



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO
CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN**

**SISTEMA METEOROLÓGICO CON
TECNOLOGÍA IOT PARA LA
ASIGNATURA DE ENERGÍAS
RENOVABLES**

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Electrónica y Automatización

AUTORES: Vinicio Alejandro Basantes Correa

Josué Ricardo Castro Bolaño

TUTOR: William Paul Oñate Amaguaña

Quito-Ecuador

2024

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Vinicio Alejandro Basantes Correa con documento de identificación N° 1726482613 y Josué Ricardo Castro Bolaño con documento de identificación N° 1718550179; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 28 de febrero del año 2024

Atentamente,



Vinicio Alejandro Basantes Correa
1726482613



Josué Ricardo Castro Bolaño
1718550179

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Vinicio Alejandro Basantes Correa con documento de identificación N° 1726482613 y Josué Ricardo Castro Bolaño con documento de identificación N° 1718550179, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “Sistema meteorológico con tecnología IoT para la asignatura de energías renovables”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Electrónica y Automatización, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 28 de febrero del año 2024

Atentamente,



Vinicio Alejandro Basantes Correa

1726482613



Josue Ricardo Castro Bolaño

1718550179

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, William Paul Oñate Amaguaña con documento de identificación N° 1715580500 docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: SISTEMA METEOROLÓGICO CON TECNOLOGÍA IOT PARA LA ASIGNATURA DE ENERGÍAS RENOVABLES, realizado por Vinicio Alejandro Basantes Correa con documento de identificación N° 1726482613 y por Josué Ricardo Castro Bolaño con documento de identificación N° 1718550179, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto Técnico, que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 28 de febrero del año 2024

Atentamente,

A handwritten signature in blue ink, appearing to be 'WPA', is written over a horizontal line.

Ing. William Paul Oñate Amaguaña, Msc.

1715580500

DEDICATORIA

La presente Tesis se encuentra dedicada con profundo agradecimiento a mi familia, cuyo apoyo ha sido fundamental para culminar exitosamente mi carrera académica. Especial reconocimiento va dirigido a mis padres, Ricardo y Elvia, quienes siempre han estado a mi lado, brindándome apoyo incondicional y valiosos consejos. Asimismo, deseo expresar mi gratitud a mis hermanas Mónica, Tatiana y Karely, cuyas palabras de aliento y compañía han sido un sostén significativo a lo largo de este trayecto.

Josué Castro

DEDICATORIA

Agradezco a toda mi familia por estar apoyándome en todo momento por ser el soporte y la guía en el proceso de culminación de mi carrera. Quiero agradecer en especial mi mamá Mayra Correa que fue padre y madre para poder salir adelante y ser mi motor en mi vida te amo mamá, también a mis abuelitos Jaime Correa, Fanny Vizúete y mi hermano Santiago Correa por ser mis padres les dedico este logro y por saberme guiar y apoyar en todo lo que necesite se los agradezco de corazón, a Maritza Guanoles por ser el apoyo incondicional que tuve siempre presente.

Vinicio Basantes

ÍNDICE GENERAL

1	INTRODUCCIÓN.....	1
1.1	Justificación	1
1.2	Descripción del problema	2
1.3	Objetivos.....	3
1.3.1	Objetivo general	3
1.3.2	Objetivos específicos.....	3
2	Marco Teórico	4
2.1	Estación Meteorológica	4
2.1.1	Climatología.....	4
2.2	Tipos de estación meteorológica.....	5
2.2.1	Estación meteorológica tipo A	5
2.3	Variables de medición para la estación meteorológica.....	5
2.3.1	Velocidad de viento.....	5
2.3.2	Humedad	5
2.3.3	Temperatura.....	6
2.3.4	Radiación UV	6
2.4	Sensores	7
2.4.1	Temperatura y humedad.....	7
2.4.2	Radiación UV	8
2.4.3	Anemómetro.....	9
2.5	MCU y MPUS	10
2.5.1	ESP 32.....	10
2.5.2	Jetson nano	10
2.6	Enlaces virtuales	11
2.6.1	Grafana Cloud	11

2.6.2	InfluxDB.....	12
2.6.3	Thingspeak	12
3	Metodología.....	13
3.1	Diseño del prototipo IoT para la estación meteorológica	13
3.2	Instalación del sistema operativo	14
3.3	Diagrama esquemático.....	15
3.4	Ensamblaje de la estructura meteorológica	16
3.5	Ensamblaje de la estructura de monitoreo	16
3.6	Comunicación servidor local (NVIDIA) -nube	17
3.6.1	Creación de la base de datos.....	17
3.6.2	Programación en Python	19
3.6.3	Programación en Arduino IDE.....	22
3.7	Tablas de selección de sensores.....	24
3.7.1	Anemómetro	24
3.7.2	Sensor de Radiación UV	24
3.7.3	Sensor de temperatura y humedad.....	25
3.8	Acondicionamientos de señales en MCU (ESP 32).....	25
3.8.1	ADC en MCU anemómetro.....	25
3.8.2	ADC en MCU radiación UV	26
3.8.3	ADC en MCU temperatura y humedad	28
3.9	Visualización gráfica	28
3.9.1	Comunicación de InfluxDB con Grafana	28
3.9.2	Comunicación de Python con Thingspeak	29
3.9.3	Protocolo telnet.....	32
4	Análisis y resultados	34
5	cronograma y presupuesto	41
5.1	Cronograma de actividades.....	41

5.2	Presupuesto	42
6	Conclusiones y Recomendaciones	43
6.1	Conclusiones	43
6.2	Recomendaciones	44
	Referencias	45

ÍNDICE DE TABLAS

Tabla 1. Tipos de radiación	7
Tabla 2. Características de medición sensor de velocidad del viento	24
Tabla 3. Características de medición sensor de radiación UV	24
Tabla 4. Características de medición sensor de Humedad	25
Tabla 5. Parámetros Red U.P.S	34
Tabla 6. Direcciones IP	34
Tabla 7. NVIDIA rendimiento.....	35
Tabla 8. Resultados pregunta 1.....	35
Tabla 9. Resultados pregunta 2.....	36
Tabla 10. Resultados pregunta 3.....	36
Tabla 11. Comparación humedad estación INAMHI U.P.S vs estación meteorológica	37
Tabla 12. Comparación temperatura estación INAMHI U.P.S vs estación meteorológica	38
Tabla 13. Comparación velocidad de viento estación INAMHI U.P.S vs estación meteorológica	39
Tabla 14. Comparación velocidad de viento estación INAMHI U.P.S vs estación meteorológica	40
Tabla 15. Cronograma	41
Tabla 16. Presupuesto.....	42

ÍNDICE DE FIGURAS

Figura 1. Estación Meteorológica.....	4
Figura 2. Porcentaje de humedad	6
Figura 3. Índice UV	7
Figura 4. Sensor de humedad y temperatura	8
Figura 5. Sensor de radiación UV	9
Figura 6. Anemómetro.....	9
Figura 7. Modulo ESP32	10
Figura 8. NVIDIA Jetson Nano.....	11
Figura 9. Arquitectura IoT.....	14
Figura 10. Balena Etcher	14
Figura 11. Alimentación de los sensores de la estación meteorológica	15
Figura 12. Estructura estación metrológica	16
Figura 13. Estructura estación metrológica	17
Figura 14. Interfaz InfluxDB	18
Figura 15. Librerías para la comunicación HTTP	18
Figura 16. Credenciales del cliente	18
Figura 17. Visualización de la data	19
Figura 18. Arquitectura HTTP serve	20
Figura 19. Http server y socket.....	20
Figura 20. Librería InfluxDB	21
Figura 21. Credenciales InfluxDB.....	21
Figura 22. Credenciales InfluxDB.....	22
Figura 23. Configuración servidor HTTP	22
Figura 24. Envío de datos de Arduino IDE	23
Figura 25. Datos recibidos Esp32.....	23
Figura 26. Variables del Anemómetro	25
Figura 27. Cálculos de velocidad	26
Figura 28. Índice UV	27
Figura 29. Ajuste de valores de índice UV.....	27
Figura 30. Definición del módulo DHT22	28

Figura 31. Selección de la base de datos	29
Figura 32. Gestión de los datos	29
Figura 33. Canales en Thingspeak.....	30
Figura 34. Variables de los Canales	30
Figura 35. Api keys	31
Figura 36. Enviar datos a Thingspeak	31
Figura 37. Verifica la solicitud a Thingspeak	32
Figura 38. Configuración de fecha y hora	33
Figura 39. Envíos de datos por telnet	33
Figura 40. Humedad del Aire	37
Figura 41. Temperatura del Aire	38
Figura 42. Velocidad del viento	39
Figura 43. Redición solar	40

RESUMEN

Debido a que, en la Universidad Politécnica Salesiana Campus Sur, disponen de dos centros meteorológicos, el cual uno es del Municipio de Quito y otro de la carrera Eléctrica, estos se usaron como guía para la realización de nuestro sistema. En este proyecto de titulación se desarrolló un sistema meteorológico para la asignatura de energías renovables para la medición de variables meteorológicas por medio de una arquitectura IoT.

Para esto se realizó una estación meteorológica la cual se establece y se configura en la Universidad Politécnica Salesiana Bloque C, donde se realizan las pruebas necesarias para el acondicionamiento de los circuitos, tanto para los dispositivos MCU en el panel de control y los sensores utilizados. Previamente se realizó la calibración de los sensores para obtener un manejo y recepción de datos más precisos. Se utilizó un arreglo de pantallas para el monitoreo, visualización y registro de la data meteorológica enviados por los MCUS.

la tarjeta NVIDIA Jetson Nano se usó como una máquina para establecer conexión entre los dispositivos meteorológicos (humedad, temperatura, radiación UV y velocidad de viento), para ello se realizó la programación en Arduino IDE y Python, estableciendo un servidor HTTPS para el envío de datos. Por lo cual se creó una base de datos en InfluxDB por medio de código SQL, esta gestiona el filtro de datos enviados por los sensores para su respectiva medición.

En este proyecto se realizó la comparación de dashboards generadas en Grafana y Thingspeak con la finalidad de monitorear los datos recibidos que son enviados de la estación meteorológica en tiempo real. Por medio de protocolo Telnet permite generar un datalogger para el registro de datos en una hoja de cálculo.

Palabras Clave: Climatología, estación meteorológica, meteorología, tecnología IoT.

ABSTRACT

Due to the fact that, at the Salesian Polytechnic University South Campus, they have two meteorological centers, one of which is from the Municipality of Quito and the other from the Electrical career, these were used as a guide for the realization of our system. In this degree project, a meteorological system was developed for the subject of renewable energies for the measurement of meteorological variables by means of an IoT architecture.

For this, a meteorological station was set up and configured at the Salesian Polytechnic University Block C, where the necessary tests are carried out for the conditioning of the circuits, both for the MCU devices in the control panel and the sensors used. Previously, the sensors were calibrated to obtain a more accurate handling and reception of data. An array of screens was used to monitor, display, and record the meteorological data sent by the MCUS.

The NVIDIA Jetson Nano card was used as a machine to establish a connection between the meteorological devices (humidity, temperature, UV radiation and wind speed), for this the programming was carried out in Arduino IDE and Python, establishing an HTTPS server for sending data. For this reason, a database was created in Influx DB by means of SQL code, which manages the filter of data sent by the sensors to be measured.

In this project, the comparison of dashboards generated in Grafana and Thing speak was carried out in order to monitor the data received that is sent from the weather station in real time. By means of the Telnet protocol, it is possible to generate a datalogger for the recording of data in a spreadsheet.

Keywords: Climatology, weather station, meteorology, IoT technology.

CAPÍTULO 1

1 INTRODUCCIÓN

1.1 Justificación

Un sistema meteorológico es una infraestructura y conjunto de herramientas que se utilizan para recopilar, analizar y predecir las condiciones atmosféricas y climáticas en una determinada región o área geográfica. Estos sistemas están diseñados para monitorear y medir variables meteorológicas, como temperatura, humedad, presión atmosférica, velocidad y dirección del viento, precipitación, radiación solar, entre otros. Un ejemplo de esto es que en la Universidad Politécnica Salesiana Campus Sur dispone 2 sistemas uno implementado por el Distrito Metropolitano de Quito y otro la carrera de electricidad.

De lo antes mencionado este proyecto técnico pretende desarrollar un sistema meteorológico que mida y registre las variables como humedad, temperatura, radiación y viento en un data logger con comunicación inalámbrica, esto quiere decir que si la infraestructura de red falla se registrarán en la base de datos local hasta que nuevamente exista conexión de red, para de esta forma enviar los datos almacenados a la base de datos de Cloud, es así que se desarrollará una red de sensores empleando MCU's y se comunicará a un bróker en nube, de esta manera se podrá monitorear, recolectar y visualizar las variables en cuestión por medio de un arreglo de pantallas recicladas y una trajera NVIDIA.

1.2 Descripción del problema

Si un sistema meteorológico sería deficiente conocer las variables para un análisis energético (Wang et al., 2024), esto se menciona porque es de facto ya que el panorama energético que se avecina es desafiante, en cuanto a las energías primarias no renovables se agoten (Dyshlyuk et al., 2024), un ejemplo ; son las reservas de petróleo en el Ecuador es de apenas son 0.01% (Vallejo et al., 2019), por otro lado existe una ambigüedad porque en el mismo país existen diferentes puntos de interés para la explotación de la energía renovable solar (Torres-Quezada & Lituma-Saetama, 2023) sin embargo, la política actual no vela por estos intereses medios ambientales. No conocer aquellas energías renovables que según ERNL serían para el 2050 cruciales para la producción energética volviéndonos dependientes energéticamente (Guo et al., 2024).

De lo antes mencionado, existen varios países desarrollados que desde hace varios años atrás explotan diferentes recursos renovables (EL Bouji et al., 2023) sin embargo en Ecuador se tiene un desinterés en estos temas y solo se explota a la energía hidráulica y en parte la energía eólica (Villacreses et al., 2022), otro inconveniente es la falta de cultura energética que presenta la población ecuatoriana y el desconocimiento de fuentes de energías renovables, en cambio desde las entidades gubernamentales no existen un apoyo reducir los aranceles de dispositivos de elementos que constituye un sistema fotovoltaico (Benavides et al., 2022), de igual manera el sector agrícola no presenta inversión energética lo cual lo deja fuera de juego para la competitividad internacional en exportación de productos (Bolaños & Tapia, 2019).

De lo antes mencionado se ha identificado algunos estudios para dar solución a las problemáticas anteriores, es así que (Constante K, 2021) realizó el análisis, diseño, construcción e implementación de un sistema web para la gestión de datos meteorológicos mediante la utilización de un api, el cual se ejecutará de una manera autónoma sin ninguna dependencia de formatos ambientales tradicionales.

(Lopez Mateos et al., 2018) diseñó un sistema meteorológico digital usando tecnologías que faciliten el acceso de los datos de manera remota. haciendo factible el uso de una interfaz web que ayude al monitoreo de los sensores utilizados en cada una de las torres meteorológicas utilizadas.

(Felipe et al., 2021) La solución a la problemática es crear un prototipo de estación meteorológica IOT, permitiendo de esta manera que la estación establezca una conexión a internet y así se podrán obtener mediciones a través del uso de sensores, los datos obtenidos serán almacenados en una base de datos y consultados a través de un servicio web, haciendo uso de una tarjeta Raspberry Pi 4 y Arduino Uno que nos posibilitan hacer uso al máximo de las tecnologías IOT.

1.3 OBJETIVOS

1.3.1 Objetivo general

Desarrollar un sistema meteorológico basado en tecnología IoT que permita la recolección, monitoreo y visualización de datos en tiempo real.

1.3.2 Objetivos específicos

- Investigar sobre dispositivos meteorológicos inalámbricos, arquitectura IoT y servicio web que permitan la elaboración de un sistema meteorológico con tecnología actual, mediante base de datos y repositorios bibliográficos.
- Crear una estación meteorológica para la medición de variables de radiación solar, velocidad del viento, temperatura ambiente y humedad mediante instrumentación.
- Diseñar un sistema de comunicación entre un datalogger y Cloud para el envío de información meteorológica a una base de datos mediante una arquitectura IoT.
- Realizar una interfaz para el monitoreo y visualización de la data meteorológica desde un sistema de análisis de datos y una estructura constituida por un arreglo de pantallas recicladas.
- Verificar el funcionamiento del sistema metrológico con comunicación inalámbrica en la arquitectura IoT para su validación mediante pruebas experimentales de campo.

CAPÍTULO 2

2 MARCO TEÓRICO

2.1 Estación Meteorológica

Estación meteorológica son instalaciones asignadas para valorar y estudiar regularmente, distintas variables meteorológicas. La información obtenida se usa para realizar predicciones y estudios climáticos en función de modelos numéricos obtenidos. Está compuesta por sensores que se encargan de recopilar, almacenar y transmitir datos a la Nube para el monitoreo de cada uno de ellos.

Las estaciones meteorológicas se usan considerablemente en el sector público y privado, estas utilizan las predicciones de los datos estadísticos como, fenómenos atmosféricos y climáticos los cuales permiten llevar un análisis y control detallado.

Figura 1. Estación Meteorológica



Fuente: (METEOCULTURA - ESTACIONES METEOROLÓGICAS & METEOROLOGÍA, 2023)

2.1.1 Climatología

Climatología es una de las ramas de la meteorología que se encarga de predecir las condiciones atmosféricas en distintos periodos ya sean relativamente cortos o largos. El estudio de las distintas variables meteorológicas (viento, humedad, temperatura, radiación) se usan generalmente para realizar estadísticas matemáticas del recopilado de datos de pronósticos y modelos físicos.

2.2 Tipos de estación meteorológica

Las estaciones meteorológicas tienen como objetivo medir los elementos y factores del clima como: dirección y velocidad del viento, temperatura, humedad, presión, radiación, precipitación entre otros. Según el número de parámetros que determinen se puede dividir en cinco tipos: Pluviométricas, Pluviográficas, Telemétricas, siendo la Tipo A y Tipo B las más completas.

2.2.1 Estación meteorológica tipo A

Estación Tipo A es aquella que registra datos de cantidad, tiempo de duración y potencia de las lluvias, temperatura ambiente (máxima y mínima), viento (dirección, recorrido sentido). Tan bien registra datos gráficos continuos de las horas del sol y su radiación. Además, la evaporación y temperatura del suelo.

2.3 Variables de medición para la estación meteorológica

2.3.1 Velocidad de viento

El viento por lo general mide dos variables, dirección y velocidad. Los instrumentos usados para su medición son el anemómetro y la veleta, estos recopilan y almacenan datos importantes para diversas situaciones y actividades. La medición del anemómetro puede medirse en (km/s) y (m/s), esto va a depender de la constante que indique el sensor en sus especificaciones técnicas y permita variar en los rangos deseados de medición. Por otra parte, la veleta indica la dirección en la que varían las distintas rachas de viento, los puntos cardinales sirvieron como guía para detectar el sentido en que vienen dichas rachas mencionadas para recopilar distintos datos en distintos intervalos de tiempo.

2.3.2 Humedad

La Humedad indica cuanta cantidad de vapor de agua existe en el aire, esta la podemos expresar en varias magnitudes. Por otra parte, la humedad relativa expresa la cantidad de vapor de agua que contiene el aire y la cantidad de aire que se puede contener a una máxima temperatura y esta se expresa en porcentaje %, por ejemplo: 5 RH% indica que el aire está demasiado seco.

Figura 2. Porcentaje de humedad



Fuente: (Hogar Seco, 2021)

2.3.3 Temperatura

La temperatura indica el calentamiento o enfriamiento del aire el cual, es el cambio de calor entre la atmósfera y la tierra, también muestra el nivel de energía interna que posee un lugar en específico en tiempo real a través de valores numéricos. La temperatura ambiente que las personas prefieren varía entre un rango de 20 a 25 grados centígrados lo que sería la zona de confort del usuario.

2.3.4 Radiación UV

La radiación ultravioleta UV es un tipo de radiación No Ionizante que emite el sol.

Existen 3 tipos de radiación que son: (UVA) ultravioleta A, (UVB) ultravioleta B, (UVC) ultravioleta C, cada una posee su longitud de onda. En general la mayoría de radiación UV que ingresa la tierra es UVA y una pequeña parte es UVB que también ingresa a la tierra.

De lo antes mencionado las radiaciones pueden generar daños a corto y largo plazo en la salud, pero las más peligrosa en este caso es la UVA que puede causar daños en la piel ya que es la que fue que ingresa principalmente a la tierra.

Tabla 1. Tipos de radiación

Tipo de onda	UVA	UVB	UVC
Longitud de onda	315-399 nm	280-314 nm	100-279 nm
Nivel de absorción	No absorbida por la capa de ozono	Mayormente absorbida por la capa de ozono, pero cierta cantidad llega a la superficie de la tierra	Completamente absorbida por la capa de ozono y la atmósfera

Fuente: (Centro Nacional de Salud Ambiental, 2023)

2.3.5 Índice UV

El índice UV contiene una escala que fueron establecidas por la Organización Mundial de la Salud la cual especifica cuanto es el riesgo de peligrosidad para la salud y el nivel de exposición en la piel.

el índice varía según el nivel de exposición que son:

- 0 a 2: Bajo
- 3 a 5: Moderado
- 6 a 7: Alto
- 8 a 10: Muy alto
- 11 o +: Extremo

Figura 3. Índice UV



Fuente: (EAP, 2023)

2.4 Sensores

2.4.1 Temperatura y humedad

Es un sensor digital relativo también conocido como DHT21 o AM2301 que contiene una alta precisión, perfecto para aplicaciones en exteriores y procesos como el control de temperatura y monitoreo ambiental. El rango de medición para la humedad es de 0 a

99.9% RH, la temperatura va desde -40 a 80 grados centígrados, un voltaje de 3.5V a 5.5V DC y corriente de 1mA a 1.5mA.

De lo antes mencionado el sensor es simple de usar ya que contiene plataformas como son Raspberry pi, Arduino. En software dispone de "Single bus" como protocolo principal para el uso de librerías en Arduino.

Figura 4. Sensor de humedad y temperatura



Fuente: (aelectronics, s.f.)

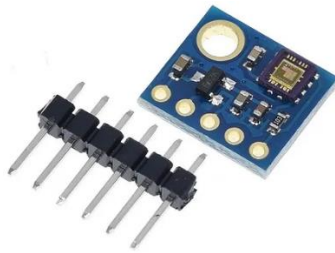
2.4.2 Radiación UV

El módulo ML85511 es un sensor analógico que sirve para la medición de luz ultravioleta (UV) con rangos de longitud de onda de 280 – 390 nm. El espectro de radiación va desde UV-B hasta el UV-A donde la medición analógica corresponde a la intensidad UV en (nW/cm²). Es Utilizado para proyectos que requieran el monitoreo de condiciones ambientales para detectar el índice UV.

Especificaciones

- Voltaje: 3.3V a 5.5V
- Salida analógica
- Bajo consumo de energía

Figura 5. Sensor de radiación UV



Fuente: (Plexilab, 2021)

2.4.3 Anemómetro

Es un sensor analógico que mide la dirección y la velocidad del viento, dicho sensor recopila datos de cada una de las rachas de viento que las produzcan. las mediciones pueden variar por las distintas rachas de viento por lo que debemos calibrar para obtener las mediciones deseadas. Cuando el sensor gira impulsa un generador pequeño de corriente continua, cuanto más rápido gire este crea un mayor voltaje y genera 2V CC con un error del 10 %.

Figura 6. Anemómetro



Fuente: (AliExpress, 2023)

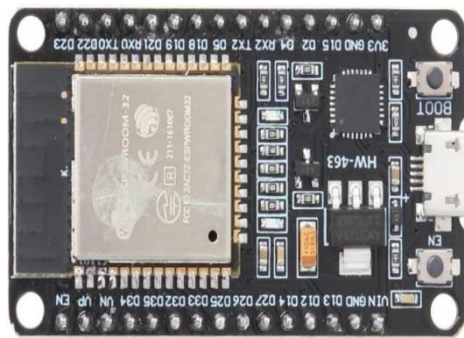
2.5 MCU Y MPUS

2.5.1 ESP 32

La ESP32 es un módulo genérico que contiene conectividad a wifi, bluetooth y da paso al crecimiento de la IoT (Internet de las cosas), esta incluye un microcontrolador de doble núcleo y funciona a una temperatura de -40 a 125 grados centígrados, lo cual hace factible su uso en entornos industriales.

La esp32 es compatible con varios lenguajes de programación como: C++ el cual es más utilizado por la plataforma Arduino, Python para realizar aplicaciones web y JavaScript que podemos utilizar con la plataforma NodeMCU para el desarrollo de aplicaciones web y ambientes virtuales.

Figura 7. Modulo ESP32



Fuente: (TEKmikro, 2023)

2.5.2 Jetson nano

la Jetson Nano es una minicomputadora, que fue diseñada para desarrollar aplicaciones básicas de las IA en Edge e impulsar nuevos dispositivos, su sistema operativo es Ubuntu e incluye librerías gráficas, archivos multimedia, visión de computación, Deep Learnig etc. También puede ejecutar múltiples aplicaciones como detección de objetos, procesamiento de vos y entrenamiento de redes neuronales.

Figura 8. NVIDIA Jetson Nano



Fuente: (NVIDIA , s.f.)

Especificaciones

- GPU: Maxwell de 128 núcleos
- UPC: ARM A57 de cuatro núcleos a 1,43 GHz
- Memoria: 4 GB LPDDR4 de 64 bits 25,6 GB/s
- Almacenamiento: microSD (no incluida)
- Codificación de vídeo: 4K a 30 | 4x 1080p a 30 | 9x 720p a 30 (H.264/H.265)
- Decodificación de vídeo: 4K a 60 | 2x 4K a 30 | 8x 1080p a 30 | 18x 720p a 30 (H.264/H.265)
- Cámara: 2 carriles MIPI CSI-2 DPHY
- Conectividad: Gigabit Ethernet, clave M.2 E
- Mostrar: HDMI y puerto de pantalla
- USB: 4x USB 3.0, USB 2.0 Micro-B
- Otros: GPIO, yo 2 C, yo 2 S, SPI, UART
- Mecánico: Conector de borde de 69 mm x 45 mm, 260 pines

2.6 ENLACES VIRTUALES

2.6.1 Grafana Cloud

Es una aplicación que permite visualizar datos en tiempo real, esto ayuda a los usuarios a ver gráficas y tablas que son monitoreadas por un panel de control lo que es más sencillo interpretar y comprender los valores que son recopilados y entregados a la base de datos.

El protocolo que usa dicha aplicación es SQL que ayuda a la administración y gestión de bases de datos relacionales.

2.6.2 InfluxDB

InfluxDB Cloud utiliza Telegraf para recopilar datos de series temporales y enviarlos a bases de datos, aplicaciones, sistemas y sensores de IoT. (influxdata, s.f.). Telegraf es un servidor con más de 300 complementos y está escrito en GO

Actualmente, InfluxDB está compuesto por una variedad de librerías de cliente como: Arduino, GO, Python, Java, Script estas librerías son sencillas de encontrar en la pestaña interfaz de usuario.

El conjunto robusto de API permite el acceso programático a InfluxDB Cloud. Estas API son comunes en InfluxDB Cloud, lo que le permite escribir código una vez y ejecutarlo localmente en una versión de código abierto de la base de datos o como servicio de la base de datos elástica.

2.6.3 Thingspeak

Thingspeak Es una plataforma con tecnología IoT que permite gestionar de forma bidireccional los datos enviados a la Nube y visualizarlos en tiempo real. Su principal característica es ser una plataforma Open Source Con una API para guardar y recopilar datos utilizando el protocolo HTTP. (Cita). Los lenguajes de programación que contiene Thingspeak son: Java Script y HTML.

Como fuerte de toda plataforma IoT es la combinación de distintos dispositivos que contengan hardware y software, las plataformas con las que comparte su integración son: Arduino, Raspberry Pi, Matlab etc.

CAPITULO 3

3 Metodología

3.1 Diseño del prototipo IoT para la estación meteorológica

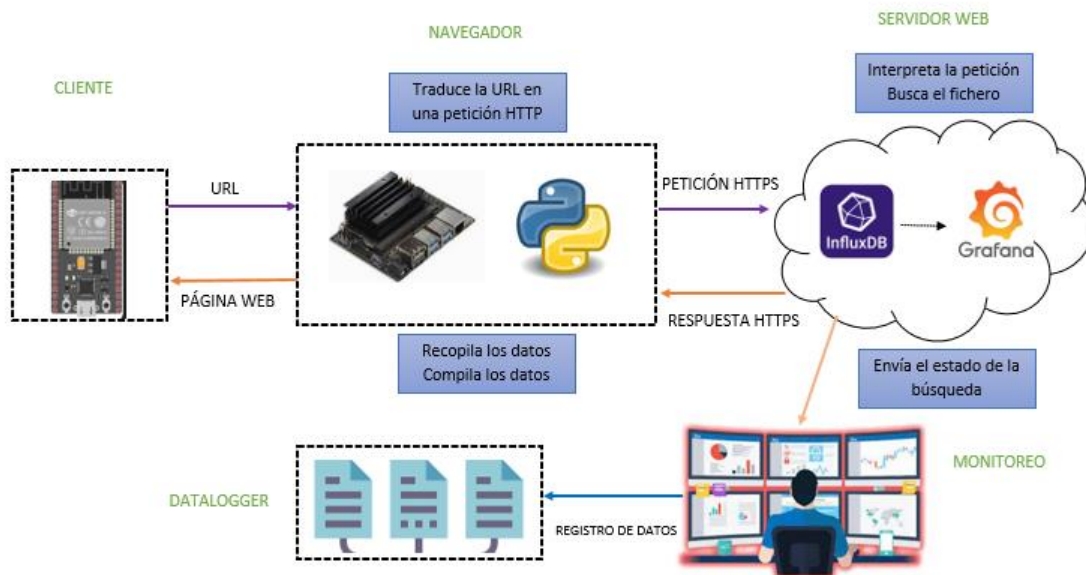
En este capítulo, se abordará la implementación de un prototipo avanzado de estación meteorológica para la materia de Energías Renovables, basado en la arquitectura de Internet de las cosas (IoT) que utiliza protocolos de comunicación HTTP, HTTPS y Telnet para un monitoreo seguro de las variables de medición. Es importante estudiar la calidad del medio ambiente y el clima y determinar cuándo las condiciones son ideales para la producción de energía renovable

La estación meteorológica IoT incluye sensores adaptados para la medición de variables ambientales como la temperatura, la humedad, la velocidad del viento y la radiación solar. Estos sensores están conectados a sistemas integrados como la placa de desarrollo Jetson Nano, que actúa como un nodo central de procesamiento y comunicación. Del procesamiento y comunicación.

Además, se implementará el protocolo telnet para respaldar el monitoreo y la gestión remota de la estación meteorológica. Esto permitirá al operador acceder a la estación de forma remota para realizar configuraciones como el registro de datos.

El monitoreo en tiempo real y el registro de datos serán aspectos fundamentales de esta arquitectura de IoT. Se establecerán mecanismos para registrar los datos recopilados en una base de datos, facilitando el análisis histórico e identificando tendencias a lo largo del tiempo. Este enfoque proporcionará información valiosa para un análisis y posibles predicciones en el uso de energías renovables, solar (fotovoltaica y térmica) y eólica vertical.

Figura 9. Arquitectura IoT

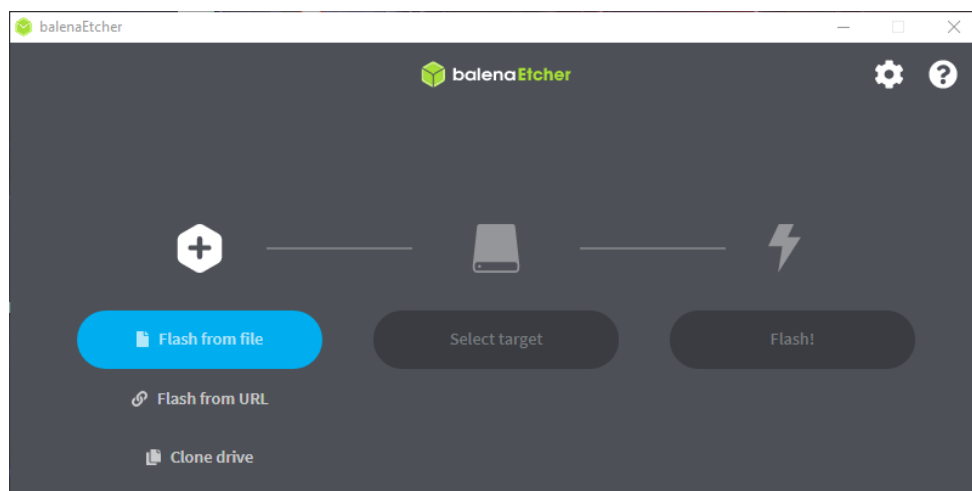


Fuente: Autores

3.2 Instalación del sistema operativo

Para la instalación del sistema operativo de la Jetson Nano, inicialmente se escribe la imagen en una tarjeta microSD y si es necesario debemos flashear. Para esto usamos el programa balenaEtcher. (ver figura 10). Al finalizar se creará un zip en el escritorio donde se encuentra la imagen de la Jetson Nano almacenada con el nombre “SDCardFormtter_V5WinEn.

Figura 10. Balena Etcher



Fuente: Autores

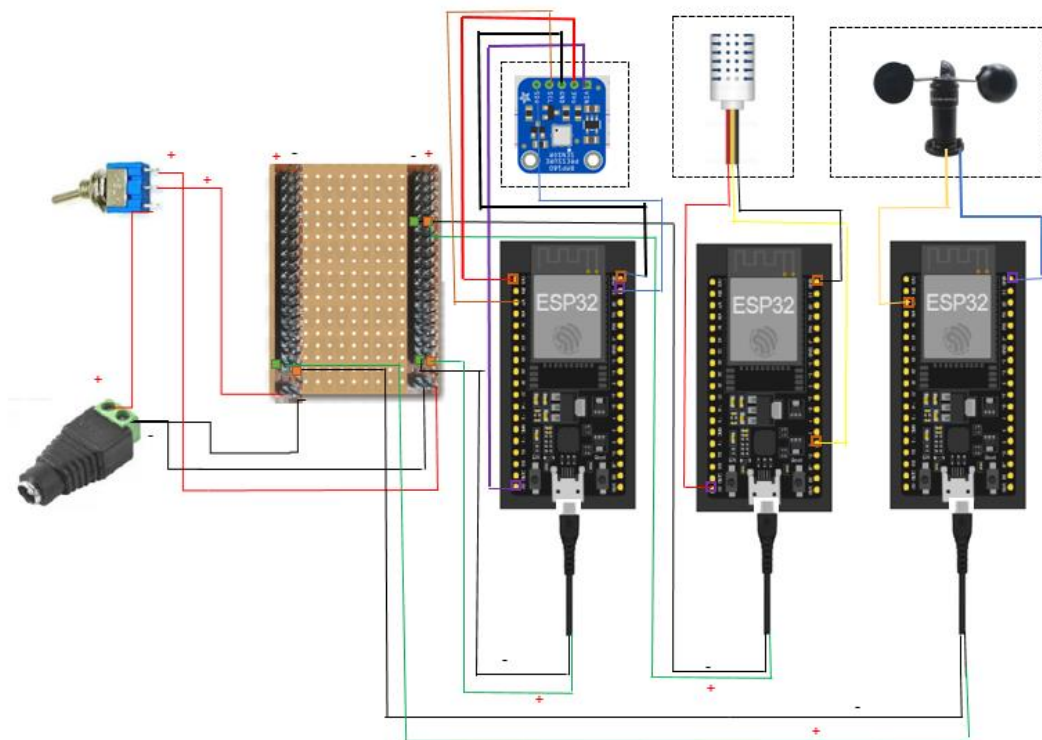
Colocamos la MicroSD en la ranura de la tarjeta Jetson Nano. Cuando inicie el administrador de ventanas aceptamos las licencias, configuramos el nombre del equipo y las credenciales para el inicio de sesión, ajustamos el tamaño de partición que nos indica en (MB), por lo cual usamos el tamaño máximo que nos permite. Realizado este proceso se genera la pantalla de inicio, el cual su sistema operativo es Ubuntu 18.4

3.3 DIAGRAMA ESQUEMÁTICO

En la figura 12 se ilustra el esquemático electrónico ha sido diseñado en función del nivel del voltaje de cada uno de los sensores, así como los equipos de adquisición y procesamiento de datos como las Esp32 en donde sus niveles de tensión están implementados de acuerdo a sus necesidades electrónicas.

Para la creación del circuito, se llevó a cabo la verificación de las especificaciones técnicas bajo hoja de datos, de esta manera se realizan las conexiones de los dispositivos y el código necesario para su funcionamiento. Se considera el tipo de ESP32 a utilizar y la configuración de los 4 sensores, que incluyen los sensores de temperatura, humedad y radiación UV.

Figura 11. Alimentación de los sensores de la estación meteorológica



Fuente: Autores

3.4 Ensamblaje de la estructura meteorológica

En la figura 12 se observa que ciertos sensores tienen la forma cilíndrica vertical que se apoya en soportes planos, y la estructura comprende diversos soportes para sus componentes. Debido a diversos cambios climáticos, es necesaria la colocación de abrazaderas y soportes en forma de L para la prueba y montaje de parlantes musicales marca "Mowey".

Figura 12. Estructura estación metrológica

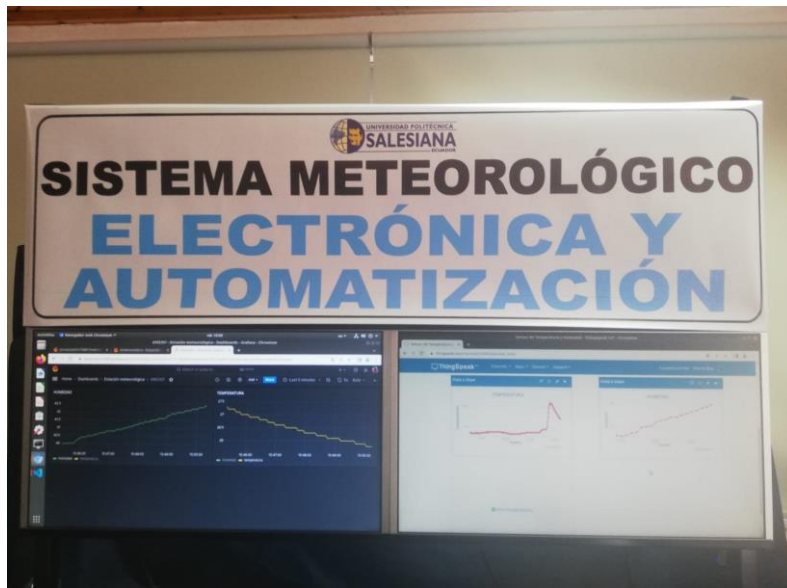


Fuente: Autores

3.5 ENSAMBLAJE DE LA ESTRUCTURA DE MONITOREO

Como se aprecia en la figura 13 la estructura está conformada para aun arreglo de 2 pantallas recicladas la cual consta también de poder moverla con facilidad en cualquier lugar, de una base para montar un splitter para VGA y la tarjeta Jetson nano.

Figura 13. Estructura estación metrológica



Fuente: Autores

3.6 Comunicación servidor local (NVIDIA) -nube

Para establecer una conexión entre los dispositivos de medición meteorológica (Temperatura, Humedad, velocidad de viento y radiación solar) y la Nube se desarrolló una programación en un entorno de desarrollo Arduino IDE y Python estableciendo así un servidor HTTP. La nube actúa como una base de datos centralizada y plataforma de visualización para facilitar el monitoreo y el análisis de los datos recolectados

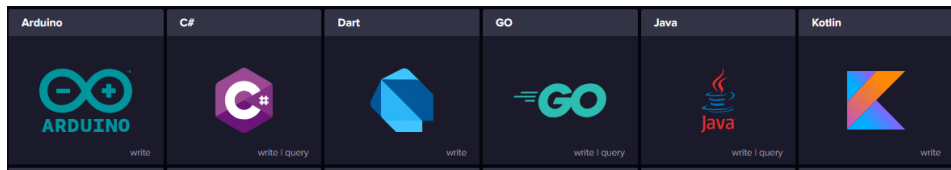
3.6.1 Creación de la base de datos

la creación de la base de datos se simplificó utilizando la plataforma de InfluxDB, donde se procedió a registrar una cuenta gratuita. Esta plataforma ofrece una interfaz intuitiva que facilita la gestión y almacenamiento de datos de series temporales de manera organizada.

Una vez creada la cuenta el usuario podrá acceder a las herramientas y servicios proporcionados por InfluxDB, de esta manera se gestiona los datos de la data meteorológica.

En la figura 14 se muestra el uso de librerías que podrá crear los Buckets para el almacenamiento de los datos medidos por los sensores.

Figura 14. Interfaz InfluxDB



Fuente: Autores

De esta manera y una vez enlazado el Arduino a la plataforma el cliente inicializará la comunicación. Por otro lado, dentro de la preparación se debe incluir las librerías para la comunicación entre ambas plataformas promedio del protocolo de comunicación HTTPPS donde los datos son enviados de una forma más segura cuando se ejecuta la petición HTTP. (ver figura 15)

Figura 15. Librerías para la comunicación HTTP

```
#include <WiFiMulti.h>
#include <InfluxDbClient.h>
#include <InfluxDbCloud.h>
#include <DHT.h>
#include <ArduinoJson.h>
#include <HTTPClient.h>
#include <TelnetStream.h>
#include <TimeLib.h>
```

libreria cliente, servicio de nube

libreria HTTPCLIENT

Fuente: Autores

Dentro de la inicialización de la plataforma se encuentran las credenciales correspondientes para incluirlas en la programación de Arduino como muestra la figura 16, permite la comunicación bidireccional de los datos con los Bucket previamente creados para cada sensor.

Figura 16. Credenciales del cliente

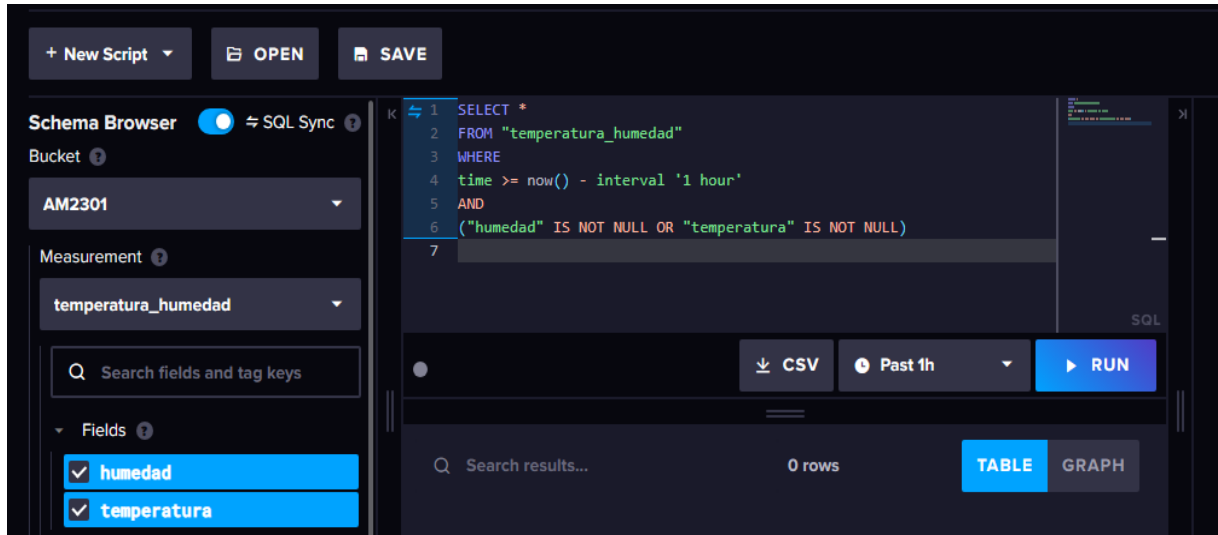
```
#define INFLUXDB_URL "https://us-east-1-1.aws.cloud2.influxdata.com"
#define INFLUXDB_TOKEN "vtyT3mJAcBtkPxdnN3f8tJ9ogf4971gvHPNS_heIGAQOYHHC-NIP0-PVvBMr-a7zTmnGbnB2TBSrimPSI75GKA=="
#define INFLUXDB_ORG "6d7a15e7ef91f632"
#define INFLUXDB_BUCKET "ML8511"
```

Fuente: Autores

Para la visualización de los datos dentro de la interfaz de InfluxDB como se muestra en la figura 17 la plataforma permite realizarlo por medio de código SQL en donde podemos

filtrar los datos enviados de los sensores como humedad, temperatura, velocidad de viento y radiación solar.

Figura 17. Visualización de la data



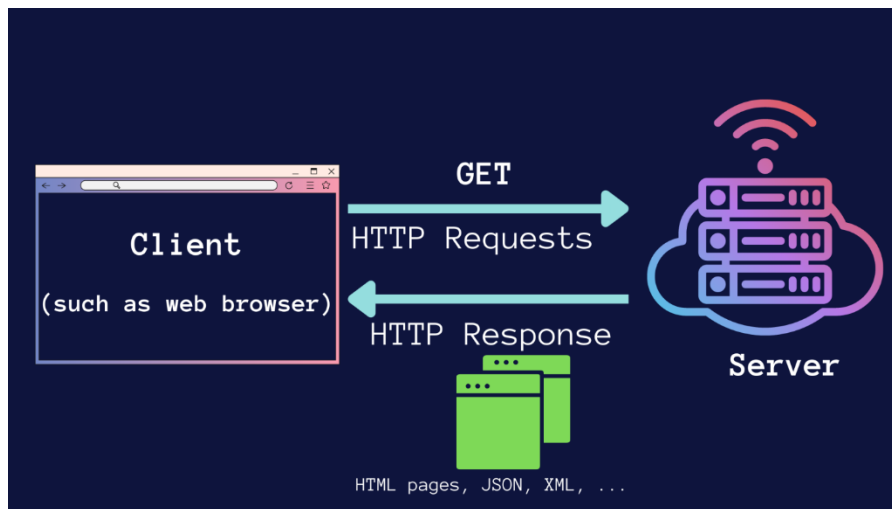
Fuente: Autores

3.6.2 Programación en Python

En la figura 18 indica la visualización de datos de los dispositivos MCU y la comunicación correspondiente con la base de datos InfluxDB se realiza mediante un servidor diseñado para manejar solicitudes HTTP de los dispositivos ESP32. Configurado para aceptar solicitudes HTTP, este servidor actúa como una interfaz que facilita la transferencia de datos desde los dispositivos de envío a la infraestructura de almacenamiento en la nube representada por InfluxDB.

La introducción de un servidor local HTTP brinda la oportunidad del procesamiento de datos provenientes de los dispositivos con poca latencia, lo que permite el almacenamiento en una base de datos temporal. Este enfoque arquitectónico se alinea con prácticas de gestión de datos de IoT, lo que garantiza una comunicación sólida y segura entre los dispositivos de borde y los servicios de almacenamiento en la nube.

Figura 18. Arquitectura HTTP serve



Fuente: (Bala, 2021)

Como se muestra en la figura 19. Se creó un servidor “http. server” en Python, la cual permite la recopilación de los datos dentro de esta. El módulo "socket" constituye un elemento fundamental en programación que facilita un mecanismo de comunicación entre procesos a través de redes. La implementación se realiza con el propósito específico de obtener la dirección IP local de la máquina en la que se ejecuta el programa. La utilización de "socket" para esta tarea se justifica por su capacidad para interactuar con las funciones subyacentes del sistema operativo relacionadas con la red, permitiendo así acceder a información esencial como la dirección IP local.

Figura 19. Http server y socket

```
from http.server import BaseHTTPRequestHandler, HTTPServer
```

Clase, servicio de HTTP

Manejo de la clase de servidor HTTP

```
host = socket.gethostname(socket.gethostname())
```

Obtención de IP de la maquina local

Fuente: Autores

Para facilitar la comunicación con la base de datos como muestra la figura 20, se ha instalado la biblioteca Json correspondiente que permite el envío de datos en formato String de forma secuencial. Esta biblioteca perteneciente al entorno de desarrollo permite analizar los datos transmitidos.

Figura 20. Librería InfluxDB

```
from http.server import BaseHTTPRequestHandler, HTTPServer
import socket
import json
from influxdb_client import InfluxDBClient, Point, WriteOptions
import requests
```

Librería de InfluxDB

Fuente: Autores

Para conectarse a una base de datos InfluxDB como indica la figura 21, es necesario especificar las credenciales proporcionadas por la interfaz InfluxDB. Estas credenciales incluyen una URL de conexión, un token de autenticación y otros parámetros que son necesarias para la autenticación de las operaciones de escritura y lectura de la base de datos. Al incluir estas credenciales correctamente en el código, se garantiza la seguridad y la legalidad de estas operaciones. El uso de un enfoque estricto para la gestión de credenciales ayuda a mantener la integridad y la confiabilidad de la conexión con InfluxDB.

Figura 21. Credenciales InfluxDB

```
url = "https://us-east-1-1.aws.cloud2.influxdata.com"
token = "Qb5FbCP6aWrb1q5vwlLjMsZ5QqYgx5Y0LrMdP470aQh9FNKERVz2_hvjS-f_lIyRz5iYA9c4-qIwyfIHvbSCBQ=="
org = "Tesis"
bucket = "AM2301"
```

Fuente: Autores

En la figura 22 se aplica una configuración que permite escribir datos en el entorno de la nube. Esta operación se realiza de manera tangible, permitiendo la identificación clara del tipo de información que se envía. Esta controvertida metodología facilita la separación y categorización de datos cuando se envía a la nube. La capacidad de transmitir datos de

manera que se distingue las variables dentro de los bucket es fundamental para la interpretación de los datos en un entorno de almacenamiento remoto.

Figura 22. Credenciales InfluxDB

```
# Verifica si la solicitud a ThingSpeak fue exitosa
if respuesta_thingspeak.status_code == 200:
    print("Datos enviados a ThingSpeak exitosamente.")
else:
    print("Error al enviar datos a ThingSpeak.")
```

Fuente: Autores

En la figura 23 indica la configuración que habilita la recepción de direcciones IP dinámicas desde dispositivos Esp32, así como se define el puerto a través de cual se escuchara las solicitudes HTTP. Los puertos seleccionados para la comunicación pueden variar y están configurados en un rango común que incluye los puertos 80,80 y 8000. Esta selección de puertos facilita la interoperabilidad y la flexibilidad en los entornos de desarrollo proporcionar opciones comúnmente utilizadas en los servicios web.

Figura 23. Configuración servidor HTTP

```
# Configura el servidor HTTP
puerto = 8081
httpd = HTTPServer(('0.0.0.0', puerto), RequestHandler)
print("Servidor en ejecución en http://{host}:{puerto}").format(host, puerto)
```

Fuente: Autores

3.6.3 Programación en Arduino IDE

Dentro del entorno de Arduino IDE es utilizado para la programación los microcontroladores Esp32. Se implementa las instrucciones que permitan la lectura de los datos de los sensores de temperatura, humedad, velocidad de viento y radicación solar. Donde estos datos luego son procesados y enviados a Python mediante un servidor de HTTP. (ver figura 24)

Figura 24. Envío de datos de Arduino IDE

```
void enviarDatosAPython(float intensidadUV, String nivelIntensidad) {
  HTTPClient http;
  String url = "http://192.168.0.106:8082";
  http.begin(url);
  http.addHeader("Content-Type", "application/json");

  String cargaUtil = "{\"intensidad_uv\": " + String(intensidadUV) + ", \"nivel_intensidad\": \"" + nivelIntensidad + "\"}";
  int codigoHTTPRespuesta = http.POST(cargaUtil);
  if (codigoHTTPRespuesta == 200) {
    Serial.println("Datos enviados a Python exitosamente");
  } else {
    Serial.print("Fallo al enviar datos a Python. Código de error HTTP: ");
    Serial.println(codigoHTTPRespuesta);
  }
}
```

Fuente: Autores

Conforme se observa en la Figura 25, en la sección de declaración de variables, se integra la dirección IP de la máquina local junto con el puerto designado para la escucha de la solicitud HTTP. Esta configuración establece una conexión efectiva entre el emisor y el receptor. La dirección IP identifica inequívocamente la ubicación del servidor, mientras que el puerto designa el canal específico a través del cual se llevará a cabo la comunicación.

Además, se destaca la creación de una carga útil en formato JSON. En este contexto, la cadena de datos resultante encapsula únicamente la información especificada para ser recibida por la máquina local, como también si los datos no fueron recibidos dando el fallo en HTTP

En Python para la recepción de los datos se realiza un arreglo String, así como la sincronización de la zona horario, en el proceso de los datos recibidos se emprime la verificación si los datos fueron recibidos o hubo un fallo en él envió.

Figura 25. Datos recibidos Esp32

```
data = json.loads(post_data.decode('utf-8'))

# Procesa los datos recibidos desde el ESP32
print("Datos recibidos desde ESP32_SensorAM2301:")
print("Temperatura:", data.get("temperatura"))
print("Humedad:", data.get("humedad"))
```

Fuente: Autores

3.7 TABLAS DE SELECCION DE SENSORES

En esta sección se presenta un análisis de selección de los componentes que constituyen los sensores del sistema metrológico como: temperatura, humedad, radiación solar y velocidad del viento, para eso se dispuso de tablas de selección donde se califica en un rango del 0 al 5 punto para seleccionar aquel de mayor puntaje

3.7.1 Anemómetro

Para la selección del sensor que permita la medición de la velocidad del viento, este debe contar de un sistema de acondicionamiento de señal, por lo que se realizó una comparación con tres tipos de sensores similares dentro del mercado nacional, tal como muestra en la tabla 1.

Tabla 2. Características de medición sensor de velocidad del viento

Características	Anemómetro Wind speed 0-2v	Anemómetro WH-SP-WS01 0-5v	Anemómetro SKU 0-12 v
Rango de medición	4	5	5
Costo	5	4	2
Disponibilidad	5	3	2
Total, puntos	14	12	9

Fuente: Autores

3.7.2 Sensor de Radiación UV

En la selección del sensor destinado para la medición la radiación uv, este debe contar con los rangos espectrales básicos, por lo que se realiza una comparación con tres tipos de sensores que mida los UVA y UVB el cual es peligroso para la vida y la salud humana como se muestra la tabla 2

Tabla 3. Características de medición sensor de radiación UV

Características	ML8511 280-390nm	GUVA-S12SD 200-370 nm	VEML6070 320-410NM
Rango de medición	5	4	5
Costo	5	4	2
Disponibilidad	5	3	2
Total, Puntos	15	11	9

Fuente: Autores

3.7.3 Sensor de temperatura y humedad

En la tabla 4 muestra las características indicadas para la medición de humedad, este debe contar con el entorno de operación y tiempo de respuesta básicos, por lo que se realiza una comparación con tres tipos de sensores.

Tabla 4. Características de medición sensor de Humedad

Características	AHT21 I2C 0 ~ 100% HR	AM201 0 – 99.9%RH	HTU31D 0-100% RH
Rango de medición	5	4	5
Costo	2	5	2
Disponibilidad	1	5	2
Total	8	14	9

Fuente: Autores

3.8 Acondicionamientos de señales en MCU (ESP 32)

3.8.1 ADC en MCU anemómetro

El principio de funcionamiento de este sensor se basa en taco dinamo el cual genera un voltaje inducido de 0 a 2V.

Como indica la figura 26 se define las variables y pin analógico de la MCU, de acuerdo a la hoja técnica del fabricante se declaró como variable entera a un valor de 0.25 que esto corresponder a la multiplicación con el voltaje inducido del pin analógico, obteniendo un resultado en m/s.

Figura 26. Variables del Anemómetro

```
#include <WiFiMulti.h>
#include <InfluxDbClient.h>
#include <InfluxDbCloud.h>
#include <ArduinoJson.h>
#include <HttpClient.h>
#include <TelnetStream.h>
#include <TimeLib.h>

#define ANEMOMETER_PIN 36
#define VOLTAGE_MULTIPLIER 0.25
```

Fuente: Autores

La figura 27 muestra la configuración de la programación del anemómetro pueda ser leído desde el pin analógico en donde la función `analogRead` recupera el valor analógico entre 0 y 1023, que representa la magnitud de la variable media de 0.25, después se realiza la multiplicación por la relación entre el voltaje máximo posible (5.0v) y el valor de `analogRead`, el valor (1023), se vuelve en el valor de lectura de voltaje proporcional entre 0 y 5.0v, por último se realiza una multiplicación por 1000.0 con el objetivo de ajustar la escala de voltaje a la escala de entrada de a anemómetro que es en función de los milivoltios

Figura 27. Cálculos de velocidad



```
anemometro

void loop() {
  anemometer.clearFields();

  // Voltaje del anemómetro
  float voltage = analogRead(ANEMOMETER_PIN) * (5.0 / 1023.0) * 1000.0;

  // Calcula la velocidad del viento en m/s
  float velocidadViento = voltage * VOLTAGE_MULTIPLIER;

  anemometer.addField("velocidad", velocidadViento);
}
```

Fuente: Autores

3.8.2 ADC en MCU radiación UV

Su salida entrega una señal analógica la cual depende de la cantidad de onda UV que detecte. Es comúnmente usado para el monitoreo de las condiciones ambientales, la longitud de onda se encuentra en rangos de 280-390nm, este rango de medición puede cubrir el espectro UV-B hasta el UV-A, la medición de la salida de la longitud se mide (mW/cm²).

La figura 28 muestra los diferentes rangos de longitud de onda que se debe considerar dentro de las condiciones del programa.

UVC: la cual se encuentra en una oscilación de 200 -280 nm, los cuales son rayos que la capa de Ozono puede ser absorbida antes de llegar a la tierra

UVB: la cual se encuentra en una oscilación de 280- 320 nm, los cuales son rayos que la capa de Ozono puede ser absorbida antes de llegar a la tierra

UVA: la cual se encuentra en una oscilación de 320-400nm, los cuales son rayos que la capa de Ozono puede ser absorbida antes de llegar a la tierra

Figura 28. Índice UV

Color	Índice	Rango	Tipo de riesgo
	0	<50	Ninguno
	1	<227	Bajo
	2	<318	
	3	<408	Moderado
	4	<503	
	5	<606	
	6	<696	Alto
	7	<795	
	8	<881	Muy alto
	9	<976	
	10	<1079	
	>= 11	<1170	Extremadamente alto

Fuente: (nagibvalejos, 2006)

Se calculó el voltaje obtenido en el pin analógico para su multiplicación con los valores de las variables mínimo que va desde 0 y máximo 1023 por el índice UV, luego estos valores se los divide por los voltajes mínimo de 1.0 y de máximo de 3.0 y se obtiene el valor del índice UV, lo que indica el valor del índice UV, donde se ajusta el rango en una escala de 0 a 11.

Figura 29. Ajuste de valores de índice UV

```
void clasificarIntensidadUV(float intensidadUV) {
    String nivelIntensidad;

    if (intensidadUV < 50) {
        indice = 0;
        Serial.println("Ninguno");
    } else if (intensidadUV < 227) {
        indice = 1;
        Serial.println("Bajo");
    } else if (intensidadUV < 318) {
        indice = 2;
        Serial.println("Bajo");
    } else if (intensidadUV < 408) {
        indice = 3;
        Serial.println("Moderado");
    } else if (intensidadUV < 503) {
        indice = 4;
        Serial.println("Moderado");
    } else if (intensidadUV < 606) {
        indice = 5;
        Serial.println("Moderado");
    } else if (intensidadUV < 696) {
        indice = 6;
    }
}
```

Fuente: Autores

3.8.3 ADC en MCU temperatura y humedad

el módulo AM2301 es un sensor digital que no necesita ser calibrado, los rangos de humedad entre 0 a 99.9%RH y de temperatura -40 a 80°C, y los rangos de precisión en humedad son $\pm 3\%$ RH y temperatura $\pm 0.5^\circ\text{C}$.

La figura 30 indica como se configuración de la librería DHT22 y la conversión de los valores de temperatura y humedad en una variable flotante, para luego visualizarlos en una cadena Sting.

Figura 30. Definición del módulo DHT22

```
#include <WiFiMulti.h>
#include <InfluxDbClient.h>
#include <InfluxDbCloud.h>
#include <DHT.h>
#include <ArduinoJson.h>
#include <HTTPClient.h>
#include <TelnetStream.h>
#include <TimeLib.h>

#define DHT_PIN 2
#define DHT_TYPE DHT22
DHT dht(DHT_PIN, DHT_TYPE);
float temperatura = dht.readTemperature();
float humedad = dht.readHumidity();
```

Fuente: Autores

3.9 Visualización gráfica

3.9.1 Comunicación de InfluxDB con Grafana

La figura 31 muestra la comunicación entre las plataformas InfluxDB y Grafana por medio de los protocolos HTTP/HTTPS, por medio de una API permite que Grafana acceda a los datos almacenados en la nube, para esto se configuró el data source donde el lenguaje consulta por Flujo si se puede incluir el URL por medio de HTTP.

Se especifica las credenciales de la base de datos las cuales son, la organización y el Token que corresponde a la clave de seguridad de los Buckets creados para almacenar los datos.

Figura 31. Selección de la base de datos



Fuente: Autores

Con la verificación el sistema, se puede acceder a los datos almacenados, Para la visualización y creación de un dashboards con los datos vinculados.

La Figura 32 muestra cómo gestionar la visualización de la data source la cual está configurado a una conexión externa de la base de datos, donde se especifica el nombre del Bucket.

Figura 32. Gestión de los datos



Fuente: Autores

3.9.2 COMUNICACIÓN DE PYTHON CON THINGSPEAK

En la figura 33 indica la visualización de un Bróker en línea para mayor monitoreo de los datos provenientes de las Esp32.

Para la recopilación y análisis de datos, se implementaron canales de comunicación dedicados, diseñados específicamente para visualización de la información generada por cada sensor. Estos canales facilitan la transmisión y presentación de datos en tiempo real, permitiendo una supervisión continua, toma de decisiones e históricos.

Figura 33. Canales en Thingspeak

Name	Created	Updated
Sensor de Temperatura y Humedad <small>Private Public Settings Sharing API Key Data Import / Export</small>	2023-11-17	2024-01-03 07:09
ANEMOMETRO <small>Private Public Settings Sharing API Key Data Import / Export</small>	2023-12-13	2024-01-13 21:59
Sensor uv <small>Private Public Settings Sharing API Key Data Import / Export</small>	2024-01-03	2024-01-03 06:57

Fuente: Autores

La figura 34 muestra la configuración de cada canal, en donde se detallan cuántas variables se visualizarán, ya que este número depende directamente de las dos variables que provienen de cada sensor. Esta personalización en la cantidad de variables permite que un usuario se adapte de mejor manera y así una interpretación más clara, detallada y eficaz.

Figura 34. Variables de los Canales

Channel Settings

Percentage Complete 50%

Channel ID 2348036

Name Sensor de Temperatura y Humedad

Description Datos de Temperatura , Humedad

Field 1 Temperatura

Field 2 Humedad

Field 3 Viento

Fuente: Autores

La figura 35 muestra el acceso al canal creado, es necesario seguir un proceso que implica adoptar las credenciales de API Key. Esta clave API proporcionan la autenticación necesaria para garantizar un acceso seguro y controlado al canal de comunicación. Al adoptar esta credencial, se establece una conexión segura que permite la transferencia confiable de datos entre los sensores y la plataforma de visualización.

Figura 35. Api keys

The screenshot shows the ThingSpeak web interface. At the top, there is a navigation bar with the ThingSpeak logo and menu items: Channels, Apps, Devices, and Support. Below this, the channel name 'Sensor de Temperatura y Humedad' is displayed. Underneath, the channel ID '2348036' is highlighted with a red box, along with the author 'mwa0000032132566' and 'Access: Private'. A breadcrumb trail shows 'Datos de Temperatura , Humedad'. A horizontal menu contains options: Private View, Public View, Channel Settings, Sharing, API Keys (which is selected), and Data Import / Export. The main content area is titled 'Write API Key' and features a text input field for the 'Key' containing '3JMKH8RSYOKS3BHZ', also highlighted with a red box. Below the input is a yellow button labeled 'Generate New Write API Key'. To the right, there is a 'Help' section with the text 'API keys enable you to keys are auto-generati' and an 'API Keys Sett' section with two bullet points: 'Write API Key: Us been compromis' and 'Read API Keys: U'.

Fuente: Autores

En la figura 36 muestra los datos recopilados de los dispositivos MCU se transmiten desde el servidor HTTP de Python para su almacenamiento en los canales de Thingspeak. Esta conexión establecida permite una transferencia de la información obtenida, facilitando su registro y organización en los canales específicos de Thingspeak.

Figura 36. Enviar datos a Thingspeak

```
# Verifica si la solicitud a ThingSpeak fue exitosa
if respuesta_thingspeak.status_code == 200:
    print("Datos enviados a ThingSpeak exitosamente.")
else:
    print("Error al enviar datos a ThingSpeak.")
```

Fuente: Autores

Dentro de Python, se implementa un arreglo para verificar si los datos han sido enviados o no. Este arreglo proporciona un mecanismo efectivo para realizar un seguimiento del estado de la transmisión de datos, permitiendo una gestión más precisa y la toma de decisiones basada en la confirmación de envío. (ver figura 37)

Figura 37. Verifica la solicitud a Thingspeak

```
# Verifica si la solicitud a ThingSpeak fue exitosa
if respuesta_thingspeak.status_code == 200:
    print("Datos enviados a ThingSpeak exitosamente.")
else:
    print("Error al enviar datos a ThingSpeak.")
```

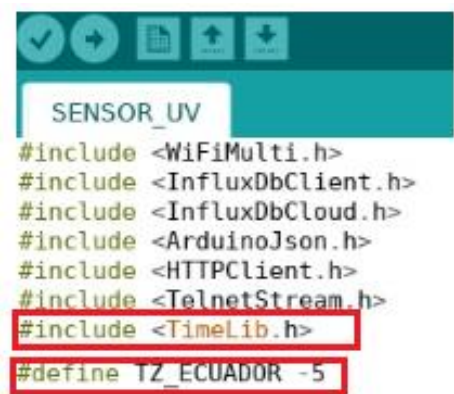
Fuente: Autores

3.9.3 Protocolo telnet

El protocolo Telnet posibilita la visualización y registro de datos en una hoja, asegurando la sincronización precisa de fecha y hora. Esta funcionalidad es esencial para la formación de un data logger, permitiendo la captura sistemática y organizada de información. La capacidad de sincronización de fecha y hora garantiza una representación precisa del tiempo en que se registraron los datos, contribuyendo a una gestión más completa y contextualizada de la información recopilada mediante este protocolo.

La figura 38 incorpora la biblioteca TimeLib y el puerto 23 que es por cual el protocolo telnet se comunica, la cual posibilita la sincronización y comunicación de fecha y hora con la zona horaria de Ecuador, sin necesidad de correcciones adicionales. Esta integración facilita la gestión precisa del tiempo en el sistema, asegurando una representación exacta y contextualizada de los datos recopilados.

Figura 38. Configuración de fecha y hora



```
SENSOR_UV
#include <WiFiMulti.h>
#include <InfluxDbClient.h>
#include <InfluxDbCloud.h>
#include <ArduinoJson.h>
#include <HTTPClient.h>
#include <TelnetStream.h>
#include <TimeLib.h>
#define TZ_ECUADOR -5
```

Fuente: Autores

La figura 39 indica la configuración que restringe la recepción de datos adicionales a través del acceso Telnet; en este caso, se limita la transmisión exclusivamente a la fecha y la hora, junto con los datos proporcionados por cada sensor. Esta parametrización permite la transmisión de la información de interés, simplificando el proceso de manejo de datos desde el acceso Telnet.

Figura 39. Envíos de datos por telnet

```
if (TelnetStream.available()) {
  TelnetStream.print("Fecha : ");
  TelnetStream.print(fecha);
  TelnetStream.print(", Hora : ");
  TelnetStream.print(hora);
  TelnetStream.print(", Intensidad UV: ");
  TelnetStream.print(intensidadUV);
  TelnetStream.print(", Nivel de Intensidad UV: ");
  TelnetStream.println(obtenerNivelIntensidadUV(intensidadUV));
}
```

Fuente: Autores

CAPITULO 4

4 Análisis y resultados

La transmisión de la data meteorológica se lo hace por dos vías tal como se había mencionado en la sección 3 de metodología, en este aspecto también se menciona que la infraestructura de red que se encuentra armada la arquitectura presenta los parámetros que indica la tabla 5.

Tabla 5. Parámetros Red U.P.S

Parámetros	Red WIFI para estudiantes de la Universidad Politécnica Salesiana sede quito campus sur
Banda	2,4GH
Descarga	1GB
Envío de archivo	5mb
Fluctuación	5 ms

Fuente: Autores

Respecto a los parámetros de la red wifi que se presentaron en la tabla anterior y donde se ancla el sistema meteorológico se registraron métricas de data enviada, perdida y el tiempo que transcurre durante la transmisión de la información hacia la nube, para este caso de estudio se dispuso de dos vías de transmisión con diferentes alojamientos de datos, es así que en la tabla 6 se presentan el promedio de 20 muestras como resultados de la transmisión de data.

Tabla 6. Direcciones IP

IP 1-Data InfluxDB		
Datos perdidos	Datos recibidos	Tiempo
20%	80%	1057 ms
IP 2- Data en Thingspeak		
Datos perdidos	Datos recibidos	Tiempo
30%	70%	1371 ms

Fuente: Autores

Desde un servidor local (NVIDIA Jetson Nano) se envía la fuente de datos por las dos vías antes mencionadas para su visualización en dos dashboards distintas, estos dos procesos presentan un consumo dentro de rangos nominales de operación en CPU, pero una sobrecarga en la memoria RAM producto del uso de los puertos DisplayPort para la expansión de pantalla y HDMI para el uso de video, en tal situación en la tabla 6 se muestran el rendimiento del proceso de visualización del sistema meteorológico.

Tabla 7. NVIDIA rendimiento

CPU	RAM	PANTALLAS EN USO
13%	3075 MB	1
20%	3079 MB	2

Fuente: Autores

De la tabla anterior muestra cuando el sistema meteorológico opera de forma normal; sin embargo, cuando arranca el boot de inicio la CPU en los dos casos de usos de pantallas aproximadamente se duplica el rendimiento del servidor local.

Este sistema fue sometido a encuestas de análisis comparativo para que el usuario compare entre dos plataformas y visualizar cual es más didáctica para su uso, el ámbito profesional que presentan y los recursos que tiene para la descarga de data.

Tabla 8. Resultados pregunta 1

1.Cuál de las 2 interfaces genera una percepción fácil para su uso.

	Excelente	Muy bueno	Bueno	Regular	Insuficiente
Grafana	20%	10%	10%	-----	-----
Thingspeak	50%	10%	-----	-----	-----

Fuente: Autores

De los 30 encuestados en la tabla 10 el dashboard con mayor porcentaje fue Thingspeak, ya que es una interfaz de fácil acceso y las gráficas que se presentan generan un aspecto más ilustrativo en comparación con Grafana.

Tabla 9. Resultados pregunta 2

2. Cuál de las 2 plataformas genera una percepción para un ámbito profesional.

	Excelente	Muy bueno	Bueno	Regular	Insuficiente
Grafana	80%	-----	-----	-----	-----
Thingspeak	20%	10%	-----	-----	-----

Fuente: Autores

Tabla 10. Resultados pregunta 3

3. Cuál de las dos interfaces según su criterio presenta una mayor funcionalidad respecto a la navegación, almacenamiento y recursos para la descarga de la data.

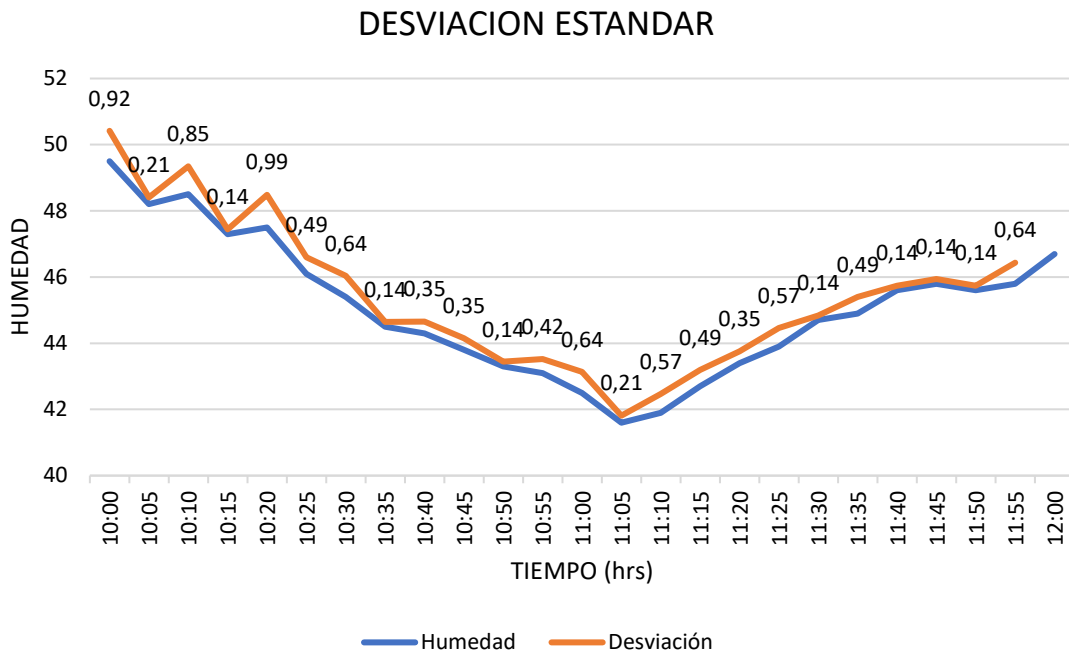
	Excelente	Muy bueno	Bueno	Regular	Insuficiente
Grafana	0%	-----	20%	10%	-----
Thingspeak	70%	-----	-----	-----	-----

Fuente: Autores

La explicación realizada a los 30 encuestados sobre la descarga de la data de cada una de las plataformas mencionadas, Thingspeak obtuvo un mayor porcentaje ya que la plataforma permite una descarga rápida de los datos recopilados.

La comparación de la data meteorológica se le realizó con los datos que se obtuvieron de la estación meteorológica de la Universidad Politécnica Salesiana sede Quito campus sur la cual cuenta con un sistema de medición por hora en donde se obtiene la desviación estándar como se muestra en la figura 40

Figura 40. Humedad del Aire



Fuente: Autores

De lo mencionado anteriormente se observa que los valores de humead tomados van en un rango de tiempo, en la cual se percibe una notable semejanza entre las dos estaciones lo cual se podría apreciar una estabilidad relativa en los niveles de humedad.

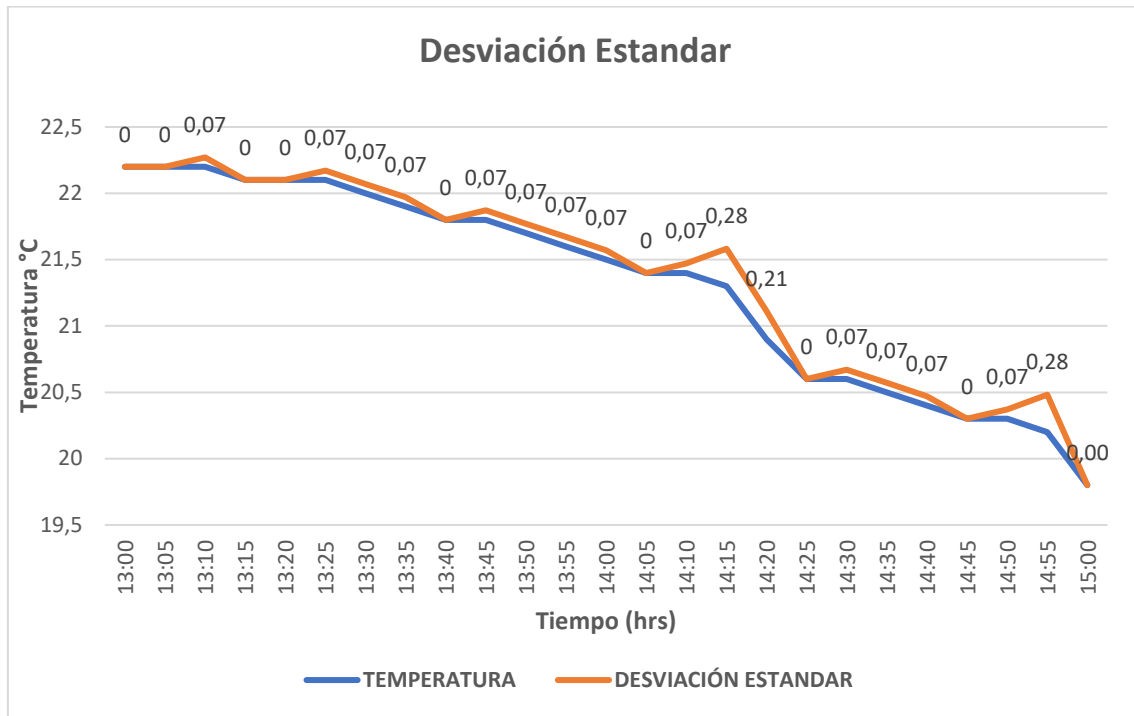
Como se muestra en la tabla 11 tanto se tomó valores de 3 horas en específico que brinda la estación de las U.P.S y se comparó con los valores de nuestra estación, dichos valores se calculó el error porcentual.

Tabla 11. Comparación humedad estación INAMHI U.P.S vs estación meteorológica

Hora	Humedad Estación INAMHI U.P.S	Humedad Estación Meteorológica	Error %
10:00	47,9	49,5	3,340292276
11:00	41,8	42,5	1,674641148
12:00	44,7	46,7	4,474272931

Fuente: Autores

Figura 41. Temperatura del Aire



Fuente: Autores

Como se visualiza en la figura 41 existe una diferencia mínima en los patrones térmicos con la toma de desviación, pero en algunas horas fluctúan de manera significativa, lo que se aprecia un cambio en las respuestas de medición.

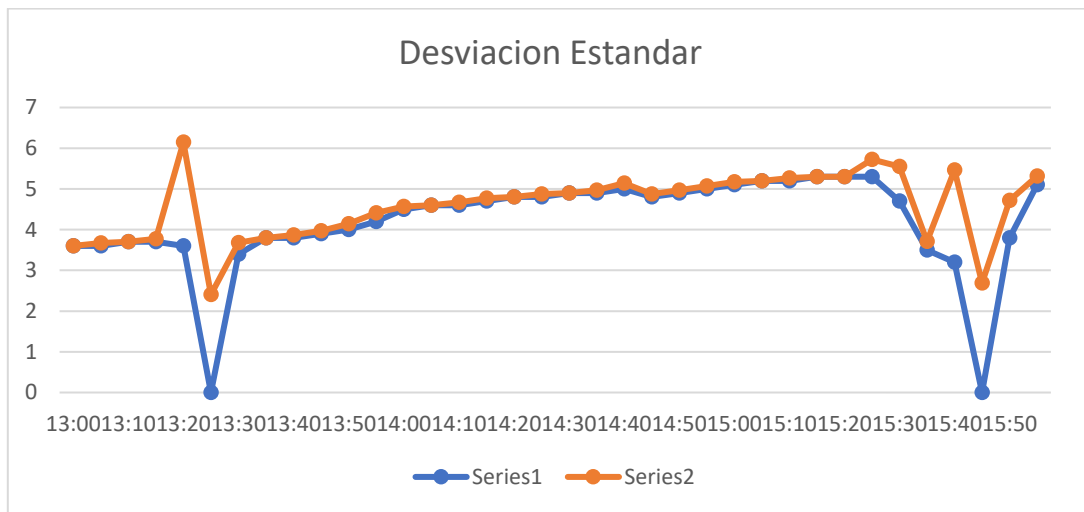
Como se muestra en la tabla 12 se tomó los tiempos de medición de cada hora que nos brinda la estación de la U.P.S, comparando así con la temperatura medida por parte de nuestra estación. Por lo que se calculó el error porcentual de cada una de ellas.

Tabla 12. Comparación temperatura estación INAMHI U.P.S vs estación meteorológica

Hora	Temperatura U.P.S	Temperatura mediada	Error %
13:00	21	22,2	5,714285714
14:00	20,9	21,3	1,913875598
15:00	19,2	19,8	3,125

Fuente: Autores

Figura 42. Velocidad del viento



Fuente: Autores

Como se aprecia en la figura 42 los valores de la desviación estándar de la velocidad del viento registran fluctuaciones notables, manifestados en los pulsos bajos de medición en condiciones atmosféricas cambiantes durante periodos de medición.

Como se muestra en la tabla 13 se comparó los valores de la estación controlada por el INAMHI que se encuentra en la U.P.S y se comparó con los valores obtenidos por parte de nuestra estación, dichos valores se realizó el cálculo del error porcentual.

Tabla 13. Comparación velocidad de viento estación INAMHI U.P.S vs estación meteorológica

Hora	Velocidad U.P.S (m/s)	Velocidad mediada (m/s)	Error %
13:00	3,1	1,2	-61,29032258
14:00	3,8	2,9	-23,68421053
15:00	5,4	3,9	-27,77777778
16:00	5,7	4,8	-15,78947368

Fuente: Autores

Como se muestra en la tabla 14 se comparó los valores de la estación controlada por el INAMHI que se encuentra en la U.P.S y se comparó con los valores obtenidos por parte de nuestra estación, dichos valores se realizó el cálculo del error porcentual.

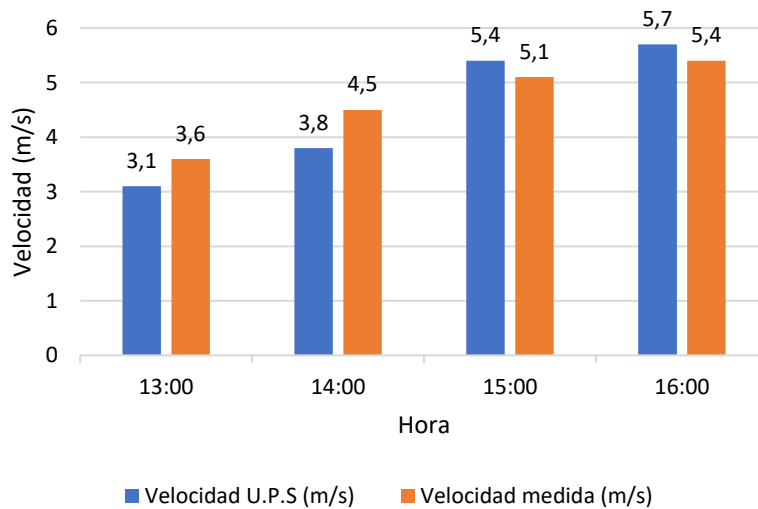
Tabla 14. Comparación velocidad de viento estación INAMHI U.P.S vs estación meteorológica

Hora	Indice UV U.P.S	Indice UV medido	Error %
10:00	5	5	0,00
11:00	7	8	14,29
12:00	11	11	0,00
13:00	10	9	-10,00

Fuente: Autores

Como se aprecia en la figura 43 los valores del cálculo de error del sensor de radiación registran valores similares, manifestados la radiación sola en condiciones atmosféricas cambiantes durante periodos de medición

Figura 43. Redición solar



Fuente: Autore

5 CRONOGRAMA Y PRESUPUESTO

5.1 Cronograma de actividades

Tabla 15. Cronograma

#	Actividades	Octubre				Noviembre				Diciembre				Enero			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Diseño de estructuras para pantallas y sensores	■	■														
2	Investigación de los tipos de variables a medir para la estación meteorológica	■	■	■													
3	Selección de los tipos de sensores			■													
4	Instalación de sistema operativo (micro SD) en tarjeta NVIDIA			■	■												
5	Ensamblaje de pantallas en estructura diseñada			■	■	■											
6	Ensamblaje de sensores en estructura diseñada				■	■	■										
7	Definir los protocolos de comunicación					■	■	■									
8	Comunicación de la nube con plataformas de programación seleccionadas						■	■	■	■							
9	Programación de los sensores								■	■	■	■					
10	Comunicación de InfluxDB con Grafana											■	■				
11	Comunicación de Python con Thingspeak.											■	■	■	■		
12	Registro de datos utilizando el protocolo Telnet.													■	■	■	
13	Visualización de datos meteorológicos recopilados por la base de datos														■	■	■
14	Realización de pruebas experimentales para evaluar el rendimiento y precisión del sistema meteorológico													■	■	■	■
15	Revisión de avance con tutor	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
16	Redacción del proyecto técnico	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Fuente: Autores

5.2 Presupuesto

Tabla 16. Presupuesto

Presupuesto						
Financiamiento	Recursos	Descripción	Valor	Cantidad	Total	
Rubros	Servicios Técnicos	Diseño de las estructuras	75%	2	150\$	
	Equipos	Sistema embebido Jetson Nano	0	0	0	
	Materiales	ESP32		12\$	3	36\$
		Sensor de temperatura y humedad		9\$	1	9
		Sensor de radiación		12.50\$	1	12.50\$
		Anemómetro		40\$	1	40\$
		Splitter Vga		19.26\$	1	19.26\$
		Adaptador HDMI a VGA		11.50\$	1	11.50\$
		Pantallas Recicladadas		0	4	0
		Software Cad & Software programación		0	0	0
Bolsa	Otros	Microcontroladores	5\$	4	20\$	
Total del Proyecto					289.26	

Fuente: Autores

6 CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

Las funcionalidades que caracteriza a un sistema meteorológico es la capacidad de almacenamiento de información, en tal virtud se dispuso de un sistema IoT con dos plataformas en vías distintas (direcciones IP) para el envío de datos hacia el almacenamiento; para esto se dispuso de InfluxDB y Thingspeak, siendo este último el que presentó un incremento en la latencia del 30% respecto al primero tras varias pruebas realizadas.

Otro parámetro que se consideró dentro sistema meteorológico fue la interfaz de monitoreo el cual está constituido por dos pantallas que se expanden por DisplayPort y se visualizan por HDMI, por tal aspecto el rendimiento de la tarjeta de desarrollo que permite esta acción, también opera como servidor receptando y enviando la data a la Cloud, esta actividad se refleja en el rendimiento de la CPU trabajando durante estado estable a temperaturas nominales de operación, sin embargo cuando se genera el arranque el booteo inicial demora 20 segundos, tiempo en el cual el rendimiento de la CPU consume un 92 % adicional y su temperatura alcanza valores de sobrecalentamiento forzando su ventilación, dato a considerar cuando se disponga de un arreglo de pantallas para mejorar el aspecto de la interfaz de monitoreo y el número de DisplayPort que disponga la tarjeta.

Para el análisis y monitoreo de la data meteorológica, se usó dos plataformas (Grafana y Thingspeak) las cuales permiten realizar dashboards para la visualización de los datos recopilados. Por lo que se generó una encuesta a 30 usuarios, obteniendo como resultado a Grafana con un 60% ya que esta permite generar un auto refrescado de la página, dashboard completo y almacena una mayor cantidad de datos, pero no genera un aspecto ilustrativo según los usuarios encuestados. Por otro lado, Thingspeak obtuvo un 70% de resultado a favor en la descarga de los datos recopilados generándolo como .CVS para una toma de decisiones inmediata.

Se realizó una comparación entre la estación meteorológica de la Universidad Politécnica Salesiana con el prototipo realizado ya que este nos brinda un dato cada hora, con los resultados obtenidos de las variables a medir se obtuvo el rango de error de cada una de las variables, en humedad obtuvimos un porcentaje de error del 3.14% , temperatura 3,58% y velocidad del viento del 32.13% ya que los valores recopilados se asemejan a los de la estación meteorológica controlada por el INAMHI lo que indica que los resultados son estables.

6.2 Recomendaciones

- Realizar las calibraciones necesarias a los sensores que se van a emplear en la estación meteorológica y así obtener medidas precisas para la recopilación de datos
- Como aplicaciones de investigación futuras se puede implementar sensores con mayor precisión de medición

REFERENCIAS

- aelectronics. (s.f.). *aelectronics*. Obtenido de Sensor de humedad y temperatura AM2301: <https://aelectronics.com.mx/metepec/sensores/51-sensor-de-humedad-y-temperatura-am2301.html>
- AliExpress. (2023). *AliExpress*. Obtenido de Sensor de velocidad del viento anemómetro Medición de velocidad del viento Sensor de viento generación de energía: <https://es.aliexpress.com/i/1005002472107422.html>
- Bala, P. (6 de Septiembre de 2021). *Geekflare*. Obtenido de <https://geekflare.com/es/python-web-servers/>
- Centro Nacional de Salud Ambiental. (13 de Julio de 2023). *Centro Nacional de Salud Ambiental*. Obtenido de Radiación UV: <https://www.cdc.gov/spanish/nceh/especiales/radiacionuv/index.html>
- EAP. (2 de Febrero de 2023). *EAP*. Obtenido de Escala de indice UV: <https://espanol.epa.gov/espanol/escala-del-indice-uv>
- Festo. (2010). *Festo-Didactic*. Obtenido de <https://www.festo-didactic.com/es-es/productos/automatizacion-de-fabricas-e-industria-4.0/modulos-de-la-fabrica-para-la-ensenanza/estaciones-mps>
- Hogar Seco. (2021). *Hogar Seco*. Obtenido de Humedad relativa del ambiente - Como medirla: <https://www.hogarseco.com/humedad-relativa-del-ambiente-como-medirla/>
- influxdata. (s.f.). *influxdata*. Obtenido de Nube InfluxDB: <https://www.influxdata.com/products/influxdb-cloud/>
- METEOCULTURA - ESTACIONES METEOROLÓGICAS & METEOROLOGÍA. (2023). Obtenido de ¿QUÉ ES UNA ESTACIÓN METEOROLÓGICA?: <https://estaciondemeteorologia.com/que-es-una-estacion-meteorologica/>
- minibots. (15 de Enero de 2021). *Notas sobre robótica, domótica, sistemas operativos y programación*. Obtenido de <https://minibots.wordpress.com/2023/07/28/instalacion-de-linux-en-nvidia-jetson-nano/>
- nagibvalejos. (12 de Marzo de 2006). *Robotics-Space-NV*. Obtenido de <https://github.com/nagibvalejos/Robotics-Space-NV/blob/master/Arduino%20I/Sesi%C3%B3n%2014/S14.pdf>
- NVIDIA . (s.f.). *NVIDIA DEVELOPER*. Obtenido de Kit de desarrollador Jetson Nano: <https://developer.nvidia.com/embedded/jetson-nano-developer-kit#>
- NVIDIA. (12 de 09 de 2019). *NVIDIA Developer*. Obtenido de <https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit#>
- Plexilab. (2021). *Plexilab*. Obtenido de Módulo sensor de luz ultravioleta UV GY-ML8511 GY-8511 para Arduino: <https://www.plexylab.com/product-page/m%C3%B3dulo-sensor-de-luz-ultravioleta-uv-gy-ml8511-gy-8511-para-arduino>

TEKmikro. (2023). *TEKmikro*. Obtenido de MODULO ESP32 WIFI BLUETOOTH:
<https://tecmikro.com/modulos-shields/630-modulo-esp32-wifi-bluetooth.html>