

Diseño e Implementación del front-end y back-end de una plataforma IoT para recepción, registro y visualización de los datos de telemetría y del payload del cohete sonda de combustible sólido Orión del semillero de Cohetería UIS Aeroespacial.

Laura Nathalia Pinto Campos,
Ana María Sánchez Castillo

Trabajo de grado para optar al título de
Ingeniero Electrónico

Director
Julián Gustavo Rodríguez Ferreira
Doctorado en Física, Especialidad Astrofísica

Co-Director
Jaime Guillermo Barrero Pérez
Maestría en Ingeniería Electrónica

Universidad Industrial de Santander
Facultad de Ingenierías Fisicomecánicas
Escuela de Ingenierías
Eléctrica, Electrónica y de Telecomunicaciones
Bucaramanga
2023

Dedicatoria

*Yo **Ana María Sánchez Castillo** nacida en **Duitama, Boyacá** en camino a obtener mi título profesional como ingeniera electrónica, le dedico la culminación de este proyecto a mis padres **Álvaro Ignacio Sánchez Vivas** y **Lidia Elsy Castillo Corredor** los cuales fueron y han sido los principales maestros en toda mi formación integral, personal y profesional. Les estoy eternamente agradecido por todo el esfuerzo que han hecho y las herramientas que me han brindado para que cada día pueda seguir avanzando por un camino ético y moralmente correcto. Por último, quiero dedicar la obra aquí presente a mi hermano **Luis Ignacio Sánchez Castillo**, quien me ha ayudado a alcanzar el equilibrio que me permite dar todo mi potencial. Ya que gracias a la sabiduría y experiencia que me ha compartido he podido avanzar a través de los retos que la vida me ha presentado teniendo en cuenta puntos de vista que personalmente nunca habría considerado.*

Dedicatoria

*Dedico con toda mi energía esta tesis a mis padres, **Fanny** y **Samuel**, por ser el ejemplo perfecto de esfuerzo y sacrificio, por enseñarme cosas básicas del día a día como disfrutar de los pequeños detalles, por creer en mí. Son los pilares de mi vida y para ustedes es que dedico todos mis triunfos. ¡Los amo hasta el infinito y más allá!*

*A mis hermanos por compartir sus vidas con la mía, a mi **Samuelito**, tus lindas acciones hablan por ti y me inspiran a ser mejor persona, ¡gracias universo por enlazar nuestras almas!, a **Angy**, por demostrarme que la fortaleza vence los obstáculos y se pueden lograr grandes éxitos.*

*A mis peques: **Salomé**, **Juliana** y **Samuel**, por llegar a este cosmos a iluminar mis días con su amor y alegría.*

*A mi familia por esa fuerza de unidad que nos motiva a perseverar y alcanzar los sueños, a la tía **Albita** por sus sabias palabras y lecciones, a mi prima **Anggy** por ser mi cómplice y enseñarme que la vida es bella.*

A mis amigos por cada momento vívido, por darme la motivación para no desfallecer y seguir avanzando a lo largo de esta etapa universitaria.

Laura Nathalia Pinto Campos

Agradecimientos

A nuestra alma Máter la Universidad Industrial de Santander, a nuestra escuela de Ingeniería Eléctrica, Electrónica y de telecomunicaciones, a nuestro director Dr. Julián Gustavo Rodríguez Ferreira y al codirector MSc. Jaime Guillermo Barrero Pérez por orientarnos en cada avance realizado, por ser una fuente de inspiración de disciplina y ser una guía indispensable para culminar con éxito este proyecto de investigación.

Al estudiante de ingeniería de sistemas Julián Eduardo Villamizar, a los ingenieros Ricardo Rodríguez y Brayan Aparicio, por tomarse el tiempo de brindarnos indicaciones y consejos durante el inicio y desarrollo de este proyecto. Gracias por la disponibilidad en aquellos días de arduo trabajo.

A los miembros del Semillero de Coherencia UIS Aeroespacial por brindarnos apoyo incondicional y los ánimos para alcanzar este sueño hecho realidad.

Tabla de Contenido

Introducción	16
1 Problema.	18
1.1 Definición del Problema	18
1.2 Planteamiento de la Misión	20
1.3 Estado Actual de las misiones	21
1.3.1 Validación	21
1.3.2 Transmisión de datos e interfaz gráfica	22
1.3.3 Estación Terrena	23
1.4 Este Proyecto Como Parte de Orión y Guanesat	25
2 Objetivos.	26
2.1 Objetivo General	26
2.2 Objetivos Específicos	26
3 Fundamento Teórico.	27
3.1 Instrumentación de alto nivel	27
3.2 Comunicación	28
3.3 Visualización	28
3.4 Visual Studio Code	30
3.5 MYSQL	31
3.6 PYTHON	32
3.7 PHP	33
3.8 HTML	33
3.9 CSS	33
4 Sistema desarrollado.	35
4.1 Requerimientos del Sistema	35
4.1.1 Requerimientos de Hardware	35
4.1.2 Requerimientos de Software	36
4.2 Arquitectura del Sistema	37
4.2.1 Requerimientos funcionales y no funcionales	39
4.2.2 Algoritmo de procesamiento	41
4.3 Diseño de Etapas del Sistema	43
4.3.1 Etapa para la validación del sistema de comunicación	43
4.3.2 Etapa de procesamiento de datos	43
4.3.3 Etapa de visualización	44
4.3.4 Etapa de Almacenamiento de Data	46
4.3.5 Etapa de validación de variables sensadas	47

4.3.6	Algoritmo de Procesamiento	49
5	Simulación del Sistema.	50
5.1	Prototipo centro de control	52
6	Implementación Del Sistema.	55
6.1	Implementación etapa para la validación del sistema de comunicación	55
6.2	Implementación etapa procesamiento de datos	59
6.3	Implementación etapa de visualización	61
6.4	Implementación etapa de almacenamiento	63
6.5	Implementación de la validación de variables sensadas	63
7	Resultados.	67
7.1	Resultados etapa para la validación del sistema de comunicación	67
7.2	Resultados etapa de procesamiento de datos	67
7.3	Resultados etapa de Visualización	68
7.4	Resultados etapa de almacenamiento	71
7.5	Resultados etapa de variables sensadas	72
8	Conclusiones	73
9	Recomendaciones.	75
10	Trabajo Futuro	76
	Referencias Bibliográficas	79
	Apéndices	80

Lista de Tablas

1	<i>Procesamiento aviónica a bordo (Fuente:[J.Vivas, 2022])</i>	28
2	<i>Diccionario trama para la aviónica</i>	29
3	<i>Diccionario trama para el CanSat</i>	29
4	<i>Condiciones ambientales misión GuaneSat</i>	36
5	<i>Condiciones ambientales misión Orión</i>	36
6	<i>Software y Hardware</i>	37
7	<i>Historial data misión Orión</i>	46
8	<i>Historial data misión GuaneSat</i>	46
9	<i>Historial Banco de pruebas</i>	47
10	<i>Condiciones ambientales misión Orión</i>	47
11	<i>Guia de observación para sensores e instrumentos</i>	48
12	<i>Normativa de Radioenlace nacional e internacional</i>	56
13	<i>Tipos de Antenas</i>	57
14	<i>MS5611-01BA03</i>	64
15	<i>BNO055</i>	64
16	<i>GY-63 MS5611</i>	65
17	<i>MPU6050</i>	65
18	<i>BMP280</i>	66
19	<i>Termocupla tipo K MAX6675</i>	66
20	<i>Consolidado de sensores aprobados para LASC2022</i>	72

Lista de Figuras

1	Diagrama de Bloques Aviónica (Fuente:[J.Vivas, 2022])	23
2	Diagrama de Bloques Aviónica CanSat (Fuente:[SCUA, 2022a])	24
3	Diagrama general Estación terrena (Fuente: propia)	24
4	Sistema de transmisión y recepción mediante radios LoRa y su respectiva alimentación. Sistema de Telemetría. (Fuente:[J.Vivas, 2022])	25
5	Circuitos integrados digitales más comunmente usados en un instrumento de alto nivel (Fuente:[Constaín, 2012])	27
6	Esquema general de un instrumento de alto nivel (Fuente:[Constaín, 2012])	27
7	Trama de datos propuesta por la aviónica durante la validación de la estación terrena. (Fuente:[J.Vivas, 2022])	30
8	Arquitectura con base a las misiones.	38
9	Arquitectura llegada de datos.	39
10	Proceso de ejecución Full Stack.	42
11	Arquitectura del back-end	44
12	Tratamiento de datos	45
13	Tratamiento de datos	45
14	Tareas de lenguajes de programación	46
15	Comparación datos históricos de temperatura máximas y mínimas en los meses de Junio, Julio y Agosto del año 2022 en Brasil y Bucaramanga-Colombia (Fuente:[the weather Channel, 2022])	50
16	Condiciones de Temperatura en el mes de Julio de 2022 en Bucaramanga (Fuente:[IDEAM, 2022])	51
17	Condiciones de Humedad en meses de Junio, Julio y Agosto de 2022 en Brasil comparada con Bucaramanga (Fuente:[the weather Channel, 2022])	51
18	Velocidad media del viento en meses de Junio, Julio y Agosto de 2022 en Brasil comparada con Bucaramanga (Fuente:[the weather Channel, 2022])	51
19	Primer dashboard mockup	53
20	Segundo dashboard mockup, pantalla aviónica	53
21	Segundo dashboard mockup, pantalla CanSat	53
22	Segundo dashboard mockup, pantalla datos de sensores CanSat	54
23	Línea de vista entre la entrada de la 30 UIS y el edificio de petróleos . .	57
24	Línea de vista entre la entrada de la 30 UIS y el edificio de petróleos con aplicativo web Ubiquiti	57
25	Línea de vista entre el edificio de petróleos y barrio los Pinos	58

26	<i>Línea de vista entre el edificio de petróleos y barrio los Pinos con aplicativo web Ubiquiti</i>	58
27	<i>Línea de vista entre el edificio de pesados y aeropuerto Palonegro</i>	58
28	<i>línea de vista entre el edificio de pesados y aeropuerto palonegro con aplicativo web Ubiquiti</i>	59
29	<i>Evidencia efectuando pruebas de radioenlace con aviónica a bordo y sistema de recepción</i>	59
30	<i>Datos sin procesar en un archivo .csv</i>	60
31	<i>Estructura de tabla para banktestingcrudadata</i>	60
32	<i>Estructura conexión entre base de datos y servidor</i>	60
33	<i>Estructura código para las características estéticas del diseño</i>	61
34	<i>Estructura código para la pantalla principal, presentación LASC</i>	61
35	<i>Estructura código para las características estéticas, vista de los usuarios .</i>	62
36	<i>Estructura código para las características estéticas para gráficas tipo Chart</i>	62
37	<i>Estructura código para las tablas de presentación de historiales.</i>	63
38	<i>Prueba con antena omnidireccional y antena modulo LoRa Ra-02</i>	67
39	<i>Trama recibida durante pruebas con antena omnidireccional y antena modulo LoRa Ra32</i>	68
40	<i>Resultados de la Etapa de procesamiento de datos</i>	68
41	<i>Pantalla principal</i>	69
42	<i>Menú desplegable</i>	69
43	<i>Pantalla para visualización datos de vuelo en misión Orión</i>	70
44	<i>Pantalla para visualización datos de vuelo en misión GuaneSat</i>	70
45	<i>Pantalla para visualización datos de vuelo en misión GuaneSat</i>	70
46	<i>Pantalla de historial de misión Orión</i>	71
47	<i>Pantalla de historial de misión GuaneSat</i>	71
48	<i>Pantalla de historial de registros en el banco de prueba del motor</i>	72
49	<i>Diagrama UML para la base de datos (Fuente: propia)</i>	81

Lista de Apéndices

Apéndice A. Electrónica empleada en la misión Orión (Fuente:[J.Vivas, 2022])	80
Apéndice B. Tabla PLAN DE BANDA. (440 – 470 MHz)	81
Apéndice C. Diagrama UML aplicado en la base de datos	81
Apéndice C. Repositorio general del proyecto - GitHub.	82
Apéndice D. Página web del trabajo de investigación	82

Glosario

- **ASCII:** Código Estadounidense Estándar para el Intercambio de Información. Sistema de codificación que permite insertar caracteres especiales y símbolos en los textos.
- **LASC:** “Latin American Space Challenge” Evento experimental de coherería y satélites realizado en Brasil.
- **Desarrollador Full Stack:** Profesional versátil capaz de desenvolverse en un desarrollo web tanto en la parte del cliente (Front End) como del servidor (Back End).
- **Back End:** Parte del desarrollo web que se encarga de que toda la lógica de una página web funcione. Se trata del conjunto de acciones que pasan en una web pero que no se ven.
- **Front End:** Parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.
- **Lenguaje Unificado de Modelado (UML):** Desempeña un rol importante en el desarrollo de software, ya que es una forma de mostrar visualmente el comportamiento y la estructura de un sistema o proceso.
- **Carga Útil:** Capacidad de carga máxima de un vehículo aéreo o espacial compuesta por un conjunto de instrumentos que cumplen un objetivo específico.
- **Cohete sonda de combustible sólido:** sistema compuesto por varios subsistemas intrínsecamente relacionados y que se disponen para transportar instrumentos científicos de observación y medición.
- **I2C:** “Inter-Integrated Circuit” o Circuito Inter-Integrado. Puerto y protocolo de comunicación serial, define la trama de datos y las conexiones físicas para transferir bits entre 2 dispositivos digitales.
- **IoT:** “Internet of Things” o Internet de las Cosas. Es la agrupación e interconexión de dispositivos y objetos a través de una red.
- **Perfil Atmosférico:** representación gráfica del cambio que presentan las distintas variables encontradas en la atmósfera en función de su altitud.

- **Gases de efecto invernadero:** Componentes gaseosos de la atmósfera, tanto naturales como antropógenos, que absorben y remiten radiación infrarroja.
- **VSC:** “Visual Studio Code”. Editor de código fuente independiente que se ejecuta en Windows, macOS y Linux..
- **Debugger:** “Depurador” Se encarga de controlar la ejecución y corrección de errores de un código de programación.
- **IDE:** “Integrated Development Environment”. Herramienta de software que proporciona un entorno de programación completo para los desarrolladores de software.
- **Radio Enlace:** Interconexión inalámbrica entre dos o mas puntos a través del envío y recepción de ondas electromagnéticas.
- **SCPI:** “Standard Command for Programmable Instruments” o Estándar de comandos para Instrumentos Programables. Define un estándar para la sintaxis y comandos utilizados para controlar dispositivos de prueba y medición programables.
- **SNR:** “Signal to Noise Ratio” o Relación Señal a Ruido. Relación existente en la potencia de una que señal que se transmite y la potencia del ruido que la corrompe.
- **SPI:** “Serial Peripheral Interface” o Interfaz de Periféricos Serial. Estándar de comunicación utilizado para la transferencia de información entre circuitos integrados en dispositivos electrónicos.
- **Trama:** Unidad de envío de datos. Se compone de una serie sucesiva de bits, organizados en forma cíclica, que transportan información, la cual, es posible extraer en la recepción.
- **UART:** “Universal Asynchronous Receiver Transmitter” o Transmisor-Receptor Asíncrono Universal. Protocolo para el intercambio bidireccional de datos entre dos dispositivos.
- **Bits:** Unidad de medida de la velocidad de transmisión de los datos.
- **Open Source:** El software de código abierto es un código que está diseñado para ser de acceso público: cualquiera puede ver, modificar y distribuir el código como mejor le parezca.
- **Script:** Documento que contiene instrucciones, escritas en códigos de programación. El script es un lenguaje de programación que ejecuta diversas funciones en el interior de un programa de computador.

- **Ganancia:** La ganancia de una antena es la relación entre la potencia que entra en una antena y la potencia que sale de esta. Esta ganancia es comúnmente referida en dBi's, y se refiere a la comparación de cuanta energía sale de la antena en cuestión, comparada con la que saldría de una antena isotrópica. Una antena isotrópica es aquella que cuenta con un patrón de radiación esférico perfecto y una ganancia lineal unitaria.
- **Directividad:** La directividad de la antena es una medida de la concentración de la potencia radiada en una dirección particular. Se puede entender también como la habilidad de la antena para direccionar la energía radiada en una dirección específica. Es usualmente una relación de intensidad de radiación en una dirección particular en comparación a la intensidad promedio isotrópica.
- **Polarización:** Es la orientación de las ondas electromagnéticas al salir de la antena. Hay dos tipos básicos de polarización que aplican a las antenas, como son: Lineal (incluye vertical, horizontal y oblicua) y circular (que incluye circular derecha, circular izquierda, elíptica derecha, y elíptica izquierda). No olvide que tomar en cuenta la polaridad de la antena es muy importante si se quiere obtener el máximo rendimiento de la antena. La antena transmisora debe de tener la misma polaridad de la antena receptora para máximo rendimiento.
- **Línea de Vista:** Se refiere a un camino (path) limpio, sin obstrucciones, entre las antenas transmisoras y receptoras. Para que exista la mejor propagación de las señales RF de alta frecuencia, es necesaria una Línea de vista sólida (limpia - sin obstrucciones).

Resumen

Título: Diseño e Implementación del front-end y back-end de una plataforma IoT para recepción, registro y visualización de los datos de telemetría y del payload del cohete sonda de combustible sólido Orión del semillero de Cohetería UIS Aeroespacial.¹

Autor: Laura Nathalia Pinto Campos, Ana María Sánchez Castillo²

Palabras Clave: Full stack, Servidor, Front End, Back End, Cohete sonda de combustible sólido, Carga Útil, Radiofrecuencia, Sensores, Validación.

Descripción: En este proyecto se implementó y validó el conjunto de sensores y sistema de telecomunicaciones en el cohete sonda de combustible sólido, para obtener las lecturas de la sensórica presente tanto en el cohete como en el satélite y proporcionar esta información en una interfaz gráfica, partiendo desde el tratamiento de datos, lógica de la interfaz o más comúnmente conocido como el Back End, para continuar diseñando el Front End o estilo visual, bandejas u pantallas con las diferentes formas de visualizar las lecturas recopiladas por los sensores.

Por ende se validaron los instrumentos capaces de operar tomando como referencia los reportes proporcionados por el IDEAM (Instituto de Hidrología, Meteorología y Estudios Ambientales) y The Weather Channel, con el fin de proporcionar información veraz; además de validar el sistema de telecomunicaciones que comprende la misión y el sistema de adquisición de datos. Siendo un proyecto que tiene como fin principal el tratamiento de data cruda y a través de este procesamiento, visualizar de forma comprensible en una interfaz gráfica cada una de las lecturas proporcionadas por la sensórica validada.

¹Trabajo de grado

²Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de telecomunicaciones. Director: Julián Rodríguez Ferreira, Doctorado en Física, Especialidad astrofísica

Abstract

Title: Design and implementation of the front-end and back-end of an IoT platform for receiving, recording and visualization of telemetry data and payload of the Orion solid fuel sounding rocket of the UIS Aerospace Rocketry research group.³

Author(s): Laura Nathalia Pinto Campos, Ana María Sánchez Castillo⁴

Key Words: Full stack, Server, Front End, Back End, Solid Fuel Sounding Rocket, Payload, Radio Frequency, Sensors, Validation.

Description: In this project we implemented and validated the set of sensors and telecommunications system in the solid fuel sounding rocket, to obtain the sensor readings present in both the rocket and the satellite and provide this information in a graphical interface, starting from the data processing, interface logic or more commonly known as the Back End, to continue designing the Front End or visual style, trays or screens with different ways to visualize the readings collected by the sensors.

Therefore, the instruments capable of operating were validated taking as a reference the reports provided by the IDEAM (Institute of Hydrology, Meteorology and Environmental Studies) and The Weather Channel, in order to provide accurate information; in addition to validating the telecommunications system comprising the mission and the data acquisition system. The main purpose of this project is the treatment of raw data and through this processing, to visualize in a comprehensible way in a graphic interface each of the readings provided by the validated sensor system.

³Bachelor Thesis

⁴Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y telecomunicaciones. Director: Julián Rodríguez Ferreira, Doctorado en Física, Especialidad astrofísica

Introducción

El aire es una mezcla homogénea presente en toda la atmósfera terrestre, rica en Oxígeno (21%), Nitrógeno (75%), Vapor de agua (1%) y otros gases (3%). A lo largo de los últimos 100 años, los desarrollos industriales y el modernismo han ocasionado un incremento considerable en la emisión de gases efecto invernadero hacia la atmósfera, provocando el aumento de la concentración de gases nocivos para la salud, hasta el punto que, según datos de la OMS (Organización Mundial de la Salud), a nivel mundial una de cada ocho muertes es ocasionada por la contaminación presente en el aire [IDEAM, 2022]. Por tal motivo, el monitoreo y control de la concentración de gases presentes en la atmósfera ha tomado cada vez más importancia para la comunidad científica. Para ello, existen diferentes alternativas y técnicas de sondeo dentro de las que se encuentran globos estratosféricos, satélites sonda y estaciones meteorológicas, entre otras.

Las sondas son dispositivos artificiales que se envían hacia un objetivo concreto. Todas las sondas se montan sobre una estructura soporte, a la que se deben incorporar al menos, los tres sistemas que se describen: Sistema energético el cual comprende baterías eléctricas y paneles solares para proveer de electricidad a los sistemas, instrumentos de observación como lo son cámaras fotográficas, sensores o analizadores de espectro y por último equipos de comunicación existiendo diversos tipos de antenas para transmitir y visualizar información recolectada durante la misión. Esta actividad actualmente posee unos records establecidos por empresas espaciales desde 1972 [Wikimedia, 2022]. Las sondas pueden incorporar: motores para efectuar maniobras, tanques de combustible, protecciones térmicas para evitar el congelamiento de la sonda, o transportar las sondas menores independientes. El peso de una sonda es determinado por la capacidad en la limitación de los cohetes.

Una vez lanzado el cohete se debe conocer los estados en los que se encuentra la misión ya que de esto depende si el satélite es eyectado en el momento indicado, al igual que la posición del cohete durante su trayectoria, es bien sabido que el cohete perderá peso con referencia a su punto de partida por consumo de combustible y la eyección de la sonda, por lo que es importante conocer como es su comportamiento vectorial de descenso. Los satélites se envían para cumplir una misión por esto es importante tener control de todos sus movimientos, sensores, cámaras, con el fin de suplir las necesidades en el campo de la investigación, ya sea de aficionados, de universidades o entidades destinadas a construir este tipo de vehículos de transporte instrumental. A partir de lo anterior, los requisitos de la investigación que se quiere realizar y tomando en cuenta las misiones Orion

y GuaneSat para su caracterización y visualización, están encaminadas en la toma de mediciones medioambientales de mayor abundancia en una altura de aproximadamente 500 metros, siendo el recorrido pronosticado en la misión GuaneSat y una altura de 3 kilómetros para la misión Orión. Se desarrolla la implementación de una plataforma IoT capaz de compilar la telemática para las dos misiones en conjunto.

En este trabajo de investigación se plasma toda la documentación relacionada con el tratamiento de la data, desde el desarrollo de back-end, frond-end y validación de una interfaz IoT partiendo del estado actual de las misiones Orión y GuaneSat y el papel que desempeña este trabajo de grado en las mismas. Un marco teórico que permita definir y explicar los conceptos físicos en los cuales se basa el funcionamiento del instrumento a partir de investigaciones previas. Se abordan los requerimientos, arquitectura y diseño de las diferentes etapas que componen el proceso de visualización y tratamiento (etapa de procesamiento de datos, etapa de visualización, etapa de almacenamiento, validación de variables sensadas). Se documenta el proceso de selección, adquisición y rendimiento. Se explican los algoritmos realizados para la interfaz gráfica y los software utilizados. Finalmente, se presenta la implementación por etapas y una integración final del sistema, los resultados obtenidos, recomendaciones, trabajo futuro y conclusiones.

1 Problema

1.1 Definición del Problema

Actualmente, numerosas personas utilizan procesadores de texto para escribir documentos; hojas de cálculo para realizar presupuestos, nóminas o aplicaciones estadísticas; navegadores (browsers) para explorar internet, y programas de correo electrónico para enviar información y documentos por la red. Los procesadores de texto, las hojas de cálculo y los gestores de correo electrónico son ejemplos de programas o software que se ejecutan en computadoras. El software se desarrolla utilizando lenguajes de programación como Java, C, C++, C# o XML [Joyanes Aguilar, 2011].

Una aplicación de software se desarrolla a través de un proceso. No es algo que se fabrica a partir de materia prima, ni se ensambla a partir de piezas más pequeñas. Las aplicaciones de software son “programas aislados que resuelven una necesidad comercial específica”. Algunos ejemplos son el software que procesa datos comerciales o técnicos que facilitan las operaciones y la gestión de una investigación, negocio u empresa. Además del código fuente, el software incluye toda la documentación y los datos necesarios para que el programa funcione correctamente [Ramos Cardozzo, 2016].

Representando así un reto a la hora de trabajar con datos ya sean almacenados o en tiempo real, lo que conlleva visualizarlos de tal forma que sean comprensibles y que a su vez permitan concluir afirmaciones que contribuyen conocimiento al campo con el que estos datos estén relacionados. Las alternativas que se encuentran varían de acuerdo a los gustos de cómo se desea presentar esta información, a la cantidad de programas y lenguajes que permiten crear a partir de plantillas existentes plataformas capaces de hacer que la experiencia de visualizar datos sin tratamiento, sea entendible y a la vez generar una interacción agradable de estos con los usuarios.

Si un ser humano desea comunicarse lo hace mediante los lenguajes humanos, bien sea el español, el inglés o el portugués, pero si quiere interactuar con una máquina necesita un medio para poder hacerlo. Ese medio es, precisamente, la interfaz de usuario. Según [Gallego Carrillo, 2005], “la interfaz es un lugar de encuentro entre los bits y las personas”; la interfaz gráfica de usuario es la parte visible de las aplicaciones, es lo que se percibe del sistema y, por ello, tiene mucha importancia. A través de ella y, de forma transparente, se debe proporcionar, entre otras cosas: seguridad, fiabilidad, sencillez y precisión.

La interfaz de usuario es el medio a través del cual un usuario interactúa con un sistema informático. Esta interfaz puede tomar muchas formas, desde una simple línea de comandos hasta una interfaz gráfica de usuario (GUI) altamente visual e interactiva. La GUI es el tipo más común de interfaz de usuario y utiliza elementos visuales, como iconos, botones y menús, para permitir que los usuarios interactúen con el sistema. Los usuarios pueden hacer clic en los elementos de la GUI con el mouse o utilizar teclas de acceso directo en el teclado para realizar tareas específicas.

Otras formas de interfaz de usuario incluyen la línea de comandos, que es un tipo de interfaz basada en texto que utiliza comandos escritos para interactuar con el sistema, y la interfaz de voz, que permite a los usuarios interactuar con el sistema mediante el uso de comandos de voz. En general, una buena interfaz de usuario debe ser fácil de usar y que permita a los usuarios realizar las tareas que necesitan de manera eficiente y sin confusión. Debe ser fácil de entender y utilizar, y debe proporcionar retroalimentación clara y útil sobre las acciones del usuario.

Siendo algunos de los requisitos básicos para plantear y desarrollar una herramienta capaz de visualizar, dar tratamiento y que a la vez almacene data cruda en tiempo real. Además es de vital importancia comprender el funcionamiento de una computadora, en cuanto a cómo representa la información; considerando la codificación en patrones de bits que los elementos internos de la computadora almacenen y procesen con facilidad esta información. Implicando el desarrollo de lenguajes ensambladores puesto que la programación escrita en lenguaje máquina es un proceso tedioso y difícil de realizar [Joyanes Aguilar, 2011].

Los lenguajes de programación son sistemas formales diseñados para expresar las instrucciones que pueden ser ejecutadas por una computadora. Cada lenguaje de programación tiene sus propias reglas sintácticas y semánticas que definen la estructura y el significado de las instrucciones. Existen muchos lenguajes de programación diferentes, cada uno con su propia historia y conjunto de características.

De acuerdo con lo mencionado, se plantea desarrollar una interfaz gráfica que permita visibilizar las lecturas tomadas por dos misiones, las cuales estarán presentes en el concurso de coherencia experimental denominado LASC por sus siglas en inglés **Latin American Space Challenge** [LASC, 2022] y que además pertenecen al **Semillero de Coherencia UIS Aeroespacial - SCUA** [SCUA, 2022c]. El presente proyecto, se encarga de diseñar e implementar el front-end y back-end de una interfaz gráfica para la

recepción, registro y visualización de los datos de telemetría y del payload del cohete sonda de combustible sólido Orión del semillero de Cohetería UIS Aeroespacial.

1.2 Planteamiento de la Misión

Basándose en la definición del problema planteado previamente, se realiza el diseño y construcción de una interfaz gráfica la cual permite visualizar la telemática para el cohete sonda de combustible sólido Orión y su carga útil GuaneSat, los cuales participaron en el concurso latinoamericano LASC 2022, con el objetivo principal de llevar a cabo un vuelo de 3 Km de altura y en su descenso a 500 metros de la superficie expulsar la carga útil que comprende un satélite en lata, el cual también es controlado desde la interfaz, esto con el fin de poder monitorear la calidad del aire y construir un perfil atmosférico, a partir de los instrumentos de medición y electrónica necesaria tanto de la aviónica del cohete como de su carga útil.

La misión se encuentra segmentada en 3 etapas:

1. **Validación:** Esta etapa hace referencia a evaluar cada una de las mediciones que se tienen mediante la telemetría presente en el cohete sonda de estado sólido Orión del grupo de cohetería UIS Aeroespacial y a su vez a la carga útil de este cohete un satélite tipo CanSat que lleva como nombre de la misión GuaneSat. La misión GuaneSat tiene como objetivo final hacer un registro de mediciones físicas en su vuelo como: temperatura, humedad, CO, CO₂, NO₂, NH₃, tVOC, y también cuenta con sensores para su funcionamiento mecánico como lo son el magnetómetro, presión, brújula, altímetro, giroscopio y acelerómetro. En segundo lugar para el cohete Orión se tiene como objetivo validar cada una de las variables físicas como lo son: presión atmosférica, altímetro, GPS, giroscopio, temperatura, cámaras, acelerómetro, encendido y apagado, por consiguiente se describirán a fondo más adelante en la sección fases de la misión y banco de pruebas.
2. **Estación Terrena:** Esta etapa se encarga de la adecuación de las señales emitidas por el Cohete sonda Orión durante el ascenso y descenso, registros de datos por el satélite GuaneSat; estos se componen por antenas, sistemas de transmisión y recepción por radiofrecuencia, para las cuales se llevan a cabo pruebas para su validación y obtener un correcto funcionamiento en sus buses de datos.
3. **Interfaz gráfica:** Esta sección abarca todo lo relacionado con la correcta visualización de los datos obtenidos. De igual forma, se espera que sea sencilla y de fácil acceso para que los datos generados puedan ser visibilizados por aquellas personas

interesadas en el monitoreo de la calidad del aire y por supuesto por quienes en su momento esten ejecutando cada una de las misiones citadas.

1.3 Estado Actual de las misiones

1. **Cohete Orión:** La misión cohete sonda de combustible sólido Orión es un proyecto que se presentó por el semillero de cohetería SCUA, el cual participó en el concurso latinoamericano de cohetería LASC 2022 que se llevó a cabo en el mes de agosto del presente año en la ciudad de Tatuí - Brasil. Esta misión está desarrollada por varios sistemas: Sistema de estructura, sistema de propulsión, sistema estructural, Aerodinámica, Sistema de recuperación y aviónica.[SCUA, 2022b]
2. **Satélite GuaneSat:** La misión satélite GuaneSat es un proyecto que se presentó por el semillero de cohetería SCUA, el cual participó en el concurso latinoamericano de cohetería LASC 2022 en la categoría de satélites y más específicamente en satélites tipo CanSat que se llevó a cabo en el mes de agosto del presente año en la ciudad de Tatuí - Brasil. Esta misión tiene como fin hacer una recopilación de mediciones atmosféricas tomadas durante su vuelo como: *temperatura, humedad, presión, CO₂, CO, NO₂, NH₃, tVOC* [SCUA, 2022a].

Con base en la estación terrena capaz de recibir la data transmitida por el cohete y su carga útil, se ve la necesidad de diseñar e implementar un servidor el cual actúa como visualizador de los datos, control de vuelo del CanSat y control de cohete, base fundamental para la realización de este proyecto de grado.

1.3.1 Validación

En cuanto a la interpretación y tratamiento de la información, la electrónica analógica y digital presentan una gran diferencia, la primera está más cerca del fenómeno físico y la segunda más cerca a su tecnología digital. Por consiguiente, la validación durante el ascenso y el descenso de las dos misiones, se espera que todas las mediciones registradas por los sensores sea de forma coherente con las previstas mediante datos registrados en bases medioambientales, las cuales son recogidas por estaciones metereológicas y clasificadas hacia los sensores que se tienen en las misiones. Estas mediciones son de vital importancia puesto que al experimentar cambios bruscos puede afectar la calidad de la misión, es decir, los componentes electrónicos deben seleccionarse con cualidades como: capacidad de respuesta rápida alrededor de 30 [Mbps], con frecuencias de operación de más o menos 420 - 450 MHz y deben estar entre los rangos establecidos para operar correctamente durante la misión, entre ellos a temperaturas de -40 a 60 [°C], presión

atmosférica de 1013.25 a 540 [mBar].

La validación se identifica como la verificación de requisitos específicos para gestionar procesos y métodos de trabajo adecuados para un uso específico. En la validación se demuestra lo que es más idóneo para cierta medición de acuerdo a la documentación previa [IDEAM, 2022]. Una de las constantes que son legítimas para una validación es el uso de métodos normalizados y se refiere al identificar y evaluar referente a publicaciones por organismos reconocidos del medio ambiente, donde es oportuno estudiar las características para garantizar que el método normalizado sea satisfactorio.

Además es de resaltar como parte fundamental la tesis titulada ‘Diseño, implementación y validación de la aviónica para el cohete sonda de combustible sólido ORION UIS partícipe en el Latin American Space Challenge 2022’, desarrollada por los estudiantes de ingeniería electrónica de la Universidad Industrial de Santander, Juan Martín Vivas Camargo y Santiago Hernández Rincón [J.Vivas, 2022], en la que se establecen todos los requisitos, sistemas y funcionalidad electrónica de la aviónica del cohete sonda misión Orión, requerimientos y procesos de manufactura que dieron lugar a la validación de cada una de sus mediciones registradas, resumiendo, se presentan los resultados obtenidos.

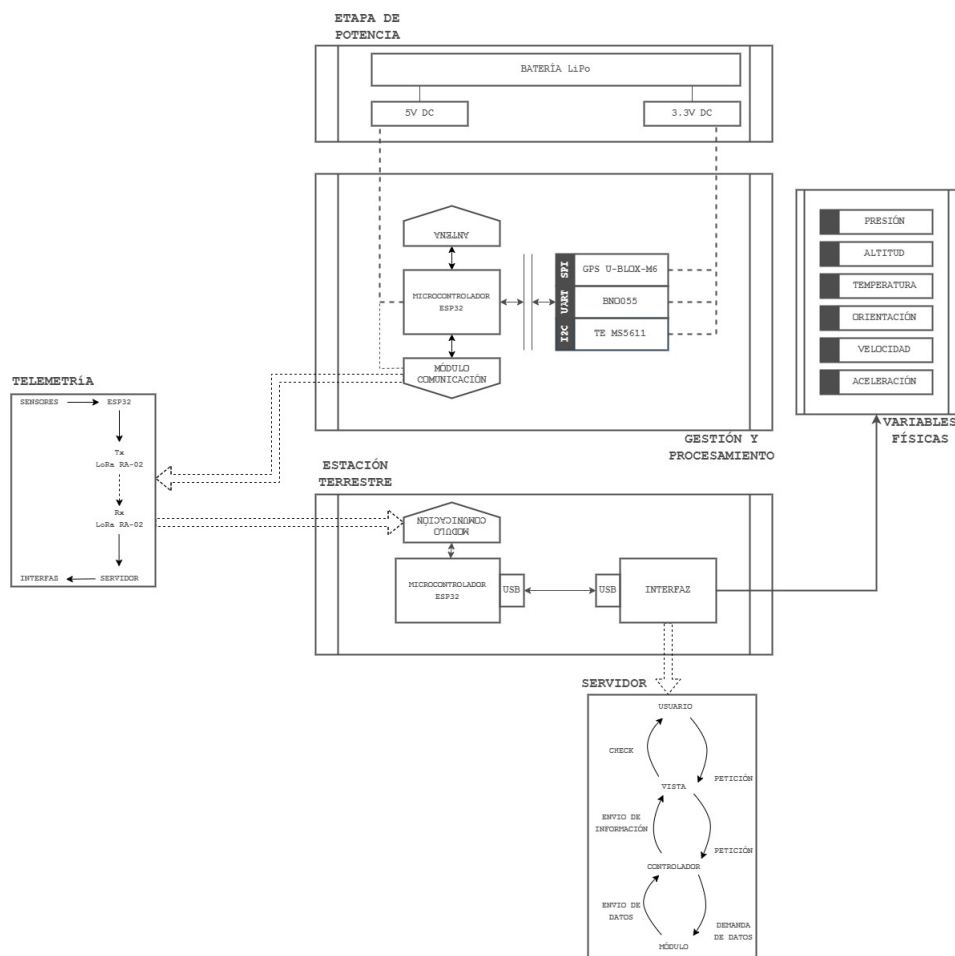
1.3.2 Transmisión de datos e interfaz gráfica

Las misiones de cohetes y su carga útil se han convertido en una forma de desarrollar misiones específicas como en el caso de Orión y GuaneSat, siendo su objetivo principal el captar mediciones atmosféricas; también se encuentran otras misiones como por ejemplo recoger muestras biológicas la cual fue desarrollada por el equipo de cohetaría de Paraguay en el LASC 2022, llevar satélites a 3Km, 1km y 500m. Sin embargo, para obtener todos los datos que registren los instrumentos a bordo de la misión, se hace necesario establecer un sistema de comunicaciones que permita transmitir la información recolectada y en paralelo, ejecutar la recepción de dicha información en Tierra.

El sistema de telemetría que utiliza Orión, se llevó a cabo en la tesis titulada ‘Diseño, implementación y validación de la aviónica para el cohete sonda de combustible sólido ORIÓN UIS partícipe en el Latin American Space Challenge 2022’, por Juan Vivas y Santiago Hernández, estudiantes de la Universidad Industrial de Santander, en la cual, se utilizan dos radios LoRa, uno presente en la estación terrena y otro como parte de la aviónica del cohete, dicha aviónica propone el diagrama de bloques ilustrado en la figura 1; para establecer un radio enlace punto a punto que permita garantizar la comunicación y transferencia sin pérdidas de los paquetes de datos. Se seleccionaron ciertos instrumentos

de medición, aptos para el vuelo real, que permitieron adquirir datos y llevar a cabo pruebas in-situ para verificar la correcta comunicación entre el cohete y la estación terrena. Lo anterior fue diseñado a partir de la interconexión de los sensores meteorológicos, una IMU ESP32, TX LoRa, RX LoRa, lo que supone la necesidad de crear un servidor y por último la interfaz gráfica que facilita la visualización de los datos obtenidos. Por otra parte, se expone la aviónica para el CanSat en la figura 2, de igual forma que en el cohete cuenta con módulos LoRa para la comunicación y posterior conexión con la interfaz gráfica.

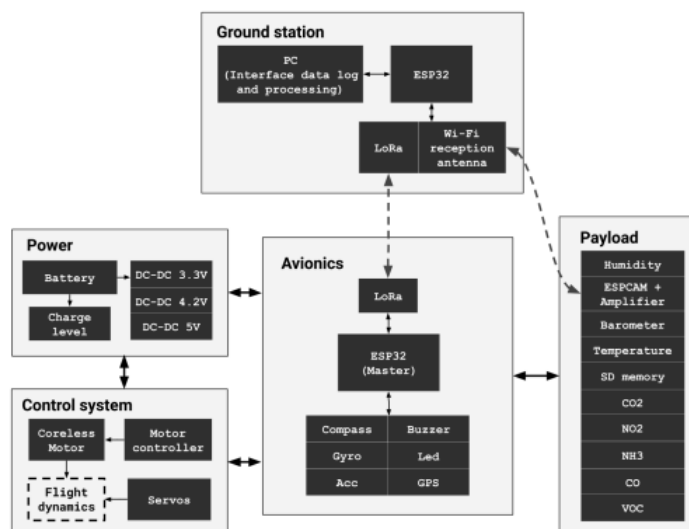
Figura 1: *Diagrama de Bloques Aviónica (Fuente:[J.Vivas, 2022])*



1.3.3 Estación Terrena

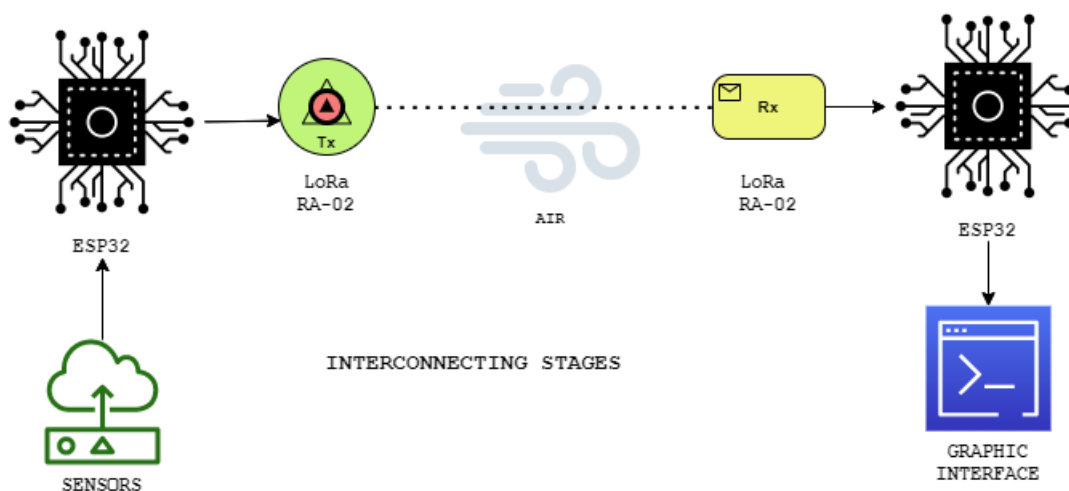
A partir del objetivo planteado sobre validar el sistema de comunicación y desarrollado en conjunto con el proyecto de grado *Diseño, implementación y validación de la aviónica para el cohete sonda de combustible sólido Orion UIS participante en el Latin American Space Challenge 2022*, en la figura 3 se establece el diagrama general de conexión entre la

Figura 2: Diagrama de Bloques Aviónica CanSat (Fuente:[SCUA, 2022a])



electrónica presente en el cohete y la interfaz gráfica ubicada en tierra, optando por los radios LoRa Ra-02, módulos transceptores que permiten ser configurados bien sea como transmisor o receptor, dependen directamente de su composición y arquitectura interna para suplir radioenlaces que alcancen diferentes distancias, llevando a cabo el respectivo montaje que permite establecer una correcta comunicación en un radioenlace en punto de vista.

Figura 3: Diagrama general Estación terrena (Fuente: propia)



El radio en la estación terrena está conectado a un computador personal por medio de un puerto USB, el cual, permite ajustar vía protocolo serial los parámetros del radio como lo son: selección de la banda de frecuencia de operación, potencia de transmisión,

velocidad de enlace, reiniciar el dispositivo, actualizar versión del firmware, entre otras opciones. Para mejor comprensión remítase a los diagramas de bloques de las figuras 1 y 2, además en la figura 4 se puede ver el sistema de transmisión y recepción mediante los módulos LoRa y su conexión por puerto USB.

Figura 4: *Sistema de transmisión y recepción mediante radios LoRa y su respectiva alimentación. Sistema de Telemetría. (Fuente:[J. Vivas, 2022])*



1.4 Este Proyecto Como Parte de Orión y Guanesat

El presente trabajo de investigación tiene como objetivo principal validar el sistema de adquisición y transmisión de datos para dos misiones desarrolladas por el Semillero de investigación SCUA, mediante una interfaz gráfica que permite la visualización del comportamiento de la instrumentación que compone el cohete sonda, denominado misión Orión y su carga útil denominada GuaneSat. En dicha interfaz se contará con el desarrollo en conjunto de la telemática de las dos misiones (Orión y GuaneSat) y de igual forma una pantalla de visualización para las pruebas de temperatura y empuje de un banco de pruebas, destinado para el estudio del motor de combustible sólido. Esta interfaz gráfica está diseñada para enlazar las dos misiones y recopilar todos los datos registrados y ordenados para su posterior estudio en estas áreas de la ciencia, para monitorear la calidad del aire y construir perfiles atmosféricos.

2 Objetivos

2.1 Objetivo General

Diseñar una interfaz gráfica para la visualización del comportamiento de la instrumentación que compone la carga útil del cohete sonda de combustible sólido Orión del semillero de Cohetería UIS Aeroespacial y validar el sistema de adquisición y transmisión.

2.2 Objetivos Específicos

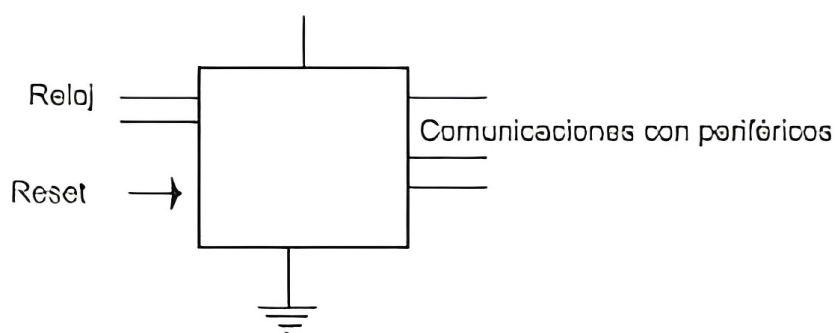
- Validar el sistema de comunicación de la aviónica utilizada en el cohete sonda de combustible sólido Orión, realizado por un radio enlace que permite en línea de vista directa la recepción y transmisión de datos en un alcance de 3 kilómetros y documentar las pruebas para su correcto funcionamiento.
- Diseñar una interfaz gráfica basada en los lenguajes como: PHP, Python, JavaScript y servidor HTTP para la visualización de las variables sensadas y el estado del cohete y su carga útil.
- Validar mediante guías de observación de instrumentación de alto nivel cada una de las lecturas de los sensores seleccionados para el cohete sonda de combustible sólido Orión y su carga útil del semillero de Cohetería UIS Aeroespacial.
- Integrar el sistema de visualización generado a partir del tratamiento de datos del cohete sonda junto con los datos recolectados de su carga útil.

3 Fundamento Teórico

3.1 Instrumentación de alto nivel

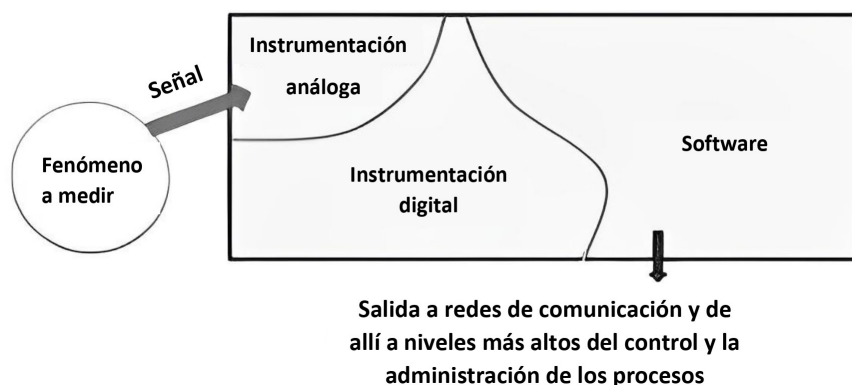
Se considera un instrumento de alto nivel si esta compuesto por dos partes fundamentales, circuitos digitales como el de la figura 5 y software para el manejo posterior de las mediciones.

Figura 5: *Circuitos integrados digitales más comúnmente usados en un instrumento de alto nivel (Fuente:[Constaín, 2012])*



De acuerdo con la figura 6, se toma en consideración el fenómeno que se mide y obteniendo la señal física y eléctrica característica, la sección analógica toma parte con un nivel de importancia considerable, porque la calidad de la medida depende en realidad de esta etapa.

Figura 6: *Esquema general de un instrumento de alto nivel (Fuente:[Constaín, 2012])*



Los sensores de la aviónica implementados en el proyecto de grado '*Diseño, implementación y validación de la aviónica para el cohete sonda de combustible sólido Orion UIS participante en el Latin American Space Challenge 2022*', que comprenden la carga útil se listan en la tabla 1 con sus respectivos protocolos y pines de conexión al microcontrolador ESP32-WROOM-32D, además en el apéndice A se describen con mas detalle:

Tabla 1: *Procesamiento aviónica a bordo (Fuente:[J.Vivas, 2022])*

Elemento	Protocolo	PIN(ES)	Potencia consumida [mW]
ESP32-WROOM-32D	SPI/I2C/ I2S/UART/ USART/GPIO	4/2/ 2/3/ 0	528
GPS NEO-6M	UART	RX/TX	148.5
BAR TE MS5611	I2C	SDA/SCL	4.62
IMU BNO055	I2C	SDA/SCL	41.25
LoRa RA-02	SPI	CS1, MOSI, MISO, SCK	396
SD	SPI	CS2, MOSI, MISO, SCK	200
BATERÍA LiPo Zippy Compact 2200mAh		3.3V	
		Total	1318.37

3.2 Comunicación

Con el modelo funcional de comunicaciones se procede a establecer la organización de los datos en una trama de caracteres en formato ASCII para posteriormente visualizarlos de forma ordenada. En la siguiente tabla, se puede observar los valores de los datos con su respectivo significado, generando así, una especie de diccionario que permite simplificar la incrustación de toda la trama en la interfaz gráfica, partiendo del principio Key-Value muy utilizado en Python. La trama para la aviónica que se plantea para enviar es la siguiente: “Accx, Accy, Accz, Apogeo1, Apogeo2, Tiempo, Altura, Presión, GPSlat, GPSlong, Power, Fases, temp, girox, giroy, giroz” en total son 16 datos donde cada dato se encuentra separado por ‘,’ en la tabla 2 se enuncia el significado de cada una de las siglas; el bus de datos comienza y finaliza con llaves ‘{bus}’, en cuanto a la trama de CanSat se tiene: “GPSLAT, GPSLON, GPSALT, GPSSPE, HUM, TEMP, PRES, CO, CO₂, NO₂, NH₃, TVOC, COMX, COMY, COMZ, ACCX, ACCY, ACCZ, GYRX, GYRY, GYRZ, BAT, VEL, ANGA, ANGB” con una longitud de 25 datos cada uno separado por “ ” el bus de datos comienza y finaliza con llaves ‘< bus >’, en la tabla 3 se enuncia el significado de cada una de las siglas.

3.3 Visualización

Con base a modelos de interfaces, modelos IoT y las características de la misión, entre ellas: la implementación de dispositivos de bajo consumo de potencia capaces de manejar protocolos de comunicación digitales. La plataforma se encuentra alojada en el servidor local. El computador de la estación terrena es el encargado de adquirir los datos

Tabla 2: *Diccionario trama para la aviónica*

Sigla	Significado
Accx	Acelerómetro x
Accy	Acelerómetro y
Accz	Acelerómetro z
Apogeo1	Apogeo
Apogeo2	Apogeo
Tiempo	Tiempo del vuelo
Altura	Altura
Presion	Presión
GPSlat	GPS latitud
GPSlong	GPS longitud
Power	Luz on/off
Fases	Línea de tiempo parte inferior
temp	Temperatura
girox	Giroscopio x
giroy	Giroscopio y
giroz	Giroscopio z

Tabla 3: *Diccionario trama para el CanSat*

Sigla	Significado
GPSLAT	GPS latitud
GPSLON	GPS longitud
GPSALT	GPS altitud
GPSSPE	GPS velocidad
HUM	Humedad
TEMP	Temperatura
PRES	Presión
CO	CO
CO ₂	CO ₂
NO ₂	NO ₂
NH ₃	NH ₃
TVOC	Total compuestos orgánicos volátiles
COMX	Brújula x
COMY	Brújula y
COMZ	Brújula z
ACCX	Acelerómetro x
ACCY	Acelerómetro y
ACCZ	Acelerómetro z
GYRX	Giroscopio x
GYRY	Giroscopio y
GYRZ	Giroscopio z
BAT	Estado de batería
VEL	Velocidad del motor
ANGA	Ángulo servo 1
ANGB	Ángulo servo 2

del radio transmisor mediante el módulo LoRa y convertirlos en un diccionario de trama pura [J.Vivas, 2022], como la que se presenta en la figura 7, propuesta por la aviónica para la etapa de validación de la telemetría.

Figura 7: Trama de datos propuesta por la aviónica durante la validación de la estación terrena. (Fuente:[J.Vivas, 2022])

```

TRYFINI.TXT: Bloc de notas
Archivo Edición Formato Ver Ayuda
{ Acc x: 0.00 , Acc y: 0.00 , Acc z: 0.00 , Apogeo: 0.00 , Apogeo2: 0.00 , Tiempo seg: , 0 , ap ^
{ Acc x: 0.00 , Acc y: 0.00 , Acc z: 0.00 , Apogeo: 0.00 , Apogeo2: 0.00 , Tiempo seg: , 0 , ap
{ Acc x: 0.00 , Acc y: 0.00 , Acc z: 0.00 , Apogeo: 0.00 , Apogeo2: 0.00 , Tiempo seg: , 0 , ap
{ Acc x: 0.00 , Acc y: 0.00 , Acc z: 0.00 , Apogeo: 0.00 , Apogeo2: 0.00 , Tiempo seg: , 0 , ap
{ Acc x: 0.00 , Acc y: 0.00 , Acc z: 0.00 , Apogeo: 0.00 , Apogeo2: 0.00 , Tiempo seg: , 0 , ap
{ Acc x: 0.00 , Acc y: 0.00 , Acc z: 0.00 , Apogeo: 0.00 , Apogeo2: 0.00 , Tiempo seg: , 0 , ap
{ Acc x: -0.25 , Acc y: -0.43 , Acc z: 9.53 , Apogeo: 9.54 , Apogeo2: 0.00 , Tiempo seg: , 4 ,
{ Acc x: -0.27 , Acc y: -0.45 , Acc z: 9.51 , Apogeo: 9.52 , Apogeo2: 0.00 , Tiempo seg: , 4 ,
{ Acc x: -0.25 , Acc y: -0.43 , Acc z: 9.51 , Apogeo: 9.52 , Apogeo2: 0.00 , Tiempo seg: , 4 ,
{ Acc x: -0.28 , Acc y: -0.45 , Acc z: 9.53 , Apogeo: 9.54 , Apogeo2: 0.00 , Tiempo seg: , 4 ,
{ Acc x: -0.26 , Acc y: -0.44 , Acc z: 9.50 , Apogeo: 9.51 , Apogeo2: 0.00 , Tiempo seg: , 4 ,
{ Acc x: -0.26 , Acc y: -0.47 , Acc z: 9.51 , Apogeo: 9.53 , Apogeo2: 0.00 , Tiempo seg: , 4 ,
{ Acc x: -0.29 , Acc y: -0.46 , Acc z: 9.52 , Apogeo: 9.54 , Apogeo2: 0.00 , Tiempo seg: , 4 ,
{ Acc x: -0.27 , Acc y: -0.45 , Acc z: 9.50 , Apogeo: 9.51 , Apogeo2: 0.00 , Tiempo seg: , 4 ,
{ Acc x: -0.25 , Acc y: -0.43 , Acc z: 9.53 , Apogeo: 9.54 , Apogeo2: -0.07 , Tiempo seg: , 6 ,
{ Acc x: -0.27 , Acc y: -0.46 , Acc z: 9.50 , Apogeo: 9.51 , Apogeo2: -0.07 , Tiempo seg: , 6 ,
{ Acc x: -0.27 , Acc y: -0.45 , Acc z: 9.50 , Apogeo: 9.51 , Apogeo2: -0.07 , Tiempo seg: , 6 ,
{ Acc x: -0.21 , Acc y: -0.43 , Acc z: 9.54 , Apogeo: 9.55 , Apogeo2: 0.00 , Tiempo seg: , 6 ,
{ Acc x: -0.23 , Acc y: -0.42 , Acc z: 9.53 , Apogeo: 9.54 , Apogeo2: 0.00 , Tiempo seg: , 6 ,
{ Acc x: -0.25 , Acc y: -0.42 , Acc z: 9.49 , Apogeo: 9.50 , Apogeo2: 0.00 , Tiempo seg: , 6 ,
{ Acc x: -0.22 , Acc y: -0.44 , Acc z: 9.45 , Apogeo: 9.46 , Apogeo2: 0.00 , Tiempo seg: , 6 ,
{ Acc x: -0.26 , Acc y: -0.44 , Acc z: 9.51 , Apogeo: 9.52 , Apogeo2: 0.00 , Tiempo seg: , 6 ,
{ Acc x: -0.25 , Acc y: -0.47 , Acc z: 9.50 , Apogeo: 9.51 , Apogeo2: 0.00 , Tiempo seg: , 7 ,
{ Acc x: -0.25 , Acc y: -0.45 , Acc z: 9.50 , Apogeo: 9.51 , Apogeo2: 0.24 , Tiempo seg: , 7 ,
{ Acc x: -0.26 , Acc y: -0.45 , Acc z: 9.51 , Apogeo: 9.52 , Apogeo2: 0.24 , Tiempo seg: , 7 ,

```

3.4 Visual Studio Code

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como C++, C#, Java, Python, PHP, Go, .NET) [Code, 2022].

Visual Studio Code es un editor de texto plano desarrollado por Microsoft totalmente gratuito y de código abierto para ofrecer a los usuarios una herramienta de programación avanzada como alternativa al Bloc de Notas [Rubén, 2021].

Este editor está escrito totalmente en Electron, un framework utilizado para unir Chromium y Node.js en forma de aplicación de escritorio. No se caracteriza precisamente por un bajo consumo de memoria (ya que tiene que cargar todo el core de Chrome), pero es muy sencillo de programar, muy potente y flexible. Este editor utiliza el mismo componente editor que Visual Studio Team Services (Monaco) en lugar del clásico Atom [Rubén, 2021].

Visual Studio Code o VS Code cuenta con una variedad de características útiles para optimizar el trabajo, siendo una de las herramientas preferidas a la hora de programar y desarrollar contenido.

- **Multiplataforma:** es una herramienta tan versátil que se puede ejecutar en sistemas operativos como Windows, macOS y GNU/Linux.
- **IntelliSense:** Esta característica está relacionada con la edición de código, auto-completado y resaltado de sintaxis, lo que permite ser más ágil a la hora de escribir código. Como su nombre lo indica, proporciona sugerencias de código y terminaciones inteligentes en base a los tipos de variables, funciones, etc. Con la ayuda de extensiones se puede personalizar y conseguir un IntelliSense más completo para cualquier lenguaje.[Frankier, 2022]
- **Depuración:** esta opción detecta los errores en el código, facilitando la edición y corrección de los mismos.
- **Control de versiones:** Visual Studio Code tiene compatibilidad con Git, por lo que se pueden revisar diferencias o lo que conocemos con git diff, organizar archivos, realizar commits desde el editor, y hacer push y pull desde cualquier servicio de gestión de código fuente (SMC). Los demás SMC están disponible por medio de extensiones.[Frankier, 2022]
- **Extensiones:** Cuenta con una infinidad de extensiones que permiten programar en diferentes lenguajes, además de personalizar y agregar funciones adicionales, convirtiéndolo en una herramienta potente y con una mejor experiencia a la hora de programar.

3.5 MYSQL

En programación es prácticamente inevitable trabajar con algún tipo de sistema de gestión de bases de datos. Cualquier programa que imaginemos tarde o temprano necesitará almacenar datos en algún lugar, como mínimo para poder almacenar la lista de usuarios autorizados, sus permisos y propiedades. MySQL es un sistema de gestión basado en código fuente abierto de fácil acceso. Actualmente varios programadores de dedican al desarrollo web utilizando MySQL para la programación y gestión de datos, ya que es multihilo y multiusuario que permite ser utilizado por varias personas al mismo tiempo, además de su amplia difusión, también cuenta con una enorme comunidad que puede brindar soporte para otros usuarios [Ángel, 2019].

Algunas características de MySQL son:

- MySQL basa su funcionamiento en un modelo cliente y servidor.
- Compatibilidad con SQL, este es un lenguaje generalizado dentro de la industria. Al ser un estándar MySQL ofrece plena compatibilidad por lo que si se ha trabajado en otro motor de bases de datos no se tendrán problemas en migrar a MySQL.
- MySQL posee la característica de no procesar las tablas directamente sino que a través de procedimientos almacenados es posible incrementar la eficacia de nuestra implementación.
- En el momento que se produce un evento otro es lanzado para actualizar registros o optimizar su funcionalidad.
- El sistema de base de registros avala que todos los procedimientos se establezcan correctamente o ninguna de ellas. En caso por ejemplo de una falla de energía, cuando el monitor falla u ocurre algún otro inconveniente, el sistema opta por preservar la integridad de la base de datos resguardando la información [Ángel, 2019].

3.6 PYTHON

Python es un lenguaje de secuencias de comandos, interpretado o de script que apunta a objetos y listo para ejecutar todo tipo de programas. Además se ejecuta utilizando un programa intermedio o intérprete, en lugar de compilar el código a lenguaje máquina que pueda comprender y ejecutar directamente una computadora (lenguajes compilados).

Algunas de las características que han llevado a Python en convertirse en un programa con una sintaxis muy limpia y a la vez legible son:

- **Tipado dinámico:** se refiere a que no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que su tipo se determinará en tiempo de ejecución según el tipo del valor al que se asigne, y el tipo de esta variable puede cambiar si se le asigna un valor de otro tipo.
- **Fuertemente tipado:** No se permite tratar a una variable como si fuera de un tipo distinto al que tiene, es necesario convertir de forma explícita dicha variable al nuevo tipo previamente.
- **Multiplataforma:** El intérprete de Python está disponible en multitud de plataformas (UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS, etc.) por lo que si no se utilizan librerías específicas de cada plataforma el programa no podrá correr en todos estos sistemas sin grandes cambios.

- **Orientado a objetos:** La ejecución del programa consiste en una serie de interacciones entre los objetos. Python también permite la programación imperativa, programación funcional y programación orientada a aspectos [Raúl, 2011].

3.7 PHP

PHP son las siglas en inglés del acrónimo Hypertext Pre-Processor, es decir, pre-procesador de hipertexto [Alex, 2019]. Es un lenguaje de script que se ejecuta del lado del servidor, este se incluye en una página HTML normal. A diferencia de un lenguaje como JavaScript, donde el código se ejecuta del lado del cliente (en el explorador), en PHP dicho código se ejecuta del lado del servidor. Al ejecutarse se incrusta en la página HTML, que es enviada al navegador, el cual no tiene conocimiento del procesamiento que se está llevando en el servidor.

La técnica permite realizar páginas web dinámicas, generando contenido ya sea de forma total o parcialmente al realizar la llamada de la página, gracias a la información recopilada en un formulario o se extrae de una base de datos [Olivier, 2016].

3.8 HTML

HTML es un lenguaje artificial que los ordenadores son capaces de interpretar y diseñado para que los programadores redacten instrucciones que los navegadores ejecutan para originar la página web. Por lo tanto este es un lenguaje de programación, o un "idioma que la máquina entiende y procesa para dar una respuesta" [Vértice, 2009].

Las siglas HTML significan HyperText Markup Language (Lenguaje de marcas de hipertexto). Este código estructura y despliega una página web y sus contenidos, es decir posee referencias como hipervínculos, links o enlaces a otro texto. En conclusión, el hipertexto es el texto que se pulsa con el mouse del ordenador, redirigiendo a otro texto, tabla, imagen u otro elemento.

3.9 CSS

CSS (Cascading Style Sheets) u Hojas de estilo en cascada en español, no es realmente un lenguaje de programación, tampoco es un lenguaje de marcado. Es un lenguaje de hojas de estilo, es decir, permite aplicar estilos de manera selectiva a elementos en documentos HTML. Por ejemplo, para seleccionar todos los elementos de párrafo en una página HTML y volver el texto dentro de ellos de color rojo [MDN, 2022].

Con CSS se pueden crear reglas para visualizar un sitio web cómo se desea mostrar la información y guardar los comandos para elementos de estilo (fuentes, colores, tamaños, etc.) separados de los que configuran el contenido. Además, se pueden crear formatos específicos útiles para comunicar las ideas y producir experiencias más agradables visualmente para los usuarios del sitio web. Facilitando procesos de edición para cualquier página web de modo que se resalten algunas de sus ventajas:

- El lenguaje CSS tiene la ventaja de ser mucho más simple, por lo que implica menos código y probabilidad de errores, así como una mayor velocidad de carga y facilidad de lectura.
- **Optimiza la edición:** CSS permite crear estilos que pueden aplicarse a todas las páginas de un sitio web. Esto ahorra tiempo y posibilita crear una imagen de la marca a través de tipografías, colores y recursos visuales.
- **Facilita la accesibilidad al usuario:** Adecuar la plataforma para medios de acceso como teléfonos, tabletas, computadoras de escritorio o laptops puede resultar una tarea complicada. CSS tiene la ventaja de facilitar la accesibilidad al usuario, gracias a las hojas de estilo estandarizadas.
- **Promueve la creatividad:** Acelera el proceso de personalización de los sitios, que pueden ser construidos con especificaciones claras o bien sujetarse a las particularidades de los distintos navegadores.
- **Prioriza la limpieza del código:** CSS puede separar todo el código relacionado con el estilo de un sitio web del contenido base de una página, que puede estar en lenguaje HTML. De este modo, mantiene limpieza en ambos conjuntos de información y evita tener líneas de código de contenido interfiriendo entre sí [Diego, 2022].

4 Sistema desarrollado

A lo largo del tiempo el desarrollo de tecnologías para obtener diseños, procesos y técnicas de servicios o aplicaciones que permiten transmitir datos, en la ciencia ha sido gracias a la telemática, esta rama engloba conocimientos de ingeniería como la informática, comunicación e instrumentación. Siendo el punto de partida, dentro del grupo de investigación CEMOS se encuentra el semillero de cohetaría SCUA, el cual está adelantando dos misiones para el concurso que se lleva a cabo cada año en Brasil que tiene como nombre LASC, para el año 2022 el semillero de investigación presenta la propuesta para la implementación de un dashboard en el cual se requiere una herramienta de gestión de la información que monitorea, analiza y muestra de manera visual los indicadores de sus misiones. La primera misión es ‘Autonomus UAV Cansat for Meteorological Studies’ ó GuaneSat y desarrolla el monitoreo de la contaminación atmosférica de forma autónoma, para ello requiere de una interfaz de visualización en la que se muestre de forma fácil: visualización del vuelo, control del registro de los sensores y manejo del satélite desde tierra. La segunda misión es ‘Rocket vehicle for 4 km payload’ designandose como Orión, este cohete alcanzará una altura de 3 kilómetros y llevará una carga útil con masa de aproximadamente 4 kilogramos, también llevará un vehículo aéreo no tripulado que se eyectará a 500 metros de altura. Lo anterior permite tener un enfoque claro de todo el desarrollo de este proyecto.

4.1 Requerimientos del Sistema

Durante el desarrollo del diseño es necesario contar con operaciones del Hardware y software, así que a continuación se definen cada una de estas.

4.1.1 Requerimientos de Hardware

Aclarando lo que se expuso en el **Planteamiento de la Misión**, se consideran dos campos importantes: las misiones Orión y GuaneSat. Para una comprensión mas idónea se desglosan los requerimientos de cada una de estas.

1. **Misión GuaneSat:** La carga útil será sometida a temperaturas y presiones como las que se muestran en la tabla 4, por lo tanto se requiere del uso de componentes que sean aptos para vuelos espaciales y que funcionen bajo dichas condiciones que se presentan durante el vuelo en la atmósfera.
2. **Misión Orión:** El cohete Orión será sometido a altas velocidades, temperaturas y presiones, por lo que es necesario que el uso de componentes sean aptos para este tipo de vuelos espaciales y que funcionen bajo condiciones presentes durante el

Tabla 4: *Condiciones ambientales misión GuaneSat*

Variable	Bucaramanga	Tatuí
Humedad	77% y 83%	66.68% a 79.99%
Temperatura	19°C a 27°C	16°C a 28°C
Presión	1015 hPa	1016 hPa
Monóxido de carbono	186.69 $\mu\text{g}/\text{m}^3$	245.75 $\mu\text{g}/\text{m}^3$
Compuestos orgánicos volátiles	8.47 $\mu\text{g}/\text{m}^3$	PM2.5
Dióxido de nitrógeno	1.65 $\mu\text{g}/\text{m}^3$	2.73 $\mu\text{g}/\text{m}^3$
Campo magnético	30.000 nT	24.000 nT

vuelo en la atmósfera. Tomando en consideración las condiciones ambientales que se esponen en la tabla 5.

Tabla 5: *Condiciones ambientales misión Orión*

Variable	Bucaramanga	Tatuí
Temperatura	19°C a 27°C	16°C a 28°C
Presión	1015 hPa	1016 hPa

3. **Banco de pruebas:** Adicionalmente para el correcto funcionamiento del motor es importante hacer pruebas con este, para eso se recopilan y estudian los datos de este banco. Las características a tener en cuenta son la Presión y temperatura, esto va a depender específicamente del motor, mas no de las condiciones del ambiente.

4.1.2 Requerimientos de Software

Para los requerimientos de software es necesario contar con la tecnología que permita al sistema realizar las actividades que se enumeran:

1. Operar en modo manual (pruebas en tierra) y modo automático (en vuelo).
2. Verificar en tiempo real el estado de las conexiones presentes: alimentación, sensores y memoria.
3. Comunicar el estado de las conexiones del sistema en tiempo real (funcionamiento correcto).
4. Ejecutar órdenes de lectura y escritura en segundo plano, y en diversos protocolos de comunicación.
5. Clasificar data.
6. Almacenar datos en sus respectivas bases de datos.

7. Permitir la visualización del rendimiento del sistema.
8. Operar en diversas fases de la misión (Despegue, máxima aceleración, apogeo, despliegue paracaídas estabilización, despliegue paracaídas principal, recuperación y control CanSat).

4.2 Arquitectura del Sistema

En la arquitectura del sistema es indispensable cumplir con ciertos requisitos para que esta herramienta sea viable y ejecutada correctamente; el sistema debe estar al alcance de un trabajo con tecnología distribuida, es decir, donde los objetos puedan interactuar entre ellos, escalable, eficiente y segura.

La base fundamental de un sistema IoT es poder distribuir objetos, sensores y dispositivos conectados en todo nuestro entorno, esto conlleva a que la información adquirida puede provenir de diferentes lugares y es procesada por un servidor. Todo esto implica que se va a contar con variedad de sensores, separados físicamente pero conectados entre si por una red de comunicación. Cada componente dispone de su propio software y hardware los cuales se muestran en la tabla 6.

Tabla 6: *Software y Hardware*

Misión	Variable	Software	Hardware
Orión	Presión	MS5611	ESP32 Arduino IDE
Orión	Aceleración	BNO055	ESP32 Arduino IDE
Orión	Posición	BNO055	ESP32 Arduino IDE
GuaneSat	Imagen	ESP32CAM OV2640-160 ^o	ESP32 Arduino IDE
GuaneSat	Humedad	SHT31	ESP32 Arduino IDE
GuaneSat	Temperatura	SHT31	ESP32 Arduino IDE
GuaneSat	CO2	MH-Z14A	ESP32 Arduino IDE
GuaneSat	NO2		ESP32 Arduino IDE
GuaneSat	CO2	MiCS-6814	ESP32 Arduino IDE
GuaneSat	NH2	MiCS-6814	ESP32 Arduino IDE
GuaneSat	Orgánicos	CCS811	ESP32 Arduino IDE
GuaneSat	Polvo y material particulado	PMS7003	ESP32 Arduino IDE
GuaneSat	Aceleración	MPU6050	ESP32 Arduino IDE
GuaneSat	Magnetómetro	HMC5983	ESP32 Arduino IDE
GuaneSat	Presión	BMP180	ESP32 Arduino IDE
GuaneSat	GPS	NEO6	ESP32 Arduino IDE

La arquitectura de la interfaz gráfica debe ser capaz de mostrar todos los componentes como un único sistema a los ojos de los usuarios, por esta razón se analiza desde la

recepción de la información, el procesamiento de esta, formatos de llegada a las bases de datos locales, base de datos en el servidor local y su visualización en los diferentes dashboard a usar, cosa que se clarifica en la figura 8.

Figura 8: *Arquitectura con base a las misiones.*

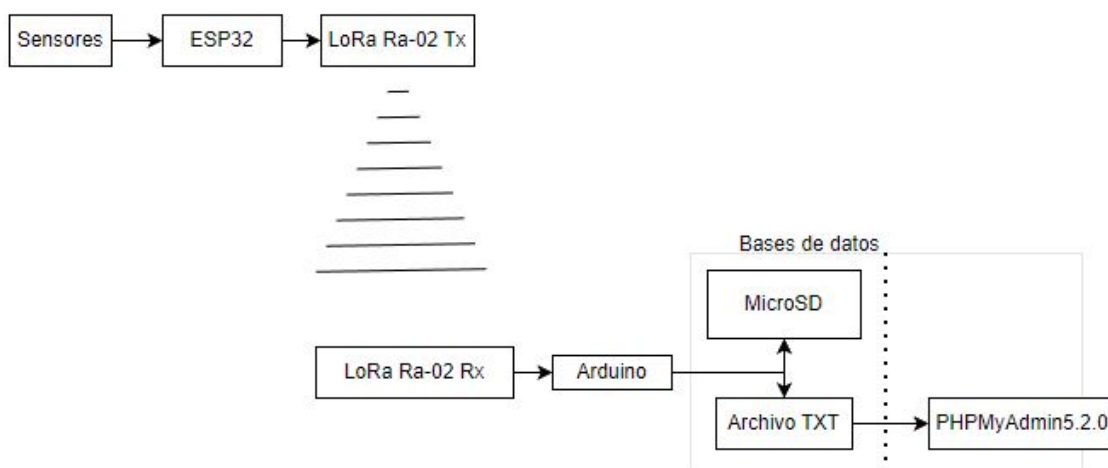


El sistema desarrollado en la fase de comunicación tiene un comportamiento similar en la misión Orión y la misión GuaneSat, en la figura 9, se puede ver el diagrama básico para este sistema de comunicación desde el sensado de la variable, hasta la llegada a las bases de datos.

El Arduino optado para la recepción de información en la base terrena es Arduino Nano de la familia Arduino Nano Every, los cuales están orientados al IoT, su voltaje de operación se reduce de 5 V a 3,3 V, cuenta con un microcontrolador tipo ATmega4809 con un 50 % más de memoria Flash (48 KB), un 200 % más de memoria SRAM (6 KB) y una velocidad de reloj mejorada de 20 MHz e incorpora un conector micro-USB y gracias a su tamaño compacto es ideal para la función designada.

La ESP32 con la que se cuenta en la transmisión de información en la aviónica, es de la familia de chips SoC de bajo coste y consumo de energía, con tecnología Wi-Fi y Bluetooth, emplea un microprocesador Tensilica Xtensa LX6, operando a 160 o 240 MHz con memoria de 520 KiB tipo SRAM, cuenta con protocolos 4 SPI, 2 interfaces I2S, 2 interfaces I2C, 3 UART.

Figura 9: *Arquitectura llegada de datos.*



4.2.1 Requerimientos funcionales y no funcionales

Se puede pensar en un requisito funcional como una característica u actividad que un sistema deba realizar, es decir, como se comporta o funciona específicamente dicho sistema u software, siendo este último el caso de estudio y con el cual se cumplen ciertas condiciones.

- **¿Qué son los requisitos funcionales?**

Citando a [Ricardo, 2018] “Los requerimientos funcionales son las descripciones explícitas del comportamiento que debe tener una solución de software y que información debe manejar.”, lo que se traduce en lo que el sistema debe hacer para satisfacer las necesidades o expectativas del usuario. Los requisitos funcionales se pueden considerar como características que el usuario detecta. Estos difieren de los requisitos no funcionales, que definen cómo debe funcionar internamente el sistema (p. Ej., rendimiento, seguridad, etc).

Los requisitos funcionales se componen de dos partes: función y comportamiento. La función es lo que hace el sistema (por ejemplo, “Visualización en tiempo real de la lectura

de datos”). El comportamiento es cómo lo hace el sistema (p. Ej., “El sistema mostrará esta data de forma ordenada, puesto que ha sido tratada por diferentes algoritmos desarrollados y los cuales se han ido mencionando”).

Algunas características que deben seguir los requisitos funcionales se mencionan a continuación:

- Son declaraciones de los servicios que debe proporcionar el sistema.
- Especifican la manera en que este debe reaccionar a determinadas entradas.
- Especifican cómo debe comportarse el sistema en situaciones particulares.
- Pueden declarar explícitamente lo que el sistema no debe hacer.

En consecuencia, los requisitos funcionales del sistema propuesto son los siguientes:

- Visualizar información
- Descargar información
- Guardar información
- Facilidad de comprensión, aprendizaje y uso
- Procesar datos (mostrando gráficos)
- Consultar estadísticas
- Actualizar información en tiempo real
- Diseño ergonómico mediante el establecimiento de menús, barras de acciones e iconos de fácil acceso

Puesto que los requisitos funcionales especifican lo que debe hacer un sistema, los requisitos no funcionales describen cómo lo hará. Por ejemplo, la nueva aplicación nos proporcionará la data procesada, en gráficos, tablas y comandos de control. Eso es parte de los requisitos funcionales.

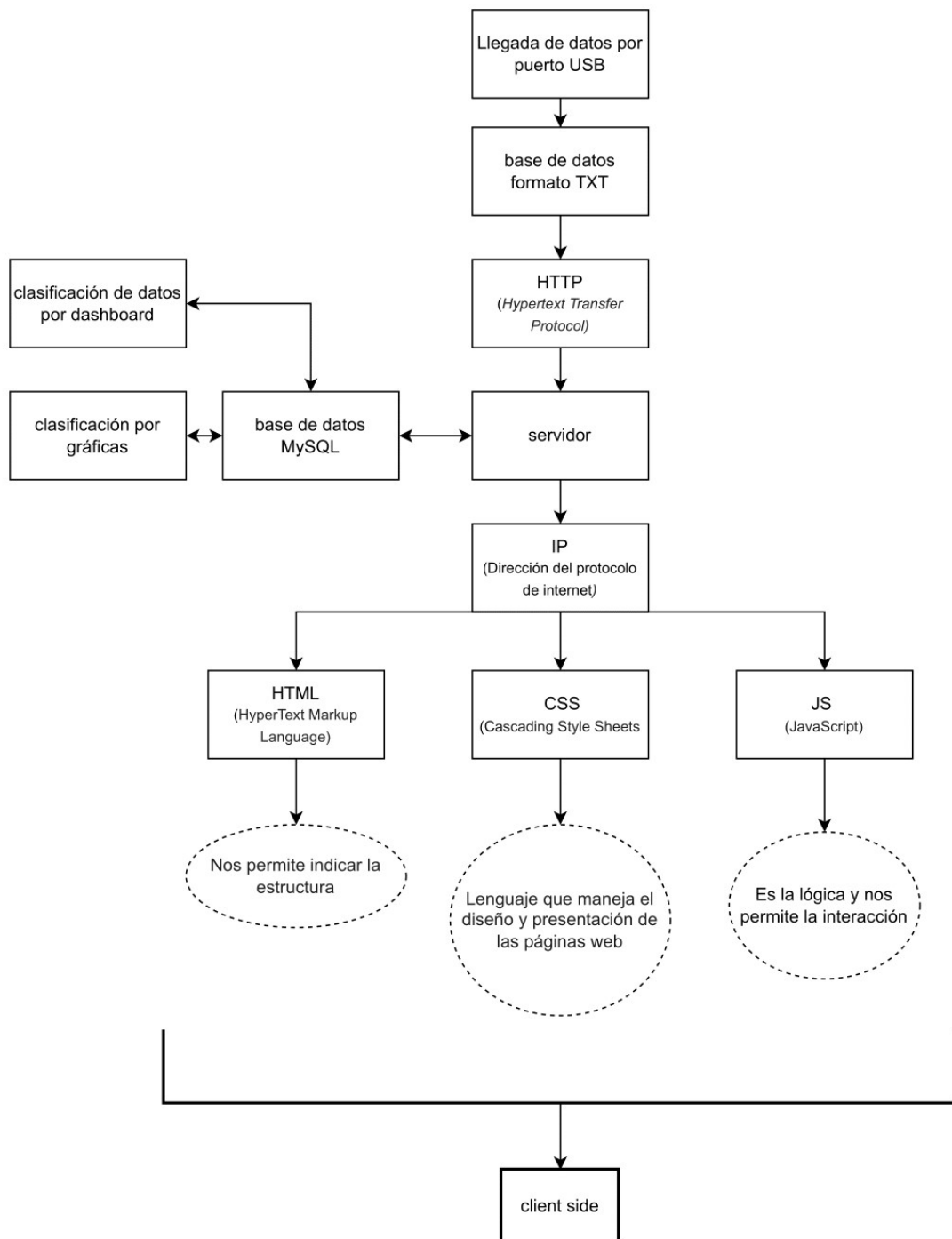
4.2.2 Algoritmo de procesamiento

Para el algoritmo de procesamiento primero se considera que el desarrollo del proyecto se debe hacer con un proceso de ejecución web Full Stack lo que se refiere a toda la profundidad de una aplicación de sistema informático, y los desarrolladores de pila completa abarcan dos dominios de desarrollo web separados: el front-end y el back-end. El front-end incluye todo lo que un cliente, o visitante del sitio, puede ver e interactuar. Por el contrario, el back-end se refiere a todos los servidores, bases de datos y otra arquitectura interna que impulsa la aplicación; por lo general, el usuario final nunca interactúa directamente con este reino [University, 2020]. El cual se aplica para mantener y optimizar cualquier sitio web, en la figura 10, se muestra el diagrama de flujo ejecutado para el full stack que se desarrolla en este proyecto.

El algoritmo de procesamiento para la misión Orión y Ganesat se define en 3 niveles de desarrollo: Nivel 1, back-end enfocado a tratamiento en bases de datos. Nivel 2, front-end enfocado a visualización diseño y actualización de data tratada. Nivel 3, bases de datos enfocado a historiales de las misiones. Lo anterior, con el fin de asignar una estructura para tareas designadas y poder tener vinculación asertiva del sistema.

Cabe resaltar que la arquitectura planteada no contempla el sistema en un servidor web, esto será propuesto como trabajo futuro en dicha sección. Es importante tener en cuenta que los terminos back-end y front-end son dos términos usados en el desarrollo de productos tecnológicos en el mundo IOT para el desarrollo web, pero son bastante diferentes entre sí. Cada lado debe comunicarse y operar de manera efectiva con el otro como un solo conjunto para mejorar la funcionalidad del sitio web.

- **Back End:** Hace referencia al proceso del servidor. Es la encargada del sistema que recoge y guarda información y la organiza. Para estas tareas en el proyecto son usados los lenguajes PHP y Python.
- **Fronnd End:** Es el encargado de que el usuario interactúe directamente, esto se denomina interfaz. También se conoce como el 'lado del cliente o client side' de la aplicación. Incluye todo lo que los usuarios experimentan directamente: colores y estilos de texto, imágenes, gráficos y tablas, botones y menú de navegación. HTML, CSS y JavaScript son los lenguajes utilizados para el desarrollo de este Front-end.
- **Bases de datos históricos:** Es una estructura fundamental para una administración eficiente de datos en el campo de la investigación y tecnología. Es un conjunto de ficheros interrelacionados con distribución determinada por el proceso.

Figura 10: *Proceso de ejecución Full Stack.*

Las bases de datos se clasifican en bases de datos dinámicas y estáticas, donde los datos se actualizan o incluso se modifican y otras donde están referidas a las que no pueden modificarse respectivamente.

4.3 Diseño de Etapas del Sistema

Tomando como punto de partida los niveles planteados previamente, se realiza la definición de las etapas que integran todo el desarrollo de la arquitectura del sistema propuesto en la figura 8: Etapa para la validación del sistema de comunicación, etapa de procesamiento de datos (back-end), etapa de visualización (frond-end), almacenamiento de data (historial), validación de variables sensadas basada en tablas. En consecuencia, para cada etapa se define el proceso de ejecución y se sustenta el ¿por qué? de esta base a los requerimientos de hardware y software.

4.3.1 Etapa para la validación del sistema de comunicación

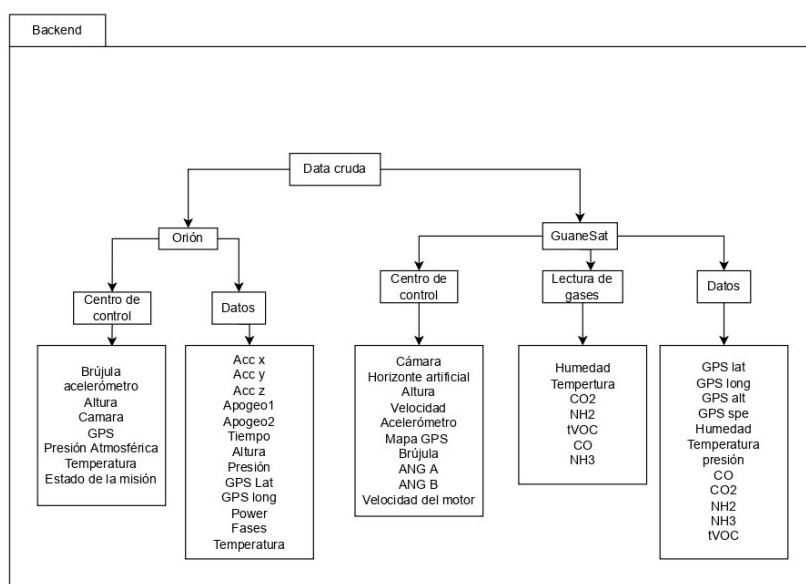
La principal importancia de esta etapa es conseguir la forma correcta de recepción del mensaje que viene de la misión Orión, para esto se tienen en cuenta varios procesos como lo son:

- Recepción de data en tiempo real, esta data es enviada desde la aviónica instalada en el cohete, mediante la antena transmisora y es recibida en la estación terrena por la antena receptora y almacenada en dos partes, un archivo .txt para su tratamiento en la interfaz gráfica y un respaldo de seguridad en una MicroSD. El bus de datos destinado tiene una estructura definida en la trama de datos nombrada anteriormente para poder garantizar el almacenamiento de cada uno de los datos que llegan.
- Definición de una trama ideal para el bus de datos (esta se especifica mas a fondo en la sección de procesamiento de datos)
- Elección de la antena adecuada para la transmisión y la recepción.

4.3.2 Etapa de procesamiento de datos

En detalle, en la figura 11 se desglosa la distribución principal del nivel de back-end con el objetivo de aclarar su función, conexión y direccionamiento de datos para el proceso de ejecución.

Como complemento de la estructura base de la distribución de la data, se establece el diagrama en lenguaje unificado de modelado (UML), siendo una herramienta que ayuda

Figura 11: *Arquitectura del back-end*

a captar mediante un conjunto de símbolos y diagramas a comunicar la idea del sistema, el cual se presenta en el apéndice C de este texto.

Partiendo de la data que se recibe para el tratamiento de datos y que comienza con una distribución en SQL de dos carpetas referidas a cada misión, cada una de ellas esta compuesta de la siguiente manera:

- Orión: distribución centro de control y datos.
- GuaneSat: centro de control, lectura de gases y datos.

En la **sección 3.2** sobre la comunicación, se relaciona sobre la trama de datos para cada una de las misiones, con el fin de entender el orden de los datos recibidos y a qué dato hace referencia cada número. En los paquetes de información que se reciben en el puerto hay datos sin tratamiento a los cuales se le hace su respectivo tratamiento, en las figuras 12 y 13 se puntualizan cada uno de los pasos para cumplir con este tratamiento.

4.3.3 Etapa de visualización

Su función va de la mano con ser el puente de enlace entre el usuario con los datos y la visualización de estos. Para su desarrollo se requiere del uso de tres lenguajes de programación HTML, CSS y JS. Cada uno de estos tiene una estructuración independiente y en la figura 14 se encuentran funcionalidades para cada uno de ellos.

Figura 12: *Tratamiento de datos*



Figura 13: *Tratamiento de datos*

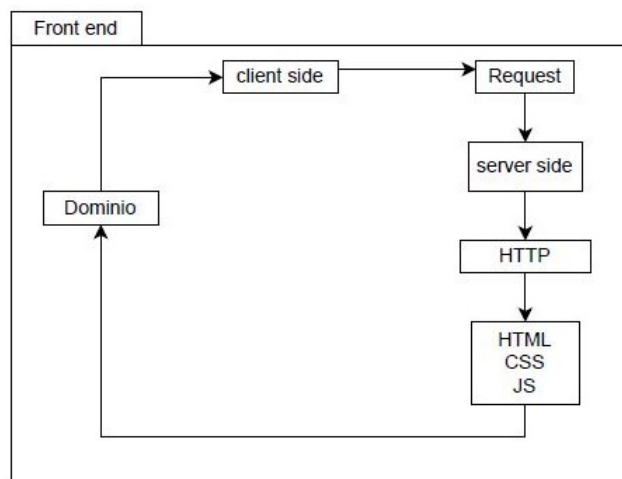


Figura 14: *Tareas de lenguajes de programación*

4.3.4 Etapa de Almacenamiento de Data

Tiene como única función proveer al usuario un historial de toda la data obtenida durante las misiones, con el fin de servir para estudios de las lecturas en el campo medioambiental y corroborar el comportamiento de los sensores adquiridos para futuras mejoras.

La estructura para presentar el almacenamiento de los datos por cada una de las misiones, se estructura en: la tabla 7 para el historial de la misión Orión, tabla 8 para el historial de la misión GuaneSat y en la tabla 9 para el historial de un banco de pruebas del motor del cual no se hace énfasis. Estas estructuras son sugeridas de forma legible, fácil de identificar para cada una de las misiones y el banco de pruebas del motor.

Tabla 7: *Historial data misión Orión*

Orión															
Accx	Accy	Accz	Apogeo1	Apogeo2	Tiempo	Altura	Presión	GPSlat	GPSlong	Power	Fases	temp	girox	giroy	giroz

Tabla 8: *Historial data misión GuaneSat*

GuaneSat											
GPSlat	GPSlong	GPSalt	GPSspe	Hum	Temp	Pres	CO	CO2	NH2	NH3	tVOC

Tabla 9: *Historial Banco de pruebas*

Banco de pruebas			
Empuje	Temp 1	Temp 2	Temp 3

4.3.5 Etapa de validación de variables sensadas

Esta etapa enfatiza principalmente en la importancia de la validación como un proceso articulado que debe trascender de la confiabilidad a la validez, condiciones indispensables en todo proceso de medición en la investigación científica. Centrándose, en la creación de un referente conceptual relacionado con el diseño y el ecosistema.

Un instrumento de medida es una técnica o conjunto de técnicas que permiten una asignación numérica que cuantifica las manifestaciones de un constructo que es medible solo de manera indirecta [Martínez Ramírez, 2019]. Es importante tener claridad de los objetivos y posicionamientos teóricos de la investigación, y en este proyecto es la tarea de la cual parte la elección de los sensores y además saber si estos funcionan adecuadamente para las situaciones en las que serán empleados. Por ello se aclaran tanto las condiciones ambientales presentes en la ciudad de Bucaramanga como en Tatuí de acuerdo con la 10.

Tabla 10: *Condiciones ambientales misión Orión*

Misión Orión		
Variable	Ambiente	
	Bucaramanga	Tatuí
Temperatura	19°C a 27°C	16°C a 28°C
Presión	1015 hPa	1016 hPa
Monóxido de carbono	186.69 $\mu\text{g}/\text{m}^3$	245.75 $\mu\text{g}/\text{m}^3$
Compuestos orgánicos volátiles	8.47 $\mu\text{g}/\text{m}^3$	PM2.5
Dióxido de nitrógeno	1.65 $\mu\text{g}/\text{m}^3$	2.73 $\mu\text{g}/\text{m}^3$
Campo magnético	30.000 nT	24.000 nT

En la tabla 11 se presenta la guía de observación para los instrumentos y un resumen de variables ya registradas en la web que se emplean con la finalidad de validar correctamente la medidas que se esperan registrar en cada misión.

En esta guía se compila la información de los sensores a usar donde cada ítem representa:

- Guía de observación para el instrumento: se encuentra el nombre del sensor o instrumento a estudiar

Tabla 11: *Guía de observación para sensores e instrumentos*

Guía de observación para instrumentos			
Objetivo de la guía			
Objetivo de la investigación			
Rango del instrumento			
Criterios a evaluar			
Criterio	Si	No	Observación
La medición se encuentra en los rangos aceptables			
El sensor tiene comportamiento estable con el resto del sistema			
Contribuye a los objetivos de la investigación			
Contribuye a medir el constructo en estudio			
Cantidad de unidades es clara para el estudio			
Coherente para uso en Bucaramanga			
Coherente para uso en Tatuí			

- Objetivo de la guía: medida que se va a estudiar.
- Objetivo de investigación: Misión en donde se encuentra
- Rangos del instrumento: rangos según datasheet
- La medición se encuentra en los rangos aceptables: si está la medición del medio ambiente entre los rangos permitidos por el sensor.
- El sensor tiene comportamiento estable con el resto del sistema: si /no
- Contribuye con los objetivos de la investigación: El sensor o instrumento aporta a la investigación si/no
- Contribuye a medir el constructo del estudio: cumple con objetos autónomos, aun cuando no tengan existencia real si/no
- Cantidad de unidades es clara para el estudio: si/no presenta unidades necesarias en su medición para tener exactitud en su medida
- Coherente para uso en Bucaramanga: si/no la medida promedio en Bucaramanga está entre los rangos del sensor o instrumento
- Coherente para uso en Tatuí: si/no la medida promedio en Tatuí está entre los rangos del sensor o instrumento.

4.3.6 Algoritmo de Procesamiento

La etapa de procesamiento de datos es de vital importancia para recopilar la información de la etapa de sensado y como su nombre lo indica, procesar la información obtenida de cada sensor. EL algoritmo de procesamiento comprende una metodología básica, concepto básicos y asignaciones concretas para resolver el problema a cabalidad.

El algoritmo de procesamiento está desarrollado de forma precisa, definida y finita, a continuación se explica este desarrollo:

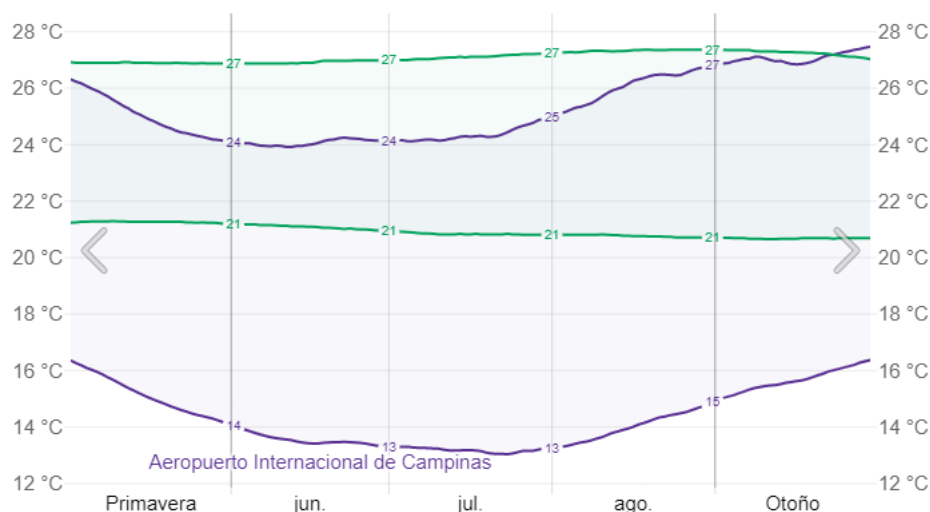
- Recibiendo en el puerto USB de forma correcta el vector de datos.
- Se tiene una conexión directa con el vector de datos por medio de python y las librerías de python flask, las cuales son: serial, matplotlib.pyplot, threading.
- Usando el comando numpy desde python generamos las matrices, llenamos y organizamos acorde a la tabla global de datos dependiendo de las estructuras condicionales sujetas de la información sin procesar recibida.
- Se establece una conexión con la base de datos de MySQL mediante sentencias SQL y python-flask que es usado para las condiciones en tiempo real.

5 Simulación del Sistema

Uno de los pasos más importantes al momento de llevar a cabo un proyecto de investigación consiste en determinar el mockup, es decir la estructura a donde se desea llegar, teniendo en cuenta parámetros y variables presentes en el campo de estudio para analizar su posible comportamiento en el montaje experimental. Tomando en cuenta lo anterior, este proyecto no está exento de un mockup para el dashboard final y herramienta de validación de datos. Esta sección se encuentra segmentada en dos modelos de simulación con ayuda de los datos presentes en fuentes como ‘The Weather Channel’ [the weather Channel, 2022] e ‘IDEAM’ [IDEAM, 2022]. A continuación se presenta la información recolectada en este campo por las empresas nombradas anteriormente, datos usados para la validación y buen funcionamiento de los sensores y que cada uno de estos cuenta con rangos de funcionamiento específicos desde fábrica.

En las figuras 15 y 16, se observa que las temperaturas ambiente para los meses de junio, julio y agosto están en un rango de 12°C a 28°C aproximadamente. El color verde es asignado a Bucaramanga y color morado asignado a Brasil (datos tomados de aeropuertos cercanos a cada una de las ciudades). Igualmente se complementa la información con temperaturas registradas en el mes de julio en Bucaramanga.

Figura 15: *Comparación datos históricos de temperatura máximas y mínimas en los meses de Junio, Julio y Agosto del año 2022 en Brasil y Bucaramanga-Colombia (Fuente:[the weather Channel, 2022])*



En la figura 17 se observa que la Humedad para los meses de Junio, Julio y Agosto están en un rango de 82 a 0 % aproximadamente. El color verde es asignado a Bucaramanga y el color morado asignado a Brasil (datos tomados de aeropuertos cercanos a cada una de las ciudades).

Figura 16: *Condiciones de Temperatura en el mes de Julio de 2022 en Bucaramanga* (Fuente:[IDEAM, 2022])

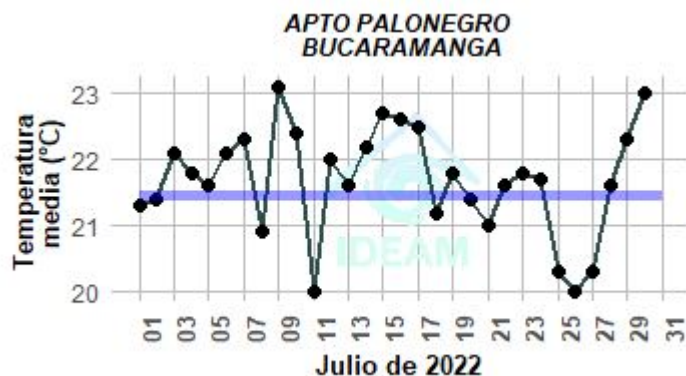


Figura 17: *Condiciones de Humedad en meses de Junio, Julio y Agosto de 2022 en Brasil comparada con Bucaramanga* (Fuente:[the weather Channel, 2022])

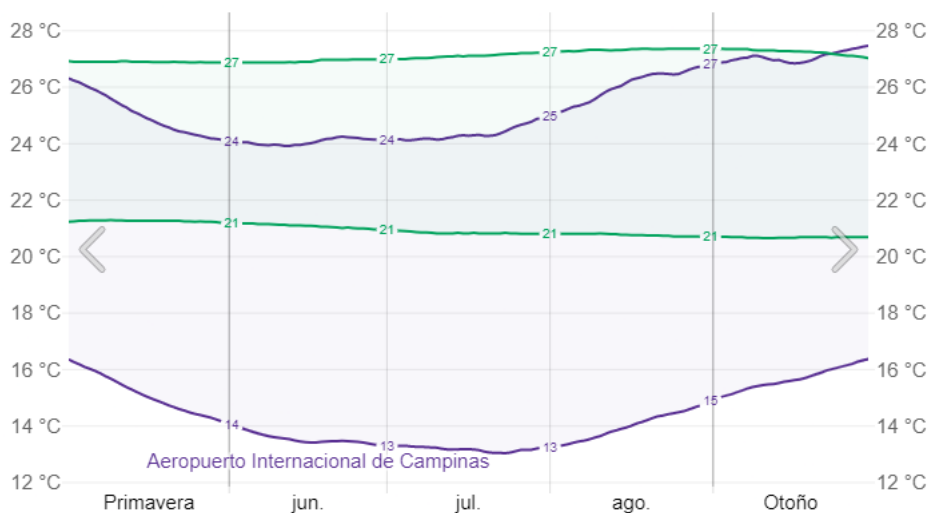
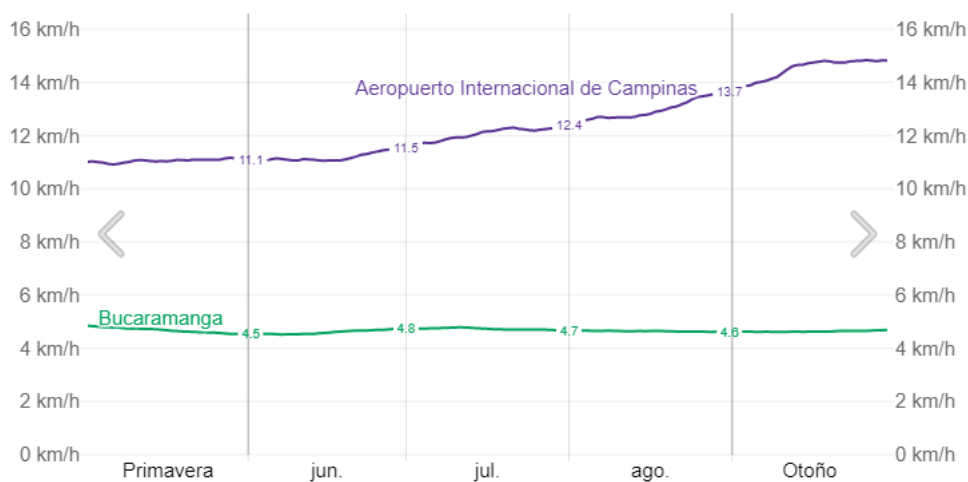


Figura 18: *Velocidad media del viento en meses de Junio, Julio y Agosto de 2022 en Brasil comparada con Bucaramanga* (Fuente:[the weather Channel, 2022])



En la figura 18 se observa los vientos en los meses de Junio, Julio y Agosto en un rango de 4 a 15 Km/h aproximadamente. El color verde es asignado a Bucaramanga y el color morado asignado a Brasil (datos tomados de aeropuertos cercanos a cada una de las ciudades).

La segunda parte que se especifica en esta sección comprende la presentación del mockup de las pantallas que se tendrán, siendo muy importante debido a que hay que contar con la información de los vuelos de una forma clara y concisa. Los criterios a tener en cuenta para este diseño son:

- Claridad: Debe transmitir la información de manera precisa para evitar que el usuario cometa errores durante la interacción.
- Conciso: Se pretende darle al usuario solo la información que necesita y pide.
- Coherencia: La interfaz debe ser creada de forma intuitiva, permitiéndole a la persona distinguir patrones de uso de manera sencilla y práctica.
- Atractivo visual: Debe contar con la importancia del atractivo visual. Además de tener grandes funcionalidades, una buena interfaz es agradable a la vista.

5.1 Prototipo centro de control

En este proyecto se plantean dos estilos generales de mockup:

- En la figura 19 se presenta el primer boceto propuesto para el mockup, es un estilo general en el que se plantean dos cámaras, una para la misión Orión y otra para la misión GuaneSat, alrededor de estos se hace una distribución de las gráficas de los datos relevantes para el momento del vuelo, como por ejemplo: Presión atmosférica, Altura, temperatura, giroscopio, aceleraciones, Raw Data, variables sensadas, estados de la misión y los escudos de la Universidad Industrial de Santander y el semillero de investigación SCUA.
- En las figuras 20, 21 y 22 se señala el segundo boceto para el mockup, el cual está compuesto por varias pantallas para la visualización de cada una de las misiones y la visualización de los datos organizados en tablas, gráficas y elementos interactivos.

Con estas distribuciones de diseño se pretende tener un acceso en 3 pantallas diferentes al mismo tiempo. La pantalla asignada para la aviónica está distribuida únicamente con información de la misión Orión en donde se tienen elementos como la imagen de la cámara, brújula, Raw data, estados de la misión, acelerómetro, altímetro, barómetro

Figura 19: *Primer dashboard mockup*



Figura 20: *Segundo dashboard mockup, pantalla aviónica*

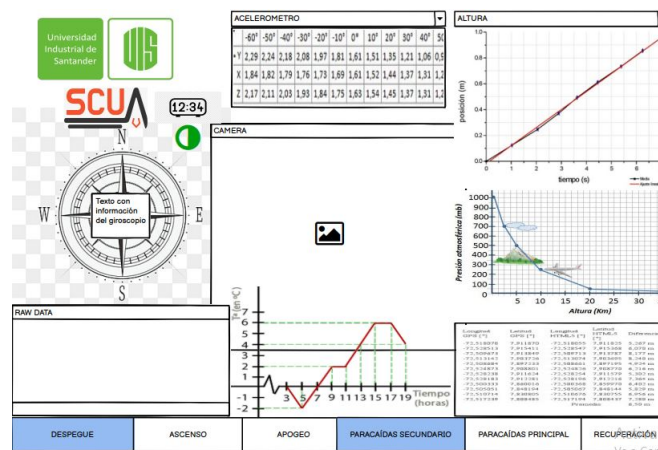


Figura 21: *Segundo dashboard mockup, pantalla CanSat*

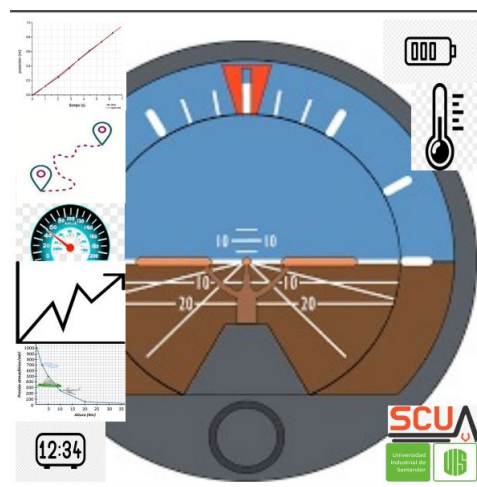
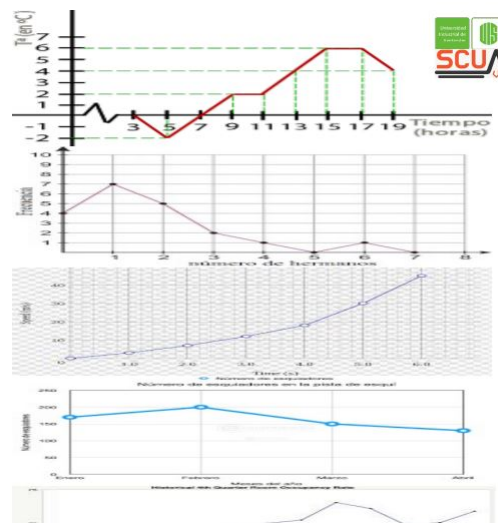


Figura 22: Segundo dashboard mockup, pantalla datos de sensores CanSat



y temperatura. En la pantalla de CanSat se detalla la información únicamente de la misión GuaneSat, en la que contiene el altímetro, ubicación, temperatura, velocidad, hora, barómetro, nivel de batería y horizonte artificial. Pantallas en las que se pretende controlar el vuelo de las misiones. Finalmente, en la pantalla ‘datos de sensor CanSat’ se consideran las gráficas de líneas de cada uno de los sensores presentes en la misión.

6 Implementación Del Sistema

Esta sección comprende la implementación del sistema de visualización. A continuación, se muestran fragmentos del código ejecutado, enfatizando en los scripts desarrollados en distintos lenguajes de programación.

6.1 Implementación etapa para la validación del sistema de comunicación

La implementación de esta sección es de vital importancia, debe considerarse la normativa para radioenlaces, configuración, velocidad de transmisión, rangos permitidos por las antenas de enlace, practicidad de transporte y la adecuada recepción de las tramas de datos. Así pues, la descripción para cada uno de estos enfoques que se tienen en cuenta para la validación del sistema de comunicación se señalan a continuación:

- Normativa nacional e internacional: En la tabla 12 se presenta a modo de resumen las principales normativas de radioenlace nacional e internacional, además en la sección de anexos, apéndice B se adjunta una tabla adicional que respecta al plan de banda.
- Configuración de radioenlace: Un factor importante a tener en cuenta en los radioenlaces es la zona de fresnel pero en el caso, como las trayectorias se presentan de forma perpendicular a la tierra no se tiene en cuenta esta zona, si es importante tener presente la línea de vista desde la estación terrena y las misiones. Además es posible calcular la pérdida de trayectoria entre un transmisor y un receptor mediante la siguiente ecuación:

$$FSPL(dB) = 20 \log_{10}(D) + 20 \log_{10}(f) + 32.44 \quad (1)$$

Donde:

- D = Distancia [km]
- f = Frecuencia [Hz]

$$FSPL = 94.65dB \quad (2)$$

Otros aspectos a examinar en los radioenlaces son: Patrón de radiación (dipolo) y su característica de ser omnidireccional. Es sabido que existe una amplia gama de antenas, en la tabla 13 se presentan diferentes tipos de antenas con el fin único de servir como apoyo en la elección de la antena correcta para su aplicación, se provee como un medio de comparación entre los diferentes tipos:

Tabla 12: *Normativa de Radioenlace nacional e internacional*

Rango	Nota Nacional	Resolución	Descripción	Nota Internacional	Resolución
430-432MHz	CLM3	https://goo.su/TGTtQ0	Normatividad relacionada con los límites de las emisiones y las condiciones técnicas y operativas tanto generales como específicas de las aplicaciones permitidas para utilizar el espectro bajo la modalidad de uso libre dentro del territorio nacional.	5,271	Atribución adicional: en Belarús, China, India, Kirguistán y Turkmenistán, la banda 420-460 MHz está también atribuida, a título secundario, al servicio de radionavegación aeronáutica (radioaltímetros). (CMR-07)
	CLM7	https://goo.su/oy2nn2S	Atribución establecida por la Unión Internacional de Telecomunicaciones (UIT) para el servicio de aficionados y aficionados por satélite - Región 2 y los planes de banda LF/MF/HF de la Unión Internacional de Radioaficionados Región II (IARU Región 2) en el Cuadro Nacional de Atribución de Bandas de Frecuencias (CN-ABF). Cuenta con reglamentación especial expedida por el Ministerio de Tecnologías de la Información y las Comunicaciones.	5,276	Atribución adicional: en Afganistán, Argelia, Arabia Saudita, Bahrein, Bangladesh, Brunei Darussalam, Burkina Faso, Djibouti, Egipto, Emiratos Árabes Unidos, Ecuador, Eritrea, Etiopía, Grecia, Guinea, India, Indonesia, Irán, Iraq, Israel, Italia, Jordania, Kenya, Kuwait, Libia, Malasia, Níger, Nigeria, Omán, Pakistán, Filipinas, Qatar, República Árabe Siria, Rep. Pop. Dem. de Corea, Singapur, Somalia, Sudán, Suiza, Tailandia, Togo, Turquía y Yemen, la banda de frecuencias 430-440 MHz está también atribuida, a título primario, al servicio fijo y las bandas de frecuencias 430-435 MHz y 438- 440 MHz están también atribuidas, excepto en Ecuador, a título primario, al servicio móvil, salvo móvil aeronáutico. (MOD CMR-15)
	CLM24	https://goo.su/sr75	Frecuencias radioeléctricas para ser utilizadas en la realización de pruebas técnicas conforme a las condiciones establecidas por el Ministerio de Tecnologías de la Información y las Comunicaciones en la normatividad que se referencia.	5,278 5,279	Categoría de servicio diferente: en Argentina, Brasil, Colombia, Costa Rica, Cuba, Guyana, Honduras, Panamá, Paraguay, Uruguay y Venezuela, la atribución de la banda de frecuencias 430-440 MHz al servicio de aficionados es a título primario (véase el número 5.33). (MOD CMR-19) Atribución adicional: en México las bandas de frecuencias 430-435 MHz y 438-440 MHz están también atribuidas, a título primario, al servicio móvil, salvo móvil aeronáutico, y a título secundario, al servicio fijo, a reserva de obtener el acuerdo indicado en el número 9.21. (MOD CMR-19)
432-438MHz	CLM3			5,271	
	CLM7			5,276	
	CLM24			5,278	
				5,279	
				5,281	Atribución adicional: en los Departamentos y colectividades franceses de Ultramar de la Región 2, y en India, la banda 433,75-434,25 MHz está también atribuida, a título primario, al servicio de operaciones espaciales (Tierra-espacio). En Francia y en Brasil esta banda se encuentra atribuida, a título secundario, al mismo servicio.
			5,282	El servicio de aficionados por satélite podrá explotarse en las bandas 435-438 MHz, 1 260-1 270 MHz, 2 400-2 450 MHz, 3 400-3 410 MHz (en las Regiones 2 y 3 solamente), y 5 650-5 670 MHz, siempre que no cause interferencia perjudicial a otros servicios explotados de conformidad con el Cuadro (véase el número 5.43). Las administraciones que autoricen tal utilización se asegurarán de que toda interferencia perjudicial causada por emisiones de una estación del servicio de aficionados por satélite sea inmediatamente eliminada, en cumplimiento de lo dispuesto en el número 25.11. La utilización de las bandas 1 260-1 270 MHz y 5 650-5 670 MHz por el servicio de aficionados por satélite se limitará al sentido Tierra-espacio	
438-440MHz	CLM3			5,271	
	CLM7			5,276	
	CLM24			5,278	
				5,279	

Tabla 13: *Tipos de Antenas*

	Patron de radiación	Ganancia	Directividad	Polarización
Dipolo	amplio	baja	baja	lineal
Dipolo multielemento	amplio	baja/media	baja	lineal
Panel plano	amplio	media	media/alta	lineal/circular
Plato parabólico	amplio	alta	alta	lineal/circular
Yagi	Endfire	media/alta	media/alta	lineal
Ranura	amplio	baja/media	baja/media	lineal
MicroStrip	Endfire	media	media	lineal

- Pruebas de distancia en radioenlace: Para efectuar estas pruebas se definen 3 rutas de diferentes distancias y perfiles geográficos, entre el edificio de pesados de la Universidad Industrial de Santander y el edificio de entrada de la carrera 30 de la misma, el edificio de petróleos de la Universidad Industrial de Santander y barrio los Pinos de la ciudad de Bucaramanga y finalmente del edificio de pesados de la Universidad Industrial de Santander y el aeropuerto Palonegro ubicado en el municipio de Lebrija - Santander. Se debe aclarar con respecto a las misiones, que al ser un lanzamiento perpendicular a la tierra este no supondrá tantos obstáculos de línea de vista, por el contrario será más despejado.

Observando las rutas propuestas, en las figuras 23 hasta 28 se presentan las comparaciones en perfiles y distancias de los radioenlaces ejecutados en Google Earth y web Ubiquiti para cada una de las distancias mencionadas.

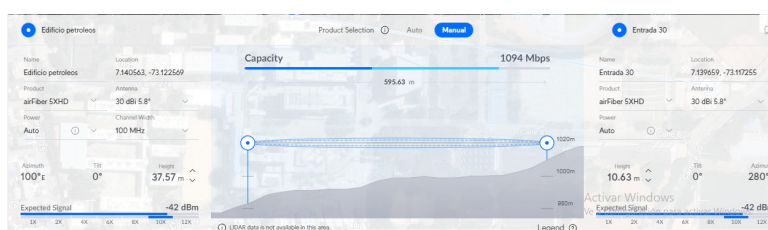
Figura 23: *Línea de vista entre la entrada de la 30 UIS y el edificio de petróleos*Figura 24: *Línea de vista entre la entrada de la 30 UIS y el edificio de petróleos con aplicativo web Ubiquiti*

Figura 25: Línea de vista entre el edificio de petróleos y barrio los Pinos



Figura 26: Línea de vista entre el edificio de petróleos y barrio los Pinos con aplicativo web Ubiquiti



Figura 27: Línea de vista entre el edificio de pesados y aeropuerto Palonegro

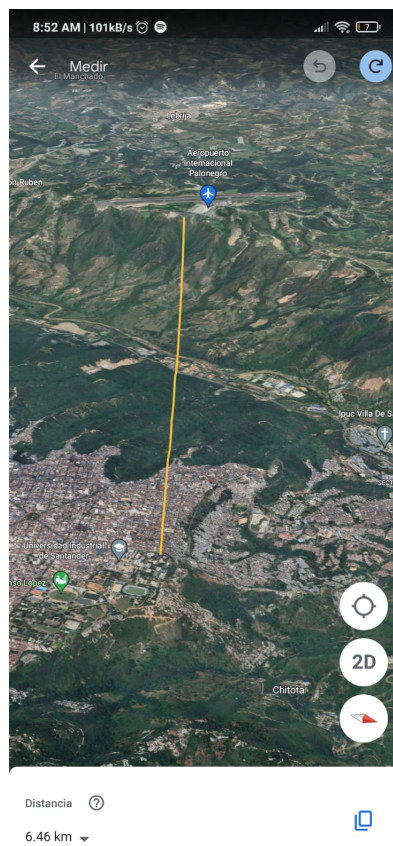
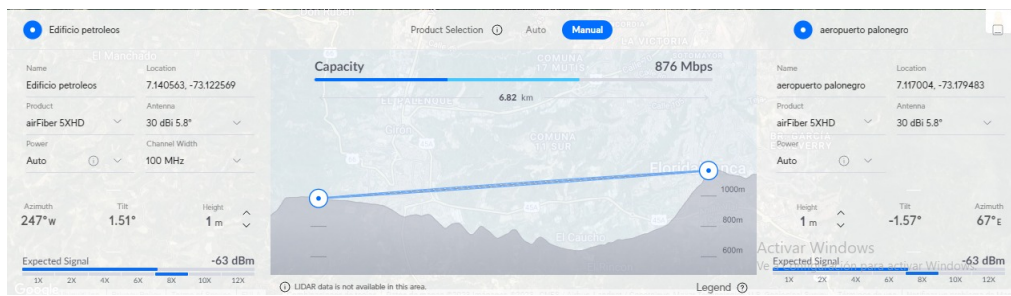
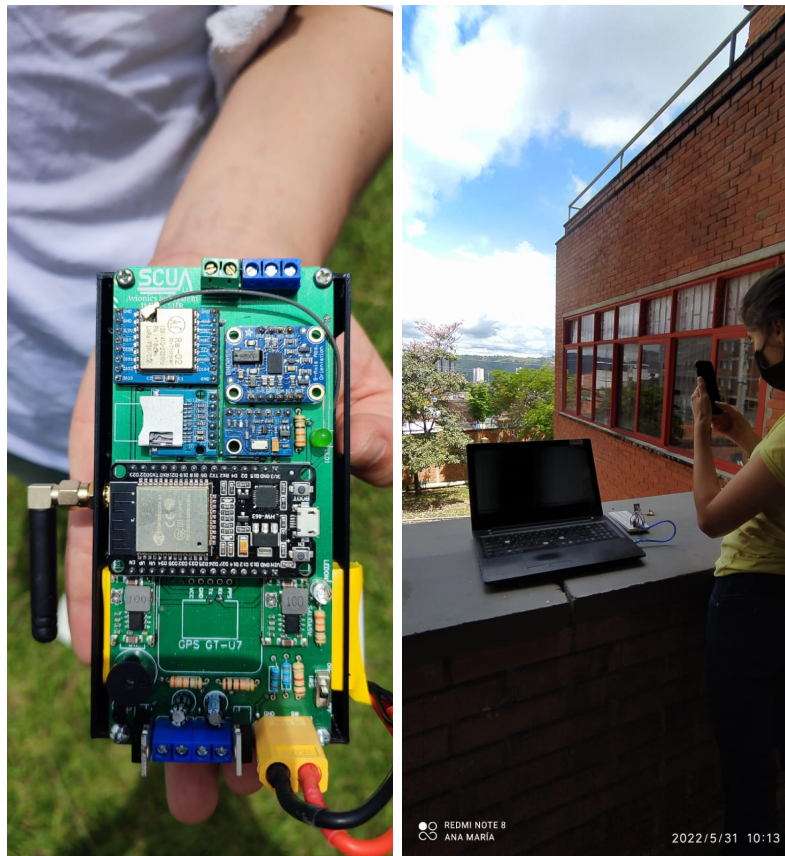


Figura 28: *línea de vista entre el edificio de pesados y aeropuerto palonegro con aplicativo web Ubiquiti*



Por otra parte, en la figura 29 se exhibe la evidencia de las pruebas realizadas para los radioenlaces en las distancias mencionadas.

Figura 29: *Evidencia efectuando pruebas de radioenlace con aviónica a bordo y sistema de recepción*



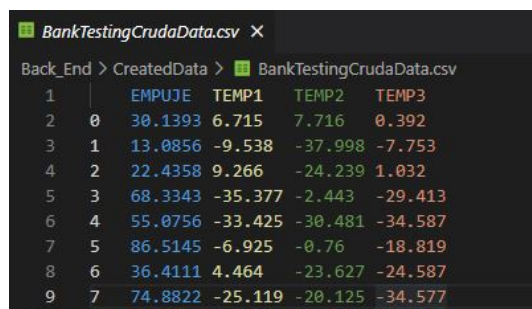
6.2 Implementación etapa procesamiento de datos

La implementación de la etapa de procesamiento de datos se basa en el diseño y estructura de código para entregar datos correctamente a la etapa de visualización, teniendo

presente la cantidad de datos en cada bus, su ubicación y relación de estos.

Con el objetivo principal de poseer claridad en los datos, se inicia con la estructura de la clasificación de los datos sin procesar en un archivo .csv como se muestra en la figura 30, para luego usarlo en un archivo SQL, indicándose en la figura 31, donde se declaran los elementos que se tendrán en la datos recibidos, para el caso de la imagen del ejemplo se declaro la siguiente trama: 'empuje', 'TEMP1', 'TEMP2', 'TEMP3'.

Figura 30: *Datos sin procesar en un archivo .csv*



Back_End	EMPUJE	TEMP1	TEMP2	TEMP3
0	30.1393	6.715	7.716	0.392
1	13.0856	-9.538	-37.998	-7.753
2	22.4358	9.266	-24.239	1.032
3	68.3343	-35.377	-2.443	-29.413
4	55.0756	-33.425	-30.481	-34.587
5	86.5145	-6.925	-0.76	-18.819
6	36.4111	4.464	-23.627	-24.587
7	74.8822	-25.119	-20.125	-34.577

Figura 31: *Estructura de tabla para banktestingcrudadata*

```
-- Estructura de tabla para la tabla `banktestingcrudadata`
--
CREATE TABLE `banktestingcrudadata` (
  `empuje` varchar(250) COLLATE latin1_spanish_ci NOT NULL,
  `TEMP1` varchar(250) COLLATE latin1_spanish_ci NOT NULL,
  `TEMP2` varchar(250) COLLATE latin1_spanish_ci NOT NULL,
  `TEMP3` varchar(250) COLLATE latin1_spanish_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;
```

El archivo fuente para la conexión entre la base de datos y servidor, se desarrolla en lenguaje PHP y su estructura se presenta en la figura 32, se recopilan los valores clasificándolos en cada uno de los elementos asignados, en una estructura tipo SQL usada en el tratamiento de datos Back-end.

Figura 32: *Estructura conexión entre base de datos y servidor*

```
<?php
function conectar(){
    $host = "localhost";
    $user = "root";
    $password = "";
    $db = "adminlte3";

    $conectar = mysqli_connect($host,$user,$password,$db);
    return $conectar;
}
```

6.3 Implementación etapa de visualización

En la etapa de visualización se implementa un código donde se asignan características relacionadas con los tamaños de tablas, colores, imágenes, videos y muchos más elementos estéticos, parte de este proceso se ejemplifica en la figura 33, en donde se estructura el código para dichas características. Para ello, se hace uso de lenguajes PHP y JavaScript. Toda la programación de esta etapa se ejecuta en Visual Studio Code, el cual engloba los diferentes lenguajes empleados.

Figura 33: Estructura código para las características estéticas del diseño

```

LauraPinto96, 6 months ago | 1 author (LauraPinto96)
.chart{
  height: 250px;
}

julian, 6 months ago | 1 author (julian)
.button {
  background-color: #F25430; /* Green */
  border: none;
  color: white;
  padding: 7px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 0px 0px;
  transition-duration: 0.4s;
  cursor: pointer;
}

```

Asimismo, en la figura 34 se especifica la estructura del código que comprende la pantalla principal de la interfaz gráfica, en donde se encontrará un menú desplegable para cada una de las misiones y de que se trata en sí el concurso de cohetaría LASC.

Figura 34: Estructura código para la pantalla principal, presentación LASC

```

<div class="content-wrapper" id="contenido_principal" style="text-align: center;">
  <!-- Main content -->
  <span class="card-title-center" style="color: #D8D9D7; font-family: 'Mochiy Pop One', sans-serif; font-size:
    <strong><b>LATIN AMERICAN SPACE CHALLENGE</b></strong>
  </span>
  <div>
    
  </div><br>
  <div>
    <body text= 'black'>
      <b><big>The Latin American Space Challenge (LASC) has a mission to <br/>
      motivate people to develop and launch a rocket with a satellite <br/>
      as a payload. There are two challenges with different categories: <br/>
      the Rocket Challenge and the Satellite Challenge. In 2022, LASC <br/>
      will be onsite at the Cape Canavial!</big><br/>
    </body>
  </div><br><br>
  <div>
    
  </div>

```

En la figura 35, al igual que en la figura 34 se precisa la estructura de código, solo que en el caso es para la vista de las pantallas de cada una de las misiones, en las que se encuentran las gráficas, tablas y demás elementos interactivos de cada una de las variables definidas en las misiones.

Figura 35: *Estructura código para las características estéticas, vista de los usuarios*

```

<style>
  LauraPinto96, 6 months ago | 1 author (LauraPinto96)
  .content-wrapper {
    background-image: url('FONDO.png');
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: cover;
  }

  LauraPinto96, 6 months ago | 1 author (LauraPinto96)
  .card-body{
    padding: 1px;
  }
</style>

<!-- Main content -->
<section class="content" style="padding: 0px">
  <div class="body">
    <div class="row">
      <div class="col-md-12">
        <div class="card card-danger">
          <div class="card-body" style="background-color: #D8D9D7">
            <h3 class="card-title" style="color: #F25430"><b>RAW DATA</b></h3>
          </div>
          <div class="card-body">
            <div class="chart">

```

Por último, en la figura 36 se indica el código que realiza la configuración estética para gráficos tipo chart en la interfaz.

Figura 36: *Estructura código para las características estéticas para gráficas tipo Chart*

```

//-----
//-TEMPERATURA -
//-----
var areaChartCanvas = $('#areaChart').get(0).getContext('2d');

var areaChartDataset = {
  label: 'T (°C)',
  backgroundColor: '#122126',
  borderColor: '#122126',
  fill: false,
  data: []
};

datos.forEach((el, idx) => {
  areaChartDataset.data.push(parseInt(el.temp));
});

var areaChartOptions = {
  type: 'line',
  data: {

```

6.4 Implementación etapa de almacenamiento

Para la implementación de esta etapa se realiza mediante Bootstrap o biblioteca multiplataforma, este es un repositorio donde se encuentran varios ejemplos de código para visualización de data y otras más funciones del entorno de trabajo o framework. En esta etapa se tiene como resultado junto con las etapas anteriores el objetivo de obtener la visualización de las lecturas tomadas en las misiones Orión y GuaneSat. Por tanto, en la figura 37 se expone el código que ordena los datos en filas y columnas para su posterior almacenamiento, en el encabezado de esta tabla generada se encuentra el nombre de la variable y bajo esta cada una de las lecturas, este proceso engloba los datos para cada una de las misiones, por lo que se genera otro archivo similar para la otra misión en cuestión.

Figura 37: Estructura código para las tablas de presentación de historiales.

```

section class="content" style="padding: 0px">
<div class="body">
  <div class="row">
    <div class="col-md-12">
      <div class="card card-danger">
        <div class="card-body" style="background-color: #D8D9D7">
          <h3 class="card-title" style="color: #F25430"><b>RAW DATA</b></h3>
        </div>
        <div class="card-body">
          <div class="chart">
            <!-- /.card-header -->
            <div class="card-body">
              <div id="example1_wrapper" class="dataTables_wrapper dt-bootstrap4">
                <div class="row">
                  <div class="col-sm-12" style="padding: 0px">
                    <table id="example1" class="table table-bordered table-striped dataTable dtr-inline" aria-describedby
                    <thead>
                      <tr>
                        <th class="sorting sorting_asc" tabindex="0" aria-controls="example1" rowspan="1" colspan="1" a
                        <th class="sorting" tabindex="0" aria-controls="example1" rowspan="1" colspan="1" aria-label="A
                        <th class="sorting" tabindex="0" aria-controls="example1" rowspan="1" colspan="1" aria-label="A
                        <th class="sorting" tabindex="0" aria-controls="example1" rowspan="1" colspan="1" aria-label="A
                        <th class="sorting" tabindex="0" aria-controls="example1" rowspan="1" colspan="1" aria-label="A
                        <th class="sorting" tabindex="0" aria-controls="example1" rowspan="1" colspan="1" aria-label="T
                        <th class="sorting" tabindex="0" aria-controls="example1" rowspan="1" colspan="1" aria-label="A

```

6.5 Implementación de la validación de variables sensadas

A fin de implementar esta etapa, en las tablas 14 a 19 se entregan las guías de observación para cada uno de los sensores que fueron usados en la misión GuaneSat y Orión. En estas tablas se especifica a que misión va dirigido el instrumento, que variable mide, rango en el que opera dicho instrumento y cada uno de los criterios establecidos para verificar su funcionalidad, entre ellos, si es apto para operar en diferentes ambientes, la precisión y resolución del instrumento, etc.

Tabla 14: *MS5611-01BA03*

Guia de observación para el instrumento MS5611-01BA03			
Objetivo de la guía	Presión		
Objetivo de la investigación	Misión Orión		
Rango del instrumento	10 - 1200 mbar		
Criterios a evaluar			
Criterio	Si	No	Observación
Claridad de la medida	X		
Coherencia interna con el sistema	X		
Respuesta puede estar orientada a la deseabilidad social	X		
Contribuye a los objetivos de la investigación	X		
Contribuye a medir el constructo en estudio	X		
Cantidad de unidades es clara para el objetivo	X		
Coherente para uso en Bucaramanga	X		1016 mbar
Coherente para uso en Tatuí	X		1021mbar

Tabla 15: *BNO055*

Guia de observación para el instrumento BNO055			
Objetivo de la guía	Acelerómetro [g] Giroscopio [°/s] Magnetómetro [μ T]		
Objetivo de la investigación	Misión Orión		
Rangos del instrumento	$\pm 2 - \pm 16$ 125 - 2000 $\pm 1200 - \pm 2500$		
Criterios a evaluar			
Criterio	Si	No	Observación
Claridad de la medida	X		
Coherencia interna con el sistema	X		Se encuentra en módulo y en SMD
Respuesta puede estar orientada a la deseabilidad social	X		
Contribuye a los objetivos de la investigación	X		
Contribuye a medir el constructo en estudio	X		
Cantidad de unidades es clara para el objetivo	X		
Coherente para uso en Bucaramanga	X		
Coherente para uso en Tatuí	X		

Tabla 16: *GY-63 MS5611*

Guia de observación para el instrumento GY-63 MS5611			
Objetivo de la guía	Temperatura y presión atmosférica		
Objetivo de la investigación	Versión inicial, pruebas Misión GuaneSat		
Rango del instrumento	10 - 1200 mbar		
Criterios a evaluar			
Criterio	Si	No	Observación
Claridad de la medida	X		
Coherencia interna con el sistema	X		Se encuentra en módulo y en SMD
Respuesta puede estar orientada a la deseabilidad social	X		
Contribuye a los objetivos de la investigación	X		
Contribuye a medir el constructo en estudio	X		
Cantidad de unidades es clara para el objetivo	X		
Coherente para uso en Bucaramanga	X		
Coherente para uso en Tatuí	X		

Tabla 17: *MPU6050*

Guia de observación para el instrumento MPU6050			
Objetivo de la guía	Medidas inerciales		
Objetivo de la investigación	Versión alternativa, pruebas Misión GuaneSat		
Rango del instrumento	0°-360°		
Criterios a evaluar			
Criterio	Si	No	Observación
Claridad de la medida	X		
Coherencia interna con el sistema	X		Se encuentra en módulo y en SMD
Respuesta puede estar orientada a la deseabilidad social	X		
Contribuye a los objetivos de la investigación	X		
Contribuye a medir el constructo en estudio	X		
Cantidad de unidades es clara para el objetivo	X		
Coherente para uso en Bucaramanga	X		
Coherente para uso en Tatuí	X		

Tabla 18: *BMP280*

Guía de observación para el instrumento BMP280			
Objetivo de la guía	Mediciones de temperatura y presión atmosférica		
Objetivo de la investigación	Versión alternativa, pruebas Misión GuaneSat		
Rango del instrumento	10 - 1200 mbar		
Criterios a evaluar			
Criterio	Si	No	Observación
Claridad de la medida	X		
Coherencia interna con el sistema	X		Se encuentra en módulo y en SMD
Respuesta puede estar orientada a la deseabilidad social	X		
Contribuye a los objetivos de la investigación	X		
Contribuye a medir el constructo en estudio	X		
Cantidad de unidades es clara para el objetivo	X		
Coherente para uso en Bucaramanga	X		
Coherente para uso en Tatuí	X		

Tabla 19: *Termocupla tipo K MAX6675*

Guía de observación para el instrumento Termocupla tipo K MAX6675			
Objetivo de la guía	Registrar altas temperaturas		
Objetivo de la investigación	Banco de pruebas		
Rango del instrumento	-200°C a 1200°C		
Criterios a evaluar			
Criterio	Si	No	Observación
Claridad de la medida	X		
Coherencia interna con el sistema	X		
Respuesta puede estar orientada a la deseabilidad social	X		
Contribuye a los objetivos de la investigación	X		
Contribuye a medir el constructo en estudio	X		
Cantidad de unidades es clara para el objetivo	X		
Coherente para uso en Bucaramanga	X		
Coherente para uso en Tatuí	X		

7 Resultados

7.1 Resultados etapa para la validación del sistema de comunicación

Se presenta como resultado que el uso de antenas para módulo LoRa Ra32 tanto en receptor como transmisor; durante las pruebas se usó antena lora en transmisor y antena tipo Antena omnidireccional VHF UHF 3G 4G 65 a 3500MHz (sombriilla) la cual no fue elegida por la practicidad en tierra, con estas antenas se genera una buena compatibilidad y óptima recepción de los mensajes, al igual que teniendo antena de módulo lora tanto en transmisor como en receptor. Por lo que, se determina el uso de antena módulo LoRa Ra-02. Como evidencia de las pruebas realizadas se muestra en la figura 38 la antena omnidireccional (tipo sombrilla), junto con el módulo LoRa.

Figura 38: *Prueba con antena omnidireccional y antena modulo LoRa Ra-02*



Por otra parte, como resultado de estas pruebas, en la figura 39 en un archivo .txt se visualiza la trama de datos recibida mediante esta antena; se aclara que esta no es la trama final definida para las misiones, solo para la práctica de prueba y error.

7.2 Resultados etapa de procesamiento de datos

Se presenta como resultado de la etapa de procesamiento en phpMyAdmin, donde se cuenta con la base de datos en una aplicación web que sirve para administrar bases de datos MySQL de forma sencilla y con una interfaz amistosa.

Figura 39: *Trama recibida durante pruebas con antena omnidireccional y antena modulo LoRa Ra32*

```

TRYFINI.TXT: Bloc de notas
Archivo Edición Formato Ver Ayuda
{ Acc x: 0.00 , Acc y: 0.00 , Acc z: 0.00 , Apogeo: 0.00 , Apogeo2: 0.00 , Tiempo seg: , 0 , ap
{ Acc x: 0.00 , Acc y: 0.00 , Acc z: 0.00 , Apogeo: 0.00 , Apogeo2: 0.00 , Tiempo seg: , 0 , ap
{ Acc x: 0.00 , Acc y: 0.00 , Acc z: 0.00 , Apogeo: 0.00 , Apogeo2: 0.00 , Tiempo seg: , 0 , ap
{ Acc x: 0.00 , Acc y: 0.00 , Acc z: 0.00 , Apogeo: 0.00 , Apogeo2: 0.00 , Tiempo seg: , 0 , ap
{ Acc x: 0.00 , Acc y: 0.00 , Acc z: 0.00 , Apogeo: 0.00 , Apogeo2: 0.00 , Tiempo seg: , 0 , ap
{ Acc x: 0.00 , Acc y: 0.00 , Acc z: 0.00 , Apogeo: 0.00 , Apogeo2: 0.00 , Tiempo seg: , 0 , ap
{ Acc x: -0.25 , Acc y: -0.43 , Acc z: 9.53 , Apogeo: 9.54 , Apogeo2: 0.00 , Tiempo seg: , 4 ,
{ Acc x: -0.27 , Acc y: -0.45 , Acc z: 9.51 , Apogeo: 9.52 , Apogeo2: 0.00 , Tiempo seg: , 4 ,
{ Acc x: -0.25 , Acc y: -0.43 , Acc z: 9.51 , Apogeo: 9.52 , Apogeo2: 0.00 , Tiempo seg: , 4 ,
{ Acc x: -0.28 , Acc y: -0.45 , Acc z: 9.53 , Apogeo: 9.54 , Apogeo2: 0.00 , Tiempo seg: , 4 ,
{ Acc x: -0.26 , Acc y: -0.44 , Acc z: 9.50 , Apogeo: 9.51 , Apogeo2: 0.00 , Tiempo seg: , 4 ,
{ Acc x: -0.26 , Acc y: -0.47 , Acc z: 9.51 , Apogeo: 9.53 , Apogeo2: 0.08 , Tiempo seg: , 4 ,
{ Acc x: -0.29 , Acc y: -0.46 , Acc z: 9.52 , Apogeo: 9.54 , Apogeo2: 0.08 , Tiempo seg: , 4 ,
{ Acc x: -0.27 , Acc y: -0.45 , Acc z: 9.50 , Apogeo: 9.51 , Apogeo2: 0.08 , Tiempo seg: , 4 ,
{ Acc x: -0.25 , Acc y: -0.43 , Acc z: 9.53 , Apogeo: 9.54 , Apogeo2: -0.07 , Tiempo seg: , 6 ,
{ Acc x: -0.27 , Acc y: -0.46 , Acc z: 9.50 , Apogeo: 9.51 , Apogeo2: -0.07 , Tiempo seg: , 6 ,
{ Acc x: -0.27 , Acc y: -0.45 , Acc z: 9.50 , Apogeo: 9.51 , Apogeo2: -0.07 , Tiempo seg: , 6 ,
{ Acc x: -0.21 , Acc y: -0.43 , Acc z: 9.54 , Apogeo: 9.55 , Apogeo2: 0.00 , Tiempo seg: , 6 ,
{ Acc x: -0.23 , Acc y: -0.42 , Acc z: 9.53 , Apogeo: 9.54 , Apogeo2: 0.00 , Tiempo seg: , 6 ,
{ Acc x: -0.25 , Acc y: -0.42 , Acc z: 9.49 , Apogeo: 9.50 , Apogeo2: 0.00 , Tiempo seg: , 6 ,
{ Acc x: -0.22 , Acc y: -0.44 , Acc z: 9.45 , Apogeo: 9.46 , Apogeo2: 0.00 , Tiempo seg: , 6 ,
{ Acc x: -0.26 , Acc y: -0.44 , Acc z: 9.51 , Apogeo: 9.52 , Apogeo2: 0.00 , Tiempo seg: , 6 ,
{ Acc x: -0.25 , Acc y: -0.47 , Acc z: 9.50 , Apogeo: 9.51 , Apogeo2: 0.00 , Tiempo seg: , 7 ,
{ Acc x: -0.25 , Acc y: -0.45 , Acc z: 9.50 , Apogeo: 9.51 , Apogeo2: 0.24 , Tiempo seg: , 7 ,
{ Acc x: -0.26 , Acc y: -0.45 , Acc z: 9.51 , Apogeo: 9.52 , Apogeo2: 0.24 , Tiempo seg: , 7

```

En la figura 40, se indica el entorno de phpMyAdmin, si bien es de resaltar que: para cada una de las misiones se genera una base de datos, y que a su vez se encierran dentro de una sola denominada adminte3.

Figura 40: *Resultados de la Etapa de procesamiento de datos*

empuje	TEMP1	TEMP2	TEMP3
301.393	6.715	7.716	0.392
130.856	-9.538	-37.998	-7.753
224.358	9.266	-24.239	1.032
683.343	-35.377	-2.443	-29.413
550.756	-33.425	-30.481	-34.587
865.145	-6.925	-0.76	-18.819
364.111	4.464	-23.627	-24.587
740.892	25.416	20.425	24.577

7.3 Resultados etapa de Visualización

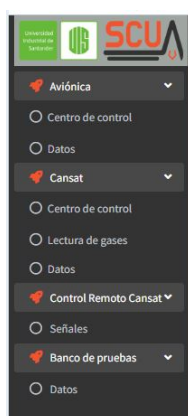
Como resultado de esta etapa se presenta cada una de las pantallas de visualización desarrolladas durante el desarrollo de este proyecto.

En la figura 41 se muestra la pantalla principal, comprende una breve introducción relacionada con el concurso denominado LASC y en la izquierda un menú que remite a cada una de las misiones, al seleccionar una de estas opciones y de acuerdo con la figura 42 para cada misión se cuenta con mas pantallas, en las que se pueden ver el centro de control y los datos organizados en una tabla. Además, se destaca que se cuenta con una opción mas, la cual se propone como trabajo futuro para su profundización, comprendiendo el control remoto del CanSat.

Figura 41: *Pantalla principal*



Figura 42: *Menú desplegable*



Además, en las figuras 43 y 44 se presentan las pantallas finales para cada una de las misiones, conteniendo cada una de las variables mencionadas a lo largo de este texto.

Figura 43: Pantalla para visualización datos de vuelo en misión Orión

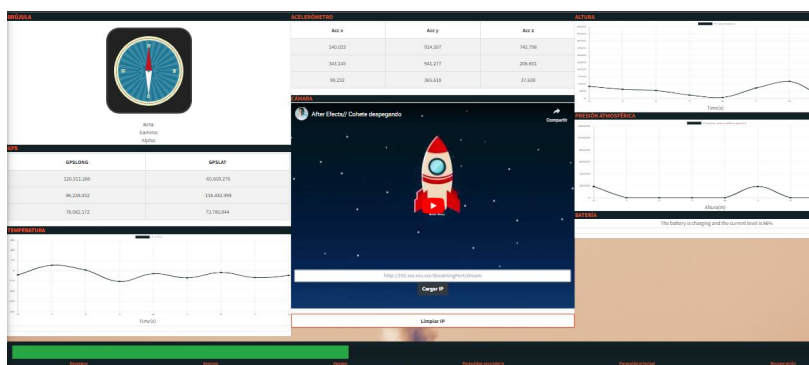
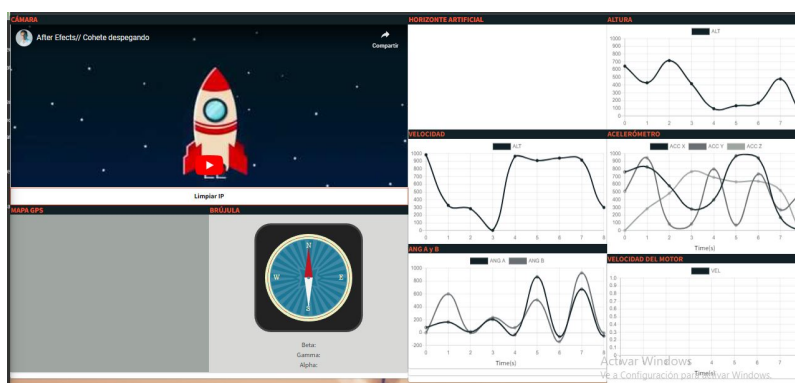
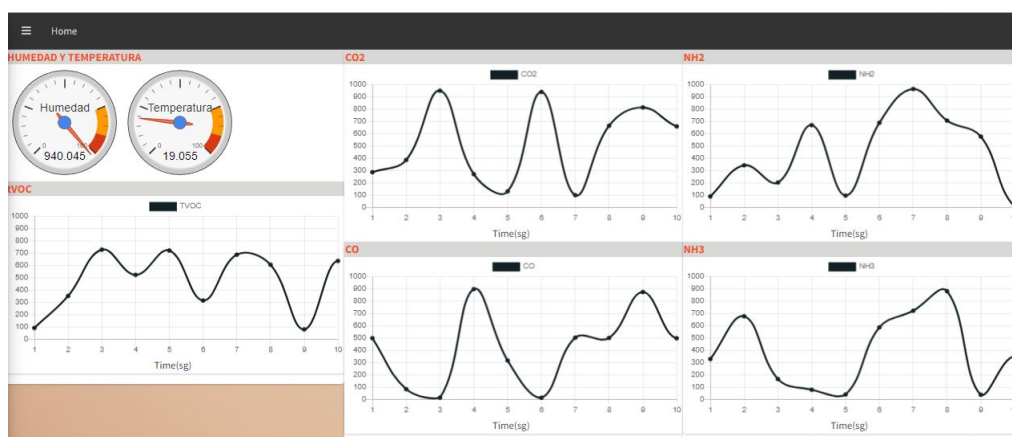


Figura 44: Pantalla para visualización datos de vuelo en misión GuaneSat



Debido a la cantidad de variables con las que cuenta la misión Orión, se opta por generar una pantalla mas, en la figura 45 se separan los gases de las demás mediciones para no generar confusión con las lecturas que son de interés para el control del CanSat.

Figura 45: Pantalla para visualización datos de vuelo en misión GuaneSat



7.4 Resultados etapa de almacenamiento

Como resultado de la etapa de almacenamiento se entregan las tablas con el historial de cada una de las misiones.

En las figuras 46 y 47, se exhiben las tablas generadas durante la etapa de almacenamiento, ordenadas de acuerdo a la trama de datos recibida para cada una de las misiones, en la figura 46 la pantalla de historial para la misión Orión, siendo de esta forma la visualización dentro de la interfaz y de igual manera en la figura 47 para GuaneSat, además que en phpMyAdmin también se genera una tabla con esta estructura. Por otra parte, en la figura 48 se refleja la pantalla para el banco de pruebas del motor, que al igual que el control del CanSat, cuenta con un espacio para visualizar estas medidas dentro de la interfaz.

Figura 46: *Pantalla de historial de misión Orión*

RAW DATA															
Acc x	Acc y	Acc z	Apege1	Apege2	Tiempo	Altura	Presión	GPS lat	GPS long	Power	Fasec	Temp	Giro x	Giro y	Giro z
340.033	954.307	745.799	394.424	436.698	1354	3478.64	103326.0	65.609.276	120.511.206	1	2	14.636	53.7	309.26	124.42
343.145	943.277	288.601	93.278	234.753	408	2611.94	101.192.678	156.482.998	86.238.452	1	3	18.283	250.49	353.52	47.23
98.232	365.438	37.638	207.138	958.936	1022	2264.96	103.098.256	73.780.844	76.082.172	1	0	2.533	70.83	78.22	47.4
145.822	864.019	145.908	92.044	137.375	1622	956.33	100.926.034	134.838.558	99.841.869	1	1	36.035	122.25	136.08	233.95
624.538	940.149	352.436	122.775	883.174	231	297.99	100.792.732	45.428.524	3.533.208	1	5	10.889	334.24	98.52	126.08
922.748	471.748	772.039	439.731	11.591	961	2389.78	100659.39	60.041.784	16.997.792	0	2	24.659	104.36	82.38	133.76
28.734	293.434	755.509	450.862	194.421	3827	4804.39	100.526.068	122.346.068	167.522.777	1	3	6.832	146.7	287.32	6.79
93.503	122.206	922.834	11.681	980.147	854	816.34	100.392.746	16.402.867	23.809.408	1	4	23.423	108.01	270.45	163.42
854.137	102.847	64.721	162.934	502.545	246	1983.63	100.259.424	14.335.093	41.826.837	1	1	36.947	4.21	233.11	49.27
679.784	930.966	62.096	954.962	36.671	710	156.15	100.128.102	57.123.962	168.240.560	0	3	2.227	8.96	38.43	6.78
338.096	546.889	868.127	203.637	568.319	641	2015.39	99992.78	17.261.303	30.985.158	1	3	17.104	308.05	286.66	227.83
769.142	755.363	426.631	222.759	89.268	1	1832.17	99.858.458	98.812.194	95.519.459	0	0	15.989	183.71	237.62	223.61
229.963	54.538	856.284	0.9795	70.195	1796	1405.62	99.726.136	63.482.548	93.147.968	0	5	7.984	122.34	311.57	133.35
935.833	861.531	329.637	88.362	113.939	389	2298.49	99.192.834	1.530.417	128.757.905	1	5	26.456	357.57	223.25	156.01
179.703	344.818	931.790	907.852	381.726	401	3246.93	99.459.492	30.260.734	44.113.812	0	6	36.831	75.63	270.84	46.05

Figura 47: *Pantalla de historial de misión GuaneSat*

RAW DATA												
GPS lat	GPS long	GPS alt	GPS spe	Hum	Temp	Pres	CO	CO2	NH2	NH3	tVOC	
596.777	288.569	644.828	984.679	203.887	-60.287	101326.0	500.656	288.442	91.724	330.959	93.335	
606.345	492.707	430.179	328.985	666.342	4.874	101.192.678	85.918	388.011	344.955	678.002	353.356	
927.509	160.803	716.626	284.293	828.195	-67.323	101.059.356	16.354	951.444	204.011	167.362	730.197	
26.245	551.191	417.507	3.633	461.193	-62.913	100.926.034	899.665	272.019	671.829	80.652	525.078	
645.786	135.018	95.134	964.283	546.143	49.185	100.792.712	318.188	132.779	98.535	41.526	723.622	
433.853	451.476	133.732	909.163	901.195	72.502	100659.39	15.714	941.162	690.262	588.187	316.274	

Figura 48: *Pantalla de historial de registros en el banco de prueba del motor*

RAW DATA			
EMPUJE	TEMP 1	TEMP 2	TEMP 3
301.393	6.715	7.716	0.392
130.856	-9.538	-37.998	-7.753
224.358	9.266	-24.239	1.032
683.343	-35.377	-2.443	-29.413
550.756	-33.425	-30.481	-34.587
865.145	-6.925	-0.76	-18.819
364.111	4.464	-23.627	-24.587
748.822	-25.119	-20.125	-34.577
91.852	29.566	14.001	14.887
17.888	-33.699	-5.811	-32.97
909.904	11.501	-13.431	-25.949

7.5 Resultados etapa de variables sensadas

En la tabla 20, se presenta el consolidado de sensores que cumplen con los requisitos de funcionalidad en las condiciones ambientales presentes tanto en Tatuí como en Bucaramanga para la temporada del año que comprende los meses de Junio, Julio y Agosto.

Tabla 20: *Consolidado de sensores aprobados para LASC2022*

Instrumento y/o sensor	Aprobado		Misión
	Si	No	
MS5611-01BA03	X		Orión
BNO055	X		Orión
BNO055 módulo	X		GuaneSat
BNO055 SMD	X		GuaneSat
GY-63 MS5611 módulo	X		GuaneSat
GY-63 MS5611 SMD	X		GuaneSat
MPU6050 módulo	X		GuaneSat
MPU6050 SMD	X		GuaneSat
BMP280 módulo	X		GuaneSat
BMP280 SMD	X		GuaneSat
MAX6675	X		Banco de pruebas

8 Conclusiones

Vale la pena aclarar que la importancia de tener un radioenlace efectivo define la adecuada recepción de información, durante la transmisión en la distancia entre la UIS y el Aeropuerto Palonegro se presentaron algunos obstáculos como árboles y la posición de las antenas, generando errores en el mensaje recibido como letras y símbolos que no estaban descritos en la trama enviada. Otro factor importante es saber que tipo de antenas están disponibles para el uso en las frecuencias que trabaja el radiotransmisor LoRa Ra-02 que es de 430 MHz, para esto llegamos a la conclusión que las antenas más prácticas y con cobertura necesaria para las misiones son las antenas dipolo, debido a su patrón de radiación, ganancia, directividad y dirección. La pérdida de trayectoria en espacio libre es de 94.652 dB para el objetivo de los 3 km. La frecuencia de 430 MHz según la ANE, está en una banda abierta para aficionados en Colombia y en Brasil.

Visual Studio Code fue una herramienta que de manera integrada y fácil comprensión, ofreció un entorno en el cual se pudo programar y sintetizar los scripts desarrollados en PHP, Python, Javascript y en el servidor HTML, siendo de gran importancia durante el diseño de la interfaz gráfica y la codificación para el procesamiento y visualización de datos de las variables sensadas, estado del cohete y su carga útil. Sin embargo, utilizar este software requiere de la previa instalación de extensiones, las cuales permiten que los diferentes lenguajes de programación no generen conflictos entre sí y se ejecuten en completa armonía. Por otra parte, este se integra muy bien con el control de código fuente, es decir, con Git. Haciendo que el flujo de trabajo sea más eficiente y sólido.

Al tener conocimiento del comportamiento registrado por empresas como IDEAM y Weather Company, sabemos a qué condiciones deben funcionar los sensores, por esta razón en el trabajo se presenta un estudio detallado de estas, por motivos de que a la fecha no se puede presentar un vuelo del cohete y del cansat en estas altitudes de 500 metros para la misión GuaneSat y 3 kilómetros para la misión Orión a fin de realizar estos vuelos en Colombia se debe contar con permisos de instituciones como la Aereocivil e INDUMIL, se deja este ítem abierto para hacer la comparación con las mediciones descritas en el cuerpo del trabajo.

A lo largo de este estudio, se expuso de manera objetiva las bases y elementos que en conjunto originan la interfaz gráfica en la cual se visualizan las distintas variables de las misiones Orión y GuaneSat, siendo este un reto a la hora de unificar los diferentes lenguajes de programación empleados, puesto que se generaron uno que otro conflictos

debido a que la conexión entre puertos, la capacidad de recepción de la antena, hasta el sistema operativo de la computadora fueron motivo de inconvenientes. Lo que supuso grandes retos, pero no imposibles de encontrar soluciones.

9 Recomendaciones

Considerando que el alcance de este proyecto de investigación está direccionado a visualizar data captada durante el vuelo del cohete Orión y el cansat GuaneSat en una interfaz gráfica de forma local, se recomienda que esta plataforma se pueda ejecutar en la nube, lo que la haría mas eficiente ya que cualquier usuario contaría con acceso a ella sin estar presente en el lugar que se este desarrollando la misión y acceder a esta data en tiempo real. Además su diseño es flexible a cualquier tipo de cambios en su codificación para tal fin, de igual forma es recomendable ejecutar la interfaz como sigue:

1. Ejecutar el programa XAMPP Control Panel e iniciar los modulos Apache y MySQL, con el propósito de conectar a la base de datos.
2. Ejecutar el programa Visual Studio Code.
3. Siguiendo el enlace presentado en el apéndice C, se procede a seleccionar la rama nombrada Back_End, en la cual se encuentra el repositorio de este proyecto.
4. Para iniciar la interfaz de forma local, se procede a descargar el repositorio mencionado.
5. Desde el programa VSC se despliega los diferentes scripts que hacen parte de la interfaz.
6. Para visualizarla desde el navegador de su preferencia, se procede a ejecutar el archivo index.php, el cual hace el llamado a todos los scripts enlazados a este. Cabe aclarar que previo a ejecutar dicho archivo se deben instalar las diferentes extensiones para la correcta ejecución, entre ellas las relacionadas con php, css, html, javascript.

10 Trabajo Futuro

El trabajo implementado hasta el momento en este proyecto de investigación concierne a un buen avance relacionado en el campo aeroespacial dado que, es la primera vez que se desarrolla un proyecto en Colombia para un concurso internacional que trata esta temática. Para próximos trabajos en la línea expuesta se propone ampliar la implementación de esta plataforma con acceso en la nube para servidores públicos, ya que hasta el momento es desarrollado para servidores locales. El modo de ejecución de todos los códigos de implementación están desarrollados de una forma modular de tal manera que estos pueden ser ampliados hacia la finalidad que deseen los usuarios.

El repositorio que contiene los diferentes scripts construidos, se encuentran en el desarrollador de software GitHub, el cual aloja proyectos utilizando el sistema de control de versiones Git, ver Apéndice C. Cabe aclarar que en este se encuentran dos ramas: main y Back_End, siendo la segunda con el código definitivo y que a su vez esta abierto para ser optimizado y mejorado.

Referencias Bibliográficas

- [Alex, 2019] Alex, A. S. (2019). ¿qué es php: Características y usos? In <https://openwebinars.net/blog/que-es-php/>. OpenWebinars.
- [Code, 2022] Code, V. S. (2022). Docs. In <https://code.visualstudio.com/docs/nodejs-articles>. Microsoft.
- [Constaín, 2012] Constaín, Alfredo José; Alzate, E. B. (2012). Metodología básica de instrumentación industrial y electrónica. In *Metodología básica de instrumentación industrial y electrónica*. Universidad de la Salle.
- [Diego, 2022] Diego, S. (2022). Introducción al css: qué es, para qué sirve y otras 10 preguntas frecuentes. In <https://blog.hubspot.es/website/que-es-css>. HubSpot.
- [Frankier, 2022] Frankier, F. (2022). ¿qué es visual studio code y qué ventajas ofrece? In <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>. OpenWebinars.
- [Gallego Carrillo, 2005] Gallego Carrillo, M. Montalvo Herranz, M. (2005). Interfaz gráfica. In *Interfaces gráficas en Java*, page 15. Editorial Universitaria Ramon Areces.
- [IDEAM, 2022] IDEAM (2022). Pronósticos y alertas, metereología aeronautica, cambio climático. In <http://www.ideam.gov.co/>. Ministerio de ambiente y desarrollo sostenible.
- [Joyanes Aguilar, 2011] Joyanes Aguilar, L., Z. M. I. (2011). Lenguaje de computadora. In *Programación en Java 6: algoritmos, programación orientada a objetos e interfaz gráfica de usuario.*, pages 1–11–12–13. McGraw-Hill.
- [J.Vivas, 2022] J.Vivas, S. (2022). Misión orión. In *Diseño, implementación y validación de la aviónica para el cohete sonda de combustible sólido ORION UIS partícipe en el Latin American Space Challenge 2022*. UIS.
- [LASC, 2022] LASC (2022). Latin american space challenge. In <https://www.lasc.space/>. LASC.
- [Martínez Ramírez, 2019] Martínez Ramírez, J. L. (2019). The process of preparing and validating a documentary research measurement instrument. In <http://portal.amelica.org/ameli/jatsRepo/226/226955004/index.html>. Acción y Reflexión Educativa.
- [MDN, 2022] MDN, w. d. (2022). Css básico. In https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/CSS_basics. MDN.

- [Olivier, 2016] Olivier, H. (2016). Php 7: Desarrollar un sitio web dinámico e interactivo. In <https://books.google.es/books?id=EJ1t1GsMHbIC>, Recursos Informáticos, page 15. ENI.
- [Ramos Cardozzo, 2016] Ramos Cardozzo, D. (2016). Introducción a la ingeniería de software. In *Desarrollo de Software: Requisitos, Estimaciones y Análisis.*, pages 1–4. IT Campus Academy.
- [Raúl, 2011] Raúl, G. D. (2011). ¿qué es python? In *PYTHON para todos.*, pages 7–8. UCI.
- [Ricardo, 2018] Ricardo, R. M. (2018). ¿qué es un requerimiento funcional? In <http://www.pmoinformatica.com/2018/05/que-es-requerimiento-funcional.html>. PMOinformatica.coml.
- [Rubén, 2021] Rubén, V. (2021). Visual studio code: el editor de código de microsoft que querrás instalar. In <https://www.softzone.es/programas/utilidades/visual-studio-code/>. Soft Zone.
- [SCUA, 2022a] SCUA (2022a). Autonomus uav cansat for meteorological studies. In *Autonomus UAV Cansat for Meteorological Studies. Team 110 Project Technical Report for the 2022 Latin American Space Challenge*, pages 1–23. CEMOS.
- [SCUA, 2022b] SCUA (2022b). Orión rocket technical report. In *Rocket Vehicle for a 4Kg Payload Team 39 Project Technical Report for the 2022 Latin American Space Challenge*, pages 1–28. CEMOS.
- [SCUA, 2022c] SCUA (2022c). Semillero de cohetería uis aeroespacial. In <https://www.scua.space/>. SCUA.
- [the weather Channel, 2022] the weather Channel (2022). Calidad del aire hoy. In <https://weather.com/es-UY/forecast/air-quality/l/d7384fb1d7f39ff5c6a8fe6019195d49c8c54e1ef1f7714583f825e3b49c97ac>. The weather company.
- [University, 2020] University, o. T. (2020). What is a full stack developer what do they do? In <https://bootcamp.learn.utoronto.ca/blog/what-is-a-full-stack-developer/>. University of Toronto School of Continuing Studies.
- [Vértice, 2009] Vértice, E. (2009). Diseño básico de páginas web en html. In <https://books.google.es/books?id=Q4VL8Tiy7gcC>, Internet / Vértice, page 12. Publicaciones Vértice SL.

- [Wikimedia, 2022] Wikimedia, P. (2022). Sondasespaciales. In *https://es.wikipedia.org/wiki/SondaEspacial.MediaWiki*.
- [Ángel, 2019] Ángel, R. (2019). ¿qué es mysql: Características y ventajas? In *https://openwebinars.net/blog/que-es-mysql/*. OpenWebinars.

Apéndices

Apéndice A. Electrónica empleada en la misión Orión

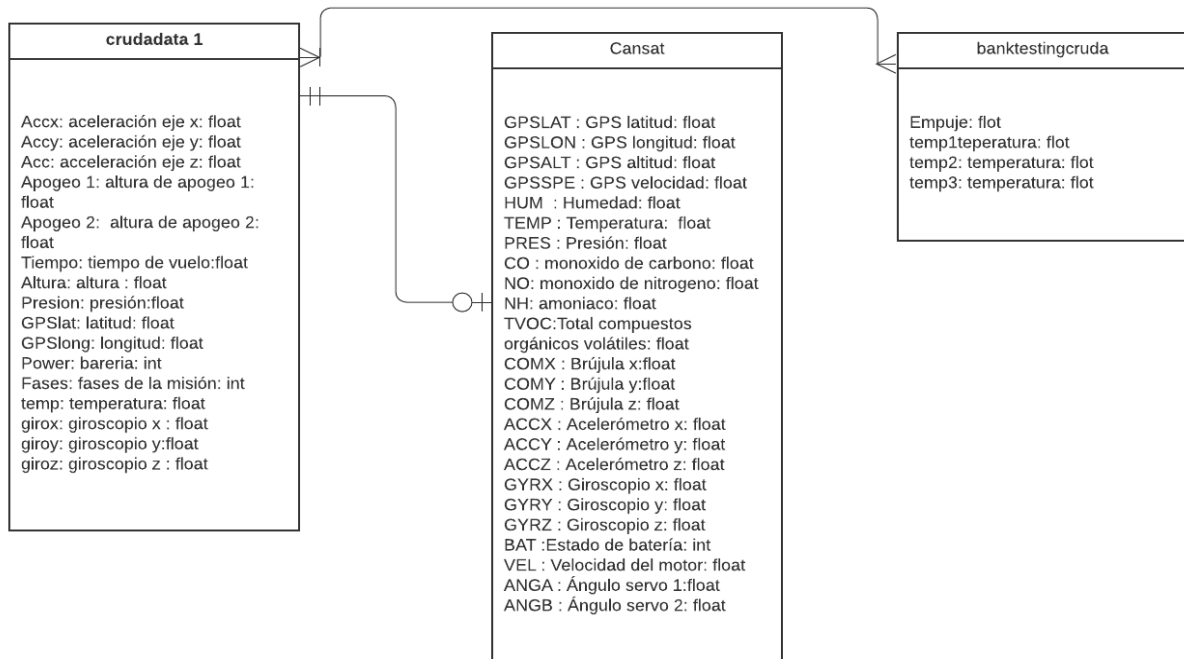
Microcontradores/PLacas de desarrollo		
Elemento	Descripción	Tipo
Módulo DEVKIT ESP32 30 pines	Microcontrolador tecnología THT para la etapa de procesamiento y control de banco de pruebas. Versión inicial, pruebas.	Micro - módulo
Módulo DEVKIT ESP32 38 pines	Microcontrolador alterno tecnología THT para la etapa de procesamiento y control de banco de pruebas. Versión inicial pruebas.	Micro - módulo
SMD ESP32S	Microcontrolador tecnología SMT para la etapa de procesamiento y control de banco de pruebas. Versión final implementable.	Micro - SMD
Módulo de comunicación		
Módulo LORA32 433/470 MHz Ra02	Módulo para la etapa de telemetría y transmisión de datos tomados por banco de pruebas. Versión inicial, pruebas THT.	LoRa - módulo
SMD LORA32 433/470 MHz Ra02	Módulo para la etapa de telemetría y transmisión de datos tomados por banco de pruebas. Versión final, implementación SMD.	LoRa - SMD
Módulo GPS Neo M8N	Módulo para la etapa de procesamiento y control, versión 1.	GPS - módulo
Módulo GPS GT-U7	Módulo para la etapa de procesamiento y control, versión 2.	GPS - módulo
Módulo GPS Neo 6M	Módulo para la etapa de procesamiento y control, versión 3.	GPS - módulo
Adaptador IPX a SMA	Adaptador para acoplar las antenas registradas a los respectivos módulos.	LoRa - Antena
Antena 433 MHz transmisor/receptor	Antena necesaria para la etapa de telemetría, a través de estas es posible establecer comunicación entre emisor y receptor.	LoRa - Antena
Antena Yagi 433 MHz 15dBi	Antena de alta potencia necesaria para asegurar alta potencia en recepción que permita confiar en la comunicación emisor/receptor.	LoRa - Antena
Cámara fpv	Cámara que permitirá transmitir y registrar video de los diferentes eventos de pruebas en el banco.	Cámara
Sensores		
Módulo BNO055	Módulo THT sensor indispensable para la etapa de procesamiento y control. A través de él se podrá registrar medidas inerciales. Versión inicial, pruebas.	IMU - módulo
SMD BNO055	Circuito integrado SMD sensor indispensable para la etapa de procesamiento y control. A través de él se podrá registrar medidas inerciales. Versión final, implementable.	IMU - SMD
Módulo GY-63 MS5611	Módulo THT sensor indispensable para la etapa de procesamiento y control. A través de él se podrá registrar mediciones de temperatura y presión atmosférica. Versión inicial, pruebas.	Barómetro - módulo
SMD MS5611	Circuito integrado SMD sensor indispensable para la etapa de procesamiento y control. A través de él se podrá registrar medidas de temperatura y presión atmosférica. Versión final, implementable.	Barómetro - SMD
Módulo MPU6050	Módulo THT sensor indispensable para la etapa de procesamiento y control. A través de él se registran medidas inerciales. Versión alternativa, pruebas.	IMU - módulo
SMD MPU6050	Circuito integrado SMD sensor indispensable para la etapa de procesamiento y control. A través de él se podrá registrar medidas inerciales. Versión alternativa, implementable.	IMU - SMD
Módulo BMP280	Módulo THT sensor indispensable para la etapa de procesamiento y control. A través de él se registran mediciones de temperatura y presión atmosférica. Versión alternativa, pruebas.	Barómetro - módulo
SMD BMP280	Circuito integrado SMD sensor indispensable para la etapa de procesamiento y control. A través de él se registran medidas de temperatura y presión atmosférica. Versión alternativa, implementable.	Barómetro - SMD
Módulo Termocupla tipo K MAX6675	Módulo independiente necesario para el banco de pruebas, registra altas temperaturas.	Temperatura - módulo
Memoria		
Adaptador MicroSD	Módulo necesario para el almacenamiento local de los datos procesados.	SD - módulo
SMD Flash / EEPROM	Circuito integrado SMD necesario para el almacenamiento local de datos procesados.	Flash - SMD
Tarjeta SD Clase 10 32GB	Tarjetas microSD necesarias para almacenar los datos.	SD - microSD
Componentes		
Módulo MP2307 Regulador	Módulos indispensables para la etapa de potencia, a través de ellos se obtendrán los niveles de tensión deseados y requeridos de cada componente.	Regulador - módulo
Conector JST 4P hembra/macho	Conectores necesarios para las placas de prueba y de implementación.	Conector
Conector XT60 hembra/macho	Conectores necesarios para las placas de prueba y de implementación.	Conector
Bornera terminal 2P 5mm	Borneras necesarias para las respectivas salidas de la etapa de control.	Bornera
Bornera terminal 3P 5mm	Borneras necesarias para las respectivas salidas de la etapa de control.	Bornera
Buzzer pasivo 7.6mm	Dispositivo necesario para indicadores auditivos.	Buzzer
Transistor IRL520 TO-220	Dispositivo THT necesario para la señal de activación en la etapa de control.	Transistor Pyro
Transistor IFRZ44N TO-220	Dispositivo THT necesario para la señal de activación en la etapa de control.	Transistor Pyro
Transistor IFRZ44N TO-263	Dispositivo SMD necesario para la señal de activación en la etapa de control. Versión implementable.	Transistor Pyro - SMD
Led RGB 2 pines	Dispositivo necesario para indicadores visuales.	Led

Apéndice B. Tabla PLAN DE BANDA. (440 – 470 MHz)

Banda	Rango de frecuencias	Modo de uso
A	440 a 443 MHz	Dúplex con la banda A' con separación de 5 MHz
S1	443 a 445 MHz	Simplex
A'	445 a 448 MHz	Dúplex con la banda A con separación de 5 MHz
S2	448 a 450 MHz	Simplex
S3	450 a 450.6 MHz	Simplex
B	450.6 a 451.9 MHz	Dúplex con la banda B' con separación de 10 MHz
C	451.9 a 452.5 MHz	Dúplex con la banda C' con separación de 7.5 MHz
D	457.5 a 459.4 MHz	Dúplex con la banda D' con separación de 10 MHz
C'	459.4 a 460 MHz	Dúplex con la banda C con separación de 7.5 MHz
S4	460 a 460.6 MHz	Simplex
B'	460.6 a 461.9 MHz	Dúplex con la banda B con separación de 10 MHz
S5	461.9 a 462.5 MHz	Simplex
D'	467.5 a 469.4 MHz	Dúplex con la banda D con separación de 10 MHz

Apéndice C. Diagrama UML para la base de datos

Figura 49: Diagrama UML para la base de datos (Fuente: propia)



Apéndice D. Repositorio general del proyecto - GitHub

<https://github.com/Katheji/adminlte3>

Apéndice E. Página web del trabajo de investigación

<https://sites.google.com/view/orion-guanesat/inicio>