PRESENTACIÓN DE TRABAJO COMPLETO

Ingeniería y Tecnología

Desarrollo de un subsistema de gestión de imágenes a bordo para nanosatélites

Autor: Chaparro Mancuello, Miguel Angel; chaparromiguel87@fpuna.edu.py

Orientador: Manabe Safi, Ariel Mazaru; amanabe@pol.una.py

Universidad Nacional de Asunción, Facultad Politécnica - Grupo de Investigación en Electrónica y Mecatrónica

---

Resumen

Paraguay inició su programa espacial con el nanosatélite GuaraniSat-1 y actualmente desarrolla el GuaraniSat-2, una plataforma CubeSat sujeta a restricciones energéticas, espaciales y a la limitada capacidad de transmisión por enlace UHF. Dado que la descarga de una sola imagen puede demorar más de una semana y que la operatividad en órbita baja es breve, resulta imprescindible reducir la obtención de datos de escaso interés, como las capturas afectadas por errores de apuntado o sobreexposición solar. Para mitigar esa limitación se diseñó un prototipo del subsistema de adquisición, clasificación y almacenamiento de imágenes de una Unidad de Clasificación de Imágenes (ICU), concebida como interfaz entre la cámara de carga útil y la computadora de a bordo (OBC). El subsistema se articuló sobre un microcontrolador PIC18F67J94, administrador de bajo consumo y disponibilidad continua que gestiona una memoria Flash NOR de 128 MiB y los enlaces seriales, y una computadora monoplaca Raspberry Pi Zero 2W, encendida a demanda, que captura con una cámara HQ de 12 MP, etiqueta cada imagen con su categoría mediante un modelo de clasificación externo y la entrega al microcontrolador, de modo que en Tierra se descarguen solo las relevantes. Se seleccionaron componentes COTS con herencia de vuelo, se diseñó el hardware sobre una placa de cuatro capas, se desarrollaron el firmware embebido y la aplicación de orquestación, y se validó el prototipo en banco. Se obtuvo un subsistema funcional y replicable, orientado a elevar la proporción de capturas útiles recibidas en tierra, como aporte reutilizable para las futuras misiones espaciales del Paraguay.

Palabras clave: nanosatélite, clasificación de imágenes a bordo, memoria flash

Introducción

Durante las últimas dos décadas, el acceso al espacio dejó de ser exclusivo de las grandes potencias para abrirse a gobiernos, empresas privadas e instituciones educativas. La miniaturización de los componentes, los nuevos materiales y los algoritmos de inteligencia artificial permitieron concentrar capacidades antes reservadas a grandes plataformas en satélites pequeños y económicos, fenómeno que se conoce como «Nuevo Espacio» (Santillán, 2024; Henriquet, 2022). En ese marco, los nanosatélites (satélites de entre uno y diez kilogramos) replican casi todas las funciones de un satélite convencional a un costo significativamente menor, lo que despertó el interés de programas espaciales emergentes (Pilkington, 2021; Singh, 2016).

El Paraguay se sumó a esta tendencia con la creación de la Agencia Espacial del Paraguay (AEP) y el desarrollo de los nanosatélites GuaraniSat-1 (2021) y GuaraniSat-2 (2024), conducidos por la AEP en colaboración con la Facultad Politécnica de la Universidad Nacional de Asunción (FPUNA) y con socios internacionales. Entre las misiones del GuaraniSat-1 figuró la captura de imágenes del territorio paraguayo mediante una cámara embarcada, una aplicación de observación terrestre por teledetección satelital.

La operación del GuaraniSat-1, sin embargo, dejó en evidencia una limitación concreta. Una porción significativa de las imágenes obtenidas resultó poco aprovechable: capturas dominadas por el espacio sidéreo por errores de apuntado, destellos en los elementos ópticos por incidencia directa de la luz solar, sobreexposición sobre superficies muy iluminadas, encuadres ocupados por la curvatura terrestre o el horizonte, y cubiertas de nubes que ocultaban el territorio de interés. Estas imágenes no aportaban información relevante a los objetivos de la misión y, pese a ello, ocupaban el mismo recurso escaso de almacenamiento y de enlace que las capturas útiles. La restricción se agravó por las características del enlace de descenso: en una órbita terrestre baja (LEO) a unos 500 km de altitud, el nanosatélite dispuso de aproximadamente cuatro pasadas útiles diarias sobre la estación terrena, con ventanas efectivas del orden de 8 a 10 minutos, de modo que la descarga completa de una sola imagen por el enlace en banda UHF demoró entre una y dos semanas. La corta vida útil de un nanosatélite en LEO, del orden de meses a pocos años, agrava la situación: cada pasada no aprovechada y cada imagen sin valor que ocupa el enlace restan peso a una misión de horizonte temporal acotado.

El procesamiento de datos a bordo constituye una palanca reconocida para atenuar este problema: seleccionar, comprimir o descartar la información antes

de transmitirla evita consumir recursos críticos en datos de escaso interés (KP Labs, 2025). Las alternativas por la vía del hardware óptico, del subsistema de apuntado o de un enlace más veloz tienen alcance limitado, pues los factores que producen imágenes inútiles (orbitales, meteorológicos y solares) son independientes entre sí y persisten ante mejoras en cualquiera de esas etapas. Mientras la decisión de qué imágenes enviar a tierra se difiera al postprocesamiento en estación terrena, la energía, el tiempo de enlace y la vida útil del nanosatélite se gastan en transmisiones que no aportan valor.

De ahí la conveniencia de incorporar a bordo un mecanismo capaz de discriminar las capturas según su utilidad antes de que ocupen almacenamiento o tiempo de enlace. El presente trabajo aborda el diseño y la validación funcional de un prototipo de ese mecanismo, denominado Unidad de Clasificación de Imágenes (ICU, por sus siglas en inglés), concebido como interfaz entre la cámara de carga útil y la computadora de a bordo (OBC) del nanosatélite. El prototipo se planteó como propuesta de ingeniería para una misión futura que la AEP decida lanzar, y no como carga integrada al GuaraniSat-2. El subsistema se articuló sobre dos planos computacionales de perfiles de consumo disímiles: un microcontrolador de bajo consumo, siempre disponible, como administrador del subsistema, y una computadora de placa única, encendida a demanda, como motor de inferencia que ejecuta el modelo de clasificación. Las secciones siguientes presentan los objetivos del trabajo, los materiales y métodos empleados en el diseño y la construcción, los resultados de la validación en banco, y las conclusiones derivadas.

Objetivos

General

Desarrollar un subsistema de gestión para la adquisición, clasificación y almacenamiento de imágenes a bordo de nanosatélites.

Específicos

1. Diseñar la arquitectura del subsistema de gestión de imágenes útiles para descargar a Tierra.
2. Desarrollar el hardware del subsistema de gestión para su operación en órbita baja terrestre.
3. Desarrollar el software para la descarga efectiva de imágenes útiles.
4. Validar el funcionamiento del subsistema de gestión en pruebas de laboratorio.

Materiales y Métodos

El trabajo se enmarcó en la investigación aplicada, con enfoque cuantitativo, de nivel descriptivo y diseño experimental. La componente de ingeniería se materializó en el diseño, la construcción y la validación funcional en laboratorio de un prototipo de ICU, cuyo desempeño se verificó mediante ensayos sobre el hardware desarrollado

bajo condiciones controladas que emularon aspectos seleccionados de la operación en órbita.

Relevamiento de requisitos

El relevamiento se llevó a cabo mediante reuniones de trabajo con el ingeniero a cargo del desarrollo del GuaraniSat-2 en el edificio CETUNA, quien aportó además la experiencia acumulada en la misión precedente, complementadas con el estudio de la documentación pública de la plataforma de bus abierto BIRDS del Instituto de Tecnología de Kyushu (Kyutech) y de la misión Kitsune (un nanosatélite de formato 6U), de la cual GuaraniSat-1 y GuaraniSat-2 heredan componentes y decisiones de arquitectura. De allí se derivaron tres conjuntos de restricciones: físicas (factor de forma del bus abierto BIRDS: contorno de 80 mm por 83 mm, cuatro orificios de sujeción de 3,8 mm y un conector de 50 pines), eléctricas y térmicas (carril lógico de 3,3 V, compatibilidad con los ciclos de eclipse e iluminación solar de LEO, preferencia por componentes con herencia de vuelo o, como piso, de grado industrial o automotriz) y lógicas (ausencia, al momento del diseño, de una especificación cerrada de protocolo para el enlace serial con la OBC, lo que trasladó al trabajo la definición del formato de trama, el conjunto de comandos y el esquema de validación por CRC-16).

A partir de los objetivos específicos y de las restricciones señaladas se definieron los requerimientos del subsistema, diferenciados entre funcionales y no funcionales, cada uno con su criterio de aceptación. Los funcionales establecen lo que el subsistema debe hacer: adquirir pares de imágenes (resolución plena más versión reducida para inferencia), invocar el modelo de clasificación externo, descartar las imágenes no relevantes, almacenar de forma no volátil las aceptadas, intercambiar tramas con la OBC y con la computadora de placa única, transferir imágenes íntegras a la OBC, ejecutar el borrado escalonado de la Flash, controlar el carril de alimentación de la computadora de placa única y gestionar un manifiesto consultable como telemetría. Los no funcionales fijan las condiciones de calidad y diseño: operación con carril lógico de 3,3 V, respeto del factor de forma BIRDS, uso exclusivo de COTS con herencia o grado industrial, firmware sin recursión, sin asignación dinámica de memoria ni punteros a función en el camino crítico (Holzmann, 2006), y tolerancia a reinicios sin pérdida de datos ni corrupción del índice. La Tabla 1 resume los parámetros operacionales adoptados como valores de diseño.

Arquitectura del subsistema

El principio rector dispuso el subsistema en dos planos computacionales con perfiles de consumo disímiles, de modo que cada función residiera en el plano cuyo costo energético y cuya capacidad de cómputo le

resultaran proporcionales (decomposición orientada a carga útil; Araguz et al., 2018). El microcontrolador PIC18F67J94, administrador alimentado de manera continua mientras la ICU permaneciera energizada, asumió la interlocución con la OBC, la gestión de la memoria Flash NOR local, el arbitraje del bus SPI compartido y el control del carril de alimentación de la computadora de placa única. La Raspberry Pi Zero 2W, motor de inferencia encendido a demanda, asumió la captura del par de imágenes, la invocación del modelo de clasificación externo y la entrega de las imágenes aceptadas al microcontrolador. Esta partición se justificó en tres planos: energético (mantener la computadora de placa única siempre encendida resultaba incompatible con el presupuesto de un nanosatélite en LEO, en tanto que el microcontrolador admite el régimen continuo por su bajo consumo), de cómputo (la inferencia excede los recursos del microcontrolador y exige una plataforma con sistema operativo de propósito general; Carlson, 2020) y de disponibilidad (la pasarela única de comunicaciones con la OBC no admite las interrupciones del ciclo de encendido y apagado de la computadora de placa única).

La operación se vertebra en cuatro flujos de datos: (I) captura, clasificación y almacenamiento intermedio en la computadora de placa única; (II) transferencia desde la computadora de placa única a la Flash local por SPI; (III) transferencia desde la Flash local hacia la OBC por tubería DMA de doble búfer; y (IV) liberación de espacio en la Flash local mediante borrado escalonado una vez que la OBC consumió la imagen.

Hardware del prototipo

La selección de componentes COTS se guió por cuatro criterios: herencia de vuelo en misiones previas, bajo consumo, rango de temperatura compatible con el entorno térmico interno del nanosatélite en LEO y, como piso mínimo, grado industrial o automotriz; el recurso a componentes no cualificados se asume bajo un caso de aseguramiento explícito, conforme a la práctica documentada para COTS en órbita baja (Budroweit y Patscheider, 2021). El microcontrolador PIC18F67J94 y la memoria Flash NOR MT25QL01GBBB (128 MiB) replicaron decisiones validadas en el GuaraniSat-1 y en la misión Kitsune, lo que permitió reutilizar el entorno de desarrollo asociado (CCS C Compiler). La computadora de placa única Raspberry Pi Zero 2W se sustentó en su uso previo en la misión GASPACS de la Universidad Estatal de Utah, y el conjunto óptico lo integraron la cámara HQ de 12 MP con lente de 16 mm. Como alternativa se evaluó un microcontrolador STM32H7 para asumir íntegramente la orquestación, opción descartada en favor de la computadora de placa única por simplicidad de desarrollo.

El esquemático se diseñó en Autodesk

Fusion 360 y reprodujo dos patrones validados en Kitsune: el enlace UART entre la OBC y el microcontrolador precedido por un búfer de nivel lógico, y un multiplexor digital aguas arriba de la memoria Flash NOR que arbitra su acceso entre el microcontrolador y la computadora de placa única; el bus SPI es estrictamente monomaster, por lo que el acceso simultáneo de dos controladores debe evitarse por hardware. Se incorporaron circuitos de protección contra sobrecorriente (LTC4361) heredados del mismo antecedente. El ruteo se resolvió sobre una placa de cuatro capas con apilamiento señal-superior, masa, alimentación y señal-inferior, con polígonos de masa continuos en ambas capas de señal; en las líneas SPI se aplicó control de impedancia aprovechando dicho apilamiento. Las consideraciones térmicas y de compatibilidad electromagnética se materializaron en la continuidad del plano de masa (retorno y blindaje) y en vías térmicas bajo los componentes disipadores. El factor de forma siguió el estándar de bus abierto BIRDS de Kyutech (80 mm por 83 mm, cuatro orificios de 3,8 mm y conector de 50 pines). La manufactura se encargó al servicio comercial JLCPCB en una única tirada de cuatro unidades.

Firmware del microcontrolador

El firmware se desarrolló en lenguaje C con el compilador CCS, ejecutado sobre Linux mediante la capa de compatibilidad Wine, y se organizó en una arquitectura por capas. En la capa inferior residen los controladores de periférico (SPI para la Flash NOR, UART para las interfaces con la OBC y con la computadora de placa única, y líneas digitales para el arbitraje del multiplexor SPI y la habilitación del carril de alimentación). Sobre ellos se ubican los módulos de protocolo (CRC-16, conversiones de orden de byte, construcción de tramas y envoltura ACK/NACK) y los de almacenamiento (índice de entradas, gestión de metadatos, asignación de espacio por búsqueda de huecos, borrado escalonado en sectores de 4 KiB, 32 KiB y 64 KiB, y tubería DMA de doble búfer para la transferencia de imágenes hacia la Flash de la OBC). En la capa superior, un enrutador de comandos despacha cada trama al manejador correspondiente. La ejecución adoptó el esquema de super-loop con recepción UART atendida por interrupciones, complementado con máquinas de estado explícitas para la escritura en Flash y la transferencia por DMA; este modelo cooperativo es determinista por construcción y resulta preferible a un sistema operativo en tiempo real cuando la carga de trabajo es planificable (Jones, 2024).

El protocolo con la OBC se especificó en el marco del trabajo: trama con encabezado, identificador de comando, longitud, carga útil de hasta 39 B, suma de verificación CRC-16 y byte de cierre (tamaño de 6 a 45 B), validada con la variante CRC-16/CCITT

(polinomio 0x1021, semilla 0x1D0F) por sus garantías de detección sobre los errores típicos de los canales serie (Williams, 1993). Para depurar el algoritmo de administración de la memoria Flash sin someterlo al ciclo de compilación y carga sobre el microcontrolador, se construyó una simulación del subsistema de almacenamiento en C estándar (desarrollo *off-target*), empleando un archivo binario de 128 MiB como sustituto de la memoria física. La validación final del firmware se efectuó sobre el hardware construido, con el programador PICkit 3.

Orquestación en la computadora monoplaca

La orquestación en la Raspberry Pi Zero 2W se implementó como una aplicación en Python, organizada en módulos por área de responsabilidad: configuración y registro, núcleo (orquestador y *pipeline* de imágenes), interfaces de hardware (cámara, memoria Flash y UART) y procesamiento (CRC-16, metadatos, red neuronal, base de datos de manifiesto y generación de miniaturas). El núcleo ejecuta un bucle de eventos no bloqueante (intervalo de 10 ms) que atiende los comandos recibidos por UART desde el microcontrolador, supervisa en segundo plano las escrituras a la Flash SPI ejecutadas por un hilo trabajador y refresca un *watchdog* de software (período de 10 s).

El *pipeline*, disparado por el comando de captura, adquiere de la cámara una pareja compuesta por la imagen JPEG a resolución plena y una versión reducida a 224×224 píxeles en RGB para inferencia, invoca el modelo de clasificación (componente de terceros, tratado como caja negra que recibe un arreglo de píxeles y devuelve un vector de puntuaciones; Krizhevsky et al., 2012; Howard et al., 2017) y, cuando la clase inferida corresponde a una categoría de interés, inyecta los metadatos EXIF pertinentes, registra la imagen en una base de datos SQLite que actúa como manifiesto y la persiste como archivo; las imágenes rechazadas se descartan sin almacenarse. El uso de una base de datos embebida para el manifiesto se sustentó en sus garantías de atomicidad, indexado e integridad estructural frente a la escritura de ficheros sueltos (SQLite, 2017). La comunicación con el microcontrolador se efectuó por UART a 115200 baudios, reutilizando el formato de trama y la validación CRC-16 del enlace con la OBC. La escritura sobre la Flash descompone la imagen en páginas de 256 B y las graba en tres pasadas (alineamiento del residuo inicial, ráfaga de páginas completas y residuo final), con el ciclo WREN + PAGE_PROGRAM por página y sondeo del bit de ocupación interno del dispositivo.

Banco de ensayos y caracterización energética

Los ensayos funcionales se efectuaron sobre la placa fabricada con todos los

componentes montados en su configuración definitiva. La OBC se emuló desde una computadora personal mediante el terminal serial libre *tio*, que generó las tramas UART del protocolo definido y registró la totalidad del tráfico para su depuración. La alimentación se suministró con una fuente de laboratorio sobre los carriles de 3,3 V y 5 V, y la instrumentación de medición se limitó al multímetro digital. La verificación frente a condiciones térmicas extremas se restringió al análisis teórico sobre los rangos declarados en las hojas de datos; las pruebas ambientales (vibración, vacío, ciclado térmico y radiación), reservadas a las unidades embarcadas, quedaron fuera del alcance por tratarse de un prototipo no destinado al vuelo.

Resultados y Discusión

El trabajo culminó en un prototipo electrónico funcional, junto con el firmware embebido y la aplicación de orquestación, validados sobre el hardware construido frente a los criterios de aceptación definidos.

Realización de la arquitectura y verificación funcional

La partición entre el microcontrolador (administrador de bajo consumo y disponibilidad continua) y la computadora de placa única (motor de inferencia encendido a demanda) se implementó conforme al reparto de responsabilidades, y los cuatro flujos de datos operaron de extremo a extremo sobre el hardware construido. La verificación funcional se articuló en torno a los criterios de aceptación de los requerimientos:

- **Respuesta del prototipo a cada comando** del protocolo definido, con reconocimiento o rechazo de las tramas según el CRC-16 y el formato, y emisión de ACK/NACK.
- **Almacenamiento simultáneo de imágenes en alta resolución** en la Flash local (meta: 25 imágenes), con recuperación sin pérdida.
- **Transferencia íntegra de imágenes** entre la OBC emulada y la memoria Flash local, verificada por comparación byte a byte entre la copia provista por la computadora de placa única, la almacenada en la Flash del prototipo y la recuperada por la OBC emulada; la reconstrucción del archivo JPEG a partir de su volcado hexadecimal se empleó como verificación visual complementaria.

Discusión

Los resultados convergen con la literatura sobre carga útil de imágenes en nanosatélites. La partición adoptada (un microcontrolador dedicado al arbitraje de la memoria Flash y al protocolo con la OBC, una computadora monoplaca como motor de inferencia, y una memoria Flash NOR como repositorio no volátil) reproduce, con ajustes propios, la decomposición funcional

reportada por Maldonado (2022) para el sistema de captura y preprocesamiento del CubeSat SUCHAI 2 (que combinó una Raspberry Pi Zero W con un microcontrolador PIC para el control de la cámara y la interlocución con la OBC) y la topología de carga útil *off-the-shelf* descrita por Azami et al. (2022) para un CubeSat 6U de clasificación de incendios forestales con red neuronal convolucional a bordo. La decisión de procesar y clasificar a bordo antes del envío a tierra, en lugar de delegar la selección al postprocesamiento en estación terrena, distingue al presente trabajo de propuestas centradas únicamente en la captura y el apuntado (Hudson et al., 2024; Orts, 2020). La descomposición por planos de carga útil se alinea con las guías de arquitectura de software orientadas a carga útil para nanosatélites (Araguz et al., 2018); la elección del super-loop sobre un sistema operativo en tiempo real sigue el criterio de adoptar el mecanismo más simple capaz de satisfacer la carga de trabajo (Jones, 2024; Carlson, 2020); y el uso de CRC-16, de una base de datos embebida y de un modelo de clasificación liviano se apoya en la práctica establecida (Williams, 1993; SQLite, 2017; Howard et al., 2017).

El prototipo presenta limitaciones que acotan el alcance de sus conclusiones y definen líneas de trabajo futuro. La validación se efectuó en banco, sin cámara térmica, vibración, vacío ni exposición a radiación, por tratarse de un prototipo. El modelo de clasificación se trató como caja negra externa, de modo que su tasa de acierto (que gobierna la eficacia global del filtrado) no se evaluó en este trabajo. La vida útil de la memoria Flash se estimó a partir de las hojas de datos, sin caracterización experimental. Cada una de estas decisiones obedeció a un compromiso explícito entre simplicidad, herencia de vuelo y alcance de un prototipo de propuesta.

Conclusiones

En atención a los objetivos planteados, se desarrolló un prototipo funcional del subsistema de gestión para la adquisición, clasificación y almacenamiento de imágenes a bordo. Se diseñó su arquitectura sobre dos planos computacionales de perfiles de consumo disímiles: el microcontrolador como administrador siempre disponible y la computadora de placa única encendida a demanda. Se desarrolló el hardware sobre una placa de cuatro capas con componentes COTS de herencia de vuelo, control de impedancia en las líneas SPI y el factor de forma del bus abierto BIRDS, apto para la operación en órbita terrestre baja. Se desarrolló el software en ambos planos: el firmware del microcontrolador, bajo reglas de codificación segura, y la aplicación de orquestación de la computadora de placa única (captura, invocación del modelo externo, etiquetado por metadatos, manifiesto y entrega al microcontrolador), que habilita la descarga selectiva de las

imágenes útiles. El prototipo integrado se validó funcionalmente en banco contra los criterios de aceptación definidos.

La partición entre un microcontrolador administrador, de bajo consumo y disponibilidad continua, y una computadora de placa única encendida a demanda demostró ser una arquitectura coherente con las restricciones de volumen, energía y temperatura de un CubeSat, y consistente con la práctica internacional reportada para cargas útiles de imágenes en nanosatélites. Al trasladar a bordo la decisión de qué imágenes conservar, el subsistema está orientado a elevar la proporción de capturas útiles efectivamente recibidas en tierra, y constituye una base reutilizable y económicamente viable para las futuras misiones espaciales del Paraguay.

Referencias Bibliográficas

Araguz, C., et al. (2018). Design guidelines for general-purpose payload-oriented nanosatellite software architectures. *Journal of Aerospace Information Systems, 15*(3), 107–119. https://doi.org/10.2514/1.I010537 [completar autores]

Azami, M. H., Orger, N. C., Schulz, V. H., Oshiro, T., & Cho, M. (2022). *Earth observation mission of a 6U CubeSat with 5-meter resolution for wildfire image classification using convolutional neural network approach* [título en la fuente: «Misión de observación de la Tierra de un CubeSat 6U con una resolución de 5 metros para la clasificación de imágenes de incendios forestales mediante un enfoque de red neuronal convolucional»]. [completar: publicación, volumen, páginas, DOI]

Budroweit, J., & Patscheider, H. (2021). Risk assessment for the use of COTS devices in space systems under consideration of radiation effects. *Electronics, 10*(9), 1008. https://doi.org/10.3390/electronics10091008

Carlson, J. (2020, 26 de octubre). *So you want to build an embedded Linux system?* jaycarlson.net. https://jaycarlson.net/embedded-linux/

Holzmann, G. J. (2006). The power of 10: Rules for developing safety-critical code. *Computer, 39*(6), 95–97. https://doi.org/10.1109/MC.2006.212

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient convolutional neural networks for mobile vision applications* (arXiv:1704.04861). arXiv. https://arxiv.org/abs/1704.04861

Hudson, J., Smith, … De Backer (2024). *Diseño de un CubeSat de bajo coste para observación terrestre y transmisión de datos* [Carolina del Sur, EE. UU.]. [completar: autores completos, publicación, páginas]

Jones, N. (2024, 11 de abril). *You don't need an RTOS (Part 1)*. EmbeddedRelated. https://www.embeddedrelated.com/showarticle/1636.php

KP Labs. (2025, 28 de enero). *What is on-board data processing?* https://www.kplabs.space/news/what-is-on-board-data-processing

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. En *Advances in Neural Information Processing Systems 25 (NIPS 2012)* (pp. 1097–1105). Curran Associates.

Maldonado, D. (2022). *Implementación del sistema de captura de imágenes y preprocesamiento a bordo del CubeSat SUCHAI 2* [Chile]. [completar: tipo de documento, institución, URL]

Micron Technology. (2024). *NOR/NAND flash guide: Selecting a flash memory solution for embedded applications*. https://www.micron.com/products/storage/nor-flash

Microchip Technology. (2017). *PIC18F67J94 family data sheet: 64/80-pin, low-power, high-performance microcontrollers with nanoWatt XLP technology*. Microchip Technology Inc.

Orts, [iniciales] (2020). *Propuesta y diseño de un nanosatélite de comunicaciones para fines educativos* [España]. [completar: nombre del autor, tipo de documento, institución]

Pilkington, B. (2021, 15 de octubre). *What are nanosatellites?* AZoNano. https://www.azonano.com/article.aspx?ArticleID=5840

Santillán, M. L. (2024, 24 de abril). *New Space o el presente de la exploración espacial*. Ciencia UNAM. https://ciencia.unam.mx/leer/1398/new-space-o-el-futuro-de-la-exploracion-espacial

Singh, I. (2016, 17 de octubre). *What are nanosatellites and why do they matter?* Geospatial World. https://geospatialworld.net/blogs/nanosatellites-or-small-satellites-are-going-to-play-a-big-role/

SQLite. (2017). *35% faster than the filesystem*. https://sqlite.org/fasterthanfs.html

Williams, R. N. (1993). *A painless guide to CRC error detection algorithms* (versión

3.0). Rocksoft. http://www.ross.net/crc/download/crc_v3.txt

Agradecimientos

Se agradece a la Agencia Espacial del Paraguay (AEP) y a su laboratorio SpaceLab, al Grupo de Investigación en Electrónica y Mecatrónica (GIEM) y a la Facultad Politécnica de la Universidad Nacional de Asunción por la infraestructura y el acompañamiento. Al orientador, Prof. MSc. Ariel Mazaru Manabe Safi y a la orientadora metodológica, Prof. Mst. María Soledad Ayala Rodríguez.

Financiamiento

El trabajo se desarrolló en el marco de un Trabajo Final de Grado bajo cobertura institucional de la FPUNA–UNA, con acceso a la infraestructura del SpaceLab y del GIEM. El hardware de las cuatro unidades del prototipo constituyó la única erogación material directa.