

Tecnológico de Costa Rica

Área Académica de Ingeniería en Mecatrónica

Programa de Licenciatura de Ingeniería en Mecatrónica

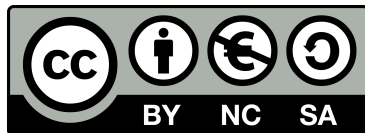


**Diseño de un sistema para detección automática de anomalías
posterior a una prueba de vibración para CubeSats**

Informe de Proyecto de Graduación para optar por el título de Ingeniero en Mecatrónica
con el grado académico de Licenciatura

Jairo Rodríguez Blanco

Cartago, 13 de marzo del 2024



Esta obra está bajo una
Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional.

Declaro que el presente avance de Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.



Jairo Rodríguez Blanco
Cartago, 13 de marzo del 2024
Céd: 207910809

INSTITUTO TECNOLÓGICO DE COSTA RICA
PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA
PROYECTO FINAL DE GRADUACIÓN
ACTA DE APROBACIÓN

El profesor asesor del presente trabajo final de graduación, indica que el documento presentado por el estudiante cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica del Instituto Tecnológico de Costa Rica para ser defendido ante el jurado evaluador, como requisito final para aprobar el curso Proyecto Final de Graduación y optar así por el título de Ingeniero(a) en Mecatrónica, con el grado académico de Licenciatura.

Estudiante: Jairo Rodríguez Blanco

Proyecto: Diseño de un sistema para detección automática de anomalías posterior a una prueba de vibración para CubeSats.



Dr. Ing. Carlos Adrián Salazar
García
Asesor

Cartago, 13 de marzo de 2024

INSTITUTO TECNOLÓGICO DE COSTA RICA
PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA
PROYECTO FINAL DE GRADUACIÓN
ACTA DE APROBACIÓN

Proyecto final de graduación defendido ante el presente jurado evaluador como requisito para optar por el título de Ingeniero(a) en Mecatrónica con el grado académico de Licenciatura, según lo establecido por el programa de Licenciatura en Ingeniería Mecatrónica, del Instituto Tecnológico de Costa Rica.

Estudiante: Jairo Rodríguez Blanco.

Proyecto: Diseño de un sistema para detección automática de anomalías posterior a una prueba de vibración para CubeSat.

Miembros del jurado evaluador



Msc. Ing. Roger Melendez Poltronieri

Jurado



Ing. Paula Elena Monge Chanto

Jurado

Los miembros de este jurado dan fe de que el presente proyecto final de graduación ha sido aprobado y cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica.

Cartago, 13 de marzo de 2024

Resumen

El presente documento detalla el diseño de un sistema autónomo de visión por computadora para la inspección visual e identificación de anomalías posterior a pruebas de vibración para CubeSats, desarrollado en el Instituto Tecnológico de Kyushu, Japón. El objetivo principal del proyecto es agilizar el proceso de inspección visual para habilitar un mayor volumen de pruebas en el laboratorio. Se construyó una estructura mecánica que aísla lumínicamente y soporta el satélite y un sistema de cámaras. Se toman fotografías de todas las caras del satélite antes y después de la prueba de vibraciones, y utilizando algoritmos de visión por computadora se encuentra cualquier anomalía de tipo grieta, rasguño o marca de torque. El sistema se controla en una computadora a través de una interfaz gráfica amigable con el usuario y produce reportes tanto de fallas, si las hay, como de certificación, en caso de que no haya, para presentar ante la Agencia Espacial Japonesa. Se obtuvo una mejora de tiempo que ronda entre el 2802 % al 5296 % con un 100 % de eficacia, disminuyendo de aproximadamente 30 minutos a 1 minuto el tiempo de inspección y redacción.

Palabras clave: CubeSats, Vibración, Pruebas Ambientales, Visión por Computadora, IA

Abstract

This document outlines the design of an autonomous computer vision system for post-vibration test visual inspection and anomaly identification for CubeSats, developed at the Kyushu Institute of Technology, Japan. The main goal of the project is to streamline the visual inspection process to enable a higher volume of tests in the laboratory. A mechanical structure was constructed for luminary isolation and satellite and camera system support. Photographs of all satellite faces are taken before and after vibration testing, and using computer vision algorithms, any cracks, scratches, or torque mark anomalies are identified. The system is controlled on a computer through a user-friendly graphical interface and generates reports of either failures, if any, or certification, in the absence of failures, to present to the Japan Aerospace Exploration Agency. A time improvement ranging between 2802% and 5296% was achieved with 100% efficacy, reducing the inspection and reporting time from approximately 30 minutes to 1 minute.

Keywords: Cubesats, Vibration, Environmental Testing, Computer Vision, AI

A Juan José Rojas, profesor y amigo

Agradecimientos

A mis compañeros y amigos David Solórzano, Olman Quirós y David Montero, entre otros, con quienes compartí la carrera y muchos otros momentos.

Un agradecimiento especial al profesor Carlos Salazar, Juan José Rojas y Mengu Cho, cuyos aportes y apoyos para el proyecto fueron indispensables.

A Andrea Zúñiga, Zompi y Niji, quienes me apoyaron en los momentos más críticos y emocionales durante la carrera y el proyecto.

A Ezra Fielding, Keenan Chatar, Tasuku Matsui, Merissa Kosiyakul, Reynel Galindo y Aekjira Kuyyakanont quienes me dieron consejos técnicos y de vida que hicieron que mi estancia en Japón fuera el mejor tiempo de mi vida, y apoyaron remotamente para pruebas finales del proyecto.

Finalmente el agradecimiento más grande a mi familia, especialmente a mis padres, por siempre soportar mis entregas de último momento con más estrés que yo mismo, y aún así apoyarme incondicionalmente en cada paso del camino.

Jairo Rodríguez Blanco

Cartago, 13 de marzo del 2024

Índice general

Índice de figuras	IV
Índice de tablas	VI
1. Introducción	1
1.1. Entorno del proyecto	1
1.2. Problema existente	1
1.2.1. Generalidades	1
1.2.2. Justificación	2
1.2.3. Síntesis del problema	2
1.3. Limitaciones del proyecto	2
1.4. Objetivo general	3
1.5. Objetivos específicos	3
1.6. Enfoque de la solución	3
1.7. Estructura del documento	5
2. Marco teórico	7
2.1. Satélites	7
2.2. Cubesats	8
2.3. Pruebas ambientales para satélites	9
2.4. Visión por computadora	10
2.5. Aprendizaje de máquina para visión por computadora	10
2.6. Técnicas tradicionales de visión por computadora	12
2.6.1. Preprocesamiento de imágenes	12
2.6.2. Umbralización	12
2.6.3. Operaciones morfológicas	13
2.6.4. Filtrado	14
2.6.5. Detección de bordes	14
2.6.6. Detección de contornos	14
2.6.7. Extracción de características	14
2.6.8. Comparación de plantillas	15
2.6.9. Transformaciones geométricas	15
2.6.10. Sustracción de imágenes	15
2.7. Obtención de imágenes	16

2.7.1.	Tamaño del sensor	16
2.7.2.	DSLR vs. Cámaras Compactas/Empotradas	17
2.7.3.	Tipos de objetivos	17
2.7.4.	Resolución	17
2.7.5.	Campo de visión (FOV)	17
2.7.6.	Profundidad de campo (DOF)	18
2.7.7.	Ruido de la imagen	18
2.7.8.	Sensibilidad ISO	18
3.	Metodología	19
3.1.	Pasos de la metodología	19
3.2.	Identificación de necesidades	20
3.3.	Generación de métricas	22
3.4.	Generación de conceptos	25
3.5.	Selección del concepto	27
3.6.	Concepto ganador	30
3.7.	Limitaciones para la validación	31
4.	Propuesta de diseño	33
4.1.	Segmento Eléctrico	33
4.1.1.	Identificación de los requerimientos	33
4.1.2.	Componentes seleccionados	34
4.1.3.	Conexión de cámaras al computador	37
4.1.4.	Iluminación	38
4.2.	Segmento Mecánico	39
4.2.1.	Identificación de los requerimientos	39
4.2.2.	Estructura principal	40
4.2.3.	Soporte del satélite	42
4.2.4.	Soportes de cámaras	43
4.2.5.	Sistema ensamblado	44
4.3.	Segmento Software	45
4.3.1.	Identificación de los requerimientos	45
4.3.2.	Ambiente de desarrollo	46
4.3.3.	Acceso y control de cámaras	47
4.3.4.	Identificador de anomalías	49
4.3.5.	Interfaz de usuario	50
4.3.6.	Generación de reportes	51
5.	Resultados y análisis	53
5.1.	Construcción del prototipo	53
5.2.	Identificador de anomalías	58
5.3.	Interfaz Gráfica	63
5.4.	Generación de reportes	65
5.5.	Estrategia de validación	66

5.6. Resultados de validación	68
5.7. Análisis económico	70
6. Conclusiones y trabajos futuros	72
6.1. Conclusiones	72
6.2. Trabajos futuros	73
Bibliografía	75
A. Ejemplo de reporte de éxito generado por la aplicación	78
B. Ejemplo de reporte de fallo generado por la aplicación	94
C. Planos mecánicos	101

Índice de figuras

1.1. Diagrama de bloques que describe el enfoque de la solución preliminar. . . .	5
2.1. Cubesats de 1, 2 y 3 unidades [8].	8
2.2. Comparación de diferentes métodos de umbralización [16].	12
2.3. Operaciones morfológicas comunes [17]	13
2.4. Canny edge y Sobel [18].	14
2.5. Dos métodos de sustracción de imágenes.	16
3.1. Fases de la metodología de Ulrich y Eppinger [25].	19
3.2. División funcional de primer nivel.	25
3.3. División funcional de segundo nivel.	26
3.4. Boceto del concepto ganador.	31
4.1. Multiplexer 4 a 1 de Arducam [27].	34
4.2. Jetson Orin Nano [28].	36
4.3. Kit del multiplexer con cámaras [27].	36
4.4. Cableado de cámaras con multiplexor y Jetson Nano. Fuente propia.	38
4.5. Kit del sistema de iluminación. [29]	39
4.6. Kit del sistema de iluminación. Fuente propia	40
4.7. Desplazamiento máximo y estrés von Mises en la viga de la puerta. Fuente propia	41
4.8. Modelo CAD del soporte del satélite. Fuente propia	43
4.9. Modelo CAD del soporte de la cámara. Fuente propia	44
4.10. Modelo CAD de ensamble completo. Fuente propia	45
5.1. Cable formado con dos cables para obtener los conectores correctos. Fuente propia	54
5.2. Jetson Orin Nano con una cámara conectada. Fuente propia	54
5.3. Primer ensamble del armazón. Fuente propia	55
5.4. Soporte y estructura de satélite temporal usados para pruebas de cámara. Fuente propia	56
5.5. Interior de la estructura final. Pendiente en este punto: soporte en acrílico, montaje de cámaras. Fuente propia	57
5.6. Interior de la estructura final. Pendiente en este punto: soporte en acrílico, montaje de cámaras. Fuente propia	58

5.7. Resultados intermedios para un caso de tornillo flojo. Fuente propia.	60
5.8. Resultados intermedios para un caso de grieta en el panel. Fuente propia. . .	61
5.9. Resultados intermedios para un caso de rasguño en el marco. Fuente propia.	62
5.10. Vista inicial de la ventana principal. Fuente propia	63
5.11. Vista final de la ventana principal. Fuente propia	64
5.12. Vista de la ventana de generación de reporte. Fuente propia	65
5.13. Tornillo flojo en la cara -Y. Fuente propia.	67
5.14. Tornillo flojo en la cara +Y. Fuente propia.	67
5.15. Tornillo flojo en la cara +Z. Fuente propia.	68
5.16. Operador realizando la prueba de validación planteada. Fuente propia . . .	69

Índice de tablas

3.1. Nivel de importancia con su respectivo significado.	21
3.2. Lista de necesidades y sus calificaciones.	21
3.3. Lista de métricas.	23
3.4. Matriz necesidades métricas.	24
3.5. Lista de conceptos para cada función.	27
3.6. Lista de combinaciones.	29
3.7. Matriz de filtrado para conceptos de detección y identificación.	29
3.8. Matriz de selección para detección de presencia y identificación.	30
4.1. Opciones de cámaras de formato seleccionado	35
5.1. Resultados de validación de tiempo ahorrado. Fuente propia.	70
5.2. Componentes y costos aproximados. Fuente propia.	71

Lista de símbolos y abreviaciones

Abreviaciones

AC-DC	Corriente Alterna - Corriente Continua
CAD	Computer Aided Design por sus singlas en inglés
CNN	Redes Neuronales Convolucionales por sus siglas en inglés
CPU	Por su inglés, Central Processing Unit
CubeSat	Formato de satélite estandarizado con forma de cubo de 10x10x10 cm
CUDA	Compute Unified Device Architecture por sus siglas en inglés, un modelo de computación paralela desarrollado por Nvidia
DOF	Profundidad de Campo, por sus siglas en inglés
Dragonfly	Nombre de satélite en desarrollo en Kyutech
DSLR	De su inglés, Digital Single-Lens Reflex camera
FOV	Campo de visión, por sus siglas en inglés
GAN	Redes Generativas Adversariales por sus siglas en inglés
GEO	Órbita geoestacionaria
GPS	Sistema de Posicionamiento Global por sus siglas en inglés
GTO	Órbita de transferencia geoestacionaria
GUI	Interfaz Gráfica de Usuario por sus siglas en inglés
I2C	Inter-Integrated Circuit por sus siglas en inglés, estándar de comunicación
IA	Inteligencia Artificial
IoT	Por su inglés, Internet of Things
ISO	Organización Internacional de Estandarización, en referencia a estándares de sensibilidad lumínica de las cámaras
Kyutech	Instituto Tecnológico de Kyushu
L4T	Linux for Tegra, versión de Ubuntu personalizado a la plataforma Jetson Nano
LaSEINE	Laboratorio de investigación, Laboratory of Lean Satellite Enterprises and In-Orbit Experiments
LEO	Órbita terrestre baja
MEO	Órbita terrestre media
MIPI CSI-2	Mobile Industry Processor Interface Camera Serial Interface 2, por sus siglas en inglés. Conector para cámaras empotradas
MPa	Mega Pascales, unidad de medida de presión y estrés interno
nU	Unidad, refiriéndose a estándar de tamaños de CubeSats
SSIM	Índice de Similaridad Estructural

SSO	Órbita sol-sincrónica
TEC	Instituto Tecnológico de Costa Rica
USB	Universal Serial Bus

Capítulo 1

Introducción

1.1. Entorno del proyecto

El proyecto consiste en desarrollar un dispositivo en el laboratorio LaSEINE del Instituto Tecnológico de Kyushu en Japón, especializado en sistemas espaciales, específicamente satélites tipo CubeSat. Utilizando una metodología ágil, el laboratorio busca agilizar el proceso de desarrollo de satélites, priorizando el tiempo y el costo sobre la confiabilidad. Además de trabajar en misiones propias lanzadas al espacio, el laboratorio participa en programas de colaboración internacionales, como el proyecto BIRDS, que proporciona a equipos de países en vías de desarrollo una plantilla de satélite para su primera misión espacial. El proyecto Irazú, primer satélite costarricense, fue beneficiario de esta iniciativa.

El Laboratorio Delta del Tecnológico de Costa Rica estableció contacto con el Laboratorio LaSEINE a través de la visita del Dr. Mengu CHO de Kyutech, quien invitó a funcionarios de la escuela de electromecánica a realizar su doctorado en Japón. Juan José Rojas Hernández, profesor de electromecánica, regresó de este programa y continuó trabajando en sistemas satelitales en el Laboratorio Delta del TEC. El laboratorio LaSEINE es el mejor equipado de todo Japón para la ejecución de pruebas ambientales en satélites.

Este proyecto se llevó a cabo en Kyutech gracias a la conexión entre el Laboratorio Delta del TEC y el Laboratorio LaSEINE de Kyutech, y el dispositivo resultante es beneficioso para ambas instituciones.

1.2. Problema existente

1.2.1. Generalidades

El proceso de verificación de CubeSats para obtener la certificación de vuelo es un procedimiento complejo y costoso tanto en términos de tiempo como de otros recursos. Existen varios tipos de pruebas. Uno de estos tipos son las pruebas ambientales. En estas, se

somete el satélite terminado a diversas condiciones ambientales similares al de su misión espacial. Una de las pruebas ambientales es la prueba de vibración, donde se instala el satélite en una superficie que procederá a vibrarlo con parámetros dados por estándares de agencias espaciales gubernamentales, para simular los efectos de un lanzamiento en un cohete. En un alto nivel, el satélite debe funcionar correctamente posterior a la prueba (prueba funcional) pero además debe someterse a una inspección visual, donde se asegura que el satélite no tiene ningún rasguño, grieta, cable flojo, tornillo flojo, o similar.

1.2.2. Justificación

En el laboratorio LaSEINE se han identificado ineficiencias significativas en el proceso de verificación y certificación de CubeSats. El tiempo y otros recursos necesarios para realizar estas pruebas son excesivos, con un promedio de varios meses y costos que alcanzan cifras significativas. Lograr reducir el tiempo que toman las pruebas implica reducir los costos de proyectos. Es posible realizar las inspecciones visuales mencionadas anteriormente de las pruebas de vibración para proyectos sencillos, como un sólo satélite tipo CubeSat. Sin embargo, la industria espacial está cada vez más utilizando constelaciones de satélites, grupos de hasta cientos de satélites idénticos, al mismo tiempo. SpaceX con Starlink y PlanetLabs con Planet son dos ejemplos de empresas que utilizan cientos o miles de satélites simultáneamente. Cada uno de estos satélites debe ser inspeccionado manualmente, lo que lleva a costos enormes, y en el caso del laboratorio LaSEINE, no se cuenta con el ancho de banda en cuanto a mano de obra para poder aceptar este tipo de proyectos durante el desarrollo de otros. Además, un operador que pase todo el día durante varios días realizando la misma inspección visual (especialmente cuando es un estudiante, como es el caso del laboratorio LaSEINE) puede verse afectado por fatiga, lo que reduce la calidad de las inspecciones. Un sistema que automatice el proceso de inspección visual es necesario.

1.2.3. Síntesis del problema

Para el proceso de certificación de satélites tipo CubeSat se requieren diversas pruebas, entre ellas están las pruebas de vibración. Estas no escalan fácilmente a volúmenes altos de satélites si se utilizan las técnicas actuales usadas por el laboratorio LaSEINE. Un sistema innovador que automatice y agilice el proceso de inspección es necesario.

1.3. Limitaciones del proyecto

El proyecto presenta diversas limitaciones, de las cuales muchas no son directamente técnicas.

Si bien el laboratorio va a financiar todos los componentes necesarios para el desarrollo

de prototipos, el proceso de adquisición de componentes toma tiempo, debido a toda la tramitología necesaria para el financiamiento. Por esto, no existe la posibilidad de comprar un componente de manera de emergencia, si fuera necesario. Deben planificarse sumamente bien las compras, tomando en cuenta posibles necesidades adicionales.

Una gran parte del proyecto es la detección de anomalías en imágenes. Por la naturaleza de este trabajo, diversas implementaciones de Inteligencia Artificial serán evaluadas durante el proceso de diseño ingenieril. La mayoría, si no todas, de los paradigmas de inteligencia artificial requieren datos para entrenamiento. En el caso de este proyecto, probablemente serán imágenes de satélites. Será muy complicado obtener un conjunto de imágenes de satélites, por lo que se limita las posibles tecnologías de detección de anomalías que se pueden usar.

La principal limitante es el tiempo. El intercambio internacional es de solo 90 días, por lo que se debe terminar el proyecto en este tiempo, o encontrar la forma de transportar el prototipo construido a Costa Rica, un desafío logístico.

1.4. Objetivo general

Diseñar un sistema automático de detección de anomalías en pruebas de vibración de satélites tipo CubeSat.

1.5. Objetivos específicos

- Emplear una metodología de diseño sistemática y estructurada para identificar y definir correctamente las necesidades, requerimientos y especificaciones del sistema de detección de anomalías, a fin de encontrar la solución óptima al problema planteado.
- Desarrollar el sistema de montaje y toma de imágenes para sistemas CubeSat considerando los principios y requisitos técnicos establecidos durante la etapa de diseño
- Implementar un sistema de detección de anomalías en imágenes antes y después de pruebas de vibración.
- Validar el sistema de detección automático de anomalías para verificar su capacidad de correctamente identificar problemas de CubeSats posterior a pruebas de vibración

1.6. Enfoque de la solución

Inicialmente, se realizó una investigación exhaustiva sobre todo lo relacionado con pruebas ambientales para la verificación y certificación de componentes de satélite, con énfasis particular a las pruebas de vibración.

Seguidamente se inició el proceso de ejecución de la metodología descrita en el presente

proyecto, específicamente en este punto, lo referente a la identificación de problemática y necesidades. La previa investigación es clave para concluir esta etapa de manera exitosa.

Anterior al inicio del diseño técnico como tal, durante la fase de planificación previa, se identificaron las partes que conformarán el sistema de detección de anomalías, específicamente el subsistema mecánico de montaje, el subsistema de toma de imágenes, los algoritmos de detección de anomalías, y todo lo referente a la generación de reportes. Al concluir esta fase se tiene claro también todo lo referente a los tipos de satélites a probar: tamaños, formas de sujeción, potenciales errores a encontrar, entre otros.

Teniendo todo este proceso claro, se procedió con el sistema de sujeción y obtención de imágenes. El sistema de sujeción sujeta el satélite, los instrumentos que vayan a tomar imágenes (una o múltiples cámaras), el satélite como tal, y cualquier otra infraestructura que se necesite para asegurar las mismas condiciones de iluminación entre fotografías.

Concluido el diseño de esta estructura, hubo tiempo disponible mientras se fabricaba la estructura y se conseguían las cámaras y otras partes. En este tiempo y posterior a la construcción del sistema, se procede a desarrollar el sistema de software que detecta las anomalías utilizando imágenes. Existen dos principales propuestas para abordar este problema: técnicas de visión por computadora tradicionales, y visión por computadora basada en inteligencia artificial. La primera tiene la ventaja que no ocupa datos de entrenamiento, pero puede ser susceptible a cambios en los satélites, iluminación, y demás. La segunda es más robusta ante estos cambios, pero requiere datos de entrenamiento. Durante la fase de ejecución de la metodología de diseño ingenieril se evaluaron estas herramientas para ver cuál es la más factible.

Finalmente, se desarrolló una aplicación de fácil uso para el usuario que tome las imágenes y utilizando estas junto con la respuesta a frecuencia determine si pasa o no la prueba de vibración. Se desarrolló un sistema de generación de reportes que demuestre estos resultados, siguiendo estándares de documentación de la agencia gubernamental a la que está asociado el satélite.

Cuando se finalizó el desarrollo de todo el sistema, se sometió a pruebas para verificar que correctamente identifica anomalías.

En la figura 1.1 se ve este plan de trabajo de una manera gráfica.

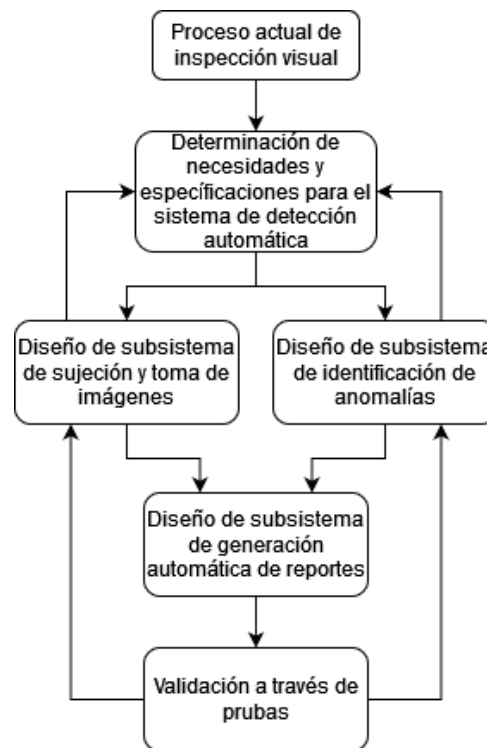


Figura 1.1: Diagrama de bloques que describe el enfoque de la solución preliminar.

1.7. Estructura del documento

El documento está compuesto por capítulos principales, la presente introducción incluida. El siguiente capítulo es el marco teórico, en el cual se describe toda la información necesaria para el entendimiento y desarrollo del proyecto. Se incluye información que no se utilizó para la implementación final, pero fue necesaria para la toma de decisiones que llevaron al prototipo.

En el tercer capítulo describe la metodología utilizada en el proyecto, además de la ejecución de las primeras fases de esta. Específicamente, el procesamiento de las peticiones del cliente en necesidades y métricas, la generación del concepto a desarrollar y una explicación de las limitaciones presentes para la validación del proyecto.

El cuarto capítulo es la propuesta de diseño, donde se describe el proceso de diseño y construcción del prototipo final. Se dividió este proceso en tres segmentos: eléctrico, mecánico y software, y se explora todo el proceso de desarrollo de cada uno de estos.

El quinto capítulo es la discusión y evaluación de los resultados obtenidos tras la finalización del proyecto. Aquí se analizan todos los aspectos de los resultados, incluyendo las imágenes obtenidas y la capacidad de identificar las anomalías, la validación del prototipo y un breve análisis financiero.

El sexto y último capítulo principal son las conclusiones, donde se analizan el proyecto y se resumen los puntos importantes logrados.

Posterior a las conclusiones, se encuentra la bibliografía y los anexos.

Capítulo 2

Marco teórico

En este capítulo se exploran conceptos fundamentales para el desarrollo del proyecto. Algunos de estos no fueron necesarios para la elaboración de la versión final, pero fueron indispensables para la toma de decisiones que llevaron al diseño que se construyó. Se describen temas como satélites, tipos y pruebas a las que se someten, visión por computadora y diversos paradigmas de esta área, y un análisis a mayor profundidad sobre diferentes técnicas de manipulación de imágenes.

2.1. Satélites

Los satélites son objetos artificiales que se ponen en órbita alrededor de un cuerpo celeste. Tienen diversos usos: retransmisión de comunicaciones, predicción meteorológica, navegación (GPS), radiodifusión, investigación científica y observación de la Tierra.[1]

Las tecnologías de los satélites evolucionan y avanzan constantemente. Algunas de las principales tendencias y tecnologías satelitales para 2023 son los satélites pequeños o *smallsats*, especialmente los *NanoSats*, que impulsan la próxima generación de capacidades satelitales. Además, el bajo coste de fabricación de los *smallsats* está allanando el camino para la producción masiva de satélites. El Internet de las Cosas (IoT) por satélite es otra tendencia importante que permite una conectividad sin precedentes en todos los sectores y potencia las capacidades 5G y las próximas 6G. Al mismo tiempo, los fabricantes y operadores de satélites están aportando innovaciones tecnológicas a las estaciones terrestres y a los servicios orbitales. Las cargas útiles digitalizadas, los sistemas de propulsión y tecnologías como la inteligencia artificial (IA) permiten a los satélites realizar funciones más complejas de forma autónoma.[2]

Las metodologías de diseño de satélites varían en función de la misión específica y la finalidad del satélite. El proceso de diseño de ingeniería se utiliza para diseñar, construir, probar y mejorar un modelo de satélite. El satélite debe cumplir determinados criterios de instrumentación y restricciones de robustez para superar con éxito las pruebas de certificación. La ingeniería de sistemas también se utiliza habitualmente en la industria

del diseño de satélites para aprovechar las complejidades subyacentes.[3]

Las pruebas son una parte fundamental del proceso de desarrollo de satélites. Los satélites deben ser capaces de sobrevivir a las vibraciones y niveles acústicos extremos del lanzamiento, así como de funcionar en condiciones muy desafiantes, como un vacío casi total, al tiempo que soportan altos niveles de electrorradiación y fluctuaciones en temperaturas que van desde las más cálidas a las más frías. Las pruebas también son vitales porque los satélites son inversiones costosas para los gobiernos o las empresas privadas.[4]

Existen varios tipos de satélites en función de su órbita: órbita terrestre baja (LEO), órbita terrestre media (MEO), órbita geoestacionaria (GEO), órbita sincrónica solar (SSO) y órbita de transferencia geoestacionaria (GTO). Los satélites también pueden clasificarse según sus funciones, como satélites de comunicaciones, meteorológicos, de navegación, de investigación científica y de observación de la Tierra.[5]

2.2. Cubesats

Un CubeSat es un tipo de satélite miniaturizado para la investigación espacial formado por múltiplos de unidades cúbicas de $10\text{ cm} \times 10\text{ cm} \times 10\text{ cm}$. Los CubeSat tienen una masa máxima de 1,33 kg por unidad y suelen utilizar componentes comerciales para su electrónica y estructura. [6]

Los CubeSats tienen diferentes tamaños, en función del número de unidades (U) que los componen. El tamaño estándar del CubeSat es una unidad.° "1U" que mide $10 \times 10 \times 10\text{ cm}$ y puede ampliarse a tamaños mayores: 1,5, 2, 3, 6 e incluso 12U. El diseño de CubeSat más pequeño que existe es de 0,25U y el más grande de 27U.[7] Los tres tamaños más usados en la academia (1U, 2U y 3U) se pueden ver representados en la figura 2.1.

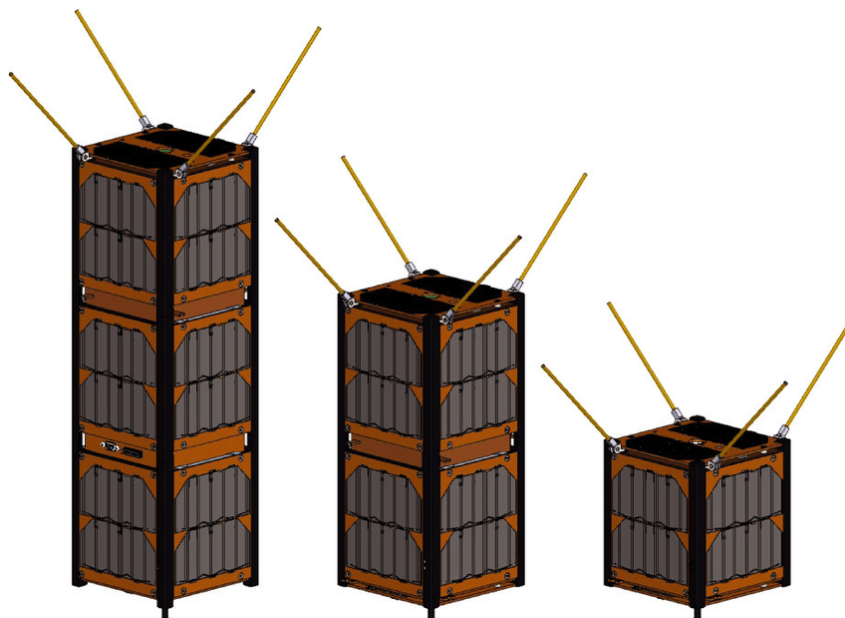


Figura 2.1: Cubesats de 1, 2 y 3 unidades [8].

Los CubeSats tienen varias ventajas sobre los satélites convencionales. Son más accesibles para empresas de todo tipo y tamaño, tienen precios más asequibles, plazos de desarrollo más cortos, tecnología más actualizada y ofrecen servicios más flexibles. Un satélite convencional puede costar entre 100 y 300 millones de euros y requiere plazos de desarrollo muy largos, entre 5 y 15 años. En cambio, el primer CubeSat de una constelación puede estar en el espacio en 8 meses, y si hablamos de réplicas, esta espera se reduce a sólo 2 meses.[9]

2.3. Pruebas ambientales para satélites

Los CubeSat, como todos los satélites, deben someterse a una serie de pruebas ambientales antes de ser certificados para el vuelo. Estas pruebas están diseñadas para garantizar que el satélite pueda soportar las duras condiciones del espacio y las tensiones del lanzamiento. Algunas de las pruebas ambientales más comunes a las que deben someterse los CubeSats son las siguientes[6][10]:

- **Pruebas de vibración:** Esta prueba simula las intensas vibraciones que experimentará el satélite durante el lanzamiento. El satélite se monta en una mesa vibratoria y se somete a una serie de frecuencias y amplitudes para garantizar que puede soportar el entorno de lanzamiento.
- **Pruebas de vacío térmico:** Esta prueba simula las temperaturas extremas y las condiciones de vacío que experimentará el satélite en el espacio. El satélite se coloca en una cámara de vacío térmico y se somete a ciclos de altas y bajas temperaturas mientras está en vacío para garantizar que puede funcionar correctamente en estas condiciones.
- **Pruebas de compatibilidad electromagnética (CEM):** Esta prueba garantiza que los sistemas electrónicos del satélite no interfieren entre sí ni con otros sistemas del vehículo de lanzamiento o en el espacio. El satélite se somete a diversos campos electromagnéticos y se miden sus emisiones para garantizar el cumplimiento de las normas pertinentes.
- **Pruebas de choque:** Esta prueba simula los choques mecánicos que sufrirá el satélite durante el lanzamiento y la separación del vehículo lanzador. El satélite se somete a una serie de choques de alta intensidad para garantizar su resistencia.

Tras las pruebas de choque y vibración, se suele realizar una inspección visual para asegurarse de que no hay grietas ni tornillos sueltos como resultado de las pruebas. Esta inspección es importante para detectar posibles problemas antes de lanzar el satélite.[6][10]

2.4. Visión por computadora

La visión por computadora es un campo de la informática que permite a los ordenadores y sistemas obtener información significativa a partir de imágenes digitales, vídeos y otras entradas visuales. Permite a los ordenadores ver, observar y comprender el mundo que les rodea. La visión por computadora consiste en entrenar a las máquinas para que reconozcan y comprendan datos visuales mediante algoritmos de aprendizaje automático, redes neuronales y grandes cantidades de datos.[11]

La visión por computadora es un campo interdisciplinario y multidisciplinario que trata de cómo se puede hacer que los ordenadores obtengan una comprensión de alto nivel a partir de imágenes o vídeos digitales. Desde el punto de vista de la ingeniería, trata de automatizar tareas que realiza el sistema visual humano. Algunas partes clave de la visión por computadora son las siguientes[12]:

- **Procesamiento de imágenes:** En esencia, la visión por computadora consiste en manipular y analizar imágenes o fotogramas de vídeo. Esto incluye operaciones como filtrado, detección de bordes, umbralización y transformaciones geométricas para mejorar o extraer características específicas de los datos visuales.
- **Extracción de características:** Los sistemas de visión por computadora a menudo requieren la extracción de información relevante o características de las imágenes. Las características pueden ser puntos, líneas, contornos o estructuras más complejas como objetos, texturas o patrones.
- **Reconocimiento y seguimiento de objetos:** Uno de los principales objetivos de la visión por computadora es identificar y localizar objetos en imágenes o vídeos. Esto puede implicar el reconocimiento de objetos predefinidos o la detección y seguimiento de objetos en tiempo real.

La visión por computadora tiene una amplia gama de aplicaciones en diversos sectores como la energía, los servicios públicos, la fabricación y la automoción. Se utiliza para tareas como la reconstrucción de escenas, la detección de objetos, la detección de eventos, el reconocimiento de actividades, el seguimiento de vídeos, el reconocimiento de objetos, la estimación de poses en 3D, el aprendizaje, la indexación, la estimación del movimiento, el modelado de escenas en 3D, la segmentación de defectos y la restauración de imágenes.[11]

2.5. Aprendizaje de máquina para visión por computadora

La inteligencia artificial (IA) desempeña un papel cada vez más importante en el campo de la visión por computadora. Las técnicas de IA, como el aprendizaje automático y

el aprendizaje profundo, se utilizan para desarrollar algoritmos que pueden aprender a reconocer patrones y tomar decisiones basadas en datos visuales. [13]

Una de las formas clave en que se utiliza la IA en la visión por computadora es a través del desarrollo de redes neuronales convolucionales (CNN). Se trata de un tipo de algoritmo de aprendizaje profundo que puede aprender a reconocer patrones en datos visuales analizando las relaciones entre los píxeles de una imagen. Las CNN han tenido mucho éxito en tareas como la clasificación de imágenes, la detección de objetos y la segmentación semántica. [14]

Otra forma en la que la IA se utiliza en visión por computadora es a través del desarrollo de redes generativas adversariales (GAN). Se trata de un tipo de algoritmo de aprendizaje profundo que puede generar nuevas imágenes aprendiendo la distribución subyacente de un conjunto de datos. Las GAN se han utilizado para tareas como la síntesis de imágenes, la transferencia de estilos y la superresolución. [14]

La IA también se utiliza en visión por computadora para tareas como la mejora de imágenes, donde los algoritmos se entrenan para mejorar la calidad de las imágenes eliminando el ruido, aumentando el contraste o corrigiendo las distorsiones. Además, las técnicas de IA se utilizan para el análisis de vídeo, donde los algoritmos se entrenan para reconocer acciones, eventos o anomalías en los datos de vídeo.

Hay dos tipos de redes que se deben destacar para el presente proyecto: los autocodificadores y las redes en forma de U. Estos son dos tipos de arquitecturas de redes neuronales utilizadas habitualmente en la IA para la visión por ordenador, en particular para tareas como la segmentación de defectos.

Un autocodificador es un tipo de red neuronal que se entrena para reconstruir sus datos de entrada. Consta de dos partes: un codificador, que comprime los datos de entrada en una representación de menor dimensión, y un decodificador, que reconstruye los datos originales a partir de la representación comprimida. Los autocodificadores pueden utilizarse para tareas como la compresión de datos, la eliminación de ruido y la detección de anomalías. [14]

Una red en forma de U, también conocida como U-Net, es un tipo de red neuronal convolucional que suele utilizarse para tareas de segmentación de imágenes. Tiene una arquitectura simétrica, con una ruta de codificación que captura el contexto y una ruta de decodificación que permite una localización precisa. La red está diseñada para trabajar con muy pocas imágenes de entrenamiento y producir segmentaciones más precisas.[14]

En el contexto de la segmentación de defectos, los autocodificadores y las redes en forma de U pueden utilizarse para identificar y segmentar defectos en imágenes. Por ejemplo, un autocodificador puede entrenarse para reconstruir imágenes de piezas no defectuosas y, a continuación, utilizarse para identificar defectos buscando diferencias entre la imagen reconstruida y la imagen original. Una red en forma de U puede entrenarse para segmentar defectos en imágenes aprendiendo a distinguir entre regiones defectuosas y no defectuosas.[15]

2.6. Técnicas tradicionales de visión por computadora

Un resumen de algunas técnicas tradicionales de visión por computadora utilizadas habitualmente en el tratamiento y análisis de imágenes:

2.6.1. Preprocesamiento de imágenes

- **Conversión a escala de grises:** Convierte imágenes en color a escala de grises para simplificar y reducir la complejidad computacional.
- **Mejora de imágenes:** Aplicar técnicas como la ecualización de histogramas para mejorar el contraste y la visibilidad de los detalles de la imagen.
- **Reducción del ruido:** Utilizar filtros (por ejemplo, Gaussiano, mediana) para reducir el ruido y suavizar la imagen.

2.6.2. Umbralización

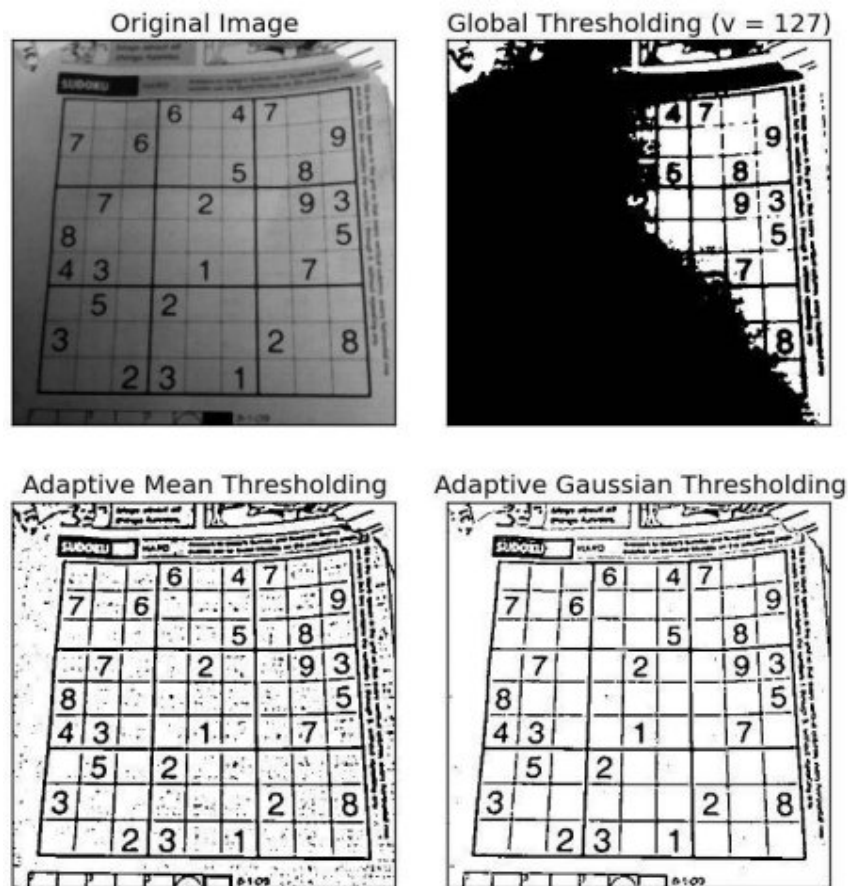


Figura 2.2: Comparación de diferentes métodos de umbralización [16].

- **Umbralización binaria:** Convierte una imagen en escala de grises en una imagen binaria ajustando los píxeles por encima de un determinado umbral a un valor y los píxeles por debajo a otro.
- **Umbralización adaptativa:** Determinar automáticamente el valor del umbral en función de las características locales de la imagen.

2.6.3. Operaciones morfológicas

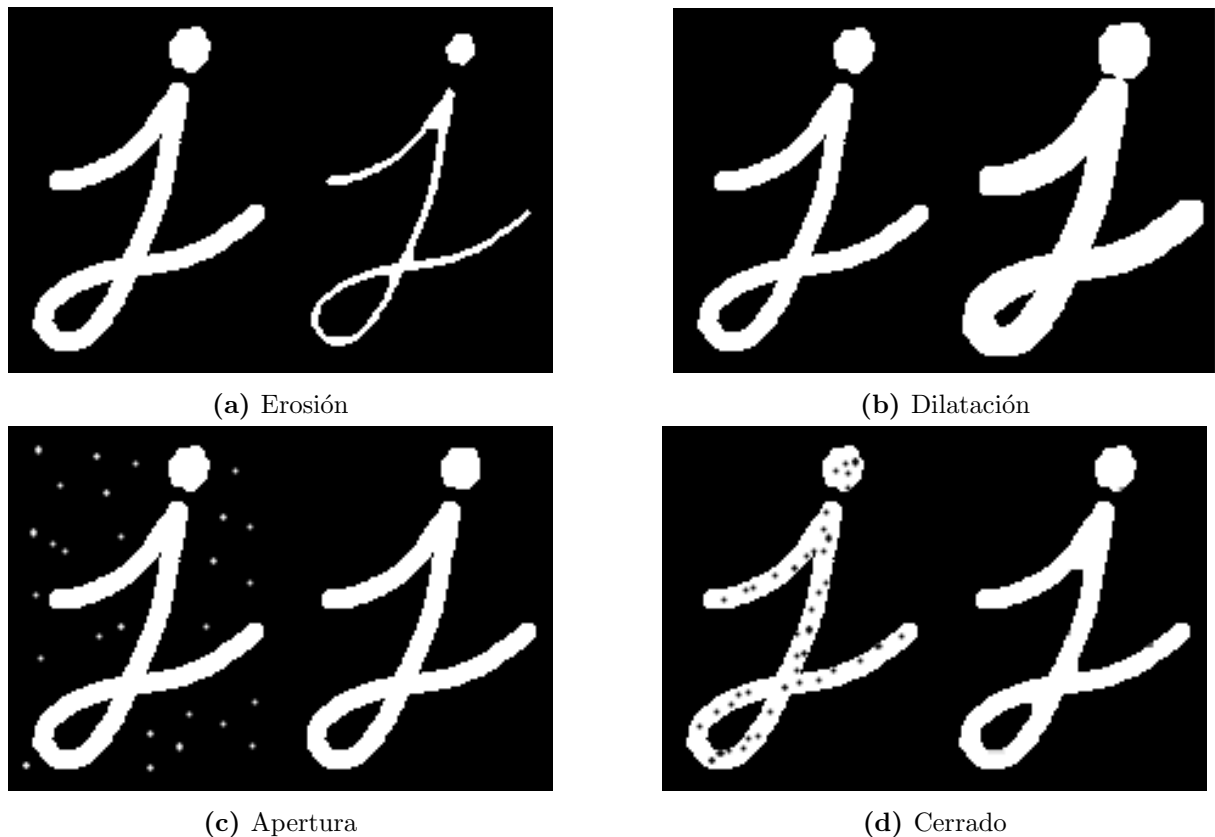


Figura 2.3: Operaciones morfológicas comunes [17]

- **Erosión:** Contrae los límites de las regiones blancas en una imagen binaria.
- **Dilatación:** Expande los límites de las regiones blancas en una imagen binaria.
- **Apertura:** Erosión seguida de dilatación, útil para eliminar ruido y separar objetos.
- **Cerrar:** Dilatación seguida de erosión, útil para cerrar pequeños huecos entre objetos.

2.6.4. Filtrado

- **Filtros de suavizado:** Aplica filtros como el Gaussiano o el de promediado para desenfocar la imagen y reducir el ruido.
- **Filtros de nitidez:** Mejora los bordes y los detalles finos utilizando filtros como el Laplaciano o la máscara de desenfoque.

2.6.5. Detección de bordes



Figura 2.4: Canny edge y Sobel [18].

- **Operador Sobel:** Detecta los bordes convolviendo la imagen con kernels Sobel en dirección horizontal y vertical.
- **Detector de bordes Canny:** Algoritmo de detección de bordes en varias etapas que incluye suavizado, cálculo de gradiente, supresión no máxima y trazado de bordes por histéresis.

2.6.6. Detección de contornos

- **Contornos:** Identificar y extraer los contornos de los objetos de una imagen mediante técnicas como el trazado de contornos.
- **Casco convexo:** Encontrar el casco convexo de los objetos, que puede ser útil para el análisis de formas.

2.6.7. Extracción de características

- **Histogramas:** Calcular histogramas de intensidades de píxeles para el análisis de imágenes.
- **Análisis de texturas:** Evaluar la textura de la imagen utilizando técnicas como las matrices de co-ocurrencia.

- **Detección de esquinas:** Identificar puntos clave o esquinas en una imagen utilizando algoritmos como el detector de esquinas de Harris.

2.6.8. Comparación de plantillas

- **Matching:** Localizar instancias de una plantilla (imagen pequeña) dentro de una imagen más grande.
- **Correlación cruzada:** Medir la similitud entre la plantilla y las regiones de la imagen.

2.6.9. Transformaciones geométricas

- **Rotación:** Rota una imagen un ángulo especificado.
- **Escalado:** Cambia el tamaño de una imagen manteniendo su relación de aspecto.
- **Transformaciones afines:** Realiza transformaciones más complejas como el cizallamiento o la inclinación.
- **Alinear imágenes:** Alinear dos o más imágenes para superponerlas con precisión.

Estas técnicas tradicionales de visión por computadora sirven de base para diversas tareas de análisis de imágenes. Son herramientas esenciales para tareas como el reconocimiento de objetos, la segmentación de imágenes y la extracción de características, y a menudo pueden combinarse para resolver problemas de visión por computadora más complejos.

2.6.10. Sustracción de imágenes

La sustracción de imágenes y el índice de similitud estructural (SSIM) son dos técnicas que pueden utilizarse en visión por computadora para comparar imágenes.

La sustracción de imágenes es una técnica sencilla que consiste en restar los valores de los píxeles de una imagen de otra. Puede utilizarse para resaltar las diferencias entre dos imágenes, como cambios en la iluminación o la presencia de nuevos objetos. La imagen de diferencia resultante tendrá valores de píxel altos en las regiones en las que las dos imágenes difieren significativamente, y valores de píxel bajos en las regiones en las que son similares. Esta se puede ver en la figura 2.5b.

El SSIM, por su parte, es una técnica más sofisticada para medir la similitud entre dos imágenes. No sólo tiene en cuenta los valores de los píxeles, sino también la información estructural de las imágenes. El SSIM compara patrones locales de intensidades de píxeles en las dos imágenes, teniendo en cuenta la luminancia, el contraste y la estructura. El índice SSIM resultante es un valor comprendido entre -1 y 1, donde 1 indica que las dos

imágenes son idénticas y -1 que son completamente diferentes. El resultado de esta se puede ver en la figura 2.5a [19].

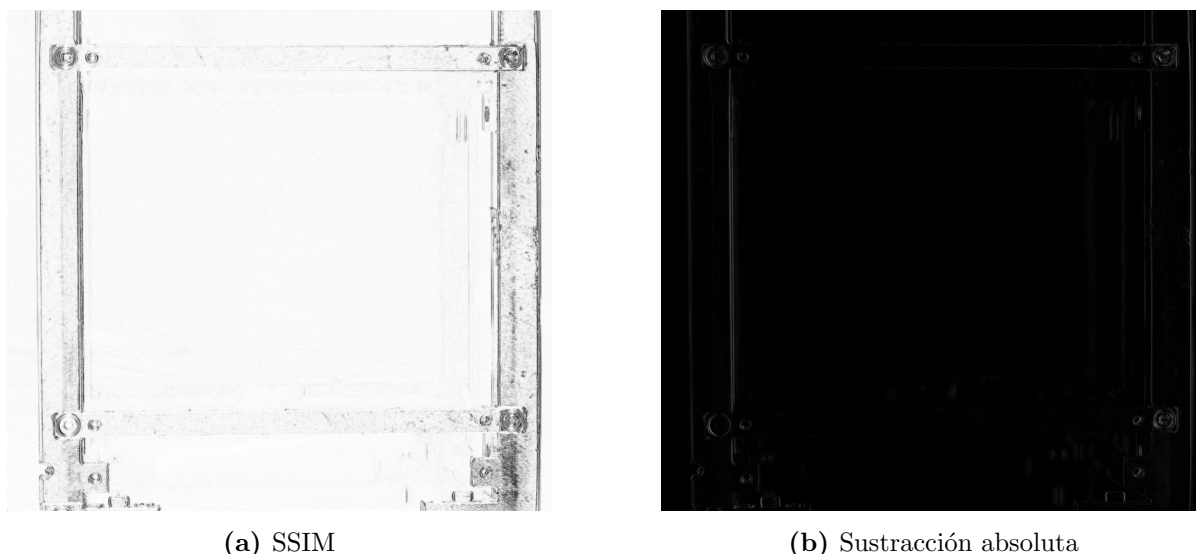


Figura 2.5: Dos métodos de sustracción de imágenes.

La principal diferencia entre la sustracción de imágenes y el SSIM es que la sustracción de imágenes produce una imagen de diferencia que resalta las diferencias entre dos imágenes, mientras que el SSIM produce una única puntuación de similitud que mide lo parecidas que son las dos imágenes. La sustracción de imágenes es una técnica sencilla y rápida, mientras que el SSIM es más compleja y requiere más cálculos.

2.7. Obtención de imágenes

En visión por computadora, la elección de la cámara depende de los requisitos específicos de la aplicación. Para capturar imágenes de alta calidad, se prefieren las cámaras DSLR con sensores grandes y objetivos de alta resolución. Sin embargo, para soluciones integradas o rentables, pueden ser adecuadas cámaras más pequeñas como el módulo de cámara Raspberry Pi[20]. Comprender estos aspectos de la cámara puede ayudarle a seleccionar la cámara adecuada para su proyecto de visión por computadora y optimizar la calidad de la imagen. A continuación se ofrece un resumen de alto nivel de aspectos importantes relacionados con las cámaras digitales:

2.7.1. Tamaño del sensor

El tamaño del sensor se refiere a las dimensiones físicas del sensor de imagen dentro de la cámara. Los sensores más grandes suelen capturar más luz y proporcionar una mejor calidad de imagen, especialmente en condiciones de poca luz. Las cámaras con sensores más grandes tienden a producir imágenes con menos ruido y mejor rango dinámico[21].

2.7.2. DSLR vs. Cámaras Compactas/Empotradas

Si bien el principio fundamental de las cámaras son las mismas, se pueden clasificar de acuerdo a características físicas. Se describen dos grandes categorías a continuación [20].

Cámaras DSLR (Digital Single-Lens Reflex): Estas cámaras ofrecen objetivos intercambiables, sensores más grandes y un control manual avanzado, lo que las hace versátiles y adecuadas para la fotografía de alta calidad.

Cámaras compactas: Las cámaras compactas, incluidos módulos como la Raspberry Pi Camera, son más pequeñas y portátiles, pero suelen tener sensores más pequeños y objetivos fijos. Son adecuadas para diversas aplicaciones, incluidos los sistemas embebidos.

2.7.3. Tipos de objetivos

Existen múltiples tipos de objetivos o lentes para utilizar en cámaras. Algunos de estos tipos son[22]:

Objetivos de focal fija: Objetivos de distancia focal fija sin capacidad de zoom. Suelen ofrecer una excelente calidad de imagen y baja distorsión.

Objetivos con zoom: Objetivos de distancia focal variable que permiten acercar y alejar el zoom. Son versátiles, pero pueden sacrificar algo de calidad de imagen por comodidad.

Objetivos gran angular: Capturan un campo de visión más amplio, por lo que son ideales para paisajes y arquitectura.

Teleobjetivos: Ofrecen un campo de visión más estrecho y se utilizan para captar sujetos lejanos, como la fauna salvaje o eventos deportivos.

2.7.4. Resolución

La resolución se refiere al número de píxeles de una imagen, que suele expresarse como ancho x alto (por ejemplo, 1920x1080 para Full HD). Las imágenes de mayor resolución contienen más detalles, pero también requieren más capacidad de almacenamiento y procesamiento. La resolución es esencial para tareas como el análisis de imágenes y la visión por computadora [23].

2.7.5. Campo de visión (FOV)

El campo de visión es la extensión de la escena que una cámara puede captar con una distancia focal determinada. Determina qué parte del entorno es visible en la imagen. Los objetivos gran angular tienen un FOV más amplio, mientras que los teleobjetivos tienen un FOV más estrecho[23].

2.7.6. Profundidad de campo (DOF)

La DOF se refiere al rango de distancias en una imagen donde los objetos aparecen enfocados. Una DOF poco profunda desenfoca el fondo, resaltando el sujeto. Una DOF profunda mantiene enfocada una mayor parte de la escena. El tamaño de diafragma y el tamaño del sensor influyen en la DOF[23].

2.7.7. Ruido de la imagen

El ruido de imagen aparece como variaciones aleatorias en los valores de los píxeles y puede degradar la calidad de la imagen. Factores como el tamaño del sensor, el ajuste ISO y el tiempo de exposición afectan al ruido. Los sensores más pequeños y los ajustes ISO más altos tienden a producir imágenes más ruidosas. Las cámaras suelen incluir algoritmos de reducción de ruido para mitigarlo, especialmente en condiciones de poca luz.[24]

2.7.8. Sensibilidad ISO

ISO mide la sensibilidad del sensor de la cámara a la luz. Los ajustes ISO más altos se pueden utilizar en condiciones de poca luz, pero pueden introducir ruido. Aumentar el ISO permite velocidades de obturación más rápidas con poca luz, reduciendo el desenfoque de movimiento, pero puede reducir la calidad de la imagen debido al ruido.[24]

Capítulo 3

Metodología

3.1. Pasos de la metodología

La metodología empleada en el desarrollo de este proyecto es aquella propuesta por Ulrich en su obra “Diseño y desarrollo de productos” [25]. Dicha metodología descompone el proceso en diversas etapas, las cuales se llevan a cabo de manera secuencial, tal como se ilustra en la Figura 3.1. Es importante señalar que este procedimiento es repetitivo, lo que significa que si se detecta alguna insuficiencia en cualquiera de las fases, se retrocederá a dicha etapa en cualquier momento para incorporar las mejoras pertinentes. Este capítulo abarcará el proceso desde la fase inicial, que implica la identificación de las necesidades del cliente, hasta alcanzar la etapa en la que se selecciona el concepto del producto.

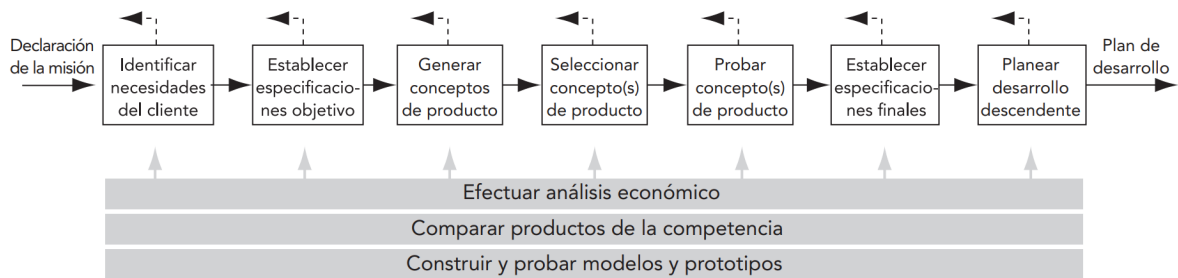


Figura 3.1: Fases de la metodología de Ulrich y Eppinger [25].

Las etapas por abordar en este capítulo serán las siguientes::

- **Identificar las necesidades del cliente:** Constituye la fase inicial de la metodología. Implica identificar la totalidad de los requerimientos que deben ser satisfechos por la solución [25]. Este reconocimiento se logra mediante diversas interacciones con el cliente, en las cuales cada deseo es identificado y transformado en una necesidad con el propósito de evitar cualquier sesgo. Además, se otorga a cada necesidad un nivel de importancia establecido por el cliente, el cual desempeña un papel en etapas posteriores como factor determinante para la toma de decisiones.

- **Establecer especificaciones objetivo:** Los valores objetivo y métricas son traducciones técnicas de las necesidades previamente delimitadas y reflejan las expectativas del equipo de desarrollo [25]. Cada métrica está vinculada al menos a una necesidad y permite evaluar el grado en que la solución satisface dicha necesidad.
- **Generar conceptos del producto:** Constituye una fase crítica, ya que es aquí donde se proponen las diversas soluciones posibles. Se inicia con una descomposición funcional, en la cual se identifican subfunciones potenciales de la solución. Posteriormente, se plantean soluciones o conceptos para cada subfunción. La generación de ideas se efectúa tanto internamente como mediante fuentes externas, asegurando así la consideración de una amplia gama de opciones.
- **Seleccionar concepto del producto:** Representa la fase culminante de este capítulo. Se combinan los conceptos generados previamente, y cada alternativa se evalúa utilizando una matriz de selección. El proceso involucra una etapa inicial de filtrado para retener las combinaciones más prometedoras, seguida por una fase de evaluación que determina los conceptos ganadores. Para cada matriz, se elige un punto de referencia que sirve como base comparativa, y se establecen criterios basados en las necesidades previamente identificadas.

3.2. Identificación de necesidades

Para la identificación de las necesidades primero se conversó con el cliente, para ver qué quiere exactamente. Para esto se utilizaron diversos métodos como correos electrónicos, documentos internos y diversas conversaciones. Además, se tuvieron conversaciones con el profesor Masui, encargado del laboratorio de pruebas ambientales, y con Tasuku Matsui, estudiante que trabaja como asistente en este laboratorio. Todas las necesidades fueron planteadas a partir de las limitaciones adquiridas de estas discusiones. De cada uno de estos intercambios se extrajeron todas las frases o comentarios que se pudieran identificar como un y se procesaron apegándose lo mejor posible al siguiente conjunto de reglas [26]:

- Se expresa la necesidad en términos de lo que el producto tiene que hacer, no como tiene que hacerlo.
- Se expresa la información tan específica como la información recompilada, no realizar afirmaciones de detalles no presentes en la declaración original.
- Utilizar enunciados de forma positiva.
- Se expresa la necesidad como un atributo del producto.
- Se evitan las palabras debe y debería.

Por último, después de haber identificado y analizado todas las necesidades, es esencial asignar a cada una de ellas un grado de importancia relativa. Este nivel de importancia

desempeña un papel crucial en la determinación de la prioridad de cada necesidad en relación con las demás. Además, cumple una función adicional al actuar como un mecanismo de verificación para asegurar la comprensión precisa por parte del cliente. Esta información reviste una importancia fundamental para orientar las decisiones en las fases posteriores del proceso. El sistema de importancia elegido consiste en asignar una calificación entre uno y cinco a cada necesidad, donde cada valor tiene su propio significado [26], como se muestra en Tabla 3.1. Se envió el formulario al cliente donde se le entregó una lista con los significados de cada nivel de importancia, un cuadro con todas las necesidades y un espacio para llenar la importancia asociada.

Tabla 3.1: Nivel de importancia con su respectivo significado.

Importancia	Significado
1	La función es indeseable. No consideraría un producto con esta función.
2	La función no es importante, pero no me importaría tenerla.
3	Sería bueno tener esta función, pero no es necesario.
4	La función es altamente deseable, pero consideraría un producto sin ella.
5	La función es de importancia crítica. No consideraría un producto sin esta función.

Todas las necesidades se observan en la Tabla 3.2. Es importante mencionar que debido a que el cliente habla inglés, estas necesidades fueron originalmente planteadas en inglés, y fueron traducidas para la elaboración de este documento.

Tabla 3.2: Lista de necesidades y sus calificaciones.

Num	Necesidad	Importancia
1	Ser capaz de inspeccionar todos los paneles de una vez	3
2	Detectar errores de marca de torque	5
3	Detectar rasguños en el vidrio	5
4	Detectar grietas en el vidrio	5
5	Soportar tamaños diferentes a 2U	3
6	Funcionar con tolerancia a pequeños cambios de luz y posición	3
7	Automáticamente crear reportes de prueba si es exitoso	4
8	Automáticamente crear reporte de fallos si no es exitoso	4
9	Ser fácil de instalar en diferentes computadoras	3
10	Ser simple de utilizar por nuevos usuarios	3
11	Permitir que entradas de usuario definan si una detección es una falla	4
12	Integrarse en los sistemas de pruebas actuales en Labview	3
13	Tener una interfaz de usuario gráfica amigable	3

Las necesidades más críticas (2, 3 y 4) tienen este nivel de importancia debido a que estas anomalías son exactamente las que se deben encontrar en una inspección visual. Cualquier rasguño, grieta o error de marca de torque implica algún problema geométrico o de ensamble del satélite que se debe corregir antes de enviarlo al espacio.

Además de mostrar información a los operadores, los reportes (necesidades 7 y 8) son fundamentales, ya que estos son requisitos explícitos por agencias espaciales. La documentación de todas las pruebas del satélite es necesaria para poder lanzarlo al espacio.

La capacidad de poder editar los resultados del sistema automático fue una solicitud de parte del profesor encargado, con el fin de poder fácilmente editar cualquier falso positivo, o bien, descartar un error de baja importancia.

3.3. Generación de métricas

Las métricas se crean a partir de las necesidades previamente reconocidas. Deben ser cuantificables, ya que se emplean para medir el nivel de satisfacción de la solución y deben evitar términos subjetivos. Es esencial que cada necesidad esté respaldada por al menos una métrica, ya que, de lo contrario, no se podrá garantizar que se cumplan los deseos del cliente.

Para cada métrica, se establece un valor objetivo, que representa el valor deseado que se aspira alcanzar, y un valor mínimo, que delimita lo que se considera aceptable en la solución final y que es menos restrictivo. La flexibilidad de estos intervalos se determina en función de la importancia de cada necesidad; las necesidades más críticas requieren valores más ajustados, mientras que las menos cruciales pueden ser más flexibles en cuanto a los criterios de aceptación. Dos tipos principales de criterios fueron empleados: magnitudes, que son valores numéricos que cuantifican propiedades físicas y binarios, que se basan en la conformidad o no con lo establecido en la métrica [25].

Las métricas generadas se observan en Tabla 3.3. Además, con una matriz necesidades - métricas, se comprobó que todas las necesidades estuvieran representada por al menos una métrica. Esta matriz se observa en la Tabla 3.4.

La elección de los valores marginales e ideales se efectuó considerando tanto los deseos del cliente como fuentes externas. Cuando las métricas se relacionaban con funciones específicas expresamente solicitadas por el cliente, como el cumplimiento de métricas binarias o listas de funciones, se utilizaron los deseos del cliente como base. En los demás casos, se emplearon referencias externas en conjunto con los deseos del cliente para establecer los rangos marginales e ideales para cada métrica.

Varias necesidades son de naturaleza subjetiva. Por ejemplo, la necesidad 10 y 13. Definir “simple de utilizar” o “amigable” puede ser problemático. Se pueden seguir guías de diseño de interfaces de usuario o minimizar el número de pasos para realizar la prueba durante el proceso de diseño, ¿pero cómo se puede cuantificar esto durante la fase de planteamiento

de métricas? Si un sistema debe armarse y desarmarse fácilmente, se podría cuantificar el tiempo que se tarda, el número de herramientas, etc. En este caso, se aplicará una pequeña encuesta a los usuarios, quienes calificarán el sistema de manera subjetiva. Los valores que deben alcanzarse en esta encuesta son definidos por la importancia asignada a su respectiva necesidad.

La primera necesidad habla sobre la capacidad de la solución implementada de analizar todas las caras del satélite al mismo tiempo, o si se analizan solo algunas caras, se modifican factores como la posición o la orientación, y se ejecuta nuevamente la prueba con las caras faltantes. Se seleccionó que idealmente se analizan las 6 caras al mismo tiempo, ya que ahorraría tiempo. Sin embargo, si es necesario analizar parcialmente el satélite, rotarlo, y volver a ejecutar la prueba, es aceptable. Por eso, al menos se deben analizar 4 caras a la vez.

Los CubeSats, como describió anteriormente, pueden tener distintos tamaños. El cliente expresó interés en construir este sistema para ser utilizado en el satélite Dragonfly. Este satélite es de 2 unidades (10cm x 10cm x 22.7cm). Sin embargo, estaría bien si el sistema soporta otros tamaños, para minimizar el trabajo futuro necesario para adaptar el sistema para ser utilizado con otros satélites. Los más comunes en el laboratorio LaSEINE son de 1, 2 y 3 unidades.

La facilidad de instalación del sistema se refiere a que el segmento de software debe ser fácilmente instalado en la plataforma que se vaya a usar. Cualquier software que sea parte del proyecto va a ser desarrollado en Linux, que fue algo aceptado por el cliente durante una conversación, sin embargo, expresó interés en que fuera compatible con los otros sistemas operativos que se utilizan en el laboratorio.

Tabla 3.3: Lista de métricas.

Num	Num. Necesidad	Métrica	Importancia	Unidades	Valor Marginal	Valor ideal
1	1	Número de paneles que se pueden inspeccionar a la vez	3	# de paneles	4	6
2	2	Puede detectar errores de marca de torque	5	Binario	Sí	Sí
3	3	Puede detectar rasguños en el vidrio	5	Binario	Sí	Sí
4	4	Puede detectar grietas en el vidrio	5	Binario	Sí	Sí
5	5	Tamaños que soporta	3	Lista	2U	1U 2U 3U
6	6	Tiene tolerancia a cambios de luz y posición	3	Binario	No	Sí
7	7	Crea reportes de prueba cuando esta es exitosa	4	Binario	Sí	Sí
8	8	Crea reportes de fallos cuando estos existen	4	Binario	Sí	Sí
9	9	Sistemas operativos con los que es compatible	3	Lista	Linux	Windows MacOS Linux
10	10	Facilidad subjetiva del operario de utilizar el sistema	3	Escala subjetiva	7	9
11	11	Usuario puede sobrescribir decisión del algoritmo	4	Binario	Sí	Sí
12	12	Sistema integrado con el resto del sistema de pruebas del laboratorio	3	Binario	Sí	Sí
13	13	Facilidad subjetiva del operario de utilizar el sistema	3	Escala subjetiva	7	9

3.4. Generación de conceptos

Antes de emprender la generación de conceptos, es imperativo discernir todas las funciones con el propósito de proponer conceptos específicos para cada una. Para este propósito, se lleva a cabo una subdivisión funcional en bloques, en la que las funciones se desglosan en unidades más pequeñas en una progresión gradual hasta alcanzar un nivel que permita la formulación de conceptos destinados a resolver cada una de ellas. El proceso se inicia mediante la creación de un diagrama de primer nivel en el que se identifican exclusivamente las entradas y salidas del sistema. Este es presentado en la Figura 3.2 como un punto de partida. Las entradas del sistema en este caso son: la energía que alimentará el sistema, el satélite tanto antes como después de la prueba de vibración, señal de inicio y aprobación por parte del operador. Por otro lado, las salidas son los reportes de la prueba y las fotografías para ser archivadas para futura referencia.

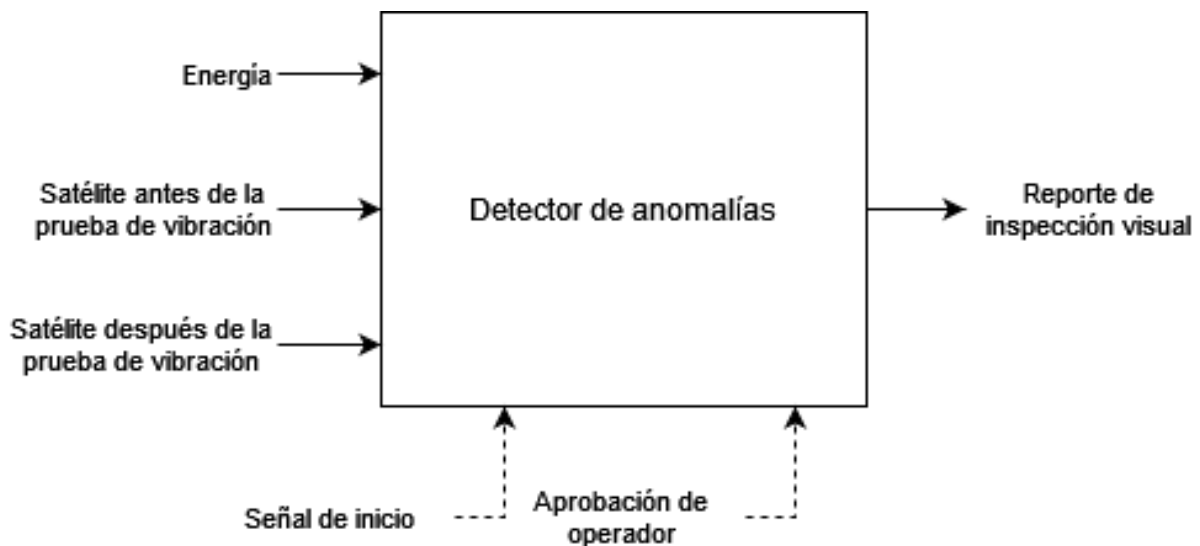


Figura 3.2: División funcional de primer nivel.

Para el segundo nivel se divide este primer bloque en bloques pequeños donde cada uno representa una función específica. Las funciones que se tienen son las siguientes:

- Admitir energía eléctrica que es la etapa que se encarga de admitir la energía externa, adaptarla en caso de ser necesario para que sea compatible con los otros subsistemas y distribuirla a las partes que la requieran
- El sistema de fotografiado, que es el sistema donde se montará el satélite y se tomarán las imágenes necesarias para la comparación y para el archivado
- El identificador de anomalías, que es el algoritmo que detecta las diferencias en el satélite antes y después de someterse a la prueba de vibración
- El generador de reportes que tiene como salida el reporte como tal

Este diagrama se puede observar en la Fig. 3.3.

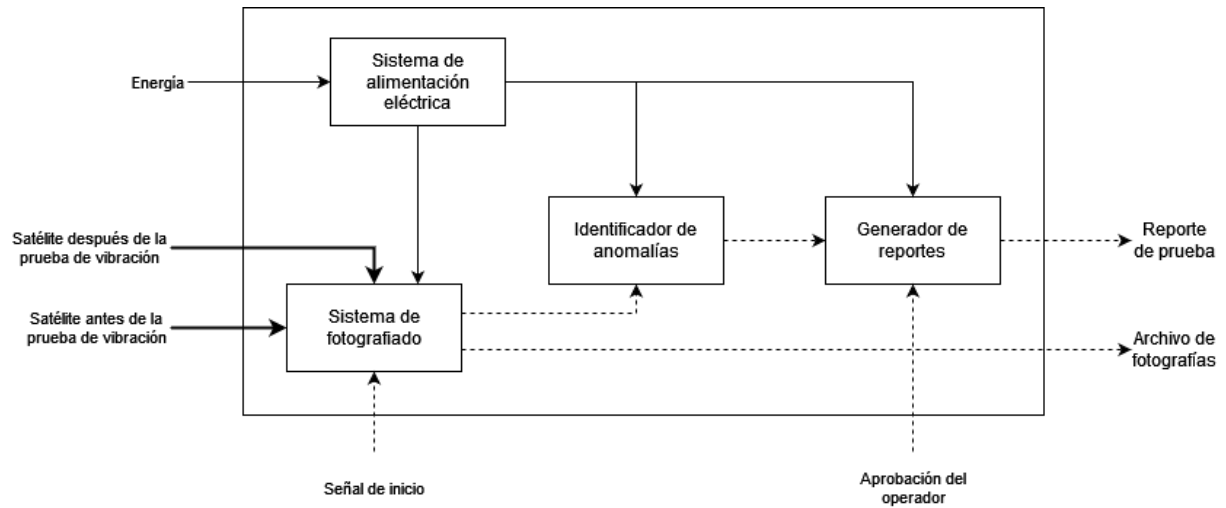


Figura 3.3: División funcional de segundo nivel.

Una vez que se han identificado y desglosado todas las funciones, el siguiente paso implica la creación de una lista de conceptos que aborden cada una de estas funciones. Para cada función, se llevó a cabo inicialmente una búsqueda interna mediante una lluvia de ideas para generar conceptos iniciales. Posteriormente, se complementó esta búsqueda con una exploración externa con el fin de identificar soluciones adicionales que pudieran no haberse considerado en la primera fase. La lista resultante para cada función se encuentra detallada en la Tabla 3.5. A continuación, se presentan algunos de los conceptos descubiertos para cada función, dejando el análisis detallado de sus características para la fase de selección.

El sistema de fotografiado es la función más compleja del sistema. Este se podría decir que está compuesto por tres subsistemas: montaje del satélite, fotografiado, y la iluminación. Para el montaje de satélite se tiene que considerar que este debe permitir ver lo más posible cada cara del satélite, y que este debe estar en exactamente la misma posición cada vez, para minimizar falsos positivos durante la detección. Por esto, se consideraron pequeños soportes impresos en 3D donde el satélite encaja; soportes hechos en aluminio o similar en los cuales el satélite es atornillado; colgar el satélite de cuerdas y un brazo robótico que rota el satélite como sea mejor para las fotografías. Para el sistema de fotografiado como tal, se consideraron formas de obtener imágenes del satélite que provean información de su geometría y apariencia para detectar cambios. En este caso, se consideraron escaneo 3D usando LiDAR, escaneo 3D usando fotogrametría, ultrasonido, cámaras ultraspectrales, cámaras simples embebidas y cámaras de fotografía de alta calidad como DSLRs. Finalmente, para la iluminación del satélite, se consideraron únicamente luces LED en diferentes formatos: tiras LED, barras LED, flash de cámaras y un cajón iluminado, como el que es usado en fotografía de productos para venta.

Para el algoritmo de identificación de anomalías se requirió mucho trabajo de investigación para si quiera conocer diferentes paradigmas y posibilidades para usar en este sistema. Se

tienen dos ramas principales para trabajar: sistemas de inteligencia artificial, y sistemas de técnicas tradicionales de visión por computadora. Ambas ramas cumplen la misma función. Para sistemas de inteligencia artificial, se tienen redes tipo \ddot{u} z auto-encoders, mientras que para técnicas tradicionales se tienen las opciones de substracción de imágenes y índice de similaridad estructural (SSIM).

El generador de reportes es un programa que tomará los resultados de las pruebas y las fotografías tomadas, llenará una plantilla y creará un reporte de la prueba con evidencias. Se tienen dos propuestas actuales para la generación de reportes: utilizar macros en Visual Basic for Applications, que es código que corre dentro de los programas Office de Microsoft, y se pueden usar para generar documentos .docx con la información deseada, o directamente crear un pdf desde el mismo código que detecta las anomalías.

Finalmente, el sistema de alimentación eléctrica. Ninguna parte del sistema es mecánico, por lo que toda la energía debe mantenerse eléctrica. Sin embargo, esta debe adecuarse a lo sistemas que la necesiten. Se tienen diversas opciones, sin embargo, la opción ideal dependerá de cuáles otras opciones sean escogidas en los otros sistemas. Por ejemplo, si se utilizarán cámaras DSLR con sus propios flash, no se necesita alimentación eléctrica para las luces. Pero en general, se tienen opciones de adaptadores AC-DC convencionales, alimentación directa desde el computador que correrá el algoritmo de detección o una combinación de ambas.

Estas opciones se pueden ver representadas en la tabla 3.5

Tabla 3.5: Lista de conceptos para cada función.

Fotografiado					
Montaje del satélite	Obtención de imágenes	Iluminación	Identificación de Anomalía	Generador de Reporte	Alimentación Eléctrica
Soportes impresos en 3D	LiDAR	LED Strips	SSIM	VBA Macros	Adaptadores AC-DC
Soportes de aluminio atornillado	Ultrasonido	Flash DSLR	Subtraction	Python PDFs	Alimentación directa
Colgar de cuerdas	Cámaras infrarrojas	Led Lamps	U networks		
Robotic gripper	Cámaras embebidas	Lightbox	Auto-encoders		
	Cámaras DSLR				
	Fotogrametría				

3.5. Selección del concepto

Para la etapa de selección se decidió separar la comparación de los conceptos en dos grupos, el primero se refiere a todo el sistema menos el subsistema de alimentación eléctrica, y el segundo se refiere al subsistema de alimentación eléctrica. Esto ya que, como se mencionó anteriormente, el subsistema de alimentación eléctrica depende completamente de cuál solución final se va a utilizar.

En esta etapa se utilizaron dos tipos de mecanismos, el primero es una etapa de filtrado donde se eliminan los conceptos menos interesantes y el segundo es una matriz donde se asignan un peso a cada criterio y se pondera cada combinación para determinar el

concepto ganador. En cada etapa se debe seleccionar un concepto de referencia el cual se utiliza como base para comparar el rendimiento de la combinación en cada criterio seleccionado. En la etapa de filtrado solo se asigna si el criterio empeoró o mejoró para concepto mientras que en la matriz final se pesos y se asigna valores entre uno y cinco a cada criterio. El lugar se obtiene diferente en ambas matrices, el filtrado utiliza la suma de + y - mientras que la matriz de selección utiliza la suma ponderada de todos los criterios.

Los criterios se asignan con base en las necesidades, se buscan todas las necesidades que pueden ser afectadas por la selección de estos conceptos y se generan métricas que puedan comparar los conceptos. En el caso de la matriz de selección a cada criterio se le asigna un peso que se basa en la importancia de la necesidad de forma que todos sumen 100%. La clasificación se hace asignando valores entre uno y cinco que determina como se compara el concepto con la referencia en cada criterio: 1 es peor a la referencia, 2 ligeramente peor, 3 igual, 4 ligeramente mejor y 5 mejor.

Para empezar, se genera la lista de combinaciones de la Tabla 3.6 mezclando los diferentes conceptos previamente generados, para esto utilizó criterio personal ya que si se utilizaban todas las posibilidades se tendría que comparar una cantidad muy elevadas de combinaciones, se conservaron las opciones que se consideraron más interesantes y se eliminaron las que no se consideraron adecuadas. A cada concepto se le asigna una letra la cual se va a utilizar como identificación en las siguientes etapas. El concepto A fue elegido referencia después de discutirlo con expertos en el laboratorio, preliminarmente se considera el mejor.

Es importante mencionar que se descartaron todas las opciones de identificación de anomalías que utilicen inteligencia artificial ya que usando técnicas de aprendizaje incluso no supervisadas se necesitan datos de entrenamiento. En este caso, como mínimo, se ocuparían muchísimas fotos de satélites tomadas de la misma manera que se tomarán en este sistema, y no se cuenta con este conjunto de datos de entrenamiento. Además, se descartó utilizar macros VBA ya que implicaría utilizar un programa extra, lo que va en contra de ser simple de usar. Colgar el satélite de cuerdas se descartó rápidamente también debido a que es sumamente propensa a mover el satélite con estímulos mecánicos simples, sin mencionar el riesgo de la manipulación del satélite. Y el brazo robótico que sostiene al satélite también se eliminó ya que implica muchísima complejidad en el sistema. Y finalmente, se descartó utilizar un lightbox debido a que se debería modificar mucho estructuralmente para montar las cámaras y el satélite, lo que podría llevar a mala repetitibilidad en las fotografías.

La etapa de filtrado se observa en la Tabla 3.7.

Tabla 3.6: Lista de combinaciones.

Conceptos	
A	Supp. 3D + Emb. Cam. + LED Strips + Subtraction
B	Supp. Alum. + Emb. Cam + LED Strips + Subtraction
C	Supp. 3D + DSLR + LED Lamps + SSIM
D	Supp. Alum. + DSLR + LED Lamps + Subtraction
E	Supp. 3D + Emb. Cam. + LED Lamps + SSIM
F	Supp. Alum. + Emb. Cam + LED Lamps + SSIM
G	Supp. 3D + DSLR + Flash + Subtraction

Tabla 3.7: Matriz de filtrado para conceptos de detección y identificación.

Conceptos							
Criterios de selección	A (Referencia)	B	C	D	E	F	G
Repetibilidad de fotografiado	0	+	0	+	0	+	0
Propensidad a falsos positivos	0	0	-	+	-	-	0
Costo	0	-	-	-	0	-	-
Tolerancia a ruido	0	0	-	+	-	-	+
Facilidad de operación	0	-	-	-	0	-	+
Suma +	0	1	0	3	0	1	2
Suma 0	5	2	1	0	3	0	2
Suma -	0	2	4	2	2	4	1
Evaluación neta	0	-1	-3	1	-2	-3	1
Lugar	3	5	6	1	4	6	1
¿Continuar?	Sí	No	No	Sí	No	No	Sí

Para la selección final se utiliza la misma referencia, se dejan los mismos criterios, pero esta vez se le asigna peso a cada uno basándose en la importancia de las necesidades que lo generaron. Los criterios con mayor peso asociado son los relacionados con la repetibilidad y rechazo de ruido ya que se originaron de necesidades de importancia crítica, el resto se divide entre los criterios de menor importancia. Las calificaciones y los puntajes finales se muestran en la Tabla 3.8.

Tabla 3.8: Matriz de selección para detección de presencia y identificación.

Criterios de selección	Peso	Conceptos					
		A		D		G	
		Calificacion	Ponderada	Calificacion	Ponderada	Calificacion	Ponderada
Repetibilidad de fotografiado	25 %	5	1.25	5	1.25	3	0.75
Propensidad a falsos positivos	25 %	4	1	4	1	3	0.75
Tolerancia a ruido	25 %	4	1	5	1.25	2	0.5
Costo	15 %	5	0.75	2	0.3	2	0.3
Facilidad de operación	10 %	4	0.4	2	0.1	3	0.3
Total de puntos	100.0 %		4.4		3.9		2.6
Lugar			1		2		3
¿Continuar?			Sí		No		No

El concepto ganador es el concepto A de la tabla 3.6. Está compuesto por soportes impresos en 3D, cámaras embebidas, tiras de LEDs y utiliza la técnica de sustracción.

3.6. Concepto ganador

Recapitulando lo discutido en este capítulo, a partir del procesamiento de las necesidades se determina que se necesita un dispositivo capaz de detectar e identificar anomalías en el vidrio de paneles solares y en marcas de torque a partir de fotografías y que genere un reporte de los resultados del análisis permitiendo que un usuario tenga control sobre este.

El concepto que mejor cumplió con estas características y que mejor se acopla con los intereses del laboratorio es el concepto A de la tabla 3.6

Para facilitar la visualización del concepto ganador se construye el diagrama de la Fig. 3.4, el cual muestra un boceto de una vista superior de cómo sería la distribución de estos componentes y una lista numerada con los diferentes conceptos que fueron elegidos para cada función, cabe mencionar que las posición exacta de cada componente, particularmente las luces, y las dimensiones pueden variar en la solución final.

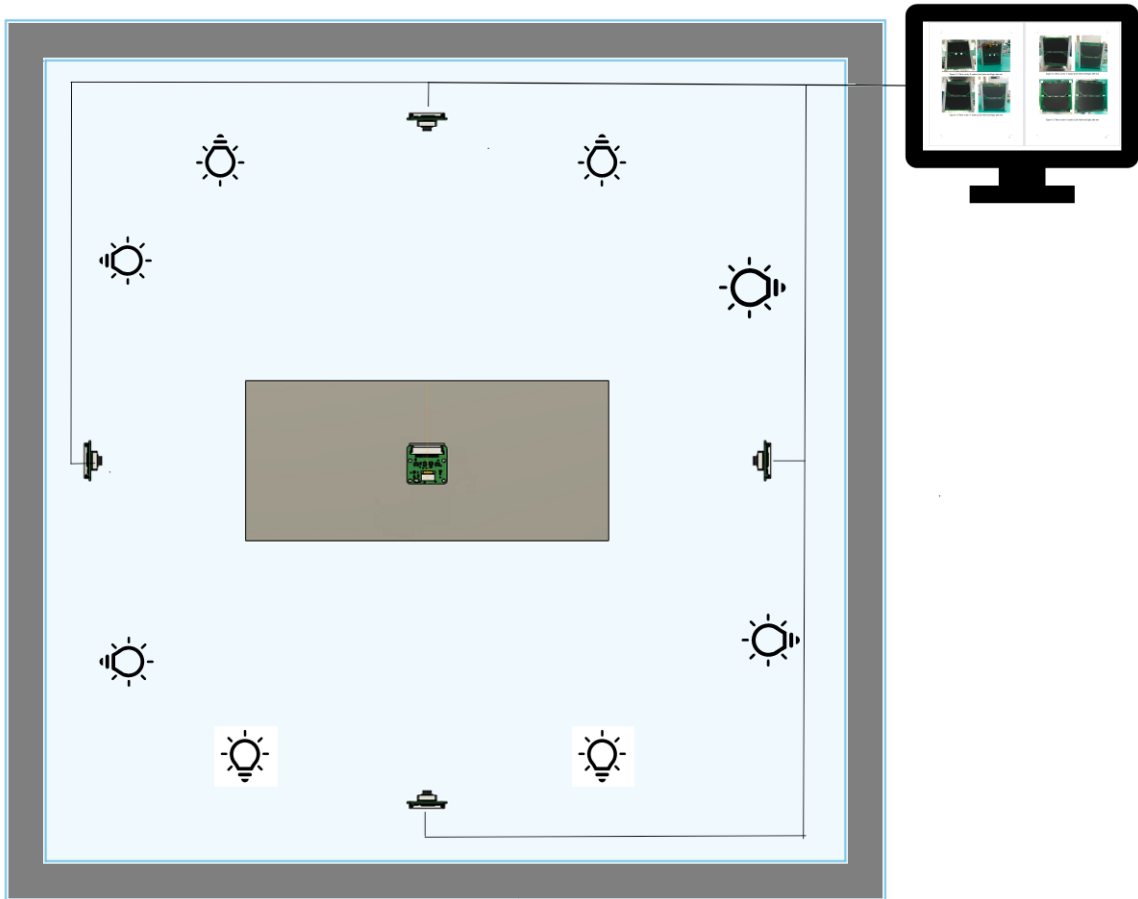


Figura 3.4: Boceto del concepto ganador.

Con el concepto ganador elegido se procede a realizar la propuesta de diseño. Por orden, se divide el diseño en dos secciones grandes: el diseño del sistema de fotografiado, que tiene que ver con todo lo relacionado a la estructura mecánica, cámaras y electrónica que procesa y alimenta las cámaras y luces, y posteriormente el segmento de software que tiene el algoritmo de detección de anomalías y generación de reportes.

3.7. Limitaciones para la validación

La última fase de la metodología, la validación, se ejecuta una vez construido el prototipo. En esta fase se prueba exhaustivamente la solución desarrollada con el fin de verificar que se cumplen todos los requerimientos del proyecto. Estas pruebas deben ser estadísticamente válidas [25].

Esta es una de las principales limitaciones del proyecto. Debido a que no es posible obtener múltiples satélites en perfecto estado y otros dañados, no se pueden tomar comparaciones masivamente. Además, los operadores cambian para cada proyecto. No existen expertos, aparte del profesor asesor industrial.

Para validar la funcionalidad del proyecto, se logró reservar el satélite Dragonfly de 2U

durante un día laboral. Durante esta jornada, se le pedirán a la mayor cantidad de operadores de proyectos pasados que realicen la inspección visual y redacción de los reportes de manera manual. Seguidamente se les pedirá que utilicen el sistema automático.

Los resultados de la validación se encuentran en la tabla 5.1 en el capítulo de resultados. En resumen, solo se pudieron ejecutar 7 pruebas en el tiempo que se contaba con el único satélite alocado para el proyecto.

Capítulo 4

Propuesta de diseño

Para el desarrollo del prototipo de la solución seleccionada en el capítulo anterior se procedió a dividir el sistema completo en tres segmentos: eléctrico, mecánico y software, aptamente mecatrónico. A continuación se describen los detalles de cada segmento.

4.1. Segmento Eléctrico

El segmento eléctrico consiste en todas las partes del prototipo que sea hardware electrónico, como lo son la selección de componentes, diseño de la arquitectura del sistema, alimentación eléctrica y el sistema de iluminación interno.

4.1.1. Identificación de los requerimientos

De las necesidades de la tabla 3.2 se extraen las directamente relevantes a este segmento, y se añaden otras que emergen durante el desarrollo del proyecto. Esto puede darse por que se tiene nueva información obtenida tras mayor investigación, o también por que se van definiendo restricciones en recursos. Los requerimientos resultantes son los siguientes:

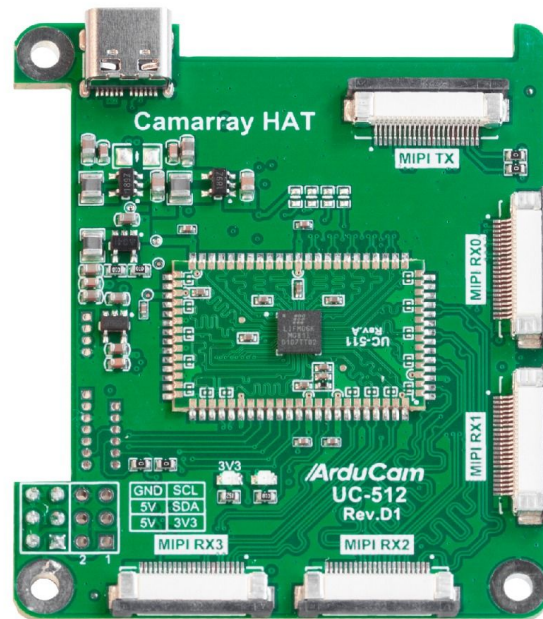
- Ser capaz de inspeccionar todos los paneles a la vez
- Detectar errores en la marca de torque
- Detectar rasguños en el vidrio
- Detectar grietas en el vidrio
- Funcionar con tolerancia a pequeños cambios de luz y posición
- Ser fácil de instalar en diferentes computadoras

La capacidad de detectar todos los paneles a la vez en el concepto ganador implica que se utilizarán seis cámaras, por lo que se requiere un computador que pueda ser conectado a seis cámaras simultáneamente. Además, se debe maximizar el nivel de detalle manteniendo el ruido al mínimo para mejorar la capacidad de pequeñas características como los rasguños. Por esto, se tendrá iluminación controlada.

4.1.2. Componentes seleccionados

Existieron dos restricciones principales para la selección de componentes eléctricos. El precio y el tiempo de adquisición, por lo que opciones muy exóticas debieron ser descartadas rápidamente, y se procedió a buscar opciones con las que hubiera algún nivel de familiaridad.

Durante el proceso metodológico se decidió utilizar cámaras embebidas. Estas son similares a los módulos de cámara de Raspberry Pi, o incluso estos mismos pueden ser. Además, deben conectarse seis simultáneamente al procesador. Tras una investigación, se identificó una plataforma que permite la conexión de hasta ocho cámaras a través de una interfaz CSI, la misma encontrada en una Raspberry Pi. Esta plataforma es el multiplexor de 4 a 1 del fabricante Arducam, que observa en la figura 4.1.



ArduCam

Figura 4.1: Multiplexer 4 a 1 de Arducam [27].

Este recibe hasta cuatro cámaras, y tiene un solo cable de salida, el cuál se conecta ya sea a una Raspberry Pi, o a un Jetson de NVidia. Esto ayuda significativamente a filtrar opciones de cámaras y computadoras. Ahora se procedió a seleccionar cuáles cámaras de las compatibles con el multiplexor son las mejores. Se identificaron tres opciones que Arducam vende directamente en un kit con el multiplexor. Estas se analizan en la tabla 4.1.

Tabla 4.1: Opciones de cámaras de formato seleccionado

Cámara	Resolución	Plataformas Compatibles	Precio
IMX219	8 Megapíxeles	Raspberry Pi Jetson	\$140
477P	12 Megapíxeles	Raspberry Pi Jetson	\$320
IMX519	16 Megapíxeles	Raspberry Pi Jetson	\$169
64MP	64 Megapíxeles	Raspberry Pi	\$299

Por supuesto existen otros parámetros aparte de los presentes en la tabla que son relevantes en el mundo de las cámaras. Sin embargo, como se mencionó en la sección 4.1.1, se busca maximizar la resolución. Esto implicaría que por defecto se utilicen las cámaras de 64 megapíxeles. Sin embargo, el proyecto como tal tiene un objetivo claro: minimizar el tiempo que se tarda llevando a cabo la inspección visual. Por esto, se busca que el tiempo de procesamiento y creación de reporte sea el menor posible. Por esto, entre las dos plataformas a escoger, se utilizará un Jetson Orin Nano, la versión más nueva de la plataforma Jetson para desarrolladores. Esta se observa en la figura 4.2. Esto descarta las cámaras de 64 megapíxeles. La siguiente mejor opción por lo tanto sería el sensor IMX519, de 16 megapíxeles. Todas estas opciones de cámaras son vendidas por Arducam en kits que contienen el multiplexor y los accesorios para conectar todo. El correspondiente a la opción seleccionada se observa en la figura 4.3.

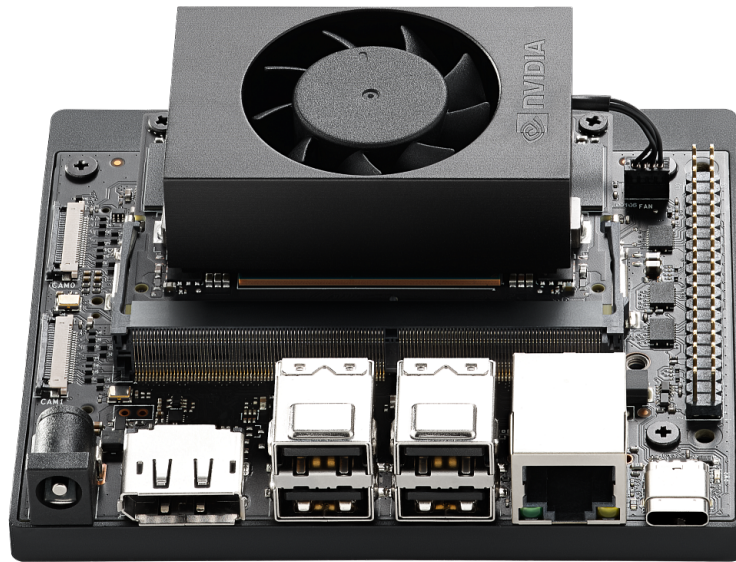


Figura 4.2: Jetson Orin Nano [28].

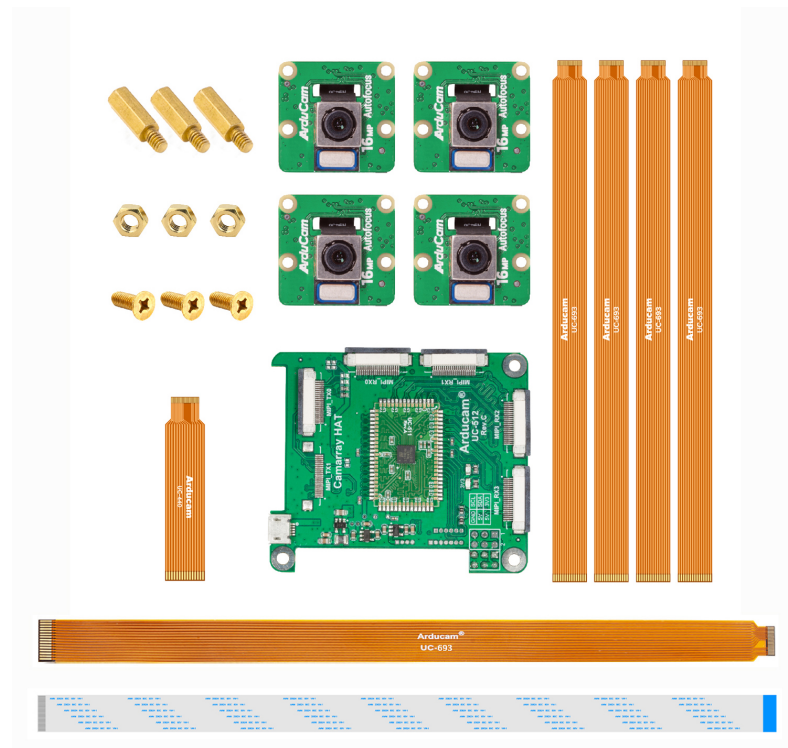


Figura 4.3: Kit del multiplexer con cámaras [27].

Nótese que el multiplexor tiene un máximo de cuatro entradas. Otra ventaja que tiene el Jetson Nano sobre la Raspberry Pi es que el Jetson tiene dos puertos CSI, ambos con el máximo de ancho de banda disponible. Si bien el microprocesador de la Raspberry Pi 4 y 5 tienen esta capacidad, la Raspberry Pi en su formato tradicional no tiene los dos puertos disponibles a nivel de hardware. Debe utilizarse el Compute Module y su placa madre. El Compute Module es, esencialmente, la Raspberry Pi en formato SODIMM, lo que reduce su tamaño. La placa de desarrollo para utilizar con el development kit sí tiene disponibles los dos puertos. Sin embargo, es sumamente difícil adquirir el Compute Module 4. En contraste, no hubo ningún problema con la adquisición del Jetson Nano. Incluso, había un descuento por educación disponible.

4.1.3. Conexión de cámaras al computador

Las cámaras son conectadas al computador a través de puertos CSI. Sin embargo, es muy importante mencionar que existen dos tipos de puertos CSI utilizados tradicionalmente: 15 pines, y 22 pines. 15 pines es el tamaño utilizado por versiones antiguas del Jetson Nano, y por la Raspberry Pi. El cable de 22 pines es utilizado por la Raspberry Pi Zero y Zero W, y por la Jetson Orin Nano, que es la Jetson Nano que se adquirió para el proyecto.

Los cables incluidos en el kit son de 15 pines en un extremo y de 22 pines en el otro extremo. Esto porque las cámaras utilizan un puerto de 15 pines, pero el multiplexor 2. No hay problema para conectar las cámaras al multiplexor. Sin embargo, tanto el multiplexor como el Jetson Nano utilizan 22 pines, por lo que no es posible conectarlos. Por fortuna, Arducam incluye dos cables para la conexión entre el multiplexor y el Jetson Nano. Se soldaron los dos cables en su lado de 15 pines y con esto se construyó un cable de 22 pines en ambos extremos, y de esta manera se conectan los multiplexores al Jetson Nano. El cableado de los componentes se observa en la figura 4.4

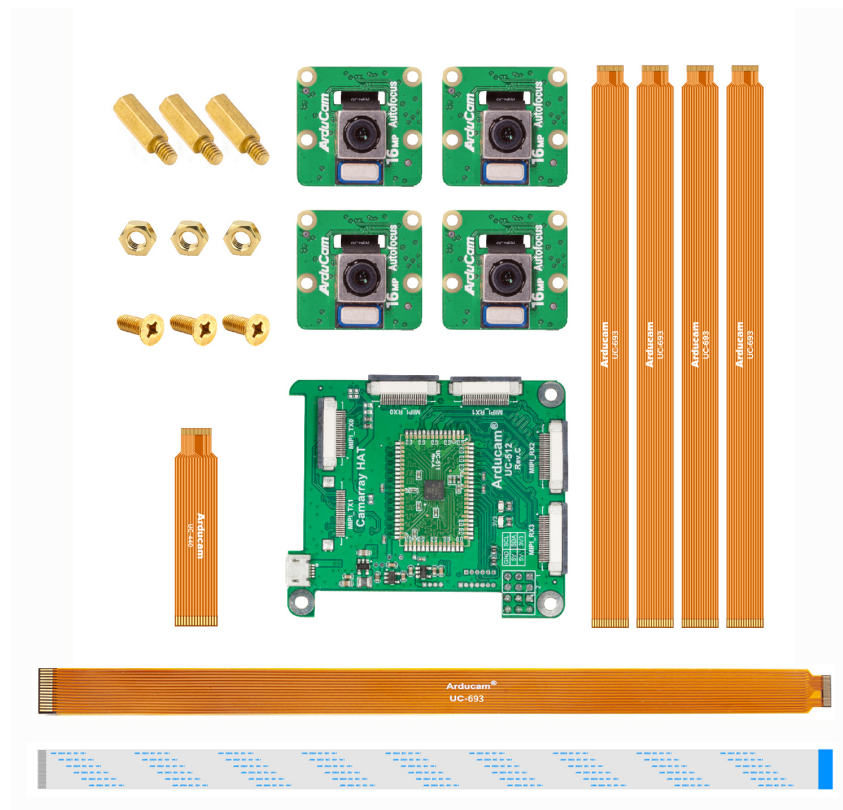


Figura 4.4: Cableado de cámaras con multiplexor y Jetson Nano. Fuente propia.

4.1.4. Iluminación

Para mejorar la calidad de las detecciones es de muy alta importancia minimizar el ruido. Una de las mejores maneras de lograr esto es aumentando la luz que hay en la escena al máximo. Esto permite reducir el ISO de la cámara, lo que se refleja en un menor ruido. El ruido se refleja como píxeles aleatorios en la imagen, y aparecerán como anomalías durante el procesamiento de imágenes.

Inicialmente durante la fase de metodología se decidió utilizar bombillas LED. Una de las razones de esto era por que se pensaba que tiras LED convencionales no tendrían suficiente luminiscencia para iluminar correctamente el satélite. Al llevar a cabo pruebas preliminares en el laboratorio fue evidente que no es cierto. Por otro lado las tiras LED tienen una gran ventaja en que no son un solo punto, sino que la fuente de luz está distribuida en un área mayor. Esto lleva a que la escena esté iluminada más uniformemente, y reduce las sombras.

Originalmente se pensaba comprar tiras LED y hacer un sistema de alimentación y control personalizado. Sin embargo, tras una muy corta búsqueda en catálogos electrónicos, se encontraron múltiples kits de tiras LED blancas de alta potencia con alimentación, cableado y control ya incluido. Esto facilitó significativamente el trabajo del sistema de iluminación. El kit es hecho por Inspire LED, y contiene múltiples barras de 21 LEDs de

alta potencia de aproximadamente 4300K que corresponde a un blanco neutro. La foto del kit del catálogo puede verse en la figura 4.5.



Figura 4.5: Kit del sistema de iluminación. [29]

4.2. Segmento Mecánico

El segmento mecánico es sin lugar a duda el más visual del proyecto, es como lo que se trabaja y lo que se manipula. Este incluye la estructura mecánica que soportará todos los componentes, los sistemas de montaje de las cámaras y el satélite, y el aislamiento de la luz externa respecto al interior de esta.

4.2.1. Identificación de los requerimientos

Nuevamente, de las necesidades de la tabla 3.2 se extraen las directamente relevantes a este segmento, y se añaden otras que emergen durante el desarrollo del proyecto. Esto puede darse por que se tiene nueva información obtenida tras mayor investigación, o también por que se van definiendo restricciones en recursos, en particular el tiempo. Los requerimientos resultantes son los siguientes:

- Ser capaz de inspeccionar todos los paneles a la vez

- Soportar tamaños diferentes a 2U
- Funcionar con tolerancia a pequeños cambios de luz y posición
- Ser simple de utilizar por nuevos usuarios

Aparte de un marco para montar todos los componentes, el aislamiento de la iluminación implica la instalación de paredes. Además, los sistemas de montaje tanto del satélite como de las cámaras deben ser robustos y rígidos, para minimizar falsos positivos que surjan a raíz de pequeños movimientos durante las pruebas.

El trabajo de diseño CAD fue llevado a cabo en Fusion 360 debido a su fácil uso y que contiene todas las características necesarias para este proyecto.

4.2.2. Estructura principal

La estructura es esencialmente un cubo, donde cada arista es un trozo de extrusión de aluminio 6061. Estas piezas son unidas a través de soportes en L. El modelo 3D se aprecia en la figura 4.6.

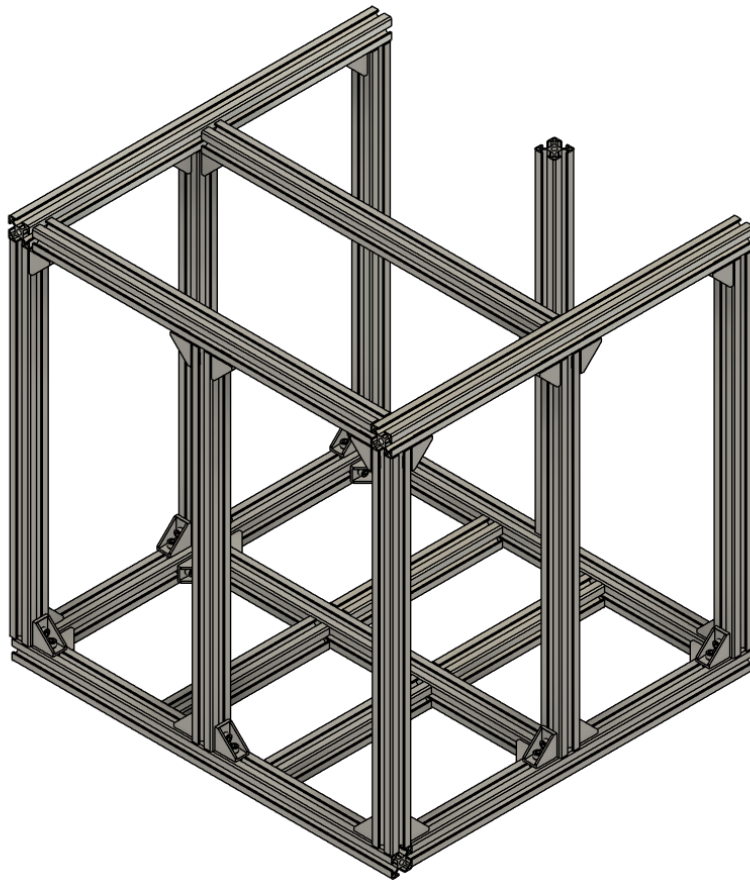


Figura 4.6: Kit del sistema de iluminación. Fuente propia

Nótese que en la cara trasera derecha no existe una viga en el lado superior. Esto es porque esta cara fue designada como la puerta al interior de la maquina, y por lo tanto la barra superior no es necesaria. La barra inferior sí, ya que está conectada a otras barras sobre las cuales se montará el soporte del satélite. La barra central está fijada a la cara inferior. La pared que funciona como puerta será fijada a una de las caras con una bisagra.

Las únicas cargas mecánicas ejercidas sobre este armazón es la fuerza de la gravedad de los componentes. Por ejemplo, en la figura 4.7 se observan los resultados de una simulación de análisis estático, donde la viga potencialmente más débil es sometida ante los esfuerzos de gravedad causados por la cámara y su propio peso. El desplazamiento máximo es de 0.65 micrómetros, y el estrés máximo von Mises es de 0.2 MPa. Esto despeja cualquier duda acerca de un posible efecto en el ángulo de la cámara o estrés estructural. Las demás vigas son aún más resistentes ya que están fijas en ambos extremos.

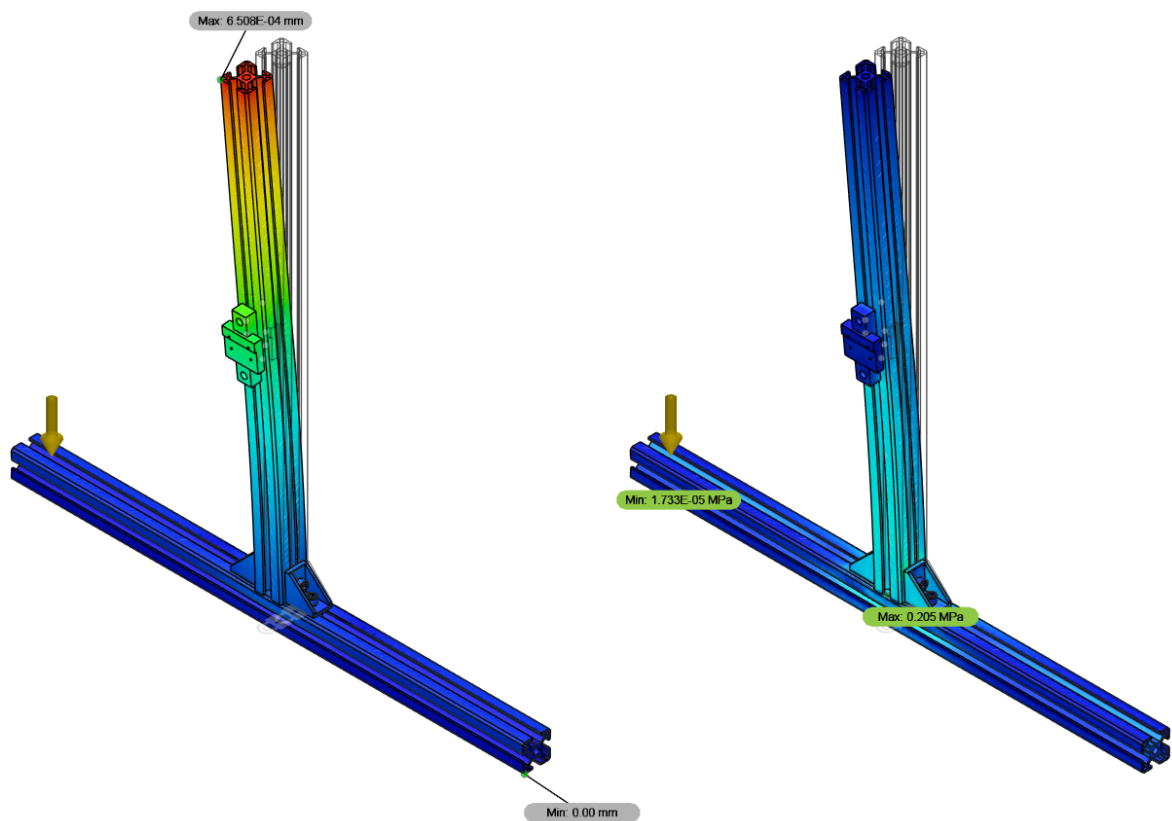


Figura 4.7: Desplazamiento máximo y estrés von Mises en la viga de la puerta. Fuente propia

El tamaño del marco fue determinado por el tamaño del satélite a probar, y por las distancias focales de las cámaras. De acuerdo a la documentación de las cámaras [27] la distancia focal mínima es de 8 cm. Con este dato, se colocan las cámaras a esta distancia de cada cara del satélite para maximizar los píxeles por centímetro. En el caso del lado largo del satélite, la cámara se coloca de tal forma que los bordes del satélite queden justo dentro del marco del video, maximizando nuevamente la resolución por centímetro. Esto significa que se podrán probar únicamente satélites de 1 y 2 unidades, cumpliendo el requerimiento de soportar múltiples tamaños. Una vez que las cámaras están en sus

posiciones finales, se colocan las vigas. Con el diseño CAD terminado, se procede al corte y construcción del marco.

4.2.3. Soporte del satélite

El soporte del satélite es el componente mecánico más crítico de todos, y definitivamente el que pasó por más iteraciones. El satélite debe encajar perfectamente en el soporte **cada vez**, ya que cualquier varianza va a afectar significativamente la identificación de diferencias. Esta fue la principal dificultad a la hora de realizar pruebas del software en paralelo con la estructura mecánica: fue extremadamente difícil tener fotos del satélite antes y después de aflojar un tornillo sin que el satélite como tal se moviera.

Otro punto muy importante es que el soporte no puede ocultar ninguna parte del satélite. Originalmente se pensaba que simplemente dejando agujeros para que las cámaras fotografíen los tornillos era suficiente. Sin embargo, por cuestiones de certificación de metodologías de pruebas ambientales para JAXA, se debe mostrar la cara completa del satélite.

La primera idea fue tomar fotos de tres caras del satélite, y luego rotar a este para tomar fotos a las otras tres caras, pero esto va completamente en contra del objetivo de agilizar el proceso. Tras discutir con el experto de las certificaciones y pruebas ambientales, se llegó al compromiso de que es posible utilizar materiales transparentes, como el acrílico, para soportar el satélite y aún así ver el marco.

En la figura 4.8 se aprecia la última iteración del soporte. Este está hecho de acrílico y se monta en cuatro vigas de extrusión de aluminio que en sus otros extremos se fijan en la cara inferior de la cámara. En la misma imagen se observa la estructura del satélite Dragonfly, con el que se realizará la validación del sistema.

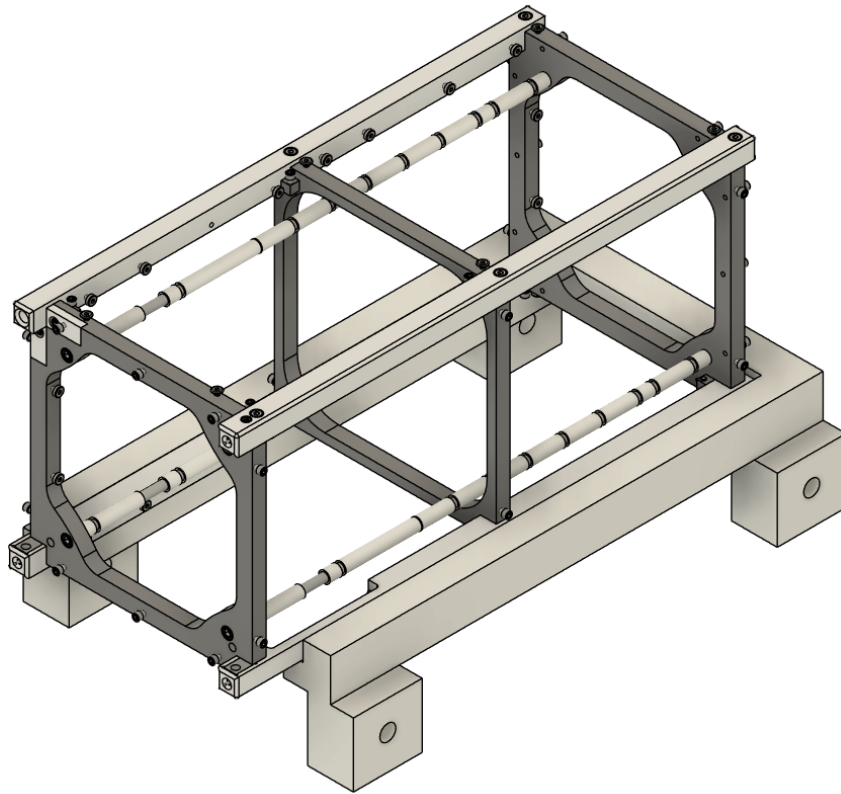


Figura 4.8: Modelo CAD del soporte del satélite. Fuente propia

4.2.4. Soportes de cámaras

Los soportes de las cámaras son pequeñas piezas plásticas que fueron impresas en 3D que unen las cámaras con el marco. Estas también pasaron por múltiples iteraciones, pero principalmente para afinar la geometría de la superficie para que la cámara, que tiene una superficie irregular, se instale lo más paralelamente posible al soporte, tal que esté apuntando directamente al centro del armazón. Es una pieza relativamente sencilla, y su modelo CAD se aprecia a en la figura 4.9. Se utilizó un material opaco para facilitar su visualización.

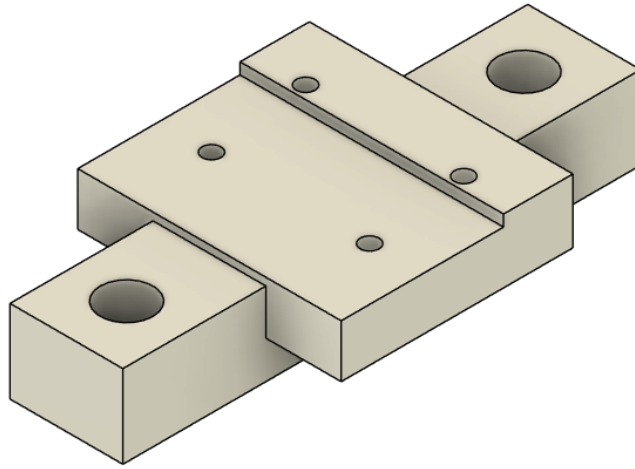


Figura 4.9: Modelo CAD del soporte de la cámara. Fuente propia

El soporte se fija al marco a través de dos tornillos que pasan a través del soporte y se atornillan en dos tuercas especiales dentro del perfil de aluminio. Al atornillar, las tuercas forman una unión de fricción y se vuelve prácticamente imposible que se deslice el soporte a lo largo del perfil, manteniendo así la posición de la cámara sin importar cuánto se mueva la estructura.

4.2.5. Sistema ensamblado

El ensamble completo se observa en la figura [4.10](#).

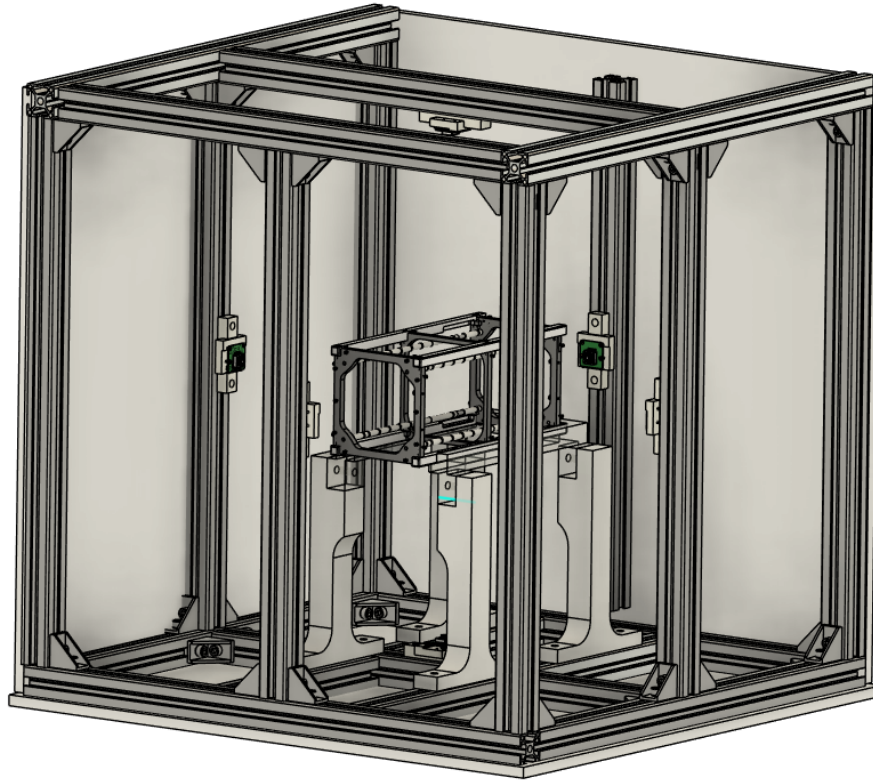


Figura 4.10: Modelo CAD de ensamblaje completo. Fuente propia

4.3. Segmento Software

El segmento de software presentó desafíos significativos comparado con los otros segmentos, en particular el control del enfoque de las cámaras y la generación de reportes.

4.3.1. Identificación de los requerimientos

Por última vez, se extraen las necesidades de la tabla 3.2 las relevantes al segmento, y se añaden otras que surgen durante el desarrollo del proyecto.. Los requerimientos resultantes son los siguientes:

- Ser capaz de inspeccionar todos los paneles de una vez
- Detectar errores de marca de torque
- Detectar rasguños en el vidrio
- Detectar grietas en el vidrio
- Funcionar con tolerancia a pequeños cambios de luz y posición

- Automáticamente crear reportes de prueba si es exitoso
- Automáticamente crear reporte de fallos si no es exitoso
- Ser fácil de instalar en diferentes computadoras
- Ser simple de utilizar por nuevos usuarios
- Permitir que entradas de usuario definan si una detección es una falla
- Integrarse en los sistemas de pruebas actuales en Labview
- Tener una interfaz de usuario gráfica amigable

Nótese que son todos los requerimientos. Esto demuestra lo complejo que es el segmento, y se podría decir que es el componente principal del proyecto.

Rápidamente se descartó el punto de que sea compatible con el sistema actual de Labview, y se dejará como un posible proyecto futuro. El trabajo necesario para esto aumentaría el alcance del proyecto significativamente, y no sería posible realizarlo con los recursos disponibles, en particular el tiempo.

4.3.2. Ambiente de desarrollo

Primero se debe definir un ambiente de desarrollo y crear un flujo de trabajo. Es muy importante que se pueda trabajar en el sistema de manera remota incluso fuera de la red de área local. Esto porque la estancia en Japón no alcanzó para terminar el segmento de software, por lo que se dió un enfoque a terminar todo lo relacionado a hardware, y llevar a cabo la programación desde Costa Rica.

La Jetson Orin Nano usa una versión de Ubuntu 20.04 modificada por NVidia, llamado Linux For Tegra (L4T). Esta versión modificada incluye muchas herramientas para desarrollo de software que haga uso de las capacidades del hardware especializado de NVidia, como por ejemplo núcleos CUDA, o más relevante para este proyecto, el hardware que soporta las cámaras.

Para trabajar remotamente se probaron muchas soluciones de uso de escritorio remoto. Al final, se decidió utilizar Teamviewer, por ser gratis de uso muy sencillo. Teamviewer permite controlar el otro computador a través de usuario y contraseña, en vez de IP que en el caso de Kyutech, es dinámica. Mientras el Jetson esté conectado al internet y a la red eléctrica, se puede trabajar desde cualquier parte del mundo.

Por cuestiones de familiaridad, se decidió hacer el procesamiento de imágenes con el módulo OpenCV 2. Este es un conjunto de herramientas para análisis y manipulación de imágenes de manera *tradicional*, es decir, sin utilizar ninguna red neural. L4T ya incluye este módulo, por lo que ni si quiera su instalación es necesaria.

Un problema muy importante es que no fue posible hacer que la Jetson funcionara de manera *headless*. Esto significa que la computadora funcione sin un monitor conectado. Se determinó que si el procesador no detecta un dispositivo de video, otros periféricos no funcionan, como lo son las cámaras. No se logró encontrar una solución a este problema a tiempo, por lo que se tiene un monitor dedicado a este proyecto a pesar que todo el trabajo se hace remoto.

4.3.3. Acceso y control de cámaras

El acceso y control de cámaras fue el principal problema dentro del segmento de software. Reiterando, se está utilizando una Jetson **Orin Nano**, Orin siendo la línea más nueva de este dispositivo, lanzado al mercado unas escasas semanas previo al inicio del proyecto. Si bien esto sonaba como una excelente idea, pensando en sus capacidades computacionales, esto trajo muchos problemas en cuanto a compatibilidad de código existente por parte de Arducam.

El acceso al video de las cámaras se da a través de un flujo llamado GStreamer. Este es una herramienta sumamente potente sobre la cual muchas aplicaciones de transmisión de video están construidas. Un flujo o conducto es el conjunto de operaciones y elementos que utiliza GStreamer para la transmisión de video, y esencialmente une el sensor de la cámara con la aplicación que usará el video.

La cantidad de elementos y parámetros a usar en GStreamer es funcionalmente infinita, y es completamente dependiente de hardware, especialmente en el caso de L4T que tiene su propia versión de GStreamer que hace uso de hardware de NVidia.

Tras muchas pruebas no funcionales y fusionando múltiples ejemplos de NVidia, se obtuvo el siguiente conducto:

```
nvarguscamerasrc ! video/x-raw(memory:NVMM), width=(int)4654,
height=(int)3496, format=(string)NV12, framerate=(fraction)9/1 ! nvvidconv !
video/x-raw, format=(string)BGRx ! videoconvert ! appsink
```

Toda la información relevante se puede obtener en la documentación de GStream [30]. A continuación se descompone cada parte del conducto:

- **nvarguscamerasrc**: es el elemento fuente provisto por NVidia. Directamente toma los cuadros del sensor y los deja disponibles para el siguiente elemento
- **video/x-raw(memory:NVMM)**: este elemento especifica cómo almacenar los cuadros en memoria, y en qué tipo de memoria. NVMM es un tipo de memoria propietario de NVidia optimizado para sus procesadores gráficos.
- **width=(int)4654, height=(int)3496**: especifica el tamaño de los cuadros en píxeles

- `format=(string)NV12`: es un formato de codificación de píxeles, comúnmente utilizado en aplicaciones de procesamiento de video.
- `framerate=(fraction)9/1`: especifica el número de cuadros por segundo. Las cámaras determinan qué opciones hay. A su máxima resolución solo se pueden utilizar 9 cuadros. A 1080P, se pueden utilizar hasta 60.
- `nvvidconv`: es un elemento desarrollado por NVidia para convertir formato de video utilizando hardware optimizado.
- `video/x-raw, format=(string)BGRx`: es necesario convertir el formato de video utilizado por NVidia a nivel de hardware a un formato utilizado por OpenCV para procesamiento de imágenes.
- `videoconvert`: un conversor de formato de video genérico, utilizado para asegurar que el video es compatible con `appsink`
- `appsink`: es el elemento de salida, es la interfaz entre el conducto y la aplicación final, en este caso, Python. Otros sinks pueden abrir una ventana directamente en el sistema operativo y mostrar el video.

Al ejecutar esto, se obtendrá video desenfocado de cuatro cámaras al mismo tiempo. Por defecto, el multiplexor disminuye la resolución del video y junta las cuatro cámaras en un solo flujo de video. Esto debe configurarse para tener una transmisión de video individual de cada cámara, y además debe controlarse el motor de enfoque.

Ambas operaciones utilizan la interfaz I2C. El multiplexor está conectado al Jetson a través del puerto CSI para la transmisión de video, pero para el control de las cámaras utiliza esta interfaz. Esencialmente, se envían bytes en una línea mientras que otra línea funciona para sincronizar el reloj de la transmisión. Para controlar cuál cámara se ve en un determinado momento, se refiere a la documentación de Arducam [31]:

```
# Set to single channel 0
i2cset -y 10 0x24 0x24 0x02
# Set to single channel 1
i2cset -y 10 0x24 0x24 0x12
# Set to single channel 2
i2cset -y 10 0x24 0x24 0x22
# Set to single channel 3
i2cset -y 10 0x24 0x24 0x32
# Set to double channel (single channel 0 and single channel 1)
i2cset -y 10 0x24 0x24 0x01
# Set to double channel (single channel 2 and single channel 3)
i2cset -y 10 0x24 0x24 0x11
# Set to four in one mode (Default)
i2cset -y 10 0x24 0x24 0x00
```

Esencialmente, dependiendo si se quieren cuatro cámaras, dos o una, se envían ciertos bytes por I2C para configurar el multiplexor. En la solución del proyecto se fija el canal 1, se toma la fotografía, se cambia al canal, se toma la segunda fotografía, y así sucesivamente para las cuatro cámaras. Recuérdese que para las seis cámaras se utilizan dos multiplexores.

Finalmente, para controlar el enfoque, se envía de la misma manera un comando por I2C. Este comando sí resultó significativamente más complejo, por lo que se reutilizó código dado por el fabricante Arducam. Se envía un número de 0 a 1023, donde 0 es el enfoque más lejano y 1023 el más cercano, de 8 centímetros aproximadamente. El número se afinó manualmente para cada cámara una vez instaladas en el marco.

4.3.4. Identificador de anomalías

El identificador de anomalías es el núcleo del proyecto. Este toma las fotos tomadas por las cámaras antes y después de la prueba de vibración y las compara para buscar algún cambio. Como se mencionó anteriormente, este procesamiento se realizó en OpenCV por la familiaridad previa con este módulo.

Los pasos para identificar anomalías siguen el siguiente flujo:

1. **Toma de imágenes:** los cuadros se obtienen del conducto de GStreamer y se almacenan permanentemente en el computador para archivado.
2. **Cálculo de diferencia:** utilizando el comando de substracción `cv2.absdiff` se calcula la diferencia entre los píxeles de las imágenes, dando como resultado una imagen a color `diff` con únicamente los cambios.
3. **Conversión a grises:** en preparación a etapas de filtrado la imagen a color se convierte a escala de grises.
4. **Umbralizado de la imagen:** se le aplica una umbralización a la imagen en escala de grises. Esta operación define un umbral, y los valores debajo y sobre este umbral se convierten a un absoluto negro o blanco respectivamente.
5. **Filtrado:** formado con dos subetapas, el filtrado elimina ruido y otras píxeles no deseados de la imagen, dejando únicamente las regiones principales de la resta. Primero se erosiona la imagen y luego se dilata, como fue explicado en la sección [2.6](#).
6. **Extracción de contornos:** se dibujan rectángulos alrededor de las regiones finales y luego se superponen sobre la segunda imagen original, mostrando así las regiones donde se encuentra un cambio.

Las etapas de umbralizado y filtrado tienen múltiples parámetros. Para encontrar los valores ideales para estas operaciones se ejecutó una prueba masiva. Se construye una

matriz con muchos valores para cada parámetro y se extraen las diferencias a todas las posibles combinaciones. Posteriormente se analizan manualmente los resultados y se definen los valores ideales. En esta prueba se incluye también el enfoque de las cámaras. La prueba masiva tuvo un total de 1600 pruebas. Cientos de estas pruebas fueron fácilmente descartables porque no quitaban nada de ruido o eliminaban todas las características. Un análisis minucioso fue llevado a cabo con 64 combinaciones de valores. Los parámetros resultantes fueron:

- Umbral: 50
- Iteraciones de erosión: 1
- Iteraciones de dilatación: 15
- Tamaño de kernel: 5x5
- Enfoque: 420 y 650 (para lado corto)

4.3.5. Interfaz de usuario

La interfaz de usuario es la unión entre el hardware y el producto final del proyecto, los reportes. El proceso de generación de reportes como tal se describe en la sección 4.3.6. La interfaz de usuario (GUI por sus siglas en inglés) diseñada tiene dos ventanas principales: la ventana principal de toma de imágenes, el procesamiento de estas y la inspección de anomalías encontradas, y la ventana de toma de datos para la generación de reportes. Todos los aspectos de interfaz e interacción con el usuario se hicieron utilizando el módulo de Python `TKInter`.

En la ventana principal se podrán ver dos transmisiones de video. Estas provienen de la una misma cámara. Debajo de estas transmisiones se encontrarán botones. Al presionar uno de estos botones, se tomará una foto y se congelará la transmisión de video correspondiente. Esta foto tomada se usará para la identificación de anomalías, ya sea como la foto de “antes” o la foto de “después” de la prueba de vibraciones. Se tendrá un indicador extra para confirmar que la foto fue tomada correctamente. Se tiene la transmisión de video de una sola cámara de las 6, ya que estas transmisiones son simplemente para confirmar visualmente que las cámaras funcionan correctamente. Internamente, de una manera no expuesta al usuario a través del GUI se toman fotos de las demás cámaras.

Existe otro botón, esta vez con el texto “procesar”. Al presionar este botón, se inicia todo el proceso de identificación de anomalías con las fotos tomadas. Al terminar, se muestra la foto de la etapa de dilatación, que funciona como una máscara que muestra dónde están las anomalías encontradas, si existen, y la foto “después” anteriormente tomada, esta vez con rectángulos que indican dónde están las anomalías.

Finalmente, en la parte derecha de la ventana hay una lista. Esta lista se llena con las coordenadas de las anomalías encontradas en la etapa de identificación de anomalías. Al

hacer click en un elemento de la lista, se abre una imagen con bastante acercamiento que permite analizar muy detalladamente la anomalía, y decidir si realmente es un error o no.

Al hacer click en el botón con el texto "generar reporte" se transforma la ventana a una mucho más pequeña y simple. Esta tiene 3 espacios de texto. En estos se escribe el nombre del proyecto, el nombre del satélite y el encargado de la prueba de vibración. Estos son los datos, junto con la fecha y las fotos tomadas, que varían para cada prueba en cuanto a la elaboración de un reporte para aspectos de certificación. El resto del contenido del reporte es constante entre pruebas.

4.3.6. Generación de reportes

El resultado final del proyecto es un reporte con la información correspondiente a la prueba. Se planteó generar dos reportes: un reporte de fallas, en caso de que se encuentren anomalías, y un reporte de certificación, en caso de que no hayan errores.

El primero es el más sencillo. Consiste en una portada con identificación de la prueba, el proyecto y el encargado, seguido de un listado de las imágenes de las caras con un antes y después, y finalmente una sección con las fotos con aumento donde detalladamente se analizan las anomalías.

El reporte de certificación es significativamente más complejo. Para crearlo primero se debe identificar el formato y contenido requeridos por la universidad y por la Agencia Espacial Japonesa. Por fortuna, existe un repositorio de reportes de misiones pasadas. Al estudiarlos y compararlos, se identificó qué contenido cambia entre pruebas y qué contenido es constante. Entre el contenido que cambia están algunos datos simples como nombre del encargado, del proyecto y del satélite, además de las fotos correspondientes a la inspección visual. Es importante mencionar que además existen mucho contenido adicional que cambia, específicamente relacionado a los análisis de respuesta a frecuencia. Esto incluye imágenes y contenido en tablas. Como todos los aspectos de análisis de respuesta a frecuencia están fuera del alcance de este proyecto, el contenido relacionado a esto en el reporte se marcará en un color llamativo y se hará una nota para que el encargado proceda a llenar esta información manualmente.

Tomando esto en cuenta, la solución para esta parte fue tomar una plantilla del reporte, y en la misma aplicación del procesado de imágenes solicitar al usuario la información básica del proyecto. Seguidamente con los resultados de las pruebas se llenan las secciones respectivas de la plantilla, y se produce un documento de Word .docx.

Para la edición del documento .docx se utiliza un módulo llamado python-docx. Este genera una lista de todos los párrafos del documento. El inicio y final de un párrafo puede identificarse en la aplicación Word por el símbolo ¶. Para verse debe habilitarse presionando el botón con el mismo símbolo, o bien con el atajo `ctrl + *`. La lista generada por python-docx son todos estos párrafos de manera secuencial. Ya que la plantilla es constante, se conoce previamente dónde deben insertarse los datos relevantes para cada

sección.

Capítulo 5

Resultados y análisis

El principal entregable del proyecto es la máquina que ejecuta las pruebas, por lo que se discutirá la construcción de esta como un resultado. Seguidamente se tratará el tema de la validación del funcionamiento, y se terminará con un resumen sobre los gastos correspondientes al proyecto.

5.1. Construcción del prototipo

Los primeros componentes que llegaron al laboratorio fueron las cámaras y el Jetson Nano, por lo que se empezó con el segmento eléctrico. Lo primero fue cablear todo, y aquí fue donde se descubrió el problema con los cables mencionado en el capítulo 4. En la figura 5.1 se observa la solución mencionada al problema de los tamaños de los cables. En la figura 5.2 se aprecia el jetson con un módulo multiplexor y una cámara conectada a este. Durante una gran parte del desarrollo del prototipo se utilizó una sola cámara para las pruebas, ya que una vez que funcionara con una, solo se debe iterar sobre las demás cámaras el mismo proceso.

Nótese que la cámara ya está montada sobre un soporte. Este fue el primer soporte impreso en 3D que calzó correctamente sin interferir con los componentes en la placa de la cámara.



Figura 5.1: Cable formado con dos cables para obtener los conectores correctos. Fuente propia

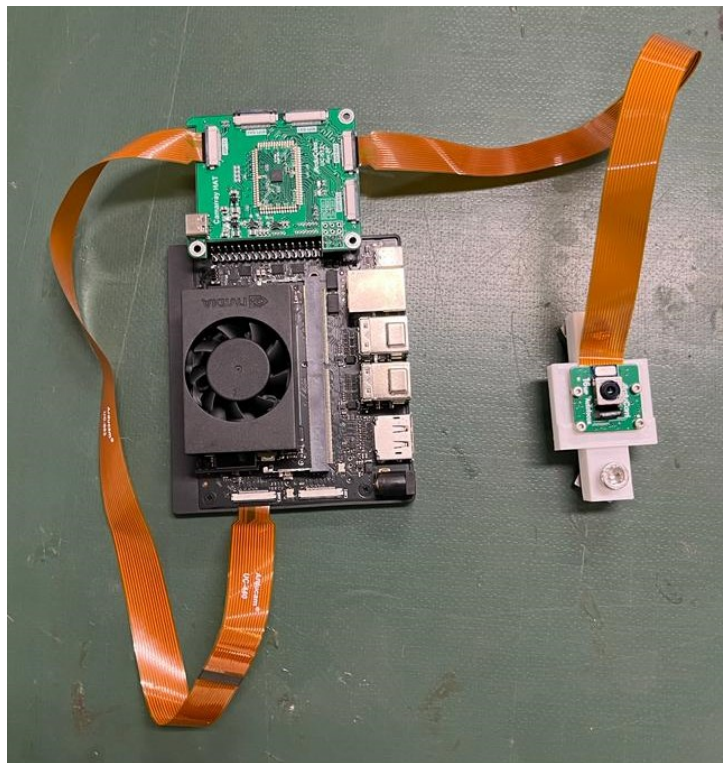


Figura 5.2: Jetson Orin Nano con una cámara conectada. Fuente propia

Mientras se trabajaba en el software, las vigas de extrusión de aluminio llegaron, por lo que se procedió a construcciones iniciales del armazón, durante los cuales se afinaban detalles y se modificaba el diseño hasta el que se apreció en el capítulo 4. En la figura 5.3 se aprecia el primer ensamble de la estructura. Es un prisma de 50cm x 50cm x 54cm.

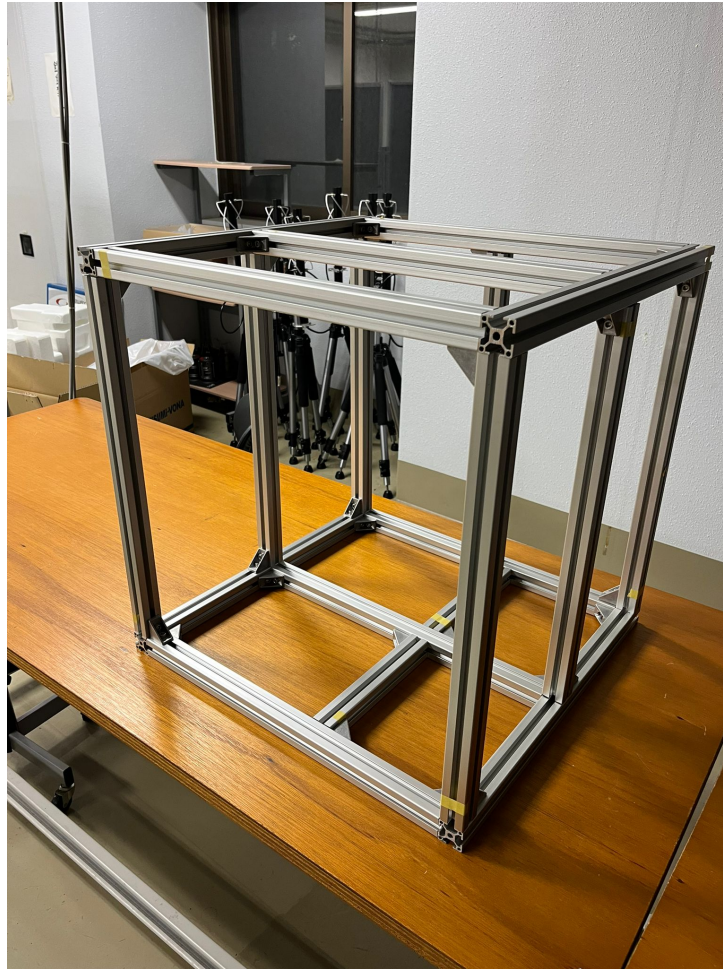


Figura 5.3: Primer ensamble del armazón. Fuente propia

Con esta estructura lista, se adquirió una estructura de satélite vieja que se podía manipular sin problema para realizar prueba. Para esto, se construyó un soporte temporal para sostener la pieza mientras se hacían pruebas con las cámaras. Esta estructura de satélite con el soporte temporal se aprecia en la figura 5.4.

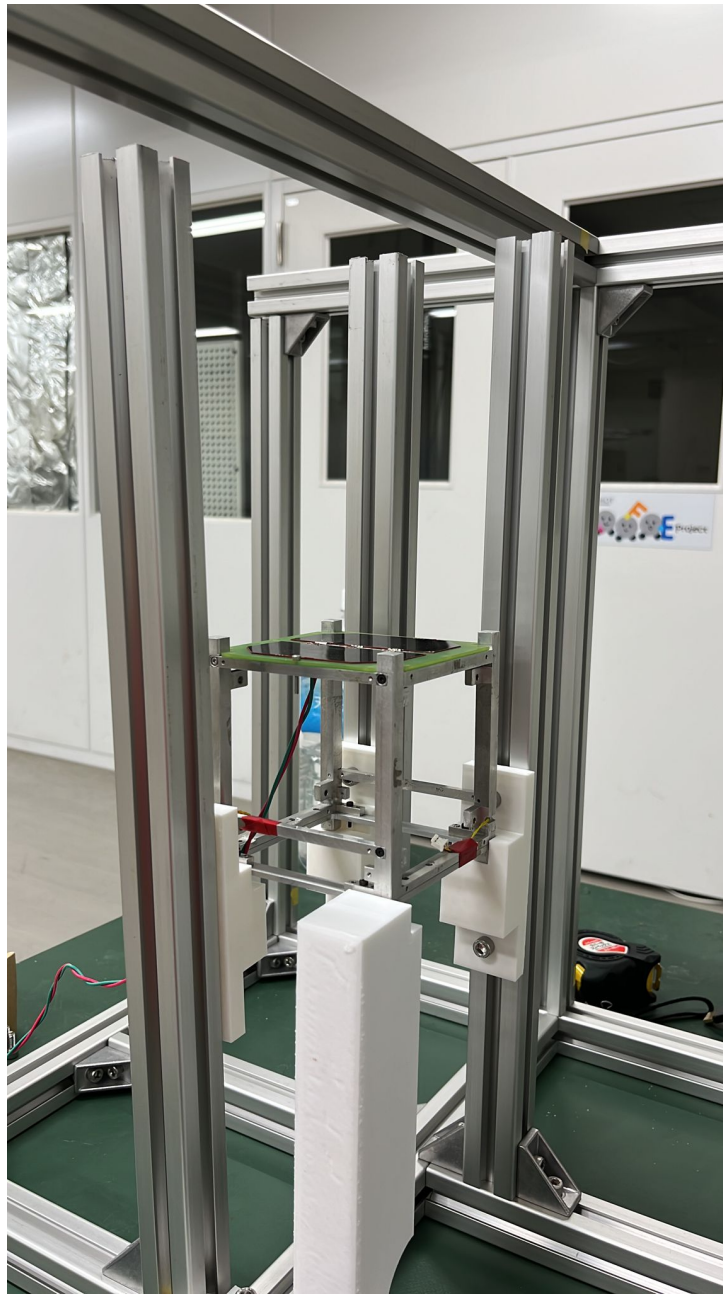


Figura 5.4: Soporte y estructura de satélite temporal usados para pruebas de cámara. Fuente propia

Con esto se lograron hacer pruebas preliminares suficientes para lograr diseñar y posteriormente construir el sistema final. Se decidió que las paredes iban a tener un interior de papel aluminio para que reflejen la luz de manera uniforme, y se imprimió el soporte final mientras se fabrica en acrílico. Esta estructura terminada se aprecia en la figura 5.5.

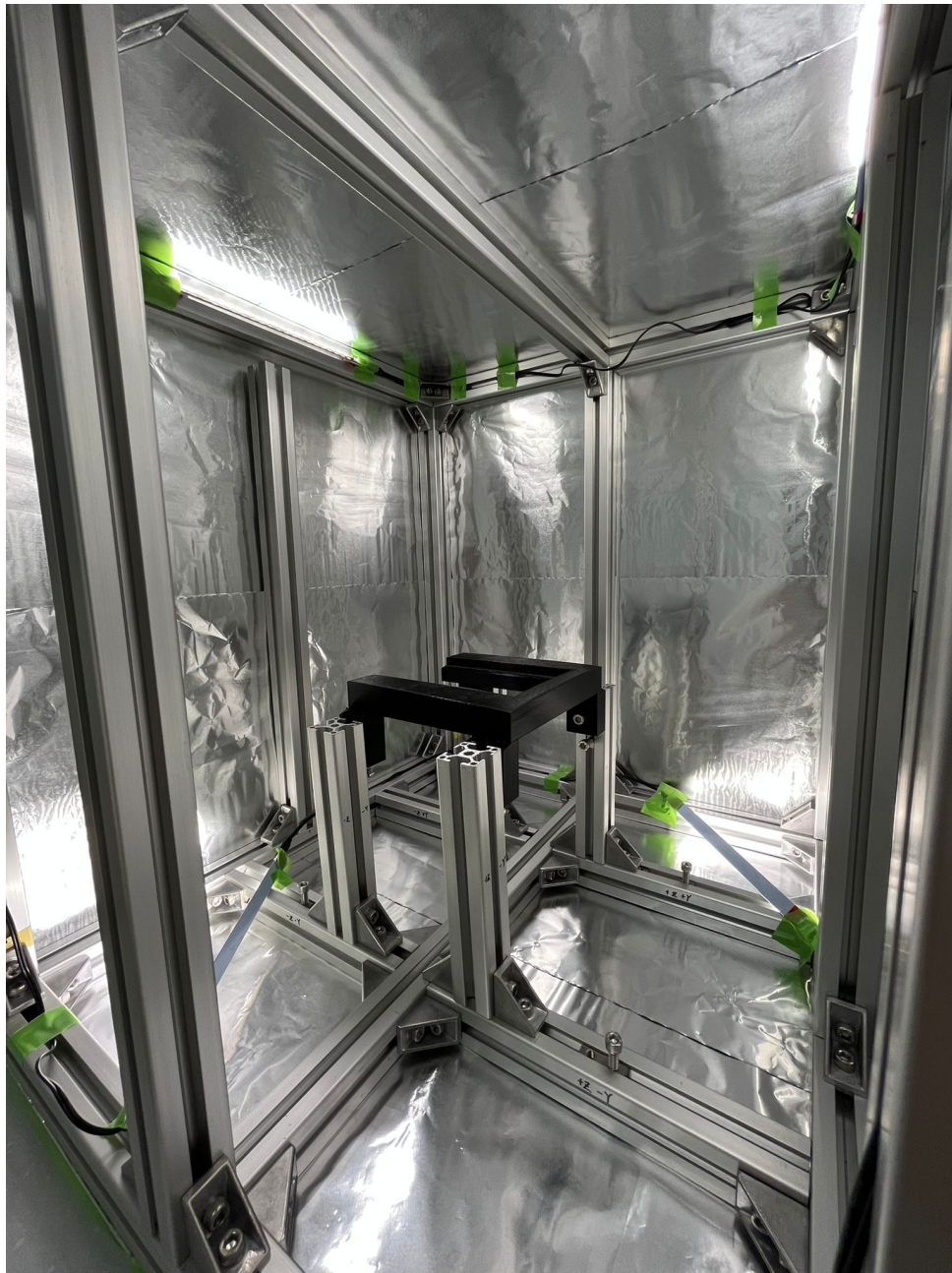


Figura 5.5: Interior de la estructura final. Pendiente en este punto: soporte en acrílico, montaje de cámaras. Fuente propia

En la figura 5.6 se aprecia el interior de la estructura terminado, con las cámaras montadas y el soporte fabricado con acrílico, para ver el satélite completamente.



Figura 5.6: Interior de la estructura final. Pendiente en este punto: soporte en acrílico, montaje de cámaras. Fuente propia

El cableado de las cámaras volvió a ser un desafío. Los cables de las cámaras incluidos en los kits eran de 30cm de longitud, lo cuál no es suficiente considerando que el tamaño de un lado de la estructura es de al menos 50cm. Por esto, se construyeron cables con extensiones para lograr alcanzar. Aún con estas extensiones fue necesario separar los multiplexores de la Jetson. Funcionalmente no hay ningún problema, pero estéticamente se podría mejorar.

5.2. Identificador de anomalías

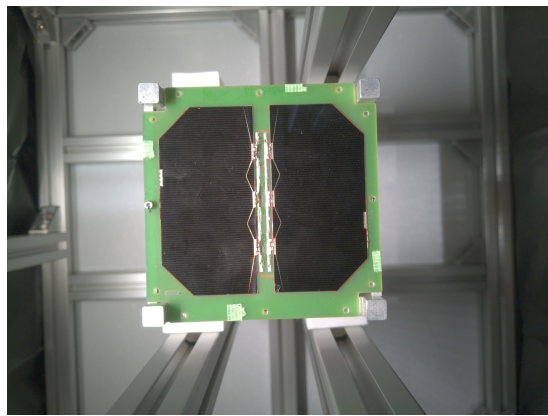
Al finalizar la afinación de parámetros se tiene un sistema que identifica rasguños, grietas y tornillos flojos **de manera exitosa**. En la figura 5.7 se observan las 8 etapas. Se invita al lector a que intente encontrar la diferencia en las dos primeras imágenes 5.7a y 5.7b. Este tamaño aproximadamente tienen las imágenes en los reportes oficiales a la agencia espacial japonesa.

En la etapa de erosionado (Fig. 5.7e) y dilatación (Fig. 5.7f) se aprecia por qué controlar las variables ambientales como la iluminación es tan crítico. Si se tuviera más ruido que filtrar, se perdería la información. Sí es importante notar que las imágenes adjuntas en este documento son capturas de pantalla de una previsualización con un 30% de la resolución original. El algoritmo trabaja con significativamente más información que la mostrada. Cabe destacar que estas imágenes son un caso extremo, donde el satélite está desalineado entre las fotos por un par de milímetros aproximadamente. Con las versiones nuevas del

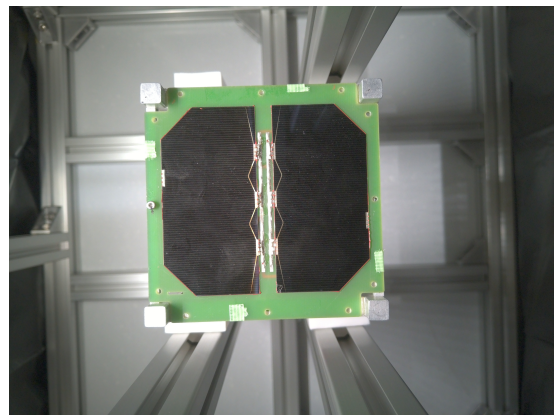
soporte del satélite esto no sucede.

En la figura 5.8 se aprecia el mismo comportamiento para el caso de una grieta en el panel. Para esto se tomo una pieza del mismo cristal que cubre celdas solares y se creo una grieta idéntica a las causadas por errores de manipulación. Se puede notar cómo identifica esta grieta de manera exitosa.

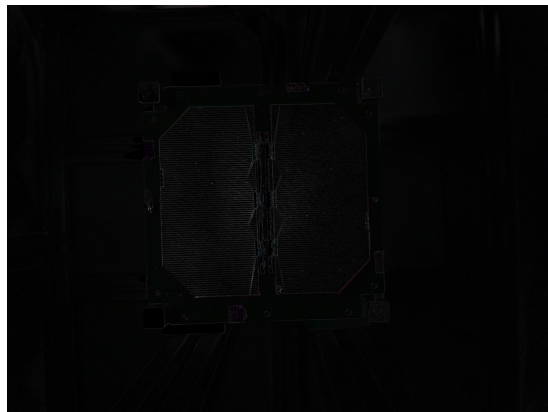
Finalmente, en la figura 5.9 se aprecia el mismo comportamiento exitoso para rasguños en el marco estructural del satélite.



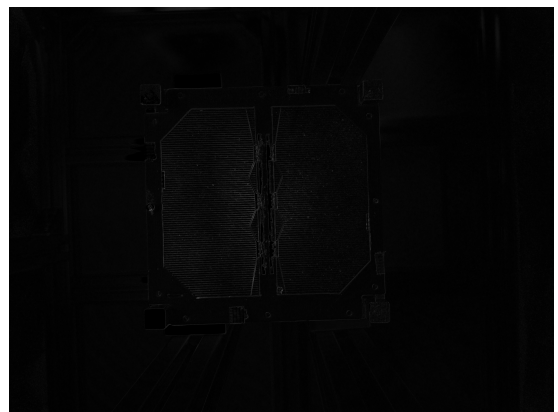
(a) Imagen pre-vibración.



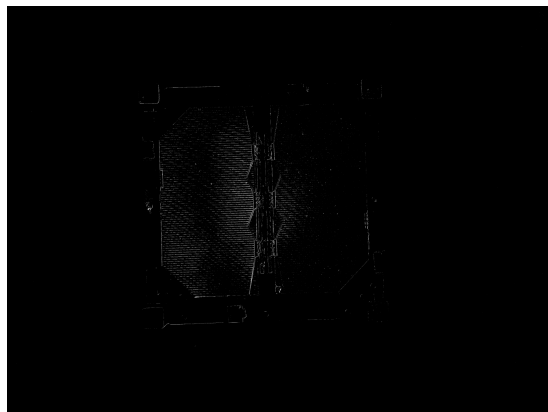
(b) Imagen post-vibración.



(c) Diferencia



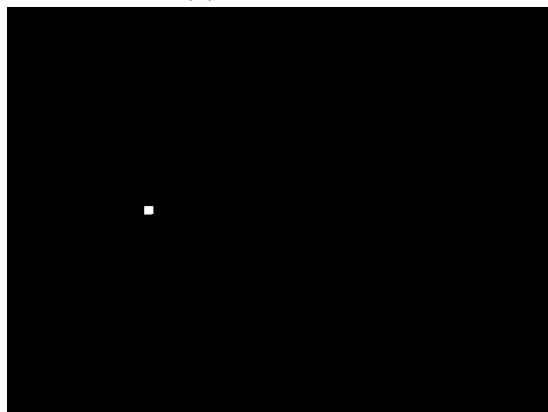
(d) Diferencia en gris



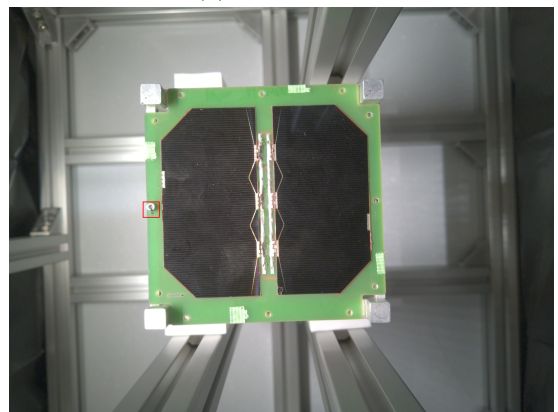
(e) Umbralizado



(f) Erosionado

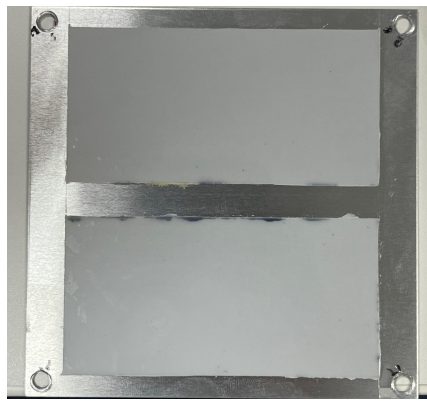


(g) Dilatación

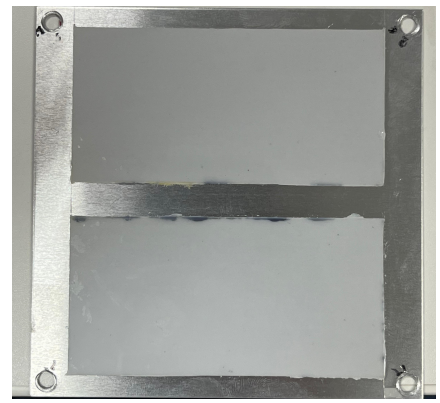


(h) Final

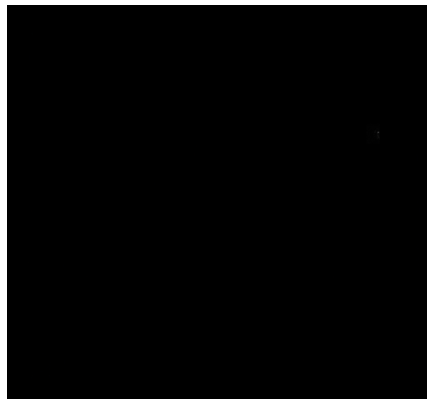
Figura 5.7: Resultados intermedios para un caso de tornillo flojo. Fuente propia.



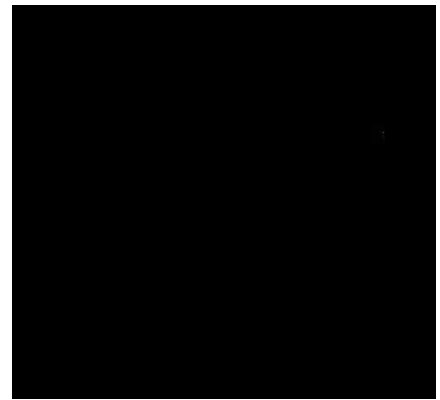
(a) Imagen pre-vibración.



(b) Imagen post-vibración.



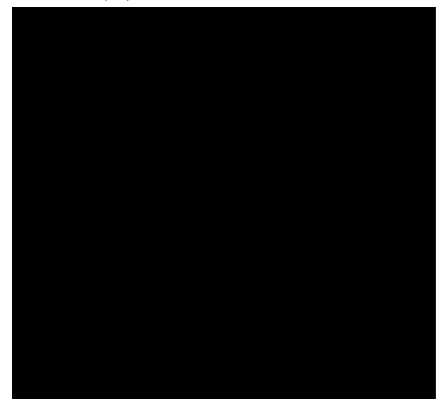
(c) Diferencia



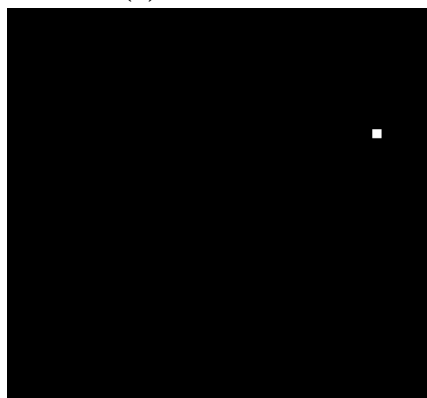
(d) Diferencia en gris



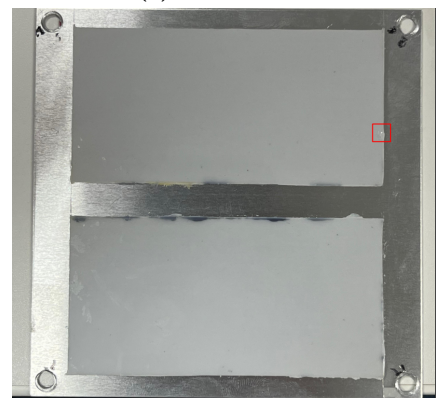
(e) Umbralizado



(f) Erosionado

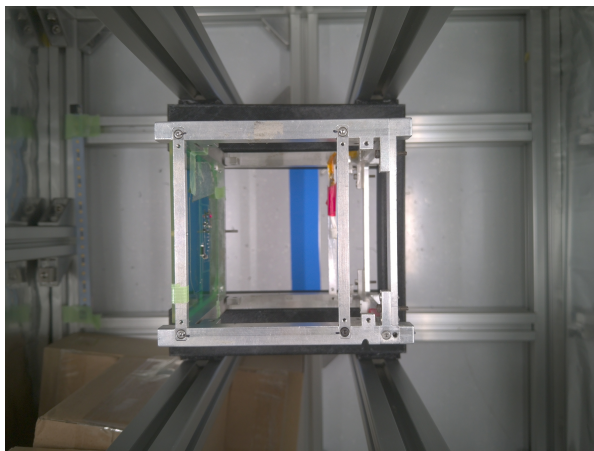


(g) Dilatación

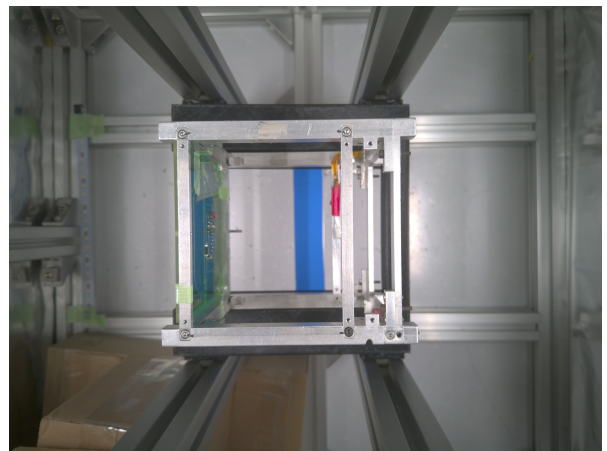


(h) Final

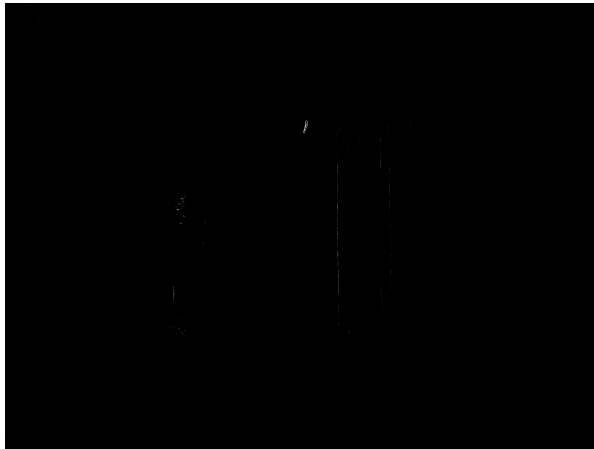
Figura 5.8: Resultados intermedios para un caso de grieta en el panel. Fuente propia.



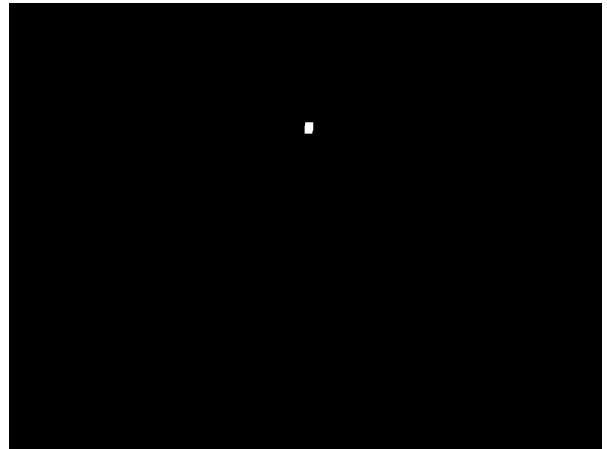
(a) Imagen pre-vibración.



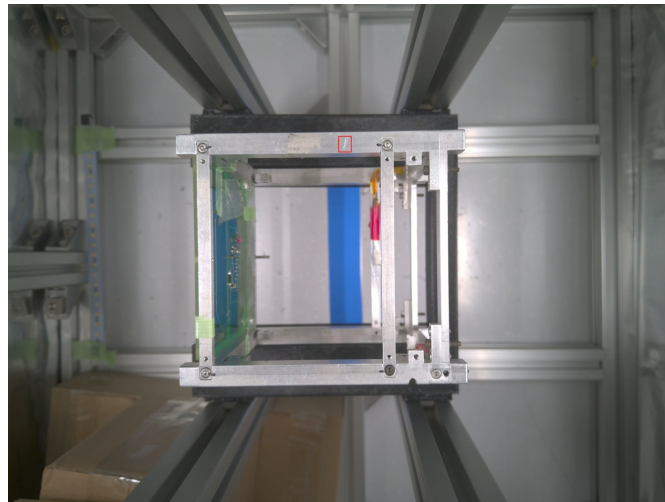
(b) Imagen post-vibración.



(c) Umbralizado



(d) Filtrado



(e) Final

Figura 5.9: Resultados intermedios para un caso de rasguño en el marco. Fuente propia.

5.3. Interfaz Gráfica

En la figura 5.10 se aprecia la ventana principal de la interfaz, como se ve al iniciar la aplicación.

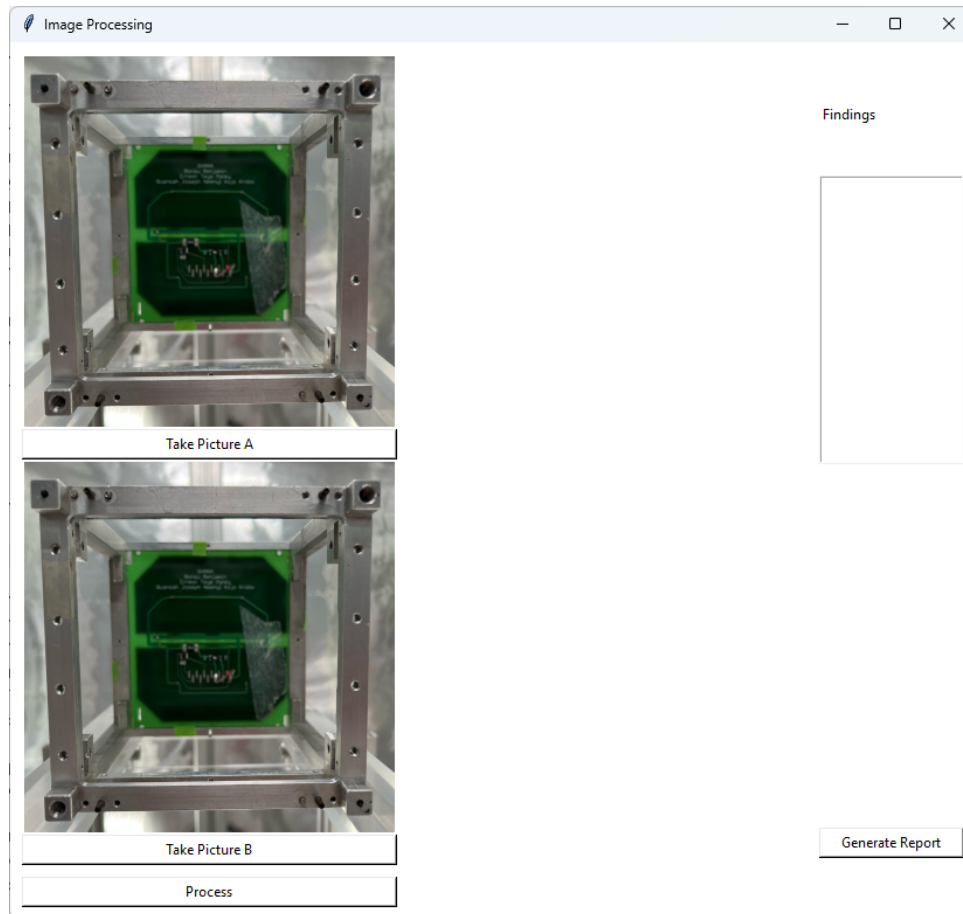


Figura 5.10: Vista inicial de la ventana principal. Fuente propia

TK Inter crea interfaces uniendo bloques básicos llamados **widgets**. Un widget es un elemento que puede tomar muchas formas, como campos de textos, imágenes, listas, botones y demás. En la ventana se aprecian cuatro widgets en la izquierda, dos que muestran transmisión de video y dos botones para toma de las fotos correspondientes a antes y después de la prueba.

Al presionar los botones estos se tornan color verde para mostrar que la toma de la imagen fue exitosa. Cuando se han tomado ambas fotografías, se procede a presionar el botón en el centro con el texto "procesar". Esto inicia el proceso de identificación de anomalías, que extra cualquier problema que exista. Al finalizar este procesamiento se muestran dos imágenes en el centro de la ventana, uno con una máscara que muestra la ubicación de las anomalías, si existen, y otra donde se ve la imagen posterior a la prueba con rectángulos superpuestos para identificar de manera más visual la ubicación de estas anomalías.

A la derecha de la ventana se encuentra una lista. Esta lista se llena con filas que co-

responden a las coordenadas y caras donde se encontraron anomalías. Al dar click a uno de estos elementos se despliega una nueva imagen abajo de la lista que corresponde a un recorte con aumento significativo de la posición de la anomalía, para que se pueda apreciar mejor la ubicación de esta.

En la figura 5.11 se aprecia la interfaz una vez concluido el proceso de identificación de anomalías. El caso del ejemplo mostrado es con una imagen alterada digitalmente, para mostrar de manera más obvia un error de manera que fuera notable en el presente documento.

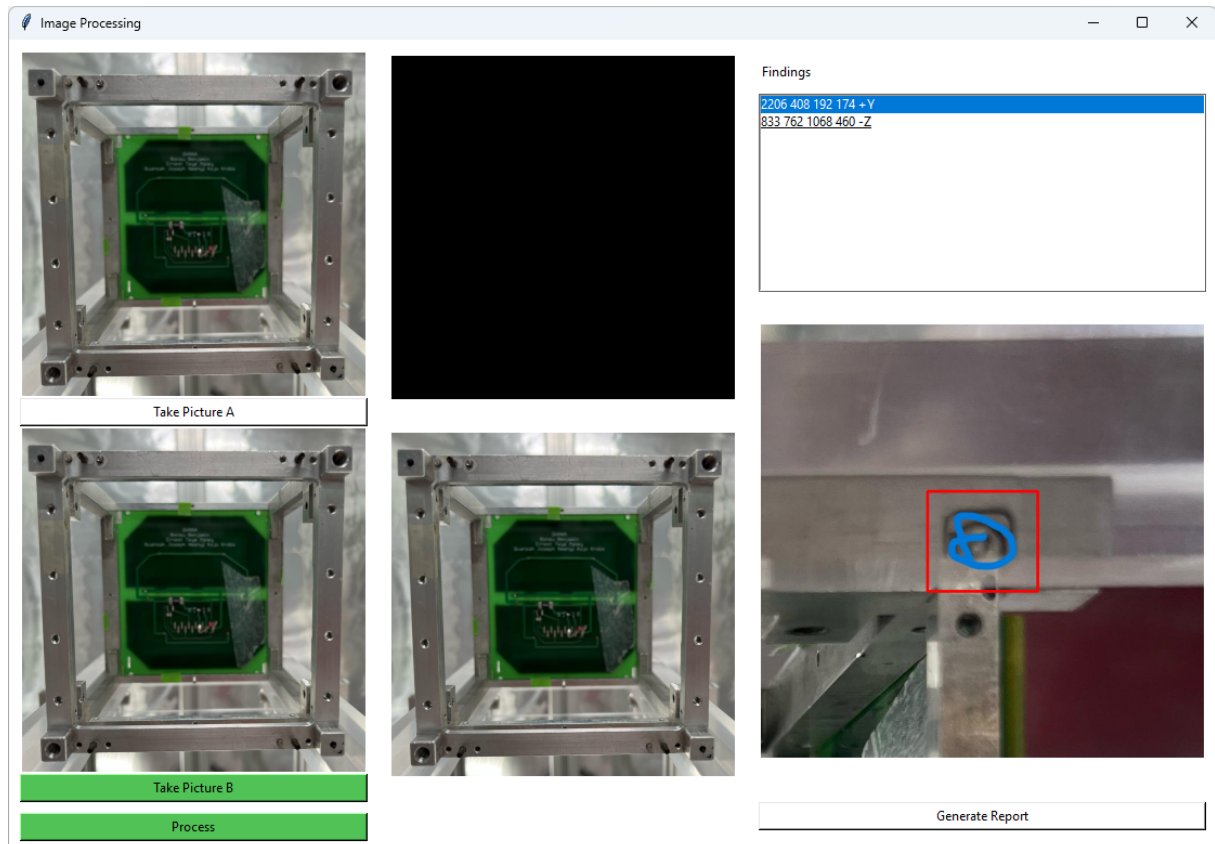


Figura 5.11: Vista final de la ventana principal. Fuente propia

Como se mencionó en la sección 4.3.6, se deben generar dos tipos reportes dependiendo de si se encontraron o no anomalías. El software identifica automáticamente cuál es el caso, y al presionar el botón correspondiente a "Generar Reporte" se despliega una pequeña ventana que solicita información al usuario, específicamente el nombre del encargado, del satélite y del proyecto. Esta ventana se aprecia en la figura 5.12. El texto del botón cambia dependiendo de si es un reporte de fallos o un reporte de certificado.

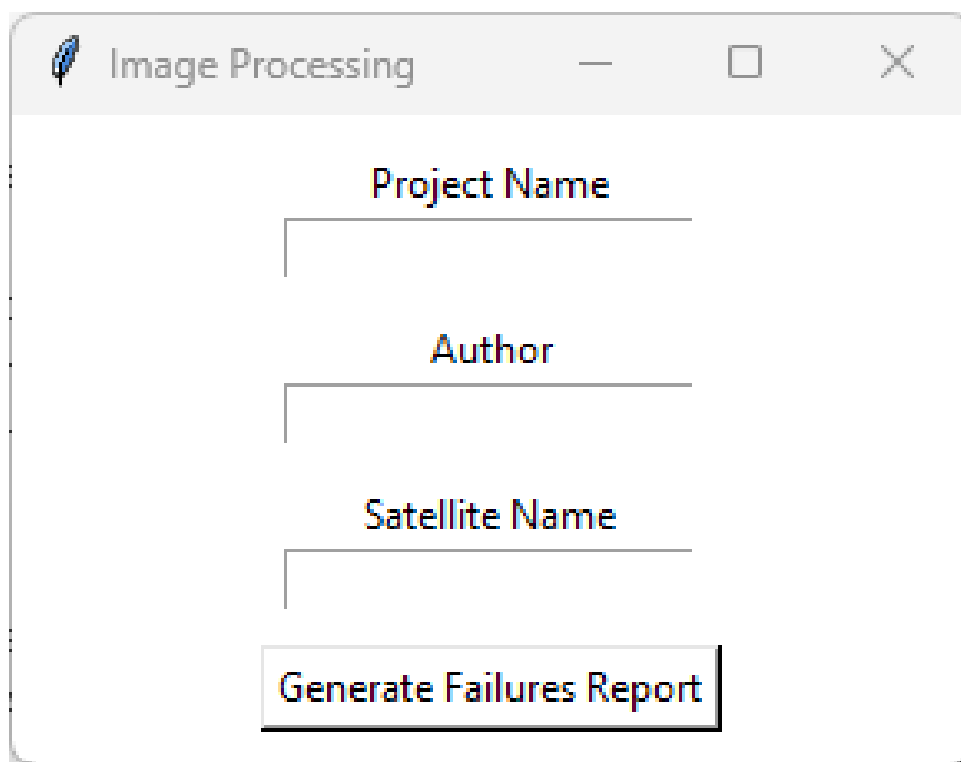


Figura 5.12: Vista de la ventana de generación de reporte. Fuente propia

En el capítulo 3 se descartó utilizar paradigmas de inteligencia artificial por la falta de datasets para poder entrenarlos. Por esto, como un objetivo secundario no crítico del proyecto, todas las imágenes de antes y después de la prueba de vibración, así como los pasos intermedios y las anomalías encontradas, son archivados con etiquetas descriptivas. Esto con el fin de iniciar la creación pasiva de un dataset que habilite el desarrollo de futuros proyectos con paradigmas de inteligencia artificial que requieran fotos claras y cercanas de caras de cubesats.

5.4. Generación de reportes

La generación del reporte como tal se realizó usando el módulo de Python `python-docx`. Este permite abrir un documento `.docx` y trabajar a nivel de párrafos y tablas. Word divide un documento `.docx` en párrafos, y estos a su vez están divididos en corridas. Una corrida es una secuencia de caracteres divididos por formato o por un símbolo especial. Si se edita un párrafo completo, se pierde cualquier formato específico que variaba entre corridas.

Para editar la plantilla contó el número de párrafo de cada párrafo que contenía información que debía ser editada por el software. Una vez identificado el párrafo, se separaba en sus corridas y se editaba la información de manera correspondiente, y finalmente se reconstruye el párrafo.

Algunas de las imágenes insertadas son concatenaciones de dos imágenes obtenidas durante el proceso de identificación de anomalías. Se tuvo que programar la funcionalidad de concatenación de imágenes para luego insertarlas en el documento .docx. La inserción de imágenes fue la etapa más difícil de la generación de reportes, debido a cómo python-docx manipula imágenes. No es posible reemplazar una imagen, debe borrarse una corrida y luego insertar una imagen nueva, creando así una corrida nueva.

El resultado de los reportes, tanto el reporte de fallas como el reporte de certificados fue un éxito, y ejemplos de estos reportes se encuentran en la sección de anexos.

5.5. Estrategia de validación

La validación del funcionamiento del detector de anomalías es un desafío interesante, ya que no es posiblemente simplemente buscar 50 satélites y dañarlos para notar un antes y un después diferente. Por esto para todas las pruebas se ha utilizado una estructura extra y un panel solar extra que sobraron de una misión antigua.

El funcionamiento del detector de anomalías es sencillo. Este debe detectar si hay diferencias en el antes y en el después de la vibración, y para esto, se puede reemplazar el satélite real con los extras que se tienen, y la vibración con daños manuales que reproduzcan daños iguales a los posibles durante vibración o manipulación. Debe detectar cambios si existen, y si no existen, no debe arrojar falsos negativos.

La determinación de un porcentaje de error se decidió que no es posible realizarse para este proyecto. Los operadores de la máquina de pruebas son los mismos miembros del equipo del satélite, y por la naturaleza de los proyectos de esta universidad, siempre es la primera vez que llevan a cabo las pruebas. No se tiene personal con larga experiencia aparte de un profesor, por lo que no se puede tener una muestra estadística de un operario que se pueda tomar como referencia perfecta para comparar el detector automático. Fuera de esto, el objetivo es que el sistema automatizado encuentre todos los errores existentes en la estructura.

Por eso, se creó un conjunto de pruebas de tornillos flojos, de rasguños y de grietas. Si el detector logra identificar todas las pruebas correctamente, se considera exitoso. Este criterio fue aceptado por el profesor encargado del laboratorio de pruebas ambientales y por el profesor supervisor del proyecto.

Lo que sí es posible cuantificar es el tiempo que se tarda haciendo las inspecciones. Como un requerimiento es que las pruebas sean sencillas de realizar, se considera que está bien utilizar un operario principiante. Para esto, se le solicitará que tome fotografías del satélite, se le entregue al conductor de la prueba quien procederá a causar un daño, se le regresa al operario y este debe tomar el segundo juego de fotografías, identificar el error y llenar el informe. El sistema automático debe vencer el tiempo que le toma al operario hacer esto. Si lo vence, se considera que el proyecto funciona. En las figuras 5.13, 5.14 y 5.15 se aprecian las anomalías insertadas que los participantes debían encontrar.



(a) Imagen pre-vibración.



(b) Imagen post-vibración.

Figura 5.13: Tornillo flojo en la cara -Y. Fuente propia.



(a) Imagen pre-vibración.



(b) Imagen post-vibración.

Figura 5.14: Tornillo flojo en la cara +Y. Fuente propia.

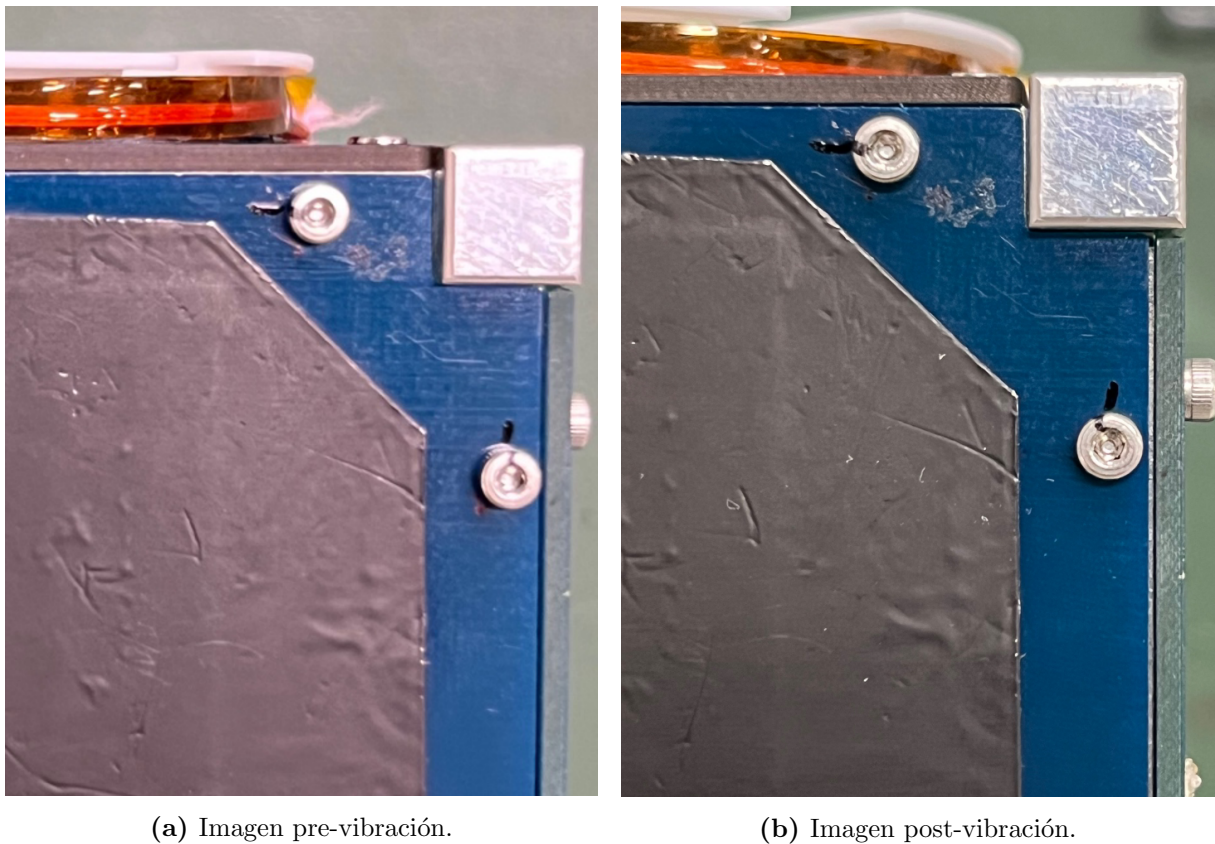


Figura 5.15: Tornillo flojo en la cara +Z. Fuente propia.

Inicialmente, si todas las pruebas de validación planteadas anteriormente eran exitosas, se iba a utilizar el modelo de vuelo del satélite Dragonfly, el cuál será sometido a sus pruebas finales de vibración a inicios de febrero del 2024. Además de ser sometido a las pruebas tradicionales, iba a ser el primer satélite real que será sometido ante el detector automático. Lamentablemente, el proyecto del satélite tuvo que mover las fechas de sus pruebas de vibración para el modelo de vuelo fuera de lo planificado. Por esto, solo se utilizará el modelo de ingeniería, para el cuál no se escribió un reporte de vibraciones cuando este fue sometido a la prueba. Sin embargo, tiene las mismas geometrías del modelo de vuelo.

5.6. Resultados de validación

Las pruebas específicamente consistían en pedirle a un operario voluntario que hiciera una inspección minuciosa de cada uno de los 68 tornillos y otros elementos estructurales y electrónicos del satélite Dragonfly, y notaran cuáles eran los errores presentes. A este satélite se le aflojaron ligeramente 3 tornillos, los mismos para cada participante.

Una vez que finalizaron la inspección visual, deben redactar el informe de certificación señalando en las imágenes los problemas encontrados. Se anota el tiempo que se tardó

en la inspección visual y en la redacción del documento. En la figura 5.16 se aprecia al operador Javier Ferrer llevando a cabo la inspección visual inicial de esta prueba.



Figura 5.16: Operador realizando la prueba de validación planteada. Fuente propia

Posteriormente, se les piden que repitan la prueba esta vez utilizando la máquina desarrollada en el presente proyecto. Se mide el tiempo que tardan usándola. Los resultados de la prueba se aprecian en la tabla 5.1.

Tabla 5.1: Resultados de validación de tiempo ahorrado. Fuente propia.

N	Nombre de Operador	Tiempo de inspección [mm'ss"]	Tiempo de documentación [mm'ss"]	Tiempo total [mm'ss"]	Errores reales detectados (de 3)	Errores inexistentes encontrados	Tiempo con sistema automatizado [mm'ss"]	Mejora
1	Jorge	14'10"	12'49"	26'59"	3	1	00'38"	4161 %
2	Nastya	9'49"	25'48"	35'37"	3	8	01'04"	3239 %
3	Yudai	11'37"	11'37"	23'13"	3	1	00'48"	2802 %
4	Sirash	8'27"	28'32"	37'00"	3	0	00'43"	5063 %
5	Altea	9'34"	20'04"	29'39"	3	6	01'00"	2865 %
6	Zamba	09'55"	36'02"	44'58"	1	1	00'50"	5296 %
7	Javier	16'53"	28'52"	45'45"	2	2	00'53"	5079 %

En la última columna se aprecia cómo la mejora va desde un 2802 % hasta un 5296 %.

En cuanto a la eficacia del sistema para identificar las anomalías, efectivamente identificó las 3 anomalías insertadas, además de todas las anomalías presentes en un conjunto de imágenes de prueba de otro satélite. Algunas de estas se mostraron en la sección 5.2.

Un punto de discusión importante es la columna de errores inexistentes encontrados. Estos son errores que no estaban presentes en el satélite, pero fueron equivocadamente identificados como anomalías por los operadores en el reporte que escribieron. Al ser un satélite didáctico, el reporte primero pasaría por algún profesor de la facultad que revisaría y descartaría estos falsos negativos. Sin embargo, el sistema automatizado no identificó ninguna anomalía que no fue insertada, ni con el satélite Dragonfly ni con los otros conjuntos de imágenes de pruebas. Esto muestra cómo la solución desarrollada es más fiable que operadores principiantes.

5.7. Análisis económico

Este proyecto no es de índole económica, en el sentido que se espera producir en masa. Si bien es cierto que busca agilizar el proceso y ahorrar tiempo, los profesores no mostraron ningún interés en ningún momento sobre potenciales ahorros económicos que podría traer. No necesitan agilizar el proceso para que sus costos disminuyan. Simplemente buscan aumentar capacidad de pruebas, sin importar los costos de la máquina. Esto porque, al menos actualmente, el proyecto del detector de anomalías y los proyectos de satélites que se desarrollan en la universidad son didácticos. El presupuesto no fue un factor ni si quiera a la hora de la selección de componentes.

Dicho esto, aún así describirán los costos asociados al desarrollo del prototipo en cuanto a componentes y tiempo de trabajo por parte del estudiante que ejecutó el proyecto. Los materiales utilizados y sus costos aproximados (por tipo de cambio de Yen a Colones) se observan en la tabla 5.2.

Tabla 5.2: Componentes y costos aproximados. Fuente propia.

Componente	Cantidad	Precio Individual (¢)	Precio Total (¢)
Jetson Orin Nano	2	235400	470800
Kit Cámaras	2	99900	199800
Cables Extra	2	8800	17600
Kits Iluminación	2	92000	184000
Monitor	1	80000	80000
Componentes Mecánicos	1	200000	200000
Tiempo Ingeniero	240	18500	4400000
		Total	5552200

Esto sin embargo no toma en cuenta el tiempo de trabajo. El estudiante trabajó 20 horas semanales durante 3 meses en este proyecto. Esto se traduce a una cantidad de horas de 240. De los datos del Colegio Federado de Ingenieros y Arquitectos, la hora profesional de un ingeniero es de 37000 colones [32]. Considerando que el estudiante no tiene título universitario, se tomará la mitad de este valor, para un total de 18500 colones la hora profesional. A lo largo de 240 horas, esto suma 4 440 000 colones.

Se insiste en que este dato no es muy significativo. Si bien el proyecto tiene un potencial futuro donde le brinde ahorros al laboratorio, se planteó meramente didáctico. La universidad proveyó al estudiante con espacio para hospedaje además de acceso a equipo especializado de manera gratuita, que no fue contemplado en el desglose anterior, como estructuras y paneles solares antiguos. Por todo esto, y concordando con la opinión del profesor supervisor del proyecto, se considera que los gastos estuvieron justificados.

Un factor difícil de calcular económicamente, pero relevante a esta sección, es el tiempo ahorrado por los operadores. Por lo general, la fase de pruebas ambientales de un satélite es el tiempo más crítico del ciclo de vida de este tipo de proyectos en Kyutech. Los estudiantes por lo general abandonan sus investigaciones personales y hasta sus clases para finalizar esta fase, ya que un atraso puede significar pérdidas económicas grandes para el departamento. Los operadores son estudiantes becados, y la beca que reciben en este laboratorio es de 144 mil yenes japoneses al mes. Se podría tomar este valor y dividirlo entre las horas de trabajo, pero la realidad del laboratorio es que no todas las horas valen lo mismo, y la cantidad de estas varía significativamente dependiendo del periodo. Por esto, ponerle un monto económico exacto al ahorro que trae esta disminución del tiempo que toman las pruebas es sumamente difícil, pero es definitivamente significativo.

El tiempo ahorrado sí implica beneficios no tangibles. Al reducir significativamente el tiempo de inspección visual, se elimina este cuello de botella. Esto permite la inspección de constelaciones de satélites en tiempos realistas. Además, se reduce el tiempo de espera, lo que permite que otros posibles proyectos se terminen antes. Esto lleva a mayor capacidad de proyectos, y eventualmente, mayor cantidad de publicaciones del laboratorio.

Capítulo 6

Conclusiones y trabajos futuros

6.1. Conclusiones

El proyecto consiste en el desarrollo de un identificador de anomalías autónomo para pruebas de vibración en cubesats, el cual se diseña siguiendo la metodología propuesta por Ulrich [25], donde primero se identifican las necesidades dadas por los deseos del cliente, se generan métricas para cumplir con estas necesidades y se generan diferentes conceptos hasta generar un concepto ganador. Para las conclusiones se incluyen los valores generados a partir de metodología y las validaciones. Todas las conclusiones del proyecto se citan a continuación:

- Se diseñó un sistema mecatrónico que puede realizar inspección visual e identificación de anomalías de manera autónoma en las 6 caras de un satélite.
- Se diseñó un sistema de visión por computadora que identifica con eficacia total errores de tipo rasguños en la estructura, grietas en los paneles solares y errores de marcas de torque en los tornillos.
- Se diseñó una estructura mecánica que soporta todos los componentes permitiendo acceder visualmente a todo el satélite simultáneamente, y soporta satélites de tamaños de 1U y 2U.
- Se construyó un sistema de iluminación aislado del exterior, lo que mantiene condiciones de iluminación constantes entre pruebas.
- Se diseñó un sistema de software de generación de reportes, que toma los datos obtenidos con el identificador de anomalías y genera reportes de certificación de prueba de vibración si fue exitoso, o de fallos si se encontraron anomalías.
- Se integraron todos los segmentos de software con una interfaz gráfica para el usuario que notablemente fue recibida muy positivamente por los usuarios, que es compatible con Windows, MacOS y Ubuntu, tanto en sistemas x86 como ARM64.

- Se validó el sistema de identificación de anomalías utilizando datos limpios como también datos alterados artificialmente, y se obtuvo una capacidad de identificación sin errores.
- Se diseñó todo el proyecto siguiendo minuciosamente la metodología planteada por Ulrich-Eppinger, tomando en cuenta en todo momento las necesidades planteadas por el cliente.
- Se analizan los costos del proyecto, tanto en compras de componentes como en tiempo de trabajo, en relación con la mejora calculada de hasta un 5296 % en tiempo que se toma en realizar la inspección visual posterior a las pruebas de vibración.

6.2. Trabajos futuros

Debido al enorme alcance que tendría el proyecto para abordar todo lo solicitado por el cliente, se excluyeron tres partes importantes para enfocarse así en la inspección visual superficial. A continuación se resaltan cuales son estas tres partes.

- El análisis de resultados de la prueba de vibración consiste en la inspección visual, que fue lo abordado por el proyecto, como también el análisis de respuesta a frecuencia. Esta parte consiste en la toma de datos dados por la máquina de vibración como tal que mide la respuesta a frecuencia del satélite ante un impulso unitario, antes y después de la vibración principal. Esto permite identificar si algún componente interno del satélite está suelto. El análisis de la respuesta a frecuencia es un análisis numérico que no tiene ningún traslape con el trabajo de visión por computadora que es el enfoque del presente proyecto. Ahora que se tienen las bases del generador de reportes, un futuro trabajo de análisis de respuesta a frecuencia tendría parte de su alcance terminado. Quedaría pendiente la interfaz con la máquina de vibración, cálculos y generación de gráficos.
- Actualmente el laboratorio tiene todos los equipos de pruebas ambientales integrados con el software LabView, que es mayoritariamente un sistema de programación gráfica con bloques. Si bien esto facilita el desarrollo de aplicaciones simples que utilicen operaciones soportadas por labview, otras operaciones complejas como visión por computadora no son soportadas. Para poder integrar completamente el sistema desarrollado, en un futuro deberá escribirse un código envoltorio que funcione como interfaz con el sistema de bloques de LabView, para así integrar datos como el análisis de respuesta a frecuencia que actualmente se lleva a cabo en el programa de LabView.
- Uno de los principales desafíos a lo largo del proyecto para alcanzar el nivel de eficacia obtenido fue la variación en la posición del satélite cuando se toman las fotos antes y después de la prueba de vibración. Se tomaron múltiples medidas en

las diferentes versiones del soporte del satélite, sin embargo, en la construcción final del soporte de acrílico las tolerancias obtenidas no correspondieron con las solicitadas al taller de manufactura. Se desconoce la causa de esto, pero se asume que es por problemas al usar el material relativamente exótico que es acrílico transparente con maquinaria de precisión de metales. El Comité de Estandarización de CubeSats se enteró del proyecto, y se comunicó con el profesor supervisor, sugiriendo utilizar marca fiduciarias en las placas de circuitos impresos de los paneles externos, para posteriormente utilizar visión por computadora y alinear digitalmente el satélite. Esto mitigaría potenciales problemas de alineación entre pruebas. Se recalca que no se observaron problemas por alineación. Sin embargo, si en un futuro se cambian las cámaras a unas con mejor resolución y menor ruido, una mejor alineación permitiría identificar anomalías sumamente pequeñas comparadas con las actuales. Por otro lado, los expertos de la universidad en pruebas ambientales no esperan necesitar mejor resolución que la que se posee actualmente.

Bibliografía

- [1] J. P. Laboratory. «Build a Satellite.» (2019), dirección: <https://www.jpl.nasa.gov/edu/teach/activity/build-a-satellite>.
- [2] S. Insights. «Top 10 Satellite Trends, Technologies and Innovations for 2023.» (2023), dirección: <https://www.startus-insights.com/innovators-guide/satellite-trends-innovation/>.
- [3] S. R. Hirshorn, *NASA Systems Engineering Handbook*, 2da. ed. Washington, DC: NASA, 2020.
- [4] N. T. Systems. «What is satellite testing?» (2022), dirección: <https://nts.com/ntsblog/what-is-satellite-testing/>.
- [5] E. Analytics. «Types of Satellites.» (), dirección: <https://eos.com/blog/types-of-satellites/>.
- [6] A. Johnstone, *CubeSat Design Specification*, 14.1. California: Calpoly University, 2022.
- [7] NASA. «What are smallsats and cubesats?» (), dirección: <https://www.nasa.gov/content/what-are-smallsats-and-cubesats>.
- [8] R. Bedington, X. Bai, E. Truong-Cao et al., «Nanosatellite experiments to enable future space-based QKD mission,» *EPJ Quantum Technology*, 2016.
- [9] A. Space. «10 Advantages of CubeSats vs. Conventional Satellites.» (2019), dirección: <https://info.alen.space/advantages-of-cubesats-vs-conventional-satellites>.
- [10] M. Viens, *General Environmental Verification Standard (GEVS) for GSFC Flight Programs and Projects*, 3ra. ed. Washington, DC: NASA, 2021.
- [11] IBM. «What is computer vision?» (), dirección: <https://www.ibm.com/topics/computer-vision>.
- [12] Microsoft. «What is computer vision?» (), dirección: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-computer-vision/>.
- [13] R. Szeliski, *Computer Vision: Algorithms and Applications*, 1ra. ed. Springer, 2011.
- [14] A. Rosebrock, *Deep Learning for Computer Vision*, 2da. ed. 2021.

- [15] P. e. a. Bergman, «Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders,» 2019.
- [16] OpenCV. «Image Thresholding.» (), dirección: https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html.
- [17] OpenCV. «Morphological Operations.» (), dirección: https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html.
- [18] L. Haidar. «Sobel vs. Canny Edge Detection Techniques: Step by Step Implementation.» (), dirección: <https://medium.com/@haidarlina4/sobel-vs-canny-edge-detection-techniques-step-by-step-implementation-11ae6103a56a>.
- [19] Z. Wang, A. Bovik, H. Sheikh y E. Simoncelli, «Image quality assessment: from error visibility to structural similarity,» *IEEE Transactions on Image Processing*, 2004.
- [20] C. Imaging. «Choosing a Camera for Embedded Vision.» (2019), dirección: <https://www.clearview-imaging.com/en/blog/choosing-a-camera-for-embedded-vision>.
- [21] D. Zafra. «Camera Sensor Size in Photography.» (2020), dirección: <https://capturetheatlas.com/camera-sensor-size/>.
- [22] Masterclass. «Photography 101: Understanding Camera Lenses Basics.» (2021), dirección: <https://www.masterclass.com/articles/basic-photography-101-understanding-camera-lenses>.
- [23] O. Engineering. «Optics basics.» (2021), dirección: <https://www.opto-e.com/en/basics/optics-basics>.
- [24] I. Engineering. «Image Quality Factors: Noise.» (2021), dirección: <https://www.image-engineering.de/library/image-quality/factors/1080-noise>.
- [25] K. T. Ulrich, *Diseño y desarrollo de productos*, 5a. ed. México: McGrawHill, 2013.
- [26] J. Crespo, «Módulo 1. Sesión 2: “Fase 0” (o de problema percibido) y determinación de necesidades,» 2022.
- [27] Arducam. «16MP IMX519 Autofocus Synchronized Quad-Camera Kit for Raspberry Pi, NVIDIA® Jetson Nano/Xavier NX/AGX Orin/Orin Nano/Orin NX.» (), dirección: <https://www.arducam.com/product/arducam-16mp-imx519-autofocus-synchronized-quad-camera-kit-for-raspberry-pi-nvidia-jetson-nano-xavier-nx/>.
- [28] NVidia. «Jetson Orin Nano Developer Kit User Guide.» (), dirección: <https://developer.nvidia.com/embedded/learn/jetson-orin-nano-devkit-user-guide/index.html>.
- [29] D. Inspire LED. «PRO Series General Purpose Lighting Kit White, Neutral 4200K 12VDC 2 A.» (), dirección: <https://www.digikey.jp/en/products/detail/inspired-led-llc/3583/5977973>.

-
- [30] GStreamer. «Official GStreamer Command Line Documentation.» (), dirección: <https://gstreamer.freedesktop.org/documentation/tools/gst-launch.html?gi-language=python>.
- [31] ArduCAM. «CamArray Getting Started Guide.» (), dirección: <https://docs.arducam.com/Raspberry-Pi-Camera/Multi-Camera-CamArray/quick-start/#switch-composition-mode>.
- [32] Colegio Federado de Ingenieros y de Arquitectos de Costa Rica. «Salarios mínimos (Primer semestre 2023).» (2023), dirección: <https://cfia.or.cr/descargas/2020/dep/hora-profesional.pdf>.

Apéndice A

Ejemplo de reporte de éxito
generado por la aplicación

Kyushu Institute of Technology
Department of Applied Science for Integrated System Engineering



BIRDS-X Project Vibration Test Report

Initial Release: 2024/02/29

Revision History

Date	Version Number	Writer	Annotation
2024/02/29	NC	Jairo Rodríguez	Initial Release

Contents

1. Purpose.....	3
2. Applicable Document	3
3. Test method	3
3.1. Test Objectives.....	3
3.2. Acceleration measurement point.....	3
3.5. Test Tolerance	6
3.6. Test Sequence.....	6
4. Test Results.....	6
4.1. Visual inspection	7
4.2. Random Vibration Test	9
4.2.1. X-axis vibration	Error! Bookmark not defined.
4.2.2. Z-axis vibration.....	11
4.2.3. Y-axis vibration	12
4.3. Function Test.....	14
5. Conclusion	15

1. Purpose

This document summarizes the results of the Vibration Test for BIRDS-X which will be deployed from JEM Small Satellites Orbital Deployer-R (J-SSOD-R).

2. Applicable Document

- | | |
|---------------------------|---|
| (1) JX-ESPC-101133-E | JEM Payload Accommodation Handbook-Vol.8-
Small Satellite Deployment Interface Control Document |
| (2) JMX-2011303
JAXA | Structure Verification and Fracture Control Plan for
Selected Small Satellite Released from J-SSOD |
| (3) 15_BIRDS-X-FCE-02(Φ3) | Structure Fracture Control Evaluation Form for
Small Satellite deployed from J-SSOD |
| (4) 16_BIRDS-X-SAR-02(Φ3) | BIRDS-X Flight Safety Assessment Report
for phase III |

3. Test method

The verification points are as follows,

- (1) No breakage in main structure
- (2) Main structure needs to satisfy specified natural frequency
- (3) Natural frequency before and after tests need to remain unchanged
- (4) No improper antenna deployment, and no malfunction to CubeSat
- (5) No breakage in glass material
- (6) No loosening in all fasteners

3.1. Test Objectives

Table 3.1-1 Test Articles

	Article name	Number	Manufacturer
1	PearlAfricaSat-1	1	HMD
4	2U Pod	1	

3.2. Acceleration measurement point

The acceleration measurement points are shown in Figures 3.2-1 to 3.2-6 and Table 3.2-1. “Ch.○” are acceleration measurement points for data acquisition. “Control point ○ (CP○)” are acceleration measurement points for average value control for the vibration machine.

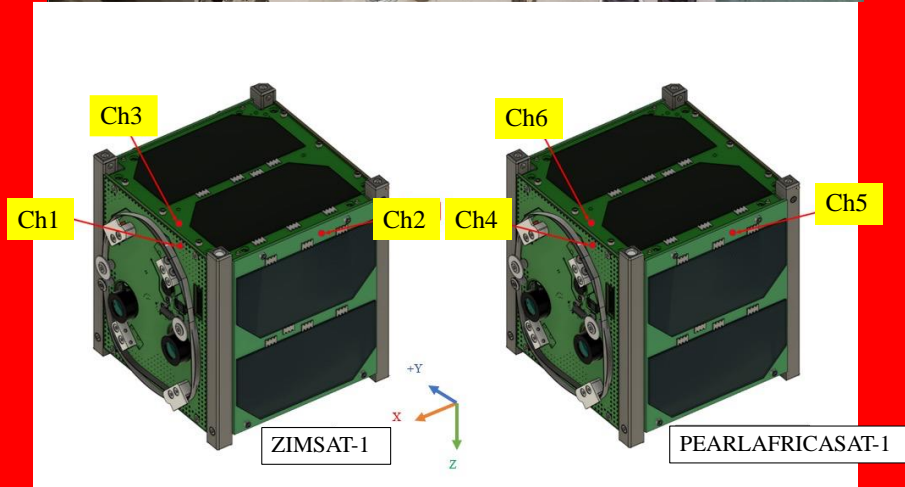
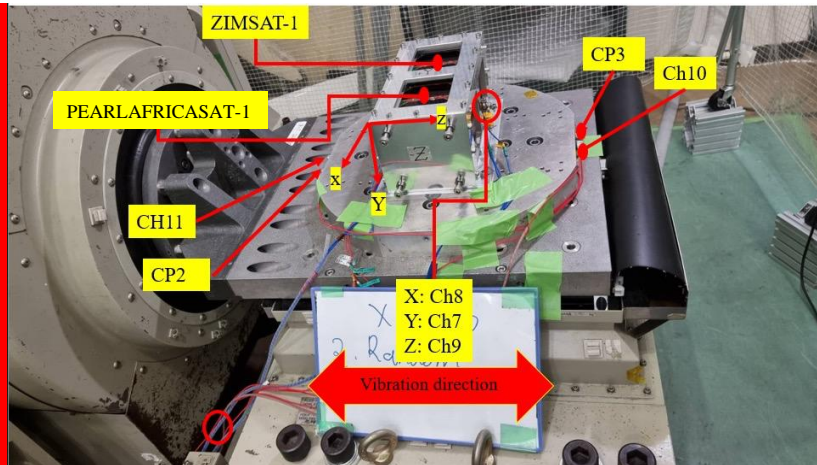


Figure 3.2-1 Acceleration measurement point (X-axis) in Dragonfly-1 test.

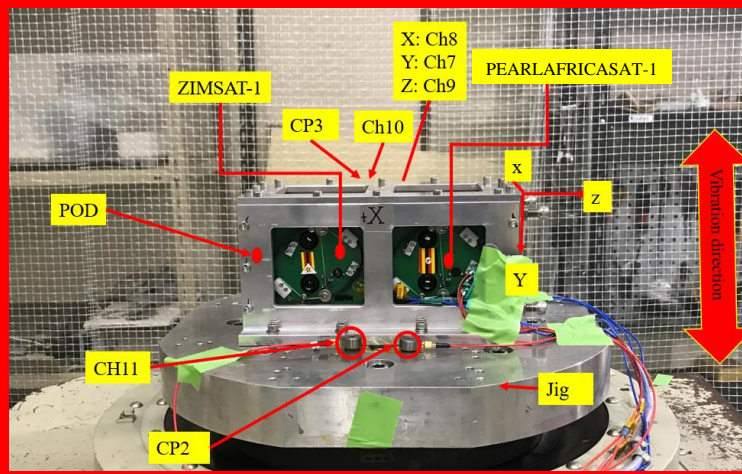


Figure 3.2-2 Acceleration measurement point (Y-axis) in Dragonfly-1 test.

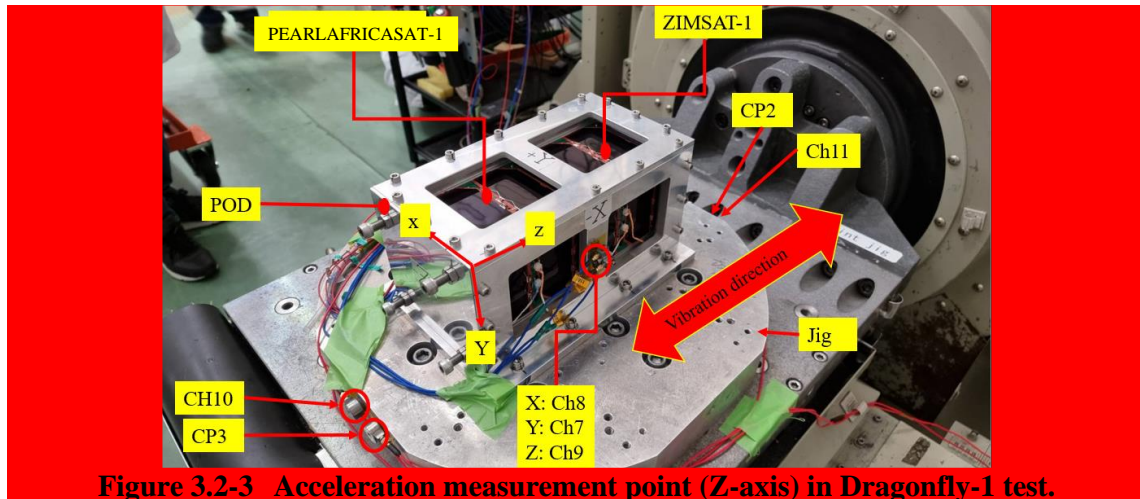


Figure 3.2-3 Acceleration measurement point (Z-axis) in Dragonfly-1 test.

Table 3.2-1 Acceleration measurement point

a) Dragonfly-1 test

Measurement point	Measurement axis	Acceleration sensor	CH	Note
Control Sensor	1-axis	EM15	CP2	
	1-axis	EM13	CP3	
Jig	1-axis	EM16	Ch.10	
	1-axis	EM17	Ch.11	
Dragonfly-1	X	tt04	Ch.1	
	Y	tt05	Ch.2	
	Z	tt06	Ch.3	
Pod	X	tt02	Ch.7	
	Y	tt01	Ch.8	
	Z	tt03	Ch.9	

3.3. Test Contents

Following vibration tests were performed along X, Z and Y axes, respectively.

- (1) Low level white noise vibration (Modal survey)
- (2) Random vibration

3.4. Vibration Test Level

- (1) Low level white noise vibration (Modal Survey)

Modal survey test parameters are described in Table 3.4-1

Table 3.4-1 Modal survey vibration environment

	Frequency [Hz]	Acceleration [Grms]	Time [sec]
Each axis	20~2000	0.5	60

- (2) Random Vibration

The random vibration test level is shown in Table 3.4-2, the level is the envelope of the environments for HTV-X, Dragon and Cygnus (reference: JX-ESPC-101132-D). This test level was defined by Structure Fracture Control Evaluation Form.

**Table 3.4-2 Random vibration test level
(Envelope of the environments for HTV-X, Dragon and Cygnus)**

Freq. [Hz]	PSD [G^2/Hz]
20	0.02
50	0.02
120	0.031
230	0.031
1000	0.0045
2000	0.0013
Overall	4.08 Grms
Duration	60 sec/axis
Direction	3 axes each

3.5. Test Tolerance

<Random Vibration Test>

- (a) Overall [Grms] : $\pm 1.5dB$
- (b) PSD [G^2/Hz] : $+3.0dB / -1.0dB$
- (c) Frequency [Hz]: larger one of $\pm 2\%$ or 1 [Hz]
- (d) Duration [sec]: $+10\% / -0\%$

3.6. Test Sequence

Table 3.6-1 shows the test sequence.

Table 3.6-1 Test Sequence

	Test Contents	Test ID
Before vibration test	Visual inspection of satellite	
Z-axis test	Satellite setup	
	Modal survey	1-modal
	Random vibration test	2-random
	Modal survey	3-modal
X-axis test	Axis change	
	Modal survey	4-modal
	Random vibration test	5-random
	Modal survey	6-modal
Y-axis test	Axis change	
	Modal survey	7-modal
	Random vibration test	8-random
	Modal survey	9-modal
After vibration test	Visual inspection of satellite	
	Function test of satellite	

4. Test Results

Date: 2024/02/29

Location : Center for Nanosatellite Testing (CeNT)
 Laboratory of Lean Satellite Enterprises and In-Orbit Experiments
 Kyushu Institute of Technology
 1-1 Sensui, Tobata, Kitakyushu, 804-8550 Fukuoka, Japan

4.1. Visual inspection

The visual inspection for the satellite was conducted before and after the test. No breakage was found in the satellites after the test. No deformation and damage were seen on the deployment switches. Also, no breakage in the solar cell cover glass. No loosening was found in all fasteners. There was also no inadvertent antenna deployment during and after the vibration. After the vibration test, the knot of the wire restraining the antenna was not loosened, and the wire was not damaged or stretched. No malfunction occurred to CubeSat, either. No scratches or cracks of any significance were found on the structural components, panels, and glass parts. Test results show in Table 4.1-1 and Figure 4.1-1 to Figure 4.1-18.

Table 4.1-1 Inspection Result

	Part Name	Quantity	Inspection Result
1	Dragonfly-1		
1-1	External panel	6	Pass
1-2	Solar panel cover glass	10	Pass
1-3	Camera lens	3	Pass
1-4	Camera filters	2	Pass
1-5	Torque mark	53	Pass
1-6	Antenna mechanism	4	Pass

a) Dragonfly-1



Figure 4.1-1 Photo on the +X surface (Left: before test Right: after test)

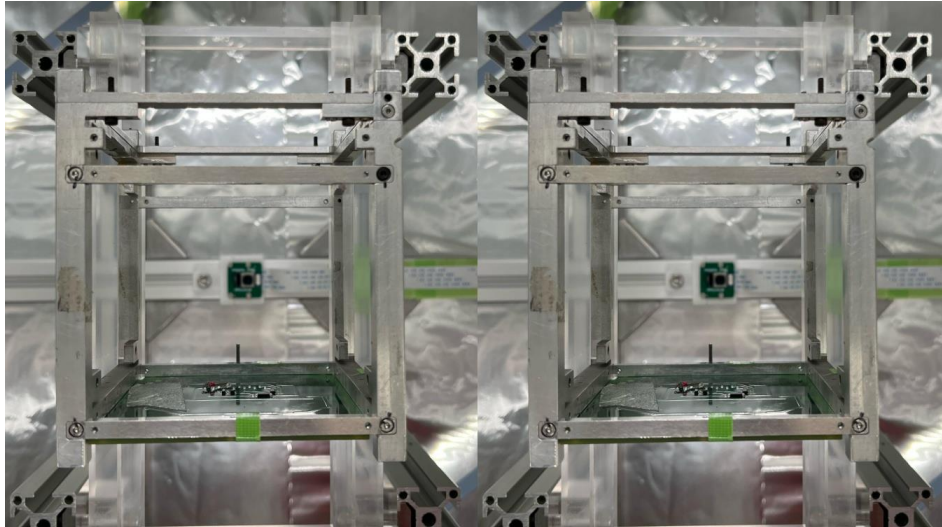


Figure 4.1-2 Photo on the -X surface (Left: before test Right: after test)

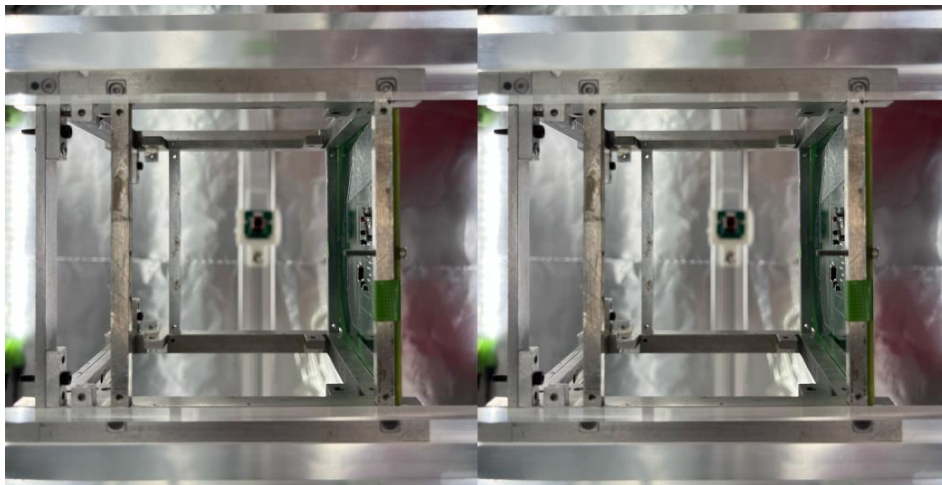


Figure 4.1-3 Photo on the +Y surface (Left: before test Right: after test)



Figure 4.1-4 Photo on the -Y surface (Left: before test Right: after test)

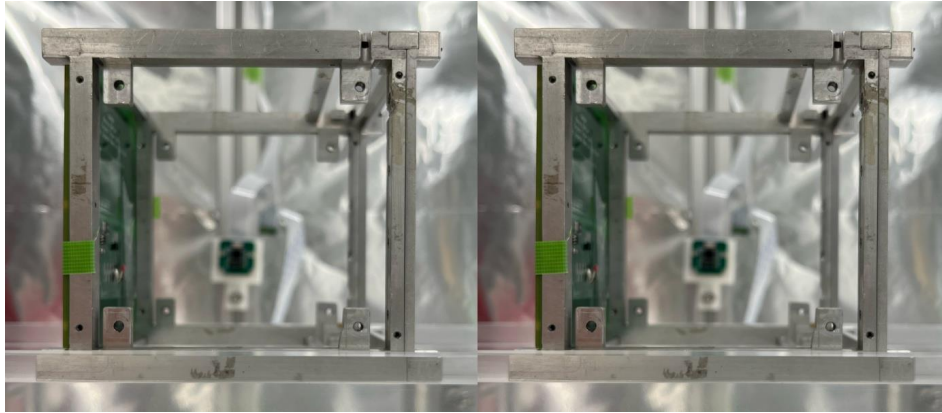


Figure 4.1-5 Photo on the +Z surface (Left: before test Right: after test)

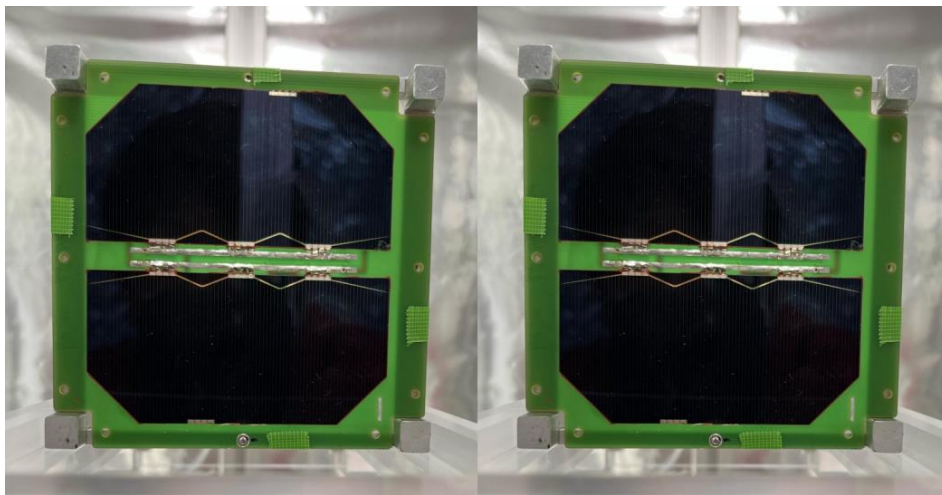


Figure 4.1-6 Photo on the -Z surface (Left: before test Right: after test)

4.2. Random Vibration Test

4.2.1. Dragonfly-1

4.2.1.1. X-axis vibration

The required acceptance level of vibration was met by the vibration machine. Test results show in Figure 4.2.1.1-1 to Figure 4.2.1.1-2.

In the modal survey before and after the random vibration, the resonant frequency was **347 Hz and 339 Hz**, respectively. There is no significant change in the minimum natural frequency before and after the random vibration. It is 60Hz or higher and meets the requirements.

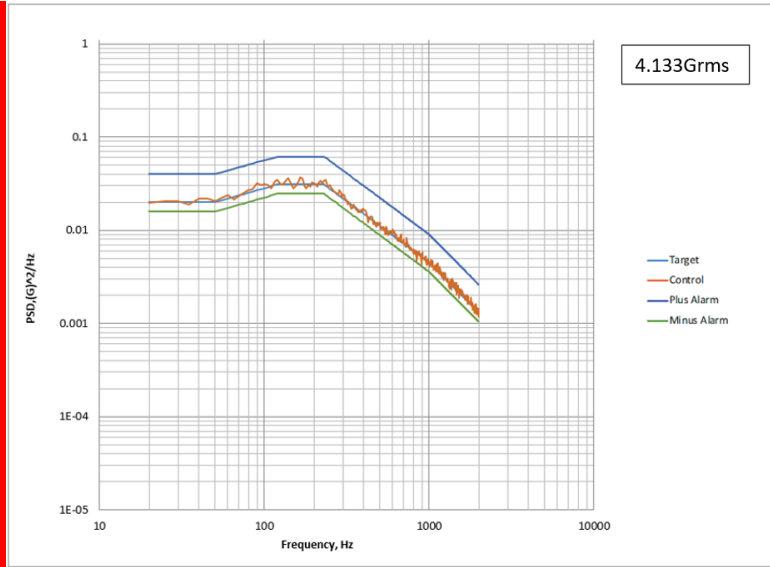
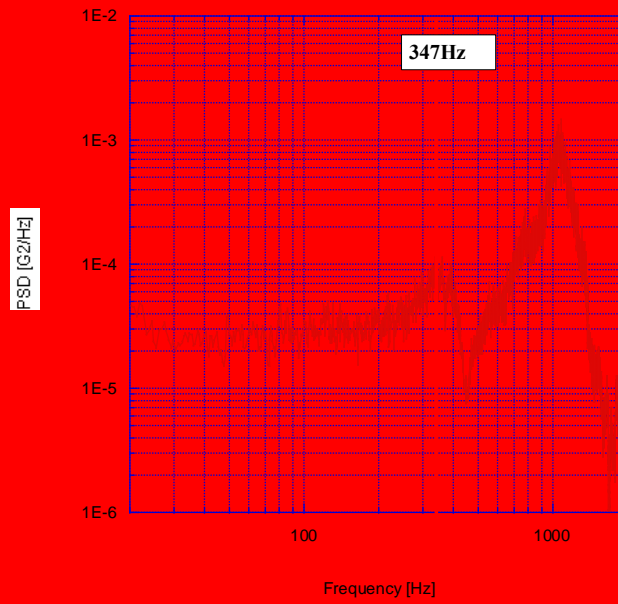


Figure 4.2.1.1-1 X-axis random vibration at Control Sensor



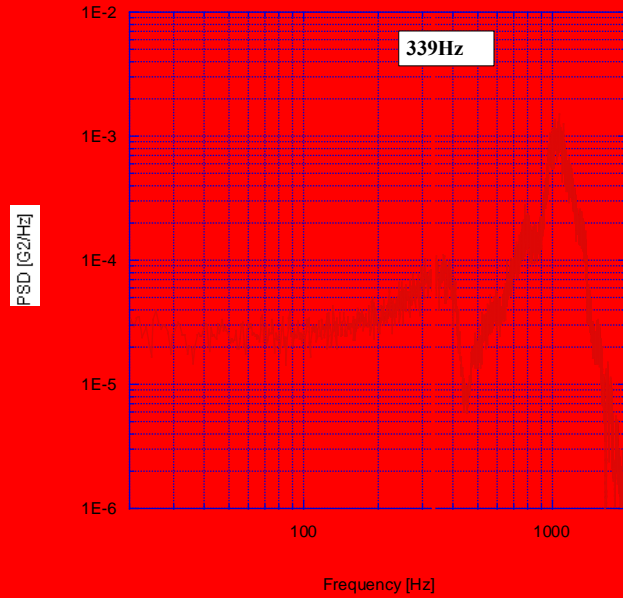


Figure 4.2.1.1-2 Modal Survey before Vibrating X-axis at Measurement Point (Ch.4)
(Left: before, Right: after)

4.2.1.2. Z-axis vibration

The required acceptance level of vibration was met by the vibration machine. Test results show in Figure 4.2.1.2-1 to Figure 4.2.1.2-2.

In the modal survey before and after the random vibration, the resonant frequency was **317 Hz and 324 Hz**, respectively. There is no significant change in the minimum natural frequency before and after the random vibration. It is 60Hz or higher and meets the requirements.

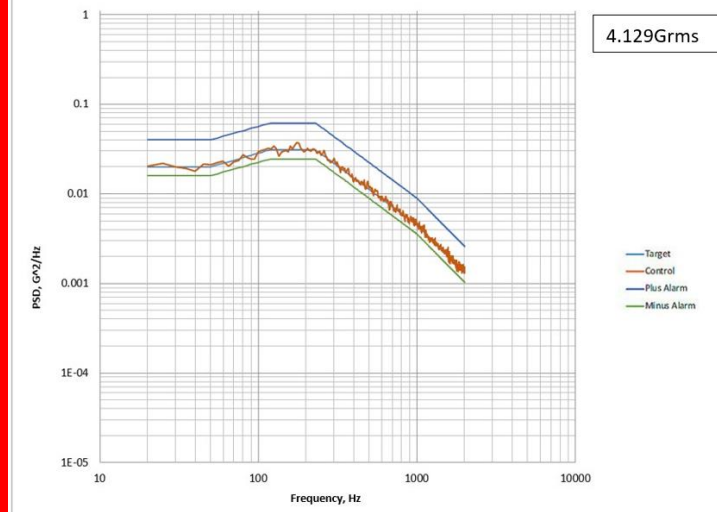
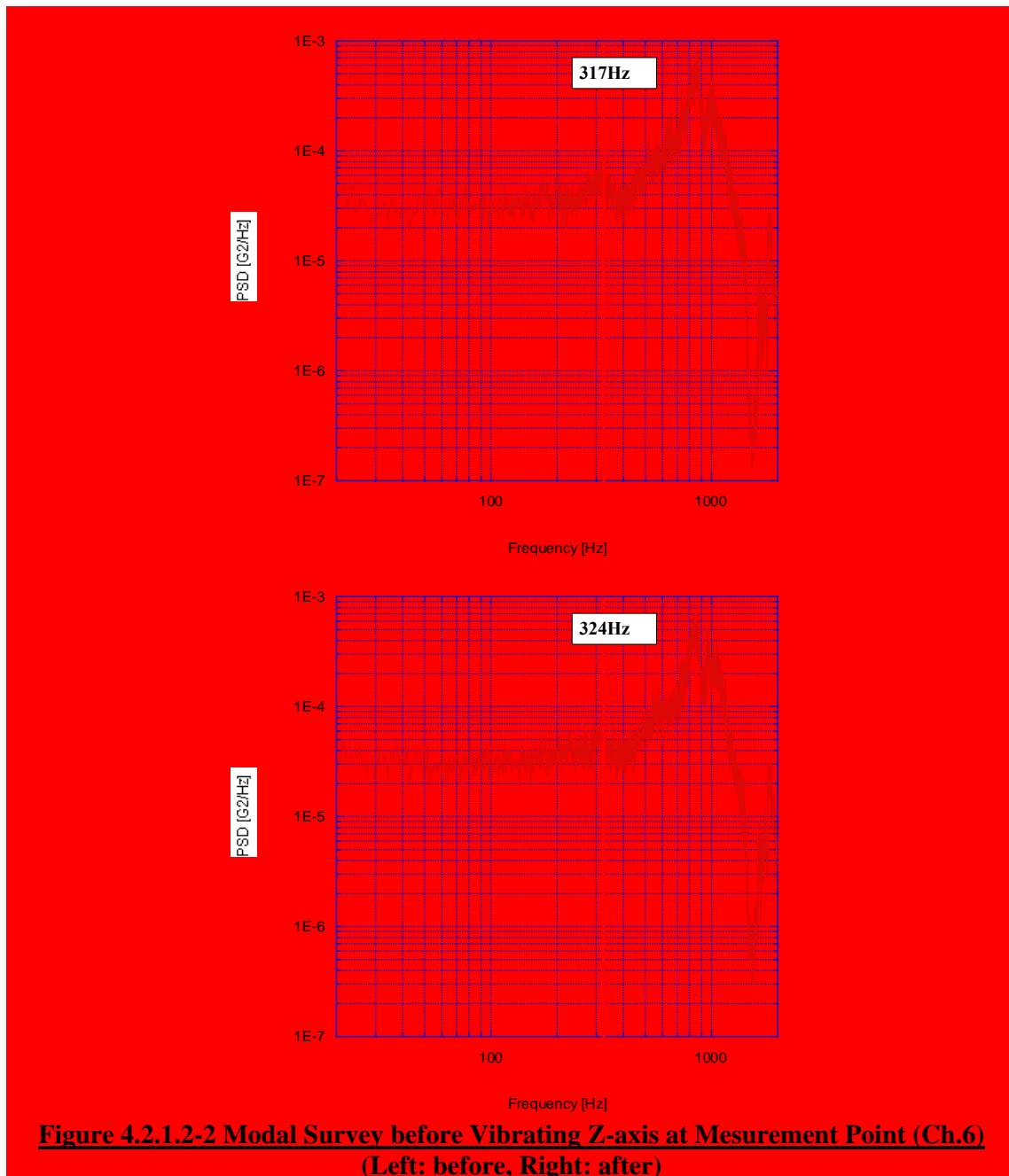


Figure 4.2.1.2-1 Z-axis random vibration at Control Sensor



4.2.1.3. Y-axis vibration

The required acceptance level of vibration was met by the vibration machine. Test results show in Figure 4.2.1.3-1 to Figure 4.2.1.3-2.

In the modal survey before and after the random vibration, the resonant frequency was **321 Hz and 323 Hz**, respectively. There is no significant change in the minimum natural frequency before and after the random vibration. It is 60Hz or higher and meets the requirements.

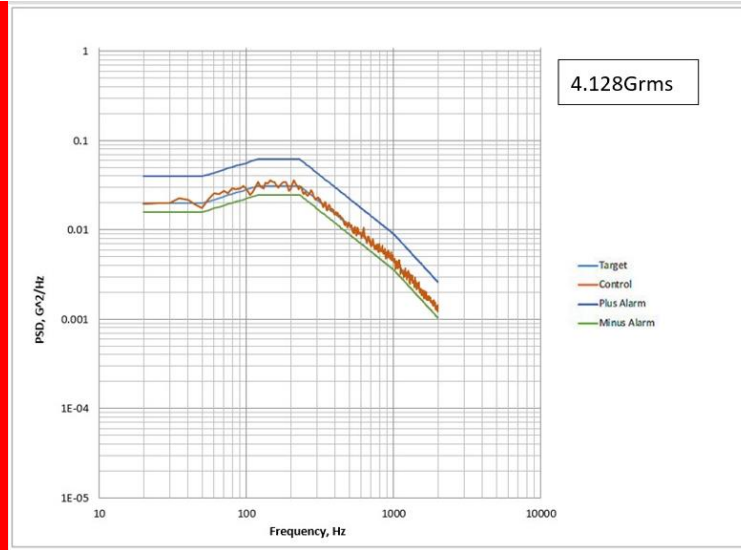
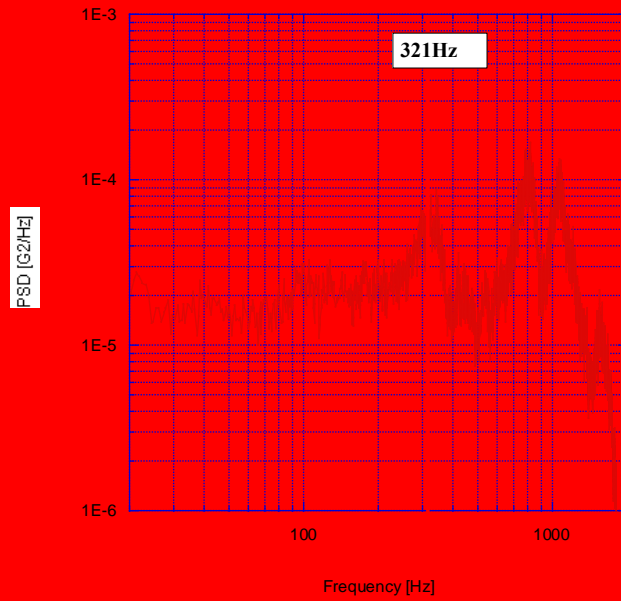
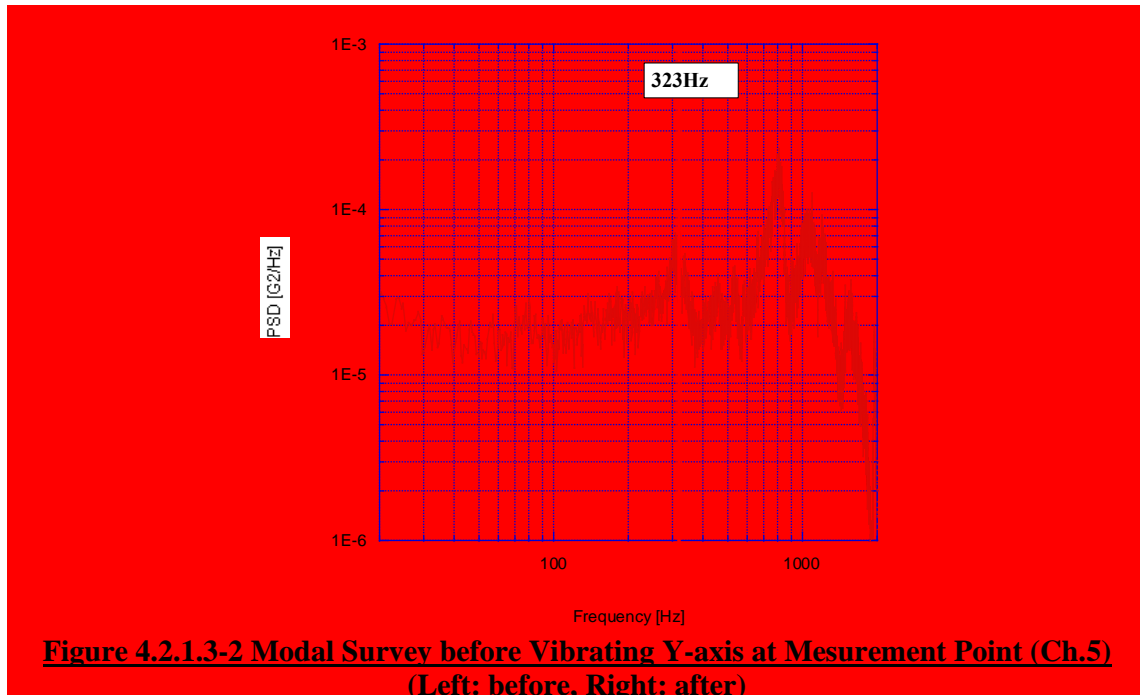


Figure 4.2.1.3-1 Y-axis random vibration at Control Sensor





4.3.Function Test

There was no failure or abnormal status on the voltage of battery and health data of the OBC (On-Board Computer) after vibration test. The satellite had no damage from vibration environment.

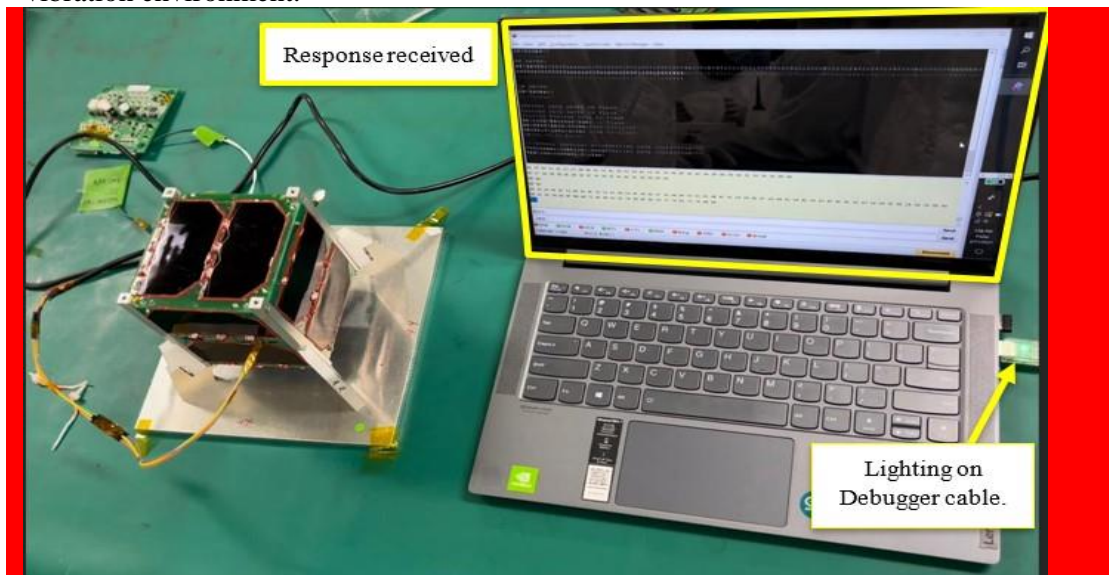


Figure 4.3-1 Dragonfly-1: When DepSW1 was released, SepSW3 got closed, and battery supplied power to the load.

5. Conclusion

No distortion or damage was found on the BIRDS-X flight models. No loosening was found in the fasteners which constitute a main structure. Moreover, no breakage, scratching or cracking of the external panels, the solar cell cover glass or structural components were found based on visual inspection after the vibration.

The minimum natural frequency of each axis is 60Hz or higher, which meets the requirements. The minimum natural frequency of the measured satellite did not change significantly before and after the random vibration test.

There was no inadvertent antenna deployment during and after the vibration. The knot of the wire restraining the antenna was not loosened, and the wire was not damaged or stretched.

No malfunction occurred to CubeSat, either.

In conclusion, the results of the vibration test show that the Dragonfly-1 flight model have successfully passed all requirements and is ready to launch.

No	Verification Items	Results
1	Dragonfly-1	
1-1	No breakage in main structure	Pass
1-2	Main structure needs to satisfy specified natural frequency	Pass
1-3	Natural frequency before and after tests need to remain unchanged	Pass
1-4	No improper antenna deployment, and no malfunction to CubeSat	Pass
1-5	No breakage in glass material such as solar cell cover, camera lens and filters	Pass
1-6	No loosening in all fasteners as stated in the structural analysis report and is listed below: #1-1 to #1-8, #2-1 to #2-6, #3-1 to #3-19, #4-1-#4-4, #5-1-#5-4, #6-1-#6-4, #7-9-#7-16, #7-33-#7-39 The screws are represented in fig 4.3.3-1 to 7 and fig 4.3.3-9 to 10	Pass

Satellite		Natural Frequency (Hz)		a/b
		a. Pre-Modal	b. Analysis	
Dragonfly-1	X	340	435	78%
	Y	345	513	67%
	Z	317	587	54%

Apéndice B

Ejemplo de reporte de fallo generado por la aplicación

Carlos Adrián
Visual Inspection Test Report

Tested Date : 2024/02/10

Responsible: Jairo

This document reports the result of the visual inspection for the vibration test. In this report, the following three items; torque mark shift, scratch, and crack are called error.

The David got the following errors by exposing the vibration environment. Red circle shows the position of detected errors. (If there is no red rectangle, it means the panel does not have error)

- +X panel



Figure 1 +X Panel (left:Before / right:after)

- -X panel

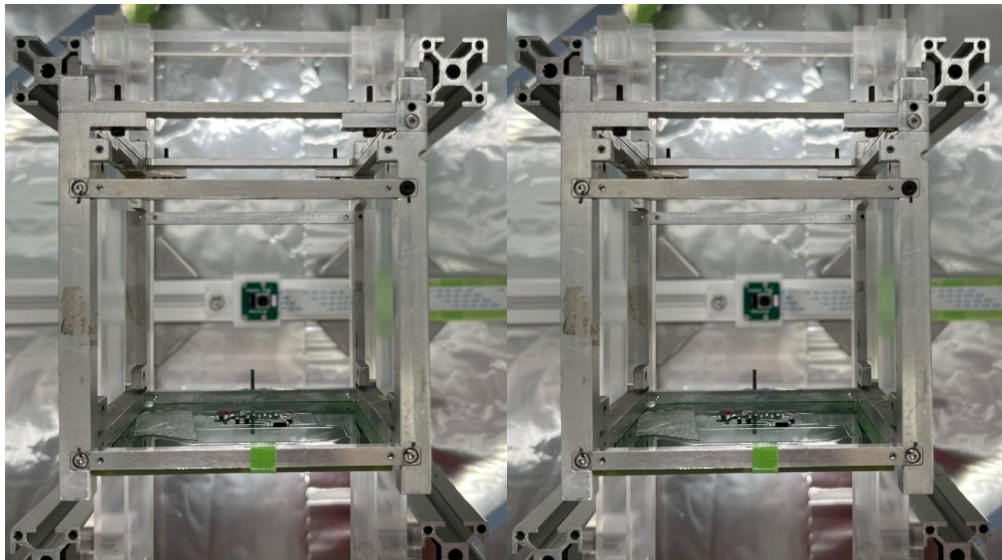


Figure 2 -X Panel(left:Before / right:after)

- +Y panel

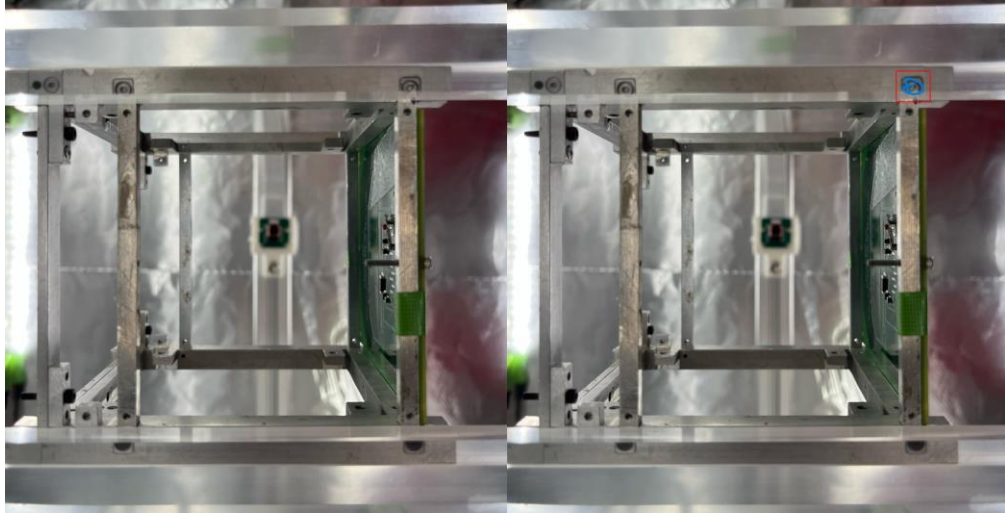


Figure 3 +Y Panel(left:Before / right:after)

- -Y panel

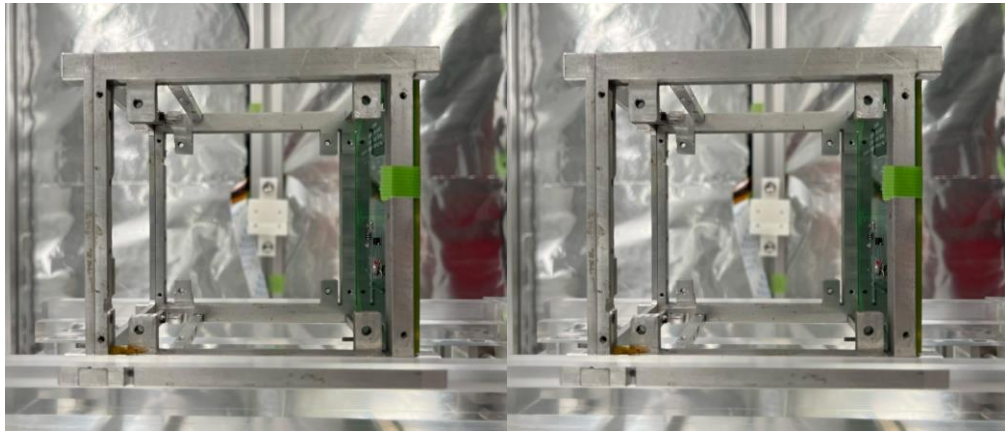


Figure 4 -Y Panel(left:Before / right:after)

- +Z panel

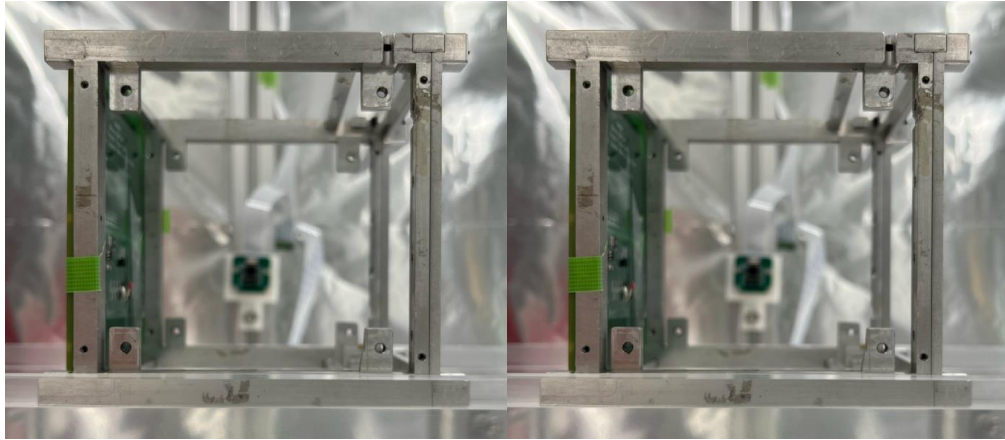


Figure 5 +Z Panel (left: Before / right: after)

- -Z panel

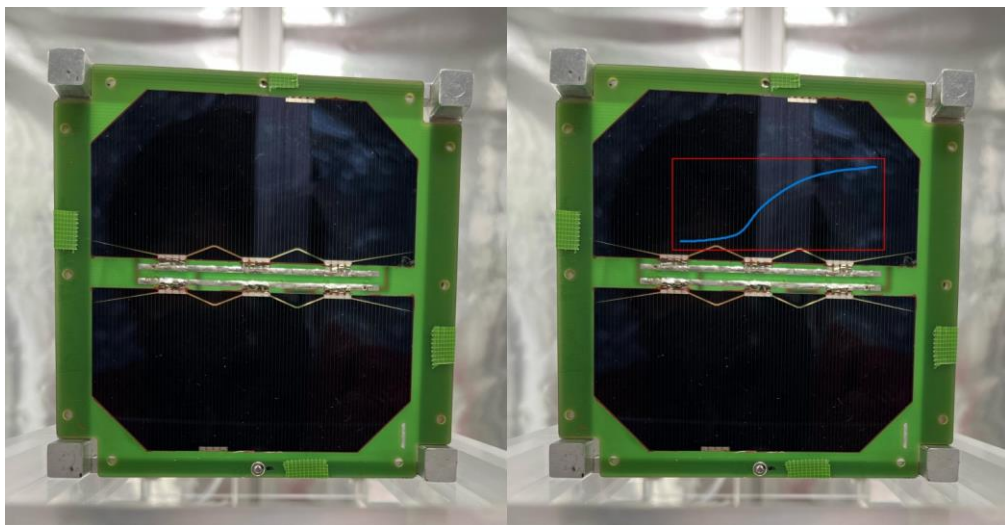


Figure 6 -Z Panel (left: Before / right: after)

In the following images you will be able to see a close up of the detected anomaly.

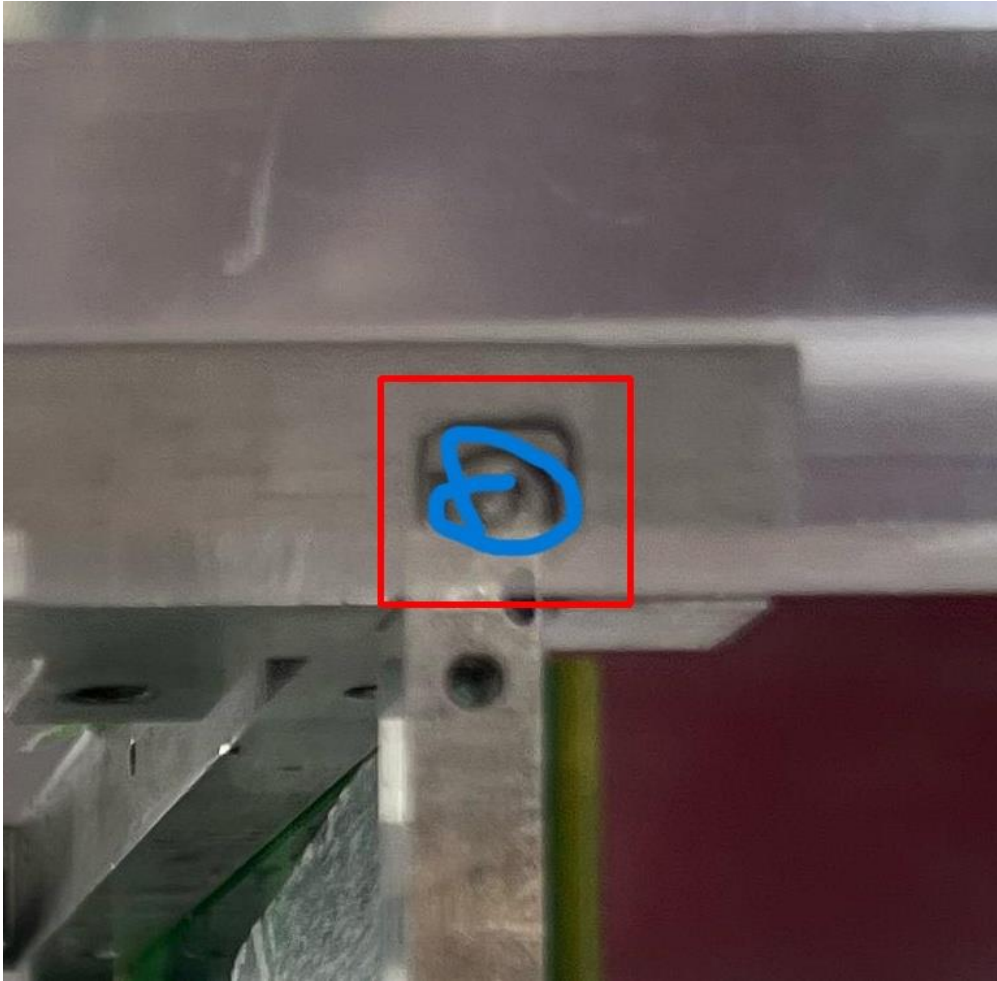


Image coordinates: 2206, 408, 192, 174, '+Y'

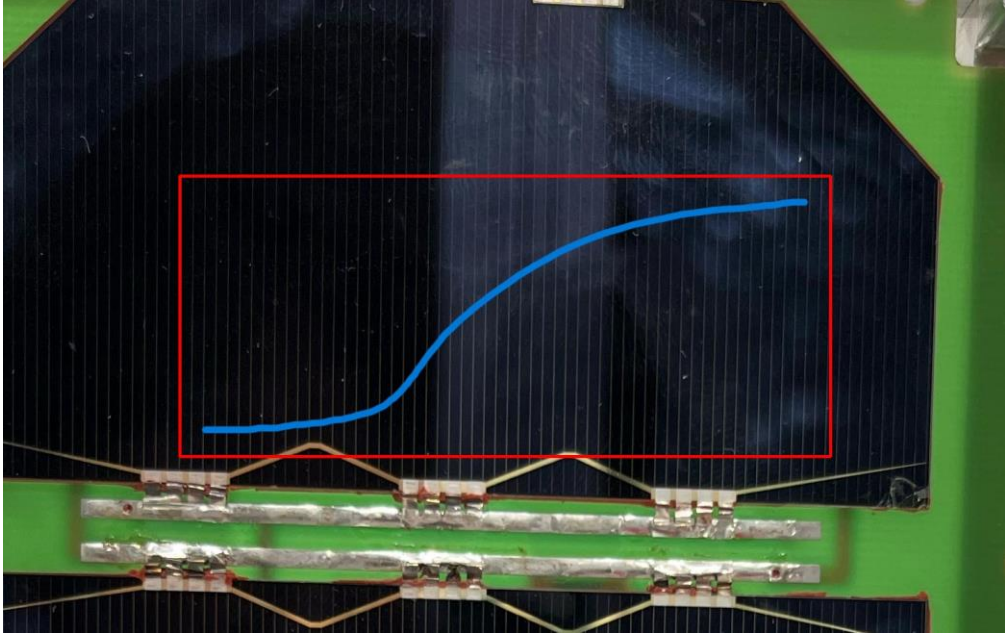
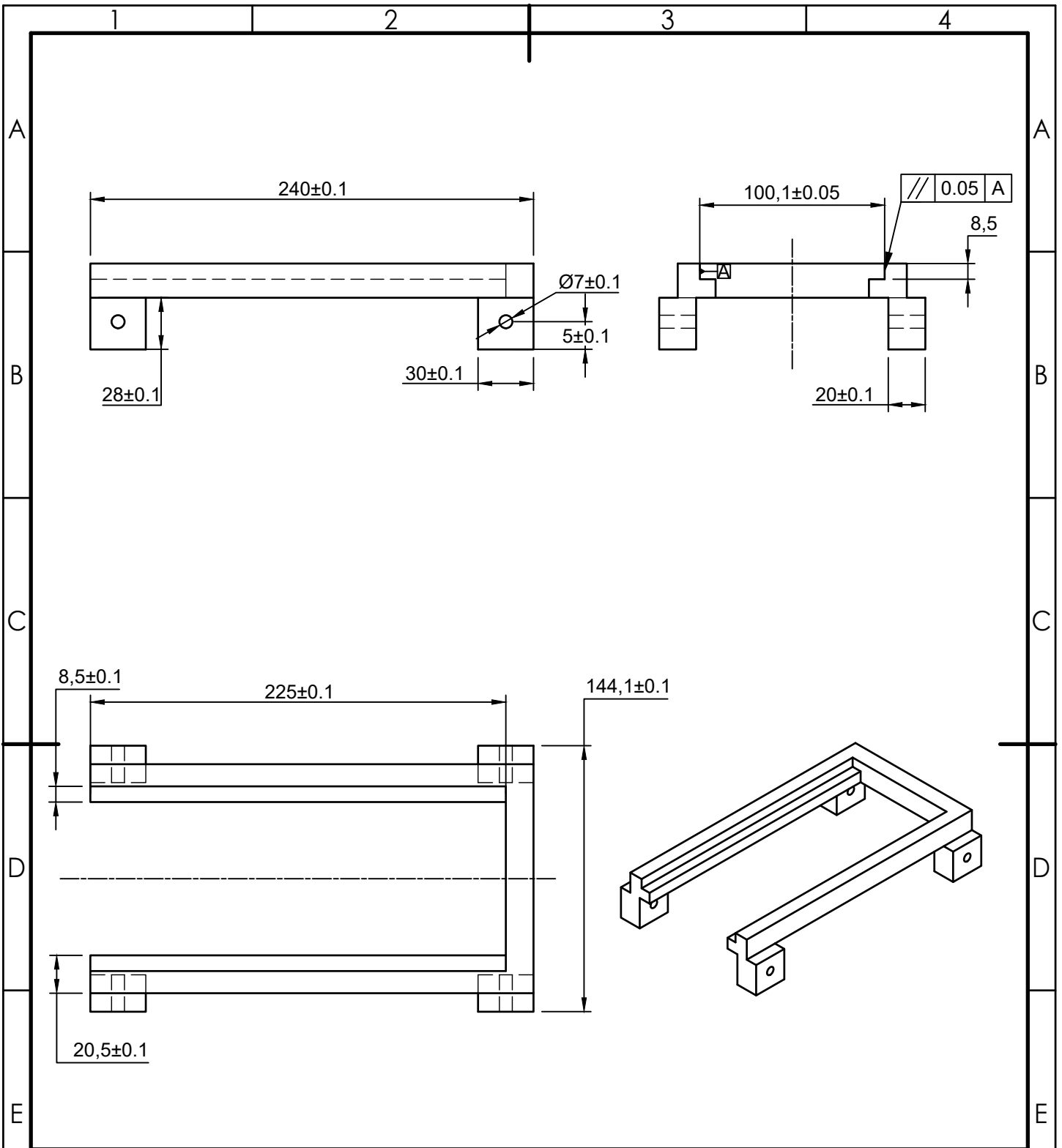


Image coordinates: 833, 762, 1068, 460, '-Z'

Apéndice C

Planos mecánicos



TEC | Tecnológico de Costa Rica

ESCUELA DE INGENIERÍA ELECTROMECAÁNICA
CURSO Proyecto Final de Graduación

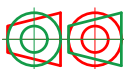
REVISIÓN
A

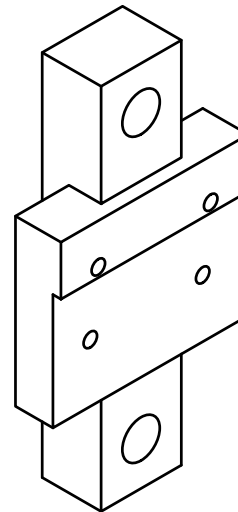
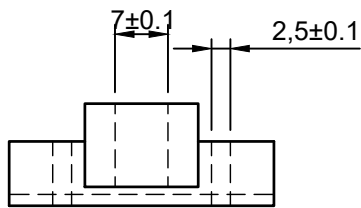
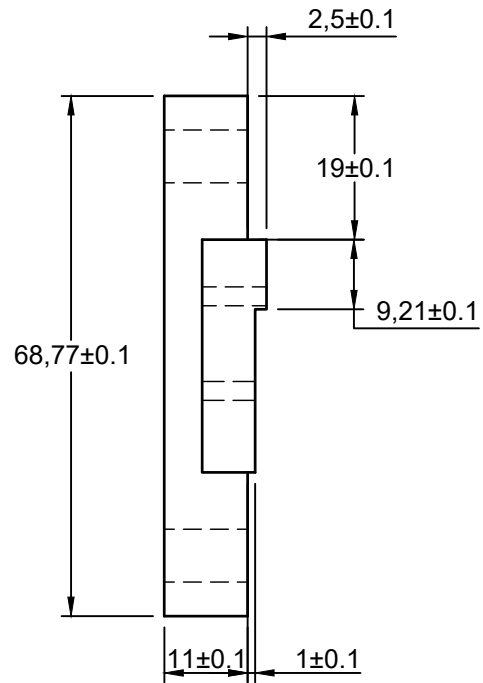
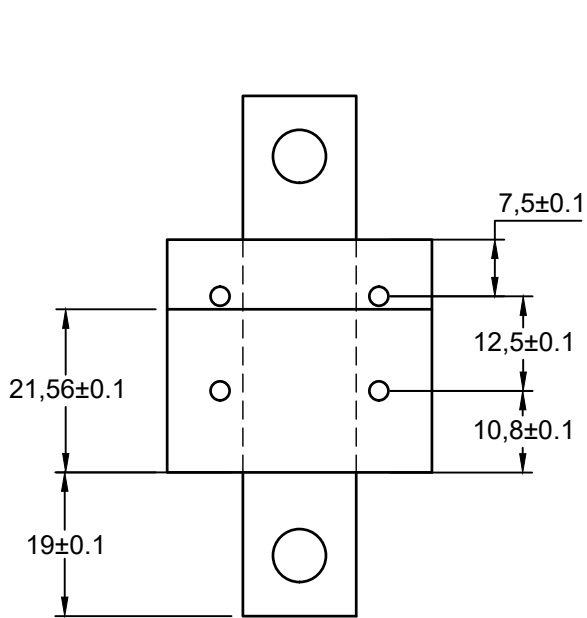
	NOMBRE	FIRMA	FECHA	MATERIAL
DIBUJA	J.Rodríguez		29/02/24	AISI 316
VERIFICA	J.Rodríguez		29/02/24	
				ACABADO GENERAL IT6

PARTE
Soporte de Satélite

SISTEMA	TOLERANCIA SEGÚN INTE ISO 2768
---------	-----------------------------------

PROYECTO Detector de Anomalías	ESCALA 1:3	FORMATO CARTA
UNIDADES mm	HOJA 1 DE 4	





Nota: este componente fue impreso en 3D utilizando PLA, con grosor de capa de 0.09mm

TEC | Tecnológico de Costa Rica

ESCUELA DE INGENIERÍA ELECTROMECÁNICA

REVISIÓN

CURSO

Proyecto Final de Graduación

A

	NOMBRE	FIRMA	FECHA	MATERIAL
DIBUJA	J.Rodríguez		29/02/24	AISI 316
VERIFICA	J.Rodríguez		29/02/24	
				ACABADO GENERAL
				IT6

PARTE

Soporte Cámaras

SISTEMA	TOLERANCIA SEGÚN
	INTE ISO 2768

PROYECTO

Identificador de Anomalías

ESCALA

1:3

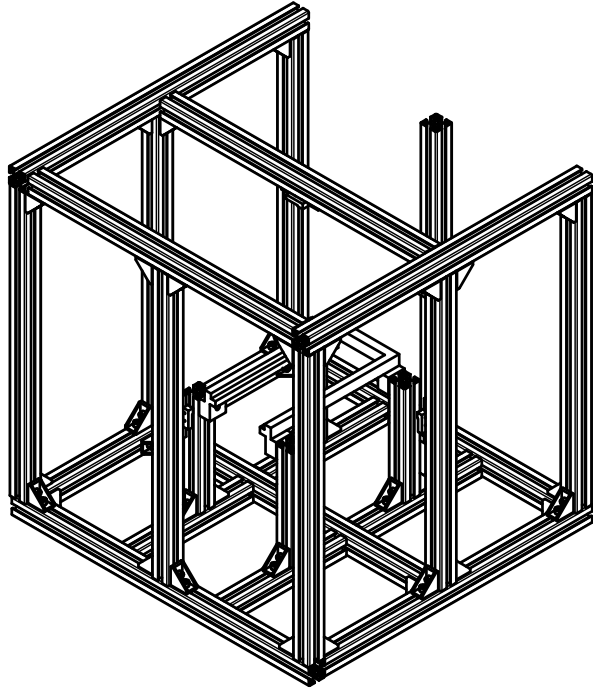
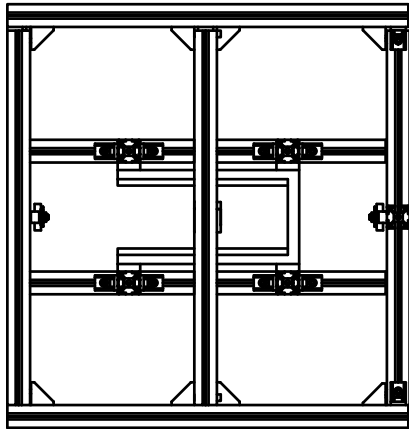
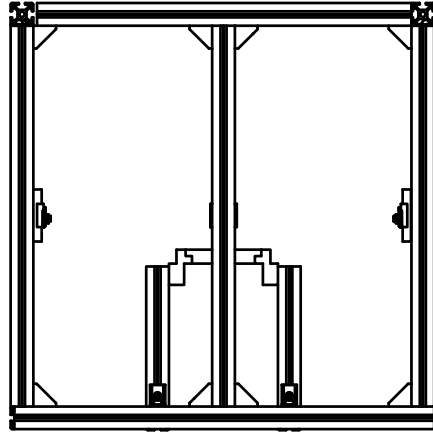
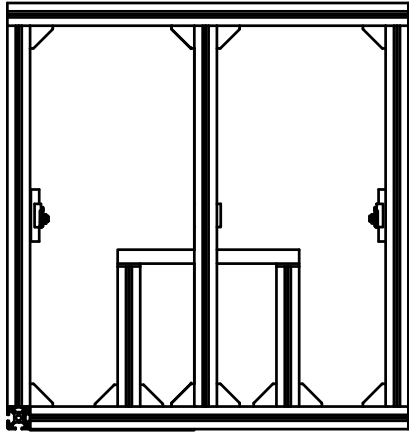
FORMATO

CARTA

UNIDADES

mm

HOJA 2 DE 4



TEC | Tecnológico de Costa Rica

ESCUELA DE INGENIERÍA ELECTROMECÁNICA

REVISIÓN

CURSO
Proyecto Final de Graduación

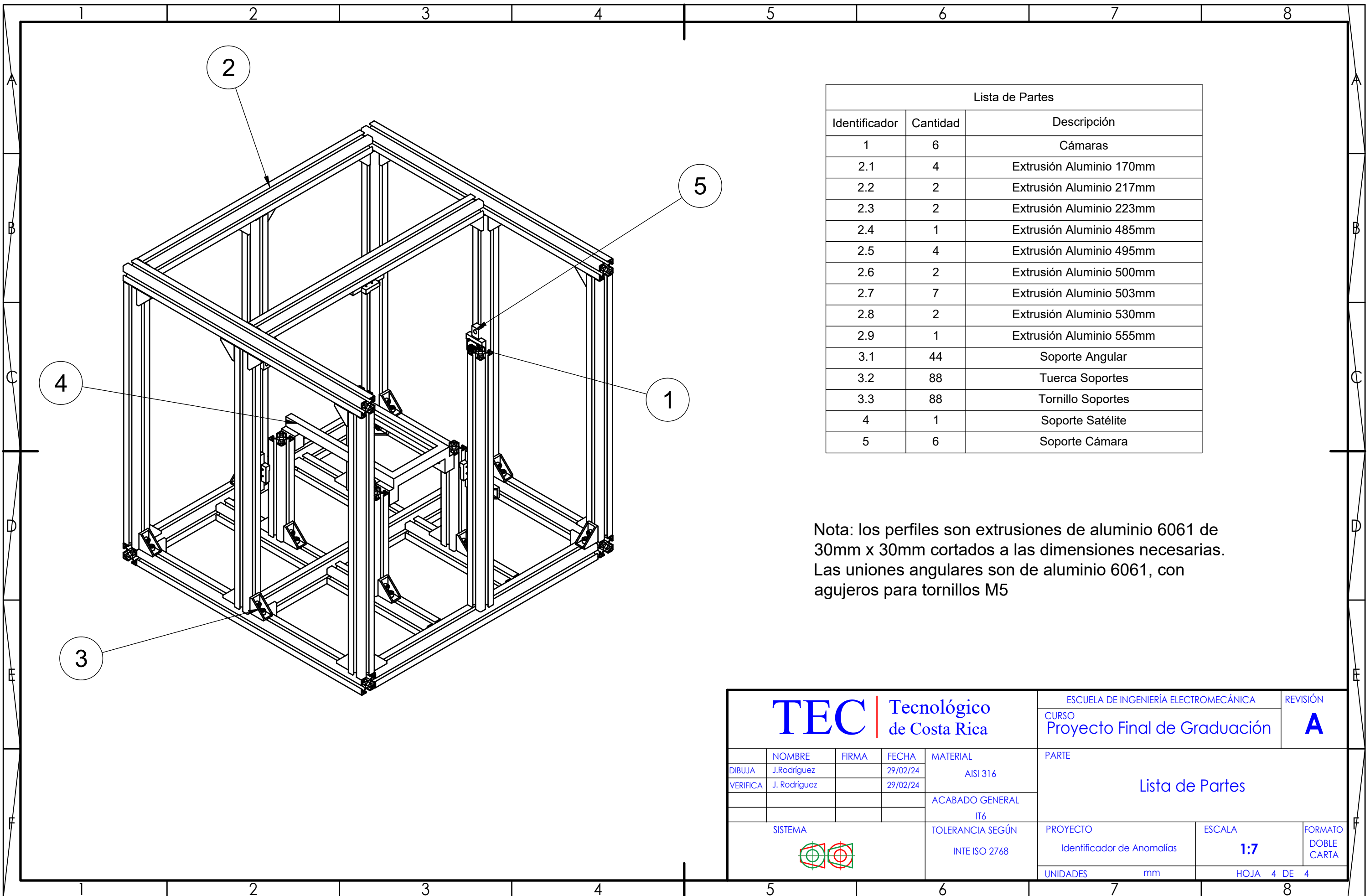
A

	NOMBRE	FIRMA	FECHA	MATERIAL
DIBUJA	J. Rodríguez		29/02/24	AISI 316
VERIFICA	J. Rodríguez		29/02/24	
				ACABADO GENERAL
				IT6

PARTE
Ensamble

SISTEMA	TOLERANCIA SEGÚN
	INTE ISO 2768

PROYECTO	ESCALA	FORMATO
Identificador de Anomalías	1:10	CARTA
UNIDADES	mm	HOJA 3 DE 4



Lista de Partes		
Identificador	Cantidad	Descripción
1	6	Cámaras
2.1	4	Extrusión Aluminio 170mm
2.2	2	Extrusión Aluminio 217mm
2.3	2	Extrusión Aluminio 223mm
2.4	1	Extrusión Aluminio 485mm
2.5	4	Extrusión Aluminio 495mm
2.6	2	Extrusión Aluminio 500mm
2.7	7	Extrusión Aluminio 503mm
2.8	2	Extrusión Aluminio 530mm
2.9	1	Extrusión Aluminio 555mm
3.1	44	Soporte Angular
3.2	88	Tuerca Soportes
3.3	88	Tornillo Soportes
4	1	Soporte Satélite
5	6	Soporte Cámara

Nota: los perfiles son extrusiones de aluminio 6061 de 30mm x 30mm cortados a las dimensiones necesarias. Las uniones angulares son de aluminio 6061, con agujeros para tornillos M5

 Tecnológico de Costa Rica				ESCUELA DE INGENIERÍA ELECTROMECÁNICA		REVISIÓN	
				CURSO Proyecto Final de Graduación		A	
NOMBRE: J. Rodríguez FIRMA: [Firma] FECHA: 29/02/24 MATERIAL: AISI 316				PARTE <h2 style="text-align: center;">Lista de Partes</h2>			
VERIFICA: J. Rodríguez FECHA: 29/02/24 ACABADO GENERAL: IT6							
SISTEMA: 		TOLERANCIA SEGÚN INTE ISO 2768		PROYECTO Identificador de Anomalías		ESCALA 1:7	FORMATO DOBLE CARTA
				UNIDADES: mm		HOJA 4 DE 4	