Implementación de una estación climática

**Tecnologías Emergentes**



**Integrantes:**Lázaro Yucra, Grover Eduardo

Quesada Tinco, Wilson Alejandro

Rivera Polín, Johnny Eduar

Timoteo Torres, John Fabian

**Docente:**

Arce Zárate, César

**Carrera:**

“Diseño de Software e Integración de Sistemas”  
**Fecha de entrega:**

21/11/2018

**ÍNDICE**

No se encontraron entradas de tabla de contenido.

// Segundo índice

**INTRODUCCIÓN**

Debido a los constantes cambios climáticos generados por el calentamiento global, se ha detectado su efecto negativo en el ambiente, por lo cual se ha decidido desarrollar una estación que registre la actividad climatológica durante el transcurso de un tiempo determinado y que ayude a establecer soluciones a medio y largo plazo. El proyecto a desarrollar a continuación está basado en el uso de distintos sensores que ayuden a la lectura y captura de datos del ambiente para luego ser almacenados en un servidor y luego mostrados y analizados en una aplicación móvil.

Los sensores que se utilizarán en este proyecto son algunos como: sensores de lluvia, sensores de rayos UVA, sensores de temperatura y humedad. Estos sensores son capaces de brindar información de las condiciones climatológicas más habituales y de mayor importancia para las personas o incluso para la industria general.

* 1. **Sensor de UVA (Ultra Violeta)**
     1. **Conceptos teóricos**

(Carita, 2018) UVA es de duración prolongada y penetración profunda, genera radicales libres que dañan las células y puede producir cáncer, además se encuentra incluso en días nublados. UVB no penetra tanto como UVA, pero más del 90% de esta radiación es reflejada por la atmósfera y es la responsable del bronceado y quemaduras.

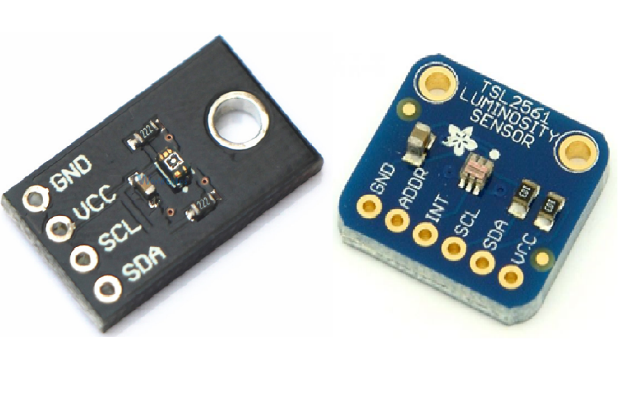


Ilustración Google imágenes Recuperado de: <http://gidahatari.com/ih-es/sensores-de-radiacion-uva>

* + 1. **Diagramas**

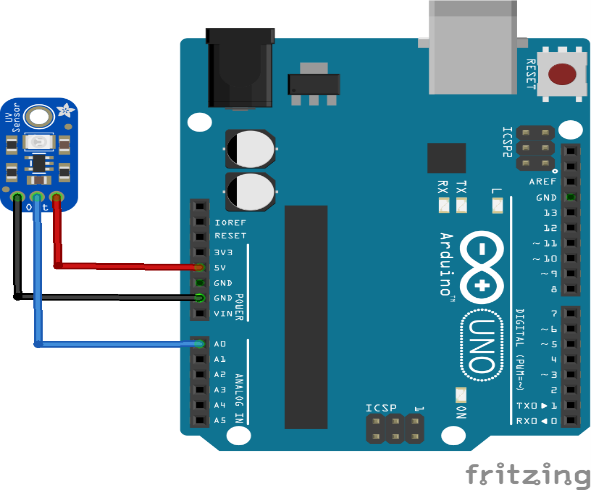


Ilustración conexión sensor UVA - Arduino UNO Recuperado de: https://2.bp.blogspot.com/-H-Wo6amWnxI/WIEXTPp8VfI/AAAAAAAADGM/MTfiafqw0LoglQa9OQqs\_ssL8Jf0LwXmgCLcB/s1600/UV7.png

* + 1. **Requerimientos**
* Arduino UNO.
* Sensor UVA.
* Conocimientos en voltaje para convertir el valor obtenido de acuerdo a una tabla de intensidad.

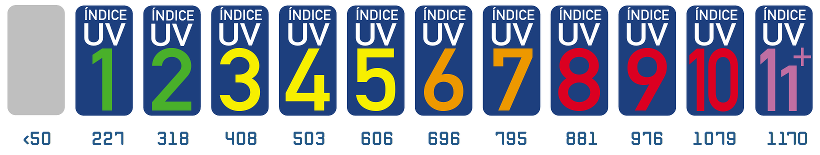


Ilustración Ilustración de la diversa intensidad de radiación de acuerdo al voltaje Recuperado de <https://polaridad.es/sensor-radiacion-ultravioleta-arduino-indice-uv-uvm30a-guva-s12sd/>

* + 1. **Programa**

|  |
| --- |
| void setup()  {   Serial.begin(9600);  }    void loop()  {   float sensorVoltage;   float sensorValue;     sensorValue = analogRead(A0);   sensorVoltage = sensorValue/1024\*3.3;   Serial.print("sensor reading = ");   Serial.print(sensorValue);   Serial.println("");   Serial.print("sensor voltage = ");   Serial.print(sensorVoltage);   Serial.println(" V");   delay(1000);  } |

* 1. **Sensor de Humedad y Temperatura**
     1. **Conceptos teóricos**

(PROMETEC, 2018) […] Los sensores de temperatura de la familia DHT nos proporcionan de forma digital la temperatura y la humedad, con diferente precisión según el modelo.

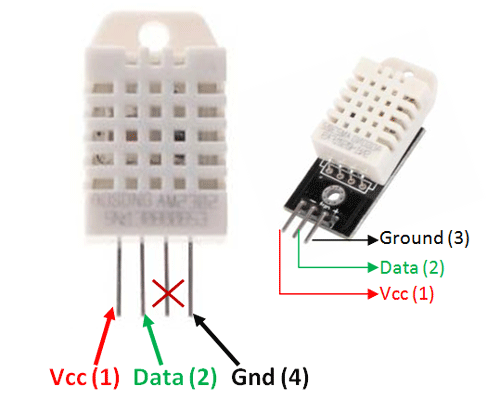


Ilustración sensor DHT22 Recuperado de:: https://components101.com/sites/default/files/component\_pin/DHT22-Sensor-Pinout.png

Básicamente hay dos variantes DHT11 y DHT22; los sensores de la gama DHT, se caracterizan por ser usados en proyectos de bajo costo, ya que su fabricación no supera los 2 euros, además están diseñados para soportar hasta 5V de temperatura. Asimismo, es preciso el uso de una resistencia en modo pull up para evitar dañar el dispositivo.

* + 1. **Diagramas**

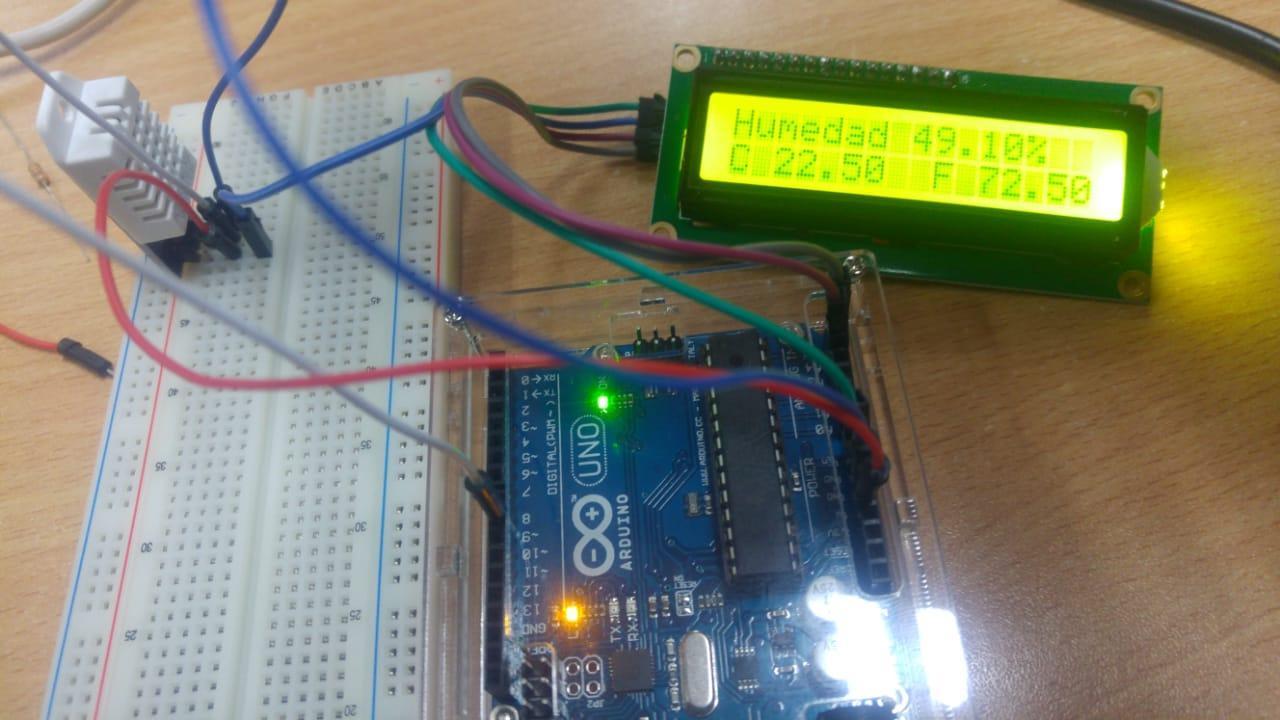
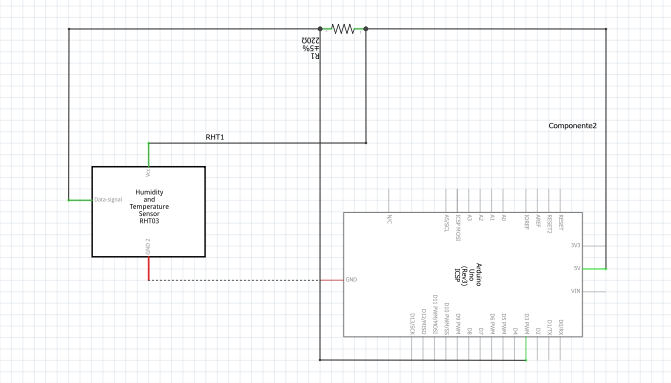


Ilustración Demostración conexión Sensor DTH22 fotografía propia

Diagrama del circuito eléctrico sensor de humedad y temperatura:



|  |  |
| --- | --- |
| Resultado de imagen para sensor dht22  Ilustración 6 Recuperado de: http://electronicazyc.cl/1747-large\_default/sensor-de-temperatura-y-humedad-dht22-am2302.jpg | Ilustración 7 esquema conexión sensor DTH22 Recuperado de: Google imágenes |

* 1. **Sensor de lluvia**
     1. **Concepto teórico**

Este tipo de sensores detectan la presencia de lluvia por la variación de conductividad del sensor al entrar en contacto con el agua. Existen varios modelos de sensores similares, como el FC-37 y el YL-83. Constructivamente son sensores sencillos. Se dispone de dos contactos, unidos a unas pistas conductoras entrelazadas entre sí a una pequeña distancia, sin existir contacto entre ambas. Al depositarse agua sobre la superficie, se pone en contacto eléctrico ambos conductores, lo que puede ser detectado por un sensor.

Ambos modelos se envían con una placa de medición estándar con el comparador LM393, que permite obtener la lectura tanto como un valor analógico como de forma digital cuando se supera un cierto umbral, que se regula a través de un potenciómetro ubicado en la propia placa.



Ilustración Sensor de lluvia Recuperado de: <https://www.luisllamas.es/arduino-lluvia/>

* + 1. **Diagramas**

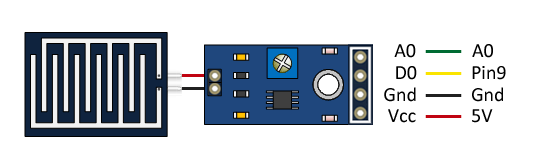


Ilustración Esquema de montaje sensor de lluvia Recuperado de: https://www.luisllamas.es/arduino-lluvia/

|  |  |
| --- | --- |
| Ilustración Pines usados en Arduino para conexión | Ilustración Indicadores de la placa de medición del sensor de lluvia |

* + 1. **Requerimientos**
* Arduino Uno.
* Detector de lluvia.
* Sensor FC-37 o YL-83.

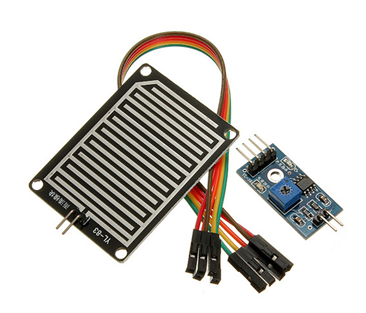


Diagrama del circuito eléctrico sensor de lluvia:

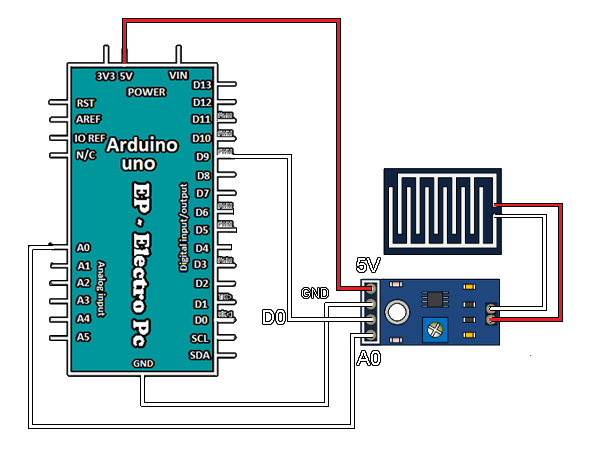


Ilustración Sensor FC37

* + 1. **Programa**

|  |
| --- |
| // lowest and highest sensor readings:  const int sensorMin = 0;     // sensor minimum  const int sensorMax = 1024;  // sensor maximum  void setup() {   // initialize serial communication @ 9600 baud:   Serial.begin(9600);  }  void loop() {   // read the sensor on analog A0:  int sensorReading = analogRead(A0);   // map the sensor range (four options):   // ex: 'long int map(long int, long int, long int, long int, long int)'  int range = map(sensorReading, sensorMin, sensorMax, 0, 3);     // range value:   switch (range) {  case 0:    // Sensor getting wet     Serial.println("Flood");     break;  case 1:    // Sensor getting wet     Serial.println("Rain Warning");     break;  case 2:    // Sensor dry - To shut this up delete the " Serial.println("Not Raining"); " below.     Serial.println("Not Raining");     break;   }   delay(1);  // delay between reads  } |

* 1. **Sensor de ultrasonido**
     1. **Concepto teórico**

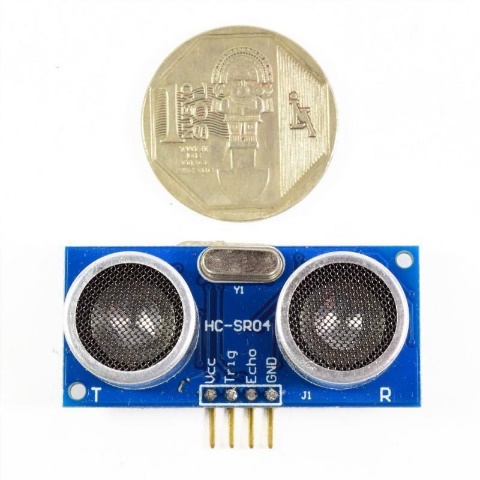
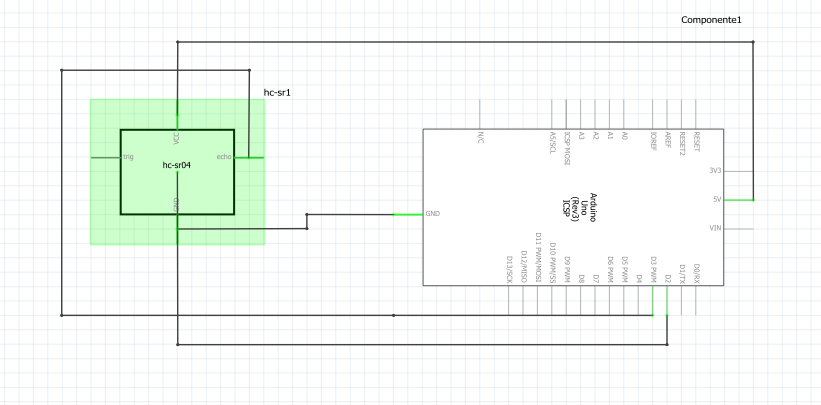


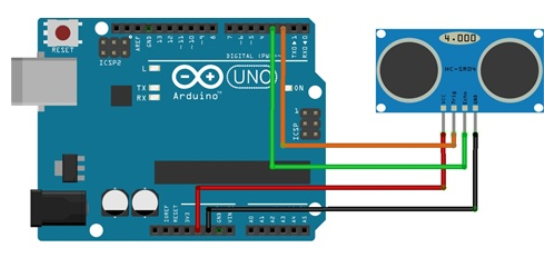
Ilustración Sensor Ultrasonido Recuperado de: https://naylampmechatronics.com/740-large\_default/sensor-ultrasonido-hc-sr04.jpg

(MECHATRONICS, 2018) El sensor HC-SR04 es un sensor de distancia de bajo costo que utiliza ultrasonido para determinar la distancia de un objeto en un rango de 2 a 450 cm. Destaca por su pequeño tamaño, bajo consumo energético, buena precisión y excelente precio. El sensor HC-SR04 es el más utilizado dentro de los sensores de tipo ultrasonido, principalmente por la cantidad de información y proyectos disponibles en la web. De igual forma es el más empleado en proyectos de robótica como robots laberinto o sumo, y en proyectos de automatización como sistemas de medición de nivel o distancia.

* + 1. **Diagramas**

Diagrama del circuito eléctrico sensor de ultrasonido:



Ilustración Diagrama conexión sensor ultrasonido Recuperado de: http://www.naylampmechatronics.com/img/cms/Blog/Tutorial%20arduino%20y%20HC-SR04/Tutorial%20sensor%20ultrasonico%202.jpg

* + 1. **Requerimientos**
* Arduino Uno.
* Sensor ultrasónico HC-SR04.
* Conocimientos concretos de física y matemáticas (velocidad = espacio /tiempo).

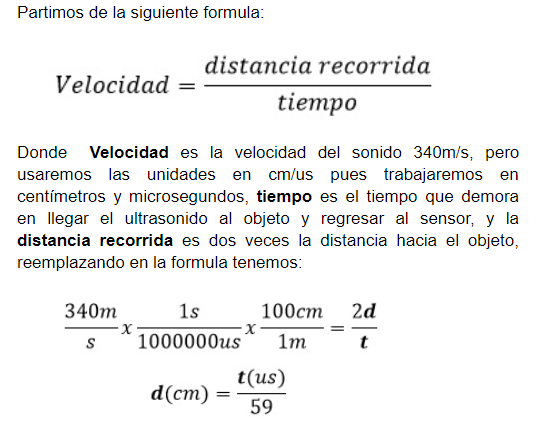


Ilustración conocimientos matemáticos para la programación del sensor ultrasonido Recuperado de: https://naylampmechatronics.com/blog/10\_Tutorial-de-Arduino-y-sensor-ultras%C3%B3nico-HC-S.html

***Fuente:***[*https://naylampmechatronics.com/blog/10\_Tutorial-de-Arduino-y-sensor-ultras%C3%B3nico-HC-S.html*](https://naylampmechatronics.com/blog/10_Tutorial-de-Arduino-y-sensor-ultras%C3%B3nico-HC-S.html)

* + 1. **Programa**

|  |
| --- |
| const int Trigger = 2;   //Pin digital 2 para el Trigger del sensor  const int Echo = 3;   //Pin digital 3 para el Echo del sensor  void **setup**() {  **Serial**.begin(9600);//iniciailzamos la comunicación    pinMode(Trigger, OUTPUT); //pin como salida    pinMode(Echo, INPUT);  //pin como entrada    digitalWrite(Trigger, LOW);//Inicializamos el pin con 0  }  void **loop**()  {    long t; //timepo que demora en llegar el eco    long d; //distancia en centimetros    digitalWrite(Trigger, HIGH);    delayMicroseconds(10);          //Enviamos un pulso de 10us    digitalWrite(Trigger, LOW);      t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso    d = t/59;             //escalamos el tiempo a una distancia en cm    **Serial**.print("Distancia: ");  **Serial**.print(d);      //Enviamos serialmente el valor de la distancia  **Serial**.print("cm");  **Serial**.println();    delay(100);          //Hacemos una pausa de 100ms  } |

* 1. **Módulo Wifi ESP8266-PCB:**
     1. **Concepto teórico**

(Llamas, 2017) El ESP8266 es un microprocesador de bajo coste con Wifi integrado fabricado por Espressif. Podemos usar el ESP8266 para conectar nuestros proyectos de electrónica y robótica con Arduino.

Antes del ESP8266, las opciones disponibles para conectar un Arduino a Wifi (como el Wifi Shield) eran prohibitivamente caras. La aparición del ESP8266 supuso una pequeña revolución al ser el primer dispositivo realmente barato que proporcionaba conectividad WiFi.

En realidad, el ESP8266 es mucho más que un módulo Wifi para Arduino. Es un procesador completo, con mucha más potencia que la mayoría de modelos de Arduino. De hecho es uno de los principales “competidores” a los que se enfrenta Arduino.

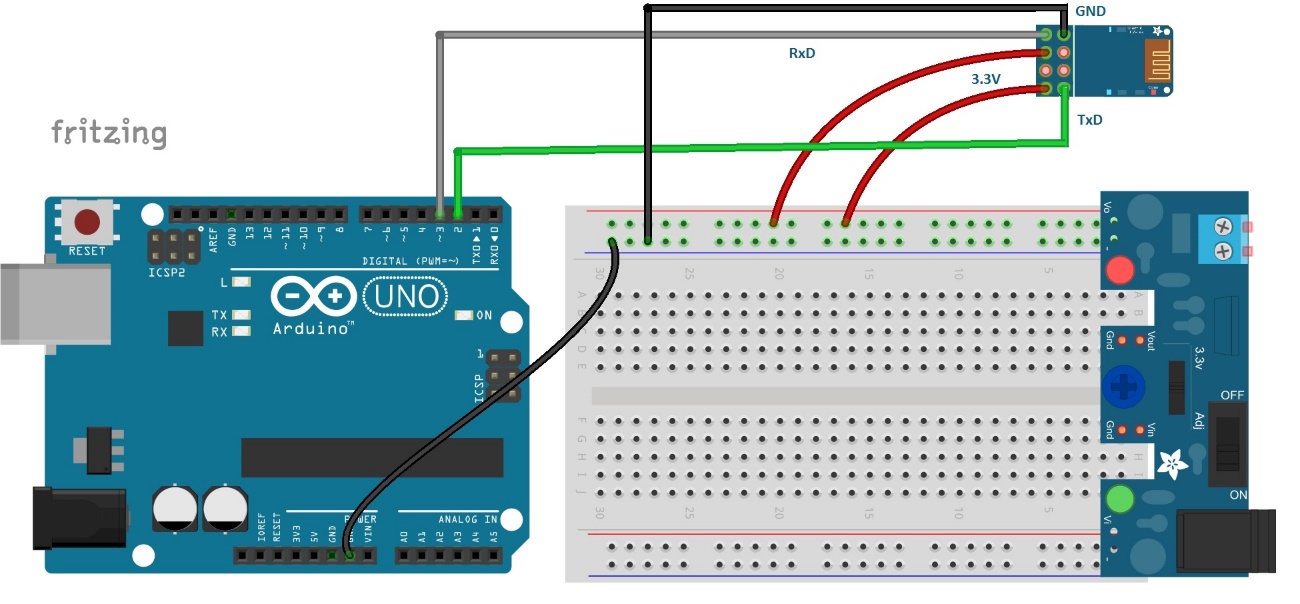
Existen muchos modelos de placas que integran el ESP8266, y una creciente comunidad de usuarios en torno a este procesador. Trataremos extensamente sobre más detalles y modelos con ESP8266 en la sección de tutoriales avanzados.

* + 1. **Diagramas**



Ilustración Sensor WIFI esp8266. [Figura4]. Recuperado de <https://www.prometec.net/wp-content/uploads/2014/12/descarga.jpg>

Diagrama del circuito de modulo wi-fi:



* + 1. **Programa**

|  |
| --- |
| #Codigo para configurar para el módulo wifi en conjunto para el envío de datos emdiante un webserver.  #include <SoftwareSerial.h>  SoftwareSerial SerialESP8266(10,11); // RX, TX  String server = "https://arduino-api-jfabiantimoteotorres.c9users.io";  //variables para enviar al servidor  int variable1=364;  float variable2=3.14;  String cadena="";  void setup() {   SerialESP8266.begin(9600);   Serial.begin(9600);   SerialESP8266.setTimeout(2000);     //Verificamos si el ESP8266 responde   SerialESP8266.println("AT");   if(SerialESP8266.find("OK"))     Serial.println("Respuesta AT correcto");   else     Serial.println("Error en ESP8266");   //-----Configuración de red-------//Podemos comentar si el ESP ya está configurado     //ESP8266 en modo estación (nos conectaremos a una red existente)     SerialESP8266.println("AT+CWMODE=1");     if(SerialESP8266.find("OK"))       Serial.println("ESP8266 en modo Estacion");       //Nos conectamos a una red wifi     SerialESP8266.println("AT+CWJAP=\"internet\",\"\"");     Serial.println("Conectandose a la red ...");     SerialESP8266.setTimeout(10000); //Aumentar si demora la conexion     if(SerialESP8266.find("OK"))       Serial.println("WIFI conectado");     else       Serial.println("Error al conectarse en la red");     SerialESP8266.setTimeout(2000);     //Desabilitamos las conexiones multiples     SerialESP8266.println("AT+CIPMUX=0");     if(SerialESP8266.find("OK"))       Serial.println("Multiconexiones deshabilitadas");     //------fin de configuracion-------------------   delay(1000);    }  void loop() {     //--- programa----------------------     variable1=analogRead(A0);     variable2=3.14;     //otras operaciones     // . . .       //---------enviamos las variables al servidor---------------------         //Nos conectamos con el servidor:         SerialESP8266.println("AT+CIPSTART=\"TCP\",\"" + server + "\",80");       if( SerialESP8266.find("OK"))       {           Serial.println();           Serial.println();           Serial.println();           Serial.println("ESP8266 conectado con el servidor...");             //Armamos el encabezado de la peticion http           String peticionHTTP= "GET /api/data/?a=";           peticionHTTP=peticionHTTP+String(variable1)+"&b="+String(variable2)+" HTTP/1.1\r\n";           peticionHTTP=peticionHTTP+"Host: www.aprende-web.net\r\n\r\n";             //Enviamos el tamaño en caracteres de la peticion http:           SerialESP8266.print("AT+CIPSEND=");           SerialESP8266.println(peticionHTTP.length());           //esperamos a ">" para enviar la petcion  http           if(SerialESP8266.find(">")) // ">" indica que podemos enviar la peticion http           {             Serial.println("Enviando HTTP . . .");             SerialESP8266.println(peticionHTTP);             if( SerialESP8266.find("SEND OK"))             {               Serial.println("Peticion HTTP enviada:");               Serial.println();               Serial.println(peticionHTTP);               Serial.println("Esperando respuesta...");                 boolean fin\_respuesta=false;               long tiempo\_inicio=millis();               cadena="";                 while(fin\_respuesta==false)               {                   while(SerialESP8266.available()>0)                   {                       char c=SerialESP8266.read();                       Serial.write(c);                       cadena.concat(c); //guardamos la respuesta en el string "cadena"                   }                   //finalizamos si la respuesta es mayor a 500 caracteres                   if(cadena.length()>500) //Pueden aumentar si tenen suficiente espacio en la memoria                   {                     Serial.println("La respuesta a excedido el tamaño maximo");                       SerialESP8266.println("AT+CIPCLOSE");                     if( SerialESP8266.find("OK"))                       Serial.println("Conexion finalizada");                     fin\_respuesta=true;                   }                   if((millis()-tiempo\_inicio)>10000) //Finalizamos si ya han transcurrido 10 seg                   {                     Serial.println("Tiempo de espera agotado");                     SerialESP8266.println("AT+CIPCLOSE");                     if( SerialESP8266.find("OK"))                       Serial.println("Conexion finalizada");                     fin\_respuesta=true;                   }                   if(cadena.indexOf("CLOSED")>0) //si recibimos un CLOSED significa que ha finalizado la respuesta                   {                     Serial.println();                     Serial.println("Cadena recibida correctamente, conexion finalizada");                     fin\_respuesta=true;                   }               }                 }             else             {               Serial.println("No se ha podido enviar HTTP.....");            }           }       }       else       {         Serial.println("No se ha podido conectarse con el servidor");       }     //-------------------------------------------------------------------------------   delay(5000); //pausa de 10seg antes de conectarse nuevamente al servidor (opcional)  } |

**CONCLUSIONES**

Colocar las conclusiones obtenidas al completar el trabajo.

* Como resultado de los datos obtenidos se logran hacer estadísticas las cuales a un futuro se podrá tener un control y predecir posibles fenómenos en la región local.
* Se concluye que los datos obtenidos pueden ser de mucha ayuda previniendo por ejemplo a los alumnos de Tecsup, que se protejan de los rayos UVA en temporadas de radiación muy alta.

**RECOMENDACIONES**

Colocar las recomendaciones sobre el trabajo realizado.

* Se recomienda realiza las conexiones con sumo cuidado y con la introducción debida ya que algún mal cálculo puede quemar todo el circuito.
* Tener en cuenta siempre el voltaje y amperaje al que trabajan cada dispositivo ya que un valor fuera de lo recomendable podría estropear el dispositivo, por ejemplo, el módulo wifi trabaja con 3.3 voltios.
* Hacer uso de los protoboard para tener más ordenado las conexiones.

**BIBLIOGRAFÍA**

* Carita, G. (13 de Febrero de 2018). *Gestión sostenible del agua.* Obtenido de Gidahatari: <http://gidahatari.com/ih-es/sensores-de-radiacion-uva-uvb-luz-visible-e-infrarroja-para-raspberry-pi>
* MECHATRONICS, N. (14 de Noviembre de 2018). *Naylmap Mechatronics*. Obtenido de <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>
* PROMETEC. (2018). prometec. Obtenido de <https://www.prometec.net/sensores-dht11/>
* Llamas, L. (27 de mayo de 2017). Luis Llamas. Obtenido de <https://www.luisllamas.es/arduino-wifi-esp8266-esp01/>
* ARDUINO (2017). *Getting Started with Arduino and Genuino UNO* Obtenido de <https://www.arduino.cc/en/Guide/ArduinoUno>

**GLOSARIO**

* **Wifi:** Es originalmente una abreviación de la famosa marca ***Wireless Fidelity*,** que en ingles significa “Fidelidad sin cables o inalámbrica”. En su funcionamiento este depende de un equipo (router o enrutador) el cual tenga conexión a internet y una antena para que así a su vez redistribuya esta señal, los equipos receptores que se encuentren dentro del área de cobertura podrán conectarse a esta y poder tener así acceso a internet.
* **Arduino UNO:** El Arduino UNO es una placa basada en el micro-controlador ATMEL, estos micro-controladores son circuitos integrados en los que se puede grabar instrucciones, estos se escriben bajo el lenguaje de programación C, y su IDE (Entorno de Desarrollo Integrado) es Arduino IDE, estas instrucciones sirven para interactuar con los circuitos que tiene la placa.
* **Sensor UVA:** Sensor UVA es una abreviación de ***Sensor Ultra Violet,*** que en ingles significa “Sensor Ultra Violeta”, este sensor dispone de una carcasa de acero inoxidable que permite usarlo en condiciones ambientales difíciles.
* **Protoboard:** Es un tablero con pequeños orificios que se encuentran conectados eléctricamente entre sí de manera interna, estos siguen un patrón de líneas en los cuales se puedan insertar componentes electrónicos, también se pueden insertar cables para poder así armar un prototipo.

**ANEXOS**

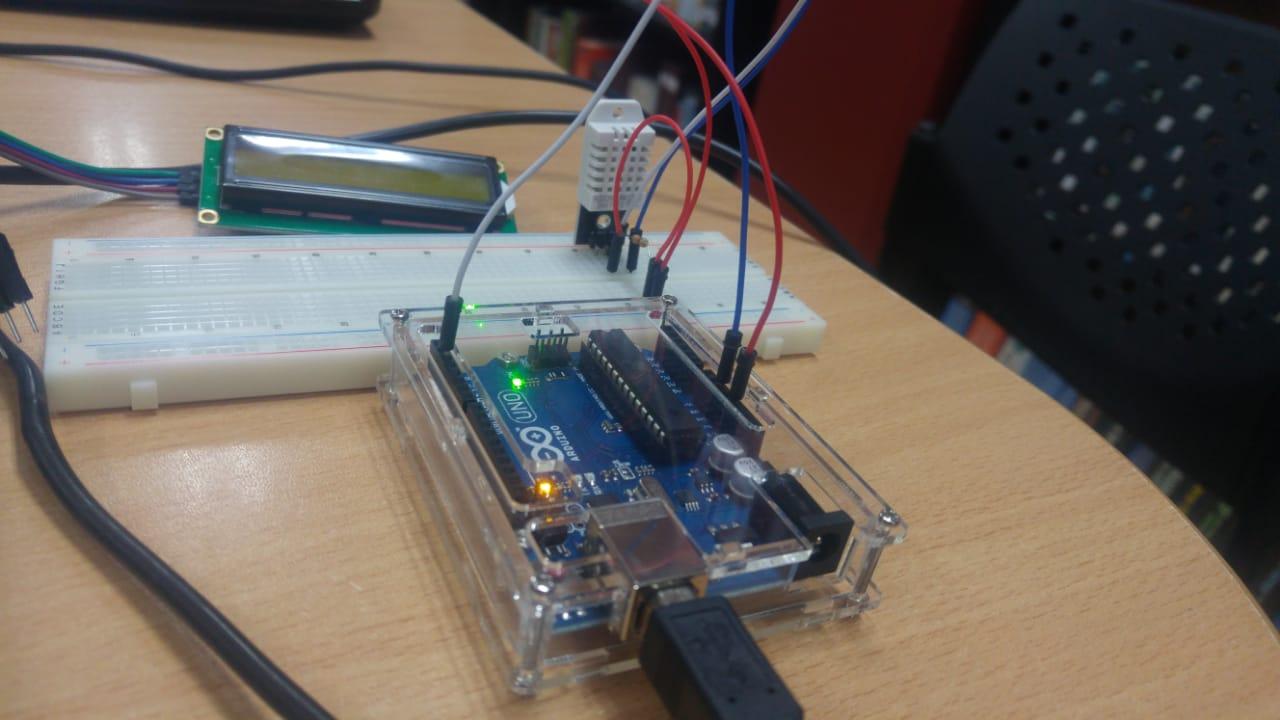


Ilustración Conexión sensor de temperatura Recuperado: fotografía propia

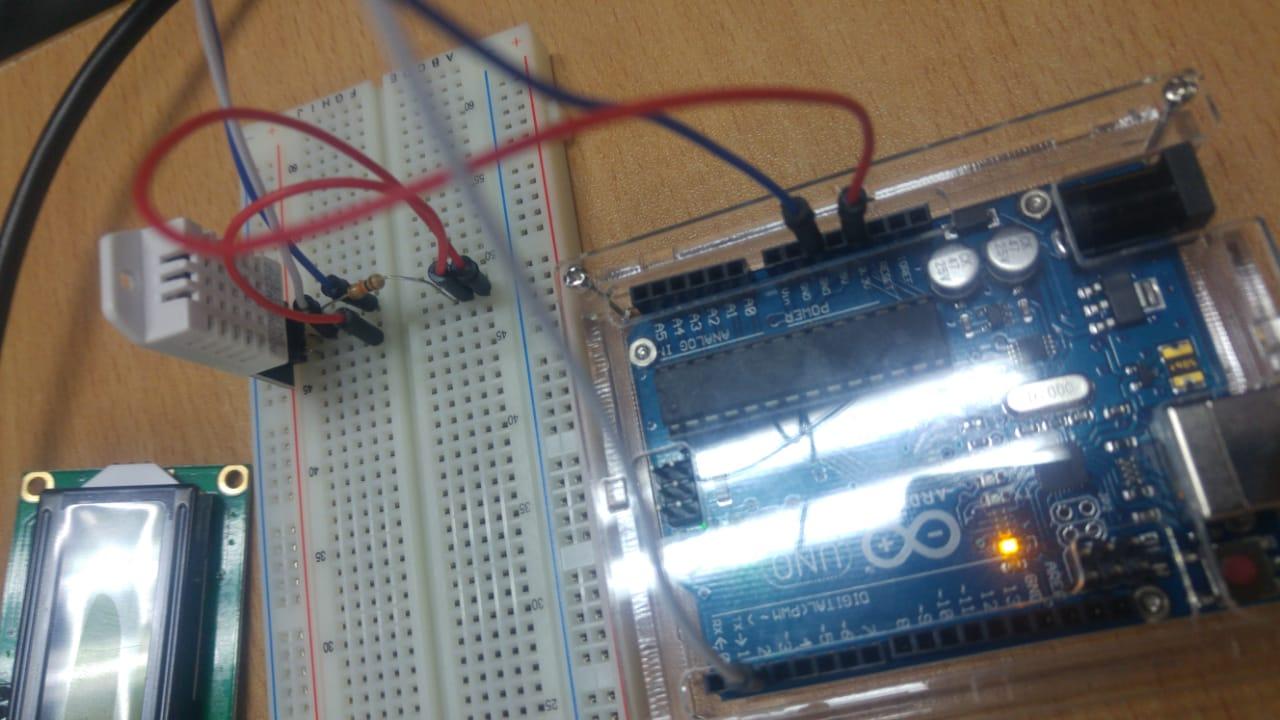


Ilustración Conexión sensor de temperatura Recuperado: fotografía propia



Ilustración Conexión sensor de temperatura Recuperado: fotografía propia