

## Monitoreo de variables atmosféricas y percepción remota

Hermes Moreno-Álvarez<sup>1</sup>, Juan Flores-García<sup>2</sup>,  
Jesús Eduardo González<sup>2</sup>, Carmen Santiago-Díaz<sup>3</sup>,  
Gustavo Rubín-Linares<sup>3</sup>, Catalina Rivera-Morales<sup>4</sup>,  
Vanessa Baeza-Olivas<sup>1</sup>, Xochitl de Jesus-Rojas<sup>3</sup>

<sup>1</sup> Universidad Autónoma de Chihuahua,  
Departamento de Ingeniería Aeroespacial,  
México

<sup>2</sup> Tecnológico Nacional de México,  
Instituto Tecnológico Superior de Poza Rica,  
México

<sup>3</sup> Benemérita Universidad Autónoma de Puebla,  
Facultad de Ciencias de la Computación,  
México

<sup>4</sup> Benemérita Universidad Autónoma de Puebla,  
Facultad de Ingeniería Química,  
México

{hm1713a, jesus95hidalgo}@gmail.com, {hmoreno,  
vbaeza}@uach.mx, juan.flores@itspozarica.edu.mx,  
{marycarmen.santiago, gustavo.rubin,  
maria.riveram}@correo.buap.mx, xochitl.de@alumno.buap.mx

**Resumen.** En el presente trabajo se integran en una plataforma de bajo costo los diferentes subsistemas satelitales concluyendo la implementación y pruebas de funcionamiento. Una de las principales tareas es reportar la información proporcionada por los sensores que miden las variables ambientales en tiempo real, tales como la presión, altura, temperatura y humedad, de manera simultánea la información de la trayectoria que toma durante el ascenso y descenso de estos sistemas considerando una elevación mínima de 1000m con un dispositivo externo. El objetivo del diseño de este prototipo es ahorrar tiempo en la elaboración de prototipos espaciales educativos, en donde por este medio podrán verificar que sensores utilizar, observar el diseño electrónico, el microcontrolador, las plataformas y la programación utilizada para el desarrollo de prototipos futuros que ayuden al análisis de los datos atmosféricos. Sin embargo, también se pretende aportar en otras disciplinas con los diferentes subsistemas creados en el prototipo los cuales pueden tener diferentes aplicaciones.

**Palabras clave:** Subsistemas, satelital, sensores, ambiental, variables, programación, implementación, plataforma.

## **Monitoring of atmospheric variables and remote sensing**

**Abstract.** In the present work, the different satellite subsystems are integrated into a low-cost platform, concluding the implementation and performance tests. One of the main tasks is to report the information provided by the sensors that measure environmental variables in real time, such as pressure, height, temperature and humidity, simultaneously with information on the trajectory it takes during the ascent and descent of these systems considering a minimum elevation of 1000m with an external device. The objective of the design of this prototype is to save time in the elaboration of educational spatial prototypes, where by this means they will be able to verify which sensors to use, observe the electronic design, the microcontroller, the platforms and the programming used for the development of future prototypes that assist in the analysis of atmospheric data. However, it is also intended to contribute to other disciplines with the different subsystems created in the prototype, which may have different applications.

**Keywords:** Subsystems, satellite, sensors, environmental, variables, programming, implementation, platform.

### **1. Introducción**

El uso de plataformas anfitrionas con sensores como parte de un sistema conformado por elementos que representan los subsistemas satelitales cada vez más intensivos en datos, impulsan la necesidad de una capacidad de enlace descendente de gastos de mayor velocidad y eficiencia energética.

Los CubeSats, el estándar dominante para los nanosatélites, tienen un tamaño, peso y potencia limitados, lo que dificulta la instalación de antenas de radiofrecuencia (RF) de alta ganancia. Esto empuja a la mayoría de las misiones de CubeSat con necesidades de alta velocidad de datos a utilizar estaciones terrestres con grandes aperturas de alta ganancia (diámetros de plato que oscilan entre 5 y 20 m).

También existen desafíos regulatorios para obtener licencias de RF con un ancho de banda sustancial para misiones CubeSat; incluso la gestión de las numerosas solicitudes de licencias de ancho de banda estrecho se ha convertido en un problema. Una alternativa a las comunicaciones de RF tradicionales es deseable para CubeSats.

Actualmente las cosas han cambiado, ya que conforme ha avanzado la tecnología, y la estandarización, ha bajado el costo de los componentes utilizados para este tipo de misiones, dando oportunidad a estudiantes y empresas que con un menor presupuesto tienen el alcance de realizar sus propias investigaciones [1].

Una alternativa que ha ayudado a realizar misiones espaciales, sobre todo a los estudiantes, son los llamados Cubesat, se trata de sistemas que representan los elementos que componen un satélite de pequeñas dimensiones, son más baratos en cuanto a materiales, su construcción y sus lanzamientos [2]. Debido a sus características como su peso, su geometría y carga útil, estos dispositivos suelen ser menos complejos

que los satélites comerciales. Sin embargo, eso no los exime de tener un gran potencial y un gran alcance dentro de las investigaciones [4].

Este proyecto se enfoca en realizar un prototipo de un nanosatélite con un presupuesto bajo que tenga la capacidad de comunicación bidireccional con un centro de control en el cual se puedan monitorear algunas variables de la atmósfera. Se puedan analizar las variables y gestionar los datos obtenidos. A través de esta información se podría obtener un análisis concreto acerca de algunas variables atmosféricas.

Para lograr este cometido, primero se tendrá que seleccionar y probar cada uno de los elementos, integrarlos con un microcontrolador para obtener el conjunto de variables que se requieren, para posteriormente realizar la comunicación bidireccional del prototipo con una estación que tenga acceso en tiempo real de los datos que se quieren obtener. Finalmente, se obtendrán resultados de su capacidad en la obtención de datos, visualización de los gráficos de los datos obtenidos, tiempo de transmisión y costos.

## 2. Marco teórico

Subsistemas que conforman los satélites “Un satélite está dividido principalmente en dos partes, la plataforma y la carga útil, la carga útil es la encargada de realizar la misión del satélite y la plataforma es la encargada de darle soporte al funcionamiento del satélite, como energía, posicionamiento, comunicaciones, trabajo a determinados niveles de temperatura, entre otros.

De este modo la plataforma, es el conjunto de subsistemas que dependen uno del otro para lograr el correcto funcionamiento del satélite” [7]. El subsistema de control de orientación es un sistema muy importante para el satélite, consta de componentes que le ayudan a corregir su posición, en dado caso de que alguna antena no esté apuntando para el lugar correcto.

Esta corrección se realiza mediante propulsores utilizando un sensor giroscopio como referencia o un GPS. El satélite puede corregir su posición mediante algoritmos preestablecidos, el cual le da un alto grado de autonomía.

Para el subsistema de energía está constituido por paneles solares como la fuente principal de energía, además de que cuenta con baterías que almacenan la energía que se obtiene de los paneles y sirven para dar inicio a las operaciones un instante después de ser lanzado del vehículo que lo transporta antes de que los paneles solares empiecen a alimentar al satélite.

Las baterías reabastecen cuando no hay suficientes rayos del sol para alimentar todo el satélite. En el caso del *subsistema de telemetría* se da a conocer el estado de los demás subsistemas, en donde mediante sensores se miden variables vitales del satélite como es: voltaje, corriente, temperatura, presión, entre otras variables más. Estas mediciones se envían a través de canal de comunicación espacial para informar a la base lo que ocurre con el satélite.

Otro de los subsistemas a considerar es el de telemetría y comando, este nos permite mandar instrucciones desde la base de control a los subsistemas del satélite, a través de una línea de comunicación, las instrucciones son codificadas para que solo la base tenga el control del satélite y ningún otro equipo pueda acceder al control del satélite.

```
#define SSID "REDWIFISAT"
#define SSID_PASSWORD "PassWordSat"

#define USERNAME "Usuario_sat"
#define DEVICE_ID "prototipo"
#define DEVICE_CREDENTIAL "163928"
```

Fig. 3. Líneas de código de configuración de usuario, ID del dispositivo, credenciales del dispositivo y la configuración de la red.

```
thing["dht22"] >> [] (pson& out){
out["Temperatura"] = dht.readTemperature();
out["Humedad"] = dht.readHumidity();
out["Temperature"] = bmp.readTemperature();
out["Presion"] = bmp.readPressure();
out["Altitud"] = bmp.readAltitude(1013.25);
out["AnguloX"] = AcX;
out["AnguloY"] = AcY;
out["Latitud"] = gps.location.lat();
out["Longitud"] = gps.location.lng();
```

Fig. 4. Líneas de código de la configuración de envío de variables a la plataforma.

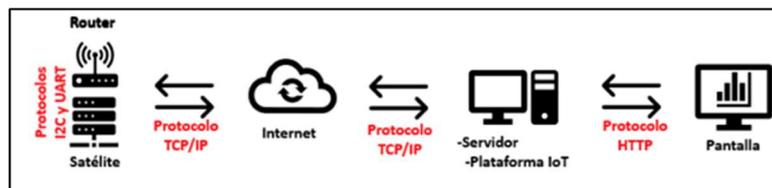


Fig. 5. Esquema de interacción de protocolos para la transferencia de datos.

En cuanto al subsistema de control térmico podemos decir que controla las temperaturas dentro de la nave, para que los componentes y sistemas trabajen en óptimo funcionamiento, ya que las temperaturas pueden ser muy extremas en el espacio exterior, sobre todo cuando hay eclipse o cuando los rayos del sol no pegan directamente al satélite. Para el control térmico, se emplean conductores de calor, radiadores y calefactores eléctricos.

### 3. Estado del arte

Los modelos denominados CanSats se han utilizado principalmente en los niveles de educación superior y especialidad, utilizando una amplia variedad de hardware y software, con algunos intentos de estandarización a través de "kits" [2, 3, 4]. Esto debido a una notable falta de material educativo que pueda ser utilizado para crear clases donde pueda aprovecharse esta tecnología, pues los modelos reales implican electrónica de grado espacial con costos que sobrepasan las posibilidades de los centros educativos.

Los cubesat en su presentación de simuladores de mesa, así como los modelos reales puestos en órbita como lo fue el AztechSat-1 se han utilizado principalmente a nivel

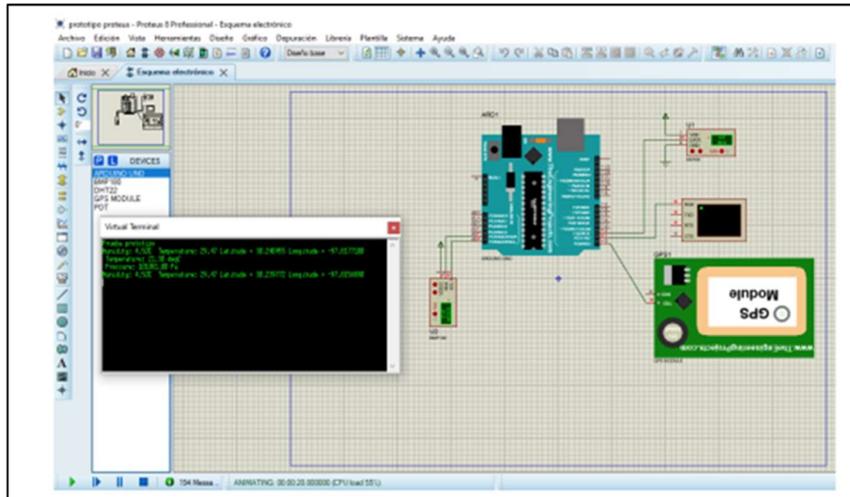


Fig. 6. Simulación de adquisición de datos a través de Proteus y la salida en la terminal virtual.

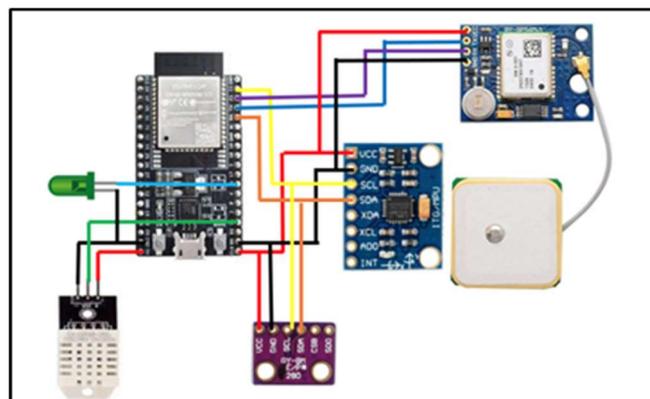
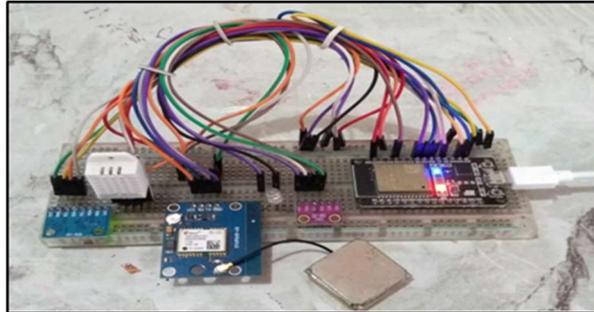


Fig. 7. Integración del circuito.

universitario teniendo una variada participación de disciplinas las cuales en mayor o menor grado están relacionadas con la tecnología espacial. Por su lado los cansats pueden incluso cubrir un sector más amplio en los niveles educativos que pueden empezar a nivel de media y media superior.

Los satélites en general poseen una estructura similar y con los cansats. Se pretende reproducir el funcionamiento de éstos subsistemas, sin embargo, las tecnologías que se utilizan son variadas esto básicamente dependiendo de los objetivos o misiones. Es importante mencionar que las aplicaciones de éstos subsistemas pueden extrapolarse a otras áreas, quienes pueden aprovechar ésta tecnología tanto en la parte de sensado como de transmisión y almacenamiento de datos.

El dispositivo que se presenta proporciona la capacidad de medir in situ diferentes variables atmosféricas, las cuales son integradas en el dispositivo, documentando en



**Fig. 8.** Prototipo armado en funcionamiento.

tiempo real los parámetros obtenidos con un volumen considerable de datos almacenados para su manipulación y bajo costo. Se realiza una descripción de los diferentes elementos utilizados en el dispositivo y el diseño del mismo tomando en cuenta los requisitos de medición. Los resultados presentados ilustran la calidad de los datos proporcionados por las mediciones realizadas de manera simultánea.

## 4. Desarrollo

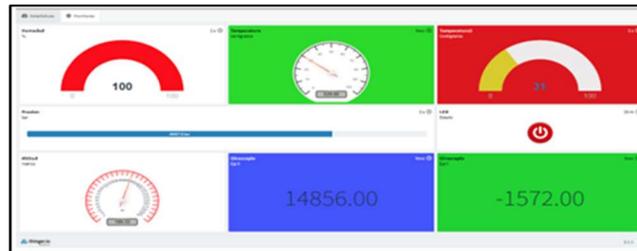
Los componentes son piezas fundamentales dentro de cada dispositivo y es importante seleccionar minuciosamente cada uno de ellos. En el caso de los sensores, puede variar el rango de medición o el voltaje de operación entre un sensor y otro, motivo por el cual, se verifica cada una de las características de los elementos que se van a integrar. Enumerando a continuación algunas de las características de los componentes a utilizar en este prototipo.

### 4.1. Sensor GY-521 MPU6050

Tiene todo lo necesario para medir movimiento en 6 grados de libertad, combinando un giroscopio de 3 ejes y un acelerómetro de 3 ejes. Trabaja con un voltaje de operación de 3 a 5 V. Puede medir las aceleraciones en los rangos de 2g/4g/8g/16g, además cuenta con giroscopio, cuyos rangos son: 250Grad/Seg, 500Grad/Seg, 1000Grad/Seg, 2000Grad/Seg, con sensibilidad de giroscopio de 131 LSBs/dps, interface de I2C, conversor AD de 16 bits (salida digital). Sus dimensiones son de 19x15x5 mm y tiene un peso de 15 gramos.

### 4.2. Cámara ESP32 CAM módulo

Esta cámara opera con un voltaje de 5 volts, cuenta con módulo de wi-fi 802.11b/g/n, con un tipo de cámara OV2640 2MP, cuenta con CPU de 32 bits de doble núcleo de baja potencia, frecuencia principal de 240 MHz, potencia informática de hasta 600 DMIPS, velocidad de reloj de hasta 160 MHz. Incorpora SRAM 520 Kb, 4MPSRAM externa, soporta interfaces UART/SPI/I2C/PWM/ADC/DAC, soporta cámaras OV2640/OV7670, flash incorporado, soporta tarjetas TF micro SD (máximo 4 GB),



**Fig. 9.** Pantalla de visualización de las variables atmosféricas humedad, temperatura, presión barométrica, altitud, nivel en el eje x y y, medición del giroscopio, sin unidades.



**Fig. 10.** Pantalla de visualización que muestra el comportamiento de las variables, temperatura, altitud, presión, humedad y nivel del prototipo con respecto al tiempo y la ubicación en tiempo real.

compatible con modos de operación STA/AP/STA+AP, con antena PCB, cuenta con conectores u.FL y FPC. Sus dimensiones son de 27x40.5x4.5 mm y tiene un peso de 20 gramos. En la figura se muestra el aspecto físico del sensor con sus entradas y salidas del mismo.

## 5. Prueba de funcionamiento de componentes

Se realizaron las pruebas con los sensores a través del gestor del IDE de Arduino, verificando la funcionalidad de cada uno de los sensores, considerando que en el prototipo final se utilizará el microcontrolador ESP32.

### 5.1. Programación

Una vez integrada la programación se cargó el código de todos los sensores en el microcontrolador ESP32, para corroborar el correcto funcionamiento del código los datos obtenidos de esta prueba se mostraban en la consola del IDE de Arduino.

Para programar el código de la comunicación se configuró la red wifi y los parámetros como el nombre del usuario, ID del dispositivo y credenciales para la comunicación con la plataforma IoT Thingier.io, como se muestra en la figura 3.

Después de unir el código de comunicación con el código que integran los sensores, se agrega el código que indica que datos se envían, como se muestra en la figura 4.

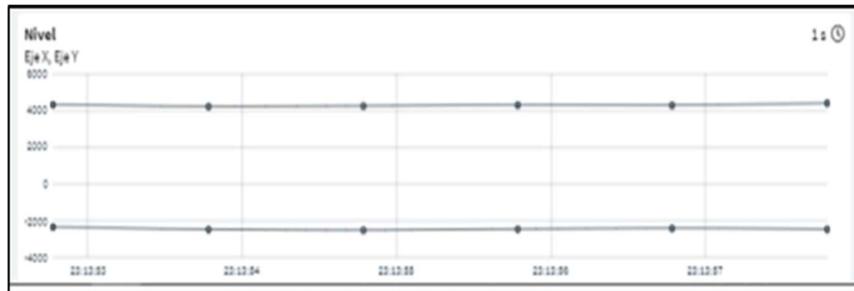


Fig. 11. Muestra el comportamiento del eje x y eje y con respecto al tiempo, cuando el prototipo se encuentra nivelado.

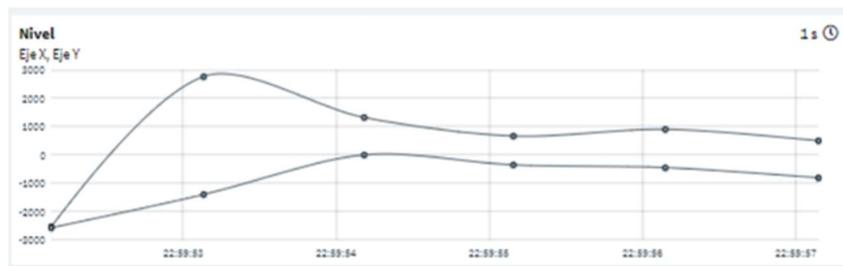


Fig. 12. Muestra el comportamiento del eje x y eje y con respecto al tiempo, cuando el prototipo ha sufrido alguna inclinación.

## 5.2. Comunicación

Se configura el dashboard con la opción From Device Resource, para indicar que la fuente del valor viene de un dispositivo, el nombre del dispositivo, el nombre del grupo de variables, la variable y el tiempo con el cual se estaría actualizando la información.

En la figura 5 se muestra cómo interactúan los protocolos de comunicación, el protocolo I2C fue utilizado para obtener los datos del sensor giroscopio y del sensor BMP280, y el protocolo UART se utilizó para obtener los datos del sensor GPS. Una vez recopilados los datos en el microcontrolador se utilizó el protocolo TCP/IP para mandar todos estos datos a través de internet a la plataforma IoT,

Thingier.io, y posteriormente se utilizó el protocolo HTTP para mostrar los datos a través de gráfico mejor conocidos como dashboards

## 6. Resultados

En la figura 6 se muestra el resultado del circuito de la simulación en el software Proteus, en donde se puede observar el sensor de temperatura, módulo de GPS y al sensor de presión y altura conectados de manera virtual a un microcontrolador Arduino UNO, que a su vez está conectada a una terminal virtual para mostrar los datos obtenidos de los sensores.

Date	Altitud	Angulo X	Angulo Y	Humedad	Latitud	Longitud	Presion
2021-08-07T22:45:53.453Z	153.2822898777544	19280	2718	84	19.8889946524219	25.4448912548727	99487.625
2021-08-07T22:44:53.281Z	153.3022499233475	19288	4934	85.5	25.4448912548727	25.4448912548727	99484.48575
2021-08-07T22:44:53.348Z	153.388493191872	19124	4812	85.400011238769	25.4448912548727	25.4448912548727	99484.51125
2021-08-07T22:44:53.352Z	154.3728191838888	19380	5038	84	25.4448912548727	25.4448912548727	99488.5825
2021-08-07T22:45:53.633Z	154.1321152479042	14888	7008	85.400011238769	0	0	99487.46175
2021-08-07T22:37:41.896Z	154.5389399121938	188	-1828	85.8889946524219	0	0	99488.4858175
2021-08-07T22:38:38.547Z	154.5889399444219	188	-1834	85.800011238769	0	0	99487.895825
2021-08-07T22:38:38.646Z	153.3022499233475	288	-1822	85.8889946524219	0	0	99489
2021-08-07T22:37:41.281Z	44838	-1	-1	Null	0	0	0
2021-08-07T22:20:01.122Z	154.8589946524219	15284	-780	85.5	25.4448912548727	25.4448912548727	99481.7138175

Fig. 13. Base de datos con los registros de las variables obtenidos a través de los sensores.

Angulo X	Humedad	Latitud	Longitud	Presion	Temperature	Temperature
2718	84	19.8889946524219	25.4448912548727	99487.625	28.800000000000004	28.170000000000004
4934	85.5	25.4448912548727	25.4448912548727	99484.48575	28.800000000000004	28.120000000000004
4812	85.400011238769	25.4448912548727	25.4448912548727	99484.51125	28.700000000000004	28.180000000000004
5038	84	25.4448912548727	25.4448912548727	99488.5825	28.700000000000004	28.200000000000004
7008	85.400011238769	0	0	99487.46175	28.700000000000004	28.198888888888888
2384	85.5	0	0	99487.34375	28.700000000000004	28.158888888888888
-1828	85.8889946524219	0	0	99488.4858175	28.700000000000004	28.148888888888888
-1834	85.800011238769	0	0	99487.895825	28.788888888888888	28.200000000000004
-1822	85.8889946524219	0	0	99489	28.700000000000004	28.158888888888888
-1	Null	0	0	Null	0	0
-780	85.5	25.4448912548727	25.4448912548727	99481.7138175	28.888888888888888	28.238888888888888

Fig. 14. Continuación de la base de datos obtenida de la plataforma Thingie.io.

En la figura 7 se muestra el esquema resultante del circuito del prototipo, realizando la integración de los componentes, respetando las entradas de los protocolos de comunicación como fue el caso de los sensores BMP280, GY-521 MPU6050 y el módulo de GPS. Todo el circuito fue alimentado con 3.3 volts que son los valores de desempeño del microcontrolador ESP32.

En la figura 8 se muestra la integración de los componentes para realizar las pruebas. En lo que respecta a la velocidad de la comunicación se obtuvieron resultados muy favorables, sobre todo con la rapidez y fluidez en la transmisión de los datos del microcontrolador con la plataforma Thingier.io, realizamos una prueba, la cual consistía en cronometrar el tiempo en que tardaba en encenderse el led, accionando el botón desde los gráficos, y se obtuvo un tiempo de 0.40 segundos con una velocidad de carga y descarga en internet de 5.03 Mbps y 8.66 Mbps respectivamente, por lo que la latencia puede variar dependiendo la velocidad del internet.

En la figura 9 se observa gráficamente el resultado de las variables interactuando en tiempo real con las opciones monitoreo y estadística, en el primer caso se muestra la variable de Humedad obtenida por el sensor DHT22, con la unidad de porcentaje en un gráfico llamado Gauge, a un lado se encuentra la variable de temperatura obtenida por el sensor DHT22, con la unidad de centígrados en un gráfico llamado Tachometer, en el cual se configuraron alarmas en el color del grafico de acuerdo a un rango en la escala establecido, también se muestra la variable de temperatura obtenida por el sensor BMP280, con la unidad de centígrados en un gráfico llamado Gauge también con alarmas, enseguida la variable de presión proveniente del sensor BMP280 con la unidad

bar en un gráfico llamado Progressbar, finalmente la variable altitud obtenida por el sensor BMP280, en metros y en un gráfico llamado Tachometer, se muestran las variables de nivel del eje x y eje y, las cuales no tienen ninguna unidad en unos gráficos llamados Text/Value, los cuales cuentan con tres alarmas, cambian de color, verde o rojo dependiendo si está o no equilibrado el eje respectivo.

La figura 10 muestra las *Estadísticas* que genera la plataforma, en donde se puede ver el comportamiento de las variables de temperatura, altitud, presión, humedad y nivel en los ejes X y Y, con respecto al tiempo, las cuales se van registrando gráficamente cada 5 segundos, así como la ubicación en tiempo real a través de un mapa, gracias a las coordenadas de latitud y longitud obtenidas del módulo de GPS.

En la figura 11 se puede observar el comportamiento del eje x y eje y con respecto al tiempo, cuando los dos ejes se encuentran nivelados, se puede apreciar que la gráfica se mantiene estable en comparación con la figura 12 la cual muestra cuando el eje x y eje y, tienen alguna inclinación o movimiento, se ve como los puntos graficados sufren un cambio significativo. Por lo que este tipo de gráficas son ideales para identificar cualquier anomalía que suceda en el prototipo.

Las figuras 11 y 12 muestran los resultados de la base de datos, que en este caso se programó para almacenar los datos de todas las variables cada minuto la cual solo puede tener 100 registros almacenados y posteriormente se va eliminando el último. En la figura 13 se muestra la base de datos con 8 columnas etiquetadas como *Date* para la fecha y hora, enseguida las correspondientes a las variables Altitud, Ángulo x, Ángulo y, Humedad, Latitud, Longitud y Presión. Finalmente, en la figura 14 se muestra la continuidad de la base de datos en las columnas con el nombre de las variables faltantes, las cuales son las dos variables de Temperatura obtenidas.

Se compararon las mediciones de los sensores en el prototipo físico con las que se midieron en la plataforma obteniéndose precisiones menores al 0.05%.

## 7. Conclusiones

Se puede hacer las siguientes conclusiones.

1. En este prototipo se integraron diferentes subsistemas electrónicos, que permitieron recopilar la información de variables atmosféricas en tiempo real a una altura de mil metros sobre el nivel del mar para simular una misión satelital de comunicación y percepción remota.
2. Se diseñó un prototipo con los subsistemas mínimos requeridos considerando las limitaciones en peso, dimensiones, transmisión de información y la creación de las bases de datos a través del software Proteus y la plataforma Thingier.io que sirvieron de apoyo para integrar dichas soluciones.
3. Se establecieron las variables atmosféricas que se miden en el prototipo, en donde se determinó que las variables de temperatura, humedad, altura, localización y el nivel en los ejes X y Y son indispensables para vigilar el comportamiento de la atmósfera.
4. Se seleccionó el microcontrolador ESP32, se debió a que tiene integrado un módulo wifi, además de que es compatible con la IDE de Arduino y sobre todo porque es compatible con la plataforma de IoT que se eligió para realizar la comunicación.

5. Se configuró la comunicación entre los diferentes protocolos de comunicaciones que los diferentes subsistemas requerían.
6. Se seleccionó la plataforma Thinger.io para una comunicación y visualización de los datos obtenidos en tiempo real, así como también el almacenamiento en la base de datos que la misma plataforma proporciona.
7. Este prototipo electrónico se convierte en una herramienta útil, no solo para el sector aeroespacial sino para múltiples áreas en donde se les puede obtener aplicaciones para el monitoreo de diferentes variables, ocupando como base información expuesta en este documento las cuales facilita una alternativa más clara de telemetría.

## **Referencias**

1. Orts-Torres, C.: Propuesta y diseño de un nano-satélite de comunicaciones para fines educativos (2021)
2. Wertz, J. R., Larson, W. J.: Space mission analysis and design. Springer Netherlands (1999)
3. Cubesat project: Cubesat Design Specification (CDS) (2013)
4. Opazo-Toro, T. I.: Requerimientos, implementación y verificación del nano-satélite Suchai (2013)
5. Chin, J., Coelho, R., Foley, J., Johnstone, A., Nugent, R., Pignatelli, D., Pignatelli, S., Powell, N., Puig-Suari, J.: Basic concepts and processes for first-time cubesat developers. NASA, pp. 96 (2017)
6. Toorian, A.: CubeSat Design Specification (2015)
7. Bohorquez Garzón, Y. S.: Diseño conceptual y preliminar de un CubeSat de bajo costo (2019)
8. Jiménez-Fernández, C. J., López-Ojeda, A., León de Mora, C.: Metodología de diseño electrónico dentro de prácticas obligatorias de laboratorio, no. 37, pp. 19–27 (2010)
9. Barrón, M.: Uso didáctico del software de ayuda al diseño electrónico PROTEUS (1970)
10. Corona, A. E.: Protocolos TCP/IP de internet (2004)
11. Luna del Águila, F.: Implementación del protocolo HTTP paralelizado en cliente y servidor (1999)
12. Sánchez-Martínez, I.: Empleo de bus 12C en el microcontrolador PIC 16F73. Universidad Central Marta Abreu de Las Villas (2008)
13. Lozano-Cruz, D., López-Jiménez, J. A., Cruz-Avilés, D., Granillo-Macías, R.: Diseño de una red de distribución mediante datos obtenidos de una interfaz de programación de aplicaciones. vol. 7, no. 13, pp. 42–48 (2020)
14. Monge De La Cruz, L. A., Torres-Herrera, J. P., López Chico, L. E., Navarro Cota, C. X.: Análisis comparativo de servidores de mapas. no. 10, pp. 1–10 (2010)