

REPORTTE TÉCNICO

DISEÑO, CONSTRUCCIÓN Y
PRUEBAS DE UN COHETE

ELABORADO POR:

ILIHUÍCATL
XOXOAUHCO



2024

IBERO
CIUDAD DE MÉXICO ®



ALAS
AEROSPACE LATIN AMERICAN SUBORBITALS

PATROCINADORES:

TRANSCOM

iniat

REPORTE TÉCNICO

DISEÑO, CONSTRUCCIÓN Y PRUEBAS DE UN COHETE

Autor

ILIHUÍCATL XOXOAUHCO

José Eduardo Crescencio Dávila - Director de Misión
Juan José Vidaña Hernandez - Ingeniero de Análisis de Datos de Vuelo
José Pablo Mejía Ibarra - Contador
Thalía Isabel Bravo Pimentel - Ingeniera de Diseño
Jorge Rojas Hernández - Ingeniero de Materiales
Juan Camilo Tejada Orjuela - Asesor
Mario Ramírez Neria - Asesor
Fernando Ian Yedra Lavalle - Asesor

Director de competencia

Corporación ALAS (@corp_alas)

Competencia

Concurso Colombiano de Cohetería Deportiva CCCD2024

Universidad Iberoamericana Ciudad de México

Ciudad de México, México

2024

IBERO
CIUDAD DE MÉXICO ®



ALAS
AEROSPACE LATIN AMERICAN SUBORBITALS

PATROCINADORES:

TRANSCOM

iniat

“Cuando lo creas todo perdido, no olvides que aún te queda el futuro, tu cerebro, tu voluntad y dos manos para cambiar tu destino.”

- Wernher von Braun

AGRADECIMIENTOS

Expresamos nuestro más sincero agradecimiento a la Universidad Iberoamericana Ciudad de México por su apoyo invaluable durante el proceso de diseño, construcción y fabricación. También extendemos nuestra gratitud al INIAT por el soporte brindado y el préstamo de equipos de laboratorio, esenciales para el prototipado. Agradecemos profundamente a Transcom por el generoso patrocinio que hizo posible cumplir este sueño. Finalmente, nuestro reconocimiento a ALAS Corp. por su valioso apoyo y la provisión de materiales que fueron clave para el desarrollo del proyecto.



“Plantilla de reporte elaborada para la Universidad Iberoamericana, licensed under a Creative Commons Attribution 4.0 International License.

ÍNDICE

AGRADECIMIENTOS	III
ÍNDICE DE TABLAS	VII
ÍNDICE DE FIGURAS	IX
ÍNDICE DE CÓDIGOS	X
1. INTRODUCCIÓN	1
2. METODOLOGÍA	3
3. DISEÑO Y ANÁLISIS	5
3.1. Diseño Preliminar	5
3.1.1. Contexto del problema	5
3.1.2. Requerimientos de usuario	6
3.1.2.1. Requerimientos Generales	6
3.1.2.2. 2. Requerimientos para el Cohete	7
3.1.2.3. 3. Requerimientos para la Carga Útil (CanSat)	7
3.1.2.4. 4. Requerimientos de Operación	8
3.1.3. Quality Function Deployment	8
3.2. Arquitectura Funcional	13
3.3. Estructura funcional	15
3.4. Diseño Básico	17
3.4.1. Cohete	17
3.4.1.1. Arquitectura física	17
3.4.2. Matriz morfológica	19
3.4.2.1. Matriz de selección	20
3.4.3. Cansat	21
3.4.3.1. Arquitectura física	21

3.4.4. Matriz morfológica	21
3.4.4.1. Matriz de selección	24
3.5. Diseño Detalle	24
3.5.1. Cohete	24
3.5.1.1. Cono	24
3.5.1.2. Estabilizadores	27
3.5.1.3. Estructura	28
3.5.1.4. Cordón de choque	28
3.5.1.5. Paracaídas	29
3.5.1.6. Cálculo del Centro de Presión	30
3.5.1.7. Simulaciones en Open Rocket	32
3.5.1.8. Diseño en software CAD	34
3.5.2. Cansat	35
3.5.2.1. Estructura	35
3.5.2.2. Sistema de Despliegue	36
3.5.2.3. Sistema Electrónico	36
3.5.2.4. Sistema de Comunicación	38
3.5.2.5. Fuente de Energía	39
3.5.2.6. Códigos para la medición y comunicación	39
4. PROTOTIPADO Y CONSTRUCCIÓN	52
4.1. Cohete	52
4.1.1. Motor	52
4.1.2. Cuerpo (Fuselaje)	52
4.1.3. Cono	53
4.1.4. Aletas y Estabilizadores	54
4.1.5. Cordón de Choque	54
4.1.6. Paracaídas	55
4.1.7. Anillos de Soporte para el Motor	55

ÍNDICE

4.1.8. Papel Ignífugo	56
4.1.8.1. Integración y montaje	56
4.2. Cansat	56
4.2.1. Materiales y Componentes Utilizados	57
4.2.2. Técnicas de Fabricación	57
4.2.3. Integración y Montaje	57
5. PRUEBAS Y VALIDACIÓN	59
5.1. Cohete	59
5.2. Cansat	60
6. LECCIONES APRENDIDAS	63
7. EXPERIENCIAS SIGNIFICATIVAS	64
8. FICHAS TÉCNICAS	65
9. BROCHURE	68
BIBLIOGRAFÍA	70
ANEXOS	71
A. Primer anexo	71

ÍNDICE DE TABLAS

3.1.	Casa de la Calidad - Matriz de Relaciones	12
3.2.	Matriz Morfológica para el Diseño del Cohete	19
3.3.	Matriz de Selección Basada en Características de Ingeniería	20
3.4.	Matriz Morfológica para el Diseño del CanSat	23
3.5.	Matriz de Selección para el Diseño del CanSat	24
3.6.	Detalles de las partes del cohete.	32
3.7.	Condiciones de simulación para el lanzamiento.	32
3.8.	Consumo energético de los componentes del CanSat	39

ÍNDICE DE FIGURAS

2.1. Descripción de la etapa de diseño del proyecto.	4
2.2. Descripción de las etapas de fabricación y pruebas.	4
3.1. Arquitectura Funcional del Sistema	14
3.2. Estructura Funcional del Sistema, azul corresponde a entradas, verde a procesos y rojo a salidas	16
3.3. Arquitectura física del cohete.	18
3.4. Arquitectura física del Cansat.	22
3.5. Vista isométrica del cono.	25
3.6. Plano del cono.	26
3.7. Vista isométrica de los estabilizadores.	27
3.8. Plano de los estabilizadores.	28
3.9. Triángulo isósceles.	29
3.10. Geometría de simulación Open Rocket.	33
3.11. Perfil de vuelo simulado en Open Rocket.	33
3.12. Estabilidad vs Tiempo.	34
3.13. Estabilidad vs Tiempo.	34
3.14. a) Vista del modelo del cohete ensamblado. b) Vista de los componentes internos del cohete. c) Vista explosionada de los componentes finales del cohete.	35
3.15. Modelo del CanSat en inventor	36
3.16. Circuito del sistema electrónico.	38
4.1. a)Fuselaje del cohete. b) Acople. c) Sección carga útil.	53
4.2. Cono del cohete.	53
4.3. Aletas y estabilizadores del cohete.	54
4.4. Paracaídas extendido	55
4.5. Anillos de soporte para el motor del cohete.	55
4.6. Ensamble del cohete final.	56
4.7. a) Placas base de desarrollo. b) Integración de componentes. c) Ensamble final de la electrónica	58

4.8. CanSat Final Armado	58
5.1. Primer lanzamiento del cohete.	59
5.2. Segundo lanzamiento de cohete tomando en cuenta la reducción del tamaño y peso simulado en OpenRocket.	60
5.3. Lanzamiento del prototipo final de cohete.	61
5.4. Fotos de la prueba de Cansat con dron.	61
5.5. Resultados de la prueba final.	62

ÍNDICE DE CÓDIGOS

3.1. Código utilizado para la captura y envío de datos en el microcontrolador, parte 1.	41
3.2. Código utilizado para la captura y envío de datos en el microcontrolador, parte 2.	42
3.3. Código utilizado para la captura y envío de datos en el microcontrolador, parte 3.	43
3.4. Código utilizado para la recepción de datos y transmisión al computador.	44
3.5. Código utilizado para el procesamiento de datos y generación de gráficas en tiempo real.	45
3.6. Código utilizado para el procesamiento de datos y generación de gráficas en tiempo real. Parte 2.	46
3.7. Código utilizado para el procesamiento de datos y generación de gráficas en tiempo real. Parte 3.	47
3.8. Código utilizado para el procesamiento de datos y generación de gráficas en tiempo real. Parte 4.	48
3.9. Código utilizado para el procesamiento final de datos. Parte 1.	49
3.10. Código utilizado para el procesamiento final de datos. Parte 2.	50
3.11. Código utilizado para el procesamiento final de datos. Parte 3.	51

1. INTRODUCCIÓN

En la mitología mexica, Ilhuícatl-Xoxoauhco es el séptimo cielo en la estructura del universo según su cosmovisión. Este lugar, que representa el sitio donde el sol aparece al amanecer, está bajo el control de Huitzilopochtli, el dios del sol, la voluntad, la guerra, las estrategias militares, las batallas y el fuego, además de ser el protector del sur. Esta rica simbología de nuestro antepasados nos inspiró para tomar este nombre como identidad de nuestro equipo, misión y cohete, reflejando la fuerza, estrategia y propósito que guía nuestro proyecto.

En este documento se detalla el desarrollo de un cohete de propelente sólido de clase G con motor AEROTECH G80-7T, diseñado y construido para la participación en la categoría universitaria del Concurso Colombiano de Cohetería Deportiva (CCCD2024). Este proyecto tiene como objetivo principal cumplir con los estrictos requisitos técnicos establecidos por el comité organizador, los cuales abarcan tanto aspectos de diseño como de seguridad y desempeño durante el vuelo.

El cohete, diseñado como una única etapa con una longitud mínima de 80 cm y una masa máxima de 1200 g, está configurado para ser reutilizable. Su sistema de propulsión emplea el motor comercial AEROTECH G80-7T, certificado por la Tripoli Rocketry Association y suministrado durante el evento. Además, la estabilidad del vuelo se garantiza mediante el cumplimiento de parámetros como la relación empuje-peso mínima de 5:1 y una velocidad inicial superior a 55 km/h.

La carga útil, cuyo diseño es de libre elección, debe cumplir con una masa mínima de 250 g y estará equipada con un altímetro electrónico de precisión. Es importante resaltar que esta carga deberá cumplir objetivos funcionales bien definidos, no podrá incluir materiales peligrosos ni animales, y debe permanecer unida al cohete durante toda la misión, asegurando un descenso conjunto y seguro mediante el uso de un sistema de paracaídas. Para este caso en particular se ha definido usar un CANSAT que permite participar en esta categoría adicional.

Por último, la operación del lanzamiento se llevará a cabo utilizando las plataformas proporcionadas por ALAS, que incluyen un riel de aluminio de 2 metros con una inclinación máxima de 15°. Este sistema garantiza condiciones estándar y seguras para todos los equipos participantes.

A lo largo del presente informe se describen en detalle las etapas de diseño, construcción y validación del cohete y su carga útil, asegurando el cumplimiento de los requisitos establecidos en los documentos *Requerimientos Técnicos* y *Guía de Participantes*. Asimismo, se abordan los retos enfrentados durante el desarrollo del proyecto y las soluciones implementadas para alcanzar los objetivos propuestos.

Como principales resultados del proyecto se logra obtener diversos prototipos de cohetes propulsados por motores tipo C y D de prueba preliminar y un prototipo de cohete para la competencia. Así mismo se logra obtener un prototipo de carga útil basado en un CANSAT que permite la medición de variables ambientales y a su vez permite establecer un análisis atmosférico en el diámetro de descenso.

El documento está organizado de la siguiente forma: sección 2 metodología que permite establecer la forma en la que se aborda el estudio, la sección 3 diseño y análisis que permite establecer las etapas y elementos considerados para el diseño del cohete y del cansat, la sección 4 prototipado y construcción que permite conocer las etapas de fabricación, materiales y estrategias de integración para permitir reducir la incertidumbre y abordar de forma adecuada la solución del reto, la sección 5 pruebas y validación que permite conocer las pruebas realizadas, sus resultados y análisis. Finalmente las secciones 6 y 8 permiten conocer los aprendizajes más significativos

1. INTRODUCCIÓN

como equipo. Finalmente se presenta la ficha técnica y el brochure de los desarrollos obtenidos.

2. METODOLOGÍA

El desarrollo de este proyecto se estructuró siguiendo las metodologías propuestas por Dieter y Schmidt, 2012 en *Engineering Design* y por Ulrich y Eppinger, 2016 en *Product Design and Development*. Estas metodologías ofrecen un marco sistemático y robusto que permite abordar proyectos de diseño de ingeniería desde su conceptualización hasta su implementación. En particular, se adoptaron los principios de diseño preliminar, básico y de detalle descritos por Dieter, integrándolos con los enfoques iterativos de desarrollo de productos y toma de decisiones jerárquica planteados por Ulrich.

El enfoque metodológico aplicado no solo proporcionó un camino estructurado para establecer los criterios de diseño, sino que también permitió la definición precisa de las etapas del proyecto. Este marco aseguró que el cohete y su carga útil cumplieran con los estándares técnicos y de seguridad exigidos por el Concurso Colombiano de Cohetería Deportiva (CCCD2024), promoviendo un diseño funcional y competitivo.

Las etapas del desarrollo se definieron de la siguiente manera:

1. **Diseño preliminar:** Se identifican las partes del proyecto, los requerimientos del usuario y las características generales de los sistemas. Esta etapa culmina con la elaboración de una arquitectura funcional que detalla las funciones principales y los problemas a resolver, definiendo claramente el *qué* del proyecto.
2. **Diseño básico:** Se especifican las partes y componentes que conformarán la solución propuesta. En esta etapa se responde al *cómo*, con base en los requerimientos funcionales y técnicos establecidos previamente.
3. **Diseño de detalle:** Se definen las dimensiones exactas, materiales, elementos y ensamblajes de cada componente. Este nivel de diseño ofrece una representación realista y detallada de la solución final a implementar.
4. **Fabricación:** A partir del diseño de detalle, se procede a la construcción de los prototipos utilizando los materiales seleccionados. Esta etapa busca traducir el diseño conceptual en un producto tangible.
5. **Pruebas:** Se realizan pruebas funcionales y de desempeño para validar los prototipos y ajustar los diseños según sea necesario. Estas pruebas son críticas para garantizar que el modelo final cumpla con los requerimientos de la competencia.

En la Figura 2.1 se presenta un desglose de las etapas internas relacionadas con el diseño del proyecto, mientras que la Figura 2.2 detalla las fases de fabricación y pruebas. Estas representaciones gráficas ofrecen una visión clara y comprensible del proceso metodológico aplicado.

2. METODOLOGÍA

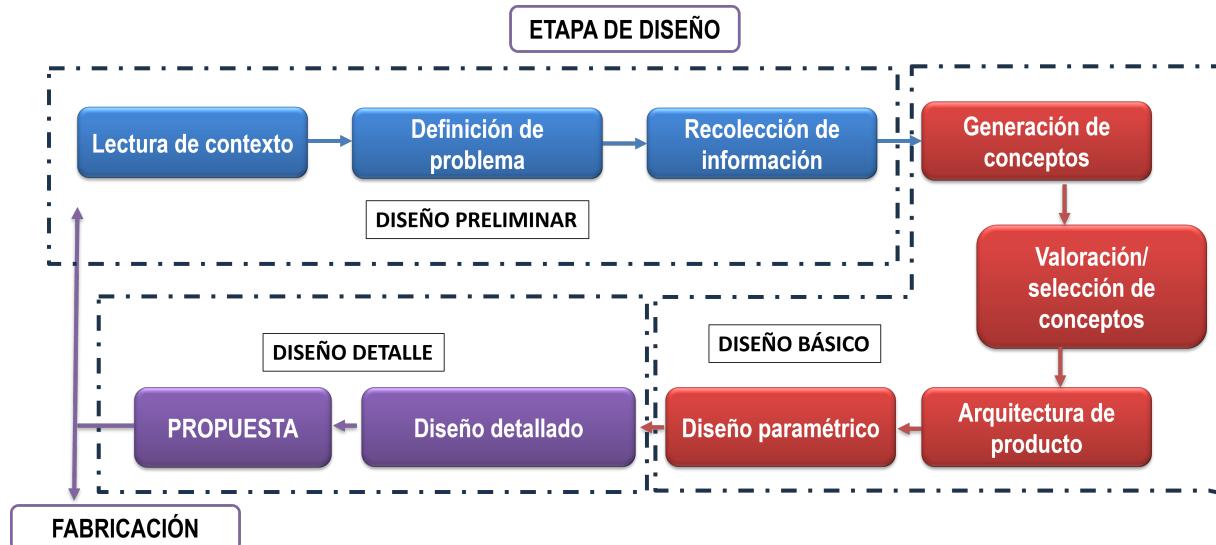


Figura 2.1: Descripción de la etapa de diseño del proyecto.

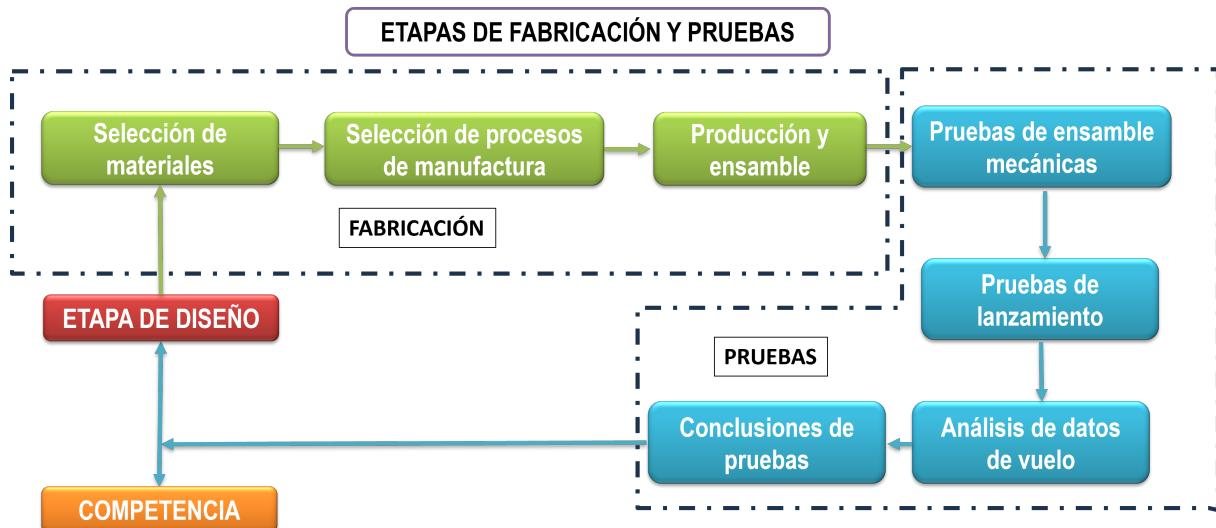


Figura 2.2: Descripción de las etapas de fabricación y pruebas.

Las siguientes secciones, (3) Diseño y Análisis, (4) Prototipado y Construcción, y (5) Pruebas y Validación, presentan un análisis detallado de los resultados alcanzados en cada fase, destacando los logros y aprendizajes obtenidos en el desarrollo del proyecto.

3. DISEÑO Y ANÁLISIS

El diseño y análisis constituyen una de las fases más críticas en el desarrollo del proyecto, ya que permiten transformar los requerimientos funcionales en soluciones técnicas viables. En esta etapa, se llevó a cabo la definición de la arquitectura funcional del cohete y su carga útil, especificando las funciones principales y secundarias de cada componente. Asimismo, se realizaron cálculos y simulaciones para garantizar el cumplimiento de parámetros clave, como estabilidad, relación empuje-peso y desempeño aerodinámico. Estas actividades estuvieron alineadas con los principios metodológicos establecidos en la etapa de diseño preliminar, básico y de detalle, y fueron fundamentales para asegurar que el cohete y su carga útil cumplieran con las especificaciones establecidas por el Concurso Colombiano de Cohetería Deportiva (CCCD2024). A lo largo de esta sección, se presentan los resultados del diseño conceptual y detallado, así como los análisis técnicos que respaldan la viabilidad del proyecto.

3.1. Diseño Preliminar

3.1.1. Contexto del problema

La cohetería ha sido históricamente uno de los pilares fundamentales en el desarrollo de la tecnología aeroespacial, permitiendo el acceso al espacio y habilitando misiones científicas, comerciales y educativas. En su nivel más básico, el diseño y construcción de cohetes implican la integración de múltiples disciplinas, como la dinámica de fluidos, la propulsión, la aerodinámica y los sistemas de recuperación, lo que los convierte en herramientas ideales para la formación práctica en ingeniería.

En el ámbito académico, la cohetería experimental ha ganado relevancia como una forma de introducir a estudiantes en los principios de diseño, análisis y validación de vehículos aeroespaciales. Según Portocarrero-Hermann et al., 2021, el desarrollo de motores cohete de combustible sólido tipo Candy ha demostrado ser una metodología accesible y práctica para proyectos educativos, al combinar un enfoque experimental con requisitos técnicos rigurosos. Estos motores, junto con el diseño conceptual y las pruebas estáticas de cohetes, ofrecen una base sólida para el aprendizaje práctico y la validación de modelos teóricos.

Un aspecto crucial en el diseño de cohetes es garantizar la estabilidad durante el vuelo. Trabajos como el de Riveros Enciso y Rodríguez Herrera, 2010 han explorado tecnologías avanzadas, como toberas de empuje vectorial, para mejorar la maniobrabilidad y la estabilidad de los cohetes aficionados. Aunque estas técnicas no se aplican directamente en competiciones como el Concurso Colombiano de Cohetería Deportiva (CCCD2024), subrayan la importancia de priorizar diseños que aseguren un vuelo estable y seguro.

Por su parte, el CCCD2024 establece requerimientos específicos para los cohetes en la categoría universitaria:

- El cohete debe ser de una única etapa, con una longitud mínima de 80 cm y una masa máxima de 1200 g al momento del lanzamiento.
- Debe cumplir con una relación empuje-peso mínima de 5:1 y garantizar una velocidad inicial superior a 55 km h^{-1} .
- El sistema de recuperación debe garantizar el descenso seguro mediante el uso de para-

3. DISEÑO Y ANÁLISIS

caídas.

- El motor AEROTECH G80-7T, certificado y proporcionado por la competencia, debe ser utilizado para la propulsión.

Por otro lado, la carga útil tipo CanSat representa un desafío adicional, ya que no solo debe integrarse perfectamente en el cohete, sino también cumplir con su propia misión científica o de ingeniería. Los CanSat, concebidos originalmente como herramientas educativas, simulan las funciones de un satélite real en un volumen compacto equivalente al de una lata de refresco. Según Ramírez et al., 2019, los CanSat ofrecen una experiencia práctica en el diseño y construcción de sistemas espaciales en miniatura, incluyendo subsistemas como potencia, sensores y comunicación.

El CCCD2024 establece requerimientos específicos para la carga útil tipo CanSat:

- **Masa:** Entre 250 y 350 g.
- **Dimensiones:** Cilindro de 115 mm de altura y 66 mm de diámetro.
- **Sensores:** Debe incluir una Unidad de Medición Inercial (IMU) y sensores que registren al menos tres variables físicas diferentes, como temperatura, presión y humedad.
- **Almacenamiento:** Los datos deben guardarse en formato CSV y ser accesibles inmediatamente tras la recuperación.
- **Fuente de alimentación:** Una batería con una duración mínima de 4 horas, fácilmente reemplazable.
- **Montaje:** La carga útil debe estar conectada al cohete mediante un cordón de choque en caso de eyeción.

3.1.2. Requerimientos de usuario

A continuación, se presenta la lista de requerimientos de usuario para el diseño y construcción del cohete y la carga útil tipo CanSat, de acuerdo con los lineamientos del Concurso Colombiano de Cohetería Deportiva (CCCD2024).

3.1.2.1 Requerimientos Generales

- **Seguridad:**
 - El sistema debe garantizar la seguridad de los operadores y espectadores durante el lanzamiento, vuelo y recuperación.
 - El cohete y la carga útil no deben contener materiales peligrosos o sustancias inflamables, excepto las previstas por los organizadores.
 - Deben emplearse sistemas de recuperación que eviten daños al cohete, la carga útil y el entorno al aterrizar.
- **Conformidad:**
 - El diseño debe cumplir con los lineamientos establecidos por la competencia (CCCD2024), incluyendo dimensiones, peso y materiales permitidos.

- Todos los componentes deben ser legales y certificados según las regulaciones aplicables.

- **Facilidad de Ensamblaje:**

- El cohete debe ser fácil de ensamblar y desensamblar en el lugar de lanzamiento, con instrucciones claras para su montaje.

- **Reutilización:**

- El cohete debe ser reutilizable, permitiendo múltiples lanzamientos con un mantenimiento mínimo.

- **Fiabilidad:**

- El sistema debe garantizar un vuelo estable y preciso, alcanzando la altura objetivo de 400 metros.

3.1.2.2 2. Requerimientos para el Cohete

- **Diseño Aerodinámico:**

- El cohete debe ser aerodinámicamente estable con un margen de estabilidad estática no inferior a 2.0.
- Debe incluir un sistema de aletas optimizado para mantener la dirección durante el vuelo.

- **Propulsión:**

- El cohete debe emplear el motor AEROTECH G80-7T proporcionado por los organizadores.
- La relación empuje-peso debe ser al menos de 5:1 para garantizar un despegue seguro.

- **Sistemas de Recuperación:**

- Debe contar con un sistema de paracaídas que garantice un descenso conjunto y seguro del cohete y su carga útil.

- **Identificación:**

- El cohete debe incluir el logotipo de la competencia y un identificador único asignado por los organizadores.

3.1.2.3 3. Requerimientos para la Carga Útil (CanSat)

- **Misión:**

- La carga útil debe cumplir con una misión científica o de ingeniería bien definida.
- Debe registrar variables ambientales (por ejemplo, temperatura, presión, humedad) y almacenar los datos en formato CSV.

- **Dimensiones y Peso:**

3. DISEÑO Y ANÁLISIS

- El CanSat debe ser compacto, con un tamaño máximo de 115 mm de altura y 66 mm de diámetro, y pesar entre 250 y 350 gramos.

- **Sensores y Electrónica:**

- Debe incluir una Unidad de Medición Inercial (IMU) y sensores para registrar al menos tres variables físicas.
- Debe tener capacidad de almacenamiento interno y/o transmisión de datos.

- **Fuente de Energía:**

- El sistema debe contar con una batería de al menos 4 horas de duración, fácilmente reemplazable.

- **Integración:**

- El CanSat debe estar firmemente acoplado al cohete y no separarse durante el vuelo, excepto si está conectado mediante un cordón de choque.

3.1.2.4 4. Requerimientos de Operación

- **Lanzamiento:**

- El sistema debe ser compatible con las plataformas de lanzamiento proporcionadas por los organizadores, con un riel de 2 metros y una inclinación máxima de 15°.

- **Pruebas:**

- El sistema debe ser probado previamente para validar la estabilidad, el rendimiento y el cumplimiento de los requisitos técnicos.

- **Recuperación de Datos:**

- Los datos registrados por la carga útil deben ser accesibles inmediatamente tras el aterrizaje para análisis posteriores.

3.1.3. Quality Function Deployment

De todo lo anterior se extrajeron los 20 requerimientos más importantes y se convirtieron en características de ingeniería.

Requerimientos de Usuario

1. Garantizar la seguridad de operadores y espectadores.
2. Cumplir con los lineamientos del CCCD2024.
3. Facilidad de ensamblaje y desmontaje.
4. Reutilización del cohete para múltiples lanzamientos.
5. Fiabilidad del sistema para alcanzar 400 m de altura.

6. Diseño aerodinámico estable.
7. Uso del motor AEROTECH G80-7T.
8. Relación empuje-peso de al menos 5:1.
9. Sistema de recuperación con paracaídas.
10. Incluye logotipo y número de identificación.
11. Dimensiones del CanSat compactas (115 mm x 66 mm).
12. Peso del CanSat entre 250 y 350 g.
13. Sensores de medición inercial y variables físicas.
14. Datos en formato CSV accesibles tras recuperación.
15. Batería de 4 horas de duración mínima.
16. CanSat acoplado firmemente al cohete.
17. Compatibilidad con la plataforma de lanzamiento.
18. Pruebas previas para validar estabilidad y rendimiento.
19. Recuperación de datos para análisis post-lanzamiento.
20. Misión científica o de ingeniería bien definida.

Características de Ingeniería

1. Materiales seguros y certificados.
2. Sistemas de propulsión confiables.
3. Conexiones modulares.
4. Resistencia del fuselaje.
5. Sistema de estabilización aerodinámica.
6. Relación peso-potencia óptima.
7. Compatibilidad con el motor AEROTECH G80-7T.
8. Durabilidad del sistema de recuperación.
9. Precisión en la fabricación del CanSat.
10. Optimización del almacenamiento de datos.
11. Capacidad de transmisión inalámbrica.
12. Autonomía de la fuente de energía.
13. Robustez de los sensores del CanSat.
14. Configuración de software para análisis de datos.

3. DISEÑO Y ANÁLISIS

15. Adaptabilidad a diferentes entornos de lanzamiento.
16. Fiabilidad del sistema de eyección.
17. Diseño compacto y liviano del CanSat.
18. Control de calidad en materiales y ensamblaje.
19. Cumplimiento de las tolerancias estructurales.
20. Estabilidad durante el vuelo.

La siguiente matriz de relaciones establece las conexiones entre los requerimientos de usuario y las características de ingeniería más importantes para el diseño del cohete y su carga útil tipo CanSat, se presenta como una simplificación de la casa de calidad de la metodología. La matriz de relaciones entre los requerimientos de usuario y las características de ingeniería se muestra en la Tabla 3.1.

A continuación, se presenta un análisis de los resultados obtenidos en la matriz de relaciones:

Relaciones Clave

- **Materiales seguros y certificados:**

- Altamente relacionados con la seguridad (9), la durabilidad del sistema de recuperación (9), y la estabilidad durante el vuelo (9).
- Fundamental para garantizar un diseño confiable y seguro.

- **Sistemas de propulsión confiables:**

- Relación fuerte con cumplir los lineamientos del CCCD2024 (9), la relación empuje-peso (9), y el motor (9).
- Impacta directamente en la capacidad del cohete para alcanzar la altura objetivo y mantener la estabilidad.

- **Resistencia del fuselaje:**

- Relacionada con la seguridad (9), la fiabilidad del sistema (9), y la estabilidad aerodinámica (9).
- Asegura la integridad estructural durante las condiciones de vuelo y recuperación.

- **Durabilidad del sistema de recuperación:**

- Conexión fuerte con el sistema de recuperación (9), la seguridad (9), y las pruebas (9).
- Garantiza que el cohete y el CanSat desciendan de forma segura y funcional.

- **Precisión en la fabricación del CanSat:**

- Alta correlación con las dimensiones del CanSat (9), los sensores (9), y el almacenamiento de datos (9).
- Impacta directamente en el cumplimiento de la misión científica o de ingeniería.

- **Estabilidad durante el vuelo:**

- Relación fuerte con la aerodinámica (9), el cumplimiento de las tolerancias estructurales (9), y el sistema de estabilización (9).
- Es esencial para garantizar el éxito del lanzamiento y el vuelo.

Características de Ingeniería Más Importantes

Basado en el análisis de las relaciones clave, las características de ingeniería más importantes son:

1. **Materiales seguros y certificados:** Garantizan seguridad y cumplimiento de normas técnicas, afectando múltiples requerimientos, como durabilidad y estabilidad.
2. **Sistemas de propulsión confiables:** Esencial para el rendimiento del cohete, cumplimiento del empuje-peso y uso correcto del motor.
3. **Resistencia del fuselaje:** Asegura que el cohete soporte las cargas dinámicas durante el lanzamiento y recuperación.
4. **Durabilidad del sistema de recuperación:** Crítica para la integridad del cohete y el CanSat tras el aterrizaje.
5. **Precisión en la fabricación del CanSat:** Fundamental para cumplir con los objetivos de misión científica y de ingeniería.
6. **Estabilidad durante el vuelo:** Garantiza un lanzamiento recto y seguro, minimizando riesgos.

Tabla 3.1: Casa de la Calidad - Matriz de Relaciones

Características de Ingeniería	Seguridad	Cumplir CCCD2024	Ensamblaje	Reutilización	Riabilidad	Aerodinámica	Motor	Empuje-peso	Recuperación	Identificación	Dimensiones CanSat	Peso CanSat	Sensores	Datos CSV	Batería	Acople CanSat	Plataforma	Pruebas	Datos Post	Misión
Materiales seguros	9	3	1	3	9	3	9	1	9	3	1	3	1	3	1	9	3	3	1	9
Sistemas de propulsión	3	9	3	1	3	9	9	9	3	3	3	9	3	9	3	3	3	1	3	3
Conexiones modulares	1	1	9	3	1	9	1	1	3	3	3	3	1	9	3	1	3	3	9	1
Resistencia del fuselaje	3	3	3	9	1	9	1	3	1	3	1	3	1	9	1	3	3	9	3	3
Sistema de estabilización aerodinámica	9	3	1	1	9	9	3	3	1	3	3	3	3	3	1	9	1	9	3	9
Relación peso-potencia óptima	3	9	1	3	9	9	3	1	3	3	3	3	1	1	3	9	3	9	1	9
Compatibilidad con el motor	9	9	3	9	3	3	9	3	9	3	3	3	9	3	1	3	3	9	1	3
Durabilidad del sistema de recuperación	3	1	1	9	3	9	1	9	3	3	3	3	3	3	3	3	1	9	9	3
Precisión en fabricación del CanSat	9	3	9	1	1	9	9	3	9	9	9	3	9	9	1	1	1	3	3	9
Optimización del almacenamiento de datos	3	3	3	3	9	3	3	3	3	9	3	9	3	1	9	3	3	1	3	9
Capacidad de transmisión inalámbrica	1	9	1	3	3	1	1	9	3	9	9	9	3	1	3	3	1	3	9	3
Autonomía de la fuente de energía	3	9	3	9	1	3	9	1	9	9	9	9	9	9	9	3	9	1	3	1
Robustez de los sensores del CanSat	9	3	3	1	1	3	9	9	3	9	3	9	9	3	3	3	1	3	9	1
Configuración de software para datos	3	9	9	3	3	9	3	3	3	9	9	3	3	9	3	9	3	3	9	1
Adaptabilidad a entornos de lanzamiento	1	3	3	9	1	3	9	9	1	3	3	1	3	3	3	3	3	3	9	1
Fiabilidad del sistema de eyección	9	3	1	3	9	9	3	3	1	3	9	1	1	3	3	3	9	9	9	9
Diseño compacto y liviano del CanSat	3	3	3	1	3	1	3	9	9	3	3	1	1	9	9	9	3	9	9	3
Control de calidad en ensamblaje	9	1	9	3	9	9	9	1	1	3	3	1	1	9	3	1	9	1	3	9
Cumplimiento de tolerancias estructurales	1	1	9	3	9	3	3	1	1	3	9	1	9	3	3	9	1	9	9	1
Estabilidad durante el vuelo	9	9	3	9	3	1	1	9	9	9	3	9	3	1	1	3	9	9	3	9

3.2. Arquitectura Funcional

La arquitectura funcional del sistema se organiza en bloques modulares que representan las principales funciones y subfunciones requeridas para garantizar el éxito del cohete y su carga útil tipo CanSat. La Figura 3.1 muestra un diagrama esquemático que detalla estas funciones y sus interacciones.

■ **Sistema de Propulsión:**

- Generar el empuje necesario para alcanzar la altura objetivo.
- Proveer estabilidad inicial al cohete durante el lanzamiento.

■ **Sistema Aerodinámico:**

- Mantener la estabilidad del vuelo mediante aletas estabilizadoras.
- Reducir el arrastre aerodinámico para optimizar el rendimiento.

■ **Sistema de Recuperación:**

- Garantizar un aterrizaje seguro del cohete y su carga útil.
- Implementar paracaídas para desacelerar el descenso.

■ **Carga Útil (CanSat):**

- Cumplir con una misión científica o de ingeniería definida.
- Medir y registrar variables ambientales (temperatura, presión, humedad).
- Almacenar datos en formato accesible para análisis posterior.

■ **Sistema de Energía:**

- Alimentar los componentes electrónicos del CanSat.
- Garantizar una autonomía mínima de 4 horas.

■ **Sistema de Comunicación:**

- Transmitir datos del CanSat a la estación base durante el vuelo (opcional).
- Facilitar la recuperación y análisis de datos tras el aterrizaje.

■ **Sistema de Control y Estructura:**

- Integrar los subsistemas en un diseño robusto y compacto.
- Soportar las fuerzas dinámicas durante el lanzamiento y el vuelo.

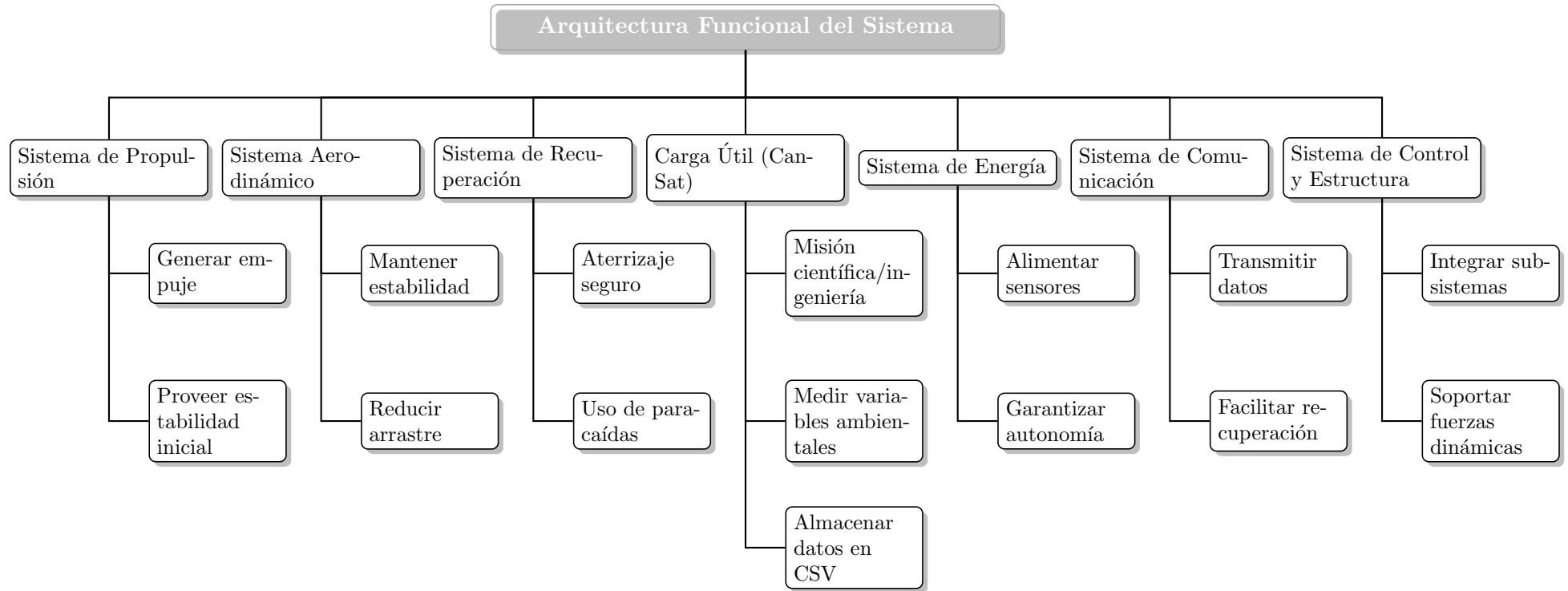


Figura 3.1: Arquitectura Funcional del Sistema

3.3. Estructura funcional

La estructura funcional del sistema representa las interacciones clave entre los diferentes subsistemas involucrados en el diseño del cohete y la carga útil tipo CanSat. Cada subsistema cumple un rol específico para garantizar el éxito de la misión. Las entradas principales, como energía, materiales, datos de misión y comandos de operación, alimentan los procesos centrales del sistema, que incluyen propulsión, aerodinámica, recuperación, carga útil y comunicación. Estos procesos trabajan de manera coordinada para producir las salidas esperadas, como alcanzar la altura objetivo, mantener la estabilidad durante el vuelo, asegurar un aterrizaje seguro y registrar datos científicos. Esta representación permite una visualización clara de cómo los elementos se integran en un flujo lógico, destacando la importancia de cada componente en el cumplimiento de los objetivos generales del sistema. Además, se emplean flechas de doble sentido para reflejar relaciones bidireccionales entre los subsistemas, como la dependencia entre los sensores de la carga útil y el sistema de comunicación. En la Figura 3.2 se observa esta relación entre las entradas, procesos y salidas del sistema. Esta estructura junto a la arquitectura son la salida del diseño preliminar permitiendo estructurar de forma adecuada el diseño básico y de detalle.

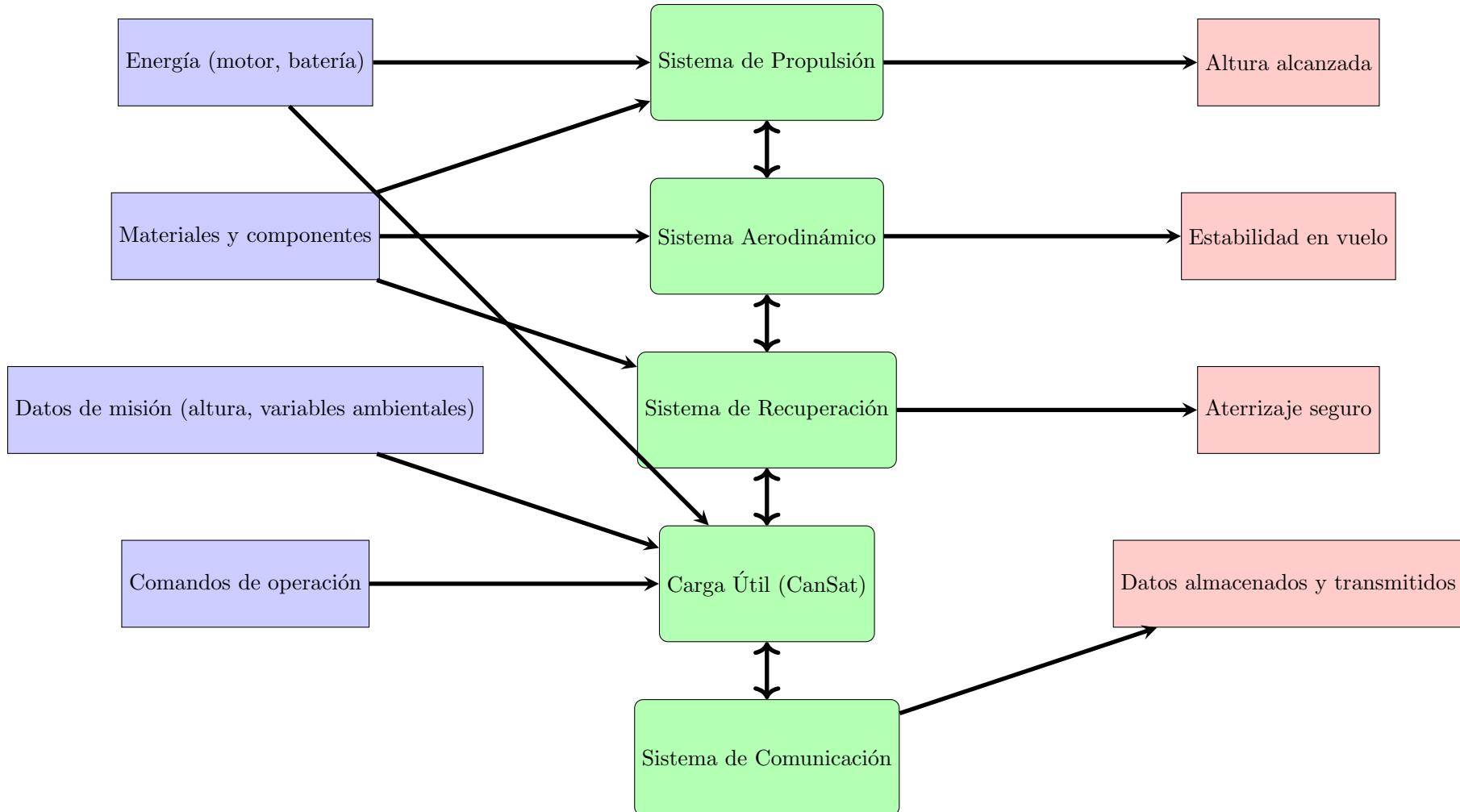


Figura 3.2: Estructura Funcional del Sistema, azul corresponde a entradas, verde a procesos y rojo a salidas

3.4. Diseño Básico

El diseño básico del sistema establece las bases para la configuración final del cohete y la carga útil tipo CanSat, integrando las decisiones técnicas necesarias para cumplir con los objetivos del proyecto. En esta etapa, se emplearon herramientas de diseño como la arquitectura física, la matriz morfológica y la matriz de selección para definir las principales características del sistema y evaluar diferentes alternativas de diseño.

La **arquitectura física** se utilizó para descomponer el sistema en subsistemas principales, identificando las interacciones físicas entre ellos. Esta herramienta permitió detallar los componentes clave, como el sistema de propulsión, la estructura aerodinámica, el mecanismo de recuperación y los módulos de la carga útil, asegurando su integración eficiente.

La **matriz morfológica** facilitó la generación de posibles soluciones técnicas para cada función principal del sistema. Al identificar múltiples alternativas para funciones como la estabilidad en vuelo, la recuperación segura y el registro de datos, se exploraron combinaciones que maximizan el rendimiento general del cohete.

Finalmente, la **matriz de selección** permitió evaluar y priorizar las alternativas generadas en la matriz morfológica. Utilizando criterios específicos como costo, facilidad de fabricación, confiabilidad y alineación con los requerimientos del CCCD2024, se seleccionaron las opciones que optimizan el desempeño técnico y funcional del sistema.

Al trabajar de esta forma estructurada, se busca asegurar que el diseño básico no solo cumpla con los requerimientos técnicos, sino que también provea una solución integrada, robusta y alineada con los objetivos de la misión. En las siguientes secciones, se presenta un análisis detallado de cada herramienta aplicada y los resultados obtenidos por separado para el cansat y el cohete.

3.4.1. Cohete

3.4.1.1 Arquitectura física

El diseño básico del sistema parte de la arquitectura funcional para identificar y estructurar los componentes físicos necesarios. A continuación, se describen las partes principales que se deben considerar en la arquitectura física del cohete:

■ Sistema de Propulsión:

- *Motor de propulsión sólida*: Proporciona el empuje necesario para alcanzar la altura objetivo de 400 metros.

■ Sistema Aerodinámico:

- *Aletas estabilizadoras*: Mantienen la estabilidad direccional del cohete durante el ascenso.
- *Nariz*: Reduce el arrastre aerodinámico y protege los componentes internos durante el vuelo.
- *Cuerpo del cohete*: Proporciona la estructura principal y aloja los sistemas internos.

■ Sistema de Recuperación:

- *Paracaídas*: Asegura un descenso controlado y seguro para el cohete y la carga útil.
- *Mecanismo de eyeción*: Libera el paracaídas en el momento adecuado.
- *Cordón de choque*: Conecta el paracaídas al cuerpo del cohete para evitar daños en el aterrizaje.

■ **Sistema de seguridad y Estructura:**

- *Armazón estructural*: Proporciona rigidez y soporte a todos los subsistemas.
- *Conexiones modulares*: Facilitan el ensamblaje y desensamblaje para pruebas y ajustes.
- *Protección térmica*: Evita daños por el calor generado durante el vuelo.

En la Figura 3.3 se pueden observar los componentes de la arquitectura física del cohete y sus conexiones entre ellos.

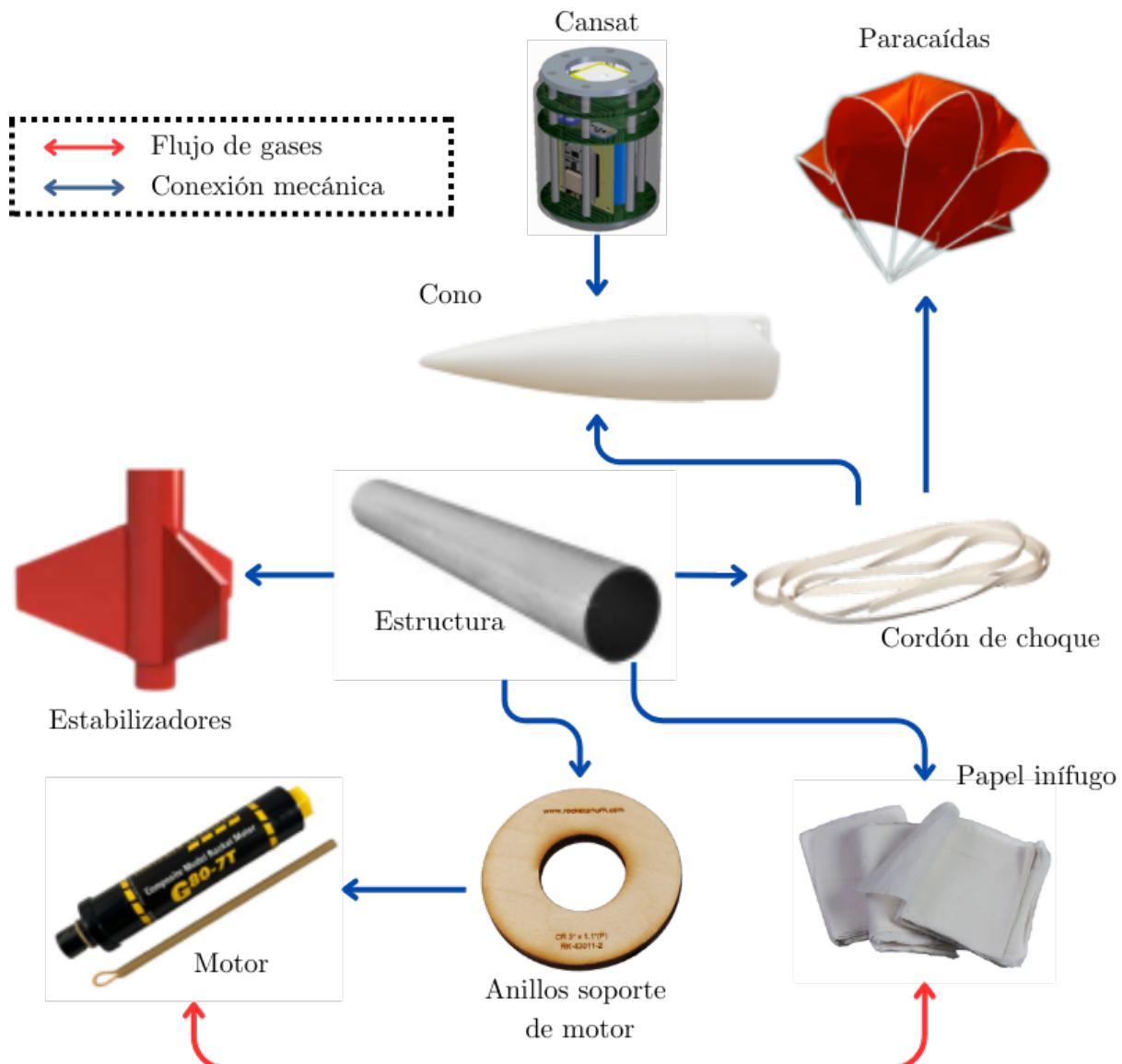


Figura 3.3: Arquitectura física del cohete.

3.4.2. Matriz morfológica

La matriz morfológica permite explorar soluciones técnicas para las funciones principales del cohete, como generar empuje, mantener estabilidad, reducir arrastre y recuperar el sistema. También considera aspectos específicos como la protección del sistema de eyección y el soporte del motor. Esta herramienta facilita la identificación de alternativas innovadoras y proporciona una base sólida para seleccionar combinaciones óptimas en etapas posteriores. En la Tabla 3.2 se presenta la matriz con las opciones propuestas para cada función.

Tabla 3.2: Matriz Morfológica para el Diseño del Cohete

Función	Solución 1	Solución 2	Solución 3
Generar empuje	Motor sólido clase G		
Reducir arrastre	Nariz cónica	Nariz parabólica	Nariz en forma de lágrima
Mantener estabilidad	Aletas trapezoidales	Aletas elípticas	Aletas rectangulares
Recuperar el cohete	Paracaídas hemisférico	Paracaídas cruciforme	Paracaídas cónico
Liberar el paracaídas	Sistema pirotécnico		
Proteger el sistema de eyección	Material ignífugo de fibra cerámica	Material ignífugo de papel inifugo	Algodón y talco
Proteger componentes internos	Fuselaje de fibra de carbono	Fuselaje de PVC	Fuselaje de plástico reforzado
Soportar y centrar el motor	Anillo de retención metálico	Soporte impreso en 3D	Cuna ajustable de aluminio
Disipar impactos durante la eyección	Cordón de nylon	Cordón de kevlar	Cordón elástico reforzado
Conectar subsistemas	Acoplamiento enroscado	Conexión magnética	Uniones a presión

Combinaciones de Diseño Basadas en la Matriz Morfológica

A continuación, se presentan tres combinaciones seleccionadas de la matriz morfológica para el diseño del cohete:

■ Combinación 1:

- **Generar empuje:** Motor sólido clase G.
- **Reducir arrastre:** Nariz cónica.
- **Mantener estabilidad:** Aletas elípticas.
- **Recuperar el cohete:** Paracaídas cruciforme.
- **Liberar el paracaídas:** Sistema pirotécnico.
- **Proteger el sistema de eyección:** Material ignífugo de fibra cerámica.
- **Proteger componentes internos:** Fuselaje de plástico reforzado.
- **Soportar y centrar el motor:** Anillo de retención metálico.
- **Disipar impactos durante la eyección:** Cordón de nylon.
- **Conectar subsistemas:** Acoplamiento enroscado.

■ Combinación 2:

- **Generar empuje:** Motor sólido clase G.
- **Reducir arrastre:** Nariz parabólica.
- **Mantener estabilidad:** Aletas trapezoidales.
- **Recuperar el cohete:** Paracaídas hemisférico.
- **Liberar el paracaídas:** Sistema pirotécnico.
- **Proteger el sistema de eyección:** Material ignífugo de papel.
- **Proteger componentes internos:** Fuselaje de PVC.
- **Soportar y centrar el motor:** Soporte impreso en 3D.
- **Disipar impactos durante la eyección:** Cordón elástico reforzado.
- **Conectar subsistemas:** Uniones a presión.

▪ **Combinación 3:**

- **Generar empuje:** Motor sólido clase G.
- **Reducir arrastre:** Nariz en forma de lágrima.
- **Mantener estabilidad:** Aletas rectangulares.
- **Recuperar el cohete:** Paracaídas cónico.
- **Liberar el paracaídas:** Sistema pirotécnico.
- **Proteger el sistema de eyección:** Algodón y talco.
- **Proteger componentes internos:** Fuselaje de fibra de carbono.
- **Soportar y centrar el motor:** Cuna ajustable de aluminio.
- **Disipar impactos durante la eyección:** Cordón elástico reforzado.
- **Conectar subsistemas:** Conexión magnética.

3.4.2.1 Matriz de selección

La matriz de selección evalúa las combinaciones propuestas en la matriz morfológica basándose en características de ingeniería clave, como estabilidad, peso, facilidad de ensamblaje y durabilidad. Cada criterio tiene un peso relativo, y las combinaciones se calificaron en una escala de 1 a 10, obteniendo un puntaje total ponderado. En la Tabla 3.3 se presenta la matriz de selección, destacando la combinación ganadora que mejor cumple con los objetivos del diseño.

Tabla 3.3: Matriz de Selección Basada en Características de Ingeniería

Criterios (Características de Ingeniería)	Peso (%)	C-1	C-2	C-3
Peso total reducido	20	8	9	7
Estabilidad durante el vuelo	15	7	9	8
Facilidad de ensamblaje	15	7	9	6
Resistencia al impacto en el aterrizaje	15	7	9	8
Durabilidad de los materiales	15	6	8	9
Aerodinámica eficiente	10	8	9	7
Sistema de eyección confiable	10	7	9	8
Protección térmica adecuada	10	7	9	8
Puntaje Total	100	7.35	8.85	7.7

3.4.3. Cansat

3.4.3.1 Arquitectura física

El diseño básico del sistema parte de la arquitectura funcional para identificar y estructurar los componentes físicos necesarios. A continuación, se describen las partes principales que se deben considerar en la arquitectura física del Casnsat:

▪ Sistema electrónico y de control:

- *Módulo de sensores:* Conjunto de sensores para medir parámetros como temperatura, presión, aceleraciones y posición GPS.
 - Medir temperatura.
 - Medir humedad.
 - Medir presión y altitud.
 - Medir aceleración.
 - Medir inclinación.
 - Medir posición global.
- *Unidad de almacenamiento:* Guarda los datos recolectados en un formato accesible (CSV).
- *Fuente de energía:* Batería que suministrará la energía necesaria para el funcionamiento de todos los sistemas durante un mínimo de 4 horas continuas.

▪ Sistema de Comunicación:

- *Transmisor inalámbrico:* Envía datos recolectados por el CanSat a una estación base en tiempo real.
- *Antena:* Mejora la señal para la transmisión de datos.

▪ Sistema de seguridad y Estructura:

- *Carcasa protectora:* de forma cilíndrica, resguarda los componentes internos del CanSat durante el lanzamiento y el aterrizaje.
- *Conexiones modulares:* Facilitan el ensamblaje y desensamblaje para pruebas y ajustes.
- *Switch de encendido:* Habilita el sistema electrónico completo de medición del Cansat.

En la Figura 3.4 se pueden observar los componentes de la arquitectura física del Cansat y sus conexiones entre ellos.

3.4.4. Matriz morfológica

La matriz morfológica permite explorar y analizar múltiples alternativas de diseño para las funciones clave del CanSat. Cada función principal, como la medición de parámetros, almacenamiento, comunicación y estructura, cuenta con diferentes soluciones técnicas. En la Tabla 3.4 se puede observar las alternativas planteadas.

Combinaciones Propuestas para el Diseño del CanSat

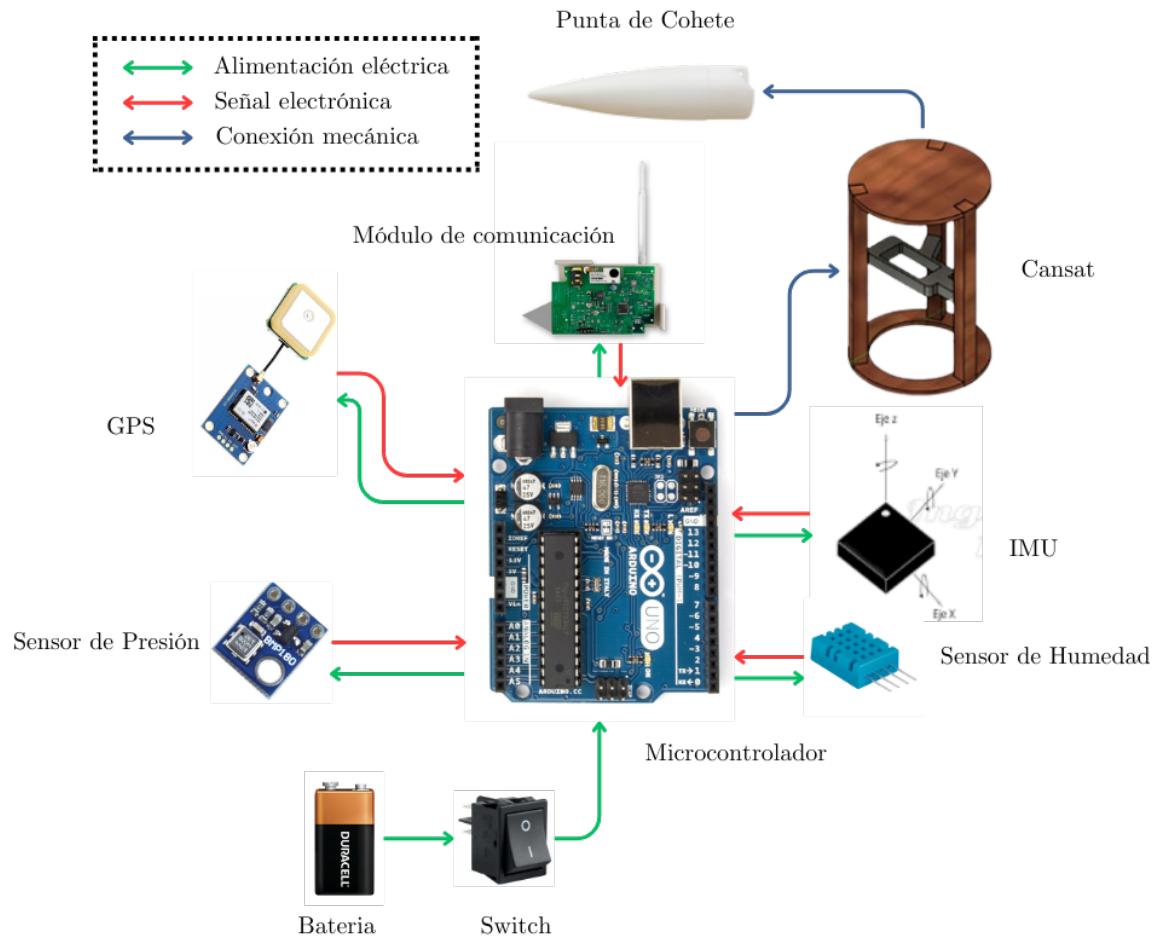


Figura 3.4: Arquitectura física del Cansat.

A continuación, se presentan tres combinaciones seleccionadas de la matriz morfológica final, adaptadas a diferentes objetivos de diseño:

■ **Combinación 1: Diseño Básico y Económico**

- **Medir temperatura:** Sensor digital (AHT20).
- **Medir humedad:** Sensor digital (AHT20).
- **Medir presión y altitud:** Barómetro digital (BMP280).
- **Medir aceleración:** Acelerómetro de 3 ejes (MPU6050).
- **Medir inclinación:** Giroscopio de 3 ejes (MPU6050).
- **Medir posición global:** Módulo GPS (NEO-6M v2).
- **Unidad de almacenamiento:** Tarjeta SD.
- **Fuente de energía:** Batería de iones de litio (Li-ion).
- **Transmisor inalámbrico:** Módulo LoRa.
- **Antena:** Antena monopolo.
- **Carcasa protectora:** Carcasa de PLA.
- **Conexiones modulares:** Conexiones de tornillo.
- **Switch de encendido:** Interruptor basculante.

Tabla 3.4: Matriz Morfológica para el Diseño del CanSat

Función	Solución 1	Solución 2	Solución 3
Medir temperatura	Sensor digital (AHT20)	Termistor	Sensor integrado (IMU)
Medir humedad	Sensor digital (AHT20)	Sensor capacitivo	Sensor integrado (IMU)
Medir presión y altitud	Barómetro digital (BMP280)	Barómetro MEMS	Barómetro integrado en IMU
Medir aceleración	Acelerómetro de 3 ejes (MPU6050)	Acelerómetro analógico	Acelerómetro integrado (IMU)
Medir inclinación	Giroscopio de 3 ejes (MPU6050)	Giroscopio analógico	Sensor integrado (IMU)
Medir posición global	Módulo GPS (NEO-6M v2)	Módulo GNSS (u-blox ZED-F9P)	GPS integrado con LoRa
Unidad de almacenamiento	Tarjeta SD	Memoria flash interna	Memoria EEPROM externa
Fuente de energía	Batería de polímero de litio (LiPo)	Batería de iones de litio (Li-ion)	Batería alcalina recargable
Transmisor inalámbrico	Módulo LoRa	Módulo Zigbee	Módulo WiFi (ESP8266)
Antena	Antena monopolo	Antena dipolo	Antena helicoidal
Carcasa protectora	Carcasa de ABS	Carcasa de PLA	Carcasa de fibra de carbono
Conexiones modulares	Conexiones de tornillo	Conexiones magnéticas	Conexiones a presión
Switch de encendido	Interruptor deslizante	Interruptor de botón	Interruptor basculante

■ Combinación 2: Diseño Equilibrado

- **Medir temperatura:** Termistor.
- **Medir humedad:** Sensor capacitivo.
- **Medir presión y altitud:** Barómetro MEMS.
- **Medir aceleración:** Acelerómetro analógico.
- **Medir inclinación:** Giroscopio analógico.
- **Medir posición global:** Módulo GNSS (u-blox ZED-F9P).
- **Unidad de almacenamiento:** Memoria flash interna.
- **Fuente de energía:** Batería de polímero de litio (LiPo).
- **Transmisor inalámbrico:** Módulo Zigbee.
- **Antena:** Antena dipolo.
- **Carcasa protectora:** Carcasa de ABS.
- **Conexiones modulares:** Conexiones magnéticas.
- **Switch de encendido:** Interruptor de botón.

■ Combinación 3: Diseño Avanzado y Resistente

- **Medir temperatura:** Sensor integrado (IMU).
- **Medir humedad:** Sensor integrado (IMU).

- **Medir presión y altitud:** Barómetro integrado en IMU.
- **Medir aceleración:** Acelerómetro integrado (IMU).
- **Medir inclinación:** Sensor integrado (IMU).
- **Medir posición global:** GPS integrado con LoRa.
- **Unidad de almacenamiento:** Memoria EEPROM externa.
- **Fuente de energía:** Batería alcalina recargable.
- **Transmisor inalámbrico:** Módulo WiFi (ESP8266).
- **Antena:** Antena helicoidal.
- **Carcasa protectora:** Carcasa de fibra de carbono.
- **Conexiones modulares:** Conexiones a presión.
- **Switch de encendido:** Interruptor deslizante.

3.4.4.1 Matriz de selección

Tabla 3.5: Matriz de Selección para el Diseño del CanSat

Criterios (Características de Ingeniería)	Peso (%)	C-1	C-2	C-3
Precisión de sensores	25	9	8	7
Eficiencia energética	20	9	8	7
Facilidad de ensamblaje	20	9	8	7
Confiabilidad de almacenamiento	15	9	8	7
Durabilidad de la carcasa	10	8	9	9
Calidad de comunicación	10	9	8	8
Puntaje Total	100	8.95	8.25	7.65

3.5. Diseño Detalle

El diseño de detalle se centra en la especificación precisa de los componentes seleccionados para el cohete y el CanSat, asegurando su integración eficiente. En esta etapa se definen dimensiones, materiales y conexiones, optimizando características clave como la disposición interna del CanSat y la fijación de los subsistemas del cohete, incluyendo el sistema de recuperación y la carga útil. Además, se seleccionan los materiales finales considerando resistencia, peso y compatibilidad, garantizando que el sistema completo cumpla con los requerimientos técnicos y pueda ser fabricado de manera eficiente Components, n.d.

3.5.1. Cohete

3.5.1.1 Cono

La nariz o cono del cohete es un componente crucial debido a su función como la punta del cohete, ya que su diseño impacta directamente en el desempeño aerodinámico. Es fundamental que tenga una geometría que minimice la resistencia al aire, conocida como "drag", para permitir un desplazamiento eficiente durante el vuelo. Existen diversas geometrías que pueden emplearse para fabricar el cono, ordenadas de menor a mayor resistencia aerodinámica:

1. Parabólico: Junto con las elípticas son una de las mejores geometrías ya que reducen significativamente la resistencia del aire incluso a diferentes velocidades.
2. Elíptico: Muy similar en su geometría a las parabólicas reduciendo considerablemente la resistencia del aire.
3. Ojiva tangente: Es un diseño muy utilizado por ejemplo en la geometría de cohetes o proyectiles como balas debido a su fácil construcción y buena resistencia aerodinámica.
4. Cónico: Quizás en contra de lo que uno pueda pensar, un cono es en realidad una mala geometría ya que tiene muy alta resistencia aerodinámica.

Para la construcción de nuestro cono se optó por una geometría tipo parabólica/elíptica, esta es una geometría más difícil de elaborar sin embargo su baja resistencia la hace la mejor geometría para el vuelo. Tal es el caso que la mayoría de aviones comerciales y cohetes usan esta geometría debido al ahorro de combustible Team, n.d.[b].



Figura 3.5: Vista isométrica del cono.

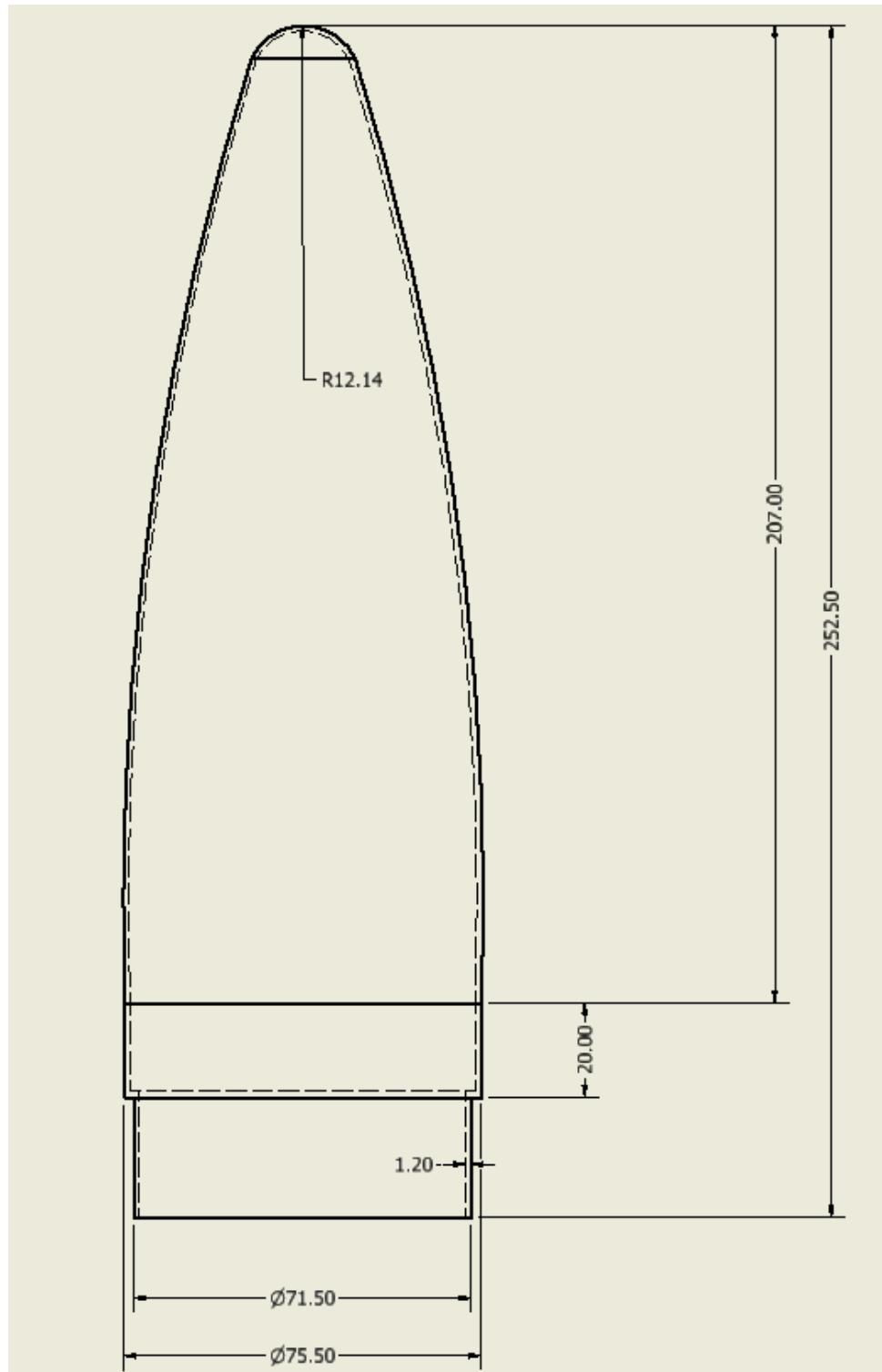


Figura 3.6: Plano del cono.

3.5.1.2 Estabilizadores

Los estabilizadores del cohete son elementos esenciales para mantener su estabilidad y trayectoria durante el vuelo. Estos componentes contribuyen a corregir desviaciones, resistir fuerzas externas y garantizar que el cohete mantenga su curso hacia el objetivo. La geometría de los estabilizadores juega un papel crítico, ya que influye directamente en el balance entre la resistencia al aire (drag) y la fuerza de recuperación generada Team, n.d.[a].

Existen diferentes geometrías de estabilizadores que pueden ser utilizadas, dependiendo del diseño del cohete y sus objetivos específicos. Entre las más comunes:

- Trapecial: Combinan baja resistencia al aire con un área mayor, lo que incrementa la fuerza de recuperación aerodinámica. Son ideales para cohetes que requieren estabilidad sin comprometer el rendimiento en vuelo.
- Rectangular: Fáciles de fabricar y con buen desempeño a bajas velocidades, aunque generan mayor resistencia en comparación con otras geometrías más refinadas.
- Elíptico: Ofrecen la menor resistencia aerodinámica gracias a su forma suavizada, pero son más complejos de construir.
- Delta: Es una figura aerodinámica pero tiene menor impacto a la estabilidad del cohete.

En el diseño de nuestro cohete, se eligieron estabilizadores trapezoidales debido a su balance óptimo entre baja resistencia al aire y una mayor superficie útil para generar la fuerza de recuperación necesaria durante el vuelo. Aunque su fabricación puede ser ligeramente más compleja que la de estabilizadores rectangulares, esta geometría permite un desempeño superior, proporcionando mayor estabilidad y eficiencia aerodinámica. Además, este diseño es ampliamente utilizado en cohetes de alto rendimiento debido a su versatilidad y su capacidad para adaptarse a diferentes velocidades y condiciones de vuelo.

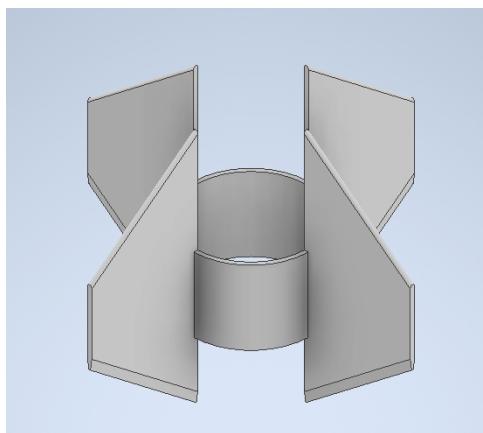


Figura 3.7: Vista isométrica de los estabilizadores.

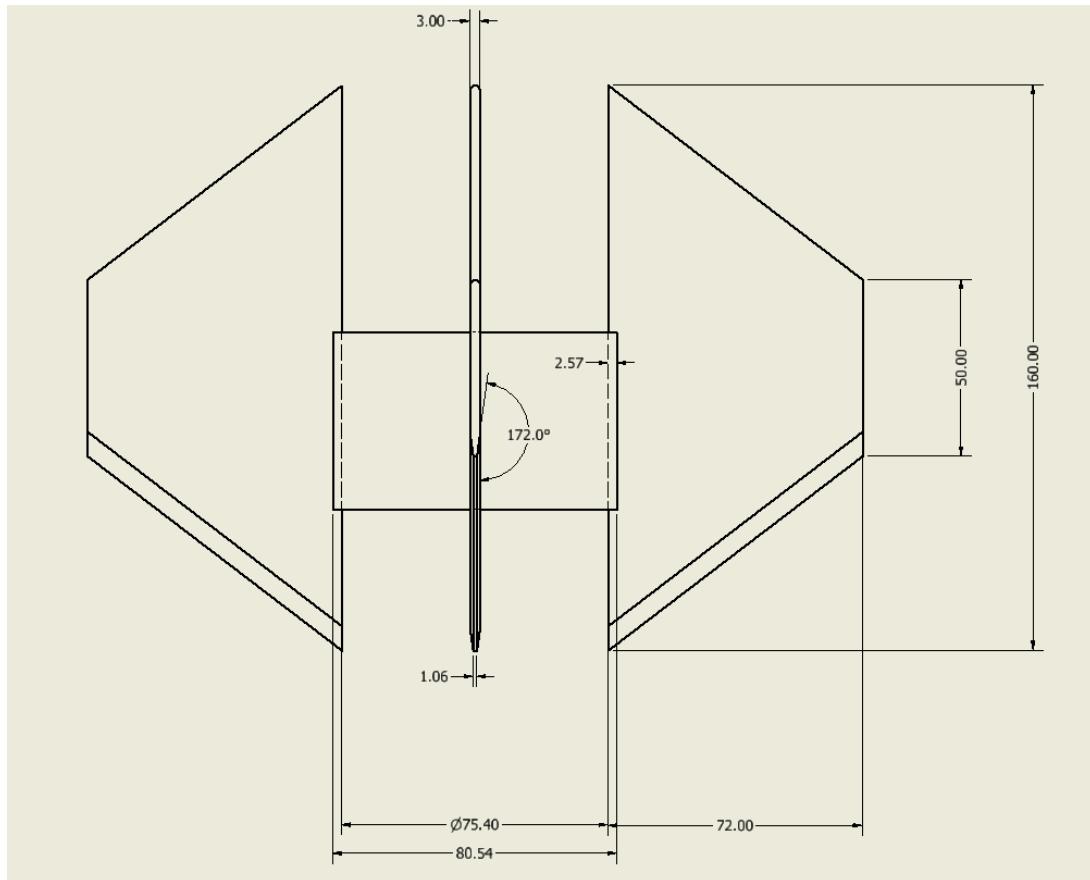


Figura 3.8: Plano de los estabilizadores.

3.5.1.3 Estructura

La estructura del cohete se diseñó a partir de un tubo de cartón con un diámetro interior de 70 mm, seleccionado por su ligereza, resistencia a la compresión y facilidad de manipulación. Las dimensiones del tubo se determinaron en función del espacio necesario para alojar el motor, el CanSat y el sistema de recuperación, asegurando un margen adecuado para la estabilidad estructural. Se reforzó con anillos internos de soporte y una capa de vinilo para mejorar su resistencia a la humedad y las cargas aerodinámicas Ponder, 2013.

3.5.1.4 Cordón de choque

El cordón de choque de un cohete es un componente del sistema para la recuperación del mismo, el cordón es diseñado para absorber y disminuir la energía generada después de la explosión del propulsor y previo a la apertura del paracaídas. Esto hace que los daños en la estructura del cohete sea la menor posible.

En nuestro caso y en nuestro prototipo, nuestro cordón fue fabricado con poliestireno, con el fin que al elongarse, aseguremos reducir el impacto brusco de la desaceleración, protegiendo en su totalidad tanto al cohete como al paracaídas, esto con el fin de asegurar una transición suave durante la fase de recuperación y aumentando la probabilidad de que el cohete aterrice de forma segura y sin daños significativos.

Características

3.5.1.5 Paracaídas

El sistema de recuperación, que incluye el paracaídas, es un componente esencial del cohete, diseñado para desplegarse en el apogeo o en los instantes cercanos a este. Su función principal es garantizar la recuperación íntegra del cohete, incluyendo todos sus componentes: el CanSat, el cono, el cuerpo principal, los estabilizadores y el motor.

Existen dos sistemas de recuperación comúnmente utilizados: el paracaídas y el sistema de hélices. Aunque el uso de hélices puede resultar más innovador e interesante, el paracaídas se destaca por su simplicidad en diseño, facilidad de construcción y la posibilidad de calcular con precisión sus parámetros para un descenso controlado.

A continuación, se presenta una descripción detallada de este sistema:

Sabiendo que un octágono tiene 8 triángulos isósceles con ángulos externos de 45° cada uno. Es muy sencillo fabricar un paracaídas octagonal.

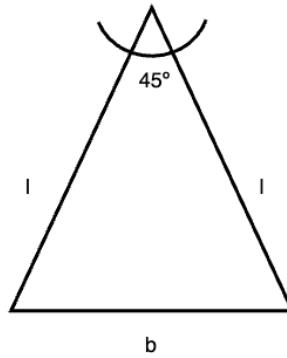


Figura 3.9: Triángulo isósceles.

$$l = r_{octagono} = 0,715m$$

$$b = 2L \sin \frac{\theta}{2}$$

$$b = (2)(0,715m) \sin \frac{45}{2}$$

Ecuaciones para la fabricación del paracaídas:

$$F_D = \frac{1}{2} \rho C_d A v^2 \quad (3.1)$$

$$F_D = mg \quad (3.2)$$

$$A = \frac{2mg}{\rho C_D v^2} \quad (3.3)$$

$$D_{octagono} = 2 \sqrt{\frac{A}{2(1 + \sqrt{2})}} \quad (3.4)$$

Dónde:

- ρ : Densidad del aire en Antioquia = $1,1Kg/m^3$.
- C_D : Coeficiente de arrastre del paracaídas $\approx 1,5$.
- A : Área proyectada de paracaídas.
- v : Velocidad de descenso = $2,2m/s$.
- F_D : Fuerza de resistencia aerodinámica.
- m : Masa aproximada del paracaídas = $1Kg$.
- g : Gravedad = $9,81m/s^2$.

Sustituyendo y despejando ecuaciones:

$$\begin{aligned} \text{Si: } mg &= \frac{1}{2} \rho C_d A v^2 \\ \dots \quad A &= \frac{2mg}{\rho C_D v^2} \\ \dots \quad A &= \frac{(2)(1Kg)(9,81m/s^2)}{(1,1Kg/m^3)(1,5)(2,2m/s)^2} \\ \therefore \quad A &= 2,4568m^2 \end{aligned}$$

$$\begin{aligned} \text{Si: } D_{octagono} &= 2 \sqrt{\frac{A}{2(1 + \sqrt{2})}} \\ \dots \quad D_{octagono} &= 2 \sqrt{\frac{2,4568m^2}{2(1 + \sqrt{2})}} \\ \therefore \quad D_{octagono} &= 1,43m \end{aligned}$$

3.5.1.6 Cálculo del Centro de Presión

La estabilidad de un cohete depende crucialmente de la ubicación relativa de su centro de masa (CM) y su centro de presión (CP). El CM es el punto donde se puede considerar concentrada toda la masa del cohete, mientras que el CP es el punto donde se concentran las fuerzas aerodinámicas. Para una estabilidad pasiva, el CM debe ubicarse por delante del CP. De esta manera, las fuerzas aerodinámicas sobre las aletas estabilizadoras generan un momento que corrige cualquier desviación de la trayectoria. La posición de ambos centros se puede ajustar modificando la distribución de masa y la geometría del cohete (cono, cuerpo y aletas). Generalmente, es más

fácil ajustar el CM. Para ello, es fundamental conocer la posición estimada del CP, la cual puede determinarse mediante: ensayos en túnel de viento, prueba del péndulo (o prueba del cordón) y proyección 2D del cohete.

Estos métodos experimentales proporcionan una estimación útil. También se puede calcular el CP utilizando la siguiente ecuación:

$$CP = \frac{d_n a_n + d_b a_b + d_f a_f}{A} \quad (3.5)$$

donde CP representa la posición del centro de presión, A es el área total de la superficie del cohete, y d_i y a_i son la distancia al extremo inferior del cohete y el área de cada componente, respectivamente: cono (n), cuerpo cilíndrico (b) y estabilizadores (f).

A continuación, se detalla el cálculo del CP para nuestro cohete:

Cálculo de Áreas:

Cono:

$$\begin{aligned} a_n &= \pi r \sqrt{r^2 + h^2} \\ \dots \quad a_n &= \pi(3,69\text{cm}) \sqrt{(3,69\text{cm})^2 + (22\text{cm})^2} \\ \therefore \quad a_n &\approx 258,59\text{cm}^2 \end{aligned}$$

Cuerpo Cilíndrico:

$$\begin{aligned} a_b &= 2\pi r h \\ \dots \quad a_b &= 2\pi(3,79\text{cm})(89,7\text{cm}) \\ \therefore \quad a_b &\approx 2136,05\text{cm}^2 \end{aligned}$$

Estabilizadores (4 estabilizadores):

$$\begin{aligned} a_f &= \frac{(b_1 + b_2)h}{2} \\ \dots \quad a_f &= \frac{(5\text{cm} + 16\text{cm})(7\text{cm})}{2} \\ \dots \quad a_f &= 73,5\text{cm}^2 \\ \therefore \quad 4a_f &= 294\text{cm}^2 \end{aligned}$$

Área Total:

$$\begin{aligned} A &= a_n + a_b + 4a_f \\ \dots \quad A &\approx 258,59\text{cm}^2 + 2136,05\text{cm}^2 + 294\text{cm}^2 \\ \therefore \quad A &\approx 2688,64\text{cm}^2 \end{aligned}$$

Distancias (desde la base del cohete):

$$\begin{aligned} d_n &= 90\text{cm} \\ d_b &= 41,5\text{cm} \\ d_f &= 8\text{cm} \end{aligned}$$

3. DISEÑO Y ANÁLISIS

Cálculo del Centro de Presión:

$$CP = \frac{d_n a_n + d_b a_b + d_f (4a_f)}{A}$$

$$\dots CP \approx \frac{(90\text{cm})(258,59\text{cm}^2) + (41,5\text{cm})(2136,05\text{cm}^2) + (8\text{cm})(294\text{cm}^2)}{2688,64\text{cm}^2}$$

$$\therefore CP \approx 42,5\text{cm}$$

3.5.1.7 Simulaciones en Open Rocket

Una vez finalizada la selección y diseño de los componentes se procede a realizar la simulación del vuelo y estabilidad del cohete diseñado. Es importante resaltar que a partir de las geometrías escogidas para el cono, estabilizadores y demás componentes, se realiza una serie de iteraciones en el software Open Rocket que permite establecer unas geometría dentro de los límites de diseño y capacidades de manufactura del equipo.

Los parámetros geométricos y de partes de la simulación se puede obserbar en la Tabla 3.6

Componente	Material	Dimensiones	Peso
Cono	PLA - 100 % infill	Elíptica, Len: 22 cm	105 g
Cansat	—	Diámetro ext: 6.6 cm, Len: 11.5 cm	250 g
Altímetro	—	Diámetro ext: 1.8 cm, Len: 5.5 cm	12 g
Sección Cansat	PLA - 100 % infill	Diámetro int: 6.98 cm, ext: 7.38 cm, Len: 12.7 cm	71.6 g
Acople	PLA - 100 % infill	Diámetro int: 6.98 cm, ext: 7.38 cm, Len: 2 cm	11.3 g
Cuerpo principal	Cartón (0.68 g/cm ³)	Diámetro int: 7.14 cm, ext: 7.38 cm, Len: 61 cm	140 g
Estabilizadores (4)	PLA - 100 % infill	Espesor: 0.3 cm	110 g
Cordón de choque	Cordón elástico	Longitud: 50 cm	0.9 g
Paracaídas	Tela PVC	Diámetro ext: 140 cm, Longitud: 16.2 cm	53.9 g
Shroud Lines	Hilo cáñamo	Líneas: 8, Longitud: 100 cm	—
Anillo de soporte (1)	PLA - 100 % infill	Diámetro int: 2.87 cm, ext: 7.5 cm, Len: 1.6 cm	75.4 g
Anillo soporte (2)	PLA - 100 % infill	Diámetro int: 2.87 cm, ext: 7.5 cm, Len: 1.6 cm	75.4 g
Botón de riel (1)	PLA - 100 % infill	Longitud: 0 cm	1.56 g
Botón de riel (2)	PLA - 100 % infill	Longitud: 0 cm	1.56 g

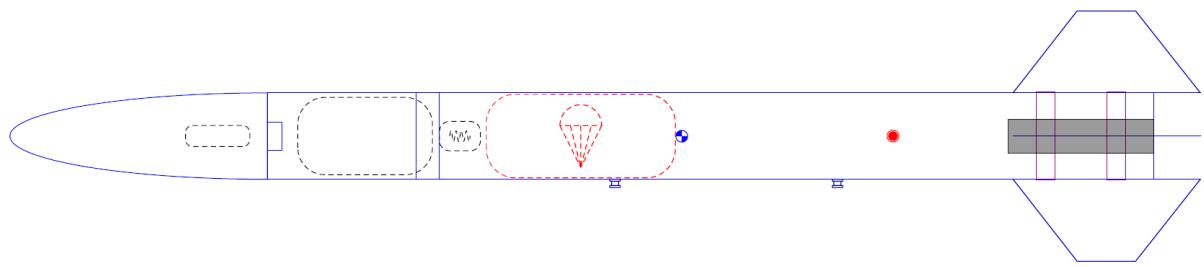
Tabla 3.6: Detalles de las partes del cohete.

Las condiciones de simulación para el lanzamiento se pueden obserbar en la Tabla 3.7.

Parámetro	Valor
Viento	
Velocidad media del viento	0.5 m/s
Desviación estándar	0.05 m/s
Intensidad de la turbulencia	10 % (Medio)
Dirección del viento	90°
Lugar de lanzamiento	
Latitud	6° N
Longitud	75° E
Altitud	2150 m

Tabla 3.7: Condiciones de simulación para el lanzamiento.

Se procede a realizar la simulación del vuelo cuya geometría puede ser observada en la Figura 3.10. Se obtienen los resultados del perfil del vuelo en la Figura 3.11, de estabilidad hasta el apogeo en la Figura 3.12 y el desplazamiento lateral del cohete en la Figura 3.13.



CoheteXoxo

Stages: 1

Mass (with motor): 1010 g

Stability: 2.44 cal / 17.7 %

CG: 57.4 cm

CP: 75.4 cm

Figura 3.10: Geometría de simulación Open Rocket.

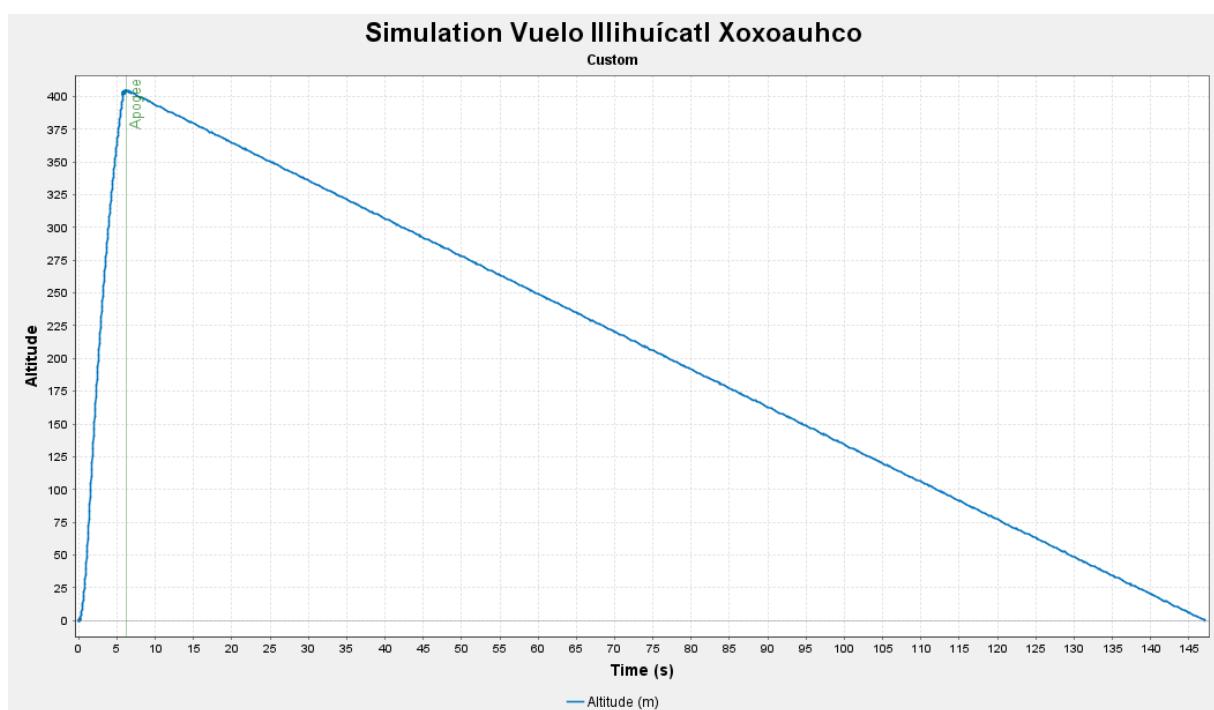


Figura 3.11: Perfil de vuelo simulado en Open Rocket.

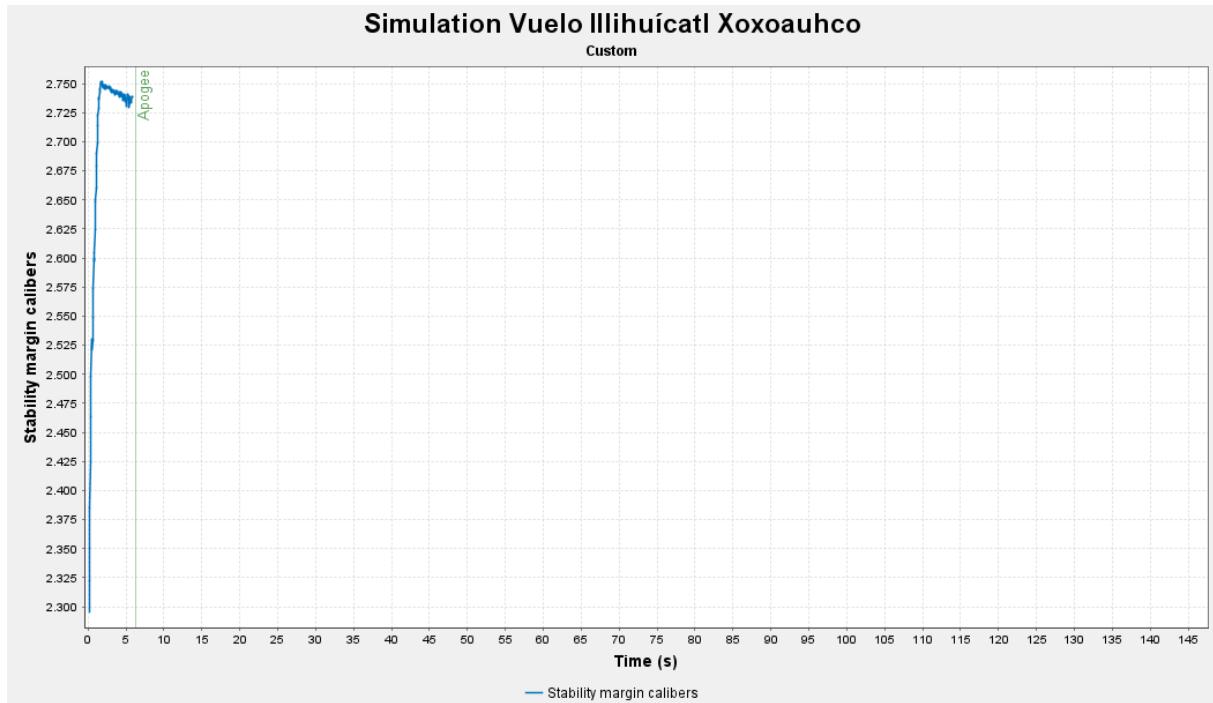


Figura 3.12: Estabilidad vs Tiempo.

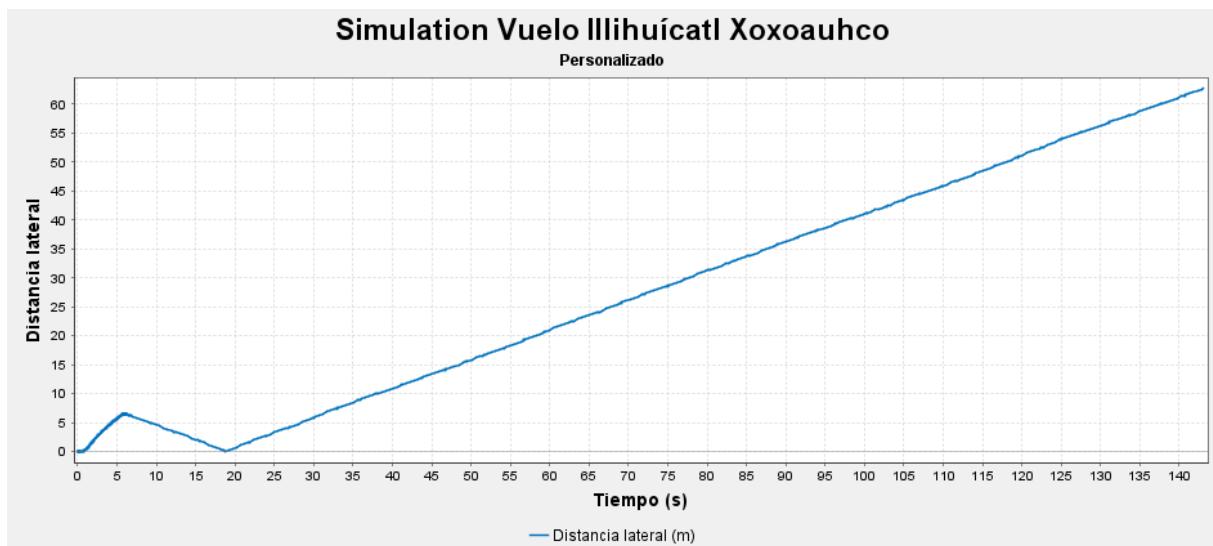


Figura 3.13: Estabilidad vs Tiempo.

Dentro de los resultados principales encontrados se pudo establecer que la altura máxima simulada es de 404 m. Uno de los elementos importante fue determinar que el margen de estabilidad siempre se mantuviera mayor a 2, logrado un margen de estabilidad mínima de 2,32 y máxima de 2,75. La distancia de caída del cohete es de 65 m. El tiempo total de vuelo es de 143 s.

3.5.1.8 Diseño en software CAD

Una vez finalizada la etapa de simulaciones con las geometrías seleccionadas se procede a realizar el CAD final del cohete que nos permite verificar el correcto ensamblaje de los componentes y el

correcto ensamble de los mismos. En la Figura 3.14 se puede observar vistas del cohete diseñado con sus componentes internos y la vista de explosionado.

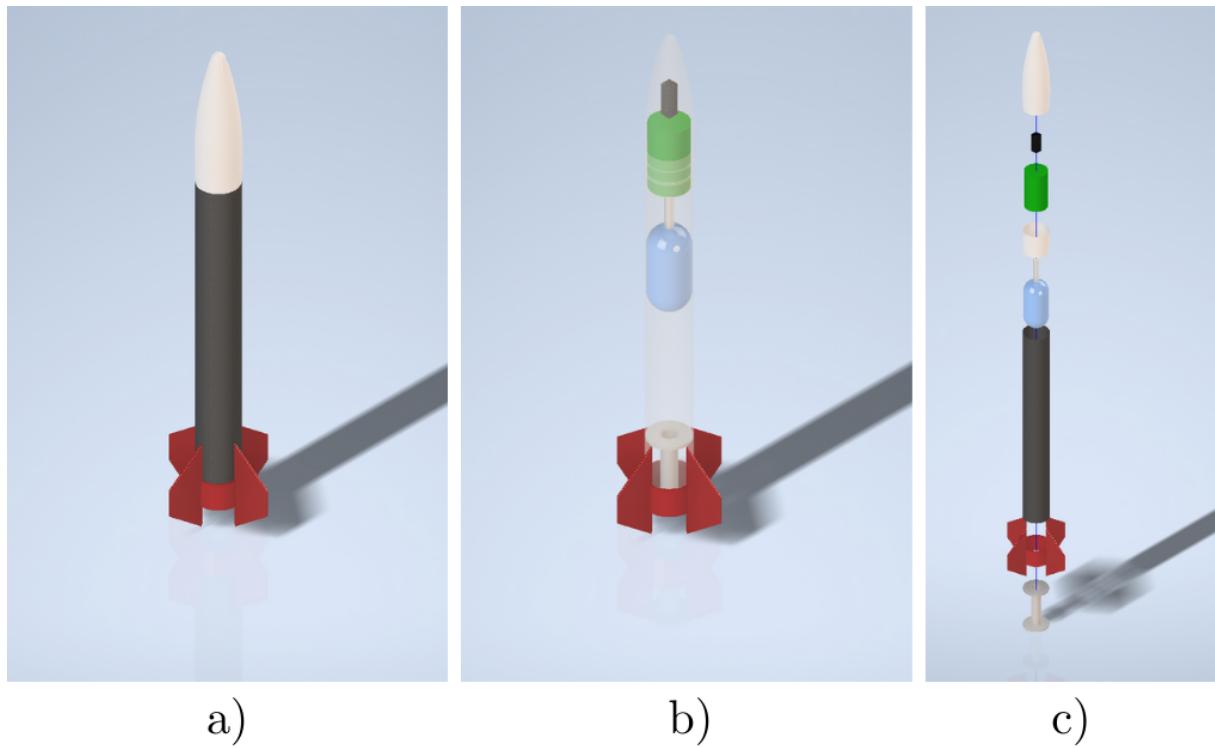


Figura 3.14: a) Vista del modelo del cohete ensamblado. b) Vista de los componentes internos del cohete. c) Vista explosionada de los componentes finales del cohete.

3.5.2. Cansat

3.5.2.1 Estructura

- **Material:** PLA estándar, seleccionado por su relación resistencia-peso, su económico precio, su disponibilidad y versatilidad para prototipado.
- **Dimensiones:** Diámetro de 66 mm y altura de 115 mm, cumpliendo con las restricciones del concurso y normativas de un CanSat.
- **Peso Total:** 250 gramos [CAMBIAR AL TENER PESO REAL], incluyendo todos los componentes internos para cumplir con las especificaciones establecidas por los organizadores del concurso.

El CanSat fue diseñado en Inventor tomando como base un prototipo previamente elaborado. El diseño incluye la separación de dos placas de prototipado, sobre las cuales posteriormente se montarán los circuitos. Además, se añadió un espacio destinado para un interruptor y una pantalla segmentada, que permitirá indicar el voltaje de la batería. Una representación en CAD del Cansat puede ser observada en la Figura 3.15.



Figura 3.15: Modelo del CanSat en inventor

3.5.2.2 Sistema de Despliegue

El CanSat estará alojado en la punta del cohete, diseñada para ser eyectada automáticamente al término de la fase de ascenso. Una vez que el motor del cohete deja de proporcionar impulso, genera una pequeña explosión que fuerza la expulsión de la punta.

La punta del cohete permanecerá conectada al fuselaje mediante un cordón elástico, asegurando que no se pierda durante el descenso. Este mecanismo garantiza que el CanSat sea liberado de manera segura para iniciar su descenso controlado, activando el sistema de paracaídas una vez que queda libre de la estructura del cohete sin que el CanSat se separe del cohete.

3.5.2.3 Sistema Electrónico

Sistema de Medición

- **Sensor de Temperatura y Humedad:** AHT20, rango de temperatura -40°C a 85°C, con una precisión de $\pm 0.3^\circ\text{C}$; rango de humedad 0 % a 100 % con una precisión de $\pm 0.2\%$ RH.
- **Sensor de Presión y Altitud:** BMP280, rango de 300 a 1100 hPa, precisión de ± 1 hPa. Este sensor se ha conseguido en el mismo módulo que el sensor anterior.
- **Acelerómetro y Giroscopio:** MPU6050, rango de ± 16 g y ± 2000 °/s, respectivamente.
- **Módulo GPS:** NEO6MV2, precisión horizontal de ± 2.5 m.

Microcontrolador

- **Microcontrolador:** ESP32 NodeMCU.
- **Puertos de Entrada/Salida:**
 - GPIO: Hasta 34 pines programables.
 - ADC: Resolución de 12 bits, útil para la lectura de sensores analógicos.
 - Interfaz I2C y SPI para comunicación con los sensores AHT20, BMP280, MPU6050 y el módulo Blueee LoRa.
 - Voltaje de Operación: 3.3-5 V, ajustado a 3.3 V el cual es compatible con todos los sensores y módulos utilizados.
- **Razones para su Selección:**
 - Procesamiento Rápido: Su potente procesador de doble núcleo permite manejar las operaciones de los sensores y el sistema de comunicación simultáneamente.
 - Capacidad de Conectividad: Aunque el LoRa es el principal sistema de comunicación, el Wi-Fi y Bluetooth integrados del ESP32 facilitan el diagnóstico, configuración y pruebas previas al lanzamiento.
 - Eficiencia Energética: Optimizado para operar a bajo consumo, prolongando la autonomía de la batería.
 - Compatibilidad: Es fácilmente integrable con los sensores y módulos utilizados en el CanSat, gracias a su variedad de interfaces de comunicación.
 - Tamaño Compacto: Su diseño ligero y pequeño lo hace ideal para proyectos donde el espacio es limitado.

Diagrama Electrónico

Se creó un diagrama electrónico en EasyEDA con el objetivo de representar detalladamente las conexiones entre los componentes principales del CanSat y describir su funcionamiento. Este diagrama incluye la disposición de los módulos esenciales, como el microcontrolador, el sistema de alimentación, los sensores y el módulo de transmisión de datos (ver Figura 3.16). Además, se especificaron las rutas de señal y alimentación para garantizar una comprensión clara del flujo eléctrico y de datos dentro del sistema. El diseño no solo sirve como guía para el ensamblaje, sino también como referencia para el diagnóstico y la solución de problemas durante las pruebas y ajustes del CanSat.

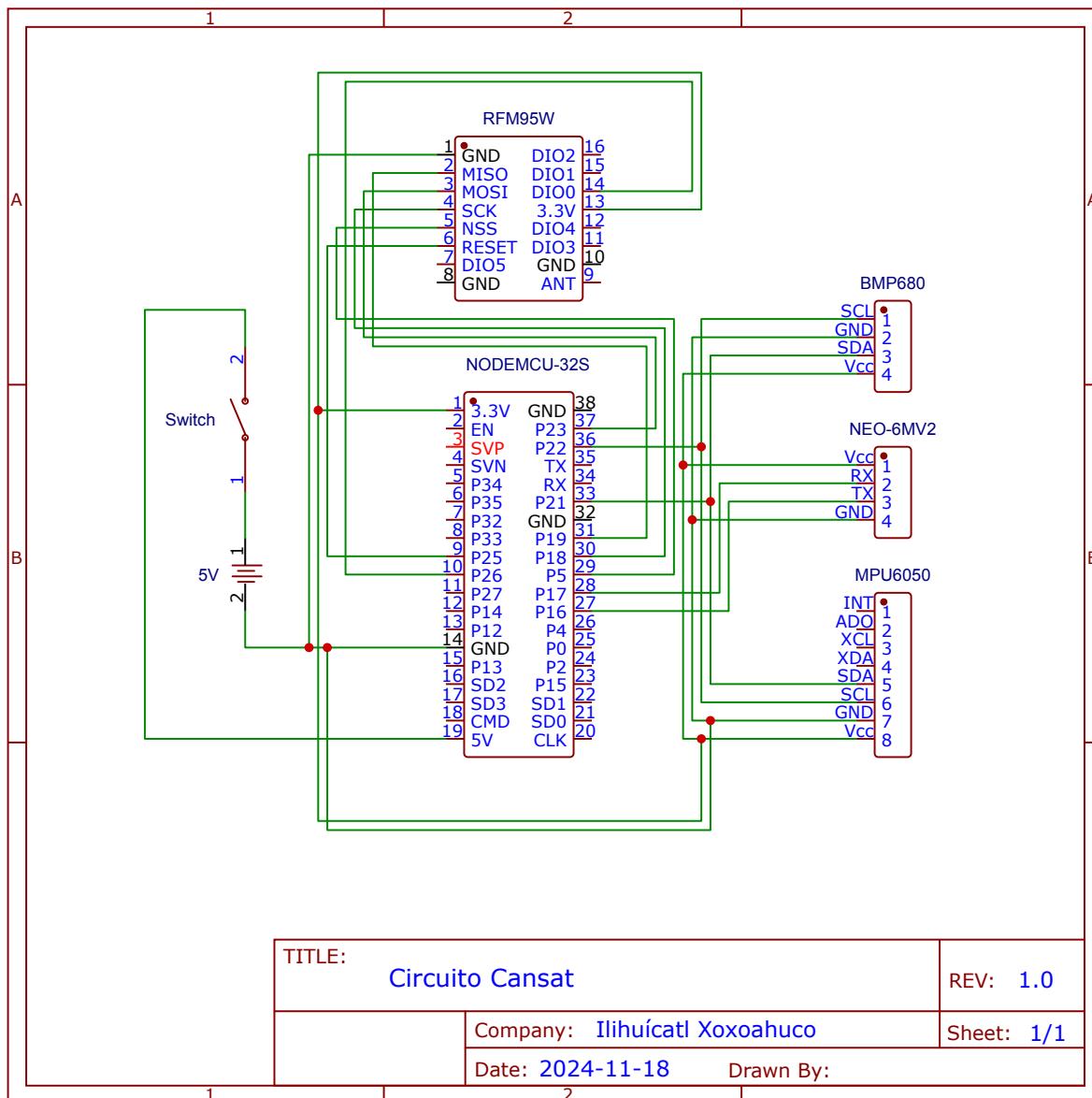


Figura 3.16: Circuito del sistema electrónico.

3.5.2.4 Sistema de Comunicación

- Transmisor RF:** El sistema de comunicación del CanSat utiliza el módulo **RFM95W**, un dispositivo compacto y eficiente basado en tecnología LoRa (Long Range). Este módulo opera en la banda ISM (433 MHz, 868 MHz o 915 MHz, dependiendo de la región) y es capaz de transmitir datos a largas distancias, alcanzando hasta **15 km** en línea de vista** en condiciones ideales y entre **1-2 km**** en entornos con interferencias moderadas.
- Antena:**
 - Tipo:** Antena omnidireccional.
 - Frecuencia de Operación:** Compatible con las bandas del módulo RFM95W, seleccionada según la configuración regional (915 MHz para América, 868 MHz para Europa, 433 MHz en otras regiones).

- **Ganancia:** Entre 2-3 dBi, optimizando la señal en todas las direcciones.
- **Montaje:** La antena está ubicada en la parte superior del CanSat para minimizar interferencias de otros componentes y maximizar la transmisión de datos.

▪ **Ventajas del RFM95W:**

- **Largo Alcance:** Permite transmisiones confiables en distancias de hasta 15 km en línea de vista.
- **Bajo Consumo Energético:** Ideal para aplicaciones como el CanSat donde la autonomía es crítica.
- **Flexibilidad de Configuración:** Operación en múltiples bandas ISM según las restricciones regionales.
- **Compatibilidad:** Fácil integración con microcontroladores como el ESP32 mediante interfaces SPI.

▪ **Razones de Selección del Bluee LoRa:**

- **Alcance Extendido:** Ofrece una cobertura significativamente mayor en comparación con tecnologías como Bluetooth o Wi-Fi, lo que es crucial en entornos abiertos.
- **Fiabilidad:** La tecnología LoRa garantiza una transmisión de datos robusta y estable, incluso en condiciones de interferencia.
- **Compatibilidad:** Fácil integración con microcontroladores y otros sensores del sistema.
- **Económico:** Comparado con otros módulos que brindan la misma distancia de comunicación es el de mejor precio disponible.

3.5.2.5 Fuente de Energía

- **Batería:** De iones de litio de 5V y 800 mAh, proporcionando una autonomía estimada de casi 9 horas continuas, considerando un consumo promedio de 85 mAh. Dicha información se fundamenta en la siguiente tabla:

Componente	Rango Mínimo Voltaje	Rango Máximo Voltaje	mA Hora
Neo 6MV2 GPS	3.3 VDC	5 VDC	45
BMP 280	2.8 VDC	5 VDC	0.58
Bluee LoRa (RFM95W)	1.8 VDC	3.7 VDC	10
AHT20	2.8 VDC	5 VDC	0.25
ESP32	3.3 VDC	5 VDC	30
Total			85.83

Tabla 3.8: Consumo energético de los componentes del CanSat

- **Regulador de Voltaje:** El ESP32 tiene un regulador de 5V a 3.3V, la salida de este es la utilizada para alimentar todos los componentes.

3.5.2.6 Códigos para la medición y comunicación

Para el desarrollo del proyecto se implementaron cuatro códigos que permiten la medición, transmisión e interpretación de los datos. En los códigos 3.1, 3.2 y 3.3 se encuentran el código

3. DISEÑO Y ANÁLISIS

programado en el ESP32 que permite la asquisición de los datos de los diferentes sensores digitales implementados y que permite la transmisión de los mismos por medio del protocolo LoRa.

El código 3.4 permite recibir los datos y los envía al puerto serial del computador cada 0,5 s.

El código compuesto por los códigos 3.5, 3.6, 3.7 y 3.5 se encarga de interpretar los datos del puerto serial y graficar las mediciones obtenidas en tiempo real.

Finalmente el código compuesto por los códigos 3.9, 3.10 y 3.11; toma los datos al final del vuelo y realiza unas gráficas con los resultados totales.

```

1 #include <LoRa.h>
2 #include <Wire.h>
3 #include <SPI.h>
4 #include <Adafruit_BMP280.h>
5
6 // Definir pines para el LoRa
7 #define ss 5
8 #define rst 25
9 #define dio0 26
10#define BMP_SCK (13)
11#define BMP_MISO (12)
12#define BMP_MOSI (11)
13#define BMP_CS (10)
14
15 // Direcciones I2C
16#define BMP280_ADDR 0x77
17#define MPU_ADDR 0x68
18
19 Adafruit_BMP280 bmp; // I2C
20
21 // Variables para el MPU6050
22 int16_t a_x, a_y, a_z;
23 int16_t g_x, g_y, g_z;
24 int16_t t;
25 int16_t ax_offset = 0, ay_offset = 0, az_offset = 0;
26 int16_t gx_offset = 0, gy_offset = 0, gz_offset = 0;
27 char tmp_str[7];
28
29 // Variables para el GPS
30#define RXD2 16
31#define TXD2 17
32#define GPS_BAUD 9600
33
34 HardwareSerial gpsSerial(2);
35 String nmeaSentence;
36
37 // Función para convertir a cadena
38 char* convert_int16_to_str(int16_t i) {
39     sprintf(tmp_str, "%6d", i);
40     return tmp_str;
41 }
42
43 void setup() {
44     // Configurar comunicación serial
45     Serial.begin(115200);
46     while (!Serial);
47     Serial.println("LoRA Sender");
48
49     // Configurar LoRa
50     LoRa.setPins(ss, rst, dio0);
51     while (!LoRa.begin(915E6)) {
52         Serial.println(".");
53         delay(200);
54     }
55
56     LoRa.setSyncWord(0xF3);
57     Serial.println("LoRa Initializing OK!");

```

Código 3.1: Código utilizado para la captura y envío de datos en el microcontrolador, parte 1.

3. DISEÑO Y ANÁLISIS

```
1 // Configurar el BMP280
2 Serial.println(F("BMP280 test"));
3 if (!bmp.begin(BMP280_ADDR)) {
4     Serial.println(F("Could not find a valid BMP280 sensor, check wiring or try
5         ↪ a different address!"));
6     while (1) delay(10);
7 }
8 bmp.setSampling(Adafruit_BMP280::MODE_NORMAL,
9                 Adafruit_BMP280::SAMPLING_X2,
10                Adafruit_BMP280::SAMPLING_X16,
11                Adafruit_BMP280::FILTER_X16,
12                Adafruit_BMP280::STANDBY_MS_500);
13 // Configurar el MPU6050
14 Wire.begin();
15 Wire.beginTransmission(MPU_ADDR);
16 Wire.write(0x6B); // Registro de encendido
17 Wire.write(0); // Despertar el sensor
18 Wire.endTransmission(true);
19 calibrateSensor(); // Calibrar los ejes al inicio
20 // Configurar el GPS
21 gpsSerial.begin(GPS_BAUD, SERIAL_8N1, RXD2, TXD2);
22 nmeaSentence.reserve(80); // Reservar memoria para la cadena NMEA
23 Serial.println("Serial and GPS initialized.");
24 }
25 void loop() {
26     // Leer datos del BMP280
27     String altura = String(bmp.readAltitude(1013.25));
28     String temperatura = String(bmp.readTemperature());
29     String presion = String(bmp.readPressure());
30     // Leer datos del MPU6050
31     Wire.beginTransmission(MPU_ADDR);
32     Wire.write(0x3B);
33     Wire.endTransmission(false);
34     Wire.requestFrom(MPU_ADDR, 7 * 2, true);
35     a_x = (Wire.read() << 8 | Wire.read()) - ax_offset;
36     a_y = (Wire.read() << 8 | Wire.read()) - ay_offset;
37     a_z = (Wire.read() << 8 | Wire.read()) - az_offset;
38     t = Wire.read() << 8 | Wire.read();
39     g_x = (Wire.read() << 8 | Wire.read()) - gx_offset;
40     g_y = (Wire.read() << 8 | Wire.read()) - gy_offset;
41     g_z = (Wire.read() << 8 | Wire.read()) - gz_offset;
42     // Leer datos del GPS
43     String gpsData = "";
44     while (gpsSerial.available() > 0) {
45         char c = gpsSerial.read();
46         if (c == '$') {
47             nmeaSentence = ""; // Limpiar la cadena para la siguiente sentencia
48         }
49         if (c != '\n' && c != '\r') { // Si no es fin de línea
50             nmeaSentence += c; // Añadir el carácter a la cadena
51         }
52         if (c == '\n') { // Nueva línea indica fin de la sentencia
53             if (nmeaSentence.startsWith("$GPGGA")) { // Solo procesar si es GPRMC
54                 gpsData = nmeaSentence; // Guardar los datos del GPS
55             }
56         }
57     }
}
```

Código 3.2: Código utilizado para la captura y envío de datos en el microcontrolador, parte 2.

```

1 // Preparar el paquete de datos
2 String paquete = temperatura + "," + presion + "," + altura + ",";
3 paquete += String(a_x) + "," + String(a_y) + "," + String(a_z) + ",";
4 paquete += String(g_x) + "," + String(g_y) + "," + String(g_z) + ","
    ↵ gpsData;
5
6 // Enviar el paquete por LoRa
7 LoRa.beginPacket();
8 LoRa.print(paquete);
9 LoRa.endPacket();
10 Serial.println("Enviando paquete: " + paquete);
11
12 delay(500);
13 }
14
15 // Función de calibración para el MPU6050
16 void calibrateSensor() {
17     const int numReadings = 1000;
18
19     long ax_sum = 0, ay_sum = 0, az_sum = 0;
20     long gx_sum = 0, gy_sum = 0, gz_sum = 0;
21
22     for (int i = 0; i < numReadings; i++) {
23         Wire.beginTransmission(MPU_ADDR);
24         Wire.write(0x3B);
25         Wire.endTransmission(false);
26         Wire.requestFrom(MPU_ADDR, 7 * 2, true);
27
28         ax_sum += Wire.read() << 8 | Wire.read();
29         ay_sum += Wire.read() << 8 | Wire.read();
30         az_sum += Wire.read() << 8 | Wire.read();
31         Wire.read() << 8 | Wire.read(); // Leer temperatura y descartarla
32         gx_sum += Wire.read() << 8 | Wire.read();
33         gy_sum += Wire.read() << 8 | Wire.read();
34         gz_sum += Wire.read() << 8 | Wire.read();
35
36         delay(10);
37     }
38
39     ax_offset = ax_sum / numReadings;
40     ay_offset = ay_sum / numReadings;
41     az_offset = az_sum / numReadings;
42     gx_offset = gx_sum / numReadings;
43     gy_offset = gy_sum / numReadings;
44     gz_offset = gz_sum / numReadings;
45 }

```

Código 3.3: Código utilizado para la captura y envío de datos en el microcontrolador, parte 3.

3. DISEÑO Y ANÁLISIS

```
1 #include <SPI.h>
2 #include <LoRa.h>
3
4 //Definir pines para el LoRa
5 #define ss 5
6 #define rst 21
7 #define dio0 25
8
9 int led = 2;
10 int counter = 0;
11
12 void setup() {
13     pinMode(led, OUTPUT);
14     //Configurar comunicación serial
15     Serial.begin(115200);
16     while(!Serial);
17     Serial.println("LoRa Reciever");
18
19     //Configurar LoRa
20     LoRa.setPins(ss, rst, dio0);
21     while(!LoRa.begin(915E6)) {
22         Serial.println(".");
23         delay(500);
24     }
25
26     LoRa.setSyncWord(0xF3);
27     Serial.println("LoRa Initializing OK!");
28     digitalWrite(led, HIGH);
29 }
30
31 void loop() {
32     int packetSize = LoRa.parsePacket();
33     if (packetSize) {
34         //Serial.print("Recieved packet ");
35
36         while (LoRa.available()) {
37             String LoRaData = LoRa.readString();
38             Serial.print(LoRaData);
39             Serial.print("\n");
40         }
41
42         //Serial.print(" with RSSI ");
43         //Serial.println(LoRa.packetRssi());
44     }
45 }
```

Código 3.4: Código utilizado para la recepción de datos y transmisión al computador.

Código 3.5: Código utilizado para el procesamiento de datos y generación de gráficas en tiempo real.

3. DISEÑO Y ANÁLISIS

```
1 # Esta función se encarga de convertir el formato GPGGA de latitud y longitud
2 # en un formato que se pueda leer para caropy, usa la coordenada GPGGA y su
3 # posición (N/S, E/W).
4 def nmea_to_decimal_degrees(nmea_str, direction):
5     degrees = float(nmea_str[:2]) # Extract degrees
6     minutes = float(nmea_str[2:]) # Extract minutes and decimal minutes
7     decimal_degrees = degrees + minutes / 60.0
8     if direction in ('S', 'W'): # Adjust for South/West
9         decimal_degrees *= -1
10    return decimal_degrees
11
12
13 # Es una función larga que en pocas palabras contiene todos los parámetros
14 # que se utilizan para hacer las gráficas.
15 def plot_data(df, fig, axes, map_ax):
16     # Plot the data
17     axes[(0, 0)].plot(df['Temperatura'], label='Temperatura', color='gold')
18     axes[(1, 0)].plot(df['Presión'], label='Presión', color='deepskyblue')
19     axes[(2, 0)].plot(df['Altitud'], label='Altitud', color='slategray')
20     axes[(2, 0)].plot(df['Altitude_GPS'], label='AltitudGPS', color='tomato')
21     axes[(2, 0)].legend()
22
23     axes[(0, 1)].plot(df['accelerationX'], label='Accel X', color='red')
24     axes[(0, 1)].plot(df['accelerationY'], label='Accel Y', color='green')
25     axes[(0, 1)].plot(df['accelerationZ'], label='Accel Z', color='blue')
26     axes[(0, 1)].set_ylabel('Aceleración [m/s^2]') # Combined label
27     axes[(0, 1)].legend()
28
29     axes[(1, 1)].plot(df['Xangle'], label='Roll (X)', color='lightcoral')
30     axes[(1, 1)].plot(df['Yangle'], label='Pitch (Y)', color='lightskyblue')
31     axes[(1, 1)].plot(df['Zangle'], label='Yaw (Z)', color='yellowgreen')
32     axes[(1, 1)].set_ylabel('Rotación [rad]')
33     axes[(1, 1)].legend()
34
35     # Add labels and titles
36     axes[(0, 0)].set_ylabel('Temperatura [°C]')
37     axes[(1, 0)].set_ylabel('Presión [hPa]')
38     axes[(2, 0)].set_ylabel('Altitud [m]')
39     axes[(0, 1)].set_ylabel('Aceleración [m/s^2]')
40     axes[(1, 1)].set_ylabel('Rotación [rad]')
41     # Add grid
42     for ax in axes.values():
43         ax.grid(True)
44
45 fig.suptitle('Sensor Data') # Overall title
46
47 #Plot GPS
48 map_ax.clear() #clear each frame
49 roads = cfeature.NaturalEarthFeature(
50     category='cultural',
51     name='roads',
52     scale='10m', # Adjust scale as needed (10m, 50m, 110m)
53     edgecolor='gray', # Road color
54     facecolor='none', # No fill
55     linewidth=0.5 # Adjust line width
56 )
57 map_ax.add_feature(roads)
58 map_ax.add_feature(cfeature.LAND)
59 map_ax.add_feature(cfeature.OCEAN)
    ↪ map_ax.add_feature(cfeature.COASTLINE)
    ↪ map_ax.add_feature(cfeature.BORDERS, linestyle=':')
```

Código 3.6: Código utilizado para el procesamiento de datos y generación de gráficas en tiempo real. Parte 2.

```

1    if not df.empty:
2        try:
3            lon = df.apply(lambda row:
4                nmea_to_decimal_degrees(str(row['Longitude']), row['E/W']), axis=1)
5            lat = df.apply(lambda row:
6                nmea_to_decimal_degrees(str(row['Latitude']), row['N/S']), axis=1)
7            map_ax.plot(lon, lat, 'ro-',
8                        transform=ccrs.Geodetic(),
9                        markersize=1)
10           map_ax.set_extent([lon.min() - 0.5, lon.max() + 0.5, lat.min() - 0.2,
11                               lat.max() + 0.2], crs=ccrs.PlateCarree())
12       except (KeyError, TypeError, ValueError) as e:
13           print(f"Error graficando GPS:
14             {RED}{e}{RESET}\t{lat.iloc[-1]},{lon.iloc[-1]})")
15
16       map_ax.set_title('Localización GPS')
17
18 # Es la función main, aquí se mandan a llamar las demás funciones y se crea y
19 # el df que se utiliza para ir guardando los datos, la función process_lora_data
20 # se manda a llamar multiples veces en el while y se actualizan las gráficas
21 # hasta que
22 # se detecta un KeyboardInterrupt es decir se usa ctrl+C para terminar el
23 # programa.
24 def main():
25     clear_terminal()
26     print('Ilihuícatl Xoxoauhco: Misión de Prueba: 2 [Prueba de altitud].')
27     print('Iniciando programa...')

28     # Serial connection
29     ser = connect_to_serial(SERIAL_PORT, BAUD_RATE)
30     if not ser:
31         return

32     # Data frame
33     df = pd.DataFrame(columns=[
34         'Temperatura', 'Presión', 'Altitud', 'accelerationX', 'accelerationY',
35         'accelerationZ',
36         'Xangle', 'Yangle', 'Zangle', 'GPGGA', 'Time', 'Latitude', 'N/S',
37         'Longitude', 'E/W', 'FixQuality',
38         'Satellites', 'HDOP', 'Altitude_GPS', 'M_Altitude', 'Geoid_Height',
39         'M_Geoid', 'DGPS_update', 'DGPS_station'])#, 'Checksum'

40     fig = plt.figure(figsize=(18, 8), facecolor="#e7e7e8")
41     gs = fig.add_gridspec(4, 2, height_ratios=[1, 1, 1, 2])

42     # Sensor plots
43     axes = {}
44     for i in range(3):
45         for j in range(2):
46             axes[(i, j)] = fig.add_subplot(gs[i, j])

```

Código 3.7: Código utilizado para el procesamiento de datos y generación de gráficas en tiempo real. Parte 3.

```
1  # Map
2  projection = ccrs.PlateCarree()
3  map_ax = fig.add_subplot(gs[3, :], projection=projection)
4  map_ax.stock_img()
5  map_ax.set_global()
6
7
8
9  plt.ion()
10 plt.show(block=False)
11
12 with open(CSV_FILE, 'w', newline=',') as csvfile:
13     writer = csv.writer(csvfile)
14     try:
15         while True:
16             df = process_lora_data(ser, writer, df)
17             for ax in axes.values():
18                 ax.clear()
19             map_ax.clear()
20             plot_data(df, fig, axes, map_ax)
21             fig.canvas.draw()
22             fig.canvas.flush_events()
23             plt.pause(0.1)
24             plt.tight_layout()
25     except KeyboardInterrupt:
26         print('\nTerminando ejecución.')
27     finally:
28         ser.close()
29         plt.show(block=True)
30
31
32 if __name__ == "__main__":
33     main()
```

Código 3.8: Código utilizado para el procesamiento de datos y generación de gráficas en tiempo real. Parte 4.

```

1 import pandas as pd
2 import cartopy.crs as ccrs
3 import matplotlib.pyplot as plt
4 import cartopy.feature as cfeature
5
6 # Function to convert NMEA coordinates to decimal degrees
7 def nmea_to_decimal_degrees(nmea_str, direction):
8     try:
9         degrees = float(nmea_str[:2])
10        minutes = float(nmea_str[2:])
11        decimal_degrees = degrees + minutes / 60.0
12        if direction in ('S', 'W'):
13            decimal_degrees *= -1
14        return decimal_degrees
15    except (ValueError, TypeError, IndexError):
16        return None # Handle cases where data is missing or malformed
17
18 # Function to plot the data
19 def plot_data(df, fig, axes, map_ax):
20     #Error Handling for missing columns
21     for col in ['Temperatura', 'Presión', 'Altitud', 'AccelX', 'AccelY',
22                 'AccelZ',
23                 'RotX', 'RotY', 'RotZ', 'Longitude', 'Latitude', 'N/S', 'E/W',
24                 'AGPS']:
25         if col not in df.columns:
26             print(f"Warning: Column '{col}' not found in CSV. Skipping related
27             plots.")
28             continue
29
30     # Plot the data (with error handling for missing columns)
31     axes[(0, 0)].plot(df['Temperatura'], label='Temperatura', color='gold')
32     axes[(1, 0)].plot(df['Presión'], label='Presión', color='deepskyblue')
33     axes[(2, 0)].plot(df['Altitud'], label='Altitud', color='slategray')
34     axes[(2, 0)].plot(df['AGPS'], label='AltitudGPS', color='tomato')
35     axes[(2, 0)].legend()
36
37     axes[(0, 1)].plot(df['AccelX'], label='Accel X', color='red')
38     axes[(0, 1)].plot(df['AccelY'], label='Accel Y', color='green')
39     axes[(0, 1)].plot(df['AccelZ'], label='Accel Z', color='blue')
40     axes[(0, 1)].set_ylabel('Acceleration [m/s^2]')
41     axes[(0, 1)].legend()
42
43     axes[(1, 1)].plot(df['RotX'], label='Roll (X)', color='lightcoral')
44     axes[(1, 1)].plot(df['RotY'], label='Pitch (Y)', color='lightskyblue')
45     axes[(1, 1)].plot(df['RotZ'], label='Yaw (Z)', color='yellowgreen')
46     axes[(1, 1)].set_ylabel('Angle [rad]')
47     axes[(1, 1)].legend()
48
49     # Add labels and titles
50     axes[(0, 0)].set_ylabel('Temperatura [°C]')
51     axes[(1, 0)].set_ylabel('Presión [hPa]')
52     axes[(2, 0)].set_ylabel('Altitud [m]')
53     axes[(0, 1)].set_ylabel('Aceleración [m/s^2]')
54     axes[(1, 1)].set_ylabel('Rotación [rad]')
55
56     # Add grid
57     for ax in axes.values():
58         ax.grid(True)
59
60     fig.suptitle('Sensor Data')

```

Código 3.9: Código utilizado para el procesamiento final de datos. Parte 1.

```
1 # Plot GPS (with robust error handling)
2 map_ax.clear()
3 roads = cfeature.NaturalEarthFeature(
4     category='cultural',
5     name='roads',
6     scale='10m',
7     edgecolor='gray',
8     facecolor='none',
9     linewidth=0.5
10 )
11 map_ax.add_feature(roads)
12 map_ax.add_feature(cfeature.LAND)
13 map_ax.add_feature(cfeature.OCEAN)
14 map_ax.add_feature(cfeature.COASTLINE)
15 map_ax.add_feature(cfeature.BORDERS, linestyle=':')
16
17 try:
18     lon = df.apply(lambda row:
19                     nmea_to_decimal_degrees(str(row['Longitude']), row['E/W']), axis=1)
20     lat = df.apply(lambda row:
21                     nmea_to_decimal_degrees(str(row['Latitude']), row['N/S']), axis=1)
22
23     #Check for NaN values before plotting
24     mask = ~lon.isna() & ~lat.isna()
25     map_ax.plot(lon[mask], lat[mask], 'ro-',
26                 transform=ccrs.Geodetic(),
27                 markersize=1)
28     map_ax.set_extent([lon[mask].min() - 0.5, lon[mask].max() + 0.5,
29                        lat[mask].min() - 0.2, lat[mask].max() + 0.2],
30                        crs=ccrs.PlateCarree())
31 except (KeyError, TypeError, ValueError) as e:
32     print(f"Error plotting GPS: {e}")
33
34 map_ax.set_title('Localización GPS')
```

Código 3.10: Código utilizado para el procesamiento final de datos. Parte 2.

```

1 # Main function to read CSV and plot
2 def main():
3     try:
4         df = pd.read_csv('lora_data.csv') #Read the CSV file
5         print(df)
6
7         #Clean the Temperatura Column
8         df['Temperatura'] = pd.to_numeric(df['Temperatura'], errors='coerce')
9         #Handle any NaN values (e.g., replace with 0 or the mean)
10        df['Temperatura'].fillna(0, inplace=True)
11
12        fig = plt.figure(figsize=(18, 8), facecolor='#e7e7e8')
13        gs = fig.add_gridspec(4, 2, height_ratios=[1, 1, 1, 2])
14        axes = {}
15        for i in range(3):
16            for j in range(2):
17                axes[(i, j)] = fig.add_subplot(gs[i, j])
18
19        projection = ccrs.PlateCarree()
20        map_ax = fig.add_subplot(gs[3, :], projection=projection)
21        map_ax.stock_img()
22        map_ax.set_global()
23
24
25        plot_data(df, fig, axes, map_ax)
26
27        plt.subplots_adjust(wspace=0.2, hspace=0.5)
28        plt.show()
29
30
31    except FileNotFoundError:
32        print("Error: 'lora_data.csv' not found.")
33    except pd.errors.EmptyDataError:
34        print("Error: 'lora_data.csv' is empty.")
35    except pd.errors.ParserError:
36        print("Error: Could not parse 'lora_data.csv'. Check the file format.")
37    except Exception as e:
38        print(f"An unexpected error occurred: {e}")
39
40 if __name__ == "__main__":
41     main()

```

Código 3.11: Código utilizado para el procesamiento final de datos. Parte 3.

4. PROTOTIPADO Y CONSTRUCCIÓN

4.1. Cohete

4.1.1. Motor

- **Modelo:** AEROTECH G80-7T.
- **Propósito:** Proporcionar el empuje necesario para alcanzar la altura objetivo de 400 metros.
- **Características técnicas:**
 - Tipo: Propelente sólido.
 - Clase: Clase G.
 - Empuje máximo: 80 Newtons.
 - Duración del pico de fuerza: Aproximadamente 0.7 segundos tras la ignición.
 - Tiempo total de combustión: Cerca de 2 segundos.
 - Relación empuje-peso: Asegura una relación mínima de 5:1, lo que garantiza un despegue seguro y estable.

4.1.2. Cuerpo (Fuselaje)

- **Material:**

- Se utiliza cartón para disminuir el peso y lograr llegar a 1 Kg.

- **Propósito:**

- Proveer soporte estructural al cohete y albergar sus componentes internos (motor, carga útil, sistema de recuperación).
 - Garantizar integridad durante el lanzamiento, vuelo y aterrizaje.
 - Reducir el arrastre aerodinámico gracias a su diseño cilíndrico.

- **Características adicionales:**

- El fuselaje debe ser rígido para soportar las fuerzas dinámicas del lanzamiento.
 - La longitud mínima del cohete es de 80 cm, cumpliendo los requisitos de la competencia.
 - Diámetro interno mínimo de 66 mm para permitir el ingreso del CanSat.
 - Se determina partir el fuselaje en 3 secciones, el cuerpo principal de 61 cm que se puede observar en la Figura 4.1a, un acople de 2 cm que se muestra en la Figura 4.1b y finalmente una sección para la carga útil. que podemos observar en la Figura 4.1c.

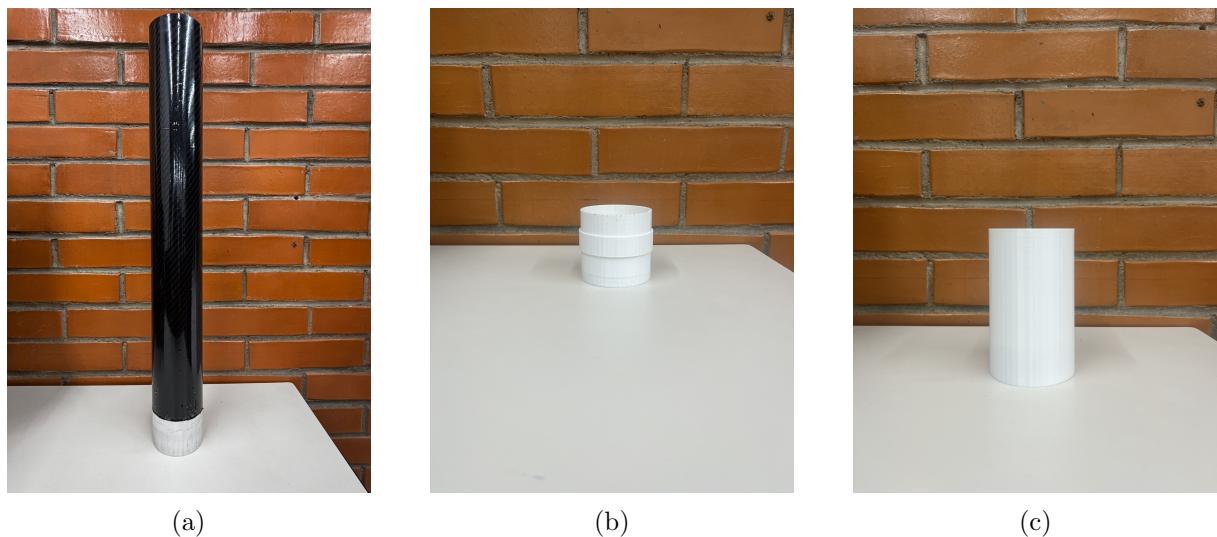


Figura 4.1: a)Fuselaje del cohete. b) Acople. c) Sección carga útil.

4.1.3. Cono

▪ **Material:**

- PLA estándar.

▪ **Propósito:**

- Disminuir el arrastre aerodinámico.
- Proteger los componentes internos, como el CanSat, del impacto inicial y el flujo de gases.

▪ **Diseño:**

- Nariz de forma parabólica seleccionada por su eficiencia aerodinámica y facilidad de fabricación. El prototipo final del cono se puede observar en la Figura 4.2.



Figura 4.2: Cono del cohete.

4.1.4. Aletas y Estabilizadores

- **Material:**

- PLA estándar.

- **Propósito:**

- Mantener la estabilidad direccional del cohete durante el ascenso.
 - Minimizar las desviaciones provocadas por vientos laterales o turbulencias.

- **Diseño:**

- Aletas trapezoidales seleccionadas por su mejor relación estabilidad/arrastre, estas se pueden observar en la Figura 4.3.
 - Posición: En la parte inferior del fuselaje, distribuidas uniformemente para optimizar la estabilidad.



Figura 4.3: Aletas y estabilizadores del cohete.

4.1.5. Cordón de Choque

- **Material:**

- PVC.

- **Propósito:**

- Conectar el paracaídas y cono al cuerpo del cohete, evitando que este se pierda o dañe en el aterrizaje.
 - Absorber el impacto generado al desplegarse el paracaídas, distribuyendo la fuerza y protegiendo las conexiones del sistema de recuperación.

- **Características adicionales:**

- Resistente a altas tensiones para soportar el peso del cohete durante el descenso.

4.1.6. Paracaídas

■ **Material:**

- Nylon ligero y resistente.

■ **Propósito:**

- Garantizar un descenso controlado y seguro del cohete junto con el CanSat.

■ **Diseño:**

- Paracaídas octagonal seleccionado por su simplicidad y eficiencia, así como un descenso controlado. Este paracaídas puede observarse en la Figura 4.4



Figura 4.4: Paracaídas extendido

4.1.7. Anillos de Soporte para el Motor

■ **Material:**

- PLA estándar.

■ **Propósito:**

- Fijar el motor en su posición dentro del fuselaje.
- Garantizar un centrado adecuado del motor para evitar vibraciones y desbalances durante el vuelo. Estos anillos pueden observarse en la Figura 4.5



Figura 4.5: Anillos de soporte para el motor del cohete.

4.1.8. Papel Ignífugo

- **Material:**

- Papel marca Estes para resistir altas temperaturas.

- **Propósito:**

- Proteger los componentes internos del calor generado por la combustión del motor.
 - Asegurar que el sistema de recuperación no se dañe antes de su activación.

4.1.8.1 Integración y montaje

Finalmente el cohete es ensamblado y está preparado para la competencia, el cohete final puede ser observado en la Figura 4.6.



Figura 4.6: Ensamble del cohete final.

4.2. Cansat

El proceso de prototipado del CanSat se centró en la construcción y validación de un modelo funcional que cumpliera con los objetivos y requisitos establecidos por el Concurso Colombiano de Cohetería Deportiva 2024. Este prototipo final fue fabricado a partir de los procesos de diseño previos.

4.2.1. Materiales y Componentes Utilizados

Para la construcción del prototipo, se seleccionaron los siguientes materiales y componentes, considerando su disponibilidad, costo y facilidad de uso:

- **Estructura:** Fabricada en PLA estándar mediante impresión 3D, seleccionada por ser un material económico, liviano, y fácilmente personalizable para ajustes rápidos. La estructura incluye soportes internos para fijar las placas electrónicas a presión, asegurando su estabilidad durante el vuelo.
- **Plataforma Electrónica:** Dos placas perforadas tipo protoboard de 5.5x7 cm, estañadas, fueron utilizadas para montar los componentes electrónicos. Estas placas se adaptaron al diseño interno del CanSat, fijándolas de manera vertical para optimizar el espacio disponible.
- **Subsistemas Electrónicos:**
 - **Sensores:** AHT20 para temperatura y humedad, BMP280 para presión y altitud, MPU6050 para aceleración y giroscopio, y NEO6MV2 para GPS, conectados al microcontrolador ESP32 mediante interfaces I2C y UART.
 - **Sistema de Comunicación:** Módulo Bluee LoRa, operando en la banda ISM (433/868 MHz).
 - **Fuente de Energía:** Batería de litio de 5V y 800 mAh con un regulador interno del ESP32 para alimentar los componentes a 3.3V.

4.2.2. Técnicas de Fabricación

El proceso de prototipado incluyó diversas técnicas para la construcción y ensamblaje:

- **Impresión 3D:** La estructura fue diseñada en CAD y fabricada con una impresora 3D. Se ajustaron las tolerancias para permitir la fijación a presión de las placas perforadas y los soportes del paracaídas.
- **Soldadura y Ensamblaje Electrónico:** Los componentes fueron soldados en las placas perforadas, asegurando conexiones firmes y organizadas para minimizar interferencias.
- **Integración del Sistema de Despliegue:** El diseño permitió que la estructura del CanSat encaje perfectamente debajo de la punta del cohete, donde es eyectado junto con el paracaídas mediante la presión generada por el motor al finalizar su ascenso.

4.2.3. Integración y Montaje

- **Plataforma Electrónica:** Se utilizaron dos placas perforadas tipo protoboard estañadas de 5.5x7 cm, ubicadas de forma vertical en el centro del CanSat para optimizar el espacio dentro del mismo.
- **Distribución Interna:** Los sensores, el microcontrolador y el módulo de comunicación se han montado sobre las placas distribuidos de tal forma que se busca optimizar el espacio dentro del CanSat.

4. PROTOTIPADO Y CONSTRUCCIÓN

En la Figura 4.7 se pueden observar las etapas de fabricación del sistema electrónico del Cansat. En la Figura 4.8 se puede observar el prototipo final armado.

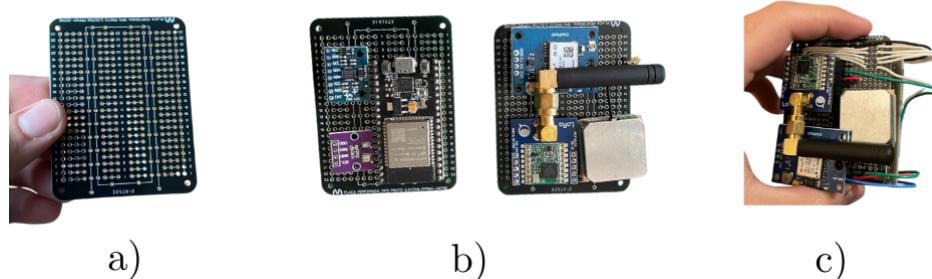


Figura 4.7: a) Placas base de desarrollo. b) Integración de componentes. c) Ensamble final de la electrónica

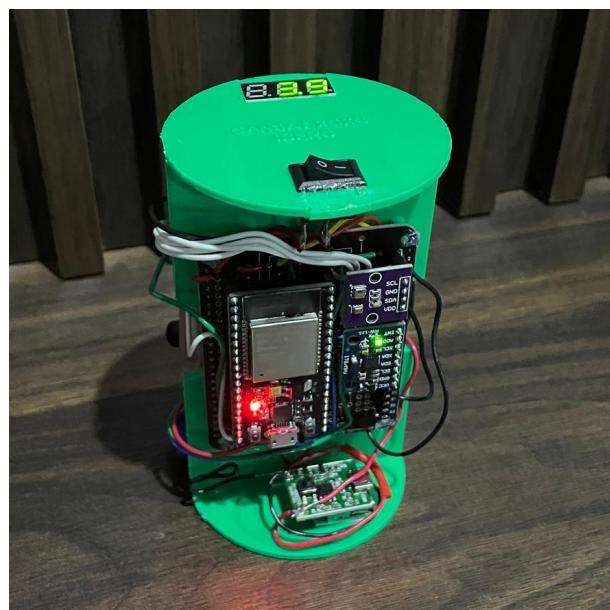


Figura 4.8: CanSat Final Armado

5. PRUEBAS Y VALIDACIÓN

5.1. Cohete

Se realizaron pruebas con cohetes de diferentes tamaños.

- **Primeras pruebas:**

En las primeras pruebas de lanzamiento se utilizaron motores de las categorías C y D. A través de estas pruebas, se observó que tanto el tamaño como el peso de los cohetes resultaron inadecuados, ya que eran demasiado grandes y pesados en relación con la capacidad de los motores empleados. Esto provocó que, durante los lanzamientos, los cohetes no lograran despegar y quedaran atascados en el riel de lanzamiento. En la Figura 5.1 podemos observar el tamaño inicial de los primeros prototipos de cohetes durante sus intentos de despegue



Figura 5.1: Primer lanzamiento del cohete.

- **Reducción de tamaño:**

Se realizaron mejoras en los cohetes considerando las simulaciones generadas en OpenRocket, las cuales sugerían reducir el tamaño y peso del cohete para optimizar los lanzamientos con los motores disponibles. Además, se experimentó con la instalación de motores en paralelo con el objetivo de replicar la potencia de lanzamiento de un motor de categoría G.

Sin embargo, durante las pruebas de lanzamiento, se observó que los motores conectados en paralelo no lograban activarse simultáneamente. Esto ocasionaba que los cohetes despegaran inclinados en diferentes direcciones y no alcanzaran la altura esperada.

Esta segunda prueba se ilustra en la Figura 5.2, donde se aprecia claramente la reducción en el tamaño del cohete, lo que permitió que, esta vez, el cohete lograra despegar.

- **Pruebas del centro de masa:**

Se llevaron a cabo pruebas para verificar la relación entre el centro de masa y el centro de presiones utilizando prototipos de cohetes modificados tras las pruebas con motores instalados en paralelo.

- **Pruebas finales**



Figura 5.2: Segundo lanzamiento de cohete tomando en cuenta la reducción del tamaño y peso simulado en OpenRocket.

Después de realizar las segundas pruebas de despegue y verificar el centro de masa, se llevaron a cabo las últimas modificaciones al cohete. Con estas mejoras, se efectuaron las pruebas finales mediante un último lanzamiento, confirmando que tanto el modelo físico del cohete como las simulaciones funcionaban correctamente.

Estas pruebas de funcionamiento se pueden observar en la Figura 5.3.

5.2. Cansat

Para validar y verificar el correcto funcionamiento del CanSat y sus componentes se realizaron las siguientes pruebas:

- **Pruebas de Vibración:**

- Simulación de las condiciones de lanzamiento utilizando un agitador mecánico.
- Ciclo de vibración sinusoidal en frecuencia de 20-2000 Hz.

- **Pruebas de Comunicación:**

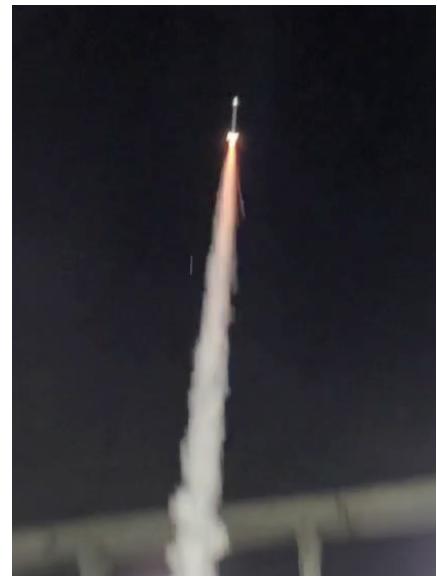
- Pruebas de transmisión en línea de vista a diferentes distancias (500 m, 1 km, y 1.5 km).
- Validación de integridad de los paquetes de datos enviados.

- **Pruebas de Sensores:** Cada sensor ha sido probado en condiciones controladas para validar su precisión y funcionalidad:

- Comparación de los datos de temperatura y presión con equipos calibrados.
- Validación de las lecturas de aceleración y giroscopio frente a movimientos programados.
- Comparación de los datos arrojados por el GPS con los obtenidos en Google Maps.



(a) Lanzamiento final de cohete antes del despegue.



(b) Lanzamiento final de cohete despegando.

Figura 5.3: Lanzamiento del prototipo final de cohete.



(a)



(b)

Figura 5.4: Fotos de la prueba de Cansat con dron.

- **Pruebas de Duración Energética:** Simulación de una misión completa, midiendo el consumo de energía bajo condiciones reales, para confirmar que la batería es suficiente para la duración requerida (mínimo 4 horas de operación continua).
- **Pruebas de Integración Final:** Ensamble completo del CanSat para pruebas funcionales, asegurando que todos los subsistemas trabajen de manera integrada. Esto incluye:
 - Prueba completa de la misión simulada (con ayuda de un dron el cual se elevó a 450 m) desde el despliegue hasta el aterrizaje, fotos de la prueba pueden ser observadas en la Figura 5.4.
 - Monitoreo en tiempo real de los datos transmitidos.

Los resultados de esta prueba nos permitieron verificar la transmisión de datos durante toda la prueba y nos permitieron verificar los códigos para graficar los datos en tiempo real y al final del proceso. Los resultados se pueden observar en la Figura 5.5

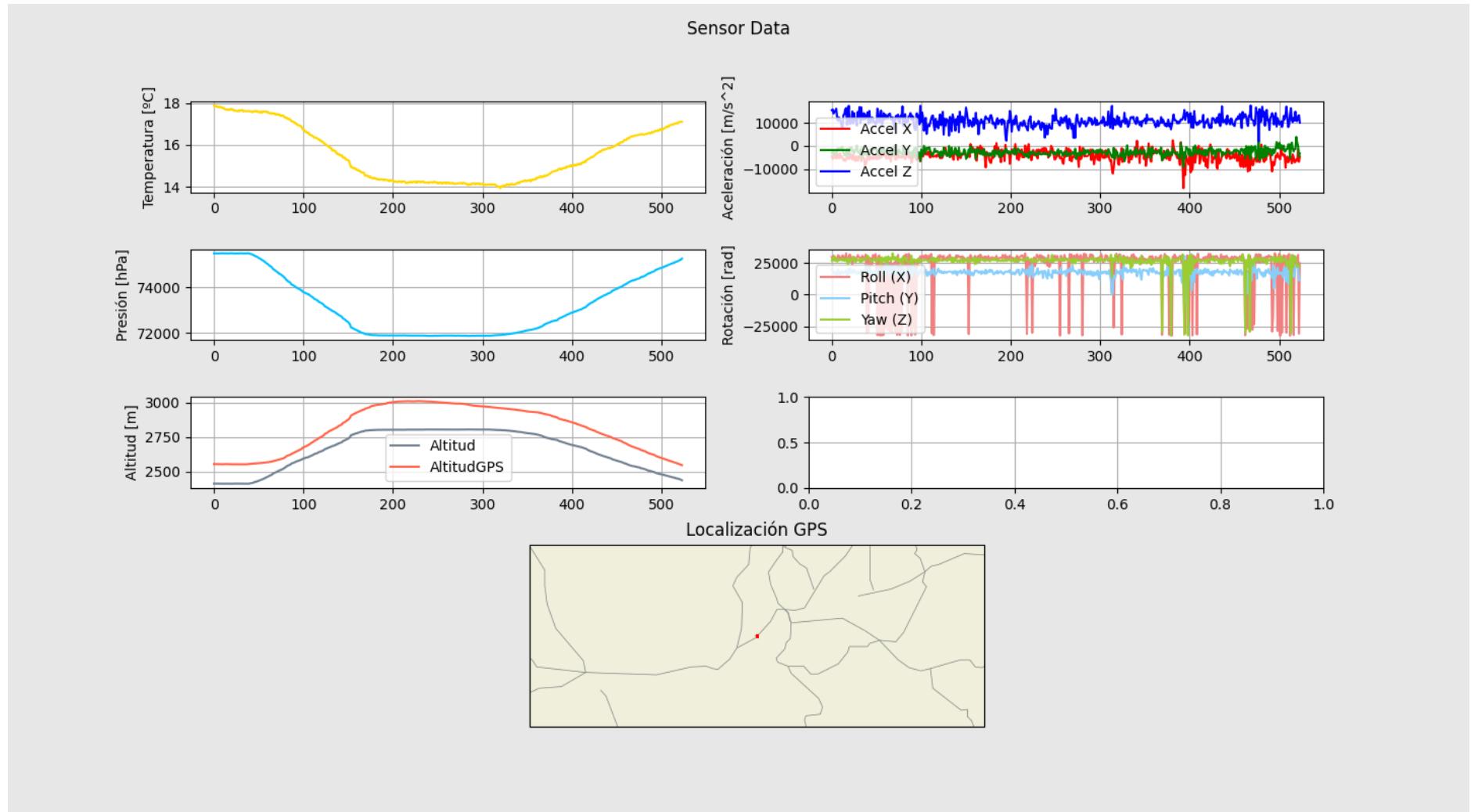


Figura 5.5: Resultados de la prueba final.

6. LECCIONES APRENDIDAS

1. Trabajo en equipo y colaboración interdisciplinaria

Aprendimos la importancia de delegar responsabilidades y aprovechar las fortalezas individuales de cada integrante. La colaboración entre las áreas de diseño, electrónica y funcionamiento fue clave para superar obstáculos y lograr un proyecto sólido.

2. Gestión del tiempo y recursos

Durante los cuatro meses del proyecto, enfrentamos el reto de equilibrar nuestras responsabilidades académicas, laborales y personales con las demandas del concurso. Esta experiencia nos enseñó a priorizar tareas, planificar eficazmente y optimizar recursos para cumplir con los plazos establecidos.

3. Adaptabilidad y resolución de problemas

La construcción del cohete implicó numerosos desafíos técnicos y situaciones imprevistas. A través de la experimentación, los análisis posteriores a cada prueba y el aprendizaje de nuestros errores, mejoramos nuestra capacidad de adaptación y desarrollamos habilidades para resolver problemas de manera creativa y eficiente.

4. Aplicación práctica de conocimientos adquiridos

Este proyecto fue una oportunidad para poner en práctica conceptos teóricos aprendidos en la carrera, como diseño mecánico, electrónica, control, programación y aerodinámica. Ver cómo estas áreas se integran en un proyecto real reforzó nuestra comprensión y consolidó nuestras habilidades técnicas.

5. Comunicación efectiva

La coordinación y el intercambio constante de ideas fueron esenciales para mantener un flujo de trabajo eficiente. Aprendimos a escuchar, expresar nuestras opiniones y resolver desacuerdos de manera constructiva, fortaleciendo la dinámica del equipo.

6. Perseverancia ante la adversidad

Enfrentamos momentos de frustración y cansancio, especialmente cuando los resultados no eran los esperados. Sin embargo, nuestra determinación como equipo nos permitió mantener el enfoque y superar las dificultades, valorando el aprendizaje en cada etapa del proceso.

7. Importancia de la iteración y la mejora continua

Cada prueba fue una oportunidad para identificar áreas de mejora y refinar nuestro diseño. Este ciclo constante de ensayo y error nos permitió alcanzar un nivel de calidad que representara con orgullo a nuestra universidad en el concurso internacional.

Esta experiencia no solo enriqueció nuestras competencias como futuros ingenieros, sino que también nos dejó una valiosa lección sobre la importancia del esfuerzo colectivo y la pasión por superar límites en proyectos desafiantes.

7. EXPERIENCIAS SIGNIFICATIVAS

Desde el momento en que surgió la idea de participar en el concurso, nos emocionaba la posibilidad de emprender un viaje como equipo. Al principio, no llegamos a dimensionar todo lo que implicaría competir en un certamen internacional; sin embargo, desde el inicio teníamos muy claro nuestro objetivo: representar orgullosamente a la Universidad Iberoamericana.

Nos consolidamos como equipo a principios de agosto, pocos días antes de iniciar un nuevo semestre. Desde ese momento, las largas jornadas en la universidad para trabajar en el cohete se fueron extendiendo cada vez más, especialmente conforme se acercaba el 21 de noviembre. Fue un desafío para todos, tanto para los integrantes del equipo como para nuestros asesores, ya que tuvimos que dejar de lado algunas responsabilidades académicas, laborales y personales para enfocarnos en alcanzar nuestra meta.

A lo largo del proyecto, enfrentamos numerosos momentos de frustración y cansancio, especialmente cuando las cosas no salían según lo planeado. En cada lanzamiento de prueba, analizábamos nuestras áreas de oportunidad tanto como equipo en conjunto como en nuestras respectivas especialidades: diseño, electrónica y funcionamiento. Esto nos permitió identificar y corregir errores, evitando repetirlos y enfocándonos en solucionar aquello que no estaba funcionando.

Aprendimos a valorar los instantes que compartimos juntos, ya fuera por la mañana, por la tarde o incluso durante largas noches de trabajo. Aunque al principio hubo desacuerdos, logramos construir una verdadera dinámica de equipo: delegamos tareas, aprovechamos las habilidades individuales y organizamos nuestras actividades para maximizar el tiempo y la energía, logrando trabajar de manera eficiente en un mismo espacio.

Durante los últimos meses, asumimos el desafío de construir un cohete prácticamente desde cero. Ha sido una experiencia profundamente enriquecedora, demandante y agotadora. Este proceso, lleno de retos, avanzó tanto con pasos pequeños como con grandes logros, cada uno aportándonos nuevos conocimientos sobre esta fascinante área. Aunque nuestro equipo está compuesto exclusivamente por ingenieros mecatrónicos, este proyecto nos permitió aplicar de manera integral lo aprendido durante nuestra carrera. Si bien el prototipo tiene un enfoque mayormente mecánico, la integración de sistemas electrónicos ha sido clave, exigiéndonos diseñar, analizar y ejecutar procesos electrónicos de calidad.

Nuestro objetivo principal era implementar un cohete que estableciera comunicación entre el CanSat y una altitud de 400 metros, asegurando su recuperación mediante un paracaídas. Vivimos momentos emocionantes y satisfactorios, como cuando el CanSat logró transmitir datos desde más de 400 metros de altura. Sin embargo, también enfrentamos situaciones tensas, como el riesgo de que el dron que transportaba el CanSat quedara sin batería, o la frustración cuando algunos cohetes no lograron despegar. Cada experiencia, tanto los logros como los desafíos, nos dejó aprendizajes valiosos y nos permitió crecer como estudiantes y como equipo.

Esta experiencia no solo nos desafió como estudiantes, sino que también nos unió como equipo, fortaleciendo nuestra capacidad de colaborar y superar obstáculos juntos. Estamos muy orgullosos del trabajo realizado y estamos seguros de que este proyecto es solamente el inicio de muchos más que tenemos por delante como grupo de amigos.

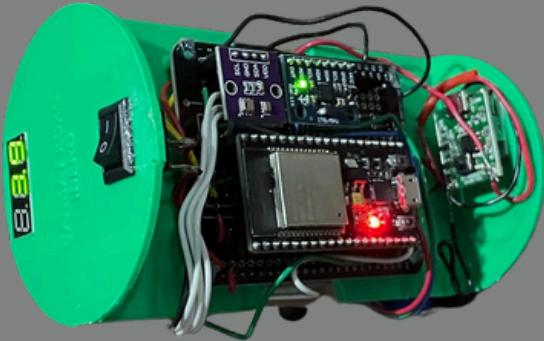
8. FICHAS TÉCNICAS

Cohete Huitzilopochtli MX-IB-24

Categoría	Componente	Especificaciones
Motor	Motor Clase G807T	<ul style="list-style-type: none"> - Tipo: Motor para cohetes modelo G807T - Clase: G - Aplicación: Cohetes de propulsión sólida - Características: Alta potencia, duración limitada, ideal para lanzamientos de CanSat - Precauciones: Uso bajo normas de seguridad aeronáutica y en áreas habilitadas para lanzamientos.
Fuselaje	Dimensiones Físicas	<ul style="list-style-type: none"> - Longitud: 1 m - Diámetro: 75 mm - Peso: 1.200 kg - Altura de Vuelo: 400 m
CanSat	ESP32 WROOM 32	<ul style="list-style-type: none"> - Microcontrolador: ESP32 Dual Core - Conectividad: Wi-Fi y Bluetooth integrados - Frecuencia: 2.4 GHz - Memoria: Flash 4 MB - Voltaje de operación: 3.3 V - Aplicación: Sistema de control y transmisión de datos de CanSat
	GPS6MV2 (NEO-6M-0-001)	<ul style="list-style-type: none"> - Tipo: Módulo GPS con antena externa - Sensibilidad: -161 dBm - Precisión: 2.5 metros (CEP) - Frecuencia: 1 Hz (actualizable a 5 Hz) - Voltaje de operación: 3.3 - 5 V - Interfaz: UART - Aplicación: Geoposicionamiento en tiempo real del CanSat durante el vuelo y recuperación post-lanzamiento



Cansat Huitzilopochtli MX-IB-24



Datasheet - Parámetros de Telemetría del CanSat

Parámetro	Rango/Especificación	Sensor/Componente	Descripción
Temperatura	5°C - 50°C	Sensor DHT22 o similar	- Mide la temperatura ambiental durante el vuelo.
Altura	0 - 500 m	Barómetro BMP280/Altímetro	- Determina la altura relativa usando presión atmosférica.
Posicionamiento Global	Coordinadas GPS en tiempo real	GPS6MV2 (NEO-6M-0-001)	- Proporciona latitud, longitud y altitud con precisión de 2.5 m (CEP).
Presión Atmosférica	300 - 1100 hPa	Barómetro BMP280	- Monitorea la presión ambiental para análisis meteorológico y cálculo de altitud.
Aceleraciones	±16 g en tres ejes	Acelerómetro MPU6050	- Mide aceleraciones lineales en los ejes X, Y y Z para detectar cambios en velocidad.
Ángulos Roll, Pitch, Yaw	±180° en cada eje	Giroscopio MPU6050	- Determina la orientación del CanSat en el espacio.
Velocidad de Transmisión	1 Hz - Actualización de datos por segundo	ESP32 WROOM 32	- Frecuencia de envío de los parámetros a la estación base.
Protocolo de Comunicación	UART o I2C	ESP32 WROOM 32	- Interfaz para la conexión de sensores y transmisión de datos.

9. BROCHURE

Cohete Huitzilopochtli MX-IB-24

Innovación en Adquisición de Datos para Misiones Aéreas

Parámetro	Rango/Especificación	Descripción
Fuselaje	1 m de longitud, 75 mm de diámetro	Diseño aerodinámico y ligero para estabilidad en vuelo.
Peso	1.200 kg	Óptimo para operaciones seguras y manejo eficiente de carga útil.
Motor	Clase G807T	Propulsión sólida con alta potencia y duración limitada.
Altura Máxima	400 m	Ideal para pruebas educativas y lanzamientos controlados.

Acerca del Cohete

Nuestro cohete está diseñado para misiones educativas, experimentales y de investigación. Equipado con tecnología avanzada, permite alcanzar alturas de hasta 400 metros mientras transporta dispositivos como CanSat, integrando sistemas de telemetría y propulsión de alta precisión.

Contacto

Equipo Ilihuícatl Xoxoauhco
Universidad Iberoamericana
Ciudad de México, México
 @ilihuicatl_xoxoauhco



Características Principales

- Fuselaje compacto y resistente: 1 metro de longitud y 75 mm de diámetro.
- Peso óptimo para operaciones seguras: 1.200 kg.
- Altura de vuelo máxima: 400 metros.
- Sistema de propulsión con motor Clase G807T para alta eficiencia y estabilidad.
- Integración con dispositivos como CanSat para misiones de recolección de datos.

BIBLIOGRAFÍA

Components, A. (n.d.). *Model Rocket Building Basics*. <https://www.youtube.com/watch?v=LZMWsN65yk0&t=683s>. Accessed on November 20, 2024.

Dieter, G. E. y Schmidt, L. C. (2012). *Engineering Design*. 5th. New York, USA: McGraw-Hill Education. ISBN: 978-0073398143.

Ponder, D. M. (2013). *Designing Your Own Model Rocket*. Includes guidelines for model rocket design, safety codes, and additional resources for advanced rocketry projects. URL: <http://www.ohio4h.org/rocketsaway>.

Portocarrero-Hermann, J. A. et al. (2021). «Diseño, construcción y prueba estática experimental de un motor-cohete de combustible sólido». En: *Revista UIS Ingenierías* 20.2, págs. 97-108. URL: <https://revistas.uis.edu.co/index.php/revistauisingenierias/article/view/11068>.

Ramírez, L. P. B., Briñez, R. C., Tejada, J. C., Marmolejo, P. y Jaramillo, O. A. (2019). «Diseño de un CanSat para medición de variables ambientales». En: *Publicaciones e Investigación* 13.2. Disponible en línea en Dialnet. ISSN: 1900-6608. URL: <https://dialnet.unirioja.es/servlet/articulo?codigo=8660106>.

Riveros Enciso, F. A. y Rodríguez Herrera, L. A. (2010). «Diseño y construcción de un cohete aficionado controlado mediante el accionamiento de una tobera de empuje vectorial». En: *Universidad Militar Nueva Granada*. URL: <https://repository.unimilitar.edu.co/bitstream/handle/10654/3303/RiverosEncisoFelipeAlejandro2010.pdf>.

Team, T. M. R. (n.d.[a]). *Model Rocket Fins 101: Purpose, Shape, Size, and Placement*. Accessed on November 20, 2024. URL: <https://themodelrocket.com/model-rocket-fins-101-purpose-shape-size-and-placement/>.

– (n.d.[b]). *The Essential Guide to Model Rocket Nose Cones*. Accessed on November 20, 2024. URL: <https://themodelrocket.com/the-essential-guide-to-model-rocket-nose-cones/>.

Ulrich, K. T. y Eppinger, S. D. (2016). *Product Design and Development*. 6th. New York, USA: McGraw-Hill Education. ISBN: 978-0078029065.

ANEXOS

A. Primer anexo

El material complementario del proyecto puede ser encontrado en el siguiente enlace, este corresponde a una carpeta de google drive con videos, fotos y demás elementos del proyecto.

https://drive.google.com/drive/folders/1dZo1bvPxe010D5z6GQTBuzs0kW35RT6e?usp=drive_link

