



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería Mecánica Eléctrica

**DISEÑO DE UNA ESTACIÓN METEOROLÓGICA CON CONTROL DE  
ACCIONAMIENTOS ELECTROMECÁNICOS Y MONITOREO DE SENSORES**

**Luis Diego Maldonado Girón**

Asesorado por el Ing. Helmut Federico Chicol Cabrera

Guatemala, junio de 2018

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DE UNA ESTACIÓN METEOROLÓGICA CON CONTROL DE ACCIONAMIENTOS  
ELECTROMECÁNICOS Y MONITOREO DE SENSORES**

TRABAJO DE GRADUACI3N

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**LUIS DIEGO MALDONADO GIR3N**  
ASESORADO POR EL ING HELMUNT FEDERICO  
CHICOL CABRERA

AL CONFERIRSELE EL TÍTULO DE

**INGENIERO ELECTR3NICO**

GUATEMALA, JUNIO DE 2018

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Oscar Humberto Galicia Núñez
VOCAL V	Br. Carlos Enrique Gómez Donis
SECRETARIA	Inga. Lesbia Magalí Herrera López

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADORA	Inga. Ingrid Salomé Rodríguez de Loukota
EXAMINADOR	Ing. Sergio Leonel Gómez Bravo
EXAMINADORA	Inga. María Magdalena Puente Romero
SECRETARIA	Inga. Lesbia Magalí Herrera López

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### **DISEÑO DE UNA ESTACIÓN METEOROLÓGICA CON CONTROL DE ACCIONAMIENTOS ELECTROMECAÓNICOS Y MONITOREO DE SENSORES**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica con fecha 18 de noviembre de 2016



**Luis Diego Maldonado Girón**

Guatemala, 26 febrero de 2018.

Ing. Otto Andrino  
Director Escuela de Ingeniería Mecánica Eléctrica  
Facultad de Ingeniería  
Universidad de San Carlos de Guatemala


Respetable Ingeniero Andrino.

Por medio de la presente informo a usted que, como asesor del Trabajo de Graduación del estudiante universitario **LUIS DIEGO MALDONADO GIRÓN**, quien se identifica con carné universitario No. **2009 – 15396**, procedí a revisar la tesis de cinco capítulos, cuyo título es: **“DISEÑO DE UNA ESTACIÓN METEOROLÓGICA CON CONTROL DE ACCIONAMIENTOS ELECTROMECAÑICOS Y MONITOREO DE SENSORES”**, el cual encuentro satisfactorio.

En tal virtud, **LA DOY POR APROBADA**, solicitándole dar el trámite correspondiente.

Sin otro particular, es grato suscribirme de usted.

Atentamente.

  
Ing. Helmut Federico Chicol Cabrera  
Colegiado Activo No. 7350





REF. EIME 17.2018.

9 DE MARZO 2018.

Señor Director  
Ing. Otto Fernando Andrino González  
Escuela de Ingeniería Mecánica Eléctrica  
Facultad de Ingeniería, USAC.

Señor Director:

Me permito dar aprobación al trabajo de Graduación titulado:  
**DISEÑO DE UNA ESTACIÓN METEOROLÓGICA CON  
CONTROL DE ACCIONAMIENTOS ELECTROMECAÑICOS Y  
MONITOREO DE SENSORES,** del estudiante Luis Diego  
Maldonado Girón, que cumple con los requisitos establecidos  
para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,  
D Y ENSEÑAD A TODOS

Ing. Julio César Solares Peñate  
Coordinador de Electrónica



STO



REF. EIME 17. 2018.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; LUIS DIEGO MALDONADO GIRÓN titulado: DISEÑO DE UNA ESTACIÓN METEOROLÓGICA CON CONTROL DE ACCIONAMIENTOS ELECTROMECÁNICOS Y MONITOREO DE SENSORES, procede a la autorización del mismo.

  
Ing. Otto Fernando Andrino González



GUATEMALA, 4 DE ABRIL 2018.

Universidad de San Carlos  
De Guatemala

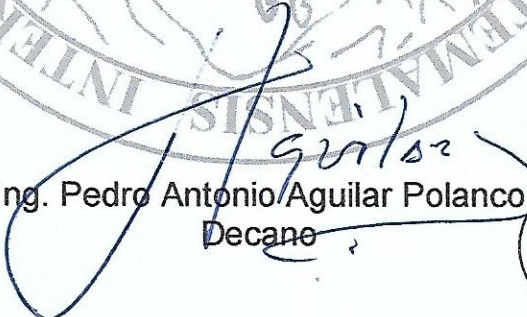


Facultad de Ingeniería  
Decanato

Ref. DTG.D.202.2018

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica al trabajo de graduación titulado: **DISEÑO DE UNA ESTACIÓN METEOROLÓGICA CON CONTROL DE ACCIONAMIENTOS ELECTROMECÁNICOS Y MONITOREO DE SENSORES**, presentado por el estudiante universitario: **Luis Diego Maldonado Girón**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.

  
Ing. Pedro Antonio Aguilar Polanco  
Decano



Guatemala, junio de 2018



## **ACTO QUE DEDICO A:**

<b>Dios</b>	Por levantarme cuando caí en momentos difíciles.
<b>Mis padres</b>	Luis Gustavo Maldonado y María Elena de Maldonado, por ser un ejemplo de superación y unión.
<b>Abuelos maternos</b>	Porque siempre me dieron un empujón extra y fueron inspiración a lo largo de la carrera.
<b>Mis hermanos</b>	Por ayudarme a comprender que las aventuras son mejor cuando se comparten.
<b>Mis amigos</b>	Porque la universidad fue un mejor lugar a su lado, compartiendo momentos.
<b>Mis tíos</b>	Por siempre estar al pendiente a pesar de la distancia
<b>Abuelos paternos</b>	Siempre me inculcaron a no perder de vista mi objetivo y apoyarme a ser mejor.
<b>Seres queridos</b>	Porque siempre preguntaron por mis avances y logros.

## **AGRADECIMIENTOS A:**

<b>Universidad de San Carlos de Guatemala</b>	Por ser mí segunda casa durante la carrera.
<b>Facultad de Ingeniería</b>	Por enseñarme a pensar diferente y tratar de hacer las cosas de la mejor manera.
<b>Mis amigos</b>	Pablo Andrés Echeverría, René Flores, Rodrigo Rivera, Daniel Ávila, José Morales, Liser Rivas, Carlos Quan, Miguel Fonseca, Sergio Contreras, Alejandro Hernández, Alex Palma.
<b>Ing. Helmunt Chicol</b>	Por dedicar su tiempo a la revisión de mi trabajo de graduación.
<b>King Kong</b>	Porque por ti es que decidí ser ingeniero electrónico.
<b>Tío Juanpi</b>	Porque aun estando tan lejos, siempre preguntaste por mí y mis logros.
<b>A ti</b>	Cindy Paola por apoyarme y darme ese empujón extra que necesitaba

## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
LISTA DE SÍMBOLOS .....	IX
GLOSARIO .....	XI
RESUMEN.....	XV
OBJETIVOS.....	XVII
INTRODUCCIÓN .....	XIX
1.    CONCEPTOS GENERALES SOBRE LA METEOROLOGÍA.....	1
1.1.    Meteorología.....	1
1.2.    Estación meteorológica .....	2
1.3.    Paneles solares, turbinas y accionamientos.....	3
1.3.1.    Paneles solares .....	4
1.3.2.    Turbinas eólicas.....	7
1.4.    Parámetros .....	8
1.4.1.    Presión atmosférica .....	8
1.4.1.1.    Factores variables .....	9
1.4.2.    Temperatura atmosférica.....	10
1.4.2.1.    Medición .....	11
1.4.3.    Velocidad y dirección del viento.....	12
1.4.3.1.    Características del viento .....	13
1.4.4.    Humedad del aire .....	14
1.4.4.1.    Clasificación y medición .....	14
1.4.5.    Precipitaciones .....	15
1.4.5.1.    Tipos de precipitaciones .....	16

2.	ARDUINO UNO Y MSP432 .....	17
2.1.	Especificaciones .....	17
2.1.1.	Arduino Uno REV3 .....	17
2.1.2.	MSP 432.....	18
2.2.	Ventajas y desventajas .....	20
2.3.	Entorno de desarrollo integrado .....	21
2.3.1.	Arduino .....	21
2.3.2.	MSP432.....	23
2.4.	Sensores de medición y módulos .....	25
2.4.1.	Sensor de presión BMP180.....	25
2.4.1.1.	Arduino UNO REV3.....	25
2.4.1.2.	MSP432.....	29
2.4.2.	Sensor de humedad DHT11 .....	31
2.4.2.1.	Arduino UNO REV3.....	31
2.4.2.2.	MSP432.....	34
2.4.3.	Sensor de precipitación .....	35
2.4.3.1.	Arduino UNO REV3.....	35
2.4.3.2.	MSP432.....	39
3.	ENVÍO Y RECEPCIÓN DE DATOS .....	41
3.1.	Comunicación <i>Bluetooth</i> . .....	41
3.1.1.	Módulo HC-05 .....	41
3.1.1.1.	Configuración en Arduino y MSP432 ...	44
3.1.1.2.	Ventajas y desventajas.....	46
3.1.2.	Módulo WiFi ESP8266 .....	46
3.1.2.1.	Configuración inicial de la ESP8266 ....	48
3.1.2.2.	Ventajas y desventajas.....	52
3.1.3.	Despliegue de información .....	53

4.	MANEJO DE LOS ACCIONAMIENTOS ELECTROMAGNÉTICOS Y PANELES SOLARES .....	55
4.1.	Implementación de Servo Motores. ....	55
4.1.1.	Servo Motor. ....	55
4.1.1.1.	Ventajas y desventajas .....	60
4.2.	Paneles Solares .....	61
4.2.1.	TP4056 .....	62
4.2.2.	<i>BoostConverter</i> .....	63
4.2.3.	Batería 18650 .....	64
4.3.	Circuito seguidor de luz solar .....	65
4.3.1.	Circuito por usar .....	65
4.3.2.	Funcionamiento .....	66
5.	PRESENTACIÓN DEL PROYECTO .....	69
5.1.	Prototipo de estación meteorológica .....	69
5.1.1.	Pruebas iniciales.....	69
5.1.2.	Organización de los sensores.....	71
5.1.3.	Organización de los servomotores .....	73
5.2.	Comportamiento de los dispositivos .....	74
5.3.	Ventajas y desventajas.....	76
5.3.1.	Ventajas.....	76
5.3.2.	Desventajas .....	77
5.4.	Mejoras a implementar .....	78
5.4.1.	Mejoras a la comunicación .....	78
5.4.1.1.	NodMCU ESP8266MOD .....	78
5.4.1.2.	Mejoras al procesamiento.....	79
	CONCLUSIONES .....	83
	RECOMENDACIONES .....	85

BIBLIOGRAFÍA.....87  
APÉNDICES.....93

# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1.	Estación meteorológica convencional .....	3
2.	Panel solar clásico .....	4
3.	Circuito de paneles solares en paralelo .....	5
4.	Radiación solar en Guatemala .....	6
5.	Esquema básico de un aerogenerador .....	7
6.	Representación de un barómetro .....	8
7.	Anemómetro convencional.....	13
8.	Pluviómetro .....	16
9.	Arduino Uno REV3.....	17
10.	Diagrama de bloques de la MSP 432.....	19
11.	Placa MSP-EXP432P401R .....	20
12.	Software Arduino IDE.....	22
13.	Software Energia MPS432 .....	24
14.	Sensor BMP180 .....	26
15.	Conexión de Arduino y BMP180 .....	26
16.	Configuración inicial .....	28
17.	Gráfica de datos .....	29
18.	Ejemplo de código.....	30
19.	Salida serial del sensor .....	30
20.	Sensor DHT11 con placa .....	31
21.	Sección de muestreo de datos.....	33
22.	Gráfica de los datos tomados.....	33
23.	Código de configuración.....	34

24.	Salida Serial.....	35
25.	Sensor de precipitación .....	36
26.	Acople de sensor y microcontrolador.....	37
27.	Toma de datos de la placa.....	37
28.	Gráfica de datos análogos .....	38
29.	Gráfica de porcentajes.....	38
30.	Configuración puerto serial .....	40
31.	Gráfica de datos.....	40
32.	Módulo HC-05 sin pines.....	42
33.	Módulo HC-05 con pines (parte trasera).....	43
34.	Módulo HC-05 con pines (parte delantera) .....	43
35.	Configuración inicial.....	45
36.	Sección del bucle.....	45
37.	Diagrama ESP8266 .....	48
38.	Configuración Serial.....	48
39.	Configuración ESP8266.....	49
40.	Comandos AT en IDE .....	50
41.	Configuración HTML.....	51
42.	Despliegue de la información.....	54
43.	Entradas de un Servo Motor .....	55
44.	Servo Motor MG996R .....	57
45.	Esquema de conexiones.....	58
46.	Librería para Arduino .....	59
47.	Configuración del servo .....	59
48.	Barrido del servomotor.....	60
49.	Panel Solar 5V .....	62
50.	Módulo TP4056.....	63
51.	Módulo <i>BoostConverter</i> .....	64
52.	Circuito del Seguidor de Luz.....	66



53.	Configuración de los pines .....	67
54.	Lectura Analógica .....	67
55.	Seguimiento solar .....	68
56.	Microcontrolador Arduino UNO R3.....	69
57.	USB con medidor de corriente .....	70
58.	Arduino Nano .....	71
59.	Proceso finito de toma de datos.....	72
60.	Proceso de seguimiento de luz solar.....	73
61.	Gráfico temperatura vs tiempo .....	75
62.	Temperatura de dispositivos .....	75
63.	NodMCU ESP8266MOD .....	79
64.	Arduino DUE .....	80
65.	Arduino MEGA .....	81

## TABLAS

I.	Unidades de conversión de presión atmosférica.....	9
II.	Clasificación según la velocidad del viento .....	12
III.	Especificaciones Arduino UNO REV3.....	18
IV.	Conexión de pines.....	27
V.	Conexión de pines.....	32
VI.	Puertos de conexión.....	39
VII.	Puertos de conexión.....	44
VIII.	Especificaciones .....	44
IX.	Parámetros de la ESP8266.....	47
X.	Comandos AT. ....	50
XI.	Ventajas y desventajas .....	53
XII.	Datos MG996R .....	56
XIII.	Características del módulo TP4056 .....	63

XIV.	Características 1865.....	65
------	---------------------------	----

## LISTA DE SÍMBOLOS

<b>Símbolo</b>	<b>Significado</b>
<b>ATM</b>	Atmosfera
<b>I<sub>T</sub></b>	Corriente total
<b>dBm</b>	Decibelios-mili vatios
<b>GHz</b>	Giga Hertz, frecuencia
<b>C</b>	Grados Celsius
<b>F</b>	Grados Farenheit
<b>K</b>	Grados Kelvin
<b>g/m<sup>3</sup></b>	Gramo por metro cúbico
<b>Kb</b>	Kilo bit
<b>K*mol</b>	Kilo mol, peso molecular
<b>kPA</b>	Kilo Pascal
<b>kWh/m<sup>2</sup></b>	Kilo Watt Hora por metro cuadrado (Energía)
<b>L/m<sup>2</sup></b>	Litro por metro cuadrado
<b>Mbps</b>	Mega bits por segundo
<b>MHz</b>	Mega Hertz, frecuencia
<b>m</b>	Metro
<b>μA</b>	Micro Amperio
<b>mA</b>	Mili Amperio
<b>mΩ</b>	Mili ohmios
<b>mm</b>	Milímetro
<b>Ω</b>	Ohmios
<b>Pa</b>	Pascal
<b>P</b>	Potencia

<b>Bar</b>	Unidad de presión
<b>PSI</b>	Unidad de presión
<b>Torr</b>	Unidad de presión
<b>b/g/n</b>	Versiones de WiFi
<b>Vcc</b>	Voltaje de entrada
<b>V</b>	Voltios

## GLOSARIO

<b>AC</b>	<i>Access Point</i> (punto de acceso).
<b>AES</b>	Encriptación de contraseñas.
<b>Beaufort</b>	Medición empírica de la velocidad del viento.
<b>Bits</b>	Número digital que representa un uno o un cero. Capacidad de almacenamiento.
<b>Boostconverter</b>	Convertidor que permite el aumento de voltaje de entrada.
<b>Char</b>	Tipo de variable utilizada para almacenar caracteres.
<b>DC</b>	Corriente directa.
<b>Energia</b>	Software empleado para el desarrollo de microcontroladores de la empresa Texas Instrument.
<b>Flash</b>	Tipo de memoria de almacenamiento. Proceso de almacenamiento de un set de instrucciones en la memoria.
<b>FTP</b>	Protocolo de transferencia de archivos.

<b><i>Full Duplex</i></b>	Tipo de transmisión de datos en la cual se cuenta con un canal de transmisión y uno de recepción.
<b><i>HIGH</i></b>	Palabra que se utiliza para representar un uno lógico.
<b><i>HTML</i></b>	Lenguaje de marcas de hipertexto.
<b><i>HTTP</i></b>	Protocolo de transferencia de hipertexto.
<b><i>IPv4</i></b>	Protocolo de Internet versión 4 para el etiquetado de de paquetes.
<b><i>LED</i></b>	Diodo emisor de luz.
<b><i>LOW</i></b>	Palabra que se utiliza para representar un cero lógico.
<b><i>MAC</i></b>	Dirección física de una tarjeta de red. Control de acceso de medios.
<b><i>Open Source</i></b>	Software de fuente libre, que es actualizado por una comunidad de desarrolladores.
<b><i>PCB</i></b>	Diseño de un circuito impreso para ser aplicado a una placa especial.
<b><i>SCL</i></b>	Protocolo de comunicación entre registros.
<b><i>SDA</i></b>	Protocolo de comunicación entre registros.

<b>TCP</b>	Protocolo de aplicaciones del modelo OSI.
<b>Texas Instrument</b>	Compañía dedicada a la fabricación de chips para microcontroladores.
<b>TKIP</b>	Encriptación de contraseñas para un dispositivo de red.
<b>UDP</b>	Protocolo de aplicaciones del modelo OSI.
<b><i>Upload</i></b>	Proceso de cargar un set de instrucciones dentro de una memoria de un microcontrolador.
<b>USB</b>	Protocolo de comunicación para la transferencia de archivos.
<b>WEP</b>	Cifrado de contraseñas de baja seguridad.
<b>WPA</b>	Cifrado de contraseñas de media seguridad.
<b>WPA2</b>	Cifrado de contraseñas de alta seguridad.





## RESUMEN

Por medio del uso de equipos de cómputo, como los microcontroladores de la rama Arduino y MSP, se diseñará una estación meteorológica capaz de procesar datos de distintos sensores análogos, que se encargan de realizar mediciones de los distintos fenómenos atmosféricos. Se utilizarán protocolos de comunicación inalámbrica, que permitirá un mayor alcance de la estación para que sea eficiente en su totalidad.

Los datos obtenidos de cada sensor serán mostrados en una página web, que se generará mediante el almacenamiento de instrucciones HTML en la memoria del microcontrolador, que permitirá, la fácil interacción entre la máquina y el usuario final.

Toda la estación meteorológica utilizará energía renovable, por medio de paneles solares se distribuirá el voltaje a los distintos sensores, accionamientos electromecánicos y a los microcontroladores, así permitirá, que el conjunto de dispositivos sea amigable al ambiente.



# OBJETIVOS

## General

Proponer el diseño de una estación meteorológica con control de accionamientos electromecánicos y monitoreo de sensores, para el estudio de las condiciones climáticas que se producen en una región determinada.

## Específicos

1. Analizar los diferentes fenómenos atmosféricos que afectan a una región de interés.
2. Desarrollar un conjunto de instrucciones capaces de procesar la información obtenida por los sensores y enviarla al microcontrolador.
3. Utilizar tecnologías inalámbricas para la comunicación entre dispositivos y sensores para la representación de la información obtenida.
4. Facilitar el uso de una herramienta capaz de representar de manera gráfica el comportamiento de un fenómeno atmosférico dentro de una región.



## INTRODUCCIÓN

A lo largo de la historia del ser humano, se ha tratado de comprender como los efectos del clima pueden alterar la vida cotidiana, y como se puede establecer una predicción que permita comprender como se comportarán los fenómenos atmosféricos en una región que se rige por leyes físicas, que afectan el ambiente, y crean cambios climáticos.

De esto nace la meteorología, ciencia capaz de realizar un estudio específico del comportamiento natural del clima, y que fenómenos se producen cuando suceden las alteraciones ambientales.

Mediante una estación meteorológica se puede establecer una base de datos, con información proveniente de sensores especializados en la medición de distintos parámetros, que componen un estudio climático y la representación de los mismos.

Gracias a esto, el ser humano ha podido comprender que no se está exento a sufrir el impacto de los fenómenos atmosféricos, pero puede mitigar los daños a futuro si se comprende el comportamiento del clima, y como la sociedad plantea un estilo de vida en base a los estudios realizados.



# **1. CONCEPTOS GENERALES SOBRE LA METEOROLOGÍA**

## **1.1. Meteorología**

Se conoce como meteorología, al estudio de los diversos parámetros atmosféricos. El principal objetivo es la recopilación diaria y estacional de los datos necesarios para desarrollar su posterior análisis, para dar una predicción de cómo se comportará el clima, según de la información obtenida en el presente. De la misma manera comprender los fenómenos atmosféricos de una región o muestra, esto durante un periodo de tiempo determinado, permite así el desarrollo de la vida.

Los parámetros de estudio de la meteorología son:

1. Presión atmosférica
2. Temperatura atmosférica
3. Velocidad y dirección del viento
4. Humedad del aire
5. Precipitaciones

Dado a que la meteorología permite establecer conclusiones a futuro de cómo se verá afectado el clima al depender de los parámetros en tiempo presente, se dice que es una ciencia auxiliar de la climatología.

Se pueden evidenciar tres grandes ramas de la meteorología:

- Meteorología aeronáutica
- Meteorología agraria
- Meteorología médica

Existen otras ramas importantes que caben resaltar como tema de interés, entre esas están:

- Meteorología dinámica
- Meteorología sinóptica
- Micro meteorología
- Meteorología tropical

## **1.2. Estación meteorológica**

Es una instalación equipada con dispositivos especializados donde su principal objetivo es medir y registrar los parámetros atmosféricos de una región específica.

Los dispositivos principales por usar para la medición de dichos parámetros son:

- Termómetro: mide la temperatura del aire
- Barómetro: mide la presión atmosférica en la superficie
- Pluviómetro: mide la cantidad de agua que cae sobre el suelo por metro cuadrado en forma de lluvia, llovizna, nieve o granizo.
- Higrómetro: medida de la humedad relativa del aire
- Anemómetro: medida de la velocidad del viento
- Veleta: instrumento que indica la dirección del viento

La función de la estación meteorológica es, recabar datos de forma planificada en relación con los datos que se utilizan para definir predicciones climáticas. Para un funcionamiento óptimo, es necesario que la estación esté automatizada en su mayoría, debido a que una vez obtenida la información necesaria para generar los cálculos y establecer la predicción de los cambios



climáticos, los resultados deben ser procesados de forma automática, sin la intervención humana.

Figura 1. **Estación meteorológica convencional**



Fuente: <http://www.mvara.com/weather.htm> Consulta: 05 de junio de 2017.

### 1.3. Paneles solares, turbinas y accionamientos

La estación meteorológica tiene como objetivo almacenar datos, sacar conclusiones y dar predicciones de cómo será el clima a futuro, sin embargo, se desea obtener la mayor exactitud de los datos. Se busca que sea una estación completamente *ecológica*, que utilice energías renovables, como lo son paneles solares o turbinas eólicas, para el funcionamiento de los dispositivos electrónicos de la estación.

### 1.3.1. Paneles solares

Un panel solar es un dispositivo capaz de capturar la energía de la radiación solar y convertirla en energía térmica o eléctrica, en otras palabras, capta la luz solar y la transforma en energía eléctrica. En el caso de la energía eléctrica, lleva de nombre panel fotovoltaico. Los paneles están conformados por pequeñas celdas fotovoltaicas, las cuales producen energía eléctrica mediante la radiación solar, es decir, la luz que impacta en el dispositivo.

Figura 2. **Panel solar clásico**

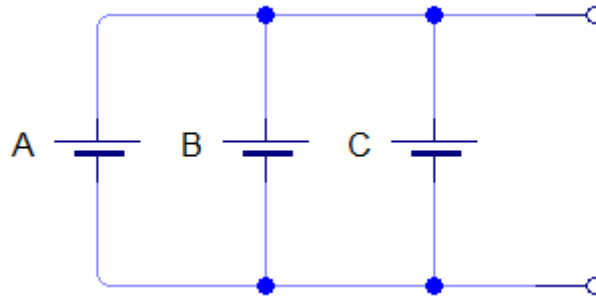


Fuente: <http://www.solarshop.cl>. Consulta: 06 de junio de 2017.

Los paneles alimentarán la estación meteorológica completa, desde los sensores hasta la carga de las baterías, para que el impacto al ambiente sea positivo.

La conexión que se realizará es colocar paneles solares en paralelo para optimizar la capacidad de miliamperios (mA).

Figura 3. **Circuito de paneles solares en paralelo**



Fuente: elaboración propia, empleando Livewire V1.11.

Para esta conexión el amperaje aumentará dependiendo de la cantidad y características de los paneles solares que se conecten. Tomando como ejemplo un panel solar de 6 voltios y 200 miliamperios se cuenta que tiene una corriente total de salida dada por la suma de cada panel solar:

$$I_T = I_1 + I_2 + \dots + I_N$$

Donde  $N$  indica el número total de paneles solares instalados en la conexión en paralelo.

Por lo que si la corriente que genera cada panel es de 200 miliamperios y se conectan tres paneles solares, se obtiene un total estimado de 600 miliamperios, esta es la cantidad total de corriente que el sistema es capaz de entregar.

Para el análisis de la energía generada en un día por el conjunto de paneles solares, se calcula mediante la siguiente ecuación:

$$E_T = I_T * HSP * V$$

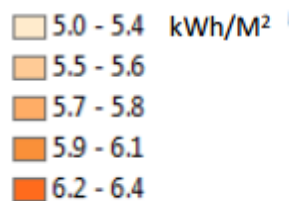
Dónde:

- $E_T$  es la energía total generada en kWh/m<sup>2</sup>
- $P$  es la potencia que entrega el panel solar en Watts
- HSP es la cantidad de horas solar pico de la región por día
- $V$  es el voltaje de salida ideal del sistema

Para ejemplo práctico se utiliza los datos de una celda solar de un Watt de potencia, seis voltios y un valor HSP de 6,0 kWh/m<sup>2</sup>, obteniendo 36 Wh/día. Multiplicando esto por 30 días da un total de 1 080 Wh/mes.

Figura 4. **Radiación solar en Guatemala**

Radiación solar global horizontal, kWh/m<sup>2</sup>



Fuente: <http://www.mem.gob.gt/>. Consulta: 06 de junio 2017.

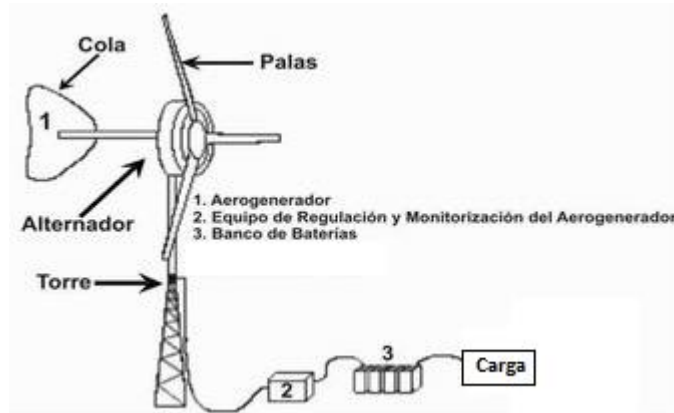
En base a los datos presentados se analiza si el sistema será capaz de suministrar energía suficiente para abastecer todos los sensores conectados en la estación meteorológica. De no ser suficiente se puede incorporar más

paneles solares en paralelo al sistema y así obtener mayor corriente para abastecer la estación.

### 1.3.2. Turbinas eólicas

También conocida como un Aerogenerador, es una turbina que está accionada por medio de la energía eólica, es decir, aprovecha el viento para generar energía eléctrica. Su funcionamiento es bastante básico ya que el viento hace girar un rotor, y convierte la energía mecánica en energía eléctrica y esta última se almacena en baterías para su uso posterior o en algunos casos, se redirige a la red eléctrica.

Figura 5. Esquema básico de un aerogenerador



Fuente: [www.monografias.com](http://www.monografias.com). Consulta: 06 de junio de 2017.

Como los paneles solares, la turbina eólica se plantea como alternativa al consumo de la red eléctrica y se configura como un suministro del mismo. De esta manera la estación contará con tres fuentes de energía; las baterías, los paneles solares y una turbina eólica.

## 1.4. Parámetros

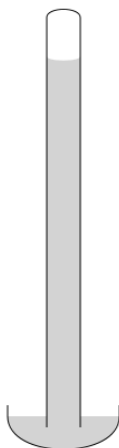
Los parámetros por tratar son los ya establecidos en la sección 1.1 y se detallarán de manera precisa a continuación.

### 1.4.1. Presión atmosférica

La presión atmosférica es la fuerza por unidad de área que ejerce el aire sobre la superficie terrestre. La unidad que la representa es el Pascal (Pa) y a 1 atmosfera, es decir la presión atmosférica normalizada es de 100kPa (100 000 Pa).

La medición de los parámetros es realizada por medio de un barómetro, y consiste en un líquido encerrado en un recipiente o tubo de vidrio. El más conocido es el barómetro de mercurio.

Figura 6. Representación de un barómetro



Fuente: [www.monografias.com](http://www.monografias.com). Consulta: 07 de junio de 2017.

El funcionamiento de un barómetro consiste en la baja o aumento del líquido en su interior, en este caso mercurio indica que, si se encuentra un tiempo despejado, el nivel del mercurio se elevará; caso contrario, decae el nivel de mercurio si se aproxima un mal clima o tormenta. Tomando este concepto, si la presión de aire es elevada, el mercurio busca una salida y tiende a subir por el recipiente. Si la presión del aire disminuye, el mercurio disminuye de la misma manera.

Tabla I. **Unidades de conversión de presión atmosférica**

<b>Unidad</b>	<b>Pa</b>	<b>Bar</b>	<b>Atm</b>	<b>Torr</b>	<b>PSI</b>
1 Pa (N/m <sup>2</sup> )	1	10 <sup>-5</sup>	0,987×10 <sup>-5</sup>	0,0075	0,00014503
1 Bar	10 <sup>5</sup>	1	0,987	750	14,5036
1 Atm	101325	1,01325	1	760	14,69480
1 Torr	133,32	0,0013332	1,32×10 <sup>-3</sup>	1	0,019336
1 PSI	6894,75729	0,068948	0,068046	51,7149	1

Fuente: elaboración propia.

#### **1.4.1.1. Factores variables**

Los factores que influyen en las mediciones de la presión atmosférica son la humedad, altura y temperatura.

- **Altura:** a mayor altura la presión disminuye y a menor altura, la presión aumenta. Al aumentar la altura, el aire soporta menos peso, por lo que se expande y ejerce menor presión.
- **Temperatura:** el aire caliente pesa menos que el aire frío y tiende a elevarse. Con altas temperaturas, el aire se calienta, se hace liviano, y asciende, esto causa baja presión.
- Con bajas temperaturas, el aire tiende a enfriarse, se hace más pesado, desciende y origina alta presión. Aquí se aplica

la regla, “*a mayor altura, menor temperatura; a menor altura, mayor temperatura*”.

- Humedad: En lugares donde hay mayor humedad, hay menor presión y viceversa, si hay menor humedad, mayor presión; esta situación se relaciona directamente con la altura.

#### **1.4.2. Temperatura atmosférica**

Se refiere al grado de calor específico del aire en un tiempo determinado sobre un lugar fijo. La temperatura y la presión se relacionan entre sí, ya que, a mayor temperatura del aire, menor presión, dado que el aire asciende. De lo contrario, a menor temperatura, mayor presión. La unidad de medida de la temperatura son los grados Celsius (°C).

La temperatura se puede dividir en tres secciones para su mejor comprensión:

- Temperatura máxima: Es la mayor temperatura que alcanza el aire en una región específica durante todo un día, un mes o un año.
- Temperatura mínima: Se refiere a la menor temperatura que se registra en un lugar durante un periodo de un día, un mes o un año.
- Temperatura media: Es el promedio estadístico que se calcula con los valores de la temperatura máxima y mínima, ya sea diario, mensual o anual.



Para la conversión entre las distintas unidades de temperatura se emplean las siguientes ecuaciones:

1. De Fahrenheit a Celsius

$$C = \frac{5}{9}(F - 32)$$

2. De Celsius a Fahrenheit

$$F = 1,8(C) + 32$$

3. De Celsius a Kelvin

$$K = C + 273,15$$

#### **1.4.2.1. Medición**

Con el termómetro de mercurio se mide la temperatura del aire. Este es el dato climatológico más conocido. El termómetro recibe el calor transmitido por conducción del aire que lo rodea.

El termómetro ordinario consta de un depósito de vidrio con paredes muy delgadas, para que las variaciones de calor se transmitan con rapidez al líquido contenido en su interior.

Este dispositivo se prolonga en un tubo capilar delgado, por el que asciende el líquido, al dilatarse indica la temperatura.

### 1.4.3. Velocidad y dirección del viento

El viento es el movimiento de una masa de aire en la atmosfera, que posee una dirección y velocidad en la que se dirige. La velocidad se mide en kilómetros por hora (Km/h) y la dirección del viento la determina el punto cardinal desde donde se origina.

Tabla II. Clasificación según la velocidad del viento

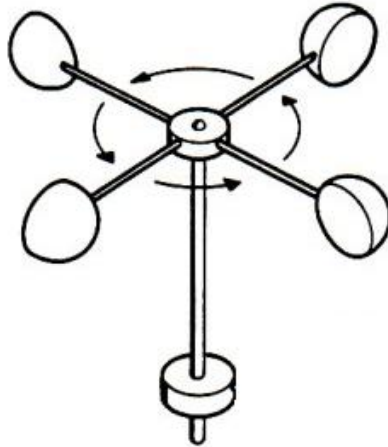
Velocidad del Viento (Km/h)	Descripción
1 - 11	Brisa suave, vientos suaves
12 - 28	Brisa moderada
29 - 61	Ventarrones, brisa moderada
62 - 102	Vientos fuertes, brisa fuerte
103 - 116	Tormenta, ventarrón constante
> 117	Huracán

Fuente: elaboración propia.

Para la medición de la velocidad y dirección del viento se usa un instrumento llamado *anemómetro*, que consta de pequeños discos cóncavos que atrapan el viento lo cual permite girar el dispositivo, mide así las revoluciones que da y calcula la velocidad del viento.

La fuerza del viento se estima en función de la magnitud que corresponde a su velocidad de desplazamiento; se registra según sus efectos mecánicos con un instrumento llamado anemómetro y se clasifican según la escala de Beaufort. La dirección del viento se mide mediante el uso de una veleta.

Figura 7. **Anemómetro convencional**



Fuente: [www.monografias.com](http://www.monografias.com). Consulta: 08 de junio de 2017.

#### **1.4.3.1. Características del viento**

Se clasifican en:

- Dirección: Se relaciona directamente con la rotación terrestre y se denota en grados sexagesimales y por su velocidad, que se suele expresar en nudos, por ejemplo; millas náuticas por hora, en metros por segundo o en kilómetros por hora.
- Intensidad: Se mide según la escala de Beaufort que va desde el 0 (calma absoluta) hasta 12 (huracán). La intensidad es directamente proporcional a la diferencia de presión entre el lugar de origen del viento y el de su llegada. Se mide con los anemómetros y es comúnmente llamada velocidad del viento.

#### 1.4.4. Humedad del aire

Se produce por el vapor de agua que se encuentra en la atmosfera. Dependiendo de la temperatura del aire, así es la cantidad de vapor que puede absorber, mayor temperatura, más vapor; Menor temperatura, menos vapor. Se expresa en gramos por metro cúbico de aire (g/m<sup>3</sup>). La humedad específica está dada por la siguiente fórmula:

$$w = \frac{p_v}{p} * \frac{R_a}{R_v} = 0,622 * \frac{p_v}{p}$$

Dónde:

$R_a$  es la constante de gases, que para el aire= 29,7 m/K

$p_v$  es la presión parcial del vapor de agua

$p$  es la presión atmosférica

$R_v$  es la constante de gases para el vapor de agua = 47,1m/k

##### 1.4.4.1. Clasificación y medición

La humedad posee tres clasificaciones:

- Humedad Relativa: Se da según la relación que exista entre la cantidad de vapor de agua en la atmosfera y la máxima cantidad que puede tener como referencia a la misma temperatura.
- Humedad Absoluta: Es la masa total de agua presente en el aire por unidad de volumen, y sus unidades de medición son los gramos por metros cúbicos de aire.

- **Humedad Específica:** Es la cantidad de masa de agua que se encuentra en estado gaseoso en un kilogramo de aire húmedo.

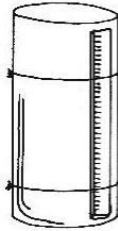
Para la medición de la humedad se utiliza el psicrómetro que es un instrumento que consta de dos termómetros: el seco que mide la temperatura real y el húmedo o mojado. Está rodeado por una muselina humedecida. Sobre la tela se evapora más o menos agua según si la humedad atmosférica es menor o mayor. La evaporación será mayor cuanto menor sea la humedad relativa. El agua para evaporarse toma calor del termómetro mojado y esto hace que descienda su temperatura. Con la lectura de los dos termómetros y tablas confeccionadas a tal efecto, se deduce la humedad relativa.

#### **1.4.5. Precipitaciones**

Es la caída de un hidrometeoro desde la atmosfera hacía la tierra, la lluvia, llovizna, nieve o granizo que se precipita hacía la superficie de la tierra. Las unidades de medida de dicho fenómeno son los litros por metro cuadrado ( $L/m^2$ ) en un área específica, mientras que de la precipitación en si son los milímetros (mm).

Para la medición de la cantidad de lluvia que cae sobre una pequeña región de interés se utiliza el pluviómetro, que es un instrumento que se usa para medir en milímetros la cantidad de agua que cae en un determinado tiempo.

Figura 8. **Pluviómetro**



Fuente: [www.monografias.com](http://www.monografias.com). Consulta: 08 de junio de 2017.

#### **1.4.5.1. Tipos de precipitaciones**

Las precipitaciones pueden clasificarse de la siguiente manera:

- Llovizna: Precipitación blanda, uniforme, formada por gotas finas de gran número y de un diámetro menor a 0,5 mm.
- Lluvia: Precipitación normal, acuosa, en forma de gotas
- Lluvia ácida: Precipitación que lleva diluido en el agua óxido de azufre y nitrógeno, creando así una lluvia contaminada, la cual crea gases dañando la atmosfera.
- Nevada: Precipitación que incluye en sus elementos a la nieve. Es decir, cuando el agua helada se separa de las nubes y llega a la tierra en forma de copos blancos o mejor conocida como nieve.

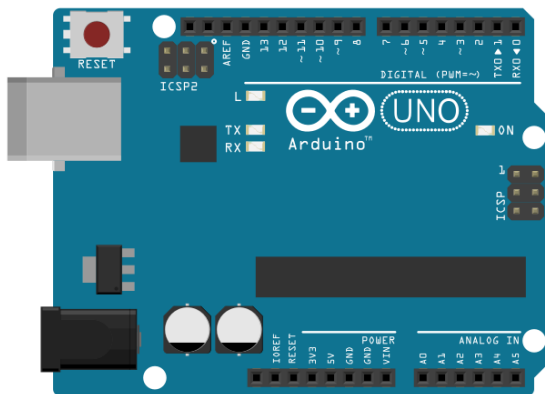
## 2. ARDUINO UNO Y MSP432

### 2.1. Especificaciones

#### 2.1.1. Arduino Uno REV3

Arduino es una placa de software libre o conocida también como *open source* en inglés, consta de un microcontrolador Atmel Atmega328p de 8 bits, que es el encargado de todo el procesamiento de instrucciones que se guardan en la memoria del dispositivo. Para la alimentación de la placa Arduino, se puede utilizar un rango de voltaje que varía desde los 7 voltios hasta los 12 voltios DC, para que el dispositivo opere a 5 voltios DC en su funcionamiento interno. Arduino Uno REV3 cuenta con 14 puertos digitales de uso general de los cuales dos de ellos son para la comunicación serial (Puerto 0 y puerto 1), y seis tienen la capacidad de enviar pulsos modulados mediante la implementación de un PWM (*pulse-width modulation* por sus siglas en inglés).

Figura 9. **Arduino Uno REV3**



Fuente: [www.arduino.cc](http://www.arduino.cc). Consulta: 15 de julio de 2017.

También cuenta con seis puertos de entrada analógica para la medición de datos que pueden ser de sensores, potenciómetros, entre otros, permite tomar medidas reales y convertirlas a digitales para el entendimiento del microcontrolador. Tanto los puertos digitales como los análogos cuentan con una tolerancia de un máximo de 20 mA por puerto, es decir que la carga que se le aplique a los puertos no debe exceder dicha corriente, para que no se arruine de manera permanente, tanto el microcontrolador como los puertos.

Para la programación del microcontrolador se utiliza la comunicación serial mediante *Universal Serial Bus* tipo B, que permite introducir instrucciones directamente a la memoria y ejecutarlas a una alta velocidad, debido a un reloj de cristal que maneja frecuencias de hasta 16 MHz.

Tabla III. **Especificaciones Arduino UNO REV3**

Microcontrolador	Atmega328p
Voltaje de operación	5V
Voltaje de entrada	7-12V
Puertos I/O digitales	14
Puertos analógicos	6
Memoria Flash	32 Kb
Velocidad del Reloj	16 MHz

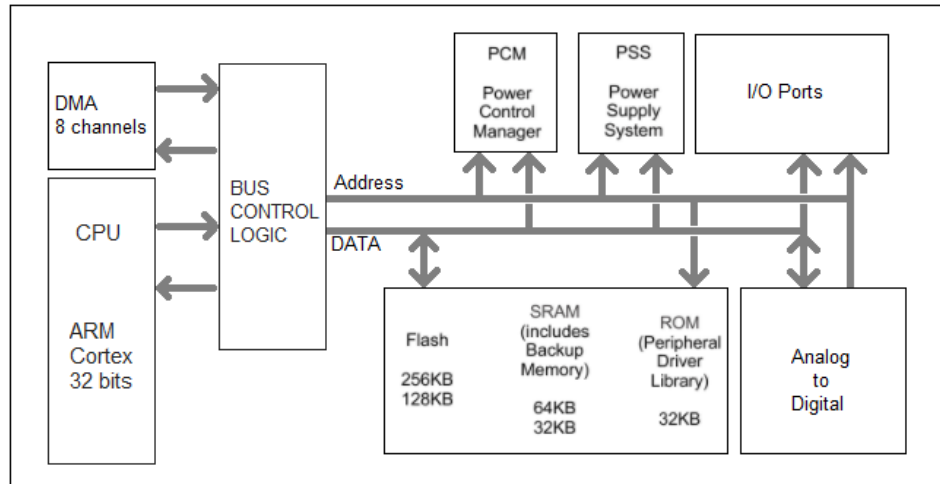
Fuente: [www.arduino.cc](http://www.arduino.cc). Consulta: 15 de julio de 2017.

### 2.1.2. **MSP 432**

Es un microcontrolador de la compañía Texas Instruments, y posee una unidad de procesamiento central de la familia *ARM Cortex* de 32 bits, con 48 MHz de frecuencia de reloj para procesar instrucciones alojadas en una memoria *Flash* de 256 Kb.



Figura 10. Diagrama de bloques de la MSP 432



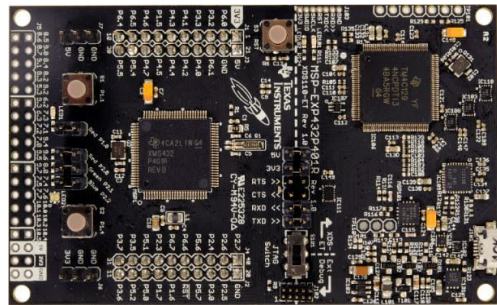
Fuente: elaboración propia, empleando Microsoft Word.

La comunicación que se utiliza en esta placa es mediante un Micro USB, y permite la interacción serial entre la computadora y el microcontrolador, como la programación del mismo. Cuenta con 48 puertos de entrada o salida con capacidad de interrupción y un suministro de voltaje de 3,3 voltios o 5 voltios, dependiendo del uso que se le fuera a dar. Mientras que internamente maneja un rango de voltaje de 1,6 voltios a 3,7 voltios. Tiene un ADC o Convertidor analógico digital de 14 bits que permite tomar muestras de los sensores con una exactitud muy alta.

El nivel de complejidad de esta placa es mayor en comparación al Arduino, pero a su vez tiene mejores características que permite obtener mayores beneficios como lo son:

- Almacenamiento
- Rapidez
- Exactitud

Figura 11. **Placa MSP-EXP432P401R**



Fuente: [www.makeelectronico.com](http://www.makeelectronico.com). Consulta: 17 de julio de 2017.

## 2.2. Ventajas y desventajas

Dado a que la velocidad con la que trabaja el microcontrolador de la MSP432 es ampliamente superior a la velocidad del Arduino Uno, es una opción más viable para leer los sensores de una manera más rápida y con un consumo de potencia menor. A su vez, la MSP432 cuenta con un ADC de 14 bits, y la hace más exacta a la hora de tomar lecturas de cada dispositivo conectado a sus puertos analógicos.

Por otro lado, Arduino cuenta con la ventaja de la compatibilidad, y muchos de los sensores propuestos no presentan ninguna restricción cuando se programan en el entorno de Arduino, a diferencia de la poca documentación que existe para la placa MSP432 ya que es una placa reciente.

Ambos presentan la desventaja que necesitan realizar una instrucción a la vez, porque siguen una estructura lógica y de manera jerárquica, debido a la programación que utilizan. La placa MSP432 presenta mayor capacidad de almacenamiento por lo que es posible crear más instrucciones en base a la posibilidad de conectar una mayor cantidad de sensores a los puertos analógicos.

## **2.3. Entorno de desarrollo integrado**

### **2.3.1. Arduino**

Conocido por sus siglas en inglés *Integrated Development Environment* (IDE), es una aplicación que se encarga de traducir las instrucciones, que un operario ingresa a información útil para el microcontrolador, y así poder realizar el proceso que el usuario desea.

El lenguaje que se usa para la programación es variado y en el caso de Arduino y de la MSP432 utilizan el lenguaje basado en C, que permite una mayor facilidad de comprensión y elaboración de códigos para realizar procesos en tiempos óptimos, y menor conjunto de instrucciones para almacenar en la memoria de la placa a utilizar.

Figura 12. **Software Arduino IDE**



Fuente: elaboración propia. [www.arduino.cc](http://www.arduino.cc)., empleando Microsoft Word.

En la imagen anterior se pueden enlistar cuatro partes importantes de la aplicación IDE de Arduino, que conforman la parte esencial de la configuración y programación de la placa.

- De izquierda a derecha se tiene el botón para verificar que las instrucciones estén bien configuradas o escritas. Luego está el botón para subir o *upload* el archivo a la memoria de la placa. Por último, están los botones para crear un nuevo archivo, abrir uno ya creado o guardar el archivo presente.
- El botón en forma de lupa representa la ventana de la comunicación serial entre el Arduino y la computadora, en la mayoría de casos es mediante el USB. Permite observar los

datos que son desplegados por el Arduino mediante la instrucción *Serial.print* y sus variaciones.

- Es el espacio disponible para escribir o configurar las instrucciones que el microcontrolador leerá luego de que se cargue el archivo a la memoria del mismo. Entre más instrucciones se escriban, más memoria utilizará el microcontrolador para almacenar dichas instrucciones.
- Indicador de que puerto COM está siendo utilizado por la placa Arduino.

### **2.3.2. MSP432**

Para el entorno de desarrollo de la MSP se encuentra con la aplicación Energia, que es una variación del IDE de Arduino, por lo que al comparar ambos IDE se observa una gran semejanza, varía solo el color, inclusive el lenguaje de programación es el mismo, y los detalles de la aplicación son los mismos enlistados con anterioridad.

Figura 13. **Software Energia MPS432**



Fuente: elaboración propia. [www.energia.ru](http://www.energia.ru), empleando Snipping Tool.

Para la programación de la MSP432 se seguirán los mismos pasos que para el Arduino, varían solo los puertos a usar de cada placa, porque ambas cuentan con distintos nombres para enlistar dichos puertos de uso general, como los PWM y los puertos análogos.

La MSP432 cuenta con mayor capacidad de almacenaje, no representa un problema el colocar todos los sensores conectados, tomando medidas y almacenando información, porque placa es capaz de realizar todas las instrucciones con mayor fluidez que la placa Arduino.

## **2.4. Sensores de medición y módulos**

Serán los encargados de tomar los datos de los parámetros atmosféricos, y luego serán procesados por el microcontrolador.

Entre los sensores a usar están:

- Sensor de presión y temperatura BMP180
- Sensor de humedad DHT11
- Sensor de precipitación

### **2.4.1. Sensor de presión BMP180**

#### **2.4.1.1. Arduino UNO REV3**

Este sensor toma las medidas absolutas de la presión atmosférica, mediante el cálculo de la temperatura. Posee la opción de calcular la altura a nivel del mar en la que se encuentra. Incluye un regulador de 3,3 voltios, para poder hacer uso de los 5 voltios del microcontrolador sin problema de dañar el sensor.

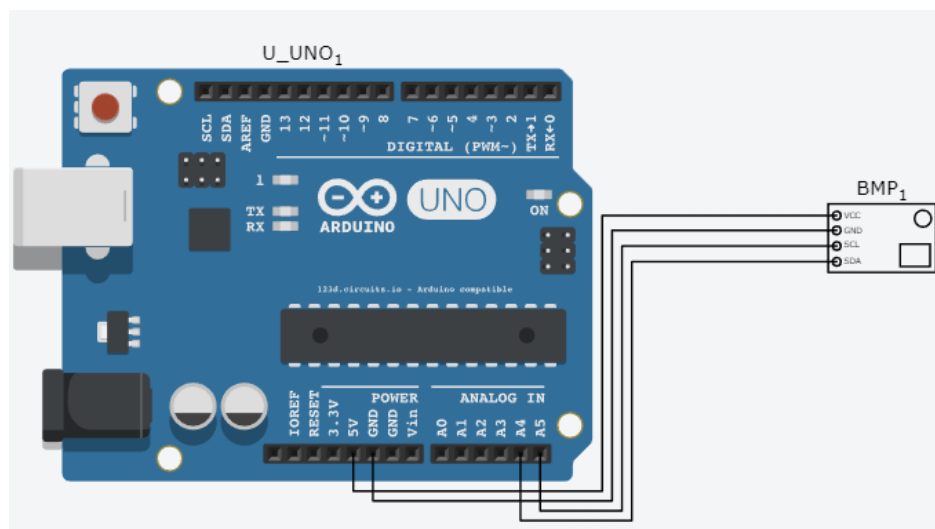
Figura 14. **Sensor BMP180**



Fuente: [http://img.dxcn.com/productimages/sku\\_435568\\_1.jpg](http://img.dxcn.com/productimages/sku_435568_1.jpg). Consulta: 20 de julio de 2017.

La conexión o configuración del *hardware* para el Arduino Uno REV3 y el sensor BMP180 es la siguiente:

Figura 15. **Conexión de Arduino y BMP180**



Fuente: elaboración propia, empleando CircuitIO.



Tabla IV. **Conexión de pines**

ARDUINO	BMP180
5V	VCC
GND	GND
A4	SDA
A5	SCL

Fuente: elaboración propia.

Para el uso de este sensor se debe descargar una librería que deberá ser instalada en el folder del microcontrolador, en la computadora que programará la placa. Esta librería se puede conseguir desde la página de *GitHub* dentro del apartado de SparkFun.

Dentro de los comandos necesarios para el correcto funcionamiento del sensor BMP180 se encuentran los siguientes:

- *begin()*: El cual inicia la librería del sensor para que éste funcione de manera correcta. Da como resultado un 1 si es correcta o un 0 si es incorrecta la inicialización.
- *startTemperature()*: Comando que se utiliza para iniciar el proceso de lectura de los datos previo al cálculo de la temperatura del lugar.
- *getTemperature(T)*: Toma la lectura y realiza el cálculo de la temperatura, almacenando la información en una variable T. Se llama primero al comando *startTemperature()* para que el funcionamiento sea el correcto.
- *startPressure()*: Comando utilizado para iniciar el proceso de toma de medición de la presión. Se complementa con un número del 0 al

3 dentro del paréntesis para indicar cuantas muestras adicionales debe tomar el sensor.

- *getPressure(P,T)*: Realiza el cálculo de la presión del medio ambiente, complementado con la variable de temperatura T, dado que dicho parámetro afecta considerablemente la presión atmosférica del lugar. Los datos se almacenan en la variable P.

Este sensor incluye la posibilidad de calcular la altitud y la presión respecto al nivel del mar en el que se encuentra mediante los comandos siguientes:

- *altitude(P,Po)*: Calcula la altitud mediante la presión calculada P y una presión de referencia Po.
- *sealevel(P,A)*: Utilizando la altitud A del comando anterior, calcula la presión respecto al nivel del mar mediante la variable P almacenada anteriormente.

### Figura 16. Configuración inicial

```
void setup()
{
  Serial.begin(9600);

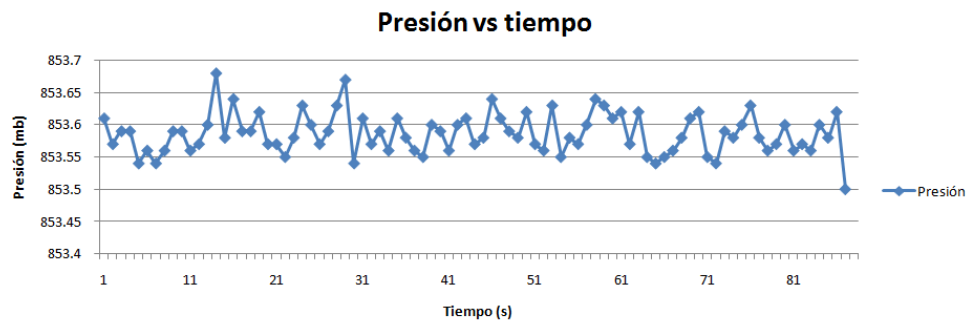
  if (bmp180.begin())
    Serial.println("BMP180 listo");
  else
  {
    Serial.println("Error en el proceso de inicializacion BMP180");
  }
}
```

Fuente: elaboración propia, empleando IDE.

La configuración inicial que lleva el sensor corresponde a la comunicación serial a 9600 bits por segundo entre ambos dispositivos, y mensajes de estados de dicho sensor al iniciar el proceso en el microcontrolador.

Para analizar el comportamiento del sensor, se toman medidas de la presión en milibares (mb) para poder graficar los datos obtenidos, respecto al tiempo que el sensor obtuvo la información.

Figura 17. **Gráfica de datos**



Gracias a que el software Energía está basado en el motor de Arduino, los códigos para programar y configurar el sensor BMP180 son los mismos enlistados en el apartado 2.4.1.1.

Figura 18. **Ejemplo de código**

```
SFE_BMP180 pressure;  
  
#define ALTITUDE 1500.0
```

Fuente: Librería SparkFun BMP180.

Mediante la toma de datos con el sensor BMP180, la placa MPS432 puede desplegar la información de manera rápida y precisa. En este caso se tomó como ejemplo la librería de SparkFun para dicho sensor, y despliega la temperatura, la presión absoluta, relativa y la altitud que uno ingresa según la localidad que se estudia.

Figura 19. **Salida serial del sensor**

```
provided altitude: 1500 meters, 4921 feet  
temperature: 27.26 deg C, 81.06 deg F  
absolute pressure: 854.66 mb, 25.24 inHg  
relative (sea-level) pressure: 1024.12 mb, 30.25 inHg  
computed altitude: 1500 meters, 4921 feet
```

Fuente: Librería SparkFun BMP180.

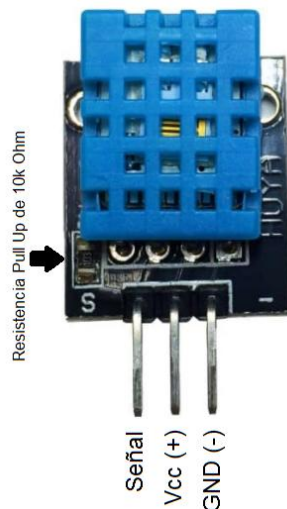
## 2.4.2. Sensor de humedad DHT11

### 2.4.2.1. Arduino UNO REV3

Este sensor se encargará de la toma de datos de la humedad del área que se está censando en ese momento. Tiene la opción de desplegar los valores de la temperatura, pero este proceso se llevará a cabo con el sensor BMP180 y solo se usará para comparar la precisión de ambos sensores.

El sensor DHT11 cuenta con un rango de exactitud de la humedad del 5,0 % y un tiempo de respuesta menor a 50 ms. Con un consumo de energía muy bajo, dicho sensor resulta muy útil para la implementación de un conjunto de componentes amigables al medio ambiente, es decir que su bajo consumo permite el uso de paneles solares para trabajar de manera eficiente.

Figura 20. **Sensor DHT11 con placa**



Fuente: elaboración propia, empleando Snipping Tool.

La configuración de este sensor es relativamente sencilla, solo necesita la instalación de la librería como se realizó con el sensor BMP180, iniciar la comunicación serial entre la placa y el sensor, chequear la existencia de errores de inicialización del sensor y por último realizar la toma de la humedad y temperatura.

En algunos casos, las terminales pueden variar, es decir que la terminal de salida se encuentra en medio de las tres terminales pero la configuración y conexión al microcontrolador sigue siendo la misma.

Tabla V. **Conexión de pines**

ARDUINO	DHT11
D2	Señal
5V	Vcc
GND	GND

Fuente: elaboración propia.

Para la toma de datos, el sensor se configura en el puerto deseado, en este caso se utiliza el puerto digital D4 para la recepción de la información. Se crean las variables que almacenaran esta información y luego se despliegan de manera serial, para tener control de que sea la información correcta.

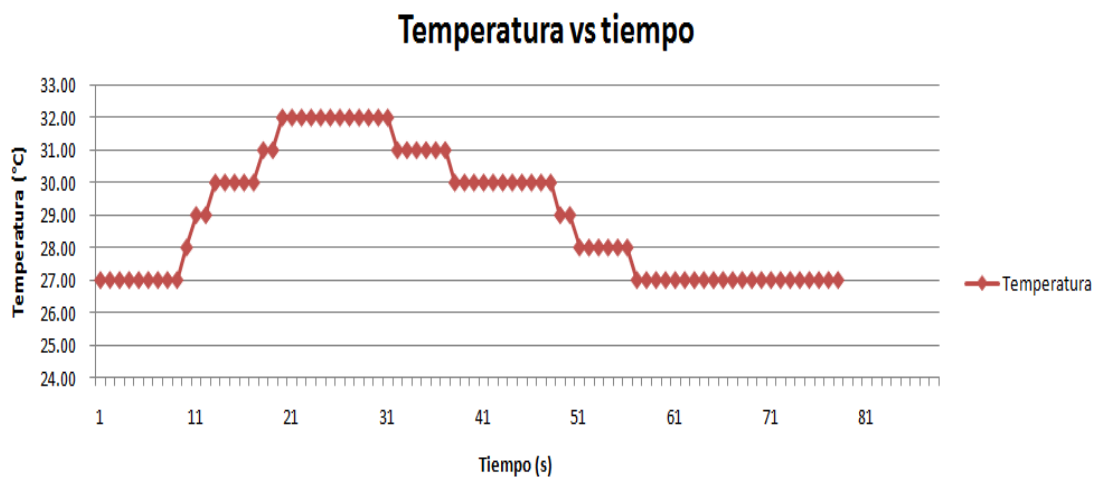
Figura 21. **Sección de muestreo de datos**

```
void loop()
{
  int er;
  float temp, hum;
  if((er=dht11.read(hum, temp))==0)
  {
    Serial.print("Temperatura: ");
    Serial.print(temp);
    Serial.print(" Humedad: ");
    Serial.print(hum);
    Serial.println();
  }
}
```

Fuente: Librería Arduino DHT11.

La información almacenada se presenta mediante una gráfica que compara la temperatura que se registró en el sensor, y el tiempo que éste estuvo activo. Se aplicó aire tibio al sensor para que aumentara la temperatura a manera demostrativa para la toma de datos.

Figura 22. **Gráfica de los datos tomados**



Fuente: elaboración propia.

### 2.4.2.2. MSP432

El proceso para el funcionamiento correcto de este sensor en la placa MSP432 es similar a la conexión y configuración realizada con el microcontrolador Arduino. Se debe instalar la librería en la carpeta de Energía y luego llamar a la toma de datos mediante un puerto digital de elección y desplegar el valor mediante la comunicación serial para su uso posterior.

Si se elige el puerto digital D2, la conexión quedará de la misma manera que se indica en la tabla V, y facilita la compatibilidad de la librería con el microcontrolador de la placa MSP43, y dada las características de dicha placa, la toma de datos sería más rápida y habría mayor almacenamiento para las variables.

Para la comunicación entre el sensor y la placa es necesario especificar qué puerto analógico se usará para la recepción de datos, en este caso se utilizará el puerto A6. De la misma manera, se debe incluir la librería que se encarga de hacer el procesamiento matemático de los datos obtenidos, es decir, pasar de valores analógicos a valores entendibles para el ser humano.

Figura 23. **Código de configuración**

```
#include <DHT11.h>
int pin=A6;
DHT11 dht11(pin);
```

Fuente: Librería DHT11.



Se despliega la información en dos columnas que representan la temperatura y la humedad que el sensor percibe. Se aplicó una pequeña fuente de calor para que el sensor mostrara el cambio susceptible en ese preciso momento.

Figura 24. **Salida Serial**

```
temperature:28.00 humidity:77.00  
temperature:28.00 humidity:90.00  
temperature:29.00 humidity:91.00  
temperature:30.00 humidity:91.00
```

Fuente: Librería DHT11.

### **2.4.3. Sensor de precipitación**

#### **2.4.3.1. Arduino UNO REV3**

La función de este sensor es bastante sencilla, ya que dependiendo de la cantidad de agua que exista sobre los contactos del sensor, es la cantidad de información que enviará al microcontrolador. Para hacer esto, se usará los puertos analógicos o digitales de la placa a usar. Si se desea obtener la información de manera analógica, se usará el puerto del ADC, es decir *Analog-to-digital Converter*, obteniendo así valores de que van de 0 a 1 024. Con estos datos se procede a realizar una relación de porcentaje, para calcular que tanta agua cubre al sensor y en base a esto dar una conclusión de tres posibles estados:

- Seco: es decir no hay agua presente en el sensor (1 024 de ADC)
- Intermedio: Poca presencia de agua (valores intermedios)
- Mojado: Alta presencia de agua en el sensor (0 de ADC)

Figura 25. **Sensor de precipitación**



Fuente: elaboración propia, empleando Snipping Tool.

Adicional a la configuración necesaria, se tiene un acople que permite la comunicación entre la placa del sensor y la placa del microcontrolador. Ésta cuenta con un potenciómetro y un acople de voltaje para el flujo de los datos, tanto en digital (*HIGH* y *LOW*) como analógico (valores de 0 a 1 024). También cuenta con indicadores visuales para cuando la placa esté en funcionamiento y con voltaje.

Figura 26. **Acople de sensor y microcontrolador**



Fuente: elaboración propia, empleando Snipping Tool.

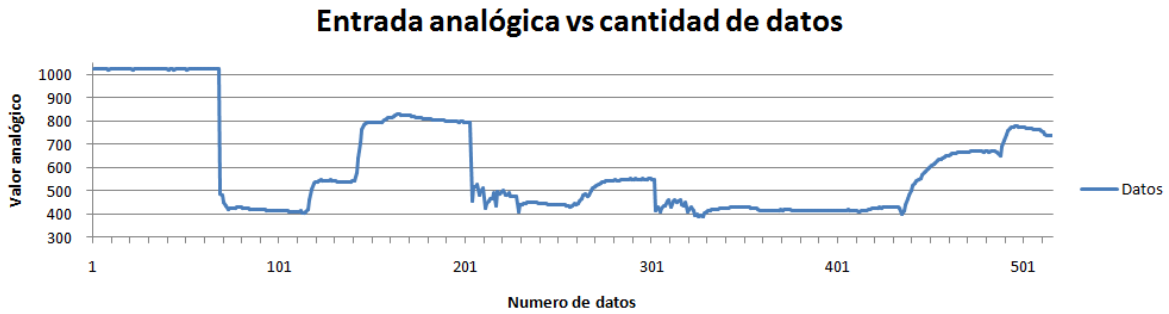
La toma de datos se realiza mediante la lectura del puerto analógico A0, en el ingresan los valores que varían en un rango de 0 a 1 023, donde éste último representa un sensor totalmente seco, es decir que no existe presencia de agua en la placa de precipitación. Por otra parte, un 0 representaría 0 % de la placa cubierta en agua. Para fines prácticos se tomó en consideración un 1 en lugar del 0.

Figura 27. **Toma de datos de la placa**

```
void loop() {  
  Serial.println(analogRead(A0)); //lectura analógica  
  delay(100);  
}
```

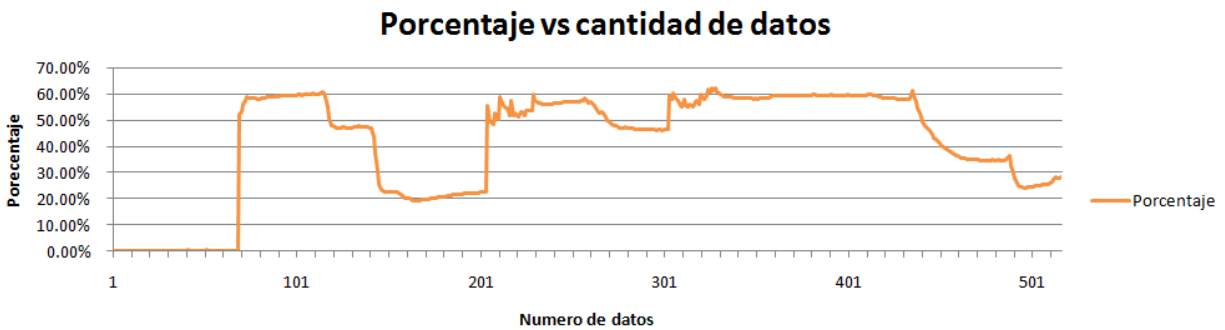
Fuente: elaboración propia, empleando IDE.

Figura 28. Gráfica de datos análogos



Fuente: elaboración propia.

Figura 29. Gráfica de porcentajes



Fuente: elaboración propia

Se observa que inicialmente la placa está 100 % seca, o bien 0 % cubierta de agua y luego de añadir un par de gotas sobre la placa, el porcentaje varía, llegando a estar en un 40 % seco o 60 % cubierta de agua.

Cada cambio presente en la gráfica representa un aumento o disminución de la cantidad de agua presente sobre los contactos de la placa de precipitaciones. Lamentablemente dicha placa está limitada solamente a

detectar el porcentaje de líquido presente sobre ella, y no la cantidad de gotas que caen en un intervalo de tiempo.

#### 2.4.3.2. MSP432

El proceso para la conexión del sensor en la placa MSP432 es el mismo que para el Arduino, solo varía en el nombre del puerto analógico que se usará, es decir que, a la hora de configurar el código para dicho microcontrolador, hay que tener en cuenta los nombres que pueden tomar los puertos analógicos.

Luego de la obtención de los datos, el procedimiento es el mismo que para la placa Arduino, los valores que se obtienen siguen dentro del rango de 0 a 1024, representando los tres estados del sensor mencionados en el apartado 2.4.3.1.

Tabla VI. Puertos de conexión

MSP432	Sensor
<i>Analog Pin</i>	Señal

Fuente: elaboración propia.

Para la configuración entre la placa y el sensor, se utiliza una comunicación serial de 9600 bits por segundo para una mayor transferencia de información. Este valor puede cambiarse a otra cantidad de bits por segundo, para que pueda tener relación con los demás sensores y se ajuste a las necesidades del usuario.

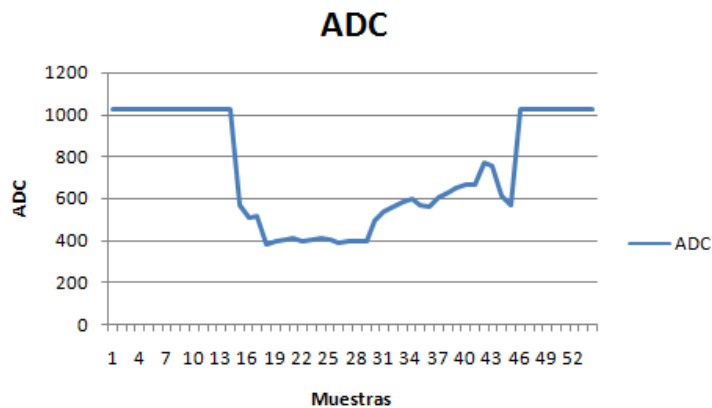
Figura 30. **Configuración puerto serial**

```
void setup()
{
  Serial.begin(9600);
}
```

Fuente: elaboración propia, empleando IDE.

De la misma manera que se hizo con Arduino, se grafican los datos obtenidos de la comunicación de la placa y el sensor, mostrando el comportamiento del sensor cuando se le aplica una cierta cantidad de agua sobre las terminales.

Figura 31. **Gráfica de datos**



Fuente: elaboración propia.

### 3. ENVÍO Y RECEPCIÓN DE DATOS

#### 3.1. Comunicación *Bluetooth*.

##### 3.1.1. Módulo HC-05

La comunicación *Bluetooth* permite el envío y recepción de datos digitales entre de dos dispositivos separados a una distancia máxima de diez metros. Esta tecnología es fácil de implementar, pero no permite una libre comunicación cuando se encuentra rodeada por objetos que bloquean el paso de las señales de las antenas. Se usa dicha tecnología para entablar comunicación entre el microcontrolador y un dispositivo o usuario final, como puede ser un celular, una tableta o un ordenador con *bluetooth* incorporado.

Mediante el uso de esta comunicación se pretende enviar, recibir y almacenar los datos tomados por los sensores indicados en el capítulo anterior, permitiendo manipular dichos datos para que sean interpretados de manera visual en cualquier dispositivo.

Como la comunicación es de dos vías, se puede enviar comandos o datos del dispositivo hacia el microcontrolador, para limitar la toma de datos o mostrar solo cierta cantidad de información, según el usuario lo requiera, es decir, mostrar uno por uno cada sensor y como estos manipulan los datos. Esto se logra mediante una aplicación desarrollada en una plataforma *Android* e instalada en un celular.

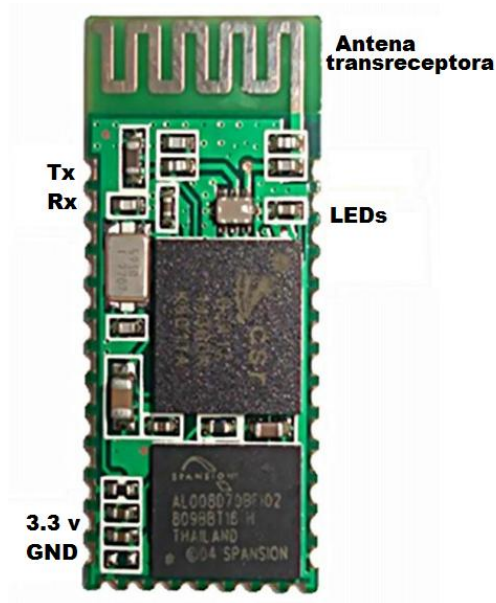
El dispositivo HC-05 cuenta con dos posibles estados de funcionamiento:

1. Maestro
2. Esclavo

En el modo maestro el sensor permite controlar la conexión entre los dispositivos, es decir que se puede conectar a un sensor esclavo. Para este caso, el módulo HC-05 funcionará como esclavo ya que el dispositivo celular se conectará a él, por lo que quien controla la conexión es el móvil, actuando como maestro. Una vez la conexión se establece, comienza la transmisión de datos.

El módulo cuenta con dos pines para la transmisión y recepción de los datos, una antena integrada y una entrada de voltaje de 3,3v.

Figura 32. **Módulo HC-05 sin pines**



Fuente: elaboración propia, empleando Snipping Tool.



El módulo que se usará cuenta con una placa extra, se le incorporan pines y un regulador de voltaje, permitiendo una entrada mayor de 3,3 voltios para su funcionamiento.

Figura 33. **Módulo HC-05 con pines (parte trasera)**



Fuente: <http://www.electronicoscaldas.com>. Consulta: 20 de octubre de 2017.

Figura 34. **Módulo HC-05 con pines (parte delantera)**



Fuente: <http://www.electronicoscaldas.com>. Consulta: 20 octubre de 2017.

Tabla VII. **Puertos de conexión**

Microcontrolador	Módulo
Vcc	3,6 V - 6 V
GND	GND
Tx	Rx
Rx	Tx

Fuente: elaboración propia.

Dicho módulo trabaja con las siguientes especificaciones:

Tabla VIII. **Especificaciones**

Modelo	HC-05
Frecuencia	2,4 GHz
Modulación	GFSK
Potencia	$\leq 4$ dBm
Sensibilidad	$\leq -84$ dBm
Seguridad	Clave encriptada
Consumo	25 mA
Niveles lógicos	3,3 V
Temperatura de op.	-20 °C a 75 °C
Voltaje de entrada	3,6 V a 6 V
Velocidad	2,1 Mbps (máx.)

Fuente: elaboración propia.

### 3.1.1.1. **Configuración en Arduino y MSP432**

Por medio de la siguiente configuración se logra establecer un canal de comunicación *full duplex* entre el dispositivo HC-05 y el microcontrolador Arduino, (se aplica la misma configuración para el microcontrolador MSP432) para poder enviar y recibir los datos de los sensores. La tasa de baudios se inicia a un valor por defecto, en este caso, 9 600, lo cual permite enviar bits a

una velocidad adecuada para evitar la pérdida de datos, ya que la calidad de la información debe ser muy precisa. Como los sensores estarán tomando mediciones cada cierto período de tiempo, no es necesario colocar una tasa de baudios más elevada.

Figura 35. **Configuración inicial**

```
char data = 0;
void setup()
{
  Serial.begin(9600);
}
```

Fuente: elaboración propia, empleando IDE.

Se utiliza una variable del tipo *char* para almacenar los datos recibidos por el dispositivo HC-05, el cual, a su vez, recibe la información del dispositivo móvil conectado a él. Esta configuración, junto a la iniciación de la tasa de baudios, se coloca dentro de la sección de configuración (*voidsetup*, en inglés).

Figura 36. **Sección del bucle**

```
void loop()
{
  if (Serial.available() > 0)
```

Fuente: elaboración propia, empleando IDE.

### **3.1.1.2. Ventajas y desventajas**

En referencia al módulo HC-05 se presentan dos claras ventajas y son el hecho de una fácil conexión entre el dispositivo maestro y el esclavo ya que no requiere de mayor cosa para emparejar ambos equipos. Se coloca una clave previamente configurada y la conexión se realiza. La otra ventaja es el bajo consumo de energía del módulo, ya que solo consume un aproximado de 25 mA, que es una buena relación para tomar en cuenta ya que se usarán fuentes renovables para el funcionamiento de dicho módulo.

Entre las desventajas que posee el módulo HC-05 se representa mediante una limitante de tecnología, ya que *bluetooth* se maneja mediante ondas, el alcance de dichas ondas está regido por los obstáculos que se presentan en el camino de la señal y de la potencia que pueda emitir la antena junto con el controlador de la placa, y como la tecnología *bluetooth* cuenta con un rango de 10 metros de alcance ideal para una señal, se debe tomar en cuenta que para una óptima comunicación se requiere que los dispositivos emparejados estén dentro de dicho rango.

### **3.1.2. Módulo WiFi ESP8266**

El módulo ESP8266 es un sistema con un chip integrado, el cual permite el uso del protocolo TCP/IP para la comunicación entre dicho dispositivo y un sistema WiFi. Es decir, actuar como un *Access Point* o un conectarse a una red, ya sea pública o privada mediante comandos AT.

Cuenta con un procesador integrado, que permite usar el módulo como un controlador, es decir se pueden agregar sensores, *LEDs*, botones, etc., para interactuar con el dispositivo. A pesar de esta ventaja, se utilizará el

microcontrolador Arduino. Para manejar la comunicación entre la ESP8266 y la red a la que se desea acceder. Para la interacción entre la red y el dispositivo, el módulo tiene integrada una antena en la placa PCB, permitiendo así la recepción y transmisión de la información, según sea el alcance de la red inalámbrica.

Tabla IX. **Parámetros de la ESP8266**

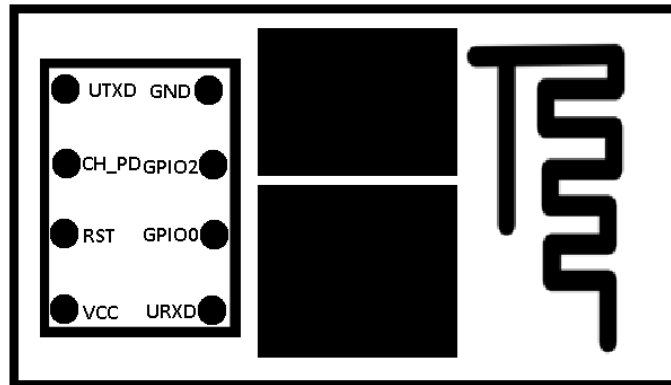
Parámetro	Valor
Protocolo WiFi	802.11 b/g/n
Frecuencia	2,4 – 2,5 GHz
Potencia de antena	20/17/14 dBm
Voltaje de operación	3,0 – 3,6 V
Temperatura de operación	-40° - 125° C
Seguridad	WPA/WPA2
Encriptación	WEP/TKIP/AES
Protocolos de Red	IPv4, TCP/UDP, FTP, HTTP
Instrucciones	Comandos AT

Fuente: elaboración propia.

El módulo cuenta con pines de Tx y RX los cuales permiten la comunicación entre el microcontrolador y la placa y se deben conectar de manera cruzada, Tx conectado a Rx. Cuenta con dos pines de entrada o salida de uso general (GPIO), y darles uso en caso de programar el módulo como controlador independiente.

La placa soporta un voltaje de entrada de 3 a 3,3 voltios y como no posee regulador, si se llegase a conectar a 5 voltios, la placa sufriría daños permanentes.

Figura 37. Diagrama ESP8266



Fuente: <https://hetpro-store.com/TUTORIALES/esp8266-arduino-crear-servidor-web/>. Consulta: 22 de octubre de 2017.

### 3.1.2.1. Configuración inicial de la ESP8266

Para la configuración del módulo se utilizaron comandos AT, que permiten reiniciar el dispositivo, verificar el modo de uso del equipo y conectarse a una red disponible.

Figura 38. Configuración Serial

```
#include <SoftwareSerial.h>  
SoftwareSerial BT(3, 2); // RX | TX
```

Fuente: <https://hetpro-store.com/TUTORIALES/esp8266-arduino-crear-servidor-web/>. Consulta: 22 de octubre de 2017.

Mediante la librería *SoftwareSerial* se puede asignar los pines a usar para Rx y Tx. Se utilizan los pines digitales dos y tres del microcontrolador para evitar interferencias con la programación de dicho controlador, ya que al cargar

el código fuente desde el IDE, se utilizan los pines Rx y Tx (cero y uno) del chip y eso podría crear una mala recepción del código o de la señal WiFi.

Figura 39. **Configuración ESP8266**

```
char txt;
void setup()
{  Serial.begin(9600);
   BT.begin(9600);
}

void loop()
{if (Serial.available())
 {txt = Serial.read();
  BT.print(txt);
 }
 if (BT.available())
 {txt = BT.read();
  Serial.print(txt);
 }
}
```

Fuente: <https://hetpro-store.com/TUTORIALES/esp8266-arduino-crear-servidor-web/>. Consulta: 23 de octubre de 2017.

Luego de cargar este código se procede a configurar el módulo mediante comandos AT para acceder a la red que se desea usar para obtener IP. Esta configuración se realiza una sola vez, ya que el módulo se conectará automáticamente a la red que nosotros accedamos, sin embargo, es necesario establecer en el arranque del controlador que se configure en modo servidor y en modo de conexiones múltiples.

Tabla X. **Comandos AT.**

Comando	Función
"AT"	Verifica la respuesta del módulo ante las instrucciones a usar.
"AT+CIOBAUD=9600"	Establece la comunicación serial a 9600 baudios.
"AT+CWMODE=3"	Permite al dispositivo actuar como un "Access Point" en modo <i>Station</i> 1 = Modo <i>Station</i> 2 = Modo AP 3 = Modo AP + <i>Station</i>
"AT+CWLAP"	Muestra un listado con las redes (SSID) en el rango de la antena del módulo-
"AT+CWJAP= "ssid", "password"	Realiza la conexión entre el módulo y la red a la que se desea conectar mediante el nombre (SSID) y la clave ( <i>Password</i> )
"AT+CIFSR"	Muestra la IP y la MAC del módulo.
"AT+CIPMUX=1"	Configura al dispositivo para tener una conexión o múltiples conexiones. 0 = <i>Single Connection</i> 1 = <i>Multiple Connections</i>
"AT+CIPSERVER=1,80"	Configura el módulo como servidor. 0 = Borrar servidor 1 = Crear servidor 80 = Puerto TCP para http

Fuente: [https://github.com/espressif/ESP8266\\_AT/wiki](https://github.com/espressif/ESP8266_AT/wiki). Consulta: 05 de noviembre de 2017.

Figura 40. **Comandos AT en IDE**

```
String ordenes[] =
{ // "AT+RST",
  "AT",
  "AT+CIPMUX=1",
  "AT+CIPSERVER=1,80",
  "END" // Fin comandos AT
};
```

Fuente: elaboración propia, empleando IDE.



Al arranque de cada sesión del microcontrolador se ejecutarán los comandos de configuración para que el módulo funcione de la manera correcta y se conecte al SSID registrado.

Para el despliegue de datos se utilizará una pequeña página web creada desde el microcontrolador, con actualización de cada 60 segundos y una IP que se asignó directo desde el *router*, mediante la MAC del módulo WiFi, por lo que en cada sesión que la ESP8266 solicite una nueva IP al servidor, éste le asignará la misma dirección que ya tiene establecida, esto para evitar el cambio de IP y perder la página *web*.

Figura 41. **Configuración HTML**

```
void webservice(void) {
  http("<!DOCTYPE HTML>");
  http("<html>");
  http("<head><title>LECTURAS SENSORES.</title>");
  http("<meta http-equiv=\"refresh\" content=\"60\"></head>");
  http("<body><h1> Situacion Ambiente</h1>");

  http("Humedad: ");
  http( String(DHT.humidity));
  http(" % ");
  http("Temp: ");
  http(String(DHT.temperature));
  http(" C ");
  http("Presion: ");
  http(String(P));
  http(" mb ");
  http("Altitud: ");
  http(String(A));
  http(" m.s.n.m ");

  http("<br />");

  http("<p><em> La pagina se actualiza cada 60 segundos.</em></p></body></html>");
}
```

Fuente: <https://www.prometec.net/servidor-web-esp8266/>. Consulta: 08 de noviembre de 2017.

Con esto se logra desplegar los datos de humedad, temperatura, presión y altitud de la región seleccionada. El intervalo de toma de datos puede ser mayor, ya que no es necesario estar tomando los datos cada minuto. El único inconveniente presente en el uso de esta tecnología es la necesidad de estar en el rango del AP y obtener la IP correcta para el despliegue de la página web.

### **3.1.2.2. Ventajas y desventajas**

Entre las desventajas más notorias es la necesidad de aplicar una dirección IP manualmente al servidor (módulo ESP8266), porque si se asigna de manera automática, se corre el riesgo de que el *router* le asigne una IP distinta a la sesión anterior, lo que causaría que no se despliegue la información de la página web.

Se debe mantener el módulo a una distancia prudente del *router*, teniendo en cuenta que entre más objetos se encuentren en el camino de las ondas, la intensidad será menor en la antena receptora, lo que provoca calentamiento de dicha antena y fallas en la página web. Dicho rango es mayor que el que se usa para la comunicación *bluetooth*, por lo que hace a esta tecnología la principal para la interacción del usuario final.

La configuración inicial para el dispositivo no es la más sencilla, ya que requiere el uso de comandos AT para obtener una IP según la red a la que se conecte y colocar al equipo en modo AP para poder hacer uso de las características web.

Tabla XI. **Ventajas y desventajas**

Ventajas	Desventajas
Mayor rango de alcance de operación	Mayor consumo de potencia debido a la antena transreceptora.
Fácil manejo de la configuración para la página web.	Consumo de memoria para la implementación de la configuración HTML.
Amplia forma de desplegar la información de los sensores.	Configuración inicial compleja
Adaptabilidad a los pines de comunicación serial Rx y Tx.	Uso de librería externa para la configuración de la comunicación serial.
Antena integrada en la placa del módulo. Se puede añadir una antena más potente.	Uso externo de <i>hardware</i> ( <i>router</i> ) para la comunicación serial.
Fácil manejo de sensores al aplicar una actualización de datos cada período de tiempo.	Requiere una fuente de voltaje externa por el alto consumo de corriente.
Cantidad moderada de pines	Voltaje de operación de 3,3 voltios

Fuente: elaboración propia.

### 3.1.3. **Despliegue de información**

Para la el despliegue de la información en la página web se ingresará en cualquier navegador web, la dirección IP que asignó el *router* al módulo. Se utilizará la dirección 192.168.50.150 y luego el módulo envía la orden de que el usuario está solicitando hacer uso de la página web.

Figura 42. **Despliegue de la información**



Fuente: elaboración propia, empleando Snipping Tool.

Desde el código HTML localizado en el microcontrolador se puede cambiar los textos que se despliegan dentro de la página *web*.

De esta manera se tiene una lectura de sensores la cual será tomada solo si el usuario desea obtener la información. En el momento que el usuario no haga uso de la página web, el microcontrolador estará en *stand by*, a la espera de que el código HTML sea solicitado para hacer el despliegue de la información.

## 4. MANEJO DE LOS ACCIONAMIENTOS ELECTROMAGNÉTICOS Y PANELES SOLARES

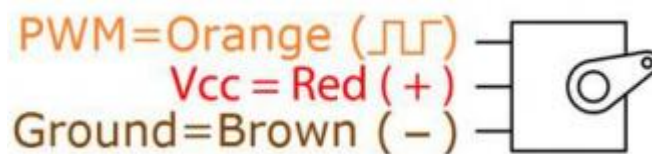
### 4.1. Implementación de Servo Motores.

#### 4.1.1. Servo Motor.

El control de los servomotores se hará mediante pulsos enviados por un microcontrolador, directamente a la entrada de la señal del motor. Los servomotores cuentan con tres entradas, Vin, GND, S, siendo el voltaje de entrada, tierra y señal de entrada respectivamente. Como los servomotores funcionan en base a grados, la señal que se enviará desde el microcontrolador representará cierta cantidad de grados que el motor debe girar en base a la señal recibida por unos sensores que tomarán los rayos del sol.

Dado que los puertos de salida del microcontrolador (ya sea Arduino o MSP432) son de baja potencia, se debe colocar una fuente externa que suministre el voltaje necesario para que los servomotores operen de manera correcta y el microcontrolador no sufra daños.

Figura 43. Entradas de un Servo Motor



Fuente: [http://www.electronicoscaldas.com/datasheet/MG996R\\_Tower-Pro.pdf](http://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf). Consulta: 15 de diciembre de 2017.

Se emplearán servomotores de 55 gramos ya que se moverá un panel solar que alimentará toda la estación meteorológica.

Tabla XII. **Datos MG996R**

Ítem	Valor
Voltaje de entrada	4,8v ~ 7,2v
Corriente máxima	500mA ~ 900mA
Corriente de parada	2,4 A
Entradas de control	1
Peso soportado	55 g
Rango de temperatura	0°C ~ 55°C

Fuente: [http://www.electronicoscaldas.com/datasheet/MG996R\\_Tower-Pro.pdf](http://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf). Consulta: 15 de diciembre de 2017.

Para poder controlar los servomotores se requieren señales enviadas desde el microcontrolador en forma de PWM, tomando en cuenta que para colocar el motor en la posición cero grados se requiere un pulso de 1,5 ms enviado, para 90 grados se envía un pulso de 2 ms de duración. Finalmente, para obtener -90 grados se utiliza un pulso de 1ms de duración. En base a esto se calcula la posición del servo motor y así calibrar el giro que tendrá dependiendo a la posición del sol.

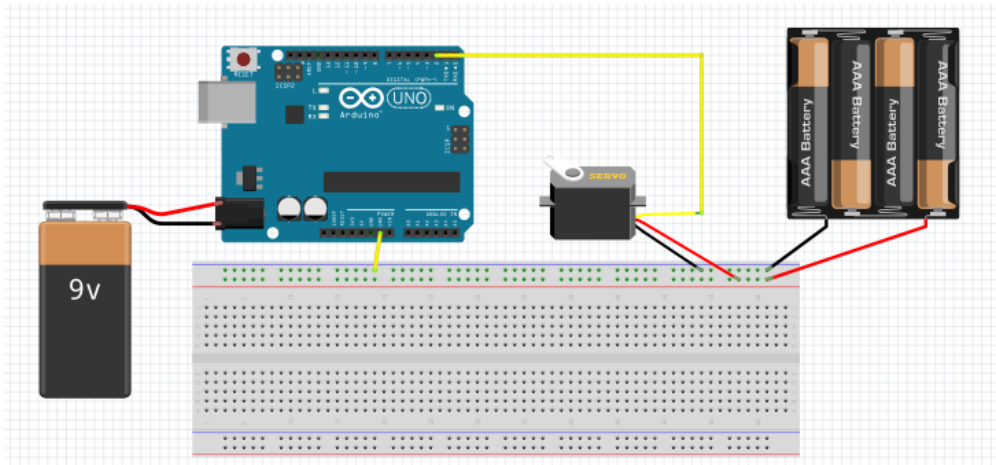
Figura 44. **Servo Motor MG996R**



Fuente: [http://www.hobbyandyou.com/content/images/thumbs/0002238\\_towerpro-mg996r-metal-gear-servo-motor.jpeg](http://www.hobbyandyou.com/content/images/thumbs/0002238_towerpro-mg996r-metal-gear-servo-motor.jpeg). Consulta: 04 de enero de 2018.

La conexión necesaria que se utilizará para manejar los motores es bastante sencilla, pero requiere una fuente de poder externa. Dado que el servo no soporta 12V, se aplicará 5V provenientes del panel solar y regulado para mantener el voltaje estable y no afectar el funcionamiento de los circuitos. En caso de no suministrar suficiente corriente, la alimentación externa podría ser sustituida por un juego de baterías AA para proporcionar 4,8V estables.

Figura 45. **Esquema de conexiones**



Fuente: elaboración propia, empleando Fritzing.

Se aplicará 9V al Arduino a manera de suministrar una fuente de voltaje estable que permita el buen funcionamiento del microcontrolador para manejar los servomotores.

Dado a que los sensores consumen muy poca corriente, el uso de servomotores no limita el funcionamiento de los demás circuitos, por lo que no habría interrupción de las mediciones realizadas. Se deberá conectar la tierra (GND), del microcontrolador y el servomotor, ya que, al compartir tierra se estabiliza la corriente y funciona de manera correcta, por tratarse del manejo de señales moduladas enviadas hacia el motor.

Para la programación de los servomotores se debe incluir una librería que permite el manejo de dichos equipos, facilitando así el ingreso de cuantos grados se deberá mover el motor y en qué dirección.



Figura 46. **Librería para Arduino**

```
#include <Servo.h>  
Servo Servo1;
```

Fuente: elaboración propia, empleando Snipping Tool.

Asimismo, se declara una variable llamada Servo1 que permitirá hacer uso de las funciones de la librería, ya que se almacena como un objeto y se llamará en cada renglón de uso del código para mover el motor. A su vez se indica el pin que se usará para la entrega de la señal por parte del microcontrolador mediante el comando Servo1.attach, es decir por donde saldrá la señal PWM para mover los grados al servomotor.

Figura 47. **Configuración del servo**

```
Servo1.attach(2);
```

Fuente: elaboración propia, empleando Snipping Tool.

Para una configuración inicial se usará un código que permite colocar el servomotor a un grado en específico, es decir ingresar de forma manual el valor al que se quiere posicionar el motor. Se aplicarán los valores de 0 grados, 90 grados y 180 grados, para obtener un barrido total de 180 grados. Esto ayudará a colocar el servomotor a la posición de 0 grados y añadirle los agarradores para sostener la estructura que llevará el panel solar.

Figura 48. **Barrido del servomotor**

```
Servol.write(0);  
delay(1000);  
Servol.write(90);  
delay(1000);  
Servol.write(180);  
delay(1000);
```

Fuente: elaboración propia, empleando Snipping Tool.

#### **4.1.1.1. Ventajas y desventajas**

La necesidad de aplicar un voltaje externo estable hace que sea una limitante para el uso de los servomotores. Sin embargo, la conexión adicional de la fuente no requiere mayor complejidad, por lo que es una desventaja que no afecta en su totalidad.

La mayor ventaja que presenta el uso de un servomotor es la precisión a la hora de girar los engranajes, ya que como es un dispositivo que se basa en grados para dar los giros necesarios. Esto se aplica mediante pulsos en ms según sea el caso. Con la librería para Arduino esto se simplifica aún más ya que se coloca el valor del grado deseado y ésta lo procesa para saber cuánto debe moverse el motor.

La configuración es bastante sencilla y ligera por lo que no hará uso excesivo de los recursos del microcontrolador, dejando suficiente memoria para cualquier otra implementación a futuro.

Dado que los servomotores se clasifican por la cantidad de torque que ejercen, se puede añadir un peso considerable al sistema que éste podrá moverlo sin problema, permitiendo así el seguimiento de los rayos solares de manera precisa y continúa.

#### **4.2. Paneles Solares**

La implementación de un panel solar tiene como fin proporcionar una fuente de voltaje de bajo impacto ambiental. Sin embargo, mucho de esto depende del estado del tiempo, con un día nublado no se obtendrá el mismo resultado que un día soleado.

Para aprovechar dicha energía se creará un circuito capaz de almacenar, potenciar y distribuir el voltaje obtenido por el panel solar. Esto se logrará mediante un módulo de carga de batería que recibe un pequeño voltaje, lo eleva para cargar una batería y luego permite la extracción de dicho voltaje para ser utilizado para alimentar los sensores, los microcontroladores y demás equipos dentro de la estación.

Figura 49. **Panel Solar 5V**



Fuente: [https://images-na.ssl-images-amazon.com/images/I/41IDD6BN3BL.\\_SY355\\_.jpg](https://images-na.ssl-images-amazon.com/images/I/41IDD6BN3BL._SY355_.jpg).

Consulta: 11 de enero de 2018.

#### **4.2.1. TP4056**

Con el módulo TP4056 se aprovechará la energía solar para poder cargar una batería 18650, y servirá como respaldo en caso de que las condiciones climáticas no permitan el uso de dichos paneles solares. Tiene incorporado un *PMOSFET* que hace innecesario el uso de un diodo protector para evitar corrientes de carga negativas y que existan daños a los sensores o microcontroladores del sistema.

Genera un voltaje de salida de 4,2V y se puede regular la corriente de salida mediante una resistencia, esto con el fin de respetar la regla de carga de una batería, que indica que se debe cargar a un décimo de la capacidad de la misma para no disminuir su tiempo de vida. Considerando esto, los tiempos de carga serán mayores a 10 horas.

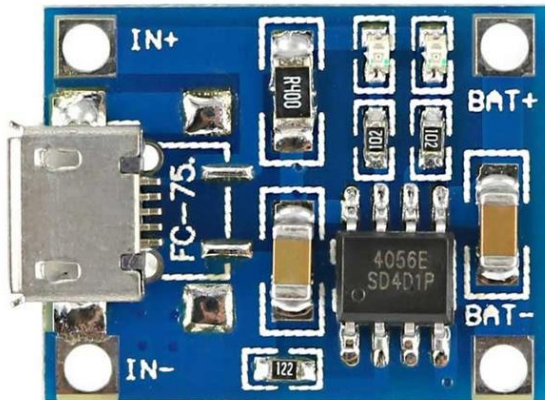
El módulo presenta las siguientes características de funcionamiento:

Tabla XIII. **Características del módulo TP4056**

Característica	Valor
V <sub>CC</sub>	4v ~ 8v
I <sub>CC</sub>	500µA
I <sub>BAT</sub>	450 mA ~ 1 050 mA
Temperatura de operación	-40°C ~ 85°C

Fuente: <https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf>. Consulta: 13 de enero de 2018.

Figura 50. **Módulo TP4056**



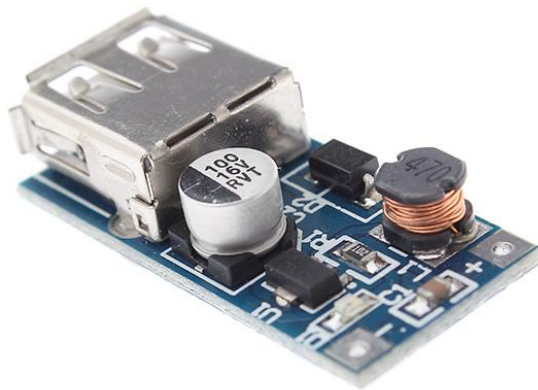
Fuente: [http://img.dxcdn.com/productimages/sku\\_373990\\_1.jpg](http://img.dxcdn.com/productimages/sku_373990_1.jpg). Consulta: 20 de enero de 2018.

#### 4.2.2. **BoostConverter**

Con este módulo se aumenta el voltaje a 5V para poder aplicarlo a un circuito, sensor o microcontrolador. La corriente almacenada en la batería se utiliza mediante el *boostconverter*, permitiendo tener una fuente de voltaje

continua y estable, para suministrar el amperaje necesario para el funcionamiento correcto de todo el sistema conectado. Cuenta con una salida USB estándar, y facilita la conexión de cualquier microcontrolador o dispositivo.

*Figura 51. Módulo BoostConverter*



Fuente: <https://img.fasttechcdn.com/121/1219201/1219201-5.jpg>. Consulta: 06 de febrero de 2018.

#### **4.2.3. Batería 18650**

Se usará una batería 18650 para almacenar la energía obtenida por el panel solar ya que se cuenta con un almacenamiento de corriente relativamente grande, siendo mayor a 2 500mAh (miliamperios hora). Con esto se logrará suministrar por un período de tiempo amplio, una corriente estable para todos los dispositivos conectados. En caso de una falta de energía solar (por malas condiciones climáticas) el uso de esta batería nos permite tener una fuente de voltaje externa lo suficientemente capaz de mantener activa toda la estación por un tiempo considerable.

Tabla XIV. **Características 18650**

Característica	Valor
Capacidad	Variable (2600mAh ~ 6000mAh)
Voltaje Nominal	3,7V
Impedancia Interna	$\leq 70\text{m}\Omega$
Voltaje Mximo de Carga	$4,20\text{V} \pm 0,05\text{V}$
Corriente de Carga Estndar	0,52A
Corriente de Carga Rpida	1,3A

Fuente: <https://www.ineltro.ch/media/downloads/SAAltem/45/45958/36e3e7f3-2049-4adb-a2a7-79c654d92915.pdf>. Consulta: 06 de febrero de 2018.

### 4.3. Circuito seguidor de luz solar

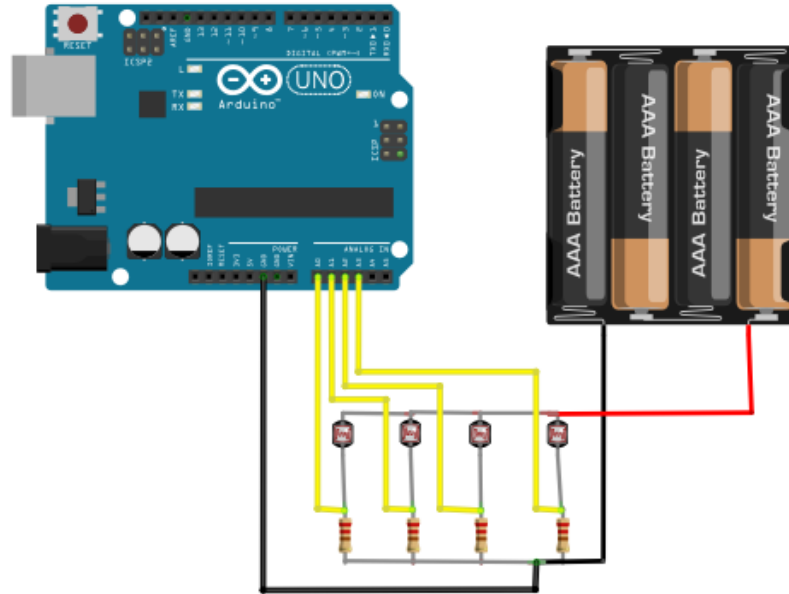
Dado a que, a cierta hora del da, un panel solar deja de recibir luz directa del sol, se pierde la eficiencia con la cual se podra contar para cargar unas bateras o alimentar el circuito completo. Por esta razn se utilizar un seguidor de luz solar, el cual consiste en crear un mecanismo capaz de mover un panel solar segn sea la trayectoria del sol. Esto se lograr mediante cuatro fotorresistencias que servirn para distinguir la direccin de los rayos solares, enviando valores anlogos a un microcontrolador, que procesar esos datos y tomar decisiones para mover los servomotores.

#### 4.3.1. Circuito por usar

Se colocarn dispositivos analgicos capaces de recibir la energa solar, como lo son las fotorresistencias. Las seales ingresarn por medio de las entradas anlogas del microcontrolador, luego sern procesadas por el ADC (*Analog-to-Digital Converter*) interno y en base a eso, el servomotor ser girado

hacia la posición correspondiente para obtener la mayor energía sobre el panel solar

Figura 52. **Circuito del Seguidor de Luz**



Fuente: <https://www.electronicshub.org/arduino-solar-tracker/> Consulta: 08 de febrero de 2018.

#### 4.3.2. **Funcionamiento**

Dado que se cuenta con cuatro fotorresistencias para ambos movimientos (horizontal y vertical) se calculará un promedio del valor obtenido por cada sensor y en base a esto, los actuadores recibirán una señal proveniente del microcontrolador. Primero se colocarán en una posición central, es decir, se configuran inicialmente para obtener una colocación de referencia.



Figura 53. **Configuración de los pines**

```
// LDR Conexión de Pines

int ldrlt = 0; //LDR Izquierda - Arriba
int ldr rt = 1; //LDR Derecha - Arriba
int ldrl d = 2; //LDR Izquierda - Abajo
int ldrr d = 3; //ldr Derecha - Abajo
```

Fuente: <http://www.instructables.com/id/Dual-Axis-300W-IOT-Solar-Tracker/>. Consulta: 12 de febrero de 2018.

Se declararán cuatro pines de entrada analógica, uno para cada fotorresistencia y se leerá el valor que ingresa a dichos puertos. Luego se convertirá mediante la función de ADC.

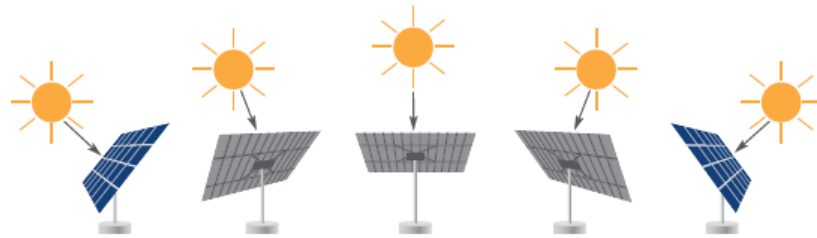
Figura 54. **Lectura Analógica**

```
int lt = analogRead(ldrlt);
int rt = analogRead(ldrrt);
int ld = analogRead(ldrld);
int rd = analogRead(ldrrd);
```

Fuente: <http://www.instructables.com/id/Dual-Axis-300W-IOT-Solar-Tracker/> Consulta: 13 de febrero de 2018.

Según sean los datos obtenidos, se enviarán señales a los servomotores para que actúen de manera horizontal y vertical. Con esto se logran que las fotorresistencias siempre que estén de frente al sol, y el panel recibirá la mayor cantidad de energía solar, generando un mayor voltaje y proporcionando una mejor estabilidad para consumo de la estación meteorológica.

Figura 55. **Seguimiento solar**



Fuente: [http://www.npspower.com/wp-content/uploads/2015/09/solar\\_tracker\\_movment.png](http://www.npspower.com/wp-content/uploads/2015/09/solar_tracker_movment.png).

Consulta: 13 de febrero de 2018.

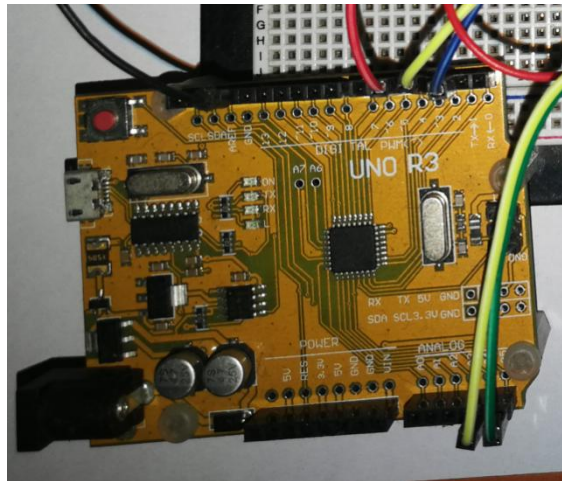
## 5. PRESENTACIÓN DEL PROYECTO

### 5.1. Prototipo de estación meteorológica

#### 5.1.1. Pruebas iniciales

Las primeras pruebas se realizarán en un *protoboard* o placa de pruebas y se harán sensor por sensor de manera independiente. Luego se unirán los tres sensores al microcontrolador para ver la cantidad de memoria que consume las instrucciones para controlar cada sensor. Mediante un USB capaz de medir corriente, se tomarán datos del comportamiento del sistema y cuanta potencia consume dicho circuito.

Figura 56. Microcontrolador Arduino UNO R3



Fuente: elaboración propia, empleando Snipping Tool.

El medidor de corriente ayudará a analizar el comportamiento del equipo completo, ya que como se mencionó anteriormente, se desea usar una fuente renovable como lo es la energía solar.

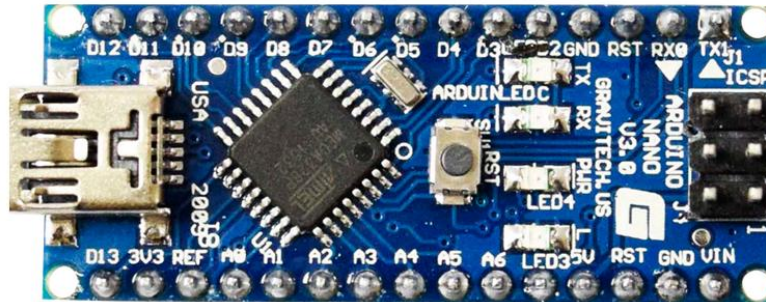
Figura 57. **USB con medidor de corriente**



Fuente: elaboración propia, empleando Snipping Tool.

Tomando en cuenta el consumo de corriente de los servomotores, se empleará un microcontrolador aparte. Esto beneficiará también a tener un margen de memoria más amplia y no perder estabilidad a la hora de ejecutar el programa principal. Por lo tanto, se hará uso del Arduino Nano, una versión reducida del microcontrolador Arduino Uno.

Figura 58. **Arduino Nano**



Fuente: [https://www.makerlab-electronics.com/my\\_uploads/2016/08/arduino-nano-1.jpg](https://www.makerlab-electronics.com/my_uploads/2016/08/arduino-nano-1.jpg).

Consulta: 18 de febrero de 2018.

Como los sensores y el equipo se mantendrán en un ambiente externo y recibirán precipitaciones, se colocarán en recipientes herméticos, que no permitirán el ingreso de agua, para no afectar o dañar totalmente los dispositivos.

### **5.1.2. Organización de los sensores**

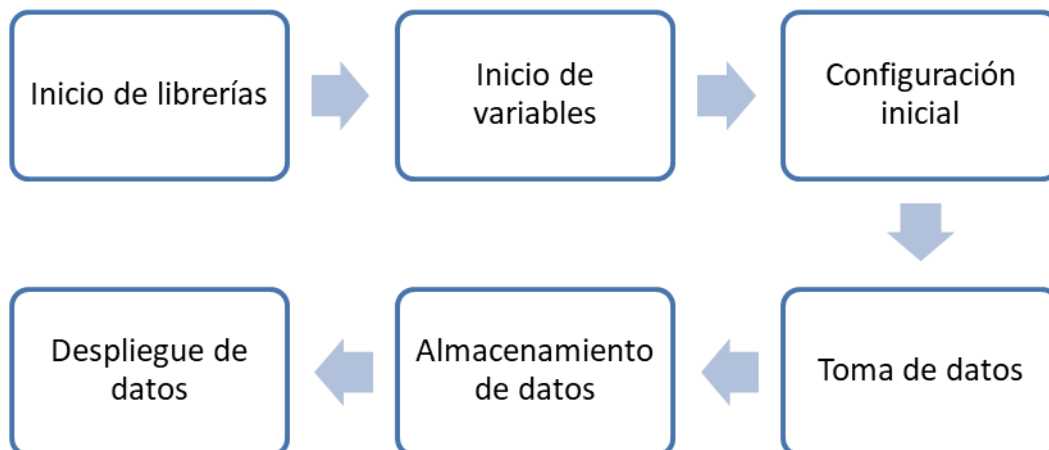
El proceso de la toma de datos debe llevar un orden específico para poder desplegar los datos de manera correcta. Tomando en cuenta los tres sensores a utilizar, el proceso sería el siguiente:

- DHT11
- BMP180
- Sensor de precipitación

Una vez se obtienen los datos de los tres sensores, se procederá al despliegue y almacenamiento de los datos de manera final. Mediante el uso del módulo WiFi se logrará transmitir de manera inalámbrica, la información obtenida en el período de tiempo censado.

El flujo de trabajo de los circuitos viene siendo un ciclo definido por el período que el usuario desee que los sensores trabajen y en base a esto se repite el proceso de funcionamiento, creando una toma de datos sistemática y progresiva. Una vez se inicializan las librerías, se declaran las variables y se configura de manera inicial, los sensores quedan en un bucle de toma de datos.

Figura 59. **Proceso finito de toma de datos**



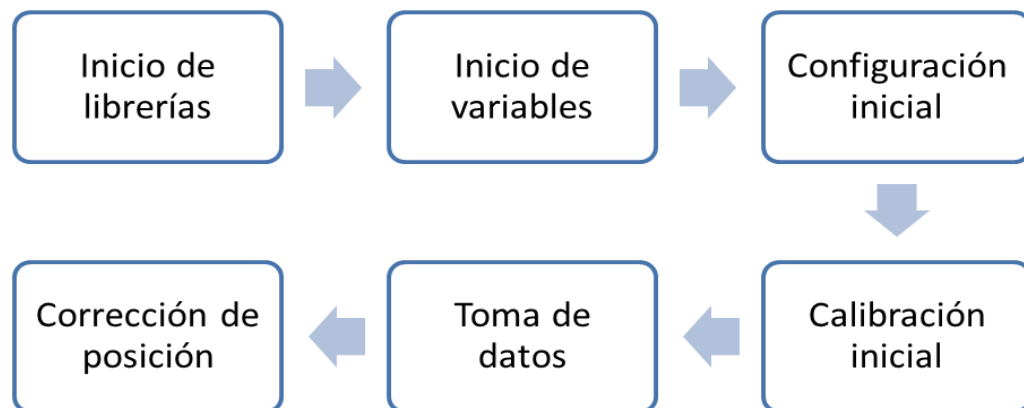
Fuente: elaboración propia.

### 5.1.3. Organización de los servomotores

Con el movimiento del panel solar se colocará un microcontrolador distinto al usado para el programa principal. Esto con el objetivo de mantener estable la ejecución del código fuente, sin tener pérdidas de información o inestabilidad del microcontrolador y así obtener el mejor posicionamiento para el panel solar, almacenando así, la mayor cantidad de energía solar.

El proceso del seguimiento de la luz solar consta de la inicialización de librerías y variables que almacenarán los datos obtenidos. Luego se pasará a la etapa de configuración inicial, la cual permite activar el uso de los servomotores. Por último se tomarán los datos y se procederá al movimiento de los motores.

Figura 60. **Proceso de seguimiento de luz solar**



Fuente: elaboración propia.

## **5.2. Comportamiento de los dispositivos**

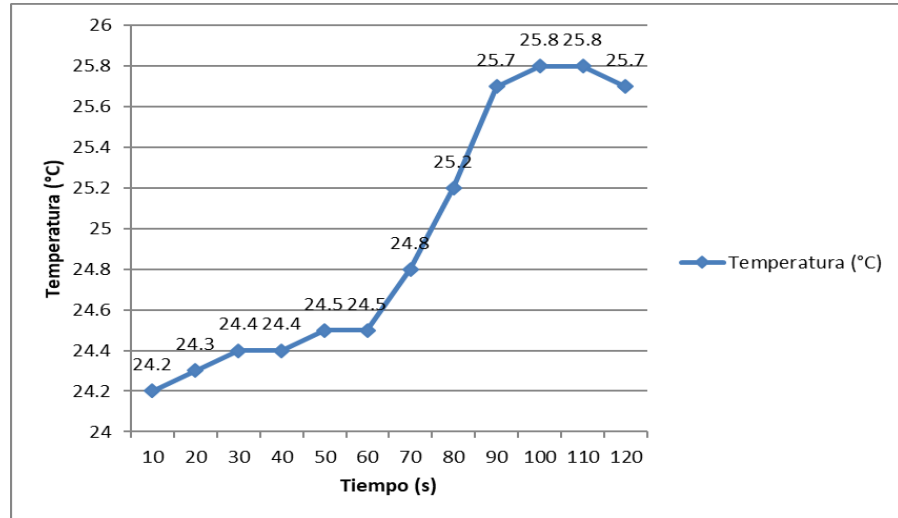
Para tener un funcionamiento correcto, cada dispositivo deberá trabajar a una temperatura adecuada respecto al tiempo de operación. Con esto, se garantiza que los sensores y microcontroladores funcionarán de manera correcta luego de horas de su puesta en marcha, ya que su objetivo es permanecer encendidos por períodos extensos de tiempo dentro de recipientes herméticos.

Si se presentan temperaturas elevadas en los dispositivos, sería necesario presentar dos soluciones. Una de ellas sería colocar un ventilador capaz de enfriar dichos dispositivos, ya sea mediante la extracción de calor o el envío de aire fresco a los sensores y microcontroladores. La otra consiste en colocar disipadores de calor pero sería complicado ya que los dispositivos no cuentan con espacio para dicho elemento.

Para el arranque del microcontrolador, la temperatura varía en los primeros 120 segundos, ya que los sensores comienzan a tomar datos y a consumir más corriente de los puertos GPIO del microcontrolador. Luego la temperatura se normaliza a un valor cercano a los 28 °C en su funcionamiento normal, cuando al iniciar operaciones el valor de la temperatura era de 23,1 °C.

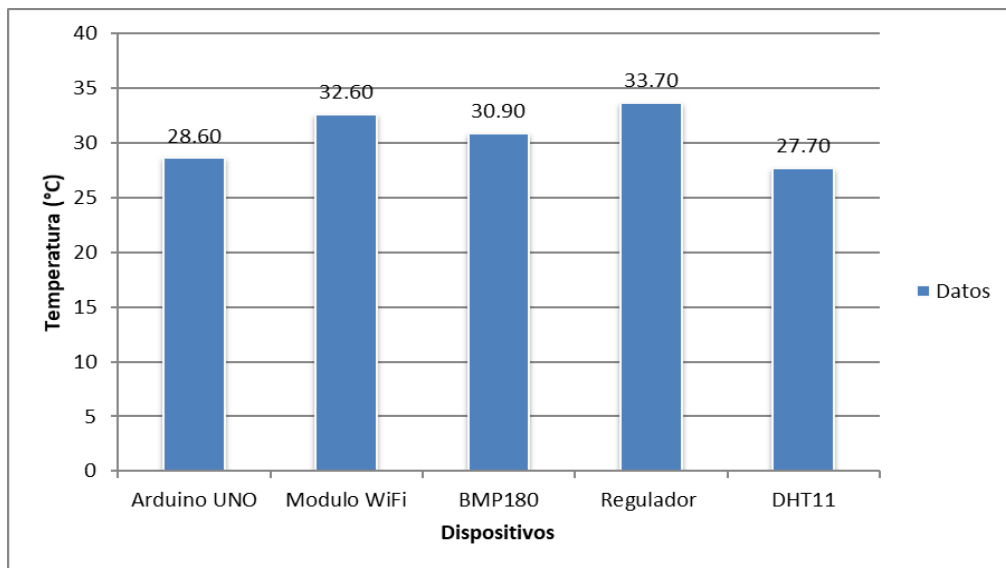


Figura 61. Gráfico temperatura vs tiempo



Fuente: elaboración propia.

Figura 62. Temperatura de dispositivos



Fuente: elaboración propia.

### **5.3. Ventajas y desventajas**

#### **5.3.1. Ventajas**

Una de las ventajas más notables es la fácil conexión de los sensores y el microcontrolador, ya que no lleva ningún buffer o acoplador entre dichos dispositivos, permitiendo así obtener los valores sin necesidad de circuitería externa. A su vez, cada sensor cuenta con una librería definida para el funcionamiento correcto y para que la configuración inicial se pueda aplicar y el microcontrolador logre obtener y almacenar las variables provenientes de dichos sensores.

La utilización de un módulo WiFi da suficiente rango de envío de datos y mejora la forma en que la información es desplegada, siendo ésta, amigable al usuario final, sin necesidad de caracteres incongruentes y con un bajo porcentaje de pérdidas de información en el momento de la transmisión.

De las ventajas más notorias se encuentran:

- Fácil conexión entre módulos y microcontrolador
- Bajo consumo de corriente por sensor
- Interacción amigable entre usuario y máquina
- Alto grado de precisión entre los datos obtenidos
- Sensores compactos
- No hay necesidad de acopladores o buffers
- Amplio rango de recepción de señal en el módulo WiFi

### 5.3.2. Desventajas

El despliegue de la información presenta complicaciones por ser los valores dispersos entre cada sensor, por lo que a cada variable se le debe asignar el tipo correcto de valor a almacenar y el espacio adecuado en la memoria, creando así la necesidad de declarar cada variable de manera correcta para el buen funcionamiento de cada sensor.

Dada la poca capacidad de almacenamiento en el microcontrolador Arduino UNO, el circuito completo presenta un porcentaje elevado de uso de memoria, provocando inestabilidad y sobre carga de datos, pero aun siendo funcional a la hora del despliegue de información sobre la página web creada.

El uso de un módulo WiFi crea la dependencia de un *router* inalámbrico que tenga la suficiente potencia para transmitir la señal hasta el punto donde se colocará el conjunto de sensores. Esto lleva consigo la asignación de una IP estática para no perder el despliegue de la información enviada a través de la señal y permitiendo así acceder a la página web donde se mostrará dicha información.

Entre las desventajas presentadas se cuenta con las siguientes:

- Uso elevado de memoria del microcontrolador
- Necesidad de un sistema externo para controlar los motores
- Utilización de IP estática para el módulo WiFi
- Colocación del sistema en el rango de la señal del *router* inalámbrico
- Obstáculos debilitan la señal recibida por el módulo WiFi
- Necesidad de un día soleado para obtener la energía suficiente para la recarga de la batería externa.

## 5.4. Mejoras a implementar

### 5.4.1. Mejoras a la comunicación

#### 5.4.1.1. NodMCU ESP8266MOD

Para una mejor recepción del módulo WiFi, se planea la implementación de un módulo mejorado del ESP8266, que permitirá mejores características para el despliegue de información. También con la posibilidad de eliminar el uso de un microcontrolador externo, ya que éste cuenta con un pequeño dispositivo montado en la placa. Con esto se obtiene la fusión del microcontrolador junto con el módulo WiFi.

Figura 63. **Módulo NodMCU ESP8266MOD**

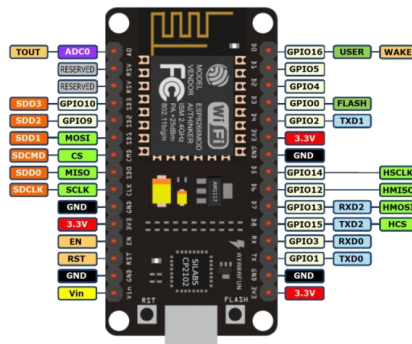


Fuente: <http://i.imgur.com/l9bpiTf.jpg>. Consulta: 22 de febrero de 2018.

Con este módulo se puede configurar un proceso de toma de datos similar al que se hará con los microcontroladores MSP432 y Arduino UNO, ya que cuenta con un pin para la conversión Análoga-Digital, pines de uso general (GPIO), y los pines necesarios para la conexión del sensor BMP180, los cuales son SDA y SCL. También cuenta con la capacidad de recepción y transmisión

de señales WiFi, con una antena integrada, por ese motivo el rango de señal es aceptable.

Figura 64. NodMCU ESP8266MOD



Fuente: [https://bennthomsen.files.wordpress.com/2015/12/nodemcu\\_pinout\\_700-2.png](https://bennthomsen.files.wordpress.com/2015/12/nodemcu_pinout_700-2.png).

Consulta: 23 de febrero de 2018.

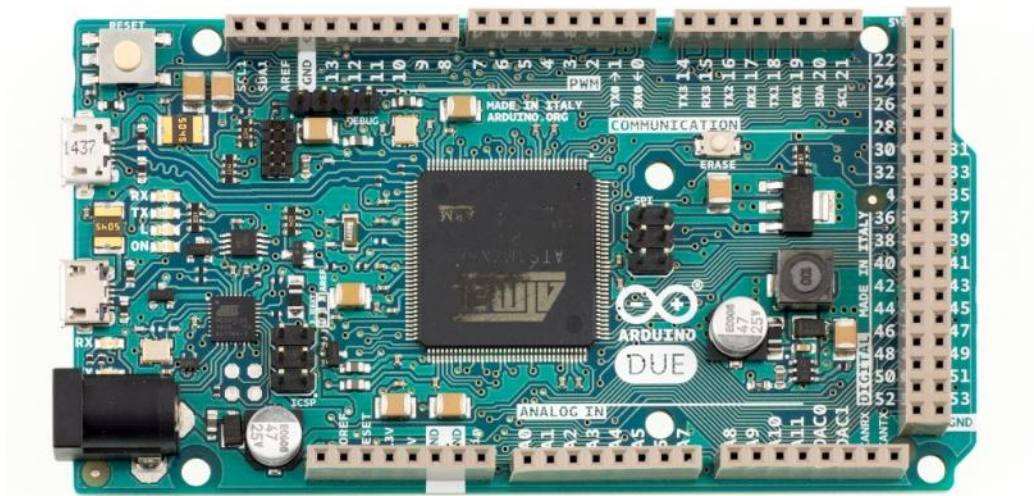
#### 5.4.1.2. Mejoras al procesamiento

Entre las mejoras a realizar, está la implementación de un microcontrolador más capacitado para el manejo de los datos, con mayor almacenamiento para las variables y con una frecuencia de procesamiento más elevada para una realización de instrucciones en un ciclo más corto. La implementación un microcontrolador con mayor capacidad de memoria permitirá unir el programa del seguimiento de luz solar con el de toma de datos de los sensores, puesto que se contará con mayor almacenamiento en las memorias del dispositivo.

Se plantean dos microcontroladores para mejorar el sistema, estos son:

- Arduino Due:
  - 3,3 v de voltaje de operación
  - 512 kb de memoria para el usuario
  - 84 MHz de frecuencia de procesamiento
  - 54 pines de uso general
  - 130 mA de corriente tolerable en cada puerto

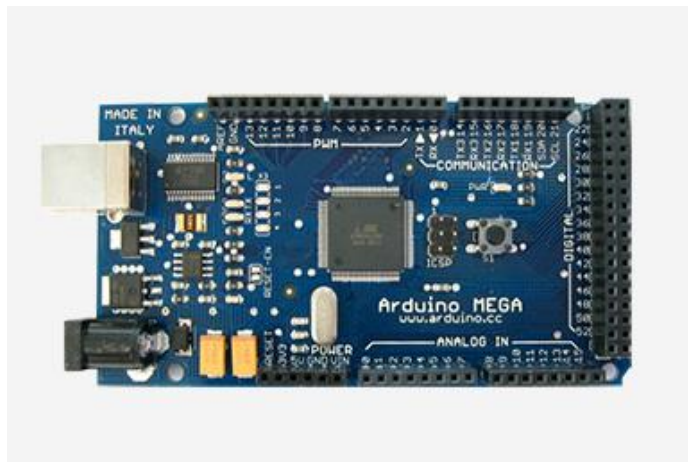
Figura 65. **Arduino DUE**



Fuente: [https://cdn.electronza.com/wp-content/uploads/2016/02/arduino\\_due.jpg](https://cdn.electronza.com/wp-content/uploads/2016/02/arduino_due.jpg). Consulta: 24 de febrero de 2018.

- Arduino Mega:
  - 5 v de voltaje de operación
  - 54 puertos de uso general
  - 128 kb de memoria para la aplicación
  - 16 MHz de frecuencia de procesamiento

Figura 66. **Arduino MEGA**



Fuente: <https://www.arduino.cc/en/uploads/Main/ArduinoMega.jpg> Consulta: 26 de febrero de 2018.





## CONCLUSIONES

1. Mediante el diseño de una estación meteorológica de fácil acceso se logró monitorear el clima en una región de interés, permitiendo así obtener datos en tiempo real.
2. Con la limitante en el almacenamiento de los microcontroladores al momento de cargar el conjunto de instrucciones programables, se necesitaron dos equipos para poder monitorear los sensores, administrar el seguimiento solar y muestrear los datos.
3. La tecnología WiFi permite que el rango de comunicación sea mayor, y los sensores pueden estar al aire libre enviando y recibiendo datos, con una falla de señal muy baja.
4. Por medio del rápido procesamiento de los dispositivos, se logró analizar los datos obtenidos para una región delimitada, y los fenómenos que afectan a la misma.



## RECOMENDACIONES

1. Instalar la estación meteorológica con accionamientos electromecánicos para monitorear el clima de una región en tiempo real, sin la necesidad de estar ubicados en el lugar de estudios, facilitando así los procesos en el ámbito de la investigación.
2. Implementar un microcontrolador más completo, que tenga un almacenamiento mayor, para los juegos de instrucciones programables y así reducir el número de dispositivos en la estación.
3. Utilizar tecnologías inalámbricas que permiten un rango de comunicación mayor, y una menor cantidad de errores en la recopilación y transmisión de datos.
4. Emplear un procesador cuya frecuencia sea mayor a 800 MHz. para poder analizar los datos obtenidos de una manera óptima.



## BIBLIOGRAFÍA

1. ALBUQUERQUE FONSECA, Igor. *Solar Charged Battery Powered Arduino Uno*. 2016. [en línea].  
<<https://create.arduino.cc/projecthub/igorF2/solar-charged-battery-powered-arduino-uno-645d89>>. [Consulta: 22 de febrero de 2017].
2. Anónimo *ESP8266 Web Server with HTML Web Page*. 2016 [en línea].  
<<https://circuits4you.com/2016/12/16/esp8266-web-server-html/>>. [Consulta: 8 de noviembre de 2017].
3. Anónimo. *Arduino Solar Tracker*. 2016. [en línea].  
<<https://www.electronicshub.org/arduino-solar-tracker/>>. [Consulta: 26 de febrero de 2017].
4. Anónimo. *Bluetooth en Arduino*. [en línea].  
<<https://aprendiendoarduino.wordpress.com/tag/hc-05/>>. [Consulta: 8 de noviembre de 2017].
5. Anónimo. *ESP8266: Understanding Web Server programming with Arduino code*. 2017 [en línea]. <<https://diyprojects.io/esp8266-web-server-tutorial-create-html-interface-connected-object/#.Wpd1yOjOXDd>>. [Consulta: 8 de noviembre de 2017].

6. Anónimo. *HOW MUCH ENERGY CAN A SOLAR PANEL PRODUCE IN A DAY?* Hopkinton, NH. [en línea].  
<<http://store.sundancesolar.com/how-much-energy-can-a-solar-panel-produce-in-a-day-1/>>. [Consulta: 20 de agosto de 2017].
7. Anónimo. *SENSORES DE TEMPERATURA DHT11*. México. 2017. [en línea]. <<https://www.prometec.net/sensores-dht11/>>. [Consulta: 3 de agosto de 2017].
8. BrownDogGadgets. *Simple dual axis solar tracker*. 2015 [en línea].  
<<http://www.instructables.com/id/Simple-Dual-Axis-Solar-Tracker/>>. [Consulta: 15 de julio de 2017].
9. CEMAER. *Panel Solar, ¿Cómo Calcular Cuánta Energía Produce?* 2013 [en línea].  
<<http://www.gstriatum.com/energiasolar/blog/2013/03/17/panel-solar-como-calcular-cuanta-energia-produce/>>. [Consulta: 7 de julio de 2017].
10. Deba168. *Solar Powered Arduino Weather Station*. 2014. [en línea].  
<<http://www.instructables.com/id/SOLAR-POWERED-ARDUINO-WEATHER-STATION/>>. [Consulta: 15 de febrero de 2017].
11. DOVEP. *SOLAR BATTERY CHARGER*. 2006 [en línea].  
<<http://www.instructables.com/id/Solar-Battery-Charging/>>. [Consulta: 18 de agosto de 2017].

12. DURÁN ROCHA, Andrés. *Bluetooth HC-06 y HC-05 Android Arduino*. 2015 [en línea]. <<https://hetpro-store.com/TUTORIALES/bluetooth-hc-06-app-arduino/>>. [Consulta: 23 de septiembre de 2017].
13. ESPRESSIF SYSTEMS. *ESP8266 AT Instruction Set* [en línea]. <[https://www.espressif.com/sites/default/files/documentation/4a-esp8266\\_at\\_instruction\\_set\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/4a-esp8266_at_instruction_set_en.pdf)>. [Consulta: 5 de enero de 2017].
14. ESPRESSIF SYSTEMS. *ESPRESSIF SMART CONNECTIVITY PLATFORM: ESP8266*. 2013. [en línea]. <[https://nurdspace.nl/images/e/e0/ESP8266\\_Specifications\\_English.pdf](https://nurdspace.nl/images/e/e0/ESP8266_Specifications_English.pdf)>. [Consulta: 10 de enero de 2017].
15. GEO, Bruce. *Arduino Solar Tracker*. 2011 [en línea]. <<http://www.instructables.com/id/Arduino-Solar-Tracker/>>. [Consulta: 12 de julio de 2017].
16. GONZÁLEZ GARCÍA, Antony. *Módulo YL-83: Un Detector de lluvia*. Panamá. 2014 [en línea]. <<http://panamahitek.com/modulo-yl-83-un-detector-de-lluvia/>>. [Consulta: 5 de septiembre de 2017].
17. JOYSON96. *Simple webserver using Arduino and ESP8266*. 2016 [en línea]. <<http://www.instructables.com/id/Simple-Webserver-Using-Arduino-UNOMega-and-ESP8266/>>. [Consulta: 8 de noviembre de 2017].

18. Lasmonedasdejudas. *Construye un cargador de batería solar para Arduino*. 2017. [en línea].  
<<https://lasmonedasdejudas.wordpress.com/2017/02/10/construye-un-cargador-de-bateria-solar-para-arduino/>>. [Consulta: 23 de febrero de 2017].
19. LLORENTE MARTÍNEZ, Fernando. *Información meteorológica tablas* [en línea]. <<http://www.rumtor.com/tablas.html>>. [Consulta: 1 de agosto de 2017].
20. Naylam. *Tutorial básico NRF24L01 con Arduino*. 2016 [en línea].  
<[http://www.naylampmechatronics.com/blog/16\\_Tutorial-b%C3%A1sico-NRF24L01-con-Arduino.html](http://www.naylampmechatronics.com/blog/16_Tutorial-b%C3%A1sico-NRF24L01-con-Arduino.html)>. [Consulta: 25 de julio de 2017].
21. Naylam. *Tutorial sensor de presión barométrica BMP180*. Trujillo, Perú, 2017 [en línea].  
<[http://www.naylampmechatronics.com/blog/43\\_Tutorial-sensor-de-presi%C3%B3n-barom%C3%A9trica-BMP180.html](http://www.naylampmechatronics.com/blog/43_Tutorial-sensor-de-presi%C3%B3n-barom%C3%A9trica-BMP180.html)>. [Consulta: 10 de julio de 2017].
22. PROMOTEC. *Arduino y WiFi ESP8266*. México. 2017 [en línea].  
<<https://www.promotec.net/arduino-wifi/>>. [Consulta: 10 de noviembre de 2017].
23. pr-ser-uj. *Servidor WEB con el ESP8266 y Arduino*. 2016 [en línea].  
<<https://hetpro-store.com/TUTORIALES/esp8266-arduino-crear-servidor-web/>>. [Consulta: 10 de noviembre de 2017].

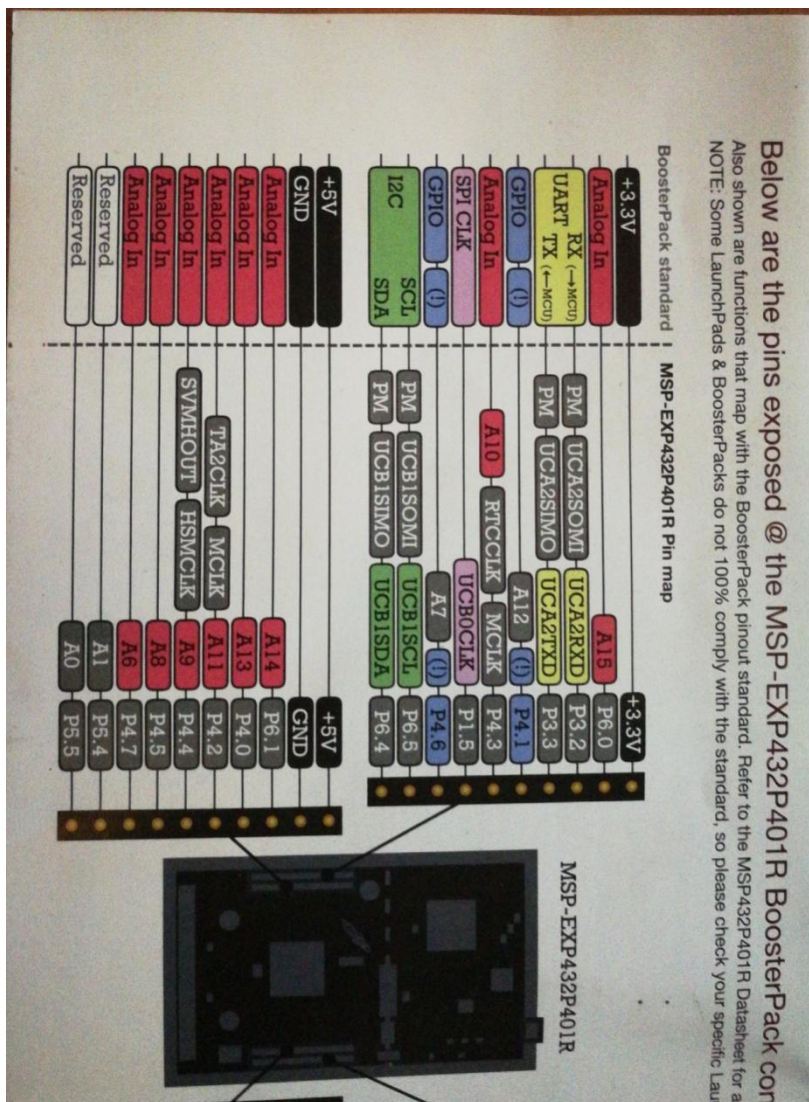


24. RIERA SALÍS, José Manuel. *Regulación Técnica de los sistemas Wi-Fi*. España. 2008. [en línea]. <[https://www.rediris.es/eventos/foro-wifi/2008/wifi\\_salud\\_2.pdf](https://www.rediris.es/eventos/foro-wifi/2008/wifi_salud_2.pdf)>. [Consulta: 10 de febrero de 2017].
25. TerryKing. *DHT11-Humidity-TempSensor*. San Francisco, CA, 2010 [en línea].  
<<https://arduino-info.wikispaces.com/DHT11-Humidity-TempSensor>>. [Consulta: 10 de julio de 2017].
26. URIARTE, Ivan. *Servidor WEB con WiFi ESP8266*. México. 2017 [en línea].  
<<https://www.prometec.net/servidor-web-esp8266/#>>. [Consulta: 20 de enero de 2017].



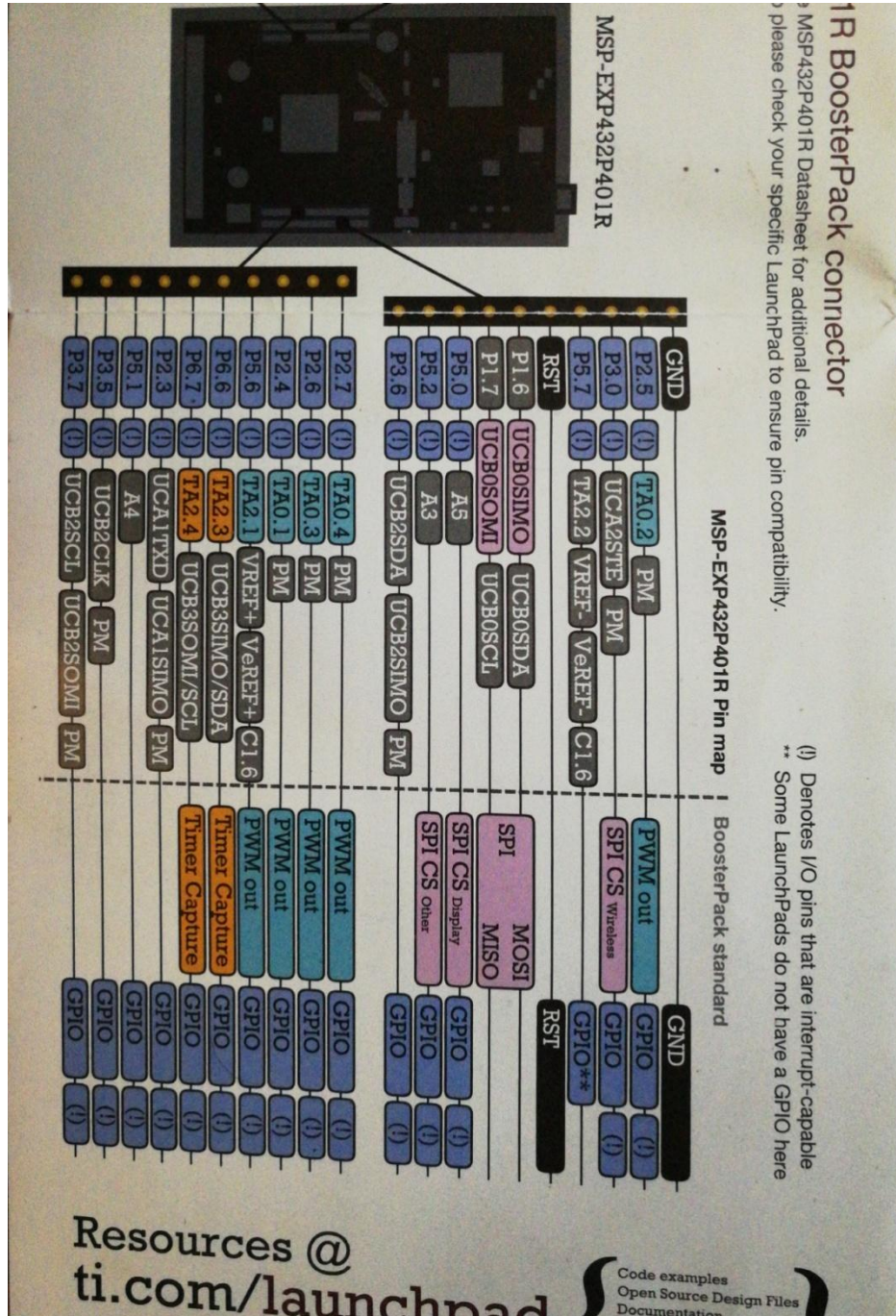
# APÉNDICES

## Apéndice 1. Puertos de uso general de microcontrolador MSP432, incluyendo registros



Fuente: elaboración propia.

## Apéndice 2. Complemento de puertos del microcontrolador MSP432



Fuente: elaboración propia.