

UNMDP - Facultad de Ingeniería - Departamento de Informática

Sistema de procesamiento de imágenes satelitales para la gestión agropecuaria

Alumno: García Federico (fedegarciamdq@gmail.com)

Director: Mg. Cristian Remon

Agradecimientos

A quienes hicieron posible llegar al resultado de este trabajo. A todos los docentes que formaron parte de mi formación a lo largo de la carrera. A todo el equipo de Pastech por la confianza y por estar siempre a disposición, y a mi director, Cristian, por su predisposición constante para responder, orientar y guiarme durante el desarrollo de este proyecto.

A mis amigos, por el apoyo constante y por estar presentes a lo largo de todo este camino.

A los compañeros que conocí durante la carrera, especialmente en el tramo final.

A mi familia, en especial a mis padres por apoyarme siempre, y a mis abuelos, que son mi motivación.

Y por último, a Sofía, que siempre estuvo a mi lado y fue una pieza fundamental para llegar hasta acá, acompañándome en los momentos difíciles y celebrando cada logro conmigo.

Índice

1. Resumen	6
2. Introducción	7
2.1 Contexto	7
2.2 Problema a resolver	7
2.3 Solución propuesta	8
2.3.1 Justificación del uso de imágenes satelitales	9
2.4 Objetivos del proyecto	9
2.4.1 Objetivo General	9
2.4.2 Objetivos Específicos	9
3. F.O.D.A.	10
3.1 Fortalezas	10
3.2 Oportunidades	10
3.3 Debilidades	11
3.4 Amenazas	11
4. Gestión de riesgos	12
5. Desarrollo	17
5.1 Planificación inicial	17
5.2 Desviaciones en las tareas	19
5.2.1 Diagrama de Gantt	19
5.2.2 Distribución semanal de horas	20
5.2.3 Comparación entre estimación y ejecución por etapa	21
5.2.4 Distribución total del esfuerzo	23
5.3 Modalidad de trabajo	24
5.3.1 Organización y herramientas de trabajo	24
5.3.1.1 Organización de tareas	24
5.3.1.2 Control de código fuente	25
5.3.2 Metodología por etapas	26
5.3.2.1 Investigación	26
5.3.2.2 Análisis	27
5.3.2.3 Diseño	27
5.3.2.4 Desarrollo	27
5.3.2.5 Testing	28
5.3.2.6 Despliegue	28
6. Sistema	30
6.1 Análisis	30
6.1.1 Requerimientos del sistema	30
6.1.1.1 Requerimientos Funcionales (RF)	30
6.1.1.2 Requerimientos No Funcionales (RNF)	32
6.1.2 Casos de uso	34
6.1.3 Requisitos de interfaces externas	35
6.1.3.1 Interfaces de Usuario	35

6.1.3.2 Interfaces de Hardware	35
6.1.3.3 Interfaces de Software	35
6.1.3.4 Interfaces de Comunicaciones	35
6.2 Diseño	36
6.2.1 Arquitectura del sistema	36
6.2.1.1 Estilo arquitectónico elegido	36
6.2.1.2 Justificación de la arquitectura	37
6.2.1.2.1 Beneficios en Atributos de Calidad	38
6.2.1.2.1 Trade-offs	38
6.2.1.3 Diagrama de despliegue	39
6.2.2 Diseño de base de datos	40
6.2.2.2 Modelo Lógico relacional	41
6.2.2.3 Justificación del modelo	42
6.2.2.3.1 Normalización	42
6.2.2.3.2 Elección de relaciones	43
6.2.3 Diseño de Tests	43
6.2.4 Diseño de despliegue / DevOps	44
6.2.4.1 Entorno de ejecución	44
6.2.4.2 Contenedores	44
6.2.5 Tecnologías utilizadas	45
6.2.5.1 Backend y framework	45
6.2.5.2 Base de datos	45
6.2.5.3 Procesamiento de imágenes satelitales	45
6.2.5.4 Almacenamiento en la nube	46
6.2.5.6 Validación, serialización y comunicación	46
6.2.5.7 Testing y análisis	46
6.2.5.8 Infraestructura y despliegue	47
6.2.6 Patrones de diseño aplicados	47
6.2.7 Manejo de idempotencia, resiliencia y tolerancia a fallos	47
6.2.8 Especificación de la API	49
6.3 Estructura de costos	49
6.3.1 Estrategia de Despliegue	49
6.3.1.1 Fase 1: Implementación Inicial (Uruguay, Paraguay, Chile)	50
6.3.1.2 Fase 2: Escalado a Producción (Argentina)	50
7. Proceso de Cálculo y Análisis de Imágenes Satelitales	51
7.1 Selección del proveedor de imágenes satelitales	51
7.2 Tipos de imágenes satelitales y sus características	52
7.2.1 Resolución espacial	52
7.2.2 Resolución temporal	53
7.2.3 Resolución espectral	53
7.3 Comparación entre Sentinel y proveedores comerciales de imágenes satelitales	53
7.3.1 Sentinel	54
7.3.2 Proveedores comerciales	54

7.3.3 Ventajas, Limitaciones y Selección del Proveedor	54
7.4 Índices de Vegetación Calculados	56
7.4.1 NDVI (Normalized Difference Vegetation Index)	56
7.4.2 Interpretación de los valores de NDVI	57
7.4.3 Aplicación del NDVI en el contexto del proyecto	58
7.5 Modelo de estimación de biomasa	59
7.6 Validación de imágenes	61
7.7 Validación de resultados	62
7.7.1 Comparación de cálculos	62
7.7.1.1 Objetivo	62
7.7.1.2 Datos utilizados	62
7.7.1.3 Análisis de NDVI	64
7.7.1.4 Análisis de Stock Forrajero	66
7.7.1.5 Conclusiones	67
7.7.2 Comparación de disponibilidad de imágenes	68
7.7.2.1 Objetivo	68
7.7.2.2 Resultados	68
7.7.2.3 Conclusiones	70
8. Memoria del proyecto	71
8.1 Cumplimientos	71
8.2 Obstáculos	71
8.3 Futuras mejoras	74
8.4 Estado actual del sistema	76
Anexos	78
Glosario	88
Bibliografía	90

Nota: Con el objetivo de facilitar la lectura y comprensión del informe, se incluye al final del documento un Glosario, donde se definen los principales términos técnicos utilizados.

1. Resumen

Este proyecto final de carrera se centró en el desarrollo de un sistema automatizado de procesamiento de imágenes satelitales orientado a optimizar la gestión agropecuaria, específicamente en la estimación de la biomasa forrajera para la ganadería. En el sector ganadero, el manejo eficiente del pasto es estratégico, sin embargo, los métodos tradicionales de medición manual y visual presentan limitaciones críticas en cuanto a precisión, frecuencia y escalabilidad. Para abordar esta problemática, surge la necesidad de la empresa Pastech de integrar herramientas tecnológicas que superen las deficiencias de resolución y los periodos de inactividad por nubosidad de los sensores gratuitos actuales.

Como solución, se diseñó y desarrolló una API modular y escalable, que automatiza el ciclo completo de adquisición y análisis de datos geoespaciales. El sistema permite la gestión de potreros mediante polígonos GeoJSON y se integra con el proveedor comercial Planet para obtener imágenes de alta resolución y frecuencia diaria. A partir de estos datos, la aplicación calcula automáticamente el índice de vegetación NDVI y aplica modelos de estimación de biomasa que contemplan la estacionalidad y el tipo de pastura. Tecnológicamente, la solución se implementó utilizando el lenguaje Python, una base de datos MySQL y la plataforma Railway para su despliegue en la nube.

Los resultados fueron validados satisfactoriamente en conjunto con ingenieros agrónomos, confirmando que las mediciones obtenidas son consistentes y aportan un valor diferencial para la toma de decisiones productivas en el ecosistema de Pastech. El proyecto cumplió con los objetivos técnicos y plazos previstos, logrando superar obstáculos como la curva de aprendizaje en análisis espectral y la complejidad de integración con servicios de terceros. De esta manera, se consolidó una herramienta profesional que no solo resuelve una necesidad operativa actual, sino que sienta las bases para futuras mejoras en el monitoreo de precisión de diversos cultivos.

2. Introducción

2.1 Contexto

La ganadería es una actividad estratégica dentro del sector agropecuario, y su rendimiento depende directamente del manejo eficiente del pasto disponible. Tradicionalmente, la estimación del stock forrajero se realizó mediante inspecciones visuales y mediciones manuales, métodos que si bien son funcionales, presentan limitaciones en precisión, escalabilidad y frecuencia de actualización. La implementación de herramientas basadas en imágenes satelitales permite superar estas barreras, ofreciendo monitoreo continuo y cobertura espacial amplia sin necesidad de presencia física en campo.

Sin embargo, la adquisición y procesamiento de estas imágenes requiere herramientas que automaticen la obtención de datos, apliquen cálculos sobre las bandas espectrales y presenten resultados para la toma de decisiones operativas. En este contexto, la empresa *Pastech* busca incorporar un sistema que complemente su integración con proveedores satelitales, calcule índices de vegetación como NDVI y traduzca estos valores en estimaciones concretas del stock de pasto disponible.

El avance de proveedores satelitales comerciales y la disponibilidad de APIs especializadas abren la oportunidad de desarrollar una solución capaz de conectarse con estos servicios, procesar datos espectrales y convertirlos en información útil para la toma de decisiones.

Por lo expuesto anteriormente, el objetivo de este trabajo final es desarrollar una API capaz de cumplir con dichos requerimientos, integrando procesamiento satelital y análisis espectral, sentando las bases para una herramienta escalable y adaptable a futuras mejoras.

2.2 Problema a resolver

La empresa **Pastech**, dedicada al desarrollo de soluciones tecnológicas para el sector agropecuario, busca mejorar la precisión y frecuencia con la que estima el stock de pasto disponible en los potreros que gestiona, integrando un nuevo proveedor de imágenes satelitales.

Actualmente, las herramientas utilizadas presentan limitaciones tanto en la disponibilidad de imágenes satelitales actualizadas como en la resolución, lo que afecta la confiabilidad de los indicadores definidos.

Por lo tanto, se plantea la necesidad de contar con una solución que permita:

- Obtener imágenes satelitales de alta frecuencia y adecuada resolución.
- Calcular índices de vegetación.
- Estimar de forma automatizada el stock forrajero disponible por potrero.

Dado que el acceso a imágenes satelitales implica el uso de servicios de terceros, será necesario integrar proveedores, priorizando aquellos que permitan consultas diarias y alta resolución.

2.3 Solución propuesta

El proyecto propone el desarrollo de una **API modular, escalable y de uso interno**, que automatice todas las etapas del trabajo con imágenes satelitales:

1. **Gestión de potreros mediante polígonos geoespaciales.**
2. **Integración con proveedores satelitales** para obtener imágenes con:
 - Alta resolución espacial
 - Alta frecuencia temporal
 - Control de nubosidad
3. **Cálculo automatizado del NDVI y otros índices espectrales relevantes.**
4. **Cálculo para estimar biomasa en función del estado del pastizal.**
5. **Devolución de información estructurada** lista para integrarse en el panel productivo de Pastech.

Además, el diseño arquitectónico contempla la incorporación futura de nuevos:

- Proveedores satelitales
- Algoritmos de análisis
- Indicadores o fuentes complementarias

Lo que garantiza la continuidad del sistema a largo plazo.

2.3.1 Justificación del uso de imágenes satelitales

La decisión de utilizar exclusivamente imágenes satelitales, en lugar de otras tecnologías como vehículos aéreos no tripulados (drones), responde a un requerimiento fundamental de Pastech: la automatización total del monitoreo. El uso de imágenes satelitales permite realizar mediciones frecuentes y a gran escala sin requerir intervención manual, logística física ni despliegue de personal en los establecimientos agropecuarios, garantizando así la escalabilidad del sistema.

No obstante, gracias al diseño modular, la solución no se encuentra estrictamente limitada a un único medio de captura. Si en el futuro la empresa decidiera incorporar drones equipados con sensores multiespectrales capaces de registrar las bandas requeridas (Rojo e Infrarrojo Cercano), el sistema cuenta con la capacidad de adaptarse para ingerir y procesar dichas imágenes mediante el desarrollo de un nuevo adaptador.

2.4 Objetivos del proyecto

2.4.1 Objetivo General

Desarrollar un sistema de automatización para el procesamiento de imágenes satelitales que permita estimar indicadores forrajeros en forma diaria para la gestión eficiente de potreros.

2.4.2 Objetivos Específicos

- Registrar potreros mediante coordenadas geográficas.
- Integrar servicios de terceros para la obtención de imágenes satelitales.
- Procesar automáticamente las imágenes para calcular índices de vegetación.
- Estimar la cantidad de biomasa disponible para la alimentación animal.
- Diseñar la arquitectura del sistema con criterios de escalabilidad, mantenibilidad y robustez.

3. F.O.D.A.

3.1 Fortalezas

Integración con una plataforma existente (Pastech):

La API propuesta no parte desde cero, sino que se integrará a Pastech, una plataforma web de gestión agropecuaria. Esto permite una implementación más ágil, aprovechando la infraestructura tecnológica y la base de usuarios ya existente.

Propuesta innovadora que agrega valor al sector agropecuario:

El desarrollo ofrece una solución tecnológica avanzada que combina observación satelital, análisis espectral y conocimiento técnico del manejo forrajero. Esto representa un diferencial competitivo para Pastech y una herramienta concreta para mejorar la eficiencia del manejo forrajero y la toma de decisiones productivas.

Uso de datos satelitales actualizados de alta frecuencia:

Esto permite un seguimiento continuo del estado vegetativo de los potreros, brindando información actualizada al productor en tiempo casi real.

3.2 Oportunidades

Creciente adopción de tecnologías satelitales en el sector agropecuario:

El sector agropecuario está cada vez más abierto a la adopción de soluciones basadas en datos geoespaciales. Esto facilita la aceptación y escalabilidad del producto, especialmente en procesos que se buscan digitalizar y basar decisiones en información objetiva.

Posibilidad de escalar la solución a otras regiones o cultivos:

Si bien el enfoque inicial está en el manejo de pasturas, la infraestructura puede adaptarse para monitorear otros cultivos o condiciones productivas. Además, la solución es aplicable a diversas regiones del país o incluso a nivel internacional, ampliando el mercado objetivo.

Intercambio de información y conocimiento con profesionales del sector agropecuario:

El desarrollo de esta solución abre la puerta a una colaboración con ingenieros agrónomos, asesores técnicos y productores, lo que permite mejorar la interpretación de los datos y validar resultados en campo. Esto también puede derivar en nuevas funcionalidades.

3.3 Debilidades

Escasa experiencia inicial con procesamiento de imágenes satelitales y análisis espectral:

Se podría enfrentar una curva de aprendizaje importante en el manejo de imágenes multispectrales, correcciones atmosféricas, selección de bandas y cálculo de índices.

3.4 Amenazas

Dependencia técnica y disponibilidad de la infraestructura satelital:

Al depender exclusivamente de plataformas externas para la obtención de datos, el sistema queda expuesto a riesgos operativos fuera de su control. Fallas en la plataforma de origen (API), alteraciones repentinas en los formatos de entrega de los datos, o la baja de un satélite pueden generar interrupciones en el servicio, dejando sin cobertura temporal o permanente ciertas zonas geográficas y representando un riesgo crítico de disponibilidad si se depende de un solo proveedor.

Cambios en la política de precios o acceso de proveedores:

El costo del servicio satelital puede variar inesperadamente por decisión del proveedor. Una suba abrupta o un cambio en el modelo de licencias puede comprometer la viabilidad económica de la solución si no se planifica adecuadamente.

Condiciones climáticas adversas (nubosidad):

La presencia frecuente de nubes en ciertas regiones limita la captación útil de imágenes ópticas.

4. Gestión de riesgos

La gestión de riesgos constituye una etapa fundamental dentro del desarrollo del proyecto, ya que permite identificar posibles eventos que puedan afectar el cumplimiento de los objetivos planteados, evaluar su impacto y probabilidad, y establecer medidas preventivas o correctivas. A continuación se presenta la matriz de riesgos elaborada para el proyecto, junto con las estrategias propuestas para su mitigación.

Para la evaluación de los riesgos identificados se asignaron valores numéricos a la **Probabilidad de Ocurrencia** e **Impacto**, en una escala del 1 al 3. Con el objetivo de cuantificar la criticidad de cada riesgo, se calculó un Peso mediante la siguiente fórmula:

$$\text{Peso} = \text{Probabilidad} * \text{Impacto}$$

El resultado permite priorizar acciones de mitigación, considerando más críticos aquellos riesgos con mayor valor numérico.

Se establece como criterio que los planes de contingencia solo se aplicarán para riesgos cuyo Peso ≥ 6 , por considerarse de alto impacto potencial sobre la continuidad del proyecto.

Tabla 1*Matriz de riesgos*

Riesgo	Consecuencia	Probabilidad	Impacto	Peso	Plan de Contingencia
R1: Cambios en la política de precios o acceso de proveedores	Aumentos en costos o cambios en las licencias afectan la viabilidad económica del proyecto	2	3	6	Utilizar contratos de corto plazo para evitar compromisos prolongados con proveedores externos, permitiendo renegociar costos o cambiar de proveedor ante modificaciones en precios o licencias.
R2: Posible discontinuación o reconfiguración de servicios satelitales	Pérdida de cobertura en ciertas zonas	1	3	3	-
R3: Condiciones climáticas adversas (nubosidad)	Inutilización de imágenes por demasiada nubosidad deja sin mediciones a un potrero por algunos días.	3	2	6	Implementar un algoritmo de filtrado espacial determinista utilizando la capa de calidad de píxeles de la imagen. Esto permite identificar y aislar las áreas afectadas por nubes o sombras, aprovechando las regiones limpias del lote para el cálculo del NDVI sin introducir ruido analítico ni requerir modelos de procesamiento complejos.
R4: Cambios en los requerimientos del cliente	Retrabajo y retrasos	3	2	6	Armar sprint cortos. En cada sprint verificar (validar) los requerimientos antes de empezar a desarrollarlo.

durante el proyecto

<p>R5: La solución no aporta un valor diferencial significativo respecto a la tecnología actual</p>	<p>Baja adopción de la solución por parte de Pastech, pérdida de justificación para la inversión.</p>	1	3	3	-
<p>R6: Aparición de nuevas tecnologías que vuelvan obsoleta o significativamente inferior la solución antes de su implementación</p>	<p>Desventaja competitiva en el mercado, necesidad de rehacer parte del desarrollo o buscar nuevos proveedores</p>	2	3	6	<p>Desacoplar la implementación a un solo proveedor, permitiendo agregar nuevos proveedores fácilmente.</p> <p>Monitorear tendencias y evaluar si la nueva tecnología ofrece mejoras sustanciales. En caso de que las nuevas tecnologías aumenten significativamente la precisión o reduzcan mucho los costos, evaluar su implementación y reestructurar el plan de trabajo.</p>
<p>R7: Los requerimientos funcionales no representan los objetivos reales del proyecto</p>	<p>Desarrollo de funcionalidades innecesarias o falta de las esenciales. Implica pérdida de tiempo y recursos</p>	2	2	4	-
<p>R8: Tiempo estimado afectado por falta de experiencia en procesamiento de</p>	<p>Retrasos, necesidad de capacitación</p>	2	3	6	<p>Realizar capacitación específica y contar con el apoyo de un referente con experiencia en análisis y procesamiento de imágenes para mitigar riesgos de retrasos.</p>

imágenes satelitales

R9: Dependencia excesiva de personal técnico agropecuario para el algoritmo de cálculo

Retrasos por la espera a una respuesta de los profesionales

3

2

6

Documentar y diversificar las fuentes de consulta. Más de dos semanas sin respuesta implicaría reestructuración de plan de trabajo.

R10: No se logra realizar la implementación

El proyecto no alcanza un estado funcional y no se puede poner en producción, generando una pérdida de inversión y tiempo para la empresa.

1

3

3

-

Durante el desarrollo del proyecto, los riesgos que efectivamente se materializaron fueron el **R1** y el **R3**.

En relación con el **R1**, la incertidumbre propia de una solución en etapa de validación hacía riesgoso comprometerse desde el inicio con un plan de larga duración y mayor inversión. Ante cambios en las condiciones del proveedor y considerando que aún se estaba evaluando el impacto real de la incorporación de nuevas imágenes satelitales en el sistema, se analizó la contratación de un plan de menor alcance para una primera etapa de validación en producción. Si bien esta alternativa implicaba un mayor costo por imagen, permitía reducir la exposición económica inicial y mantener flexibilidad hasta confirmar la viabilidad técnica y el valor agregado de la solución.

En cuanto al **R3**, vinculado a la presencia de nubosidad en las imágenes satelitales, el riesgo se mitigó exitosamente mediante la implementación de un algoritmo de filtrado espacial determinista. Utilizando la capa de calidad de píxeles (Usable Data Mask) provista por el satélite, el sistema logró aislar mediante lógica matemática las áreas afectadas por nubes y sombras. Esto permitió procesar imágenes parcialmente nubladas y aprovechar con exactitud las porciones limpias de cada captura, reduciendo la pérdida de mediciones.

5. Desarrollo

5.1 Planificación inicial

La planificación del desarrollo es una etapa esencial en todo proyecto de software, ya que permite organizar las tareas, estimar tiempos de ejecución y establecer un cronograma ordenado que guíe el avance del trabajo. Para este proyecto se elaboró un plan temporal a partir de la descomposición de tareas y la estimación de sus duraciones. Como herramienta principal se utilizó un diagrama de Gantt, que facilita la visualización de las etapas del proyecto y su relación temporal. En la Figura 1 se presenta el cronograma planificado previo al inicio del desarrollo.

Figura 1

Diagrama de Gantt inicial

ETAPA / SEMANA	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20	#21	#22	#23	#24	#25	#26	#27	#28	#29	#30	#31	#32	
Investigación	12	12	11																														
Análisis				9	9	7																											
Diseño							12	12	11																								
Desarrollo										12	(1)	10	12	10	12	10	12	(2)			8	8	6	5	5	10							
Testing												2		2		2					4	4	2	2	2	2	8						
Documentación													3										4				3						
Despliegue																											5	5					
Informe final																														12	12	12	4

Se estimó una duración total del proyecto de 32 semanas, con una dedicación promedio de 12 horas semanales a partir de la etapa de “Diseño” (semana 7). Además, en el cronograma se contemplaron semanas destinadas al estudio para parciales y/o entregas de trabajos académicos, indicadas como (1) y (2) dentro del Gantt, con el objetivo de mantener una planificación realista que considere la disponibilidad efectiva durante el periodo de desarrollo.

El proyecto se dividió en las siguientes etapas:

- 1) Investigación (duración total de 35 h): Se realiza un estudio sobre imágenes satelitales, sus tipos, características, resolución y periodicidad. Además, se analiza cómo se procesan y qué información se puede extraer.

Luego, se lleva a cabo un relevamiento de proveedores de imágenes, evaluando aspectos como costos, frecuencias de captura, condiciones de uso y resolución espacial. Esta es una etapa clave para determinar la viabilidad técnica y económica del proyecto.

2) Análisis (duración total de 25 h): Se define en conjunto con el referente funcional de Pastech cuales son los requerimientos específicos del sistema. También se analiza el flujo actual de trabajo de la empresa y la arquitectura de su sistema.

3) Diseño (duración total de 35 h): Se diseña la arquitectura general del sistema, incluyendo los componentes principales de la API, los esquemas de datos, los endpoints necesarios y el modo en el que se integrará con los servicios ya existentes de la empresa.

4) Desarrollo (duración total 120 h): Durante esta etapa se llevará a cabo la implementación del sistema. Se desarrollarán todas las funcionalidades definidas en las etapas previas cumpliendo con todos los requerimientos.

5) Testing (duración total 30 h): Se realizan pruebas unitarias y de integración para validar el correcto funcionamiento del sistema. Se prueba la precisión del cálculo de índices y estimaciones, la estabilidad de la API ante distintos volúmenes de datos y la integración con otros componentes del ecosistema Pastech.

6) Documentación (duración total 10 h): Se redacta la documentación técnica del sistema, incluyendo detalles de la arquitectura, configuración y uso de la API.

7) Despliegue (duración total 10 h): Se realiza el despliegue de la API en un entorno cloud, asegurando disponibilidad, escalabilidad y seguridad. Se llevará a cabo la integración definitiva con el ecosistema de Pastech y se realizarán pruebas finales en producción supervisadas por el equipo técnico.

8) Informe final (duración total 40 h): Se elabora un informe detallado que incluirá la descripción de cada etapa, resultados obtenidos, dificultades enfrentadas y posibles mejoras futuras. Este documento será un respaldo formal del trabajo desarrollado y base para futuras ampliaciones del sistema. Cabe aclarar que, en las etapas de investigación,

análisis y diseño, se contemplaron las horas necesarias para elaborar documentación que integrará el informe final.

5.2 Desviaciones en las tareas

Si bien, como se mencionó anteriormente, la planificación y el plan de trabajo constituyen pilares fundamentales en un proyecto de software y tienen un impacto directo en su éxito, la inexperiencia puede derivar en estimaciones poco precisas.

El objetivo de esta sección es contrastar la planificación inicial con la ejecución real del proyecto, analizando las desviaciones tanto en la cantidad de horas dedicadas a cada etapa como en su distribución a lo largo del tiempo.

5.2.1 Diagrama de Gantt

En la Figura 2 se presenta el diagrama de Gantt ejecutado, correspondiente al estado final del proyecto. Más allá de las diferencias en el tiempo empleado, que se analizan en las secciones siguientes, existen algunos aspectos cualitativos relevantes a destacar.

Figura 2

Diagrama de Gantt ejecutado

ETAPA / SEMANA	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20	#21	#22	#23	#24	#25	#26	#27	#28	#29	#30	#31	#32	
Investigación	12	12	10	3																													
Análisis				5	3	8		1	5		1			2				1				2											
Diseño							10	9	5					1			1					1											
Desarrollo										12	11	12	6	8	(1)	6	2	10	10	13	10	10	(2)	10	7	6	6						
Testing												2				10				2					3	8							
Documentación												4							4					2									
Despliegue																									3	4		2					
Informe final																									2				-10	10	10	10	

En primer lugar, tanto la etapa de análisis como la de diseño habían sido planificadas inicialmente como instancias únicas y no iterativas. Si bien se contemplaban reuniones periódicas, estas se pensaron principalmente como instancias de validación de avances, asumiendo que los ajustes menores podrían resolverse dentro de la etapa de desarrollo.

En la práctica, esto no ocurrió. Los cambios que surgieron reunión a reunión afectaron requerimientos centrales del sistema, lo que implicó la necesidad de analizar y rediseñar componentes clave. Como consecuencia, estas etapas adoptaron un enfoque más iterativo, reflejado claramente en el diagrama de Gantt mediante la aparición de sprints sucesivos, representando de manera más fiel la dinámica real del proyecto.

Otro aspecto relevante fue la etapa de testing. La planificación inicial contemplaba la realización de pruebas desde etapas tempranas del desarrollo, distribuyendo la carga horaria en varias semanas con poca dedicación. Sin embargo, al tratarse de un sistema con pocos casos de uso pero de alta complejidad, resultó difícil implementar tests significativos en las primeras etapas.

En consecuencia, el esfuerzo de testing se concentró en dos momentos bien definidos: una primera instancia dedicada a la preparación de la estructura, configuración y herramientas de prueba, y una etapa final donde fueron implementados efectivamente los tests. Esto derivó en una menor cantidad de semanas dedicadas a testing, pero con una carga horaria mayor en cada una.

Asimismo, al momento de la planificación no se contaba con un calendario académico definido, lo que explica los desplazamientos identificados como (1) y (2) en el diagrama. Estas pausas ocurrieron en semanas distintas a las originalmente previstas e incluso fue necesario reducir la duración total de las interrupciones planificadas.

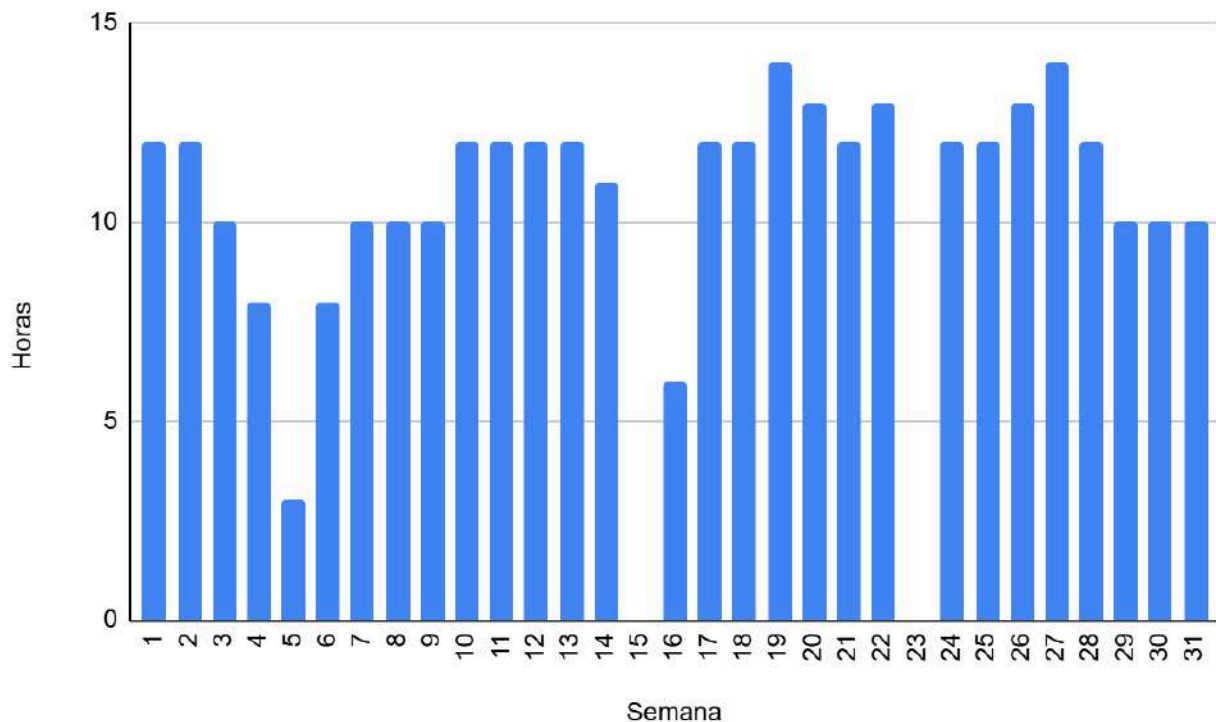
Por último, la etapa de despliegue también presentó diferencias respecto a lo planificado. Si bien se había concebido como una actividad concentrada al final del proyecto, en la práctica se adoptó un enfoque orientado al despliegue desde etapas tempranas, utilizando herramientas como Docker y pipelines de integración continua. Esto implicó dedicar tiempo a la configuración de estos componentes, pero permitió que el sistema quede listo para ser desplegado sin necesidad de configuraciones manuales adicionales por parte de Pastech.

5.2.2 Distribución semanal de horas

Durante la planificación inicial, se estimó la duración del proyecto en función de una dedicación semanal realista, considerando actividades externas al desarrollo. A partir de la etapa de diseño, se estableció una dedicación objetivo de 12 horas semanales, considerada suficiente para cumplir con los plazos propuestos.

Figura 3

Dedicación semanal de horas



La Figura 3 muestra que, en promedio, esta estimación se cumplió. Las semanas con una menor carga horaria fueron compensadas por otras con mayor dedicación, manteniendo un equilibrio general a lo largo del proyecto.

Se destacan particularmente las semanas 5 y 16. En la semana 5, correspondiente a la etapa de análisis, la menor dedicación se debió a la dependencia de la disponibilidad del personal de Pastech para la realización de reuniones clave.

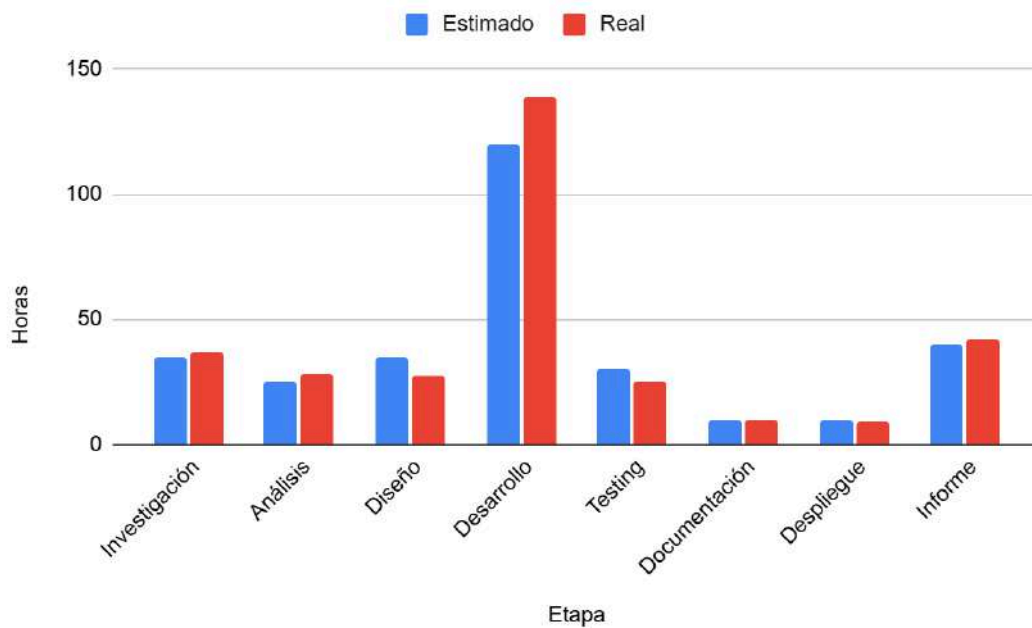
En la semana 16, la reducción de horas respondió a la coincidencia con un examen parcial. En lugar de pausar completamente el proyecto, se optó por mantener una dedicación reducida, aprovechando el tiempo disponible.

5.2.3 Comparación entre estimación y ejecución por etapa

La Figura 4 permite observar las desviaciones absolutas del proyecto.

Figura 4

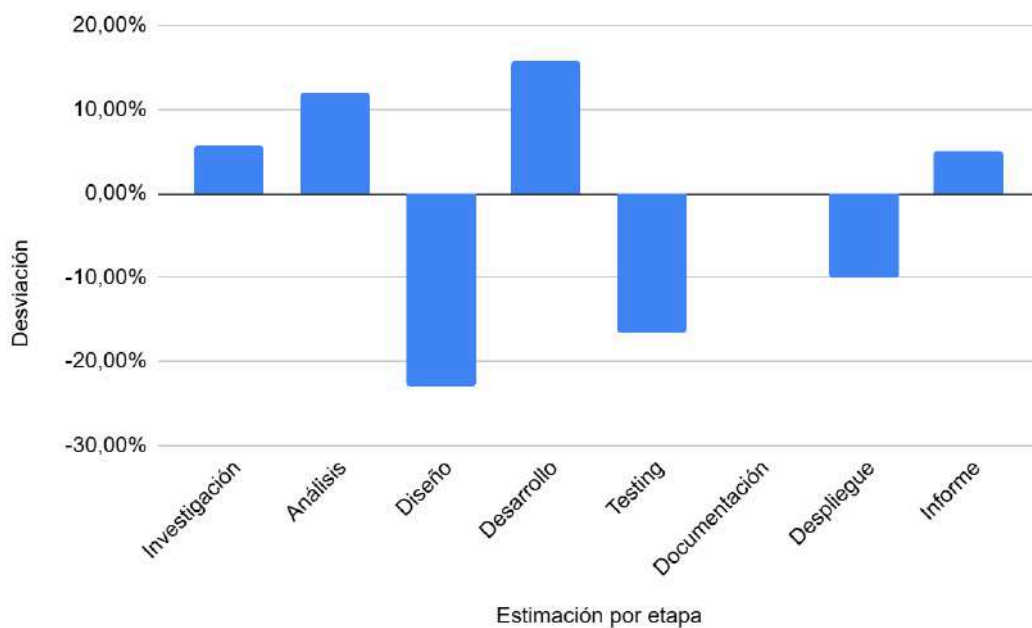
Comparación entre horas estimadas y empleadas por etapa



No obstante, estas diferencias resultan más representativas cuando se analizan en términos porcentuales, ya que una desviación de varias horas no tiene el mismo impacto según la etapa considerada. Por ejemplo, una diferencia de 10 horas en desarrollo resulta poco significativa, mientras que esa misma desviación en despliegue sí lo sería.

Figura 5

Desviación porcentual entre tiempo estimado y empleado por etapa



En este análisis, la desviación más notable se observa en la etapa de diseño, donde fueron empleadas menos horas de las inicialmente estimadas. Esto se explica, en gran medida, porque durante el análisis ya se habían identificado importantes restricciones del sistema, particularmente la necesidad de reutilizar el esquema de base de datos existente. El hecho de contar con gran parte de las entidades previamente definidas redujo considerablemente el esfuerzo requerido en la etapa de diseño.

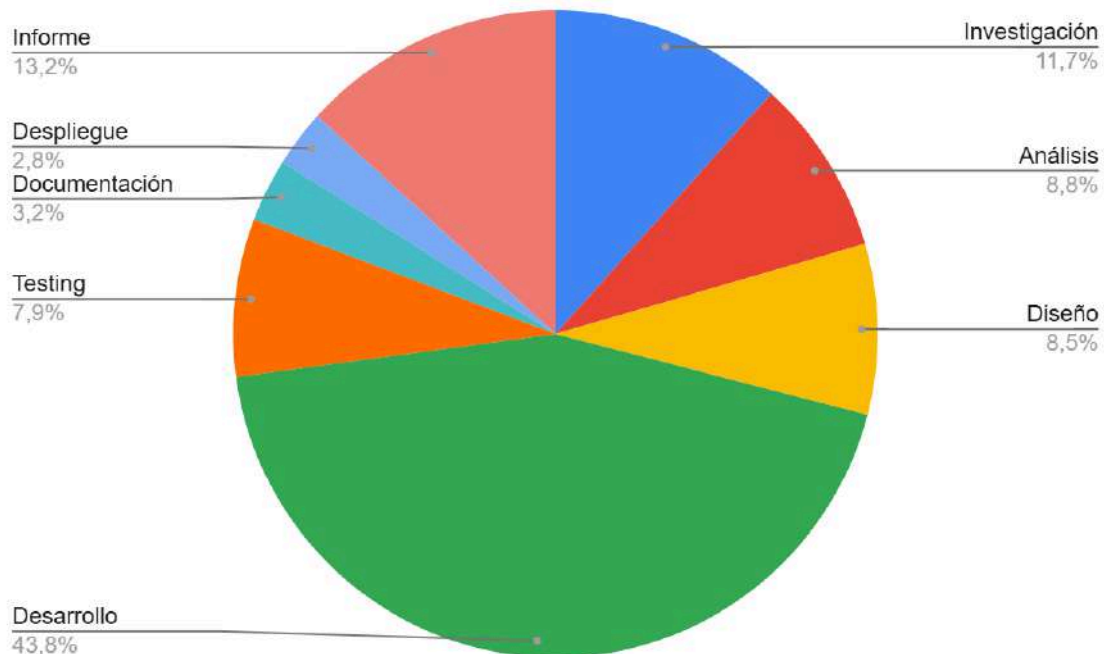
Asimismo, puede observarse una tendencia general a la sobreestimación en la planificación inicial. Este enfoque conservador permitió absorber imprevistos sin generar retrasos en las entregas y contribuyó al cumplimiento de los plazos establecidos.

5.2.4 Distribución total del esfuerzo

Finalmente, la Figura 6 muestra que la mayor parte del esfuerzo se concentró en la etapa de desarrollo, seguida por investigación y análisis. Esta distribución resulta coherente con la naturaleza del proyecto, caracterizado por un caso de uso principal y falta de conocimiento sobre imágenes satelitales al inicio del proyecto.

Figura 6

Gráfico de distribución total de horas del proyecto



En términos generales, si bien existieron desviaciones respecto a la planificación inicial, estas no impactaron negativamente en el cumplimiento de los objetivos del proyecto. Por

el contrario, el enfoque conservador adoptado en las estimaciones permitió una gestión más flexible del tiempo y facilitó la adaptación a los cambios surgidos durante el desarrollo.

5.3 Modalidad de trabajo

En esta sección se describe la modalidad de trabajo adoptada a lo largo del proyecto, tanto desde una perspectiva general, transversal a todas las etapas del proyecto, como desde un enfoque más específico, haciendo énfasis en aquellas fases que requirieron una organización particular.

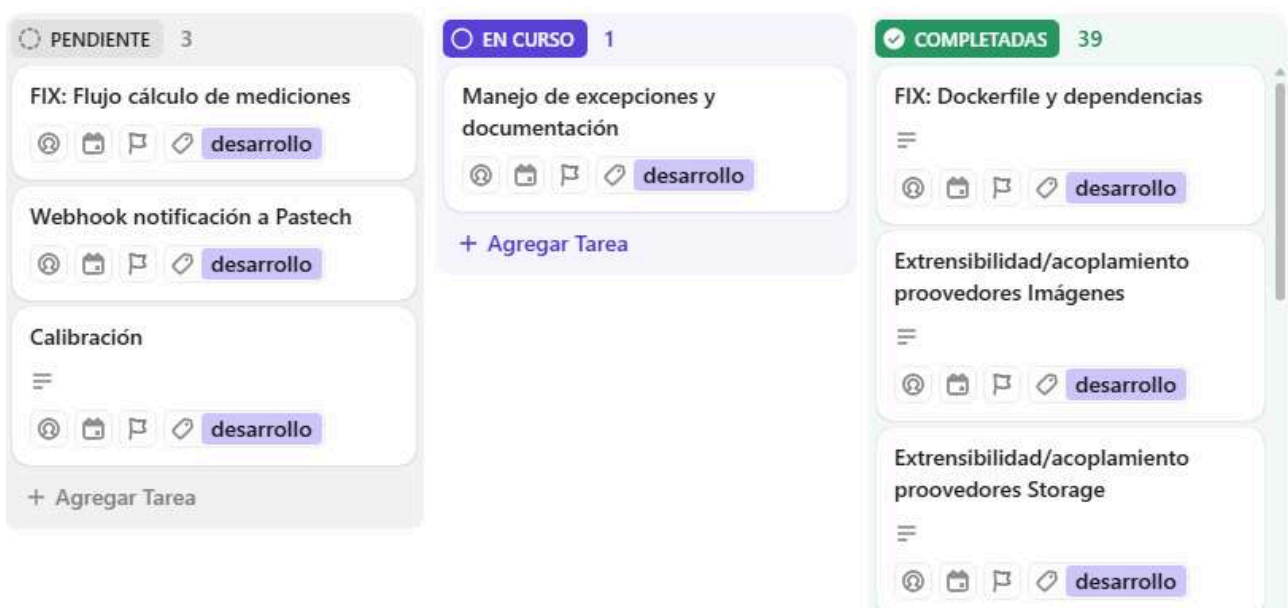
5.3.1 Organización y herramientas de trabajo

5.3.1.1 Organización de tareas

Para la organización de las tareas se utilizó una metodología ágil basada en el uso de un tablero Kanban. El objetivo principal de este enfoque fue visualizar el estado del trabajo y facilitar el seguimiento del avance del proyecto mediante columnas simples: Pendientes, en curso y completadas.

Figura 7

Captura de pantalla del tablero Kanban en etapa de desarrollo del proyecto



La herramienta seleccionada para implementar el tablero fue **ClickUp**, ya que permite gestionar tareas de forma sencilla y flexible. Dentro de sus funcionalidades básicas, la plataforma permite asignar categorías, descripciones y subtareas a cada ítem.

Si bien ClickUp ofrece un conjunto amplio de funcionalidades adicionales, muchas de ellas fueron descubiertas de forma progresiva y no llegaron a implementarse completamente en este proyecto. Un ejemplo de ello es el control de horas, que en este caso se llevó adelante mediante una hoja de cálculo independiente. Esta decisión brindó mayor versatilidad y permitió generar estadísticas personalizadas, útiles para analizar el avance del trabajo y la distribución del esfuerzo a lo largo del tiempo.

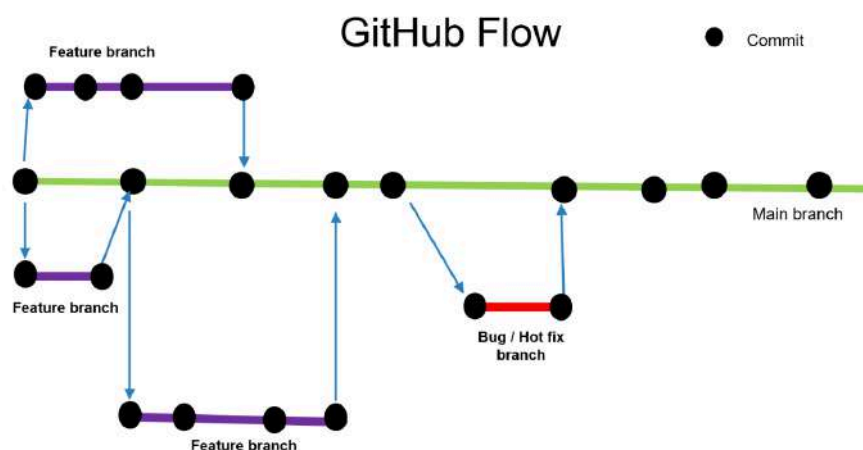
5.3.1.2 Control de código fuente

Otro aspecto fundamental del proyecto fue el control de código fuente, una práctica esencial en cualquier desarrollo de software. Para este fin se utilizó **GitHub**, principalmente debido al conocimiento previo sobre la plataforma y a que cumple con todas las necesidades del proyecto, incluyendo la gestión del repositorio y la integración con flujos de CI/CD.

Se optó por utilizar un *GitHub Flow*, manteniendo una rama principal que representa la versión estable del sistema y creando ramas independientes para cada funcionalidad o conjunto de funcionalidades relacionadas. Si bien esta estrategia muestra todo su potencial en equipos con múltiples desarrolladores trabajando en paralelo, su adopción permitió mantener un historial claro de cambios y facilitar la incorporación progresiva de nuevas funcionalidades, incluso en un desarrollo de carácter individual.

Figura 8

GitHub Flow



5.3.2 Metodología por etapas

A lo largo del proceso fue necesario interactuar y coordinar actividades con distintos actores, entre ellos el referente funcional, desarrolladores de la empresa Pastech, ingenieros agrónomos y analistas de datos. Estas instancias de intercambio resultaron fundamentales tanto para el relevamiento de requerimientos como para la validación de decisiones técnicas y resultados.

En este contexto, el trabajo se estructuró en distintas etapas ([5.1 Planificación inicial](#)), cada una con objetivos y dinámicas específicas. A continuación, se describe la metodología aplicada en cada una de estas fases, detallando cómo se abordaron las tareas y cómo se integraron los distintos aportes interdisciplinarios a lo largo del proyecto.

5.3.2.1 Investigación

Durante la etapa de investigación se trabajó bajo una modalidad de sprints semanales, ya que era necesario comprender en profundidad los objetivos del proyecto y evaluar progresivamente su viabilidad. Semana a semana se mantuvieron reuniones con el referente funcional, quien fue orientando la investigación y ayudando a enfocar el análisis hacia los requerimientos reales del negocio.

En cada sprint se presentaban los avances obtenidos, incluyendo ejemplos de imágenes satelitales provistas por distintos proveedores, con el objetivo de evaluar su utilidad y adecuación para el cálculo de los índices de vegetación y la estimación de biomasa. Esta dinámica permitió realizar ajustes tempranos en el enfoque de la investigación y descartar alternativas que no resultaban convenientes.

La participación se concentró principalmente en el referente funcional, ya que el foco estuvo puesto en comprender el problema, analizar las alternativas técnicas disponibles y delimitar el alcance del proyecto. En paralelo, se realizó una evaluación de los costos asociados a los distintos proveedores y alternativas tecnológicas, con el objetivo de comparar opciones y analizar la viabilidad económica de la solución propuesta.

5.3.2.2 Análisis

Al igual que en la etapa de investigación, el análisis se organizó mediante sprints semanales, orientados a relevar y documentar los requerimientos que luego se reflejarían en el desarrollo del sistema.

A diferencia de la etapa anterior, en esta fase comenzaron a participar otros actores, principalmente los programadores de la empresa, lo que permitió validar aspectos técnicos y funcionales desde etapas tempranas.

Además de las reuniones periódicas, la comunicación asincrónica tuvo un rol importante, ya que permitió resolver dudas puntuales de forma ágil, sin necesidad de esperar a una nueva reunión formal.

5.3.2.3 Diseño

En esta etapa no fue necesario un trabajo conjunto continuo con otros actores. A partir de la información relevada durante las fases de investigación y análisis, se realizó el diseño del sistema y se elaboró la documentación correspondiente.

Como instancia de validación, se presentaron a los desarrolladores algunos aspectos relevantes de la arquitectura propuesta y ciertos diagramas, como el diagrama de despliegue, con el objetivo de verificar que no existieran inconsistencias conceptuales o incompatibilidades con la infraestructura y los sistemas actuales de la empresa.

5.3.2.4 Desarrollo

La etapa de desarrollo también se abordó de forma iterativa, al igual que las fases de investigación y análisis. Sin embargo, las reuniones se realizaron con una mayor separación temporal, aproximadamente cada tres semanas.

Según las necesidades del momento, estas reuniones podían incluir al referente funcional, a los desarrolladores o a los ingenieros agrónomos. Generalmente, cada encuentro seguía una estructura similar:

- Presentación de un resumen de avances para poner en contexto a los participantes.
- Planteo de dudas o consultas dirigidas a los distintos referentes, ya sea para aclarar requerimientos o validar decisiones tomadas.

- Recepción de preguntas, sugerencias o recomendaciones por parte de los participantes.

Si bien algunas reuniones fueron más informales, en todas ellas se contaba con un conjunto de preguntas previamente preparadas. En la mayoría de los casos, además, se utilizó soporte visual o documentación para dejar registro de lo conversado.

Un aspecto recurrente de esta etapa fue que muchas de las reuniones derivaron en cambios en los requerimientos, lo que implicó actualizar la documentación de análisis y, en algunos casos, realizar ajustes en el diseño del sistema. Este punto representó una dificultad del proyecto y será retomado más adelante en el capítulo [8. Memoria del proyecto.](#)

5.3.2.5 Testing

El testing fue una etapa que se llevó a cabo de manera iterativa, intercalado con el desarrollo de la aplicación a lo largo del proyecto. La particularidad de esta fase fue que la distribución del esfuerzo no se realizó de la forma originalmente planificada, ya que al intentar incorporar pruebas en etapas tempranas fue necesario modificarlas como consecuencia de pequeños cambios en los comportamientos esperados del sistema.

Ante esta situación, se decidió concentrar un mayor esfuerzo en la primera iteración de testing, con el objetivo de construir una base sólida que pudiera sostener el proceso de validación en las fases posteriores. En esta instancia inicial se trabajó principalmente en la creación y configuración del entorno de pruebas, la definición de las clases y estructuras a utilizar, así como en la determinación de qué métodos serían testeados y bajo qué criterios.

Posteriormente, en la iteración final, se realizó otro esfuerzo significativo orientado a completar la implementación de las pruebas y a validar de manera efectiva el funcionamiento del sistema en su conjunto.

5.3.2.6 Despliegue

En la planificación inicial, la etapa de despliegue y las prácticas ligadas a DevOps habían sido concebidas como una actividad concentrada estrictamente al final del ciclo de vida del proyecto. Sin embargo, durante el proceso de desarrollo, se identificó que posponer esta fase representaba una limitación técnica importante, transformándose en la práctica

en un proceso continuo, iterativo y progresivo desde etapas tempranas. Esta adopción temprana no fue accidental, sino que estuvo profundamente condicionada por la naturaleza del sistema y su fuerte dependencia de integraciones con servicios en la nube como el cloud storage y la API de Planet.

Para validar el flujo completo y asíncrono de descarga y cálculo de imágenes satelitales, el proveedor requería endpoints públicos y accesibles en internet para la correcta entrega de los webhooks de notificación de tareas. Si bien existían alternativas para simular estos entornos (como el uso de proxies locales), se optó por adoptar de manera temprana prácticas reales de Integración y Despliegue Continuo (CI/CD).

Esta adopción progresiva de metodologías DevOps permitió realizar despliegues iterativos en Railway a medida que se completaban nuevas funcionalidades, operando bajo un esquema de entregas incrementales (MVP). Esta estrategia facilitó la validación temprana de la comunicación real entre los distintos servicios cloud, permitiendo detectar y corregir de forma prematura diversos problemas relacionados con la configuración del entorno, variables de entorno y dependencias, mitigando los riesgos de integración en la recta final del proyecto.

Una vez completado el despliegue técnico, se definió una estrategia de validación controlada para una futura puesta en producción. En una primera instancia, la implementación sería utilizada únicamente en un conjunto reducido de potreros de Pastech, permitiendo a los ingenieros agrónomos y desarrolladores evaluar el funcionamiento del sistema en un entorno real y escalar gradualmente la demanda. Esta aproximación buscaba evitar la contratación anticipada de planes de infraestructura de mayor costo antes de contar con una validación suficiente de la solución, permitiendo un crecimiento progresivo y acorde a los resultados obtenidos. No obstante, tal como se detalla en el apartado [8.4 Estado actual del sistema](#), la contratación del plan comercial por parte del comitente fue postergada, por lo que esta etapa de validación en producción no llegó a implementarse formalmente.

6. Sistema

6.1 Análisis

Como parte de la etapa de análisis, se elaboró un Documento de Especificación de Requerimientos de Software (ERS) con el objetivo de relevar, organizar y documentar los requerimientos del sistema. Dicho documento constituye la base para el diseño y posterior desarrollo de la solución propuesta.

En el presente informe se exponen únicamente los aspectos más relevantes de esta especificación. La versión completa del documento de requerimientos se encuentra disponible en los anexos, para su consulta en caso de ser necesario.

6.1.1 Requerimientos del sistema

A continuación se presentan los requerimientos del sistema identificados durante la etapa de análisis. Estos requerimientos describen el comportamiento esperado de la solución y las condiciones que debe cumplir para satisfacer las necesidades relevadas. Para una mejor organización, se los clasifica en requerimientos funcionales y no funcionales.

6.1.1.1 Requerimientos Funcionales (RF)

RF1. Autenticación

- **RF1.1** El sistema deberá autenticar cada solicitud mediante el uso de una API KEY en el header de las solicitudes.
- **RF1.2** La API KEY se deberá generar manualmente por un administrador de la base de datos. Esto se debe a que por ahora solo Pastech hará uso de la API y será monousuario.

RF2. Gestión de potreros

- **RF2.1** El sistema deberá permitir registrar en el proveedor de imágenes satelitales las nuevas áreas (potreros) a medida que se agreguen a la plataforma de Pastech.
- **RF2.2** Cuando se agregue un nuevo potrero a la plataforma de Pastech se deberá, además de registrar el potrero, cargar las últimas 5 mediciones disponibles.
- **RF2.3** El sistema deberá permitir registrar en el proveedor de imágenes satelitales las áreas ya existentes en la plataforma de Pastech.

- **RF2.4** El sistema deberá permitir eliminar áreas registradas en el proveedor de imágenes, para liberar cuota de uso.

RF3. Consulta y cálculo por potrero individual

- **RF3.1** El sistema deberá permitir solicitar información satelital para un potrero, dado el identificador y un rango de fechas.
- **RF3.2** El sistema deberá:
 - Persistir la imagen original correspondiente a cada fecha.
 - Procesar y persistir la imagen satelital con la capa NDVI.
 - Verificar que la cantidad de píxeles utilizables (porcentaje claros de píxeles) sea mayor a un umbral definido en un archivo de configuración.
 - Aplicar los algoritmos pertinentes para el cálculo de stock forrajero y persistir los resultados.
- **RF3.3** El sistema deberá notificar vía webhook al sistema de Pastech que el potrero se terminó de procesar.

RF4. Consulta y cálculo por múltiples potreros

- **RF4.1** El sistema deberá aceptar una lista de identificadores de potreros y un rango de fechas como parámetros.
- **RF4.2** Por cada potrero el sistema deberá replicar el comportamiento de RF3.2

RF5. Consulta y cálculo por usuario

- **RF5.1** El sistema deberá permitir obtener imágenes e índices calculados para todos los potreros asociados a un usuario, replicando el comportamiento de RF4 para la totalidad de los potreros.

RF6. Consulta del estado de la cuenta del proveedor satelital

- **RF6.1** El sistema deberá exponer un método para consultar:
 - Cuota de uso actual del plan del proveedor de imágenes (porcentaje y cantidad de áreas usadas).
 - Estado actual de facturación.
 - Número total de áreas y superficie (en km²) registradas.

RF7. Cálculo de índices

- **RF7.1** El sistema deberá calcular el índice de vegetación NDVI a partir de imágenes satelitales provistas.
- **RF7.2** El sistema deberá calcular la cantidad estimada de pasto por hectárea, utilizando fórmulas proporcionadas por Pastech.

RF8. Reproceso y corrección de índices

- **RF8.1** El sistema deberá permitir el recálculo de índices para fechas pasadas.
- **RF8.2** El sistema deberá aplicar las correcciones necesarias si se actualizan los algoritmos de cálculo o alguna calibración.

RF9. Filtrado de Nubosidad y Calidad de Píxeles

- **RF9.1** El sistema deberá implementar un algoritmo determinista de filtrado espacial utilizando la capa de clasificación de calidad de píxeles (Usable Data Mask) provista de forma nativa por el proveedor satelital. El sistema calculará el porcentaje de píxeles útiles libres de nubes y sombras para validar la viabilidad del cálculo del NDVI, descartando de forma automática aquellas imágenes que no superen el umbral de claridad configurado.

6.1.1.2 Requerimientos No Funcionales (RNF)

RNF1. Contratos e interfaces

- **RNF1.1** El sistema deberá contar con interfaces y contratos documentados que permitan una correcta integración con el ecosistema de Pastech, preferentemente, basadas en especificación OpenAPI .

RNF2. Integración con proveedor de imágenes satelitales

- **RNF2.1** El sistema deberá integrarse con los servicios del proveedor Planet.
- **RNF2.2** La integración deberá realizarse mediante los endpoints oficiales y bajo los términos de servicio vigentes.

RNF3. Persistencia de datos

- **RNF3.1** La API deberá almacenar los datos procesados en una base de datos relacional.
- **RNF3.2** El sistema deberá utilizar un motor de base de datos SQL, específicamente MySQL.

RNF4. Lenguaje de desarrollo

- **RNF4.1** La solución deberá ser desarrollada utilizando el lenguaje de programación Python.

RNF5. Entorno de ejecución

- **RNF5.1** El sistema deberá estar preparado para su despliegue en Railway como plataforma de ejecución.

RNF6. Seguridad

- **RNF6.1** Todos los métodos expuestos por la API deberán estar protegidos mediante API KEY.
- **RNF6.2** El sistema deberá rechazar peticiones no autenticadas o con keys inválidas, retornando el código de error correspondiente (401).

RNF7. Rendimiento

- **RNF7.1** Cualquier tarea que involucre un alto costo computacional (como la descarga de archivos pesados, manipulación de matrices geospaciales y cálculo del NDVI) no deberá bloquear el hilo principal de ejecución y deberá ser delegada a tareas asíncronas en segundo plano (background tasks).

RNF8. Operación bajo carga

- **RNF8.1** El sistema deberá ser capaz de soportar la recepción de solicitudes de cálculo para múltiples potreros simultáneamente, encolando los procesos.

RNF9. Resiliencia y tolerancia a fallos

- **RNF9.1** Los endpoints expuestos, particularmente aquellos que consumen webhooks de proveedores externos, deberán ser idempotentes. Si un evento de red se duplica, el sistema deberá identificar el estado previo y evitar la duplicación de inserciones en la base de datos y la repetición del cálculo intensivo.
- **RNF9.2** El sistema deberá implementar políticas de reintento (retries) frente a caídas transitorias de red o indisponibilidad temporal.

RNF10. Observabilidad

- **RNF10.1** El sistema deberá emitir logs para registrar eventos críticos del ciclo de vida de la aplicación. Estos deben incluir: inicio y fin del procesamiento asíncrono por potrero, fallos de autenticación, errores de comunicación con APIs de terceros y timeouts.

- **RNF10.2** Al operar en un entorno contenerizado, los logs generados deberán enviarse a la salida estándar (stdout/stderr) para permitir su fácil recolección, visualización y monitoreo a través del panel de control de la infraestructura en la nube, facilitando la detección temprana de anomalías en producción.

6.1.2 Casos de uso

Figura 9

Diagrama de casos de uso



6.1.3 Requisitos de interfaces externas

6.1.3.1 Interfaces de Usuario

- La API no tiene una interfaz de usuario. Las interacciones con los usuarios finales se gestionan a través del Sistema Pastech, que consume los servicios de la API.

6.1.3.2 Interfaces de Hardware

- No se especifican requisitos de hardware directos. La API se despliega en la plataforma Railway.

6.1.3.3 Interfaces de Software

Interfaz con Proveedor Planet

- **Descripción:** La API se integra con los servicios de Planet para obtener imágenes satelitales, registrar/eliminar potreros y consultar el estado de la cuenta.
- **Protocolo:** HTTPS/RESTful API.
- **Métodos:** GET, POST , DELETE.
- **Formato de Datos:** GeoJSON para coordenadas, JSON para respuestas.
- **Autenticación:** Token Planet.
- **Endpoints:** Según la documentación oficial de Planet.
- **Restricciones:** Cumplimiento de los términos de servicio de Planet.

Interfaz con Sistema Pastech

- **Descripción:** La API provee datos al Sistema Pastech en formato JSON o insertando en su propia base de datos para su integración en la plataforma.
- **Protocolo:** HTTPS/RESTful API.
- **Métodos:** GET, POST, PUT, DELETE, PATCH.
- **Formato de Datos:** JSON.
- **Autenticación:** Requiere API KEY válida.

6.1.3.4 Interfaces de Comunicaciones

- **Descripción:** Utiliza HTTPS.
- **Requisitos:** Conexión a Internet.

6.2 Diseño

De manera similar a la sección de Análisis, en este apartado se describen los principales aspectos relacionados con el **diseño del sistema**. La documentación completa del diseño se encuentra incluida en el anexo correspondiente.

6.2.1 Arquitectura del sistema

6.2.1.1 Estilo arquitectónico elegido

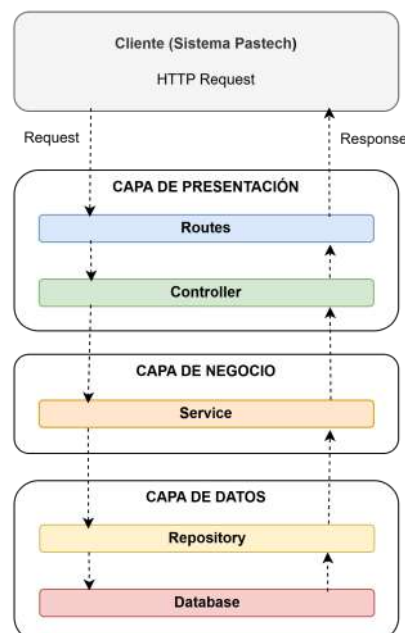
Se optó por desarrollar un monolito con Arquitectura en capas con enfoque de Clean Architecture, donde cada capa tiene una responsabilidad específica y sólo interactúa con la capa inmediata inferior.

- **Capa de Presentación:**
 - Routes: define los endpoints que expone la API al Sistema Pastech.
 - Controllors: coordina las solicitudes entre rutas y servicios. Hace las respectivas validaciones y orquesta la lógica de negocio.
- **Capa de Lógica de Negocio:**
 - Services: Implementa la lógica principal, incluyendo cálculos e integración con servicios externos.
- **Capa de Persistencia:**
 - Repository: gestiona el acceso a MySQL (database).

La arquitectura en capas se ve reflejada de la siguiente manera:

Figura 10

Arquitectura del sistema



La comunicación fluye desde arriba hacia abajo, lo que es típico de una arquitectura en capas. Cada capa tiene una responsabilidad específica y sólo interactúa con su inmediato inferior.

Clean Architecture propone una arquitectura centrada en la lógica de negocio, con dependencias que apuntan hacia el interior (entidades) y adaptadores externos (ej. API de Planet) en la periferia.

En la API se definirá una clase denominada `planet_satellite_provider.py`, la cual implementa la interfaz `satellite_service_provider.py` y actúa como un adaptador externo. Este diseño permite encapsular y aislar la lógica específica de integración con el proveedor Planet, evitando que el resto del sistema dependa directamente de sus particularidades.

De esta manera, en caso de requerirse la incorporación de un nuevo proveedor satelital, será suficiente con desarrollar una nueva clase que implemente la interfaz `satellite_service_provider.py`, sin necesidad de modificar la lógica principal del sistema.

También, habrá modelos que representen las entidades centrales (ej. `potrero_model.py`), y la separación de `services` y `repository` asegura que la lógica de negocio no dependa directamente de la base de datos o servicios externos, alineándose con los principios de Clean Architecture.

6.2.1.2 Justificación de la arquitectura

La aplicación se desarrolla bajo una arquitectura monolítica con una arquitectura en capas y un enfoque basado en Clean Architecture.

Esta organización permite mantener una separación clara entre las responsabilidades del sistema, facilitando su mantenimiento, pruebas y escalabilidad futura.

Las dependencias entre capas están diseñadas para que la lógica de negocio permanezca independiente de detalles de infraestructura o presentación, siguiendo los principios definidos en Clean Architecture.

Este tipo de arquitectura aporta importantes beneficios en atributos de calidad del software, pero también implica ciertos tradeoffs.

6.2.1.2.1 Beneficios en Atributos de Calidad

Tabla 2

Beneficios de la arquitectura en capas sobre los atributos de calidad

Atributo de Calidad	Cómo lo beneficia
Mantenibilidad	La separación clara de responsabilidades hace más fácil modificar o extender funcionalidades.
Testabilidad	Las capas desacopladas facilitan pruebas unitarias y de integración.
Portabilidad	Al desacoplar la lógica de negocio de la infraestructura, se puede migrar fácilmente a otra base de datos, API, o incluso dividir en microservicios.
Seguridad	Las responsabilidades bien delimitadas y las capas intermedias permiten aplicar medidas de autenticación y validación en puntos estratégicos.

6.2.1.2.1 Trade-offs

La arquitectura adoptada prioriza ciertos atributos de calidad en función de los objetivos del proyecto, lo que implica, de manera inevitable, comprometer otros atributos en menor medida. En particular, las decisiones orientadas a mejorar la mantenibilidad, testabilidad y portabilidad introducen impactos en atributos como el rendimiento, la simplicidad y la rapidez de desarrollo inicial. Estos trade-offs son consecuencia directa de favorecer una arquitectura más estructurada y desacoplada, adecuada para la evolución futura del sistema, por sobre soluciones más simples u optimizadas para escenarios de corto plazo.

Tabla 3

Trade-offs entre atributos de calidad de la arquitectura

Atributo priorizado	Atributo afectado	Descripción del trade-off
Mantenibilidad	Simplicidad	La separación en capas y componentes mejora la mantenibilidad del sistema, pero incrementa la

complejidad estructural y conceptual de la solución.

Testabilidad Tiempo de desarrollo

El diseño facilita la validación del sistema, pero requiere un mayor esfuerzo inicial en la definición de interfaces, mocks y configuraciones.

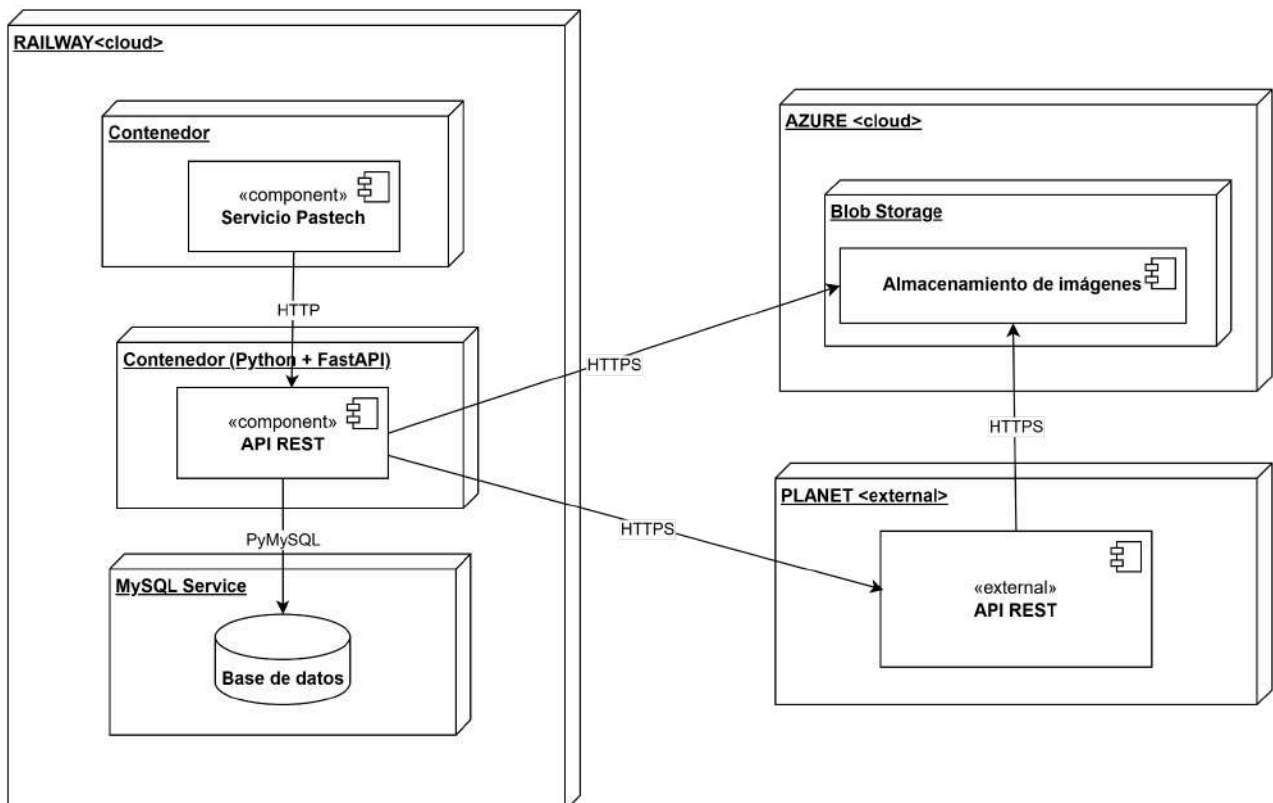
Portabilidad Rendimiento

El uso de adaptadores y capas intermedias favorece la independencia de la infraestructura, pero introduce una leve sobrecarga en la ejecución.

6.2.1.3 Diagrama de despliegue

Figura 11

Diagrama de despliegue



6.2.2 Diseño de base de datos

Como una definición clave en el diseño del sistema, se estableció que la API compartirá el mismo entorno de persistencia que la plataforma central de Pastech. El uso de una base de datos MySQL compartida con el ecosistema principal representa una decisión arquitectónica consciente, fundamentada en un análisis de costo-beneficio (trade-off). Si bien el uso compartido de un motor genera un acoplamiento a nivel de infraestructura, el sistema mantiene intacta la independencia de sus componentes lógicos, respetando el principio de Clean Architecture. Esta decisión prioriza la eficiencia operativa y el rendimiento general de la solución: evita la latencia de red y la saturación de ancho de banda que implicaría transferir masivamente los resultados de los cálculos espaciales mediante peticiones HTTP cada vez que se cargan nuevas mediciones.

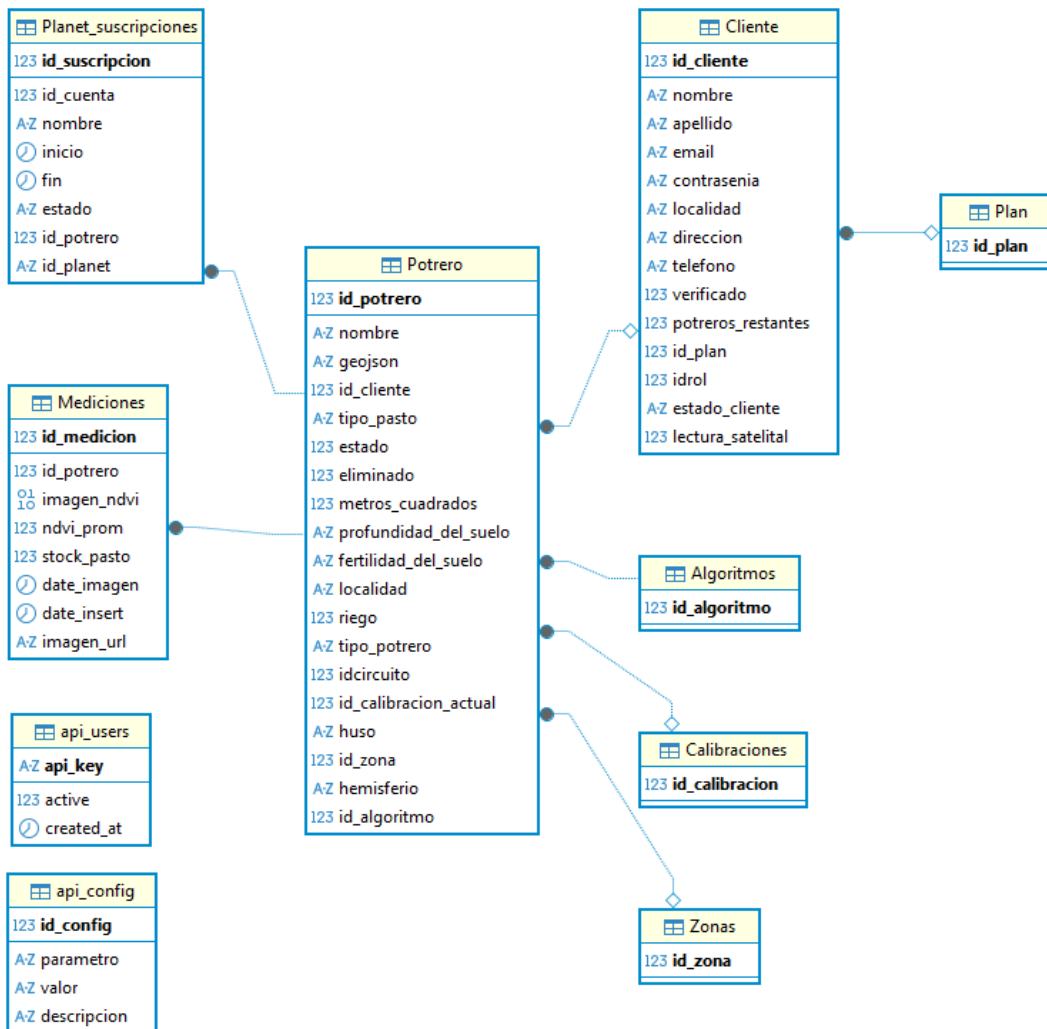
A pesar de este acoplamiento físico, el atributo de calidad de portabilidad se encuentra garantizado a nivel de código mediante el patrón de diseño Repository. La lógica de negocio se encuentra totalmente aislada de la tecnología de persistencia. Si en el futuro la escalabilidad exigiera desacoplar físicamente las bases de datos, bastaría con modificar la implementación concreta del adaptador del repositorio para comunicarse con otro servicio, sin alterar las reglas de negocio.

Debido a este enfoque compartido, no fue necesario diseñar la base de datos desde cero, sino comprender su estructura existente y realizar ajustes estratégicos mediante la incorporación de nuevos atributos, tablas y relaciones (como las entidades de suscripciones y mediciones satelitales). Este proceso permitió adaptar el modelo a los requerimientos específicos del sistema desarrollado.

A continuación, se presenta un diagrama (Figura 12) cuyo objetivo es facilitar la comprensión del dominio del problema y visualizar las principales entidades del sistema junto con sus relaciones. Para optimizar la claridad visual del modelo, se exponen únicamente aquellos atributos que resultan relevantes para el desarrollo de esta solución.

Figura 12

Diseño lógico de la base de datos del sistema



6.2.2.2 Modelo Lógico relacional

Cliente (id_cliente, nombre, apellido, email, contraseña, localidad, dirección, telefono, verificado, potrerost_restantes, id_plan, idrol, estado_cliente, lectura_satelital)

PK = { id_cliente }

CK = { id_cliente }

FK = { id_plan }

- Cliente.id_plan debe estar en Plan.id_plan o puede ser NULL.
- Plan.id_plan puede o no estar en Cliente.id_plan.

Potrero (id_potrero, nombre, geojson, id_cliente, tipo_pasto, estado, eliminado, metros_cuadrados, profundidad_del_suelo, fertilidad_del_suelo, localidad, riego,

tipo_potrero, idcircuito, *id_calibracion_actual*, huso, *id_zona*, hemisferio, *id_algoritmo*)

PK = { id_potrero }

CK = { id_potrero }

FK = { id_cliente, id_zona, id_calibracion_actual, id_algoritmo }

- Potrero.id_cliente debe estar en Cliente.id_cliente (no puede ser NULL).
- Potrero.id_zona debe estar en Zonas.id_zona o puede ser NULL.
- Potrero.id_calibracion_actual debe estar en Calibraciones.id_calibracion o puede ser NULL.
- Potrero.id_algoritmo debe estar en Algoritmos.id_algoritmo o puede ser NULL.

Planet_suscripciones (*id_suscripcion*, *id_cuenta*, nombre, inicio, fin, estado, *id_potrero*)

PK = { id_suscripcion }

CK = { id_suscripcion }

FK = { id_cuenta, id_potrero }

- Planet_suscripciones.id_cuenta debe estar en Planet_cuentas.id_cuenta (no puede ser NULL).
- Planet_suscripciones.id_potrero debe estar en Potrero.id_potrero (no puede ser NULL).

Mediciones (*id_medicion*, *id_potrero*, imagen_ndvi, ndvi_prom, stock_pasto, date_imagen, date_insert, imagen_url)

PK = { id_medicion }

CK = { id_medicion }

FK = { id_potrero }

- Mediciones.id_potrero debe estar en Potrero.id_potrero (no puede ser NULL).

6.2.2.3 Justificación del modelo

6.2.2.3.1 Normalización

- El modelo evita la redundancia separando entidades como Cliente, Plan, Potrero, Zonas, Algoritmos, etc.

- Se respetan las formas normales (1FN, 2FN, 3FN), asegurando integridad y facilidad de mantenimiento.

6.2.2.3.2 Elección de relaciones

- Las relaciones 1:N y N:1 reflejan la lógica de negocio: un cliente tiene varios potreros, pero un potrero solo puede pertenecer a un cliente.
- Las relaciones entre Potrero y otras entidades como Algoritmos, Calibraciones, y Zonas permiten modular y actualizar configuraciones de forma independiente.
- Planet_suscripciones y Mediciones permiten manejar múltiples suscripciones satelitales por potrero sin duplicar información.

6.2.3 Diseño de Tests

La estrategia de testing del sistema se define con un enfoque centrado en los casos de uso, priorizando la validación de los escenarios funcionales más relevantes. Para ello, se establece una combinación de pruebas unitarias y pruebas de integración.

Las pruebas unitarias se plantean para métodos críticos de la lógica de negocio, como el cálculo del NDVI, con el objetivo de validar su correcto funcionamiento de manera aislada y asegurar la consistencia de los resultados generados por el sistema.

Por otro lado, las pruebas de integración se diseñan para validar el comportamiento del sistema de forma completa, a través de la invocación directa de los endpoints expuestos por la API. Este enfoque permite verificar la correcta interacción entre las distintas capas de la arquitectura y el cumplimiento de los casos de uso definidos.

Con el fin de evitar dependencias externas durante el proceso de testing, se define el uso de mocks para los proveedores externos, tanto para los servicios de almacenamiento (blob storage) como para los servicios de imágenes satelitales. Esta decisión permite aislar el sistema bajo prueba y evitar el consumo de cuotas de uso de servicios de terceros.

Asimismo, se establece la utilización de una base de datos exclusiva para el entorno de testing. Para su correcto funcionamiento, se prevé la ejecución de scripts de inicialización previos a la ejecución de los tests, con el objetivo de preparar el estado de la base de

datos, así como la ejecución de verificaciones posteriores orientadas a validar los resultados obtenidos durante el proceso de prueba.

6.2.4 Diseño de despliegue / DevOps

6.2.4.1 Entorno de ejecución

El entorno de ejecución será Railway. Esta decisión se tomó debido a que los servicios actuales de Pastech corren sobre esta plataforma. Si bien se podría haber optado por otros proveedores, esto ahorrará costo de mantenimiento, debido a que el equipo técnico no tendrá que familiarizarse con una nueva plataforma y mantener los servicios.

Por otro lado, el sistema requiere un servicio de almacenamiento que permita al proveedor de imágenes satelitales depositar los archivos generados. Para esta necesidad se realizó una evaluación de alternativas y se optó por **Azure Blob Storage** durante la etapa de desarrollo, principalmente debido a que ofrece un plan gratuito para estudiantes, lo que permite trabajar sin incurrir en costos adicionales en el contexto académico del proyecto.

A futuro, en un entorno productivo, se contempla la posibilidad de montar un servicio propio de almacenamiento dentro de Railway utilizando herramientas como **MinIO**, una solución open source compatible con las interfaces y protocolos de Azure Blob Storage. Esta compatibilidad facilita una eventual migración entre servicios sin necesidad de realizar cambios significativos en la lógica de la aplicación.

El análisis de viabilidad económica asociado al almacenamiento y la infraestructura en la nube para entornos productivos se detalla formalmente en la [sección 6.3](#).

6.2.4.2 Contenedores

La API fue diseñada para ejecutarse dentro de contenedores, lo que permite asegurar portabilidad, aislamiento del entorno y consistencia entre los distintos ambientes (desarrollo, testing y producción).

Además, se configuró un flujo de **integración y despliegue continuo (CI/CD)** mediante GitHub Actions, el cual se encarga de desplegar la aplicación ante cambios en el

repositorio. Este enfoque contribuye a mejorar la calidad del software y a reducir errores durante el proceso de despliegue.

6.2.5 Tecnologías utilizadas

En esta sección se presentan las principales tecnologías empleadas en el desarrollo del sistema, organizadas por categoría. La selección de cada una de ellas responde a criterios de compatibilidad con la arquitectura definida, facilidad de mantenimiento y adecuación a los requerimientos técnicos del proyecto.

6.2.5.1 Backend y framework

- **Python 3.11:** Lenguaje principal del proyecto. Fue elegido por su ecosistema maduro, amplia adopción en el procesamiento de datos geoespaciales y buena integración con librerías científicas.
- **FastAPI:** Framework web asincrónico utilizado para la implementación de la API. Permite definir endpoints de forma clara, validar datos automáticamente mediante esquemas y generar documentación interactiva de manera nativa.
- **Uvicorn:** Servidor ASGI utilizado para la ejecución de la aplicación, compatible con el enfoque asincrónico de FastAPI.
- **SQLModel:** ORM basado en SQLAlchemy y Pydantic, utilizado para el modelado de entidades y el acceso a la base de datos. Facilita la validación de datos y mantiene consistencia entre modelos de dominio y persistencia.

6.2.5.2 Base de datos

- **MySQL:** Base de datos relacional principal del sistema. Se utilizó debido a que ya forma parte del ecosistema tecnológico de Pastech, lo que simplifica la integración y el mantenimiento.
- **PyMySQL:** Driver utilizado para la conexión entre la aplicación y la base de datos MySQL.

6.2.5.3 Procesamiento de imágenes satelitales

- **Planet SDK:** Librería utilizada para la integración con el proveedor de imágenes satelitales durante el desarrollo. Su uso se encuentra encapsulado mediante una abstracción para evitar dependencias directas con el proveedor.

- **Rasterio:** Herramienta clave para la lectura y escritura de archivos geoespaciales (GeoTIFF), utilizada en el procesamiento de imágenes y extracción de valores espectrales.
- **NumPy:** Biblioteca de computación numérica empleada para realizar los cálculos de índices de vegetación, como el NDVI.
- **Scikit-learn:** Utilizada para implementar regresiones lineales en tareas de calibración y análisis comparativo de resultados.

6.2.5.4 Almacenamiento en la nube

- **Azure Blob Storage:** Servicio de almacenamiento utilizado durante el desarrollo para el manejo de imágenes satelitales.
- **MinIO:** Alternativa contemplada a futuro para entornos productivos, compatible con interfaces de almacenamiento tipo S3, lo que facilita una eventual migración sin cambios significativos en la lógica del sistema.

6.2.5.6 Validación, serialización y comunicación

- **Pydantic:** Utilizado para la validación de datos de entrada y salida, así como para la definición de esquemas de request y response de la API.
- **Requests:** Cliente HTTP utilizado para el manejo de webhooks y comunicaciones con servicios externos.
- **python-dotenv:** Herramienta empleada para la gestión de variables de entorno en los distintos entornos de ejecución.

6.2.5.7 Testing y análisis

- **pytest:** Framework principal para la ejecución de pruebas unitarias y de integración.
- **pytest-mock / unittest.mock:** Utilizados para simular dependencias externas, como proveedores satelitales o servicios de almacenamiento.
- **Matplotlib y Pandas:** Empleados en scripts y notebooks auxiliares para el análisis de resultados y la visualización de métricas como NDVI y comparaciones entre proveedores.

6.2.5.8 Infraestructura y despliegue

- **Docker:** Utilizado para ejecutar la API en contenedores, asegurando consistencia entre los entornos de desarrollo, testing y producción.
- **Railway:** Plataforma de despliegue seleccionada para alojar la aplicación, alineada con la infraestructura existente de Pastech.

6.2.6 Patrones de diseño aplicados

A partir de la arquitectura definida y los objetivos del proyecto, se aplicaron distintos patrones de diseño que permiten mejorar la mantenibilidad, extensibilidad y claridad del sistema.

- **Arquitectura en capas:** Este patrón permite desacoplar la lógica de presentación, la lógica de negocio y el acceso a datos. Facilita el testing y evita dependencias directas entre capas no relacionadas.
- **Repository Pattern:** El acceso a la base de datos se encapsula en repositorios, evitando que la lógica de negocio dependa directamente del ORM o del motor de base de datos. Esto favorece la mantenibilidad y la posibilidad de cambios futuros.
- **Provider Pattern:** Los servicios externos, como proveedores de imágenes satelitales o almacenamiento en la nube, se integran mediante interfaces comunes. Esto evita acoplamientos directos con implementaciones específicas (por ejemplo, Planet o Azure).
- **Factory Pattern:** Se utiliza un patrón Factory para instanciar proveedores satelitales o servicios de almacenamiento en función de la configuración del sistema. Este enfoque permite agregar nuevos proveedores sin modificar la lógica central de la aplicación.

6.2.7 Manejo de idempotencia, resiliencia y tolerancia a fallos

Debido a la naturaleza distribuida y asíncrona del sistema, donde la API interactúa con proveedores externos (Planet, Azure) y notifica los resultados hacia la plataforma de Pastech mediante webhooks, se diseñaron e implementaron mecanismos específicos para garantizar la resiliencia, la tolerancia a fallos y la consistencia de los datos desde tres perspectivas operativas fundamentales:

- **Recepción y Procesamiento de Webhooks (Desde Planet):** La API está preparada para manejar la posible duplicación de notificaciones asincrónicas por parte del proveedor satelital. Si ocurriera un fallo de red que impidiera la recepción del ACK (código HTTP 200) y Planet reintentara el envío del webhook, el sistema realiza una verificación previa en la base de datos para constatar si el cálculo para esa imagen específica (identificada por `id_potrero` y `date_imagen`) ya ha sido procesado. De existir, se omite el procesamiento pesado, evitando la duplicación de tareas de cálculo intensivo, transferencias de red y almacenamiento redundante.
- **Emisión de Notificaciones Asincrónicas (Hacia Pastech):** El envío de confirmaciones y resultados hacia los endpoints del cliente está sujeto a fallos transitorios de conectividad. Para mitigar esto, el sistema incorpora un algoritmo de reintentos automáticos basado en backoff exponencial. Ante un fallo en la entrega (timeouts o errores HTTP 5xx), la API programa un nuevo intento incrementando el tiempo de espera de manera exponencial entre cada iteración, evitando saturar al receptor y elevando la tasa de éxito.
Asimismo, para mitigar los efectos de una entrega duplicada (por ejemplo, si el cliente procesa el cálculo pero el ACK de red se pierde), cada payload emitido incluye un identificador único global de evento (`event_id`) generado vía UUIDv4. Esto traslada formalmente al receptor la capacidad de registrar los eventos procesados; ante un `event_id` idéntico subsecuente, el consumidor puede ignorar la petición de forma segura, garantizando la idempotencia de extremo a extremo.
- **Peticiones de Recálculo del Cliente (Desde Pastech):** Desde la perspectiva de la interacción directa con las solicitudes del usuario, el sistema implementa una lógica de actualización o inserción (Upsert). Dado que el requerimiento funcional exige permitir el recálculo de mediciones para fechas pasadas, si Pastech solicita procesar una medición correspondiente a una fecha que ya existe en los registros, el sistema actualiza la fila existente en lugar de insertar un nuevo registro con un `id_medicion` autoincremental distinto. De esta manera, se evita la duplicación de datos y se asegura la consistencia del modelo relacional.

6.2.8 Especificación de la API

A continuación se presenta la especificación formal de la interfaz de programación del sistema, documentada bajo el estándar de OpenAPI. Este documento en formato JSON actúa como el contrato estricto de comunicación entre la plataforma central de Pastech y el servicio de procesamiento satelital.

- **Rutas y Operaciones (paths):** La semántica de los endpoints disponibles, incluyendo la gestión asincrónica de cálculos, el alta/baja de potrereros y la recepción de webhooks.
- **Modelos de Datos (schemas):** La estructura exacta de los payloads de entrada y salida.
- **Manejo de Errores Tipificados:** La definición de las respuestas HTTP, incluyendo los errores y sus respectivos mensajes de detalle generados por el framework.
- **Políticas de Seguridad:** La exigencia documentada de inyectar las credenciales de acceso mediante el encabezado personalizado x-api-key para todas las rutas protegidas.

Debido a su extensión y naturaleza estrictamente técnica, la definición estructurada de la arquitectura de la API y su contrato completo en formato JSON se incluyen detalladamente en el anexo [“Especificación de la API”](#) al final del presente documento.

6.3 Estructura de costos

6.3.1 Estrategia de Despliegue

Nota: datos de Agosto 2025.

Se propone una implementación progresiva del servicio, iniciando con el procesamiento exclusivo de parcelas registradas en Uruguay.

Esta aproximación inicial permite:

- **Reducción de costos operativos:** el costo anual de imágenes satelitales para Uruguay es aproximadamente un 50% menor que para Argentina.
- **Validación temprana:** posibilita obtener retroalimentación real de usuarios en un entorno productivo controlado antes de la expansión a todo el territorio argentino.
- **Ajustes iterativos:** facilita la optimización de algoritmos, arquitectura y flujos de datos sin incurrir en los costos máximos desde la fase inicial.

6.3.1.1 Fase 1: Implementación Inicial (Uruguay, Paraguay, Chile)

- **Cobertura:** 5.000 hectáreas
- **Costo anual de imágenes satelitales:** USD 2.700/año (USD 0,54/ha/año)
- **Costo anual API en la nube:** USD 5/mes -> USD 60/año
- **Costo anual almacenamiento imágenes (20GB/mes):** USD 3/mes -> USD 36/año

Cálculo de capacidad de almacenamiento

$Capacidad \approx N_{potreros} * \text{área (km}^2) * 1,78 \text{ MB/km}^2 * (1 + \text{overhead}) * \text{días}$

- $N_{potreros} = 5000$
- Área = 0,01 km² (1 ha)
- Tamaño imagen = 1,78 MB/km²
- Overhead (archivos y metadatos) = 0,3
- Días = 180 (90 días * 2 capturas/día)

Resultado: 20.826 MB ≈ 20 GB

6.3.1.2 Fase 2: Escalado a Producción (Argentina)

Una vez validados los resultados y alcanzada la adopción esperada, se procederá a contratar un plan de mayor capacidad para cubrir el mercado argentino.

- **Cobertura:** 15.000 hectáreas
- **Costo anual de imágenes satelitales:** USD 17.300/año (USD 1,15/ha/año)
- **Costo anual API en la nube:** USD 15/mes -> USD 180/año
- **Costo anual almacenamiento imágenes (60 GB/mes):** USD 9/mes -> USD 108/año

Cálculo de capacidad de almacenamiento

Mismo cálculo anterior, ajustando **$N_{potreros} = 15.000$**

Resultado: 62.478 MB ≈ 60 GB

Este enfoque progresivo permite minimizar la inversión inicial, reducir riesgos técnicos y financieros, y garantizar la escalabilidad controlada del sistema. La metodología asegura que la arquitectura y los procesos de negocio se validen en un entorno real antes de su despliegue a gran escala.

7. Proceso de Cálculo y Análisis de Imágenes Satelitales

7.1 Selección del proveedor de imágenes satelitales

Uno de los aspectos más relevantes del trabajo fue la selección de un proveedor de imágenes satelitales que cumpliera con los requerimientos técnicos de Pastech. Esta etapa resultó particularmente desafiante, ya que previo al inicio del proyecto el conocimiento específico sobre imágenes satelitales y sensores remotos era limitado. En consecuencia, fue necesario realizar una investigación sobre los distintos tipos de imágenes disponibles, sus características técnicas y su aplicabilidad en el contexto agropecuario, lo que implicó una curva de aprendizaje considerable.

Hasta el momento, Pastech utilizaba exclusivamente imágenes provenientes del satélite **Sentinel**, el cual captura información de una misma región aproximadamente cada cinco días. Si bien Sentinel constituye una fuente de datos confiable y de acceso gratuito, presenta una limitación importante relacionada con las condiciones atmosféricas. En particular, cuando la captura de una imagen coincide con presencia de nubosidad, resulta inutilizable, lo que puede provocar períodos prolongados sin datos disponibles. En escenarios climáticos desfavorables, esta situación puede extenderse a diez, quince días o más, afectando directamente la continuidad y confiabilidad de los indicadores calculados.

Dado que el objetivo del proyecto es estimar el stock forrajero de manera frecuente y actualizada, la dependencia exclusiva de Sentinel representaba una restricción significativa. Por este motivo, uno de los criterios fundamentales en la selección del nuevo proveedor fue la **mayor frecuencia de captura de imágenes**, permitiendo reducir la probabilidad de pérdida de mediciones debido a nubosidad y mejorar la disponibilidad temporal de datos.

Otro aspecto crítico considerado fue la **resolución espacial de las imágenes**. Sentinel ofrece una resolución adecuada para potreros homogéneos; sin embargo, en situaciones donde existe heterogeneidad en el terreno una resolución más alta permite obtener estimaciones más precisas y representativas del estado real del pastizal. La incorporación de imágenes con mayor detalle espacial contribuye directamente a mejorar la calidad de

los índices de vegetación calculados y, en consecuencia, de los modelos de estimación de biomasa.

Finalmente, se consideró como requisito indispensable la **compatibilidad con el flujo de trabajo existente** en Pastech. El nuevo proveedor debía permitir trabajar en conjunto con Sentinel, minimizando la necesidad de modificar los algoritmos de estimación ya implementados. Este criterio buscó preservar la lógica de cálculo previamente validada, facilitando la integración de nuevas fuentes de datos sin introducir cambios estructurales significativos en el sistema.

En función de lo anterior, los principales requerimientos definidos para la selección del proveedor de imágenes satelitales fueron los siguientes:

- Mayor frecuencia de captura de imágenes.
- Mayor resolución espacial en comparación con Sentinel.
- Compatibilidad con Sentinel y con los algoritmos de estimación existentes.
- Capacidad de integración dentro del sistema propuesto.

7.2 Tipos de imágenes satelitales y sus características

Las imágenes satelitales pueden clasificarse según distintas características técnicas que determinan su utilidad para aplicaciones específicas. En el contexto de este proyecto, resulta fundamental comprender estas propiedades para evaluar el impacto que tienen en el cálculo de índices de vegetación y en la estimación del stock forrajero. Las principales dimensiones consideradas son la **resolución espacial**, la **resolución temporal** y la **resolución espectral**.

7.2.1 Resolución espacial

La resolución espacial hace referencia al tamaño del área del terreno que representa cada píxel de la imagen. A menor tamaño de píxel, mayor nivel de detalle se obtiene sobre la superficie observada. En aplicaciones agropecuarias, esta característica es especialmente relevante, ya que los potreros pueden presentar variaciones significativas en la cobertura vegetal dentro de áreas relativamente pequeñas.

Imágenes de baja resolución pueden resultar suficientes en escenarios homogéneos, pero tienden a generar promedios que ocultan variaciones locales. En cambio, imágenes de mayor resolución espacial permiten identificar con mayor precisión zonas con distinto

estado vegetativo, mejorando la calidad de los índices calculados y reduciendo errores en la estimación de biomasa.

7.2.2 Resolución temporal

La resolución temporal indica la frecuencia con la que un satélite captura imágenes de una misma región. Este factor es crítico cuando se requiere monitorear cambios frecuentes en el estado de la vegetación, como ocurre en sistemas ganaderos donde el crecimiento del pasto y el consumo animal pueden variar rápidamente.

Una baja resolución temporal implica mayores intervalos entre capturas, lo que aumenta la probabilidad de perder información relevante, especialmente en presencia de nubosidad. Por el contrario, una alta frecuencia de captura permite disponer de múltiples imágenes en períodos cortos, incrementando las posibilidades de obtener datos utilizables y mejorar la continuidad del monitoreo.

7.2.3 Resolución espectral

La resolución espectral se relaciona con la cantidad y el ancho de las bandas del espectro electromagnético que un sensor es capaz de captar. Para el cálculo de índices de vegetación, resulta indispensable contar con bandas en el espectro visible y en el infrarrojo cercano, ya que estas permiten evaluar la actividad fotosintética de la vegetación.

Sensores con una adecuada resolución espectral facilitan el cálculo de índices como el NDVI y otros indicadores complementarios, permitiendo caracterizar el estado fisiológico del pastizal de forma más precisa. En este proyecto, se priorizó la disponibilidad de bandas compatibles con los algoritmos de cálculo ya utilizados por Pastech.

7.3 Comparación entre Sentinel y proveedores comerciales de imágenes satelitales

La elección del proveedor de imágenes satelitales implicó una evaluación comparativa entre la solución previamente utilizada por Pastech, basada en Sentinel, y distintas alternativas comerciales disponibles en el mercado.

7.3.1 Sentinel

Sentinel es un programa de observación de la Tierra desarrollado por la Agencia Espacial Europea (ESA), que ofrece imágenes satelitales de acceso libre y gratuito. Entre sus principales ventajas se destacan su amplia cobertura geográfica, la disponibilidad de múltiples bandas espectrales y su integración directa con herramientas de análisis ampliamente utilizadas.

No obstante, Sentinel presenta limitaciones relevantes para el contexto de este proyecto. Su resolución temporal, cercana a los cinco días para una misma región, sumada a la dependencia de condiciones atmosféricas favorables, puede generar períodos prolongados sin datos válidos. Asimismo, su resolución espacial, si bien adecuada para análisis a gran escala, puede resultar insuficiente en potreros con alta heterogeneidad.

7.3.2 Proveedores comerciales

Los proveedores comerciales de imágenes satelitales ofrecen soluciones que, en general, se caracterizan por una **mayor resolución espacial y temporal**. Estas plataformas cuentan con constelaciones de satélites que permiten obtener imágenes diarias, e incluso múltiples capturas por día, reduciendo significativamente el impacto de la nubosidad sobre la disponibilidad de datos.

Otra ventaja de estas soluciones es la posibilidad de acceder a imágenes con mayor nivel de detalle, lo que resulta especialmente beneficioso para el análisis de áreas productivas pequeñas o con variabilidad interna. Sin embargo, estas mejoras suelen estar asociadas a costos económicos y a restricciones de uso, lo que obliga a evaluar cuidadosamente su viabilidad dentro del proyecto.

7.3.3 Ventajas, Limitaciones y Selección del Proveedor

Tras evaluar múltiples alternativas disponibles en el mercado, la decisión fue seleccionar **Planet** como proveedor principal de imágenes satelitales complementarias a Sentinel. Esta elección respondió a un análisis integral que consideró **características técnicas, costos, soporte, facilidad de integración y potencial de crecimiento futuro** para asegurar la sostenibilidad del sistema desarrollado.

La Tabla 4 resume un análisis comparativo entre **Sentinel** y **Planet**, destacando los aspectos que resultaron relevantes para la toma de decisión:

Tabla 4*Comparación entre Sentinel y Planet*

Característica	Sentinel (ESA)	Planet
Resolución espacial	10–20 m	3–5 m
Resolución temporal	5 días aprox.	Diaria o superior.
Número de bandas útiles para NDVI	Bandas visible + infrarrojo cercano	Bandas visible + infrarrojo cercano
Acceso y costo	Gratuito	Pago
Integración API/Workflow	Bueno	Excelente (API robusta y documentada)
Soporte y Documentación	Comunitario / abierto	Dedicado

Justificación de la elección**1. Frecuencia de Captura**

Planet ofrece imágenes diarias o incluso múltiples veces al día, lo que reduce significativamente los vacíos temporales sin datos útiles, especialmente en condiciones de nubosidad, y mejora la continuidad del monitoreo a corto y mediano plazo. Esto representa una ventaja clave respecto a la ventana de aproximadamente 5 días promedio de Sentinel.

2. Resolución espacial mejorada

La capacidad de obtener imágenes con mayor nivel de detalle (3–5 metros) permite diferenciar características más finas del terreno. Esto se traduce en estimaciones de índices de vegetación más precisas para potreros heterogéneos, mejorando la representatividad de las mediciones en áreas productivas con variabilidad interna.

3. Escalabilidad y soporte técnico

Planet cuenta con soporte profesional y una API robusta, que facilitan la integración directa con la arquitectura de la API desarrollada en el proyecto. Tanto la documentación como el ecosistema de herramientas disponibles reducen la

complejidad de implementación y mantenimiento.

4. **Costos y alineación con el proyecto**

Si bien Planet es un servicio de pago, su modelo de precios ofrece planes adaptables al volumen de descargas e integración, lo cual permitió evaluar y seleccionar una opción económicamente viable para los requerimientos de Pastech. La relación entre costo, rendimiento y beneficios técnicos resultó favorable comparada con otras alternativas comerciales analizadas.

5. **Visión a futuro**

La estrategia de integración planteada considera la evolución de las necesidades de Pastech. Planet ofrece una plataforma flexible que puede escalar junto con el crecimiento del proyecto, sin requerir cambios estructurales importantes en los algoritmos de cálculo ya implementados.

7.4 **Índices de Vegetación Calculados**

El análisis de imágenes satelitales aplicado al monitoreo agropecuario se basa en el cálculo de **índices de vegetación**, los cuales permiten cuantificar el estado de la cobertura vegetal a partir de la información espectral captada por los sensores remotos. Estos índices constituyen la base para la estimación indirecta de variables productivas, como la biomasa disponible o el estado del pastizal.

En el presente proyecto, el cálculo de los índices de vegetación representa una etapa fundamental del proceso de análisis, ya que los valores obtenidos son utilizados posteriormente como insumo para el modelo de estimación del stock forrajero.

7.4.1 **NDVI (Normalized Difference Vegetation Index)**

El **NDVI** es uno de los índices de vegetación más utilizados a nivel mundial para evaluar el estado de la vegetación. Su popularidad se debe a su simplicidad de cálculo, robustez y fuerte correlación con la actividad fotosintética de las plantas.

Este índice se basa en la diferencia normalizada entre la reflectancia registrada en la banda del **infrarrojo cercano (NIR)** y la banda del **rojo (Red)**, de acuerdo con la siguiente expresión:

$$NDVI = \frac{NIR - Red}{NIR + Red}$$

Las plantas sanas absorben gran parte de la radiación en la banda roja para realizar la fotosíntesis y reflejan en mayor medida la radiación en el infrarrojo cercano. Como consecuencia, valores altos de NDVI suelen asociarse a vegetación densa y activa, mientras que valores bajos indican suelos desnudos, vegetación escasa o en mal estado.

Figura 13

Comparación de la reflectancia en las bandas azul, verde, roja y NIR para distintos estados de la vegetación



7.4.2 Interpretación de los valores de NDVI

El NDVI produce valores que oscilan generalmente entre **-1 y 1**, aunque en aplicaciones agropecuarias los rangos de interés suelen ubicarse entre **0 y 1**. A continuación se presenta una interpretación general de los valores obtenidos:

- **NDVI < 0:** Superficies no vegetadas (agua, nubes, nieve).
- **NDVI entre 0 y 0.2:** Suelo desnudo o vegetación muy escasa.
- **NDVI entre 0.2 y 0.5:** Vegetación de baja a media densidad.
- **NDVI > 0.5:** Vegetación densa y saludable.

Estos rangos permiten realizar un primer análisis cualitativo del estado del potrero y facilitan la detección de cambios significativos a lo largo del tiempo.

7.4.3 Aplicación del NDVI en el contexto del proyecto

En el marco de este proyecto, el NDVI se utiliza como **indicador principal del estado vegetativo del pastizal**. A partir de la delimitación de cada potrero mediante polígonos geográficos, se calcula un **valor promedio de NDVI** para cada unidad productiva y para cada fecha disponible.

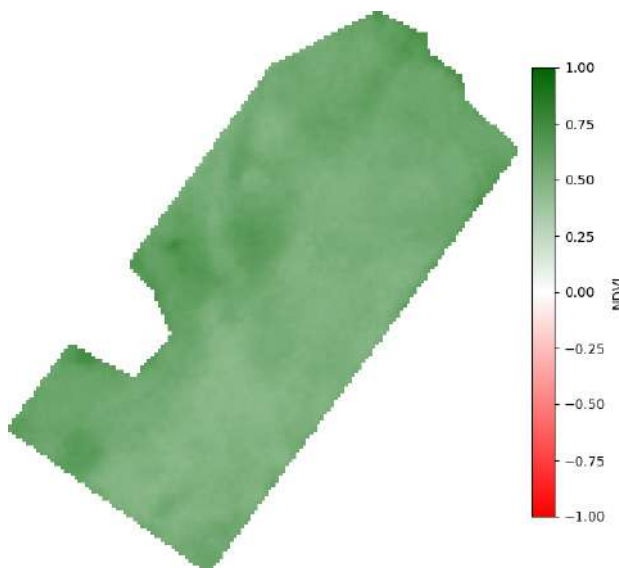
Este valor promedio permite:

- Monitorear la evolución temporal del pasto.
- Identificar zonas con menor vigor vegetativo.
- Servir como variable de entrada para el modelo de estimación de biomasa.

El cálculo se realiza de manera automatizada dentro del sistema, asegurando consistencia en el procesamiento y permitiendo comparaciones directas entre distintos proveedores de imágenes satelitales.

Figura 14

Ejemplo de imagen NDVI para un potrero registrado en Pastech.



Una vez definidos los índices de vegetación utilizados y su interpretación, el siguiente paso consiste en describir el **modelo de estimación de biomasa**, en el cual los valores de NDVI calculados son correlacionados con la cantidad de pasto disponible por potrero.

7.5 Modelo de estimación de biomasa

La estimación de biomasa se realiza a partir de información satelital, utilizando el índice NDVI como variable principal de entrada. El modelo de estimación utilizado en este trabajo fue provisto por la empresa Pastech, por lo que no forma parte del desarrollo propio del proyecto. Sin embargo, su incorporación implicó un proceso de integración, utilización y validación de resultados, lo que requirió un trabajo interdisciplinario con personal de la empresa.

El modelo no se limita únicamente al valor instantáneo de NDVI, sino que incorpora información contextual clave que permite ajustar la estimación a condiciones productivas reales. En particular, se consideran dos factores adicionales: **el tipo de pastura y el momento del año** en el que se realiza la medición. Esta combinación permite capturar las diferencias entre especies forrajeras y las variaciones estacionales en su crecimiento.

El tipo de pasto determina el comportamiento productivo general de la vegetación. Distintas especies presentan tasas de crecimiento diferentes y relaciones distintas entre el NDVI observado y la biomasa efectivamente disponible. Por este motivo, el modelo agrupa las pasturas en categorías y asigna a cada una un conjunto de curvas de respuesta específicas, diferenciadas por estación del año.

La estacionalidad es un componente central del modelo. La producción de biomasa no es constante a lo largo del año, sino que depende fuertemente de factores climáticos, los cuales varían de forma sistemática entre estaciones. Para reflejar este comportamiento, el modelo contempla las cuatro estaciones (otoño, invierno, primavera y verano) y ajusta la contribución de cada una según la fecha de la medición.

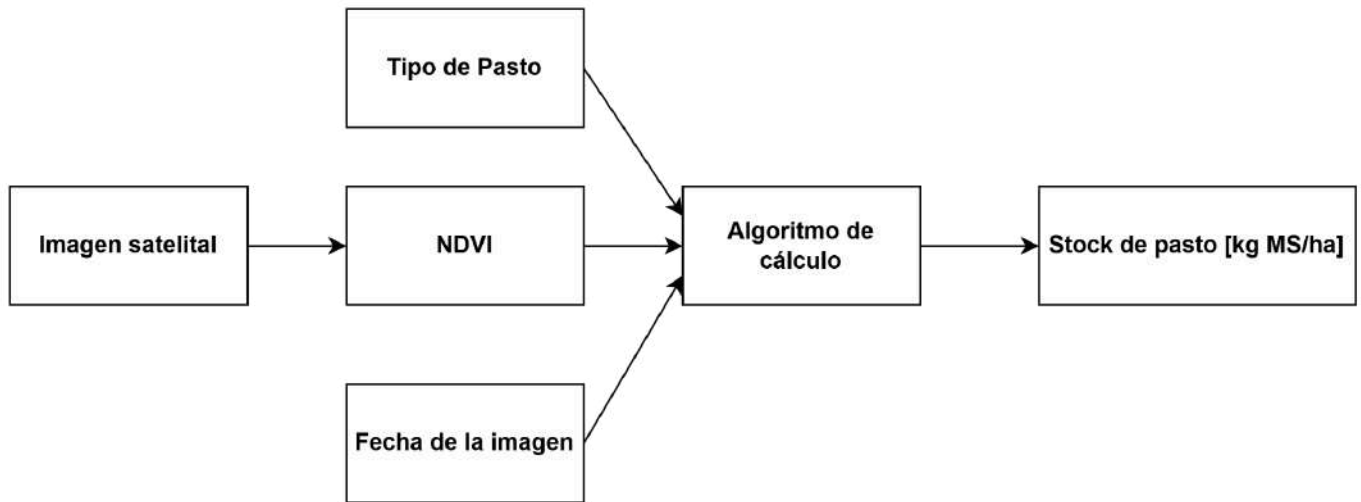
Cada estación aporta un valor estimado en función del NDVI y del tipo de pastura, y estas contribuciones se combinan de acuerdo con los pesos definidos por la estacionalidad. El resultado se expresa como cantidad de materia seca por unidad de superficie, permitiendo su uso directo en análisis productivos y de planificación ganadera.

Finalmente, una vez descritos los componentes principales del modelo y las variables que intervienen en la estimación, es posible sintetizar el proceso de cálculo como un flujo integrado de etapas. A partir de la obtención de la imagen satelital, se calcula el índice NDVI, el cual, junto con la información sobre el tipo de pastura y la fecha de la medición, constituye la entrada al modelo de estimación de biomasa. Estas variables permiten ajustar la estimación a las condiciones productivas y estacionales correspondientes,

dando como resultado el cálculo del stock de pasto expresado en materia seca por unidad de superficie. En la Figura 15 se presenta un esquema general del flujo de información y procesamiento involucrado en este procedimiento.

Figura 15

Flujo de cálculo del stock de pasto a partir de imágenes satelitales



Con el objetivo de brindar mayor contexto sobre el funcionamiento del modelo, se presentan a continuación las Figuras 16 y 17, en las cuales se detalla el flujo de cálculo del stock de pasto aplicado a un mismo potrero en dos momentos distintos del año. Estas figuras permiten contrastar las diferencias en las imágenes satelitales utilizadas, los valores de NDVI obtenidos y, finalmente, el stock de pasto estimado.

Figura 16

Flujo de cálculo del stock de pasto en el mes de octubre para un potrero registrado en Pastech.

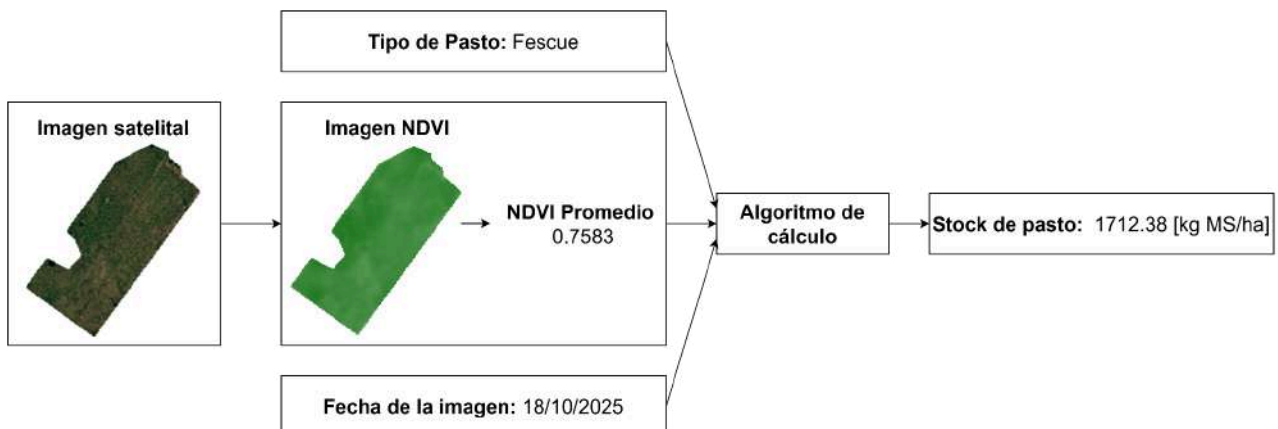
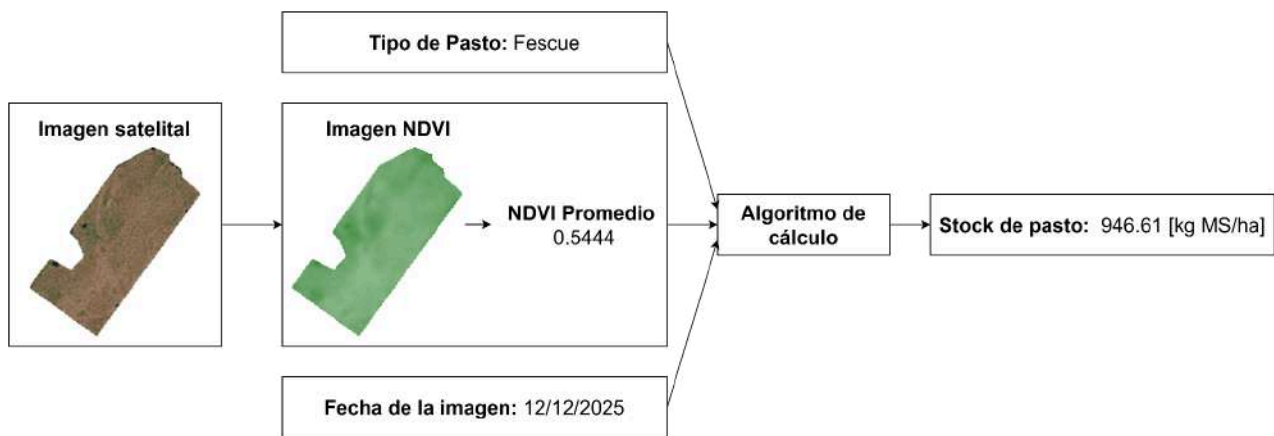


Figura 17

Flujo de cálculo del stock de pasto en el mes de diciembre para un potrero registrado en Pastech.



7.6 Validación de imágenes

Para garantizar la confiabilidad del cálculo del NDVI y la posterior estimación de biomasa, el sistema implementa un proceso de validación automatizado sobre cada imagen obtenida antes de proceder a su análisis. Este proceso consta de tres etapas de verificación:

- 1. Control de integridad:** Se comprueba que el paquete de datos descargado contenga la totalidad de los archivos requeridos y las bandas espectrales necesarias para la ejecución de los algoritmos.
- 2. Verificación de autenticidad:** Se valida que el archivo corresponda efectivamente a una escena de Planet, evitando el procesamiento de datos corruptos o de origen inválido.
- 3. Análisis de píxeles utilizables (Control de calidad):** Se evalúa la capa de clasificación adjunta a la imagen (Usable Data Mask), la cual provee información detallada píxel a píxel sobre la presencia de anomalías como nubosidad, sombras o saturación del sensor. El sistema calcula el porcentaje real de píxeles claros y lo compara contra un umbral de tolerancia mínimo. Este umbral es un parámetro dinámico configurado por el usuario y persistido en la tabla de configuración de la base de datos. Si la imagen no supera dicho porcentaje, es descartada del flujo de procesamiento para evitar que datos distorsionen la estimación del stock forrajero.

7.7 Validación de resultados

Una vez realizada la selección del proveedor de imágenes satelitales, fue necesario respaldar dicha decisión con resultados cuantitativos y evidencia empírica, que permitieran justificar su incorporación dentro del sistema de Pastech. Con este objetivo, se llevó a cabo un análisis comparativo entre imágenes provenientes de Sentinel y Planet, evaluando su impacto en los cálculos de índices de vegetación y en la estimación del stock forrajero.

El propósito de esta sección es presentar datos numéricos, gráficos y análisis comparativos que permitan visualizar las diferencias entre ambos proveedores y demostrar el valor agregado que implica la incorporación de Planet al flujo de trabajo existente. Los resultados obtenidos sirvieron como base para el intercambio con el referente funcional, permitiendo contrastar ambos proveedores con datos concretos y evaluar el aporte de Planet en aspectos clave como la frecuencia de medición, la resolución espacial y la continuidad de la información.

A continuación, se exponen los resultados del análisis realizado y las principales conclusiones derivadas de la comparación.

7.7.1 Comparación de cálculos

7.7.1.1 Objetivo

Realizar una comparativa entre los satélites Sentinel-2 y PlanetScope (SuperDove), evaluando su comportamiento en la estimación del NDVI y el stock forrajero (kilos de pasto por hectárea).

Para el análisis se utilizaron datos provistos por Pastech, correspondientes a un potrero ubicado en Uruguay.

7.7.1.2 Datos utilizados

Los datos de Sentinel incluyen mediciones de NDVI y kilos de pasto en distintas fechas durante el año 2025.

Posteriormente, se realizaron las mismas estimaciones utilizando imágenes de Planet, tanto en su versión original como en su versión armonizada con Sentinel (una opción que Planet ofrece para calibrar/normalizar sus datos con respecto a Sentinel).

Tabla 5*Valores de NDVI y stock forrajero obtenidos por Sentinel*

FECHA	NDVI	KILOS [kg/ha]
17/04/2025	0,780	2.240,56
2/05/2025	0,689	1.512,84
19/05/2025	0,810	1.864,97
29/05/2025	0,836	1.903,46
11/06/2025	0,754	1362,18
23/06/2025	0,760	1.341,08
6/07/2025	0,709	1.077,66
21/07/2025	0,743	1.244,52
9/08/2025	0,788	1.509,12
25/08/2025	0,790	1.519,85
6/09/2025	0,828	1.785,79

Tabla 6*Valores de NDVI y stock forrajero obtenidos por Planet.*

FECHA	NDVI	KILOS [kg/ha]
17/04/2025	0,772	2.175,34
2/05/2025	0,695	1.542,64
19/05/2025	0,768	1.581,41
29/05/2025	0,755	1.365,71
11/06/2025	0,743	1.297,91
23/06/2025	0,703	1.048,23
6/07/2025	0,692	999,89
21/07/2025	0,656	856,26
9/08/2025	0,777	1.432,84

25/08/2025	0,775	1.422,80
6/09/2025	0,798	1.570,79

Tabla 7

Valores de NDVI y stock forrajero obtenidos por Planet Armonizado.

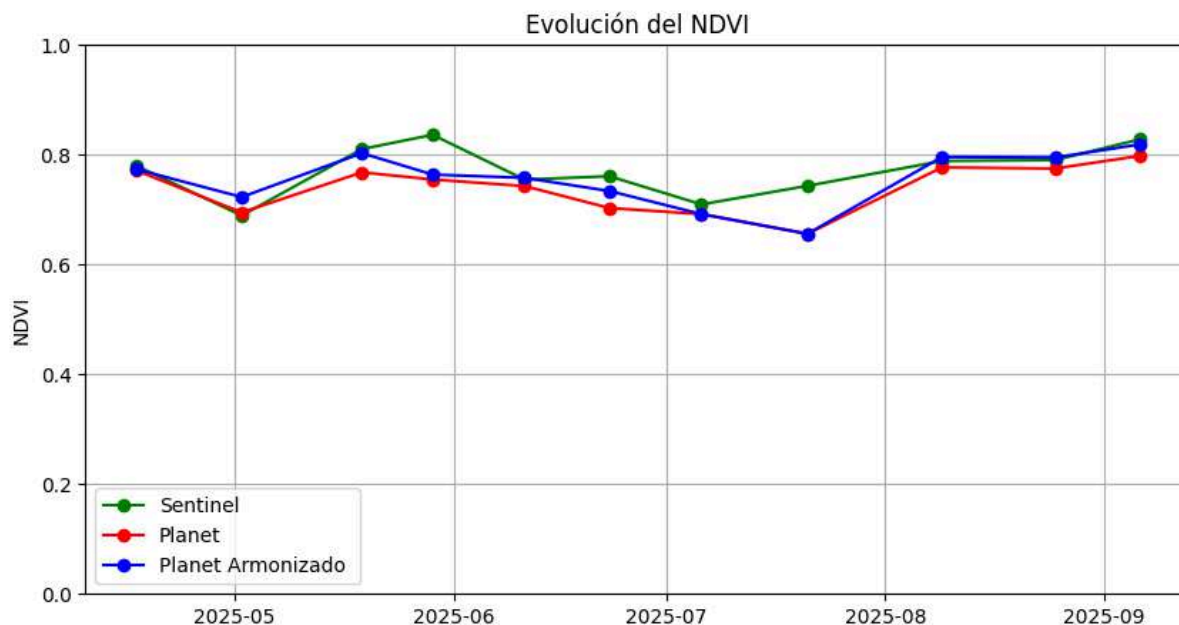
FECHA	NDVI	KILOS [kg/ha]
17/04/2025	0,774	2.191,37
2/05/2025	0,723	1.707,11
19/05/2025	0,803	1.808,20
29/05/2025	0,764	1.413,27
11/06/2025	0,758	1.378,66
23/06/2025	0,734	1.194,89
6/07/2025	0,692	999,89
21/07/2025	0,656	856,26
9/08/2025	0,796	1.559,68
25/08/2025	0,795	1.549,01
6/09/2025	0,819	1.714,63

7.7.1.3 Análisis de NDVI

Al comparar las series de NDVI obtenidas por ambos satélites, se observa que:

Figura 18

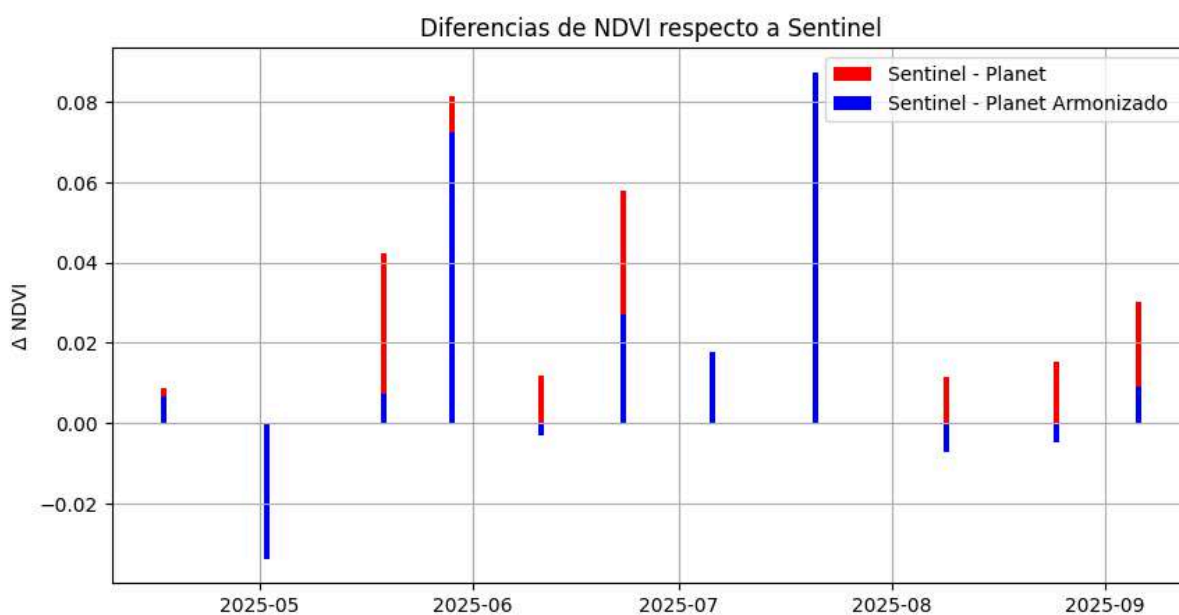
Comparación de NDVI entre Sentinel-2, Planet y Planet Armonizado.



- Las curvas presentan una tendencia general similar.
- Existen pequeñas diferencias puntuales, principalmente en dos fechas, aunque el comportamiento general se mantiene coherente.

Figura 19

Variación de NDVI respecto a Sentinel.



Se calculó el **sesgo** (bias) de cada serie con respecto a Sentinel:

- Sesgo Planet = -0.0326
- Sesgo Planet Armonizado = -0.0163

Interpretación

Un valor negativo en el sesgo indica que las mediciones de Planet reportan valores de NDVI numéricamente inferiores a los de Sentinel. Es fundamental destacar que esto no implica que el sensor de Planet subestime los valores reales del terreno, sino que corrige la sobreestimación introducida por Sentinel.

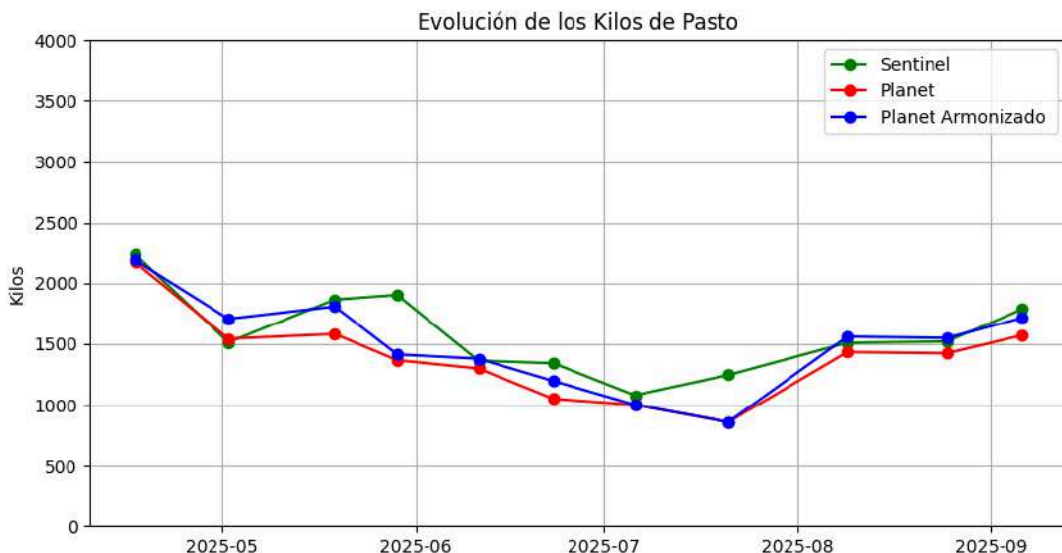
Esta diferencia radica en la resolución espacial: Sentinel (con píxeles de 10 a 20 metros) sufre del "efecto de píxel mixto", promediando áreas de vegetación vigorosa con suelo desnudo y sobreestimando el índice global del lote. Planet, al poseer una resolución de 3 metros, aísla con mayor precisión la vegetación real. Asimismo, se observa que la versión armonizada reduce la brecha numérica a la mitad, demostrando una excelente calibración que facilita la compatibilidad de las series temporales sin perder la alta fidelidad espacial. En síntesis, el proceso de armonización mejora significativamente la concordancia entre ambos satélites.

7.7.1.4 Análisis de Stock Forrajero

Se repitió el mismo procedimiento con el stock forrajero.

Figura 20

Comparación del stock forrajero estimado por Sentinel-2, Planet y Planet Armonizado.



Los sesgos resultaron:

- Sesgo Planet = -188.02
- Sesgo Planet Armonizado = -89.91

Interpretación

Al igual que con el NDVI, ambas versiones de Planet reportan un stock forrajero numéricamente menor al de Sentinel. Reiterando el principio espacial, esto no representa una subestimación del forraje, sino una medición mucho más ajustada a la realidad del potrero. Al contar con una mayor resolución espacial, Planet detecta con precisión las variaciones locales del terreno y los espacios de menor densidad que Sentinel promedia artificialmente hacia arriba.

Cabe destacar que el producto armonizado reduce la diferencia numérica a menos de la mitad (-89.91 kg/ha), lo que minimiza el impacto en el cambio de tecnología para el usuario final.

7.7.1.5 Conclusiones

Los resultados muestran una alta coherencia entre Sentinel y Planet, especialmente cuando se utilizan los productos armonizados. Las diferencias numéricas observadas, donde Planet reporta valores menores de NDVI y stock forrajero, no representan un problema ni una falencia de los sensores, sino una mejora directa en la precisión de la medición.

La adopción de las imágenes de Planet, al contar con mayor resolución espacial, permite corregir el sesgo de sobreestimación propio de Sentinel en zonas heterogéneas, ofreciendo estimaciones más detalladas y fieles a la realidad del lote. En términos prácticos, una desviación promedio de aproximadamente 90 kg/ha en el stock armonizado corrige la medición sin representar un impacto disruptivo en la escala de gestión ganadera de Pastech, validando así la superioridad técnica del nuevo proveedor.

7.7.2 Comparación de disponibilidad de imágenes

7.7.2.1 Objetivo

Analizar y comparar la frecuencia de adquisición de imágenes entre los satélites **Sentinel-2** y **PlanetScope**, considerando distintos niveles de tolerancia a la nubosidad:

1. = 0%
2. ≤ 15%
3. ≤ 30%

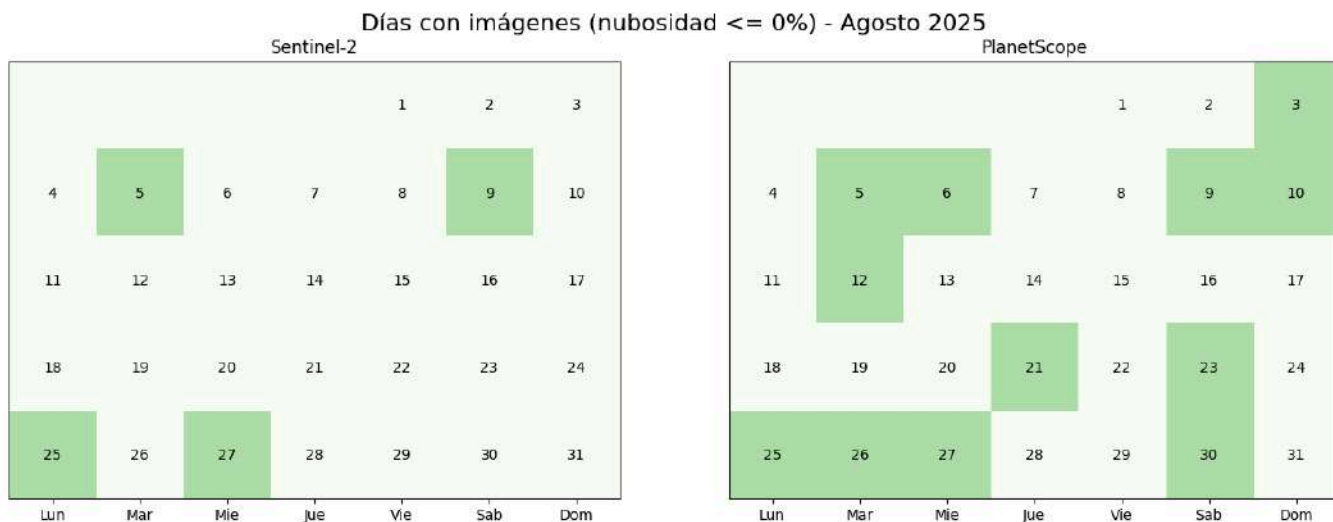
El análisis se realiza sobre el mes de agosto de 2025, para un área específica de interés (potrero en Uruguay).

7.7.2.2 Resultados

1. Nubosidad = 0%

Figura 21

Comparación de disponibilidad de imágenes entre Sentinel-2 y Planet con nubosidad = 0%



- **Sentinel-2** cuenta con imágenes sin nubosidad los días 5, 9, 25 y 27 de agosto, es decir, **4 días disponibles**.
- **PlanetScope**, en cambio, presenta imágenes en **12 días** del mes.

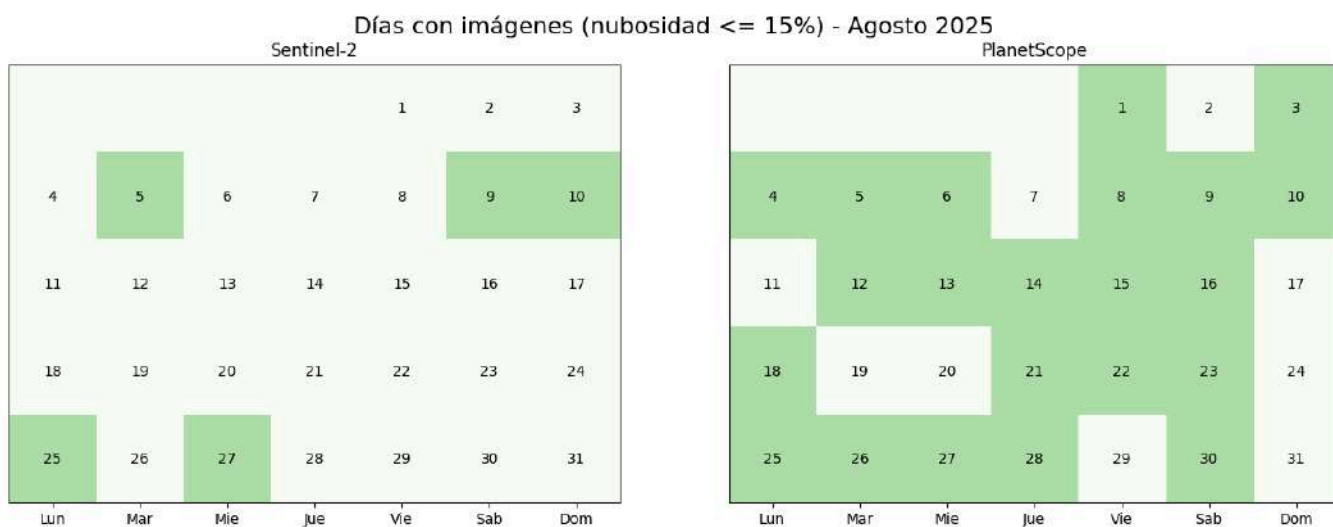
PlanetScope ofrece **más del doble de disponibilidad** de imágenes completamente despejadas respecto a Sentinel-2.

Esto se debe a la **mayor frecuencia de revisita** de la constelación Planet (diaria o casi diaria), frente al ciclo de Sentinel-2.

2. Nubosidad $\leq 15\%$

Figura 22

Comparación de disponibilidad de imágenes entre Sentinel-2 y Planet con nubosidad $\leq 15\%$



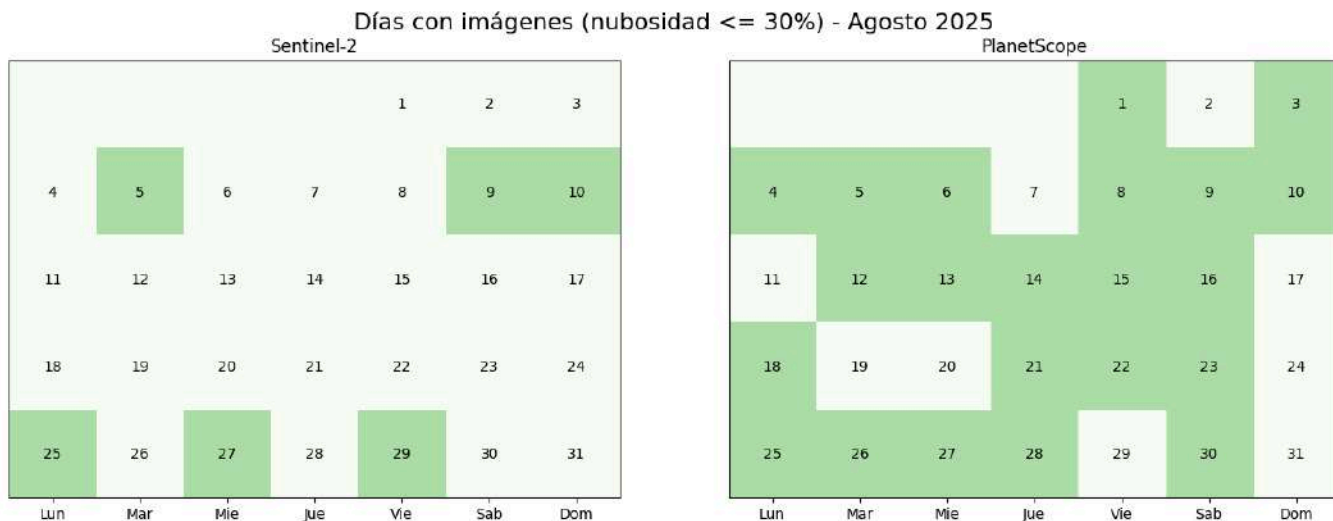
- Al permitir un margen de nubosidad del **15%**, Sentinel-2 amplía su disponibilidad a **5 fechas**, agregando una imagen adicional.
- PlanetScope muestra imágenes casi diarias, alcanzando **22 días** de observación utilizable.

La diferencia se amplía aún más a favor de PlanetScope.

3. Nubosidad \leq 30%

Figura 23

Comparación de disponibilidad de imágenes entre Sentinel-2 y Planet con nubosidad \leq 30%



- Con un umbral del 30% de nubosidad, Sentinel-2 logra 6 días de imágenes utilizables.
- PlanetScope alcanza 22 días con cobertura aceptable, lo que representa una disponibilidad superior al 70% del mes.

7.7.2.3 Conclusiones

- Sentinel-2 con su frecuencia limitada y la interferencia por nubosidad reducen significativamente la cantidad de observaciones útiles.
- PlanetScope, con su constelación de múltiples satélites, garantiza una mayor continuidad temporal, lo que permite realizar seguimientos casi diarios.
- Al aumentar el umbral de nubosidad, la diferencia de disponibilidad se amplía a favor de Planet.

8. Memoria del proyecto

8.1 Cumplimientos

Se puede afirmar que el sistema logró cumplir con los requerimientos funcionales definidos desde el inicio del proyecto. El principal caso de uso, y objetivo central del sistema, fue la carga y procesamiento de mediciones, el cual fue implementado de manera correcta y conforme a lo esperado.

Más allá de la correcta implementación técnica, se llevó a cabo una etapa de validación de resultados en conjunto con ingenieros agrónomos. Esta instancia fue fundamental, ya que el correcto funcionamiento del sistema no resultaba suficiente si las mediciones obtenidas no eran consistentes desde el punto de vista agronómico. En caso de que los resultados no hubieran aportado valor o las imágenes no resultaran útiles para el ecosistema de Pastech, se habría considerado un incumplimiento significativo del objetivo del proyecto.

Otro aspecto que se cumplió de manera satisfactoria fue el plazo de finalización. Si bien existieron desvíos en algunas etapas y fue necesaria una redistribución del esfuerzo a lo largo del desarrollo, el proyecto se completó dentro de los tiempos previstos. Un factor clave para lograr este cumplimiento fue la planificación realista de la dedicación horaria semanal, así como la consideración de un margen adicional para absorber posibles imprevistos.

Por último, el trabajo interdisciplinario se desarrolló de forma exitosa. La colaboración entre los desarrolladores de la empresa, referente funcional y los ingenieros agrónomos permitió una comunicación fluida, la construcción de consensos y la toma de decisiones conjuntas, lo cual se vio reflejado positivamente en el producto final.

8.2 Obstáculos

Uno de los principales obstáculos encontrados en las etapas iniciales del proyecto, particularmente durante la fase de investigación, fue la falta de transparencia por parte de varios proveedores de imágenes satelitales. En muchos casos, la información disponible de manera pública sobre planes, tipos de imágenes, frecuencia de captura y resolución

era escasa o poco clara. Esta situación dificultó el relevamiento y la comparación de proveedores para seleccionar la opción más adecuada.

Si bien se intentó establecer contacto a través de correo electrónico, en varios casos se obtuvo poca o ninguna respuesta. Esto llevó a priorizar aquellos proveedores que contaban con información más accesible y que, además, ofrecían beneficios o planes orientados a estudiantes.

Otro inconveniente relevante surgió luego de la elección del proveedor, ya que la contratación de un plan pago se realizaría en etapas avanzadas del proyecto. Esta situación limitó el desarrollo, al impedir la realización de pruebas en un entorno realista. No obstante, también tuvo un impacto positivo, ya que fue necesario profundizar en el estudio de la documentación técnica y diseñar la implementación basándose principalmente en ella.

Una dificultad adicional estuvo relacionada con la curva de aprendizaje una vez iniciado el desarrollo. La experiencia previa con Python y el framework FastAPI era prácticamente nula, lo que implicó un esfuerzo adicional para adquirir los conocimientos necesarios sobre el lenguaje, el modelo asincrónico y las buenas prácticas asociadas al desarrollo de APIs modernas. Asimismo, si bien la etapa de investigación teórica sobre imágenes satelitales resultó de gran utilidad, el procesamiento de este tipo de datos presentó una complejidad mayor a la esperada, requiriendo múltiples iteraciones hasta lograr resultados consistentes y confiables.

Finalmente, se presentaron desafíos vinculados al manejo del asincronismo y la idempotencia en el sistema. Al solicitar la carga de mediciones para una gran cantidad de potreros en una única operación, los tiempos de procesamiento excedían los límites razonables de respuesta, provocando *timeouts*. Para resolver este problema, se implementó una solución basada en *background tasks*, que permiten ejecutar procesos de larga duración de manera asincrónica sin bloquear la respuesta inmediata de la API. Complementariamente, se incorporó el uso de webhooks, mediante los cuales el sistema notifica a Pastech cuando finaliza el procesamiento de cada potrero, garantizando así una comunicación eficiente y desacoplada entre los sistemas.

Ante este escenario, resulta pertinente realizar un análisis crítico sobre la estrategia de escalabilidad y la elección del estilo arquitectónico. Esta detección de *timeouts* hacia las etapas finales del proyecto no representa un fallo conceptual en la selección del estilo

arquitectónico, sino una consecuencia esperable al procesar archivos geoespaciales y matrices de datos numéricos.

El objetivo central de este trabajo fue el diseño y desarrollo de una API preparada para operar en un entorno productivo real, orientada específicamente a validar la viabilidad técnica, la compatibilidad operativa y el valor agronómico de incorporar un nuevo proveedor de imágenes satelitales en el ecosistema de Pastech. Definir una arquitectura distribuida basada en microservicios desde el inicio, sin haber validado previamente la consistencia de los cálculos con el nuevo proveedor, habría supuesto una complejidad arquitectónica temprana que habría incrementado la dificultad del despliegue y los costos de infraestructura sin una justificación real que la respaldara.

Al haber diseñado e implementado la API para ejecutarse de forma aislada dentro de contenedores, el sistema ya cuenta con la estructura óptima para responder ante escenarios de alta demanda o un incremento masivo en el volumen de potreros a procesar. El uso de contenedores garantiza la portabilidad, el aislamiento del entorno y la consistencia entre ambientes, lo que permite que este sistema funcional pueda escalar frente a una alta carga mediante dos estrategias claras:

- **Escalabilidad Horizontal del Monolito:** Dado que el procesamiento de imágenes y el cálculo de índices se desacopló del ciclo de vida de la petición HTTP mediante la implementación de tareas en segundo plano (*background tasks*) y *webhooks*, la aplicación puede replicarse horizontalmente en múltiples contenedores idénticos de manera transparente. Esto permite que un balanceador de carga distribuya las solicitudes entrantes eficientemente entre las distintas instancias activas.
- **Descomposición en microservicios:** Si bien el sistema no califica estrictamente como un monolito modular en su concepción teórica más pura, el dominio central de la aplicación se encuentra profundamente modularizado. Gracias al uso de Clean Architecture, el módulo de procesamiento de imágenes y cálculo de índices (el componente más crítico y demandante en términos de CPU y memoria) mantiene un bajo nivel de acoplamiento respecto al resto del sistema. Si la demanda futura de la empresa lo exigiera, esta separación lógica permite extraer de forma limpia este submódulo y migrarlo hacia un microservicio dedicado o una arquitectura Serverless, sin necesidad de reestructurar la totalidad del código base.

Por último, otro obstáculo estuvo relacionado con la variación de los requerimientos a lo largo de las distintas reuniones. Si bien el referente funcional estuvo presente para supervisar las decisiones, los aportes de los diferentes actores involucrados generaron cambios frecuentes en los requerimientos, lo que implicó retrabajo. Esta situación había sido contemplada en la planificación inicial, dado que se trataba de pocos casos de uso pero con un alto nivel de complejidad. Aun así, el proyecto podría haberse beneficiado destinando mayor tiempo a la etapa de análisis y menos a la de desarrollo.

8.3 Futuras mejoras

Si bien la arquitectura del sistema permite la incorporación de nuevos proveedores satelitales, actualmente no contempla la posibilidad de operar con múltiples proveedores de manera simultánea. En el estado actual, Pastech utiliza su propio sistema para el procesamiento de imágenes de Sentinel y recurre al sistema desarrollado en este proyecto únicamente para imágenes de Planet. Una posible mejora futura consiste en unificar todos los servicios de imágenes satelitales en una única solución centralizada.

Otra mejora identificada, surgida durante reuniones con ingenieros agrónomos, es la aplicación de algún método de calibración sobre el algoritmo de cálculo, por ejemplo mediante regresión lineal, con el objetivo de reducir aún más las diferencias entre los valores obtenidos a partir de imágenes de Sentinel y Planet. En una etapa inicial se evaluó incorporar esta calibración directamente en el flujo de cálculo, e incluso se desarrolló un script para tal fin. Sin embargo, se decidió excluir esta funcionalidad del alcance del proyecto debido al desvío que implicaba, ya que su automatización requería una recolección masiva de imágenes de ambos proveedores, análisis comparativos y un tiempo de dedicación considerable.

Al momento de seleccionar el proveedor comercial de imágenes satelitales, si bien el requerimiento funcional para la estimación de biomasa era el cálculo del NDVI (el cual utiliza únicamente las bandas del Rojo y del Infrarrojo Cercano), el comitente estableció como un requerimiento estratégico deseable contar con una mayor resolución espectral. Específicamente, se buscaba disponer de una cantidad de bandas similar o superior a las que provee Sentinel, previendo la necesidad futura de implementar nuevos algoritmos e índices de productividad. Esta previsión técnica fue un factor clave en la selección de PlanetScope. Es importante aclarar que satisfacer este requerimiento del comitente no implicó un costo económico añadido para el proyecto. En los modelos de comercialización

actuales de estos proveedores satelitales, el paquete de imágenes multiespectrales (habitualmente de 4 u 8 bandas) se entrega como una característica nativa y estándar del producto, sin que el espectro ampliado represente un recargo sobre el precio base enfocado en la resolución espacial y temporal. En consecuencia, una mejora futura relevante consistirá en el desarrollo de módulos para calcular nuevos índices de vegetación. Gracias a la tecnología seleccionada, la concreción de estas mejoras no exigirá ampliar el espectro contratado ni representará un incremento en los costos operativos de adquisición de datos para Pastech.

En esta misma línea, una posible evolución del sistema consiste en la incorporación de algoritmos basados en técnicas de inteligencia artificial o aprendizaje automático. El uso de modelos entrenados sobre imágenes multiespectrales podría permitir estimaciones más precisas de biomasa, detección temprana de anomalías o clasificación de estados productivos, ampliando significativamente el alcance funcional del sistema.

La estrategia de testing automatizado implementada en este proyecto cumplió con el objetivo de validar de forma aislada y determinista la robustez de la API. El uso intensivo de componentes de simulación (mocks) para los proveedores de almacenamiento y de imágenes satelitales constituye un estándar para las fases de pruebas unitarias y de integración. Esto permitió verificar el algoritmo de cálculo del NDVI, los mecanismos de persistencia en la base de datos y el cumplimiento estricto de los contratos de la interfaz de programación. No obstante, desde una perspectiva estrictamente metodológica, toda estrategia basada en el aislamiento mediante mocks posee un límite de alcance natural: no replica las latencias reales de la red, los tiempos de procesamiento variables de las infraestructuras de terceros ni los casos de borde imprevistos que puedan surgir en las APIs externas en producción. La realización de pruebas de sistema End-to-End (E2E) con datos e imágenes reales de Planet quedó fuera del alcance de esta etapa debido a la postergación en la contratación del plan comercial por parte de la empresa comitente, lo cual impidió disponer de un entorno de staging productivo con cuotas operativas reales. Por lo tanto, se define como una mejora futura crítica la expansión de las pruebas mediante la incorporación de tests End-to-End en un entorno sandbox o de preproducción provisto por Planet. Esto permitirá controlar el comportamiento del sistema bajo condiciones de carga real, evaluar el manejo de errores ante caídas de los servicios externos y validar el sistema antes del despliegue en producción.

Finalmente, en la actual versión del sistema, la gestión de la *API Key* para la autenticación Server-to-Server se diseñó delegando la creación e inyección del token a una inserción manual en la base de datos relacional (como se especifica en el RF1.2). Tras una revisión arquitectónica y en consonancia con las prácticas modernas de DevOps, se identificó que esta estrategia representa una deuda técnica y operativa. Delegar la seguridad a un mantenimiento manual sobre la base de datos en caliente es una práctica propensa a errores humanos. Ante un escenario crítico que exija la revocación de la clave en producción, el sistema actual carece de un flujo automatizado de rotación de credenciales, lo que podría dejar la API expuesta durante el tiempo de intervención manual. Por lo tanto, se establece como una mejora la refactorización del mecanismo de autenticación que contemple la posibilidad de rotar credenciales.

8.4 Estado actual del sistema

El sistema se encuentra actualmente desarrollado, validado y correctamente desplegado en Railway, en condiciones técnicas de operar en un entorno productivo real. Sin embargo, al momento de cierre del presente documento, la solución no está siendo utilizada activamente por Pastech. Es de suma importancia clarificar que esta situación responde a una postergación estratégica y financiera por parte del comitente, y no a un fracaso o impedimento técnico en la arquitectura del software.

Durante la etapa de diseño y desarrollo, se trabajó utilizando un plan de acceso para estudiantes provisto por el proveedor satelital. Este plan establece restricciones contractuales estrictas de uso exclusivamente académico, lo que imposibilitaba legalmente su utilización en entornos comerciales. Para avanzar hacia la puesta en marcha efectiva, tal como se desglosa en el análisis de la Sección 6.3 (Estructura de costos), se constató que la contratación de los planes comerciales necesarios para cubrir el volumen operativo del mercado objetivo en Argentina requería una inversión inicial superior a los 15.000 USD.

Ante este requerimiento de capital, y en consonancia con la evaluación del riesgo R1, se analizó originalmente la alternativa de contratar un plan de menor escala para una primera etapa de validación. No obstante, la dirección de Pastech tomó la decisión de pausar temporalmente cualquier tipo de contratación satelital comercial, con el objetivo de evitar una inversión inicial desproporcionada.

Por lo tanto, el proyecto finaliza con una solución de software robusta y preparada ("lista para operar"), que queda a la entera disposición de la empresa para ser activada en el momento en que las condiciones de negocio justifiquen el inicio de las operaciones productivas.

Anexos

- **Documento de Especificación de Requerimientos de Software.**

Disponible en: [Documento de Especificación de Requerimientos de Software](#)

- **Documento de Decisiones de Diseño.**

Disponible en: [Documento de Decisiones de Diseño](#)

Nota: Los mencionados se entregan como material complementario en formato digital, formando parte de la documentación final del proyecto.

- **Especificación de la API**

```
{
  "openapi": "3.1.0",
  "info": {
    "title": "Pastech API.",
    "version": "1.0.0",
    "description": "API para el sistema de procesamiento de imágenes satelitales de Pastech. Permite gestionar la suscripción de parcelas (potreros) al proveedor de imágenes, recibir eventos asíncronos mediante webhooks y orquestar el cálculo automatizado de índices de vegetación (NDVI) y estimación de biomasa forrajera."
  },
  "tags": [
    {
      "name": "Cálculos",
      "description": "Orquestación y procesamiento de imágenes y cálculo de índices."
    },
    {
      "name": "Potreros",
      "description": "Gestión de parcelas y su sincronización/suscripción con la API de satélites."
    },
    {
      "name": "Webhooks",
      "description": "Recepción de eventos asíncronos de servicios externos."
    }
  ],
  "paths": {
    "/calculos/cargar-mediciones": {
      "post": {
        "tags": ["Cálculos"],
        "summary": "Orquestar cálculo de mediciones forrajeras",
        "description": "Dispara un proceso asíncrono en segundo plano (background task) para calcular las estimaciones forrajeras en base a imágenes satelitales.\n\n-
```

Potreros: Si se envía un listado, se procesan únicamente esos IDs.
Usuario: Permite filtrar los potreros pertenecientes a un ID de usuario específico.

Fecha: Si se especifica un rango, el sistema recalcula o fuerza la obtención de imágenes en dicho período. Caso contrario, se procesan únicamente las imágenes nuevas desde la última medición registrada.

La respuesta es inmediata (200 OK) indicando que el procesamiento ha iniciado.

```

"operationId": "cargar_mediciones_calculos_cargar_mediciones_post",
"requestBody": {
  "description": "Criterios de filtro para determinar qué parcelas e imágenes se procesarán.",
  "content": {
    "application/json": {
      "schema": {
        "$ref": "#/components/schemas/CargarMedicionesRequest"
      }
    }
  },
  "required": true
},
"responses": {
  "200": {
    "description": "Solicitud aceptada. El cálculo se realizará en segundo plano.",
    "content": {
      "application/json": {
        "schema": {
          "type": "string",
          "example": "El procesamiento ha comenzado en segundo plano"
        }
      }
    }
  },
  "422": {
    "description": "Error de validación en los parámetros enviados.",
    "content": {
      "application/json": {
        "schema": {
          "$ref": "#/components/schemas/HTTPValidationError"
        }
      }
    }
  }
},
"security": [
  {
    "APIKeyHeader": []
  }
]
}

```

```

},
"/potreros/alta": {
  "post": {
    "tags": ["Potreros"],
    "summary": "Alta de suscripción de potrero",
    "description": "Registra un potrero en el sistema e inicializa una suscripción automatizada en Planet Labs. A partir de este momento, Planet alertará al webhook cuando haya sobrepasos de satélite y nuevas imágenes disponibles sobre el polígono.",
    "operationId": "alta_potrero_potreros_alta_post",
    "requestBody": {
      "description": "Identificador del potrero que desea suscribirse.",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/PotreroAltaRequest"
          }
        }
      },
      "required": true
    },
    "responses": {
      "200": {
        "description": "Suscripción iniciada correctamente.",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/PotreroResponse"
            }
          }
        }
      },
      "422": {
        "description": "Error de validación.",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/HTTPValidationError"
            }
          }
        }
      }
    },
    "security": [
      {
        "APIKeyHeader": []
      }
    ]
  }
},

```

```

"/potreros/baja": {
  "delete": {
    "tags": ["Potreros"],
    "summary": "Baja de suscripción de potrero",
    "description": "Finaliza y da de baja la suscripción satelital activa de un
potrero en Planet. Detiene la descarga de nuevas imágenes y cancela las notificaciones
al webhook, evitando así el consumo innecesario de cuota y presupuesto.",
    "operationId": "baja_potrero_potreros_baja_delete",
    "requestBody": {
      "description": "Identificador del potrero que se desea dar de baja.",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/PotreroBajaRequest"
          }
        }
      },
      "required": true
    },
    "responses": {
      "200": {
        "description": "Potrero dado de baja con éxito de las suscripciones.",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/PotreroResponse"
            }
          }
        }
      },
      "422": {
        "description": "Error de validación.",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/HTTPValidationError"
            }
          }
        }
      }
    },
    "security": [
      {
        "APIKeyHeader": []
      }
    ]
  }
},
"/potreros/": {

```

```

"get": {
  "tags": ["Potreros"],
  "summary": "Consultar potreros suscritos e información de suscripciones",
  "description": "Retorna un registro con todos los potreros actualmente
suscritos. Permite el monitoreo del estado de las suscripciones. Se puede acotar la
búsqueda utilizando los filtros opcionales de ID de potrero y/o ID de suscripción.",
  "operationId": "obtener_potreros_potreros__get",
  "security": [
    {
      "APIKeyHeader": []
    }
  ],
  "parameters": [
    {
      "name": "idpotrero",
      "in": "query",
      "required": false,
      "schema": {
        "type": "integer",
        "nullable": true,
        "description": "ID del potrero a buscar."
      }
    },
    {
      "name": "idsuscripcion",
      "in": "query",
      "required": false,
      "schema": {
        "type": "integer",
        "nullable": true,
        "description": "ID de la suscripción de la API satelital a filtrar."
      }
    }
  ],
  "responses": {
    "200": {
      "description": "Lista de suscripciones obtenida correctamente.",
      "content": {
        "application/json": {
          "schema": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "id_potrero": { "type": "integer" },
                "id_suscripcion": { "type": "string" },
                "estado": { "type": "string" }
              }
            }
          }
        }
      }
    }
  }
}

```

```

    }
  }
},
"422": {
  "description": "Error de parámetros de búsqueda.",
  "content": {
    "application/json": {
      "schema": {
        "$ref": "#/components/schemas/HTTPValidationError"
      }
    }
  }
},
"/webhook/planet": {
  "post": {
    "tags": ["Webhooks"],
    "summary": "Webhook receptor de notificaciones (Planet)",
    "description": "Punto de entrada asíncrono diseñado para ser invocado por los servicios de Planet Labs tras generar o actualizar eventos de suscripción o de recortes de imagen. El sistema procesa los metadatos y encola procesos secundarios en caso de que existan nuevas capturas para descargar y evaluar.",
    "operationId": "webhook_planet_webhook_planet_post",
    "requestBody": {
      "description": "Carga útil genérica enviada por el Webhook de Planet.",
      "content": {
        "application/json": {
          "schema": {
            "type": "object",
            "additionalProperties": true
          }
        }
      }
    }
  },
  "responses": {
    "200": {
      "description": "Notificación recibida y procesada correctamente.",
      "content": {
        "application/json": {
          "schema": {
            "type": "object",
            "properties": {
              "status": {
                "type": "string",
                "example": "ok"
              }
            }
          }
        }
      }
    }
  }
}

```

```

    }
    }
  }
}
},
"components": {
  "schemas": {
    "CargarMedicionesRequest": {
      "type": "object",
      "title": "Configuración de Cálculo de Mediciones (Request)",
      "description": "Entidad que define las restricciones a aplicar durante la recolección satelital y el cálculo de estimaciones. Todos los campos son opcionales y actúan como filtros inclusivos.",
      "properties": {
        "potreros": {
          "type": "array",
          "items": {
            "type": "integer"
          },
          "nullable": true,
          "title": "IDs de potreros",
          "description": "Lista explícita de identificadores de los potreros a calcular.",
          "example": [101, 102, 105]
        },
        "usuario": {
          "type": "integer",
          "nullable": true,
          "title": "ID de Usuario",
          "description": "Filtro que acota el procesamiento únicamente a los potreros pertenecientes a este usuario.",
          "example": 42
        },
        "fecha": {
          "nullable": true,
          "allOf": [
            { "$ref": "#/components/schemas/FechaRango" }
          ],
          "description": "Período temporal sobre el cual se deben reprocesar u obtener las estimaciones. Si se omite, se asume desde el último procesamiento registrado."
        }
      },
      "FechaRango": {

```

```

    "type": "object",
    "required": ["inicio", "fin"],
    "title": "Rango de Fechas",
    "description": "Definición del bloque temporal para buscar capturas
satelitales.",
    "properties": {
      "inicio": {
        "type": "string",
        "format": "date-time",
        "title": "Fecha de Inicio",
        "description": "Límite inferior del rango de análisis.",
        "example": "2025-01-01T00:00:00Z"
      },
      "fin": {
        "type": "string",
        "format": "date-time",
        "title": "Fecha de Fin",
        "description": "Límite superior del rango de análisis.",
        "example": "2025-01-31T23:59:59Z"
      }
    }
  },
  "PotreroAltaRequest": {
    "type": "object",
    "title": "Solicitud de Alta de Potrero",
    "description": "Esquema necesario para solicitar un alta de monitorización en
un potrero. Se debe enviar el ID del potrero o, de manera opcional, el GeoJSON que
describe su geometría.",
    "properties": {
      "id_potrero": {
        "type": "integer",
        "nullable": true,
        "title": "ID Potrero",
        "description": "Identificador único asignado al potrero en el sistema
central. Opcional si se envía un geojson.",
        "example": 145
      },
      "geojson": {
        "type": "object",
        "nullable": true,
        "title": "Polígono (GeoJSON)",
        "description": "Objeto GeoJSON que representa los límites geográficos del
potrero. Requerido si no se envía el id_potrero.",
        "example": {
          "type": "Polygon",
          "coordinates": [
            [
              [-60.5, -34.5],
              [-60.5, -34.6],

```

```

        [-60.4, -34.6],
        [-60.4, -34.5],
        [-60.5, -34.5]
    ]
  ]
}
}
},
"PotreroBajaRequest": {
  "type": "object",
  "required": ["id_potrero"],
  "title": "Solicitud de Baja de Potrero",
  "description": "Esquema necesario para revocar las suscripciones del potrero.",
  "properties": {
    "id_potrero": {
      "type": "integer",
      "title": "ID Potrero",
      "description": "Identificador único asignado al potrero a desactivar.",
      "example": 145
    }
  }
},
"PotreroResponse": {
  "type": "object",
  "required": ["mensaje"],
  "title": "Respuesta Operación de Potrero",
  "description": "Respuesta estándar frente a una acción sobre una suscripción de potrero.",
  "properties": {
    "mensaje": {
      "type": "string",
      "title": "Mensaje de Estado",
      "description": "Resultado del proceso.",
      "example": "Suscripción del potrero '145' creada exitosamente"
    }
  }
},
"HTTPValidationError": {
  "type": "object",
  "title": "Error de Validación HTTP",
  "properties": {
    "detail": {
      "type": "array",
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      }
    },
    "title": "Detalle del error"
  }
}

```

```

    }
  },
  "ValidationError": {
    "type": "object",
    "required": ["loc", "msg", "type"],
    "title": "Error de Validación (Detalle)",
    "properties": {
      "loc": {
        "type": "array",
        "items": {
          "type": "string"
        },
        "title": "Ubicación",
        "description": "Ruta dentro del cuerpo de la petición que presenta fallas."
      },
      "msg": {
        "type": "string",
        "title": "Mensaje",
        "description": "Explicación del fallo de validación detectado."
      },
      "type": {
        "type": "string",
        "title": "Tipo de Error",
        "description": "Clasificador interno del error en Pydantic/FastAPI."
      }
    }
  }
},
"securitySchemes": {
  "APIKeyHeader": {
    "type": "apiKey",
    "in": "header",
    "name": "x-api-key",
    "description": "Autenticación mediada por Token enviada en el Header
'x-api-key'."
  }
}
}
}
}
}

```

Glosario

- **Biomasa Forrajera:** Cantidad de materia vegetal disponible en una superficie determinada (generalmente expresada en kg MS/ha), utilizada como recurso alimenticio para el ganado.
- **Infrarrojo Cercano (NIR - Near Infrared):** Región del espectro electromagnético (aprox. 700-1400 nm) fundamental para el monitoreo de vegetación, ya que las plantas saludables reflejan gran cantidad de energía en esta banda.
- **NDVI (Normalized Difference Vegetation Index):** Índice de vegetación normalizado que cuantifica la salud y densidad de la vegetación calculando la diferencia entre la reflectancia del infrarrojo cercano (NIR) y la luz roja visible.
- **PlanetScope:** Constelación de satélites comerciales de la empresa Planet Labs, caracterizada por ofrecer imágenes de alta resolución espacial (3-5 metros) y una frecuencia de revisita diaria.
- **Potrero:** Unidad de división de tierra dentro de un establecimiento agropecuario destinada al pastoreo rotativo del ganado.
- **Resolución Espectral:** Capacidad de un sensor satelital para distinguir diferentes bandas de longitud de onda (ej. Azul, Verde, Rojo, NIR).
- **Resolución Espacial:** Medida que determina el tamaño del píxel proyectado en el terreno (ej. 10m para Sentinel-2, 3m para Planet), definiendo el nivel de detalle visible.
- **Resolución Temporal:** Frecuencia con la que un satélite vuelve a capturar imágenes sobre la misma ubicación geográfica (tiempo de revisita).
- **Sentinel-2:** Misión de observación terrestre de la Agencia Espacial Europea (ESA) que provee imágenes ópticas gratuitas con una resolución espacial de 10 a 60 metros y una revisita aproximada de 5 días.
- **API REST (Representational State Transfer):** Estilo de arquitectura de software para sistemas distribuidos, utilizado en este proyecto para la comunicación entre el backend y el sistema Pastech mediante protocolo HTTP.

- **Clean Architecture:** Patrón de diseño de software propuesto por Robert C. Martin que promueve la separación de responsabilidades en capas concéntricas, aislando la lógica de negocio de frameworks, bases de datos e interfaces externas.
- **GeoJSON:** Formato estándar basado en JSON diseñado para representar estructuras de datos geográficos simples, utilizado aquí para definir los polígonos de los potreros.
- **Idempotencia:** Propiedad de las operaciones en una API donde realizar la misma solicitud múltiples veces produce el mismo resultado que realizarla una sola vez, garantizando consistencia ante reintentos.
- **ORM (Object-Relational Mapping):** Técnica de programación utilizada para convertir datos entre el sistema de tipos de un lenguaje orientado a objetos (Python) y una base de datos relacional (MySQL). En este proyecto se utilizó SQLAlchemy.
- **Patrón Repository:** Patrón de diseño que abstrae el acceso a los datos, permitiendo que la lógica de negocio acceda a la base de datos sin conocer los detalles de la implementación del almacenamiento.
- **Webhook:** Mecanismo de comunicación mediante el cual una aplicación envía datos en tiempo real a otra aplicación tan pronto como ocurre un evento específico, evitando la necesidad de sondeo constante (polling).
- **CI/CD (Continuous Integration / Continuous Deployment):** Práctica de ingeniería de software que permite integrar cambios en el código y desplegarlos automáticamente en producción.
- **Docker:** Plataforma de código abierto que utiliza virtualización a nivel de sistema operativo para entregar software en paquetes llamados contenedores, asegurando consistencia entre entornos de desarrollo y producción.
- **Blob Storage:** Servicio de almacenamiento en la nube optimizado para guardar grandes cantidades de datos no estructurados, como imágenes satelitales y archivos binarios.

Bibliografía

1. **Ávila Briceño, D.** (2024). *¿Qué es resolución espacial, temporal y espectral?* LinkedIn. <https://www.linkedin.com/pulse/qu%C3%A9-es-resoluci%C3%B3n-espacial-temporal-y-espectral-%C3%A1vila-brice%C3%B1o-wuu6e/>
2. **FastAPI.** (s.f.). *FastAPI.* <https://fastapi.tiangolo.com/>
3. **GeeksforGeeks.** (2025). *Complete Guide to Clean Architecture - System Design.* <https://www.geeksforgeeks.org/system-design/complete-guide-to-clean-architecture/>
4. **IBM.** (s.f.). *What is a monolithic architecture?* <https://www.ibm.com/think/topics/monolithic-architecture>
5. **Baldin, C. M., & Casella, V. M.** (2025). *Comparison of PlanetScope and Sentinel-2 Spectral Channels and Their Alignment via Linear Regression for Enhanced Index Derivation.* *Geosciences*, 15(5), 184. <https://doi.org/10.3390/geosciences15050184>
6. **Microsoft.** (s.f.). *Azure Blob Storage documentation.* <https://learn.microsoft.com/en-us/azure/storage/blobs/>
7. **Osborne, J** (2024). *FastAPI with API Key Authentication.* Medium. <https://medium.com/@joerosborne/fastapi-with-api-key-authentication-f630c22ce851>
8. **Planet Labs.** (s.f.-a). *APIs.* Planet Developer Center. <https://docs.planet.com/develop/apis/>
9. **Planet Labs.** (s.f.-b). *Planet University.* <https://university.planet.com/>
10. **Planet Labs.** (s.f.-c). *PlanetScope Imagery.* Planet Documentation. <https://docs.planet.com/data/imagery/planetscope/>
11. **Planet Labs.** (2025). *Planet PSScene Imagery Product Specification.* https://assets.planet.com/docs/Planet_PSScene_Imagery_Product_Spec_letter_screen.pdf
12. **Pydantic.** (s.f.). *Pydantic.* <https://docs.pydantic.dev/>
13. **Red Hat.** (2024). *¿Qué es un webhook?* <https://www.redhat.com/es/topics/automation/what-is-a-webhook>
14. **Sentinel Hub.** (s.f.). *PlanetScope NDVI.* Custom Scripts. <https://custom-scripts.sentinel-hub.com/planet/planetscope/ndvi/>