



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

**Departamento:**  
**Departamento de Ingeniería e Investigaciones Tecnológicas**

**Programa de acreditación:**  
**PROINCE**

**Programa de Investigación<sup>1</sup>:**

**Código del Proyecto: C211**

**Título del proyecto**  
**Sistemas de segmento terreno satelital de próxima generación**

**PIDC:**   
Elija un elemento.

**PII:**   
Elija un elemento.

**Director: Ierache, Jorge Salvador**

**Codirector: Soligo, Pablo**

**Integrantes: Pose, Aníbal**

**Investigador Externo, Asesor- Especialista, Graduado UNLaM: Valenti, Maria Cecilia**

**Resolución Rectoral de acreditación: N° 035/2019**

**Fecha de inicio: 01/01/2018**

**Fecha de finalización: 31/12/2019**

---

<sup>1</sup> Los Programas de Investigación de la UNLaM están acreditados con resolución rectoral, según lo indica la Resolución HCS N° 014/15 sobre **Lineamientos generales para el establecimiento, desarrollo y gestión de Programas de Investigación a desarrollarse en la Universidad Nacional de La Matanza**. Consultar en el departamento académico correspondiente la inscripción del proyecto en un Programa acreditado.



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

## A. Desarrollo del proyecto (adjuntar el protocolo)

### A.1. Grado de ejecución de los objetivos inicialmente planteados, modificaciones o ampliaciones u obstáculos encontrados para su realización (desarrolle en no más de dos (2) páginas)

Los objetivos han sido mayormente cumplidos durante los dos años del proyecto. Los resultados parciales han sido publicados en los congresos y workshops detallados en el presente documento (Soligo & Ierache, Arquitectura de segmento terreno satelital adaptada para el control de límites de telemetría dinámicos, 2019), (Soligo & Ierache, Segmento Terreno Para Misiones Espaciales de Próxima Generación) y (Soligo & Ierache, Software de segmento terreno de próxima generación, 2018). El trabajo final ha sido presentado en el “2nd International Academy of Astronautics Latin American Symposium on Small Satellites: Advanced Technologies and Distributed Systems” durante noviembre del 2019 (Soligo & Ierache, Experiences and lessons learned developing a next-generation ground segment prototype, 2019).

El siguiente cuadro muestra los objetivos específicos planteados y su grado de ejecución.

Objetivo	Cumplimiento		
	Total	Parcial	No cumplido
Consumir, decodificar y publicar telemetría de al menos 3 misiones satelitales	X		
Realizar los procesos utilizando una misma herramienta multimisión.	X		
Utilizar únicamente lenguajes e intérpretes de propósito general.	X		
Utilizar exclusivamente frameworks y herramientas de persistencia COTS.	X		
Sumar a la UNLAM como miembro activo de la comunidad SatNOGS ( <a href="https://satnogs.org/">https://satnogs.org/</a> ), ya sea como proveedor y/o como consumidor de datos de misiones satelitales.		X	

Para las misiones satélites el objetivo se ha sobre cumplido, el sistema se ha demostrado adaptable a telemetría de 4 misiones (SAC-D, TITA, Lituunicasat 2 y FS2017).

La desnormalización de datos que permita el almacenamiento y recuperación de datos con tiempos relativamente determinísticos (Morel, Garcia, Palsson, & Gil, 2010) no ha sido desarrollada en favor de investigaciones que permitan el uso de la minería de datos para la predicción de anomalías.

Como ampliación, fue necesaria una integración parcial con el producto NASA OPENMCT a fin de tener una interfaz de usuario en el prototipo. El desarrollo de interfaces de acceso web multiplataformas se relevó excesivamente costoso optando por una integración con productos open source publicados por NASA (<https://nasa.github.io/openmct/>).

La integración con la red SatNOGS ha sido cumplida parcialmente. El prototipo desarrollado consume datos de la organización descargados manualmente. La subida de datos provenientes de la estación terrena UNLaM es aún un punto pendiente y su implementación queda fuera de la capacidad del presupuesto asignado. Se realizaron pruebas exploratorias con resultados parciales.



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

## B. Principales resultados de la investigación

### B.1. Publicaciones en revistas (informar cada producción por separado)

Artículo 1:	
Autores	<i>Pablo Soligo, Jorge Salvador Ierache</i>
Título del artículo	<i>Software de segmento terreno de próxima generación</i>
N° de fascículo	-
N° de Volumen	-
Revista	<i>Libro de Actas XXIV Congreso Argentino de Ciencias de la Computación CACIC 2018</i>
Año	<i>2018</i>
Institución editora de la revista	<i>Facultad de Ciencias Exactas - Universidad Nacional del Centro de la Provincia de Buenos Aires</i>
País de procedencia de institución editora	<i>Argentina</i>
Arbitraje	SI
ISBN:	<i>978-950-658-472-6</i>
URL de descarga del artículo	<a href="http://cacic2018.exa.unicen.edu.ar/pdf/LibroDeActasCACIC2018.pdf">http://cacic2018.exa.unicen.edu.ar/pdf/LibroDeActasCACIC2018.pdf</a>
N° DOI	-



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

Artículo 2:	
Autores	<i>Pablo Soligo, Jorge Salvador Ierache</i>
Título del artículo	<i>Segmento Terreno Para Misiones Espaciales de Próxima Generación</i>
N° de fascículo	-
N° de Volumen	-
Revista	<i>XXI Workshop de Investigadores en Ciencias de la Computación</i>
Año	2019
Institución editora de la revista	<i>Facultad de Ciencias Exactas, Físicas y Naturales Universidad Nacional de San Juan</i>
País de procedencia de institución editora	<i>Argentina</i>
Arbitraje	SI
ISBN:	978-987-3984-85-3
URL de descarga del artículo	<a href="http://www.wicc2019.unsj.edu.ar/descargas/Libro_WICC2019.pdf">http://www.wicc2019.unsj.edu.ar/descargas/Libro_WICC2019.pdf</a>
N° DOI	-

Artículo 3:	
Autores	<i>Pablo Soligo, Jorge Salvador Ierache</i>
Título del artículo	<i>Arquitectura de segmento terreno satelital adaptada para el control de límites de telemetría dinámicos</i>
N° de fascículo	-
N° de Volumen	-
Revista	<i>Libro de Actas XXV Congreso Argentino de Ciencias de la Computación CACIC 2019</i>
Año	2019
Institución editora de la revista	<i>Facultad de Ciencias Exactas, Físico-Químicas y Naturales de la Universidad Nacional de Río Cuarto</i>
País de procedencia de institución editora	<i>Argentina</i>
Arbitraje	SI
ISBN:	<i>En Prensa</i>
URL de descarga del artículo	-
N° DOI	-



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

## B.2. Libros

Libro 1	
Autores	
Título del Libro	
Año	
Editorial	
Lugar de impresión	
Arbitraje	Elija un elemento.
ISBN:	
URL de descarga del libro	
N° DOI	

## B.3. Capítulos de libros

Autores	
Título del Capítulo	
Título del Libro	
Año	
Editores del libro/Compiladores	
Lugar de impresión	
Arbitraje	Elija un elemento.
ISBN:	
URL de descarga del capítulo	
N° DOI	

## B.4. Trabajos presentados a congresos y/o seminarios



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

<b>Autores</b>	<i>Pablo Soligo, Jorge Salvador Ierache</i>
<b>Título</b>	<i>Segmento Terreno Para Misiones Espaciales de Próxima Generación</i>
<b>Año</b>	<i>2019</i>
<b>Evento</b>	<i>XXI Workshop de Investigadores en Ciencias de la Computación</i>
<b>Lugar de realización</b>	<i>Facultad de Ciencias Exactas, Físicas y Naturales Universidad Nacional de San Juan</i>
<b>Fecha de presentación de la ponencia</b>	<i>26/04/2019</i>
<b>Entidad que organiza</b>	<i>Facultad de Ciencias Exactas, Físicas y Naturales Universidad Nacional de San Juan</i>
<b>URL de descarga del trabajo</b> (especificar solo si es la descarga del trabajo; formatos pdf, e-pub, etc.)	<a href="http://www.wicc2019.unsj.edu.ar/descargas/Libro_WICC2019.pdf">http://www.wicc2019.unsj.edu.ar/descargas/Libro_WICC2019.pdf</a>

<b>Autores</b>	<i>Pablo Soligo, Jorge Salvador Ierache</i>
<b>Título</b>	<i>Software de segmento terreno de próxima generación</i>
<b>Año</b>	<i>2018</i>
<b>Evento</b>	<i>Congreso Argentino de Ciencias de la Computación 2018</i>
<b>Lugar de realización</b>	<i>Tandil, Buenos Aires, Argentina.</i>
<b>Fecha de presentación de la ponencia</b>	<i>10/10/2018</i>
<b>Entidad que organiza</b>	<i>Facultad de Ciencias Exactas - Universidad Nacional del Centro de la Provincia de Buenos Aires</i>
<b>URL de descarga del trabajo</b> (especificar solo si es la descarga del trabajo; formatos pdf, e-pub, etc.)	



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

<b>Autores</b>	<i>Pablo Soligo, Jorge Salvador Ierache</i>
<b>Título</b>	<i>Arquitectura de segmento terreno satelital adaptada para el control de límites de telemetría dinámicos</i>
<b>Año</b>	<i>2019</i>
<b>Evento</b>	<i>Congreso Argentino de Ciencias de la Computación 2019</i>
<b>Lugar de realización</b>	<i>Río Cuarto, Córdoba, Argentina</i>
<b>Fecha de presentación de la ponencia</b>	<i>17/10/2019</i>
<b>Entidad que organiza</b>	<i>Universidad Nacional de Río Cuarto</i>
<b>URL de descarga del trabajo</b> (especificar solo si es la descarga del trabajo; formatos pdf, e-pub, etc.)	<i>En prensa.</i>

<b>Autores</b>	<i>Pablo Soligo, Jorge Salvador Ierache</i>
<b>Título</b>	<i>Experiences and lessons learned developing a next-generation ground segment prototype</i>
<b>Año</b>	<i>2019</i>
<b>Evento</b>	<i>2nd International Academy of Astronautics Latin American Symposium on Small Satellites: Advanced Technologies and Distributed Systems</i>
<b>Lugar de realización</b>	<i>Centro Cultural de la Ciencia. Buenos Aires, Argentina.</i>
<b>Fecha de presentación de la ponencia</b>	<i>12/11/2019</i>
<b>Entidad que organiza</b>	<i>International Academy of Astronautics, Instituto Colomb, UNSaM.</i>
<b>URL de descarga del trabajo</b> (especificar solo si es la descarga del trabajo; formatos pdf, e-pub, etc.)	<i>En prensa.</i>



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

## B.5. Otras publicaciones

Autores	
Año	
Título	
Medio de Publicación	

**C. Otros resultados. Indicar aquellos resultados pasibles de ser protegidos a través de instrumentos de propiedad intelectual, como patentes, derechos de autor, derechos de obtentor, etc. y desarrollos que no pueden ser protegidos por instrumentos de propiedad intelectual, como las tecnologías organizacionales y otros. Complete un cuadro por cada uno de estos dos tipos de productos.**

C.1. Títulos de propiedad intelectual. Indicar: Tipo (marcas, patentes, modelos y diseños, la transferencia tecnológica) de desarrollo o producto, Titular, Fecha de solicitud, Fecha de otorgamiento

Tipo	Titular	Fecha de Solicitud	Fecha de Emisión

C.2. Otros desarrollos no pasibles de ser protegidos por títulos de propiedad intelectual. Indicar: Producto y Descripción.

Producto	Descripción
UGS (UNLaM Ground Segment)	Prototipo de software de segmento terreno multimisión. En versión actual ingiere, interpreta, calibra, almacena y permite la visualización de telemetría de varias misiones satelitales. Recientemente Incorpora una primera aproximación a la minería de datos y el aprendizaje automático con el objetivo de detectar comportamientos anómalos en los sistemas.



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

**D. Formación de recursos humanos. Trabajos finales de graduación, tesis de grado y posgrado. Completar un cuadro por cada uno de los trabajos generados en el marco del proyecto.**

**D.1. Tesis de grado**

Director (apellido y nombre)	y Autor (apellido y nombre)	Institución	Calificación	Fecha /En curso	Título de la tesis

**D.2 Trabajo Final de Especialización**

Director (apellido y nombre)	y Autor (apellido y nombre)	Institución	Calificación	Fecha /En curso	Título del Trabajo Final

**D.2. Tesis de posgrado: Maestría**

Director (apellido y nombre)	y Tesista (apellido y nombre)	Institución	Calificación	Fecha /En curso	Título de la tesis

**D.3. Tesis de posgrado: Doctorado**

Director (apellido y nombre)	y Tesista (apellido y nombre)	Institución	Calificación	Fecha /En curso	Título de la tesis

**D.4. Trabajos de Posdoctorado**

Director (apellido y nombre)	Posdoctorando (apellido y nombre)	Institución	Calificación	Fecha /En curso	Título del trabajo	Publicación

**E. Otros recursos humanos en formación: estudiantes/ investigadores (grado/posgrado/ posdoctorado)**

Apellido y nombre del Recurso Humano	Tipo	Institución	Período (desde/hasta)	Actividad asignada <sup>2</sup>

<sup>2</sup> Descripción de la/s actividad/es a cargo (máximo 30 palabras)



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

**F. Vinculación<sup>3</sup>:** Indicar conformación de redes, intercambio científico, etc. con otros grupos de investigación; con el ámbito productivo o con entidades públicas. Desarrolle en no más de dos (2) páginas.

Dentro de las actividades de vinculación e intercambio se destacan.

1. Dictado de M4 (Lenguajes y Sistemas Operativos de Aplicación Espacial) para la CONAE (Comisión Nacional de Actividades Espaciales). Durante el segundo cuatrimestre del 2019 se dictó la capacitación para más de 30 alumnos con una carga horaria total de 40 horas presenciales. Se impartieron conceptos teóricos y prácticos del desarrollo de software en general y de su implementación en el área espacial en particular. El curso consta de unidades dedicadas al desarrollo de software sobre Python, lenguaje C y sistemas operativos de tiempo real. Los módulos dedicados a lenguajes de definición de hardware no han sido dictados por limitaciones de tiempo. Durante el dictado del curso se abordaron problemáticas vinculadas al desarrollo de software en el sector, alternativas, evaluaciones y soluciones posibles. Durante el curso se trabajó intensivamente con conjuntos de datos derivados de las investigaciones del proyecto. Se aprovechó la oportunidad para intercambiar experiencias y lecciones aprendidas con equipos técnicos de CONAE y para intercambiar datos e información sensible a las investigaciones en curso y futuras. Para el dictado del curso fue necesario la generación de material adicional, filmas, ejercicios, ejemplos, juegos de datos, proyectos y maquinas de virtuales de uso didáctico. El material queda disponible para futuros cursos o como material de la materia en futuras cohortes de MDIAE.
2. Otras actividades de vinculación:
  - a. International Space Forum at Ministerial Level, the Latin American and Caribbean Chapter – ISF Buenos Aires (<http://www.iafastro.org/events/international-space-forum-the-latin-american-chapter-isfbuenosaires/>).
  - b. PyDay 2018: Presentación “Desarrollo de un segmento terreno satelital de próxima generación con python” (<https://pydaylp.python.org.ar/#/#agenda>)
  - c. UNSaM: Conferencia “TUB and space. Satellite Missions from TUB” (<http://www.unsam.edu.ar/institutos/Colomb/tub.asp>).
  - d. UNSaM: Workshop “Operation of University Satellite Missions. Ground Stations. Communications and Antennas” (<http://www.unsam.edu.ar/institutos/Colomb/tub.asp>).

---

<sup>3</sup> Entendemos por acciones de “vinculación” aquellas que tienen por objetivo dar respuesta a problemas, generando la creación de productos o servicios innovadores y confeccionados “a medida” de sus contrapartes.



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

**G. Otra información. Incluir toda otra información que se considere pertinente.**

La exposición CACIC 2019 fue premiada como la mejor del workshop para la jornada.  
[https://cacic2019.exa.unrc.edu.ar/workshops/innovacion\\_en\\_sistemas\\_de\\_software](https://cacic2019.exa.unrc.edu.ar/workshops/innovacion_en_sistemas_de_software)

**Referencias**

- Morel, T., Garcia, G., Palsson, M., & Gil, J. C. (2010). High Performance Telemetry Archiving and Trending for Satellite Control Centers. *SpaceOps 2010 Conference Delivering on the Dream Hosted by NASA Marshall Space Flight Center and Organized by AIAA*, (pág. 2111).
- Soligo, P., & Ierache, J. S. (2018). Software de segmento terreno de próxima generación. *XXIV Congreso Argentino de Ciencias de la Computación (La Plata, 2018)*.
- Soligo, P., & Ierache, J. S. (2019). Arquitectura de segmento terreno satelital adaptada para el control de límites de telemetría dinámicos. *XXV Congreso Argentino de Ciencias de la Computación (La Plata, 2019)*.
- Soligo, P., & Ierache, J. S. (2019). Experiences and lessons learned developing a next-generation ground segment prototype. *2nd International Academy of Astronautics Latin American Symposium on Small Satellites: Advanced Technologies and Distributed Systems (November 11-16, 2019)*.
- Soligo, P., & Ierache, J. S. (s.f.). Segmento Terreno Para Misiones Espaciales de Próxima Generación. *WICC 2019*.



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

## H. Cuerpo de anexos:

### Anexo I

#### Certificados y trabajos



Se certifica que  
**Jorge Ierache**  
ha participado en calidad de Autor del artículo  
"Software de segmento terreno de proxima  
generacion" aceptado en el IX Workshop  
Procesamiento de Señales y Sistemas de Tiempo  
Real (WPSTR),  
realizado en la ciudad de Tandil del 8 al 12 de  
octubre de 2018.

Mg. Claudio Aciti  
Comité organizador  
CACIC 2018

Lic. Patricia Pesado  
coordinadora titular  
RedUNCI



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019



Se certifica que  
**Pablo Soligo**  
ha participado en calidad de Autor del artículo  
"Software de segmento terreno de proxima  
generacion Camera Ready" aceptado en el IX  
Workshop Procesamiento de Señales y Sistemas de  
Tiempo Real (WPSTR),  
realizado en la ciudad de Tandil del 8 al 12 de  
octubre de 2018.

Mg. Claudio Aciti  
Comité organizador  
CACIC 2018

Lic. Patricia Pesado  
coordinadora titular  
RedUNCI



FEDERACION DE CIENCIAS  
**EXACTAS**  
INSTITUTO ARGENTINO DE CIENCIAS  
EXACTAS Y TECNICAS





<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

# Universidad Nacional de Río Cuarto

FACULTAD DE CIENCIAS EXACTAS, FÍSICO-QUÍMICAS Y NATURALES  
DEPARTAMENTO DE COMPUTACIÓN

Por cuanto

**SOLIGO, Pablo - DNI/Pasaporte 25127140**

ha **PRESENTADO** el trabajo "*Arquitectura de segmento terreno satelital adaptada para el control de límites de telemetría dinámicos*", en el **XXV Congreso Argentino de Ciencias de la Computación - CACIC 2019**, organizado por la Red de Universidades Nacionales con carreras en Informática (RedUNCI) y el Departamento de Computación de la Facultad de Ciencias Exactas, Físico-Químicas y Naturales de la Universidad Nacional de Río Cuarto, realizado en la ciudad de Río Cuarto entre los días 14 y 18 de octubre de 2019.

Por tanto, conferimos y firmamos la presente constancia. Dado en la ciudad de Río Cuarto, el día 18 de octubre de 2019.



Dr. MARÍA MARÍA REYNOSO  
Secretaría Fac. Cs. Exactas Físico-Quím. y Nat.



Dr. MARÍA ROVERA  
Secretaría Fac. Cs. Exactas Físico-Quím. y Nat.



Certificado generado según Resolución Rectoral 446/18 - UNRC  
Verificar en: Departamento de Computación, FCEQyN, Universidad Nacional de Río Cuarto.  
secretaria@dc.exa.unrc.edu.ar - +54 0358 4676235  
[https://dc.exa.unrc.edu.ar/certificados/cacic2019/congr\\_expo/050214c11d5348375ad502e5921c5284442aed2b.pdf](https://dc.exa.unrc.edu.ar/certificados/cacic2019/congr_expo/050214c11d5348375ad502e5921c5284442aed2b.pdf)

Firmado digitalmente por: ROVERA Marisa  
Motivo: Oficina FCEQyN UNRC  
Localización: Río Cuarto  
Fecha y hora: 26.12.2019 20:10:45

Firmado digitalmente por: BEASSONI Paola Rita  
Motivo: Secretaría Técnica FCEQyN UNRC  
Localización: Río Cuarto  
Fecha y hora: 26.12.2019 20:28:17



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

# Universidad Nacional de Río Cuarto

FACULTAD DE CIENCIAS EXACTAS, FÍSICO-QUÍMICAS Y NATURALES

DEPARTAMENTO DE COMPUTACIÓN

Por cuanto

**SOLIGO, Pablo - DNI/Pasaporte 25127140**

ha **ASISTIDO** al **XXV Congreso Argentino de Ciencias de la Computación - CACIC 2019**, organizado por la Red de Universidades Nacionales con carreras en Informática (RedUNCI) y el Departamento de Computación de la Facultad de Ciencias Exactas, Físico-Químicas y Naturales de la Universidad Nacional de Río Cuarto, realizado en la ciudad de Río Cuarto entre los días 14 y 18 de octubre de 2019.

Por tanto, conferimos y firmamos la presente constancia. Dado en la ciudad de Río Cuarto, el día 18 de octubre de 2019.



Dr. MARÍA INÉS RETNOSO  
Directora de Computación



Dr. MARÍA ROVERA  
Directora de Ciencias Físico-Químicas y Naturales



Certificado generado según Resolución Rectoral 446/18 - UNRC  
Verificar en: Departamento de Computación, FCEQyN, Universidad Nacional de Río Cuarto.  
secretaria@dc.exa.unrc.edu.ar, +54 0358 4676235  
[https://dc.exa.unrc.edu.ar/certificados/cacic2019/congr\\_asistencia/c38bae7380f3610e9a458cfbf06680f641a38.pdf](https://dc.exa.unrc.edu.ar/certificados/cacic2019/congr_asistencia/c38bae7380f3610e9a458cfbf06680f641a38.pdf)

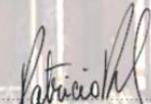


<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019



## XXI Workshop de Investigadores en Ciencias de la Computación

Se certifica que Pablo Soligo ha participado en calidad de Autor del artículo Segmento Terreno Para Misiones Espaciales de Próxima Generación aceptado en el marco del XXI Workshop de Investigadores en Ciencias de la Computación (WICC 2019) realizado en la ciudad de San Juan entre los días 25 y 26 de abril de 2019.

  
Lic. Patricia PESADO  
Coordinador Titular  
Red UNCI

  
Ing. Rodolfo BLOCH  
DECANO  
Facultad de Ciencias Exactas  
Físicas y Naturales



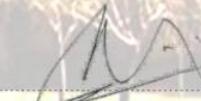
<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019



## XXI Workshop de Investigadores en Ciencias de la Computación

Se certifica que Jorge Ierache ha participado en calidad de Autor del artículo Segmento Terreno Para Misiones Espaciales de Próxima Generación aceptado en el marco del XXI Workshop de Investigadores en Ciencias de la Computación (WICC 2019) realizado en la ciudad de San Juan entre los días 25 y 26 de abril de 2019.

  
Lic. Patricia PESADO  
Coordinador Titular  
Red UNCI

  
Ing. Rodolfo BLOCH  
DECANO  
Facultad de Ciencias Exactas  
Físicas y Naturales



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLAM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

## Software de segmento terreno de próxima generación

Pablo Soligo, Jorge Salvador Ierache

Universidad Nacional de La Matanza  
Departamento de Ingeniería e Investigaciones Tecnológicas.  
Florencio Varela 1903, B1754JEC San Justo, Buenos Aires.  
[psoligo@unlam.edu.ar](mailto:psoligo@unlam.edu.ar)  
<http://www.unlam.edu.ar/>

**Resumen** Este trabajo presenta los resultados de las experiencias obtenidas en el desarrollo de la primera versión de un prototipo funcional de segmento terreno de próxima generación. El proyecto fue inicialmente desarrollado como parte de la materia “proyecto integrador” de la Maestría en Desarrollos Informáticos de Aplicación Espacial (MDIAE) y evolucionado por el grupo de investigación aplicada de la Universidad Nacional de La Matanza (UNLAM). Las unidades de software desarrolladas permiten la ingestión y decodificación de telemetría de múltiples misiones con independencia del origen de datos y el comando y control de satélites. Se aplicaron técnicas y herramientas como camino alternativo a las tecnologías propietarias para desarrollar un entorno multimisión y multiplataforma. Como segmento de vuelo se ha utilizado el satélite de formación 2017 (FS2017) [1] y datos de misiones de terceros (Satnogs).

**Keywords:** Satelites Artificiales, Segmento Terreno, Diseño de Software, Tecnologia Espacial, Sistemas Distribuidos, Telemetría, Telecomandos

### 1. Introducción

El presente trabajo detalla las experiencias y resultados en el desarrollo de un prototipo de sistema de segmento terreno satelital costo-efectivo. El diseño e implementación del prototipo tiene como raíz las experiencias realizadas durante las maestrías con sistemas de segmentos terrenos actualmente en producción y su vinculación con los sistemas de software de propósito general de uso extendido en la industria.

Se priorizo el desarrollo sobre herramientas de amplia aceptación y de probada robustez en contraste con soluciones comunes en la industria espacial, las cuales en algunos casos muestra un exceso de conservadurismo [2]. Parte del objetivo de este trabajo es confrontar dichas implementaciones, con predominancia en tres dominios:

- **Lenguajes de decodificación, comando y control:** Se propone el uso de lenguajes e intérpretes de propósito general para la decodificación de telemetría y el desarrollo de scripts de comandos.



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

2 Software de segmento terreno de próxima generación.

- **Persistencia<sup>1</sup> de datos:** Se plantea una arquitectura donde la persistencia de datos se realiza sobre motores de base de datos relacionales y su acceso es siempre mediante un ORM(Object Relational Mapper).
- **Interfaces:** Las interfaces con sistemas internos y externos, tanto para la comunicación de módulos como para publicación e ingestión de datos se realiza mediante protocolos extendidos y aceptados.

Se marginan de la solución, estándares, formatos o protocolos *ad-hoc* o de escasa penetración en la industria de software de uso masivo y se prioriza las soluciones, bibliotecas, componentes y frameworks COTS(Commercial Off-The-Shelf). Los lineamientos de diseño adhieren a los manuales de buenas practicas del área espacial [3], casi todas ellas comunes y aplicables a las buenas practicas conocidas en la industria de software general.

## 2. Lenguajes de decodificación, comando y control

Una practica común, en el área espacial, cuando se trata de comandar un satélite, es el uso de secuencias de comandos sobre estructuras de control. Empresas y agencias espaciales en todo el mundo han desarrollado sus propios lenguajes e intérpretes para cumplir con este objetivo [4]:

- STOL: Satellite Test and Operation Language. Desarrollado por la Nasa y ampliamente utilizado en varias misiones.
- PLUTO: presente en algunas misiones de la ESA (Satellite Control and Operation System 2000).
- Otros: desarrollados o utilizados por diferentes compañías SOL(GMV), CCL (Harris), PIL (Astrium), SCL (ICS).

Estos lenguajes son típicamente propietarios e incompatibles entre sí [5], dificultando las migraciones entre sistemas y la posibilidad de compartir procedimientos entre distintas misiones.

El FS2017 [1] fue el primer segmento de vuelo utilizado como prueba experimental para el desarrollo de los sistemas planteados en este trabajo y se optó por usar un lenguaje de propósito general en lugar de crear un lenguaje específico o utilizar los existentes en CONAE (Comisión Nacional de Actividades Espaciales).

Este enfoque presenta múltiples ventajas, el cuadro 1 compara los pros y contras de cada tipo de implementación.

En el caso del proyecto integrador la opción de lenguaje propio de la misión se descartó por no disponer del tiempo que implica desarrollar y validar un intérprete de propósito específico. La opción de utilizar los intérpretes de CONAE quedó relegada por sobre la opción Python, con una base de usuarios mucho más grande. El índice de popularidad TIOBE [6] ubica al lenguaje en la posición 5. Incluso en su versión más popular, las alternativas de propósito específico, al ser parte de una única o un grupo de organizaciones siquiera figuran

<sup>1</sup> Se refiere a la propiedad de los datos para que estos sobrevivan de alguna manera.



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

**Cuadro 1.** Comparativo Lenguajes Propósito específico vs Propósito General

<b>Propósito específico.</b>	<b>Propósito general.</b>
<b>Ventajas</b> Pueden ser más amigables para usuarios no programadores. Pueden presentar adaptaciones específicas a los problemas de operación del satélite. Pueden presentar adaptaciones específicas a los problemas de operación del satélite.	<b>Desventajas.</b> Es una tecnología propietaria. Problemas de portabilidad. Requiere desarrollo propio. Requiere mantenimiento y las mejoras o ampliaciones son costosas. Muy baja base de usuarios.
<b>Desventajas</b> Demasiado poderoso. Menos amigable a la lectura.	<b>Ventajas.</b> Portable. Mas poderoso que las versiones de propósito específico. Gran cantidad de usuarios.

entre las primeras 50.

Utilizando Python también se consigue generar scripts multiplataforma, mejores capacidades de depuración, documentación disponible en la comunidad y un lenguaje de dominio de todos los estudiantes.

El uso de los interprete disponibles en CONAE sufre de las mismas restricciones que cualquier lenguaje de uso interno y de propósito específico, una base de usuarios restringida a un grupo acotado y finito, documentación limitada, capacidades circunscritas del lenguaje y las herramientas, pero por sobre todo, y quizá lo más evidente, es la dificultad o el esfuerzo que implica de adaptar el intérprete a las nuevas plataformas como indica [4].

### 3. Persistencia de datos

Las definiciones de los datos (Alarmas, variables de telemetría, comandos disponibles, etc.) como los datos en sí, en cualquier nivel de procesamiento, están administrados por un motor de base de datos relacional. Para la telemetría satelital existen entidades para el almacenamiento de los datos crudos, independientemente del satélite al que pertenezcan, la fuente o el origen de la ingestión. A medida que estos datos son procesados pasan a tablas donde están las variables de ingeniería ya procesadas y disponibles para cualquier modulo que las necesite, incluyendo el procesador de comandos.

Previendo para futuras versiones la implantación de tablas que permitan el acceso eficiente a datos históricos, KPIs (Key Performance Indicators) y desnormalizaciones que permitan el almacenamiento de largo plazo.

Todo el acceso a datos se realiza exclusivamente a través de un ORM, el uso de esta capa intermedia ofrece además de una mejora en la productividad, una independencia del proveedor del motor de base de datos.

La solución propuesta en términos de persistencia, prácticamente un estándar de facto en la industria del software, aun no tiene presencia en muchos centros



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

4 Software de segmento terreno de próxima generación.

de control de misión. Su incorporación, exploración y dominio representan un valor agregado al trabajo.

#### 4. Interfaces

En términos de interfaces tanto internas como externas en el área espacial abundan las soluciones *ad-hoc* o implementaciones sobre tecnologías de escasa penetración.

Un caso remarcable es el de CORBA [7] (Common Object Request Broker Architecture), que, si bien es un estándar propuesto por varios jugadores de la industria del software, no ha sido adoptado masivamente y se vio rápidamente relegado por implementaciones basadas en HTTP (Hypertext Transfer Protocol) [8].

Las interfaces del sistema propuesto se basan completamente en esta última tecnología, por ejemplo, toda ingestión de telemetría se realiza mediante un servicio REST que puede ser consumido tanto por módulos de terceros como por módulos internos que adapten formatos, tecnologías o características de antiguas implementaciones.

La figura 1 muestra como los adaptadores ajustan dos tecnologías diferentes (TCP/IP y FTP) a una entrada normalizada sobre servicios REST. Así mismo la evolución del sistema a una arquitectura distribuida prevé distribución, sincronización y comunicación de tareas mediante brokers basados también en http.

#### 5. Arquitectura

La solución propuesta para el segmento terreno evoluciona desde su concepto cliente-servidor tradicional a una arquitectura de sistema distribuido multiprocesador. Un subconjunto finito de procesos se ejecuta de manera distribuida dentro de la infraestructura de hardware disponible. La tabla 2 enumera algunos de los servicios a ejecutar.

Esta arquitectura permite escalar horizontalmente en función de las misiones que se puedan ir sumando al centro de control. La figura 1 muestra un diagrama de alto nivel donde se pueden observar adaptadores que ajustan la telemetría provista por distintas tecnologías a un estándar REST (Representational state transfer). Los procesos 1, 2 ... n son los especificados en la tabla 2.

En términos de desarrollo el lenguaje utilizado es Python, normalizando en un único lenguaje el desarrollo de los módulos de software, los scripts de comandos y las funciones de calibración.

El framework utilizado es Django, lo que motiva un análisis y diseño orientado a objetos con un alta productividad tanto desde el framework como en el lenguaje [10].

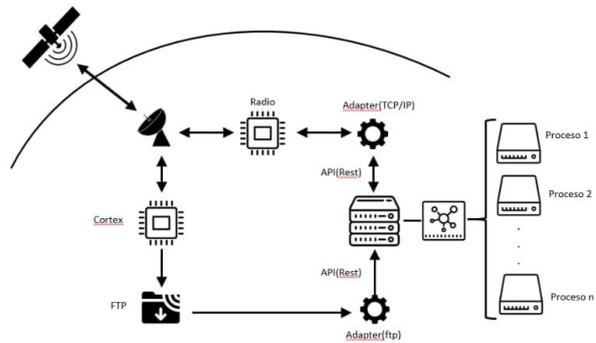
A esto último se debe agregar la capacidad de incorporar a los scripts de comandos y de calibración el acceso a herramientas de comunicación como Sockets,



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

**Cuadro 2.** Servicios distribuidos

Proceso	Descripción
Central.	Servicio central, responsable de ofrecer las interfaces, tanto sea de usuario como las API de comunicación.
Procesador de Telemetría (n).	Procesos responsables de la decodificación de la telemetría cruda.
Procesador de Telecomandos (n)	Procesos responsables de la ejecución de comandos
Servicios CODS	Responsable de generar y actualizar las efemérides, eclipses y oportunidades de contacto (Pasadas).
Procesos de datos	Procesamiento de datos, desnormalizaciones, consolidados, estadísticas y tendencias[9].
Procesos de calibración	Verificar novedades en funciones de calibración.
Otros	Procesos adicionales a ser relevados.



**Figura 1.** Arquitectura conceptual



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

6 Software de segmento terreno de próxima generación.

http, RPC (Remote Procedure Call), Web Services, email y a bibliotecas de manejo de archivos XML entre otros como explica [4].

## 6. Telemetría

Si bien la arquitectura del segmento terreno es agnóstica de la fuente de datos se han utilizado como segmento de vuelo el FS2017 y datos de la misión LITUANICASAT 2. El FS2017 envía la telemetría en tramas AX.25 que son recibidas por un equipo de radio y publicadas por TCP/IP. La trama tiene un fragmento de Payload donde viene codificada la telemetría. La telemetría como su definición (Tipo, límites, rangos, ubicación) están almacenadas en tablas de un motor de base de datos relacional. Es común en este tipo de sistemas recibir los valores crudos o *raw*, para poder transformar esta telemetría cruda a valores de ingeniería es necesario someterlos a una función que realice la conversión, la cual también podría cambiar en función del desgaste o recalibración del sensor.

Estos ajustes pueden ir desde un simple ajuste por función lineal hasta una discretización de valores por tablas de *look-up*. El sistema debe ser lo suficientemente flexible como para permitir aplicar cualquier función de transformación a toda variable de telemetría y permitir ajustarla en el tiempo.

En lugar de utilizar interpretes de desarrollo propio, la solución propuesta, utiliza técnicas de reflexión de software para cargar en tiempo de ejecución la función seleccionada de calibración y aplicarla al valor *raw* extraído de la trama de telemetría.

La función de calibración puede ser cualquier secuencia de comandos programable en python sin ningún tipo de restricción más allá del tiempo de procesamiento. Todas las bibliotecas y estructuras de datos están disponibles incluyendo funciones matemáticas e incluso el acceso a la base de datos completa mediante ORM. Al tener acceso a la base de datos se pueden obtener coeficientes actualizados de calibración permitiendo:

- **Reutilizar funciones:** Es posible crear una única función para ajustes típicos (lineal, cuadrático) y reutilizarla con distintos coeficientes según la variable de telemetría requiera
- **Calibración fina:** Creada una función es posible modificar los coeficientes para a medida que el sensor presenta desgaste.

Para que un método sea considerado de calibración o ajuste debe estar desarrollado como método de una clase heredada de *BaseCalibration*. El software

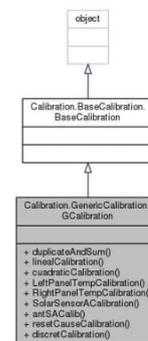


Figura 2. Calibración



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

realiza periódicamente una exploración de todos los métodos públicos de clases derivadas (Ver procesos tabla 2) de *BaseCalibration* y los disponibiliza para su aplicación a las distintas variables de telemetría. La figura 2 muestra la clase *GCalibration*, heredada de *BaseCalibration*, donde se implementan algunos métodos de calibración incluido el mencionado *linealCalibration*.

El siguiente fragmento de código muestra parte de la implementación del método, encargado de llamar a la función de calibración para todo nuevo valor *raw*.  
*Calibración de variable raw*

```
class TlmyVar(models.Model):
    def setValue( self,
raw,
saveifchange=False,
dt=datetime.now(utc)):

    value = self.tlmyVarType.getValue()
    if (raw!=self.tlmyVarType.lastRawValue) or
        (self.tlmyVarType.lastUpdate==None):
        self.tlmyVarType.lastRawValue = raw
    if self.tlmyVarType.calibrationMethod:
    if not self.tlmyVarType.calibrationLogic:
        klass =
            globals()[self.tlmyVarType.calibrationMethod.aClass]

        instance = klass()

        methodToCall =
            getattr(instance,
                self.tlmyVarType.calibrationMethod.aMethod)
        self.tlmyVarType.calibrationLogic = methodToCall
    else:
        pass #Calibracion ya cargada

    ...
```

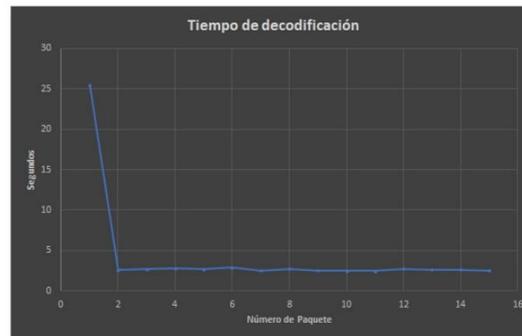
(Extracción parcial del método setValue)

Si el valor *raw* para una variable determinada no cambió desde la última actualización y tampoco se recargo la función, entonces se guarda el registro de la recepción, pero no se aplican las calibraciones. Esto último hace ganar eficiencia ya que no se vuelve a transformar cuando se sabe que el resultado será el mismo. La carga del método de calibración a un atributo del tipo de variable de telemetría se realiza, si no ocurren cambios, una única vez durante la ejecución. Esto también permite mejorar los tiempos de ejecución ya que evita la carga por cada valor de telemetría recibido. Con una frecuencia configurable, el software analiza si alguna función de calibración fue actualizada, si así ocurriera se procede al



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

8 Software de segmento terreno de próxima generación.



**Figura 3.** Tiempo de decodificación

limpiar el atributo que contiene la función para forzar su recarga.

La figura 3 muestra los tiempos de procesamiento para un conjunto de simulación de 15 paquetes con 5000 variables de telemetría cada uno, cantidad compatible con un satélite científico de envergadura (Ejemplo SABIAMAR <http://www.conae.gov.ar/index.php/espanol/introduccion-sace>).

La primera decodificación demora más que las siguientes dado que tiene que realizar la carga de las funciones de transformación. La primera telemetría recibida difiere de la anterior, ya que al ser la primera calibración la función no está precargada. Luego el tiempo se estabiliza entre 3 y 3.5 segundos para el conjunto de las 5000 variables con una probabilidad de cambio de 10% entre muestra y muestra. El tiempo de procesamiento incluye además la verificación de que los valores de las variables de ingeniería estén dentro del rango de seguridad y el almacenamiento en el histórico. Las pruebas han sido realizadas en máquinas virtuales sobre equipos de escritorio.

## 7. Telecomandos

<sup>2</sup> En el caso del FS2017 los telecomandos deben ser enviados al segmento de vuelo por el mismo canal en donde se recibe la telemetría, TCP/IP puerto 3210. Los comandos deben ser codificados en una trama AX.25. Para permitir la creación de scripts de comandos, de la misma forma que con la telemetría (6) se utilizaran las capacidades de reflexión para analizar, en tiempo de ejecución los scripts a ejecutar. Los scripts de comandos pueden ejecutarse por acción explícita de un operador o porque fueron aplicados a una pasada. El operador tiene accesos a las bibliotecas de comandos y operación, las cuales están desarrolladas sobre el mismo lenguaje e intérprete de la aplicación. Las bibliotecas tienen pleno

<sup>2</sup> Los módulos que procesan comandos se están en proceso de desarrollo



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

acceso al ORM de donde puede obtener el diccionario de comandos y los valores de telemetría si necesitara aplicar condicionales que dependieran del estado del segmento de vuelo u otro valor disponible en el sistema.

## 8. Resultados

El lenguaje (python) es sencillo aprender, pero por sobre todo, está disponible una extensa comunidad con ayudas ante los problemas o desafíos que puedan aparecer. Para las posibles limitaciones es fácil encontrar soluciones alternativas (workarounds). Esto presenta un fuerte contraste con las experiencias realizadas durante la maestría con lenguajes de propósito específico que requirieron de la asistencia constante de un experto.

Las herramientas de depuración y análisis se mostraron poderosas, incluso superando las expectativas más allá de las limitaciones en las capacidades de Code Insight/IntelliSense naturales de un lenguaje interpretado.

La solución fue ejecutada en ubuntu linux 15.10 como en windows 10.0 y windows 7.0. Se realizaron pruebas sobre dispositivos físicos y virtuales. En todos los casos el software se ejecutó sin cambios. En este punto tanto el intérprete como el framework mostraron uno de sus atributos más destacables.

El ORM de Django se mostró estable, se registraron pocos problemas o *bugs* y cuando ocurrieron se encontró información disponible que lo documentaba. Para la creación y modificación de modelos, su flujo dividido en dos etapas denominadas *Make Migrations* y *Migrate* permitió el trabajo colaborativo y con múltiples servidores, demostración empírica de la madurez del framework.

El acceso a datos requiere de la práctica de convenciones propias (*Convention over configuration/coding by convention*) que, aunque muy bien documentadas, afectan la curva de aprendizaje. Por otro lado el modelo no acepta atributos privados, cuestión que afecta el encapsulamiento. La capacidad del ORM para trabajar con polimorfismo está limitada y se requiere añadir paquetes especiales, este último punto se pueden considerar como una limitación de relevancia.

Las capacidades de reflexión fueron suficientes para cumplir con los objetivos y los tiempos de respuesta no presentan una limitación a su implementación. La reutilización de métodos previamente cargados colabora en mantener los tiempos entre márgenes aceptables.

## 9. Conclusiones

Los lenguajes específicos del sector espacial fueron creados hace décadas. Durante los 70, si bien existían opciones interpretadas, éstas no eran de uso extendido. En la actualidad existen varias opciones (Python, Perl, VBScript, JavaScript) con una amplia cantidad de usuarios de base y múltiples proyectos que avalan su robustez lo que plantea un nuevo escenario a evaluar en lo que respecta a scripts de comandos o calibración de telemetría. Las opciones ampliamente probadas no justifican el desarrollo de un lenguaje propietario, no tener una base de usuarios implica ausencia de documentación, soporte, herramientas



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

10 Software de segmento terreno de próxima generación.

y por sobre todo recursos como explica [11] refiriéndose a ADA.

En términos de persistencia de datos, la vinculación entre el ORM y el motor relacional ha favorecido la productividad. Se ha cuidado de eliminar cualquier consulta directa al motor, para favorecer la compatibilidad con múltiples motores.

Las interfaces aun no se han explotado en su totalidad. Se espera su aplicación efectiva durante las integraciones con diferentes actores o socios del proyecto.

En el estado actual, la solución permite cumplir los objetivos haciendo uso únicamente de componentes COTS y sobre herramientas de propósito general y uso extendido en la industria del software.

## Referencias

1. Pablo Soligo, Ezequiel González, Eduardo Sufán, Emmanuel Arias, Ricardo Barbieri, Pablo Estrada, Alfonso Montilla, José Robin, Javier Uranga, M Cecilia Valenti, et al. Misión cubesat fs2017: Desarrollo de software para una misión satelital universitaria. *WICC 2017*, page 843.
2. José J Ramos Pérez. A design for an advanced architecture of satellite ground segments.
3. Ken Galal and Roger P Hogan. Satellite mission operations best practices. 2001.
4. Gonzalo Garcia. Use of python as a satellite operations and testing automation language. In *GSAW2008 Conference, Redondo Beach, California*, 2008.
5. Geraldine Chaudhri, Jim Cater, and Brad Kizzort. A model for a spacecraft operations language. In *SpaceOps 2006 Conference*, page 5708, 2006.
6. TIOBE Software. TIOBE programming community index, september 2017, 2017. [Online; accessed 26-September-2017].
7. L Foti. Corba technology for ground segment system software development. In *DASIA 98-Data Systems in Aerospace*, volume 422, page 303, 1998.
8. Michi Henning. The rise and fall of corba. *Queue*, 4(5):28–34, 2006.
9. Thomas Morel, Gonzalo Garcia, Mike Palsson, and Juan Carlos Gil. High performance telemetry archiving and trending for satellite control centers. In *SpaceOps 2010 Conference Delivering on the Dream Hosted by NASA Marshall Space Flight Center and Organized by AIAA*, page 2111, 2010.
10. Lutz Prechelt. An empirical comparison of seven programming languages. *Computer*, 33(10):23–29, 2000.
11. II Smith et al. What about ada? the state of the technology in 2003. 2003.



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

# Segmento Terreno Para Misiones Espaciales de Próxima Generación

Pablo Soligo y Jorge S. Ierache

Universidad Nacional de La Matanza, [psoligo@unlam.edu.ar](mailto:psoligo@unlam.edu.ar), [jierache@unlam.edu.ar](mailto:jierache@unlam.edu.ar)

## RESUMEN

Las líneas de investigación y desarrollo presentadas tienen por objetivo demostrar la factibilidad de desarrollar sistemas de segmento terreno satelital costo-efectivos (Del inglés cost-effective) utilizando exclusivamente componentes dentro de los denominados, de estantería, en cualquiera de sus variantes OTS (Del Inglés Off-the-Shelf), COTS (Del Inglés Commercial-Off-The-Shelf) y OSS (Del Inglés Open Source Software) prescindiendo de soluciones propias o de herramientas de escasa penetración en la industria de software de propósito general.

**Palabras clave:** *Segmento Terreno, Software, Costo-Efectivo.*

## CONTEXTO

La UNLaM como socio académico de la CONAE dicta una maestría en desarrollos informáticos de aplicación espacial (MDIAE). Las experiencias realizadas en la maestría, tanto de manera directa, operando unidades de software de segmento terreno de varias agencias, como mediante investigación general de las soluciones implementadas en el área, propiciaron la creación de un grupo de investigación en el marco de proyecto "Proince C-211". Este grupo está dedicado a investigar e implementar prototipos de software alternativos de bajo costo basados en las soluciones ampliamente aceptadas, de probada madurez y con penetración en la industria de software de propósito general.

## INTRODUCCIÓN

Aunque no es una regla absoluta y muchas veces el alcance depende de la agencia u organización, la industria espacial muestra un marcado

conservadurismo en las soluciones implementadas [1], [2]. Esta realidad es definida en parte por el alto costo de las misiones y la dificultad (muchas veces imposibilidad) de realizar ajustes o reparaciones, condición que motiva la baja predisposición a tomar riesgos.

El ciclo de vida de una misión espacial está fuertemente guiado por el estricto cumplimiento de los requerimientos (requirement-driven) y esta estrategia dificulta una visión más amplia y de largo plazo [2]. No es difícil encontrar que cada agencia espacial implementa sus propias soluciones *ad-hoc*, normalmente basadas en sus protocolos y en algunos casos utilizando herramientas de escasa penetración o con años de obsolescencia.

## LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

Las líneas de investigación se proponen la exploración de propuestas alternativas basadas exclusivamente en técnicas y herramientas de uso masivo en la industria del software, en particular aplicadas a la industria espacial. El objetivo primario que se persigue es demostrar que las herramientas utilizadas en la industria del software de propósito general son aplicables a la industria espacial, puntualmente, objeto de esta investigación, el segmento terreno. Entendiendo también que la aplicación de estas herramientas colabora en el desarrollo de soluciones de bajo costo y alto nivel de mantenibilidad. Se destacan para el segmento terreno tres áreas principales de trabajo:

### Lenguajes de comando y control.

Se propone el uso de lenguajes e intérpretes de propósito general para la decodificación de telemetría y el desarrollo de scripts de comandos, marginando las soluciones propias de uso común en



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

la industria espacial, como los lenguajes STOL , PLUTO (ESA), lenguajes de propósito específico y normalmente incompatibles entre sí [3] La propuesta tiene como argumento, no solo la estandarización o la posibilidad de compartir procedimientos, los problemas derivados de los cambios de plataforma y los esfuerzos en las adaptaciones quedan parcialmente cubiertos por la comunidad u organización responsable del lenguaje/interprete. Esto último colabora con la reducción de costos y, en algunos casos, permite que los intérpretes puedan ejecutarse en plataformas actualizadas [4] Los recursos disponibles y el tiempo de entrenamiento en herramientas propietarias también es un costo que considerar. Los recursos disponibles y el tiempo de entrenamiento en herramientas propietarias también es un costo que considerar.

#### **Persistencia, recuperación y explotación de datos.**

Se ha optado por la utilización de un motor de base de datos relacional para el almacenamiento y recuperación tanto de los datos de la misión como de su definición. Los sistemas de bases de datos sean relacionales o no, se han transformado en una herramienta de uso extendido abarcando prácticamente todas las áreas de la industria. Desde sencillas base de datos embebidas como Sqlite hasta complejos RDBMS como SqlServer u Oracle. En esta área, si bien existen trabajos y experiencias documentadas [5], [6] y su uso es recomendado [7], todavía existen agencias que deciden almacenar sus datos en archivos con formatos propietarios dificultando el control de integridad, el acceso, la explotación y la compatibilidad entre aplicaciones. Se exploran así las alternativas de bajo costo disponibles, los diseños y las opciones de optimización de los motores (RDBMS) que ofrezcan un rendimiento acorde a los requerimientos de un sistema informático moderno.

Castigando el rendimiento, todo el acceso a los datos se realiza mediante un ORM (Object-relational mapping), en virtud de mantener las aplicaciones tan independientes del motor de la

base de datos (RDBMS) como sea posible y colaborar con ya mencionada reducción de costos.

#### **Interfaces y distribución de procesos**

En términos de interfaces tanto internas como externas en el área espacial se pueden encontrar desde soluciones propietarias hasta la implementación de tecnologías de escasa aceptación como CORBA [8]. Si bien CORBA es un estándar propuesto por varios jugadores importante de la industria del software, no ha sido adoptado masivamente y se vieron relegadas por implementaciones basadas en HTTP (Hypertext Transfer Protocol) [9].

Las interfaces del sistema propuesto se basan completamente en esta última tecnología. Toda ingestión de telemetría se realiza mediante un servicio REST que puede ser consumido tanto por módulos de terceros como por módulos internos que adapten formatos, tecnologías o características de antiguas implementaciones. De manera consistente la publicación de telemetría y datos en general se ofrece también utilizando HTTP lo que ha permitido la integración con herramientas como el NASA OPENMCT [10] (<https://nasa.github.io/openmct/>). (Véase Ilustración 1 - Datos de la misión argentina).



*Ilustración 1 - Datos de la misión argentina SAC-D accedidos mediante el NASA OPENMCT.*

#### **RESULTADOS OBTENIDOS/ESPERADOS**

El FS2017 [11] fue el primer segmento de vuelo utilizado como prueba experimental para el desarrollo de los sistemas planteados en las diferentes líneas de investigación. Durante esta primera integración se estableció como objetivo



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

desarrollar un sistema multimisión pero prescindiendo de lenguajes propietarios para la extracción y decodificación de la telemetría y utilizando motores de base de datos para la persistencia y posterior procesamiento. Utilizando un intérprete de propósito general (Python) y las capacidades de reflexión de este, se logró decodificar completamente la telemetría del modelo de ingeniería sin que esto supusiera un costo computacional relevante [12] (Ver Ilustración 2 - Tiempos de decodificación para paquetes de 5000 variables de telemetría.)

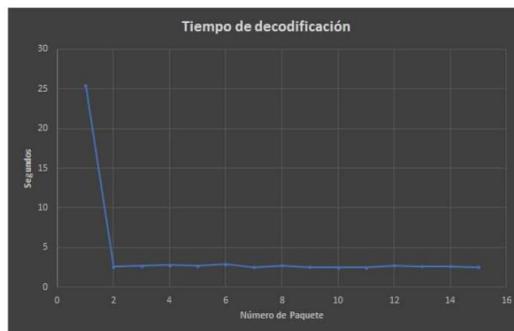


Ilustración 2 - Tiempos de decodificación para paquetes de 5000 variables de telemetría

Durante el 2018 se incorporó telemetría proveniente de la red satnogs de las misiones Lituanicasat 2 y Bugsat-1 demostrando el carácter multimisión del sistema. En el 2019 se está incorporando la telemetría histórica de la misión argentina SAC-D, siendo esta la primera misión con telemetría histórica almacenada incorporada al sistema lo que supone un desafío en términos de recuperación de datos en función del volumen. Con esta nueva incorporación se espera demostrar la capacidad de los medios de persistencia utilizados (RDBMS) para manejar de manera eficiente y segura los datos de plataforma de una misión espacial científica completa.

#### FORMACIÓN DE RECURSOS HUMANOS

Los trabajos desarrollados presentan una plataforma de experimentación realista con los sistemas espaciales. Permite a investigadores y

alumnos probar soluciones, comprobar límites, comparar alternativas y establecer criterios de decisión. La posibilidad de trabajar con varias misiones, desde pequeñas misiones universitarias hasta datos de satélites científicos permite responder de manera concreta a las premisas de desarrollar un sistema transparente al satélite en órbita.

Junto con las líneas de investigación presentadas en este documento se desarrolla una estación terrena que permite la descarga de datos de satélites en tiempo real y la formación de recursos en los aspectos de comunicaciones y mecánica orbital. La interacción con satélites actualmente en órbita permite validar los desarrollos de dinámica orbital. Actualmente el grupo se encuentra formado por un investigador formado, tres investigadores en formación.

#### AGRADECIMIENTOS

Se agradece al DIIT (Departamento de Ingeniería e Investigaciones Tecnológicas) de la UNLaM por el soporte en las investigaciones. Se hace extensivo el agradecimiento al Instituto Mario Gulich y a CONAE por el acceso a la telemetría de la misión argentina SAC-D.

#### ACRÓNIMOS

CONAE	Comisión Nacional de Actividades Espaciales.....	3
CORBA	Common Object Request Broker Architecture .....	2
COTS	Commercial-Off-The-Shelf .....	1
DIIT	Departamento de Ingeniería e Investigaciones Tecnológicas.....	3
MDIAE	Maestría en Desarrollos Informáticos de Aplicación Espacial .....	1
ORM	Object-relational mapping.....	2
OSS	Open Source Software.....	1
OTS	Off-the-Shelf .....	1
RDBMS	Relational database management system.....	2
SAC-D	Satélite de Aplicaciones Científicas - D.....	3
STOL	Satellite Test and Operation Language.....	2
UNLaM	Universidad Nacional de La Matanza.....	3



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

### BIBLIOGRAFÍA

- [1] M. Petković, R. Boumghar, M. Breskvar, S. Džeroski, D. Kocev, J. Levatić, L. Lucas, A. Osojnik, B. Ženko y N. Simidjievski, «Machine learning for predicting thermal power consumption of the Mars Express Spacecraft,» *arXiv preprint arXiv:1809.00542*, 2018.
- [2] J. J. Ramos Pérez, «A design for an advanced architecture of satellite ground segments,» 2014.
- [3] G. Chaudhri, J. Cater y B. Kizzort, «A model for a spacecraft operations language,» de *SpaceOps 2006 Conference*, 2006.
- [4] G. Garcia, «Use of Python as a Satellite Operations and Testing Automation Language,» de *GSAW2008 Conference, Redondo Beach, California*, 2008.
- [5] J. Houser y M. Pecchioli, «Database Administration for Spacecraft Operations-The Integral Experience,» *ESA BULLETIN*, pp. 100-107, 2000.
- [6] P. Cruce, B. Roberts, M. Bester y T. Quinn, «A database centered approach to satellite engineering data storage, access, and display,» 2007.
- [7] K. Galal y R. P. Hogan, «Satellite Mission Operations Best Practices,» 2001.
- [8] L. Foti, «CORBA Technology for Ground Segment System Software Development,» de *DASIA 98-Data Systems in Aerospace*, 1998.
- [9] M. Henning, «The rise and fall of CORBA,» *Queue*, vol. 4, pp. 28-34, 2006.
- [10] J. Trimble y G. Rinker, «Open Source Next Generation Visualization Software for Interplanetary Missions,» de *14th International Conference on Space Operations*, 2016.
- [11] P. Soligo, E. González, E. Sufán, E. Arias, R. Barbieri, P. Estrada, A. Montilla, J. Robin, J. Uranga, M. C. Valenti y others, «Misión CubeSat FS2017: Desarrollo de Software para una Misión Satelital Universitaria,» *WICC 2017*, p. 843.
- [12] P. Soligo y J. S. Ierache, «Software de segmento terreno de próxima generación,» *CACIC 2018*.



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

## Arquitectura de segmento terreno satelital adaptada para el control de límites de telemetría dinámicos

Pablo Soligo, Jorge Salvador Ierache 

Universidad Nacional de La Matanza  
Departamento de Ingeniería e Investigaciones Tecnológicas.  
Florencio Varela 1903, B1754JEC San Justo, Buenos Aires.  
[psoligo@unlam.edu.ar](mailto:psoligo@unlam.edu.ar)  
<http://www.unlam.edu.ar/>

**Resumen** El presente trabajo muestra la implementación de un prototipo funcional para el monitoreo del estado de salud de un satélite artificial científico. Aunque aplicable a otros sistemas, la implementación inicial ha sido probada utilizando como fuente de datos telemetría satelital y como prototipo inicial de desarrollo el Prototipo de Segmento Terreno de la UNLaM (UGS) derivado de la misión académica Formador Satelital 2017 (FS2017). Utilizando herramientas de estantería en cualquiera de sus variantes Off-the-Shelf (OTS), Commercial-Off-The-Shelf (COTS) u Open Source Software (OSS), el prototipo analiza los datos históricos para obtener patrones de comportamiento que permitan ajustar los límites de control y efectuar un monitoreo más cercano y preciso de la telemetría, en tiempo real, manteniendo los costos operativos acotados y utilizando hardware de propósito general.

### 1. Introducción

Existen tres métodos extendidos para analizar la telemetría de un satélite y controlar el estado de salud; control de límites, sistemas expertos y sistemas basados en modelos [1].

**Control de límites:** El control de límites es el sistema más simple y difundido, consiste en establecer, con la ayuda de un experto, los valores máximos y mínimos aceptables para un sensor (temperaturas, voltajes, corrientes) o conjunto de sensores y verificar que los mismos no sean violados. Estos valores usualmente deben ser ajustados con el sistema en vuelo y la cantidad de sensores puede volver impracticable por este método el control de todos los sensores. La simplicidad vuelve a este sistema el más popular, aunque se puede deducir que el mismo es absolutamente insensible al contexto.

**Sistemas expertos:** Durante los 80 y inicio de los 90 sistemas expertos han sido probados en diferentes misiones (Ej: GEOTAIL, NOZOMI y HAYABUSA). Estas técnicas si bien han mejorado el rendimiento de los sistemas



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

basados exclusivamente en control de límites no están exentas de problemas. Los sistemas expertos no pueden encontrar desperfectos desconocidos o no contemplados y requieren una laboriosa configuración y el conocimiento incorporado puede volverse inconsistente por cambios de diseño o por desconocer el comportamiento del satélite después del lanzamiento.

**Sistemas basados en modelos:** En esos sistemas la idea principal es detectar las anomalías y sus razones comparando simulaciones obtenidas por modelos computacionales contra el comportamiento actual del sistema.[1].

Cualquiera de los métodos mencionados requieren un asesoramiento casi continuo del experto en el área, ya sea actualizando límites, generando reglas o parametrizando simulaciones.

Por otro lado el aprendizaje automático ofrece un amplio rango de posibilidades para la predicción de comportamientos. Las técnicas de aprendizaje automático intentan servirse de los datos disponibles para predecir comportamientos futuros. En lugar de requerir a personal experto que infiera reglas y construya modelos, el aprendizaje automático ofrece una forma más eficiente para capturar el conocimiento, ha salido de los laboratorios y hoy es utilizado en el reconocimiento de voz e imágenes, en las búsquedas en Internet, en el ofrecimiento de productos y las fronteras de aplicación todavía no están establecidas.

Específicamente en el área de machine learning y telemetría satelital existen algunos antecedentes que abren el camino a estos estudios, los trabajos de [2], [1] y más recientemente [3] ofrecen alternativas distintas para la solución del mismo problema.

Los trabajos citados se concentran en establecer conclusiones, mayormente prometedoras, de las posibilidades que tiene el aprendizaje automático en el monitoreo del estado de salud de un satélite sin especificar una implementación práctica, objetivo principal de este trabajo.

## 2. Antecedentes

Durante las experiencias obtenidas durante la primera cohorte de la Maestría en Desarrollos Informáticos de Aplicación Espacial (MDIAE) (2015-2017) se generó el primer prototipo funcional del segmento terreno ahora denominado UGS para la misión FS2017. Se utilizó como segmento de vuelo un modelo de ingeniería de cubesat de 2U de la compañía ISIS (<https://www.isispace.nl/>) [4], [5] y [6]. En particular para el control del estado de salud se utilizó un sistema de control de límites, durante la MDIAE se tuvieron experiencias con múltiples sistemas del área espacial que hacen uso de esta técnica. Se han priorizado sobre las decisiones de diseño las recomendaciones que publicara en 2003 el Instituto Norteamericano de Aeronáutica y Astronáutica, del inglés Institute of Aeronautics and Astronautics (AIAA) [7], intentando cumplir los siguientes objetivos:

- Independencia de la arquitectura de hardware
- Independencia del fabricante del hardware
- Tolerancia a fallas



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

- Implementable en computadoras personales comerciales
- Independencia del sistema operativo

Con estas premisas y priorizando soluciones comúnmente utilizadas en los sistemas de propósito general se ha optado por:

1. Utilizar interpretes de propósito general para la decodificación de telemetría y los scripts de comandos en lugar de desarrollar interpretes propios (Solución extendida en el área espacial)
2. Utilizar base de datos relacionales (Sistema de base de datos relacional, del inglés Relational Database Management System (RDBMS)) y no relacionales para el almacenamiento de datos y metadatos.
3. Utilizar herramientas de máxima penetración para la distribución y ejecución de tareas.
4. Utilizar capas de acceso a los datos (Object Relational Mapper (ORM)) que faciliten la portabilidad entre motores de RDBMS y otras tecnologías de almacenamiento.
5. Utilizar protocolos de comunicación a nivel de aplicación comúnmente aceptados en el ámbito informático de propósito general.

El UGS, desde su primera versión, opta por el uso de un lenguaje de propósito general para el procesamiento de telemetría como para la generación de scripts de comandos. Por popularidad [8], productividad y facilidad de uso la opción elegida fue python. Al tratarse de un interprete, se ha utilizado las capacidades de reflexión de software como herramienta para la carga dinámica de procedimientos y posterior el procesamiento en tiempo real de los datos satelitales [5] y [4]. Puntualmente respecto al uso de lenguajes de propósito general, ya en el 2008, trabajos en la industria privada explicitaban las ventajas de este enfoque, considerando: [9]:

**Procesamiento:** XML, SOAP, XTCE parsing

**Comunicaciones:** Sockets, Acceso a Internet, Llamada a procedimiento remoto, del inglés Remote Procedure Call (RPC), Correo Electrónico

**Mediciones de performance:** Contadores, acceso al hardware y sistema operativo.

**Otros:** Acceso a bases de datos, funciones matemáticas, compresión de datos, hilos, criptografía.

Procesar telemetría utilizando carga en tiempo de ejecución y reflexión de software y la opción de utilizar un lenguaje e interprete de propósito general habilitan a una fracción de costo la incorporación de bibliotecas para la minería de datos y el aprendizaje automático. Todo esto suma un argumento mas a la lista expresada en [9].

### 3. Arquitectura e implementación

Originalmente el primer prototipo desarrollado de segmento terreno desarrollado para el FS2017 [6] contaba de un servidor de base de datos (RDBMS), una



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

aplicación web y una serie de productos a ejecutar manualmente. Actualmente el sistema trabaja sobre procesos distribuidos. Un subconjunto variable de procesos se ejecuta de manera distribuida dentro de la infraestructura de hardware disponible. Cada nueva ingesta de telemetría, independientemente de la fuente, dispara un proceso distribuido encargado de procesar el paquete, generar y persistir las variables de telemetría que el paquete contenga. Se generan todas las variables de telemetría que el satélite tiene configuradas, activas y se corresponden con el tipo de paquete. Actualmente el prototipo puede trabajar con telemetría de múltiples misiones incluida Satélite Argentino Científico - D (SAC-D) ([10]) utilizado en esta prueba. Desde su primera versión, el prototipo verifica el estado de salud del satélite mediante un control de límites (1), tanto para sus variables directas como derivadas, estas últimas son variables que se crean artificialmente a partir de la combinación de otras. La adaptación mostrada en este trabajo propone crear una entidad asociada al tipo de telemetría donde almacenar un modelo de predicción automáticamente creado. La figura 1 muestra el diagrama de clases (Solo se muestran los campos sensibles al análisis) donde se pueden encontrar los campos para almacenamiento del modelo (Clase **TlmyPrediction**, atributo *data*) y fecha de expiración (Clase **TlmyPrediction**, atributo *expiration*), esta entidad está asociada al tipo de telemetría **TlmyVarType**.

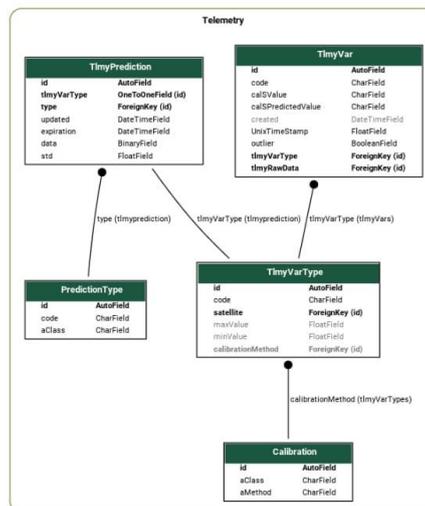


Figura 1: Diagrama de clases



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

La predicción puede no estar disponible, no ser conveniente o no presentar relevancia para el tipo de telemetría, por esta razón la relación no es obligatoria. Si se recibe un nuevo paquete de datos (independientemente de la fuente) y este contiene un tipo de telemetría determinado y este tipo tiene un modelo para la predicción, el mismo es aplicado modificando los límites máximos y mínimos de ese tipo de telemetría en un valor desplazado desde el valor predicho, positivamente y negativamente una distancia configurada. En las pruebas desarrolladas se ha tomado un  $\sigma$  (Sigma) como desplazamiento (1 y 2).

$$max = p + \sigma \quad (1)$$

$$min = p - \sigma \quad (2)$$

Estableciendo los máximos y mínimos dinámicamente en función de una predicción se pueden obtener controles sensibles al contexto actualizado del satélite. En el caso de análisis de este trabajo se ha creado una variable derivada denominada **eclipseElapsedTime**, esta variable contiene el tiempo en segundos que el satélite permanece eclipsado o la cantidad de segundos (negativos) hasta el próximo eclipse. La figura 2 muestra las variable directa **vBatAverage** (Voltaje medio de batería) del SAC-D y la variable derivada **eclipseElapsedTime** calculada en *runtime* dentro del mismo sistema utilizando pyephem [11] (Esta última variable podría ser provista por los servicios del área de determinación orbital en una agencia espacial [12]).

En la figura 2 se puede observar como la tensión decrece (variable **vBatAverage**) durante el eclipse, los paneles solares quedan parcial o totalmente tapados por la tierra durante algunos minutos.

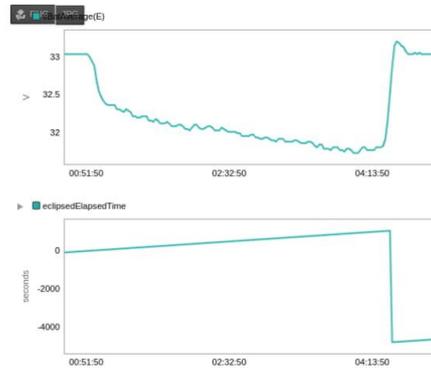


Figura 2: Caída de tensión al entrar en eclipse, ambos gráficos apilados muestran el mismo periodo de tiempo. La figura superior muestra el voltaje y la figura inferior la variable derivada **eclipseElapsedTime**.



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

Una vez que los paneles vuelven a tener la iluminación solar ocurre una rápida recuperación durante el periodo entre umbra y penumbra. La figura 3 muestra que el patrón se repite durante varios eclipses.

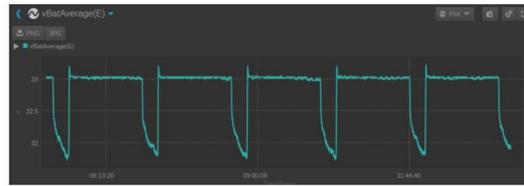


Figura 3: Variable directa **vBatAverage** visualizada por OPENMCT

Utilizando como técnica el control únicamente límites estáticos(1), para el ejemplo dado, se deberían establecer límites máximos y mínimos del orden de los 30v y 34v respectivamente. Mediciones de 31v durante los periodos entre eclipses no serían tomadas como una anomalía cuando en realidad se podría estar presentando un problema grave en la carga y un comportamiento anormal. Un conjunto de reglas explicitadas por un experto pueden salvar esta situación, aunque es necesario el experto y por otro lado los valores finales solo se conocen con el sistema en vuelo.

En el sistema propuesto, un proceso distribuido (Figura 4, Tareas Periódicas) revisa los tipos de telemetría que tienen modelos de predicción vencidos, es decir, superan su fecha expiración (propiedad *expiration*, entidad **TlmyPrediction**). Previamente ese tipo de telemetría fue configurado como sensible a predicción. Los modelos tienen una fecha de vencimiento que puede forzar su regeneración. Para cada tipo de telemetría en esta situación se realiza una nueva preparación de datos según configuración, se genera un nuevo modelo de predicción y en el caso de ser lo suficientemente preciso se persiste, caso contrario se informa mediante una alarma.

Al utilizar un lenguaje de propósito general para el procesamiento de telemetría se puede hacer uso de las múltiples bibliotecas disponibles para aprendizaje automático y normalmente existen herramientas para la persistencia (Almacenamiento en el algún medio permanente) de los modelos de predicción. Cuando un nuevo paquete de telemetría ingresa al sistema mediante la Interfaz de programación de aplicaciones, del inglés API: Application Programming Interface (API) se procesa el paquete y se llama a la función de calibración para cada uno de los tipos de variables de telemetría configuradas para ese paquete (Figura 4, Procesador de telemetría). Si el tipo de telemetría tiene asociado un modelo de predicción entonces este también se aplica y se utilizan el resultado para ajustar los máximos y mínimos tolerables, si el valor calibrado está por fuera de los límites dinámicos entonces se genera una alarma siguiendo el flujo establecido en el sistema.



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

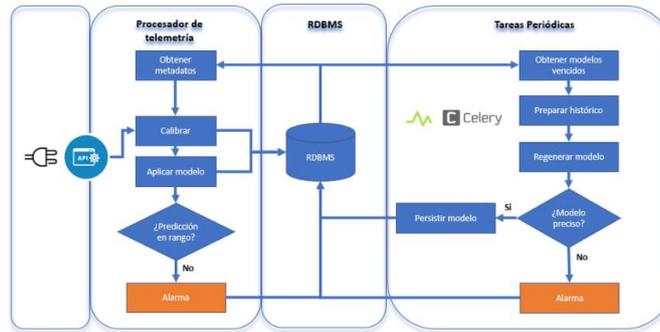


Figura 4: Arquitectura conceptual para la verificación de límites en la telemetría

#### 4. Resultados

La figura 5 muestra los resultados aplicando regresión polinómica y aplicando un árbol de regresión lineal (Biblioteca scikit-learn [13]). En ambos casos los valores predichos están dentro del rango de un  $\sigma$  establecido. Se puede observar tanto en los periodos entre eclipses como dentro del eclipse como los máximos y mínimos se ajustan permitiendo un control mas estricto y por sobre todo, sensible al contexto.



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

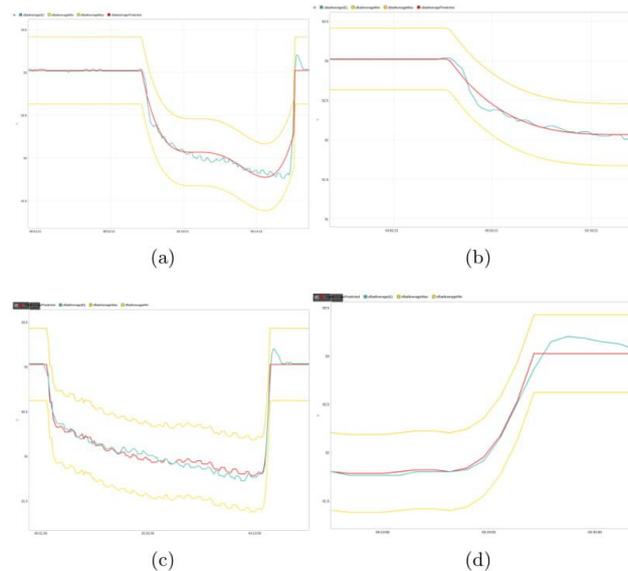
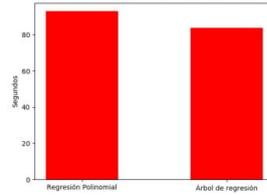


Figura 5: (a) Predicción utilizando regresión polinómica (Eclipse completo) (b) Predicción utilizando regresión polinómica (Entrada en eclipse) (c) Predicción utilizando árbol de regresión (Eclipse completo) y (d) Predicción utilizando árbol de regresión (Salida del eclipse), donde celeste es valor real, rojo valor predicho y amarillos máximos y mínimos

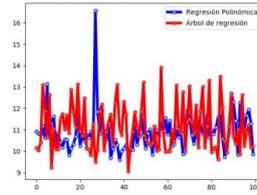
Los máximos y mínimos son ahora sensibles al contexto, entendiendo por contexto a la variable derivada que indica el periodo entre eclipses en segundos. La figura 6 (a) muestra los tiempos de generación del modelo incluyendo la búsqueda de los datos (12 horas), la preparación y persistencia de los datos para alimentar el algoritmo y la generación y persistencia del modelo en sí. La figura 6 (b) muestra (En milisegundos) los tiempos necesarios para la generación, utilizando el modelo persistido, del valor predicho y el ajuste de los límites para la variable **vBatAverage**. Se muestran los tiempos para 100 paquetes SAC-D, los tiempos no contemplan el control de límites que es independiente de si el límite es modificado dinámicamente o establecido por configuración.



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019



(a) Tiempo de generación del modelo de predicción



(b) Tiempo en milisegundos para la predicción de variables **vBatAverage** pertenecientes a satélite SAC-D (100 muestras).

Figura 6

## 5. Conclusiones

El presente trabajo muestra una propuesta de implementación que permite aplicar las herramientas contemporáneas de aprendizaje automático a un segmento terreno desarrollado bajo la premisa expresada en trabajos anteriores [6], [5] y [4]. La capacidad añadida no implicó modificación alguna al núcleo del segmento terreno original y su implementación es transparente. Como se mostró en las figuras 6 (a) y 6 (b), sin generar un costo relevante en cuanto a tiempo de procesamiento, la idea permite un control más cercano de la telemetría de house-keeping sin necesidad de intervención humana durante la operación. El costo y riesgo de este tipo de misiones justifican el desarrollo de estas técnicas y suman un argumento adicional a favor del trabajo con herramientas de propósito general en el área espacial.

## 6. Trabajo a Futuro

La propuesta presentada en este trabajo no está exenta de problemas o puntos a profundizar. La búsqueda de relaciones, o variables contextuales aún requieren del asesoramiento de un experto para su correcta identificación. En próximos trabajos se abordará esta problemática en función de encontrar correlaciones y explotárselas de manera automática, buscando explotar tanto como sea posible el poder de cómputo actual, no como remplazo del experto, sino como un aliado en la búsqueda e identificación de patrones ocultos. Por otro lado, aunque se ha contemplado la eliminación de *outliers* durante la preparación de los datos de entrenamiento, la aparición de estos durante la operación y su impacto en la generación de falsos positivos no ha sido tenida en



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

cuenta en esta primera versión, la incorporación de un filtro que detecte *outliers* en tiempo real esta pendiente.

## Referencias

1. Takehisa Yairi, Yoshinobu Kawahara, Ryohei Fujimaki, Yuichi Sato, and Kazuo Machida. Telemetry-mining: a machine learning approach to anomaly detection and fault diagnosis for space systems. In *Space Mission Challenges for Information Technology, 2006. SMC-IT 2006. Second IEEE International Conference on*, pages 8–pp. IEEE, 2006.
2. Takehisa Yairi, Minoru Nakatsugawa, Koichi Hori, Shinichi Nakasuka, Kazuo Machida, and Naoki Ishihama. Adaptive limit checking for spacecraft telemetry data using regression tree learning. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, volume 6, pages 5130–5135. IEEE, 2004.
3. L.Amoruso C.Brighenti, A.Brighenti D.Evans, M.Ricatto D.Moretto, and M.Carbone F.Ferrari. Advances in context aware spacecraft telemetry checking. In *2018 IAC*. IAC, 2018.
4. Pablo Soligo and Jorge Salvador Ierache. Segmento terreno para misiones espaciales de próxima generación. *WICC 2019*.
5. Pablo Soligo and Jorge Salvador Ierache. Software de segmento terreno de próxima generación. In *XXIV Congreso Argentino de Ciencias de la Computación (La Plata, 2018)*, 2018.
6. Pablo Soligo, Ezequiel González, Eduardo Sufán, Emmanuel Arias, Ricardo Barbieri, Pablo Estrada, Alfonso Montilla, José Robin, Javier Uranga, M Cecilia Valenti, et al. Misión cubesat fs2017: Desarrollo de software para una misión satelital universitaria. *WICC 2017*, page 843.
7. Ken Galal and Roger P Hogan. Satellite mission operations best practices. 2001.
8. TIOBE Software. TIOBE programming community index, september 2017, 2017. [Online; accessed 26-September-2017].
9. Gonzalo Garcia. Use of python as a satellite operations and testing automation language. In *GSAW2008 Conference, Redondo Beach, California*, 2008.
10. Amit Sen, Yunjin Kim, Daniel Caruso, Gary Lagerloef, Raul Colomb, Simon Yueh, and David Le Vine. Aquarius/sac-d mission overview. In *Sensors, Systems, and Next-Generation Satellites X*, volume 6361, page 63610I. International Society for Optics and Photonics, 2006.
11. Brandon Craig Rhodes. Pyephem: astronomical ephemeris for python. *Astrophysics Source Code Library*, 2011.
12. S. Hackel, Y. Wasser, M. Meinel, and R. Kahle. Flight dynamics microservices. In *Proceedings of the International Astronautical Congress, IAC*, volume 2018-October, 2018.
13. scikit-learn machine learning in python. <https://scikit-learn.org/stable/>. Accessed: 2019-23-07.



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

## Experiences and lessons learned developing a next-generation ground segment prototype

M.Sc. Eng. Pablo Soligo  
*Departamento de Ingeniería e Investigaciones  
Técnicas - Grupo de Investigación y Desarrollo  
de Software Aeroespacial.  
Universidad Nacional de La Plata  
Buenos Aires, Argentina  
[psoligo@unlam.edu.ar](mailto:psoligo@unlam.edu.ar)*

PhD. Jorge Salvador Ierache  
*Departamento de Ingeniería e Investigaciones  
Técnicas - Grupo de Investigación y Desarrollo  
de Software Aeroespacial.  
Universidad Nacional de La Plata  
Buenos Aires, Argentina  
 <https://orcid.org/0000-0002-1772-9186>*

*Index Terms*— Artificial satellites, Small Satellites, Ground support, Ground segment, Software design, Space technology, Telecommands, Telemetry, COTS.

*Abstract - In this paper we discuss how, using general purpose software tools and advanced programming techniques, it is possible to create a multi-mission ground segment software application that can be applied in almost (virtually in) any mission. We developed a functional prototype and worked with real satellite telemetry and commands. Different alternatives were explored and several analyses have shown the advantages of this approach in terms of productivity, maintainability and cost. One of the main goals was to use modern general-purpose tools instead of old or spatial industry classical ones, providing a cost-effective and multi-platform solution. We used general purpose databases, a well-known Object Relational Mapper (ORM), popular programming languages, libraries, frameworks and advanced programming techniques. We avoid any tool, protocol or strategy not used in software industry. Telemetry data from several sources is processed, stored and finally showed through Open Source Mission Control Software (OPENMCT) application by National Aeronautics and Space Administration (NASA). External software units can interact with the system using (REST)/JavaScript Object Notation (JSON) and web sockets interfaces. The interfaces are available to receive telemetry from different external systems or to publish telemetry to different external clients. In particular in this work, the*

*published telemetry interface is used by OPENMCT. An initial prototype was developed to support the flight segment of a cubesat engineering model provided by Innovative Solutions In Space (ISIS) in the context of the academic mission Formador Satelital 2017 (FS2017), as part of Master in software development for space application (MDIAE). Currently we test compatibility (without commands) not just for cubesat but also for medium or large missions. The system evolved from a traditional client-server architecture to a distributed system in order to improve the horizontal scalability and, in its most recent version, data mining and machine learning techniques were incorporated. Using well-known libraries this telemetry processor can predict future telemetry values and compare them with the real ones in order to detect anomalies or unknown patterns.*

### I. INTRODUCTION

This paper elaborates on the experiences and results of the development a cost-effective satellite ground segment system prototype. The prototype design and implementation is based on the experiences carried out during MDIAE with ground segment systems currently being produced and their relationship to general purpose software systems widely used in the software industry. During the design phase, the use of widely accepted and proven robustness tools and solutions was prioritized as



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

opposed to common proposals of the space industry. Part of the goal of this and previous papers is to collate these implementations, with prevalence in three domains:

#### A. Decoding, command and control languages

The use of general-purpose languages and interpreters is proposed to decode telemetry and develop command scripts.

#### B. Data Persistence

An architecture where data persistence<sup>1</sup> is performed on relational database engines solely accessed through an ORM is proposed. Both the data in all processing levels and the definitions are stored in a single central repository.

#### C. Interfaces

Interfaces with internal and external systems for module communication as well as data publication and ingestion are performed through extended and accepted protocols.

Ad-hoc or low-penetration standards, formats and protocols in the massive use industry are marginalized from the solution and COTS solutions, libraries, components and frameworks are prioritized. Even though part of the goals were related to a proposal different from the collected systems, the recommendations published in AIAA [1] have been taken into consideration, for being in accordance with good practices of software engineering.

In chapter 2, the background encouraging the creation of the GIDSA research group and the established technological premises are explained. Chapter 3 describes the approach adopted to control the satellite, decode its telemetry and implementation details. Chapter 4 presents the storage methods used and the restrictions imposed in order to achieve a better compatibility in different scenarios. Chapter 5 describes some characteristics of the implemented interfaces and the currently integrated systems. Chapter 6 describes the latest system adaptations; whose goal is to achieve a predictive behavior in terms of state of health. Finally, chapters 7 and 8 list results and conclusions.

## II. BACKGROUND

During the experiences undergone in the first

<sup>1</sup> It refers to the data attribute for them to survive somehow.

cohort of MDIAE (2015-2017), the first functional ground segment prototype was developed, now called UNLaM Ground Segment Prototype (UGS) for mission FS2017 [2]. A 2U cubesat engineering model from the ISIS company (<https://www.isispace.nl/>) was used as flight segment.

The first prototype developed was based on a client/server architecture which resulted in the current distributed architecture [3]. In order to develop this first prototype, it was necessary to reverse engineer the model since it did not have manufacturer support.

With these premises and prioritizing commonly used solutions in general purpose systems, we have opted for:

- Using general purpose interpreters for decoding telemetry and command scripts instead of developing our own interpreters.
- Using relational (Relational Database Management System (RDBMS)) and non-relational databases for data and metadata storage.
- Using data access layers (ORM) enabling portability between RDBMS motors and other storage technologies.
- Using maximum penetration tools for task distribution and execution.
- Using communication protocols at a level of applications commonly accepted within the general-purpose IT field.

## III. DECODING, COMMAND AND CONTROL LANGUAGES

In the space field, when dealing with controlling a satellite or decoding a telemetry frame, a common practice is the use of proprietary languages for specific purposes. Generally, command sequences are applied to control structures in these languages, as happens with telemetry interpretation in multimission systems. Space companies and agencies around the world have developed their own languages and interpreters in order to achieve this goal [4]:

STOL: Satellite Test and Operation Language. It has been developed by Nasa and widely used in



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

several missions.

PLUTO: it has been used in some ESA missions (Satellite Control and Operation System 2000).

Others: developed or used by different companies SOL (GMV), CCL (Harris), PIL (Astrium), SCL (ICS).

These languages are typically proprietary and incompatible with one another [5], hindering migrations between systems and the possibility of sharing procedures between different missions.

FS2017 [2] was the first flight segment used as experimental test for developing the systems presented in this paper, subsequently broadening to telemetry in Lituanicasat2, Bug-SAT Tita and SAC-D missions. We opted for a general purpose language instead of creating a specific language or using the existing ones in CONAE, academic partner in MDIAE.

This approach has many advantages. The following table compares pros and cons of each implementation type.

	Specific purpose	General purpose
Advantages	They can be more user-friendly for non-programmer users. They can offer specific adaptations to satellite operation problems.	Portable, more powerful than specific purpose versions. Great number of users.
Disadvantages	It is a proprietary technology. Portability issues. Own development. Poor tools and lack of documentation	Too many options, Too powerful, Could be less readable.

In the case of FS2017 ground segment, the idea of a proper mission language was rejected due to lack of time and resources needed for developing and validating a specific purpose interpreter. This condition is particularly true in small satellite missions, which lack this capability due to the cost of incorporation, resulting in projects without this capability. Python was the language used due to its popularity [6], ease of learning and reflexive characteristics, even though the concepts presented here are applicable to other interpreters with similar characteristics.

By using a general-purpose interpreter, we can also build scripts with better debug capabilities, community available documents and specially avoid the need to take or update the interpreter in several platforms.

### A. Telemetry processor

The system allows setting up telemetry variables showing type of frame, position, number of bytes, and form of interpretation, and it is possible to assign a calibration function that changes raw information into engineering variables. This calibration function is developed by python independently from the system by the specialist. When including new telemetry, the system distributes the processing and the unit in charge of processing the telemetry variable will look for the applicable calibration method in the configuration (Figure 1 - Calibration Method Selection). If not previously uploaded, it will be uploaded in runtime and called up to perform the processing.



Figure 1 - Calibration Method Selection

A distributed process regularly monitors the calibration methods that have been included to perform the load and checks the methods that have modified their shape to perform a re-load of the updated version. Figure 1 - Calibration Method Selection shows the selection of a calibration method for a specific telemetry variable. Methods are uploaded and updated on runtime.

The software reflection technique used for uploading and running calibration functions is generally costly in processing time. Two types of optimization are used to improve performance:

- If no changes occur, functions are uploaded once. Any change to the source files means re-uploading the function.
- The function only runs if the new raw datum is different to the previous one or if the calibration function has been recently uploaded, which implies a possible change.

By applying these simple optimizations, the tests performed proved that it is possible to



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

calibrate/decode around 5000 engineering variables between 2.5 and 3.5 seconds -a capability enough to provide a satellite with characteristics similar to the ones of SAC-D with a desktop computer (Intel® Core™ i5-5200U Dual Core, 2.20 GHz). Figure 2 - Decode Time show the decoding time for 16 samples of 5000 telemetry variables, each of them with a 10% exchange rate mostly using parameterized linear conversions. It can be observed that the first sample takes slightly over 25 seconds. The first decoding requires uploading and running all calibration functions. As from the second sample, the optimizations previously described are executed.

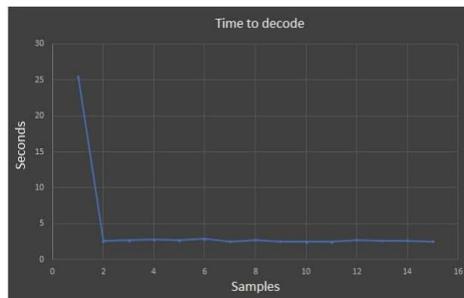


Figure 2 - Decode Time

### B. Command processor

Similarly to how telemetry is processed, command scripts are uploaded and called up on runtime. The command processor has only been tested with the FS2017 model. The system allows to set up specific commands for each satellite. The execution of a certain command type can trigger a call to an adaptor, or the execution of a code snippet related to manufacturer's specifications.

Particularly in the case of FS2017, the execution of a command type produces the output of the code defined by the manufacturer in a TCP/IP port working as communication interface with radio software.

```
satellite = Satellite.objects.get(code=="FS2017")

#Get a
tt = TlmyVarType.objects.get(code="obcT1")
#Get the last five temperatures
tmpls = tt.vars
    _order_by('-created')[:5]
    .values_list('calFValue', flat=True)
#Average, using numpy, an external library
med = np.average(tmpls)
#Check a condition
if med>TT.max:
    #Condition is true, I'm going to send a command
```

```
ct = satellite.getCommandType()
    .get(code="telemetryOBC")
cmd = satellite.newCommand(ct)
    satellite.sendCommand(cmd)
```

Source 1 – Example of a conditional command script shows a conditional script, where a “telemetryOBC” command is sent if it is detected that the obcT1 telemetry is over the maximum limit. It is also observed that the limit is controlled with a mean value estimated with an external library (numpy), showing the use of external libraries in a command script with an allegedly simple example.

```
satellite = Satellite.objects.get(code=="FS2017")
#Get a
tt = TlmyVarType.objects.get(code="obcT1")
#Get the last five temperatures
tmpls = tt.vars
    _order_by('-created')[:5]
    .values_list('calFValue', flat=True)
#Average, using numpy, an external library
med = np.average(tmpls)
#Check a condition
if med>TT.max:
    #Condition is true, I'm going to send a command
ct = satellite.getCommandType()
    .get(code="telemetryOBC")
cmd = satellite.newCommand(ct)
    satellite.sendCommand(cmd)
```

Source 1 – Example of a conditional command script

## IV. DATA PERSISTENCE

In the general-purpose software industry, the use of relational or non-relational DBMS has spread from large banking and accounting systems to small embedded devices. Even though it is not an absolute rule and there exist papers stating experiences on the implementation of formal storage, search and recovery strategies ([7] and [8]) in the space industry, it is not difficult to find complex and advanced systems avoiding the use of these tools [9]. During MDIAE, all the ground segment systems studied avoided storage on a DBMS. Instead, data or metadata were available in binary or flat files. The absence of DBMS hinders effective and standardized data storage and recovery, eliminates a concept of mandatory security such as referential integrity and overrides the capability of several systems working on data in a safe and concurrent way.

In the implemented prototype, both data definitions (alarms, telemetry variables, available commands, etc.) and data itself are run by a relational database motor at any processing level. In satellite telemetry, there are entities for raw data storage independently from the satellite they belong to, or the



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

source of ingestion. As this data is processed, they are moved to tables where the engineering variables are already processed and available for any module that requires them, including the command processor.

For future versions, implantation of tables that enable efficient access to historic data, KPIs and denormalizations to allow for long term storage is foreseen. All data is exclusively accessed with ORM. The use of this intermediate layer offers productivity improvements as well as independence from the database motor provider.

## V. INTERFACES

In terms of internal and external interfaces, there are plenty of ad-hoc solutions or implementations of low-penetration technologies in the space field. CORBA is a remarkable example because it has not been massively adopted and has been quickly changed for HTTP-based implementations [10], even though it is a standard presented by several players in the software industry. The interfaces of the proposed system are fully based on this latest technology. All telemetry is ingested with a REST service which can be used by both third-party modules and internal modules that adapt formats, technologies or characteristics from former implementations. Task distribution, synchronization and general communication are also done with http-based brokers.

## VI. STATE OF HEALTH

Particularly, in order to check the state of health, a limit control system, which is one of the most simple and generally used techniques for checking state of health, was initially used ([11] and [12]). During MDIAE, experiences with several systems from the space field that use this technique were carried out; and as from the first version, the prototype checks the satellite's state of health with a limit control (1), for direct and derivative variables (these derivative variables are artificially created from a combination of the others). Even though it is simple to implement, this technique is context-insensitive and the history value [13] is completely unknown. As it was explained in Chapter 3, the use of a general-purpose language for telemetry decoding-calibration allows to use the available libraries to enhance scripts. In the

case of UGS, this capability is used with machine learning scikit-learn libraries [14] in order to find detours during regular running by using historic behavior and contextual variable data.

In its latest version, the system has included entities into the data model to allow decoding-calibration scripts to modify maximum and minimum limits previously configured according to predicted values. The system analyzes historic information and context variables to create a model allowing to predict future behaviour. As from a prediction, the system modifies one-sigma limits to enhance control and avoid false positives [13].

## VII. RESULTS AND LESSON LEARNED

### A. Graphical User Interfaces

Even though user interfaces were not the main goal during this first project phase, it was necessary to create a telemetry administration and visualization tool that would allow to visualize and share results. The evolution of the prototype, for research use, to an available system with production output capabilities also requires a user interface. As a current solution, we decided to develop interactive web interfaces that would run in any device (PC, Tablet, Cellphone). The development of desktop applications was avoided due to the cost and difficulty experienced when porting to different platforms, such as Windows, Linux and Android.

For administration purposes, the development of static pages was enough. This fast development was enough for the research team to administer data, even though it is not suitable for a production system. To visualize historic and real-time telemetry, it was necessary to use web development frameworks.

#### 1) First prototype

To visualize historic and real-time telemetry, user interfaces were initially developed to run on the browser interacting with the backend through packages JSON. Experienced developers (even though they did not have experience with the used tool) invested more than 120 hours achieving poor results, which only accounted for a fraction of the needs and had serious adaptability issues with different screen sizes. Figure 3 - First Prototype UI shows this first prototype.



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

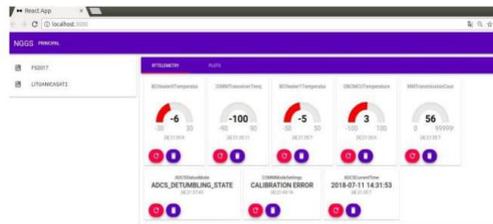


Figure 3 - First Prototype UI

## 2) NASA Open MCT

As an alternative, an integration with NASA Open MCT (<https://github.com/nasa/openmct>) was evaluated and finally, was partially implemented. The Open MCT proved to be an excellent, modern alternative with a wide range of possibilities but lacks application examples that would help developers, even though it is well documented. Figure 4 - Open MCT Showing telemetry published by UGS shows 2 using UGS data.

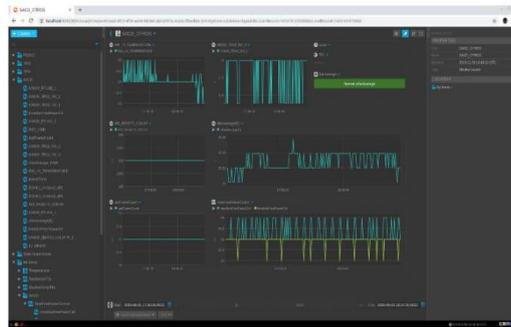


Figure 4 - Open MCT Showing telemetry published by UGS

For the partial integration, which does not cover mandatory aspects necessary for a production output, with NASA Open MCT, no less than 90 man-hours of development were needed.

The initial development (5) was discontinued since it jeopardized other research tasks due to its high development cost. The current state of the integration with NASA Open MCT only includes historic and real-time telemetry and telemetry metadata partially. Even though it is an already developed tool, it needs a developer heavily involved in the technology being implemented. To conclude, modern user interfaces can take more than 40% of total development time

[15].

## B. Commands and Telemetry with general purpose languages

At present, general purpose languages have characteristics, capabilities and tools outweighing specific purpose ones, which can hardly be regarded as complex. Several, large-scale projects support their robustness. Well-proven options do not justify a proprietary language development; not having a user database means lacking documentation, support and tools. Small satellite missions do not need to do without capabilities available for larger missions. The use of software reflection allowed to only include capabilities previously used in larger missions at a low cost. The tests performed showed that it is applicable even in general purpose hardware.

## C. Telemetry mining

The latest prototype version showed a practical, alternative, and low-cost implementation to a simple limit control. The importance of this implementation regarding early fault detection is yet to be proven and it is one of the future lines of research presented.

## D. Other results

The use of mature ORMs has allowed to develop the prototype independently from the database motor. The use of relational databases helps to process distribution and referential integrity maintenance. However, denormalizations allowing to work with data from past years for large satellite missions have not been applied to the prototype. These design adjustments are necessary to get relatively constant response times for large amounts of data. The denormalizations to implement are known and there are excellent style guides [16].

## VIII. CONCLUSIONS

The experiences carried out show an experimentation platform that is realistic regarding space systems. They allow researchers and students to test solutions, control limits, compare alternatives and set decision criteria. The possibility of working



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

with several missions, from small college missions to scientific satellite data, allows to specifically answer to the premise of developing a system transparent to an orbiting satellite. The experiences, progress made, and results show that a promising path has been chosen. The prototype showed that it is low-cost and adaptable, and offers some functionalities even outweighing larger-scale systems. Products such as NASA OPENMCT, directly integrable with the developed interfaces or with papers such as [17] support the results. Future lines of work will continue exploring the proposed alternatives exclusively based on techniques and tools of massive use in the software industry, specially applied to the space industry within the ground segment context. Lines of research will focus on applying data exploitation and automated learning techniques and tools for determining the state of health of the satellite platform.

#### ACKNOWLEDGMENTS

This research was funded by PROINCE C211. We would like to thank the Engineering and Technological Research Department (DIIT - Departamento de Ingeniería e Investigaciones Tecnológicas) of UNLaM for supporting research. We would also like to thank CONAE for allowing access to the telemetry of the Argentine mission SAC-D.

#### IX. REFERENCES

- [1] K. Galal y R. P. Hogan, «Satellite Mission Operations Best Practices,» 2001.
- [2] P. Soligo, E. González, E. Sufán, E. Arias, R. Barbieri, P. Estrada, A. Montilla, J. Robin, J. Uranga, M. C. Valenti y others, «Misión CubeSat FS2017: Desarrollo de Software para una Misión Satelital Universitaria,» *WICC 2017*, p. 843.
- [3] P. Soligo y J. S. Ierache, «Segmento Terreno Para Misiones Espaciales de Próxima Generación,» *WICC 2019*.
- [4] G. Garcia, «Use of Python as a Satellite Operations and Testing Automation Language,» de *GSAW2008 Conference, Redondo Beach, California*, 2008.
- [5] G. Chaudhri, J. Cater y B. Kizzort, «A model for a spacecraft operations language,» de *SpaceOps 2006 Conference*, 2006.
- [6] T. I. O. B. E. Software, «TIOBE Programming Community Index, September 2017,» 2017. [En línea]. Available: <https://www.tiobe.com/tiobe-index/>.
- [7] J. Houser y M. Pecchioli, «Database Administration for Spacecraft Operations-The Integral Experience,» *ESA BULLETIN*, pp. 100-107, 2000.
- [8] P. Cruce, B. Roberts, M. Bester y T. Quinn, «A database centered approach to satellite engineering data storage, access, and display,» 2007.
- [9] O. Montenbruck, M. C. Eckstein y J. Gonner, «The geo-control system for station keeping and colocation of geostationary satellites,» 1993.
- [10] M. Henning, «The rise and fall of CORBA,» *Queue*, vol. 4, pp. 28-34, 2006.
- [11] T. Yairi, M. Nakatsugawa, K. Hori, S. Nakasuka, K. Machida y N. Ishihama, «Adaptive limit checking for spacecraft telemetry data using regression tree learning,» de *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, 2004.
- [12] T. Yairi, Y. Kawahara, R. Fujimaki, Y. Sato y K. Machida, «Telemetry-mining: a machine learning approach to anomaly detection and fault diagnosis for space systems,» de *Space Mission Challenges for Information Technology, 2006. SMC-IT 2006. Second IEEE International Conference on*, 2006.
- [13] P. Soligo y J. S. Ierache, «Arquitectura de segmento terreno satelital adaptada para el control de límites de telemetría dinámicos,» de *XXV Congreso Argentino de Ciencias de la Computación (La Plata, 2019)*, 2019.
- [14] *scikit-learn Machine Learning in Python*.
- [15] G. Calvary y J. Coutaz, «Introduction to model-based user interfaces,» *Group Note*, vol. 7, p. W3C, 2014.
- [16] T. Morel, G. Garcia, M. Palsson y J. C. Gil, «High Performance Telemetry Archiving and Trending for Satellite Control Centers,» de *SpaceOps 2010 Conference Delivering on the Dream Hosted by NASA Marshall Space Flight Center and Organized by AIAA*, 2010.
- [17] L. A. C. Brighenti, A. B. D. Evans, M. R. D. Moretto y M. C. F. Ferrari, «Advances in Context Aware Spacecraft telemetry Checking,» de *2018 IAC*, 2018.
- [18] T. Yairi, N. Takeishi, T. Oda, Y. Nakajima, N. Nishimura y N. Takata, «A Data-driven Health Monitoring Method for Satellite Housekeeping Data based on Probabilistic Clustering and Dimensionality Reduction,» *IEEE Transactions on Aerospace and Electronic Systems*, 2017.

#### X. ACRONYMS

AIAA	Institute of Aeronautics and Astronautics .....	2
CONAE	Comisión Nacional de Actividades Espaciales.....	3



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

COTS	
Commercial Off-TheShelf .....	1, 2
DBMS	
Database Management System .....	4
DIIT	
Departamento de Ingeniería e Investigaciones Tecnológicas .....	7
FS2017	
Formador Satelital 2017 .....	1, 2, 3, 4
HTTP	
Hypertext Transfer Protocol .....	5
MCT	
Mission Control Software .....	6
MDIAE	
Maestría en Desarrollos Informáticos de Aplicación Espacial .....	1, 2, 4, 5
NASA	
National Aeronautics and Space Administration .....	1, 6
ORM	
Object Relational Mapper .....	1, 2, 4
RDBMS	
Relational Database Management System .....	2
REST	
Representational State Transfer .....	1
SAC-D	
(Spanish	
Satélite de Aplicaciones Científicas-D, meaning Satellite for	
Scientific Applications-D) .....	4
STOL	
Satellite Test and Operation Language .....	2
UGS	
Unlam Prototype Ground Segment .....	2, 5, 6
UNLaM	
Universidad Nacional de La Matanza .....	2



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

- Presentar una copia impresa firmada del presente documento junto con los Anexos, y enviar todo en archivo PDF por correo electrónico a la Secretaría de Investigación Departamental. **Límite de entrega: 28 de febrero de 2020**

\_\_\_\_\_  
Firma y aclaración  
del director del proyecto.

Lugar y fecha :.....