

**UNIVERSIDAD NACIONAL DEL CALLAO**  
**UNIDAD DE INVESTIGACIÓN DE LA FACULTAD DE**  
**INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**



**“SISTEMA DE ACONDICIONAMIENTO DE SEÑALES USANDO  
MCU PARA MEJORAR LA PRECISIÓN DE LAS VARIABLES  
OBTENIDAS POR UNA ESTACIÓN METEOROLÓGICA  
AUTOMÁTICA DE BAJO COSTO”**

**AUTOR: RUBIÑOS JIMENEZ, SANTIAGO LINDER**

**DOCENTE COLABORADOR:**  
Mg. Ing. NIKO ALAIN ALARCON CUEVA

**ESTUDIANTES DE APOYO:**  
Est. JOSHUA ANDERSSON MAYTA TABOADA  
Est. KEVIN LEONCIO SEDANO GONZÁLEZ

(PERIODO DE EJECUCIÓN: Del 01 de febrero de 2021 al 31 de enero de  
2022)

(Resolución de aprobación N° 118-2021-R)

**Callao, 2022**

A handwritten signature in black ink, appearing to be "G. J. ...".

NH-10

## **DEDICATORIA**

A Dios, por ser el que guía mis pasos cada día, a mi Madre e hijas, por su constante apoyo para que luche por mis objetivos.



Nelly C

## AGRADECIMIENTO

Agradezco a Dios por permitirme seguir y lograr mis objetivos, a la Universidad Nacional del Callao por ser la que me enseñó a trazar metas y gracias a mi familia por apoyarme en cada decisión y proyecto.

  
Nelly C

## ÍNDICE

<b>RESUMEN</b>	11
<b>ABSTRAC</b>	12
<b>INTRODUCCION</b>	13
<b>CAPITULO I: PLANTEAMIENTO DEL PROBLEMA</b>	14
1.1 Descripción de la realidad problemática	14
1.2 Formulación del problema	15
1.3 Objetivos (general y específicos)	15
1.4 Limitantes de la investigación (teórico, temporal, espacial)	16
<b>CAPITULO II: MARCO TEÓRICO</b>	18
2.1 Antecedentes (internacional y nacional)	18
2.2 Marco:	22
2.2.1 Teórico	22
2.2.2 Conceptual	25
2.3 Definición de términos básicos.	39
<b>CAPITULO III: HIPÓTESIS Y VARIABLES</b>	41
3.1 Hipótesis	41
3.2 Definición conceptual de variables	41
3.3 Operacionalización de variables	42
<b>CAPITULO IV: DISEÑO METODOLÓGICO</b>	44
4.1 Tipo y diseño de la investigación	44
4.2 Método de investigación	45



4.3 Población y muestra	59
4.4 Lugar de estudio y periodo desarrollado	60
4.5 Técnicas e instrumentos para la recolección de la información	60
4.6 Análisis y procesamiento de datos	60
<b>CAPITULO V: RESULTADOS</b>	61
5.1 Resultados descriptivos	61
5.2 Resultados inferenciales	61
<b>CAPITULO VI: DISCUSIÓN DE RESULTADOS</b>	66
6.1 Contrastación y demostración de la hipótesis con los resultados	66
<b>CONCLUSIONES</b>	67
<b>RECOMENDACIONES</b>	67
<b>REFERENCIAS BIBLIOGRÁFICAS</b>	68
<b>ANEXOS</b>	<b>70</b>
ANEXO N°1: MATRIZ DE CONSISTENCIA	71
ANEXO N°2: DATASHEET DE ARDUINO MEGA	73
ANEXO N°3: DATASHEET DE LM35	78
ANEXO N°4: DATASHEET DE DHT22	81



Handwritten signature and initials.

## TABLA DE FIGURAS

Figura N°1: Sensor TMP36	26
Figura N°2: Curva característica del sensor TMP35 y TMP36	26
Figura N°3: Circuito para medir la temperatura del sensor LM35	27
Figura N°4: Sensor DHT22	27
Figura N°5: Conexión directa del Sensor DTH22 al Arduino Mega	28
Figura N°6: Sensor modelo RK400-04	29
Figura N°7: Características del Sensor RK400-04	29
Figura N°8: Funcionamiento del pluviómetro	30
Figura N°9: Sensor Anemo4403	31
Figura N°10: Características del Anemo4403	31
Figura N°11 Funcionamiento del anemómetro	32
Figura N°12: Sensor EPD-440-0-1.4	32
Figura N°13: Características del Sensor EPD-440-0-1.4	33
Figura N°14: Diagrama esquemático de las partes principales del medidor de radiación	33
Figura N°15: Conversor ADC ASD1115	34
Figura N°16: Diagrama interno del ASD1115	35
Figura N°17: Partes de un Arduino Mega	36
Figura N°18: Módulo Wifi ESP8266	37
Figura N°19: Diagrama interno del módulo ESP8266	38
Figura N°20: Comunicación entre el Arduino y una PC	



mediante ThingSpeak	39
Figura N°21. Diagrama de bloques del sistema	45
Figura N°22. Tarjeta de Desarrollo Arduino Mega	46
Figura N°23. ESP8266	46
Figura N°24. Diagrama de bloques del ESP8266	47
Figura N°25. Acondicionamiento de Señal del pluviómetro	48
Figura N°26. Acondicionamiento de señal del anemómetro	48
Figura N°27. Simulación del DHT22 y LM35	49
Figura N°28. Prueba de comunicación entre Simulink y Arduino	49
Figura N°29. Configuración de Hardware y puerto serial en Simulink	50
Figura N°30. Demostración de comunicación entre Simulink y Arduino	50
Figura N°31: Diagrama de conexión de ESP8266 y Arduino	51
Figura N°32. Comunicación Serial entre Arduino y NodeMCU	51
Figura N°33: Diagrama de bloques de Simulink	52
Figura N°34: Testeo de comunicación entre Arduino, Simulink y el NodeMCU	53
Figura N°35. Envío de números a través del puerto serial desde el Arduino al NodeMCU.	55
Figura N°36. Selección de variable a usar en Ubidots	55
Figura N°37. Obtención de Ubidots token a través de su página	56
Figura N°38. Comunicación del NodeMCU con la página Ubidots	58
Figura N°39. Comunicación del sensor de temperatura y Ubidots	58
Figura N°40. Comunicación del sensor de DTH22 y Ubidots	59

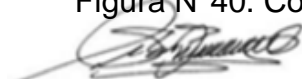
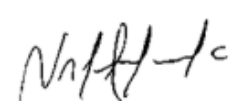

  




Figura N°41. Comunicación del sensor de UV y Ubidots	59
Figura N°42: Gráfica de Temperatura	62
Figura N°43: Gráfica de Humedad	63
Figura N°44: Gráfica de Índice UV	64
Figura N°45: Gráfica de Precipitación	65

  
Núñez

## RESUMEN

El presente trabajo de investigación contiene información acerca de el diseño de un SISTEMA DE ACONDICIONAMIENTO DE SEÑALES USANDO MCU PARA MEJORAR LA PRECISIÓN DE LAS VARIABLES OBTENIDAS POR UNA ESTACIÓN METEOROLÓGICA AUTOMÁTICA DE BAJO COSTO por medio del uso de sensores y actuadores electrónicos de poco costo en el mercado y misma eficacia al momento de obtener los parámetros deseados del sistema. En el contenido del texto se presenta la problemática, los objetivos, antecedentes relativos al estudio, así como las hipótesis a contrastar en los resultados. De igual forma contiene el diseño del sistema tanto en hardware como software.

Palabras clave: señales, mcu, meteorológica, bajo costo



Handwritten signature and initials.

## ABSTRACT

This research work contains information about the design of a SIGNAL CONDITIONING SYSTEM USING MCU TO IMPROVE THE ACCURACY OF THE VARIABLES OBTAINED BY A LOW-COST AUTOMATIC WEATHER STATION through the use of low-cost electronic sensors and actuators in the market and same efficiency at the time of obtaining the desired parameters of the system. The content of the text presents the problem, the objectives, the antecedents related to the study, as well as the hypotheses to be tested in the results. It also contains the design of the system in both hardware and software.

Keywords: signals, mcu, meteorological, low cost

  
Nishu K

## INTRODUCCIÓN

En la actualidad muchas entidades publicas y privadas se dedican a la actividad del muestreo de señales meteorológicas, con distintos fines, sean para prevenir fenómenos naturales como para anticipar fenómenos aprovechables a la energética. Para ello se utilizan máquinas de instrumentación capaces de poder capturar y procesar dichas señales, muchas veces costosas para poder ser utilizada con fines personales o suponen un gran esfuerzo de inversión para empresas dedicadas. Ante ello, la presente investigación plantea el diseño de un sistema que captura señales meteorológicas, el sistema posee como base fundamental la placa Arduino Mega, hoy muy comercializada y con precio razonablemente bajo en el mercado, así como el uso del módulo ESP6288 de Arduino, con el cual podemos procesar las señales en tiempo real mediante in servidor web. De este modo la investigación que se presenta a continuación pretende mostrar el desarrollo de un sistema de señales meteorológicas, como temperatura, fuerza de viento, entre otras, con un costo de inversión bajo.



Handwritten signature and initials.

# CAPÍTULO I

## PLANTEAMIENTO DEL PROBLEMA

### 1.1. Descripción de la realidad problemática

Una estación meteorológica es una instalación que se encarga de medir y registrar los parámetros ambientales por medio de sensores que interactúan con el entorno, algunos de estos parámetros son: Temperatura, humedad, velocidad del viento, dirección del viento y precipitaciones. Estas instalaciones han ido mejorando tanto en tecnología como en diseño, a tal grado que hoy en día se puedan ubicar en terrenos hostiles, como lo hacen las estaciones meteorológicas automáticas, las cuales, a diferencia de las estaciones meteorológicas convencionales, adquieren los datos con los instrumentos de medición y de forma automática efectúan, almacenan y transmiten estas variables al Datalogger (Registrador), sin necesidad de la presencia de personal.

Sin embargo, estos equipos llegan a ser muy caros para un estudio de los parámetros climáticos por parte de la comunidad universitaria, o para un agricultor que desea hacer uso de esta información para mejorar su producción. Aunque una posible solución es que uno implemente su propia estación meteorológica automática, comprando sensores y controladores de bajo costo, el problema persistiría al no tener una predicción correcta del estado del clima. Esto se da porque los sensores de bajo costo por lo general no se encuentran acondicionados.

Por ello en este proyecto se pretende diseñar un sistema que tendrá como objetivo aplicar el acondicionamiento para el tratamiento de los sensores de una estación meteorológica automática con el fin de amplificar y filtrar el ruido proveniente del ambiente y así poder obtener señales más precisas, la cuales podrán ser enviadas al controlador. El controlador se encargará de comunicarse con un módulo de wifi, el cual tendrá como misión, enviar la información a la nube. Cuando se encuentre en una base de datos de la plataforma "Thingspeak" se podrá realizar un monitoreo en tiempo real de las señales filtradas de los



sensores de la estación y que a su vez proporcionara información a la comunidad universitaria o a cualquiera que lo desee.

## **1.2. Formulación del problema**

### **1.2.1. Problema General**

- ¿Es posible diseñar un sistema que nos permita acondicionar las señales obtenidas por medio de los sensores de una estación meteorológica automática de bajo costo para obtener señales más precisas de los parámetros ambientales y que a su vez este sistema envíe los datos a internet para su respectivo monitoreo?

### **1.2.2. Problemas Específicos**

- ¿Se puede realizar un sistema capaz de poder acondicionar las señales provenientes de los sensores de una estación meteorológica automática de bajo costo?
- ¿Será posible usar la conversión ADC de un microcontrolador para las señales de los sensores de una estación meteorológica automática de bajo costo?
- ¿Existirá diferencia entre las señales filtradas y señales no filtradas de los sensores de una estación meteorológica automática de bajo costo?
- ¿Se podrá realizar una conexión exitosa entre un microcontrolador y la red de internet?
- ¿Es posible realizar un monitoreo en tiempo real vía internet de las señales filtradas de los sensores de una estación meteorológica automática de bajo costo?

## **1.3. Objetivos**

### **1.3.1. Objetivo general**

Diseñar un sistema que nos permita acondicionar las señales obtenidas por medio de los sensores de una estación meteorológica automática de bajo costo



para obtener señales más precisas de los parámetros ambientales y que a su vez este sistema envíe los datos a internet para su respectivo monitoreo

### **1.3.2. Objetivo Especifico**

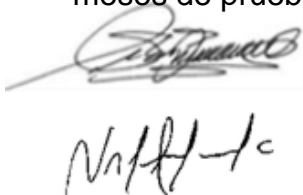
- Realizar un sistema capaz de poder acondicionar las señales provenientes de los sensores de una estación meteorológica automática de bajo costo.
- Usar la conversión ADC de un microcontrolador para las señales de los sensores de una estación meteorológica automática de bajo costo.
- Comprobar la diferencia entre las señales filtradas y señales no filtradas de los sensores de una estación meteorológica automática de bajo costo.
- Realizar una conexión exitosa entre un microcontrolador y la red de internet.
- Realizar un monitoreo en tiempo real vía internet de las señales filtradas de los sensores de una estación meteorológica automática de bajo costo.

### **1.4. Limitantes de la Investigación**

El proyecto de investigación se enfoca en la obtención de un resultado óptimo en la medida de variables meteorológicas con dispositivos de bajo costo. La principal limitante para el desarrollo del proceso de investigación es la obtención de materiales y el tiempo que demanda la construcción del prototipo, ya que debido a la realidad actual del país respecto a la pandemia causada por el virus covid-19 nos vemos restringidos en cuanto a la movilización.

#### **- Teórica**

Existen numerosas investigaciones afines a las estaciones meteorológicas, sin embargo, pocas investigaciones usan dispositivos de bajo costo e intentan conseguir los mismos resultados que una estación meteorológica automática con dispositivos de alto costo. Otras investigaciones se centran más en analizar como varía el clima, y deducir posibles eventos climáticos a lo largo de muchos meses de pruebas.



- **Temporal**

La prueba del prototipo debe compararse a los resultados obtenidos por ciertas entidades como el Senamhi, para contrastar la veracidad de los mismos. Esto involucra la toma de muestras a lo largo de un periodo extenso y en lugares con distintos microclimas.

- **Espacial**

El proyecto de investigación se ve afectado debido a la actual coyuntura de cuarentena, la cual impide la reunión y la búsqueda de materiales para el desarrollo del proyecto en zonas de difícil alcance debido a los protocolos de aislamiento social.

  
N.H.H.-C



## CAPÍTULO II

### MARCO TEÓRICO

#### 2.1. ANTECEDENTES: INTERNACIONAL Y NACIONAL

##### 2.1.1. Antecedentes Internacionales

MEDINA MOLINA & RAMIREZ GALVIS (2019) en su tesis titulada “Diseño y construcción de estación meteorológica, ubicada en la Vereda El Callejón, Ricaurte-Cundinamarca.”, donde los autores plantean el diseñar y construir una estación meteorológica para un estudio del comportamiento meteorológico dentro de la vereda y a su vez comparar los registros obtenidos con los datos de las estaciones meteorológicas cercanas que tiene instaladas el IDEAM. Tomando como base para su metodología un flujograma que consta de 8 fases las cuales se mencionan a continuación: prefactibilidad, diseño, ubicación de la estación, construcción, calibración, recolección de datos, análisis de datos y entrega de la información de la estación al IDEAM, llegando a la conclusión que a través de la plataforma de Arduino se logró crear una adecuada programación para la medición de variables como presión atmosférica, temperatura, humedad relativa y precipitación. La metodología aplicada en el trabajo de investigación de los autores servirá de aporte a nuestro proyecto de investigación.

Un trabajo de investigación de los autores VILLALTA CRUZ & SORTO PERDOMO (2013) que en su tesis titulada “Implementación de una estación meteorológica”, donde se tiene como objetivo realizar una estación meteorológica capaz de realizar medición y el registro de los diferentes parámetros climáticos. Para su desarrollo se busca los sensores para el monitoreo climático, logrando obtener información sobre los métodos aplicados para su medición, así como también los diversos medios que pueden ser usados para obtener la medida de la variable climatológica. Logrando obtener como conclusión que las pruebas realizadas por la estación permiten obtener parámetros climatológicos como velocidad y dirección del viento, temperatura, humedad, presión atmosférica y radiación solar. Así también recomendando la



importancia del diseño con la finalidad de poder movilizarnos en diferentes zonas geográficas. El aporte del antecedente mencionado para nuestra investigación sería el poder facilitarnos las mejores mediciones hechas por diferentes medios a fin de poder ahorrar costos.

Murcia Almanza & Rojas Paternina (2015), en su trabajo titulado “Diseño e implementación de un prototipo de estación meteorológica remota de bajo costo empleando el enfoque de internet de las cosas.”, donde se tiene como objetivo diseñar e implementar un prototipo de estación meteorológica que permita obtener las mediciones de variables físicas como la temperatura, la humedad, la presión atmosférica, la velocidad del viento y la detección de lluvia, a partir de una investigación basada en la utilización de nuevas tecnologías de software y hardware libres, que aunada al enfoque del internet de las cosas y la reducción de costos, de la investigación mencionada podemos obtener como aporte el uso de lenguajes de programación como C y Python para la elaboración de código fuente así como también el uso de Raspberry Pi como alternativa al Arduino.

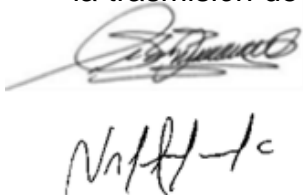
Para el trabajo correspondiente a Ferrer Sanabria (2017) en su tesis titulada “Desarrollo de una estación meteorológica de bajo costo”, teniendo como finalidad desarrollar una estación meteorológica autónoma de bajo costo utilizando un microcontrolador Arduino, un sistema de comunicación, sensores y un sistema de alimentación fotovoltaica para el sector agrícola. Realizando un conjunto de pasos para el desarrollo del proyecto, como realizar el diseño inicial, seleccionar las variables climatológicas, realizar el acondicionamiento de los sensores. Las conclusiones que se obtuvieron del trabajo fueron en primer lugar la viabilidad de la estación para el monitoreo de variables ambientales y climatológicas en sitios remotos, así como la utilización eficiente de la red móvil que brindó al sistema una mayor cobertura en referencia a otras tecnologías. De trabajo realizado por los autores podemos obtener los siguientes aportes necesarios para nuestras que serían: la metodología empleada para realizar el acondicionamiento adecuado para los sensores enfocados en las variables meteorológicas y el uso de la red móvil para el aumento de cobertura al sistema.



### 2.1.2. Antecedentes Nacionales

Gómez Vega (2017) en su tesis titulada “Niveles de radiación ultravioleta (UV) en la ciudad de Ayacucho periodo 2006 - 2011 e incidencia de cáncer de piel en la población” determina el comportamiento de las variables climatológicas en Ayacucho, teniendo como variable principal a la radiación ultravioleta haciendo uso de la información registrada por la estación meteorológica de la Universidad Nacional de San Cristóbal de Huamanga, el cual tiene como finalidad conocer los niveles de radiación ultravioleta en la ciudad de Ayacucho en el periodo 2006-2011 y la incidencia de cáncer de piel en la población. Para el desarrollo de la investigación fueron consideradas las siguientes variables climatológicas: temperatura externa, temperatura máxima, temperatura mínima, humedad externa (h.r.), radiación solar, energía solar, máxima radiación solar, índice UV, dosis UV, máxima dosis UV tomados en el horario de 9:00 am hasta 5:00 pm del periodo 2006 - 2011. Los datos fueron obtenidos por medio de la consola Vantage Pro 2 de la Estación Meteorológica de la Universidad Nacional de San Cristóbal de Huamanga, software compatible con la consola (WeatherLink 5.6), además de hacer uso de programas de análisis, como SPSS, Excel y otros. El aporte que nos brinda este antecedente para nuestro trabajo de investigación son los datos obtenidos mediante la estación meteorológica de la UNSCH que demuestran diferentes índices de radiación UV para diferentes meses del año, así como también la importancia de las variables meteorológicas y su relación para evitar la incidencia de cáncer en la piel.

En un segundo trabajo realizado por CÁRDENAS ROJAS (2017) en su tesis titulada “Implementación de un sistema de gestión de información de las estaciones meteorológicas, en la cuenca del rio Cachi, región Ayacucho” donde se plantea la existencia de una deficiente comunicación y sistematización de los datos climáticos en los procesos de atención a los usuarios que solicitan para los diversos estudios de inversión pública y privada, el cual tiene como finalidad el diseño e implementación de un Sistema de Gestión de Información de las Estaciones Meteorológicas de la Cuenca del Rio Cachi, Región Ayacucho. Con la trasmisión de la información en tiempo real de una estación meteorológica, a



través de un dispositivo móvil al Sistema de Gestión Meteorológica a desarrollar, obteniendo registros de información de parámetros climatológicos de cualquier estación. El aporte que nos brinda este antecedente para el presente trabajo de investigación es el desarrollo de un sistema de Gestión Meteorológica que permite obtener los datos en tiempo real a través de un dispositivo móvil.

Para el trabajo de investigación de Vasquez Reynaldo (2015) en su tesis titulada “Cambio de uso del suelo y su efecto en la temperatura y humedad del ambiente de la Universidad Peruana Unión, Ñaña, Lima, Perú.” donde se plantea determinar el efecto del cambio de uso del suelo y su efecto sobre las variables climáticas de temperatura y humedad del ambiente en la localidad de la Universidad Peruana Unión, Ñaña, Lima, Perú., el cual tiene como finalidad recolectar datos históricos climáticos de la Universidad Peruana Unión (UPeU) desde 1964 en los meses de enero, febrero, marzo, junio, julio y agosto hasta 2010, además de monitorear las variables climáticas de temperatura y humedad en el 2014. Utilizando a de regresión lineal simple y regresión lineal múltiple con los datos climáticos de 1964 a 2014 y su relación con el desarrollo urbano de la Universidad. El aporte que nos brinda este antecedente para el presente trabajo de investigación son los resultados obtenidos que demuestran que el cambio de uso del suelo agrícola a suelo urbano afecta las variables climáticas relacionadas a la temperatura y humedad.

Para el autor GUILLÉN FERNÁNDEZ (2018) en su tesis titulada “Identificación espacial de zonas vulnerables ante lluvias extremas en Lima Metropolitana y el Callao, a partir de los impactos reportados por la inusual lluvia del 15 de enero de 1970 y otros eventos”, teniendo como finalidad la implementación de nuevas herramientas y métodos para realizar un modelamiento espacial que nos permita identificar las zonas más vulnerables frente a una lluvia extrema en Lima Metropolitana, así también, identificar la tipología predominante de las viviendas en zonas más vulnerables. Utilizando reportes históricos de una de las lluvias atípicas más catastróficas que ha ocurrido en Lima, la del 15 de enero de 1970 coloquialmente conocida como el “diluvio setentero”, quedando así la ciudad en condiciones críticas. También se usó el programa Maxent y un conjunto de



herramientas de análisis espacial, como los sistemas de información geográfica para identificar los patrones y características de estos espacios. El aporte que nos brinda este antecedente para el presente trabajo de investigación son los resultados que indican las vulnerabilidades que se presentan en la mayoría de las periferias de Lima Metropolitana y el Callao, dándonos sustento para la investigación de variables meteorológicas relacionadas a las lluvias.

En el trabajo de investigación de Vargas Seminario (2015) en su tesis titulada “Sistema de información en línea para el seguimiento del fenómeno El Niño en la costa norte del Perú” donde se desea aportar con información consolidada a través de la implementación de un sistema de seguimiento mediante una página web. La información internacional y nacional será adquirida de otras páginas web, mientras que la información regional se generará a partir de las estaciones meteorológicas de la UDEP. Para el desarrollo del trabajo se realizaron dos fases la primera es de investigación acerca de los temas referidos a El Niño y la segunda, de aplicación donde se presenta el análisis, diseño e implementación de un sistema de adquisición de información en línea. El aporte que nos brinda este antecedente para el presente trabajo de investigación es que el sistema implementado por medio de las estaciones meteorológicas facilitará el acceso a la información y será de gran relevancia para diferentes sectores productivos y sociales de la región norte del Perú.

## **2.2. MARCO**

### **2.2.1. MARCO TEÓRICO**

#### **2.2.1.1. Importancia de la Estación meteorológica automática - Medina, Ramón, & Fresdo (2017)**

Los datos obtenidos a través de la meteorología son fundamentales para las actividades sociales y económicas, establecer factores de riesgo y garantizar medidas anticipadas para los recursos híbridos y alimenticios. La información es obtenida por medio de estaciones meteorológicas ubicadas en distintos lugares y que durante mucho tiempo se encuentran registrando las variables climatológicas. La indagación sobre la información meteorológica ha tenido un



crecimiento en tiempos recientes debido a que existe un cambio rápido y global del clima.

Arteaga Ramírez, Cervantes Osornio, Vásquez Peña, & Ojeda Bustamante (2017)

No es suficiente que las estaciones meteorológicas solo capten los parámetros del clima, sino que también la información que proporcionen debe de ser confiables y representar el clima de una zona, esto contribuiría a una buena planificación en urbanismo, ingeniería o en ordenamiento del territorio. Los elementos del clima primordiales en una estación meteorológica son: temperatura, humedad del aire, viento, radiación solar, precipitación y evaporación. Donde hay que tener en cuenta que el valor de la información obtenida debe estar relacionada con su utilidad y finalidad de la misma. Habitualmente para la realización de los trabajos meteorológicos se viene usando las estaciones convencionales (EC), sin embargo, actualmente está surgiendo un incremento en el empleo de las estaciones automáticas (EA). Esto se debe a que las estaciones automáticas aportan a la estandarización de la información meteorológica lo que provoca que se corrijan algunos problemas tradicionales como la falta de objetividad del observador y los errores de digitación.

Zúñiga Vinueza, Polanco Risquet, & Nagy (2016)

El costo y mantenimientos de las estaciones meteorológicas automáticas (EMA) comerciales es alto, por ello en algunas ocasiones para disminuir los gastos que este implica se busca diseñar un EMA que ayudara a extender el manejo de su utilización.

Polo Bravo, Torres Muro, Ríos Marquezado, Bernabé Coaquera, & De la Gala Contreras (2017)

Una estación metereologica que consta de componentes y sensores de bajo costo adquiridos en el mercado nacional puede reducir el costo a comparación



de una importada que cuesta 200 a 300 veces más. La información que puede registrar es parecida a la que proporciona cualquier estación importada de las mejores marcas.

El país necesita de estaciones meteorológicas de bajo costo pero de buen desempeño, ya que la información que proporciona en tiempo real, es clave para poder predecir la dinámica meteorológica, fenómenos y desastres naturales como el efecto invernadero, cambio climático, sequías, friaje, fenómeno del niño y la niña.

Aunque la intención de tener un diseño propio de un EMA que se pueda implementar con el fin de minimizar los gastos puede representar una gran idea, no es tan sencillo como parece porque involucra diversos factores los cuales tienen como finalidad hacer que un sistema de bajo costo se aproxime en resultados de una EMA del mercado. Entre estos factores podemos mencionar:

- La adaptación de los sensores climatológicos (temperatura, humedad del aire, viento, radiación solar, precipitación) de bajo costo por medio de acondicionamiento de señales con el objetivo de obtener señales que sean compatibles con la entrada del convertidor analógico y digital (optamos por usar el ADS1115).
- Comunicación eficiente entre el convertidor analógico y digital de mayor precisión (ADS1115) con el microcontrolador (en nuestro caso será Arduino Mega ATmega2560).
- Programación completa que conecte las señales provenientes del conversor ADC con un módulo wifi (esp8266) que sería el primer paso para realizar el monitoreo de los parámetros ambientales
- Pruebas de conectividad con la red teniendo como base la conexión entre el módulo wifi(esp8266) y un servidor de internet, este sería el último factor que se tiene que solucionar para tener un monitoreo de señales ambientales que tengan una mayor confiabilidad en el diseño propio de un EMA



## 2.2.2. MARCO CONCEPTUAL

### Temperatura

La temperatura es una magnitud muy conocida cuando se realiza un estudio al estado de la atmósfera. En los medios de comunicación casi siempre se incluye un espacio dedicado a la temperatura. La temperatura está vinculada con la rapidez de las partículas de la materia al estar en movimiento, por lo tanto, mientras más movimiento exista, mayor será la temperatura. Para obtener datos de medición de la temperatura se tiene que conocer la relación del cambio de temperatura con las propiedades de la materia que se ven alteradas: la resistencia eléctrica de algunos materiales, el volumen de un cuerpo, el color de un objeto, etc.

### Humedad ambiental

La humedad ambiental también es conocido como ratio de evapotranspiración, y se puede obtener mediciones con bastante aproximación si se relaciona la evapotranspiración potencial (ETP) y la precipit4

A continuación, se mencionarán los posibles sensores a usar para la realización del proyecto, así también se da a conocer sobre sus datos técnicos y su forma de acondicionamiento siempre y cuando sea necesario.

### Sensor de temperatura (TMP36)

Es un sensor de temperatura en grados centígrados de precisión y bajo voltaje. Entre sus especificaciones tenemos:

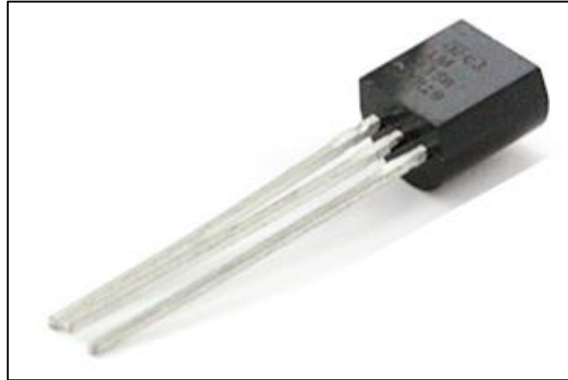
- Voltaje de entrada: 2.7 a 5.5 V DC
- Factor de escala 10 mV/°C
- Precisión de  $\pm 2$  °C sobre temperatura
- Linealidad de  $\pm 0.5$  °C
- Rango de operación: -40 °C a +125 °C





**Figura 1**

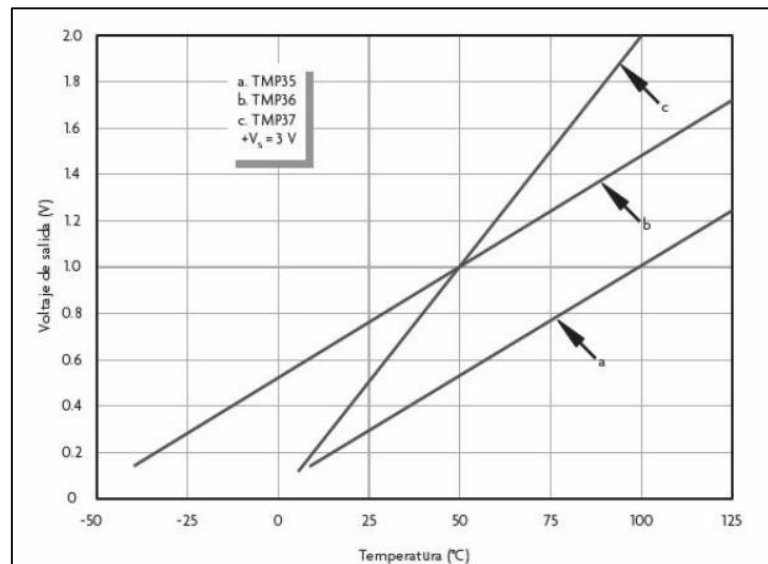
*Sensor TMP36*



Para su acondicionamiento es posible tomar de base el acondicionamiento realizado para un LM35, que también es un sensor de temperatura. Ya que ambos sensores son lineales y su diferencia radica en el rango de operación y voltaje de salida como se muestra en la siguiente Figura 2.

**Figura 2**

*Curva característica del sensor TMP35 y TMP36.*



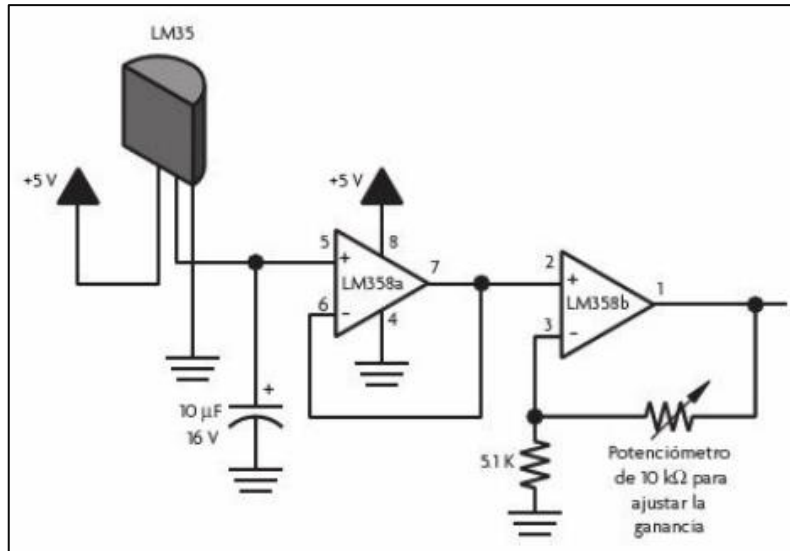
Fuente: Datasheet TMP36

*[Handwritten signature]*  
*NH-H-c*

A continuación se muestra como sería el acondicionamiento de la señal en la Figura 3.

**Figura 3**

*Circuito para medir la temperatura del sensor LM35*



Fuente: Corona Ramírez, Abarca Jiménez, & Mares Carreño, (2014). Sensores y actuadores: Aplicaciones con Arduino

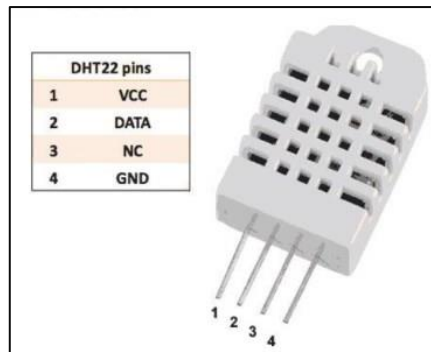
### **Sensor de humedad (DHT22)**

El sensor es un sensor de temperatura y humedad (DHT22) que nos permitirá monitorear la temperatura y humedad relativa de forma precisa y sencilla a un bajo precio. La salida suministrada es de tipo digital, no requiriéndose utilizar entradas analógicas. En la Figura 4 se muestra los pines del sensor.

*[Handwritten signature]*  
*Niff-1c*

## Figura 4

Sensor DHT22



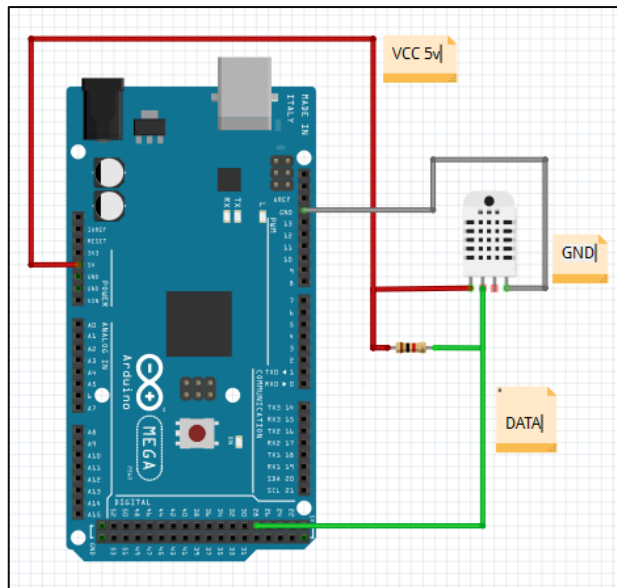
Como se puede observar el sensor tiene cuatro pines solo tenemos que alimentar a 5V la patilla 1, realizar la conexión a tierra de la patilla 4, y obtener la información usando una salida digital del Arduino por la patilla 2. La patilla 3 es NC lo cual quiere decir que no se tiene que conectar a nada

Sin embargo, aunque se una señal digital, este sensor necesitara una resistencia de pull-up (que según las especificaciones debe estar entre 4.7kOhm a 10kOhm) que se debería ubicar entre en la patilla 2(datos) y la alimentación (Vcc), esto garantizaría los niveles lógicos requeridos para realizar el cambio(switch) de entrada a salida y viceversa (los pines son de tipo I/O); si se omite este paso el sensor no funcionaría correctamente. (Cano León, 2016)

Handwritten signature and initials.

**Figura 5**

*Conexión directa del Sensor DTH22 al Arduino Mega*



Fuente: UNAC 2020, elaboración propia

### **Sensor de precipitación**

Para nuestro sensor de precipitación se puede usar pluviómetro marca RIKA modelo RK400-04 de alta precisión con filtro protector para evitar entrada de materiales sólidos sistema de pulsos.

**Figura 6**

*Sensor modelo RK400-04*



A continuación, se muestra sus especificaciones que nos da el fabricante en la Figura 7.

Una firma manuscrita en tinta negra, que parece ser 'N. H. H. - c'.

## Figura 7

### Características del Sensor RK400-04

Item	Specification
Collector	diameter : $\phi$ 200mm, height: 271mm
Measuring range	$\leq 4$ mm/min
Resolution	0.2mm
Accuracy	$\pm 2\%$
Maximum load voltage	30VDC(pulse output)
Maximum load current	20mA
Output	5Pulses/mm(@10k $\Omega$ &0.01uF)
Operating temperature	-20-80 $^{\circ}$ C
Main material	ABS
Weight(unpacked)	1kg

Fuente: Especificaciones del Fabricante

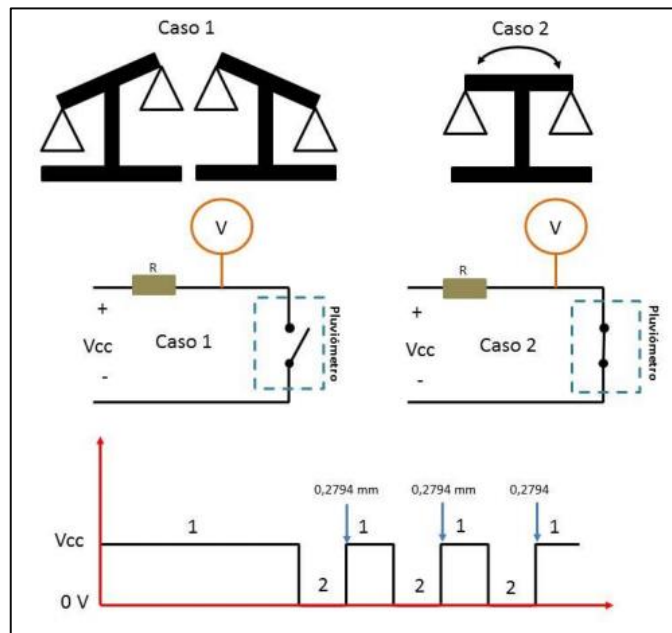
El comportamiento del pluviómetro se puede parecer a un sensor analógico, pero se puede caracterizar como digital. Su funcionamiento radica en un sensor electromecánico basado en un embudo que direcciona la precipitación hacia un colector, este se encarga de recoger la precipitación para su posterior medición.

El colector tiene una semejanza a una balanza con 2 cuencos que se encuentran ubicados a los extremos del pluviómetro, por lo que la balanza tendrá dos posiciones, cuando un extremo ese recibiendo agua y se llena la balanza cambia de posición debido al peso del agua almacenada, en ese instante cuando el interruptor se comporta como un interruptor cerrado. (Tobajas Garcia, 2016). En la Figura 8 se muestra este procedimiento y su sistema de pulsos



**Figura 8**

*Funcionamiento del pluviómetro*



Fuente: Tobajas Garcia (2016), Diseño e implementación de una estación meteorológica con Raspberry Pi

### **Sensor de viento**

Para nuestro sensor de viento se puede usar un anemómetro modelo Anemo4403 de alta precisión con filtro protector para evitar entrada de materiales sólidos sistema de pulsos.

**Figura 9**

*Sensor Anemo4403*



*[Handwritten signature]*  
*NH-H-c*

A continuación, se muestra sus especificaciones que nos da el fabricante en la Figura 10.

**Figura 10**

*Características del Anemo4403*

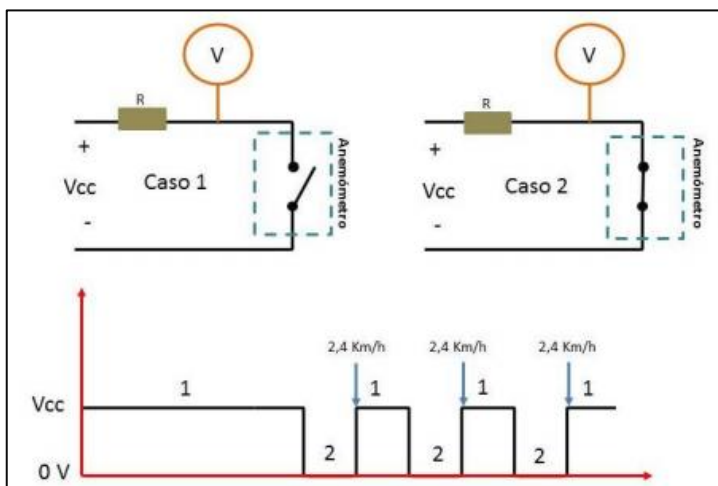
TECHNICAL CHARACTERISTICS			
Power Supply	12 to 30Vdc	Connectable load impedance	$R_L < \frac{V_{CC}-8V}{0.02A} [\Omega]$
Power consumption	<2,5W	Storage temp.	-35°C... +80°C
Starting speed	8 km/h	Operating temp. (ice free)	-20°C... +70°C
Measure range	3...180 km/h	Weight	155 g approx.
Accuracy	<3%	Analog output	Two-wires 4-20 mA

Fuente: Especificaciones del Fabricante

El anemómetro se comporta como un interruptor que se abre y se cierra, cuando éste de una vuelta. En ese caso se verán pulsos cada vez que el interruptor se abre y se cierre. (Tobajas Garcia, 2016). A continuación se muestra su funcionamiento y su sistema de pulsos en la Figura 11.

**Figura 11**

*Funcionamiento del anemómetro*



Fuente: Tobajas Garcia (2016), Diseño e implementación de una estación meteorológica con Raspberry Pi

*[Handwritten signatures]*

## Sensor de radiación UV

El EPD-440-0-1.4 es un fotodiodo con un rango de longitud de onda de 440 nm, ancho de banda de 180 nm, capacitancia de 300 pF, corriente oscura de 10 a 30 pA, capacidad de respuesta / fotosensibilidad de 0.17 A / W.

**Figura 12**

Sensor EPD-440-0-1.4



A continuación, se muestra sus especificaciones que nos da el fabricante en la Figura 13.

**Figura 13**

Características del Sensor EPD-440-0-1.4

Parameter	Test conditions	Symbol	Min	Typ	Max	Unit
Breakdown voltage <sup>1)</sup>	$I_R = 10 \mu A$	$V_R$	5			V
Dark current	$V_R = 5 V$	$I_D$		10	30	pA
Peak sensitivity wavelength	$V_R = 0 V$	$\lambda_p$		440		nm
Responsivity at $\lambda_p$	$V_R = 0 V$	$S_\lambda$		0.17		A/W
Sensitivity range at 1%	$V_R = 0 V$	$\lambda_{min}, \lambda_{max}$	190		570	nm
Spectral bandwidth at 50%	$V_R = 0 V$	$\Delta\lambda_{0.5}$		180		nm
Shunt resistance	$V_R = 10 mV$	$R_{SH}$	150	200		G $\Omega$
Noise equivalent power	$\lambda = 440 nm$	NEP		$1.1 \times 10^{-14}$		W/ $\sqrt{Hz}$
Specific detectivity	$\lambda = 440 nm$	$D^*$		$1.0 \times 10^{12}$		cm $\cdot\sqrt{Hz} \cdot W^{-1}$
Junction capacitance	$V_R = 0 V$	$C_J$		300		pF
Switching time ( $R_L = 50 \Omega$ )	$V_R = 5 V$	$t_r, t_f$		1/20		ns
Photo current at $\lambda = 440 nm$ <sup>1,2)</sup>	$V_R = 0 V$ $E_e = 1 mW/cm^2$	$I_{Ph}$		1.55		$\mu A$

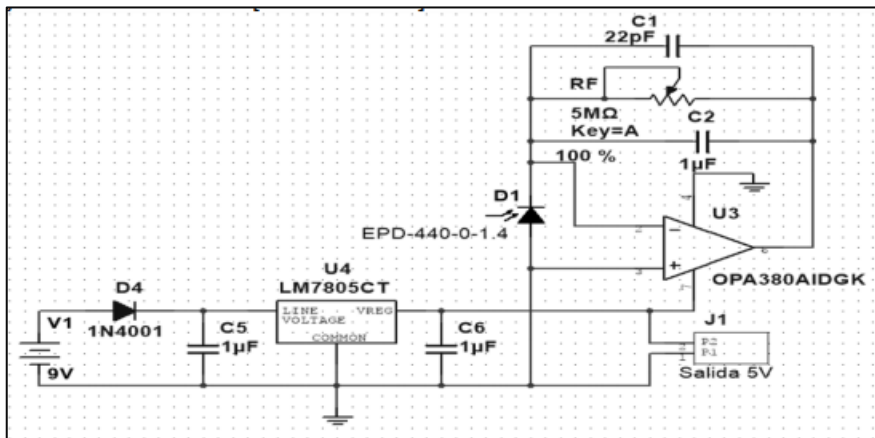
Fuente: Especificaciones del Fabricante



El fotodiodo EPD-440-0-1.4 son dispositivos que son sensibles a luz, pueden ser instalados en un soporte a fin de permitir la entrada de luz o partículas fotónicas para que sensor pueda realizar su respectiva medición. (Castelan Carreto & Cisnetos, 2015). Uno de sus posibles acondicionamientos para la entrada al Arduino Mega se muestra a continuación en la Figura 14.

**Figura 14**

*Diagrama esquemático de las partes principales del medidor de radiación*



Fuente: Castelan Carreto & Cisnetos (2015), Diseño de una medidor de radiación ultravioleta utilizando el fotodiodo EPD-440-0-1.4

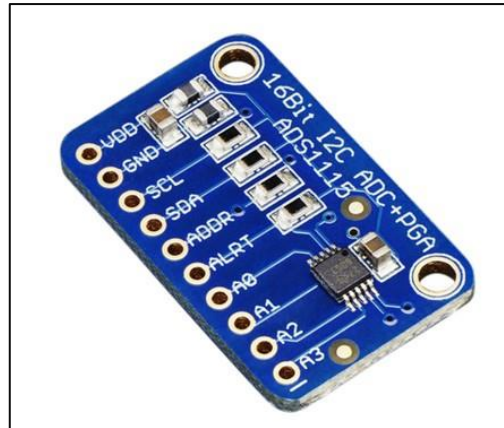
### Conversor ADC

El ADS 1115 es un ADC que cuenta con una resolución de 16bits y una frecuencia de 860 muestreos por segundo. Tiene 10 puentes, 4 de entrada analógica, 2 correspondientes a la alimentación (Vdd y GND), un puerto SDA que se encarga de la sincronización de datos y un puerto SCL que se encarga de la sincronización de reloj, ambos en conjunto se encargan de la comunicación del ADC con el microcontrolador. El puerto ADDR depende del modo de funcionamiento del ADS1115 y el puerto ALERT se usa para la verificación en tiempo real de la transmisión de datos.

*[Handwritten signature]*  
*[Handwritten signature]*

**Figura 15**

*Conversor ADC ASD1115*



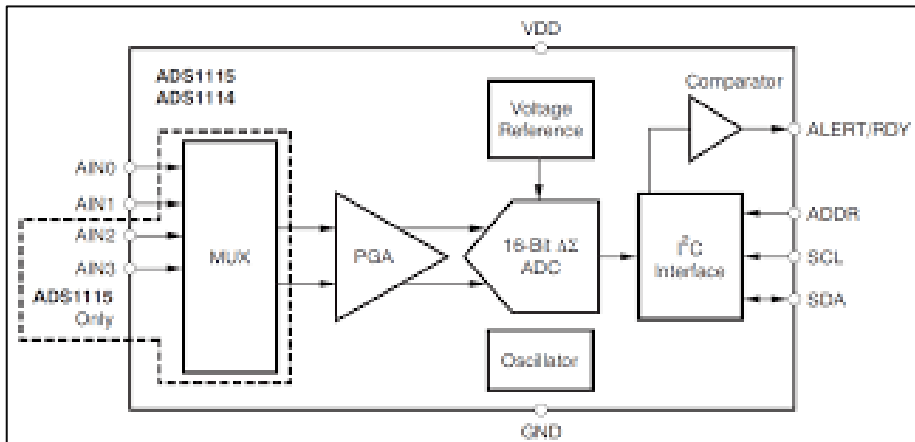
El conversor ADC ASD1115 utiliza protocolo de comunicación I2C, el cual es compatible con Arduino y otros microcontroladores (PIC). Este dispositivo contribuye a mejorar la precisión de lectura de las variables analógicas medidas, puesto que el ADC de los microcontroladores PIC es de 10 bits mientras que el del ADS1115 es de 16bits.

- Cuenta con tres modos de conexión: Single Ended, Diferencial y Comparador.
- Single Ended: Dispone de 4 canales de 16bits
- Diferencial: Se usa dos ADC para cada medición, el número de canales se reduce a 2, las ventajas es que se pueden medir tensiones negativas y tiene mayor inmunidad al ruido.
- Comparador: Funciona como alerta en el cual se activa en uno de los canales cuando el valor registrado supera el umbral seleccionado.

En la Figura 16 se muestra el diagrama interno del ASD1115

**Figura 16**

*Diagrama interno del ASD1115*



Microcontrolador (ATMEGA 2560: ARDUINO MEGA)

Arduino Mega 2560 es una tarjeta de desarrollo de Hardware libre construida con el microcontrolador ATmega2560. Está pensado para proyecto GRANDES, con muchos sensores y actuadores, como una impresora 3D, un proyecto de domótica (casa inteligente) control y automatización de máquinas.

Especificaciones Técnicas:

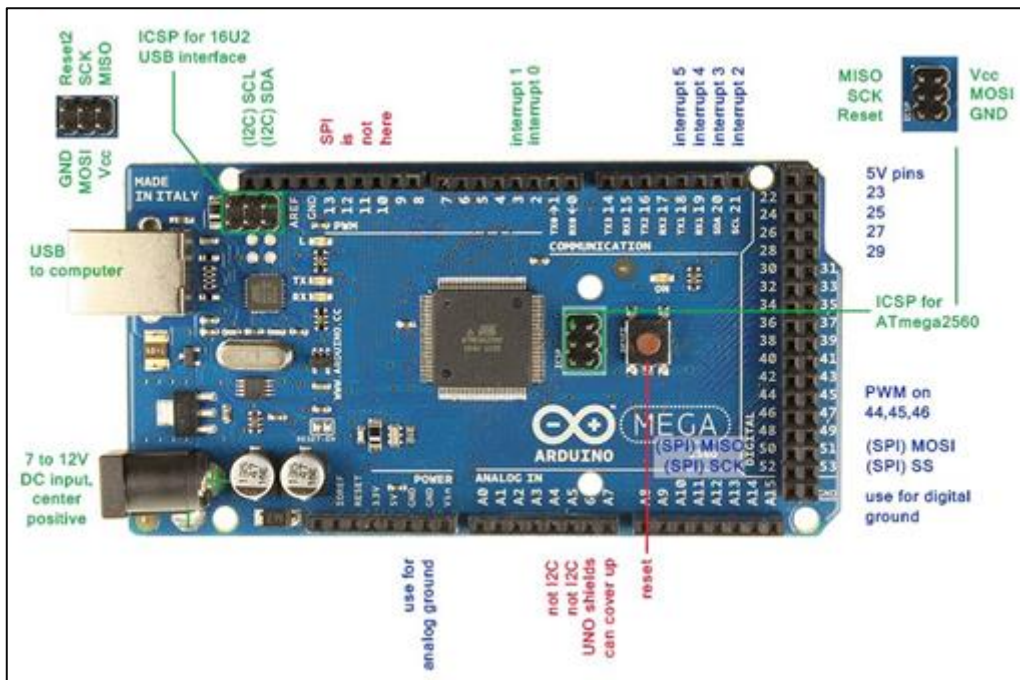
- Microcontrolador: ATmega2560
- Voltaje Operativo: 5V
- Tensión de Entrada: 7-12V
- Voltaje de Entrada(límites): 6-20V
- Pines digitales de Entrada/Salida: 54
- Pines PWM= 14
- Pines análogos de entrada: 16
- Corriente DC por cada Pin Entrada/Salida: 40 mA
- Corriente DC entregada en el Pin 3.3V: 50 mA
- Memoria Flash: 256 KB (8KB usados por el bootloader)
- SRAM: 8KB
- EEPROM: 4KB

- Clock Speed: 16MHz

En la Figura 17 se muestra la distribución de conexiones que se pueden realizar en un Arduino Mega:

**Figura 17**

*Partes de un Arduino Mega*



MODULO WIFI (ESP8266)

El esp8266 es un dispositivo el cual permite el uso del protocolo TCP/IP para la comunicación entre dicho dispositivo y un sistema WI-FI. Este dispositivo actúa como Access Point. Se pueden agregar sensores, es decir también funciona como controlador. En la parte posterior de la placa PCB, el módulo cuenta con una antena, la cual permite la recepción y transmisión de la información dentro del alcance de la red inalámbrica.

*[Handwritten signature]*  
*[Handwritten signature]*

## Figura 18

Módulo Wifi ESP8266



El ESP8266 es un dispositivo el cual permite el uso del protocolo TCP/IP para la comunicación entre dicho dispositivo y un sistema WI-FI. Este dispositivo actúa como Access Point. Se pueden agregar sensores, es decir también funciona como controlador. En la parte posterior de la placa PCB, el módulo cuenta con una antena, la cual permite la recepción y transmisión de la información dentro del alcance de la red inalámbrica.

Datos técnicos:

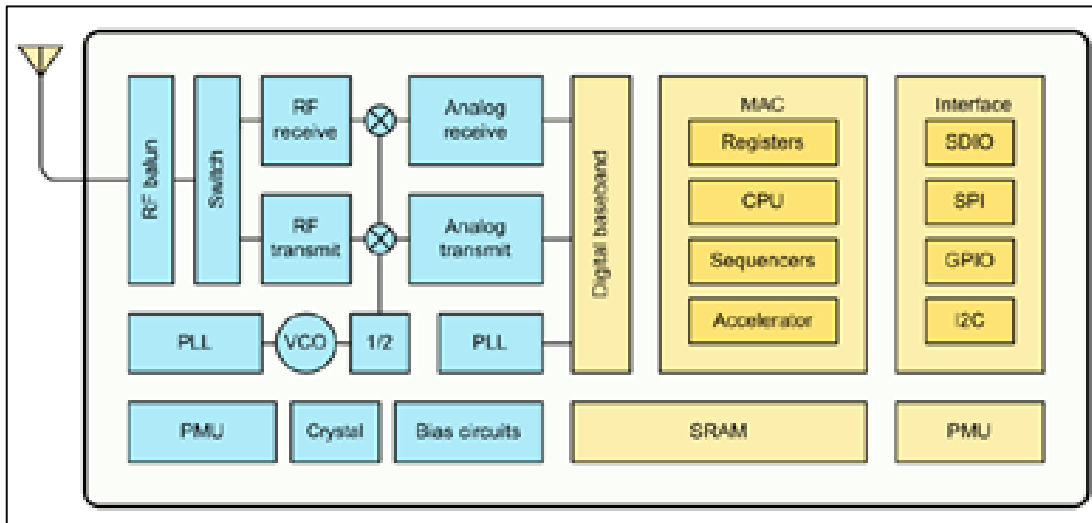
- Frecuencia: 2.4 – 2.5 GHz
- Potencia de la Antena: 20/17/14 dBm
- Voltaje de operación: 3.0 – 3.6V
- Temperatura de operación: -40° - 125°C
- Protocolos de Red: IPv4, TCP/UDP, FTP, HTTP

El módulo cuenta con pines de transmisión y recepción los cuales permiten la comunicación entre el microcontrolador y la placa. Se conectan de manera cruzada.

Handwritten signature and initials in black ink, located at the bottom left of the page.

**Figura 19**

*Diagrama interno del módulo ESP8266*



Fuente: Maldonado Girón (2018), *Diseño de una Estación meteorológica con control de accionamiento electromecánicos y monitoreo de sensores*

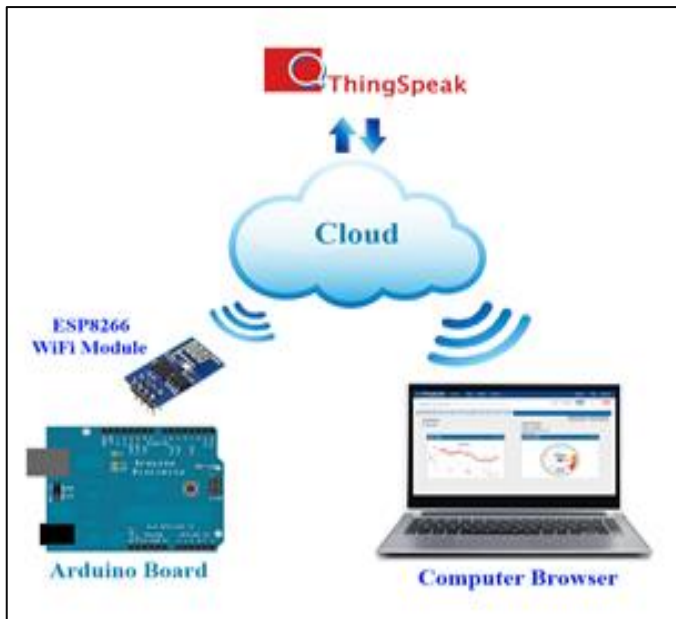
## **SERVIDOR WEB (THINGSPEAK)**

En la actualidad existen un gran número de plataformas IoT debido a la incorporación del internet a los procesos industriales y a la automatización. Estas plataformas permiten registrar datos adquiridos por sensores, supervisarlos en tiempo real y controlar a distancia otros dispositivos. Además, ofrecen una interfaz interactiva con lenguajes de alto nivel como por ejemplo Python, que se usa bastante en la ciencia de datos.

ThingSpeak es un servicio de plataforma que permite almacenar y recuperar datos de sensores en la nube y desarrollar aplicaciones de IoT. A la vez, permite realizar análisis en tiempo real de las variables y actuar sobre ellos. Los datos que proporciona el sensor pueden ser enviados a ThingSpeak haciendo uso de dispositivos que cuenten con protocolo de comunicación MQTT. MQTT está diseñado para dispositivos que estén limitados por el bajo ancho de banda y latencia.

**Figura 20**

*Comunicación entre el Arduino y una PC mediante ThingSpeak*



### **2.3. DEFINICIONES DE TÉRMINOS**

#### **ADQUISICIÓN DE DATOS**

La adquisición de datos es el proceso por el cual un dispositivo pueda medir un fenómeno eléctrico o físico como voltaje, corriente, temperatura, presión o sonido. En la adquisición de datos se emplea los acondicionadores de señales con el fin de obtener resultados más confiables

#### **CONVERSION ADC**

El conversor ADC es un arreglo con componentes electrónicos que tiene puertos de entrada, salida y alimentación eléctrica donde la entrada corresponde a una señal analógica y la salida a una señal digital correspondiente a la conversión de la señal de entrada. Se pueden clasificar por la modulación que utilizan: Modulación por Rampa, Modulación por Aproximaciones sucesivas, Modulación por memoria Flash, Modulación por tiempo de pulso

## ESTACIÓN METEOROLÓGICA AUTOMÁTICA

La estación meteorológica automática es una instalación donde se recopilan datos para el análisis del clima meteorológico, consta de instrumentos de medición precisos para el análisis. Las estaciones meteorológicas se pueden clasificar en manuales y automáticas, dependiendo del proceso de ingreso de datos. En la estación meteorológica manual los datos son ingresados por un usuario que monitorea los instrumentos de medición. Mientras que en una estación meteorológica automática el usuario programa el equipo para que se ingresen los datos sin la necesidad de la intervención humana en el campo. A la vez almacenan en tiempo real estos datos disminuyendo la probabilidad de error en todo el proceso.



Handwritten signature and initials.



## CAPÍTULO III FORMULACIÓN DE LA HIPÓTESIS

### 3.1. HIPOTESIS GENERAL Y ESPECÍFICA

#### 3.1.1. Hipótesis Principal

- El desarrollo de un sistema de acondicionamiento para las señales obtenidas por medio de los sensores de una estación meteorológica automática de bajo costo para obtener señales más precisas de los parámetros ambientales y que a su vez este sistema envíe los datos a internet para su respectivo monitoreo.

#### 3.1.2. Hipótesis Específica.

- Es posible diseñar un sistema capaz de acondicionar señales provenientes de los sensores de una estación meteorológica automática de bajo costo
- Se puede hacer uso de la conversión ADC de un microcontrolador para las señales de los sensores de una estación meteorológica automática de bajo costo.
- Existen diferencias entre las señales filtradas y señales no filtradas de los sensores de una estación meteorológica automática de bajo costo
- Es posible realizar una conexión exitosa entre un microcontrolador y la red de internet
- Es posible realizar un monitoreo en tiempo real vía internet de las señales filtradas de los sensores de una estación meteorológica automática de bajo costo

### 3.2. DEFINICION DE VARIABLES

- **Variables Dependiente**

Calidad de las Variables Climatológicas

- **Variables Independientes:**

Variables Climatológicas



### 3.3. OPERACIONALIZACIÓN DE VARIABLES

VARIABLE	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	DIMENSIÓN	INDICADORES	INSTRUMENTOS
VARIABLES INDEPENDIENTES	VARIABLES CLIMATOLÓGICAS	Variables Principales que se miden por una estación meteorológica para la obtención de datos, análisis del tiempo, prevención contra factores de riesgo	Temperatura	Medición en grados centígrados de la temperatura ambiente	TMP36
			Humedad	Cantidad de agua en el aire	DHT22
			Radiación Solar	Índice de radiación UV	EPD-440-0-1
			Precipitación	Cantidad de pulsos por almacenamiento de agua	RK400-04
			Velocidad del viento	Cantidad de pulsos por vueltas	Anemo4403
VARIABLES DEPENDIENTES	CALIDAD DE LAS VARIABLES CLIMATOLÓGICAS	Precisión y confiabilidad de las variables del clima actual por medio de circuitos acondicionadores de señal	Señal de temperatura acondicionada	Señal Amplificada y filtrada de Temperatura	TMP36
			Señal de humedad acondicionada	Señal Amplificada y filtrada de Humedad	DHT22
			Índice de Radiación Solar acondicionado	Señal Amplificada y filtrada de Radiación Solar	EPD-440-0-1
			Señal de pulsos de Precipitación acondicionado	Señal Amplificada y filtrada de Precipitación	RK400-04
			Señal de pulsos de viento acondicionado	Señal Amplificada y filtrada de velocidad de viento	Anemo4403

Fuente: UNAC 2020, Elaboración Propia



Handwritten signature and initials, likely representing the author or reviewer of the document.

## CAPÍTULO IV

### DISEÑO METODOLÓGICO

#### 4.1. TIPO Y DISEÑO DE INVESTIGACIÓN

##### 4.1.1. Tipo de Investigación

La presente investigación es de tipo cuantitativa, pues en ella se toman datos de las diferentes variables climatológicas y estos serán acondicionados para una mayor precisión.

Es de tipo de método deductivo, ya que nos estaremos basando en variables reales con el fin de predecir comportamientos climáticos.

El nivel de la investigación es Descriptivo-Explicativa. Descriptiva porque describe las características de los sistemas de acondicionamiento de señales y se realiza una descripción detallada de todos los elementos del sistema. Es Explicativa porque se analiza las relaciones causa efecto del clima respecto a las variables tomadas.

##### 4.1.2. Diseño de Investigación

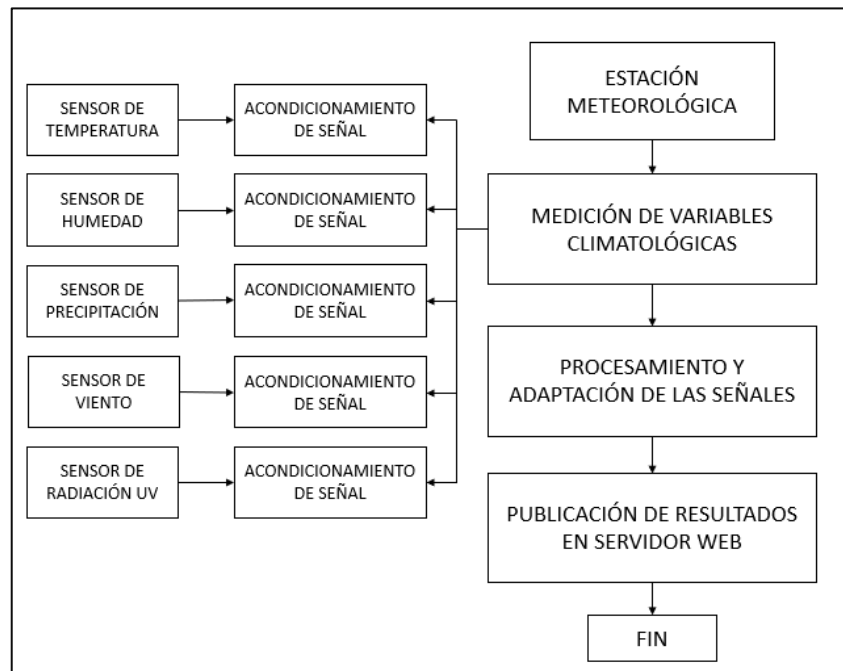
La presente investigación desarrolla un sistema de estación meteorológica automática a fin de obtener una información del clima que sea confiable y precisa, utilizando dispositivos de bajo costo. Para la elaboración del proyecto se consideran los siguientes procesos en distintas etapas donde se realiza el acondicionamiento de la señal deseada a obtener:



Handwritten signature and initials.

**Figura 21**

*Diagrama de bloques del sistema*



## 4.2. MÉTODO DE INVESTIGACIÓN

Para el diseño de la estación meteorológica automática se utiliza la placa de desarrollo Arduino Mega 2560 que se basa en el microcontrolador ATmega2560. Nos permite contar con una gran cantidad de pines, siendo en total 54 de entrada/salida, de los cuales 15 de ellos se pueden usar como salidas de PWM, 16 entradas analógicas y 4 UARTs (puertos serial). Tiene incorporado un Cristal de 16MHz y una memoria Flash de 256KB. La comunicación con la computadora es a través del puerto serie.

Otras características técnicas:

-Voltaje Operativo: 5V

-Tensión de Entrada: 7-12V

-Voltaje de Entrada(límites): 6-20V

-Corriente DC por cada Pin Entrada/Salida: 20 mA

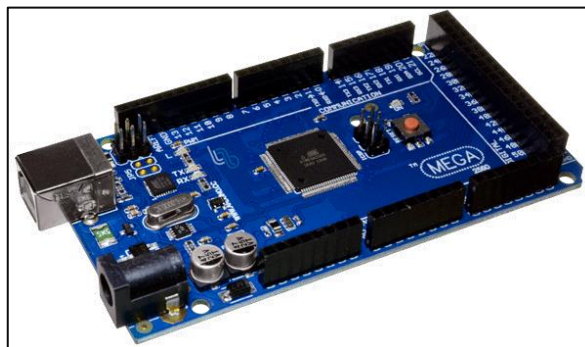
-Corriente DC entregada en el Pin 3.3V: 50 mA

-SRAM: 8KB

-EEPROM: 4KB

### Figura 22

*Tarjeta de Desarrollo Arduino Mega*



El ESP8266 es un dispositivo el cual permite el uso del protocolo TCP/IP para la comunicación entre dicho dispositivo y un sistema WI-FI. Este dispositivo actúa como Access Point. Se pueden agregar sensores, es decir también funciona como controlador. En la parte posterior de la placa PCB, el módulo cuenta con una antena, la cual permite la recepción y transmisión de la información dentro del alcance de la red inalámbrica.

### Figura 23

*ESP8266*

A handwritten signature in black ink, appearing to be 'N. H. H. C.' or similar, located at the bottom left of the page.

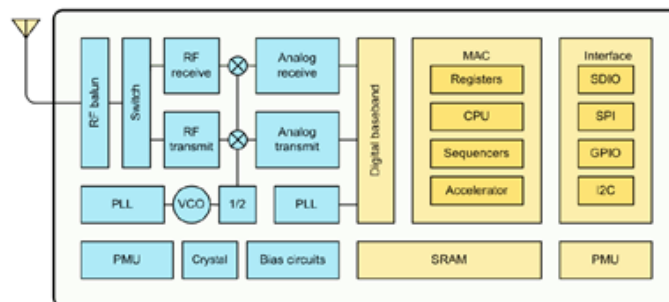
Datos técnicos:

- Frecuencia: 2.4 – 2.5 GHz
- Potencia de la Antena: 20/17/14 dBm
- Voltaje de operación: 3.0 – 3.6V
- Temperatura de operación: -40° - 125°C
- Protocolos de Red: IPv4, TCP/UDP, FTP, HTTP

El módulo cuenta con pines de transmisión y recepción los cuales permiten la comunicación entre el microcontrolador y la placa. Se conectan de manera cruzada.

### Figura 24

*Diagrama de bloques del ESP8266*

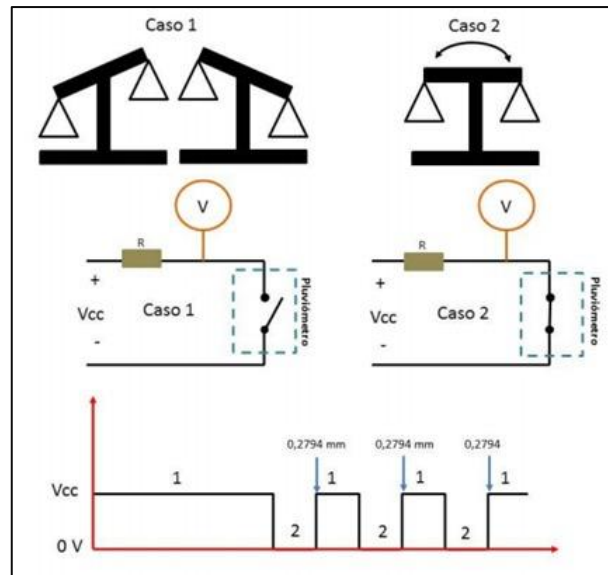


Para la etapa de obtención del acondicionamiento debemos saber cómo se maneja cada sensor de nuestro sistema. Simular cada sensor implicaría disminuir las posibilidades de error ya que cada sensor debe ser acondicionado en caso lo necesite.

Para el sensado de la lluvia usamos un pluviómetro, este dispositivo emite pulsos cada vez que su sistema es llenado de agua, podemos usar esos como señales digitales, para ser más específicos como un PWM que luego podrá ser usado por nuestro microcontrolador. Este dispositivo se puede simular como un generador de señales sabiendo cual es la magnitud que envía el sensor al controlador. También el tiempo de frecuencia de los pulsos debe ser considerado en la realidad, pero si nosotros evaluamos el rango

**Figura 25**

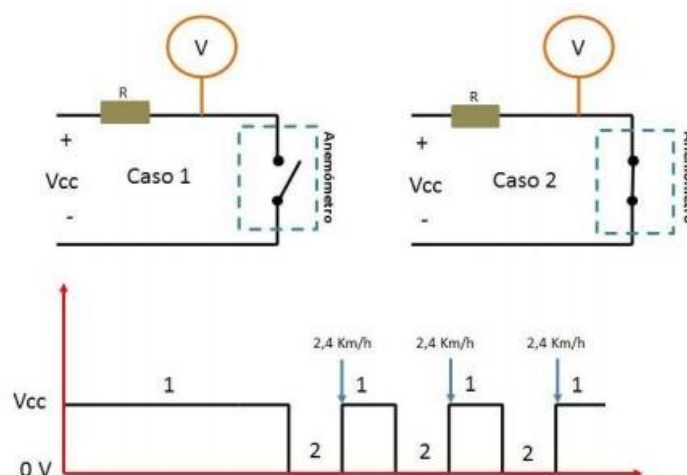
*Acondicionamiento de Señal del pluviómetro*



El mismo proceso se aplica para un anemómetro, al medir el viento se nos muestra que trabaja con una señal PWM, como ya habíamos dicho para la simulación podemos usar este un generador de señales. La velocidad de viento se mide por la frecuencia del mismo y la amplitud en estos sensores.

**Figura 26**

*Acondicionamiento de señal del anemómetro*

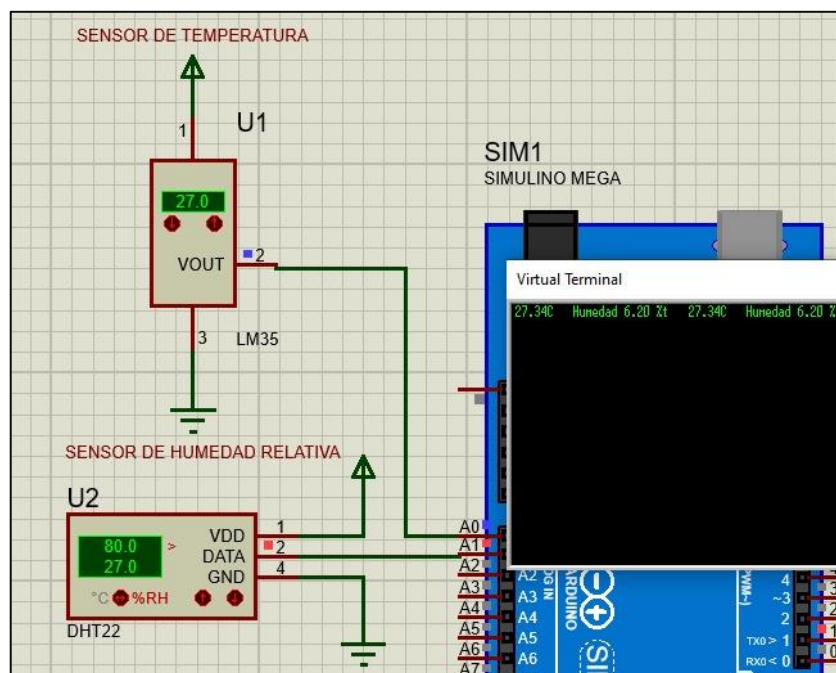


Para la obtención de la simulación del sensor de humedad podemos hacer uso del DHT22. Este sensor permite obtener tanto la temperatura y la humedad. A

pesar de ello, la medida de la temperatura del DHT22 puede ser afectada pasando el tiempo. Por ello debemos colocar un sensor que añada robustez al sistema. Un sensor extra que podemos poner es el sensor LM35, este sensor permite una medida extra de temperatura. a continuación, se muestra la simulación en proteus de los sensores

**Figura 27**

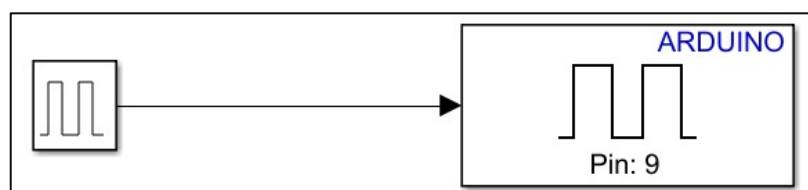
*Simulación del DHT22 y LM35*



La simulación en proteus nos puede ayudar para simular el acondicionamiento, ya sabiendo el acondicionamiento pasaremos a usar el Simulink de Matlab. Para cargar en tiempo real nuestro sistema debemos configurar el Simulink. Para ello realizamos primero una prueba en tiempo de real, la prueba consiste en prender y apagar un led a través de Simulink

**Figura 28**

*Prueba de comunicación entre Simulink y Arduino*



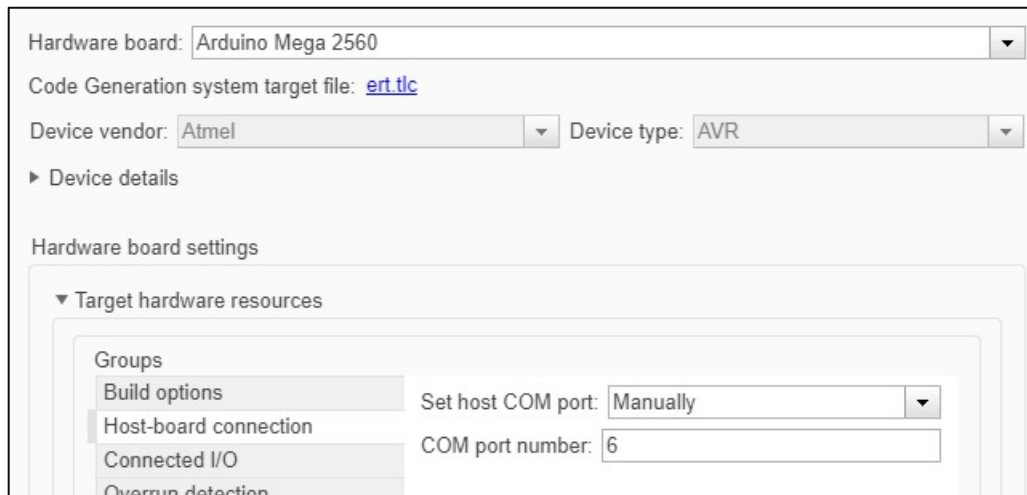
*[Handwritten signatures]*



Configurando nuestro Arduino, el puerto a usar y los baudios utilizados, podremos realizar una buena conexión.

### Figura 29

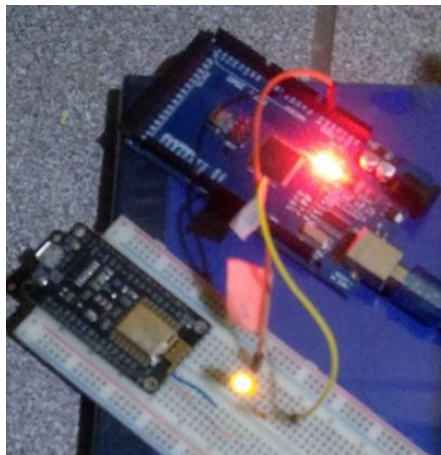
*Configuración de Hardware y puerto serial en Simulink*



Se muestra que Simulink nos permite trabajar en tiempo real, además de programar directamente el Arduino al estar conectado. Con los siguientes softwares podremos obtener gráficas y realizar el acondicionamiento debido a los sensores

### Figura 30

*Demostración de comunicación entre Simulink y Arduino*



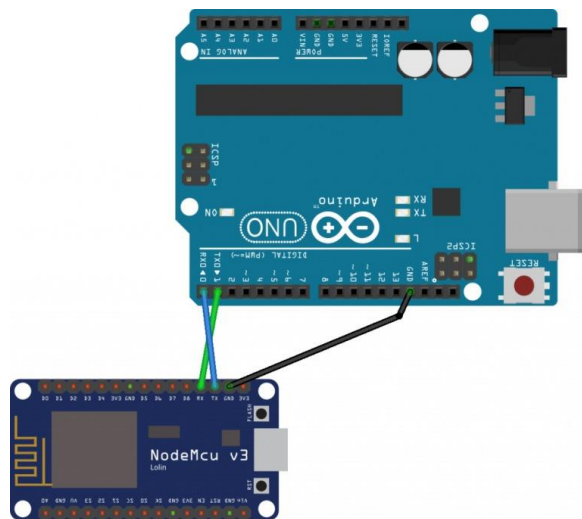
*[Handwritten signature]*  
Nishit

También es necesario realizar una comunicación serial entre el Arduino y el NodeMCU, ello lo hacemos ya que el módulo Wifi mandara las señales a la nube. El módulo de Wifi deberá adquirir todas las señales de nuestro sistema que le enviará el Arduino Mega y enviarlas a ThingSpeak para una monitorización exitosa.

En la imagen siguiente podremos ver la conexión entre el Arduino Mega y el NodeMcu.

### Figura 31

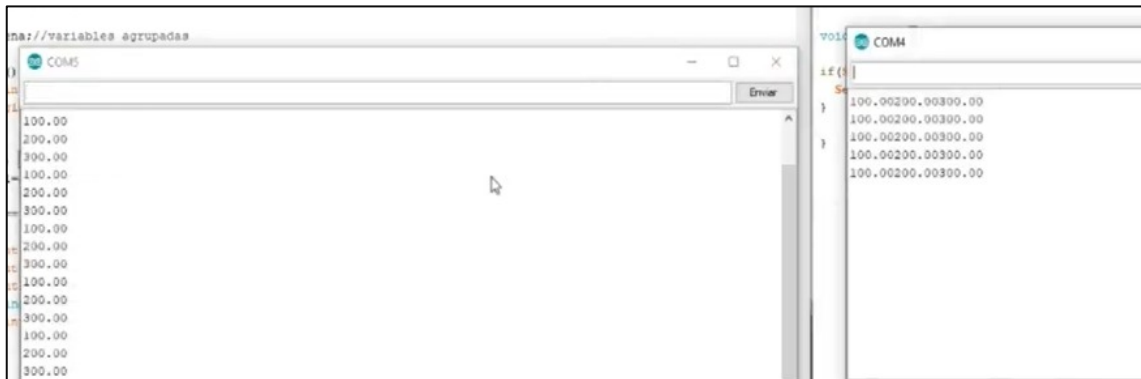
*Diagrama de conexión de ESP8266 y Arduino*



También debe tenerse en cuenta que debemos enviar una cadena de caracteres por medio del Arduino Mega, el módulo Wifi lo recibirá en cadena de caracteres y luego deberemos separar los datos, A continuación, se muestra como seria este proceso

**Figura 32**

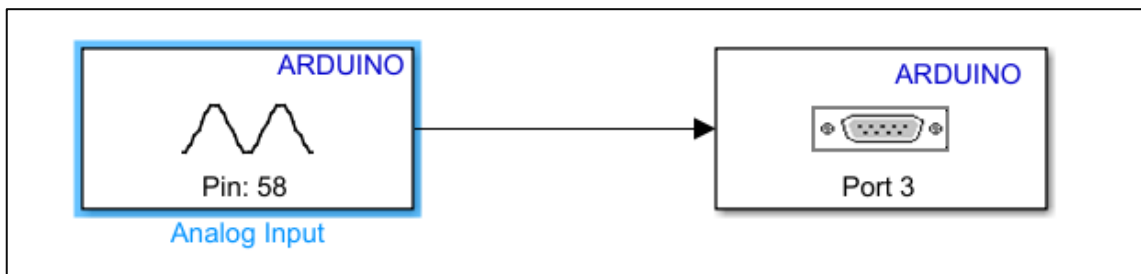
*Comunicación Serial entre Arduino y NodeMCU*



Nuestra programación de Arduino a través de simulink se realizaría de la siguiente manera, primero se deberá seleccionar el bloque Analog Input y se tendrá que escoger el pin 0. Luego se elegirá el puerto Serial 3, que sería el puerto serial que se usará del Arduino para la comunicación serial.

**Figura 33**

*Diagrama de bloques de Simulink*



### **CÓDIGO DE PRUEBA EN NODEMCU**

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

Handwritten signature and initials in black ink.

```

void loop() {

  if(Serial.available()>0){

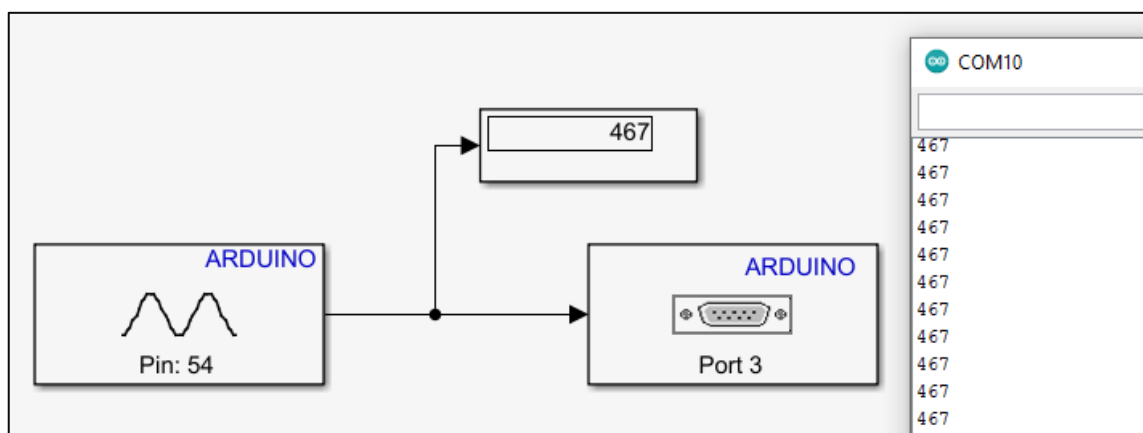
    Serial.write(Serial.read());

    delay(1); } }

```

### Figura 34

Testeo de comunicación entre Arduino, Simulink y el NodeMCU



Podemos hacer un testeo rápido con este código, sin embargo, los datos que necesitaremos mandar al NodeMCU, necesitan ser números y por lo tanto necesitaremos solucionar dos problemas, el primero sería realizar la conversión de ASCII a número y usar un delay el cual permita leer todos los números que transmitimos.

Para corregir el problema de los caracteres ASCII podemos hacer la conversión de entero a carácter usando (char) y crear un buffer para almacenar los caracteres que se están recibiendo, para que finalmente nos imprima el resultado del buffer cuando termine de recolectar la información.

Se tiene que mencionar que el delay utilizado es referencial para esta aplicación, pero puede cambiar ya que guarda relación con la cantidad de caracteres que se desee leer.

*[Firma manuscrita]*  
*[Firma manuscrita]*

Si deseamos enviar mas de un numero de un digito al NodeMCU , necesitaremos realizar una conversión de String a int, para esta acción podemos usar el comando toInt():

### **CÓDIGO PARA RECIBIR NÚMEROS DE MAS DÍGITOS POR MEDIO DEL NODEMCU.**

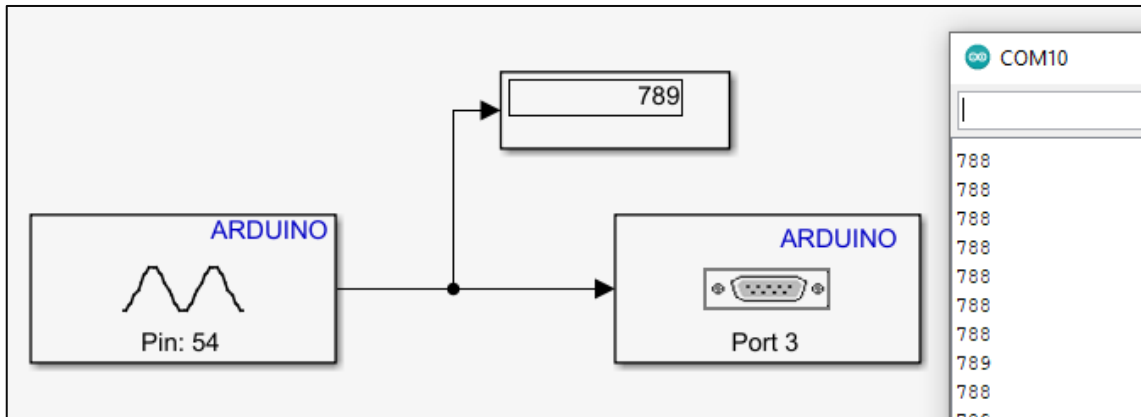
```
void setup() {  
  
  Serial.begin(9600);  
  
}  
  
void loop() {  
  
  if (Serial.available() > 0) {  
  
    //Delay especifico para la lectura de caracteres  
  
    delay(10);  
  
    //Creación de buffer  
  
    String bufferString = "";  
  
    while (Serial.available() > 0) {  
  
      bufferString += (char)Serial.read();  
  
    }  
  
    //Conversión de buffer a numero  
  
    int num = bufferString.toInt();  
  
    //Imprimimos el numero
```


```
Serial.println(num); } }
```

**Figura 35**

*Envío de números a través del puerto serial desde el Arduino al NodeMCU.*



La diferencia entre el código de prueba y el código expuesto ahora es que en el código de prueba los datos no están convertidos a números, son un conjunto de caracteres lo cuales no podemos usarlos si deseamos hacer una operación.

Una vez comunicado el Arduino, Simulink , NodeMCU pasaremos a realizar la comunicación entre el NodeMCU y Ubidots. Para ello será necesario descargar la librería de Ubidots , además de configurar nuestro acceso a Ubidots para que se transmitan los datos.

**Figura 36**

*Selección de variable a usar en Ubidots*



Tener en cuenta que además de configurar la variable que usaremos, debemos copiar nuestro Ubidots Token, esto nos permitirá conectar el NodeMCU con la página.

### Figura 37

Obtención de Ubidots token a través de su pagina



A continuación, se muestra el código para la programación.

### CÓDIGO PARA RECIBIR NÚMEROS DE MAS DÍGITOS POR MEDIO DEL NODEMCU.

```
#include <Ubidots.h>
```

```
const char* UBIDOTS_TOKEN = "BBFF-YSSI91Ex03xDgPc9BVK2V6jE5glgZV";  
// Ingrese el Ubidots TOKEN
```

```
const char* WIFI_SSID = "_____"; // Ingrese el Wi-Fi SSID
```

```
const char* WIFI_PASS = "_____"; // Ingrese el Wi-Fi password
```

```
Ubidots ubidots(UBIDOTS_TOKEN, UBI_HTTP);
```

```
int var =0; //variable donde se almacenará la información
```

```
int var1=-2; //variable para solucionar errores de datos no enviados
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    ubidots.wifiConnect(WIFI_SSID, WIFI_PASS);}
```

Two handwritten signatures in black ink. The top one is a stylized signature, and the bottom one is a more legible signature that appears to read "Nishu" followed by a symbol.

```

void loop() {

    while(var ==-2 || var1==0){

    if (Serial.available() > 0) {

        //Delay variable para cada aplicación

        delay(5);

        //Creación del buffer

        String bufferString = "";

        while (Serial.available() > 0) {

            bufferString += (char)Serial.read(); }

            var1 = bufferString.toInt();

            var = var1;

            if (var1!=0){

                var1 = 1; } } }

            ubidots.add("var", var);// Ingresamos la variable que manderemos a Ubidots

            bool bufferSent = false;

            bufferSent = ubidots.send(); // Envió de datos a una etiqueta de dispositivos que
            coincidan con el ID del dispositivo

            if (bufferSent) {

                // Mensaje para saber si los datos fueron enviados

                Serial.println("Valor enviado por el dispositivo"); }

```





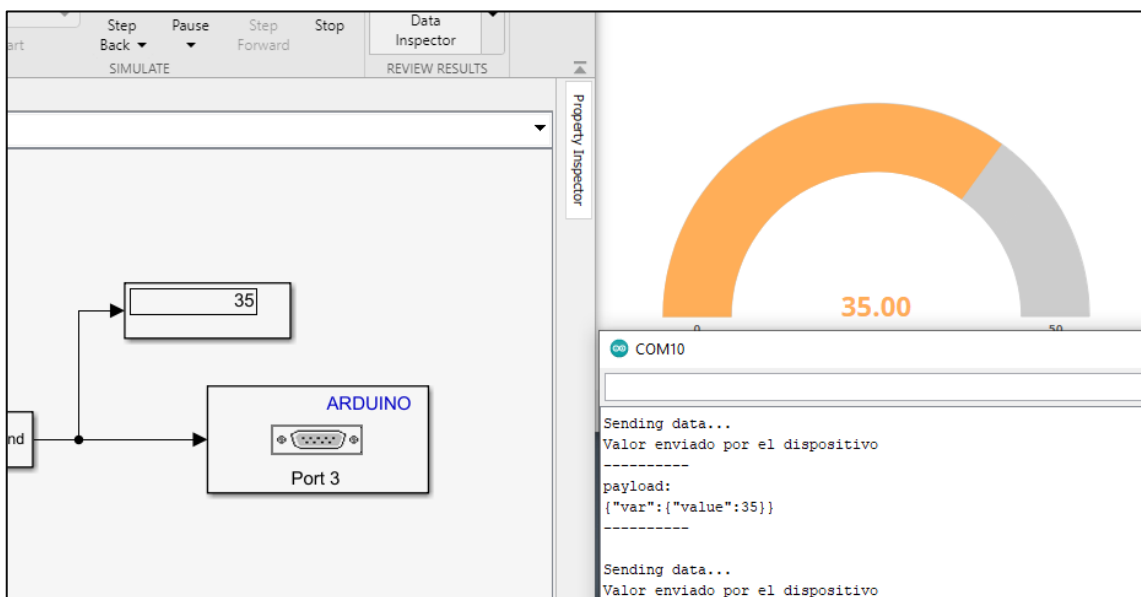

```
delay(1000); // delay usado
```

```
var1=-2; }
```

Se puede ver a continuación como se programa el Arduino para la comunicación Serial con el NodeMCU , que a su vez se comunica con la página de Ubidots.

### Figura 38

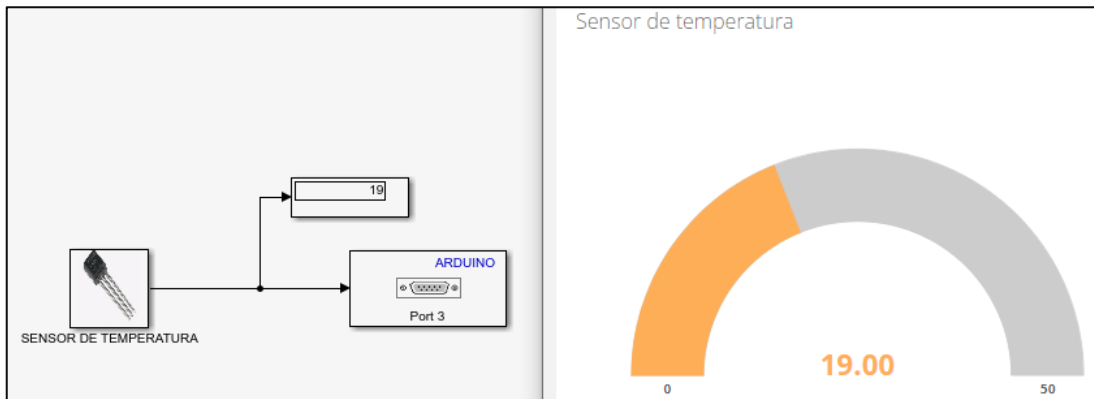
*Comunicación del NodeMCU con la página Ubidots*



Habiendo mencionado lo anterior podemos apoyarnos de simulink para hacer el acondicionamiento de señales de los diversos sensores que tiene una estación meteorológica

### Figura 39

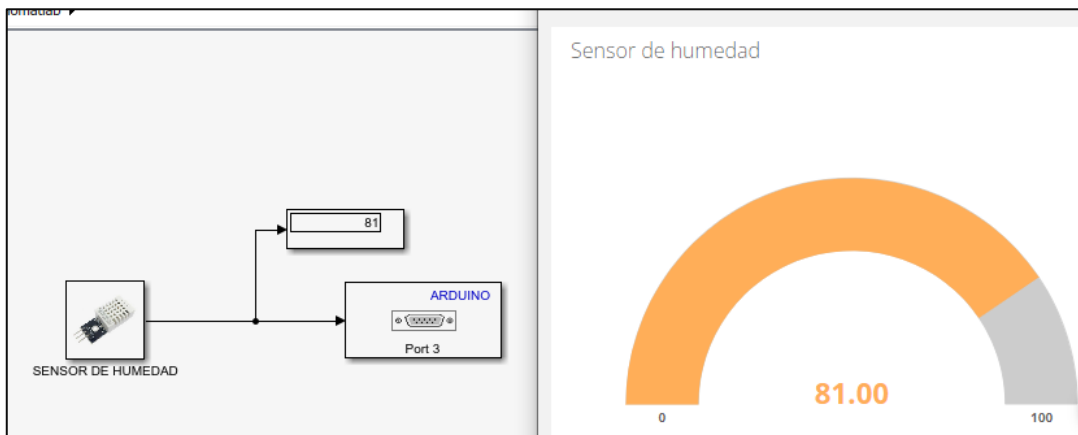
#### Comunicación del sensor de temperatura y Ubidots



La programación la tendríamos que realizar en Simulink , y ingresar los datos de los sensores a través del Arduino para poder hacer el acondicionamiento.

### Figura 40

#### Comunicación del sensor de DTH22 y Ubidots

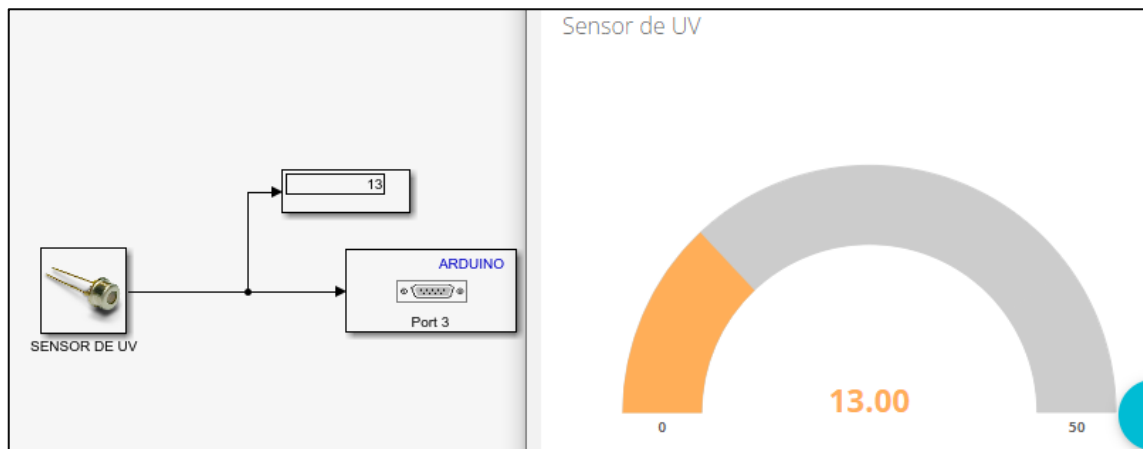


Se puede acondicionar señales de otros sensores más avanzados, previo acondicionamiento de su alimentación. (esto no podría hacerlo en Simulink)

Handwritten signature and initials, likely belonging to the author or a reviewer.

## Figura 41

### Comunicación del sensor de UV y Ubidots



## 4.3 Población y muestra

La población está determinada por la información recopilada a lo largo de 10 años.

### Características

- Análisis de la información
- Procesamiento de la Información

### Delimitación

- Modelos de Investigación Aplicada

## 4.4 Lugar de estudio y periodo desarrollado

- Ciudad de Lima
- Periodo: 2021-2022

## 4.5 Técnicas e instrumentos para la recolección de la información

Análisis Documental (Revisión de Bibliografía)

Observación Experimental (Simulaciones)

## 4.6 Análisis y procesamiento de datos

Descripción de datos. Estadísticos de una variable

Distribuciones de probabilidad e intervalos de confianza



Núñez

## CAPITULO V

### RESULTADOS

#### 5.1. Resultados Descriptivos

El resultado descriptivo realizado, nos ayuda en la comprensión de como varían los parámetros a medida de la toma de muestras en un intervalo de tiempo. Y como sobre este resultado actúan diferentes factores como el entorno, los errores de señales análogas y el ruido en el proceso.

También tenemos la información obtenida por el software SIMULINK el cuál conecta la programación con el hardware del diseño, y los resultados son enviados a un ESP32 el cual posteriormente lo enviará a UBIDOTS

#### 5.2. Resultados Inferenciales

Una vez realizado la etapa de implementación y simulación, realizamos la toma de datos de las variables descritas en la investigación: Temperatura, Precipitación, Índice UV y Humedad

Los datos de todos los sensores fueron obtenidos el día 20 de diciembre de 9:00 de la mañana a 10:00 de la mañana. Los resultados registrados fueron los siguientes:



Handwritten signature and initials.

**Temperatura:**

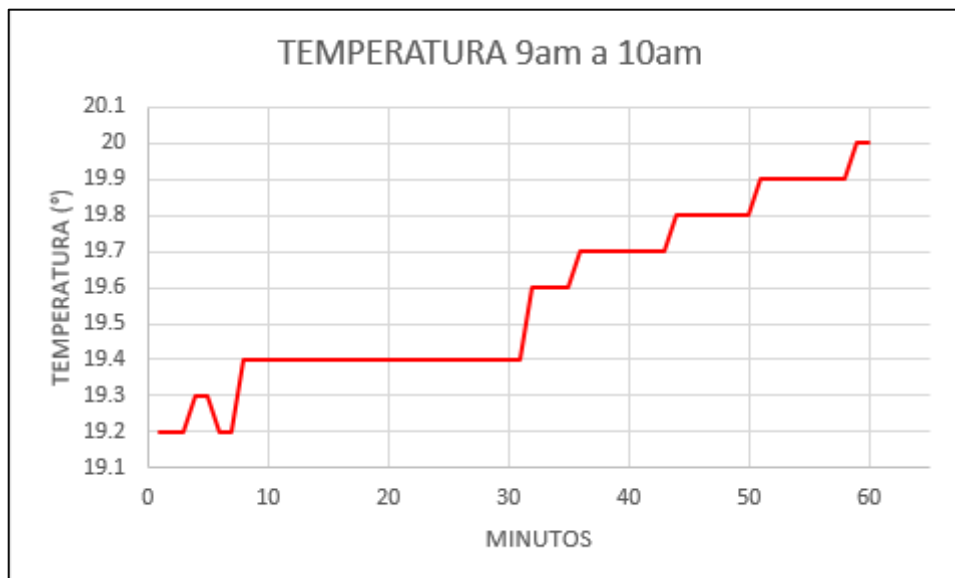
MIN	TEMPERATURA (°)
1	19.2
2	19.2
3	19.2
4	19.3
5	19.3
6	19.2
7	19.2
8	19.4
9	19.4
10	19.4
11	19.4
12	19.4
13	19.4
14	19.4
15	19.4
16	19.4
17	19.4
18	19.4
19	19.4
20	19.4
21	19.4
22	19.4
23	19.4
24	19.4
25	19.4
26	19.4
27	19.4
28	19.4
29	19.4
30	19.4
31	19.4
32	19.6
33	19.6
34	19.6
35	19.6
36	19.7
37	19.7
38	19.7
39	19.7
40	19.7
41	19.7
42	19.7
43	19.7
44	19.8
45	19.8
46	19.8
47	19.8
48	19.8
49	19.8
50	19.8
51	19.9
52	19.9
53	19.9
54	19.9
55	19.9
56	19.9
57	19.9
58	19.9
59	20
60	20

Tabla N°1: Resultados obtenidos de temperatura

Obteniendo una gráfica ascendente:

**Figura 42**

*Gráfica de Temperatura*



*[Handwritten signatures]*

**Humedad:**

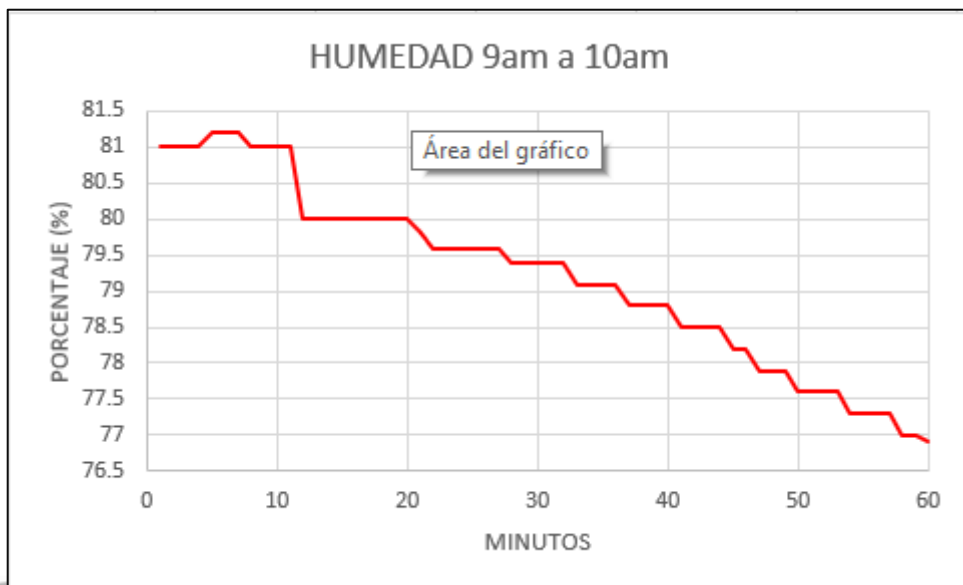
MIN	%
1	81
2	81
3	81
4	81
5	81.2
6	81.2
7	81.2
8	81
9	81
10	81
11	81
12	80
13	80
14	80
15	80
16	80
17	80
18	80
19	80
20	80
21	79.8
22	79.6
23	79.6
24	79.6
25	79.6
26	79.6
27	79.6
28	79.4
29	79.4
30	79.4
31	79.4
32	79.4
33	79.1
34	79.1
35	79.1
36	79.1
37	78.8
38	78.8
39	78.8
40	78.8
41	78.5
42	78.5
43	78.5
44	78.5
45	78.2
46	78.2
47	77.9
48	77.9
49	77.9
50	77.6
51	77.6
52	77.6
53	77.6
54	77.3
55	77.3
56	77.3
57	77.3
58	77
59	77
60	76.9

Tabla N°2: Resultados obtenidos de humedad

Obteniéndose la siguiente gráfica que se muestra en la Figura 43.

**Figura 43**

*Gráfica de Humedad*



*[Handwritten signatures]*

**Índice UV:**

MIN	INDICE UV
1	13
2	13
3	13
4	13.1
5	13.1
6	13.1
7	13.1
8	13.1
9	13.1
10	13
11	13
12	13
13	13
14	13
15	13
16	13
17	13
18	13
19	13

20	13
21	13
22	13
23	13
24	13
25	13
26	13.1
27	13.1
28	13.1
29	13.1
30	13.1
31	13.1
32	13.1
33	13.1
34	13.1
35	13.1
36	13.1
37	13.1
38	13.1
39	13.1
40	13.1

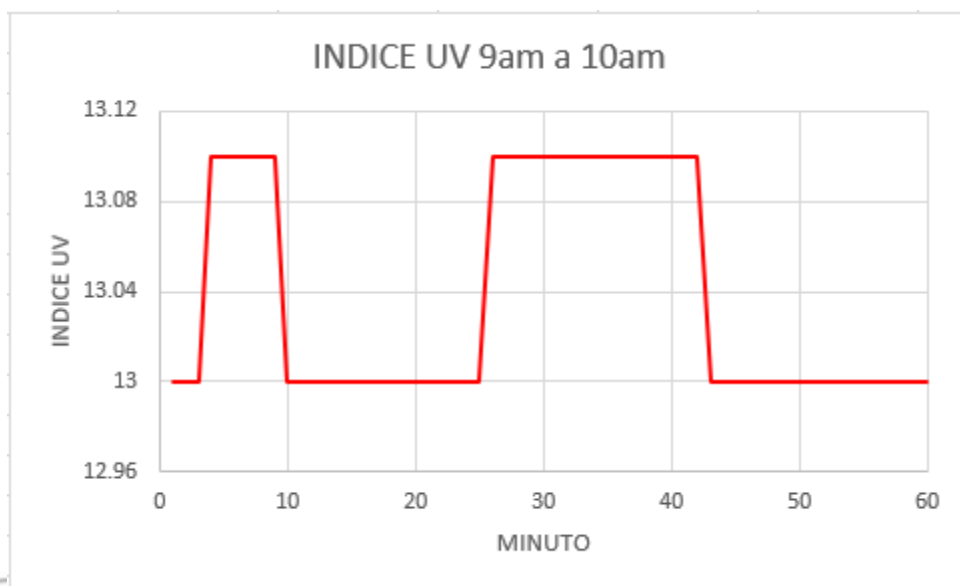
41	13.1
42	13.1
43	13
44	13
45	13
46	13
47	13
48	13
49	13
50	13
51	13
52	13
53	13
54	13
55	13
56	13
57	13
58	13
59	13
60	13

Tabla N°3: Resultados obtenidos de índice UV

Obteniéndose la siguiente gráfica que se muestra en la Figura 44:

**Figura 44**

*Gráfica de Índice UV*



*[Handwritten signatures]*



**Precipitación:**

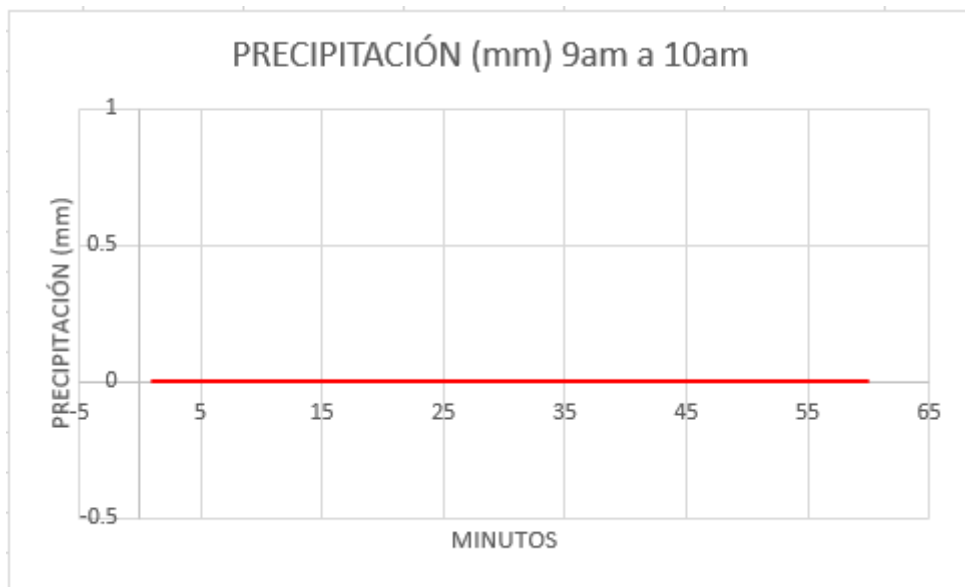
MIN	PRECIPITACIÓN (mm)				
1	0	20	0	41	0
2	0	21	0	42	0
3	0	22	0	43	0
4	0	23	0	44	0
5	0	24	0	45	0
6	0	25	0	46	0
7	0	26	0	47	0
8	0	27	0	48	0
9	0	28	0	49	0
10	0	29	0	50	0
11	0	30	0	51	0
12	0	31	0	52	0
13	0	32	0	53	0
14	0	33	0	54	0
15	0	34	0	55	0
16	0	35	0	56	0
17	0	36	0	57	0
18	0	37	0	58	0
19	0	38	0	59	0
		39	0	60	0
		40	0		

Tabla N°4: Resultados obtenidos de precipitación

Obteniendo la siguiente gráfica que se muestra en la Figura 45.

**Figura 45**

*Gráfica de Precipitación*



*[Firma manuscrita]*  
*N. H. H. - c*

## CAPÍTULO VI

### DISCUSIÓN DE RESULTADOS

#### **Contrastación y demostración de la hipótesis con los resultados**

El acondicionamiento de sensores de bajo costo para la obtención de señales aceptables en una estación meteorológica automática fue exitoso. Obtuvimos una señal limpia de ruido y los datos fueron enviados a internet para su posterior monitoreo. De esta manera corroboramos nuestra hipótesis y verificamos los resultados obtenidos con resultados de los mismos parámetros de diversas investigaciones y páginas meteorológicas.



Handwritten signature and initials.

## **CONCLUSIONES:**

Se diseñó un acondicionamiento de señales para sensores de bajo costo para una estación meteorológica automática, permitiendo demostrar que un eficiente acondicionamiento para la eliminación del ruido de la señal puede reducir costos en operación. A su vez, se diseñó un método de visualización por medio de la plataforma UBIDOTS en donde los datos se muestran en tiempo real. De esta manera se concluye que el dispositivo es funcional, mejorable y eficiente para medir variables a tiempo real.

## **RECOMENDACIONES:**

- A través de la comunicación serial se puede transmitir caracteres en vez de números y para poder ser usados, todo tipo de dato debe ser convertido a formato numérico.
- En la relación de Simulink con Arduino, se recomendaría usar un tiempo de muestreo de 0.02
- Tanto en Simulink como Arduino y ESP se debe usar el mismo formato de tipo de variable, o usar el bloque Round que permite redondear todo decimal a entero
- Los datos obtenidos en la toma de resultados deben ser corroborados con páginas de meteorología para verificar la veracidad y precisión de los valores obtenidos de sensores baratos con acondicionamiento.



Handwritten signature and initials.

## REFERENCIAS BIBLIOGRÁFICAS

1. (¿Cómo conectar NodeMCU con Arduino? Conexión serie UART, i2C y SPI, 2017) ¿Cómo conectar NodeMCU con Arduino? Conexión serie UART, i2C y SPI. (2017, November 18). BorrowBits. <https://borrowbits.com/2017/11/como-comunicar-arduino-con-nodemcu-parte-i-conexion-serieuart/>
2. (“Módulos ESP8266,” 2017) Módulos ESP8266. (2017, April 16). UN ELECTRONICA. <https://unelectronica.github.io/Modulos-esp8266/>
3. (Conversión Analogo Digital PIC, 2015). Conversión Analogo Digital PIC. (2015, August 1). Control Automático Educación. <https://controlautomaticoeducacion.com/microcontroladores-pic/14-conversion-analogodigital-ad/>
4. (EPD-440-0-1.4 - Roithner Lasertechnik, n.d.) EPD-440-0-1.4 - Roithner Lasertechnik. (n.d.). Gophotonics.Com. Retrieved January 11, 2022, from <https://www.gophotonics.com/products/photodiodes/roithner-lasertechnik-gmbh/56-452-epd-440-0-1-4>
5. (Estaciones automáticas, n.d.) Estaciones automáticas. (n.d.). Navarra.Es. Retrieved January 11, 2022, from [http://meteo.navarra.es/definiciones/estaciones\\_automaticas.cfm](http://meteo.navarra.es/definiciones/estaciones_automaticas.cfm)
6. (Qué es Adquisición de Datos - DAQ or DAS?, n.d.) Qué es Adquisición de Datos - DAQ or DAS? (n.d.). Dewesoft.com. Retrieved January 11, 2022, from <https://dewesoft.com/es/daq/que-es-adquisicion-de-datos>
7. (Sensor de Temperatura y humedad DHT22, n.d.). Sensor de Temperatura y humedad DHT22. (n.d.). Circuitos y Desarrollo en Tecnología - C&D Tecnología. Retrieved January 11, 2022, from <https://cdtecnologia.net/sensores/61-sensor-de-temperatura-y-humedad-dht22.html>
8. (SensorGO, 2020) SensorGO, M. K. T. (2020, September 15). Estaciones Meteorológicas. ¿Qué son y cómo funcionan? SensorGO. <https://sensorgo.mx/estaciones-meteorologicas/>



9. Agro Rural implementó 22 estaciones meteorológicas en 5 regiones del país para mejorar la siembra de las cosechas. (s/f). Gob.pe. Recuperado el 11 de enero de 2022, de <https://www.gob.pe/institucion/agrorural/noticias/574177-agro-rural-implemento-22-estaciones-meteorologicas-en-5-regiones-del-pais-para-mejorar-la-siembra-de-las-cosechas>
10. Arduino con Simulink. (2021, junio 24). Control Automático Educación. <https://controlautomaticoeducacion.com/arduino/arduino-con-simulink/>
11. Arduino Mega 2560 Ch340 con cable – DeTodoMaker.com. (s/f). Detodomaker.com. Recuperado el 11 de enero de 2022, de <https://detodomaker.com/producto/arduino-mega-2560-ch340-con-cable/>
12. Arduino. (2021, mayo 4). *¿Qué es Arduino y por qué se ha convertido en un estándar informal en educación y creación de prototipos?* automaticaeinstrumentacion.com. <https://www.automaticaeinstrumentacion.com/texto-diario/mostrar/2867420/arduino-convertido-estandar-informal-educacion-creacion-prototipos>
13. Arteaga Ramirez, R., Cervantes Osornio, R., Vásquez Peña, M., & Ojeda Bustamante, W. (2017). Estación meteorológica convencional versus automática sus diferencias evaluadas con índices estadísticos . *Revista Mexicana de ciencias agrícolas*, 1301-1314.
14. ATMega2560-16AU Microcontrolador Atmel 8-Bit. (s/f). Microjpm.com. Recuperado el 11 de enero de 2022, de <https://www.microjpm.com/products/ad32796/>
15. Aybar Camacho, C., & Lavado Casimiro, W. (2017). *Atlas de zonas de vida del Perú : guía explicativa*. Lima: Servicio Nacional de Meteorología e Hidrología del Perú - SENAMHI.
16. CÁRDENAS ROJAS, J. (2017). IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE INFORMACIÓN DE LAS ESTACIONES METEOROLÓGICAS, EN LA CUENCA DEL RIO CACHI, REGIÓN AYACUCHO. *Tesis*. UNIVERSIDAD PERUANA LOS ANDES, HUANCAYO - PERÚ.



17. Daniel. (s/f). Internet de las cosas: “Envío a Thingspeak de datos de temperatura, humedad y sonido”. Tecnopatafisica.com. Recuperado el 11 de enero de 2022, de <https://tecnopatafisica.com/tecno3eso/teoria/robotica/74-internet-de-las-cosas-envio-a-thingspeak-de-datos-de-temperatura-humedad-y-sonido>.
18. Del Valle Hernández, L. (2019, mayo 7). ADS1115 convertidor analógico digital ADC para Arduino y ESP8266. Programar fácil con Arduino. <https://programarfácil.com/blog/arduino-blog/ads1115-convertidor-analogico-digital-adc-arduino-esp8266/>
19. Embajadores, E. (n.d.). TMP36GT9Z - Sensor de Temperatura. Electrónica Embajadores. Retrieved January 11, 2022, from <https://www.electronicaembajadores.com/es/Productos/Detalle/SSTETMP36/sensores/sensores-de-temperatura/tmp36gt9z-sensor-de-temperatura>
20. ESP8266. (2020, mayo 29). Todo Sobre Microcontroladores. <https://microcontroladores.com/esp8266/>
21. Ferrer Sanabria, R. D. (2017). Desarrollo de una estación meteorológica autónoma de bajo costo. *Tesis*. Universidad Santo Tomás Bucaramanga, Bucaramanga.
22. Fraga, A. I. (2022, enero 10). Internet de las Cosas: la tecnología que culmina el viejo sueño de conectarlo todo. El Español. [https://www.elespanol.com/invertia/disruptores-innovadores/innovadores/investigacion/20220110/internet-cosas-tecnologia-culmina-viejo-sueno-conectarlo/640686032\\_0.html](https://www.elespanol.com/invertia/disruptores-innovadores/innovadores/investigacion/20220110/internet-cosas-tecnologia-culmina-viejo-sueno-conectarlo/640686032_0.html)
23. Gerardo de Estaciondemeteorologia.com. (2019, febrero 26). ¿QUÉ ES UNA ESTACIÓN METEOROLÓGICA? *METEOCULTURA*. <https://estaciondemeteorologia.com/que-es-una-estacion-meteorologica/>
24. Gómez Vega, S. (2017). NIVELES DE RADIACIÓN ULTRAVIOLETA (UV) EN LA CIUDAD DE AYACUCHO PERIODO 2006 - 2011 E INCIDENCIA DE CÁNCER DE PIEL EN LA POBLACIÓN. *Tesis*. UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA, AYACUCHO – PERÚ.
25. GUILLÉN FERNÁNDEZ, G. M. (2018). IDENTIFICACIÓN ESPACIAL DE ZONAS VULNERABLES ANTE LLUVIAS EXTREMAS EN LIMA



METROPOLITANA Y EL CALLAO, A PARTIR DE LOS IMPACTOS REPORTADOS POR LA INUSUAL LLUVIA DEL 15 DE ENERO DE 1970 Y OTROS EVENTOS. *Tesis*. PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ, LIMA.

26. inter-neg. (n.d.). Temperatura. Com.Ar. Retrieved January 11, 2022, from <http://www.termosistemas.com.ar/sitio/vernota.php?nota=247&cat=1>
27. Isaac. (2019, octubre 11). DHT22: el sensor de temperatura y humedad de precisión. Hardware libre. <https://www.hwlibre.com/dht22/>
28. MEDINA MOLINA , C. A., & RAMIREZ GALVIS, W. F. (2019). DISEÑO Y CONSTRUCCION DE ESTACION METEOROLOGICA, UBICADA EN LA VEREDA EL CALLEJON, RICAURTE-CUNDINAMARCA. *Tesis*. Universidad Católica de Colombia, Bogotá.
29. Medina, L., Ramón, V., & Fresdo, V. (2017). Comparación de la temperatura, precipitación, velocidad media y dirección del viento medidas por dos estaciones meteorológicas, ubicadas en Punta de Piedras, isla de Margarita. *Memoria de la Fundación La Sala de Ciencias Naturales*, 61-80.
30. Murcia Almanza, D. S., & Rojas Paternina, A. (2015). DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN METEOROLÓGICA REMOTA DE BAJO COSTO EMPLEANDO EL ENFOQUE DE INTERNET DE LAS COSAS. *Tesis*. UNIVERSIDAD PILOTO DE COLOMBIA, Bogotá D.C.
31. Organización Mundial de la Salud. (2002). *Índice UV solar mundial: Guía práctica*.
32. PDAControl. (2019, diciembre 10). Actualización Librerías Ubidots V3.0.0 para módulos ESP8266. PDAControl. <http://pdacontroles.com/actualizacion-librerias-ubidots-v3-0-0-para-modulos-esp8266/>
33. Polo Bravo, C., Torres Muro, H., Ríos Marquezado, J., Bernabé Coaquera, R., & De la Gala Contreras, A. (2017). Diseño, construcción y sistematización de una estación metereológica de registro automático y procesamiento de tres parámetros basado en la implementación de hardware libre. *Ciencias*, 30-37.



34. Protocolos y tecnologías de IoT. (s/f). Microsoft.com. Recuperado el 11 de enero de 2022, de <https://azure.microsoft.com/es-es/overview/internet-of-things-iot/iot-technology-protocols/>
35. Redacción. (2019, marzo 30). NodeMCU una plataforma para IOT de código abierto. Descubrearduino.com. <https://descubrearduino.com/nodemcu/>
36. Rodríguez Jiménez, R., Benito Capa, A., & Portela Lozano, A. (2004). Meteorología y Climatología. Madrid: FECYT (*Fundación Española para la Ciencia y la Tecnología*).
37. Sabino, E.; Lavado W. y Aybar C. (2019) Estimación de las zonas de vida de Holdridge en el Perú, Servicio Nacional De Meteorología E Hidrología Del Perú –Senamhi. Dirección de Hidrología -DHI. p. 5
38. SENAMHI - Perú. (s/f). Gob.pe. Recuperado el 11 de enero de 2022, de <https://www.senamhi.gob.pe/?p=prensa&n=498>
39. Sensor LM35. (2020, mayo 14). Blog Arduino, LabVIEW y Electrónica. <https://electronicamade.com/sensor-lm35/>
40. soloelectronicos. (2017, noviembre 22). APIs para IoT. Soloelectronicos.com. <https://soloelectronicos.com/2017/11/23/apis-para-iot/>
41. Vargas Seminario, F. (2015). Sistema de información en línea para el seguimiento del fenómeno El Niño en la costa norte del Perú. *Tesis*. Universidad de Piura, Piura.
42. Vasquez Reynaldo, J. Y. (2015). Cambio de uso del suelo y su efecto en la temperatura y humedad del ambiente de la Universidad Peruana Unión, Ñaña, Lima, Perú. *Tesis*. UNIVERSIDAD PERUANA UNIÓN, Lima.
43. Viento. (s/f). Navarra.es. Recuperado el 11 de enero de 2022, de <http://meteo.navarra.es/definiciones/viento.cfm>
44. VILLALTA CRUZ, H. A., & SORTO PERDOMO, G. (2013). Implementación de una estación meteorológica. *Tesis*. UNIVERSIDAD DE EL SALVADOR, San Salvador.



Handwritten signature in black ink, appearing to be 'N. Villalta Cruz'.



45. Zuñiga Vinueza, W., Polanco Risquet, A., & Nagy, A. S. (2016). Prototipo de mini estación meteorológica automática inalámbrica. . *Publicando*, 20-32

# ANEXOS

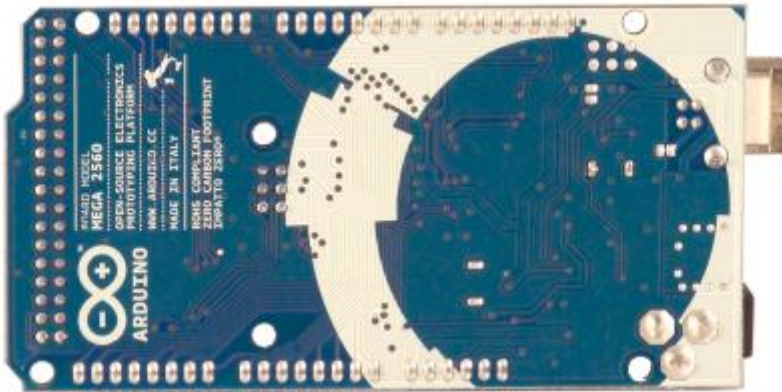


Nagy A. S.

## ANEXO Nº1: MATRIZ DE CONSISTENCIA

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES	METODOLOGÍA	POBLACIÓN
<p><b>Problema General:</b> ¿Es posible diseñar un sistema que nos permita acondicionar las señales obtenidas por medio de los sensores de una estación meteorológica automática de bajo costo para obtener señales más precisas de los parámetros ambientales y que a su vez este sistema envíe los datos a internet para su respectivo monitoreo?</p> <p><b>Problemas Específicos:</b> ¿Se puede realizar un sistema capaz de poder acondicionar las señales provenientes de los sensores de una estación meteorológica automática de bajo costo? ¿Sera posible usar la conversión ADC de un microcontrolador para las señales de los sensores de una estación meteorológica automática de bajo costo? ¿Existirá diferencia entre las señales filtradas y señales no filtradas de los sensores de una estación meteorológica automática de bajo costo? ¿Se podrá realizar una conexión exitosa entre un microcontrolador y la red de internet? ¿Es posible realizar un monitoreo en tiempo real vía internet de las señales filtradas de los sensores de una estación meteorológica automática de bajo costo?</p>	<p><b>Objetivo General:</b> Diseñar un sistema que nos permita acondicionar las señales obtenidas por medio de los sensores de una estación meteorológica automática de bajo costo para obtener señales más precisas de los parámetros ambientales y que a su vez este sistema envíe los datos a internet para su respectivo monitoreo</p> <p><b>Objetivos Específicos:</b> Realizar un sistema capaz de poder acondicionar las señales provenientes de los sensores de una estación meteorológica automática de bajo costo. Usar la conversión ADC de un microcontrolador para las señales de los sensores de una estación meteorológica automática de bajo costo. Comprobar la diferencia entre las señales filtradas y señales no filtradas de los sensores de una estación meteorológica automática de bajo costo. Realizar una conexión exitosa entre un microcontrolador y la red de internet. Realizar un monitoreo en tiempo real vía internet de las señales filtradas de los sensores de una estación meteorológica automática de bajo costo.</p>	<p><b>Hipótesis General:</b> El desarrollo de un sistema de acondicionamiento para las señales obtenidas por medio de los sensores de una estación meteorológica automática de bajo costo para obtener señales más precisas de los parámetros ambientales y que a su vez este sistema envíe los datos a internet para su respectivo monitoreo.</p> <p><b>Hipótesis Específicas:</b> Es posible diseñar un sistema capaz de acondicionar señales provenientes de los sensores de una estación meteorológica automática de bajo costo Se puede hacer uso de la conversión ADC de un microcontrolador para las señales de los sensores de una estación meteorológica automática de bajo costo. Existen diferencias entre las señales filtradas y señales no filtradas de los sensores de una estación meteorológica automática de bajo costo Es posible realizar una conexión exitosa entre un microcontrolador y la red de internet Es posible realizar un monitoreo en tiempo real vía internet de las señales filtradas de los sensores de una estación meteorológica automática de bajo costo</p>	<p><b>Variable independiente:</b> - Variable climatológicas</p> <p><b>Dimensiones:</b> - Temperatura. - Humedad. - Radiación Solar. - Precipitación. - Velocidad del viento</p> <p><b>Variable dependiente:</b> Calidad de las variables climatológicas</p> <p><b>Dimensiones:</b> - Señal de temperatura acondicionada - Señal de humedad acondicionada - Índice de radiación Solar acondicionado - Señal de pulsos de precipitación acondicionado - Señal de pulsos de viento acondicionado</p>	<p><b>Tipo de investigación:</b> INVESTIGACIÓN APLICADA</p> <p><b>Diseño de investigación:</b> EXPERIMENTAL</p> <p><b>Nivel de investigación:</b> DESCRIPTIVO-EXPLICATIVO</p> <p><b>Método de investigación:</b> HIPOTÉTICO - DEDUCTIVO</p>	<p><b>Población:</b> Población Limeña y Rural mayor de edad.</p> <p><b>Muestreo</b> Método de muestreo Probabilístico</p> <p><b>Lugar de estudio</b> Bellavista, Callao</p> <p><b>Técnicas e instrumentos de recolección de datos</b> La técnica de recolección de datos que se utilizará en la presente investigación será el método de observación</p>

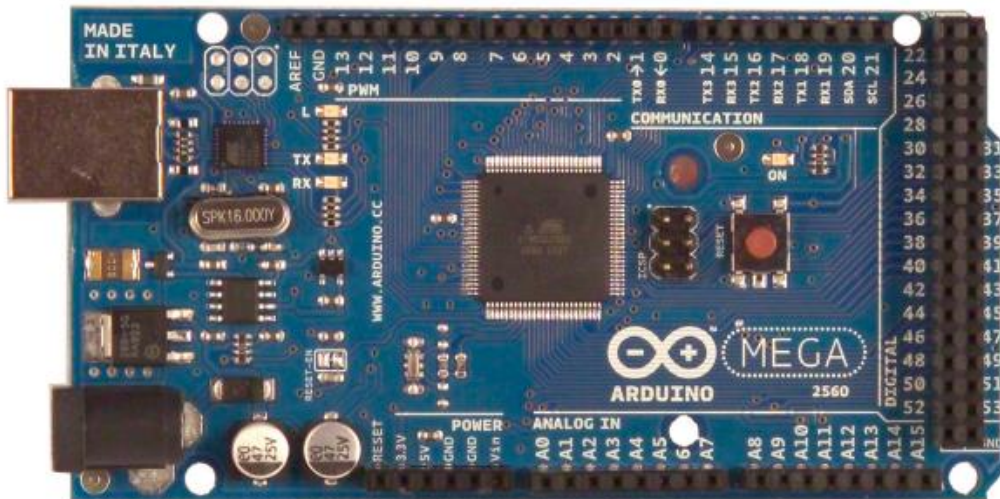
## ANEXO N°2: DATASHEET DE ARDUINO MEGA



### Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

## Arduino Mega 2560 Datasheet



*Handwritten signature*  
NafH-1c

## Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

## Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

  
Massimo Banzi

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

## Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

## Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH



Handwritten signature and initials, possibly 'N.H. 1c'.

value, the LED is on, when the pin is LOW, it's off.

- **I<sup>2</sup>C: 20 (SDA) and 21 (SCL).** Support I<sup>2</sup>C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I<sup>2</sup>C pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and `analogReference()` function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with `analogReference()`.
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

## Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I<sup>2</sup>C (TWI) and SPI communication. The Arduino software includes a `Wire` library to simplify use of the I<sup>2</sup>C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, use the [SPI library](#).

## Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It



Handwritten signature and initials, possibly 'Nuff' and 'c'.

## Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

## USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

## Physical Characteristics and Shield Compatibility



Handwritten signature and initials, likely belonging to the author or a contributor.

# ANEXO N°3: DATASHEET DE LM35

## LM35

### Precision Centigrade Temperature Sensors

#### General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of  $\pm 0.1^\circ\text{C}$  at room temperature and  $\pm 0.5^\circ\text{C}$  over a full  $-55$  to  $+150^\circ\text{C}$  temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only  $60\ \mu\text{A}$  from its supply, it has very low self-heating, less than  $0.1^\circ\text{C}$  in still air. The LM35 is rated to operate over a  $-55^\circ$  to  $+150^\circ\text{C}$  temperature range, while the LM35C is rated for a  $-40^\circ$  to  $+110^\circ\text{C}$  range ( $-10^\circ$  with improved accuracy). The LM35 series is available pack-

aged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

#### Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear  $+10.0\ \text{mV}/^\circ\text{C}$  scale factor
- $0.5^\circ\text{C}$  accuracy guaranteeable (at  $+25^\circ\text{C}$ )
- Rated for full  $-55^\circ$  to  $+150^\circ\text{C}$  range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than  $60\ \mu\text{A}$  current drain
- Low self-heating,  $0.08^\circ\text{C}$  in still air
- Nonlinearity only  $\pm 0.1^\circ\text{C}$  typical
- Low impedance output,  $0.1\ \Omega$  for  $1\ \text{mA}$  load

#### Typical Applications



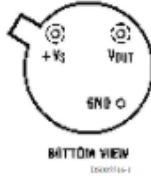
FIGURE 1. Basic Centigrade Temperature Sensor  
( $+2^\circ\text{C}$  to  $+150^\circ\text{C}$ )

*[Handwritten signature]*  
*N. H. - c*



## Connection Diagrams

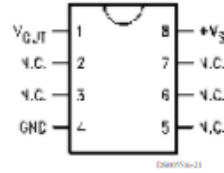
TO-46  
Metal Can Package\*



\*Case is connected to negative pin (GND)

Order Number LM35H, LM35AH, LM35CH, LM35CAH or LM35DH

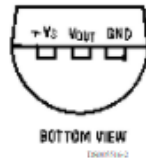
SO-8  
Small Outline Molded Package



N.C. = No Connection

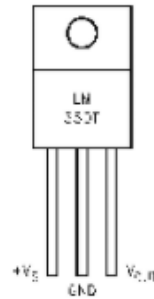
Top View  
Order Number LM35DM

TO-92  
Plastic Package



Order Number LM35CZ,  
LM35CAZ or LM35DZ

TO-220  
Plastic Package\*



### Absolute Maximum Ratings (Note 10)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	+35V to -0.2V
Output Voltage	+6V to -1.0V
Output Current	10 mA
Storage Temp.:	
TO-46 Package,	-60°C to +180°C
TO-92 Package,	-60°C to +150°C
SO-8 Package,	-65°C to +150°C
TO-220 Package,	-65°C to +150°C
Lead Temp.:	
TO-46 Package, (Soldering, 10 seconds)	300°C

TO-92 and TO-220 Package, (Soldering, 10 seconds)	260°C
SO Package (Note 12)	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 11)	2500V
Specified Operating Temperature Range: T (Note 2)	MIN to T <sub>MAX</sub>
LM35, LM35A	-55°C to +150°C
LM35C, LM35CA	-40°C to +110°C
LM35D	0°C to +100°C

*[Handwritten signature]*  
*Nufft/c*

## Electrical Characteristics

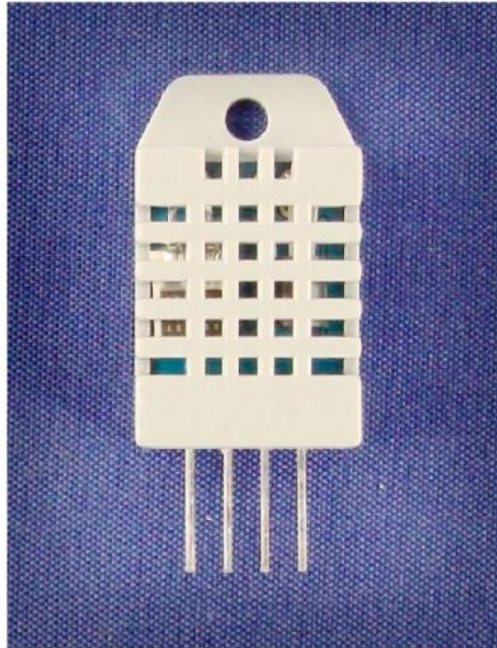
(Notes 1, 6)

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	$T_A = +25^\circ\text{C}$	$\pm 0.2$	$\pm 0.5$		$\pm 0.2$	$\pm 0.5$	$\pm 1.0$	$^\circ\text{C}$
	$T_A = -10^\circ\text{C}$	$\pm 0.3$			$\pm 0.3$			$^\circ\text{C}$
	$T_A = T_{\text{max}}$	$\pm 0.4$	$\pm 1.0$		$\pm 0.4$	$\pm 1.0$		$^\circ\text{C}$
	$T_A = T_{\text{min}}$	$\pm 0.4$	$\pm 1.0$		$\pm 0.4$		$\pm 1.5$	$^\circ\text{C}$
Nonlinearity (Note 8)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	$\pm 0.18$		$\pm 0.35$	$\pm 0.15$		$\pm 0.3$	$^\circ\text{C}$
Sensor Gain (Average Slope)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	+10.0	+9.9, +10.1		+10.0		+9.9, +10.1	mV/ $^\circ\text{C}$
Load Regulation (Note 3) $0 \leq I_L \leq 1\text{mA}$	$T_A = +25^\circ\text{C}$	$\pm 0.4$	$\pm 1.0$		$\pm 0.4$	$\pm 1.0$	$\pm 3.0$	mV/mA
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	$\pm 0.5$		$\pm 3.0$	$\pm 0.5$		$\pm 3.0$	mV/mA
Line Regulation (Note 3)	$T_A = +25^\circ\text{C}$	$\pm 0.01$	$\pm 0.05$		$\pm 0.01$	$\pm 0.05$	$\pm 0.1$	mV/V
	$4\text{V} \leq V_S \leq 30\text{V}$	$\pm 0.02$		$\pm 0.1$	$\pm 0.02$		$\pm 0.1$	mV/V
Quiescent Current (Note 9)	$V_S = +5\text{V}, +25^\circ\text{C}$	56	67		56	67	114	$\mu\text{A}$
	$V_S = +5\text{V}$	106		131	91		114	$\mu\text{A}$
	$V_S = +30\text{V}, +25^\circ\text{C}$	56.2	68		56.2	68		$\mu\text{A}$
	$V_S = +30\text{V}$	106.5		133	91.5		116	$\mu\text{A}$
Change of Quiescent Current (Note 3)	$4\text{V} \leq V_S \leq 30\text{V}, +25^\circ\text{C}$	0.2	1.0		0.2	1.0		$\mu\text{A}$
	$4\text{V} \leq V_S \leq 30\text{V}$	0.5		2.0	0.5		2.0	$\mu\text{A}$
Temperature Coefficient of Quiescent Current		+0.39		+0.5	+0.39		+0.5	$\mu\text{A}/^\circ\text{C}$
Minimum Temperature for Rated Accuracy	In circuit of Figure 1.1, $I_L = 0$	+1.5		+2.0	+1.5		+2.0	$^\circ\text{C}$
Long Term Stability	$T_J = T_{\text{max}}$ , for 1000 hours	$\pm 0.08$			$\pm 0.08$			$^\circ\text{C}$

  
Nuff-1c

## ANEXO N°4: DATASHEET DE DHT22

DHT22 (DHT22 also named as AM2302)



Capacitive-type humidity and temperature module/sensor

### 2. Description:

DHT22 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

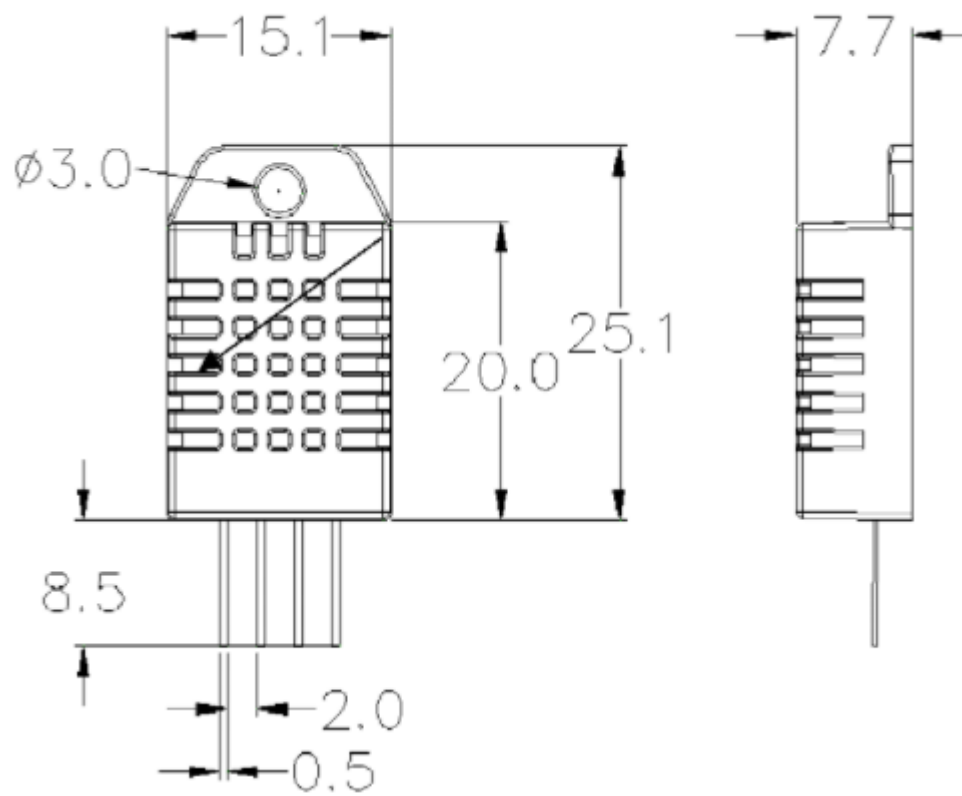
Small size & low consumption & long transmission distance(20m) enable DHT22 to be suited in all kinds of harsh application occasions.

Single-row packaged with four pins, making the connection very convenient.

### 3. Technical Specification:

Model	DHT22	
Power supply	3.3-6V DC	
Output signal	digital signal via single-bus	
Sensing element	Polymer capacitor	
Operating range	humidity 0-100%RH;	temperature -40~80Celsius
Accuracy	humidity +-2%RH(Max +-5%RH);	temperature <+-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH;	temperature 0.1Celsius
Repeatability	humidity +-1%RH;	temperature +-0.2Celsius
Humidity hysteresis	+-0.3%RH	
Long-term Stability	+-0.5%RH/year	
Sensing period	Average: 2s	
Interchangeability	fully interchangeable	
Dimensions	small size 14*18*5.5mm;	big size 22*28*5mm

*Signature*  
Naff-1c



Pin sequence number: 1 2 3 4 (from left to right direction).

Pin	Function
1	VDD----power supply
2	DATA--signal
3	NULL
4	GND

*Signature*  
Nafiz-1c