



UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO INGENIERÍA
MECÁNICA



EVALUACIÓN DE CONFIABILIDAD EN CLUSTERS DE *CUBESATS*
PARA OBSERVACIÓN TERRESTRE

POR

Nicolás Gabriel Sepúlveda Estrada

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título profesional de Ingeniero Civil Aeroespacial

Profesor Guía:

Alejandro López Telgie, PhD (C)

Profesores Comisión:

Alejandro López Telgie, PhD (C)

Frank Tinapp Dautzenberg, PhD

Bernardo Hernández Vicente, PhD

Marzo 2023

Concepción (Chile)

© 2023 Nicolás Gabriel Sepúlveda Estrada

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento

Esta memoria está dedicada a todos quienes hayan contribuido con un grano de arena a ser quien he sido, soy y seré en la vida y profesión.

A mi mamá, Susana, por tu inconmensurable sacrificio hacia mí y mi hermano, criándonos frente a la adversidad para volvernos en los hombres que te llenan de orgullo. Te amo mamá.

A mis abuelos maternos, Eriz y Ana, por ser un pilar inamovible de amor, cariño y sabiduría desde nuestra niñez hasta nuestros días.

A mi hermano, Javier, por ser mi otra mitad. Tus consejos, eternas discusiones y apañe ante cualquier circunstancia hace que seas un mellizo único en el mundo.

A mi profesor guía, Alejandro, por toda la sabiduría y conocimiento entregados a lo largo de los años. Impulsar el sector espacial en Chile requieren de gente con la pasión y habilidad que usted posee y aplica todos los días al mover proyectos adelante.

A los profesores del Departamento de Ingeniería Mecánica de la Universidad de Concepción, por las enseñanzas, meticulosidad y sabiduría traspasada en cada ramo.

A mis excolegas de la Misión Asclepios I y II, en especial a Ilaria, Simon, Arnault, Mathieu, Chiara, Chloé, Elfie, Loïc y Verónica, por la gran aventura y desafío que ha sido levantar a distancia un proyecto único en el mundo y que ha abierto tantas puertas en poco tiempo.

A mis colegas del Grupo de Interés en Propulsión (GIP), que durante años se ha trabajado en volver al equipo a lo que siempre ha querido, llegar más alto que nadie en el país.

A mis amigos, en especial a Nicolás, Jaime, Sergio, Tatiana, Diego, Sebastian, Felipe, Fernando, María Ignacia, Estefanía, Felipe Castañeda, Sebastián, Fabián y Sergio Ignacio. No existen palabras para describir cuan agradecido estoy de haberlos conocido y de tenerlos en mi corazón.

Decir gracias queda corto frente al eterno cariño y apoyo incondicional recibido, que solamente queda retribuirlo de vuelta a cada uno por toda la eternidad.

Nicolás Gabriel Sepúlveda Estrada

“Do not lose time on daily trivialities. Do not dwell on petty detail. For all these things melt away and drift apart within the obscure traffic of time. Live well and live broadly. You are alive and living now. Now is the envy of all of the dead.” – Emily

Resumen

El ascenso de los *CubeSats*, junto con la reducción significativa en los costos de lanzamiento, han permitido avances significativos en la miniaturización tecnológica en satélites con componentes más económicos y compactos, lo que ha aumentado el acceso al espacio para nuevas instituciones y países. Esto ha llamado la atención de agencias y empresas tradicionales, ya que, actualmente los utilizan como plataforma geomática debido a la creciente demanda de datos e imágenes de la superficie terrestre; y a la rapidez de desarrollo, producción y lanzamiento que estos permiten, difícil de lograr con un satélite monolítico. Sin embargo, uno de los principales problemas de los *CubeSats* es su alta tasa de fallas y la restricción de espacio disponible. A pesar de la falta de datos útiles para estimar su confiabilidad, los autores han desarrollado metodologías de cálculo y han señalado formas de mitigar las fallas. En este contexto, las misiones satelitales distribuidas (DSM) aparecen como una nueva forma de resolver el uso de los *CubeSats* para misiones de observación terrestre y mejorar la confiabilidad del sistema espacial, junto con otras estrategias de mitigación de riesgos.

De tal manera, el objetivo de este trabajo es estimar y evaluar la confiabilidad y los parámetros de ingeniería de sistemas para unidades *bus* y segmentos espaciales de *clusters* de *CubeSats* de diferentes configuraciones mediante simulaciones y análisis de estrategias de mitigación de riesgos. Para ello, se realizó un estudio del estado del arte y se construyó una arquitectura que simula y estima la confiabilidad y los parámetros de ingeniería de sistemas (masa, volumen y costo) para diferentes unidades *bus* y segmentos espaciales de *clusters* de *CubeSats*, utilizando diferentes estrategias de redundancia y mitigación de riesgos en el segmento espacial.

La arquitectura implementada considera un segmento espacial tipo *cluster*, compuesto por elementos iguales tipo *CubeSats*, cuyo número depende del área a cubrir. A su vez, cada elemento se compone de equipos denominados *bus* y *payload*. Se compararon el espacio y la masa disponibles del elemento junto con el costo del proyecto en un tradespace. Además, utilizando como base los diagramas de bloques de confiabilidad (RBD), específicamente el sistema k-out-of-n, se estimó la confiabilidad del segmento espacial a partir de las curvas del elemento. Una vez construida la arquitectura, se evaluaron las distintas estrategias de mitigación y sus efectos en la confiabilidad y costo total del segmento, así como en la masa y volumen disponibles de cada elemento.

Palabras clave: Curvas de Confiabilidad, Parámetros de Ingeniería de Sistemas, Estándar CubeSat, Observación Terrestre, Diagramas de Bloques de Confiabilidad

Abstract

The rise of CubeSats, along with the significant reduction in launch costs, has enabled significant advances in technological miniaturization on satellites with cheaper and more compact components, increasing access to space for new institutions and countries. This has caught the attention of traditional agencies and companies, which use them as a geomatics platform due to the growing demand for data and images of the earth's surface; and the speed of development, production and launch that these allow, difficult to achieve with a monolithic satellite. However, one of the main problems of CubeSats is their high failure rate and the restriction of available space. Despite the lack of useful data to estimate their reliability, the authors have developed calculation methodologies and pointed out ways to mitigate failures. In this context, distributed satellite missions (DSMs) appear as a new way to solve the use of CubeSats for Earth observation missions and improve the reliability of the space system, along with other risk mitigation strategies.

Thus, the objective of this work is to estimate and evaluate the reliability and engineering parameters of systems for bus units and spatial segments of CubeSats clusters of different configurations through simulations and analysis of risk mitigation strategies. To this end, a state-of-the-art study was conducted, and an architecture was built that simulates and estimates the reliability and systems engineering parameters (mass, volume and cost) for different bus units and space segments of CubeSat clusters, using different redundancy and risk mitigation strategies in the space segment.

The implemented architecture considers a cluster-type spatial segment, composed of CubeSats-type equal elements, whose number depends on the area to be covered. In turn, each element is composed of equipment called bus and payload. The available space and mass of the element are compared with the cost of the project in a tradespace. In addition, using as a basis the reliability block diagrams (RBD), specifically the k-out-of-n system, the reliability of the spatial segment was estimated from the curves of the element. Once the architecture was built, the different mitigation strategies and their effects on the reliability and total cost of the segment, as well as on the available mass and volume of each element, were evaluated.

Keywords: Reliability Curves, Systems Engineering Envelopes, CubeSat Standard, Earth Observation, Reliability Block Diagrams

Tabla de Contenidos

Tabla de Contenidos.....	iii
Lista de Tablas	v
Lista de Figuras	vi
Glosario	viii
1 CAPÍTULO 1: Introducción.....	9
1.1 Introducción.....	9
1.2 Planteamiento del Problema	19
1.3 Objetivos.....	23
1.4 Metodología.....	23
2 CAPÍTULO 2: Marco Teórico	25
2.1 Terminología	25
3 CAPÍTULO 3: Estado del Arte	27
3.1 Diseño de Arquitecturas y SE Envelopes	27
3.2 Confiabilidad de <i>CubeSats</i>	29
4 CAPÍTULO 4: Desarrollo del Estimador.....	39
4.1 Propuesta Inicial	39
4.2 Consideraciones de Diseño.....	41
4.3 Nueva Arquitectura.....	46
4.4 Módulo Main_est.....	48
4.5 Módulo Reliability.py.....	49
4.6 Módulo Env_tradespace.py	50
5 CAPÍTULO 5: Simulación y Resultados	51
5.1 Estrategias de Mitigación y su Tradespace.....	52
5.2 Discusión	58
6 CAPÍTULO 6: Conclusiones	59
6.1 Conclusiones.....	59
6.2 Trabajos futuros	60
Referencias	61
7 ANEXO 1: Taxonomía de la terminología DSM (le Moigne et al., 2020).....	64
8 ANEXO 2: Carta Gantt	65
9 ANEXO 3 : ISISPACE 6-Unit CubeSat Dimensions	66
10 ANEXO 4: ISISPACE 3 Unit CubeSat Dimensions.....	67
11 ANEXO 5: ISISPACE CubeSat 1 Unit Dimensions.....	68
12 ANEXO 6: Main_est.py.....	69

13 ANEXO 7: Reliability.py	72
14 ANEXO 8: Env_tradespace.py	77
15 ANEXO 9: Bases de Datos de Componentes de Subsistemas	81
16 ANEXO 10: Diagrama de Flujos Main_est	82
17 ANEXO 11: Diagrama de Flujos Reliability	83
18 ANEXO 12: Diagrama de Flujo Env_tradespace	84

Lista de Tablas

Tabla 1.1 Propuesta de Taxonomía Unificada Basado en Masa y Tamaño (Rui & Xavier, 2019)	11
Tabla 4.1 Características de masa y volumen máximo para ISISPOD de 1U, 3U y 6U.....	44
Tabla 4.2 Tabla de Diseño de DSM	45
Tabla 4.3 Costo de Lanzamiento para el Cuarto Cuatrimestre de 2023, al 6 de febrero 2023 por SpaceFlight Industries	50

Lista de Figuras

Figura 1.1 Diagrama de Descomposición de Sistema Espacial (ECSS)	10
Figura 1.2 Sondas Pioneer 10 y 11	12
Figura 1.3 Ejemplos de <i>CubeSats</i> con diferentes números de unidades (varios autores)	13
Figura 1.4 Representación gráfica de distintos tamaños estandarizados de <i>CubeSats</i> (Radius Space)	13
Figura 1.5 Nanosatélites lanzados y por lanzar según organización (Kulu, 2023)	14
Figura 1.6 Tipos de nanosatélites lanzados y planificados hasta 01/01/2022 (Kulu, 2023)	14
Figura 1.7 Ejemplos de constelaciones tipo cluster desplegadas	16
Figura 1.8 Representaciones gráficas de CubeSats 3U usadas para EO (varios autores)	17
Figura 1.9 Curva de Confiabilidad de CubeSats (Bouwmeester et al., 2022).....	20
Figura 1.10 Ejemplos de aplicación de redundancias en CubeSats para subsistemas críticos (TUDelft).....	20
Figura 1.11 Tasa de éxito de misiones CubeSats (Villela et al., 2019).....	21
Figura 1.12 Infant Mortality de CubeSats desde 2002 hasta el 31 de mayo, 2018 (se excluyen fallas de lanzamiento) (Villela et al., 2019).....	21
Figura 1.13 Demandas Nacionales del Sector Espacial 2019 (AEB)	22
Figura 2.1 <i>Bathtub Curve</i> (Bouwmeester et al., 2022).....	26
Figura 3.1 Arquitectura propuesta para el diseño de <i>SmallSats</i> (Qaisar et al, 2016).....	27
Figura 3.2 Diagrama de sistema en paralelo con 2 componentes (autoría propia)	31
Figura 3.3 Gráfica comparativa de incremento de confiabilidad vs % confiabilidad comparativa (Billinton & Adam, 1992)	32
Figura 3.4 Diagrama de sistema serie-paralelo y su reducción para su análisis (autoría propia) .	32
Figura 3.5 Diagrama de sistema standby con 2 componentes (autoría propia).....	34
Figura 3.6 Tasa de falla de un satélite tipo CubeSat construido por universidades durante 6 meses (n=113) (Langer & Bouwmeester).....	35
Figura 3.7 Curvas de confiabilidad de cada subsistema (Bouwmeester et al, 2022)	36
Figura 3.8 Representación Gráfica Simplificada de un Sistema Satelital de Varios Niveles (Xu et al, 2019).....	37
Figura 3.9 Diagrama de Flujos del Proceso de Optimización Propuesto (Xu et al, 2019)	38
Figura 4.1 Metodología y Diagramas de Modelo Iniciales (autoría propia)	39
Figura 4.2 Diagramas RBD a Usar en Modelo Inicial (autoría propia)	40
Figura 4.3 Diagrama Segmento Espacial (autoría propia)	42
Figura 4.4 Curvas de Confiabilidad para cada subsistema (Bouwmeester et al., 2022)	43
Figura 4.5 Tipos de ISISPOD desarrollados por ISISPACE (ISISPACE)	44
Figura 4.6 Diagrama IPO Proceso Definitivo (autoría propia)	47
Figura 4.7 Diagrama Arquitectura Definitiva (autoría propia)	48
Figura 4.8 Ejemplo de Segmento Espacial con despliegue en fase de periodos T (autoría propia)	49
Figura 5.1 Gráfica de Curvas de Confiabilidad de Clusters Compuesto entre 1 a 10 Elementos (autoría propia).....	51
Figura 5.2 Gráfica Comparativa de Curvas de Confiabilidad Según Cantidad de Lanzamientos por Año y Elementos Lanzados (autoría propia).....	52

Figura 5.3 Gráfica Tradespace Según Cantidad de Lanzamientos por Año y Elementos Lanzados (autoría propia).....	53
Figura 5.4 Gráfica de Curvas de Confiabilidad con Distinta Cantidad de Elementos Iniciales (autoría propia).....	54
Figura 5.5 Gráfica Curvas de Confiabilidad y Tradespace Según Cantidad de Elementos Iniciales (autoría propia).....	55
Figura 5.6 Gráfica Tradespace Según Cantidad de Elementos Iniciales (autoría propia).....	55
Figura 5.7 Gráfica Curvas de Confiabilidad Según Cantidad de Elementos con Redundancia en EPS (autoría propia).....	56
Figura 5.8 Gráficas Curvas de Confiabilidad de Estrategias de Mitigación con o sin Redundancia en EPS (autoría propia).....	57
Figura 5.9 Gráfica de Tradespace de Estrategias de Mitigación con o sin Redundancia en EPS (autoría propia).....	57

Glosario

ADCS, AD&C	: <i>Attitude Determination & Control Subsystem</i>
AEB	: <i>Agencia Espacial Brasileña</i>
BMM	: <i>Bayesian Melding Method</i>
C&DH, CDH	: <i>Command & Data Handling</i>
Cal Poly	: <i>Universidad Politécnica Estatal de California</i>
COMMS	: <i>Communications Subsystem</i>
COTS	: <i>Commercial Off-The-Shelf</i>
DOA	: <i>Dead-On-Arrival</i>
DSM	: <i>Distributed Spacecraft Missions</i>
ECSS	: <i>European Cooperation for Space Standardization</i>
EO	: <i>Earth Observation</i>
EPS	: <i>Electrical Power System</i>
ESA	: <i>European Space Agency</i>
FAA	: <i>Federal Aviation Administration</i>
FSW	: <i>Flight Software</i>
GNC	: <i>Guidance, Navigation & Control</i>
GPS	: <i>Global Positioning Service</i>
GUI	: <i>Interfaz Gráfica de Usuario</i>
IPO	: <i>Input-Process-Output</i>
LEO	: <i>Low Earth Orbit</i>
MT	: <i>Memoria de Título</i>
MTTF	: <i>Mean Time to Failure</i>
NASA	: <i>National Aeronautics and Space Agency</i>
NTNU	: <i>Norwegian University of Science and Technology</i>
ODCS	: <i>Orbit Determination & Control Subsystem</i>
PIA	: <i>Proyecto de Ingeniería Aeroespacial</i>
PL, P/L	: <i>Payload</i>
PSA	: <i>Payload and Subsystems Avionics</i>
RBD	: <i>Reliability Block Diagrams</i>
RBDO	: <i>Reliability-Based Design Optimization</i>
ROSCOSMOS	: <i>Russian State Space Corporation</i>
SE	: <i>System Engineering</i>
SEU	: <i>Single Event Upset</i>
SSA	: <i>Small Spacecraft Avionics</i>
SSDL	: <i>Laboratorio de Sistemas Satelitales y Entorno Espacial</i>
SSO	: <i>Sun-Synchronous Orbit</i>
STS, M&S, STS&M	: <i>Structure, Structure and Mechanisms</i>
TCS	: <i>Thermal Control System</i>
TT&C	: <i>Telemetry, Tracking and Control</i>
TUDelft	: <i>Universidad Tecnológica de Delft</i>
USD	: <i>United States Dollar</i>

CAPÍTULO 1: Introducción

En este capítulo se presentará la problemática que fundamenta la investigación del presente trabajo, se establecerá la terminología a utilizar, los objetivos generales y específicos para generar una solución capaz de enfrentar la problemática antes mencionada, y la metodología a utilizar para la obtención de los objetivos.

1.1 Introducción

En la actualidad, la tecnología satelital es indispensable para el diario vivir de la sociedad, siendo utilizada por varios sectores, como el civil, gubernamental, militar y científico para varios propósitos: observación espacial, exploración espacial, prevención de riesgos, monitoreo ambiental, comunicaciones, posicionamiento global y estrategia militar (Kiselyov, 2020). Desde su concepción en la carrera espacial, la filosofía de diseño satelital estuvo predominada por componentes altamente confiables y diseños robustos y conservadores, hechos para soportar las condiciones extremas del espacio por largos períodos de tiempo (Langer & Bouwmeester, 2016).

El ascenso de los *CubeSats* cambió esta filosofía inicial por una en favor de utilizar productos *Commercial-Off-The-Shelf* (COTS), haciéndolos económicamente más viables para la industria espacial, reduciendo el costo y tiempo de desarrollo asociado e incrementando el acceso al espacio a nuevas instituciones, permitiendo un mayor número de lanzamientos por año (The CubeSat Program, 2022), sacrificando confiabilidad y redundancia en el proceso. A consecuencia de ello, sumado con la reducción sustantiva del costo por kilogramo (cuánto cuesta lanzar 1 kg de carga al espacio) desde \$60 000 USD/kg del Space Shuttle (Pielke, 2008) hasta \$5 500 USD/kg del Falcon 9 (Sheetz, 2022); el impulso generado también permitió mayores avances en la miniaturización tecnológica para construir satélites con componentes de menor costo, compactos y COTS (Poghosyan & Golkar, 2017). Antes de adentrarse a plantear el problema, es importante definir varios conceptos clave que contextualicen la situación a resolver junto con la importancia del estándar *CubeSat*.

1.1.1 Descomposición de un Sistema Espacial

Según lo establecido por ECSS y observable en la Figura 1.1, un sistema espacial es el sistema de más alto nivel dentro de un proyecto espacial, el de nivel de misión (ECSS Secretariat, 2012). Este sistema espacial se desglosa en tres dominios distintos (tierra, espacio y lanzador), los cuales se denominan segmentos. Cada segmento contiene su propia terminología la cual el informe compila y define para su principal uso en el sector espacial europeo, a nivel de programa (ECSS Secretariat,

2012). Para efectos de este informe sólo se utilizarán ciertas definiciones del segmento espacial, las que son listadas a continuación.

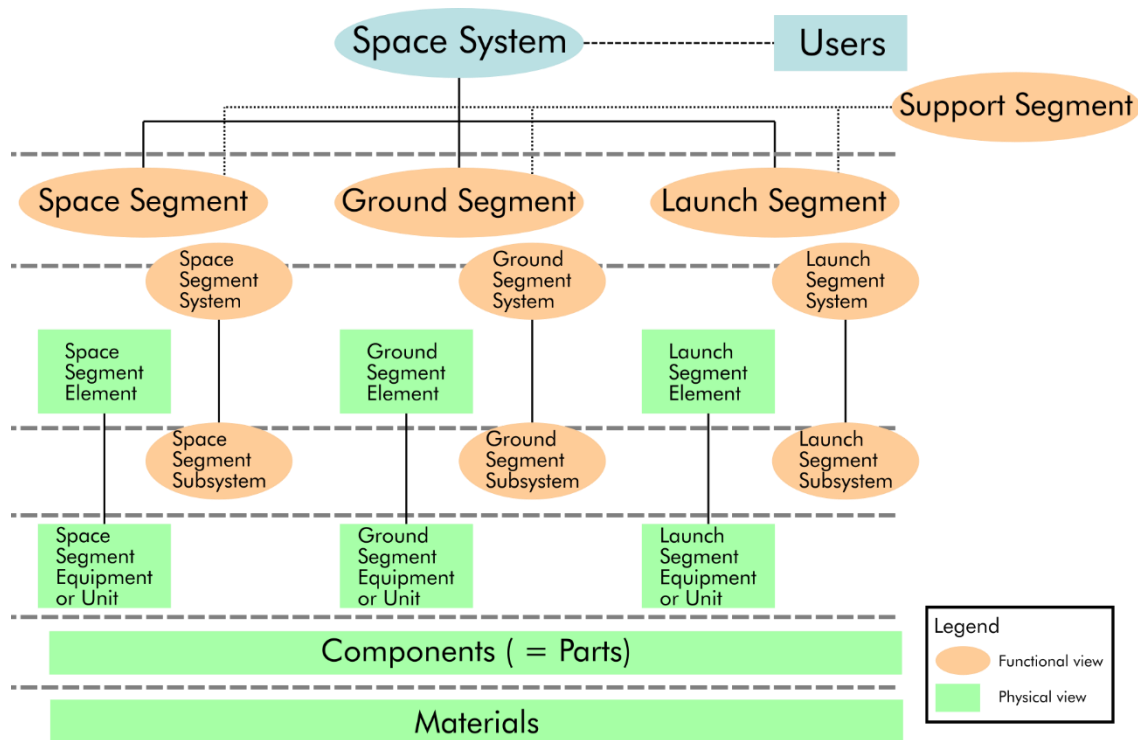


Figura 1.1 Diagrama de Descomposición de Sistema Espacial (ECSS)

- **Sistema:** conjunto de funciones interrelacionadas o interactivas constituidas para lograr un objetivo específico.
- **Segmento:** conjunto de elementos o combinación de sistemas que cumplen un subconjunto principal, autónomo, de los objetivos de la misión espacial.
- **Elemento:** combinación de equipos, componentes y piezas integrados.
- **Subsistema:** parte de un sistema que cumple una o más de sus funciones.
- **Equipo o Unidad:** Conjunto integrado de piezas y componentes que cumplen una función específica. Esta es autocontenida y clasificada como tal para los propósitos de una manufactura separada, diseño, especificaciones, almacenamiento, mantenimiento o uso.
- **Segmento espacial:** segmento espacial que forma parte de un sistema espacial, colocado en el espacio, para cumplir los objetivos de la misión espacial.
- **Sistema de segmento espacial:** sistema de sistema de segmento espacial dentro de un segmento espacial.
- **Elemento de segmento espacial:** un elemento dentro de un segmento espacial.
- **Subsistema de segmento espacial:** un subsistema dentro de un segmento espacial.
- **Equipo del segmento espacial:** Equipo dentro de un segmento espacial.
- **COTS:** componente electrónico comercial fácilmente disponible y no fabricado, inspeccionado o probado de acuerdo con las normas militares o espaciales.

1.1.2 Clasificación de Satélites según Masa y Tamaño

Actualmente, existen muchas clasificaciones y términos asociados a categorías de tamaño y rangos de masa de *NanoSats* o *SmallSats* dependiendo de la agencia o institución que las provee, tales como la *Federal Aviation Administration* (FAA) (Federal Aviation Administration (FAA), 2018), *National Aeronautics and Space Agency* (NASA) (Volynskaya & Kasyanov, 2016), *State Space Corporation* (ROSCOSMOS) (Volynskaya & Kasyanov, 2016), entre otros; haciendo la caracterización global y la comunicación, imprecisa (Rui & Xavier, 2019). Para este trabajo se definen los rangos de masas de las U estandarizadas de interés 1U, 3U y 6U, por medio de la taxonomía propuesta de A. S., Rui C. Botelho y Xavier, Ademir L. que unifica términos comunes usados en la industria y los combina en una sola descripción coherente, establecido en la Tabla 1.1.

Tabla 1.1 Propuesta de Taxonomía Unificada Basado en Masa y Tamaño (Rui & Xavier, 2019)

Clase			Subclase		Tamaño	
#	Nombre	Masa (kg)	Tipo	Masa (kg)		
3	Mini		Intermedio	[180 – 500[Small	
			Liviano	[100 – 180[
2	Micro	[10 – 100[Pesado	[60 – 100[
			Intermedio	[25 – 60[
			Liviano	[10 – 25[
1	Nano	[1 – 10[CubeSat	12U		[8 – 10[
				6U		[6 – 7.99[
				3U		[3 – 3.99[
				2U		[2 – 2.66[
				1U		[1 – 1.33[
			Otras formas	[1 – 10[

1.1.3 SmallSat

Según la taxonomía presentada previamente, los *SmallSats* corresponden a un grupo que encapsulan diferentes clases y subclases de satélites, cuyo rango de masa oscila entre los 1 y 500 kg. Históricamente hablando, su concepción como tal comenzó en *NASA Ames Research Center*, con el lanzamiento de las sondas *Pioneer 10* y *11*, lanzadas en 1972 y 1973 respectivamente como se aprecia en la Figura 1.2, que pesaban menos de 600 kg (Yost & Weston, 2023). Posterior a ello, para abarcar el constante crecimiento de masa lanzado y al costo asociado, con la alta cadencia de lanzamiento, NASA inició el Programa *Explorers* en 1988 para motivar el desarrollo de *SmallSats* con masa entre ~60 - 350 kg, extendiéndose a misiones lunares en las décadas posteriores (Yost & Weston, 2023).



a) Integración de la Sonda Pioneer 10 (NASA Office of Communications, 1972)



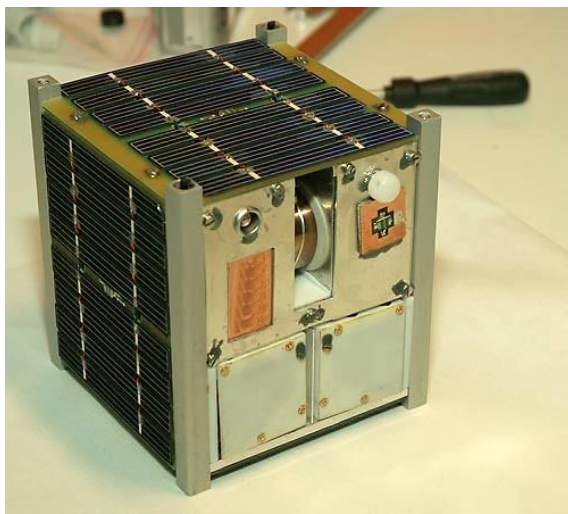
b) Integración de la Pioneer 11 a su *fairing* (John Uri, 2019)

Figura 1.2 Sondas Pioneer 10 y 11

Este decrecimiento en masa y aumento de capacidades científicas y técnica catalizó el interés por la miniaturización y maduración de tecnologías aeroespaciales a finales del siglo XX, que probaron ser capaces de producir misiones a menor costo. También, la disponibilidad de componentes y plataformas satelitales integradas en sus propios subsistemas ha aumentado significativamente desde 2010 debido a la alta demanda y por la utilidad de los *SmallSats* (Yost & Weston, 2023). En consecuencia, desde 2010 a la fecha ha resultado en la abundancia de buses ensamblados COTS, donde una performance con un costo conocido puede ser entregado regularmente. Ahora el diseño satelital ya no comienza desde cero, sino que puede ser integrado y ensamblado con mayor rapidez y seguridad, acelerando la industria comercial *SmallSat*. Dos opciones principales han emergido del mercado de bus para *SmallSats* en la última década: soluciones para bus claves de vendedores especializados en *CubeSats*, que ofrecen “ofertas” de subsistemas integrados según los requerimientos del cliente, teniendo un portafolio suficientemente diverso de subsistemas; y soluciones de unidades *bus* COTS para plataformas listas para integrar que el cliente adquiere, integra y prueba por su propia cuenta.

(1) *CubeSat*

El estándar *CubeSat*, categoría dentro del grupo de los *SmallSats* (Moe & Sullivan, 2017), fue concebido en 1999 colaborativamente entre la Universidad Politécnica Estatal de California (Cal Poly) y el Laboratorio de Sistemas Satelitales y Entorno Espacial (SSDL) de la Universidad de Stanford, lanzado por primera vez en 2003 por Dr. Jordi Puig-Suari y el Dr. Bob Twiggs (The CubeSat Program, 2022). La cualidad distintiva de los *CubeSats* es que adoptan un tamaño y factor de forma estandarizado como aparece en la Figura 1.3, cuya unidad se define en “U”. Un 1U corresponde a un cubo de 10 cm con una masa de hasta 2 kg, siendo éstos escalables desde 0.5U hasta 27U (Hevner et al., 2011). Representaciones gráficas se pueden observar en la Figura 1.4.



a) Ncube-2, misión noruega compuesta por un *CubeSat* 1U (Bjørn Pedersen, NTNU)



b) Suchai 2, *CubeSat* 3U desarrollado por la Universidad de Chile (La Tercera)

Figura 1.3 Ejemplos de *CubeSats* con diferentes números de unidades (varios autores)

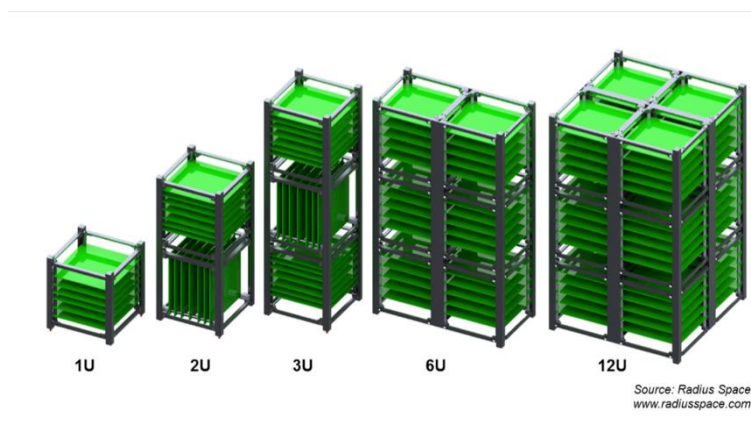


Figura 1.4 Representación gráfica de distinto tamaños estandarizados de *CubeSats* (Radius Space)

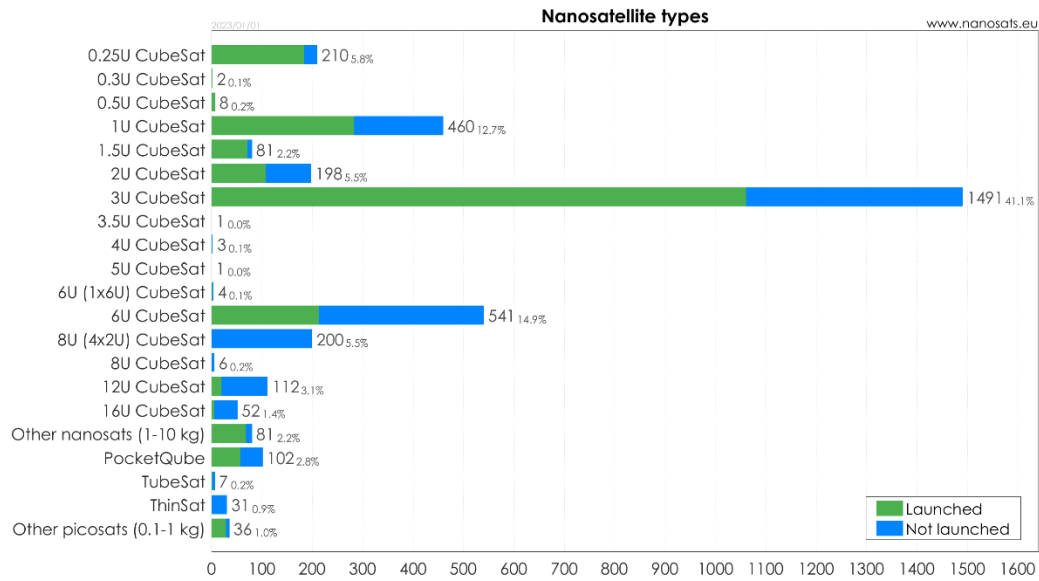


Figura 1.5 Nanosatélites lanzados y por lanzar según organización (Kulu, 2023)

Cabe destacar que, debido a su fácil acceso y bajos requerimientos para los usuarios y diseñadores, las unidades 1U, 3U y 6U han adquirido mayor popularidad, como se observa en la Figura 1.5.

A consecuencia de ello, el estándar *CubeSat* ha adquirido fama en las últimas décadas, permitiendo que nuevas instituciones se integren al área espacial, especialmente la industria y la academia como se aprecia en la Figura 1.6. No sólo eso, su irrupción en la industria, junto con nuevas tecnologías y la miniaturización, vino a la par de un movimiento económico-privado denominado NewSpace, cuyo ethos es desafiar las maneras tradicionales de la exploración espacial y su conservadurismo (Prasad, 2017).

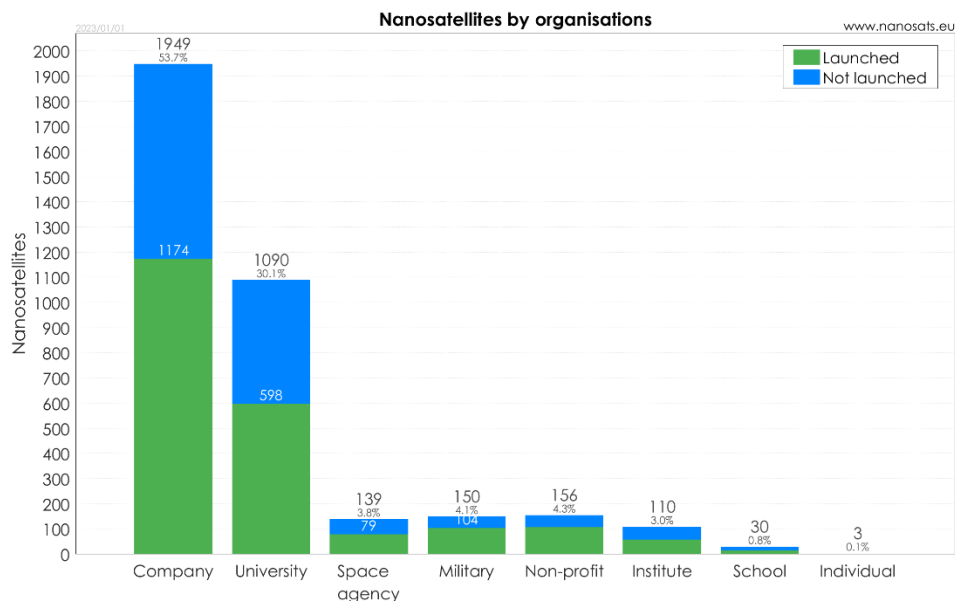


Figura 1.6 Tipos de nanosatélites lanzados y planificados hasta 01/01/2022 (Kulu, 2023)

(2) Subsistemas

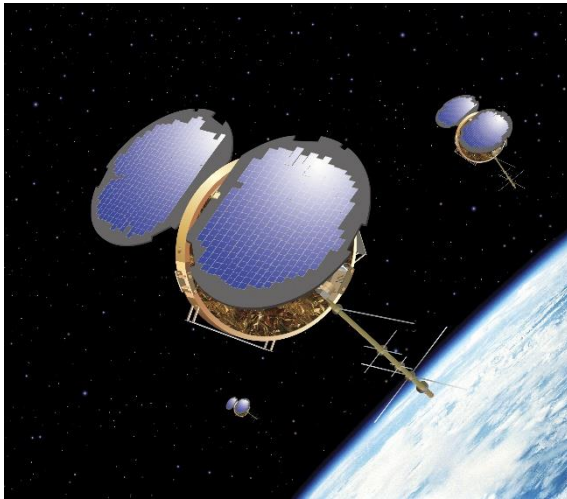
Un elemento espacial, el *CubeSat*, al igual que otros satélites, puede ser descompuesto en dos unidades críticas: el *payload* y el *bus*. El *payload* corresponde a los subsistemas del elemento que llevan consigo los instrumentos necesarios para la misión, como son cámaras, sensores, instrumentos científicos. Mientras que el *bus* compila todos los otros subsistemas esenciales para su correcto funcionamiento en la vida operacional. A continuación, se presenta la clasificación de subsistemas acorde a la definición entregada por la NASA State of the Art, reporte anual que compila y actualiza los avances en el área *SmallSat* (Yost & Weston, 2023).

- *Power*: El sistema de energía eléctrica (EPS) abarca la generación, almacenamiento y distribución de energía eléctrica. El EPS es un subsistema importante y fundamental, y comúnmente comprende una gran porción del volumen y la masa en cualquier nave espacial dada.
- *In-Space Propulsion*: Subsistema que permite la capacidad del elemento espacial para controlar su trayectoria y órbita que se encuentra. Las tecnologías de propulsión de naves espaciales pequeñas en el espacio generalmente se clasifican como químicas, eléctricas o sin propelente.
- *Guidance, Navigation and Control*: Compila tanto el subsistema de *Guidance, Navigation and Control (GNC)* y el *Attitude Determination and Control System (ADCS)*. El primero obtiene la posición del objeto mediante el uso de GPS y/o radares en tierra conectados a un propagador de trayectoria, mientras que la segunda estima y controla la actitud de éste por medio de sensores y actuadores dentro del *SmallSat*.
- *Structures, Materials and Mechanisms*: Compila a las estructuras que encapsula al *SmallSat*, sus mecanismos de despliegue y sus métodos de manufactura y selección de materiales. Es un subsistema que engloba y depende de los requerimientos de los demás para su diseño.
- *Thermal Control*: Corresponde a mecanismos que permiten el control de temperatura dentro del *SmallSat*, dependiendo de sus requerimientos de funcionamiento. Existen lo que son los sistemas pasivos (recubiertas, *thermal straps*, *heat pipes*, etc) y sistemas activos (*electrical heaters*).
- *Small Spacecraft Avionics (SSA)*: se describen como todos los subsistemas, componentes, instrumentos y elementos funcionales electrónicos incluidos en la plataforma de la nave espacial. Estos incluyen principalmente subelementos de vuelo de comando y manejo de datos (CDH), software de vuelo (FSW) y otros subsistemas de vuelo críticos, incluida la carga útil y los subsistemas de aviónica (PSA).
- *Communication*: el sistema de comunicación permite a la nave espacial transmitir datos y telemetría a la Tierra, recibir comandos de la Tierra y transmitir información de una nave espacial a otra. Un sistema de comunicaciones consiste en el segmento terrestre: una o más estaciones terrestres ubicadas en la Tierra, y el segmento espacial: una o más naves espaciales y sus respectivas cargas útiles de comunicación. Las tres funciones de un sistema de comunicaciones son recibir comandos desde la Tierra (enlace ascendente), transmitir datos a la Tierra (enlace descendente) y transmitir o recibir información de otro satélite (enlace cruzado o enlace entre satélites).

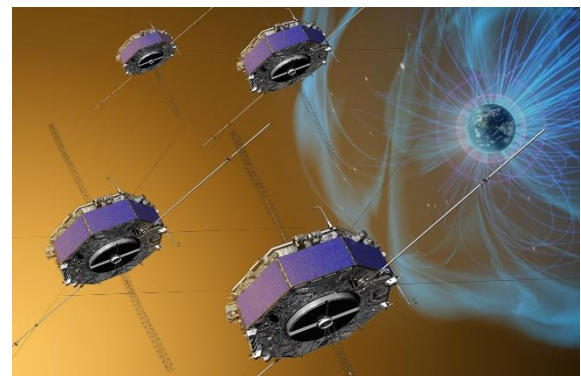
1.1.4 Distributed Spacecraft Mission

Al apilarse varios elementos bajo una misma misión, aparece el concepto segmento espacial de *Distributed Spacecraft Missions* (DSM), que son misiones que involucran múltiples satélites para lograr uno o varios objetivos comunes o distintos. Esta definición generalizada, lejos de asentar su concepto adrede, no especifica cualidades del sistema (le Moigne et al., 2020); por ende, es necesario generar una taxonomía de la definición entregada que clasifique estas características. Le Moigne, J; Adams, J y Nag, S detallan expresamente los elementos en el Anexo 1, concordante también con los alcances obtenidos por Cote et al. (Cote et al., 2018). Por medio de esta taxonomía se define a los *clusters* dentro de la categoría de relación espacial 2.1 del Anexo 1; como una colección de naves que no se encuentran uniformemente distribuidas en una región espacial particular, en contraste a las constelaciones Walker (distribución equitativa de un conjunto de cuerpos celestes). Se apilan en ciertas regiones debido a una necesidad en común, como los mostrados en la Figura 1.7. Para este trabajo se utilizará esta taxonomía para definir en mayor alcance las características del *cluster* a modelar.

1.1.5 Observación Terrestre



a) *Constellation Observing System for Meteorology Ionosphere & Climate (COSMIC)*. Constelación tipo cluster de 6 *SmallSats* que estudian la atmósfera Terrestre. (Brian Dunbar, 2014)



b) *Magnetospheric Multiscale (MMS) Mission*. Constelación tipo cluster de 4 satélites idénticos diseñados para el estudio de la magnetosfera terrestre. (Bridgman & Fox, 2016)

Figura 1.7 Ejemplos de constelaciones tipo cluster desplegadas

La Observación Terrestre (EO) es la recopilación de datos sobre los sistemas físicos, químicos y biológicos del planeta Tierra vía tecnologías de percepción remota, como son los satélites con cámaras de observación. Estos últimos entregan sets de datos confiables y repetibles que, en conjunto con la investigación y desarrollo de métodos de análisis apropiados, proveen una manera única de obtener información del planeta. Ejemplos incluyen el monitoreo del estado y evolución del medioambiente, sea por tierra, mar o aire, y la capacidad de asistir situaciones durante crisis climáticas o conflictos humanos (*Earth Observation*, 2022).

Hasta 2022 se han contabilizado 5 465 satélites actualmente operativos, donde 1 130 de ellos son satélites de EO (Union of Concerned Scientists, 2022) donde el 52,2% de ellos son para uso comercial, siendo Planet Labs y Spire Global que concentran la mayor cantidad de satélites EO en órbitas *Low Earth Orbit* (LEO), 197 y 114 respectivamente. Ambas compañías mencionadas anteriormente utilizan constelaciones de satélites tipo *CubeSat* 3U dedicados a la EO como se aprecia en la Figura 1.8.



a) Low Earth Multi-Use Receiver (LEMUR) 3U
CubeSat de Spire Global (Spire Global)



b) Despliegue de satélites Dove de Planet Labs
(Planet Labs)

Figura 1.8 Representaciones gráficas de CubeSats 3U usadas para EO (varios autores)

Muchos satélites EO transportan consigo instrumentos que deben ser operados a alturas relativamente bajas, sobre los 500 y 600 km, correspondiente al 94.3% de satélites EO lanzados en la actualidad (Union of Concerned Scientists, 2022). Los requisitos de operación, el diseño y la cobertura solicitada condicionan las órbitas a la que se encontrarán los elementos. Para uno o un *cluster* en configuración espacial Swarm, se utiliza órbitas polares o *Sun-Synchronous Orbit* (SSO) para una cobertura global, como es el satélite FASat Charlie, o si es una constelación de satélite se utilizan las constelaciones Walker o una mezcla o personalizada, como son los casos de Planet Labs y Spire Global.

1.1.6 Fallas en Satélites

La falla de los satélites en órbita, o de uno de sus subsistemas, pueden afectar significativamente las capacidades que los sectores comerciales, defensa y civil poseen. Las anomalías en órbita son eventos degradantes de la misión recurrentes en la vida operacional de un elemento. Todo cuerpo celeste artificial presenta anomalías en cualquier momento de su vida operacional, con severidad desde una degradación gradual de la eficiencia de los paneles solares y batería hasta el cese de comunicaciones y la falla catastrófica (Galvan et al., 2014).

Las causas de las fallas se deben tanto al ambiente espacial (radiación solar, *Single Event Upset* (SEU), entre otros), como al error humano (falta de experiencia, error de diseño, error de operación, entre otros) (Kiselyov, 2020). Por ende, se ha vuelto imperante la necesidad de una herramienta que permita medir la calidad de un objeto a lo largo del tiempo (Billinton & Allan, 1992); por tanto, el concepto de confiabilidad aparece como una variable dependiente del tiempo y concierne a las fallas posibles dentro de ese dominio. De esta forma, se genera una diferencia sustancial en lo que respecta a control de calidad -la mantención y monitoreo de capacidad de un sistema a la hora ejecutar la acción para la que fue hecho- e ingeniería en confiabilidad, ameritando su propio estudio.

1.1.7 Confiabilidad

La **confiabilidad** se define como la probabilidad de que un ítem realizará una función requerida sin falla, en determinadas condiciones de operación y por un tiempo definido (Billinton & Allan, 1992). También se puede establecer como el número de fallas dentro de un periodo de tiempo. La confiabilidad como concepto estimable debe entenderse como una probabilidad en la que cualquier intento por cuantificarla involucrará el uso de métodos estadísticos. Para elementos no reparables, se define como la probabilidad de supervivencia sobre la vida esperada de ítems no reparables (como ampollitas, transistores o satélites) (Billinton & Allan, 1992).

La definición establecida se puede separar en 4 secciones: probabilidad, performance, tiempo y condiciones de operación. La probabilidad provee la entrada numérica para el tratamiento de la confiabilidad y usualmente se la considera la más relevante, referente a los registros históricos para su análisis. Para el resto de las secciones se analizan como parámetros ingenieriles, que dependen del diseño y condiciones externas, donde la teoría de probabilidad no es útil o solamente complementa el diseño (Billinton & Allan, 1992). Para estas últimas secciones, la confiabilidad más que a un valor estadístico, responde a una metodología de diseño y fabricación (O'Connor & Kleyner, 2012).

También la **durabilidad** se define como la habilidad de soportar los efectos del tiempo (distancia recorrida, ciclos de operación, etc.) en mecanismos tales como la fatiga, corrosión, cambios en parámetros eléctricos, etc. También se expresa como el tiempo mínimo antes de la ocurrencia de fallos de desgaste; la durabilidad puede ser una métrica de medición referente a los parámetros

ingenieriles mencionados previamente. Para elementos no reparables, se traduce cuantitativamente como *Mean Time To Failure* (MTTF), siendo el tiempo medio hasta la falla.

1.2 Planteamiento del Problema

Uno de los problemas principales que los *CubeSats* presentan a la hora de implementarse en un programa, es su alta tasa de falla y, por ende, baja confiabilidad. Una de las aristas de una posible causa investigada es la institución desarrolladora de la misión, donde Swartwout en 2017 establece que el 25% de las misiones de *CubeSats* desarrolladas por universidades están *Dead-On-Arrival* (DOA), es decir, que llegan al espacio con falla catastrófica (Swartwout & Jayne, 2017). El autor también concluye que las instituciones que construyen más satélites reflejan un aumento en el éxito de las misiones, declarando basado en su propia experiencia que “los proyectos dirigidos por estudiantes usualmente fracasan debido a una falta de tiempo y recursos entregada al testeado a nivel de sistema” (Swartwout & Jayne, 2017).

Mirando la problemática desde una perspectiva global, se requiere saber y poder determinar la confiabilidad de un *CubeSat* para efectos de diseño. Todo esto comienza con el levantamiento de datos y recopilación de anomalías en órbitas, fallas y eventos a misiones pasadas y actuales. Un reporte de NASA presentó un análisis estadístico en la confiabilidad de *CubeSats* (GSFC/Code 371, 2015). Por medio de una base de datos creada por Swartwout (actualmente fusionada con Seradata, empresa líder de recolección de datos y anomalías en órbita). De las 390 misiones analizados, solo 21 pudieron considerarse apropiadas (que la misión tiene suficientes datos) y completos (que todas las anomalías han sido registradas) para ser usadas. Se concluyó que la ausencia de data útil hace que la estimación práctica de confiabilidad se vuelva difícil (GSFC/Code 371, 2015).

e esa insuficiencia, J. Bouwmeester et al. realiza otro acercamiento por medio de la inferencia bayesiana, método estadístico que infiere las características de una muestra a partir de un conjunto mayor que lo engloba, donde la base de datos de fallas *SmallSats* puede ser utilizada para proveer una distribución a priori para *CubeSats*, basado en el hecho que estos últimos son un subconjunto de *SmallSats* (Bouwmeester et al., 2022). Los resultados se pueden apreciar en la Figura 1.9, donde se concluye que la curva producto Lognormal-Gompertz es la que se asemeja más a los datos obtenidos. Se observa a su vez el desplome de confiabilidad en los primeros 2 años de operabilidad, llegando a ~58% pasado ese tiempo.

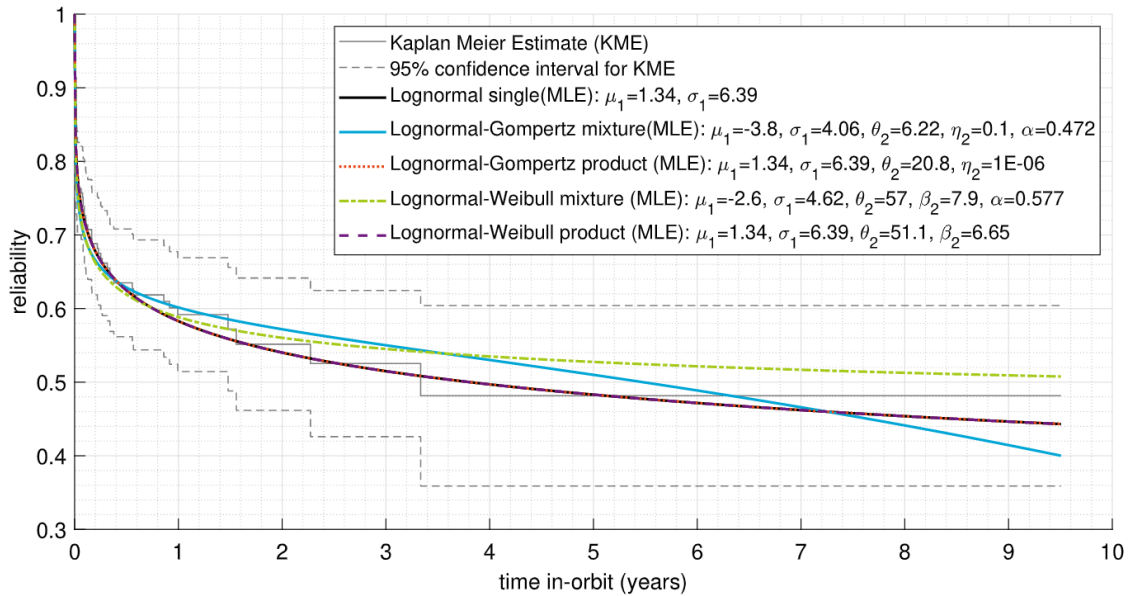
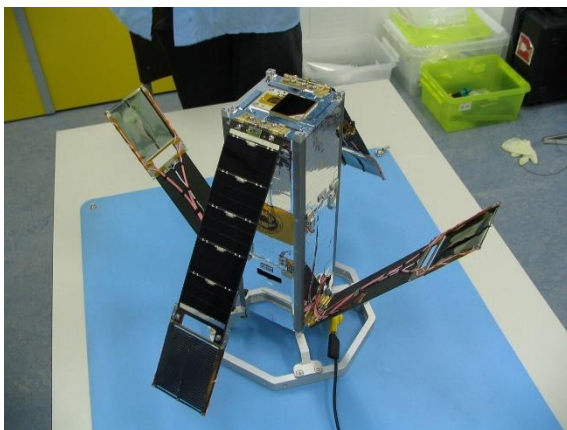
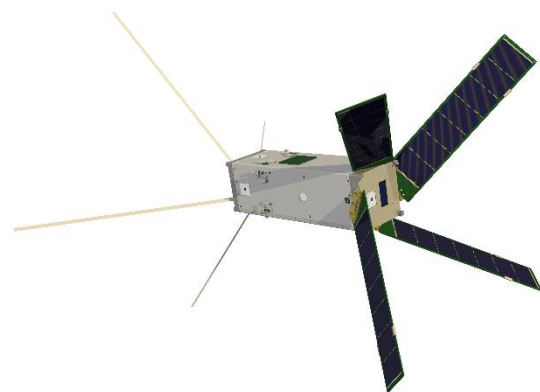


Figura 1.9 Curva de Confiabilidad de CubeSats (Bouwmeester et al., 2022)

Una aproximación común para mejorar la confiabilidad en satélites de mayor envergadura es la aplicación de redundancia a subsistemas críticos. Sin embargo, el limitado volumen comúnmente restringe la implementación acabada de redundancia en el sistema (Bouwmeester et al., 2022). Delfi-C3, un *CubeSat* 3U de la Universidad Tecnológica de Delft (TUDelft) como se muestra en la Figura 1.11, implementó redundancia por medio de un sistema *backup* para el C&DH de uno de sus *payload* (Bouwmeester et al., 2008).



a) Delfi-C3



b) Delfi-N3XT

Figura 1.10 Ejemplos de aplicación de redundancias en CubeSats para subsistemas críticos (TUDelft)

Aunque no pudo probarse a falta de tiempo, su sucesor como se muestra en la Figura 1.10, utilizó la filosofía de diseño sin *single-point-of-failure* (eventos que encadenan una falla catastrófica) para aplicar redundancias en sus subsistemas más críticos. Experiencias como estas son una motivación clave para investigar el beneficio de esta estrategia. En contraste con la relativa baja confiabilidad de la plataforma, de acuerdo con Villela et al. realizó un extenso estudio en 2019 en 855 misiones *CubeSats* (Villela et al., 2019), concluyendo que la tasa de éxito de misión ha ascendido desde un 30% en 2005, balanceándose en un 75% en 2018 como aparece en la Figura 1.11. Se concluye también que la *infant mortality* (falla temprana o de despliegue) ha sido dominante como se aprecia en la Figura 1.12 (Villela et al., 2019).

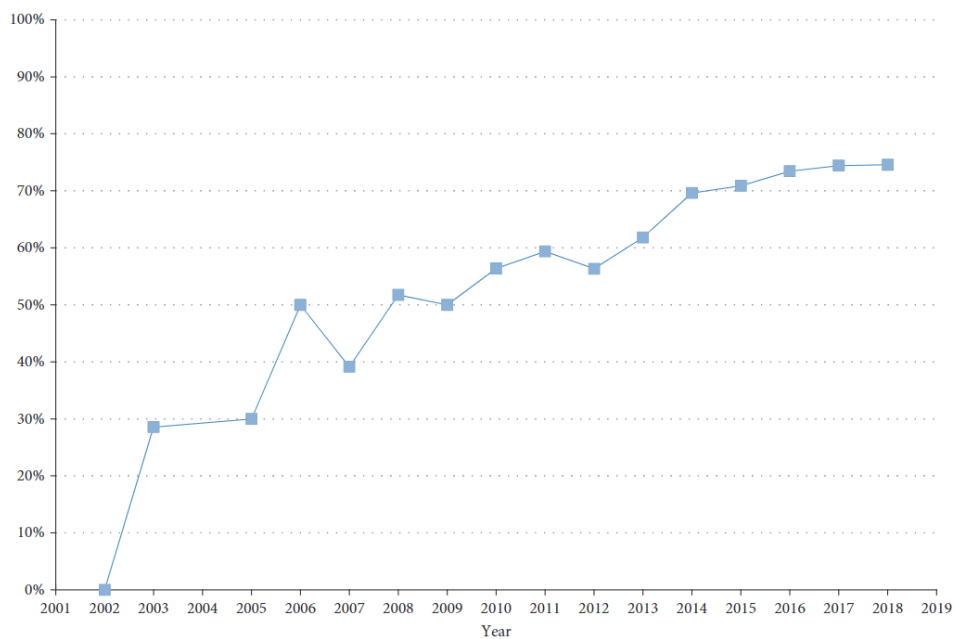


Figura 1.11 Tasa de éxito de misiones CubeSats (Villela et al., 2019)

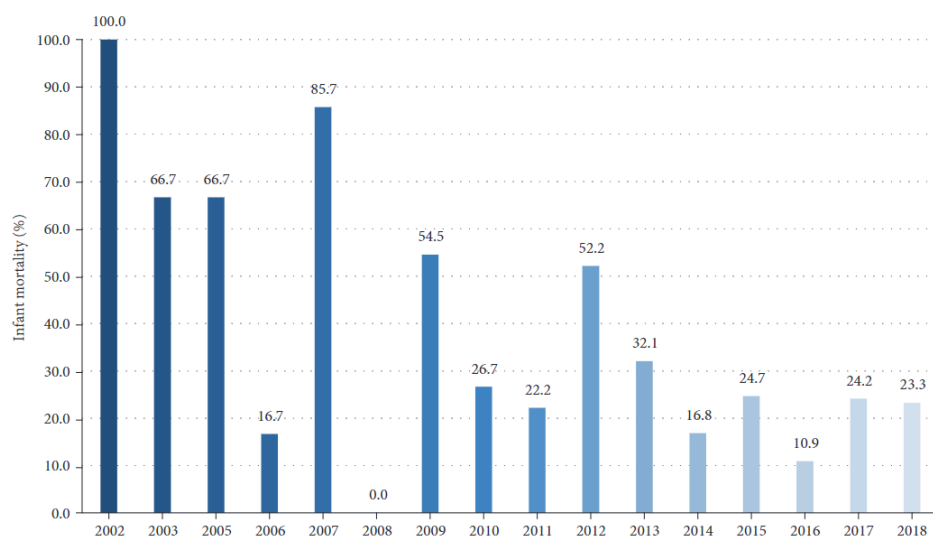


Figura 1.12 Infant Mortality de CubeSats desde 2002 hasta el 31 de mayo, 2018 (se excluyen fallas de lanzamiento) (Villela et al., 2019)

El aumento de la tasa de éxito, sumado a todo lo mencionado anteriormente, ha atraído últimamente la atención de agencias tradicionales e industrias pujantes a usarlos como plataforma de EO, por medio de programas propios como la *European Space Agency* (ESA) con FSSCat (NanoSats Lab, 2020), el sector privado como los mencionados en Figura 1.8 o una colaboración público-privada como fue la reciente expansión del contrato entre Spire Global y NASA para data de EO (News Desk, 2022). Esto debido a la mayor demanda de datos e imágenes del suelo terrestre, así como la rapidez de desarrollo, producción y lanzamiento de estos, clave para proveer mediciones regulares y precisas, incluyendo áreas difíciles de llegar como las zonas polares. Esto se refuerza por lo establecido por la Agencia Espacial Brasileña (AEB) en la Figura 1.13, exponiendo el gran interés que tiene el sector público y la academia de la observación terrestre en resolución media. Lo más remarcable es que mayoritariamente requieren resoluciones de 20 metros o menos, con una tasa de refresco de 30 días o menos. Este tipo de requerimientos es muy difícil de lograr con un satélite monolítico.

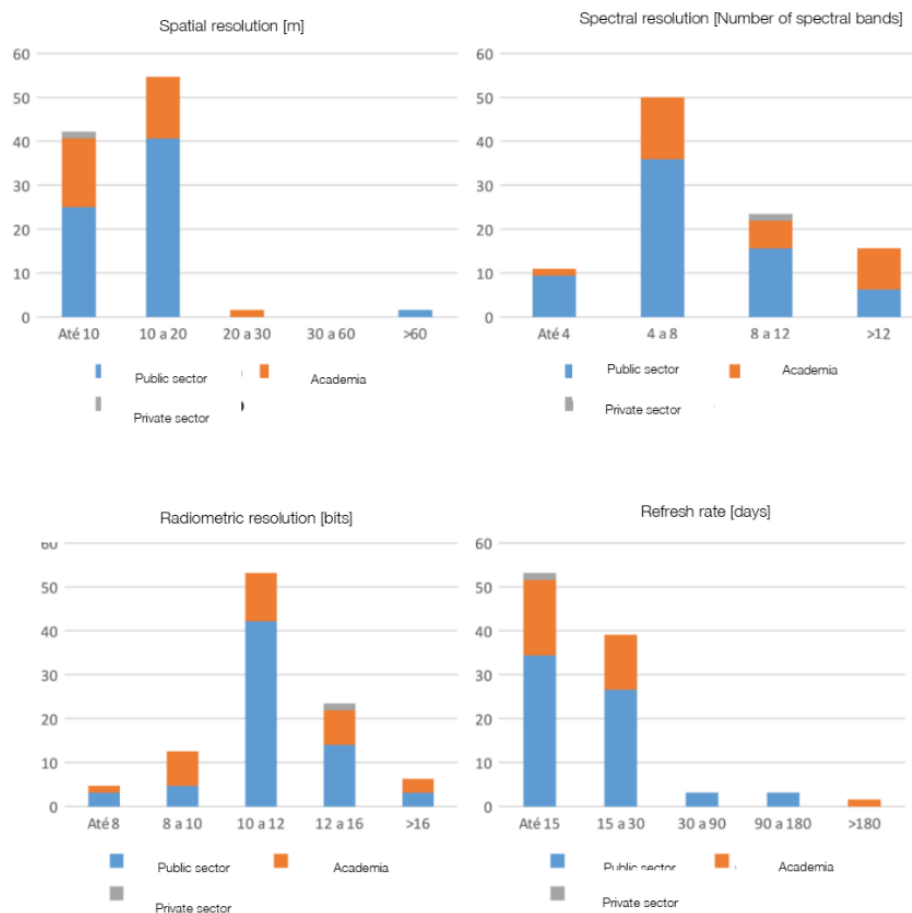


Figura 1.13 Demandas Nacionales del Sector Espacial 2019 (AEB)

Ante esa situación, los DSM aparecen como una manera de resolver el uso de *CubeSats* para misiones EO y mejorar la confiabilidad total del sistema espacial, al poder desplegar más elementos de los requeridos. Esta opción por estudiar sin embargo abre más preguntas. ¿Se puede estimar la confiabilidad de un *cluster* de *CubeSats* a partir de sus curvas de confiabilidad? ¿Cuánto

afecta la confiabilidad y *System Engineering (SE) Envelopes* (costo, masa y volumen) de un *cluster* al implementarle estrategias de redundancias a *CubeSats*?

En el presente trabajo se dará continuidad al trabajo realizado en el Proyecto de Ingeniería Aeroespacial (PIA) denominado: Confiabilidad en *CubeSats*, construyendo una arquitectura que simule y estime la confiabilidad y sus *Systems Engineering Envelopes* para unidades *bus* y segmentos espaciales *clusters* de *CubeSats*, al aplicarle estrategias de redundancia y mitigación de riesgos comparando los resultados con el trabajo de Qaisar et al (Qaisar et al., 2016).

1.3 Objetivos

1.3.1 Objetivo General

Estimar la confiabilidad y *Systems Engineering Envelopes* para unidades *bus* y segmentos espaciales *clusters* de *CubeSats* de diferentes configuraciones por medio de modelado, simulación y análisis de estrategias de redundancia y mitigación de riesgos.

1.3.2 Objetivos Específicos

- Determinar los parámetros y componentes necesarios para estimar la confiabilidad y *SE Envelopes* de subsistemas de un *cluster* compuesto por *CubeSats*, del *CubeSat* como tal y sus subsistemas.
- Diseñar la arquitectura definitiva que traduzca las estrategias de mitigación de falla en la simulación para la obtención de confiabilidad y *System Engineering Envelopes*, utilizando como referencia la propuesta inicial planteada en el PIA y el Estado del Arte.
- Simular una misión EO de 5 a 7 años por medio de un *cluster* de *CubeSats* (Torres, 2017).
- Evaluar los resultados de la simulación con casos de estudio.

1.4 Metodología

Primero, se realizará un estudio bibliográfico que complemente lo establecido en el trabajo previo (PIA) en materia de estrategias de mitigación de riesgo y redundancias aplicables. Luego de generar un levantamiento de información recopilando todos los elementos necesarios para su implementación, tales como las curvas de confiabilidad de *CubeSats*, los *SE Envelopes* de cada subsistema a partir de datos de proveedores, entre otros; se diseñará la arquitectura definitiva,

utilizando como referencia inicial la propuesta inicial planteada en el PIA y cualquier otro modelo diseñado previamente que se haya encontrado en el Estado del Arte.

Posterior a ello, una vez definida la arquitectura definitiva y sus requerimientos, se procederá a la construcción de ésta. Una vez construido se simulará una misión de EO equivalente de 5 a 7 años, siendo la vida promedio de diseño satélite monolítico de observación, por medio de un *cluster* de *CubeSats* en distintas configuraciones de mitigación de confiabilidad.

CAPÍTULO 2: Marco Teórico

En este capítulo se resumirá brevemente la terminología que enmarca las teorías y métodos de obtención de curvas de confiabilidad que se expondrán en el Estado del Arte. Se definirán las expresiones matemáticas de confiabilidad y tasa de falla, así como su comportamiento y cómo en la práctica la falla en sistemas se expresa a lo largo del tiempo

2.1 Terminología

Se define matemáticamente que la confiabilidad y la probabilidad de falla son 2 fenómenos complementarios uno del otro.

$$R(t) = 1 - Q(t) [-] \quad (1)$$

La derivada de una función distribuida cumulativa de una variable aleatoria es la probabilidad de densidad. En evaluación de confiabilidad, $Q(t)$, entrega la función de densidad de falla $f(t)$, donde

$$f(t) = \frac{dQ(t)}{dt} = -\frac{dR(t)}{dt} [-] \quad (2)$$

O

$$Q(t) = \int_0^t f(t)dt [-] \quad (3)$$

Y

$$R(t) = 1 - \int_0^t f(t)dt = \int_t^{\infty} f(t)dt [-] \quad (4)$$

Durante la vida del objeto no reparable, a la probabilidad instantánea de la única falla catastrófica se le denomina tasa de riesgo o *hazard rate*, designado $\lambda(t)$. En términos de falla, la tasa de fallo es una métrica, valga la redundancia, que describe la tasa a la que las fallas ocurren en el sistema. Esta variable es dependiente tanto del número de fallas en cierto periodo de tiempo y el número de componentes expuestos a la falla.

$$\lambda(t) = \frac{\text{número de fallas por unidad de tiempo}}{\text{número de componentes expuestos a la falla}} [-] \quad (5)$$

En (Billinton & Allan, 1992) se relaciona matemáticamente a la confiabilidad con la tasa de fallo de la siguiente manera, siendo esta descripción independiente de cualquier función o forma que describa a la falla misma.

$$R(t) = \exp \left[- \int_0^t \lambda(t)dt \right] [-] \quad (6)$$

Usualmente la tasa de falla sigue un comportamiento llamado *bathtub curve*, como se observa en la siguiente Figura 2.1. Esta curva se divide en tres secciones, la primera correspondiente a la mortalidad infantil, provocada por fallas de diseño, humanas o eventuales; seguido de un periodo constante, cuya falla principal es la provocada aleatoriamente; terminando con el ascenso por el *wear-out*, provocado por el desgaste de los componentes.

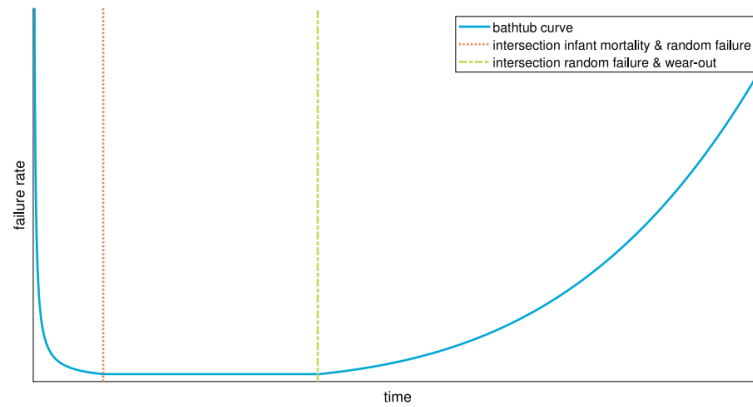


Figura 2.1 Bathtub Curve (Bouwmeester et al., 2022)

Se define el *mean time to failure* (MTTF) como el tiempo esperado a la falla de un sistema no reparable, siendo m el MTTF

$$m = \int_0^{\infty} R(t)dt = \int_0^{\infty} tf(t)dt \quad [-] \quad (7)$$

Es una herramienta útil y muy utilizada en la industria para evaluar el estado del sistema y estimar el éxito o no de los objetivos.

CAPÍTULO 3: Estado del Arte

En este capítulo se hace un resumen de trabajos previos relacionados con métodos de cálculo y estimación de confiabilidad, tanto para componentes y elementos; métodos de diseño para proyectos *SmallSats*. Esto, con tal de identificar los elementos de mayor relevancia que sirvan para la construcción de la arquitectura definitiva a desarrollar en los capítulos siguientes.

3.1 Diseño de Arquitecturas y SE Envelopes

3.1.1 A Framework for Small Satellite Architecture Design (Qaisar et al., 2016)

Los autores propusieron una arquitectura sistemática para la toma informada y óptima de decisiones en lo que respecta al diseño de *SmallSats*. Como se ilustra en la Figura 3.1, el usuario ingresa los elementos relevantes de la arquitectura del sistema, sus restricciones y SE *Envelopes*. Aplicaciones y herramientas reciben este *input*, consultan bases independientes de componentes COTS, produciendo como un resultado un *tradespace* visual que contiene las soluciones percibidas y no intuitivas. El proceso se vuelve iterativo hasta encontrar un óptimo que satisfaga los requerimientos del proyecto.

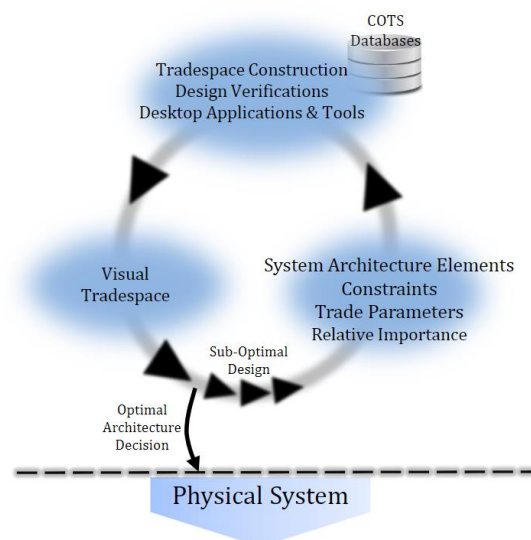
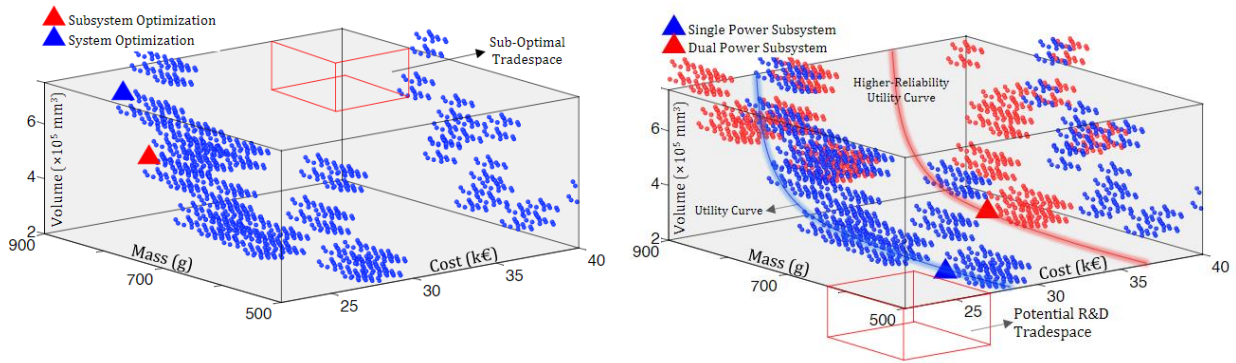


Figura 3.1 Arquitectura propuesta para el diseño de *SmallSats* (Qaisar et al, 2016)

El *tradespace* es construido tomando todas las combinaciones de una alternativa COTS a la vez. Luego los resultados son obtenidos mediante la adición de los valores de cada alternativa escogida. La optimización de estos resultados dependerá de los requerimientos del proyecto. En la Figura 3.2 se observa el ejemplo realizado por los actores que, en a) se observa el *tradespace* inicial y en b) el actualizado, se actualiza mediante el uso de un requerimiento que refine la toma de decisiones. En este caso es el uso de 2 *Power Systems* (componente del EPS) para efectos de redundancia.



a) Tradespace inicial según la base de datos del autor

b) Tradespace final con condición “Dual Power Subsystem”

Figura 3.2 Ejemplo de Tradespace realizado por el autor (Qaisar et al, 2016)

3.1.2 Mathematical Models for The Study of The Reliability of Systems (Kaufmann et al., 1977)

Según el autor, el término “redundancia” es uno general usado cuando una pieza o componente pueda reemplazar a otro que ha fallado. Cualquier estructura que no sea una en serie puede considerarse redundante en el sentido que hay presencia de componentes fallidos y sin embargo el elemento no falla. Por el contrario, una estructura en serie no tiene redundancia por sí misma.

En un estilo particular, la redundancia es la técnica para aumentar la confiabilidad de un sistema, donde se adosan en paralelo dos o más componentes, subestructuras o dispositivos completos. Estos pueden separarse en dos tipos de redundancia:

Redundancia activa: Todos los componentes del elemento funcionan permanentemente incluso si no es necesario.

Redundancia pasiva: la redundancia pasiva consiste en establecer ciertos componentes como “en reserva”, que no sean usados mientras no sean requeridos. Ante la presencia de una falla tolerable dentro del sistema, estos se pondrán en servicio, cambiando circuitos o reemplazando el componente en su totalidad. Uno generalmente supone que los componentes usados nos están siendo sujetos a deterioro, por ende la tasa de falla de estos depende de si estos han o no han sido puestos en servicio, dejándolos en un tiempo de operación distinto al del elemento en su totalidad.

(1) Redundancia al costo mínimo

Para este medio los autores utilizaron un método secuencial, propuesto inicialmente por Barlow y Proschan y posteriormente modificado levemente por ellos, que permite la búsqueda de un óptimo entre confiabilidad del sistema y el costo asociado al uso de componentes redundantes. Se hace definiendo un coeficiente para cada subsistema que compara la diferencia de confiabilidad entre el componente y el elemento total y el costo por la redundancia extra aplicada.

$$\mu_i = \frac{1}{c_i} [\log r(k_i, \dots, k_i + 1, \dots, k_n) - \log r(k)] [-] \quad (9)$$

Donde μ_i es el coeficiente del i-componente por optimizar, c_i es el costo por i-componente redundante, $r(k_i, \dots, k_i + 1, \dots, k_n)$ la confiabilidad del componente al aplicarle redundancia y $r(k)$ la confiabilidad del elemento. Esto permite un método comparable entre variables relevantes en el diseño y la confiabilidad del elemento en su totalidad, y como estos interactúan entre sí, claves para el diseño preliminar de cualquier sistema.

3.2 Confiabilidad de *CubeSats*

3.2.1 Practical Reliability Engineering (Billinton & Allan, 1992)

La confiabilidad como métrica depende sobre qué objeto se analiza, a su vez, este mismo no actúa en solitario, sino muchas veces forma parte de otros objetos aledaños cuya interacción suele ser simple o compleja. La interacción entre ellos puede beneficiar o afectar la confiabilidad total del sistema armado, por lo cual es de suma relevancia a la hora de realizar diseños preliminares. En otras palabras, la confiabilidad de los componentes efectivamente afecta la confiabilidad o durabilidad del sistema de mayor nivel. Sin embargo, la descomposición de sistemas en sus componentes y su análisis de confiabilidad sistémica desde ese extremo no es tan directo. Esto se debe a que, en la práctica, la confiabilidad y probabilidad de falla de componentes no son constantes, siendo estas variantes en el tiempo.

Cabe destacar que existen otros métodos no lógicos de analizar la confiabilidad de un sistema, dependiendo del enfoque que uno quiera dar. Un acercamiento condicional permite establecer estados en que el sistema puede encontrarse ante un evento; el método *cut set*, árboles de eventos y de fallas permite rastrear e identificar las formas y el cómo un sistema falla y, por ende, determinar qué confiabilidad se presenta ante tales eventos, entre otros. Se escogen los sistemas lógicos por su fácil implementación y comprensión, así como admite distribución de probabilidades en las fallas que puedan presentarse.

(1) Sistemas en Serie

El sistema en serie considera dos componentes independientes A y B conectados en serie, desde un punto de vista de confiabilidad, mostrado en la Figura 3.3.

Este arreglo implica que ambos equipos tienen que funcionar para que el éxito del sistema esté asegurado y debido a que el éxito y la falla son mutuamente exclusivos y complementarios, tenemos lo siguiente:

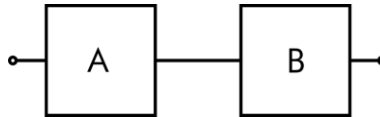


Figura 3.3 Diagrama de sistema en serie con 2 componentes (autoría propia)

Se define R_A, R_B como las probabilidades de éxito de los componentes A y B respectivamente, y Q_A, Q_B las probabilidades de falla de ambos.

$$R_A(t) + Q_A(t) = 1 \text{ y } R_B(t) + Q_B(t) = 1 \text{ [-]} \quad (10)$$

Al ser el requerimiento que ambos componentes funcionen para que el sistema completo se encuentre operativo, se define la probabilidad de éxito del sistema o confiabilidad del sistema como:

$$R_S(t) = R_A(t) \cdot R_B(t) = \exp \left[- \int_0^t (\lambda_A(t) + \lambda_B(t)) dt \right] \text{ [-]} \quad (11)$$

Para un sistema de n componentes, la confiabilidad se describe generalizando la ecuación previa

$$R_S(t) = \prod_{i=1}^n R_i(t) = \prod_{i=1}^n \exp \left[- \int_0^t \lambda_i(t) dt \right] \text{ [-]} \quad (12)$$

Esto se conoce como la regla del producto de confiabilidad, debido a que establece que la confiabilidad de un sistema en serie es la multiplicación de la de cada componente.

(2) Sistemas en Paralelo

Se considera un sistema de dos componentes independientes A y B conectados en paralelo desde un punto de vista de confiabilidad como lo observado en la Figura 3.4.

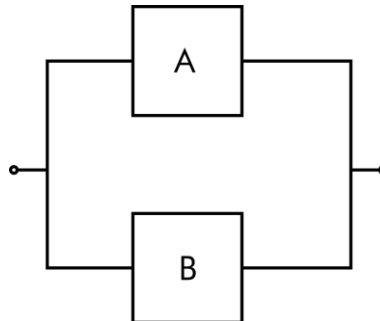


Figura 3.2 Diagrama de sistema en paralelo con 2 componentes (autoría propia)

Como se observa, el requerimiento del sistema es que solo uno de los componentes tiene que trabajar para el éxito de este. La confiabilidad del sistema se obtiene como el complemento de la falla del sistema, debido a que es “A o B”.

$$\begin{aligned} R_P(t) &= 1 - Q_A(t) \cdot Q_B(t) [-] \\ R_P(t) &= R_A(t) + R_B(t) - R_A(t) \cdot R_B(t)[-] \end{aligned} \quad (13)$$

También para un sistema de n componentes paralelos usando la terminología mencionada.

$$R_P = 1 - \prod_{i=1}^n \left(1 - \exp \left[- \int_0^t \lambda_i(t) \right] \right) [-] \quad (14)$$

A diferencia de lo visto en la subsección anterior, al incrementar la cantidad de componentes en paralelo de un mismo sistema la probabilidad de falla disminuye y, por ende, la confiabilidad del sistema incrementa en comparación al valor original. Sin embargo, cabe destacar que el incremento de componentes lleva consigo un aumento del costo inicial, el peso y el volumen del sistema, como también que al añadir más componentes a un sistema el incremento de la confiabilidad es mucho menor, tal como se observa en la Figura 3.5. Ya para el 6to elemento el beneficio es prácticamente nulo.

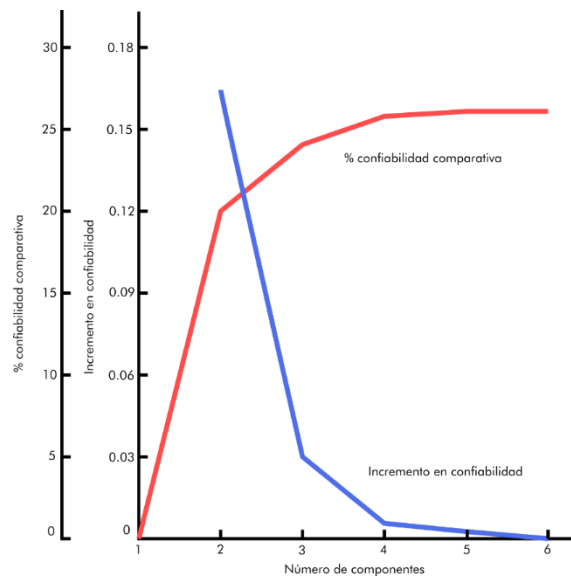


Figura 3.3 Gráfica comparativa de incremento de confiabilidad vs % confiabilidad comparativa (Billinton & Adam, 1992)

(3) Sistemas Serie-Paralelo

Los dos bloques primarios discutidos previamente conforman la base para analizar configuraciones más complejas, donde pueden ser mezclados dependiendo de la necesidad. El método utilizado para analizar tales sistemas se conoce como la técnica de reducción de redes y consiste en reducir secuencialmente la estructura en bloques en serie y en paralelo, hasta llegar a la confiabilidad de la macroestructura.

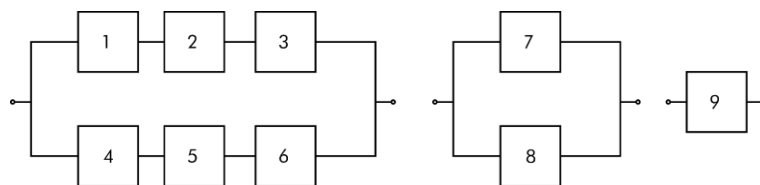


Figura 3.4 Diagrama de sistema serie-paralelo y su reducción para su análisis (autoría propia)

Se toma como ejemplo la siguiente Figura 3.4. Tenemos en cuenta que:

$$R_7(t) = R_1(t)R_2(t)R_3(t) \quad [-] \quad (15)$$

$$R_8(t) = R_4(t)R_5(t)R_6(t) \quad [-] \quad (16)$$

Y

$$R_9(t) = 1 - (1 - R_8(t))(1 - R_7(t)) \quad [-] \quad (17)$$

(4) Sistemas K-out-of-N

Los sistemas *K-out-of-N* o parcialmente redundantes, son sistemas intermedios entre los en serie y los paralelos. Esto se debe a que se establece como condición que del total de n componentes disponibles, solo k tienen que estar operativos para que el sistema no falle. Esta metodología se resuelve utilizando la distribución binomial donde, dependiendo de la cantidad de componentes requeridos para funcionar, la expresión de confiabilidad del sistema varía.

$$R_s(k, n, R) = \sum_{r=k}^n \binom{n}{r} R^r (1 - R)^{n-r}$$

La expresión mostrada anteriormente está condicionada a cuando los valores de confiabilidad de todos los componentes del sistema son iguales. Al ser de distinto valor, se le aplica otro método denominado *Event Space Method*, que es una aplicación del axioma de eventos mutuamente excluyentes. Se determinan todos los eventos mutuamente excluyentes y se consideran aquellos que resultan en el éxito del sistema. La fiabilidad del sistema es simplemente la probabilidad de la unión de todos los eventos mutuamente excluyentes que producen un éxito del sistema. Del mismo modo, la falta de fiabilidad es la probabilidad de la unión de todos los eventos mutuamente excluyentes que producen una falla del sistema.

Este método se realiza de la siguiente manera:

- Se identifican todos los eventos mutuamente excluyentes que provocan la falla del sistema.
- Se suman las probabilidades de tales eventos.
- Se le resta por 1, obteniendo el valor de confiabilidad.

Este sistema también es aplicable a componentes cuya probabilidad dependan en el tiempo.

(5) Sistemas Standby

En los sistemas anteriores mencionados se asumía que un sistema redundante consistía en dos o más ramificaciones, conectados en paralelo y que ambas ramas son operadas simultáneamente. No obstante, uno o más componentes pueden no estar activos constantemente, sino que permanecen en un estado *standby* o *idle*. Estos son activados en ciertos momentos al ser requeridos o al haber una falla. La Figura 3.8 muestra el diagrama del sistema, siendo un switch operado por el usuario que permite el cambio de dispositivo. Las ventajas y comparaciones de este sistema y el paralelo son discutibles y depende mucho del modelo a realizar.

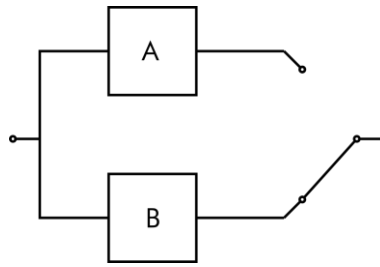


Figura 3.5 Diagrama de sistema standby con 2 componentes (autoría propia)

En la presente figura se observa la dependencia del componente *standby* B del operativo A. El estado mismo del primero depende exclusivamente del segundo, habiendo una relación bayesiana entre ellos. El *switching* puede ser perfecto o imperfecto, determinando si el propio *switch* posee una probabilidad de falla o no. En realidad, ningún *switch* es propenso a funcionar la totalidad de las veces, pero en ciertos momentos es relevante considerar aquellos casos que son estimativos.

(6) Switch Perfecto

Se considera *switch* perfecto cuando no fallan durante la operación ni al momento de cambiar o de volver a su estado original, es decir, cuando son 100% confiables. Para efectos de simpleza se establecerá un sistema de 2 componentes como en la figura anterior, donde B se considerará como el componente *standby*, y al no operar no falla, siendo solamente activado cuando A falle. Por ende, la falla del sistema es dado por A y por B, siendo que A falló. Por último, ambos componentes son iguales con una tasa de fallo λ .

Se utiliza la siguiente expresión para describir la confiabilidad en x caso, donde $P_x(t)$ denota la probabilidad de que x componente falle en tiempo t.

$$P_x(t) = \frac{(\lambda t)^x e^{-\lambda t}}{x!} [-] \quad (19)$$

Por ende, para un sistema de 2 componentes sería:

$$R_s(t) = P_0(t) + P_1(t) = (1 + \lambda t)e^{-\lambda t} [-] \quad (20)$$

3.2.2 Reliability of CubeSats – Statistical Data, Developers’ Beliefs and the Way Forward (Langer & Bouwmeester, 2016)

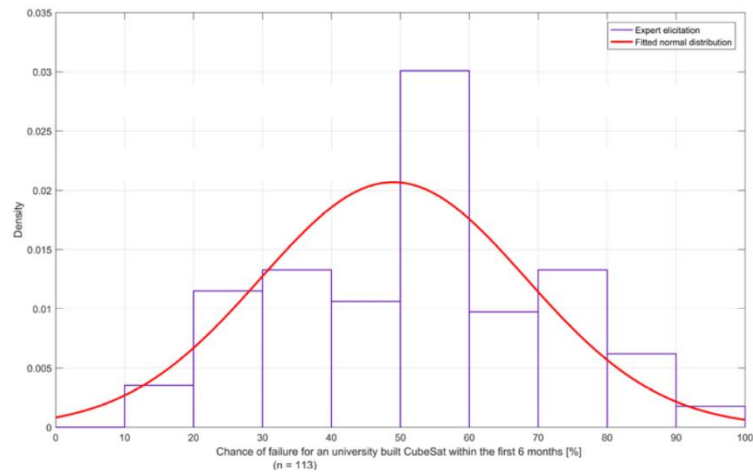


Figura 3.6 Tasa de falla de un satélite tipo CubeSat construido por universidades durante 6 meses (n=113) (Langer & Bouwmeester)

Los autores recopilaron mediante una encuesta 113 ejemplos para poder describir la falla de *CubeSats* bajo distintas lúpas. Una de estas consistió en medir la tasa de falla de un satélite tipo *CubeSat* construido por una universidad, donde se encontró que durante los primeros 6 meses se estimaba promedio encontrarse bajo el 50%. Se utilizó una distribución normal para ajustar la data recopilada, con $\mu = 48.98$ y $\sigma = 19.29$, que se observa en la Figura 3.9.

Muchos de los satélites lanzados y construido hasta la fecha de este artículo son perdidos durante la primera fase de operaciones, mencionan los autores. Swartwout menciona que mucha de las fallas iniciales se produce por falta de pruebas a nivel sistema (explicado por falta de tiempo y presupuesto). Concluyen que mucha de estas fallas iniciales puede resolverse por medio de pruebas funcionales y preocupación en la integración más que en el uso y generación de pruebas de calificación más complejas.

3.2.3 Improving CubeSat Reliability: Subsystem Redundancy or Improved Testing? (Bouwmeester et al., 2022)

Los autores estimaron las curvaturas de confiabilidad mediante el uso de la inferencia bayesiana en la base de datos obtenida por ellos mismos y categorizándolas en subsistemas los cuales son:

- *Attitude Determination and Control (ADCS)*
- *Command and Data Handling Subsystems (CDHS)*
- *Communication Subsystem (COMMS)*
- *Structure & Deployment Mechanisms (STS & DepS)*
- *Electrical Power Subsystem (EPS)*
- *Payload (P/L)*

El subsistema térmico se ha omitido debido a que no se encuentra presente en la base de datos y que se encuentra como un componente pasivo en *CubeSats* o se encuentra en otros subsistemas. Las interfaces eléctricas fueron alojadas a sus subsistemas primarios (interfaz de datos a CDHS, distribución de poder a EPS). Se recalca la existencia de una categoría extra, de tamaño no menor (23 de 71 registros), denominada *unknown* cuyo origen de la falla es desconocida. La censura de estos datos conlleva a la sobreestimación de la confiabilidad de tales subsistemas, por ende fueron removidos inicialmente previos al análisis.

Asumiendo una relación en serie entre los subsistemas, las curvas se describen según el producto Lognormal-Gompertz, se obtienen las siguientes curvas para cada subsistema según los autores:

$$R(t) = e^{-\eta\left(e^{\frac{t}{\theta}} - 1\right)} \cdot \frac{1}{\sigma\sqrt{2\pi}} \int_0^t \frac{2}{x} e^{-\frac{1(\ln x - \mu)^2}{2\sigma^2}} dx \quad (21)$$

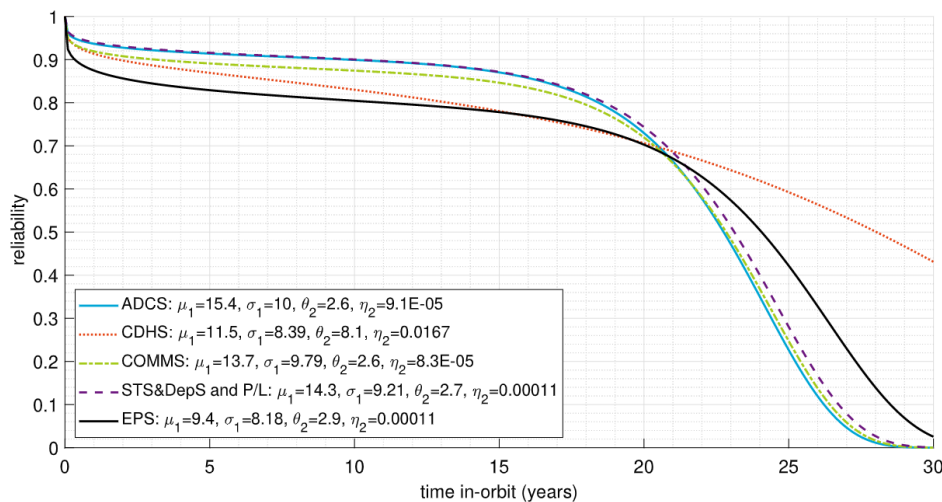


Figura 3.7 Curvas de confiabilidad de cada subsistema (Bouwmeester et al, 2022)

Para el estudio de los efectos de redundancia en los subsistemas, la ubicación de la falla en los subsistemas no es ideal, ya que la redundancia en la práctica se aplica a componentes, sus interfaz o unidades físicas. Se espera que a futuro los *CubeSats* más avanzados sean propensos al uso de redundancia de subsistemas completos. De acuerdo con el *CubeSat Data Survey* (Bouwmeester et al., 2017) aproximadamente el 10% de las unidades *bus* son redundantes. Aunque esto no permite establecer ninguna conclusión sobre la redundancia en otros subsistemas, puede ser usado como indicativo que este campo no ha sido ampliamente implementado todavía.

Finalmente los autores concuerdan en la presencia de una correlación entre la sofisticación de *CubeSats* y la experiencia de los desarrolladores, mientras la funcionalidad es añadida en proyectos siguientes. Es decir, se observa una evolución en el desarrollo de *CubeSats*, tanto en la experiencia de los desarrolladores como en la mejora y funcionalidad de estos, alcanzando por ende un mayor grado de confiabilidad.

3.2.4 Satellite system Design Optimization Based on Bayesian Melding of Multi-Level Multi-Source Lifetime (Xu et al., 2019)

En el *Reliability-Based Design Optimization* (RBDO) de un sistema satelital, el modelamiento de la confiabilidad es esencial. Ante la problemática presentada en el Capítulo 1, los autores proponen el método de *Bayesian Melding* (BMM) para fusionar la información de su vida útil a partir de múltiples fuentes, con el objetivo de actualizar las curvas de confiabilidad de los niveles superiores y reactualizar los inferiores con el objetivo de optimizar las estructuras entre cada nivel. El sistema por analizar se separa en distintos niveles de elementos y componentes, dependiendo de la complejidad a analizar (de sistema espacial a componente por ejemplo) como se aprecia en la Figura 3.8. Para cada módulo se establece una estructura entre los componentes que lo componen, influenciando la aplicación de inferencia bayesiana entre estos. Luego, se reactualizan todas las curvas de confiabilidad en los niveles superiores para luego reactualizar los inferiores.

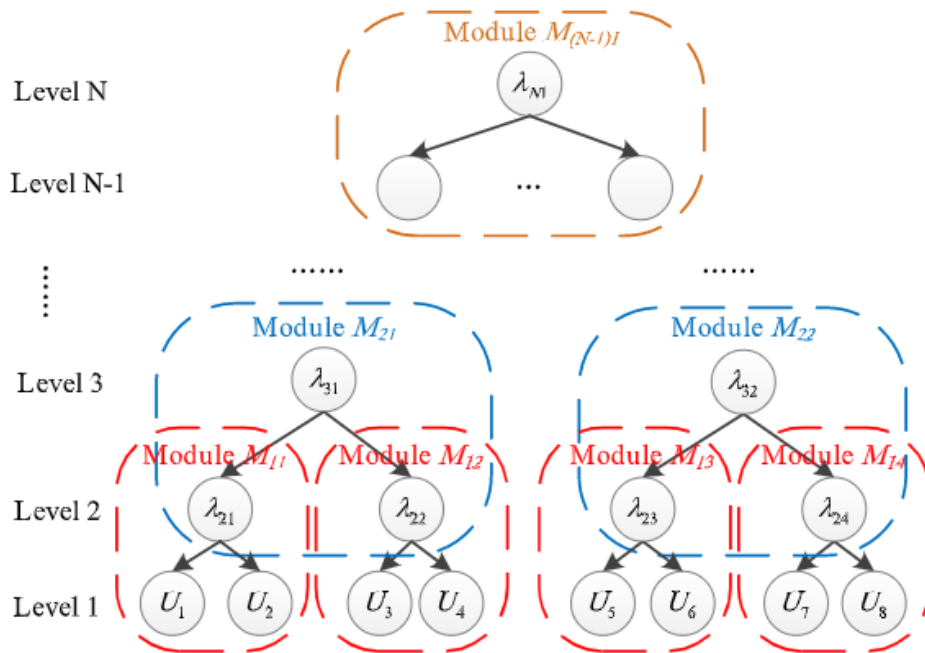


Figura 3.8 Representación Gráfica Simplificada de un Sistema Satelital de Varios Niveles (Xu et al, 2019)

Este sistema de actualización de curvas de confiabilidad viene acompañado de un método de optimización RBDO, donde se le solicita al usuario parámetros de restricción relativos al diseño de un proyecto (costo de presupuesto, confiabilidad de un satélite y su estimación de confiabilidad en cada componente que lo compone).

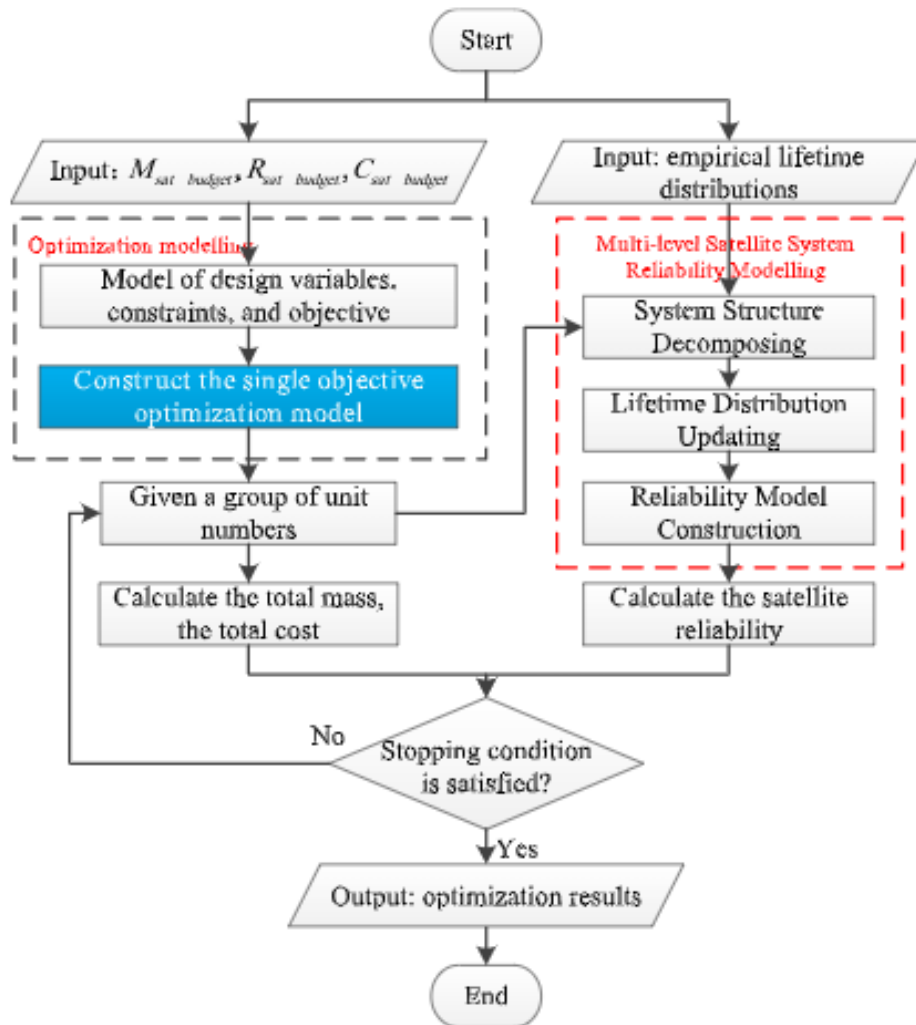


Figura 3.9 Diagrama de Flujos del Proceso de Optimización Propuesto (Xu et al, 2019)

CAPÍTULO 4: Desarrollo del Estimador

En el presente capítulo se describirá la implementación de la propuesta definitiva, que va desde la definición de los requerimientos del sistema previo a su construcción, la obtención de datos iniciales y la descripción de los programas construidos que se encuentran en Anexos 4 – 7. Se toma en cuenta los parámetros definidos inicialmente en el capítulo que aluden al tipo de misión DSM a simular.

4.1 Propuesta Inicial

Inicialmente en el PIA se definieron los subsistemas del elemento *CubeSat* a evaluar, en base a las causas de falla identificadas por (Bouwmeester et al., 2022). Estos son *Electric Power Subsystem* (EPS); *Telemetry, Tracking and Control* (TT&C); *Attitude Determination and Control* (AD&C); *Command and Data Handling* (C&DH); *Payload* (P/L); *Mechanisms and Structure* (M&S); *Thermal Control System* (TCS); *Unknown*: La razón de falla es desconocida.

Posterior a la definición se presentó el modelo IPO, como se aprecia en Figura 4.1, el cual se divide en 3 etapas: la etapa I/O recopila información al inicio del proceso que serán utilizadas en la segunda y tercera etapas llamada Nivel Satélite y Subsistema, cuyos análisis primordiales se enfocan en tomar al satélite como una unidad básica y compuesta de análisis, respectivamente. En ambos se consideran dos escenarios: el primero, si se tiene acceso a una base de datos contundente, se puede realizar el mismo procedimiento de (Bouwmeester et al., 2022) para el caso no paramétrico y paramétrico. El segundo, si no se tiene acceso a la base de datos, se podrán utilizar las curvas paramétricas obtenidas en (Bouwmeester et al., 2022), debido a su mayor análisis recabado. Finalmente, los resultados se integran al modelo propuesto, para entregar resultados.

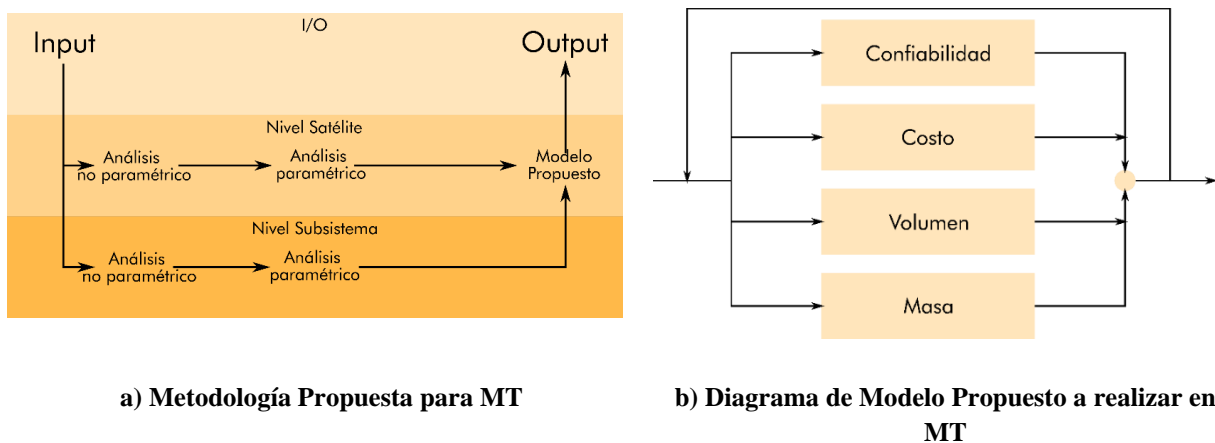
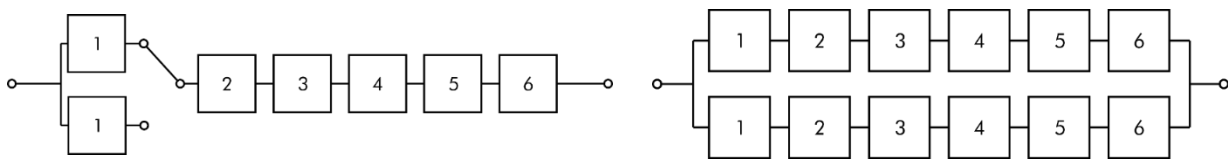


Figura 4.1 Metodología y Diagramas de Modelo Iniciales (autoría propia)

El modelo de cálculo propuesto, como se ve en la Figura 4.1, es un proceso combinatorio que estima la confiabilidad, costo, volumen y masa de las configuraciones a considerar. Inicialmente, se selecciona una configuración de subsistema, luego se evalúa en los valores mencionados para después juntar los resultados y probar nuevamente con otra configuración. Todos los resultados de cada iteración serían guardados según su configuración seleccionada para análisis.

Se definió el uso de los Diagramas de Confiabilidad (Modelamiento de Redes) para conocer los efectos de la redundancia en la confiabilidad de los *CubeSats* en distintas configuraciones. Para ello, se definen las relaciones entre subsistemas y elementos para configurar en la Figura 4.2.



a) Diagrama de Sistema con Standby en el Subsistema 1

b) Diagrama de sistema distribuido compuesto por 2 sistemas en paralelo

Figura 4.2 Diagramas RBD a Usar en Modelo Inicial (autoría propia)

Se puede apreciar que la relación entre subsistemas es en serie y la redundancia se expresa como un componente en *standby* con *switch* perfecto. Entre los 6 subsistemas se plantea la revisión de cada combinación de redundancia posible, con la posibilidad de revisar hasta 2 redundancias por subsistema. Por último, se define la combinatoria de ambos esquemas a la vez, es decir, que existan sistemas distribuidos con redundancias en satélites o sin ellos.

Para los SE Envelopes el procedimiento es similar. Para cada subsistema, la información se recolecta en referencia a las masas, costos y volúmenes de diversos componentes utilizados en cada sistema. Luego, se realiza una media aritmética obteniendo el promedio de las tres variables, esto permite una combinatoria más ágil, dependiente netamente de la configuración escogida y no de los componentes elegidos (Kiselyov, 2020). Desde su concepción en la carrera espacial, la filosofía de diseño satelital estuvo predominada por componentes altamente confiables y diseños robustos y conservadores, hechos para soportar las condiciones extremas del espacio por largos períodos de tiempo (Langer & Bouwmeester, 2016).

El ascenso de los *CubeSats* cambió esta filosofía inicial por una en favor de utilizar productos *Commercial-Off-The-Shelf* (COTS), haciéndolos económicamente más viables para la industria espacial, reduciendo el costo y tiempo de desarrollo asociado, incrementar el acceso al espacio a nuevas instituciones y permitir un mayor número de lanzamientos por año (The CubeSat Program, 2022), sacrificando confiabilidad y redundancia en el proceso. A consecuencia de ello, sumado con la reducción sustantiva del costo por kilogramo (cuánto cuesta lanzar 1 kg de carga al espacio) desde \$60 000 USD/kg del Space Shuttle (Pielke, 2008) hasta \$5 500 USD/kg del Falcon 9 (Sheetz, 2022); el impulso generado por el programa *CubeSat* también permitió mayores avances en la miniaturización tecnológica para construir satélites (será también denominado elemento) con componentes de menor costo, compactos y COTS (Poghosyan & Golkar, 2017).

4.1.1 Cambios Por Realizar

Se presentan varios aspectos a considerar para evaluar posibles mejoras a esta propuesta.

- La propuesta inicial presenta ambigüedades en el desarrollo del proceso de cálculo. Es necesario detallar y clarificar sobre cuáles son las variables relevantes solicitadas, los procesos que se realizarán y los resultados específicos a resolver.
- Un sistema distribuido por su complejidad de definición no puede ser reducido a un diagrama de confiabilidad, en comparación a los subsistemas y sus relaciones entre ellas. Es necesario el uso de un método que permita la simulación del sistema complejo en su totalidad, o la acotación del problema a abarcar.
- No se detalla de qué manera o en qué plataforma(s) se van a implementar.
- Se usarán los subsistemas y las curvas de confiabilidad obtenidas por Bouwmeester, J. para la estimación de confiabilidad de un cluster de *CubeSat* al no tener acceso a la base de datos de la cual los autores mencionados en el Estado del Arte tuvieron acceso.
- Establecer que la cantidad de satélites por lanzar está relacionado a otros factores que se originan del diseño de la misión y no por estimación del usuario.
- Según Guo y Kiselyov en la práctica se prima el estudio de redundancia netamente en el subsistema EPS debido a que estadísticamente es el que más falla (Guo et al., 2014; Kiselyov, 2020).
- El estudio debe primar en el *bus* y no en la carga útil, ya que esta última está atado a otros parámetros de misión que son irrelevantes para este estudio. Lo relevante es saber cuánto restante nos queda de carga útil, y como este cambia dependiendo de la configuración buscada. También se omite la media aritmética de los valores.
- El costo de lanzamiento no se mide en la práctica por la masa, sino que mediante el servicio de *ridesharing*, vender cupos *CubeSats* de *Pods* que se lanzarán a futuro, en las lanzaderas orbitales.

4.2 Consideraciones de Diseño

Se plantea desarrollar un sistema que estime y compare la confiabilidad y los *SE Envelopes* para *bus* y *clusters*, de distintas configuraciones de redundancias y mitigaciones de riesgo. Para ello se toman las siguientes consideraciones que entregan forma y guía a la arquitectura a construir. Como regla general se asume que todo dato técnico entregado por proveedores sobre estos subsistemas será considerado como cierto y confiable.

4.2.1 Diseño General y de la Misión

Se establece una misión que tenga una duración estimada de 5 a 7 años (Torres, 2017), similar al periodo de misión que un elemento monolítico es diseñado para durar. El propósito de la misión es EO, situando un *cluster* en órbita LEO. El diagrama del segmento espacial se muestra en la Figura 4.3, donde se expone el diseño del cluster a evaluar. Todos los satélites que conformen la constelación se encuentran en órbitas similares, con el objetivo que se encuentren en el mismo ambiente espacial. Periódicamente se añadirán más elementos al mismo ambiente espacial que se encuentre el resto. Tiene que haber una mínima cantidad operativa de *CubeSats* para considerarse que la misión sigue en curso.

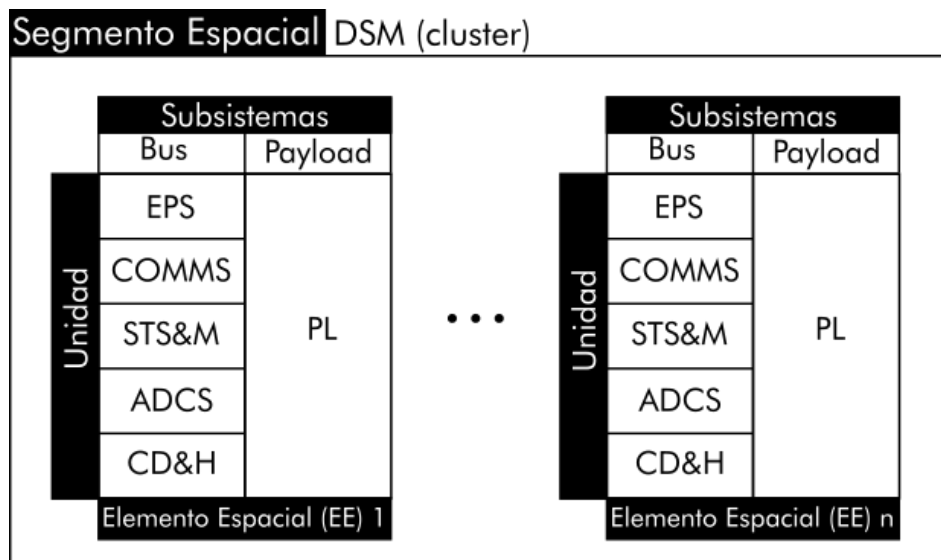


Figura 4.3 Diagrama Segmento Espacial (autoría propia)

Con respecto al programa a construir, se requiere que este sea modificable y aplicable a cualquier configuración según las necesidades contingentes a su construcción. También de una interfaz amigable que permita la ejecución rápida de la totalidad del programa o solo parte de ella. Este programa debe estimar la confiabilidad de un *cluster* definido por el usuario, por medio del uso de *Reliability Block Diagrams* (RBD) para la construcción y estimación de su curva de confiabilidad a lo largo del tiempo, en base a las curvas de confiabilidad de subsistemas obtenidas por Bouwmeester (Bouwmeester et al., 2022). Se requiere que el programa a construir genere un tradespace del cluster que describa los posibles costes totales, masa y volumen de cada *CubeSat* y de la misión, según el estándar y la configuración solicitada.

4.2.2 Subsistemas

Cada elemento que componga el *cluster* se descompone en unidades *bus* y *payload*, donde el *payload* se aboca exclusivamente a la EO y el *bus* a la subsistencia del sistema satelital. El *bus*, a

su vez, se descompone en los subsistemas definidos previamente. Para efectos de análisis de diagramas RBD se define que cada subsistema se relaciona uno con otro en serie, es decir, que para que el sistema funcione todos deben funcionar. La estrategia de redundancia para mitigación a la falla se realizará considerando subsistemas *backup* y *standby* que, en caso de falla, estos se activen en reemplazo. La probabilidad de falla y la confiabilidad de cada subsistema se obtendrá mediante las curvas obtenidas por J. Bouwmeester et al. en 2022 (Bouwmeester et al., 2022) en la Figura 4.4. Se recalca que, según establece el autor, las curvas determinadas sólo pueden entregar cierta certeza estadística en los primeros 2 años, debido a que la data se concentraba por ese período de tiempo y la forma de la función tras este tiempo está netamente ligado a la escogida por el mismo autor.

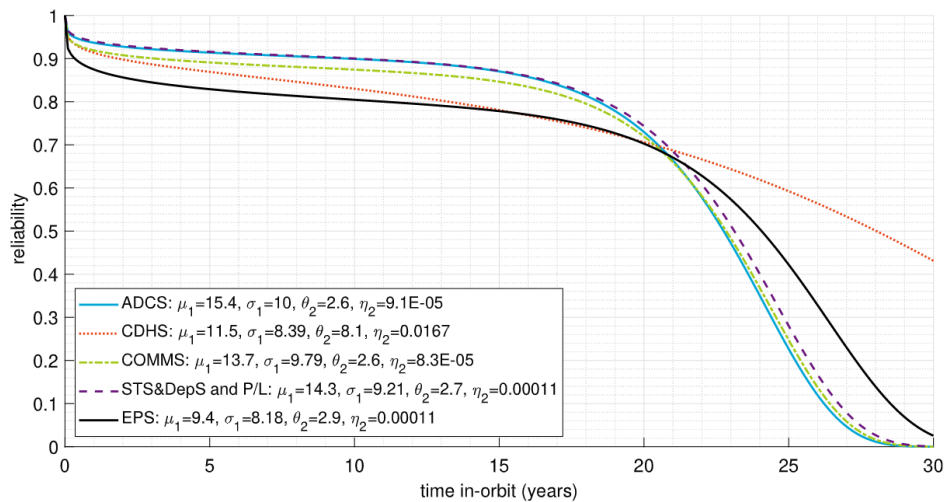


Figura 4.4 Curvas de Confiabilidad para cada subsistema (Bouwmeester et al., 2022)

4.2.3 Elemento *CubeSat*

El *cluster* estará compuesto exclusivamente por unidades *CubeSats* que pueden ser exclusivamente de 1, 3 o 6U. Los *CubeSats* tendrán una vida promedio de 1 a 2 años, cuya confiabilidad no podrá bajar del 60% (Bouwmeester et al., 2022). A su vez, la curva de confiabilidad que lo defina se obtiene por el mismo autor que entrega las curvas para cada subsistema.

Como se ha mencionado anteriormente, la cantidad de volumen y masa están delimitados por diseño. En la práctica al diseño no es el estándar que define su tamaño máximo, sino al contenedor que posteriormente despliega al *CubeSat* al espacio, comúnmente denominado *pod*. Estos con el pasar de los años han sido modificados para que permita un mayor porcentaje de espacio, más allá del estándar inicial, tanto el aumento del espesor admisible en los costados del nanosatélite como se observa en Anexo 1-3, como también la inclusión de un volumen cilíndrico en la cara superior denominado *tuna-can* (The CubeSat Program, 2022) y utilizado para la instalación de cámaras.



Figura 4.5 Tipos de ISISPOD desarrollados por ISISPACE (ISISPACE)

Para este estudio se seleccionaron los *pod* manufacturados por ISISPACE, denominados ISISPOD como se observan en la Figura 4.5. Su característica principal se debe a que el tamaño máximo admisible son uno de los mayores en el mercado (The CubeSat Program, 2022). Una vez seleccionado y obtener los planos para los *pod* 1U, 3U y 6U disponibles en Anexos 2-4, se obtienen los volúmenes y masas máximas disponibles, incluyendo al espacio de tuna can en las unidades 3U y 6U. Estos datos son relevantes ya que, dependiendo del tipo de estándar escogido, no toda combinación de componentes registrados en la base de datos es válida, y estos parámetros son cruciales como un filtrado rápido. Los datos finales se pueden observar en la Tabla 4.1.

Tabla 4.1 Características de masa y volumen máximo para ISISPOD de 1U, 3U y 6U

Estándar	Masa máxima (kg)	Volumen máximo (mm ³)
1U	2	1 496 578
3U	6	4 684 664
6U	12	10 166 098

La observación terrestre incide como requerimiento el área barrida diaria que debe realizar por día. Mientras más satélites tengas operando, mayor es la capacidad de observación. Como referencia se considera lo realizado por *Planet Labs*, quienes mediante el sucesivo lanzamiento de nanosatélites de observación pueden barrer más área a diario que uno monolítico (Devaraj, 2021). A modo de ejemplo, su modelo *CubeSat* 3U denominado “Dove”, realiza individualmente $1.5e6$ [km²/day]. En comparación con el tamaño aproximado del Amazonas, $6.7e6$ km², se han de necesitar mínimo 5 para lograr una observación diaria de la región. A modo de estudio se tomará en primera instancia el área de barrido diario del Dove como referencia al área de barrido del cluster configurado.

4.2.4 Segmento Espacial *Cluster*

Aunque la definición entregada previamente de *cluster* no es suficiente para caracterizar la estructura que se quiere simular, se utiliza la Tabla de Diseño para DSM creada por Le Moigne J, et al, cuyas características se establecieron en Tabla 4.2 y definidos a continuación.

Tabla 4.2 Tabla de Diseño de DSM

Trades DSM					
TIPOS PRINCIPALES DSM	CONSTELACIÓN GENERAL				
	Apariencia	Homogéneo		Heterogéneo	
Instrumento				Bus	Ambos
Relación Entre Naves	Ninguna	Jerárquico	Peer-To-Peer		Rendezvous
					Cooperativo
Control Espacial	Predeterminado / Pasivo	En Tierra	En Órbita / Swarm		Mezcla De En Tierra Y En Órbita
Despliegue	Todo A La Vez		Desfasado / Incrementado		
			Por Diseño	A Reacción	
Control Temporal	Predeterminado Y/O Mediciones Correlacionadas Ad Hoc Libres		Tiempo De Observación Preciso	“Flash Mob”	
Autonomía	Ninguna		Semiautónomo		Autónomo
Nº De Naves	[2-10]		[10-50]		>50
Masa De Nave	< 1	[1-10]	[10-500]	[500-5000]	>5000
Configuración De Lanzamiento	Un Lanzamiento		Múltiples Lanzamientos		
Configuración De Lanzadera	Dispensador		Lanzamientos Múltiples		

- Apariencia - Homogénea -: Constelación cuyos elementos emplean funcionalmente idénticas, en *bus*, *payload* y características operacionales.
- Relación entre Naves – Ninguna -: No existe una relación jerárquica entre naves de la misma constelación.
- Control Espacial – *Swarm* -: Varios elementos (principalmente micro o nanosatélites) que sirven un propósito común y son mayoritariamente controlados en tierra.
- Despliegue – Todo a la vez -: La constelación se implementa y se activa todo a la vez, así como los elementos que deban ser reemplazados.
- Control Temporal – “Flash Mob” -: Relacionado a mediciones interrelacionadas. Al realizar observaciones en una misma región y con distintos instrumentos pueden relacionarse entre ellos a lo largo del tiempo.
- Autonomía – Sin autonomía -.
- Nº de Naves – [2-10] -.
- Masa de Nave – [1-10] kg -.
- Configuración de Lanzamiento – Un lanzamiento -: Debido a la gran capacidad de carga disponible en lanzaderas, de ser de 2 a 10 elementos pueden ser desplegados en tan solo 1 lanzamiento.
- Configuración de Lanzadera – Dispensador -: Al ser estándar *CubeSat*, estos son desplegados por dispensadores.

Se establece por último que, para efectos de este estudio de confiabilidad, se establecerá que un cluster bajo la definición otorgada anteriormente puede ser estudiado con RBD. Esto se debe a que los elementos del sistema o cluster son independientes entre ellos con una operación lógica, es decir, o funcionan o fallan sin puntos intermedios. Se define que se utilizará el sistema k-out-of-n, ya que como mencionado en la sección anterior, se necesita una cantidad mínima de elementos operativos para considerar que la misión sigue su curso.

4.2.5 Limitaciones

Las limitaciones presentes en el diseño, abarca la propia arquitectura. La relación entre subsistemas y elementos, así como entre satélite y DSM es lógica, queriendo decir que el sistema solo puede asumir dos estados, operativo y fallido. Aunque es útil al querer visualizar y calcular la confiabilidad del sistema completo a lo largo del tiempo, no representa situaciones ante una anomalía ni cuánto se ha degradado los elementos del sistema o el propio sistema tras ésta, como son las fallas de *software*. Tampoco considera que las configuraciones de órbita, el tipo de misión ni el tipo de institución en la que fue creada, incide en la estimación de confiabilidad, siendo que se ha comprobado estadísticamente lo contrario. Sin embargo, esto puede ser estudiado para su posible integración.

4.3 Nueva Arquitectura

A diferencia de la conclusión establecida por Bouwmeester, J. sobre el cómo atacar la problemática de confiabilidad de *CubeSat*, que es la mejora de pruebas previas a lanzamiento; este trabajo ahondará en otra opción para mejorarlo, que es lanzar más elementos y periódicamente. Tomando en cuenta esto y todo lo mencionado previamente es que se presenta una nueva arquitectura, detallada en su funcionamiento y propósito, que incorpore las consideraciones de diseño y de misión tanto como sus limitaciones. La arquitectura se representa en el diagrama IPO, en la Figura 4.6.

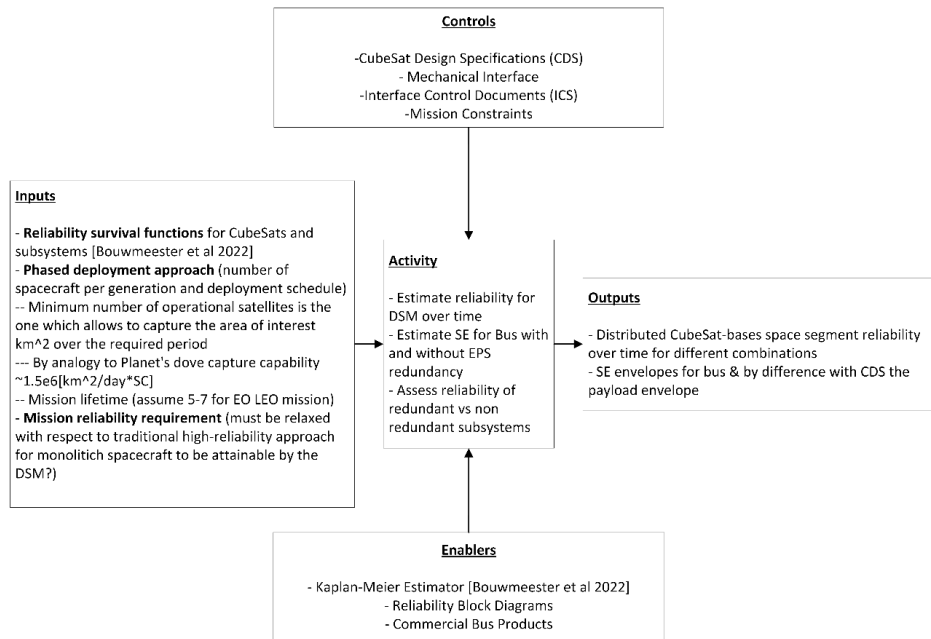


Figura 4.6 Diagrama IPO Proceso Definitivo (autoría propia)

El diagrama explica el funcionamiento de la arquitectura separándolo en 5 regiones.

- **Inputs:** Elementos que forman parte de la entrada del sistema. En este caso corresponde a las funciones de confiabilidad de *CubeSats* y Subsistemas, el acercamiento para un despliegue periódico y desfasado y el requerimiento de confiabilidad de la misión.
- **Activity:** Corresponde a las actividades que el sistema debe de realizar al input entregado para obtener el output deseado, tomando en cuenta los *enablers* y *controls*. En este caso es estimar la confiabilidad del DSM a lo largo del tiempo, estimar los SE *Envelopes* para la unidad *bus* sin o con redundancia y evaluar la confiabilidad para sistemas redundantes y no redundantes.
- **Output:** corresponde a la salida del sistema, los resultados esperados. En este caso corresponde las curvas de confiabilidad del segmento espacial DSM para distintas configuraciones y un SE *tradespace* del *bus* y del remanente para el *payload*.
- **Control:** corresponde a elementos que restringen al sistema y le dan una limitante que debe seguir. En este caso son las especificaciones de *CubeSats*, las interfaces mecánicas, los *Interface Control Documents* y restricciones de la misión.
- **Enablers:** corresponde a elementos que permiten o herramientas que el sistema usa para la ejecución de las actividades. En este caso es el Estimador de Kaplan-Meier, los *bus* para *CubeSats* disponibles en el mercado y los RBD.

Para la construcción de la arquitectura se utilizará Python por preferencia del autor y de su capacidad de integración con otras plataformas y códigos, incluso ejecutando Matlab u otros programas relevantes. Se trabajó en el compilador Pycharm en ambiente de Anaconda, conocido en juntar múltiples herramientas que trabajan con Python de distintas formas. La estructura del programa a construir se observa en la siguiente Figura 4.7.

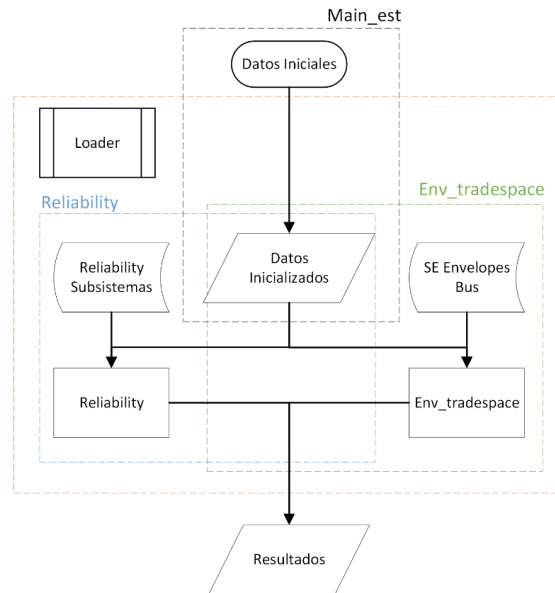


Figura 4.7 Diagrama Arquitectura Definitiva (autoría propia)

Se aprecia la presencia de tres módulos, que serán expuestos en las siguientes secciones, que estructuran el diagrama. Se comienza con el ingreso de datos en el módulo denominado `Main_est` donde, una vez inicializados, son enviados a los módulos `Reliability` y `Env_tradespace`. Estos dos últimos reciben data a partir de datos guardados u obtenidos en subprocesos (como es en el caso del bloque “Reliability Subsistemas”) para finalmente entregar resultados y dar un posterior análisis. A continuación se explicará en detalle el funcionamiento de cada módulo mostrado en la Figura 4.7.

4.4 Módulo `Main_est`

La función del módulo `Main_est`, cuyo código y diagrama de flujo se encuentra en Anexos 5 y 9, es ser el programa principal que conecta a ambos módulos subsiguientes, suministrándoles con los inputs necesarios para su funcionamiento, tales como la cantidad mínima de satélites que el cluster requiere, el tipo de *CubeSats* a analizar y la tasa de lanzamiento de satélites (cuantas veces por año). También toma decisiones sobre qué tipo de análisis quiere realizarse, si se le considera la redundancia en el sistema EPS o si quiere realizar un barrido de cuántos satélites lanzados por año influye en su funcionamiento. Una vez realizado, este abre tanto los módulos `Reliability` y `Env_tradespace` a realizar sus siguientes operaciones.

4.5 Módulo Reliability.py

La función del módulo Reliability, cuyo código y diagrama de flujo se encuentra en los Anexos 6 y 10, se centra en la obtención de la curva de confiabilidad de cada configuración de DSM que se le solicite. Para entender el concepto a construir, se tomará como ejemplo lo ilustrado en la Figura 4.8.

Consiste en un cluster donde inicialmente se compone de dos elementos tipo *CubeSat*, ambos con una misma curva de confiabilidad. Luego de cierto periodo T , dos elementos más son lanzados para ser incluidos activamente en la misión, sin embargo el valor mínimo de elementos operativos no cambia. Esto continua hasta el final de la operación. Esto se le denomina sistema *k-out-of-n* con elementos no iguales, cuyo método de resolución se explicó en el Estado del Arte. Al aplicar ese método se itera a lo largo del tiempo para obtener la curva de confiabilidad del DSM solicitado, y a su vez este itera para probar distintas configuraciones. Las distintas configuraciones o estrategias de mitigación se componen principalmente de tres métodos.

- Redundancia en el subsistema EPS.
- Lanzamientos periódicos o en fases de periodo T .
- Redundancia activa elemental.

Una vez realizado las operaciones, la confiabilidad del segmento espacial es graficada junto con el trabajo hecho en el módulo explicado a continuación.

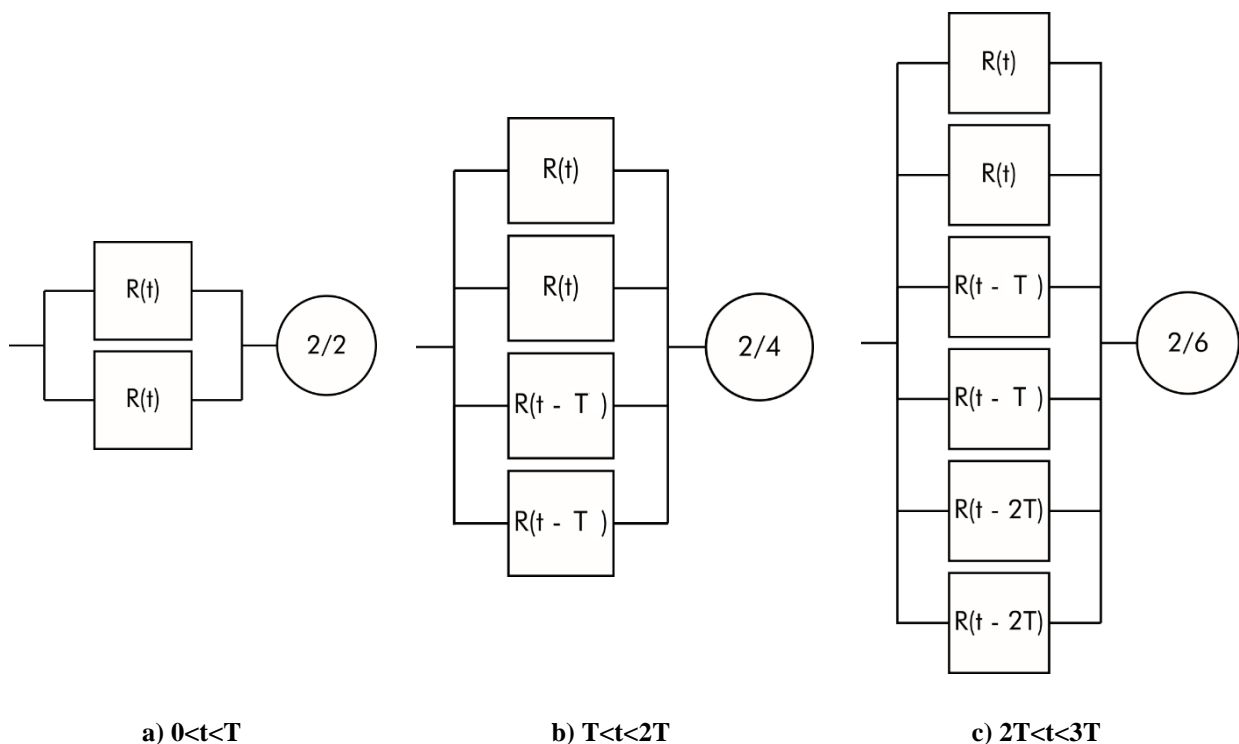


Figura 4.8 Ejemplo de Segmento Espacial con despliegue en fase de periodos T (autoría propia)

4.6 Módulo Env_tradespace.py

La función del módulo Env_tradespace, cuyo código y diagrama de flujo se encuentra en Anexos 7 y 11, es de representar gráficamente las opciones de *bus* disponibles de compañías según la configuración deseada, con el objetivo de conocer las cantidades disponibles a usar para el *payload* y visualizar como estos datos cambian dependiendo de la configuración y cantidad de satélites enviados a órbita.

El módulo realiza dos operaciones esenciales. En primer lugar, a partir de los datos de *bus* guardados en un archivo .csv y dependiendo de la configuración deseada (con o sin redundancia en EPS, tipo de *CubeSat* escogido), se realizan las operaciones como se muestran en la Ecuación 22. Estos resultados serán presentados en una gráfica scatter donde en sus ejes x e y muestren masa y volumen disponibles y en barra de color se muestre el costo asociado a este.

$$\begin{aligned}
 Masa_e &= CubeSat_{M,Max} - Bus_M [g] \\
 Volumen_e &= CubeSat_{V,Max} - Bus_V [U] \\
 Costo_T &= CubeSats_{total} \cdot \left\{ Bus_C + 1 \cdot 10^3 \cdot Masa_e \cdot Broker_{cost_{CubeSat_{tipo}}} \right\} [USD]
 \end{aligned}
 \tag{22}$$

En la práctica el costo de lanzamiento de un *CubeSat*, como paralelismo de su volumen y masa máximos, no depende exclusivamente de los parámetros del nanosatélite sino que también de la empresa quien los lanza. Para efectos de este estudio se utilizarán los datos entregados por SpaceFlight Industries, una compañía que administra y vende cupos de lanzamientos para *CubeSats* como *MicroSats* (SpaceFlight Industries, 2023). Se toma como referencia los siguientes parámetros económicos en divisa USD.

Tabla 4.3 Costo de Lanzamiento para el Cuarto Cuatrimestre de 2023, al 6 de febrero 2023 por SpaceFlight Industries

Tipo de Estándar CubeSat	Costo de Lanzamiento (Broker_cost) en \$USD
1U	70 000
3U	145 000
6U	295 000

CAPÍTULO 5: Simulación y Resultados

En el siguiente capítulo se compilarán los resultados obtenidos a partir de la arquitectura trabajada. Los resultados serán presentados según las distintas estrategias de mitigación, las cuales son:

- Lanzamientos periódicos o “en fase”.
- Redundancia activa elemental
- Redundancia en EPS.

Se compararán en la contribución que genera a la confiabilidad cada configuración con respecto a la anterior, tomando en cuenta que tiene que sobrepasar del 60% mínimo en todo momento. El análisis de tradespace será exponiendo al costo total del proyecto como una limitante del diseño comparación entre configuraciones a partir de las distintas opciones de COTS disponibles en el mercado. Los análisis de gráfica se realizarán tomando en cuenta la necesidad de que haya como mínimo cinco *CubeSats* orbitando en el segmento espacial, equivalente al barrido de *CubeSats* Dove de $7.5e6 \text{ km}^2$ diarios, sin perjuicio que este análisis puede ser realizado con una mayor o menor área de barrido diario.

Se establece el primer resultado a considerar. En la Figura 5.1 se observa diferentes curvas de confiabilidad a lo largo del curso de dos años para cada cluster de *CubeSat* lanzado. Se establece que la misión requiere que todos los elementos del segmento espacial estén operativos a lo largo de ese tiempo que, en los RBD se puede traducir en una relación en serie entre ellos. Por ende, se explica que mientras más elementos deben de estar operativos para realizar un barrido de área más extenso, más exigente se vuelve la curva de confiabilidad y más propenso que el sistema falle en menor tiempo. Cuantitativamente se traduce en, por ejemplo en la Figura 5.1, que para un sistema de 2 *CubeSats* tenga aproximadamente ~60% de falla en su primer año versus uno de 5 *CubeSats* con una falla de ~90% al primer año. Esto pone en evidencia la imperiosa necesidad de implementar estrategias de mitigación para asegurar y alargar la vida operativa de la misión, así como mitigar los efectos de *infant mortality*, cuyos efectos son mucho más prominentes al inicio.

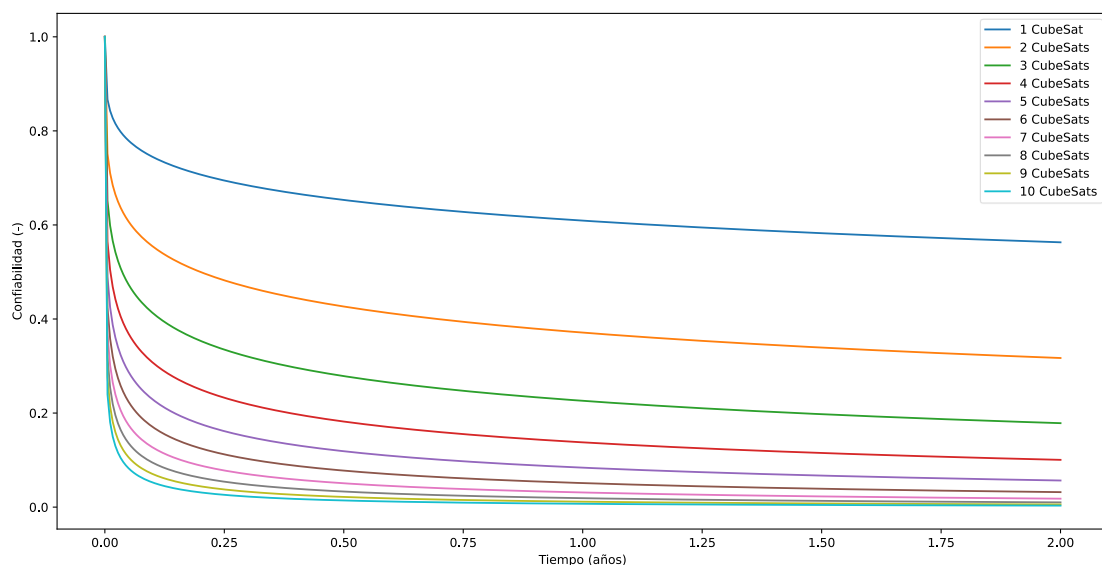


Figura 5.1 Gráfica de Curvas de Confiabilidad de Clusters Compuesto entre 1 a 10 Elementos (autoría propia)

5.1 Estrategias de Mitigación y su Tradespace

5.1.1 Lanzamientos Periódicos

La estrategia de lanzamiento periódico consiste en la mantención constante de la misión lanzando en un periodo T más elementos al espacio al integrarse al sistema. Ejemplos en la práctica son los comentados en el capítulo 1 que practican esta estrategia para mantener sus servicios por un tiempo prolongado. Esto, en la práctica genera un recambio constante de elementos nuevos ingresando al segmento espacial, compensando la posibilidad de falla a lo largo del tiempo.

Para la lectura de las siguientes figuras, en las leyendas se presenta una etiqueta donde cada componente se lee de la siguiente manera.

- EPS_red: Presencia de redundancia en el elemento (False si no presenta, True si presenta).
- Min: la cantidad mínima de elementos que el cluster debe tener operativos para que la misión pueda continuar.
- Ini: Cantidad de elementos iniciales que el cluster es inicialmente lanzado.
- Rel: Cantidad de elementos relanzados a posteriori.
- Tasa: cantidad de lanzamientos a posteriori realizados por año.

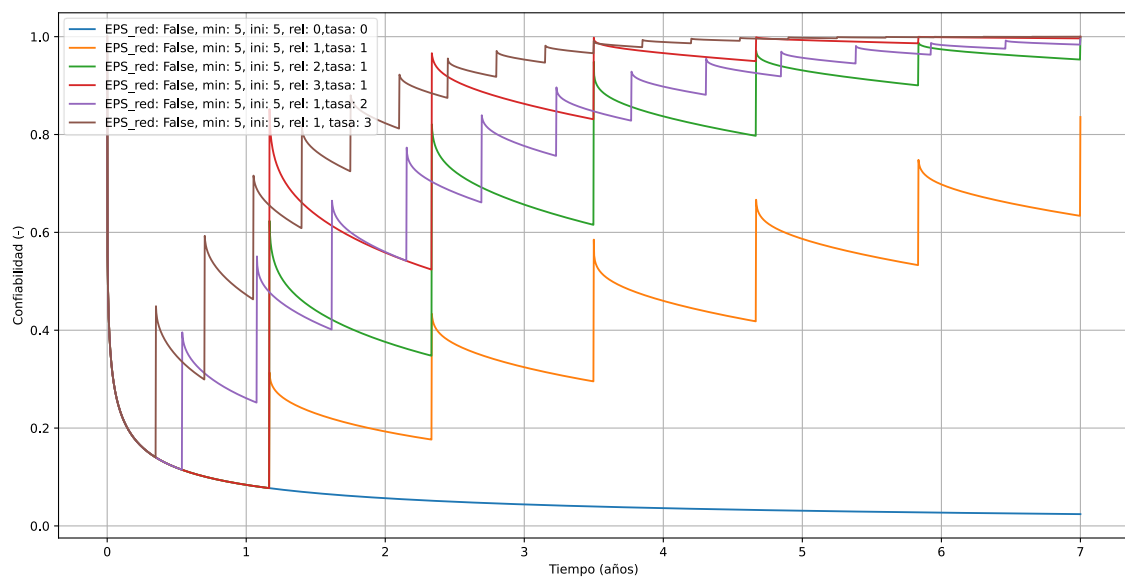


Figura 5.2 Gráfica Comparativa de Curvas de Confiabilidad Según Cantidad de Lanzamientos por Año y Elementos Lanzados (autoría propia)

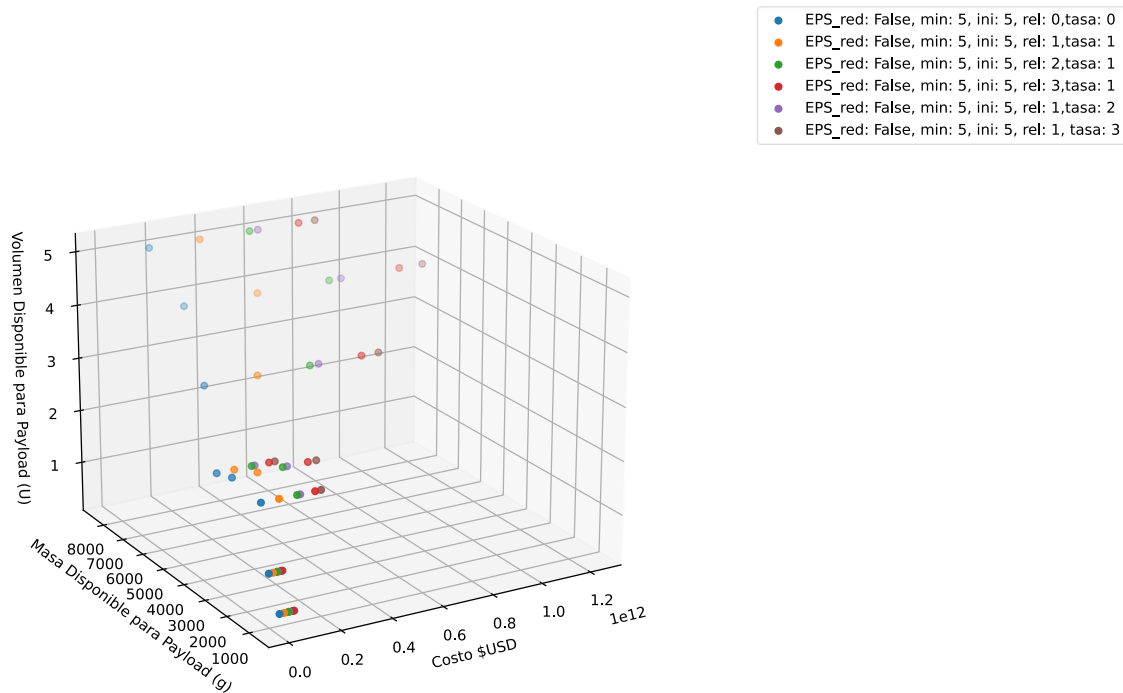


Figura 5.3 Gráfica Tradespace Según Cantidad de Lanzamientos por Año y Elementos Lanzados (autoría propia)

Existen dos opciones a revisar, que es la periodicidad de mantención y la cantidad de elementos por lanzar periódicamente. Ambos influyen en la cantidad de elementos orbitando por año, su renovación y por ende, la confiabilidad del sistema, ambos mostrados en la Figura 5.2, con un segmento espacial de cinco elementos donde cada periodo T cambia de uno a tres veces por año. En general, se observa que todas las gráficas en fase de aquí en adelante tendrán una forma de “sierra”, cuyos “*peaks*” son los relanzamientos e incorporación de nuevos elementos al segmento espacial (asumiendo que todos los lanzamientos y despliegues de la misión hayan sido exitosos). Entre medio de esos *peaks* está la curva de confiabilidad convencional, descrita por el sistema k -out-of- n , donde se destaca que progresivamente esta curva disminuye su tendencia. Se aprecia también que el lanzamiento en fase rompe con la tendencia natural de un sistema frente al desgaste, otorgándole una mayor vida operacional a la misión en su conjunto.

Con respecto al primer ítem, el periodo de lanzamiento como estrategia de mitigación en la Figura 5.2, se observa que a mayor cantidad de lanzamientos por año, mayor es la recuperación de la misión hacia su nivel de confiabilidad mínimo aceptable en un menor tiempo. Sin embargo el incremento de confiabilidad entre las tres configuraciones es la misma, solamente difiere en el tiempo. También se aprecia que es solamente a partir del segundo o tercer lanzamiento que el sistema puede volver al nivel aceptable de confiabilidad en el primer año, cuyo reflejo económico no es favorable como se observa en la gráfica tridimensional.

En segundo ítem, a diferencia del primer ítem cuyo enfoque era generar lanzamientos más seguidos y tras el segundo o tercer lanzamiento se llega a la confiabilidad mínimo del 60%, acá se refleja el impacto que genera el lanzar dos o más elementos al segmento espacial, destacando que lanzar tres elementos tras el primer año genera un impacto sustantivo de un ~5% a un 85%. Mientras más elementos lances al inicio, más disruptivo es el cambio y más rápido es el “aplanamiento” de la

curva de confiabilidad entre lanzamientos mientras se aproxima al valor idea. Su desventaja principal es que independiente de ello la confiabilidad temprana durante el primer año no es controlable. En términos económicos, presentado en la Figura 5.3, no presenta muchos cambios con respecto a la primera, debido a que el aumento de una variable o de otra lo importante es conocer la cantidad total de elementos lanzados, que define en gran medida el costo del proyecto.

5.1.2 Redundancia Activa Elemental

La estrategia de redundancia activa elemental consiste en aprovecharse en primera instancia del sistema k-out-of-n, lanzando al inicio de la misión más elementos del segmento que se requieren. Esto genera una redundancia cuyos elementos están activos y aportan a la misión, generando un barrido más amplio del requerido por la misión. Esta estrategia económicamente puede implicar un beneficio a largo plazo ya que, al tener la capacidad de observación más grande que la requerida inicialmente, el servicio de EO puede ser contratado por una entidad que lo requiera.

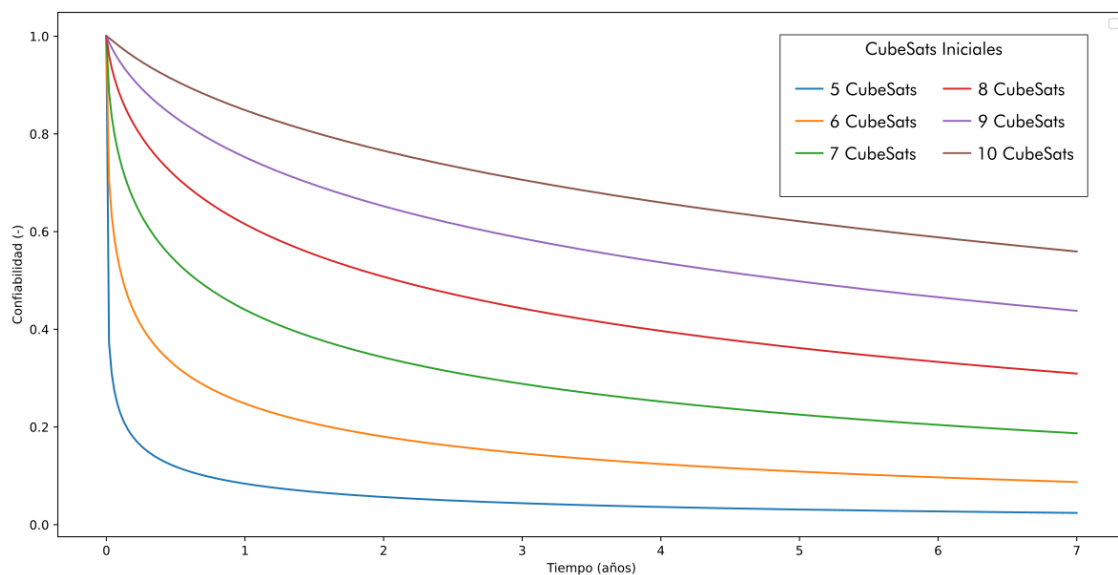


Figura 5.4 Gráfica de Curvas de Confiabilidad con Distinta Cantidad de Elementos Iniciales (autoría propia)

Se observa en la siguiente Figura 5.4 que el aumento de lanzamiento de elementos iniciales logra mitigar exitosamente la *infant mortality*, otorgando una mayor confianza al sistema al poseer redundancia activa a lo largo de la misión. Como ejemplo, para una misión de mínimo. Aunque ralentice el desgaste de la misión a lo largo del tiempo, no lo detiene. Por ende, es necesario conocer también los efectos al incluir el lanzamiento en fase en esta estrategia, ilustrado en la Figura 5.5.

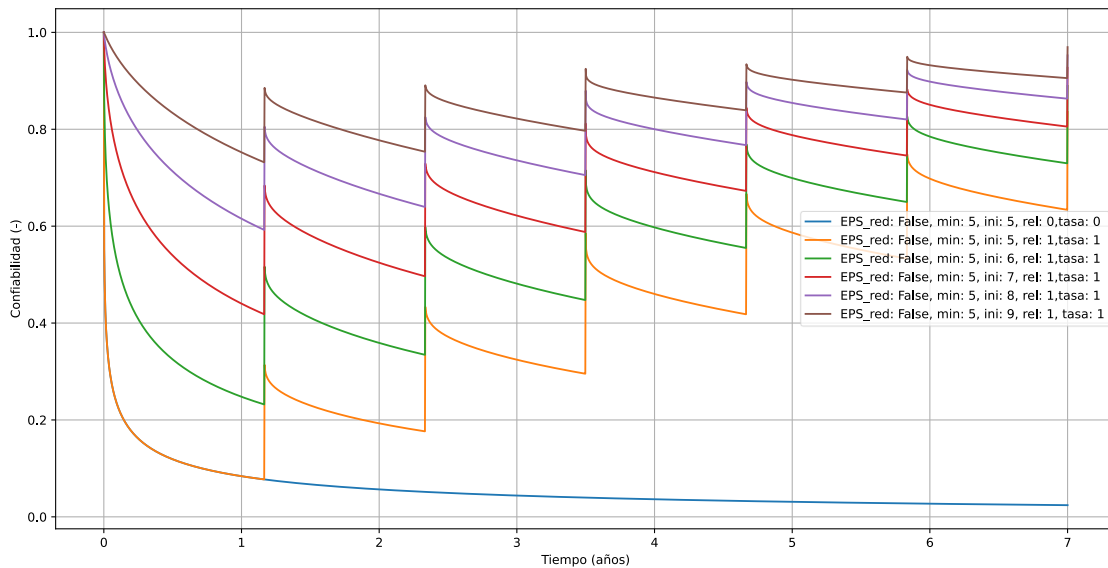


Figura 5.5 Gráfica Curvas de Confiabilidad y Tradespace Según Cantidad de Elementos Iniciales (autoría propia)

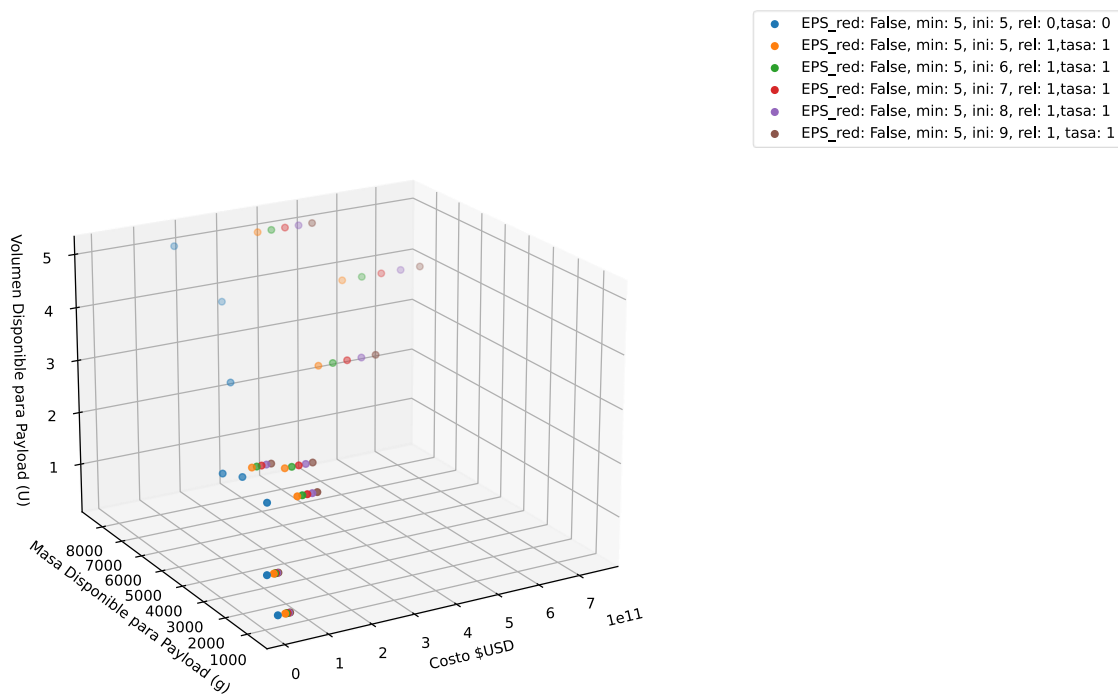


Figura 5.6 Gráfica Tradespace Según Cantidad de Elementos Iniciales (autoría propia)

En la Figura 5.5 se muestra la ventaja de combinar este y los dos efectos anteriores. Bajo este ejemplo, un segmento espacial con un porcentaje mínimo de confiabilidad de un 60%, que requiera mínimo 5 elementos espaciales para operar, puede asegurarse la operabilidad con lanzar 8 elementos inicialmente y una vez al año se relanza un elemento más, manteniendo el sistema operativo en el periodo de misión. Esta estrategia presenta una ventaja sustantiva en comparación a las mostradas previamente, ya que se encarga de la problemática de la baja confiabilidad al primer año y manteniendo un costo del proyecto menor para distintas configuraciones de *CubeSats*, mostrado en la Figura 5.6.

5.1.3 Redundancia en EPS

Como se ha comentado en el capítulo anterior, varios autores establecen como estrategia primordial para mejorar la confiabilidad de los elementos *CubeSat* la inclusión de redundancias en el subsistema EPS. Esto en la práctica se traduce en un sistema standby que, para efectos de simplificación, es un sistema standby perfecto.

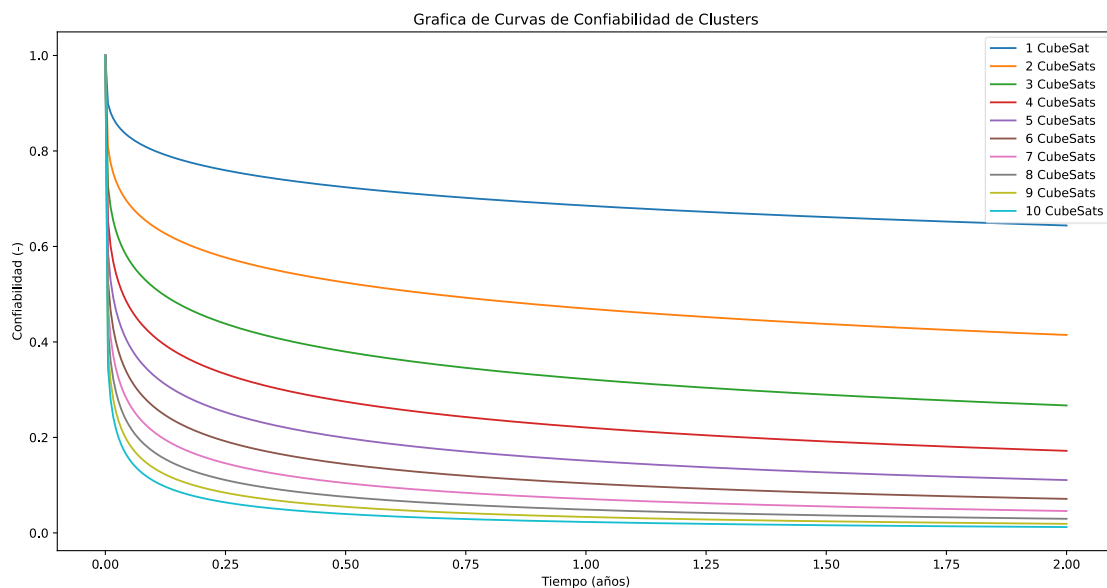


Figura 5.7 Gráfica Curvas de Confiabilidad Según Cantidad de Elementos con Redundancia en EPS (autoría propia)

En la Figura 5.7 se observa que, en comparación con la Figura 5.1, presenta una mejora inicial pero mientras más elementos agregas, menos significativa es tal mejora. Ejemplo se ve que para un sistema de 3 elementos con redundancia al primer año presenta una confiabilidad del 50% versus un ~22% sin redundancia, en comparación uno con seis elementos con redundancia presenta una confiabilidad de ~10% versus ~5% sin redundancia. Como estrategia por sí sola no es suficiente, hace falta combinarlo con otras estrategias. Se grafican las estrategias anteriores con redundancia en EPS.

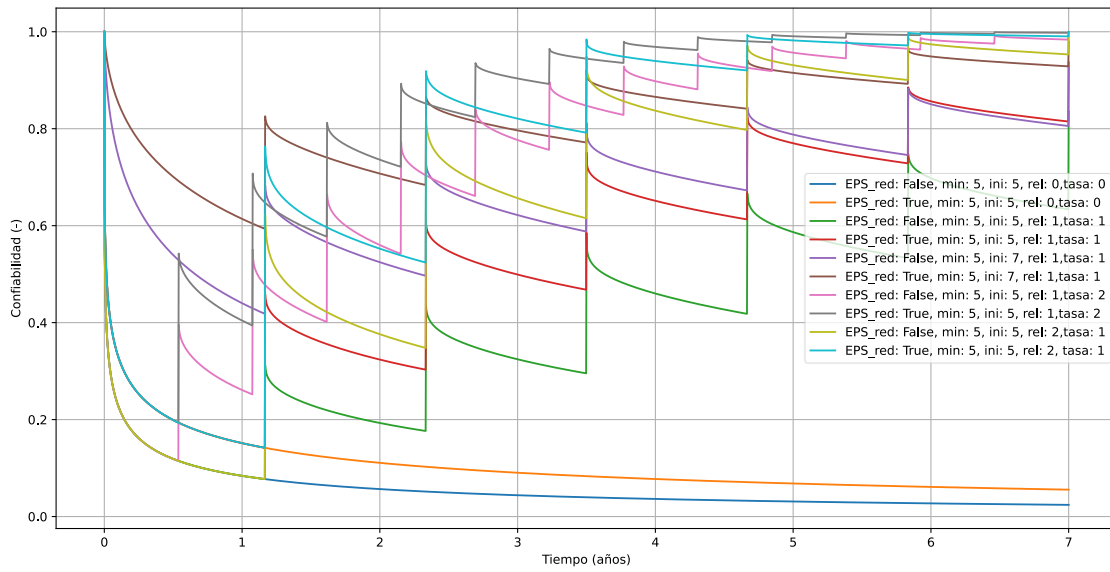


Figura 5.8 Gráficas Curvas de Confiabilidad de Estrategias de Mitigación con o sin Redundancia en EPS (autoría propia)

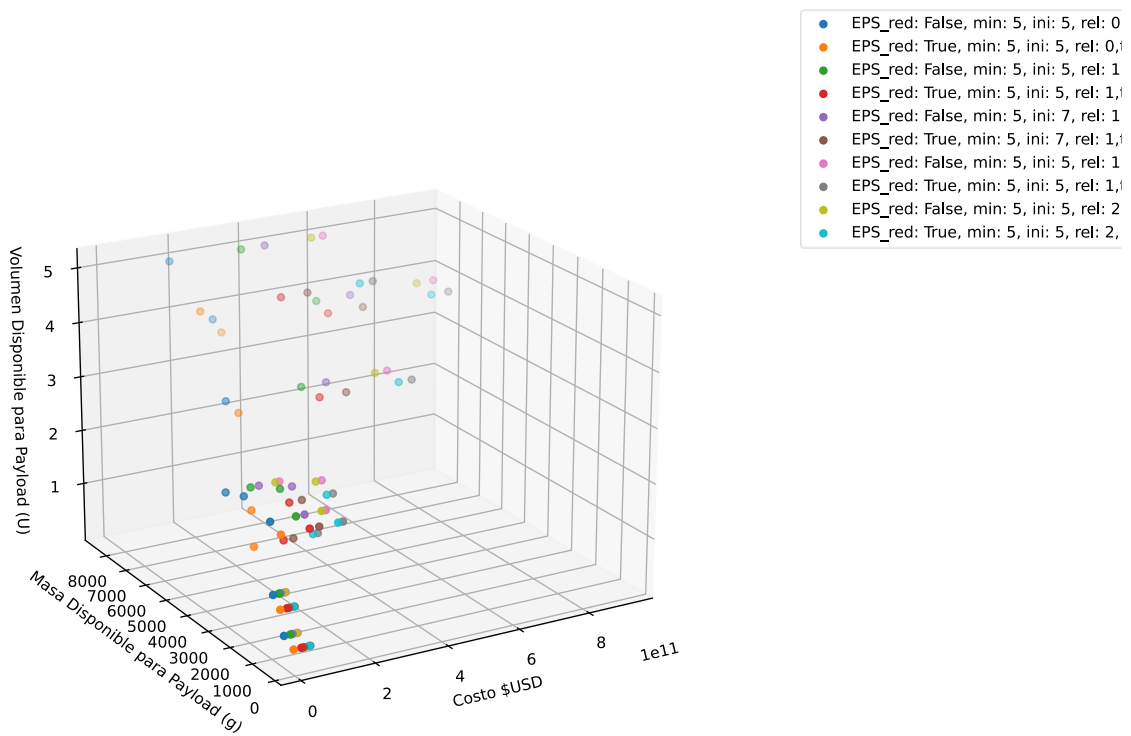


Figura 5.9 Gráfica de Tradespace de Estrategias de Mitigación con o sin Redundancia en EPS (autoría propia)

Se observa que la forma entre las configuraciones con y sin redundancia no cambia, más no su aumento. Este crecimiento se debe a que el efecto de la redundancia altera netamente la estructura interna del elemento espacial y no la del segmento, motivo en que este método permite darle un beneficio a todas las estrategias de mitigación. También se aprecia comparativamente las distintas configuraciones vistas previamente, con sus ventajas y desventajas en recuperación de confiabilidad, el tiempo de recuperación tras el primer año y los efectos de este en sus *Systems Engineering Envelopes*. Para la Figura 5.9 los efectos de la redundancia en EPS permiten, aunque disminuyendo el espacio disponible para el *payload*, la posibilidad que la configuración pueda operar con un 60% de confiabilidad mínimo con 7 elementos iniciales y no 8, como se planteaba en la sección anterior, reduciendo el costo total del proyecto.

5.2 Discusión

Resumiendo, en los efectos producidos por distintas estrategias, se encontraron que no todas generan los mismos efectos. Mientras que los lanzamientos en fase y aumento de elementos iniciales alteran disruptivamente la curva de confiabilidad del *cluster*, otros como la redundancia en EPS del elemento espacial no afecta la forma sino su fondo, incrementando proporcionalmente a cada sector. También que no existe una estrategia mejor que la otra, sino que el complemento permite un aseguramiento a largo plazo de la misión, estableciendo que toda prolongación de una misión requiere una mantención a esta. En los casos cuya reparación no es factible, la renovación de nuevos equipos al segmento espacial contribuye sustancialmente a su supervivencia. También es muy relevante y útil la adición del análisis de *Systems Engineering Envelopes* en conjunto con el de confiabilidad, ya que se puede vislumbrar íntegramente las configuraciones, estableciendo que no necesariamente la optimización de confiabilidad del segmento implica una optimización del costo o la disponibilidad de espacio para *payload*.

A modo de reflexión, la arquitectura de por sí es adaptable a múltiples usos. Desde *clusters* de satélites, sean o no estándares *CubeSats*, incluso para etapas más avanzadas del proyecto. Esto es debido a que, además de la estructura misma del segmento espacial, solamente depende de la curva de confiabilidad del elemento en cuestión. También es aplicable para un trabajo futuro expandir el concepto de la arquitectura, incorporando elementos distintos, diferentes condiciones de misión y trayectoria donde afecte la curva de confiabilidad, los lanzamientos irregulares y reprogramación de estos, la posibilidad que estos lanzamientos fallen, puesta en retiro de los elementos tras cierto tiempo, e incluso replantear la estructura misma del segmento y aproximándola a una relación más compuesta que uno k-out-of-n dependiente de este tipo de DSM, entre muchas otras. Cualquier sistema que pueda ser aproximado a una dinámica binaria (funcional y no funcional) y a una lógica binaria entre sus componentes puede ser moldeado a este tipo de arquitectura.

CAPÍTULO 6: Conclusiones

En este capítulo se presentan las conclusiones de los resultados obtenidos y se proponen mejoras para el futuro de la arquitectura creada en este proyecto.

6.1 Conclusiones

- Con base en el trabajo realizado se determinaron los parámetros y componentes necesarios para la estimación de confiabilidad y System Engineering Envelopes de un *cluster* compuesto por *CubeSats*. Esto se realizó mediante la definición exhaustiva de cada concepto y a acotación del problema a una arquitectura aproximable. Los parámetros y componentes necesarios consistieron en la cantidad de elementos a lanzar, las estrategias de mitigación a revisar, las curvas de confiabilidad de los elementos *CubeSats* y de sus subsistemas (Bouwmeester et al., 2022).
- Con base en el trabajo realizado se estudiaron los trabajos previos y conceptos relevantes, definidos en el Marco Teórico y Estado del Arte.
- Con base en el trabajo realizado se diseñó una arquitectura definitiva, a partir de la inicial plantada en el PIA y acotaciones desde el Estado del Arte, que permitieron la traducción de las estrategias de mitigación de fallas en las curvas de confiabilidad y *System Engineering Envelopes* y conocer sus efectos.
- Con base en el trabajo realizado y los resultados se simuló una misión EO de 5 a 7 años, por medio de un *cluster* de *CubeSats* (Torres, 2017) , con un *cluster* de segmento espacial compuesto inicialmente de cinco elementos, requiriendo esos cinco para lograr un barrido diario de terreno de $7.5e6 \text{ km}^2$. A partir de estos parámetros de misión se obtuvieron curvas de confiabilidad y *Systems Engineering Envelopes* para su posterior análisis.
- Con base en los resultados se evaluaron, según los efectos generados, a las propias curvas de confiabilidad, apuntando sus ventajas y desventajas frente a las otras, recopilando los resultados en el Capítulo 5.
- Con base en los resultados se concluye que las estrategias de mitigación presentadas, aunque por sí solas presentan ventajas particulares, en conjunto permiten la viabilidad de una misión extensa y con varios elementos consigo. A su vez, se permite estimar mediante especificaciones de *bus* COTS del mercado una primera estimación de la masa y el volumen disponible para el *payload*, así como el costo del proyecto hasta ese momento.
- Con base en los resultados se concluye que la integración del análisis de confiabilidad y de los *Systems Engineering Envelopes* del segmento espacial a diseñar en las fases preliminares permite vislumbrar en primera instancia las ventajas de ciertas configuraciones al aplicarles métodos de mitigación de confiabilidad, con el objetivo de asegurar la sobrevivencia de este.

6.2 Trabajos futuros

- La necesidad de implementar nuevas herramientas que afecten la confiabilidad de tales como retiro programado de elementos espaciales, los lanzamientos irregulares, probabilidad de falla de lanzamiento y periodo de misión.
- Aunque la arquitectura está pensada para el análisis de *CubeSats*, en las fases preliminares de un proyecto el usuario puede querer revisar la confiabilidad de otros modelos de micro o nanosatélites. Por ende, es importante que el usuario pueda decidir qué tipo de curva de confiabilidad quiere usar.
- La expansión de la arquitectura a otras definiciones de DSM que Le Moigne establece en el trabajo citado previamente. A consecuencia de esto, dependiendo el tipo de DSM que uno elija, corresponderá a una relación entre componentes establecida por (Billinton & Allan, 1992), permitiendo analizar y comparar una gama más amplia de DSMs como segmentos espaciales.
- La operabilidad de la arquitectura es relevante para su comprensión y correcta manipulación. Aunque nunca está demás la implementación de una Interfaz Gráfica de Usuario (GUI) permitiría una visualización mucho más clara de la información y de las acciones disponibles de esta.

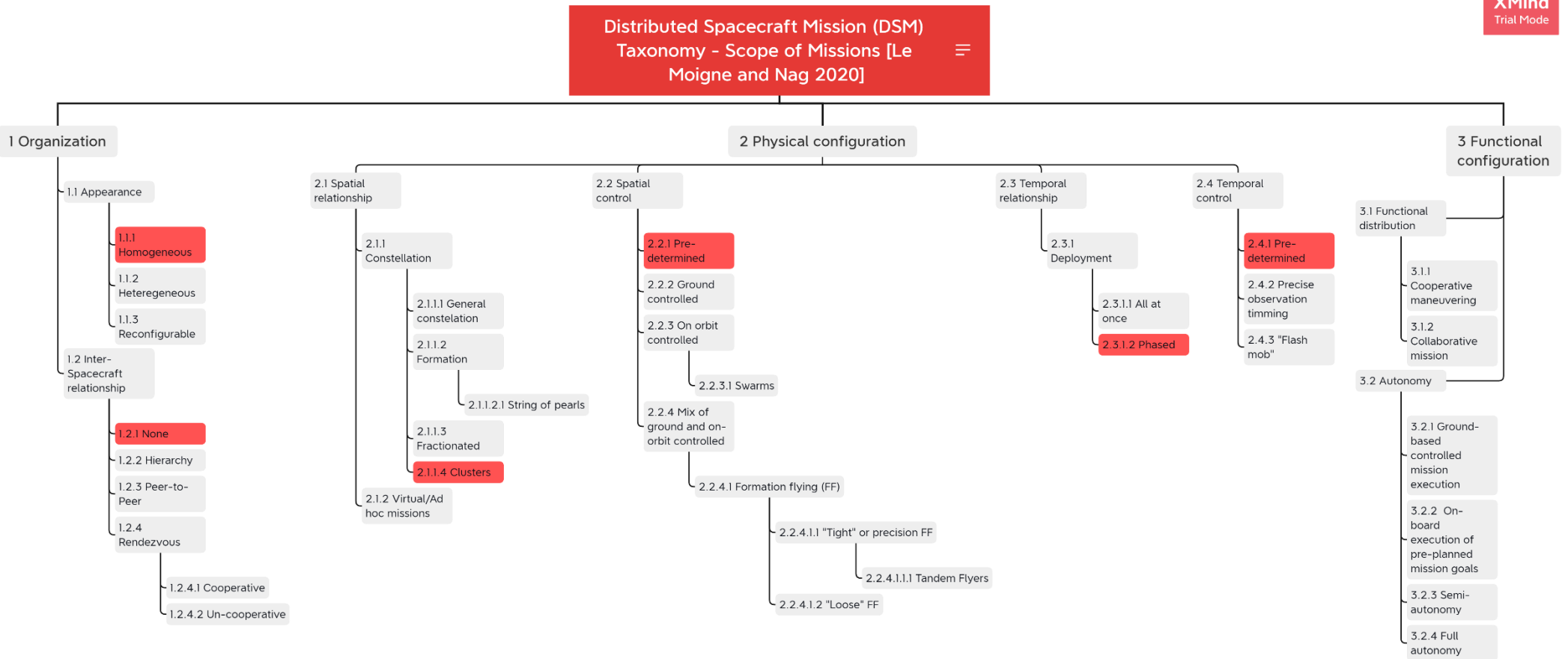
Referencias

- Billinton, R., & Allan, R. N. (1992). *Reliability Evaluation of Engineering Systems* (2nd ed.). Springer US. <https://doi.org/10.1007/978-1-4899-0685-4>
- Bouwmeester, J., Aalbers, G. T., Ubbels, W. J., Bouwmeester, J., Aalbers, G. T., & Ubbels, W. J. (2008). Preliminary Mission Results and Project Evaluation of the Delfi-C3 Nano-satellite. *ESASP*, 660, 25. <https://ui.adsabs.harvard.edu/abs/2008ESASP.660E..25B/abstract>
- Bouwmeester, J., Langer, M., & Gill, E. (2017). Survey on the implementation and reliability of CubeSat electrical bus interfaces. *CEAS Space Journal*, 9(2), 163–173. <https://doi.org/10.1007/S12567-016-0138-0/FIGURES/12>
- Bouwmeester, J., Menicucci, A., & Gill, E. K. A. (2022). Improving CubeSat reliability: Subsystem redundancy or improved testing? *Reliability Engineering and System Safety*, 220. <https://doi.org/10.1016/j.ress.2021.108288>
- Brian Dunbar. (2014, September 25). *Constellation Observing System for Meteorology Ionosphere and Climate (COSMIC)*. National Aeronautics and Space Administration. <https://www.nasa.gov/directorates/heo/scan/services/missions/earth/COSMIC.html>
- Bridgman, T., & Fox, K. (2016, May 12). *MMS First Results*. NASA Scientific Visualization Studio. https://svs.gsfc.nasa.gov/12239#media_group_96926
- Cote, J., Pierotti, S., Topczewski, S., Poghosyan, A., Lluch, I., Matevosyan, H., Lamb, A., Moreno, C. A., Taylor, C., Golkar, A., Cote, J., Mathieu, S., Pierotti, S., Grave, J., Narkiewicz, J., Topczewski, S., Sochacki, M., Lancheros, E., Park, H., & Camps, A. (2018). *Unified Classification for Distributed Satellite Systems*. <https://www.researchgate.net/publication/328020230>
- Devaraj, K. (2021, June 9). *Planet's Communication Network*. Planet Labs; Planet Labs. *Earth Observation*. (2022). European Commission. https://joint-research-centre.ec.europa.eu/scientific-activities-z/earth-observation_en
- ECSS Secretariat. (2012). *ECSS System: Glossary of Terms*.
- Federal Aviation Administration (FAA). (2018). *The Annual Compendium of Commercial Space Transportation: 2018*. <http://www.faa.gov/go/ast>
- Galvan, D. A., Hemenway, B., Welser, W. I., & Baiocchi, D. (2014). *Satellite Anomalies: Benefits of a Centralized Anomaly Database and Methods for Securely Sharing Information Among Satellite Operators*. www.rand.org
- Guo, J., Monas, L., & Gill, E. (2014). Statistical analysis and modelling of small satellite reliability. *Acta Astronautica*, 98(1), 97–110. <https://doi.org/10.1016/j.actaastro.2014.01.018>

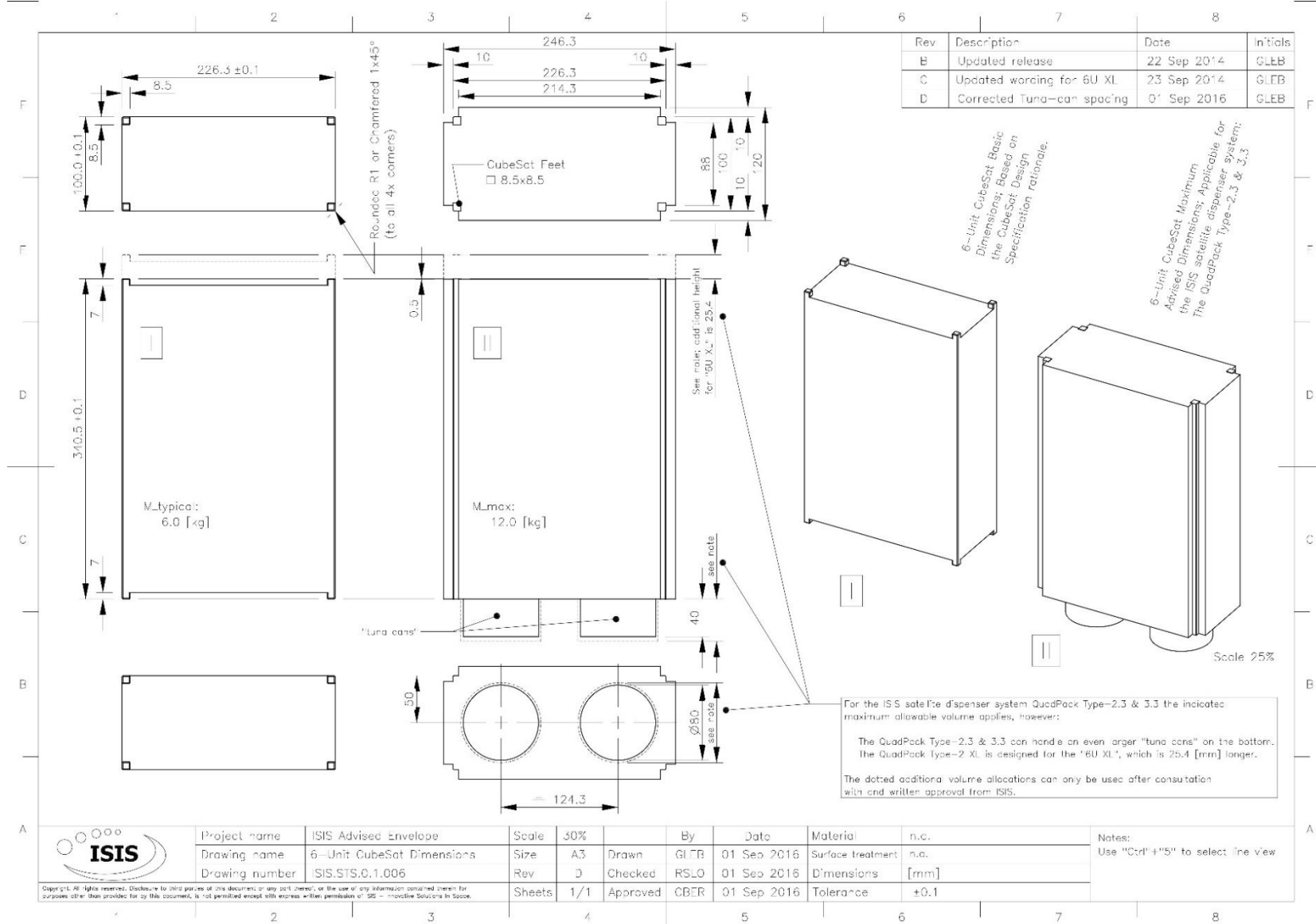
- Hevner, R., Holemans, W., Puig-Suari, J., & Twiggs, R. (2011). An Advanced Standard for CubeSats. In *25th Annual AIAA/USU Conference on Small Satellites*.
- John Uri. (2019, December 4). *45 Years Ago, Pioneer 11 Explore Jupiter*. NASA Johnson Space Center.
- Kaminskiy, M., Kashem, N., & GSFC/Code 371. (2015). *CubeSat Data Analysis Revision*.
- Kaufmann, A. (Arnold), Grouchko, Daniel., & Cruon, R. (1977). *Mathematical models for the study of the reliability of systems*. 221.
- Kiselyov, I. (2020). *Model-Based Reliability Engineering: A methodology to enhance reliability analysis in student-based CubeSat design projects* [Master Thesis, Delft University of Technology]. <http://repository.tudelft.nl/>.
- Langer, M., & Bouwmeester, J. (2016). *Reliability of CubeSats - Statistical Data, Developers' Beliefs and the Way Forward*.
- le Moigne, J., Adams, J. C., & Nag, S. (2020). A New Taxonomy for Distributed Spacecraft Missions. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 872–883. <https://doi.org/10.1109/JSTARS.2020.2964248>
- Moe, M. A., & Sullivan, E. A. (2017). *SmallSat Reliability Technology Interchange Meeting (TIM) Summary*.
- NanoSats Lab. (2020). *FSSCat*. Universitat Politècnica de Catalunya. <https://nanosatlab.upc.edu/en/missions-and-projects/fsscat>
- NASA Office of Communications. (1972, February 26). *ARC-1972-AC72-2134*. images-assets.nasa.gov/image/ARC-1972-AC72-2134/ARC-1972-AC72-2134~orig.jpg
- News Desk. (2022, September 22). *Spire Expands Contract with NASA for Earth Observation Data*. Geospatial World. <https://www.geospatialworld.net/news/spire-nasa-earth-observation/>
- O'Connor, P. D. T., & Kleyner, Andre. (2012). *Practical reliability engineering*. Wiley.
- Pielke, R. A. J. (2008). The Rise and Fall of the Space Shuttle. *American Scientist*, 96(5), 32–33.
- Poghosyan, A., & Golkar, A. (2017). CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions. In *Progress in Aerospace Sciences* (Vol. 88, pp. 59–83). Elsevier Ltd. <https://doi.org/10.1016/j.paerosci.2016.11.002>
- Prasad, N. (2017, February 28). *Traditional space and new space industry in India*. Observer Research Foundation. <https://web.archive.org/web/20170312065845/http://www.orfonline.org/expert-speaks/traditional-space-and-new-space-industry-in-india/>
- Qaisar, S. U., Ryan, M. J., & Tuttle, S. L. (2016). A Framework for Small Satellite Architecture Design. *INCOSE International Symposium*, 26(1), 1747–1758. <https://doi.org/10.1002/j.2334-5837.2016.00258.x>

- Rui, B. C., & Xavier, A. L. (2019). A Unified Satellite Taxonomy Proposal Based on Mass and Size. *Advances in Aerospace Science and Technology*, 04(04), 57–73. <https://doi.org/10.4236/aast.2019.44005>
- Sheetz, M. (2022, March 23). *SpaceX raises prices for rocket launches and Starlink satellite internet as inflation hits raw materials*. CNBC. <https://www.cnbc.com/2022/03/23/spacex-raises-prices-for-launches-and-starlink-due-to-inflation.html>
- SpaceFlight Industries. (2023, February 22). *Book My Launch*. <https://spaceflight.com/book-my-launch/>
- Swartwout, M., & Jayne, C. (2017). University-Class Spacecraft by the Numbers: Success, Failure, Debris. (But Mostly Success.). *Undefined*.
- The CubeSat Program. (2022). *CubeSat Design Specification*.
- Torres, R. (2017). *Satélite de Observación de la Tierra FASat-Charlie*. <https://support.pix4d.com/hc/en->
- Union of Concerned Scientists. (2022, May 1). *UCS Satellite Database*. <https://www.ucsusa.org/resources/satellite-database>
- Villela, T., Costa, C. A., Brandão, A. M., Bueno, F. T., & Leonardi, R. (2019). Towards the thousandth CubeSat: A statistical overview. *International Journal of Aerospace Engineering*, 2019. <https://doi.org/10.1155/2019/5063145>
- Volynskaya, O. A., & Kasyanov, R. A. (2016). Launching Numerous Small Satellites – A Flourishing Business? The Case of the Russian Federation. In *Small Satellites* (pp. 83–104). Brill | Nijhoff. https://doi.org/10.1163/9789004312234_007
- Xu, Y., Yao, W., Zheng, X., & Chen, X. (2019). Satellite system design optimization based on Bayesian melding of multi-level multi-source lifetime information. *IEEE Access*, 7, 103505–103516. <https://doi.org/10.1109/ACCESS.2019.2929758>
- Yost, B., & Weston, S. (2023). *State of the Art Small Spacecraft Technology*. <http://www.sti.nasa.gov>

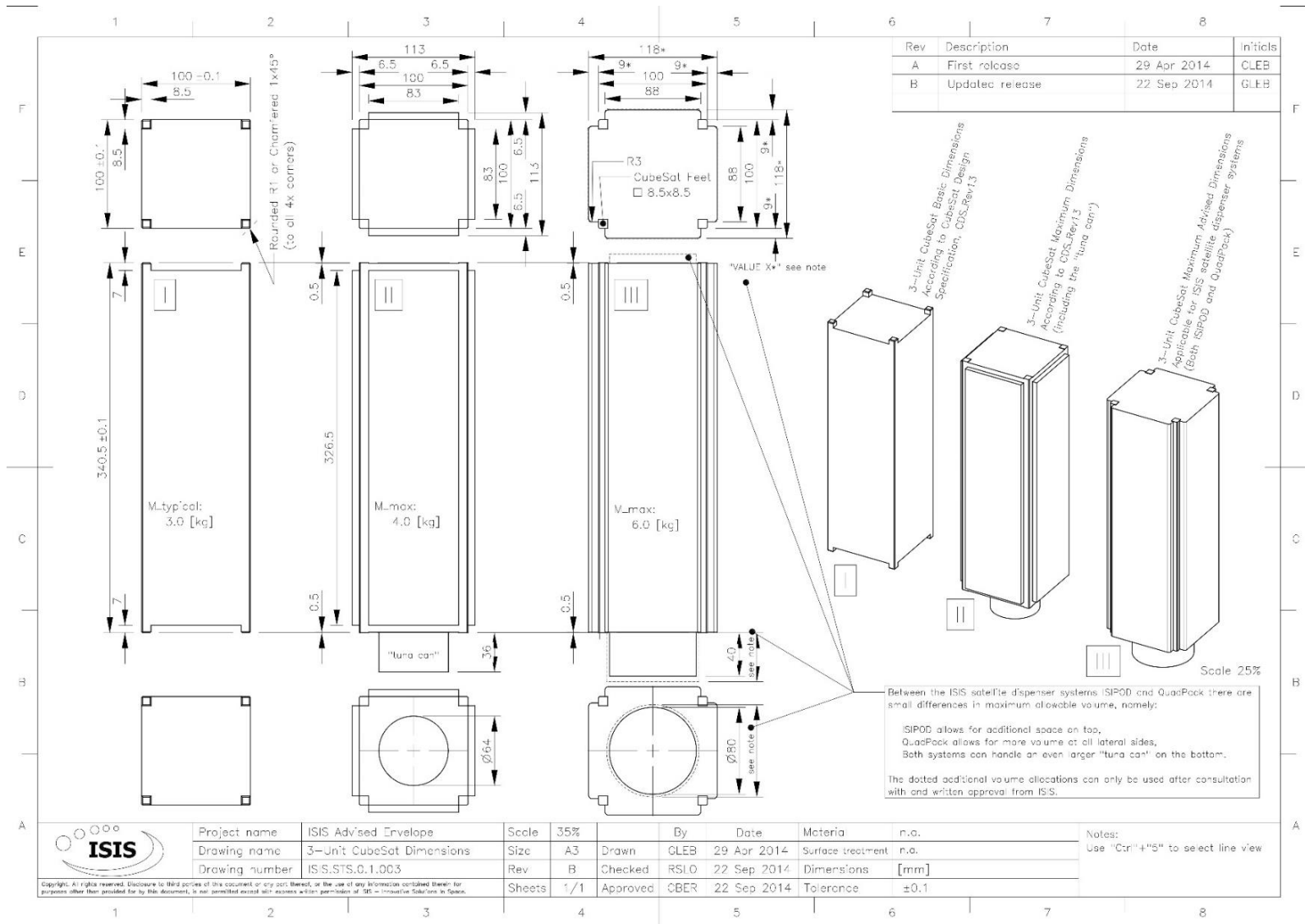
ANEXO 1: Taxonomía de la terminología DSM (le Moigne et al., 2020)



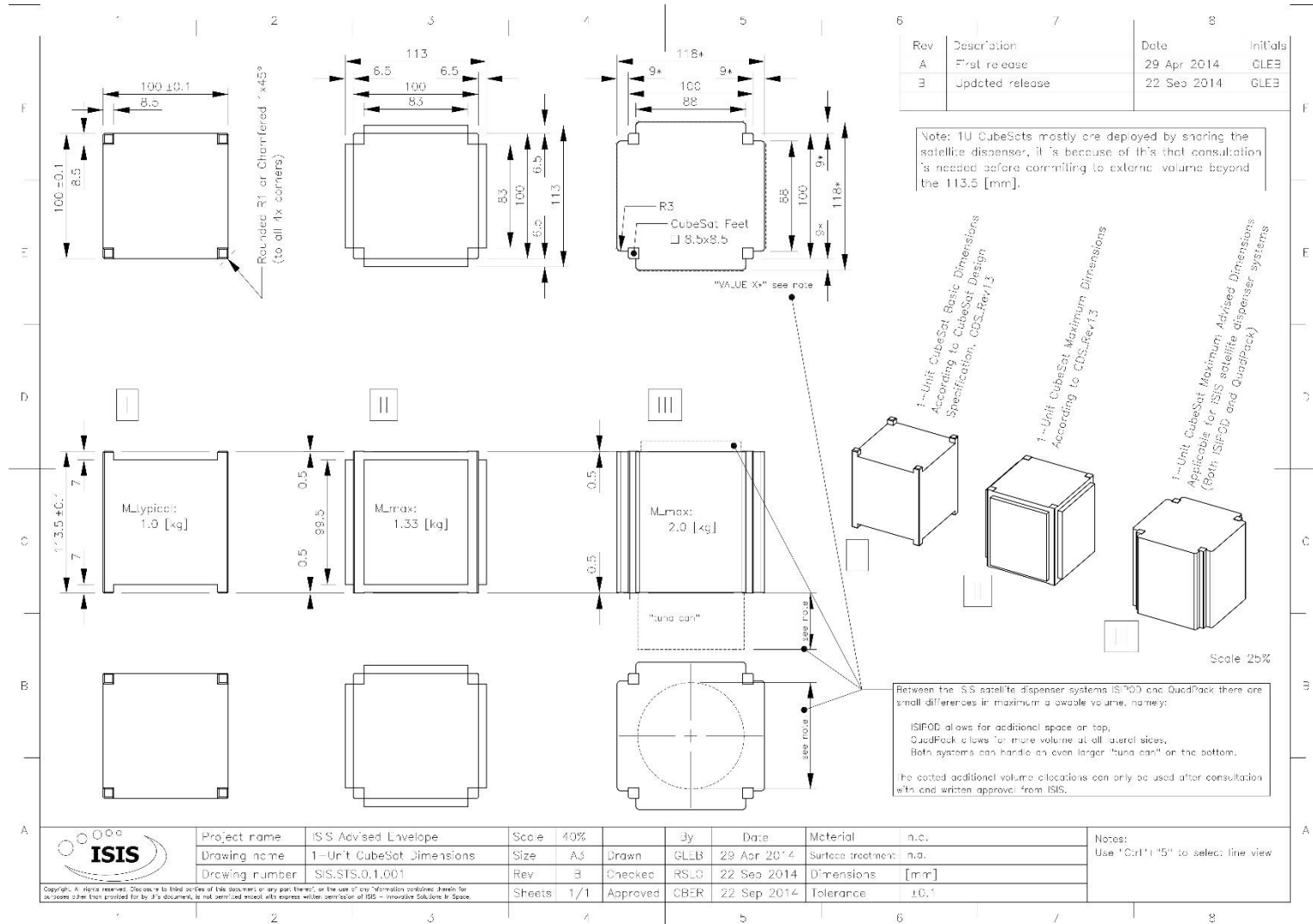
ANEXO 3 : ISISPACE 6-Unit CubeSat Dimensions



ANEXO 4: ISISPACE 3 Unit CubeSat Dimensions



ANEXO 5: ISISPACE CubeSat 1 Unit Dimensions



ANEXO 6: Main_est.py

```

from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from csv import writer

print('-----')
print('//////////////////////          //      //      //      /')
print('')
print(' ***** ***** **      **      **')
print(' ***** ***** **      **      */*****')
print(' ***** ***** **      **      *****')
print(' *****/*****')
print(' ***** ***** *****')
print('**')
print(' ***** **      **      */***** **')
print(' *****')
print(' *****')
print(' *****')
print(' **')
print(' **')
print('-----')
print('Bienvenido de nuevo')
print('Programa creado por: Nicolás Sepúlveda')
print('Para Memoria de Título')
print('-----')
print()
print()
print('-----')
R_tot = []
masa = []
volumen = []
costo = []
ciclo_et = ''
label = []
while ciclo_et != 'n' or ciclo_et != 'N':

    EPS_red = ''
    b_inicio = ''
    a_inicio = ''
    Mission_time = 7

```



```

while EPS_red != 'y' or EPS_red != 'Y' or EPS_red != 'n' or EPS_red != 'N':
    EPS_red = input('EPS posee redundancia? (y/n): ')
    if EPS_red == 'y' or EPS_red == 'Y':
        EPS_red = True
        break
    elif EPS_red == 'n' or EPS_red == 'N':
        EPS_red = False
        break
    else:
        print('ERROR. Lo ingresado no es correcto, por favor intente
nuevamente.')
        print('-----')
        ctd_sat_min = int(input('¿Cuál es la cantidad de satélites mínimos que debe
componer el cluster? Ingrese un '
                                'número: '))
        print('-----')
        ctd_sat_ini = int(input('¿Cuál es la cantidad de satélites a comenzar?
Ingrese un número: '))
        print('-----')

while a_inicio != 'y' or a_inicio != 'Y' or a_inicio != 'n' or a_inicio !=
'N':
    a_inicio = input('¿Despliegue en Fase? (y/n): ')
    print('-----')
    if a_inicio != 'y' and a_inicio != 'Y':
        from Reliability import no_phase

        x, R_sys, masa_d, volumen_d, costo_d = no_phase(EPS_red, ctd_sat_min,
ctd_sat_ini, Mission_time)
        break
    else:
        ctd_sat_rel = int(input('¿Cuál es la cantidad de satélites a
relanzar? Ingrese un número: '))
        print('-----')
        tasa_gen = int(input('¿Cuántas veces se vuelven a lanzar los CubeSats
por año? Ingrese un número: '))
        print('-----')
        from Reliability import Reliability

        x, R_sys, masa_d, volumen_d, costo_d = Reliability(EPS_red,
ctd_sat_min, ctd_sat_ini, ctd_sat_rel, tasa_gen)
        break
    ciclo_et = input('¿Quieres realizar otra configuración? (y/n): ')
    if ciclo_et == 'n' or ciclo_et == 'N':
        print('-----')
        break
    else:
        print('-----')
        R_tot.append(R_sys)

```

```

        masa.append(masa_d)
        volumen.append(volumen_d)
        costo.append(costo_d)
        label.append('EPS_red: ' + str(EPS_red) + ', min: ' + str(ctd_sat_min) +
', ini: ' + str(ctd_sat_ini) + ', rel: ' + str(ctd_sat_rel) + ', tasa: ' +
str(tasa_gen))
            continue

R_tot.append(R_sys)
masa.append(masa_d)
volumen.append(volumen_d)
costo.append(costo_d)

with open('data.csv', 'a') as file:
    writer_object = writer(file)
    writer_object.writerow([ctd_sat_min, ctd_sat_ini, ctd_sat_rel, tasa_gen])
    writer_object.writerow([R_tot])
    writer_object.writerow([masa])
    writer_object.writerow([volumen])
    writer_object.writerow([costo])
    writer_object.writerow([])
    file.close()

label.append('EPS_red: ' + str(EPS_red) + ', min: ' + str(ctd_sat_min) + ', ini:
' + str(ctd_sat_ini) + ', rel: ' + str(ctd_sat_rel) + ', tasa: ' + str(tasa_gen))

fig = plt.figure()

ax = fig.add_subplot(1, 2, 1)

for cont in range(len(R_tot)):
    ax.plot(x, R_tot[cont], label=label[cont])

ax.set_xlabel('Tiempo (años)')
ax.set_ylabel('Confiabilidad (-)')
ax.grid(True)

fig = plt.figure(2)
ax = fig.add_subplot(1, 2, 2, projection=Axes3D.name)
ax.set_ylabel('Masa Disponible para Payload (g)')
ax.set_zlabel('Volumen Disponible para Payload (U)')
ax.set_xlabel('Costo $USD')
ax.view_init(20, -120)

for count in range(len(masa)):
    ax.scatter(costo[count], masa[count], volumen[count])
fig.legend()
plt.show()

```

ANEXO 7: Reliability.py

```

import math
from decimal import Decimal
from itertools import cycle
from random import Random
from shutil import get_terminal_size
from threading import Thread
from time import sleep

import numpy as np
import scipy as sp
from matplotlib import pyplot as plt

'''
ctd_sat_min ¿Cuál es la cantidad de satélites mínimos que debe componer el
cluster?
ctd_sat_ini ¿Cuál es la cantidad de satélites a comenzar?
ctd_sat_rel ¿Cuál es la cantidad de satélites a relanzar?
tasa_gen ¿Cuántas veces se vuelven a lanzar los CubeSats por año?
'''

class Loader:
    def __init__(self, desc="Loading...", end="Done!", timeout=0.1):
        """
        A loader-like context manager

        Args:
            desc (str, optional): The loader's description. Defaults to
"Loading...".
            end (str, optional): Final print. Defaults to "Done!".
            timeout (float, optional): Sleep time between prints. Defaults to
0.1.
        """
        self.desc = desc
        self.end = end
        self.timeout = timeout

        self._thread = Thread(target=self._animate, daemon=True)
        self.steps = ["⚡", "⚡", "⚡", "⚡", "⚡", "⚡", "⚡", "⚡"]
        self.done = False

    def start(self):
        self._thread.start()
        return self

    def _animate(self):

```

```

for c in cycle(self.steps):
    if self.done:
        break
    print(f"\r{self.desc} {c}", flush=True, end="")
    sleep(self.timeout)

def __enter__(self):
    self.start()

def stop(self):
    self.done = True
    cols = get_terminal_size((80, 20)).columns
    print("\r" + " " * cols, end="", flush=True)
    print(f"\r{self.end}", flush=True)

def __exit__(self, exc_type, exc_value, tb):
    # handle exceptions with those variables ^
    self.stop()

if __name__ == "__main__":
    with Loader("Ejecutando..."):
        for i in range(10):
            sleep(0.25)

def draw(probability_distribution):
    # método de rejection sampling
    member_count = len(probability_distribution)
    step_size = 1.00 / (member_count * 1.00)

    accept = False
    r_value = 0

    while not accept:
        # generate r_temp and use it to determine R:
        random_generator = Random()
        r_temp = random_generator.random()

        bina = math.ceil(r_temp / step_size)
        binned_object = probability_distribution[bina - 1]
        r_value = binned_object['value']
        r_probability = binned_object['probability']

        # now, after we have R (r_value), draw S:
        s = random_generator.random()

        # accept or reject this observation:
        if s <= r_probability:
            accept = True

```

```

        else:
            accept = False

    return r_value

def dist(EPS_red, x):
    mu = [15.4, 11.5, 13.7, 14.3, 14.3, 9.4]
    sigma = [10, 8.39, 9.79, 9.21, 9.21, 8.18]
    theta = [2.6, 8.1, 2.6, 2.7, 2.7, 2.9]
    nu = [9.1e-5, 0.0167, 8.3e-5, 0.00011, 0.00011, 0.00011]
    rel = 1

    counter: int
    # esto tiene que entregar un número, no un array
    for counter in np.arange(len(mu)):
        if EPS_red and counter == 5:
            rel *= 1 - np.square(
                1 - sp.stats.lognorm.sf(x, sigma[counter],
scale=np.exp(mu[counter])) * sp.stats.gompertz.sf(x, nu[
counter] / theta[counter], scale=theta[counter]))
        else:
            rel *= sp.stats.lognorm.sf(x, sigma[counter],
scale=np.exp(mu[counter])) * sp.stats.gompertz.sf(x, nu[
counter] / theta[counter], scale=theta[counter])
    return rel

def rel_builder(R_elem, ctd_sat_min):
    R_s = 1
    for counter in np.arange(2 ** len(R_elem)):
        rel = 1
        bina = bin(counter)[2:].zfill(len(R_elem))
        if bina.count('1') <= len(R_elem) - ctd_sat_min:
            continue
        else:
            for k in np.arange(len(bina)):
                if bina[k] != '0':
                    rel *= 1 - R_elem[k]
                else:
                    rel *= R_elem[k]
            R_s -= rel
    return R_s

def rel_creator(t, tasa_gen, mission_time, sat_ctd_ini, sat_ctd_rel, EPS_red):
    Ans = np.ones(sat_ctd_ini) * dist(EPS_red, t)
    x = np.linspace(0, mission_time, (mission_time * tasa_gen), dtype=str)

    for a in x:

```

```

        if Decimal(str(t)) >= Decimal(a) > Decimal('0'):
            Ans = np.append(Ans, np.ones(sat_ctd_rel) * dist(EPS_red, t -
float(a)))
        else:
            continue
    return Ans.tolist()

def no_phase(EPS_red, ctd_sat_min, ctd_sat_ini, Mission_time):
    x = np.linspace(0, Mission_time, Mission_time * 365)

    loader = Loader(desc='Espere un momento...', end='Terminado!').start()
    R_sys = []
    for t in x:
        R_elem = np.ones(ctd_sat_ini) * dist(EPS_red, t)
        R_sys.append(rel_builder(R_elem, ctd_sat_min))

    loader.stop()

    from Env_tradespace import Env_tradespace
    masa = []
    volumen = []
    costo = []

    for Tip in ['1U', '3U', '6U']:
        masa.extend(Env_tradespace(EPS_red, Tip, len(R_elem))[0])
        volumen.extend(Env_tradespace(EPS_red, Tip, len(R_elem))[1])
        costo.extend(Env_tradespace(EPS_red, Tip, len(R_elem))[2])

    return x, R_sys, masa, volumen, costo

def Reliability(EPS_red, ctd_sat_min, ctd_sat_ini, ctd_sat_rel, tasa_gen):

    Mission_time = 7
    year_to_day = 365
    x = np.linspace(0, Mission_time, Mission_time * year_to_day)

    loader = Loader(desc='Espere un momento...', end='Terminado!').start()
    R_sys = []
    for t in x:
        R_elem = rel_creator(t, tasa_gen, Mission_time, ctd_sat_ini, ctd_sat_rel,
EPS_red)
        R_sys.append(rel_builder(R_elem, ctd_sat_min))
    loader.stop()

    from Env_tradespace import Env_tradespace
    masa = []
    volumen = []
    costo = []

```

```
for Tip in ['1U', '3U', '6U']:  
    masa.extend(Env_tradespace(EPS_red, Tip, len(R_elem))[0])  
    volumen.extend(Env_tradespace(EPS_red, Tip, len(R_elem))[1])  
    costo.extend(Env_tradespace(EPS_red, Tip, len(R_elem))[2])  
  
return x, R_sys, masa, volumen, costo
```

ANEXO 8: Env_tradespace.py

```

import pandas as pd

def Env_tradespace(EPS_red, CubeSt_Tipo, sat_ctd_total):
    Data = pd.read_csv("Datos Proveedores Subsistemas.csv", sep=';')
    data = Data[Data[CubeSt_Tipo].isin(['yes'])]
    mm_to_U_conv = 1000000
    broker_data = {'1U': 70000, '3U': 145000, '6U': 295000}

    if CubeSt_Tipo == '1U':

        isis_data = data[data['Nombre'].isin(['ISISPACE'])].reset_index() #
tiene EPS
        enduro_data = data[data['Nombre'].isin(['Endurosat'])].reset_index() #
tiene EPS

        if not EPS_red:
            masa_d = [isis_data._get_value(0, 'Masa')] +
[enduro_data._get_value(0, 'Masa')]
            volumen_d = [isis_data._get_value(0, 'Volumen')] +
[enduro_data._get_value(0, 'Volumen')]
            costo_d = [isis_data._get_value(0, 'Costo') * sat_ctd_total *
broker_data.get(CubeSt_Tipo)] + \
                [enduro_data._get_value(0, 'Costo') * sat_ctd_total *
broker_data.get(CubeSt_Tipo)]

        else:

            masa_d = [isis_data._get_value(0, 'Masa') - isis_data._get_value(1,
'Masa')] + \
                [enduro_data._get_value(0, 'Masa') -
enduro_data._get_value(1, 'Masa')]

            volumen_d = [isis_data._get_value(0, 'Volumen') -
isis_data._get_value(1, 'Volumen') / mm_to_U_conv] + \
                [enduro_data._get_value(0, 'Volumen') -
enduro_data._get_value(1, 'Volumen') / mm_to_U_conv]

            costo_d = [(isis_data._get_value(0, 'Costo') +
isis_data._get_value(1,
Costo')) * sat_ctd_total * broker_data.get(
                CubeSt_Tipo)] + \
                [(enduro_data._get_value(0, 'Costo') +
enduro_data._get_value(1,

```



```

'Costo')) * sat_ctd_total * broker_data.get(
    CubeSt_Tipo)]

    elif CubeSt_Tipo == '3U':
        isis_data = data[data['Nombre'].isin(['ISISPACE'])].reset_index() #
tiene EPS
        orb_data = data[data['Nombre'].isin(['OrbAstro'])].reset_index() # tiene
EPS
        enduro_data = data[data['Nombre'].isin(['Endurosat'])].reset_index() #
tiene EPS

        if not EPS_red:
            masa_d = [isis_data._get_value(0, 'Masa')] + [orb_data._get_value(0,
'Masa')] + \
                    [enduro_data._get_value(0, 'Masa')]
            volumen_d = [isis_data._get_value(0, 'Volumen')] +
[orb_data._get_value(0, 'Volumen')] + \
                    [enduro_data._get_value(0, 'Volumen')]
            costo_d = [isis_data._get_value(0, 'Costo') * sat_ctd_total *
broker_data.get(CubeSt_Tipo)] + \
                    [orb_data._get_value(0, 'Costo') * sat_ctd_total *
broker_data.get(CubeSt_Tipo)] + \
                    [enduro_data._get_value(0, 'Costo') * sat_ctd_total *
broker_data.get(CubeSt_Tipo)]

        else:
            masa_d = [isis_data._get_value(0, 'Masa') - isis_data._get_value(1,
'Masa')] + \
                    [orb_data._get_value(0, 'Masa') - orb_data._get_value(1,
'Masa')] + \
                    [enduro_data._get_value(0, 'Masa') -
enduro_data._get_value(1, 'Masa')]

            volumen_d = [isis_data._get_value(0, 'Volumen') -
isis_data._get_value(1, 'Volumen') / mm_to_U_conv] + \
                    [orb_data._get_value(0, 'Volumen') -
orb_data._get_value(1, 'Volumen') / mm_to_U_conv] + \
                    [enduro_data._get_value(0, 'Volumen') -
enduro_data._get_value(1, 'Volumen') / mm_to_U_conv]

            costo_d = [(isis_data._get_value(0, 'Costo') +
isis_data._get_value(1,
Costo')) * sat_ctd_total * broker_data.get(
    CubeSt_Tipo)] + \
                    [(orb_data._get_value(0, 'Costo') + orb_data._get_value(1,

```

```

sto')) * sat_ctd_total * broker_data.get(
    CubeSt_Tipo)] + \
    [(enduro_data._get_value(0, 'Costo') +
enduro_data._get_value(1,
    'Costo')) * sat_ctd_total * broker_data.get(
    CubeSt_Tipo)]

    elif CubeSt_Tipo == '6U':

        isis_data = data[data['Nombre'].isin(['ISISPACE'])].reset_index() #
tiene EPS

        orb_data = data[data['Nombre'].isin(['OrbAstro'])].reset_index() # tiene
EPS

        enduro_data = data[data['Nombre'].isin(['Endurosat'])].reset_index() #
tiene EPS

        if not EPS_red:
            masa_d = [isis_data._get_value(0, 'Masa')] + [orb_data._get_value(0,
'Masa')] + \
                [enduro_data._get_value(0, 'Masa')]
            volumen_d = [isis_data._get_value(0, 'Volumen')] + \
                [orb_data._get_value(0, 'Volumen')] + \
                [enduro_data._get_value(0, 'Volumen')]
            costo_d = [isis_data._get_value(0, 'Costo') * sat_ctd_total *
broker_data.get(CubeSt_Tipo)] + \
                [orb_data._get_value(0, 'Costo') * sat_ctd_total *
broker_data.get(CubeSt_Tipo)] + \
                [enduro_data._get_value(0, 'Costo') * sat_ctd_total *
broker_data.get(CubeSt_Tipo)]

        else:

            masa_d = [isis_data._get_value(0, 'Masa') - isis_data._get_value(1,
'Masa')] + \
                [orb_data._get_value(0, 'Masa') - orb_data._get_value(1,
'Masa')] + \
                [enduro_data._get_value(0, 'Masa') -
enduro_data._get_value(1, 'Masa')]

            volumen_d = [isis_data._get_value(0, 'Volumen') -
isis_data._get_value(1, 'Volumen') / mm_to_U_conv] + \
                [orb_data._get_value(0, 'Volumen') -
orb_data._get_value(1, 'Volumen') / mm_to_U_conv] + \
                [enduro_data._get_value(0, 'Volumen') -
enduro_data._get_value(1, 'Volumen') / mm_to_U_conv]

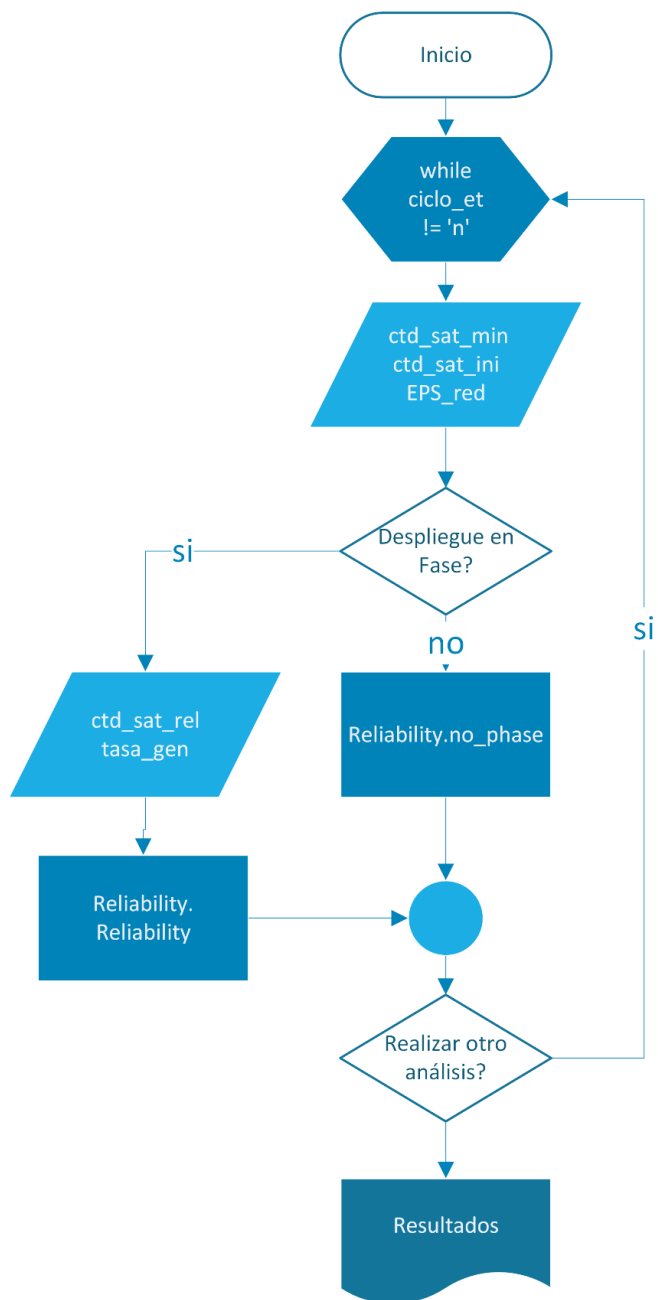
```

```
        costo_d = [(isis_data._get_value(0, 'Costo') +
isis_data._get_value(1,
Costo')) * sat_ctd_total * broker_data.get(
        CubeSt_Tipo)] + \
        [(orb_data._get_value(0, 'Costo') + orb_data._get_value(1,
sto')) * sat_ctd_total * broker_data.get(
        CubeSt_Tipo)] + \
        [(enduro_data._get_value(0, 'Costo') +
enduro_data._get_value(1,
'Costo')) * sat_ctd_total * broker_data.get(
        CubeSt_Tipo)]

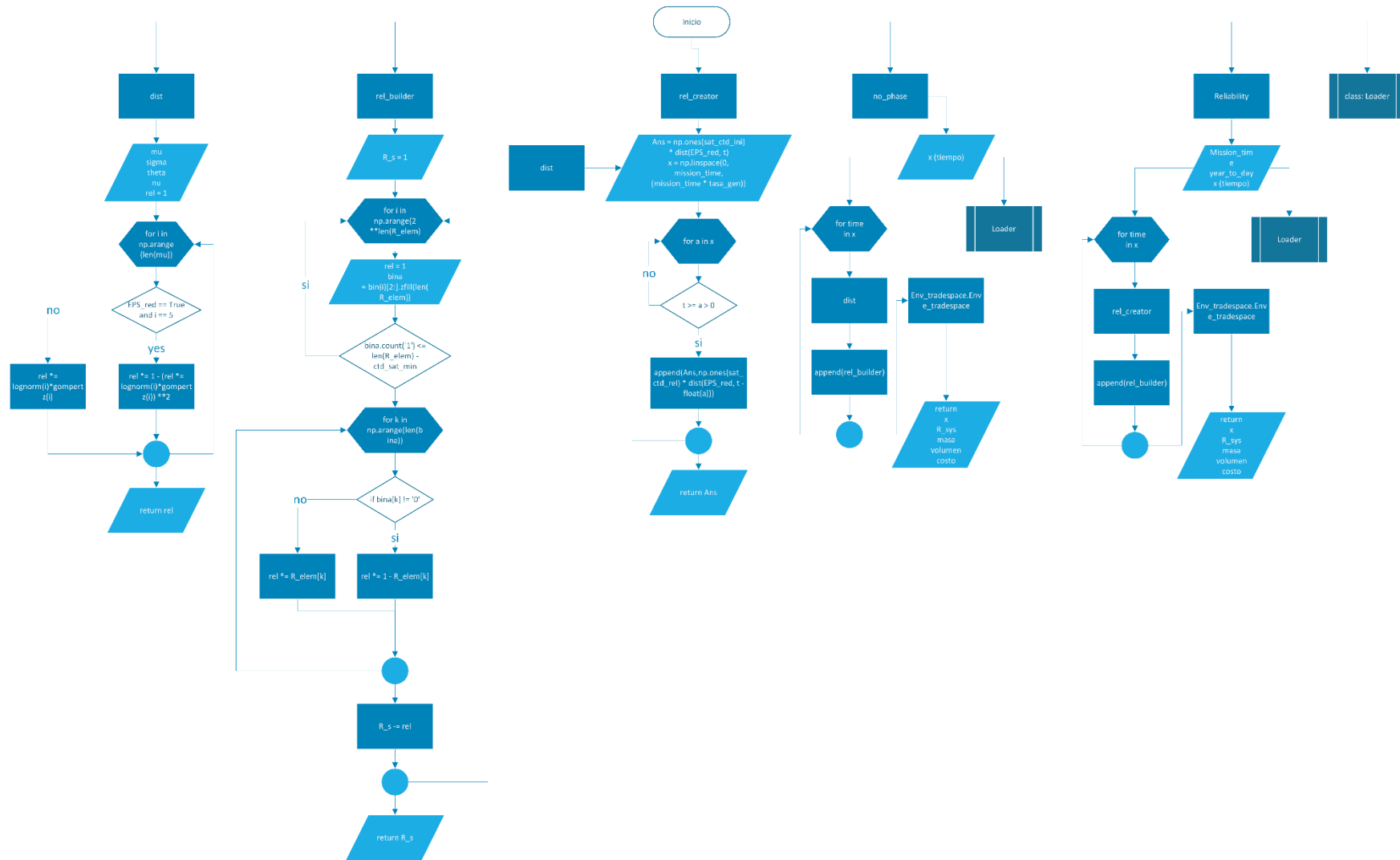
    return masa_d, volumen_d, costo_d
```

ANEXO 9: Bases de Datos de Componentes de Subsistemas

Componente	Nombre	Masa	Costo	Volumen	1U	3U	6U
Bus	ISISPACE	700	40000	0.4	yes	no	no
Bus	ISISPACE	2000	80000	2	no	yes	no
Bus	ISISPACE	6000	120000	3	no	no	yes
EPS	ISISPACE	365	9093	140870	yes	yes	yes
Bus	OrbAstro	4000	78415	2	no	yes	no
Bus	OrbAstro	8500	114608	5	no	no	yes
EPS	OrbAstro	1100	14477	700000	no	yes	yes
Bus	Endurosat	1150	37200	1	yes	no	no
Bus	Endurosat	3600	114000	2	no	yes	no
Bus	Endurosat	7800	168000	4	no	no	yes
EPS	Endurosat	283	5100	184913	yes	yes	yes

ANEXO 10: Diagrama de Flujos Main_est

ANEXO 11: Diagrama de Flujos Reliability



ANEXO 12: Diagrama de Flujo Env_tradespace