



**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y
TELECOMUNICACIONES
INGENIERÍA ELECTRÓNICA**

TRABAJO DE GRADO PARA OPTAR EL TÍTULO DE INGENIERO ELECTRÓNICO

**TÍTULO:
PLATAFORMA METEOROLÓGICA BASADA EN SISTEMA EMBEBIDO, UTILIZANDO
FIRESTORE Y FRAMEWORK WEB ANGULAR PARA LA MEDICIÓN DE VARIABLES
CLIMÁTICAS EN LA CIUDAD DE PAMPLONA**

**AUTOR:
CARLOS ADRIÁN OROZCO PINZÓN
C.C. 1118831749**

**DIRECTOR:
M.SC(C) LUIS ALBERTO MUÑOZ BEDOYA**

**CODIRECTOR
M.SC HEBER ARMANDO PABÓN CONDE**

**UNIVERSIDAD DE PAMPLONA
INGENIERÍA ELECTRÓNICA
PAMPLONA 2020**

AGRADECIMIENTOS

En primera instancia agradecerle a Dios que fue el promotor a impulsarme día a día a superar cada obstáculo y superarlo haciendo posible este nuevo paso en mi vida. A mis padres, mis hermanos, mis tíos, primos que siempre creyeron en mi. Indira Pinzón mi madre querida siempre alentándome para no rendirme aun en los momentos mas duros, Edwin Suarez mas que un padre me apoyo nunca dudo de mi, Jhon Pinzón gracias a tus consejos hoy estoy aquí logrando mis sueños gracias tío, Lindys Toro más que mi novia mi compañera mi confidente estuviste ahí incondicionalmente nunca me dejaste desistir gracias, Juan Rodríguez y Edgar Medina gracias le doy a Dios por mi paso en esta carrera fui cosechando amistades que mas que unos amigos son mis hermanos siempre estuvieron ahí en las buenas y en las malas gracias mil gracias.

También quiero agradecer a todas las personas que formaron parte de este proceso, al ramillete de profesores de la UNIVERSIDAD DE PAMPLONA, amigos, compañeros, Luis Muñoz mi director mil gracias por estar ahí siempre apoyándome y corrigiéndome para que hoy sea posible este sueño, de igual manera al codirector Heber Pabón que mas que un apoyo fue un amigo mas gracias a su experiencia y consejos hoy estoy aquí cumpliendo unos de mis sueños.

**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y
TELECOMUNICACIONES
PROGRAMA DE INGENIERÍA ELECTRÓNICA
TRABAJO PRESENTADO PARA OPTAR POR EL TÍTULO DE
INGENIERO ELECTRÓNICO**

TEMA:

**PLATAFORMA METEOROLÓGICA BASADA EN SISTEMA EMBEBIDO, UTILIZANDO
FIRESTORE Y FRAMEWORK WEB ANGULAR PARA LA MEDICIÓN DE VARIABLES
CLIMÁTICAS EN LA CIUDAD DE PAMPLONA**

**FECHA DE INICIO DEL TRABAJO: 19-MAY-2019
FECHA DE TERMINACIÓN DEL TRABAJO: 28-MAY-2020**

NOMBRES Y FIRMAS DE AUTORIZACIÓN PARA LA SUSTENTACIÓN:

CARLOS ADRIÁN OROZCO PINZÓN

AUTOR

LUIS ALBERTO MUÑOZ BEDOYA

JOSÉ DANIEL RAMÍREZ CORSO

DIRECTOR

DIRECTOR DE PROGRAMA

JURADO CALIFICADOR:

JURADO 1

JURADO 2

JURADO 3

**PAMPLONA NORTE DE SANTANDER
COLOMBIA
JUNIO-12-2020**

RESUMEN

El presente proyecto muestra la idea de desarrollo de una plataforma conformada por las siguientes herramientas tecnológicas: un sistema embebido (sobre el cual se soportará la creación de la estación meteorológica, la electrónica de sensores, la comunicación mediante internet y una base de datos para registrar las variables climatológicas), un framework de desarrollo web para visualizar la información climatológica desde una computadora o un dispositivo móvil (monitoreo remoto mediante internet), una base de datos en tiempo real para mantener actualizada la información de la estación y finalmente la implementación de un sistema de alimentación fotovoltaico autónomo para los sistemas electrónicos que lo conforman. El objetivo será ubicar el sistema dentro de las instalaciones del campus principal de la universidad donde la estación tendrá acceso a internet.

Palabras claves: Sensores, variables climatológicas, sistema embebido, angular, firebase, estación meteorológica, monitoreo remoto, panel solar, web, base datos.

ABSTRACT

This project shows the idea of developing a platform made up of the following technological tools: an embedded system (on which the creation of the meteorological station will be supported, the sensor electronics, communication via the Internet and a database for a recorder climate variables), a web development framework to view weather information from a computer or mobile device (remote monitoring via the internet), a real-time database to keep station information up to date, and finally the implementation of a autonomous photovoltaic power system for the electronic systems that comprise it. The objective will be to locate the system within the facilities of the main campus of the university where the station will have internet access.

Key words: Sensors, climatological variables, embedded system, angular, firebase, meteorological station, remote monitoring, solar panel, web, database.

CONTENIDO

AGRADECIMIENTOS	2
RESUMEN	4
ABSTRACT	5
1. INTRODUCCIÓN	10
1.1 Objetivos.....	11
1.1.1 General.....	11
1.1.2 Específicos	11
2. ESTADO DEL ARTE	12
2.1 Antecedentes.....	12
2.1.1 Estación meteorológica inalámbrica.	12
2.1.2 Construcción de una estación meteorológica de acceso remoto (EMAR).	12
2.1.3 Prototipo de una estación meteorológica.	12
2.2 Bases teóricas	13
2.2.1 Estación meteorológica.	13
2.2.2 Herramientas web.	14
2.2.3 Angular.....	14
2.2.4 Firebase.	16
3. DISEÑO DE LA PLATAFORMA METEOROLÓGICA	17
3.1 Descripción general y criterios de diseño.....	17
3.2 RASPBERRY PI 3 MODELO B.....	19
3.2.1 Descripción.....	19
3.2.2 Alimentación.....	20
3.2.3 Interfaces de usuario.	21
3.3 Medición de la presión atmosférica, la temperatura y la altitud	21
3.3.1 Características.....	22

3.3.2	Asignación de pines.....	22
3.3.3	Circuito típico de conexión.....	23
3.3.4	Protocolo de Comunicación I2C (BMP180).....	23
3.3.5	Sensor BMP180 (Lectura)	25
3.3.6	Conectar BMP180 a RASPBERRY PI 3 MODELO B.....	26
3.4	Medición de la humedad relativa	26
3.4.1	Especificaciones técnicas y características.	27
3.4.2	Asignación de pines y circuito típico de conexión.	27
3.4.3	Obtención de la humedad relativa y de la temperatura.	28
3.4.4	Diagrama de flujo.	29
3.4.5	Conectar DHT11 a RASPBERRY PI 3 MODELO B.....	31
3.5	Medición de la velocidad del viento	32
3.5.1	Descripción.....	32
3.5.2	Principio de funcionamiento.....	33
3.5.3	Medición de viento.....	33
3.6	Medición de la radiación solar (UV).....	34
3.6.1	Descripción.....	34
3.6.2	Características.....	35
3.6.3	Asignación de pines.....	35
3.6.4	Circuito típico de conexión.....	36
3.6.5	Sensor VEML-6075 (Lectura)	36
3.6.6	Conectar VEML-6075 a RASPBERRY PI 3 MODELO B.....	38
3.7	Medición de las gotas de lluvia.....	38
3.7.1	Descripción.....	39
3.7.2	Características.....	39
3.7.3	Asignación de pines.....	40
3.7.4	Conectar YL-83 a RASPBERRY PI 3 MODELO B.....	40

3.8	Cálculos para la alimentación Solar.....	41
3.8.1	Características del panel solar.....	41
3.8.2	Horas de funcionamiento.....	41
3.8.3	Banco de baterías.....	42
3.9	Diseño de la aplicación en Angular CLI.....	43
3.9.1	Consideraciones.....	43
3.9.2	Descripción.....	44
3.10	Entornos de programación.....	45
3.10.1.	Lenguaje Python.....	45
3.10.2.	Lenguaje HTML.....	46
3.10.3.	Lenguaje TypeScript.....	47
3.10.4.	Formato JSON.....	47
3.11	Firebase.....	48
3.11.1.	Herramientas de Firebase.....	48
3.11.2.	Base de datos en tiempo real.....	49
3.11.3.	Autenticación.....	49
3.11.4.	Hosting.....	50
3.12	Diseño de la estación.....	50
3.12.1.	Circuitos electrónicos de acople.....	50
3.12.2.	Diseño de la PCB.....	51
3.12.3	Diseño en Solid Edge.....	52
3.13	Programación del sistema embebido.....	53
3.13.1	Script de control en Python.....	53
3.13.2.	Librerías en Python.....	54
3.13.3.	Credenciales de Firebase para Python.....	54
3.13.4.	Configuración de los pines GPIO.....	55
3.13.5.	Control de eventos para base de datos en la Raspberry.....	56

4.	RESULTADOS Y ANÁLISIS.....	57
4.1	Funcionamiento.	57
4.2	Fiabilidad de los datos.	59
4.3	Entorno Angular CLI.	622
4.4	Análisis económico	644
5.	CONCLUSIONES.....	655
6.	RECOMENDACIONES	666
6.1	Técnicas	666
6.2	Para trabajos futuros.....	666
	Glosario	688
	Referencias.....	699

ÍNDICE DE FIGURAS

Figura 1. Representación de Herramientas Web.....	14
Figura 2. Representación de Estructura Angular.....	15
Figura 3. Representación de Estructura Firebase.....	16
Figura 4. Diagrama de bloques de la plataforma meteorológica.....	18
Figura 5. Raspberry pi 3 modelo b detalles.....	19
Figura 6. Raspberry PI, diagrama de bloques.....	20
Figura 7. Sensor BMP180, aspecto. Sensor BMP180, aspecto.....	21
Figura 8. Distribución de pines del sensor BMP180.....	22
Figura 9. Sensor BMP180, circuito típico de conexión.....	23
Figura 10. Estructura bus I2C.....	23
Figura 11. Transferencia de datos. Transferencia de datos.....	24
Figura 12. Diagrama de flujo lectura BMP180.....	25
Figura 13. Sensor BMP180, conexión Raspberry Pi.....	26
Figura 14. Sensor DHT11, aspecto.....	26
Figura 15. Sensor DHT11, circuito típico de conexión.....	28
Figura 16. Sensor DHT11, protocolo de transmisión.....	28
Figura 17. Sensor DHT11, diagrama de flujo seguido para recuperar los datos del sensor.....	30
Figura 18. Sensor DHT11, diagrama de flujo de la interrupción.....	31
Figura 19. Sensor DHT11, conexión con la tarjeta RASPBERRY PI 3B.....	31
Figura 20. Anemómetro, diseño en Solid Edge.....	32
Figura 21. Anemómetro, diseño de sus partes en 3D.....	32
Figura 22. Anemómetro, impresión de partes.....	32
Figura 23. Motor, funcionamiento como dinamo, y modulo conversor AD.....	33
Figura 24. Distribución de pines del sensor VEML-6075.....	35
Figura 25. Sensor VEML-6075, circuito típico de conexión.....	36

Figura 26. Diagrama de flujo lectura VEML-6075.....	36
Figura 27. Tabla Niveles De Índice de Radiación Rayos UV.....	37
Figura 28. Sensor VEML-6075, conexión Raspberry Pi.....	38
Figura 29. Distribución interna del sensor YL-83.....	39
Figura 30. Distribución de pines del sensor VEML-6075.....	40
Figura 31. Sensor YL-83, conexión Raspberry Pi.....	40
Figura 32. Editor Visual Studio Code, programación Angular CLI	43
Figura 33. Editor Visual Studio Code, instalación plugins.....	44
Figura 34. Interfaz Página Web, mediciones variables.....	45
Figura 35. Sintaxis básica para un archivo JSON.....	48
Figura 36. Panel de control de Firebase para crear un proyecto.....	49
Figura 37. Circuito de acople para el control de voltaje.....	51
Figura 38. Circuito de acople para el control de voltaje montado.....	51
Figura 39. Diseño PCB de la estación en la aplicación Eagle.....	52
Figura 40. circuito de acople y diseño estación en físico.....	52
Figura 41. Diseño de la Estación en el software Solid Edge.....	53
Figura 42. Librerías importadas en el código Python.....	54
Figura 43. Cargar las credenciales e iniciar el objeto Firebase en Python.....	55
Figura 44. Configuración para los pines GPIO utilizados.....	55
Figura 45. Script para el bucle repetitivo controla los eventos de cambio.....	56
Figura 46. Prototipo de estación meteorológica.....	57
Figura 47. Almacenamiento de datos en la nube Firebase.....	58
Figura 48. Comparacion de las variables en tiempo real.....	59
Figura 49. Comparación de las variables 50 minutos después.....	60
Figura 50. Comparación de pruebas con Clima.com	60
Figura 51. Comparación de pruebas con es.meteocast.net	61
Figura 52. Comparación-Obs. Meteorológicas Aeropuerto Almirante Padilla –Riohacha.....	62

Figura 53. Credenciales Firebase para la aplicación Angular CLI.	623
Figura 54. Datos reflejados en la aplicación Angular.	634
Figura 55. Ejemplo interfaz gráfica de humedad.	645

ÍNDICE DE TABLAS

Tabla 1. Distribución de pines para el sensor BMP180. Fuente: (Bosch Sensortec, 2014).....	22
Tabla 2. Sensor DHT11: distribución de pines.	27
Tabla 3. Medición de lectura de viento	34
Tabla 4. Distribución de pines para el sensor VEML-6075.	35
Tabla 5. Distribución de pines para el sensor YL-83.	40
Tabla 6. Consumo total de la plataforma meteorológica.....	42
Tabla 7. Comandos CMD para instalar SDK Firebase para Python.....	54
Tabla 8. Costo de materiales.....	655

1. INTRODUCCIÓN

La posición geográfica de Colombia posibilita que en este país se presenten una gran variedad de climas a lo largo de todo su territorio. Si bien a diario se reciben a través de los medios de información un reporte del estado del tiempo facilitado por el Instituto de Hidrología, Meteorología y Estudios Ambientales (IDEAM), esta información no representa el conjunto de factores atmosféricos que caracterizan un entorno reducido. [3]

El interés en el estudio del clima radica en que para el desarrollo de ciertas actividades económicas como la agricultura, la industria, la navegación, la investigación, entre otras; es muy importante tener conocimientos acertados sobre el comportamiento de éste, ya que de este conocimiento se fundamentan decisiones de trascendencia e importancia en aspectos relacionados con dichas labores. Por ejemplo: en la aviación se sustentan las decisiones que pueden llevar a los pilotos a tomar rutas distintas según el tiempo que se espera; en la agricultura (una de las principales actividades económicas del país y del departamento de Sucre) facilita la toma de decisiones para llevar a cabo la realización del cultivo y las distintas actividades para su sostenimiento, lo que garantiza una mayor probabilidad de éxito en el mismo. Sin embargo, la falta o distribución irregular de estaciones meteorológicas en la región no permite la elaboración de un modelo numérico del comportamiento del clima que permita hacer predicciones meteorológicas. [4]

Aunque la motivación y razón para la realización del prototipo de la estación que describe este documento no pretende satisfacer específicamente ninguna de las actividades expuestas anteriormente (por ser precisamente un prototipo experimental), mediante el uso de la misma se podría realizar un monitoreo de los factores climáticos locales; un análisis de los datos obtenidos de éste permitiría la elaboración de un modelo del comportamiento del clima a nivel local, que favorecería a los estudiantes, personas o entidades de la ciudad que a partir de ciertas condiciones climáticas deban tomar decisiones para satisfacer algún requerimiento.

Además de la utilidad práctica comentada en el párrafo anterior, el desarrollo de este proyecto también posee valor teórico en el manejo de las tecnologías de vanguardia empleadas en el mismo, como lo son el manejo de sistemas embebidos y la conexión de dispositivos o sistemas a Internet.

Si bien se consiguen en el mercado sofisticadas estaciones meteorológicas que hacen uso de sistemas avanzados para el cálculo exacto de los distintos factores que afectan el estado de la atmósfera en un momento y lugar determinado (como es el caso de las estaciones meteorológicas automáticas en las cuales no se necesita de una persona que esté tomando los datos diariamente a determinada hora, sino que éstos son procesados, almacenados y transmitidos a un servidor local), la adquisición, instalación y mantenimiento de éstas representa un costo excesivo. El prototipo de estación meteorológica que describe el presente proyecto es una opción considerablemente económica respecto a una estación comercial.

1.1 Objetivos

1.1.1 General

- Desarrollar una plataforma meteorológica basada en un sistema embebido, usando base de datos web en tiempo real y Angular de Google.

1.1.2 Específicos

- Realizar los diseños electrónicos de acuerdo a las etapas de adquisición y acondicionamiento de las variables a medir en la estación meteorológica.
- Desarrollar un aplicativo con el Framework Web Angular que visualice el registro en tiempo real de la información de la estación meteorológica.
- Analizar las variables obtenidas y generar el dimensionamiento para los equipos que cumplan con la capacidad energética medible, acorde a los modelos generados mediante técnicas de procesamiento.
- Validar la funcionalidad de la plataforma mediante múltiples pruebas de campo y una comparativa con estaciones meteorológicas en la región.

2. ESTADO DEL ARTE

2.1 Antecedentes

Referente al proyecto de investigación que se realizó se encontraron antecedentes o estudios similares realizados a nivel nacional cuyo objetivo principal era la realización de una estación meteorológica con características y funciones similares a las que describen el presente trabajo. Éstas son:

2.1.1 Estación meteorológica inalámbrica.

Presentado en la Universidad Manuela Beltrán (Bogotá, Colombia) por Arango Parrado, Daniel A.; Rincón Infante, Freddy E.; Manrique Daza, Lisandro; Buitrago Rojas, Diana C., en diciembre de 2010. En este proyecto se implementó una estación meteorológica para monitorear cinco variables físicas: la temperatura, la presión atmosférica, la humedad relativa y la dirección y velocidad del viento. El sistema utiliza un microcontrolador *PIC* en el que se procesan las señales pre-acondicionadas de los sensores adjuntos y un celular en el cual se obtienen los valores medidos de las variables climáticas estudiadas a través del servicio SMS; estos datos son registrados y visualizados (Arango, Rincón, Manrique, & Buitrago, 2010). [11]

2.1.2 Construcción de una estación meteorológica de acceso remoto (EMAR).

Presentado en la Fundación Tecnológica Antonio de Arévalo (Cartagena de Indias, Colombia) por Genadiz Cardona, G. y Bolaños, M., en 2014. Esta estación permite reconocer el estado climático y atmosférico en los alrededores de la Fundación Tecnológica Antonio de Arévalo a través del uso de sensores y monitoreo vía web (Fundación Tecnológica Antonio de Arévalo [TECNAR], 2015). [13]

2.1.3 Prototipo de una estación meteorológica.

Presentado en la Universidad de Sucre (Sincelejo, Colombia) por Sergio Andrés Herrera Mendoza y Ricardo Carazo García en Junio del 2016. Este proyecto consistió en el diseño y realización de un prototipo de una estación meteorológica de bajo costo capaz de medir la temperatura ambiente, la presión atmosférica, la humedad relativa, la velocidad y la altitud, con sistema de envío de datos por SMS a un móvil o por comunicación serial o GPRS a un ordenador con Windows. Estos datos son registrados y visualizados con la herramienta *App Inventor* de *Google* hecha en Android, y en el caso del ordenador, por una aplicación hecha en Lenguaje C#.

2.2 Bases teóricas

2.2.1 Estación meteorológica.

Una estación meteorológica consiste en una estructura o dispositivo dotado con sensores (temperatura, humedad, velocidad de viento, dirección de viento, gas metano, etc.) que responde a estímulos eléctricos, los cuales permiten registrar y coleccionar información meteorológica en forma automática y en tiempo real. Esto permite realizar un monitoreo de variaciones de la temperatura del aire, humedad relativa, radiación solar, dirección y velocidad del viento, lluvia, humedad relativa, presión atmosférica, etc. [7],[4]

A la hora de instalar una estación meteorológica, se deben poner en marcha los sensores y de esa manera realizar mediciones de algunas variables del ambiente las cuales son:

Temperatura: La temperatura es la medida del contenido de calor de un cuerpo o del medio ambiente. En la gran mayoría de estaciones, es medida en grados centígrados (°C). Mediante la medida de esta, se puede medir la influencia de la temperatura en la velocidad en que se desarrollan los cultivos. [8]

Precipitación pluvial: La precipitación es la caída directa de gotas de agua o de cristales de hielo sobre la superficie terrestre. La medida se realiza por medio de un pluviómetro generalmente se usan unidades en milímetros (mm). [7]

Humedad ambiental: La humedad ambiental es la medida que nos dice la cantidad de vapor de agua presente en el aire. Se representa en porcentaje o grado de humedad. Es medida por un psicrómetro. Su importancia radica en la predicción de posibles heladas. También es una variable climática importante presente en el desarrollo de pronósticos de posibles enfermedades de cultivos. [7],[8]

Radiación solar: La radiación solar es el flujo de energía que se recibe del sol en forma de ondas electromagnéticas de diferentes frecuencias. Se mide en calorías por centímetro cuadrado por minuto o por día (cal/cm²/día) o en watts por metro cuadrado (W/m²) con un sensor llamado piranómetro. [7]

Velocidad del viento: Se describe por dos características: velocidad y dirección. El instrumento para medir a este se llama anemómetro y se mide en metros por segundo (m/s) o kilómetros por hora (km/h). [7]

Presión atmosférica: La presión atmosférica es el peso de la columna de aire que hay sobre cualquier punto o lugar de la tierra y es por tanto el peso por unidad de superficie. Cuanto mayor es la altura, menor es la presión atmosférica y cuanto menor es la altura y más se acerque a nivel del mar, mayor será la presión.

Altitud: altitud es la distancia vertical que existe entre un punto de la tierra y el nivel del mar. El término también se utiliza como sinónimo de altura en referencia a una distancia respecto a la tierra, a la región del aire a una cierta elevación sobre la superficie o a la dimensión de un cuerpo perpendicular a su base.

2.2.2 Herramientas web.

Las Herramientas web son una asociación de aplicaciones web que facilitan la comunicación, el diseño, el usuario y la colaboración con el WWW (World Wide Web), además permite a los usuarios interactuar y colaborar con la creación de contenidos, tanto los mismos usuarios como la información dada. [3]



Figura 1. Representación de Herramientas Web. Fuente.[3]

2.2.3 Angular.

Es un framework para la creación de páginas web SPA mantenido por Google. SPA es el acrónimo de 'Single Page Application' o lo que es lo mismo, cuando un usuario entra en una web SPA, se carga todo a la vez en una misma página y Angular lo que hace por debajo es cambiar la vista al navegar por la página para que de la apariencia de una web normal. [2]

Ventajas:

- Velocidad de carga lenta la primera vez que se abre la web, pero luego navegar por la web es instantáneo debido a que se carga toda la web de golpe.
- Cómo SPA (Single Page Application) es una página solo hay una ruta que tiene que enviar el servidor.
- Aplicaciones modulares y escalares. [2]

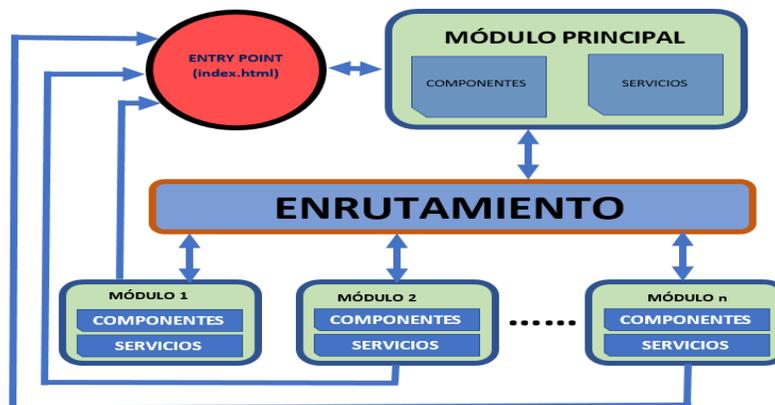


Figura 2. Representación de Estructura Angular. Fuente.[2]

Principales Características

Velocidad y Rendimiento:

- Generación de código: Angular convierte tus plantillas en código altamente optimizado para las máquinas virtuales de JavaScript de hoy en día, ofreciéndote todas las ventajas del código escrito a mano con la productividad de un framework.
- Universal: Ejecuta la primera vista de tu aplicación en node.js, .NET, PHP, y otros servidores para renderizado de forma casi instantánea obteniendo solo HTML y CSS. También abre posibilidades para la optimización del SEO del sitio.
- División del código: Las aplicaciones de Angular se cargan rápidamente gracias al nuevo enrutador de componentes. Éste ofrece una división automática de códigos para que los usuarios sólo carguen el código necesario para procesar la vista que solicitan. [2]

Productividad:

- Plantillas: Permite crear rápidamente vistas de interfaz de usuario con una sintaxis de plantilla simple y potente.
- Angular CLI: Las herramientas de línea de comandos permiten empezar a desarrollar rápidamente, añadir componentes y realizar test, así como pre visualizar de forma instantánea la aplicación.
- IDEs: Obtén sugerencias de código inteligente, detección de errores y otros comentarios en la mayoría de los editores populares e IDEs. [2]

2.2.4 Firebase.

Es la nueva y mejorada plataforma de desarrollo móvil en la nube de Google. Se trata de una plataforma disponible para diferentes plataformas (Android, iOS, web), con lo que de esta forma presentan una alternativa seria a otras opciones para ahorro de tiempo en el desarrollo. En muchas ocasiones nos planteamos cómo poder acceder a un servicio web para poder tener nuestra aplicación trabajando con datos en la nube. Por ello surgió Firebase, para proveer una API para guardar y sincronizar datos en la nube en tiempo real. [6]

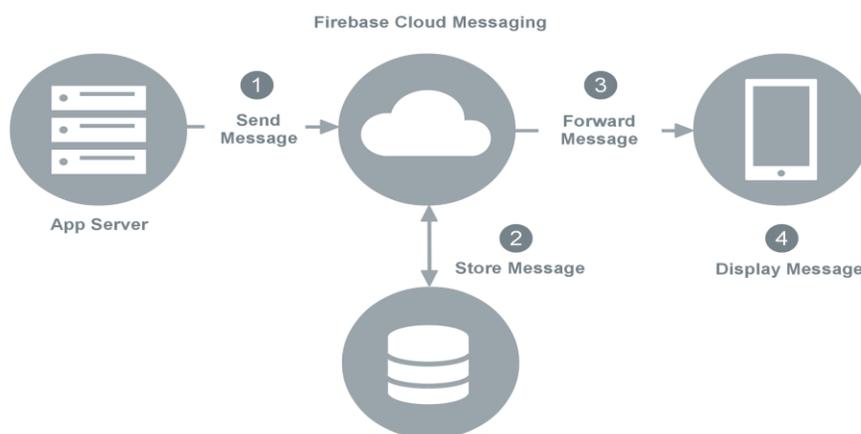


Figura 3. Representación de Estructura Firebase.
Fuente.[6]

Sus características fundamentales están divididas en varios grupos, las cuales podemos agrupar en:

- **Analíticas:** Provee una solución gratuita para tener todo tipo de medidas (hasta 500 tipos de eventos), para gestionarlo todo desde un único panel. [6]
- **Desarrollo:** Permite construir mejores apps, permitiendo delegar determinadas operaciones en Firebase, para poder ahorrar tiempo, evitar bugs y obtener un aceptable nivel de calidad. Entre sus características destacan el almacenamiento, testeo, configuración remota, mensajería en la nube o autenticación, entre otras. [6]
- **Crecimiento:** Permite gestionar los usuarios de las aplicaciones, pudiendo además captar nuevos. Para ello dispondremos de funcionalidades como las de invitaciones, indexación o notificaciones. [6]

3. DISEÑO DE LA PLATAFORMA METEOROLÓGICA

Este capítulo está dedicado al diseño de la plataforma meteorológica. Presenta una descripción completa de los elementos y subsistemas que conforman a la misma atendiendo a la metodología seguida para su elaboración y la integración de la misma en el sistema. De igual forma, presenta una justificación para el uso de dichos elementos y los procedimientos o métodos que se siguieron.

3.1 Descripción general y criterios de diseño

El objetivo general del presente proyecto señala el uso de sistema embebido Raspberry PI y un framework de desarrollo web para visualizar la información desde una computadora o un dispositivo móvil. Atendiendo a este principio, se eligió a la tarjeta Raspberry PI como unidad central de proceso. La tarjeta ejecuta el *firmware* que opera a los periféricos asociados que conforman a la estación, también procesa los datos provenientes de los sensores acerca del valor actual de las variables medidas, les da formato para que sean transmitidos inalámbricamente y posteriormente almacenados en el dispositivo remoto (celular u ordenador).

El uso del sistema embebido Raspberry PI se justifica por las altas prestaciones que esta tecnología ofrece, su flexibilidad y su bajo costo. Integra unos pines GPIO de entrada y salida de propósito general, bloques digitales y diferentes interfaces para la comunicación con el exterior suficientes para el desarrollo del prototipo. En cuanto al uso de la plataforma de firebase, es una herramienta web de google reciente, que se consideró factible su empleo en el proyecto como medio para base de datos de la información y exportarlos inalámbricamente; usar la red de telefonía celular, u ordenador resuelve el problema de la distancia o el alcance a la hora de realizar dicha labor.

Por tratarse de una plataforma cuyo fin no es satisfacer las necesidades para el desarrollo de ninguna actividad comercial en específico, se acordó medir las variables físicas más comunes o que aplican a la mayoría de actividades. Éstas fueron; la temperatura ambiente, la humedad relativa, la presión atmosférica, la precipitación de lluvia, la radiación solar, la rapidez del viento y, adicionalmente la altitud.

Otro de los requerimientos para el desarrollo de la plataforma es el empleo de hardware de bajo costo y con las prestaciones adecuadas. A la fecha, se encontraron en el mercado sensores con salida digital (el uso de sensores inteligentes ahorró realizar etapas para el acondicionamiento de señal a la salida de estos) económicos; con un rango de medición, resolución, precisión y exactitud satisfactorios; previamente calibrados o capaces de ejecutar sus propias rutinas de calibración y capaces de sensor más de una variable. Aunque su presentación no es lo suficientemente robusta para aplicaciones industriales o aplicaciones en los que el sensor es sometido a drásticas condiciones ambientales, se consideró acertado o adecuado el uso de estos sensores en la plataforma de estación meteorológica. Para medir la presión atmosférica se

eligió el sensor BMP180, además de medir la variable ya mencionada también puede medir la temperatura ambiente, y, como la presión atmosférica y la altitud se relacionan, a partir de la primera se calcula el valor de la segunda por medio de una expresión matemática de uso general. Para medir la humedad relativa se eligió al sensor DHT11, que además puede medir la temperatura, y para medir la dirección del viento se hizo necesario la construcción de un sensor Anemómetro. En cuanto a la radiación solar se utilizó el sensor VEML6075 capaz de detectar la intensidad de rayos UV y finalmente el sensor YL- 83 utilizado para detectar gotas de lluvia, como un sensor de lluvia, y seguimiento de humedad. [9]

La estación debía ser capaz de interactuar con un celular o un ordenador con conexiones previas a internet, para tal fin se elaboró una aplicación utilizando la herramienta *Angular*, de Google. La aplicación estará lanzada en un hosting adquirido a través de la herramienta *Firestore*, la cual permite subir los archivos y generar un dominio mediante el cual los clientes se podrán conectar con la aplicación. Se piden los datos de las variables a la estación, los cuales estarán leídos por la tarjeta *Raspberry PI* mediante un programa en el lenguaje *Python*, posteriormente estos datos serán actualizados en la base de datos de la herramienta *Firestore* (Real-time Database) usando archivos de texto, cuando la tarjeta *PI* actualice los datos en la *Firestore*, el cliente que ha ingresado a través del dominio se comportaría como un oyente en el momento que inicie la aplicación, este también escucharía ese cambio y se mostraría inmediatamente el valor que ha cambiado mediante la interface de la aplicación. [10]

En el diseño se consideró la realización de una fuente de alimentación dedicada a la estación. La tarjeta *Raspberry PI 3* puede ser alimentada también a través de un puerto *micro-usb* del cargador o a través de una fuente externa (Alimentación Fotovoltaica). La Figura 4 ilustra el diagrama de bloques del prototipo de estación meteorológica, el módulo de alimentación que allí se ilustra corresponde a un cargador con suficiente corriente y un panel solar de uso personal con suficiente potencia.

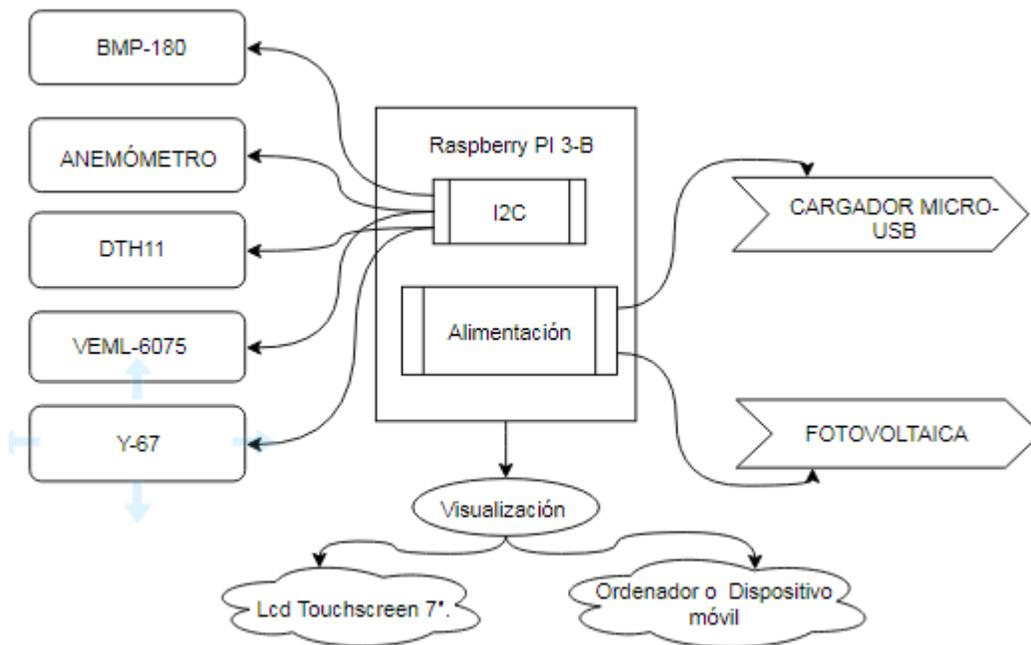


Figura 4. Diagrama de bloques de la plataforma meteorológica

En los subcapítulos siguientes a éste se presenta una descripción de cada uno de los elementos y funciones previamente mencionados/as y cómo se integraron los/as mismos/as al sistema.

3.2 RASPBERRY PI 3 MODELO B

En el Modelo 3 B el procesador Broadcom BCM2836 es sustituido por un modelo más avanzado, el Broadcom BCM2837 y sigue integrando CPU, GPU, DSP y SDRAM. (Figura 5),

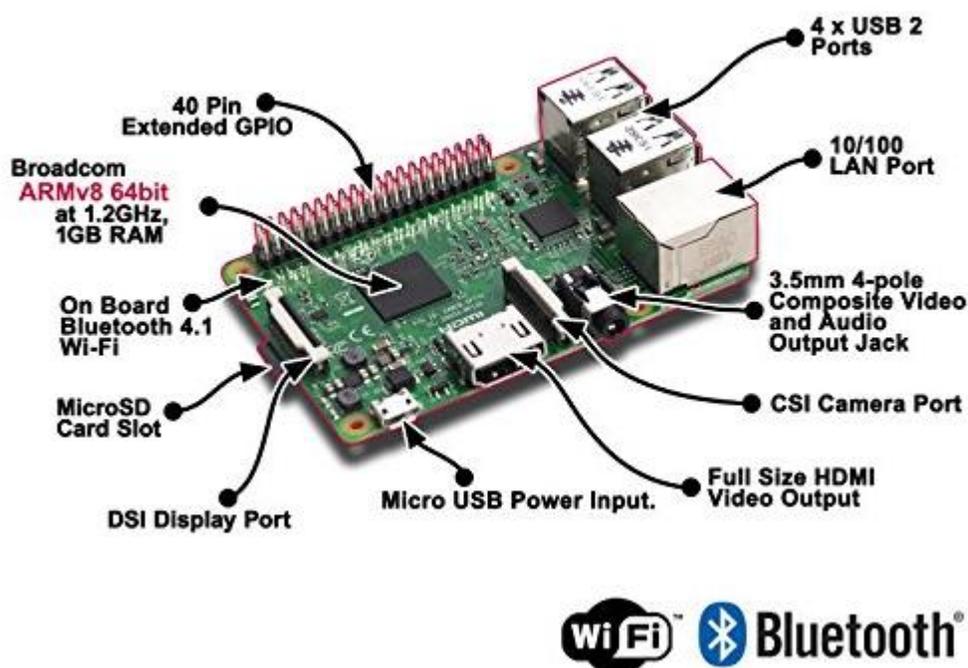


Figura 5. Raspberry pi 3 modelo b detalles.
Fuente.[1].

3.2.1 Descripción.

El procesador quad-core ARM Cortex A7 también evoluciona y tenemos un nuevo modelo, el quad-core ARM Cortex Av8 con una velocidad de 1.2GHz. El modelo 3 B ya cambia el modelo de RISC de 32 bits para pasar a trabajar en RISC de 64 bits. La GPU modelo Broadcom VideoCore IV cuenta con OPEN GL ES 2.0 MPEG-2 y VC-1 a resolución de 1080p. La memoria RAM sigue siendo de 1GB compartido con la GPU. 4 puertos USB 2.0. Entrada de video MIPI CSI para la conexión de la cámara. Dos salidas de video, un conector HDMI (rev, 1,3 y 1,4), también su interfaz DSI para conexión de panel LCD directo. Salidas de audio, minijack de 3.5mm y audio combinado del puerto HDMI. Puerto de almacenamiento micro SD. El modelo 3 B sigue

instalando el puerto de RJ45 de 100Mbps y además integra un nuevo chip de conectividad Wi-Fi N y Bluetooth 4.1. En el modelo 3 B, el conector GPIO de 17 pines y con bus HAT ID sigue montado. [1]

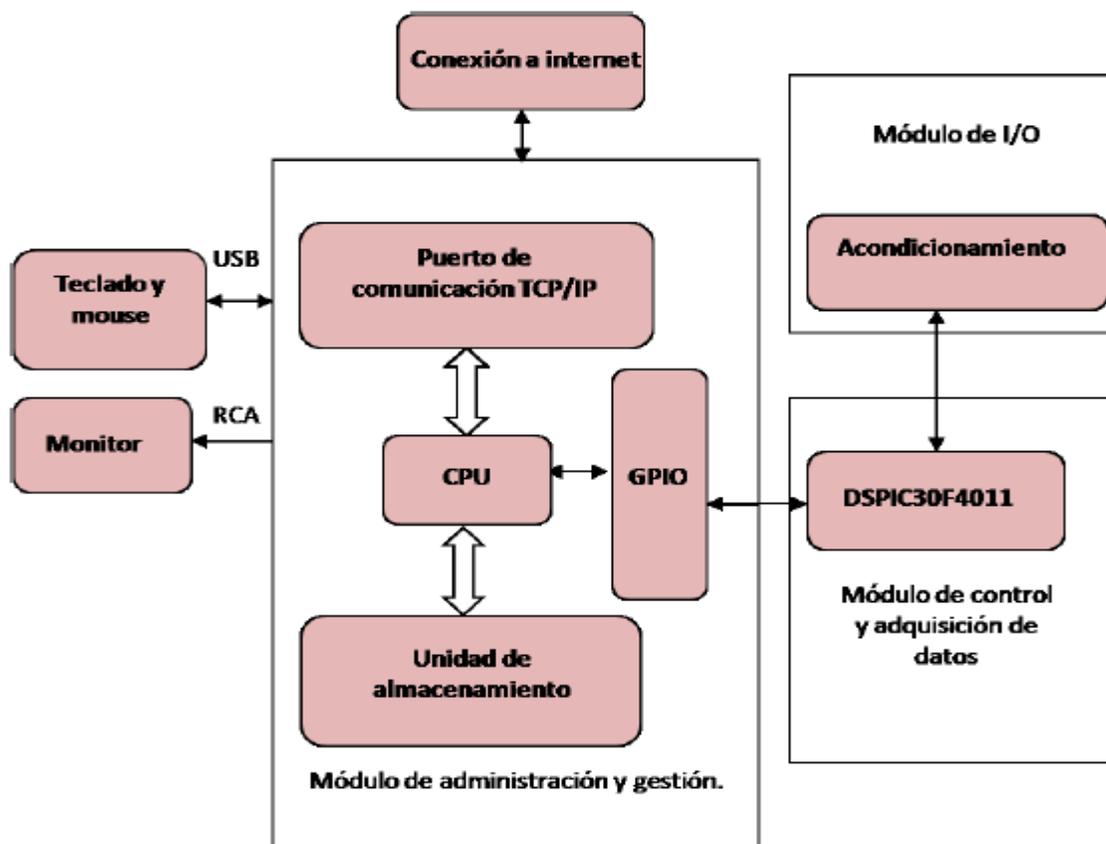


Figura 6. Raspberry Pi, diagrama de bloques.
Fuente.[4]

3.2.2 Alimentación.

Al igual que en los proyectos estáticos de Raspberry Pi, los usos portátiles de la Raspberry Pi deben ser alimentados por un adaptador de red micro USB de 5V de buena calidad con una corriente de 2A a 2.5A como máxima potencia usando todos los periféricos, y una corriente de 800mA como mínima potencia solamente usando la tarjeta sin conexión a los periféricos.

3.2.3 Interfaces de usuario.

Para la construcción de la interfaz gráfica es necesario tener en cuenta que Tkinter es una biblioteca, que posee integrada una gran cantidad de recursos que facilitan el diseño, así mismo que su desarrollo en el código es secuencial y maneja jerarquías para el control de ventanas. Además de Tkinter es necesario importar e instalar bibliotecas externas que no posee la Raspberry Pi, como la librería Adafruit la cual provee el protocolo de comunicación I2C para la captación de los sensores, urllib2 una librería que facilita el manejo y comunicación a páginas por medio de HTML, y matplotlib esta es la que permite agregar el desarrollo de gráficas y la animación en tiempo real. [1]

3.3 Medición de la presión atmosférica, la temperatura y la altitud

Para la medición de la presión atmosférica y la temperatura ambiente se utilizó el sensor inteligente BMP180 (Figura 7.) El sensor de presión barométrica BMP180 permite medir la altura respecto al nivel del mar, su funcionamiento está basado en la relación entre presión del aire y la altitud. El BMP180 es el reemplazo de BMP085, posee alta precisión y de bajo consumo de energía. Ofrece un rango de medición desde medición 0 a 1100 hPa (Hecto Pascal), con una precisión absoluta de hasta 0,03 hPa. Basado en tecnología piezo-resistiva de BOSCH con robustez EMC, alta precisión y linealidad, así como con estabilidad a largo plazo. Diseñado para ser conectado directamente a un micro controlador a través de I2C utilizando solo 2 líneas. [9]

Este tipo de sensores pueden ser utilizados para calcular la altitud con gran precisión, por lo que es un sensor muy utilizado en sistemas de Autopiloto para Drones (UAVs).

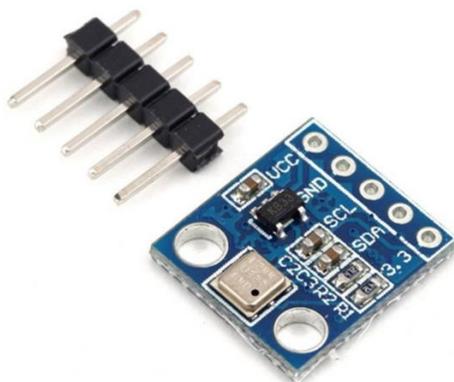


Figura 7. Sensor BMP180, aspecto. Sensor BMP180, aspecto
Fuente.[9]

3.3.1 Características.

Entre las principales características para este sensor se pueden señalar (Bosch Sensortec, 2014): Voltaje de Operación: 3.3V - 5V DC Interfaz de comunicación: I2C (3.3V). [12]

- 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.
- Completamente calibrado.
- Rango operativo de -40 a + 85 ° C.

3.3.2 Asignación de pines.

La Tabla 1. Muestra la asignación de pines y la Figura 8 ilustra el diagrama de pines para el sensor BMP180.

Tabla 1. Distribución de pines para el sensor BMP180. Fuente.[9]

Pin. No.	Nombre	Función
1	I2C	Serial Data
2	I2C	Serial Clock
3	GND	Power
4	VCC	Power



Figura 8. Distribución de pines del sensor BMP180. Fuente.[9]

3.3.3 Circuito típico de conexión.

La Figura 9 ilustra el circuito típico de conexión. El sensor se suelda en una PCB con un regulador de 3.3V, un cambiador de nivel I2C y resistencias pull-up en los pines I2C. El BMP180 reemplaza al BMP085. [11]

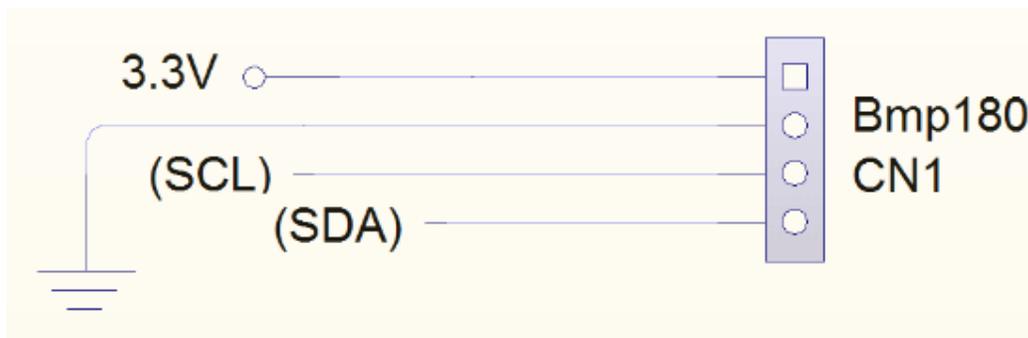


Figura 9. Sensor BMP180, circuito típico de conexión.
Fuente.[1]

3.3.4 Protocolo de Comunicación I2C (BMP180)

El sensor BMP180 posee una característica especial, los datos de la adquisición se realizan en paquetes de bytes, asociada al protocolo de comunicación I2C. El protocolo de comunicación I2C, es un protocolo sincrónico que fue desarrollado por Phillips en 1922, el cual por medio de dos hilos permite comunicar dispositivos siempre a la orden de un maestro, logrando velocidades de hasta 100Kbits por segundo. [10]

El bus de comunicación I2C, permite la conexión simple de dos o más dispositivos mediante dos hilos, que transmiten bit a bit y están dados por los siguientes.

- SCL,(Serial Clock). Señal de reloj que se usa para sincronizar los datos.
- SDA,(Serial Data). Es la línea de transferencia de datos serie.

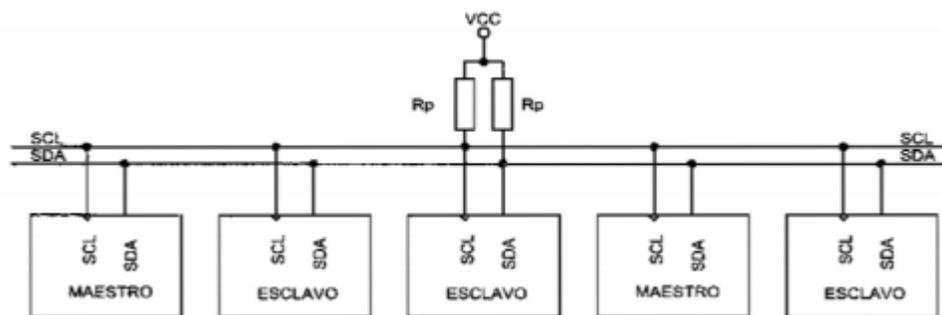


Figura 10. Estructura bus I2C.
Fuente: adaptada de (<http://visystem.ddns.net>).

En la anterior figura (Fig. 10) se muestra la topología de este bus de datos, en el cual está un maestro que por lo general se trata de un micro-controlador y los esclavos asociados a dispositivos que se comunican con el maestro.

El uso de resistencias de pull up en las líneas SCL y SDA corresponden al hardware o construcción física propia del bus I2C, en este caso se tienen dispositivos similares a conexiones de colector abierto, esto quiere decir que aniveles bajos (0 volts) el bus de comunicación se encuentra activo, bien sea enviando o recibiendo información, de lo contrario si sus niveles son altos (3.3-5.5 volts) estará inactivo. Cabe resaltar que este bus es un protocolo de comunicación para cortas distancia y comúnmente usado en implementación entre placas o tarjetas electrónicas, para una buena transmisión de información, Phillips recomienda el uso de resistencias de pull up en las líneas SCL y SDA comprendidas entre 2K Ω y 4.7K Ω . [10]

De acuerdo a lo anterior, la trama de datos enviada a través de este bus corresponde a una serie característica de bit.

- Primero un bit de start que señala el comienzo del envío.
- Siete bits de direccionamiento del esclavo.
- Un bit correspondiente a Lectura/Escritura (R/W).
- Un bit de reconocimiento (ACK).
- Mensaje dividido en BYTES.
- Un bit de stop que finaliza la comunicación.

La trama que se envía, es la anteriormente mencionada, y se puede apreciar en la siguiente figura (Fig.11).

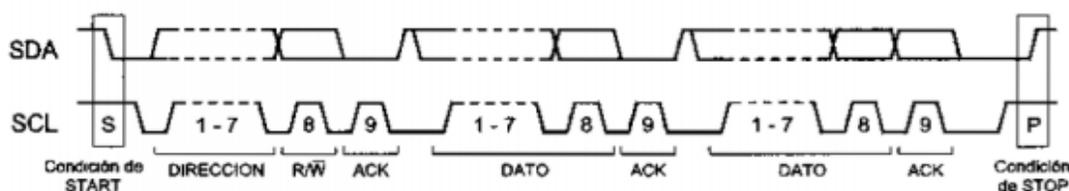


Figura 11. Transferencia de datos.
Fuente.[11]

Lo anterior corresponde a las nociones básicas que se deben tener en cuenta para la correcta implementación del protocolo de comunicación I2C, esencial para la adquisición de las variables de presión atmosférica y temperatura por parte del sensor BMP180.

3.3.5 Sensor BMP180 (Lectura)

Para la lectura correcta del sensor, el fabricante especifica el diagrama de flujo con el cual se puede realizar una adecuada adquisición de los datos. Es necesario conocer cada registro del sensor para no tener ambigüedades a la hora de pedir un dato específico a este.

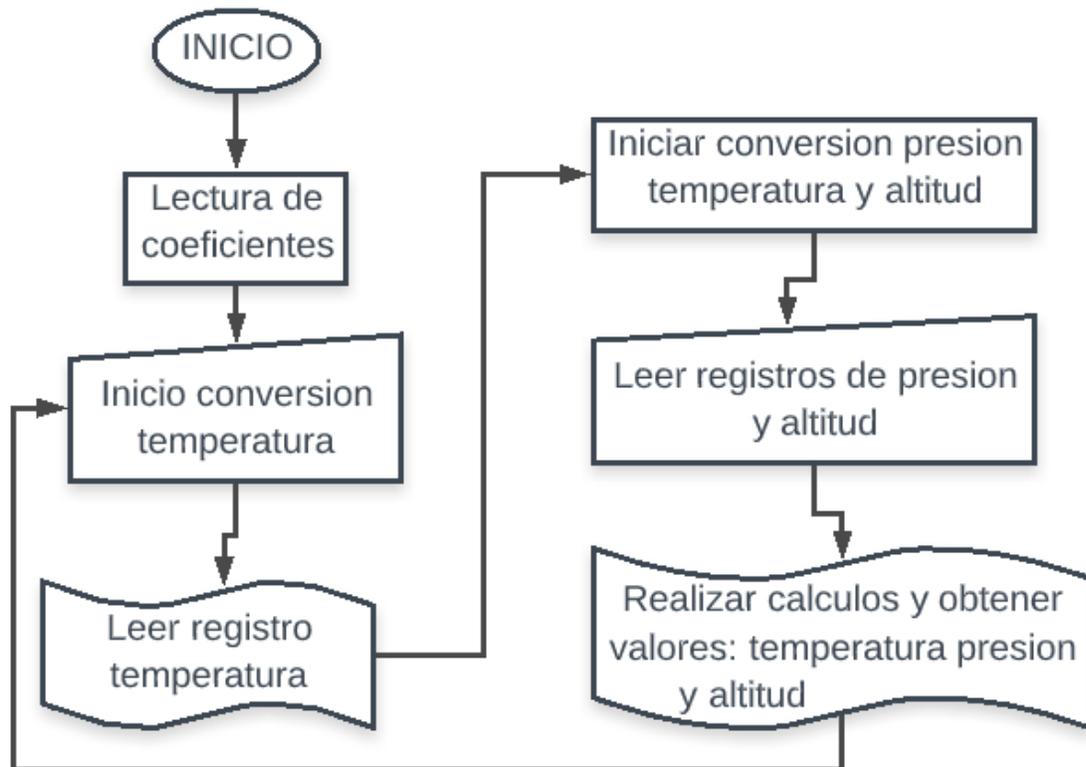


Figura 12. Diagrama de flujo lectura BMP180.
Fuente.[11]

En la anterior figura (Fig. 12) se puede detallar el diagrama de flujo que se implementó para la adquisición de los parámetros desde el sistema embebido. En primera instancia el manejo de dicho sensor indica que se debe hacer la adquisición de una serie de coeficientes, con los cuales posteriormente se realiza una serie de operaciones para así llegar a los valores correctos de temperatura y presión atmosférica, cabe resaltar que dichos coeficientes nada más se adquieren una vez en todo el proceso, y no se deben alterar de ninguna forma. [9]

Por último el datasheet del sensor BMP180 especifica el contenido en Bytes por cada variable, concluyendo que para la presión atmosférica corresponde a 3 Bytes, 2 Bytes para la parte entera y 1 Byte para la parte flotante así como para temperatura son 2 Bytes totales, 1 Byte para la parte entera y 1 Byte para la parte flotante, que luego se podrán procesar para la aplicación de envío.

3.3.6 Conectar BMP180 a RASPBERRY PI 3 MODELO B.

La Figura 13 ilustra el diagrama esquemático para la conexión del sensor BMP180 a la tarjeta RASPBERRY PI. Nótese que no aparecen las resistencias de pull up que establece la norma para el protocolo I2C, esto es porque estas resistencias ya están habilitadas internamente en el sensor. [9]

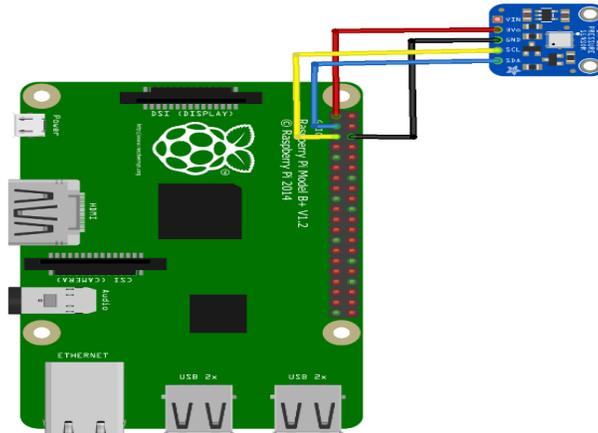


Figura 13. Sensor BMP180, conexión Raspberry Pi.
Fuente.[9]

3.4 Medición de la humedad relativa

Para la medición de la humedad relativa se utilizó el sensor inteligente DHT11 como el que se muestra en la Figura 14. El DHT11 es un sensor diseñado para la medición de la humedad relativa y de la temperatura ambiente del lugar en donde opere. Es fabricado por *D-Robotics UK* (www.droboticsonline.com). Consta de un componente para la medición de la humedad y un componente para la medición de la temperatura, además de un microcontrolador de 8-Bits. Está protegido contra interferencias y es de respuesta rápida. Almacena coeficientes de calibración para compensar el valor obtenido de las variables que soporta. [9]

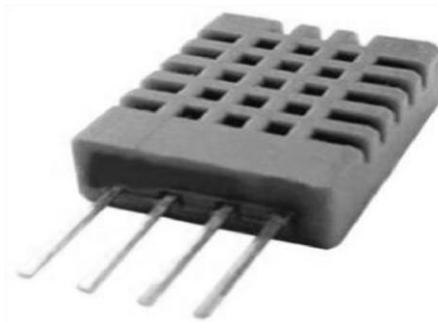


Figura 14. Sensor DHT11, aspecto.
Fuente.[9]

3.4.1 Especificaciones técnicas y características.

Entre las principales características para el DHT11 están las siguientes:

- Alimentación de 3V a 5.5V.
- Rango de medida para la humedad: 20-90%
- Error relativo en la medida de la humedad: $\pm 5\%$
- Rango de medida para la temperatura: 0-50 °C.
- Error relativo en la medida de la temperatura: $\pm 2\%$
- Resolución: 1
- Bajo consumo.
- Encapsulado de 4 pines

3.4.2 Asignación de pines y circuito típico de conexión.

La Figura 15 ilustra el circuito típico de conexión del sensor DHT11. El fabricante recomienda el uso de una resistencia de *Pull-Up* de valor 5K Ω en el pin de datos y un condensador de 100nF de para filtrar la alimentación. La Tabla 2 muestra la asignación de pines para este sensor. [12]

*Tabla 2. Sensor DHT11: distribución de pines.
Fuente: (Bosch Sensortec, 2015 pagina 2-1).*

Pin. No.	Nombre	Función
1	VCC	Power
2	Data	Digital bidireccional
3	Null	
4	GND	Power

Nota: el pin número 3 no se utiliza.

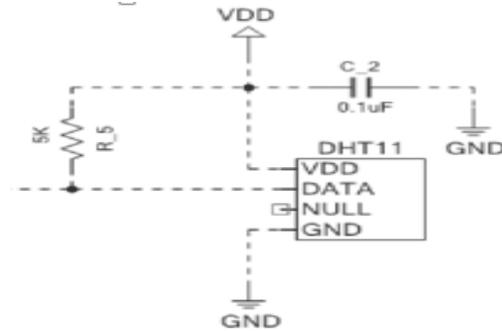


Figura 15. Sensor DHT11, circuito típico de conexión.
Fuente.[9]

3.4.3 Obtención de la humedad relativa y de la temperatura.

El DHT11 no utiliza un protocolo de comunicación estándar para la transmisión de datos con el sistema embebido, éste implementa un protocolo propio, en el cual la comunicación se realiza a través de una sola línea bidireccional (Pin 2). La Figura 16 ilustra este proceso.

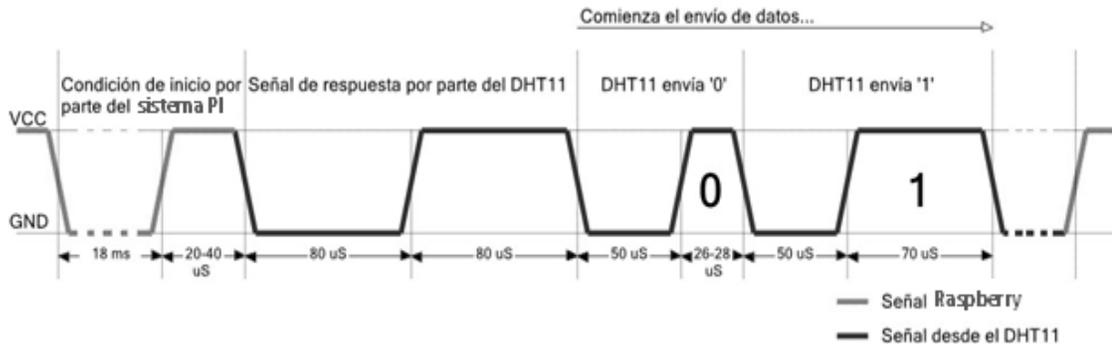


Figura 16. Sensor DHT11, protocolo de transmisión.
Fuente.[1]

Para solicitar el envío de datos la Raspberry debe poner la línea de transmisión a cero lógico durante 18 ms y luego ponerla a uno lógico para esperar la repuesta del sensor, éste se tardará entre 20 μ S y 40 μ S en responder. La señal de respuesta consiste en un pulso en bajo y un pulso en alto durante 80 μ S, seguido de la señal de respuesta empezará el envío de datos. Independientemente del estado lógico, antes de enviar un bit el sensor DHT11 pondrá a nivel bajo la línea de transmisión durante 50 μ S, inmediatamente después pondrá la línea en alto durante 26 μ S a 28 μ S si es un cero o durante 70 μ S si es un uno. El envío de datos terminará luego de haber enviado 40 bits. Al final de la transmisión la Raspberry debe poner la línea de comunicación a nivel alto para la próxima transmisión. [10]

Los 40 bits que se reciben del sensor DHT11 están divididos en 5 bytes, los cuales representan:

- El primer byte es la parte entera de la humedad relativa.
- El segundo byte es la parte decimal de la humedad relativa. En este sensor no se utiliza y siempre devuelve cero.
- El tercer byte es la parte entera de la temperatura.
- El cuarto byte es la parte decimal de la temperatura. Tampoco se utiliza en este sensor y siempre devuelve cero.
- El último byte es el *checksum* o suma de verificación, y es la suma de los anteriores bytes transmitidos.

3.4.4 Diagrama de flujo.

La Figura 17 ilustra el algoritmo en el que se basa la función programada para la obtención de los datos del sensor. En la RASPBERRY PI se configuró un pin como bidireccional y se le asignó una interrupción por flanco de subida. Se determina el estado del bit recibido testeando el estado de la línea en función de los tiempos conocidos de estos, esto se muestra en la Figura 18.

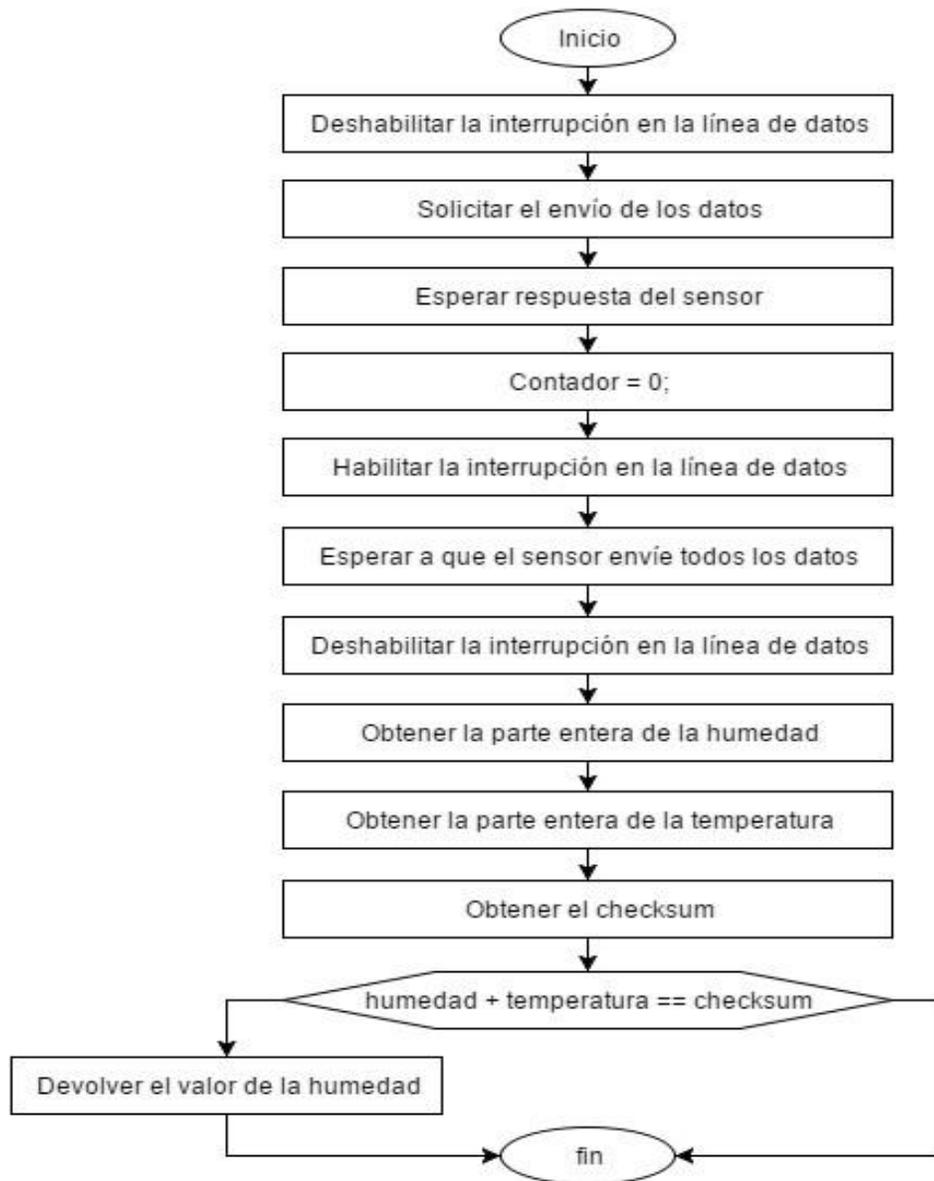


Figura 17. Sensor DHT11, diagrama de flujo seguido para recuperar los datos del sensor.
Fuente.[11]

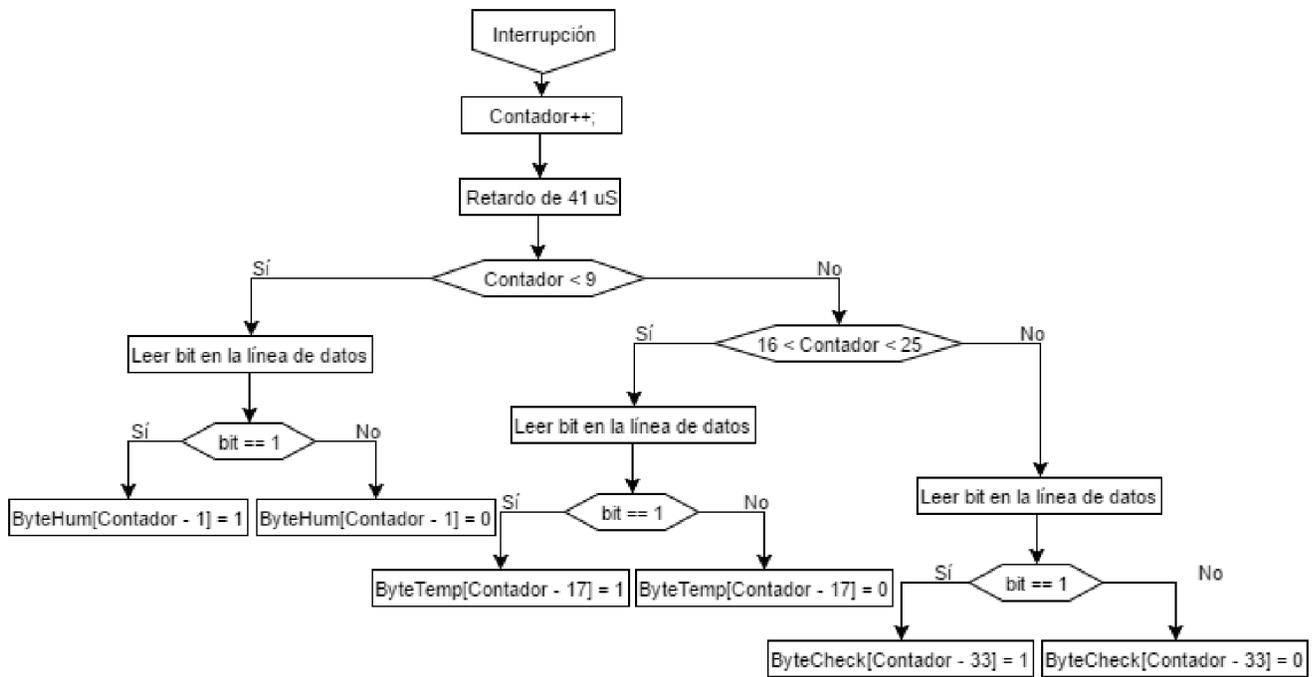


Figura 18. Sensor DHT11, diagrama de flujo de la interrupción. Fuente.[11]

3.4.5 Conectar DHT11 a RASPERRY PI 3 MODELO B.

La Figura 19 ilustra el diagrama esquemático usado en el prototipo de estación meteorológica para conectar el sensor DHT11. [15]

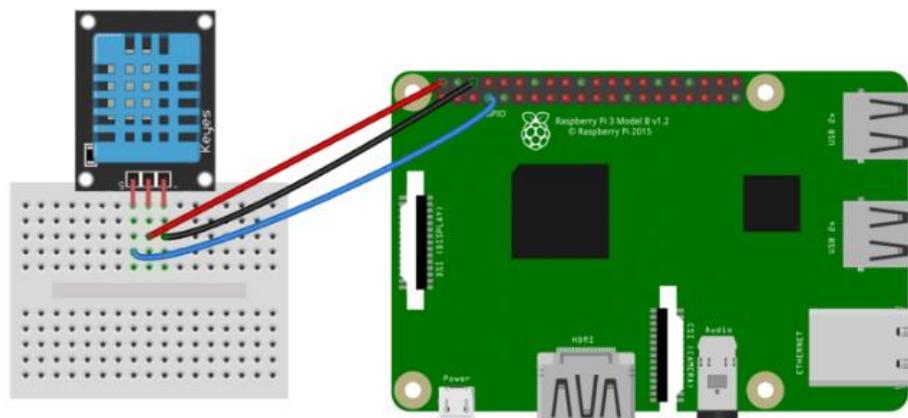


Figura 19. Sensor DHT11, conexión con la tarjeta RASPERRY PI 3B.

3.5 Medición de la velocidad del viento

Por cuestiones de costo y accesibilidad para la medición de la velocidad del viento no se pudo adquirir del mercado ningún sensor diseñado para tal fin. Así que se optó por el diseño y realización de uno propio.

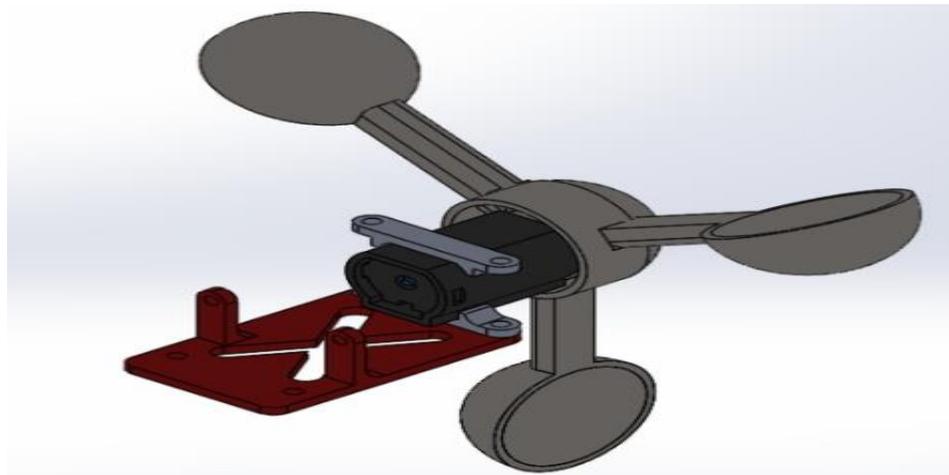


Figura 20. Anemómetro, diseño en Solid Edge.

3.5.1 Descripción.

El anemómetro diseñado se ilustra en la Figura 20, consta de tres cazoletas adheridas a uno de los extremos de un eje central, perpendiculares a éste y separadas 120° entre sí. Para esto fue necesario la ayuda del programa Solid Edge, ya que gracias a su contenido y diseño asistido por computadora, fue de gran ayuda para el diseño las piezas en tres dimensiones 3D. [16]



Figura 21. Anemómetro, diseño de sus partes en 3D.



Figura 22. Anemómetro, impresión de partes

En el diseño se utilizó un motor en corriente directa dc, que me va a servir como dimano a la hora de tomar los datos de la velocidad del viento, este va estar protegido por las piezas diseñadas en Solid Edge como se ve en la Figura 22, en el eje del motor encaja perfectamente en la base donde irán las aspas que le darán la velocidad del viento según sea la fuerza de giro. En la Figura 21 en la segunda imagen de izquierda a derecha podemos observar el diseño de una base la cual es de vital importancia a la hora emprender su funcionamiento porque determinara la estabilidad al momento de sensar. [15]

3.5.2 Principio de funcionamiento.

La forma semiesférica de las aspas, su distribución y la reducida resistencia mecánica del eje del motor facilitan el impulso de éstas por acción del viento. El *motor* gira simultáneamente con las aspas y en sus terminales tendremos la conexión tipo dinamo es decir, este genera una corriente eléctrica pulsante o continua, cuya tensión depende de la velocidad de rotación: aumentando el número de revoluciones aumenta también la tensión.[3]

Por qué un motor en DC?

El principio de funcionamiento de una dinamo es necesario referirse a una máquina elemental constituida por un imán y por una espira conductora, que se mueve entre los polos del imán. Tal y como sucede en el disco de Faraday (disco que gira en un campo magnético producido por un imán, constituyendo así el generador más sencillo de fuerza electromotriz), en el conductor se origina una tensión. Cerrando el circuito mediante una resistencia, una bombilla por ejemplo, véase en la se tiene el paso de corriente.[13]

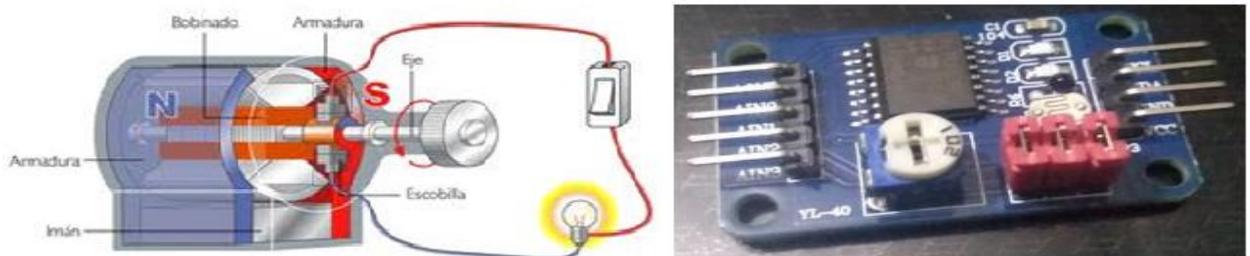


Figura 23. Motor, funcionamiento como dinamo, y modulo conversor AD.

3.5.3 Medición de viento.

Para las mediciones respectivas del anemómetro configurado como una dinamo, el multímetro encargado de tomar las lecturas en un rango de los milivoltios (mV). Fue necesario tener en cuenta el día de la medición estuviera sin viento que para las lecturas

respectivas fueran lo más correcto posible, en la tabla 3. Se observan las mediciones respectivas en un intervalo de 10Km, fue necesario utilizar un amplificador operacional para las lecturas puesto que son muy pequeñas y la tarjeta Raspberry Pi no tomaba bien este registro, y puesto que la misma no cuenta con entradas análogas se necesitó de un módulo conversor ADC, véase en la Figura 23.

Tabla 3. Medición de lectura de viento

Velocidad (Km/h)	Medición (mV)
10	35
20	82
30	125
40	190
50	250
60	320
70	390
80	430
90	480
100	550

3.6 Medición de la radiación solar (UV)

El sensor VEML-6075 tiene la peculiaridad de que los Output de luz visible e infrarroja son independientes, y a su vez las unidades son “Lux”, y que al igual que el sensor VEML-6070 se puede conectar a la Raspberry pi fácilmente y es usado para medir incidencia de luz sobre superficies. [9]

3.6.1 Descripción.

El potente VEML6075 es un chip mikroBUS que incorpora fotodiodos, amplificadores y circuitos analógico/digitales. Con este chip la luz ultravioleta proveniente el sol es convertido a valores digitales de 16 bits. UVA y UVB van en canales separados. El sensor es capaz de detectar longitud de onda de luz UV-A (315 a 400 nm) y UV-B (280 a 315 nm) y ofrece una medición digital de intensidad UV.

Para mantener una salida estable ante condiciones cambiantes de temperaturas, el chip tiene capacidades de compensación de temperatura lo que asegura un desempeño confiable en tiempos largos de exposición a los rayos UV. La placa se comunica con el micro controlador por medio de Interfaz mikroBUS I2C diseñado para una fuente de alimentación de solamente 3.3V. El sensor también viene con registros de calibración para que pueda convertir fácilmente las lecturas de UVA / UVB en el índice UV. [12]

3.6.2 Características.

- Voltaje de operación: 1.7V – 3.6V
- Corriente: 480uA
- Rango de medición UVA: 365±10nm
- Rango de medición UVB: 330±10nm
- Comunicación: I2C (dirección 0x10).
- Temperatura de trabajo mínima: - 40 C.
- Temperatura de trabajo máxima: + 85 C.
- Número de canales: 1 Channel.
- Tipo de salida: 10 KOhms Pull-up, Buffer.

3.6.3 Asignación de pines.

La Tabla 3. Muestra la asignación de pines y la Figura 8 ilustra el diagrama de pines para el sensor VEML-6075.

Tabla 4. Distribución de pines para el sensor VEML-6075.
Fuente: (mikroBUS Sensortec, 2015).

Pin. No.	Nombre	Función
1	GND	Power
2	VCC	Power
3	I2C	Serial Clock
4	I2C	Serial Data



Figura 24. Distribución de pines del sensor VEML-6075.
Fuente.[9]

3.6.4 Circuito típico de conexión.

La Figura 24 ilustra el circuito típico de conexión. El sensor se suelda en una PCB con un regulador de 3.3V, un cambiador de nivel I2C y resistencias pull-up en los pines I2C. El VEML-6075 reemplaza al VEML-6070. [11]

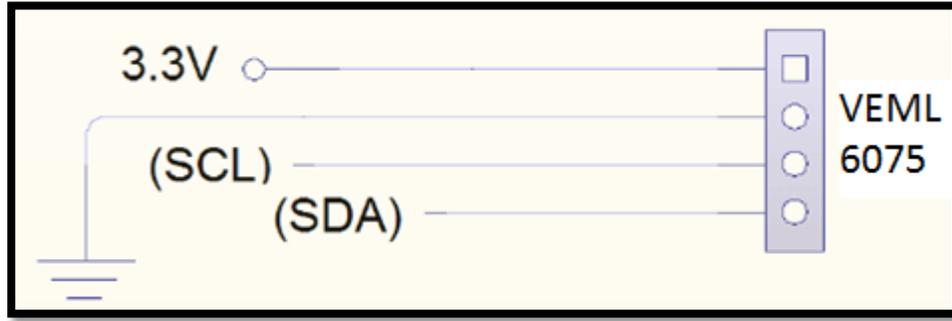


Figura 25. Sensor VEML-6075, circuito típico de conexión.
Fuente.[1]

3.6.5 Sensor VEML-6075 (Lectura)

Para la lectura correcta del sensor, el fabricante especifica el diagrama de flujo con el cual se puede realizar una adecuada adquisición de los datos. Es necesario conocer cada registro del sensor para no tener problemas en los cálculos de los datos. [9]

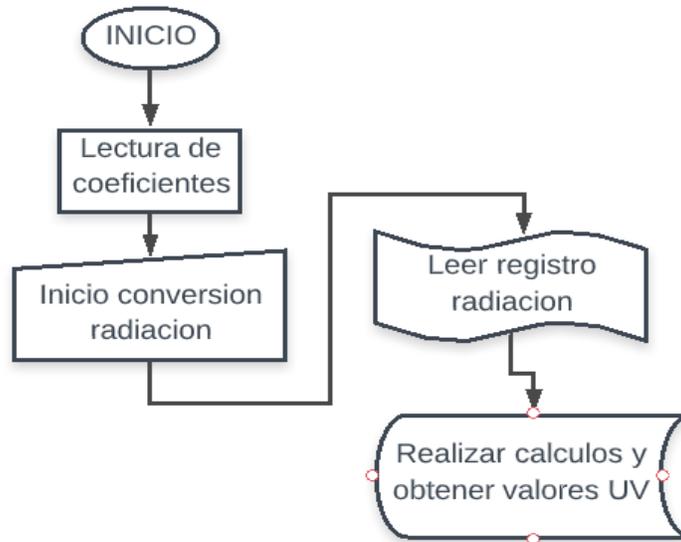


Figura 26. Diagrama de flujo lectura VEML-6075.
Fuente.[8]

En la anterior figura (Fig. 26) se puede detallar el diagrama de flujo que se implementó para la adquisición de los parámetros de radiación. En primera instancia el manejo de dicho sensor indica que la toma de datos de radiación UVA y UVB, se debe hacer a través una serie de coeficientes, con los cuales posteriormente se realiza una secuencia de calibración para así llegar a los valores correctos de UVA y UVB en un índice de radiación ultravioleta UV los cuales no se deben alterar de ninguna forma.

De acuerdo con la lectura del sensor y los índices obtenidos, es fundamental tener en cuenta la tabla Solar Global Índice UV que se puede ver en la figura (Fig.27), la cual gracias a la organización mundial de la salud (OMS) en 1995 nos alerta del nivel de radiación UV al que estamos expuestos y qué debemos hacer para evitar sufrir los efectos indeseables de esta radiación. [17], [12]



Figura 27. Tabla Niveles De Índice de Radiación Rayos UV.
Fuente.[19]

3.6.6 Conectar VEML-6075 a RASPBERRY PI 3 MODELO B.

Gracias a que el sensor UV VEML6075 es un sensor digital, podemos interactuar con él directamente desde la Raspberry Pi y no tener que depender de hardware adicional, como un convertidor analógico-digital (ADC).

La Figura 28 ilustra el diagrama esquemático para la conexión del sensor VEML-6075 a la tarjeta RASPBERRY PI. Se puede notar que no aparecen las resistencias de pull up que establece la norma para el protocolo I2C, esto es porque estas resistencias ya están habilitadas internamente en el sensor. [9]

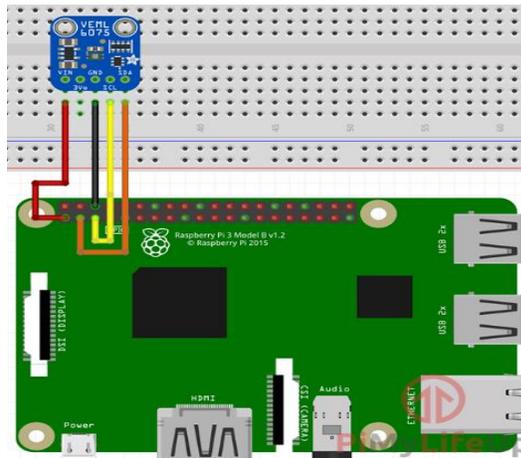


Figura 28. Sensor VEML-6075, conexión Raspberry Pi.
Fuente.[9]

- Conecte el pin VIN del sensor UV VEML6075 al pin físico 1 (3v3) en la Raspberry Pi.
- Conecte el pin GND del sensor UV VEML6075 al Pin físico 6 (GND) en la Raspberry Pi.
- Conecte el pin SDA del VEML6075 UV Sensor Pin físico 3 (SDA) en la Raspberry Pi.
- Conecte el pin SCL del sensor UV VEML6075 al Pin físico 5 (SCL) en la Raspberry Pi.

3.7 Medición de las gotas de lluvia.

El sensor YL-83 está considerado para el control de dispositivos en función de la lluvia. La placa cuenta con una superficie niquelada para resistir mejor la oxidación y alargar su vida útil.

3.7.1 Descripción.

Este dispositivo cuenta con tecnología TTL, lo que significa que proporciona una señal digital entre 0 y 1 por el pin DO. A través de la terminal AO del sensor, la cual es una salida analógica, se puede hacer la recepción de señales que irán directo al micro controlador, para así detectar la cantidad de precipitaciones que hay. La superficie de la placa sensorica se encuentra hecha en níquel plateado con propiedades antioxidantes, asegurando una buena conductividad y excelente rendimiento.

Este módulo posee un comparador LM393, el cual se encarga de asegurar que la señal de salida se encuentre limpia, para obtener una mayor calidad en el sensado de datos. Para calibrar su sensibilidad solo basta con adecuar el potenciómetro acorde a las medidas a tomar.

Cuando se polariza el dispositivo, véase la figura (Fig. 29) el módulo tiene un LED indicador, el cual, al sensar la presencia de lluvia se obtiene a la salida un nivel alto de tensión y viceversa, aunque estos niveles vienen a estar calibrados gracias al potenciómetro acorde a la necesidad del usuario. [9]

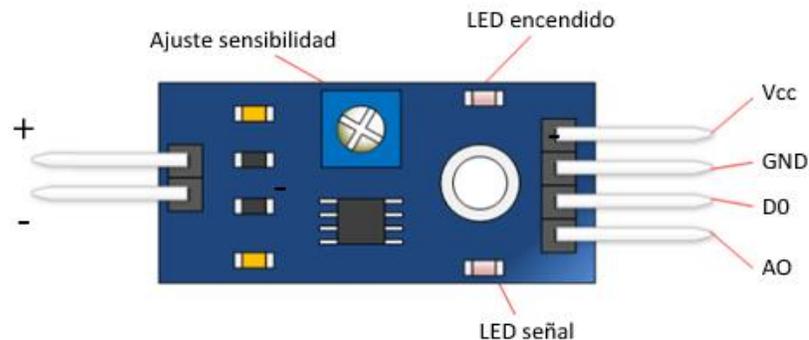


Figura 29. Distribución interna del sensor YL-83.
Fuente. [9]

3.7.2 Características.

- Voltaje de entrada: 3.3 ~ 5 V.
- Voltaje de salida: 0 ~ 4.2 V.
- Corriente de operación: 15mA.
- VCC: Tensión de alimentación.
- GND: Tierra.
- A0: Salida analógica que entrega una tensión proporcional a la humedad.
- D0: Salida digital; este módulo permite ajustar cuándo el nivel lógico en esta salida pasa de bajo a alto mediante el potenciómetro.
- Dimensiones MH-RD: 5.5 cm x 4 cm.
- Dimensiones YL-83: 3 x 1.6 cm.
- Modo de salida dual del módulo, salida digital, salida análoga más precisa.

3.7.3 Asignación de pines.

La Tabla 4. Muestra la asignación de pines y la Figura 29 ilustra el diagrama de pines para el sensor YL-83. [12]

Tabla 5. Distribución de pines para el sensor YL-83.
Fuente: (Bosch Sensortec 2014, página 4).

Pin. No.	Nombre	Función
1	AO	Analog Output
2	DO	Digital Output
3	GND	Power
4	VCC	Power

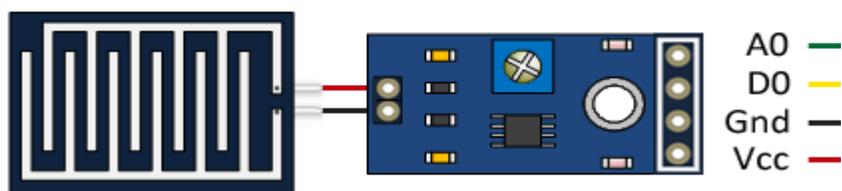


Figura 30. Distribución de pines del sensor VEML-6075.
Fuente.[9]

3.7.4 Conectar YL-83 a RASPBERRY PI 3 MODELO B.

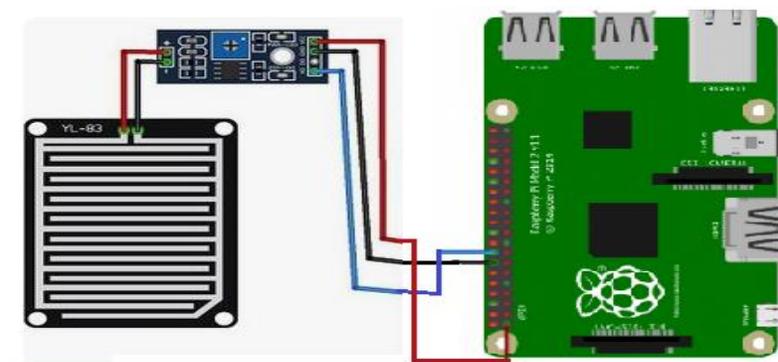


Figura 31. Sensor YL-83, conexión Raspberry Pi. Fuente.[9]

- Conecte el pin VIN del sensor YL-83 al pin físico 1 (3.3V) en la Raspberry Pi.
- Conecte el pin GND del sensor YL-83 al Pin físico 6 (GND) en la Raspberry Pi.
- Conecte el pin DO del sensor YL-83 al pin físico 7 (GPIO) en la Raspberry Pi.

3.8 Cálculos para la alimentación Solar.

Para proveer los cálculos necesarios para la alimentación de la plataforma, fue necesario investigar aspectos muy importantes al prototipo. Capacidad de carga de los sensores, panel solar y sus dimensiones, intensidad lumínica de la pantalla touch-screem, véase la figura (Fig. 32) para así determinar la potencia necesaria, los cálculos reales y las horas de funcionamiento necesarios para garantizar un buen rendimiento del prototipo y la tarjeta Raspberry PI que es la unidad de procesamiento de los datos.

3.8.1 Características del panel solar.

El panel solar utilizado en este prototipo consta de:

Potencia W_p : 5w

Voltaje: 9v

Corriente: 548 Ma

3.8.2 Horas de funcionamiento.

A partir de la ecuación 1:

$$\#paneles = \frac{E * 1,2}{Hsp * Wp}$$

despejamos **E** que hace referencia al consumo total de la instalación en watts/h por día, ecuación 2:

$$E = \# \frac{paneles * Hsp * Wp}{1,2}$$

luego con los valores obtenidos de **Wp**: valor en watts del panel solar, **Hsp**: horas de sol promedio en el lugar donde se ubicarán los paneles, y **1,2** = constante de sobre de estimación de 20%. Es un valor variable. [13]

Según el **IDEAM** y la revista **FONDO DE ADAPTACIÓN** en compañía de la **CONCESIONARIA SAN SIMÓN S.A** acogen un estudio de la población de pamplona acerca de los parámetros climáticos en la región donde el índice de radiación solar pico fue de 4,2 aproximadamente. Remplazamos los valores en la ecuación 2:

$$E = \frac{(1)(4,2)*(5)}{1,2}$$

Dando como resultado $E = 17.5Wh/día$.

Posterior hacemos el cálculo de las horas de funcionamiento (**Hf**) con la energía obtenida y la potencia del panel utilizado: ecuación 3:

$$Hf = \frac{17.5Wh / dia}{5W}$$

obteniendo $Hf = 3.5h. = 0.125días$, en la tabla 5. Podemos observar el consumo total de la plataforma.

Tabla 6. Consumo total de la plataforma meteorológica.

Descripción del Equipo	Potencia Instalada (W)	Horas de uso Diario (h)	CONSUMO DE ENERGÍA DIARIO (kWh)/día
Raspberry Pi	1,8	24	0,043
Sensor yl-83	7,5m	24	1,8
Sensor bmp-180	15 μ	24	0,0003
Sensor vcmi-6075	1,72m	24	0,042
Sensor dth11	1,5m	24	0,036
Anemómetro	550m	24	13,2
Totales	2,87		15,2

3.8.3 Banco de baterías.

Para el cálculo de las baterías según la ecuación 4:

$$I = \frac{E}{V_i}$$

reemplazamos los valores ya obtenidos para así saber la capacidad de corriente necesaria del sistema, ecuación 4:

$$I = \frac{17.5Wh/dia}{9V}$$

luego obtenemos $I = 1.94Ah/día$

Es muy importante tener en cuenta que para la capacidad de las baterías (**CB**) tenemos que tener en cuenta un factor muy importante, la profundidad de descarga de las baterías (**0.7**) ya que por medio de ella podemos saber la capacidad real con que puede contar el sistema y las pérdidas por temperatura que oscilan en un 15%. [13]

Según la ecuación 5:

$$CB = \frac{dias * I}{0,7} * perdidas,$$

reemplazamos los valores:

$$CB = \frac{0.125dias * 1.94}{0.7} * 1.15$$

obteniendo así: **CB**= 0.398 = 0.4^a

Nota: Cabe resaltar que el sistema de plataforma cuenta adicional con un cargador extra que sirve para alimentarlo en lugares donde se cuente con un suministro de alimentación de corriente en AC, esto se debe a su flexibilidad en cuanto a tamaño y peso que se puede transportar a ciertos puntos de la ciudad que cuente con acceso a internet.

3.9 Diseño de la aplicación en Angular CLI.

La realización de la plataforma meteorológica implica el desarrollo de una interfaz gráfica para visualizar los datos correspondientes a las variables físicas medidas. La programación de esta interfaz se hizo en Angular CLI, es una herramienta de trabajo para la creación de páginas web mantenido por Google, debido a sus ventajas de velocidad y compilación en una sola ruta nos facilita al momento de la creación, generación, ejecución entre otros. Su lenguaje es TypeScript, que un entorno de trabajo JavaScript de código abierto [10]. Se puede visualizar gratuitamente de la página web del fabricante en la dirección: <https://up-estacion.web.app>

3.9.1 Consideraciones.

En el diseño de la aplicación web para la plataforma meteorológica se tuvieron en cuenta las siguientes pautas:

- Contar con un ordenador en general de capacidad media para la programación web Angular. puesto que siempre estaremos en conexión con internet y nos puede servir sin ningún problema.
- Realizar las instalaciones necesarias para emplear el uso de las herramientas que nos brinda el framework web. (Google Chrome, NodeJS, TypeScript, AngularCLI)
- El editor de texto empleado para compilar nuestro programa, es de decisión personal según el usuario ya que tenemos varios de ellos que nos sirven para esta tarea (Atom, Visual Studio Code, Web Store, etc.). En este diseño se utilizó el editor Visual Studio Code véase en la Figura (Fig.33).

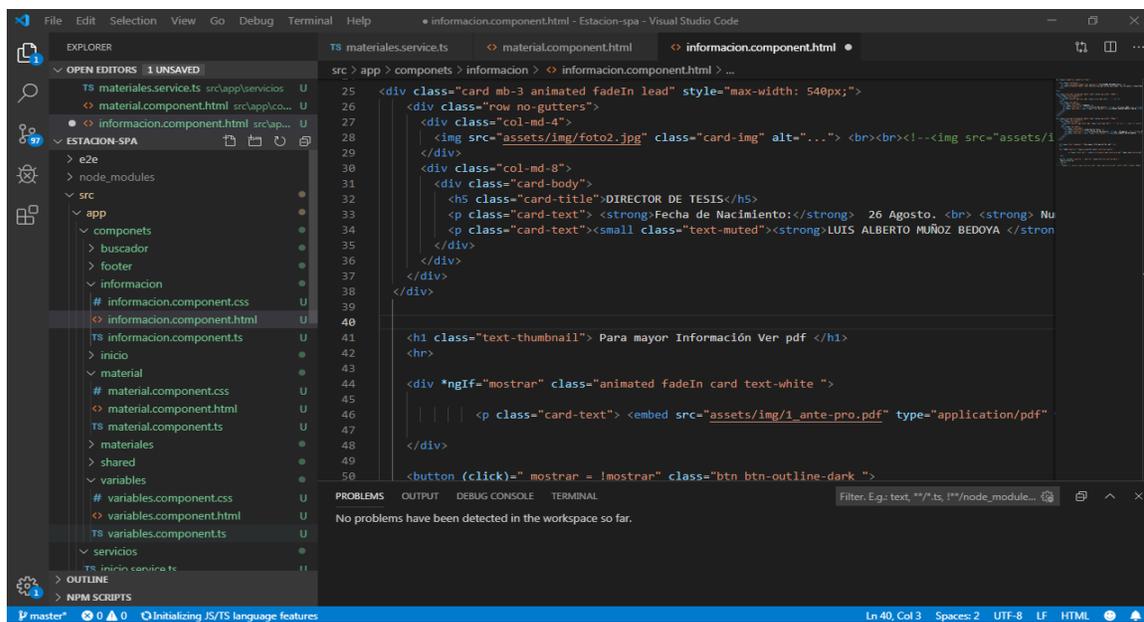


Figura 32. Editor Visual Studio Code, programación Angular CLI

- Como último paso, instalar los componentes de código (plugin) necesarios para ampliar las funciones del programa o herramienta en este caso de ámbito web, véase en la figura (Fig.34).

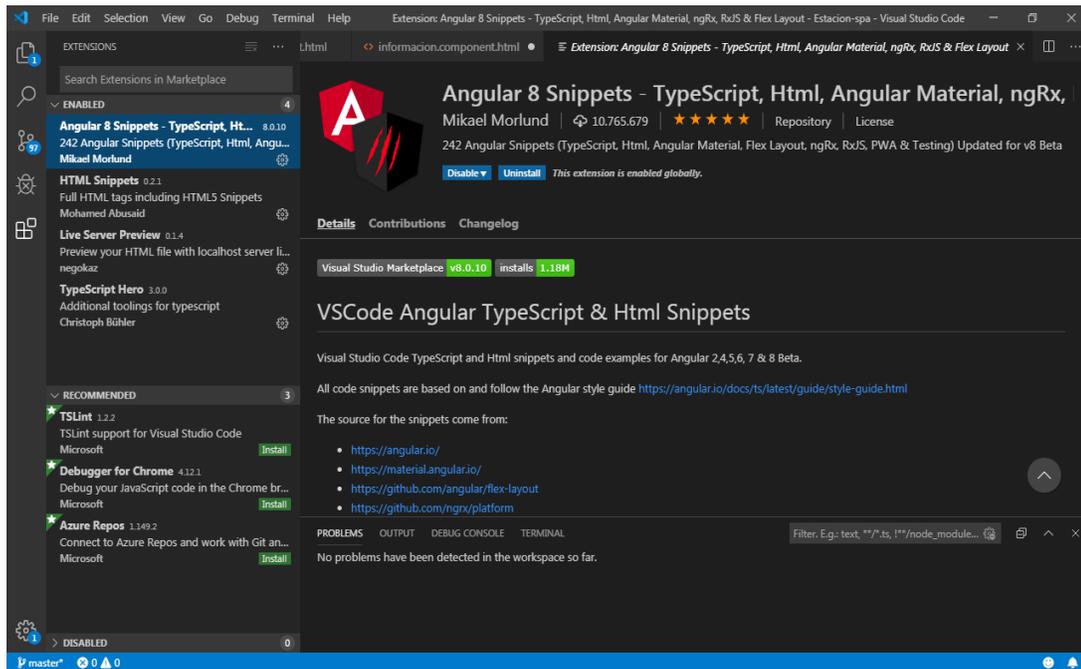


Figura 33. Editor Visual Studio Code, instalación plugins.

3.9.2 Descripción.

La Figura 35 ilustra el aspecto físico de la interfaz gráfica para la plataforma meteorológica; ésta consta, en esencia, de una barra de menú donde podremos encontrar: Inicio, Información del proyecto, materiales empleados, y las variables climatológicas medidas en unas tablas y valores en tiempo real.

Variables Climatológicas

	Nombre	Sensor	Valor / Medición
	Temperatura:	BMP-180	----
	Humedad	DTH-11	----
	Presión:	BMP-180	----
	Índice / Radiación:	VEML-6070	----
	Monitoreo / Lluvia:	Y-69	----

Figura 34. Interfaz Página Web, mediciones variables.

3.10 Entornos de programación.

En este apartado resaltaremos los entornos de programación más utilizados en el ámbito aplicativo enfocado en la plataforma meteorológica, aprovechando sus características de enlaces con dispositivos desarrolladores y su sintaxis fácil de leer con ventajas en activarse con una reacción muy cerca al tiempo real. A continuación, se especificarán los entornos de programación utilizados en este trabajo.

3.10.1. Lenguaje Python

Python es un lenguaje de programación sintetizado de alto nivel en donde se realizan códigos de programas en base de un intérprete que trabaja en conjunto con el mismo, ejecutando las sentencias en un determinado tiempo de respuesta según sea la exigencia de memoria en el Raspberry pi. Este lenguaje está inmerso en el sistema operativo Raspbian igual que su intérprete por el cual al ejecutar una sentencia de código creado este se puede mostrar por los puertos de salidos de la misma tarjeta Raspberry creando en si una aplicación interactiva de una situación real de la vida cotidiana del ser humano.

Dentro de la gran gama de decisiones que pueden llegar ejecutarse en Python se encuentra un sin número de librería que sintetizan el script aplicativo creado por el programador, reduciendo el número de líneas de códigos y dejando más fácil el análisis e interpretación del mismo para demás lectores y aficionados a esta lógica de programación. [18]

Se trata de un lenguaje de programación multiparadigma, Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional.

Hay una solución intermedia: podemos diseñar lenguajes de programación que, sin ser tan potentes y expresivos como los lenguajes naturales, eliminen buena parte de la complejidad propia de los lenguajes ensambladores y estén bien adaptados al tipo de problemas que podemos resolver con los computadores: los denominados lenguajes de programación de alto nivel. El calificativo de alto nivel señala su independencia de un ordenador concreto. Por contraposición, los códigos de máquina y los lenguajes ensambladores se denominan lenguajes de programación de bajo nivel. [18]

3.10.2. Lenguaje HTML.

Las páginas Web se localizan en sitios específicos en la red de internet llamados servidores Web. Por lo tanto, un documento WWW tiene la propiedad de enlazarse a otros documentos que se encuentran en el mismo servidor Web o en otros servidores Web, logrando la interconexión más cercana al tiempo real para diferentes aplicaciones.

El lenguaje estandarizado para la creación de páginas Web es el lenguaje HTML (HyperText Markup Language, Lenguaje de Marcas Hipertexto). HTML es un lenguaje muy sencillo que permite describir documentos hipertexto. La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado).

Por medio de páginas con programación HTML se pueden crear interface de monitoreo en donde se extrae información proveniente de un gestor de base de datos que organiza la información clasificándola y actualizándola para que la aplicación ligada a estos sistemas siempre tenga respuestas inmediatas cercanas a tiempo real. La característica más significativa de este tipo de control por página web de monitoreo es su flexibilidad de desarrollar opciones en accionamientos de control es decir insertar botones, barras de despliegue de información, hipervínculos para enlaces con más páginas web e información misma del funcionamiento de la aplicación. [18]

3.10.3. Lenguaje TypeScript.

Es un lenguaje de programación de código abierto desarrollado por Microsoft, el cual cuenta con herramientas de programación orientada a objetos, muy favorable si se tienen proyectos grandes. Anders Hejlsberg, arquitecto principal del desarrollo del lenguaje de programación C#, es el principal.

TypeScript convierte su código en Javascript común. Es llamado también Superset de Javascript, lo que significa que si el navegador está basado en Javascript, este nunca llegará a saber que el código original fue realizado con TypeScript y ejecutará el Javascript como lenguaje original. [18]

¿Qué es un superset?

Se trata de un lenguaje escrito sobre otro lenguaje. En este caso Typescript es eso, un lenguaje basado en el original, ofreciéndonos grandes beneficios como el descrito anteriormente, aunque existen otros beneficios. Por ejemplo, mientras otros Superset de JavaScript nos alejan del código original, Typescript, por el contrario, es muy similar a Javascript y a C# gracias a que su creador posee conocimientos de ambos lenguajes.

Actualmente Angular 2, uno de los Frameworks más famosos de JavaScript, está siendo desarrollado en TypeScript, para lo cual conocer este lenguaje será fundamental para entender y darle un mejor uso a la nueva versión de Angular.

3.10.4. Formato JSON

JSON es un formato de datos muy ligero basado en un subconjunto de la sintaxis de JavaScript: literales de matrices y objetos. Como usa la sintaxis JavaScript, las definiciones JSON pueden incluirse dentro de archivos JavaScript y acceder a ellas sin ningún análisis adicional como los necesarios con lenguajes basados en XML.

JSON está construido sobre dos estructuras:

- Un nombre o dato que tiene un determinado valor o estado.
- Un objeto o nodo que puede contener dentro otros nodos o datos.

En la Figura 35 se muestra un para JSON, los nodos comienzan con un "{" y terminan con un "}", mientras que un dato o nombre es seguido de ":" y el estado o valor en que se encuentra, los nodos y datos son separados por ",".[18]

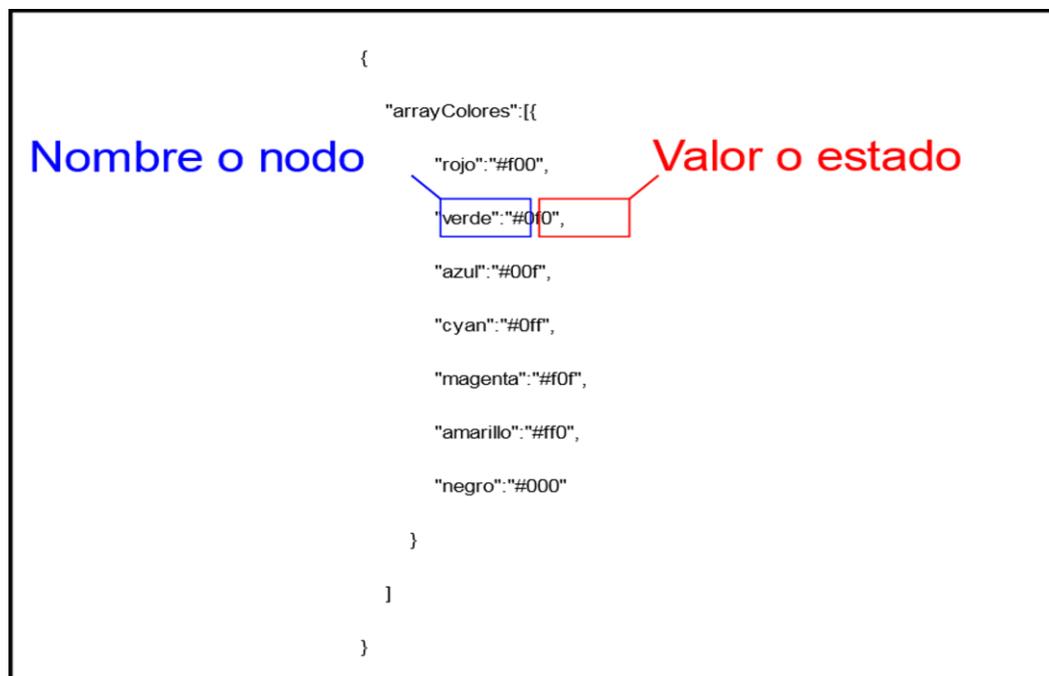


Figura 35. Sintaxis básica para un archivo JSON.
 Fuente: Implementación De Sistema De Control Remoto.
 Autor: Oscar Almanza Leones.

3.11 Firebase.

La empresa fue fundada en 2011 por ANDREW LEE Y JAMES TAMPLIN. Producto inicial de Firebase era una base de datos en tiempo real, lo que proporciona una API que permite a los desarrolladores almacenar y sincronizar los datos a través de múltiples clientes. Con el tiempo, se ha ampliado su línea de productos para convertirse en un paquete completo para el desarrollo de aplicaciones. La compañía fue adquirida por Google en octubre de 2014 y un número significativo de nuevas características se presentaron en de mayo de 2016 a Google I / O.

Firebase es una plataforma móvil que te permite desarrollar rápidamente apps de alta calidad, aumentar la base de usuarios y ganar más dinero. Firebase contiene funciones complementarias que puedes combinar y adaptar según tus necesidades. [18]

3.11.1. Herramientas de Firebase.

Firebase proporciona una solución eficaz frente no solo a problemas de desarrollo, sino también de escalabilidad a medida que la base de usuarios de la aplicación crece, ya que los servidores son proporcionados por Google. Entre sus funcionalidades se encuentra un servicio de autenticación, base de datos en tiempo real, almacenamiento de archivos,

solución de errores, funciones Backend, testeo, y medida de estadísticas recogidas de los usuarios.

3.11.2. Base de datos en tiempo real.

Firebase proporciona un gestor de base de datos (Realtime Database) NoSQL que almacena datos y los sincroniza en tiempo real. Este modelo organizacional de información ha sido utilizado para almacenar toda clase de datos, resultados que son necesario para que una aplicación trabaje en tiempo real sin perder información necesaria para luego actualizarse automáticamente. Esta base de datos utiliza un formato JSON, [18] para la gestión de los datos permitiendo una gran versatilidad al momento de generar las consultas y además mejora la velocidad para transacciones para las lecturas y actualizaciones de los nodos y estado de los datos. A continuación, se muestran los pasos usados para crear la base de datos de donde se monitorean las variables de la casa.

- **CREAR LA CUENTA:** Firebase es una herramienta de Google por lo que es necesario tener una cuenta de correo de Gmail para ingresar. En la página principal de la página esta un botón Ir a la consola, al dar clic en este botón se accede al panel de control de la cuenta.
- **CREAR EL PROYECTO:** Firebase permite crear varios proyectos lo que permite mejor organización, aquí se ingresa el nombre del proyecto, en este trabajo se le dio el nombre de up-estación, ya que se trabajara Firebase y el controlador Raspberry. En la Figura 36 y se muestra el panel de control de up-estación y la opción de agregar proyecto.



Figura 36. Panel de control de Firebase para crear un proyecto. Fuente.[5]

3.11.3. Autenticación.

La plataforma Firebase proporciona un método de registro e inicio de sesión que no solo incluye autenticación a través de correo, sino que también permite la autenticación a través de proveedores externos como Facebook, Twitter, Github y Google. Esto permite una gran facilidad al momento de desarrollar la parte de inicio de sesión de las aplicaciones mejorando el tiempo de desarrollo y despliegue del aplicativo. [18]

3.11.4. Hosting.

Firestore Hosting es un servicio de hosting de contenido web con nivel de producción orientado a programadores. Con un solo comando, puedes implementar aplicaciones web y entregar contenido dinámico y estático en una red de distribución de contenidos global rápidamente y a la vez sincronizar al actualizarse.[18]

Las principales funciones que tiene Firestore hosting son:

- Entregar contenido mediante una conexión segura
- Alojamiento de contenido dinámico y estático, además de micro servicios.
- Publicar contenido con rapidez
- Implementar versiones nuevas con un comando.
- Realizar la reversión con un clic.

3.12 Diseño de la estación.

Para el diseño de la estación meteorológica en la ciudad de Pamplona, se hizo necesario implementar el esquema de la placa de circuito impreso PCB, donde se localizaran las respectivas conexiones tanto de los sensores como del sistema embebido Raspberry Pi, también un circuito de acople para el control de voltaje de suministro y el diseño en el software Solid Edge.

3.12.1. Circuitos electrónicos de acople.

Se hizo necesario el desarrollo de un circuito electrónico de acople para el control de voltaje de suministro de alimentación de los sensores y la tarjeta Raspberry Pi, este circuito utiliza unos transistores LM 7805 y LM 318 que cumplen la función de regular el voltaje de entrada (7V-12V) proveniente del panel solar o del cargador, y así poder garantizar el correcto funcionamiento tanto de los sensores como del sistema embebido (3,3V y 5V) respectivamente. En la Figura 36 podemos asociar el circuito de acople diseñado en la aplicación de Eagle, este es un software para la fabricación y diseño de PCBs (Printed Circuit Board).

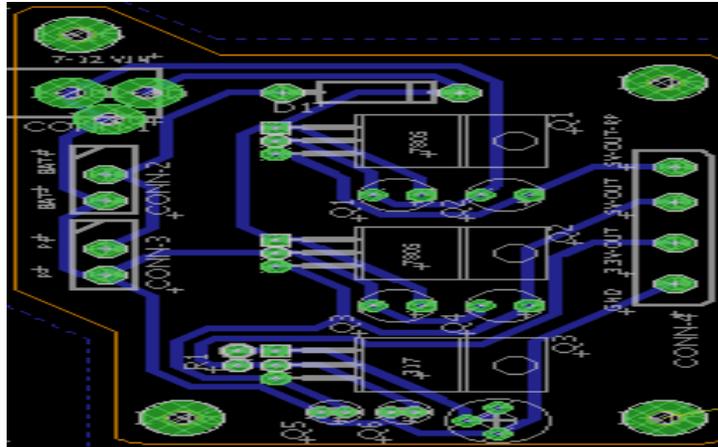


Figura 37. Circuito de acople para el control de voltaje software Eagle.

En la Figura 38 podemos observar este circuito ya montado físicamente utilizando los transistores anteriormente mencionados, fue necesario utilizar condensadores electrolíticos para proteger el circuito de alguna sobrecarga en cualquier instante que esté trabajando la estación.

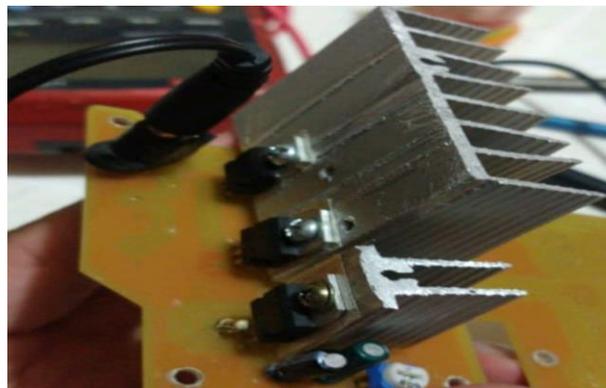


Figura 38. Circuito de acople para el control de voltaje montado.

Cabe resaltar, que la adquisición de estos transistores se dio debido a sus grandes capacidades de corriente, la cual es necesaria para el correcto funcionamiento de la Raspberry pi, y los sensores que podremos asociar al mirar el datasheet respectivamente.

3.12.2. Diseño de la PCB.

En este apartado se observa el diseño de la estación en el software Eagle, cada sensor necesita un voltaje de alimentación de 3,3v y la tarjeta Raspberry pi un voltaje de alimentación de 5v. En la Figura 38 muestra el circuito de acople mencionado anteriormente, unido con el circuito de funcionamiento de la estación.

manejo, para que a la hora de montar no tener inconvenientes de diseño y estructura. En la Figura 40 apreciaremos algunas imágenes de este diseño el cual se obtuvo a gusto personal.

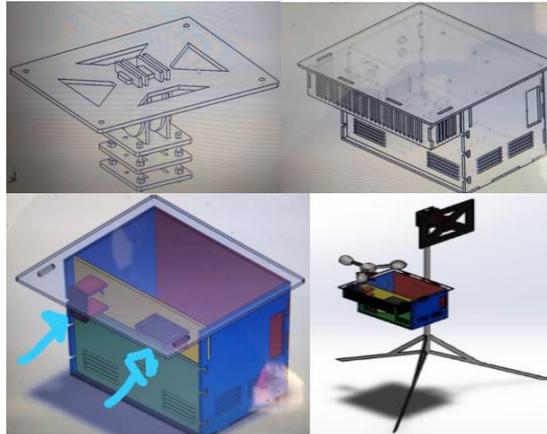


Figura 41. Diseño de la Estación en el software Solid Edge.

3.13 Programación del sistema embebido.

El sistema embebido utilizado para controlar los sensores, convertidor ADC y amplificador fue la Raspberry pi modelo B+, esta tarjeta cuenta con un procesador ARM Cortex-A53 con 1GB de memoria RAM, cuenta con puerto RJ45 para una conexión LAN a una red ethernet. El sistema operativo que se instaló en esta tarjeta fue el Raspbian en su versión 2018-03-20, Raspbian es el sistema operativo oficial para las tarjetas Raspberry, el sistema operativo trae preinstalado con Python.

3.13.1 Script de control en Python.

El algoritmo o script encargado de leer y accionar los sensores y actuadores de la casa se desarrolló utilizando el lenguaje Python, este lenguaje es de alto nivel y de propósito general para programación caracterizándose por la facilidad de legibilidad, es un lenguaje orientado a objetos haciendo excelente para el desarrollo de proyectos a pequeña y gran escala. El SDK de Firebase se puede implementar utilizando Python lo que llevo a la selección de este como lenguaje para el control de la electrónica en la estación. [18]

El código en Python desarrollado se ejecuta al iniciar la Raspberry en forma de servicio lo que asegura que a la falta energía para la alimentación de la tarjeta o después de un reinicio la lectura y control de la electrónica funcione de nuevo, internamente el servicio lee y escribe en los pines GPIO de la tarjeta estipulados permitiendo el control de la electrónica asociada constantemente debido a un ciclo que se repite, con el fin de detectar los cambios en la base de y actualizar las salidas en los pines o viceversa, leer las entrada de los pines que cambiaron para actualizar el estado en la base de datos.

En la Tabla 7 se muestra los comandos utilizados para instalar las herramientas necesarias de Firebase SDK y conectarse a la base de datos, estos comandos se ejecutaron desde el CMD de las de Raspbian en la Raspberry.

Tabla 7. Comandos CMD para instalar SDK Firebase para Python

Comando	Descripción
\$ sudo wget https://bootstrap.pypa.io/get-pip.py \$sudo python get-pip.py	Instalar PIP, para la administración de paquetes en Python.
\$sudo pip install Firebase-admin	Instalar el paquete de Firebase para Python.
\$sudo apt install Python-gpiozero	Instalar paquete para control de los pines GPIO.
\$sudo nans test fire.py	Testear la instalación de Firebase

3.13.2. Librerías en Python.

En la Figura 41 se muestra las importaciones utilizadas para el código de control en Python, la librería firebase admin es la que carga todas las herramientas de Firebase SDK para la conexión con la base de datos, RPi.GPIO se utilizó para manejar los pines digitales de entrada y salida de la tarjeta Raspberry que controlan la electrónica.

```
# libreria y dependias de firebase
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db

# dependencias para el sensor de lluvia
import RPi.GPIO as gpio
```

Figura 42. Librerías importadas en el código Python

3.13.3. Credenciales de Firebase para Python.

Las credenciales a igual que en la aplicación web necesitan ser cargadas en el objeto Firebase para que permita la conexión con la base de datos. En el script Python la variable cred importa las credenciales que se encuentran en un archivo externo en la ruta donde se encuentra el script, la carga de las credenciales se muestra en la Figura 42. Donde la variable cred carga las credenciales del archivo externo donde se guardaron, luego se inició el objeto Firebase fire_app pasando la variable cred y el URL de la base de datos. [6]

```

# credenciales para permitir la conexion con la base de datos en firebase
cred = credentials.Certificate("up-estacion.json")
# inicio de la conexion con firebase
fire_app = firebase_admin.initialize_app(cred, {
    'databaseURL':'https://up-estacion.firebaseio.com/estacion'
})

```

Figura 43. Cargar las credenciales e iniciar el objeto Firebase en Python.

3.13.4. Configuración de los pines GPIO.

Los pines GPIO de la tarjeta se configuraron como se muestra en la Figura 43, el pin GPIO-5 se configuro como salida y controla el pulso de respuesta del sensor de lluvia, el pin GPIO-10 funciona también como salida y controla el pulso de respuesta del sensor de humedad, a diferencia de los otros 2 sensores (VEML-6075 y BMP-180) funcionan a través de la comunicación I2C los cuales debemos crear el objetos para activar los pines I2C de la tarjeta Raspberry y garantizar los pulsos de comunicación de ambos sensores. [6]

```

# objeto para leer los valores del sensor BMP180
sensor_bmp180 = BMP085.BMP085()

# pobjeto para el sensor VEML6075
i2c = busio.I2C(board.SCL, board.SDA)
veml = adafruit_veml6075.VEML6075(i2c, integration_time=100)

# objeto y pin para el sensor DHT11
DHT_SENSOR = Adafruit_DHT.DHT11
DHT_PIN = 10

# Configuracion de los pines para el sensor de lluvia
gpio.setmode(gpio.BCM)
Pin_SensorLluvia = 5
gpio.setup(Pin_SensorLluvia, gpio.IN)

```

Figura 44. Configuración para los pines GPIO utilizados.

3.13.5. Control de eventos para base de datos en la Raspberry.

Los eventos detectados en la base de datos y la tarjeta son manejados por un bucle repetitivo que evalúa constantemente los cambios en las entradas GPIO de la tarjeta para actualizar su valor en la base de datos, a su vez también pregunta por los cambios que ocurren en la base de datos y que afectan el estado de las salidas GPIO de la tarjeta. En la Figura 44, se muestra el código interno del bucle repetitivo encargado de hacer lo anteriormente mencionado. [6]

```

while True:
    # Lectura del sensor de humedad
    humidity, _temperature = Adafruit_DHT.read(DHT_SENSOR, DHT_PIN)
    if humidity is None:
        humidity= 0.0
    if _temperature is None:
        _temperature=0.0

    # Lectura del sensor de lluvia
    if(gpio.input(Pin_SensorLluvia) == 0):
        lluvia = 'Rain'
    else:
        lluvia = 'No Rain'

    #Lectura del ADC para el anenometro
    viento = read_adc(0)
    # Fecha y hora de la toma de datos
    fecha = strftime("%d/%m/%y")
    hora = strftime("%H:%M:%S")

    ref = db.reference('estacion/medidas')
    ref.push({
        'date': fecha,
        'hour': hora,
        'sensors':{
            'altitude': '{0:0.2f}'.format(sensor_bmp180.read_altitude()),
            'humidity': '{0:0.2f}'.format(humidity),
            'temp': '{0:0.2f}'.format(_temperature),
            'pressure': '{0:0.2f}'.format(sensor_bmp180.read_pressure()),
            'rain': lluvia,
            'temperature': '{0:0.2f}'.format(sensor_bmp180.read_temperature()),
            'uv': '{0:0.2f}'.format(veml.uv_index),
            'wind': str(viento)
        }
    });
    print('...')
    sleep(1)

firebase_admin.delete_app(firebase_admin.get_app())
print ('Ok !')
```

Figura 45. Script para el bucle repetitivo controla los eventos de cambio.

Después de haber importado las librerías correspondientes a cada sensor, las credenciales de mi base de datos Firebase y la configuración de los pines GPIO del sistema embebido. El ciclo repetitivo o bucle se encargará de leer los datos suministrados por cada sensor los cuales serán exportados a la base de datos en tiempo real el cual trabajara con unos bakens u oyentes que estarán pendiente a cualquier cambio de evento durante la ejecución de este ciclo, posterior a esto estos datos quedaran almacenados en la base por un registro de hora y fecha para hacer más fácil su búsqueda y así poder importarlos a la aplicación Angular CLI.

4. RESULTADOS Y ANÁLISIS

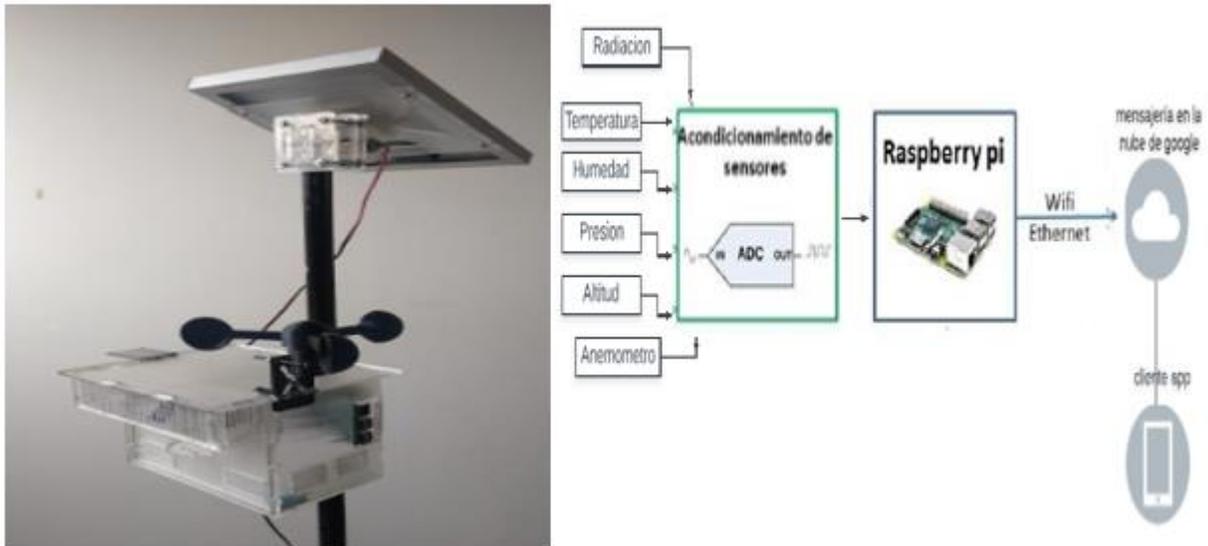


Figura 46. Prototipo de estación meteorológica. Fuente.[1],[9],[Autor]

Se logró diseñar y desarrollar un prototipo de estación meteorológica que permite sensar la temperatura ambiente, la humedad relativa, la presión atmosférica, la velocidad del viento, la altitud y la radiación solar en la ciudad de Pamplona, y enviar la información de las variables medidas a un dispositivo móvil u ordenador, a través de comunicación Ethernet.

La arquitectura general de la comunicación en el proyecto es la tarjeta Raspberry PI, la cual cuenta con un scripts (parámetros de configuración), encargado de conectarse por medio de internet, toma los datos provenientes del acondicionamiento de los sensores y sus respectivas lecturas, y posterior a esto se encarga de montarlos junto con la nube (Firebase). En la nube se encuentra una aplicación alojada la cual contara con un dominio (dirección url), que nos servirá como ruta para que el cliente pueda acceder a los datos que están en la nube, se realiza una comunicación en tiempo real (Real Time Database) que cada vez que un valor cambie en la base de datos, las aplicaciones que están puestas en marcha cuentan con unos scripts por detrás (bakers), encargados de generar un oyente que detectan los cambios en la base.

4.1 Funcionamiento.

La aplicación creada en Angular CLI se encuentra alojada en la firebase, a la cual tendrá acceso el cliente que tengo su dominio o las credenciales correctas de conexión (link), al momento de poner en marcha la tarjeta raspberry PI y ejecutar los scripts correspondientes, ella

automáticamente empieza a leer los datos provenientes de los sensores y si tiene conexión a internet los almacena en la nube (firebase). Véase en la Figura 47.

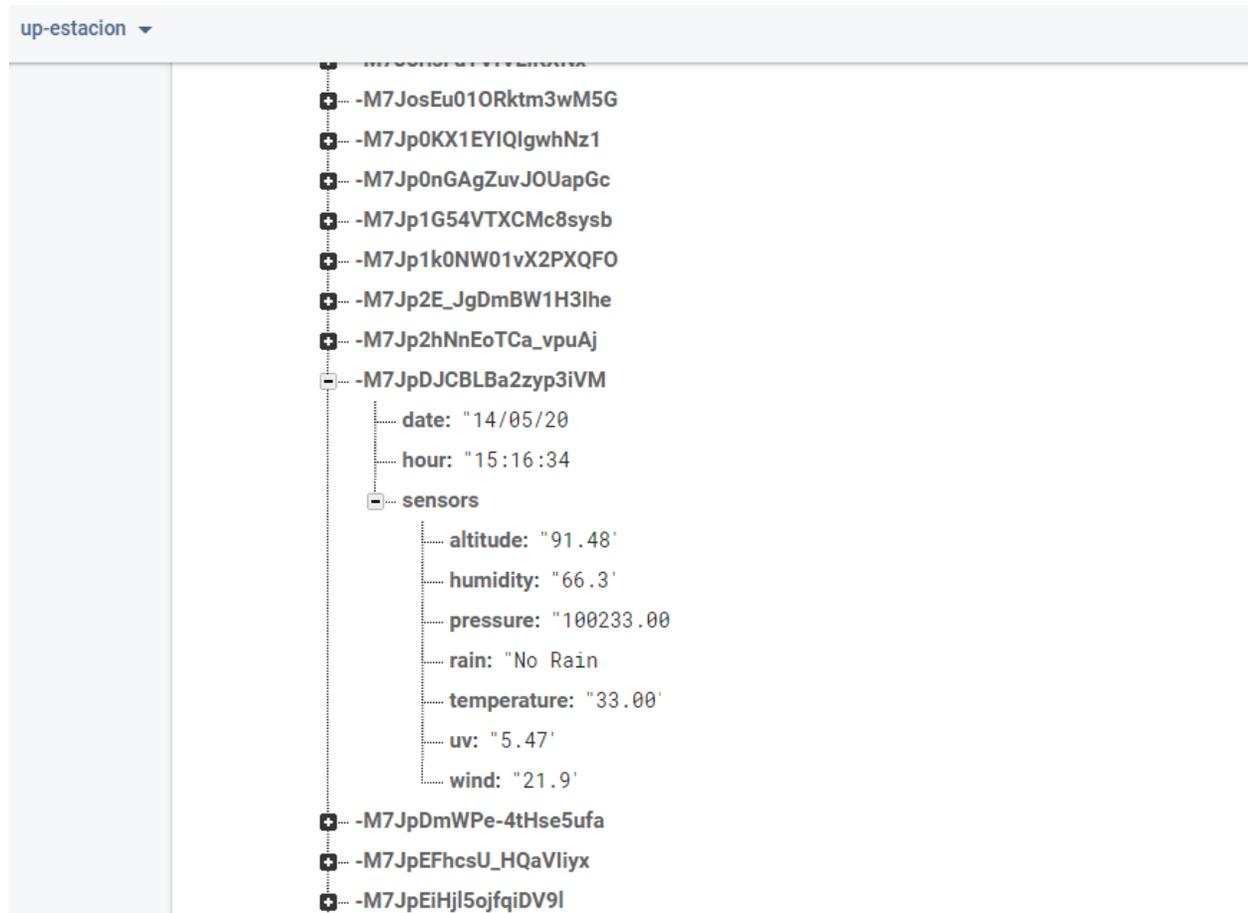


Figura 47. Almacenamiento de datos en la nube Firebase.

Cuando el cliente o usuario ingresa con el link de la aplicación y la ejecuto desde el navegador, es como si la aplicación estuviera en marcha en mi ordenador, al igual que en la tarjeta PI por detrás tiene unos scripts en java los cuales tienen unos parámetros que le permite a la aplicación conectarse (escribir o leer datos), de la base de datos que está en la nube con conexión a internet, automáticamente cuando ocurre un evento es dinámicamente creado y pasado secuencialmente a las "escuchas" (oyente), para los cambios que se presentan en la base me ejecute la función (etiqueta) que está corriendo en mi ordenador y posterior a esto sean actualizados en ese momento.

4.2 Fiabilidad de los datos.

En cuanto a la fiabilidad de los datos entregados por la estación, se consideran seguros los que se obtienen de sensores precisos, debido a que estos emplean sus propias rutinas de calibración a través de la programación en alto nivel. No obstante, la medición de las variables climatológicas fueron comparadas con los valores arrojados por la página **AccuWeather**, **Clima.com**, **Meteocast.net** y **Airport Weather Observations, Hourly** . Es una compañía estadounidense que presta servicios comerciales de pronóstico del tiempo en todo el mundo. En la Figura 48 se puede notar el almacenamiento de datos en la base de Firebase, cabe resaltar que la toma de datos es proporcional al tiempo de transmisión de los sensores.

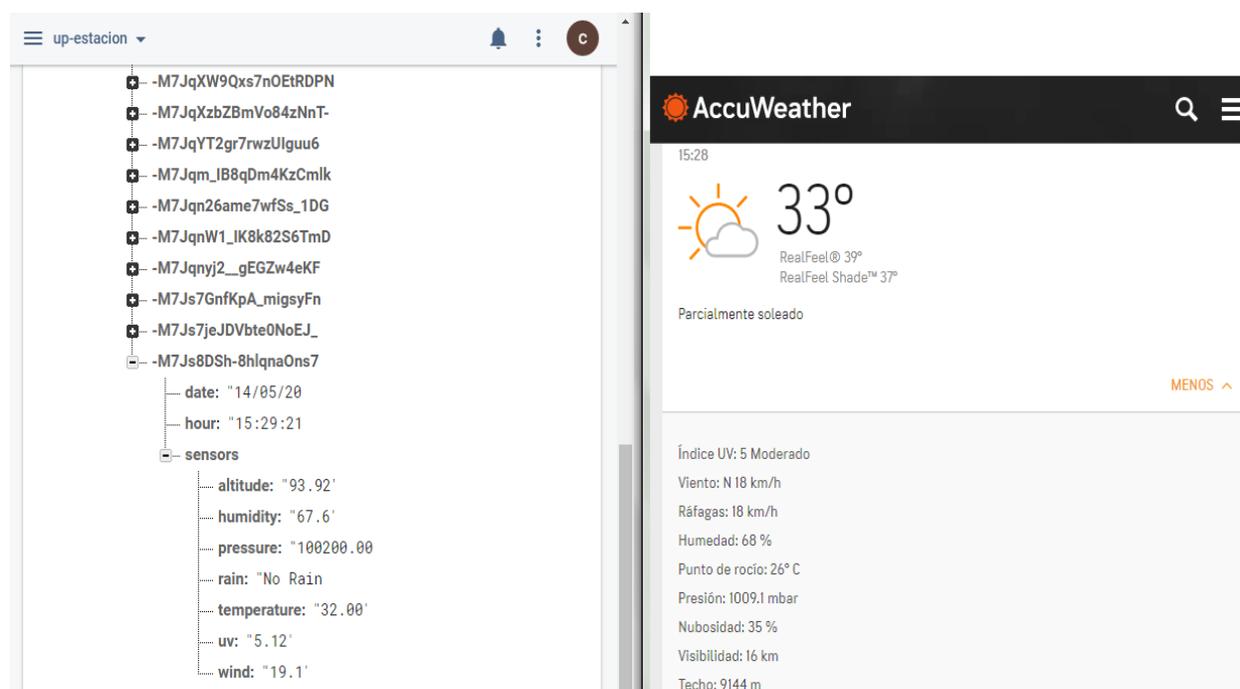


Figura 48. Comparación de las variables en tiempo real.

Un aspecto muy importante para tener en cuenta al momento de ejecutar la base de datos en la nube, es que las estructuras que manejan estas bases no son relacionales sino estructuras nodales (por nodo), son conocidos como archivos .Jason. En la Figura 48 se puede observar el comportamiento de la base de datos como me genera un nodo principal que es Estación seguido de otro llamado Medidas posterior a ellos se pueden observar unos nodos con una identificación extraña de objeto, ese es un ID que se crea automáticamente ya que es un herramienta que me brinda la nube (Firebase) para identificar el dato, para que no se repita lo cual me va a indicar que dentro de cada medición de esas va a haber una fecha y una hora y dentro de ellas van a ver unos sensores que van a medir los parámetros leídos por los sensores (Humedad, Temperatura, Presión...)

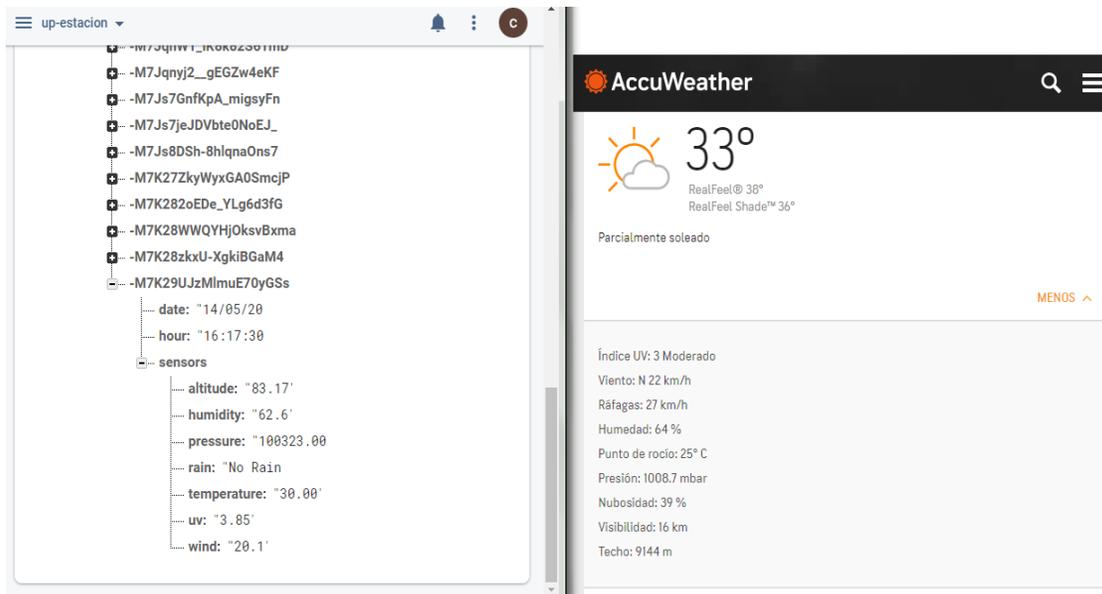


Figura 49. Comparación de las variables 50 minutos después.

En la Figura 49 se puede observar 50 minutos después los datos arrojados por la estación, cada variable concuerda con las de la página de comparación algunos tienen un margen de error de 2%, esto se da debido a la precisión de los sensores empleados de la estación meteorológica propuesta con la de la página, pero a decir verdad no es mucha la diferencia.

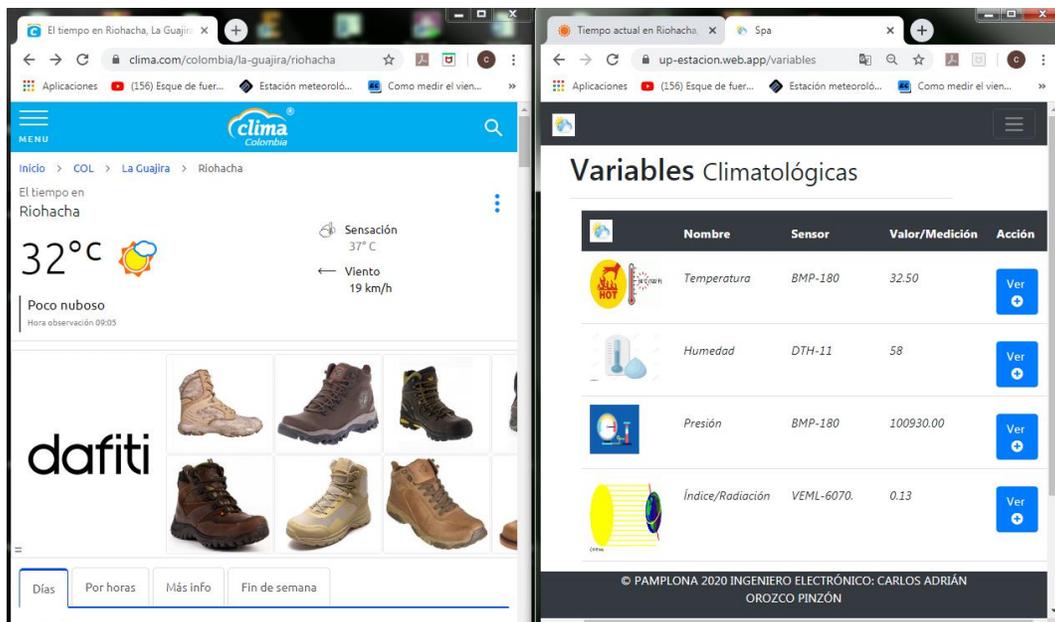


Figura 50. Comparación de pruebas con Clima.com . Fuente [20], [Autor]

En la figura 50 muestra una comparativa de la aplicación propuesta en este proyecto con una compañía de estudios, medio de comunicación online especializado en meteorología dirigido a usuarios de habla hispana en América. Se trata de una nueva marca perteneciente al grupo Pelmorex (The Weather Network, MétéoMédia, Eltiempo.es y Wetter Plus). [20]

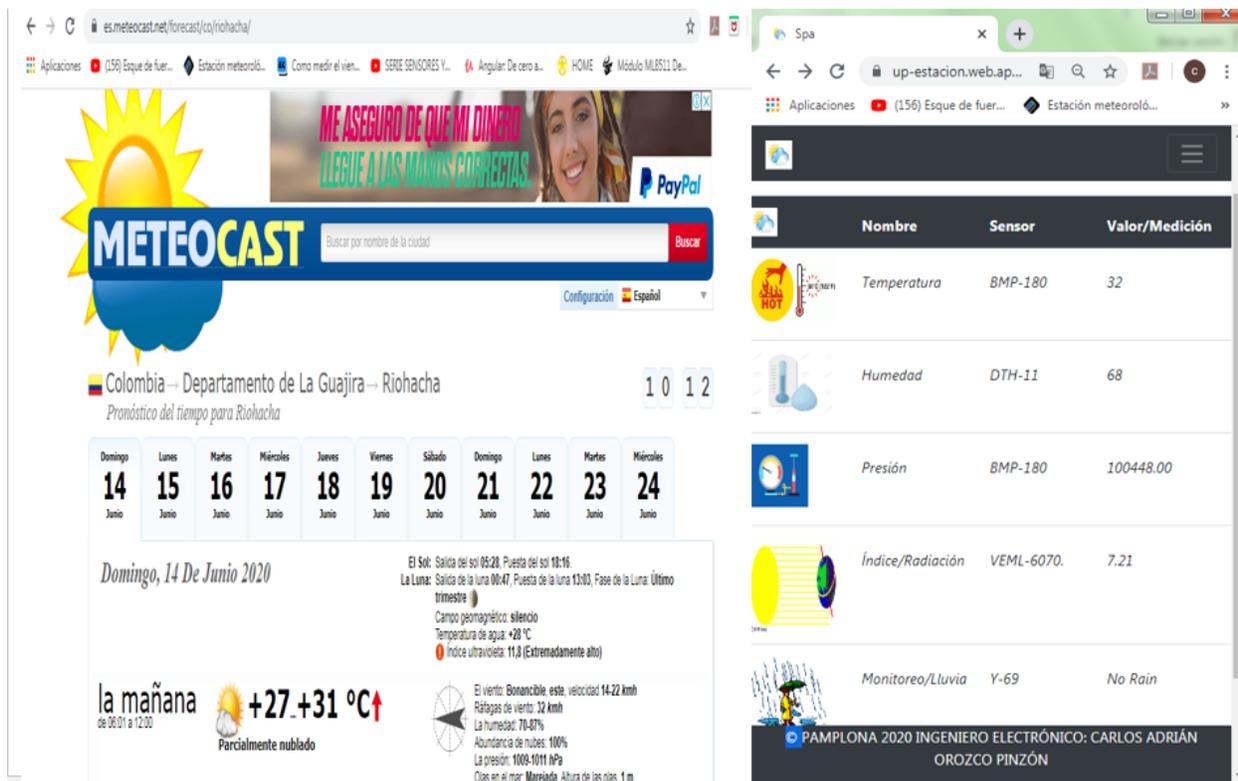


Figura 51. Comparación de pruebas con es.meteocast.net .Fuente [21],[Autor]

La compañía es.meteocast.net ofrece la gama más completa de servicios meteorológicos donde en usuario puede obtener información hacer de los factores climáticos a nivel mundial por medio de la interface de comunicación Ethernet y Wifi, esta en constante crecimiento en el ámbito tecnológico [21], su amplio servicio se debe a que trabaja en conjunto con la compañía **fdstar.com** encargada de proveer cada información del proyecto y elaboración del mismo a través del Internet global.

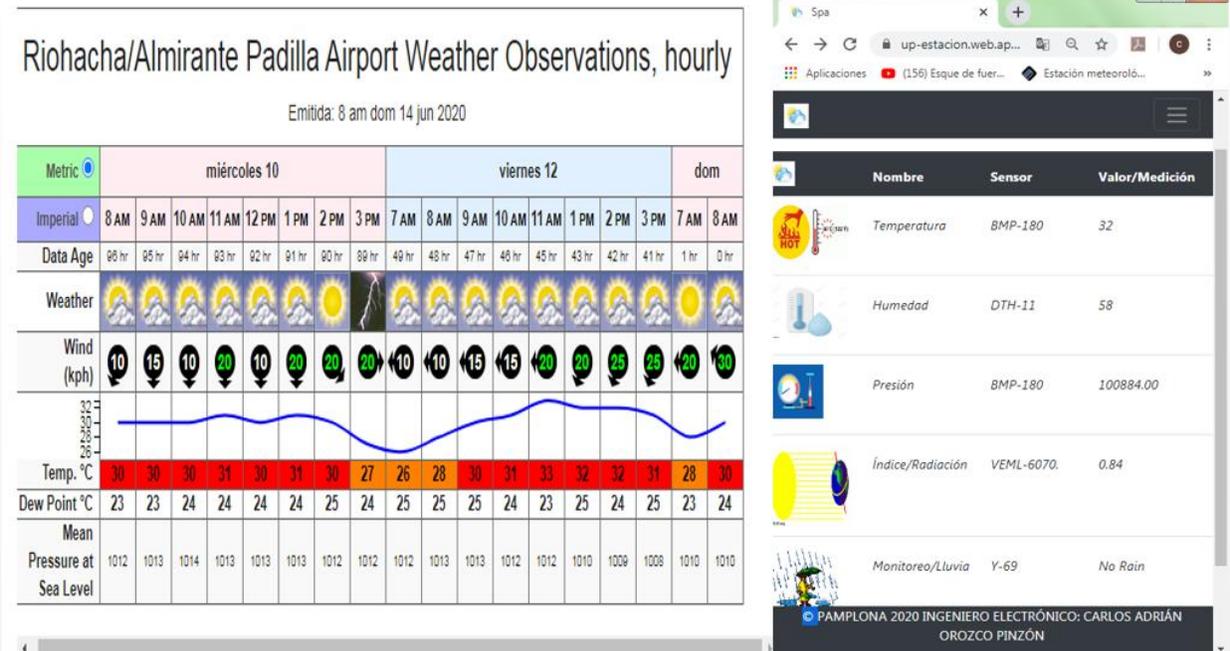


Figura 52. Comparación-Obs. Meteorológicas Aeropuerto Almirante Padilla –Riohacha. Fuente: [22],[Autor]

El informe del tiempo de **Riohacha/Almirante Padilla Airport** proporciona observaciones de la estación meteorológica actualizados cada hora. La tábula enseña la velocidad del viento, el punto de rocío, la visibilidad y la presión atmosférica con también un gráfico de temperatura detallado. Una anotación de variaciones del tiempo de los 18 últimos días también enseña observaciones durante el día y a noche. La estación meteorológica es cerca de Riohacha, Colombia como se observa en la Figura 22.

4.3 Entorno Angular CLI.

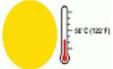
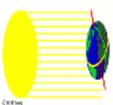
```

src > environments > TS environment.ts
1 // This file can be replaced during build by using the `fileReplacements` array.
2 // `ng build --prod` replaces `environment.ts` with `environment.prod.ts`.
3 // The list of file replacements can be found in `angular.json`.
4
5 export const environment = {
6   production: false,
7   firebaseConfig: {
8     apiKey: "AIzaSyCVZExhE07V0ESHcCbKwJhXcK61ZsVS4",
9     authDomain: "up-estacion.firebaseio.com",
10    databaseURL: "https://up-estacion.firebaseio.com",
11    projectId: "up-estacion",
12    storageBucket: "up-estacion.appspot.com",
13    messagingSenderId: "761765975530",
14    appId: "1:761765975530:web:3e2e6537cc74f6b99e0e93",
15    measurementId: "G-JC1RE4699W"
16  }
17 };
  
```

Figura 53. Credenciales Firebase para la aplicación Angular CLI.

Como se observa en la Figura 50, el entorno utilizado para editar la programación de la aplicación para el presente proyecto es Visual Studio Code. Es un editor de textos para programar en

diferentes lenguajes de programación (Python, C++, JavaScript, Typescript....), escogido por gusto personal su fácil manejo y por ser un software libre, también se pueden observar las credenciales para alojar la aplicación en la base de datos de Firebase en el archivo (Enviroments), lo cual nos garantizara que las lecturas obtenidas por los sensores se verán reflejadas en la aplicación como lo muestra la siguiente Figura 51.

	Nombre	Sensor	Valor/Medición	Acción
	Temperatura	BMP-180	28.60	Ver +
	Humedad	DTH-11	60.6	Ver +
	Presión	BMP-180	100790.00	Ver +
	Índice/Radiación	VEML-6070.	0.20	Ver +
	Monitoreo/Lluvia	Y-69	No Rain	Ver +
	Velocidad/Viento	ANEMOMETRO	19.1	Ver +

© PAMPLONA 2021 INGENIERO ELECTRÓNICO: CARLOS ADRIÁN OROZCO PINZÓN

Figura 54. Datos reflejados en la aplicación Angular.

Como se alcanza a ilustrar en la Figura 51, cada variable es alojada en una fila con su respectivo sensor, a la hora de la toma de datos cada uno de ellos reflejara el último valor durante la puesta en marcha de la estación meteorológica. Cabe resaltar que cada variable tiene su interfaz gráfica (véase en la Figura 52), puede ser observada por el cliente que quiera inspeccionar el comportamiento con más detalles al hacer (clic) en el botón “Ver” que se encuentra en la parte derecha.

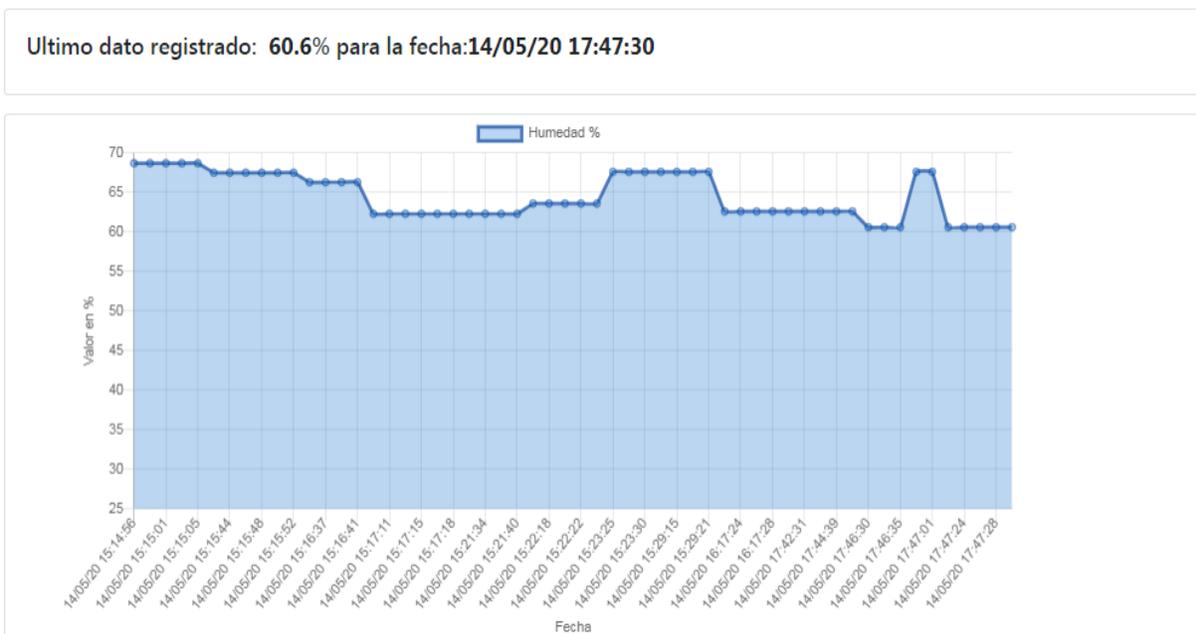


Figura 5. Ejemplo interfaz gráfica de humedad.

Aquí en esta imagen se aprecia el último dato registrado durante el día 14 del mes de mayo del año 2020 en la ciudad de Riohacha la guajira, fueron tomados una serie de datos durante el transcurso del día en 3 intervalos de tiempo, (mañana, medio día y tarde) donde la gráfica es de estilo lineal, y se le facilita al usuario ver por fechas y por horas el estado de cada variable a inspeccionar.

4.4 Análisis económico

La Tabla 7 es una lista general de los materiales que conforman la plataforma meteorológica realizada y el costo que éstos representan. En el mercado se encontraron estaciones inalámbricas robustas de características similares. Con base a los precios consultados para diferentes estaciones comerciales se estimó que adquirir una de las mismas representa un costo mínimo de aproximadamente 1.400.000 COP, una cantidad significativamente mayor a la descrita en la Tabla 7. Un ejemplo de una estación meteorológica comercial es la **Estación Meteorológica Inalámbrica Pro2, Davis Instruments 6152 cuyo precio oscila alrededor de 1.400.000 COP.**

Tabla 8. Costo de materiales.

Costo de materiales	
Rubro	Precio (COP)
Tarjeta Raspberry PI 3 modelo B	\$ 189000
Sensor de lluvia YL-83	\$ 7000
Sensor BMP180	\$ 18000
Sensor DHT11	\$ 8000
Sensor VEML6075	\$ 25000
Anemómetro	\$ 50000
Regulador de carga solar	\$ 29000
Panel solar 5W	\$ 45000
Baterías Recargables	\$ 20000
Acrílico diseño portable	\$ 40000
Base portable	\$ 25000
Modulo conversor AD	\$ 14000
Modulo Amplificador	\$ 4000
Otros	\$ 60000
Total	\$ 534000

Nota: dentro de **Otros** se incluyen diferentes componentes pasivos y activos de reducido costo. Así como cables, placa de circuito impreso, entre otros componentes y herramientas necesarias para la construcción del prototipo.

5. CONCLUSIONES

- Se realizó satisfactoriamente la plataforma meteorológica en la ciudad de Pamplona, que puede registrar diferentes variables de medición; la temperatura ambiente, la humedad relativa y la presión atmosférica, la altitud, presencia de lluvia (la medición de la velocidad del viento no se pudo comparar con un instrumento calibrado y por tanto no se garantiza la fiabilidad de los datos para estas variables). Que ofrece exportar los datos procesados de las mediciones realizadas a las variables físicas antes mencionadas a través de internet a una base de datos en tiempo real (plataforma Firebase). En el equipo terminal (móvil u ordenador) se visualizan en tiempo real, para ello se logró implementar una interfaz gráfica hecha en Angular CLI en la que se registran los datos recibidos de la estación facilitando un análisis futuro.
- La plataforma meteorológica realizada representa un costo considerablemente menor respecto a una estación meteorológica comercial. Además, es un primer paso para el diseño y realización de una estación meteorológica local que cumpla con los estándares comerciales.

- El uso del sistema embebido Raspberry PI configurado de manera correcta y el adecuado acondicionamiento de cada uno de los sensores para las lecturas climáticas, facilitaron el diseño y desarrollo de la plataforma meteorológica. Además, resulta ser competitiva en comparación con aquellas que están en el mercado permitiendo así la mejora de la misma hacia un futuro.
- En resumen, la estación cumple con los objetivos fijados en el proyecto.. Cuenta con una interfaz de comunicación Ethernet y Wifi ideal para tecnologías de vanguardia en la actualidad, al disponer de herramientas web hace que se puede acceder con cualquier dispositivo que disponga de conexión a internet independientemente de la red que se utilice. La interfaz resulta funcional, dispone de un sencillo menú brindándole al usuario la capacidad de entender, comprender, analizar y monitorear los factores climáticos obtenidos y poder navegar por el servidor.

6. RECOMENDACIONES

6.1 Técnicas

- El periodo de tiempo para el envío de las variables medidas de la plataforma es en tiempo real se recomienda una ubicación al aire libre donde la radiación del sol pueda suministrar la alimentación energética necesaria para tener así un óptimo rendimiento y una excelente respuesta.
- El área donde se encuentre operando la estación debe tener acceso a internet mientras se haga uso del envío de datos inalámbricamente.

6.2 Para trabajos futuros

- Para posteriores mejoras se puede aprovechar el hardware disponible de la tarjeta para la ampliación del prototipo. Se podría, por ejemplo, incorporar una pantalla LCD para la vista de las variables medidas, ya que son en tiempo real y aprovechar su potencia para que el usuario no necesariamente tenga que acceder a la aplicación mediante internet sino también de forma local.
- Realizar la calibración del anemómetro reemplazarlo por modelo comercial más robusto previamente calibrado.
- Por cuestiones de seguridad se debería localizar en un lugar fuera del alcance de los niños para que no tenga ningún riesgo de funcionamiento.

- Se debe tener cuidado en la alimentación de la tarjeta ya que a veces la corriente genera altos picos, lo que puede bajar la tensión en la etapa de alimentación y si la tensión se sale del rango permitido la tarjeta puede funcionar de manera errónea y tiende a apagarse.

Glosario

Altitud: distancia vertical entre un objeto o punto determinado en el espacio y la superficie del nivel del mar, la terrestre u otro punto tomado como referencia **Anemómetro:** instrumento que mide la velocidad del viento.

Clima: es el conjunto fluctuante de las condiciones atmosféricas, caracterizado por los estados y evoluciones del estado del tiempo, durante un periodo de tiempo y un lugar o región dados, y controlado por los denominados factores forzantes, factores determinantes y por la interacción entre los diferentes componentes del denominado sistema climático (atmósfera, hidrosfera, litosfera, criósfera, biosfera y antropósfera).

Humedad relativa: relación entre la cantidad de vapor de agua que tiene una masa de aire y la máxima que podría tener.

Presión atmosférica: presión que ejerce la atmósfera sobre la superficie de la Tierra.

Temperatura ambiente: temperatura ordinaria en torno a un cuerpo.

Lluvia: es un fenómeno meteorológico que consiste en caer el agua en forma de precipitación líquida desde las nubes, formadas por condensación del vapor de agua, que al cobrar tamaño y peso no pueden mantenerse suspendidas en el aire.

Radiación Solar: es el conjunto de radiaciones electromagnéticas emitidas por el Sol.

Viento: el viento es el aire en movimiento, el cual se produce en dirección horizontal, a lo largo de la superficie terrestre.

Referencias

- [1] Carlini's Blog. Sensores de humedad y temperatura DHT11 y DTH22 en la Raspberry Pi. < <http://carlini.es/sensores-de-humedad-y-temperatura-dht11-y-dth22-en-la-raspberry-pi/> > [Consulta: 30/04/2019]
- [2] Oregon Scientific. Tienda online oficial. <https://openwebinars.net/blog/angular/> [Consulta: 03/05/2019].
- [3] http://www.inet.edu.ar/wp-content/uploads/2012/11/C9Estacion_meteorologicaR.pdf >[Consulta: 01/05/2019].
- [4] E. PEGASUS, «Reporte de estación a 24 de abril de 2015,» Universidad Distrital, Bogotá, 2015.
- [5] http://firebase.google.com/EAlalQobChMI5KvDpZGI4gIVGCh31fQcjEAAYASAAEgls8vD_BwE [Consulta: 03/05/2019].
- [6] <https://repository.usta.edu.co/bitstream/handle/11634/10439/RafaelFerrer-2017.pdf?sequence=1&isAllowed=y>. > [Consulta: 02/05/2019].
- [7] G. Medina García , J. Grageda Grageda, J. A. Ruiz Corral y A. D. Báez González, «Uso de estaciones meteorológicas en la agricultura,» *Uso de estaciones meteorológicas en la agricultura*, p. 32, 2015.
- [8] Arango, D., Rincón, F., Manrique, L., & Buitrago, D. (2010). Estación Meteorológica Inalámbrica. *Umbral Científico*, 66-72.
- [9] Bosch Sensortec. (1 de 7 de 2008). BMP085 Digital Pressure Sensor Data Sheet. Alemania.
- [10] CCM. (s.f.). Obtenido de CCM: <http://es.ccm.net/contents/681-estandar-gsm-sistema-global-decomunicaciones-moviles>.
- [11] Cypress Semiconductor Corporation. (9 de 6 de 2016). Obtenido de Cypress Semiconductor: <http://www.cypress.com/file/138656/download>.
- [12] D-Robotics UK. (30 de 7 de 2010). *DHT11 Humidity & Temperature Sensor*.
- [13] Fundación Tecnológica Antonio de Arévalo [TECNAR]. (3 de 2 de 2015). Obtenido de Fundación

<http://orientacion.universia.net.co/universidades/fundacion-tecnologica-antonio-dearevalo---tecnar-40/noticias/nuevo-reconocimiento-para-tecnar--redcolsi-2840.html>

- [14] *IDEAM - Instituto de Hidrología, Meteorología y Estudios Ambientales*. (08 de 2018). Obtenido de <http://www.ideam.gov.co/>.
- [15] <http://www.scientificsales.com/Professional-Meteorological-Instruments-s/23.htm>. > [Consulta: 02/05/2019].
- [16] <http://l.facebook.com/l.php?u=https%3A%2F%2Fwww.laopinion.com.co%2Fpamplona> > [Consulta: 03/05/2019].
- [17] http://wiki.meteoclimatic.net/wiki/Resumen_normas_de_la_OMM_sobre_montaje_de_estaciones_meteorol%C3%B3gicas > [Consulta: 03/05/2019].
- [18] Oscar Almanza L, « Implementación De Sistema De Control Remoto Mediante Una Aplicación Móvil A Un Proceso Domótico Sobre Un Sistema Embebido», p. 16-21, 2019.
- [19] <https://www.eurolaser.cat/ca/proteccion-solar/>.> [Consulta: 07/08/2019].
- [20] <https://www.clima.com/informacion/ayuda/>. > [Consulta: 07/03/2020].
- [21] <https://es.meteocast.net/forecast/co/riohacha/> > [Consulta: 07/03/2020].