

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
SEDE VIÑA DEL MAR – JOSÉ MIGUEL CARRERA

DISEÑO DE UNA ESTACIÓN METEOROLÓGICA WIFI CON ESP32 Y THINGSPEAK

Trabajo de Titulación para optar al Título de
Ingeniero en Ejecución en Control e
Instrumentación Industrial

Alumno:

Felipe Ignacio Morales Palacios

Profesor Guía:

Ing. Piero Montenegro Zamora

RESUMEN

KEYWORDS: ESP32, DHT11, BMP280.

Hoy en día existen distintos dispositivos interactivos para el usuario, cuya característica principal es que son programables por lo que permiten generar diversos proyectos, mediante ese contexto se presenta una solución que consiste en un diseño de una estación meteorológica, permitiendo a cualquier estudiante o usuario programar los distintos dispositivos de forma práctica, teniendo un bajo costo y una programación fácil de entender.

El módulo a utilizar es el ESP32 donde una de sus más grandes utilidades es poder conectarse a la red WIFI local, con ello se facilita el poder transmitir los datos de las variables que se desean medir, y con la ayuda de la plataforma Thingspeak, cualquier otro usuario puede leer estos datos.

Los sensores DHT11 y BMP280 son los que permiten sustraer de ellos las variables que son necesarias para la estación, las cuales son, temperatura, presión y humedad, siendo estas las variables más necesarias a la hora de generar un proyecto de este tipo.

ÍNDICE

RESUMEN 2

SIGLA Y SIMBOLOGÍA 9

INTRODUCCIÓN 1

CAPÍTULO 1: ANTECEDENTES GENERALES	2
1 ANTECEDENTES GENERALES	3
1.1 LA UNIVERISDAD	3
1.1.1 Sede José Miguel Carrera	4
1.2 DESCRIPCIÓN DEL PROBLEMA	6
1.2.1 Definición del problema	6
1.2.1 Importancia de resolverlo	6
1.2.2 Involucrados	7
1.3 PROPOSICIÓN DE SOLUCIÓN DEL PROBLEMA	7
1.4 REQUERIMIENTOS	7
1.5 ALTERNATIVAS DE SOLUCIÓN	8
1.5.1 Alternativa N°1: “Hacer nada”	8
1.5.2 Alternativa N°2: Realización de una estación meteorológica	9
1.5.3 Alternativa N°3: Investigación en plataformas virtuales	9
1.6 EVALUACIÓN DE LAS ALTERNATIVAS DE SOLUCIÓN	9
1.6.1 Alternativa seleccionada	10
1.6 OBJETIVO GENERAL	10
1.7 OBJETIVOS ESPECÍFICOS	11
 CAPÍTULO 2: ASPECTOS TÉCNICOS	 12
2 ASPECTOS TÉCNICOS	13
2.1 MICROCONTROLADORES	13

2.1.1	Módulo ESP32	13
2.1.2	Arduino UNO	15
2.1.3	Módulo Wifi ESP-01	16
2.1.4	Raspberry Pi	17
2.1.5	Microcontrolador seleccionado	18
2.2	SENSOR DE TEMPERATURA, HUMEDAD Y PRESIÓN	18
2.2.1	Sensor DHT11	19
2.2.2	Sensor DHT22	20
2.2.3	Sensor BMP280	21
2.2.4	Sensor BMP180	22
2.2.5	Sensores seleccionados	24
2.3	PLATAFORMA WEB	24
2.4	THINGSPEAK	25
2.5	ARDUINO IDE	26
2.6	ESQUEMA DEL CIRCUITO	27
2.7	COMUNICACIÓN I2C	29
2.8	PLATAFORMA THINGSPEAK	29
2.9	PROGRAMACIÓN	32
2.10	DIAGRAMA DE FLUJO	33
CAPÍTULO 3: RESULTADOS OBTENIDOS Y EVALUACIÓN DE COSTOS		34
3.1	RESULTADOS OBTENIDOS Y EVALUACIÓN DE COSTOS	35
3.2	THINGVIEW	36
3.3	MONITOR SERIE	39
3.4	EVALUACIÓN DE COSTOS	39
3.4.1	Componentes del proyecto	40
3.4.2	Módulos WIFI	40
3.4.3	Sensor DHT11	41
3.4.4	Sensor BMP280	41

3.4.5	Cable DUPONT MACHO/MACHO	42
3.4.6	Protoboard	42
3.4.7	Licencias ThingSpeak	43
3.4.8	Costos Hora-Hombre	44
3.4.9	Costos por envíos de materiales	45
3.4.10	Costos de imprevistos	45
3.4.11	Costo de implementación	45
3.4.12	Costo final	46
3.5	RESULTADOS DE APRENDIZAJE	47
	CONCLUSIONES Y RECOMENDACIONES	48
	BIBLIOGRAFÍA	49
	ANEXO A: PROGRAMACIÓN DEL MICROCONTROLADOR ESP32	51
	ANEXO B: PRPGRAMA DE LA ASIGNATIRA	53
	ANEXO C: EXPERIENCIA DE LABORATORIO	54

ÍNDICE DE FIGURAS

Figura 1-1: Logo USM	4
Figura 1-2: Sede José Miguel Carrera	5
Figura 1-3: Ubicación de la Universidad	5
Figura 2-1: Modulo ESP32	14
Figura 2-2: Arduino UNO	15
Figura 2-3: Módulo Wifi ESP-01	16
Figura 2-4: Raspberry Pi	17
Figura 2-5: Sensor DHT11	20
Figura 2-6: Sensor DHT22	20
Figura 2-7: Sensor BMP280	22
Figura 2-8: Sensor BMP180	23
Figura 2-9: Plataforma Ubidots	24
Figura 2-10: Plataforma ThingSpeak	26
Figura 2-11: Interfaz Arduino IDE	27
Figura 2-12: Esquema del circuito	28
Figura 2-13: Nuevo canal	29
Figura 2-14: Nombre y Campos	30
Figura 2-15: Configuración de variables en campos de ThingSpeak	30
Figura 2-16: Guardar canal	31
Figura 2-17: Datos importantes	31
Figura 2-18: Gestor de librerías de Arduino IDE	32
Figura 2-19: Diagrama de flujo	33
Figura 3-1: Grafico de datos	35
Figura 3-2: Agregar canal	36
Figura 3-3: ID del canal	37
Figura 3-4: Datos en aplicación del canal	37

Figura 3-5: Gráficos en la aplicación	38
Figura 3-6: Monitor Serie	39

ÍNDICE DE TABLAS

Tabla 1-1: Requerimientos del proyecto	8
Tabla 1-2: Calificación de alternativas	9
Tabla 1-3. Exhibición de los resultados obtenidos en la comparación de alternativas	10
Tabla 2-1: Tabla comparativa	18
Tabla 2-2: Conexión Sensores	28
Tabla 3-1: Componentes del proyecto	40
Tabla 3-2: Cotización de Microcontroladores	41
Tabla 3-3: Cotización sensor DHT11	41
Tabla 3-4: Cotización Sensor BMP280	42
Tabla 3-5: Cotización cables DUPONT	42
Tabla 3-6: Cotización Protoboard	43
Tabla 3-7: Cotización ThingSpeak	43
Tabla 3-8: Costos por Hora-Hombre	44
Tabla 3-9: Carta Gantt	44
Tabla 3-10: Costos de imprevistos	45
Tabla 3-11: Costos de implementación	46
Tabla 3-12: Costos totales	46

SIGLA Y SIMBOLOGÍA

A SIGLA

BMP:	Sensor Barométrico
DHT:	sensor de temperatura y humedad
IOT:	internet de las cosas
I2C:	puerto y protocolo de comunicación serial
IDE:	entorno de desarrollo interactivo
SPI:	interfaz periférica de serie

B. SIMBOLOGÍA

°C:	Grados Celsius
Cm:	Centímetros
hPa:	Hecto pascales
Hz:	Hertz
KB:	Kilo Byte
Kg:	Kilogramos
Pa:	Pascal
RH:	Humedad relativa
V:	Voltaje

INTRODUCCIÓN

En la actualidad, el internet de las cosas (IoT, por sus siglas en ingles), hace referencia a la conexión de los objetos a internet, por ejemplo, al conectar sensores con el fin de monitorear actividades, otro ejemplo sería, para controlar el estado de las lámparas, entre otras actividades.

Hoy en día el monitoreo de variables a distancias es más común, habiendo empresas que se dedican solo a este rubro, por lo que es necesario que los alumnos tengan nociones de la programación y monitoreo a distancia. Mediante un circuito sencillo se busca que los alumnos aprendan el funcionamiento de uno de los diferentes módulos que se pueden conectar a la red WIFI.

A medida que la automatización industrial avanza con la tecnología, los ingenieros trabajan para conectar los sistemas a internet de las cosas (IoT), lo que ha hecho que las empresas industriales más tradicionales y sus maquinarias queden obsoletas con el paso del tiempo. Sin embargo, la conectividad inalámbrica a IoT mediante Wi-Fi o Bluetooth se ha hecho relativamente simple utilizando los diversos microcontroladores disponibles en el mercado.

Al comprender los conocimientos básicos en la programación en lenguaje C o C++, comenzar a usar el ESP32 no es una tarea difícil siendo este un dispositivo sencillo de utilizar, teniendo en cuenta que hay muchos detalles y bastantes decisiones que deben tomar para poder trabajar con el ESP32 por primera vez.

CAPÍTULO 1: ANTECEDENTES GENERALES

1 ANTECEDENTES GENERALES

En este capítulo se dará a conocer el marco teórico del trabajo de título, siendo este el diseño de una estación meteorológica, en donde se verán sus datos a través de la función WIFI del microcontrolador a evaluar, también a través de una página web se pueden obtener los datos de la estación meteorológica mediante la conexión WIFI.

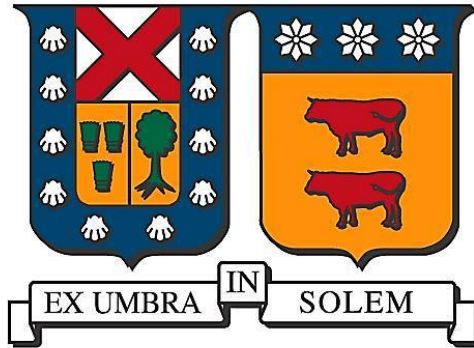
1.1 LA UNIVERSIDAD

Los albores de la Universidad Federico Santa María se remontan al sueño del altruista don Federico Santa María Carrera quien bajo su testamento manifestó el deseo de contribuir al progreso cultural de su nación, buscaba facilitar que los compatriotas desposeídos pero sobresalientes sin mayor requisito que su mérito y alto rendimiento normativo puedan optar al ingreso de la vida académica.

Nacido el 15 de agosto de 1845 en la calle el Cabo hoy Esmeralda, sector de la elite porteña. Perteneciente a una de las familias más destacadas en la emancipación de la nación chilena, sexto hijo entre nueve hermanos, sus padres Juan Antonio Santa María hombre de negocios y su madre la distinguida Magdalena Carrera Aguirre. Nieto paterno de Juan Antonio Carrera Salinas, quien fue primo hermano de José Miguel Carrera Verdugo (prócer de la patria), de quien heredara ideales que plasmara claramente en su testamento final de vida.

La Universidad Técnica Federico Santa María (Casa Central) ubicada en Cerro Placeres de la región de Valparaíso, se fundó el 20 de diciembre de 1931, al año siguiente, siguiendo el sueño de don Federico, se abren las puertas de la escuela de artes y oficio y el colegio de ingenieros José Miguel Carrera, con esto se da inicio a las actividades académicas basadas en la técnica y los oficio, que con el transcurso del tiempo se volverían un pilar esencial para el desarrollo tecnológico e industrial del país.

Actualmente la institución cuenta con un universo de 18.514 alumnos (encuesta del 2012), distribuidos en 3 campus (Casa Central Valparaíso, Vitacura Santiago, San Joaquín), y 2 sedes (José miguel Carrera Viña del Mar y Rey Balduino Bélgica Concepción).



Fuente: www.usm.cl

Figura 1-1: Logo USM

1.1.1 Sede José Miguel Carrera

En 1971 se construye y se funda La sede José Miguel Carrera (figura 1-2) se encuentra ubicada en la avenida Federico Santa María N°6090, en el límite de las comunas de Viña del Mar y Quilpué en la región de Valparaíso (figura 1-3).

La sede cuenta con carreras Técnicas Universitarias e Ingenierías, de las cuales, la mayor cantidad de ingenierías se imparten en modo vespertino.



Fuente: www.usm.cl

Figura 1-2: Sede José Miguel Carrera



Fuente: Google Maps

Figura 1-3: Ubicación de la Universidad

1.2 DESCRIPCIÓN DEL PROBLEMA

A mediados de octubre del año 2019 y todo el año 2020, las clases en la Universidad se vieron afectadas debido al estallido social y la pandemia del coronavirus, siendo aproximadamente un año y medio sin clases presenciales, en donde las clases prácticas no se realizaron de una manera óptima.

1.2.1 Definición del problema

La Universidad toma la opción de impartir las clases a través de plataformas virtuales, donde el programa utilizado fue Zoom, la modalidad virtual contempla tanto clases teóricas como laboratorios, pero en cuanto a los laboratorios o clases prácticas, esto es casi imposible debido a los materiales a utilizar, ya sean maquinas, computadores, entre otros dispositivos, los cuales los alumnos no tienen a su disposición, siendo estos materiales de un costo muy elevado, por consecuencia ningún alumno puede obtener el aprendizaje estimado por la asignatura de Microcontroladores de la carrera de Técnico Universitario en Electrónica.

1.2.1 Importancia de resolverlo

Es imposible que los alumnos en el año 2020 y parte del año 2021 puedan tener las clases prácticas o talleres en la Universidad, esto se debe a todo lo que conlleva la pandemia que se está viviendo, por lo que genera que los alumnos pierdan el conocimiento técnico de sus respectivas áreas, con el fin de que se obtenga un aprendizaje práctico, a través de un circuito con un microcontrolador, se puede obtener las aptitudes suficientes para poder cumplir con las exigencias de la asignatura en cuestión, ver anexo B.

1.2.2 Involucrados

Como se menciona anteriormente, los más afectados son los alumnos por no desarrollar todas las competencias declaradas por la carrera en cuestión y los resultados de aprendizaje de la asignatura, también los profesores quienes se ven enfrentados a la realización de laboratorios simulados online.

1.3 PROPOSICIÓN DE SOLUCIÓN DEL PROBLEMA

Gracias a este proyecto, se intenta perfeccionar las habilidades técnicas, en cuanto a la implementación de un circuito de una estación meteorológica que ayude a mejorar la programación y entender la función WIFI de un microcontrolador, este circuito está al alcance de cualquier alumno que desee aplicar los conocimientos básicos de programación y de bajo costo.

1.4 REQUERIMIENTOS

En esta sección se analizará los requerimientos funcionales necesarios para el desarrollo de este proyecto. Los requerimientos se describen en la tabla 1-1. Para el desarrollo de esto se necesitará implementos que ayuden a establecer una conexión estable entre los sensores, el microcontrolador e internet con el fin de poder ver los datos en cualquier parte sin estar presente en un lugar específico y de muy bajo costo para la operación de la estación meteorológica.

Las variables que se desean monitorear son: temperatura, humedad, altitud y presión, las cuales se especificaran más adelante.

Tabla 1-1: Requerimientos del proyecto

N°	Requerimiento	Descripción	Criterio de aceptación
1	Implementación de una conexión de internet WIFI en el hogar.	La incorporación de una red de WIFI se facilita la comunicación entre el usuario y la estación.	La conexión debe ser estable para que no existan datos incorrectos.
2	Conocimientos de programación en lenguaje C o C++.	Es necesario saber el lenguaje C o C++ para poder programar los dispositivos a utilizar.	Es necesario tener un aprendizaje básico en programación

Fuente: Elaboración propia basada en las necesidades del proyecto.

1.5 ALTERNATIVAS DE SOLUCIÓN

Para poder lograr un aprendizaje técnico de monitoreo de datos a través de internet por WIFI, se tienen que buscar alternativas que cumplan el requerimiento principal, el cual consiste en que el alumno logre obtener la habilidad práctica adecuada en cuanto a la programación.

1.5.1 Alternativa N°1: "Hacer nada"

La primera alternativa es no hacer nada ante el problema, esperar a que termine la pandemia para poder tener las clases presenciales en la Universidad, sin embargo, los alumnos no logran obtener todos los resultados de aprendizaje de la asignatura.

1.5.2 Alternativa N°2: Realización de una estación meteorológica

La segunda alternativa, es la implementación de un circuito con un dispositivo en donde los conocimientos básicos de programación y manejo de las funciones del microcontrolador, ayuda a cualquier alumno a poder desarrollar los conocimientos obtenidos en la Universidad, siendo éste de muy bajo costo. A través de una evaluación económica se insita a que la Universidad logre costear los gastos de materiales y envíos, para que los alumnos no pierdan el conocimiento técnico.

1.5.3 Alternativa N°3: Investigación en plataformas virtuales

La última opción es realizar una cotización e investigación de diversos programas virtuales existentes en el mercado, con el fin de que el alumno pueda interactuar con las plataformas virtuales solo con el computador.

1.6 EVALUACIÓN DE LAS ALTERNATIVAS DE SOLUCIÓN

Las formulaciones propuestas previstas para este proyecto son descritas en la tabla 1-3, basada en una escala de competencias, tabla 1-2.

Tabla 1-2: Calificación de alternativas.

Muy deficiente	Deficiente	Aceptable	Bueno	Óptimo
1	2	3	4	5

Fuente: Elaboración propia basada en escala de competencias

Tabla 1-3. Exhibición de los resultados obtenidos en la comparación de alternativas.

Comparación	Alternativa 1	Alternativa 2	Alternativa 3
Innovación	1	3	1
Beneficio	1	4	4
Factibilidad	1	5	2
Puntuación total	3	12	7

Fuente: Elaboración propia, basada en análisis de soluciones

1.6.1 Alternativa seleccionada

La alternativa seleccionada es la 2, debido a la contingencia que se vive en estos momentos es la mejor opción, porque ayuda tanto en los conocimientos prácticos como teóricos a la hora del aprendizaje, siendo la estación meteorológica la forma de que los alumnos desarrollen mejor sus habilidades en cuanto a la programación y emplear de manera óptima las funciones del microcontrolador.

1.6 OBJETIVO GENERAL

El objetivo general del trabajo es la implementación de una estación meteorológica con un microcontrolador a evaluar, con el fin de cumplir con los resultados de aprendizaje en la asignatura Microcontroladores de la carrera Técnico Universitario en Electrónica.

1.7 OBJETIVOS ESPECÍFICOS

- Investigar microcontroladores disponibles en el mercado y al alcance de cualquier persona.
- Determinar los componentes necesarios e idóneos para poder lograr el prototipo de estación meteorológica en casa y con un bajo costo.
- Determinar plataforma web.
- Realizar evaluación económica del proyecto.
- Diseñar una experiencia de laboratorio (anexo C).

CAPÍTULO 2: ASPECTOS TÉCNICOS

2 ASPECTOS TÉCNICOS

En este capítulo se dará a conocer el desarrollo técnico e implementación de la solución. Para poder diseñar la estación meteorológica se trabajará con 2 sensores, los cuales permiten medir los datos que entregan estos sensores y serán enviados por WIFI a la plataforma web.

2.1 MICROCONTROLADORES

En el mercado existen diversos microcontroladores con infinitudes de usos, para poder crear la estación meteorológica se evaluarán 2 dispositivos, en el cual se escogerá uno para poder realizar el proyecto.

2.1.1 Módulo ESP32

Desarrollado y creado por Espressif Systems, ESP32 es un microcontrolador de bajo consumo y bajo costo en un chip SoC (System On Chip) con Wi-Fi y modo dual con Bluetooth.

En el fondo, hay un microprocesador Tensilica Xtensa LX6 de doble núcleo o de un solo núcleo con una frecuencia de reloj de hasta 240MHz.

ESP32 (figura 2-1) está altamente integrado con switch de antena, balun para RF, amplificador de potencia, amplificador de recepción con bajo nivel de ruido, filtros y módulos de administración de energía, totalmente integrados dentro del mismo chip.

Diseñado para dispositivos móviles; tanto en las aplicaciones de electrónica, y las de IoT (Internet de las cosas), ESP32 logra un consumo de energía ultra bajo a través de funciones de ahorro de energía, Incluye la sintonización de reloj con una resolución fina, modos de potencia múltiple y escalado de potencia dinámica.



Fuente: <https://es.wikipedia.org/wiki/ESP32>

Figura 2-1: Modulo ESP32

Las características principales de este módulo son:

- Procesador principal: Tensilica Xtensa LX6 de 32 bits.
- Wi-Fi: 802.11 b / g / n / e / i (802.11n @ 2.4 GHz hasta 150 Mbit / s).
- Bluetooth: v4.2 BR / EDR y Bluetooth Low Energy (BLE).
- Frecuencia de Clock: Programable, hasta 240MHz.
- Rendimiento: hasta 600DMIPS.
- OM: 448KB, para arranque y funciones básicas.
- SRAM: 520KiB, para datos e instrucciones.

2.1.2 Arduino UNO

El Arduino Uno es un microcontrolador de código abierto basado en el microchip ATmega328P y desarrollado por Arduino, la placa está equipada con conjuntos de pines de E/S digitales y analógicas que pueden conectarse a diversos dispositivos y crear una gran cantidad de circuitos, puede ser alimentado por un cable USB o por una batería externa, también la placa uno es la primera de una serie de placas Arduino integradas con la conexión USB, figura 2-2.

Las características principales son:

- Voltaje de entrada: 7 a 20 [V]
- Pines de E/S digitales: 14
- Corriente DC por Pin de E/S: 20 mA
- Corriente CC para Pin de 3.3V: 50 mA
- Memoria Flash: 32 KB



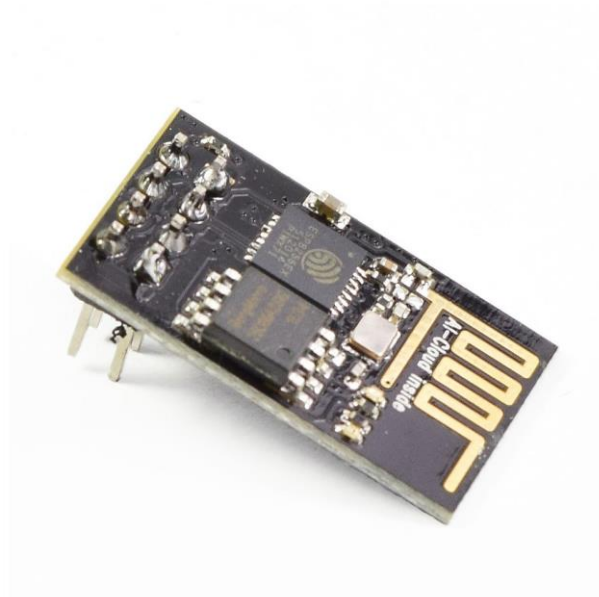
Fuente: https://es.wikipedia.org/wiki/Arduino_Uno

Figura 2-2: Arduino UNO

2.1.3 Módulo Wifi ESP-01

El módulo ESP-01 Wifi es un transceptor que permite proporcionar conectividad Wifi a los distintos microcontroladores en el mercado, en específico al Arduino UNO para que éste realice la conexión Wifi de una forma simple y económica (figura 2-3).

Integrada por un potente procesador con arquitectura de 32 bits y conectividad Wifi, permitiendo trabajar como host de aplicaciones o descargar los datos de otros procesadores, al trabajar dentro del entorno Arduino es posible utilizar un lenguaje conocido y hacer uso del IDE sin mayores problemas.



Fuente: <https://www.iberobotics.com/producto/esp-01-modulo-wifi-esp8266/>

Figura 2-3: Módulo Wifi ESP-01

2.1.4 Raspberry Pi

La Raspberry Pi es un microcontrolador de bajo costo desarrollado en el Reino Unido por la Raspberry Pi Foundation, el modelo original busca la promoción de la enseñanza de informática en las escuelas, sin embargo, termina siendo utilizado en distintas áreas como en la robótica, su sistema operativo oficial es Raspberry Pi OS, aunque este dispositivo permite usar otros sistemas operativos, incluido una versión de Windows 10. En gran parte sus versiones, incluye un procesador Broadcom, memoria RAM, GPU, puertos USB, HDMI, Ethernet, pines GPIO y un conector para cámara, figura 2-4.



Fuente: https://es.wikipedia.org/wiki/Raspberry_Pi

Figura 2-4: Raspberry Pi

2.1.5 Microcontrolador seleccionado

El módulo ESP32 fue el seleccionado debido a que este se puede conectar a la red WIFI sin tener que acoplar diversos periféricos para poder realizar dicha conexión, además de no tener un precio excesivamente alto comparado con la Raspberry 4, recalcar que este microcontrolador fue obtenido gracias a la Universidad.

Tabla 2-1: Tabla comparativa

	ESP32	Arduino uno	Raspberry pi 4
WIFI	si	no	si
Bluetooth	si	no	si
Arquitectura	32 bits	8 bits	64 bits
RAM	512 KB	2 KB	512 MB
Nucleos	2	1	4
Pines E/S	36	14	40

Fuente: Elaboración propia, basada en microcontroladores

2.2 SENSOR DE TEMPERATURA, HUMEDAD Y PRESIÓN

Para poder medir las variables de temperatura, humedad y presión es necesario tener 2 sensores, uno que proporcione los datos de temperatura y humedad, además de otro sensor que permita medir la presión para la realizar la estación.

2.2.1 Sensor DHT11

El DHT11 (figura 2-5) es un sensor digital de temperatura y humedad relativa de bajo costo y fácil uso, Integra un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos (no posee salida analógica). Utilizado en aplicaciones académicas relacionadas al control automático de temperatura, aire acondicionado, monitoreo ambiental en agricultura y más.

Es posible utilizar el sensor DHT11 con las plataformas Arduino/Raspberry Pi/Nodemcu, es muy sencillo tanto a nivel de software como hardware. A nivel de software se dispone de librerías para Arduino. En cuanto al hardware, solo es necesario conectar el pin VCC de alimentación a 3-5V, el pin GND a Tierra (0V) y el pin de datos a un pin digital en el ESP32.

El sensor sólo se puede obtener nuevos datos cada 2 segundos. El protocolo de comunicación entre el sensor y el microcontrolador emplea un único hilo o cable.

Las características del sensor son las siguientes:

- Voltaje de Operación: 3V - 5V DC
- Rango de medición de temperatura: 0 a 50 °C
- Precisión de medición de temperatura: ± 2.0 °C
- Resolución Temperatura: 0.1°C
- Rango de medición de humedad: 20% a 90% RH.
- Precisión de medición de humedad: 5% RH.
- Resolución Humedad: 1% RH
- Tiempo de sensado: 1 segundo.
- Interface digital: Single-bus (bidireccional)



Fuente: <https://programarfacil.com>

Figura 2-5: sensor DHT11

2.2.2 Sensor DHT22

Este sensor internamente es muy parecido a las descripciones del sensor DHT11, figura 2-6, con la diferencia que sus rangos de medidas son más amplias y precisas, teniendo éste un costo mayor al anterior.



Fuente: <https://naylampmechatronics.com>

Figura 2-6: sensor DHT22

Sus características son:

- Voltaje de Operación: 3V - 6V DC
- Rango de medición de temperatura: -40°C a 80°C
- Precisión de medición de temperatura: $<\pm 0.5^{\circ}\text{C}$
- Resolución Temperatura: 0.1°C
- Rango de medición de humedad: De 0 a 100% RH
- Precisión de medición de humedad: 2% RH
- Resolución Humedad: 0.1%RH
- Tiempo de sensado: 2s

2.2.3 Sensor BMP280

El sensor de presión barométrica BMP280 (figura 2-7) permite medir la altura respecto al nivel del mar, su funcionamiento está basado en la relación entre presión del aire y la altitud, posee alta precisión y un bajo consumo de energía. El BMP280 ofrece un rango de medición de 300 a 1100 hPa (Hecto Pascal). Basado en tecnología BOSCH piezo-resistiva con gran robustez, alta precisión y linealidad, así como con estabilidad a largo plazo. Se ha diseñado para ser conectado directamente a un microcontrolador a través de comunicación I2C o SPI.

Las características principales del sensor son:

- Voltaje de Operación: 1.8V - 3.3V DC
- Interfaz de comunicación: I2C o SPI
- Rango de Presión: 300 a 1100 hPa (0.3-1.1bar)
- Resolución: 0.16 Pa
- Precisión absoluta: 1 hPa
- Medición de temperatura incluida

- Resolución de temperatura: 0.01°C
- Precisión Temperatura: 1°C
- Frecuencia de Muestreo: 157 Hz (máx.)
- Rango de altura medible: 0-9100 metros
- Ultra-bajo consumo de energía

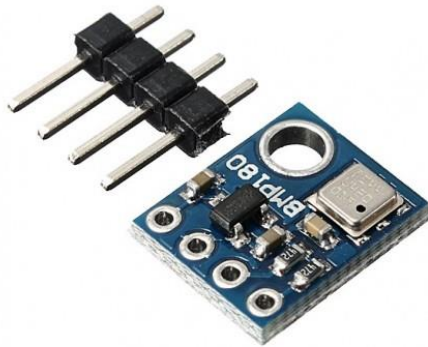


Fuente: <http://www.esp32learning.com>

Figura 2-7: Sensor BMP280

2.2.4 Sensor BMP180

Este sensor contiene gran parentesco con el sensor BMP280, siendo el bmp180 la versión anterior, por consecuencia su respectivo funcionamiento es el mismo, pero sus características de medición diferentes, figura 2-8.



Fuente: <https://naylampmechatronics.com>

Figura 2-8: Sensor BMP180

Sus características son:

- Voltaje de Operación: 3.3V - 5V DC
- Interfaz de comunicación: I2C (3.3V)
- Rango de Presión: 300 a 1100 hPa (0.3-1.1bar)
- Resolución: 1 Pa
- Precisión absoluta: 1 hPa
- Medición de temperatura incluida
- Resolución de temperatura: 0.1°C
- Precisión Temperatura: 1°C
- Frecuencia de Muestreo: 120 Hz (máx.)
- Rango de altura medible: 0-9100 metros
- Ultra-bajo consumo de energía

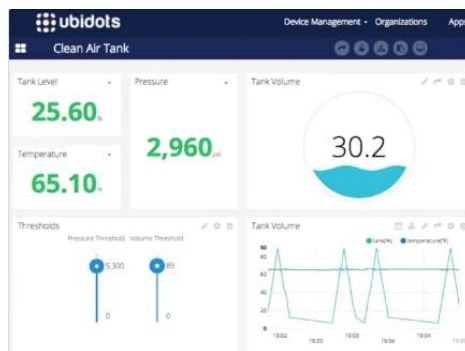
2.2.5 Sensores seleccionados

Los sensores DHT11 y BMP280 son los que están disponibles en el mercado y su costo no es excesivamente alto, su elección fue gracias a la disponibilidad del mercado en tiempos de pandemia, además de que sus diferencias son mínimas a la hora de implementar la estación meteorológica.

Con el microcontrolador y los sensores seleccionados, es posible crear la estación meteorológica con el fin de medir las variables mencionadas anteriormente.

2.3 PLATAFORMA WEB

Existen diversas plataformas web que permiten al módulo ESP32, a través de la conexión WIFI, recibir los datos enviados por microcontrolador y con ello crear gráficos con dichos datos, además de monitorear constantemente el módulo, un ejemplo seria la plataforma Ubidots, ver figura 2-9.



Fuente: www.ubidots.com

Figura 2-9: Plataforma Ubidots

Sin embargo, la licencia gratuita de la plataforma es de 1 mes, siendo muy poco tiempo para realizar experiencias o proyectos. Una mejor plataforma para realizar la comunicación es ThingSpeak, teniendo esta última una licencia gratuita de 1 año.

2.4 THINGSPEAK

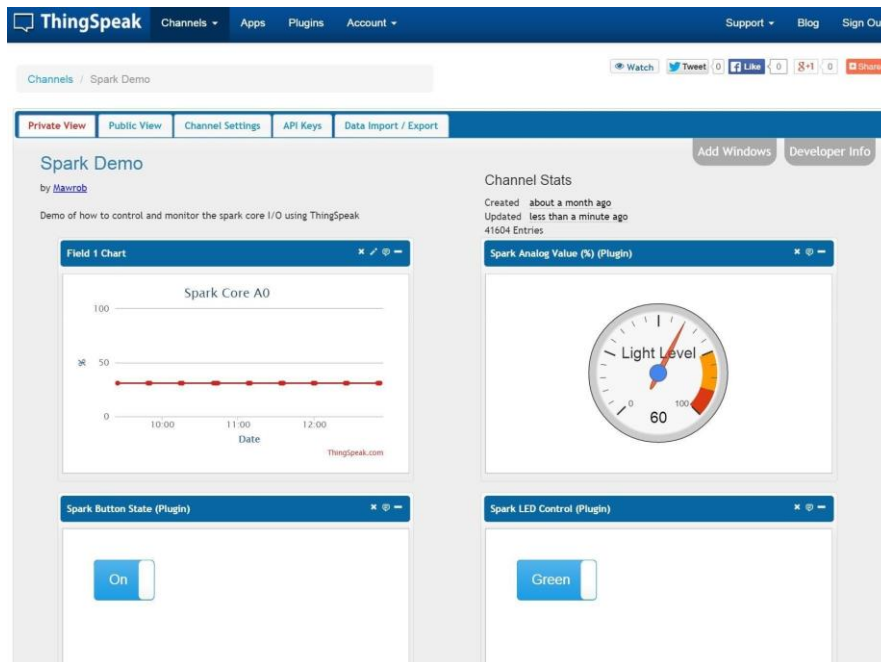
ThingSpeak es una plataforma de Internet of Things (IoT) que permite recoger y almacenar datos de sensores en la nube y desarrollar aplicaciones IoT. ThingSpeak también ofrece aplicaciones que permiten analizar y visualizar los datos en MATLAB y actuar sobre ellos.

Los datos de los sensores pueden ser enviados desde Arduino, Raspberry, entre otros dispositivos.

ThingSpeak fue lanzado originalmente por ioBridge en 2010 como un servicio en apoyo de aplicaciones de IoT, tiene soporte integrado del software de computación numérica MATLAB de MathWorks, permitiendo a los usuarios de ThingSpeak analizar y visualizar datos cargados usando Matlab sin requerir la compra de una licencia de Matlab de Mathworks.

También tiene una relación cercana con Mathworks, toda la documentación de ThingSpeak está incorporada en el sitio de documentación de Matlab e incluso habilitando cuentas de usuario registradas de Mathworks como credenciales de inicio de sesión válidas en el sitio web de ThingSpeak.

ThingSpeak maneja la adquisición de los parámetros a medir mediante canales, consta de 8 campos de datos, campos de ubicación y de estado (figura 2-10). Para los fines que uno estime conveniente, en estos canales es posible recolectar la información necesaria de cada sensor y almacenar dichos datos en esta plataforma, posteriormente, esto se podrá representar visualmente para poder desplegar los datos en pocos instantes, ya que cada 15 segundos se recopila cada parámetro medido.



Fuente: www.aprendiendoarduino.wordpress.com

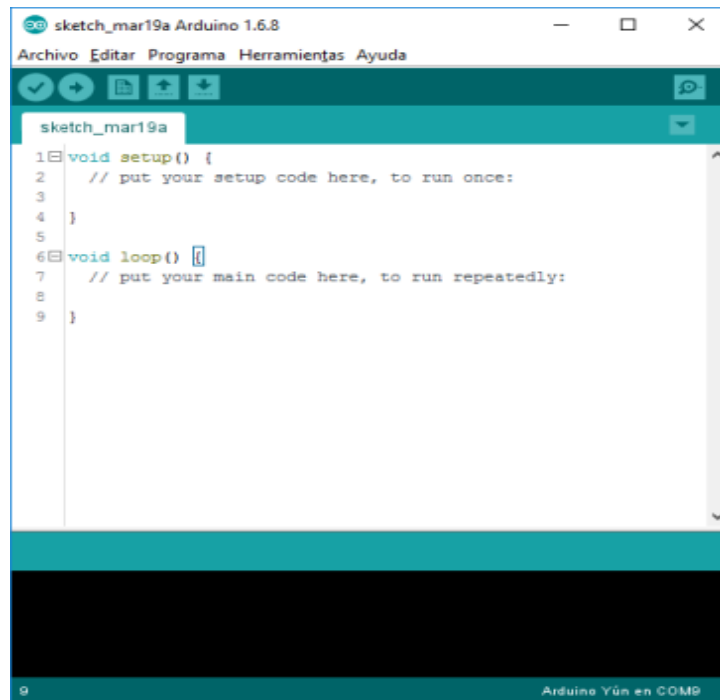
Figura 2-10: Plataforma ThingSpeak

2.5 ARDUINO IDE

El entorno de desarrollo integrado o en inglés Integrated Development Environment (IDE) de Arduino, es una aplicación multiplataforma (para Windows, macOS, Linux) que está escrita en el lenguaje de programación Java (figura 2-11). Se utiliza para escribir y cargar programas en placas compatibles con Arduino, pero también, con la ayuda de núcleos de terceros, se puede usar con placas de desarrollo de otros proveedores.

El IDE de Arduino admite los lenguajes C y C++ utilizando reglas especiales de estructuración de códigos, también suministra una vasta biblioteca de software para poder realizar el proyecto que se desee, éste proporciona muchos procedimientos comunes para los distintos periféricos de entrada y salida.

El Arduino IDE será el programa utilizado para programar el módulo ESP32, el cual es útil y sencillo de entender.

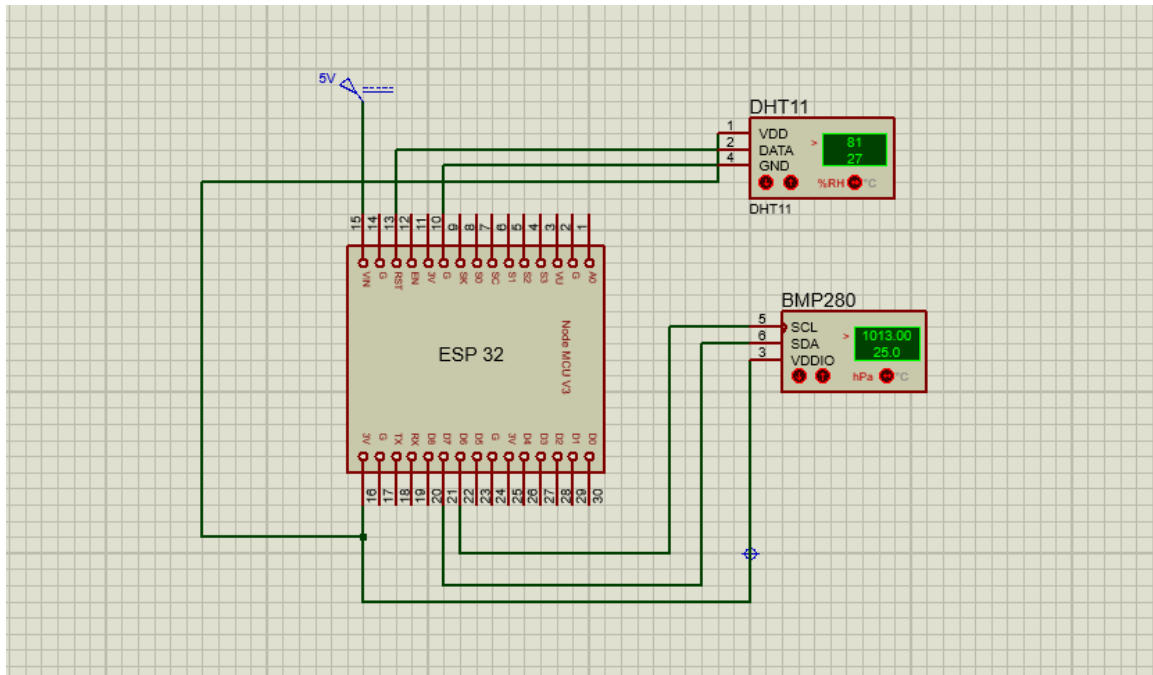


Fuente: www.aprendiendoarduino.wordpress.com

Figura 2-11: Interfaz Arduino IDE

2.6 ESQUEMA DEL CIRCUITO

Como se muestra en la figura 2-12, los pines a utilizar son: el pin de 3.3V para energizar los sensores, el pin 13 donde se conecta el sensor DTH11 debido a que por ese pin el sensor entregara la información, el pin 21 y el 22 van conectados a los pines SDA y SCL respectivamente, del sensor BMP280, con esa configuración se logra la comunicación I2C, el microcontrolador puede ser conectado desde el computador, para alimentar eléctricamente el módulo ESP32, ver tabla 2-2.



Fuente: Elaboración propia, basada en esquema del circuito

Figura 2-12: Esquema del circuito

Tabla 2-2: Conexión Sensores

Pines Microcontrolador	Sensor BMP280	Sensor DHT11
13	-	v
21	SDA	-
22	SCI	-
3.3V	IN	s
GND	-	g

Fuente: Elaboración propia, basada en esquema del circuito

2.7 COMUNICACIÓN I2C

El estándar I2C (Inter-Integrated Circuit) fue desarrollado por Philips en 1982 para la comunicación interna de dispositivos electrónicos.

El bus I2C solo necesita dos cables para funcionar, uno para la señal de reloj y otro para la señal donde se envían los datos.

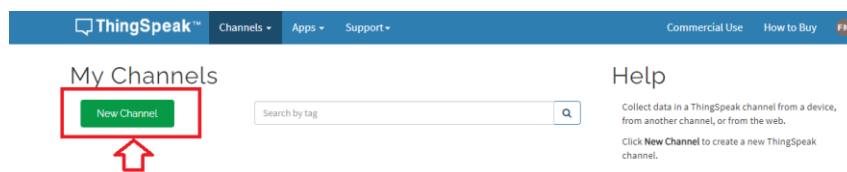
En el bus de cada dispositivo dispone de una dirección, que se ejecuta para acceder a los dispositivos de forma individual, esta dirección puede ser fijada por hardware o totalmente por software, cada dispositivo conectado al bus debe tener una dirección única.

Si se tienen diversos dispositivos similares se debe cambiar la dirección o implementar un bus secundario, también mencionar que tiene una arquitectura de tipo maestro-esclavo, el dispositivo maestro inicia la comunicación con los esclavos y puede mandar o recibir datos de los esclavos.

2.8 PLATAFORMA THINGSPEAK

Para poder crear el canal gratuito de ThingsSpeak primero hay que crear una cuenta, pero al trabajar con MathWorks, la cuenta se puede crear con el correo institucional.

Una vez conectado a la plataforma, se tiene que crear el canal (figura 2-13) en donde se ingresan todas las configuraciones.



Fuente: <https://thingspeak.com/channels>

Figura 2-13: Nuevo canal

Al crear un canal hay que asignar el nombre del canal y los campos (Fields) que se requieran, en ThingSpeak solo se pueden crear 8 con la versión gratuita, ver figura 2-14, siendo los datos de temperatura, humedad, presión y altitud entregados por los sensores y transmitidos por ESP32 a través de wifi a la plataforma.

Fuente <https://thingspeak.com/channels/new>

Figura 2-14: Nombre y Campos

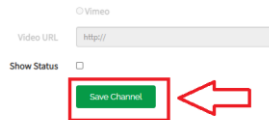
No olvidar, dependiendo del campo que se requiera utilizar, los datos hay que configurarlos en el programa del Arduino IDE, como se muestra en el anexo A, figura 2-15.

```
ThingSpeak.setField(3,temp);
ThingSpeak.setField(4,presion);
ThingSpeak.setField(5,altitud);
```

Fuente: Creación propia basado en programa del microcontrolador

Figura 2-15: Configuración de variables en campos de ThingSpeak

Una vez terminadas las configuraciones, solo se debe poner Save Channel como lo muestra la figura 2-16.

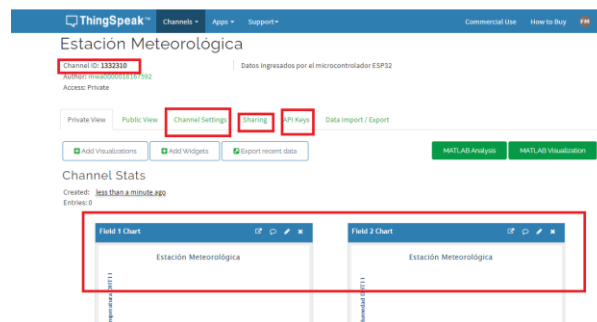


Fuente 1 <https://thingspeak.com/channels/new>

Figura 2-16: Guardar canal

Al guardar el canal, aparecerán los campos en donde se grafican los datos, también son importante los cuadros marcados en la figura 2-17, en donde Channel ID y en la pestaña APIKeys dan la clave y el ID del canal que se debe ingresar en el programa del Arduino IDE para que el módulo ESP32 envíe los datos al canal.

En Channel Settings se puede cambiar las configuraciones realizadas anteriormente cuando se creó el canal, en Sharing se puede decidir si el canal es privado o si se quiere compartir al público y los Field Chart son los gráficos que se lograrán apreciar cuando los datos se ingresen, en la versión gratuita de ThingSpeak, la captura de los datos se actualiza cada 14 segundos.



Fuente: https://thingspeak.com/channels/1332310/private_show

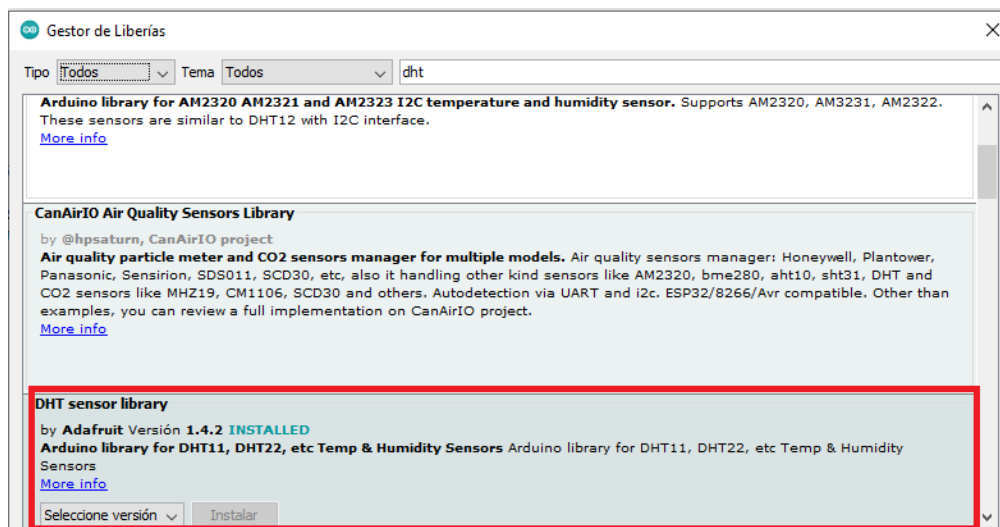
Figura 2-17: Datos importantes

2.9 PROGRAMACIÓN

En el anexo A, se puede ver toda la programación del microcontrolador, cabe señalar que antes de empezar a programar en el Arduino IDE hay que descargar las siguientes librerías a utilizar:

- DHT sensor library by adafuit: librería del sensor DHT11
- Adafruit BMP280: librería del sensro BMP280
- ThingSpeak by MathWorks: librería para ThingSpeak

Todas estas librerías se pueden descargar dentro de la misma plataforma de Arduino, en gestor de librerías, haciendo que la programación sea más reducida. Figura 2-18.

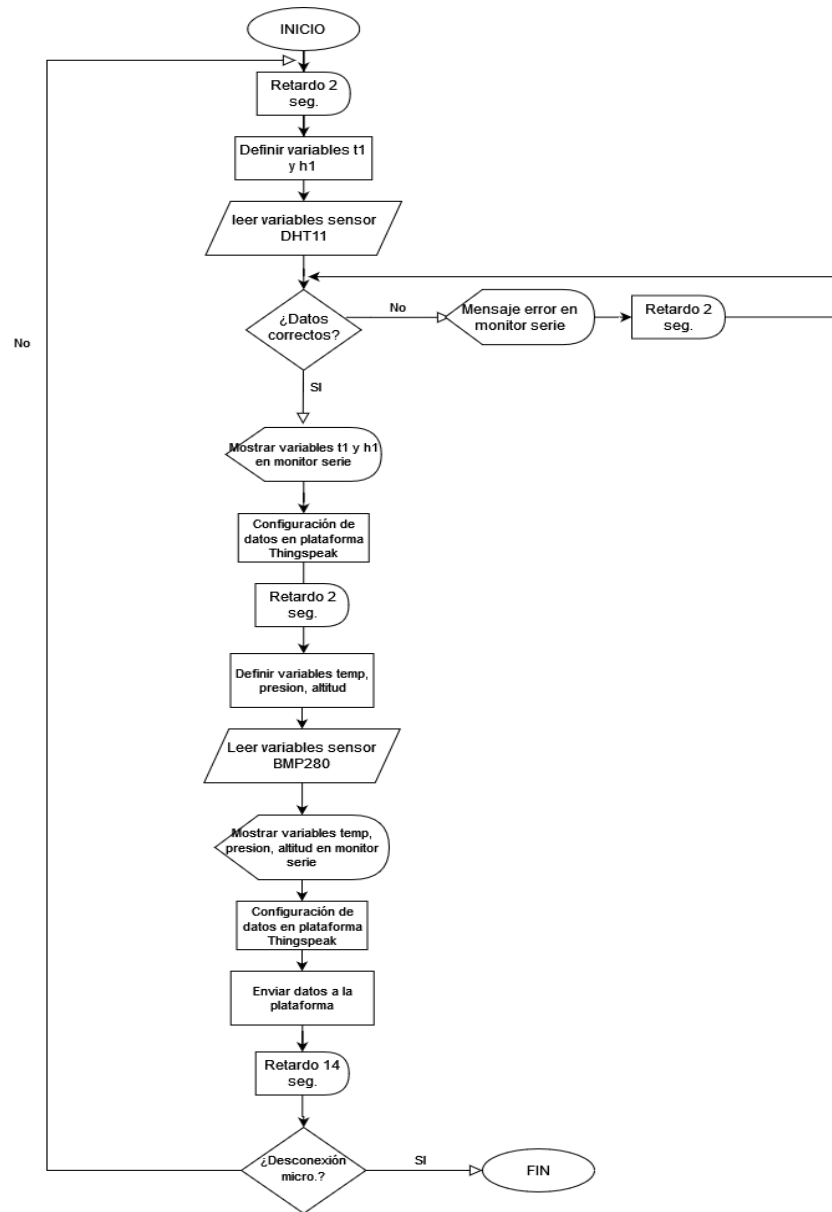


Fuente: creación propia basado en librerías de sensores

Figura 2-18: Gestor de librerías de Arduino IDE

2.10 DIAGRAMA DE FLUJO

El diagrama de flujo del proyecto es el siguiente (figura 2-19):



Fuente: Creación propia basado en diagrama del proyecto

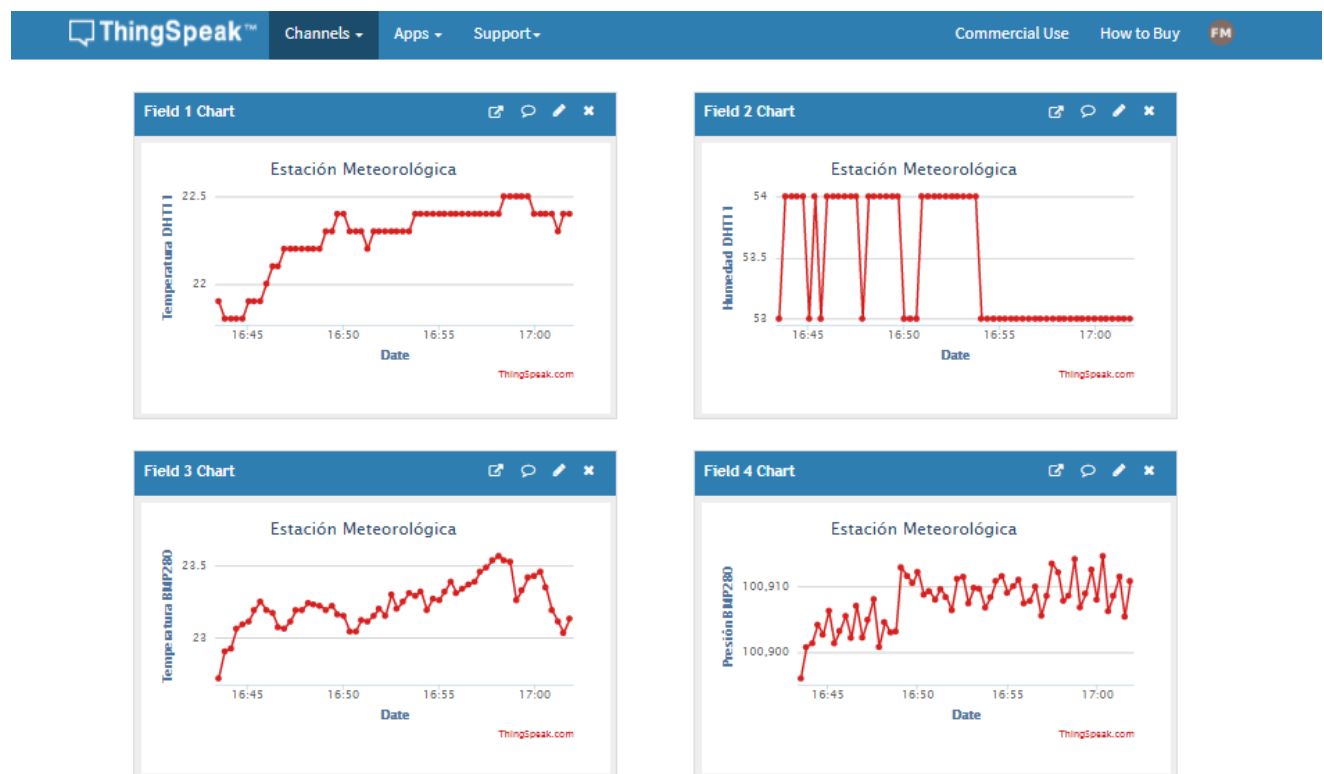
Figura 2-19: Diagrama de flujo

CAPÍTULO 3: RESULTADOS OBTENIDOS Y EVALUACIÓN DE COSTOS

3.1 RESULTADOS OBTENIDOS Y EVALUACIÓN DE COSTOS

En el presente capítulo se darán a conocer los resultados obtenidos del circuito de la estación meteorológica WIFI enviando los datos a ThingSpeak para poder observarlos de cualquier parte a través de una aplicación.

Al hacer funcionar el circuito este enviará los datos cada 20 segundos, como se ve en la programación del anexo A, una vez ya armado el circuito y programado el módulo ESP32 los datos se grafican en los campos que se adjuntaron los distintos datos (figura 3-1).



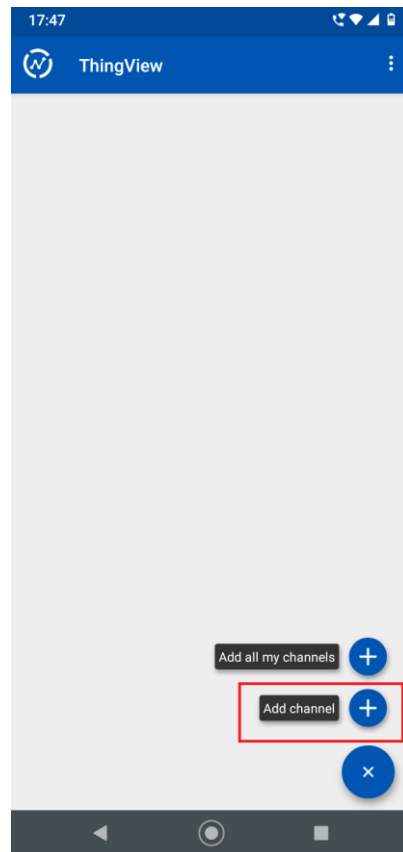
Fuente: https://thingspeak.com/channels/1332310/private_show

Figura 3-1: Gráfico de datos

3.2 THINGVIEW

ThingView permite visualizar los canales ThingSpeak de una manera fácil, basta con introducir el ID del canal si es público, si es un canal privado hay que tener el ReadAPIKey que se consigue en la pestaña de APIKey del canal en Thingspeak.

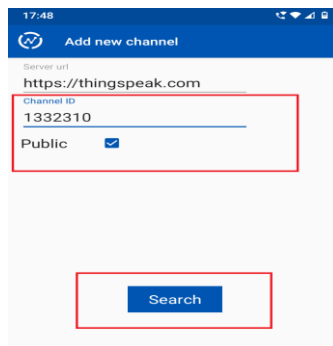
Primero se debe descargar la aplicación, como el canal fue público, solo se necesita el ID del canal, al ingresar en la aplicación agregar el canal como se muestra en la figura 3-2.



Fuente: Elaboracion propia, basada en Thingview

Figura 3-2: Agregar canal

Luego solo es requerido el ID del canal para ingresar y se selecciona la casilla que dice Public, finalmente Search (figura 3-3).



Fuente: Elaboración propia, basada en Thingview

Figura 3-3: ID del canal

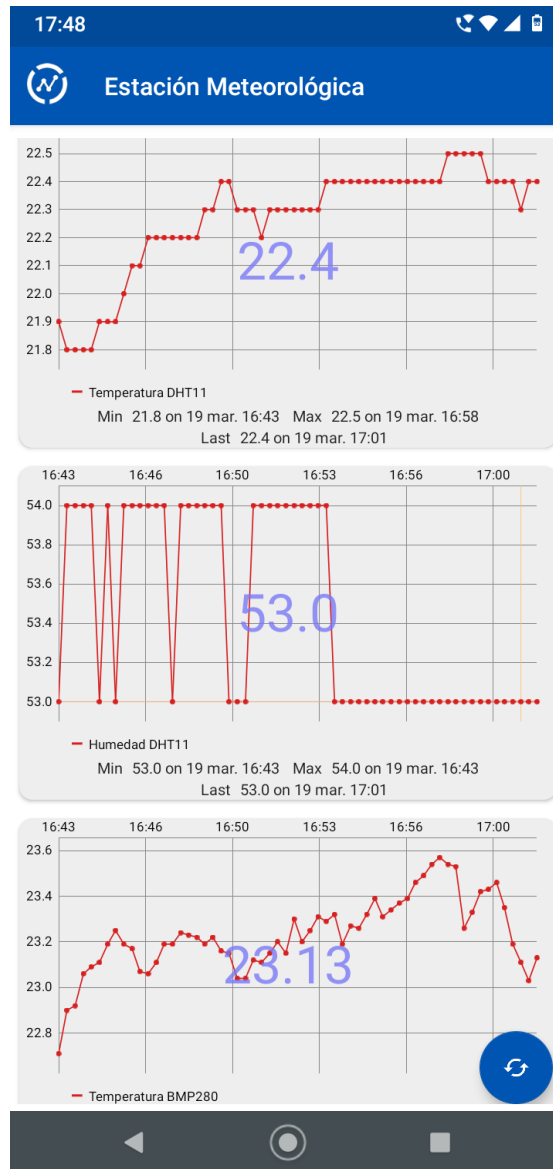
Después aparece el nombre del canal, la descripción y todos los campos que se ingresaron en ThingSpeak, como se muestra en la figura 3-4.



Fuente: Elaboración propia, basada en Thingview

Figura 3-4: Datos en aplicación del canal

Finalmente se logra ver los gráficos con los datos que el módulo ESP32 envió a Thingspeak por la red de WIFI (figura 3-5).

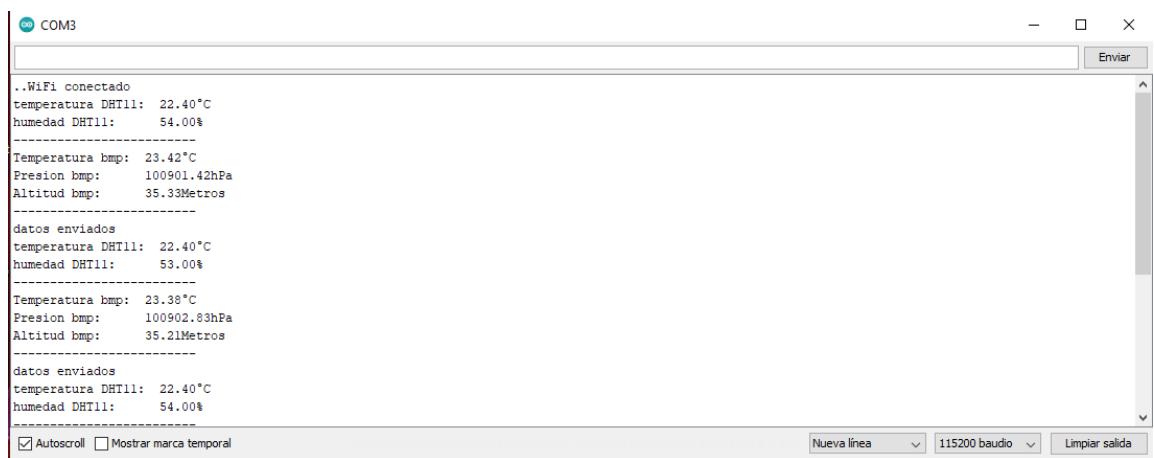


Fuente: Elaboración propia, basada en gráficos de Thingview

Figura 3-5: Gráficos en la aplicación

3.3 MONITOR SERIE

En la estación meteorológica, una forma para verificar que el microcontrolador está enviando los datos correctamente es a través del monitor serie, incluido en el IDE de Arduino como muestra la figura 3-6.



Fuente: Elaboración propia, basada en monitor serie

Figura 3-6: Monitor Serie

3.4 EVALUACIÓN DE COSTOS

El costo es el valor económico que posean los componentes que se utilizarán en el proyecto, tanto bienes tangibles e intangibles que se van a proponer en la actividad que se desea desarrollar. En este caso se verá el costo específico de los materiales a utilizar.

3.4.1 Componentes del proyecto

Para poder monitorear los datos que entrega la estación meteorológica es necesario contar con diversos componentes, en la tabla 3-1 se verán los componentes que se utilizarán en el proyecto.

Tabla 3-1: Componentes del proyecto

Componentes	Cantidad
Modulo ESP32	1
Sensor DHT11	1
Sensor BMP280	1
Cable DUPONT MACHO/MACHO	10
Protoboard 400 Pts	2

Fuente: Elaboración propia, basada en componentes del proyecto

Siendo todos estos materiales demasiados importantes y necesarios para la fabricación de la estación, siendo estos los elegidos debido a la facilidad del trabajo a implementar, además de un costo adecuado para cualquier persona o alumno que desee realizar el proyecto.

3.4.2 Módulos WIFI

Existen diversos Microcontroladores que tienen las funciones de WIFI, en la tabla 3-2 se aprecian distintos dispositivos, como dato extra para este proyecto el módulo ESP32 es proporcionado por la Universidad.

Tabla 3-2: Cotización de Microcontroladores

Componentes	Precio	Empresa
Módulo ESP32	\$ 8.490	Mechatronicstore
Raspberry Pi	\$55.900	Mercado Libre
Arduino UNO y ESP-01	\$11.021	Global Chile

Fuente: Elaboración propia, basada en costos de Microcontroladores

3.4.3 Sensor DHT11

Como se menciona anteriormente en el capítulo 2 el sensor DHT11 es un sensor de temperatura y humedad, el cual entrega los datos necesarios para la estación, tabla 3-3.

Tabla 3-3: Cotización sensor DHT11

Componentes	Precio	Empresa
DHT11	\$1.600	Global Chile
DHT11	\$2.976	Mercado Libre
DHT11	\$2.300	Afel

Fuente: Elaboración Propia, basada en costos de sensor DHT11

3.4.4 Sensor BMP280

El sensor BMP280 es un sensor barométrico que entrega los datos de temperatura, presión y altitud, ofreciendo más datos muy importantes a la hora de monitorear un proyecto que se desee, tabla 3-4.

Tabla 3-4: Cotización Sensor BMP280

Componentes	Precio	Empresa
BMP280	\$1.980	Global Chile
BMP280	\$2,891	Altronics
BMP280	\$2.300	Afel

Fuente: Elaboración propia, basada en costos de sensor BMP280

3.4.5 Cable DUPONT MACHO/MACHO

Estos cables son necesarios a la hora del armado del circuito, permiten la conexión entre los pines de los sensores, módulo y protoboard, tabla 3-5.

Tabla 3-5: Cotización cables DUPONT

Componentes	Precio	Empresa
Cable DUPONT	\$30 c/u	Global Chile
Cable DUPONT	\$1.490 – 40 Pcs.	kowka
Cable DUPONT	\$2,650- 40 Pcs	Altronics

Fuente: Elaboración propia, basada en costos de cable Dupont

3.4.6 Protoboard

La protoboard permite que todo el circuito funcione siendo la placa de pruebas para todo proyecto. En este proyecto se utilizarán 2 para mayor comodidad., tabla 3-6.

Tabla 3-6: Cotización Protoboard

Componentes	Precio	Empresa
Protoboard 400 Pts.	\$1.900	Global Chile
Protoboard 400 Pts.	\$2.300	Afel
Protoboard 400 Pts.	\$2.990	MClelectronics

Fuente: Elaboración propia, basada en costos de Protoboard

3.4.7 Licencias ThingSpeak

Las licencias académicas de ThingSpeak (tabla 3-7) son para investigación académica y docencia. El precio se basa en la cantidad de canales requeridos y en la cantidad de mensajes que se procesarán y almacenarán en un período de un año.

ThingSpeak está disponible como un servicio gratuito para pequeños proyectos académicos no comerciales con límites en ciertas funciones, para proyectos más grandes y límites ampliados, se encuentran disponibles distintas licencias.

ThingSpeak se compra en unidades, donde una unidad permite procesar y almacenar 33 millones de mensajes en un período de un año (aproximadamente 90.000 mensajes por día), una unidad también brinda la capacidad de obtener una creación de hasta 250 canales en ThingSpeak. Precio Dólar \$719.91 considerado con la fecha 1 de Marzo del 2021

Tabla 3-7: Cotización ThingSpeak

Plan	USD/ Año	CLP/ Año
Student	\$79	\$56.873
Home	\$95	\$68.391
Academic	\$250	\$179.978
Standard	\$650	\$467.942

Fuente: Elaboración propia basado en precios de ThingSpeak

3.4.8 Costos Hora-Hombre

Para medir los esfuerzos necesarios y completar la estación meteorológica, es necesario calcular los costos por Hora-Hombre, basado en el sueldo de un Ingeniero en Ejecución en Control e Instrumentación Industrial, con un ingreso promedio al cuarto año de \$1.400.000 pesos, datos obtenidos por la página www.mifuturo.cl.

También tener en cuenta las 45 horas de trabajo semanales. El costo Hora Hombre asciende a \$31.111,111. Para fines prácticos se utiliza el valor de \$31.111 pesos chilenos, ver tabla 3-8 y tabla 3-9.

Tabla 3-8: Costos por Hora-Hombre

Tarea	Horas	Valor Hora-Hombre
Investigación del proyecto	30	\$933.330
Cotización de materiales	5	\$155.555
Implementación	7	\$217.777
Pruebas de funcionamiento	5	\$155.555
Total	57	\$1.462.217

Fuente: Elaboración propia, basada en valor Hora-Hombre

Tabla 3-9: Carta Gantt

Tarea	fecha inicio	fecha final	duración	Días (9 horas)						
				1	2	3	4	5	6	7
Investigación del proyecto	01-03-2021	04-03-2021	4							
Cotización de materiales	05-03-2021	05-03-2021	1							
Implementación	08-03-2021	08-03-2021	1							
Pruebas de funcionamiento	09-03-2021	09-03-2021	1							

Fuente: Creación propia basado en carta Gantt del proyecto

3.4.9 Costos por envíos de materiales

La empresa distribuidora Starken es la que se encarga de enviar los materiales a los alumnos, teniendo esta una buena calidad además de un bajo costo.

Cabe mencionar que el paquete de envío tiene las medidas aproximadas de 15 cm de alto, 15 cm de ancho y 15 cm de largo, con un peso de 5 Kg. El costo de envío es de \$6.130, recalcar que la distancia máxima en la cotización es a la ciudad de Santiago.

Datos entregados por la página <https://www.starken.cl/cotizador>

3.4.10 Costos de imprevistos

Para un supuesto en donde 8 microcontroladores contienen fallas internas, 10 envíos no se pueden realizar por problemas de la pandemia y 15 sensores BMP280 están averiados, se genera un costo de imprevistos, como lo muestra la tabla 3-10.

Tabla 3-10: Costos de imprevistos

Imprevisto	Cantidad	Costo
Fallas internas micro.	8	\$67.920
Envíos no realizados	10	61.300
Sensores averiados	15	\$29.700
Costo total	\$158.920	

Fuente: Elaboración propia, basada en costos de imprevistos

3.4.11 Costo de implementación

Para poder analizar el costo de implementación de la estación meteorológica se estipula un curso de 30 alumnos, ver tabla 3-11.

Tabla 3-11: Costo de implementación

Componentes	Cantidad	Costo por alumno	Costo curso
Modulo ESP32	1	\$8.490	\$254.700
Sensor DHT11	1	\$1.600	\$48.000
Sensor BMP280	1	\$1.980	\$59.400
Cable DUPONT MACHO/MACHO	10	\$300	\$9.000
Protoboard 400 Pts	2	\$3.800	\$144.000
Costo total	\$485.100		

Fuente: Elaboración propia, basada en costos de implementación.

3.4.12 Costo final

Al utilizar la versión gratuita de ThingSpeak no se agrega a los costos totales, también indicar que si se compran los componentes en la empresa que ofrece más baratos sus productos, el valor total del proyecto disminuye (tabla 3-12), siendo así más accesible para cualquier alumno que desee realizar el proyecto.

Tabla 3-12: Costos totales

Componentes	Valor
Costo de componentes	\$485.100
Costo Hora Hombre	\$1.462.217
Costos por envíos	\$183.900
Costos de Imprevistos	\$158.920
Costo total	\$2.290.137

Fuente: Elaboración propia, basada en costos totales

3.5 RESULTADOS DE APRENDIZAJE

Con el proyecto en cuestión es posible adquirir la gran parte de los resultados de aprendizaje del Programa de la asignatura Microcontroladores, en la carrera de Técnico Universitario en Electrónica, siendo el RdA1, RdA2, RdA3, RdA4, RdA5, RdA6, RdA7 los resultados que se pueden obtener en este proyecto, debido a que es necesario tener un conocimiento de lenguaje de programación y de microcontroladores para poder realizar la estación meteorológica.

CONCLUSIONES Y RECOMENDACIONES

En el año 2020 y parte del año 2021 se ha hecho casi imposible que los alumnos obtengan los conocimientos prácticos en las diferentes áreas, mediante el presente proyecto se observa una solución practica y sencilla, que busca que los alumnos aprendan el funcionamiento del dispositivo ESP32, teniendo este módulo infinidad de usos.

En el proyecto se demuestra que no es necesario hacer una inversión demasiado grande para poder entender los conceptos básicos de la programación y el módulo con el que se está trabajando.

Dentro de las ventajas que ofrece la estación meteorológica, es el monitoreo de variables como la temperatura, humedad, presión y altitud, esta estación es posible utilizar en distintos proyectos, ya sea de tipo invernadero o no, lo ideal siempre es entender que es posible saber a través de WIFI lo que sucede en el lugar que se requiere monitorear

Cabe recalcar que mediante este proyecto fue posible corroborar y reforzar los conocimientos en cuanto a la programación en lenguaje C, para llevar a cabo este propósito se tuvo que evaluar distintos microcontroladores del mercado, siendo escogido el módulo ESP32, gracias a que este dispositivo es facilitado por la Universidad, además de no necesitar ningún periférico o placa extra para lograr una conexión WIFI.

BIBLIOGRAFÍA

Sensor DHT11, disponible en <https://programarfacil.com/blog/arduino-blog/sensor-dht11-temperatura-humedad-arduino/>

[Consulta: Enero/2021- Marzo/2021]

Sensor DHT22, disponible en <https://naylampmechatronics.com/sensores-temperatura-y-humedad/58-sensor-de-temperatura-y-humedad-relativa-dht22-am2302.html>

[Consulta: Marzo/2021- Julio/2021]

Modulo ESP32, disponible en <https://es.wikipedia.org/wiki/ESP32>

[Consulta: Enero/2021- Marzo/2021]

Arduino UNO, disponible en https://es.wikipedia.org/wiki/Arduino_Uno

[Consulta: Marzo/2021- Julio/2021]

Sensor BMP 280, disponible en <http://www.esp32learning.com/code/esp32-and-bmp280-sensor-example.php>

[Consulta: Enero/2021- Marzo/2021]

Sensor BMP 180, disponible en <https://naylampmechatronics.com/sensores-posicion-inerciales-gps/75-sensor-de-presion-bmp180.html>

[Consulta: Marzo/2021- Julio/2021]

Comunicación I2C, disponible en <https://hetpro-store.com/TUTORIALES/i2c/>

[Consulta: Marzo/2021- Julio/2021]

Sueldo de Técnico Universitario en Electrónica, disponible en <https://www.mifuturo.cl/buscador-de-empleabilidad-e-ingresos/>
[Consulta: Marzo/2021- Julio/2021]

Costos de envíos, disponible en <https://www.starken.cl/cotizador>
[Consulta: Marzo/2021- Julio/2021]

Módulo Wifi ESP-01, disponible en <https://www.iberobotics.com/producto/esp-01-modulo-wifi-esp8266>
[Consulta: Marzo/2021- Julio/2021]

Raspberry Pi, disponible en https://es.wikipedia.org/wiki/Raspberry_Pi
[Consulta: Marzo/2021- Julio/2021]

ANEXO A: PROGRAMACIÓN DEL MICROCONTROLADOR ESP32

```

#include <WiFi.h>
#include <ThingSpeak.h>
#include <Adafruit_BMP280.h>
#include "DHT.h"           // librerías para los sensores, ThingSpeak y WIFI!!

#define pin1 13

const char* ssid= "ssid";           // ssid y password de la red local
const char* password= "clave";

unsigned long channelID= 1332310;    // ID del canal y su apikey
const char* WriteAPIKey= "5SJCDAUGILT6TYMW";

WiFiClient cliente;                 // agregar el cliente

DHT dht1(pin1, DHT11);              // configuración de DHT11

Adafruit_BMP280 bmp;                // configuración de BMP

void setup() {
    Serial.begin(115200);             // velocidad del ESP32
    Serial.println("Conectando a WIFI");

    WiFi.begin(ssid,password);       // conexión del wifi con clave y password
    while(WiFi.status() !=WL_CONNECTED){ // bucle para conectar el WIFI
        delay(500);
        Serial.print(".");
    }
    Serial.println("WiFi conectado");

    ThingSpeak.begin(cliente);        // conexión del esp32 con thingspeak

    dht1.begin();
    bmp.begin(0x76); // se fuerza la dirección 0x76 para la comunicación i2c
}

void loop() {

    delay(2000);
    leer_dht1();

    delay(2000);
    leer_bmp();

    ThingSpeak.writeFields(channelID,WriteAPIKey); // escribe los datps de los
    sensores en los campos de thingspeak
    Serial.println("datos enviados");
    delay(14000);

```

```

void leer_dht1(){
    float t1= dht1.readTemperature(); // leer los datos entregado por el sensor
DHT11
    float h1= dht1.readHumidity();

while (isnan(t1) || isnan(h1)){ //revisa si el DHT11 entrega los
parametros correctos, pregunta si son numeros
    Serial.println("lectura incorrecta, repetir");
    delay (2000);
    t1= dht1.readTemperature();
    h1= dht1.readHumidity();
}

    Serial.print("temperatura DHT11: ");
    Serial.print(t1);
    Serial.println("°C");

    Serial.print("humedad DHT11: ");
    Serial.print(h1);
    Serial.println("%");
    Serial.println("-----");

    ThingSpeak.setField(1,t1); //se configura que los datos ingresen en los
campos 1 y 2 de thingspeak
    ThingSpeak.setField(2,h1);
}

void leer_bmp(){

    float temp= bmp.readTemperature(); // leer las varianles de presion,
temperatura y altitud del BMP280
    float presion= bmp.readPressure();
    float altitud= bmp.readAltitude();

    Serial.print("Temperatura bmp: "); // muestran las variables por el monitor
serie
    Serial.print(temp);
    Serial.println("°C");

    Serial.print("Presion bmp: ");
    Serial.print(presion);
    Serial.println("hPa");

    Serial.print("Altitud bmp: ");
    Serial.print(altitud);
    Serial.println("Metros");
    Serial.println("-----");

    ThingSpeak.setField(3,temp); // se configura que los datos ingresen en los
campos 3, 4 y 5 de thingspeak
    ThingSpeak.setField(4,presion);
    ThingSpeak.setField(5,altitud);
}
}

```

ANEXO B: PRPGRAMA DE LA ASIGNATURA

Asignatura: MICROCONTROLADORES		Sigla: EIN224-A	Fecha de aprobación 22 agosto 2019		
Créditos SCT: 5	Prerrequisitos: Sistemas Digitales (cursada) Programación (cursada)	Examen: No tiene	Departamento docente que la imparte		
			Departamento De Electrotecnia e Informática/Departamento De Electrónica e Informática		
Horas Semanal Cátedra: 1,75	Horas Semanal Ayudantía: No tiene	Horas Semanal Laboratorio: 1,75	Semestre en que se dicta		
			Impar	Par X	Ambos
Eje formativo: Áreas de Instalación, Operación y Mantenición de Equipos y Sistemas Electrónicos					
Tiempo total de dedicación a la asignatura: 123,5 horas cronológicas					

DESCRIPCIÓN DE LA ASIGNATURA

Esta asignatura forma parte del plan de estudios del Técnico Universitario en Electrónica proporcionando conocimientos de programación de microcontroladores que le permiten desarrollar proyectos en el área de electrónica, incorporando los más diversos sensores, actuadores y visualizaciones gráficas. La lógica desarrollada le permitirá resolver variados problemas o desafíos, utilizando lenguajes de programación dirigidos al uso de microcontroladores.

REQUISITOS DE ENTRADA

Poseer conocimientos de código decimal, binario y hexadecimal.
Nociones básicas de unidades de información como bit, byte, nibble, Word.
Conocimientos básicos de programación.
Implementación de circuitos básicos en protoboard.

CONTRIBUCIÓN AL PERFIL DE EGRESO

TÉCNICO UNIVERSITARIO EN ELECTRÓNICA

Competencias de egreso

- 1.1.1. Interpretar Planos, manuales y/o Software de equipos y sistemas electrónicos.
- 1.2.1. Efectuar ajustes y calibración de instrumentos de medición para revisión de equipos y sistemas electrónicos.
- 2.1.1. Utilizar equipos y sistemas electrónicos según especificaciones técnicas del fabricante.
- 2.2.1. Utilizar manuales, software, herramientas e instrumental para la operación de equipos y sistemas de tecnología electrónica.
- 3.1.1 Identificar procesos de mantenimiento preventivo y/o correctivo de equipos y sistemas electrónicos según especificaciones técnicas.

RESULTADOS DE APRENDIZAJE QUE SE ESPERAN LOGRAR EN ESTA ASIGNATURA

- RdA1:** Identifica la simbología utilizada en diagramas de flujo, para la programación secuencial de sistemas embebidos.
- RdA2:** Analiza la operación y funcionamiento de sistemas electrónicos basados en microcontroladores, considerandola información técnica del fabricante.
- RdA3:** Opera sistemas de desarrollo asociados a microcontroladores, según información técnica del fabricante.
- RdA4:** Utiliza software de simulación para contrastar los valores prácticos con los valores teóricos de señales en circuitos basados en microcontroladores, considerando la información técnica del fabricante.
- RdA5:** Aplica técnicas de programación en un microcontrolador para el desarrollo de proyectos, según especificaciones técnicas de este.
- RdA6:** Elabora informes técnicos de las soluciones propuestas en base a microcontroladores, utilizando las herramientas ofimáticas según formatos establecidos.
- RdA7:** Verifica en forma experimental el funcionamiento de sistemas electrónicos basados en microcontroladores, según especificaciones técnicas.

ANEXO C: EXPERIENCIA DE LABORATORIO

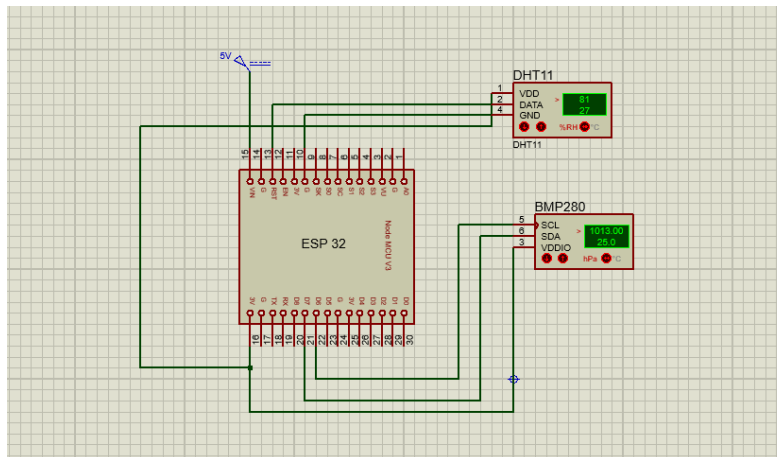
Experiencia Microcontroladores

Título: Modulo ESP32 y ThingSpeak

Objetivo: Diseñar una estación meteorológica para medir a través de la conexión WIFI, las variables de temperatura, humedad, presión y altitud.

Descripción.

1. Implemente el siguiente circuito he inserte el programa del anexo A en el microcontrolador.



2. A través del monitor serie revise si los sensores entregan los valores de las variables que se necesitan.
3. Una vez que el circuito opera de manera óptima, crear el canal en la plataforma Thingspeak.
4. Revisar los gráficos entregados por Thingspeak.
5. Descargue la aplicación Thingview o a través de la página de Thingspeak, compruebe si el canal y los gráficos están operando.