



**UNCUYO**  
UNIVERSIDAD  
NACIONAL DE CUYO



INSTITUTO TECNOLÓGICO  
UNIVERSITARIO

**PROYECTO:**

# **SENSOR DE TEMPERATURA Y HUMEDAD EN CULTIVOS**

**Integrantes: Cruz Florencia - Hertlein Hernan - Baez Paula**

**23/06/2023**

**Institución: Instituto Tecnológico Universitario**



# ÍNDICE

<b>PROPÓSITO:</b>	<b>3</b>
<b>INTRODUCCIÓN:</b>	<b>3</b>
<b>MATERIALES:</b>	<b>4</b>
<b>PROPUESTA DEL PROYECTO:</b>	<b>5</b>
<b>ANÁLISIS DE FACTIBILIDAD:</b>	<b>5</b>
<b>ETAPAS DE DESARROLLO:</b>	<b>6</b>
<b>CÓDIGO:</b>	<b>19</b>
<b>CONCLUSIONES:</b>	<b>19</b>
<b>REFERENCIAS:</b>	<b>20</b>



## PROPÓSITO:

El propósito del proyecto es implementar un sensor de humedad en cosechas para monitorear y controlar de manera eficiente el riego de las plantas. Se busca mejorar la productividad y la salud de las cosechas al proporcionar la cantidad adecuada de agua en el momento adecuado.

Para lograr esto, se utilizarán sensores de humedad del suelo que se colocarán estratégicamente en diferentes áreas del cultivo. Estos sensores medirán continuamente la humedad del suelo y obtendrán los datos recopilados a un microcontrolador Arduino.

El microcontrolador estará programado para analizar los datos recibidos y tomar decisiones basadas en ellos. Utilizando algoritmos específicos, determinará cuándo es necesario reparar las plantas y activar los sistemas de riego correspondientes.

Los resultados esperados del proyecto incluyen una gestión más precisa y eficiente del riego, evitando tanto el exceso como la falta de agua en las plantas. Esto puede ayudar a minimizar los costos de operación y mejorar la salud general de las cosechas. Además, al mantener niveles óptimos de humedad, se espera obtener una mayor productividad y calidad de los cultivos.

## INTRODUCCIÓN:

La problemática a la que queremos hacer frente, ya que nos informamos y vemos que es de gran importancia solucionar, es sobre la humedad en los cultivos ya que radica en la necesidad de mantener un nivel óptimo de humedad en el suelo para garantizar un crecimiento saludable de las plantas. Tanto el exceso como la falta de agua pueden tener consecuencias negativas en el rendimiento y la calidad de los cultivos, por lo que es crucial implementar estrategias de monitoreo y control de la humedad para optimizar el riego y maximizar la productividad agrícola.

Un sensor de temperatura y humedad creado con un microcontrolador ESP8266 es un dispositivo que combina la funcionalidad de medición de la temperatura y la humedad relativa del ambiente utilizando sensores especializados, y la capacidad de procesamiento y transmisión de datos proporcionada por el microcontrolador ESP8266.

El objetivo principal de este proyecto será prevenir la pérdida de los cultivos en el área vitivinícola de Mendoza utilizando la conexión de dos sensores de temperatura y humedad creado con microcontroladores ESP8266, midiendo y registrando la temperatura y humedad relativa del ambiente en tiempo real.

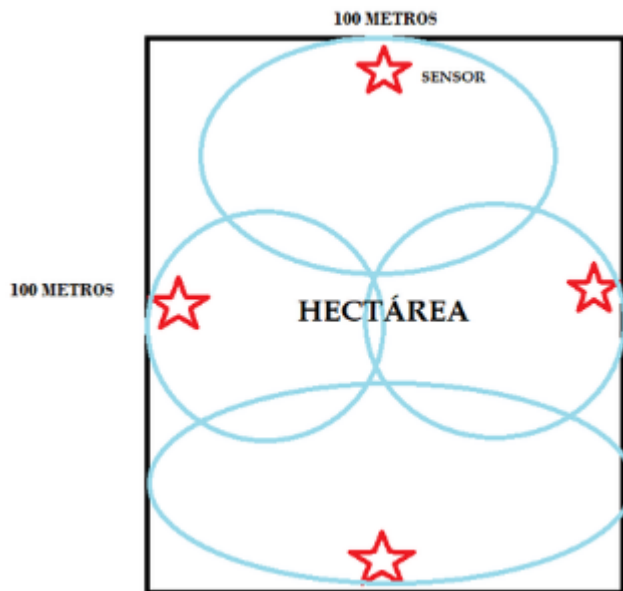
## **MATERIALES:**

- **Material 1: ESP8266** es un módulo WiFi de bajo costo y de bajo consumo de energía. Es ampliamente utilizado en proyectos de IoT (Internet de las cosas) debido a su capacidad para conectarse a redes inalámbricas y enviar/recibir datos a través de Internet. Puede actuar como un microcontrolador independiente o puede ser utilizado en conjunto con un Arduino para agregar funcionalidad WiFi.
- **Material 2: Paquete de cables Jumpers** Los cables Jumpers son cables flexibles que se utilizan para hacer conexiones entre los diferentes componentes electrónicos, que facilita la conexión y desconexión rápida de los componentes sin necesidad de soldadura.
- **Material 3: Sensor de Temperatura y Humedad DHT11** El sensor de temperatura y humedad DHT11 es un sensor digital que permite medir tanto la temperatura ambiental como el nivel de humedad relativa. Proporciona datos precisos y es fácil de usar. Es ampliamente utilizado en proyectos relacionados con el monitoreo ambiental, control de clima y aplicaciones de automatización del hogar.
- **Material 4: Protoboard:** El protoboard (placa de pruebas) es una placa con una matriz de orificios conectados eléctricamente entre sí. Se utiliza para crear circuitos temporales y realizar conexiones entre los componentes electrónicos sin la necesidad de soldarlos. Es muy útil para prototipar y probar circuitos antes de realizar una implementación permanente.
- **Material 5: Arduino Uno** El Arduino Uno es una placa de desarrollo. Es uno de los modelos más populares de Arduino y ofrece una amplia gama de pines de entrada/salida que se pueden utilizar para conectar y controlar diferentes componentes electrónicos. El Arduino Uno es programable a través de un entorno de desarrollo integrado (IDE) y es ampliamente utilizado en proyectos de electrónica y robótica.



## Propuesta del Proyecto:

Instalación de los sensores: Tenemos una hectárea de 100 metros x 100 metros, los sensores tienen un alcance de 30 metros. Se colocan en base a los puntos cardinales De esta forma tendremos toda el área cubierta.



## Análisis de Factibilidad:

**Técnico:** La tecnología de sensores para medir estos parámetros ambientales ya está desarrollada y disponible en el mercado. Los sensores pueden ser instalados en diferentes puntos de la zona agrícola para obtener mediciones precisas y en tiempo real. Además, existen sistemas de monitoreo y control que permiten recolectar y analizar los datos obtenidos de los sensores de manera eficiente. Se realiza con componentes a disposición y no es estrictamente necesario el trabajo de un programador.

**Económico:** El análisis económico debe considerar los costos de adquisición e instalación de los sensores, así como los costos de implementación de un sistema de monitoreo y control. Además, es importante evaluar los beneficios económicos que se obtienen a partir de la aplicación de estos sensores. Esto incluye la optimización del riego y fertilización, la detección temprana de condiciones desfavorables para los cultivos y la reducción de pérdidas por condiciones

ambientales adversas ahorra en pérdidas de cultivos y utiliza microcontroladores de bajo costo

**Social:** Desde el punto de vista social, la aplicación de sensores de temperatura y humedad en la agricultura puede tener varios impactos positivos. Estos sensores permiten mejorar la productividad y la calidad de los cultivos, lo que puede llevar a un aumento en la producción de alimentos. Además, al optimizar el riego y la fertilización, se pueden reducir los impactos negativos en los recursos hídricos y disminuir el uso de productos químicos, lo que contribuye a la sostenibilidad ambiental. Asimismo, la implementación de tecnologías avanzadas en la agricultura puede generar empleo y promover el desarrollo de habilidades técnicas en la comunidad agrícola.

**Ambiental:** Al monitorear y controlar estos parámetros, es posible optimizar el uso del agua y reducir el consumo de energía al ajustar el riego de manera precisa. Esto puede contribuir a la conservación de los recursos naturales, especialmente en áreas donde el agua es escasa. Además, al detectar condiciones ambientales desfavorables, como temperaturas o alta humedad, se pueden tomar medidas preventivas para proteger los cultivos y minimizar las pérdidas. En general, la implementación de sensores de temperatura y humedad en la agricultura puede ayudar a mejorar la eficiencia ambiental y la sostenibilidad del sector agrícola.

## **Etapas de desarrollo:**

### **1) Descripción**

El sensor ESP8266 es un impresionante y económico módulo de Wifi especialmente para conectarlo a un Microcontrolador y agregar comunicación a nuestro proyecto. El módulo puede ser programado para actuar como un sistema aislado.

Ya que la transmisión de datos puede llegar a consumir una corriente considerable se recomienda utilizar una fuente externa de 3.3V en lugar de conectarlo al pin del Arduino. Sin embargo para pruebas se puede utilizar este pin por periodos cortos y evitar sobrecalentar el Arduino.

Finalmente la hoja de datos del fabricante comenta que los pines de comunicación del ESP8266 (Rx y Tx) también operan a 3.3V por lo que existe el riesgo de que

tengamos problemas comunicándonos con los pines Rx y Tx del Arduino, ya que operan a 5V.

## **2)Características**

- Voltaje de operación: 3.3V
- Corriente de máxima: 240mA
- Corriente de operación: 70mA

## **3)Componentes**

- Arduino Uno
- Módulo Wifi ESP8266
- Paquete de cables Jumpers
- Sensor de Temperatura y Humedad DHT11
- Protoboard

## **4)Descripción del Proyecto**

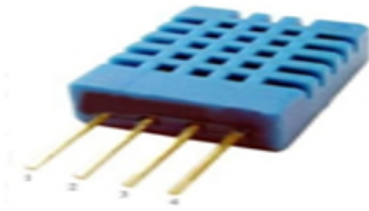
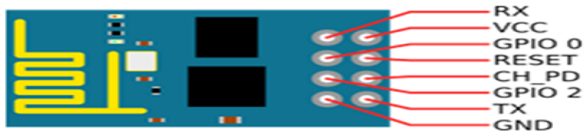
El proyecto está diseñado para transmitir y registrar la temperatura y la humedad inalámbricamente hacia un servidor (ThingSpeak).

La problemática a la que queremos hacer frente, ya que nos informamos y vemos que es de gran importancia solucionar, es sobre la humedad en los cultivos ya que radica en la necesidad de mantener un nivel óptimo de humedad en el suelo para garantizar un crecimiento saludable de las plantas. Tanto el exceso como la falta de agua pueden tener consecuencias negativas en el rendimiento y la calidad de los cultivos, por lo que es crucial implementar estrategias de monitoreo y control de la humedad para optimizar el riego y maximizar la productividad agrícola.

Un sensor de temperatura y humedad creado con un microcontrolador ESP8266 es un dispositivo que combina la funcionalidad de medición de la temperatura y la humedad relativa del ambiente utilizando sensores especializados, y la capacidad de procesamiento y transmisión de datos proporcionada por el microcontrolador ESP8266.

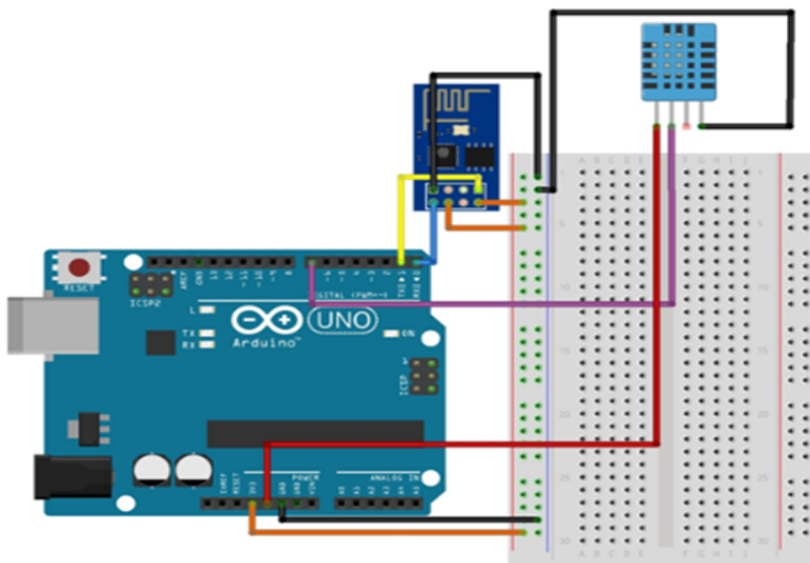
El objetivo principal de este proyecto será prevenir la pérdida de los cultivos en el área vitivinícola de Mendoza utilizando la conexión de dos sensores de temperatura y humedad creado con microcontroladores ESP8266, midiendo y registrando la temperatura y humedad relativa del ambiente en tiempo real.

## **5)Pin-Out**



Pin	Name	Description
1	VDD	Power supply 3 – 5.5 V DC
2	DATA	Serial data output
3	NC	Not connected
4	GND	Ground

## 6) Hardware



## 7) Código





```
1  "DHT.h";
2  SSID "REDWIFI" //Nombre de la Red Wifi a la que nos conectamos
3  PASS "PASSWORD" //Contraseña de la Red Wifi
4  IP "184.106.153.149" // thingspeak.com
5
6  //Sustituir [WRITE_API_KEY] por la obtenida en thingspeak.com sin []
7  String GET_TEMP = "GET /update?key=[WRITE_API_KEY]&field1="; //Campo de temperatura
8  String GET_HUM = "GET /update?key=[WRITE_API_KEY]&field2="; //Campo de humedad
9
10 DHTPIN 7 //Pin al cual se conecta el Sensor DHT11
11 DHTTYPE DHT11 //Tipo del Sensor
12 DHT dht(DHTPIN, DHTTYPE); //Declaracion del objeto DHT
13
14 void setup()
15 {
16
17   pinMode(13,OUTPUT);
18
19   dht.begin(); //Se inicializa sensor DHT
20   Serial.begin(115200); //Se inicializa el puerto Serial para comunicarse con el Modulo Wifi ESP8266
21   Serial.setTimeout(3000); //Set el Timeout en 3 segundos para Serial.find()
22   Serial.println("AT\r"); //PROBAR CON \n\r
23
24   delay(2000);
25
26   if(Serial.find("OK")){ //Verificar si Modulo Wifi responde
```

```
26   if(Serial.find("OK")){ //Verificar si Modulo Wifi responde
27     if(!connectWifi()){ //Conectarse a la red
28       Serial.println("NO SE PUDO CONECTAR A RED WIFI");
29     }
30   } else {
31     Serial.println("MODULO WIFI NO RESPONDE");
32   }
33 }
34
35 void loop(){
36   float h = dht.readHumidity(); // Lee la humedad
37   float t = dht.readTemperature(); //Lee la temperatura
38
39   char bufferTemp[8]; //Crear buffer temporal para convertir float a string
40   String temperatura = dtostrf(t, 8, 3, bufferTemp); //Convertir de float a String 8 = longitud de String, 3 = numero de decimales. (e.g. 1234.567)
41   updateTemp(temperatura); //Actualizar temperatura
42
43   digitalWrite(13,HIGH);
44   delay(20000); //Thingspeak nos pide un MINIMO de 15 segundos entre cada actualizacion
45   //Si se hace mas pronto no se captura el Dato. Esperar 20 para estar seguros
46
47   char bufferHum[8]; //Crear buffer temporal para convertir float a string
48   String humedad = dtostrf(h, 8, 2, bufferHum); //Convertir de float a string 8 = longitud de String, 3 = numero de decimales. (e.g. 1234.567)
49   updateHumedad(humedad); //Actualizar humedad
50   digitalWrite(13,LOW);
51   delay(20000); //Hacer una medicion cada 20 segundos
```



```
51 delay(20000); //Hacer una medicion cada 20 segundos
52 }
53
54 void updateTemp(String temp){
55
56 String cmd = "AT+CIPSTART=\"TCP\", \""; //Crear el comando para comenzar una conexion TCP
57 cmd += IP; //IP del sitio a conectarse (thingspeak.com)
58 cmd += "\",80"; //Puerto (80)
59 Serial.println(cmd); //Crear conexion
60 delay(2000); //Darle 2 segundos para responder
61
62 if(Serial.find("ERROR")){
63 return; //No se pudo conectar
64 }
65
66 temp.trim(); //Remover espacios e.g. "____30.00" &gt;&gt; "30.00"
67
68 cmd = GET_TEMP; //Crear datos a enviar
69 cmd += temp; //Agregar la temperatura leida
70 cmd += "\r\n"; //Agregar caracteres de newline y Carriage return
71
72 Serial.print("AT+CIPSEND="); //Indicar cuantos datos se enviaron por TCP
73 Serial.println(cmd.length()); //Tamaño de los datos a enviar por TCP
74
75 delay(2000);
76
```

```
75 delay(2000);
76
77 if(Serial.find("&gt;")){
78 Serial.print(cmd); //Enviar datos
79 }else{
80 Serial.println("AT+CIPCLOSE"); //No se pudo ejecutar, cerrar conexion TCP
81 }
82 }
83
84 void updateHumedad(String humedad){
85 String cmd = "AT+CIPSTART=\"TCP\", \"";
86 cmd += IP;
87 cmd += "\",80";
88 Serial.println(cmd);
89 delay(2000);
90
91 if(Serial.find("ERROR")){
92 return;
93 }
94
95 humedad.trim();
96
97 cmd = GET_HUM;
98 cmd += humedad;
99 cmd += "\r\n";
100 Serial.print("AT+CIPSEND=");
```

```

107 }else{
108   Serial.println("AT+CIPCLOSE");
109 }
110 }
111
112 //Funcion para crear conexion Wifi. Regresa true si se conecta exitosamente, si no false.
113 boolean connectWifi(){
114
115   Serial.println("AT+CWMODE=3"); //Configurar Modulo Wifi ESP8266 en Modo 3 = Station + Access Point (Trasmitir y Recibir)
116   delay(2000); //Darle tiempo para ejecutar comando
117
118   String cmd="AT+CWJAP=";
119   cmd+=SSID;
120   cmd+="","";
121   cmd+=PASS;
122   cmd+=""";
123
124   //cmd = AT+CWJAP="SSID","PASS"
125
126   Serial.println(cmd);
127   delay(10000); //Darle 10 segundos para conectarse
128
129   if(Serial.find("OK")){
130     return true; //Conexión exitosa
131   }else{
132     return false; //No se pudo conectar
133   }

```

## 8) Explicación de las líneas del código

**"DHT.h";**

**SSID "REDWIFI"** //Nombre de la Red Wifi a la que nos conectamos

**PASS "PASSWORD"** //Contraseña de la Red Wifi

**IP "184.106.153.149"** // thingspeak.com

//Sustituir [WRITE\_API\_KEY] por la obtenida en thingspeak.com sin []

**String GET\_TEMP = "GET /update?key=[WRITE\_API\_KEY]&field1=";** //Campo de temperatura

**String GET\_HUM = "GET /update?key=[WRITE\_API\_KEY]&field2=";** //Campo de humedad

**DHTPIN 7** //Pin al cual se conecta el Sensor DHT11

**DHTTYPE DHT11** //Tipo del Sensor

**DHT dht(DHTPIN, DHTTYPE);** //Declaracion del objeto DHT

**void setup()**

{

**pinMode(13,OUTPUT);**

**dht.begin();** //Se inicializa sensor DHT



```
Serial.begin(115200); //Se inicializa el puerto Serial para comunicarse con el  
Modulo Wifi ESP8266
```

```
Serial.setTimeout(3000); //Set el Timeout en 3 segundos para Serial.find()
```

```
Serial.println("AT\r"); //PROBAR CON \n\r
```

```
delay(2000);
```

```
if(Serial.find("OK")){ //Verificar si Modulo Wifi responde
```

```
if(!connectWiFi()){ //Conectarse a la red
```

```
Serial.println("NO SE PUDO CONECTAR A RED WIFI");
```

```
}
```

```
} else {
```

```
Serial.println("MODULO WIFI NO RESPONDE");
```

```
}
```

```
}
```

```
void loop(){
```

```
float h = dht.readHumidity(); // Lee la humedad
```

```
float t = dht.readTemperature(); //Lee la temperatura
```

```
char bufferTemp[8]; //Crear buffer temporal para convertir float a string
```

```
String temperatura = dtostrf(t, 8, 3, bufferTemp); //Convertir de float a String 8 =  
longitud de String, 3 = numero de decimales. (e.g. 1234.567)
```

```
updateTemp(temperatura); //Actualizar temperatura
```

```
digitalWrite(13,HIGH);
```

```
delay(20000); //Thingspeak nos pide un MINIMO de 15 segundos entre cada  
actualizacion
```

```
//Si se hace mas pronto no se captura el Dato. Esperar 20 para estar seguros
```

```
char bufferHum[8]; //Crear buffer temporal para convertir float a string
```

```
String humedad = dtostrf(h, 8, 2, bufferHum); //Convertir de float a string 8 =  
longitud de String, 3 = numero de decimales. (e.g. 1234.567)
```

```
updateHumedad(humedad); //Actualizar humedad
```

```
digitalWrite(13,LOW);
```

```
delay(20000); //Hacer una medicion cada 20 segundos
```

```
}
```

```
void updateTemp(String temp){
```

```
String cmd = "AT+CIPSTART=\"TCP\",\","; //Crear el comando para comenzar  
una conexion TCP
```

```
cmd += IP; //IP del sitio a conectarse (thingspeak.com)
```

```
cmd += "\",80"; //Puerto (80)
```

```
Serial.println(cmd); //Crear conexion
```

```
delay(2000); //Darle 2 segundos para responder
```

```
if(Serial.find("ERROR")){
```

```
return; //No se pudo conectar
```

```
}
```

```
temp.trim(); //Remover espacios e.g. " ____30.00" &gt;&gt; "30.00"
```

```
cmd = GET_TEMP; //Crear datos a enviar
```

```
cmd += temp; //Agregar la temperatura leida
```

```
cmd += "\r\n"; //Agregar caracteres de newline y Carriage return
```

```
Serial.print("AT+CIPSEND="); //Indicar cuantos datos se enviaron por TCP
```

```
Serial.println(cmd.length()); //Tamaño de los datos a enviar por TCP
```

```
delay(2000);
```

```
if(Serial.find("&gt;")){
```

```
Serial.print(cmd); //Enviar datos
```



```
}else{
```

```
Serial.println("AT+CIPCLOSE"); //No se pudo ejecutar, cerrar conexión TCP
```

```
}
```

```
}
```

```
void updateHumedad(String humedad){
```

```
String cmd = "AT+CIPSTART=\"TCP\",\"";
```

```
cmd += IP;
```

```
cmd += "\",80\"";
```

```
Serial.println(cmd);
```

```
delay(2000);
```

```
if(Serial.find("ERROR")){
```

```
return;
```

```
}
```

```
humedad.trim();
```

```
cmd = GET_HUM;
```

```
cmd += humedad;
```

```
cmd += "\r\n\"";
```

```
Serial.print("AT+CIPSEND=");
```

```
Serial.println(cmd.length());
```

```
delay(2000);
```

```
if(Serial.find("&gt;")){
```

```
Serial.print(cmd);
```

```
}else{
```

```
Serial.println("AT+CIPCLOSE");
```

}

}

//Funcion para crear conexion Wifi. Regresa true si se conecta exitosamente, si no false.

**boolean** connectWiFi(){

**Serial.println**("AT+CWMODE=3"); //Configurar Modulo Wifi ESP8266 en Modo 3 = Station + Access Point (Trasmitir y Recibir)

**delay**(2000); //Darle tiempo para ejecutar comando

**String** cmd="AT+CWJAP="";

cmd+=SSID;

cmd+="\","";

cmd+=PASS;

cmd+="\","";

//cmd = AT+CWJAP="SSID","PASS"

**Serial.println**(cmd);

**delay**(10000); //Darle 10 segundos para conectarse

**if**(**Serial.find**("OK")){

**return true**; //Conexión exitosa

}**else**{

**return false**; //No se pudo conectar

}

}

## 8)Registro en el servidor ThingSpeak

El servidor ThinkSpeak es muy fácil de usar y gratis. Sigue estas instrucciones y empieza a registrar datos:

1. Regístrate y crea una cuenta gratis en <https://thingspeak.com/>

2. Ir a Channelsà Create New Channel (puedes dejar los demás valores como están)
3. Ir a API Keys y obtén tu KEY
4. Prueba poniendo esto en tu navegador:  
[http://api.thingspeak.com/update?key=\[THINGSPEAK\\_KEY\]&field1=0](http://api.thingspeak.com/update?key=[THINGSPEAK_KEY]&field1=0)
5. Checa los resultados en:  
[http://api.thingspeak.com/channels/\[CHANNEL\\_ID\]/feed.json?key=\[THINGSPEAK\\_KEY\]](http://api.thingspeak.com/channels/[CHANNEL_ID]/feed.json?key=[THINGSPEAK_KEY])
6. Reemplaza tu API KEY en el código Arduino y estás listo para enviar datos.

## 9) Explicación de ThingSpeak

ThingSpeak es una plataforma de Análisis de datos para Internet de las Cosas IoT, pero podría analizar cualquier tipo de dato numérico, está en la nube y te permite de una manera muy intuitiva agregar y visualizar datos.

Algunas características:

- Configura fácilmente los dispositivos para enviar datos usando los protocolos populares de IoT.
- Visualiza los datos de tu sensor en tiempo real.
- Datos agregados bajo demanda de fuentes de terceros.
- Utiliza la potencia de MATLAB para darle sentido a tus datos IoT.
- Ejecuta tus análisis de IoT automáticamente en función de los horarios o eventos.
- Prototipado y creación de sistemas IoT sin configurar servidores o desarrollar software web.
- Actúa automáticamente sobre tus datos y se comunica contigo empleando servicios de terceros como Twilio® o Twitter®.

Parte ThingSpeak:

- Primero debes registrarte en la web de ThingSpeak.
- Configurar un canal (*channel*). Esta parte es muy importante, porque define los datos que voy a recibir. Cada canal admite 8 campos de datos, 8 posibles datos que vas a poder subir a través de este canal.  
En este artículo voy a subir datos de latencia y calidad de una conexión wifi, pero podrían ser cualquiera, por ejemplo:
  - Temperatura
  - Humedad
  - Presión barométrica
  - Velocidad del viento
  - Dirección del viento





- Lluvia
- Nivel de luz
- Nivel de batería

Una vez creado el canal, los datos que necesitas para transferir información desde nuestro dispositivo, son el Channel ID y la APIKeys, quédate con estos dos nombres.

#### Parte Dispositivo:

- Wemos D1 mini: es un ESP8266 donde la empresa/marca Lolin/Wemos ha encontrado un filón, ya que ha creado a su alrededor un monton de zócalos (*shields*), donde es posible añadirle funcionalidades sin gastar una gota de estaño, es mi favorito. Desde este dispositivo vamos a enviar datos a la nube de ThingSpeak.
- Los datos:
  - Latencia, ping al DNS de Google que es 8.8.8.8, para medir la latencia que es el tiempo en milisegundos de lo que tarda una señal/paquete en ir al DNS de Google y volver a mi dispositivo.
  - Intensidad de la señal WiFi en dBm.
- Librerías para nuestro Wemos D1 mini ESP8266
  - ESP8266WiFi: vamos a enviar datos a una nube de internet, necesitamos un cliente HTTP.
  - ESP8266Ping: una librería muy sencilla para hacer ping tanto a IP como HOST y que nos devuelve la latencia en milisegundos.
  - ThingSpeak: esta librería está desarrollada por MathWorks que es la empresa que esta detrás de ThingSpeak, en su contenedor de GitHub tienes toda la información. Nosotros añadiremos la librería desde el propio IDE Arduino.

#### Configurar el canal de ThingSpeak:

Vas a crear una cuenta en ThingSpeak y configurar un canal. Rellena los datos, aparte del nombre y descripción del canal, lo que te interesa son los 8 campos de datos.

- *Private View*: muestra la vista privada. Esta vista solo la puedes ver tú. Aquí se verán los datos de latencia e Intensidad que se envíen



desde el Wemos ESP8266 en dos gráficas. Luego veremos cómo configurar estas gráficas.

- **Public View:** si el canal es público, en esta pestaña te muestra cómo se ve desde fuera. Puede ser interesante tener un canal público para compartir con amigos o clientes (en el caso de utilizar una licencia comercial).
- **Channel Settings:** aquí encontrarás la información del canal para poder modificarla. Puedes cambiar el nombre, la descripción o los campos que quieres emplear de este canal.
- **Sharing:** en esta opción puedes seleccionar si el canal es privado (por defecto) o si lo quieres hacer público.
- **API Keys:** algunos servicios requieren de unas claves para enviar y recibir datos. ThingSpeak es uno de ellos. Luego necesitaremos entrar en este apartado para obtener las API Keys de escritura.
- **Data Import/Export:** es la pestaña donde podrás importar o exportar datos de forma masiva.

#### Escribir clave API

Llave

[Generar nueva clave API de escritura](#)

#### Leer claves API

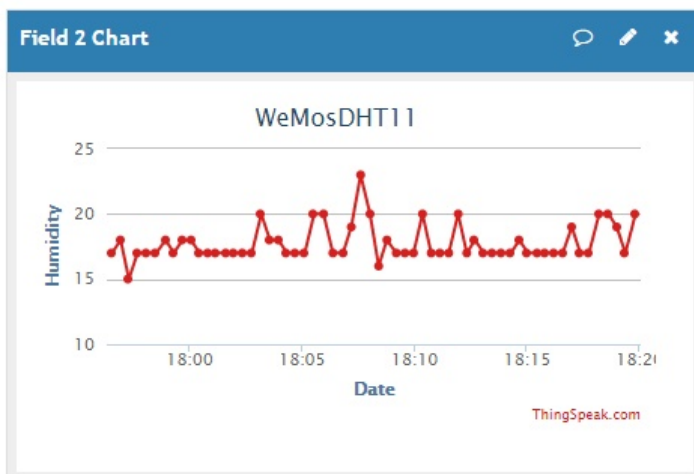
Llave

Nota

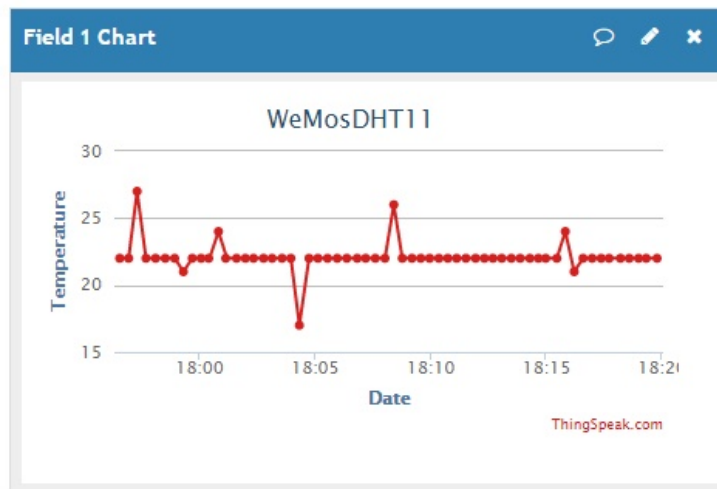
[Guardar nota](#) [Eliminar clave API](#)

#### Salida:

#### Thinkspeak DHT11 Gráfico de Temperatura:



## Thinkspeak DHT11 Grafica de Humedad:



## CÓDIGO:

<https://github.com/hernanhertlein/Sensor-de-Temperatura-y-humedad-en-cultivos>

## Conclusiones:

### Algunas dificultades:

**Costo inicial:** La adquisición e instalación de los sensores y sistemas de monitoreo puede requerir una inversión significativa. Esto puede ser una barrera para algunos agricultores o empresas agrícolas, especialmente aquellos con recursos limitados.

**Infraestructura y conectividad:** Es posible que en algunas zonas agrícolas remotas o con acceso limitado a servicios de internet y electricidad, la instalación y operación de los sensores y sistemas de monitoreo sea más complicada. La falta de infraestructura adecuada puede afectar la eficiencia y la confiabilidad de los sistemas.

**Interpretación y uso de datos:** La recolección de datos de los sensores es solo el primer paso. Es importante contar con personal capacitado que pueda interpretar correctamente los datos y tomar decisiones basadas en ellos. La falta de conocimiento técnico y capacidad de análisis puede limitar el aprovechamiento completo de los beneficios de los sensores.

Investigar opciones de sensores más económicos: Buscar alternativas de sensores de calidad a un costo más accesible puede ayudar a reducir la barrera económica para los agricultores.

#### **Posibles mejoras:**

**Capacitación y asistencia técnica:** Proporcionar capacitación y apoyo técnico a los agricultores para que puedan comprender y utilizar eficazmente los datos recopilados por los sensores.

**Integración con otras tecnologías:** Explorar la integración de los datos de los sensores con otras tecnologías, como sistemas de riego automatizado, para maximizar los beneficios y la eficiencia en la gestión de los cultivos.

**Investigación continua:** Continuar investigando y desarrollando nuevas tecnologías y enfoques para la aplicación de sensores en la agricultura, con el objetivo de mejorar la precisión, reducir costos y abordar desafíos específicos de cada región agrícola.

#### **Conclusión**

El uso de sensores de temperatura en la vid puede ser una herramienta útil para mejorar la producción y la calidad de los viñedos. Al monitorear las temperaturas de las plantas y del suelo, los viñateros pueden obtener información valiosa sobre las condiciones climáticas locales y ajustar sus prácticas de cultivo en consecuencia.

#### **Referencias:**

<https://programarfacil.com/esp8266/thingspeak/>

<https://chips.mecatronicum.com/tutorial-monitoreo-temperatura-y-humedad-por-wifi-esp8266/>

<https://www.prometec.net/wemos-d1-esp8266-wifi/>

<https://www.campusdelvino.com/blog/item/84-factores-ambientales-vinedo>

[https://elpais.com/elpais/2020/06/09/eps/1591716076\\_035335.html](https://elpais.com/elpais/2020/06/09/eps/1591716076_035335.html)