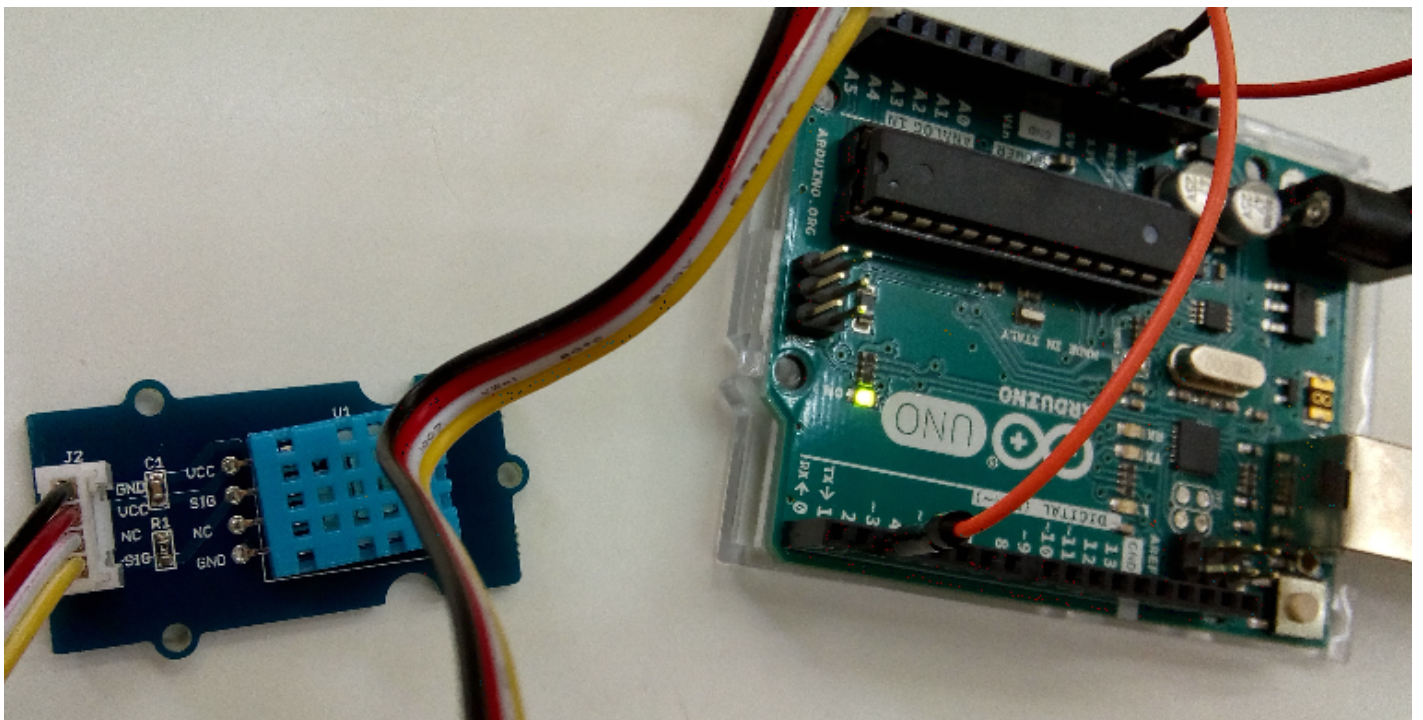
Esquema DHT11 :

Es un modelo "conectar y listo" que ya vienen con los cables preparados, pero si te fijas son en este orden : GND - 5V -NC - D2 donde NC significa NO CONECTADO y D2 son los datos

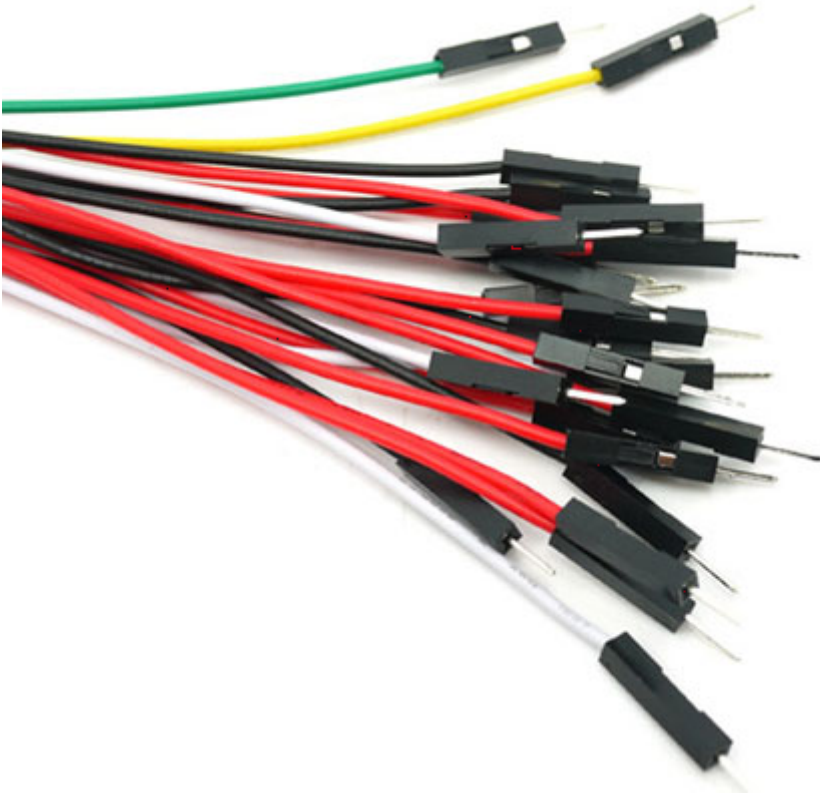


Lo conectamos en el Arduino sin necesidad de placa Protoboard:

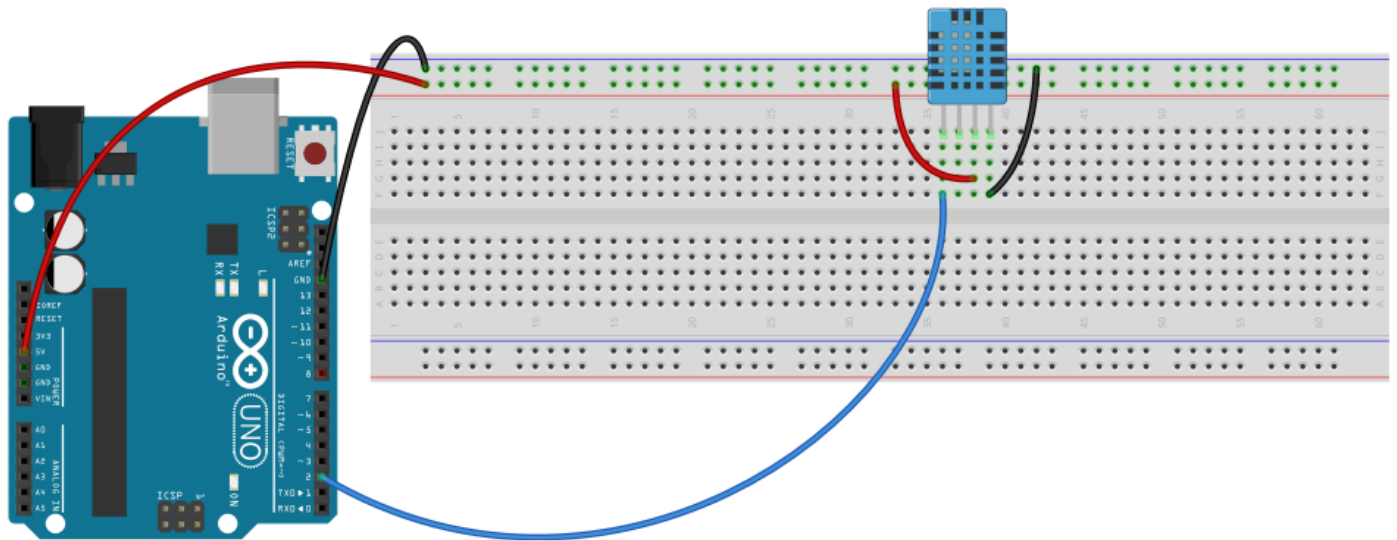
Para este montaje, la colocación de la placa Edubásica no implica ninguna ventaja.



utilizando cables macho-macho

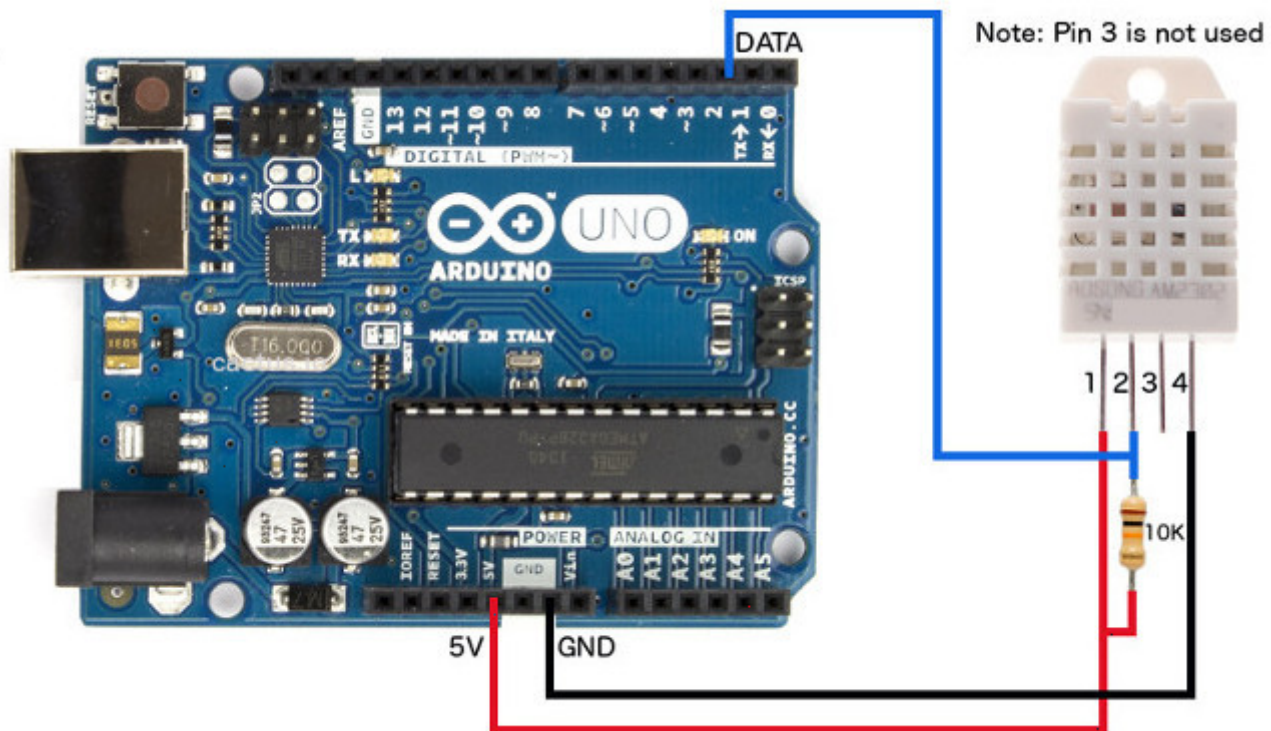


Esquema de conexión:

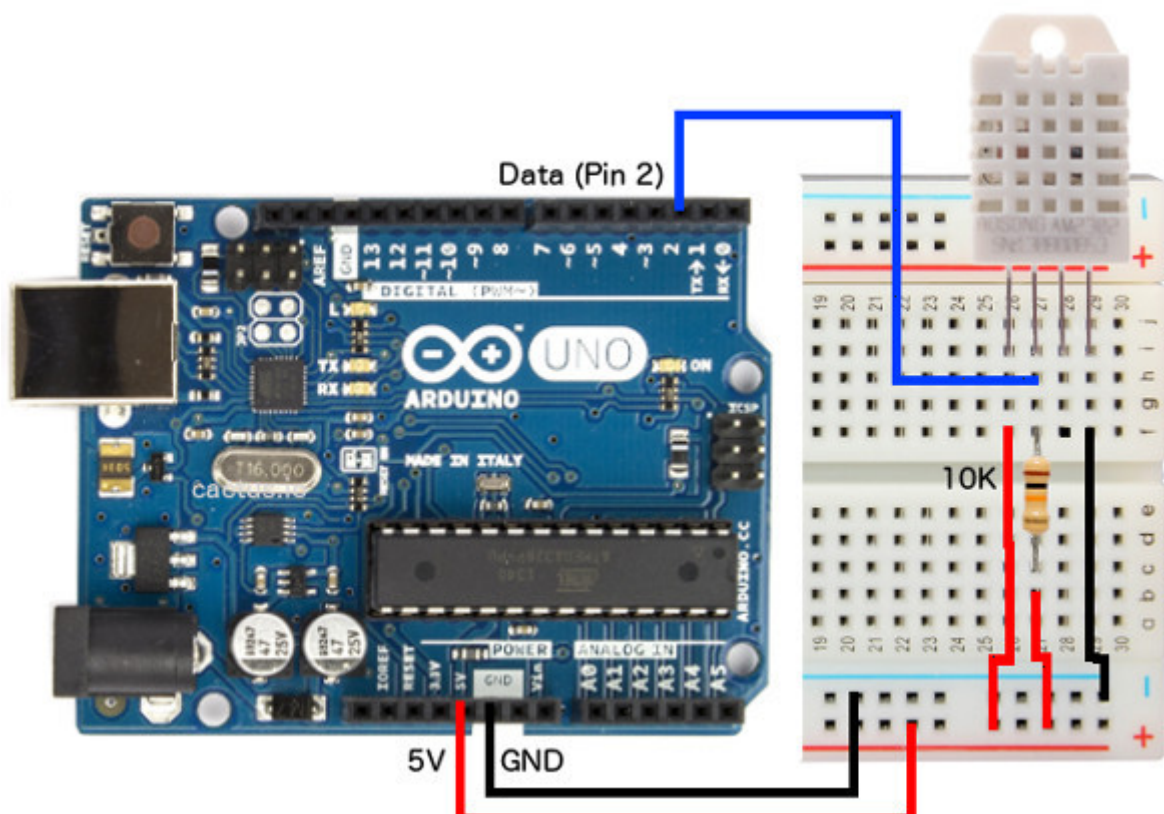


Esquema DHT12

Es un sensor que necesita 3.3V pero si trabajas con 5V necesitas hacer este puente con una resistencia de más o menos 10K



O con la placa Protoboard



Fuente [Cactus.io](https://cactus.io) CC BY-NC-SA 3.0

Librería DHT11.h

Para poder utilizar este dispositivo, necesitas esta librería, lo primero que tienes que hacerlo es descargarlo, puedes hacerlo desde <https://github.com/adafruit/DHT-sensor-library> o este enlace de [Drive](#) (zip - 4,29 KB), o simplemente buscando DHT11 library en Internet.

Una vez descargado, tienes que incorporarlo en tu librería, aquí tienes cómo hacerlo

Librería DHT12.h

Para poder utilizar este dispositivo, necesitas esta librería, lo primero que tienes que hacerlo es descargarlo, puedes hacerlo desde <https://github.com/adafruit/DHT-sensor-library> o simplemente buscando DHT12 library en Internet.

Una vez descargado, tienes que incorporarlo en tu librería, aquí tienes cómo hacerlo

Montaje 15 Medición T y H por puerto serie con DHT11:

<https://create.arduino.cc/editor/javierquintana/57997bda-7323-4f8a-9157-92f9611dbdba/preview>

<https://create.arduino.cc/editor/javierquintana/57997bda-7323-4f8a-9157-92f9611dbdba/preview?embed>

El resultado se puede ver en este vídeo, simplemente soplando nuestro vaho pasamos de 20% de humedad y 22°C a 93% y 24°C.

Si lo hacéis con niños, enseguida se les ocurre ponerlo en el sobaco, menos mal que solo son 5V ;) _

<https://www.youtube.com/embed/gLxdSaxOjbY?rel=0>

Montaje 16 Medición T y H por puerto serie con DHT12

La librería de este sensor es más potente y nos puede decir la sensación térmica:

<https://create.arduino.cc/editor/javierquintana/66a222b4-8e95-4693-b8db-df5a1fdb68a3/preview>

<https://create.arduino.cc/editor/javierquintana/66a222b4-8e95-4693-b8db-df5a1fdb68a3/preview?embed>

El resultado es:

```
DHT22 Humidity - Temperature Sensor
RH          Temp (C)    Temp (F)    Heat Index (C)  Heat Index (F)
43.60 %     21.10 *C       69.98 *F    25.05 *C       77.09 *F
43.60 %     21.10 *C       69.98 *F    25.05 *C       77.09 *F
43.70 %     21.10 *C       69.98 *F    25.05 *C       77.09 *F
```

Montaje 17 H y T por LCD

Ahora vamos a conectarlo por LCD :

<https://www.youtube.com/embed/6J8PKruEcD4>

Conexiones

Si tienes el DHT11 o si tienes el DH12 lo has visto ya Ahora añade el LCD con el I2C

Programa

En este caso lo hacemos con el DHT12 ya sabes que si utilizas DHT11 no mide la humedad y la sensación térmica

<https://create.arduino.cc/editor/javierquintana/10dcd8cb-1128-4959-8d78-70464cea939f/preview>

<https://create.arduino.cc/editor/javierquintana/10dcd8cb-1128-4959-8d78-70464cea939f/preview?embed>

Processing

Es un programa similar al IDE de Arduino que has estado manejando hasta ahora, sólo cambia en un botón de PLAY y en otro de STOP. Es software abierto desarrollado en Java por *Ben Fry* y *Casey*

Reas a raíz de una reflexiones en un congreso donde se detecto esta necesidad. Te lo puedes descargar de <http://processing.org>

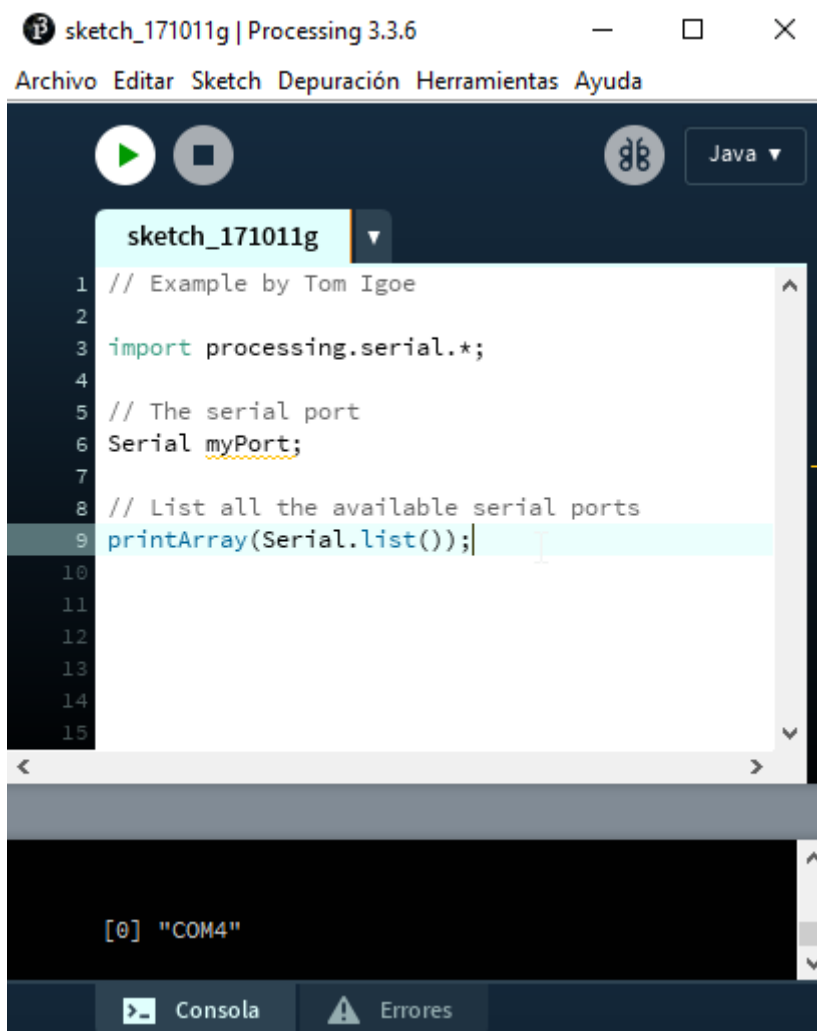
¿Para qué sirve?

Es muy común tener la necesidad de representar los datos que nos da Arduino en un entorno visual mucho más atractivo que el monitor serie que nos ofrece IDE Arduino. Si quieres saber las instrucciones que tiene y más información [consulta esta página](#).

Saber el puerto de conexión

Para empezar a utilizar Processing con nuestro Arduino necesitamos saber en qué puerto se conecta, una forma fácil es cargar y ejecutar este código con el Arduino conectado y que liste los puertos, esta instrucción **printArray(Serial.list());** nos lo puede decir

Y el resultado puedes ver que sale abajo en la consola [0] "COM4" luego es el 0 en mi caso



```
sketch_171011g | Processing 3.3.6
Archivo Editar Sketch Depuración Herramientas Ayuda

1 // Example by Tom Igoe
2
3 import processing.serial.*;
4
5 // The serial port
6 Serial myPort;
7
8 // List all the available serial ports
9 printArray(Serial.list());
10
11
12
13
14
15

[0] "COM4"
```

Montaje 18 Representación gráfica de medidas con Processing.

Una vez obtenidos los datos de temperatura y humedad a través del sensor DHT11 desde Arduino, enviamos, a través del puerto serie, estos datos al PC, donde tenemos ejecutando un programa en Processing que está "escuchando" el puerto serie, obteniendo los datos y representándolos en pantalla. Simultáneamente guardamos los datos en un archivo de texto que posteriormente podremos analizar en una hoja de cálculo.

Programa a cargar en el ARDUINO

Cargamos este programa, fíjate que sólo ponemos un valor de la temperatura, si queremos representar la humedad, quitamos el comentario de la temperatura y ponemos el de la humedad

IMPORTANTE: No hay que tener abierto el monitor serie del IDE de Arduino porque ocupa el puerto y, por lo tanto, no deja leer los datos a Processing.

Si fuera un DHT12 en vez de un DHT11 poner comentarios a las 4 primeras líneas delante // y quitárselas a las 3 siguientes

<https://create.arduino.cc/editor/javierquintana/09ac70ea-0d4b-4d26-bce4-4b45bb45d3cb/preview>

<https://create.arduino.cc/editor/javierquintana/09ac70ea-0d4b-4d26-bce4-4b45bb45d3cb/preview?embed>

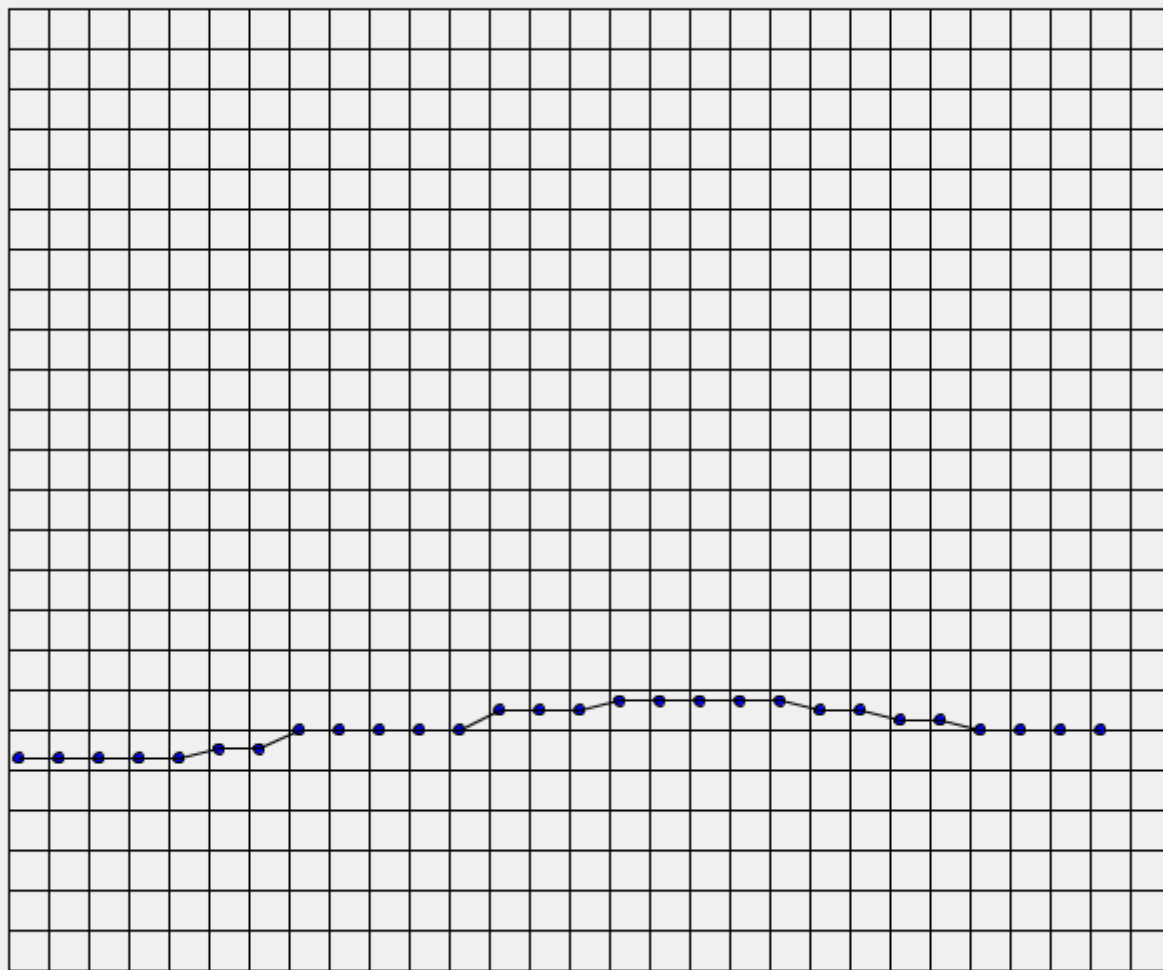
Programa a Cargar en PROCESING

Aquí lo tienes (rar - 1,96 KB), sólo representa un valor, está puesto en el puerto 0 puertoArduino = new Serial(this, Serial.list()[0], 9600);

El resultado puedes verlo aquí abajo para la temperatura, el aumento se debe a aplicar vaho al sensor:

VALORES:

25,000



Muestras

Otro programa de visualización de datos

En este caso no vamos a representar los datos en forma de gráfica, sino por colores, y además vamos a añadir un botón que encienda un LED conectado por simplicidad en el pin 13

Programa a cargar en el ARDUINO

El programa lee la temperatura y lo escribe en el puerto serie en forma de byte. También lee el puerto serie para cambiar el estado del led.

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11 // DHT 11
DHT dht(DHTPIN, DHTTYPE);
```

```

boolean status=LOW; //Estado del led

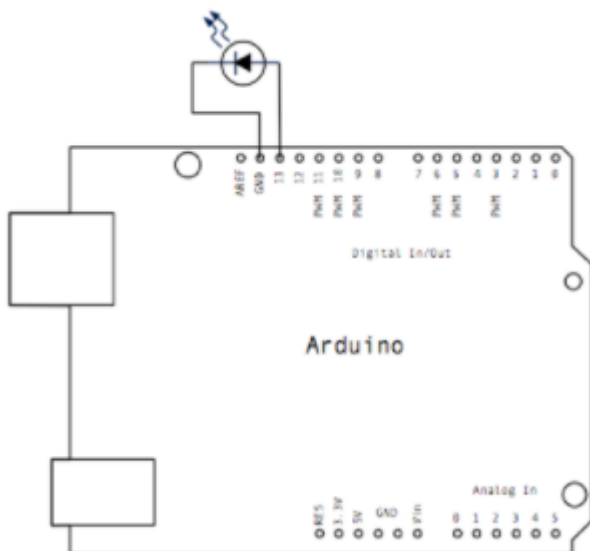
void setup() {
  Serial.begin(9600);
  pinMode(13,OUTPUT);
  dht.begin();
}

void loop() {
  delay(100);

  //float h = dht.readHumidity();
  int temp = dht.readTemperature();
  Serial.write(temp); //Enviamos los datos en forma de byte
  if(Serial.available()>0)//Si el Arduino recibe datos a través del puerto serie
  {
    byte dato = Serial.read(); //Los almacena en la variable "dato"
    if(dato==65) //Si recibe una "A" (en ASCII "65")
    {
      status=!status; //Cambia el estatus del led
    }
    digitalWrite(13,status);
  }
}

```

En el Arduino tenemos que poner el sensor de temperatura y humedad tal y como se ha explicado en el Montaje 8 y además un led en el 13



Programa en Processing

te lo puedes descargar [aquí](#) (rar - 31,02 KB)(recuerda cambiar port = new Serial(this, Serial.list()[0], 9600); por tu puerto)



LED OFF

Temperatura = 18 °C ●

dweet.io

Disponemos del portal web **dweet.io** que nos ofrece un servicio para enviar y representar datos en la nube sin necesidad, ni si quiera, de registrarnos en la plataforma.

Vamos a ver los pasos a seguir:

1. Probamos la plataforma introduciendo un dato, para ello en el navegador tecleamos por ejemplo (cambia **catedu** por tu nombre):

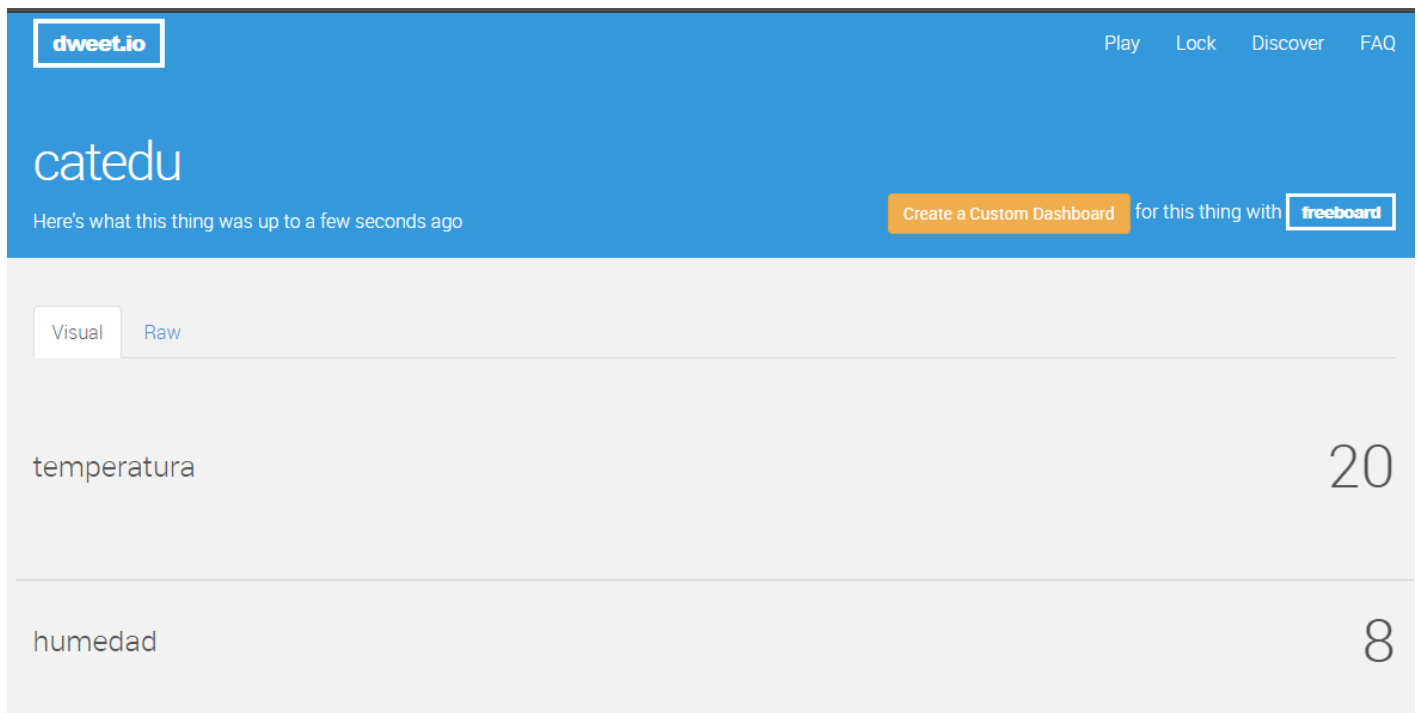
<https://dweet.io/dweet/for/catedu?temperatura=20>

2. Abre otra pestaña del navegador o utiliza un móvil para seguir el dato:

<https://dweet.io/follow/catedu>

3. Prueba añadiendo otra variable, en este caso la humedad:

<https://dweet.io/dweet/for/catedu?temperatura=20&humedad=8>



Automatizamos el proceso de recogida de datos desde Arduino con un programa en Processing, que enviará datos a través del navegador a dweet.io.

IMPORTANTE: No hay que tener abierto el monitor serie del IDE de Arduino porque ocupa el puerto y, por lo tanto, no deja leer los datos a Processing.

REPRESENTACIÓN DE DATOS EN EL NAVEGADOR:

Dweet.io nos ofrecerá los datos de la siguiente manera:

Visual Raw

Humedad

31.7

Temperatura

22.3

dweet.io is brought to you by Bug Labs, Inc. ©2016. All Rights Reserved. [Privacy and Terms](#) hello@dweet.io

Follow @dweet_io



Site by BlackTie.co - Attribution License 3.0

Si queremos algo más vistoso podemos utilizar el servicio freeboard.io aunque en este caso nos tendremos que registrar en la web.

Una vez registrados podemos crear paneles indicadores configurados a nuestro gusto para visualizar la información. Primero habrá que añadir como fuente de datos Dweet.io y nuestro nombre utilizado allí (jorgeroden en el ejemplo).

Después creamos un panel indicando que la fuente de datos que queremos utilizar y la variable en cuestión a visualizar.

freeboard

DATASOURCES

Name	Last Updated		
jorgeroden	21:20:37	↺	🗑

+ ADD PANE

WIDGET

TYPE

Gauge

▼

TITLE

Temperatura

VALUE

`datasources["jorgeroden"]["Temperatura"]`

+ DATASOURCE

📄 JS EDITOR

UNITS

°C

MINIMUM

0

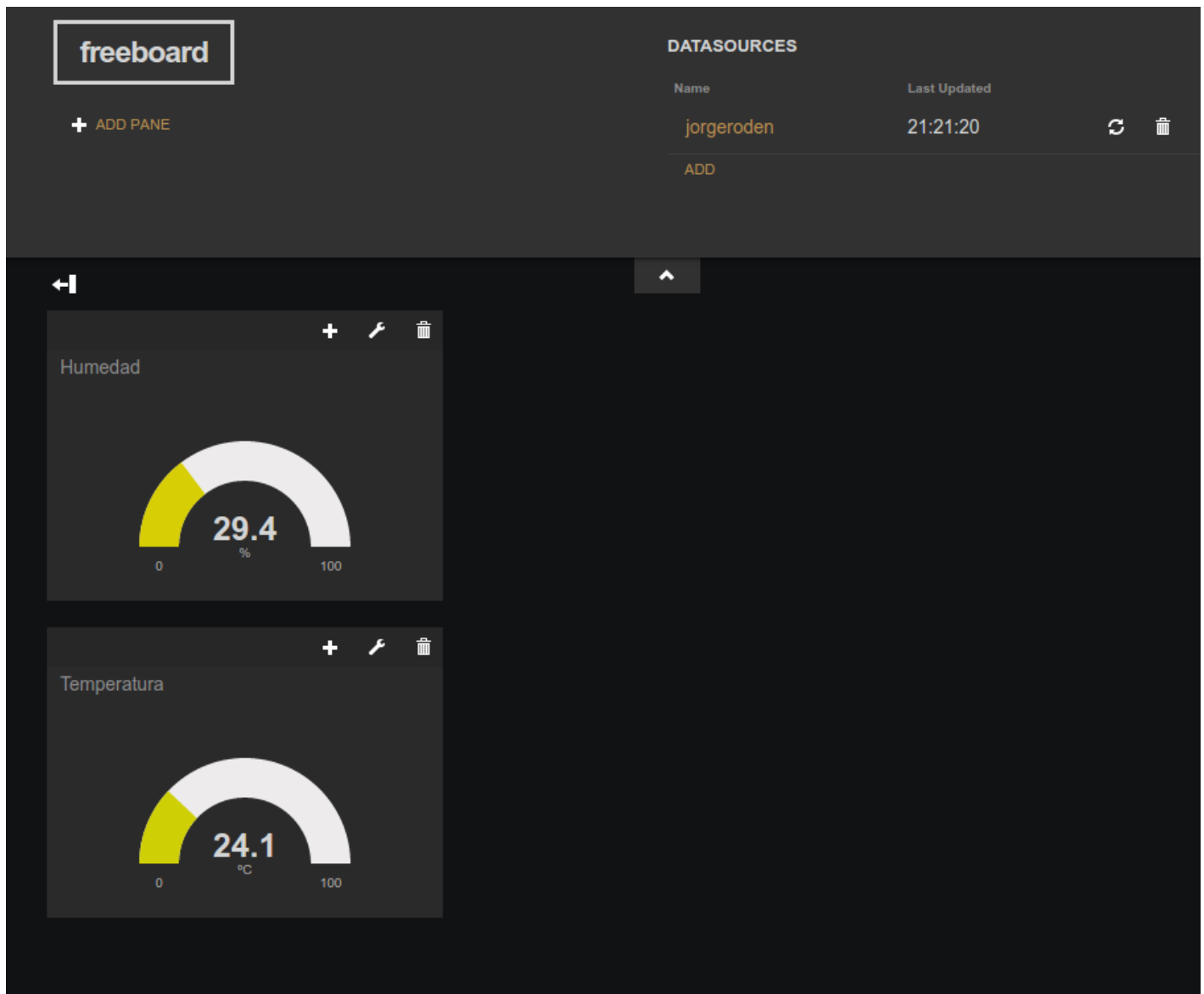
MAXIMUM

100

[SAVE]

CANCEL

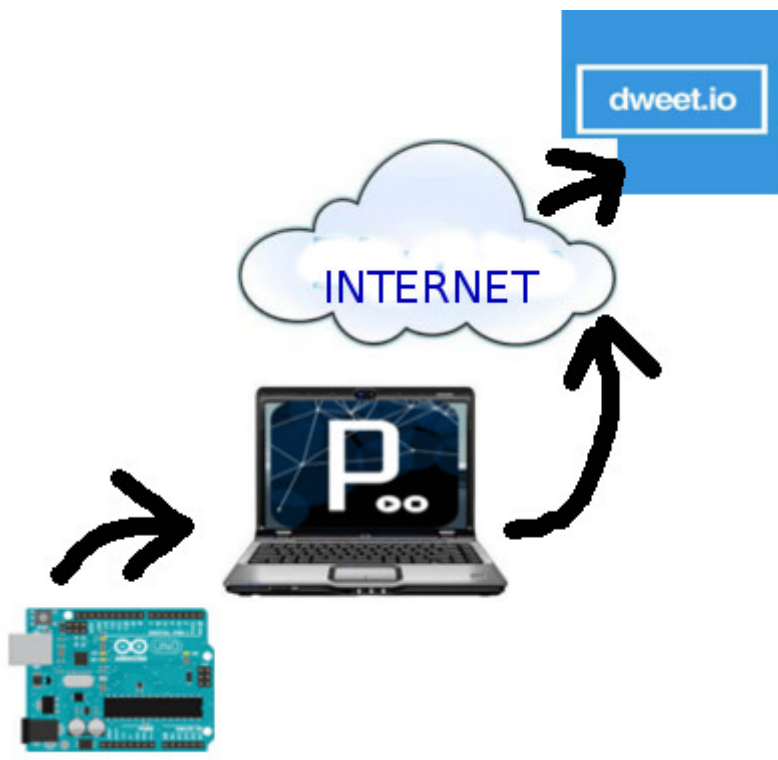
¡Y resultado puede ser de este tipo!



Montaje 19 Visualización de datos en la nube. Internet de las cosas.

Consejo: Si quieres una visualización más sencilla de los datos que recoge un Arduino y llevarlas a la nube, te recomendamos el curso ESP32 EN EL AULA

<https://libros.catedu.es/books/esp32-en-el-aula> el apartado IoT por ejemplo con Telegram



Monitorizar los datos de temperatura y humedad obtenidos del sensor DHT11 en la nube.

Para ello disponemos del portal web **dweet.io** que nos ofrece un servicio para enviar y representar datos en la nube sin necesidad, ni si quiera, de registrarnos en la plataforma.

Vamos a ver los pasos a seguir:

1. Probamos la plataforma introduciendo un dato, para ello en el navegador tecleamos por ejemplo (cambia **CATEDU** por tu nombre):
<https://dweet.io/dweet/for/CATEDU?temperatura=20>
2. Abre otra pestaña del navegador o utiliza un móvil para seguir el dato:
<https://dweet.io/follow/CATEDU>
3. Prueba añadiendo otra variable, en este caso la humedad:
<https://dweet.io/dweet/for/CATEDU?temperatura=20&humedad=8>

¿Quieres ver las cosas que ahora mismo se están difundiendo por dweet? Mira
<https://dweet.io/see>

Automatizamos el proceso de recogida de datos desde Arduino con un programa en Processing, que enviará datos a través del navegador a dweet.io.

IMPORTANTE: No hay que tener abierto el monitor serie del IDE de Arduino porque ocupael puerto y, por lo tanto, no deja leer los datos a Processing.

PROGRAMA A CARGAR EN ARDUINO:

Si fuera un DHT12 en vez de un DHT11 poner comentarios a las 4 primeras líneas delante // y quitárselas a las 3 siguientes

<https://create.arduino.cc/editor/javierquintana/8154af00-c514-4936-81fb-b03fad1963f8/preview>

<https://create.arduino.cc/editor/javierquintana/8154af00-c514-4936-81fb-b03fad1963f8/preview?embed>

PROGRAMA A EJECUTAR EN PROCESSING :

```
// El puerto serie
Serial myPort;

void setup() {
  // Lista todos los puertos serie
  printArray(Serial.list());
  // OJO: Elige el puerto donde tengas conectado Arduino.
  // Cambia el "0" de Serial.list()[0] por el orden que
  // tu puerto ocupe en la lista (0, 1, 2,...).
  // Si no lo tienes claro qué puerto ocupa Arduino mira
  // en el IDE Arduino en "Herramientas" mira el puerto que esté seleccionado.
  //Fíjate que tenemos la velocidad del puerto a la misma que pusimos en Arduino
  myPort = new Serial(this, Serial.list()[0], 9600);
}

void draw() {
  while (myPort.available() > 0) {

    String lectura = myPort.readStringUntil(If);
    if (lectura != null) {
      println(lectura);
      //IMPORTANTE! cambia CATEDU por tu nombre
      // visualiza los resultados en esta web https://dweet.io/follow/CATEDU

      loadStrings("https://dweet.io/dweet/for/CATEDU?" + lectura);
```

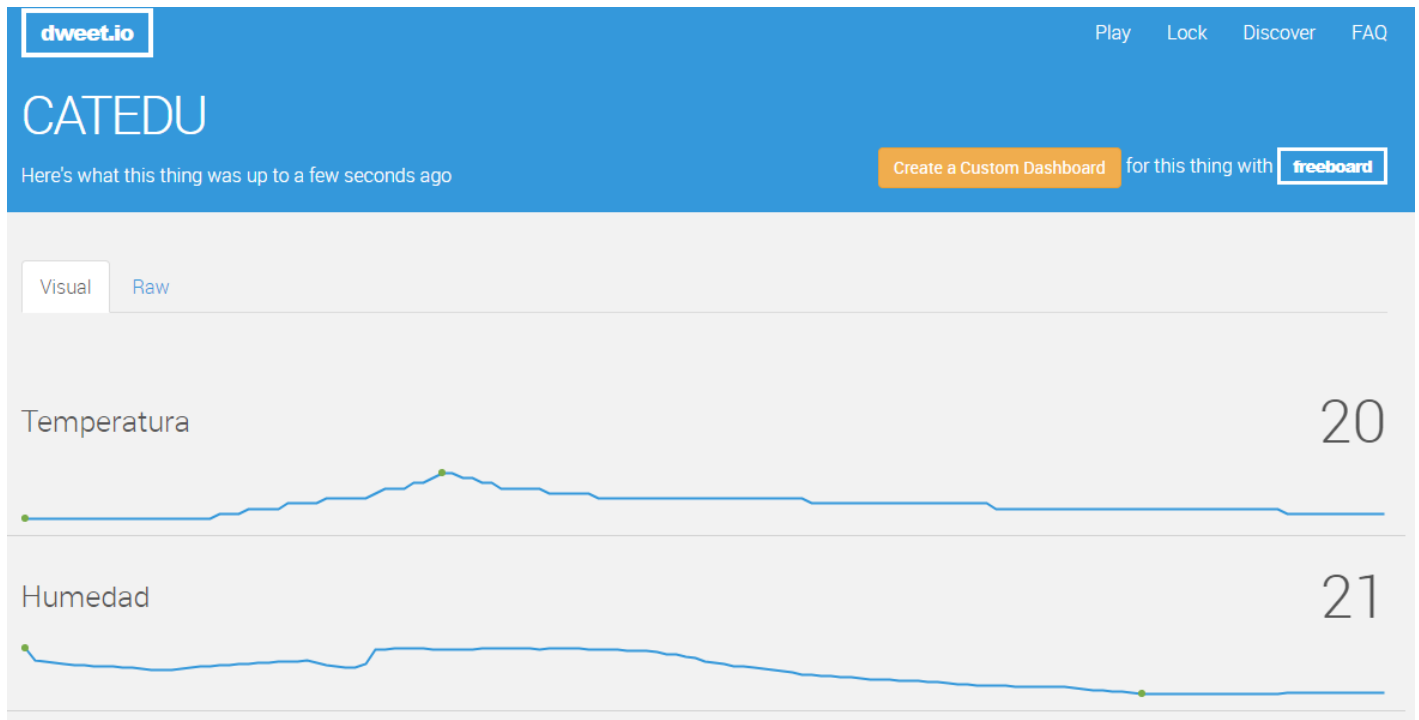
```
}
```

```
}
```

```
}
```

REPRESENTACIÓN DE DATOS EN EL NAVEGADOR:

<https://dweet.io/follow/CATEDU> nos **ofrecería** los datos de la siguiente manera:



No lo hagas, pues NO LO TENGO CONECTADO !! no sale nada !! por eso pone "*nos ofrecería*"

El resultado es espectacular

<https://giphy.com/embed/sjDV6YTbw8tig>

via GIPHY

Revision #13

Created 1 February 2022 11:20:39 by Equipo CATEDU

Updated 23 December 2023 21:35:44 by Javier Quintana