

Primer Parcial IoT

Diego Iván Perea Montealegre (2185751) diego.perea@uao.edu.co
Facultad de Ingeniería, Universidad Autónoma de Occidente
Cali, Valle del Cauca

En las regiones de clima templado la mayor parte de la producción de frutas y hortalizas es estacional, a diferencia de las de clima tropical y subtropical, en donde el período de cultivo es más amplio y la cosecha se distribuye en el tiempo. La demanda, sin embargo, es continua a lo largo del año, por lo que el almacenamiento es el proceso normal para asegurar el aprovisionamiento de los mercados por el mayor tiempo posible. El almacenamiento también puede ser una estrategia para diferir la oferta del producto hasta que el mercado se encuentre desabastecido y de esta manera obtener mejores precios.

El tiempo por el cual un producto puede ser almacenado depende de sus características intrínsecas y como extremos se tienen, por un lado, los muy perecederos como la frambuesa y berries en general, hasta aquellos que naturalmente están adaptados para una larga conservación, como por ejemplo la cebolla, papa, ajo, zapallos, etc. De estas características que les son propias, también dependen las condiciones en las que pueden ser almacenados. Por ejemplo, algunas especies soportan temperaturas cercanas al 0 °C como las hortalizas de hoja y coles en general (Tabla 1), mientras que otras no pueden ser expuestas a menos de 10 °C, como la mayor parte de las frutas de origen tropical.

A menos que sea por muy corto plazo, en donde es posible alojar más de una especie en un mismo ambiente, siempre es conveniente almacenar una sola para poder optimizar las condiciones de almacenamiento específicas de la variedad considerada. El uso del mismo espacio con diferentes productos acarrea problemas de incompatibilidad de temperaturas, humedad relativa, sensibilidad al frío y al etileno, absorción o emisión de olores contaminantes, etc.

Fruta	Tiempo Máximo de Transito y Almacenamiento*	Temperatura °C	Humedad Relativa %
Maracuyá Parchita	21 - 35	12°C	85 - 95
Guanábana, Chirimoya	14 - 28	12°C	85 - 90
Gulupa	21 - 35	12°C	85 - 95
Granadilla	22 - 28	10°C	85 - 90
Tomate de Árbol	28 - 42	0°C	85 - 95
Curuba	21 - 35	12°C	85 - 95
Uchuvas	21 - 42	10°C	65 - 80
Limonas	30 - 180	10°C	85 - 90

* Tiempo en días

Hortaliza	Tiempo Máximo de Transito y Almacenamiento	Temperatura °C	Humedad Relativa %
Zanahoria	120 - 180	0°C	95 - 100
Cebolla	30 - 240	0°C	65 - 75
Coliflor	14 - 28	0°C	95 - 98
Calabaza	60 - 150	10°C	50 - 70
Tomates	14 - 28	12°C	85 - 95
Remolacha	120 - 180	0°C	95 - 100

* Tiempo en días

Tabla1 Condiciones de almacenamiento óptimo para las principales especies de frutas y hortalizas y la vida esperada bajo esas condiciones.

En las siguientes fotografías se pueden observar algunos de productos que por no tener atender las condiciones particulares de humedad relativa y temperatura, terminaron en mal estado.



Figura 1 Producto de frutas que no atendieron las condiciones particulares de humedad relativa y temperatura

Teniendo en cuenta con los objetivos de desarrollo sostenible n° 12 Producción y consumo responsable y n° 2 Hambre cero en donde se debe fortalecer la calidad y seguridad de los alimentos, La problemática identificada fue de tener un mal control de temperatura y humedad relativa en el almacenamiento de los productos de frutas y hortalizas.

Para la solución de esta problemática se dispone a desarrollar una caja inteligente que permita saber la temperatura y humedad relativa en la que está sometida la fruta y hortaliza, Teniendo en cuenta esas características proporcione advertencias al usuario según los rangos que se tenga establecidos en el almacenamiento del producto a seleccionar: fruta o hortaliza. En detalle del funcionamiento es que la caja inteligente tomará los datos de temperatura y humedad y esa información se la enviará a un servidor, que en este mismo lo enviara hacia una página web en donde los usuarios podrán visualizar la información adquirida con un mensaje, que describe la situación de los datos (Tabla2)

El objeto de la aplicación son las frutas y hortalizas, y el objeto será una caja inteligente compuesto por sensores de temperatura y humedad relativa que permita recolectar información de la condición de almacenamiento.

En la aplicación Webb debe entregar los datos analizados de las frutas u hortalizas, en el que darán avisos de alerta según la personalización del usuario.

a) Temperatura (baja, optima, alta)

b) Humedad (baja, optima, alta)

*Se tomará como ejemplo el limón

Temperatura °C	Mensaje	Humedad relativa %	Mensaje
$T < 10$	Temperatura Baja	$H < 85$	Humedad Baja
$10 \leq T \leq 13$	Temperatura Optima	$85 \leq H \leq 90$	Humedad Optima
$T > 13$	Temperatura Alta	$H > 90$	Humedad Alta

Tabla 2 Visualización de mensaje según datos adquiridos

La solución constará de los siguientes componentes:

- Nodo: Encargado de censar la variable y enviarlas al servidor
- Servidor: Encargado de recibir los datos, procesarlos y almacenarlos.
- Aplicación web: Contendrá toda la interfaz visual que le mostrará todos los datos procesados al usuario.

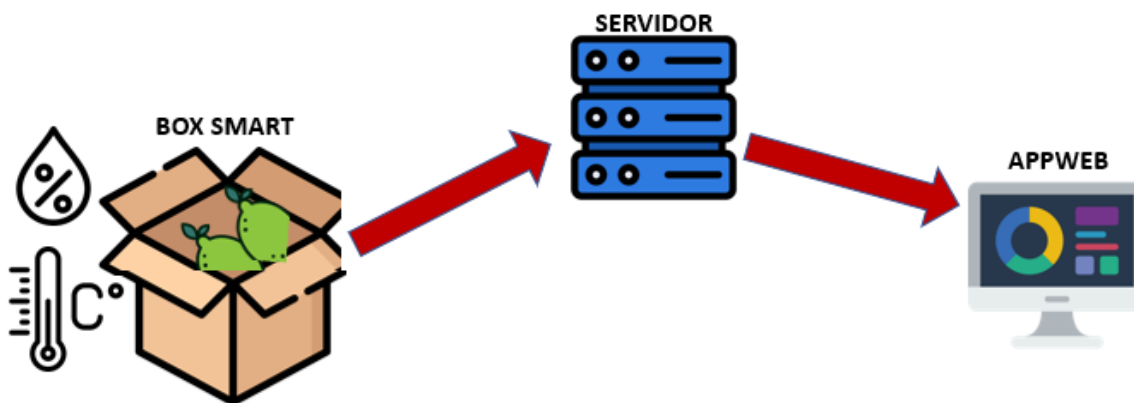


Figura 2 Comportamiento solución IoT

Las variables para capturar son temperatura y humedad relativa en donde las medidas son de Temperatura [°C] y Humedad [%].

Sensores de temperatura

Sensor de temperatura LM35 está calibrado directamente en grados Celsius (centígrados) y no debemos hacer ningún tipo de conversión ni de calibración externa. La salida es analógica y la mediremos con una entrada analógica de Arduino. El gran problema de este sensor es que realmente solo podemos medir temperaturas entre 2° C y 150° C a no ser que utilicemos voltajes negativos. Voltaje de operación: de 4 V a 30 V ,Rango de temperaturas: -55° C a 150° C ,Precisión: $\pm 0,5^\circ$ C , Conversión: 10 mV / °C, Tiempo de respuesta (100%): 4 min. ,Offset: 0 V.Costo:1 euro aprox.

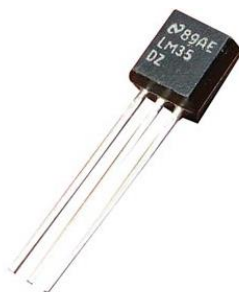


Figura 3. LM35

2



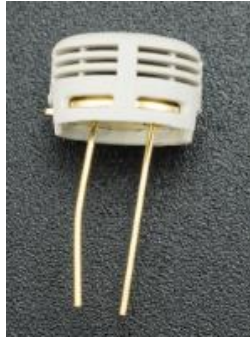


Figura 6. hs1101

Sensor de humedad HR202 es un nuevo tipo de componente sensible a la humedad que utiliza material de polímero orgánico. Tiene un amplio rango de absorción de humedad y un rendimiento estable a largo plazo. Se puede utilizar en almacenamiento, compartimentos, control de calidad del aire interior, automatización de edificios, sistema de control médico e industrial. Y una amplia gama de aplicaciones en el campo de la investigación científica. Parámetros característicos: (a 1kHz) Unidad: ohm ,Voltaje de alimentación: 1,5 V CA (onda sinusoidal máxima) ,Frecuencia de funcionamiento: 500Hz...2kHz ,Potencia nominal: 0,2 mW (onda sinusoidal máxima) ,Valor Central: (at25 °C 1kHz 1V CA 60% onda sinusoidal RH) 31.0kΩ. ,Rango de impedancia: (at25 °C 1kHz 1V CA 60% onda sinusoidal RH) 19,8... 50.2kΩ ,Precisión de detección de humedad: $\pm 5\%$ RH , costo de 3,50 USD aproximadamente

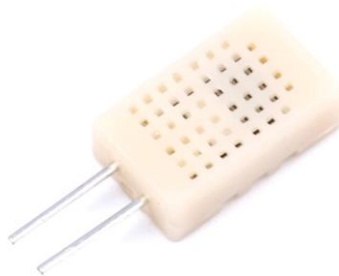


Figura 7. HR202

Sensor de humedad HDS10 revela que el sensor es un componente de cambio positivo, que es insensible a la baja humedad y sensible solo a la alta humedad. Puede funcionar bajo voltaje de CC, con calidad estable y alta fiabilidad. ,Voltaje de alimentación: 0,8 v CC (voltaje de seguridad) ,Rango de temperatura de funcionamiento: 1-80 grados ,Rango de humedad de funcionamiento: 1-100% RH ,Rango de prueba de condensación: 94-100% RH , costo de 3 USD aproximadamente.



Figura 8. HDS10

Sensores de temperatura y humedad

Sensor de temperatura y humedad HDC1080 , sensor digital , El HDC1080 tiene dos modos de funcionamiento: modo dormido y modo midiendo. Cuando el sensor de temperatura y humedad se enciende, automáticamente entra en modo dormido consumiendo una media de 100nA. En este modo, el sensor espera cualquier comando que le llegue por el protocolo I2C para despertar. Cuando recibe un comando para realizar una medida, pasa del modo dormido al modo midiendo. Una vez completada la medida, el sensor de temperatura y humedad regresa al modo dormido. Gracias a que todo esto lo hace de forma automática, se consiguen unos consumos muy bajos. Pero en realidad, el consumo también varía según el voltaje de alimentación y de la temperatura.

Relative humidity accuracy (Typ) (%RH) : ± 2

Temperature accuracy (Typ) ($^{\circ}\text{C}$) : ± 0.2

Relative humidity operating range (Typ) (%RH): 0 - 100

Supply voltage (Min): 2.7

Supply voltage (Max): 5.5

Average supply current (Typ) (uA): 1.2 @ 1 sample/sec

Operating temperature range (C): -40 to 125

Cost : 10 USD aprox



Figura 9. HDC1080

Los sensores DHT11 y DHT22 son los más básicos y los más utilizados para implementarlos con Arduino, estos sensores están compuestos en dos partes, un sensor de humedad capacitivo y un termistor, también constan de un circuito integrado básico en el interior que hace la conversión de analógico a digital y este envía una señal digital con la temperatura y la humedad. El DHT11 es un sensor de temperatura y humedad digital básico y de muy bajo costo. Es capaz de detectar la temperatura y también la humedad relativa, que es la cantidad de vapor de agua en el aire frente al punto de saturación del vapor de agua en el aire. DHT11 es el módulo de temperatura y humedad más común para Arduino y Raspberry Pi. Por lo tanto, ampliamente favorecido por los entusiastas del hardware por sus muchas ventajas.

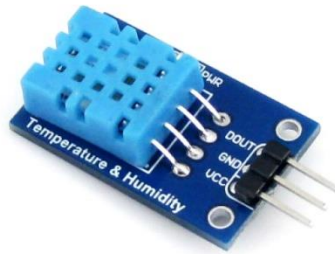


Figura 10 DHT11

El DHT22 El DHT22 también conocido como AM2302 o RHT03, incluye un sensor de humedad capacitivo y un sensor de temperatura de alta precisión. Utiliza tecnología de adquisición de módulo digital dedicada y tecnología de detección de temperatura y humedad para garantizar una alta confiabilidad y una excelente estabilidad a largo plazo. El DHT22 también tiene un elemento de detección capacitivo y un elemento de medición de temperatura de alta precisión conectado a un microcontrolador de 8 bits de alto rendimiento. Por lo tanto, tiene las ventajas de una excelente calidad, una respuesta ultrarrápida, una gran capacidad anti-interferencias y un rendimiento de alto costo.

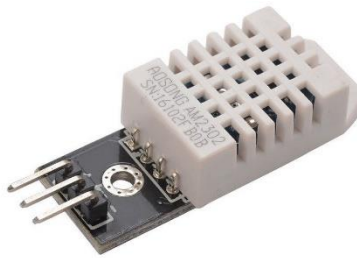


Figura 11 DHT22

Los sensores de temperatura y humedad son muy parecidos , pero ¿ cual es la diferencia entre cada uno? (Observar Tabla 3)

	<i>DHT11</i>	<i>DHT22</i>
Rango de temperatura	-20 a 60°C	-40 a 80°C
<i>Precisión de temperatura</i>	±2%	±0.5%
<i>Rango de humedad</i>	5 a 95% RH	0 a 100% RH
<i>Precisión de humedad</i>	±5%	±2%
<i>Costo</i>	5 USD	11 USD

Tabla 3 Comparativa DHT11 vs DHT22

La selección de sensores de temperatura y humedad es tomada en la disponibilidad de dichos sensores en el laboratorio de préstamo de circuitos de la Universidad Autónoma de Occidente , la disponibilidad de ellos se muestra en la siguiente figura:



Figura 12 Sensores disponibles en el laboratorio de préstamo de circuitos de la Universidad Autónoma de Occidente

Puesto en la comparación de apariencia y verificado la disponibilidad del sensor , el sensor de temperatura y humedad que más se aproximen al comportamiento de los sensores encontrados son el DHT11 y DHT22 , la comparativa de estos dos sensores se puede observar en la tabla 3;Tomando en cuenta y resaltando que el costo del sensor es de gran importancia , y fruta seleccionada es el limón y hortaliza seleccionada es la cebolla , el mejor sensor de temperatura y humedad relativa que cumple con los requisitos y disponibilidad en el laboratorio es el DHT11 .

A continuación, en la figura 13 se muestra código en el dispositivo esp32, que muestra la fecha, hora, temperatura y humedad relativa de adquisición de datos correspondientes.

En el código de la figura 13 , en las líneas 11 al 13 se da las configuraciones para el sensor dht11 en el eps32 , en las líneas 17 al 42 muestra la configuraciones para obtener la fecha y hora , en la cual se tenía que conectar a la red wifi dándole nombre de red y contraseña proporcionados .

Dentro del void loop esta los métodos adecuados para la extracción de la fecha , hora , temperatura y humedad relativa , en donde se guardaban en la siguientes variables : la fecha guardada en la variable String dayStamp, la hora guardada en la variable String timeStamp, la humedad guardada en la variable float h y la temperatura guardada en la variable float t ; Ya con las variables guardadas se procede a realizar la documentación JSON guardado en la variable doc , por lo que el doc iba estar guardado en la variable denominada String variable ;Así permitiendo la visualización de los datos adquiridos .


```

1 #include <Arduino.h>
2 #include <ArduinoJson.h>
3 //LIBRERIAS PARA FECHA Y HORA
4 #include <WiFi.h>
5 #include <NTPClient.h>
6 #include <WiFiUdp.h>
7 //LIBRERIAS PARA DHT11 (TEMPERATURA Y HUMEDAD)
8 #include <Adafruit_Sensor.h>
9 #include <DHT.h>
10
11 //DEFINICION DE PINES DHT11
12 #define DHTPIN 4 // 4 = PIN D4
13 #define DHTTYPE DHT11
14 DHT dht(DHTPIN, DHTTYPE);
15
16 //CONFIG PARA ----FECHA Y HORA-----
17 // Replace with your network credentials
18 const char* ssid = "****you**name**wifi";//name wifi
19 const char* password = "****you**password**wifi";// clave de wifi
20 // Define NTP Client to get time
21 WiFiUDP ntpUDP;
22 NTPClient timeClient(ntpUDP);
23
24 // Variables to save date and time
25 String formattedDate;
26 String dayStamp;
27 String timeStamp;
28
29
30 ;
31
32
33 void setup() {
34 // Initialize Serial Monitor
35 Serial.begin(9600);
36 //CODIGO----FECHA Y HORA-----
37 WiFi.mode(WIFI_STA);
38 WiFi.begin(ssid, password);
39 while (WiFi.status() != WL_CONNECTED) {
40 delay(500);
41 }
42 // Initialize a NTPClient to get time
43 timeClient.begin();
44 // Set offset time in seconds to adjust for your timezone, for example:
45 // COLOMBIA -5 , entonces -5*3600 -> -18000
46 timeClient.setTimeOffset(-18000); //Thailand +7 = 25200
47 }
48
49
50
51
52 void loop() {
53 while(!timeClient.update()) {
54 timeClient.forceUpdate();
55 }
56 // The formattedDate comes with the following format:
57 // 2018-05-28T16:00:13Z
58 // We need to extract date and time
59 formattedDate = timeClient.getFormattedDate();
60 // Extract date
61 int splitT = formattedDate.indexOf("T");
62 dayStamp = formattedDate.substring(0, splitT);
63 //Serial.print("DATE: ");
64 //Serial.println(dayStamp);
65 // Extract time
66 timeStamp = formattedDate.substring(splitT+1, formattedDate.length()-1);
67 //Serial.print("HOUR: ");
68 //Serial.println(timeStamp);
69
70 //CODIGO----TEMPERATURA Y HUMEDAD-----
71 float h = dht.readHumidity();
72 float t = dht.readTemperature();
73
74 //-----
75
76
77 //----CODIGO JSON-----
78 String variable;
79
80 DynamicJsonDocument doc(1024);
81
82
83
84 doc["Fecha"] = dayStamp;
85 doc["Hora"] = timeStamp;
86 doc["Temperatura(°C)"] = t;
87 doc["Humedad(%)"] = h;
88
89
90
91 serializeJson(doc, variable);
92 Serial.println(variable);
93 delay(1000);
94 }

```

Figura 13 Código de ESP32 con JSON de fecha, hora, temperatura y humedad

```

{"Fecha":"2022-08-21","Hora":"13:20:16","Temperatura(°C)":28.5,"Humedad(%)":69}
{"Fecha":"2022-08-21","Hora":"13:20:17","Temperatura(°C)":28.5,"Humedad(%)":69}
{"Fecha":"2022-08-21","Hora":"13:20:18","Temperatura(°C)":28.5,"Humedad(%)":71}
{"Fecha":"2022-08-21","Hora":"13:20:19","Temperatura(°C)":28.5,"Humedad(%)":71}
{"Fecha":"2022-08-21","Hora":"13:20:20","Temperatura(°C)":28.5,"Humedad(%)":72}
{"Fecha":"2022-08-21","Hora":"13:20:21","Temperatura(°C)":28.5,"Humedad(%)":72}
{"Fecha":"2022-08-21","Hora":"13:20:22","Temperatura(°C)":28.89999962,"Humedad(%)":72}
{"Fecha":"2022-08-21","Hora":"13:20:23","Temperatura(°C)":28.89999962,"Humedad(%)":72}
{"Fecha":"2022-08-21","Hora":"13:20:24","Temperatura(°C)":28.89999962,"Humedad(%)":73}
{"Fecha":"2022-08-21","Hora":"13:20:25","Temperatura(°C)":28.89999962,"Humedad(%)":73}

```

Figura 14 Visualización del Serial Monitor ESP32 de datos adquiridos

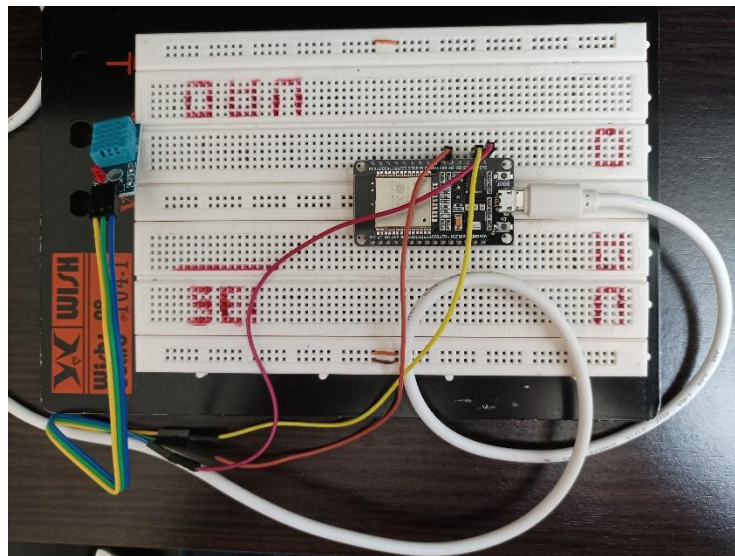


Figura 15 Montaje de ESP32 con sensor DHT11

Ahora se procede a realizar la adquisición de datos pero usando una board diferente en este caso Arduino uno sin un módulo de wifi y sin RTC , en donde no se utiliza una conexión de internet para saber precisamente la hora y fecha , sino que de manera manual se pone la fecha y hora que se debe de iniciar a implementar .

```

{"Fecha":"2022-08-21","Hora":"3:44:15","Temperatura":28,"Humedad":60}
{"Fecha":"2022-08-21","Hora":"3:44:16","Temperatura":28,"Humedad":61}
{"Fecha":"2022-08-21","Hora":"3:44:17","Temperatura":28,"Humedad":61}
{"Fecha":"2022-08-21","Hora":"3:44:17","Temperatura":28,"Humedad":62}
{"Fecha":"2022-08-21","Hora":"3:44:18","Temperatura":28,"Humedad":63}
{"Fecha":"2022-08-21","Hora":"3:44:18","Temperatura":28,"Humedad":65}
{"Fecha":"2022-08-21","Hora":"3:44:19","Temperatura":28,"Humedad":68}
{"Fecha":"2022-08-21","Hora":"3:44:19","Temperatura":29,"Humedad":68}
{"Fecha":"2022-08-21","Hora":"3:44:20","Temperatura":29,"Humedad":71}
{"Fecha":"2022-08-21","Hora":"3:44:20","Temperatura":29,"Humedad":72}
{"Fecha":"2022-08-21","Hora":"3:44:21","Temperatura":29,"Humedad":72}

```

Figura 16 Visualización del Serial Monitor de datos adquiridos en Arduino uno sin modulo wifi

```

1  #include <dht.h>
2  #include <ArduinoJson.h>
3  #include <TimeLib.h>
4
5  #define temperatura A0
6
7  dht DHT;
8  void setup(){
9
10     Serial.begin(9600);
11     setTime(3,44,0,21,8,2022); // setTime(hr,min,sec,day,mnth,yr);
12     delay(1000);
13     analogReference(INTERNAL);
14
15 }
16 String dato(int digit){
17     String cosa=String("0")+digit;
18     return cosa.substring(cosa.length()-2);
19 }
20
21 void loop(){
22
23     DHT.read11(temperatura);
24     //fecha y hora
25     String tiempo = String(hour()) + ":" + dato(minute()) + ":" +dato(second());
26     String fecha = String (year()) + "-" + dato(month()) + "-" +dato(day());
27     //-----
28
29     String variable;
30
31     DynamicJsonDocument doc(1024);
32
33     doc["Fecha"] = fecha;
34     doc["Hora"] = tiempo;
35     doc["Temperatura"] = DHT.temperature;
36     doc["Humedad"] = DHT.humidity;
37
38
39     serializeJson(doc, variable);
40     Serial.println(variable);
41     delay(500);
42 }

```

Figura 17 Código de Arduino uno con JSON de fecha, hora, temperatura y humedad sin modulo wifi

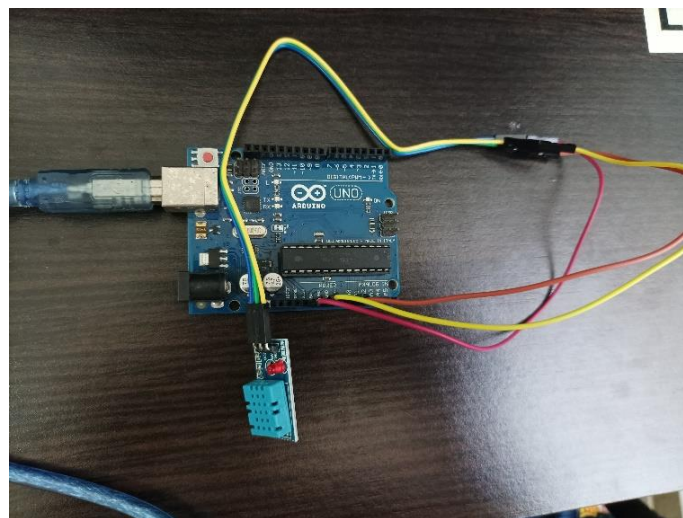


Figura 18 Montaje de Arduino uno con sensor DHT11

Bibliografía

- [1]"Capítulo 3. Almacenamiento", *Fao.org*, 2022. [Online]. Available: <https://www.fao.org/3/y4893s/y4893s06.htm>. [Accessed: 22- Aug- 2022]
- [2]"Capítulo 35: Mejoramiento de la seguridad alimentaria en el hogar", *Fao.org*, 2022. [Online]. Available: <https://www.fao.org/3/w0073s/w0073s13.htm>. [Accessed: 22- Aug- 2022]
- [3]*Procolombia.co*, 2022. [Online]. Available: https://procolombia.co/sites/all/modules/custom/mccann/mccann_ruta_exportadora/files/06-cartilla-cadena-frio.pdf. [Accessed: 22- Aug- 2022]
- [4]"Consumo y producción sostenibles - Desarrollo Sostenible", *Desarrollo Sostenible*, 2022. [Online]. Available: <https://www.un.org/sustainabledevelopment/es/sustainable-consumption-production/>. [Accessed: 22- Aug- 2022]
- [5]"Hambre y seguridad alimentaria - Desarrollo Sostenible", *Desarrollo Sostenible*, 2022. [Online]. Available: <https://www.un.org/sustainabledevelopment/es/hunger/>. [Accessed: 22- Aug- 2022]
- [6]"Sensor de temperatura, escoge el mejor para tus proyectos con Arduino", *Programar fácil con Arduino*, 2022. [Online]. Available: <https://programarfácil.com/podcast/82-escoger-mejor-sensor-temperatura-arduino/>. [Accessed: 22- Aug- 2022]
- [7]"HDC1080 Arduino y ESP8266 sensor de temperatura y humedad", *Programar fácil con Arduino*, 2022. [Online]. Available: <https://programarfácil.com/blog/arduino-blog/hdc1080-arduino-esp8266/>. [Accessed: 22- Aug- 2022]
- [8]"Sensor de Humedad Relativa HS1101", *VISTRONICA S.A.S*, 2022. [Online]. Available: <https://www.vistronica.com/sensores/sensor-de-humedad-relativa-hs1101-detail.html>. [Accessed: 22- Aug- 2022]
- [9]"0.9 £ 24% de DESCUENTO|Higrómetro con Sensor de humedad, resistencia sensible, módulo con funda, HR31 HR31D HR202 HR202L HJ3180B HDS10, 5 piezas|Sensores| - AliExpress", *aliexpress.com*, 2022. [Online]. Available: https://es.aliexpress.com/item/32951082181.html?spm=a2g0o.productlist.0.0.4f462a48CJN2YE&algo_pvid=3bcceb09-5679-4253-86c8-80d8cd0ccbec&algo_exp_id=3bcceb09-5679-4253-86c8-80d8cd0ccbec-16&pdp_ext_f=%7B%22sku_id%22%3A%2212000027223692462%22%7D&pdp_npi=2%40dis%21COP%217438.98%215646.45%21%21%2117970.06%21%21%402101d64d16610413989422613e8d0e%2112000027223692462%21sea&curPageLogUid=p6va0gBBweOo. [Accessed: 22- Aug- 2022]
- [10]*Youtube.com*, 2022. [Online]. Available: https://www.youtube.com/watch?v=SKg_4ggqz2U. [Accessed: 22- Aug- 2022]

[11]*Youtube.com*, 2022. [Online]. Available: https://www.youtube.com/watch?v=mlJxILi_xds. [Accessed: 22- Aug- 2022]

[12]"ESP32: Date and Time (NTP Client)", *Phatiphat Thounthong*, 2022. [Online]. Available: <https://phatiphatt.wordpress.com/esp32-date-and-time-ntp-client/>. [Accessed: 22- Aug- 2022]

[13]"PlatformIO Registry", *PlatformIO Registry*, 2022. [Online]. Available: <https://registry.platformio.org/libraries/paulstoffregen/Time>. [Accessed: 22- Aug- 2022]