



Integración y Refactorización de Sistemas Informáticos en Pastech



Autores:

- Ramos Kees, Teo - teoramites@gmail.com
- Romeo, Renzo Agustín - renzoromeo44@gmail.com

Director:

- Mg. Ing. Adolfo Tomás Spinelli

Proyecto final para optar al grado de Ingeniero en Informática

Mar del Plata, Febrero, 2025



RINFI es desarrollado por la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución- NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).



Integración y Refactorización de Sistemas Informáticos en Pastech



Autores:

- Ramos Kees, Teo - teoramites@gmail.com
- Romeo, Renzo Agustín - renzoromeo44@gmail.com

Director:

- Mg. Ing. Adolfo Tomás Spinelli

Proyecto final para optar al grado de Ingeniero en Informática

Mar del Plata, Febrero, 2025

Agradecimientos

Nos gustaría expresar un especial agradecimiento:

A nuestras familias, que siempre nos acompañaron y aportaron los recursos necesarios para poder progresar en nuestros estudios.

A la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata, por brindarnos el espacio para capacitarnos y aprender en un contexto ameno y de excelente trabajo.

A nuestros docentes, que a lo largo de toda la carrera nos transmitieron su compromiso y dedicación con la profesión.

A los involucrados en Pastech Satelital y Pastech Pasturómetro con quienes mantuvimos contacto durante el desarrollo de este trabajo, y que siempre estuvieron predispuestos a brindar su ayuda y contestar las dudas que surgieron.

A nuestros compañeros de facultad, que nos acompañaron a lo largo del trayecto.

Índice

Agradecimientos.....	2
Índice.....	3
Resumen.....	6
Introducción.....	7
Objetivo General.....	7
Objetivos Específicos.....	8
Etapas de Análisis y Diseño.....	10
Reuniones.....	10
Análisis de Riesgos.....	11
Análisis FODA.....	13
Estudio del Código Existente de Pastech Satelital.....	14
Diseño de la Arquitectura la Plataforma Unificada.....	16
Dependencias y Colaboraciones Externas.....	20
Definición de Tareas y Estimación de Tiempos.....	21
Metodología de Trabajo.....	24
Ejecución del Proyecto.....	26
Contexto Inicial.....	26
Sistemas Existentes y Tecnologías Utilizadas.....	26
Modificaciones Previas al Inicio del Desarrollo.....	28
Optimización de Tiempos de Compilación.....	28
Documentación de la API REST.....	30
Reestructuración de los Endpoints REST.....	31
Implementación de la Solución.....	32
Pasturómetro Electrónico.....	32
Requerimientos Funcionales.....	33
Implementación.....	35
Modificaciones a la base de datos.....	35

Nuevos endpoints REST.....	36
Vistas.....	37
Vista Integrada.....	38
Vista de Mediciones de cada Potrero.....	39
Vista de Mapa.....	40
Vista de Stock.....	41
Calibraciones.....	42
Implementación.....	43
Cálculo de curva de calibración.....	43
Nuevos endpoints REST.....	45
Tabla 3: Endpoints implementados para la ruta /api/calibraciones.....	46
Vistas.....	46
Vista principal.....	46
Vista completar calibración.....	47
Aplicación Móvil.....	48
Problemas Iniciales.....	49
Mejoras Realizadas.....	50
Creación de Calibración.....	50
Georeferenciación de las mediciones.....	54
Despliegue de la aplicación en Android e iOS.....	56
Despliegue de la Aplicación en iOS.....	57
Despliegue de la aplicación en Play Store.....	59
Validación del producto.....	59
Herramientas utilizadas.....	60
Problemas encontrados y soluciones implementadas.....	61
Errores en el Registro de Mediciones.....	61
Valores Negativos en el Cálculo de Peso del Pasto.....	62
Ubicación y Asociación de Mediciones con Potreros.....	63
Mediciones no asignadas debido a incompatibilidad en coordenadas.....	65

Memorias del Proyecto	66
Cumplimiento de objetivos.....	66
Objetivo General: Integración de subsistemas.....	66
Mejora en la experiencia de usuario.....	66
Optimización del acceso a la información.....	67
Nuevas funcionalidades que facilitan la toma de decisiones basadas en datos precisos.....	67
Objetivos Específicos.....	68
Gestión de Potreros.....	68
Visualización de Mediciones Georreferenciadas.....	68
Disponibilidad de Pasto por Grupos.....	69
Consulta de Disponibilidad Estimada mediante Imágenes Satelitales.....	69
Gestión de Circuitos de Potreros.....	69
Gestión de Calibraciones.....	69
Adaptación de la Aplicación Móvil de Pasturómetro Electrónico.....	69
Aspecto Crítico para el Éxito del Proyecto.....	70
Etapas de Análisis.....	71
Validación del Producto.....	72
Comparación entre los plazos estimados y los reales.....	74
Conclusiones	77
Bibliografía	79
Recursos Web.....	79
Libros.....	80
Anexo: Glosario	81
Anexo: Documentación de Endpoints	83
Anexo: Refactorización de Rutas	87

Resumen

El presente informe detalla el desarrollo del Trabajo Final de la carrera de Ingeniería en Informática, dictada en la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata. El trabajo se llevó a cabo sobre Pastech, una solución integral para la gestión eficiente de pasturas.

El proyecto abordó los dos productos principales de Pastech. Por un lado, Pastech Satelital, una herramienta que, mediante la lectura de imágenes satelitales estimaba la disponibilidad de pasto en cada potrero y utilizaba inteligencia artificial y modelos matemáticos para predecir el crecimiento del pasto. Por otro lado, el Pasturómetro Electrónico, un dispositivo físico que tomaba mediciones georreferenciadas para determinar la altura comprimida del pasto y enviaba los datos a una aplicación móvil conectada donde se visualizaban para facilitar la toma de decisiones.

El objetivo principal del proyecto es integrar ambas plataformas en una sola, mejorando la experiencia del usuario y ofreciendo nuevas funcionalidades. Por solicitud del demandante, se decidió mantener activa la plataforma Pastech Satelital, incorporando en ella las funcionalidades del Pasturómetro Electrónico, consolidando así una única solución integral.

Palabras clave: *Gestión de pasturas, Pasturómetro Electrónico, Imágenes Satelitales, Georreferenciación, Integración de Plataformas.*

Introducción

La gestión eficiente del pasto es un elemento clave en la producción agropecuaria, ya que el pasto constituye la principal fuente de alimento para el ganado. Optimizar su disponibilidad y calidad permite mejorar la productividad, reducir costos y hacer un uso sostenible de los recursos naturales. Sin embargo, gestionar grandes extensiones de terreno y monitorear el crecimiento del pasto de manera precisa puede ser complejo sin herramientas tecnológicas avanzadas.

En este contexto surge Pastech, una solución integral para la gestión inteligente de pasturas. Pastech combina tecnologías innovadoras para ofrecer datos precisos y en tiempo real que facilitan la toma de decisiones en el campo. La plataforma se compone de dos subsistemas principales:

- Pastech Satelital: Utiliza imágenes satelitales para estimar la disponibilidad de pasto en cada potrero y emplea inteligencia artificial y modelos matemáticos para predecir su crecimiento futuro.
- Pasturómetro Electrónico: Es un dispositivo físico que permite realizar mediciones georreferenciadas de la altura comprimida del pasto mediante sensores. Estas mediciones se pueden visualizar en tiempo real en una aplicación móvil. Los valores luego son utilizados para calcular la disponibilidad de pasto de todo el potrero mediante las calibraciones, que son funciones cuyo valor de entrada es la altura comprimida del pasto y su valor de salida es una estimación de la disponibilidad de pasto en el potrero.

Objetivo General

Inicialmente el sistema de Pastech estaba dividido en dos subsistemas independientes, lo que generaba varias desventajas. Entre ellas, una experiencia de usuario más compleja y poco conveniente, ya que los usuarios debían alternar entre ambas plataformas para acceder a la información necesaria. Esto dificulta la gestión de los datos y la toma eficiente de decisiones.

El objetivo general de este proyecto es integrar y refactorizar los subsistemas **Pastech Satelital** y **Pasturómetro Electrónico** en una única plataforma unificada que permita a los usuarios gestionar de manera eficiente sus pasturas. Para ello, la nueva versión de la plataforma buscará:

- **Mejorar la experiencia de usuario:** haciendo las interacciones con la plataforma más intuitivas y fáciles de entender.
- **Optimizar el acceso a la información:** ofreciendo en una única plataforma todas las herramientas relacionadas a Pastech Satelital y Pasturómetro Electrónico.
- **Ofrecer nuevas funcionalidades facilitando la toma de decisiones basadas en datos precisos:** ofreciendo nuevas formas de visualizar la información y ofreciendo la opción de tomar mediciones más precisas.

Objetivos Específicos

Con el objetivo general definido se identificaron los requerimientos para poder llevar a cabo la integración de las plataformas existentes en una sola. La plataforma unificada debía tener las siguientes funcionalidades:

- Proveer a los usuarios herramientas para gestionar de forma sencilla e intuitiva la información de sus potreros, incluyendo la creación, edición y eliminación de los mismos dentro de la plataforma.
- Permitir a los usuarios acceder fácilmente a las mediciones georreferenciadas realizadas con el Pasturómetro Electrónico, ofreciendo una visión detallada y gráfica de los datos recolectados. Las funcionalidades solicitadas son las siguientes:
 - Elegir una fecha para analizar las mediciones de la misma.
 - Mostrar un gráfico de barras con el stock por fecha.

- Visualizar en un mapa la ubicación de las mediciones de una fecha determinada.
- Distinguir las mediciones por sector, mostrando los datos en una tabla.
- Visualizar qué calibración fue utilizada para calcular el stock para las mediciones de una fecha.
- Cambiar la calibración con la que se calcula el stock para una fecha.
- Visualizar, por medio de gráficos de barras, la disponibilidad de pasto de los potreros, los circuitos y las zonas.
- Permitir a los usuarios consultar la disponibilidad de pasto estimada mediante imágenes satelitales, permitiendo:
 - Visualizar el stock por fecha en un gráfico.
 - Visualizar el perímetro del potrero en un mapa.
 - Visualizar los valores del índice NDVI para la superficie del potrero.
- Permitir a los usuarios la creación y gestión de circuitos de potreros, permitiendo una organización más estructurada y acorde a las prácticas productivas del usuario. Debe permitir:
 - Crear circuitos.
 - Asignar potreros a circuitos.
- Proveer una interfaz que permita al usuario crear y gestionar calibraciones en base a mediciones tomadas con el Pasturómetro Electrónico. La interfaz debe permitir:
 - Visualizar las calibraciones en una tabla.
 - Completar las calibraciones con los pesos medidos.
 - Exportar las calibraciones a una hoja de cálculo.

Etapa de Análisis y Diseño

Reuniones

Para garantizar el éxito del desarrollo e integración de los subsistemas de Pastech, se llevó a cabo una etapa de análisis que permitió definir los requerimientos del sistema, su alcance y los plazos de entrega. Este proceso comenzó con reuniones con el referente funcional, que aportó una visión detallada sobre las necesidades del proyecto y los objetivos principales que debía cumplir la solución propuesta.

En las primeras reuniones, se presentó un panorama general sobre qué es Pastech y cuál es el producto que ofrece. Se discutieron los objetivos principales de la plataforma y cómo esta contribuye al manejo eficiente de las pasturas. Además, el referente funcional describió el estado actual de los sistemas existentes y los desafíos presentes. Este contexto permitió al equipo comprender el punto de partida y la dirección que debía tomar el proyecto.

Otro aspecto importante de esta etapa fue la revisión de los sistemas que componían Pastech en ese momento: Pastech Satelital y Pasturómetro Electrónico. Durante las reuniones se analizaron sus funcionalidades principales, los datos que manejaban y las tecnologías utilizadas en su implementación.

Una vez que el equipo se familiarizó con el proyecto, se procedió a determinar los requerimientos funcionales y no funcionales del sistema. Estos requerimientos establecieron qué debía lograr la nueva plataforma, desde la integración de las funcionalidades existentes hasta la incorporación de nuevas herramientas que facilitaran la gestión de pasturas. A su vez, se definieron las prioridades para cada funcionalidad, asegurando que los objetivos más críticos fueran abordados en las primeras etapas del desarrollo.

Finalmente, se discutieron los plazos tentativos de entrega del proyecto. Este proceso incluyó la estimación del tiempo necesario para cada etapa de desarrollo y la identificación de posibles riesgos o limitaciones que pudieran influir en la planificación.

Esta etapa fue clave para establecer expectativas realistas tanto para el equipo de desarrollo como para el referente funcional.

Análisis de Riesgos

Antes de iniciar el desarrollo de cualquier proyecto, es fundamental identificar los riesgos potenciales que puedan surgir y planificar estrategias para mitigarlos. Este enfoque permite minimizar pérdidas de recursos, evitar retrabajos y garantizar un desarrollo más eficiente.

En este contexto se llevó a cabo un análisis de riesgos, una metodología ampliamente utilizada en proyectos tecnológicos que se enfoca en identificar, evaluar y priorizar los riesgos que podrían afectar el desarrollo. Para cada riesgo identificado, se asignaron dos valores clave: **probabilidad de ocurrencia** e **impacto en el proyecto**, utilizando una escala del 1 al 3. Estos valores permiten asignar un valor numérico a cada uno de los riesgos:

$$R = P \times I$$

donde R representa el nivel de riesgo, P la probabilidad e I el impacto. Según esta metodología, cualquier riesgo con un valor mayor o igual a 6 fue clasificado como importante, lo que requería acciones para prevenir que ocurriera o minimizar sus consecuencias.

El análisis comenzó con una revisión de las posibles situaciones adversas que podrían surgir durante el ciclo de vida del proyecto. Entre los aspectos evaluados se incluyeron dependencias tecnológicas, plazos ajustados y cambios en los requerimientos. Además, se consideraron factores externos, como la dependencia de terceros para la provisión de datos o servicios clave, que podrían afectar el avance del proyecto.

Una vez identificados, los riesgos fueron priorizados según su nivel de criticidad, y para los casos necesarios se propusieron estrategias de mitigación.

En la Tabla 1 se presentan los riesgos identificados, junto con su probabilidad, impacto y nivel de criticidad.

Riesgo	Consecuencias	P	I	R	Plan de contingencia
Cierre de Pastech	<ul style="list-style-type: none"> • Cancelación del proyecto. • Pérdida de tiempo y recursos. 	1	3	3	No es necesario.
Abandono del proyecto por parte de un integrante	<ul style="list-style-type: none"> • Sobrecarga de trabajo para el otro integrante del equipo. • Retraso en la entrega del producto. 	1	3	3	No es necesario.
Problemas de coordinación en el equipo	<ul style="list-style-type: none"> • Retrasos en la implementación. • Retraso en la entrega del producto. 	3	2	6	Establecer un cronograma para coordinar las sesiones de trabajo.
Dificultad para coordinar reuniones con equipo de trabajo anterior	<ul style="list-style-type: none"> • Retraso en la entrega del producto. 	2	3	6	Recopilar toda la información posible referida a los sistemas existentes antes del inicio del proyecto, para no depender tanto de las reuniones.
Cambios en los requerimientos iniciales	<ul style="list-style-type: none"> • Retrabajo. • Pérdida de tiempo. 	3	2	6	Aumentar la frecuencia de reuniones con el demandante, para evitar retrabajo.
Falta de tiempo suficiente para completar todas las tareas planificadas	<ul style="list-style-type: none"> • Incumplimiento de plazos. • Entrega de un producto incompleto y de menor calidad. 	1	2	2	No es necesario.
Falta de claridad sobre el producto final	<ul style="list-style-type: none"> • Diferencias entre las expectativas del demandante y el 	1	3	3	No es necesario.

	producto entregado.				
Falta de documentación técnica	<ul style="list-style-type: none"> Mayor tiempo invertido para comprender los sistemas existentes. 	3	2	6	Elaboración de documentación sobre los módulos del proyecto.

Tabla 1: Riesgos identificados y sus características

Análisis FODA

El análisis FODA es una etapa crucial dentro del proceso de evaluación del proyecto. Este método permite identificar las fortalezas, oportunidades, debilidades y amenazas relacionadas con el desarrollo, ofreciendo una visión más completa del contexto en el que se trabaja.

Los beneficios de esta práctica radican en su capacidad para anticipar riesgos, aprovechar ventajas estratégicas y diseñar planes que mitiguen posibles dificultades. Al realizar este análisis, el equipo pudo obtener información valiosa que posteriormente se utilizó para orientar las decisiones de desarrollo y planificación, contribuyendo a minimizar el riesgo de fracaso y a optimizar los resultados del proyecto.

A continuación, se detalla el análisis realizado por el equipo:

Fortalezas:

- Se cuenta con asesoramiento de profesionales en el área agronómica que nos instruyen en su dominio y nos aportan los conocimientos necesarios.
- Se mantiene una constante comunicación con el referente funcional del proyecto que provee una retroalimentación esencial para satisfacer los requerimientos correctamente.
- Se poseen los conocimientos necesarios sobre las tecnologías a usar en el proyecto, lo que nos facilita el desarrollo de las funcionalidades e integración de plataformas.

- Experiencia en trabajo en conjunto entre los integrantes del equipo.
- Experiencia profesional de ambos integrantes del equipo con las tecnologías utilizadas en los sistemas previamente desarrollados y tecnologías a utilizar en el nuevo sistema integrado.

Oportunidades:

- El sistema sería pionero a nivel nacional en la gestión inteligente del pasto.
- El sistema brindaría una reducción de costos para el productor y una mayor velocidad de control.

Debilidades:

- La integración de nuevas soluciones con la infraestructura existente desconocida puede presentar desafíos técnicos.
- Pocos conocimientos técnicos del equipo sobre la industria ganadera, específicamente sobre la gestión del pasto.

Amenazas:

- Resistencia al cambio por parte de los productores, pues se les estaría cambiando la forma de interactuar con el sistema.
- La competencia podría adoptar tecnologías similares en un mismo sistema, lo que destaca la necesidad de una rápida implementación con el objetivo de llegar primero al mercado.

Estudio del Código Existente de Pastech Satelital

Durante la fase inicial del proyecto, uno de los principales desafíos fue analizar el código preexistente de la plataforma. Una de las mayores preocupaciones fue el nivel de complejidad y el posible costo, en términos de tiempo y esfuerzo, que implicaría comprender a fondo dicho código para poder extenderlo con las nuevas funcionalidades requeridas.

Ante esta situación, se evaluaron tres posibles enfoques:

- **Reescribir el sistema desde cero con nuevas tecnologías:** Esta opción permitiría trabajar con una base de código completamente desarrollada por el equipo actual, lo que facilitaría su mantenimiento y evolución a futuro. Sin embargo, también representaba un impacto crítico en el costo y la duración del proyecto, y dado que el demandante tenía como prioridad una entrega rápida, esta opción fue descartada.
- **Mantener el código existente con asistencia del equipo de desarrollo anterior:** Se contempló la posibilidad de conservar la base actual e incorporar las funcionalidades necesarias con el apoyo del equipo anterior mediante reuniones. Esta alternativa fue rápidamente descartada, ya que era difícil coordinar reuniones y establecer una comunicación fluida con los desarrolladores anteriores.
- **Exploración autónoma del código existente:** La opción finalmente elegida fue realizar un análisis del código actual sin la asistencia directa del equipo anterior, salvo por algunas reuniones iniciales clave para aclarar aspectos fundamentales. Aunque esta estrategia implicaba dedicar tiempo adicional a estudiar el funcionamiento del sistema, se consideró la alternativa más viable, ya que minimizaría la dependencia de actores externos, conservaría la lógica ya implementada, y evitaría el costo elevado que supondría una reescritura completa.

Diseño de la Arquitectura la Plataforma Unificada

Luego de realizar el análisis inicial del código existente de las plataformas de Pastech Satelital y Pasturómetro Electrónico, y decidir qué módulos serían creados, modificados y conservados, se definió la arquitectura final de la plataforma unificada.

En las figuras 1 y 2 se muestran las arquitecturas de los sistemas preexistentes:

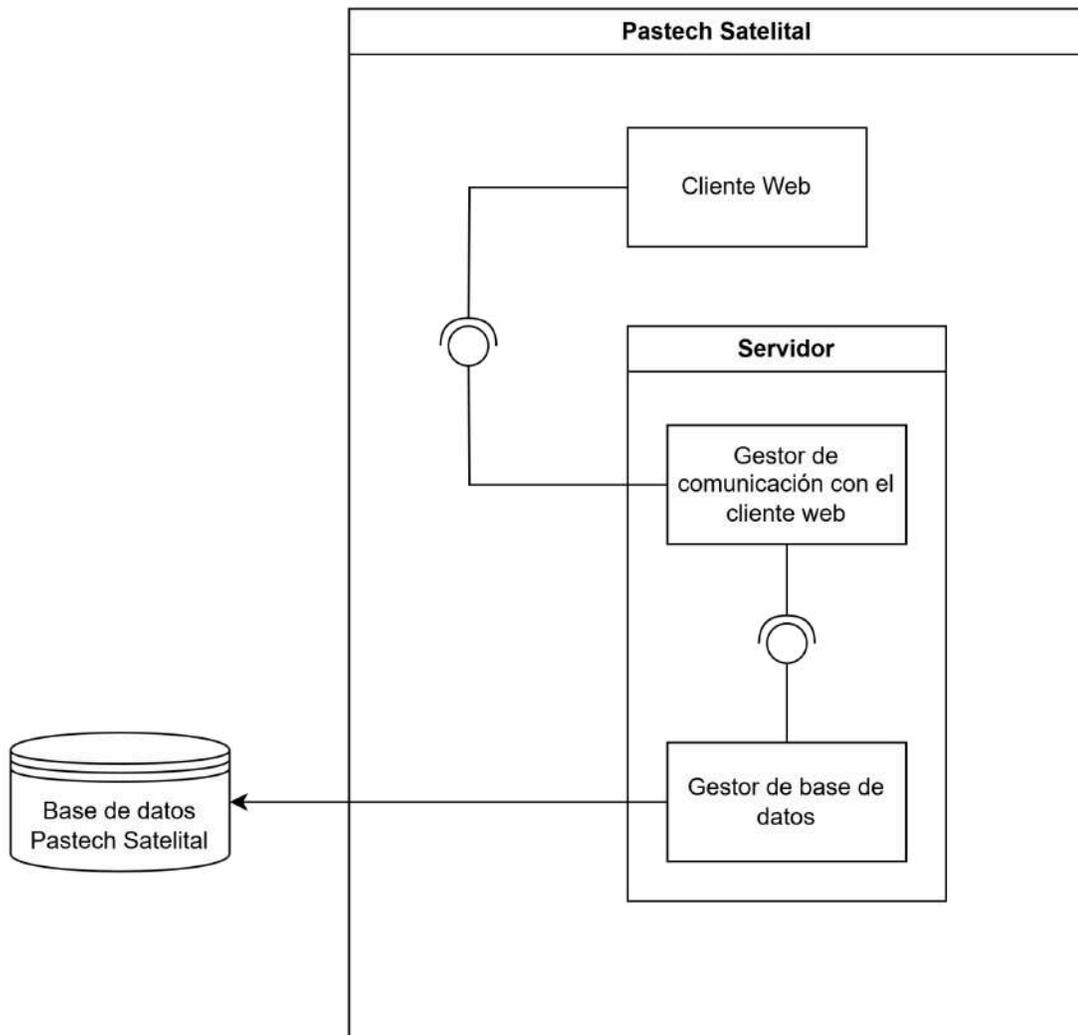


Figura 1: Arquitectura de Pastech Satelital previa al inicio del proyecto

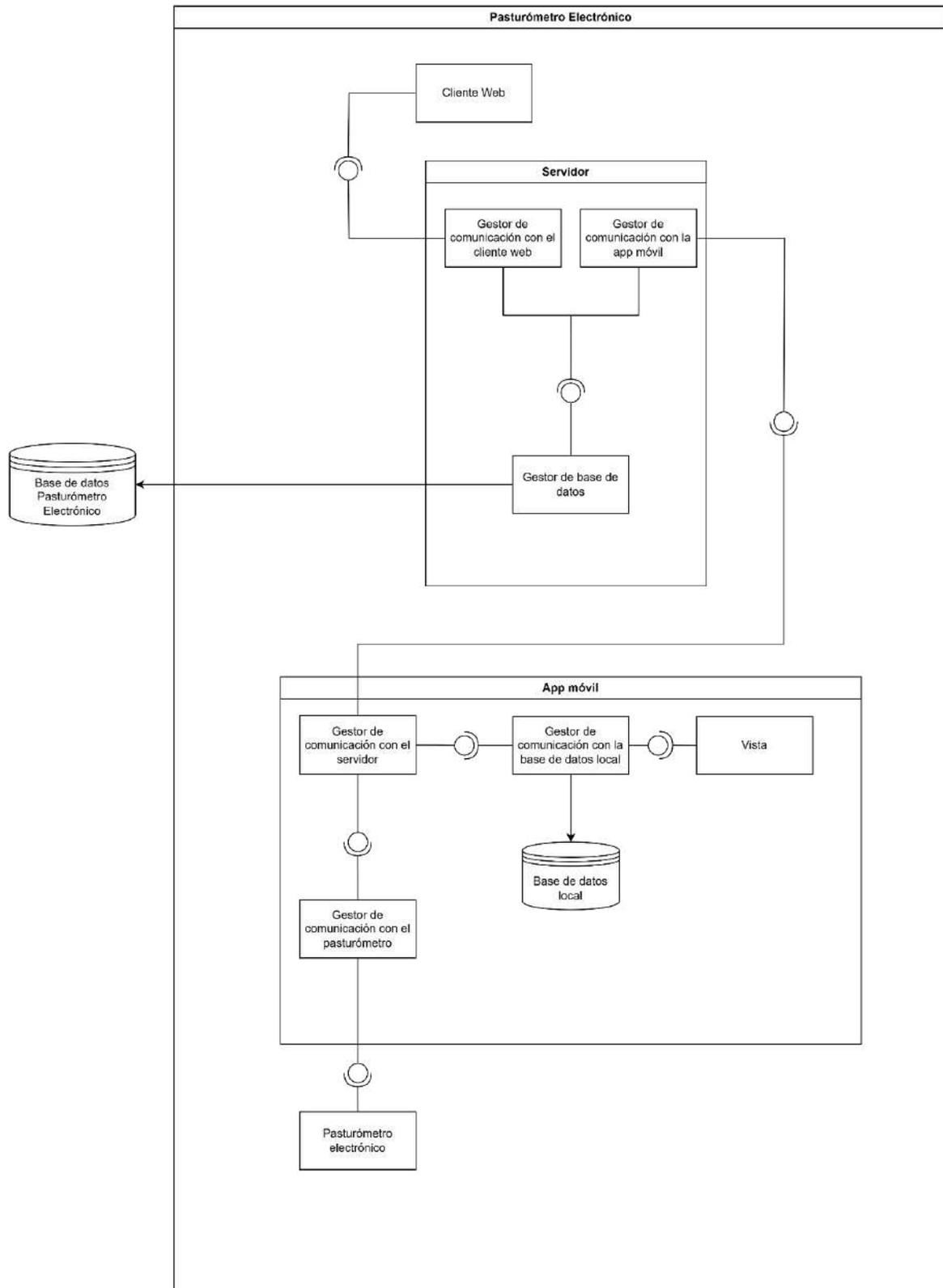


Figura 2: Arquitectura de Pasturómetro Electrónico previa al inicio del proyecto

Como se mencionó previamente, el demandante determinó que la plataforma de Pastech Satelital sería la seleccionada para incluir el resto de los módulos y de esta forma convertirse en la plataforma unificada, incluyendo toda la funcionalidad relacionada al pasturómetro.

De esta manera se decidió que la nueva arquitectura utilizaría el servidor web existente en Pastech Satelital, modificándolo para soportar las nuevas funcionalidades solicitadas por el demandante. Esto implicó la creación de un módulo para posibilitar la comunicación con la aplicación móvil, y la modificación de los gestores de comunicación con el cliente web y comunicación con la base de datos.

También se decidió que el cliente web a utilizar sea el preexistente en Pastech Satelital, añadiendo las vistas necesarias para integrar la funcionalidad del Pasturómetro Electrónico.

La aplicación móvil sería la utilizada previamente en Pasturómetro Electrónico, pero con las modificaciones necesarias para comunicarse con el servidor de la plataforma unificada, almacenar y sincronizar los nuevos datos y mostrar en nuevas vistas las funcionalidades solicitadas por el demandante.

En la figura 3 se presenta un diagrama con la arquitectura de la plataforma unificada, que incluye una referencia para mostrar qué módulos fueron creados, modificados e incluidos sin modificar.

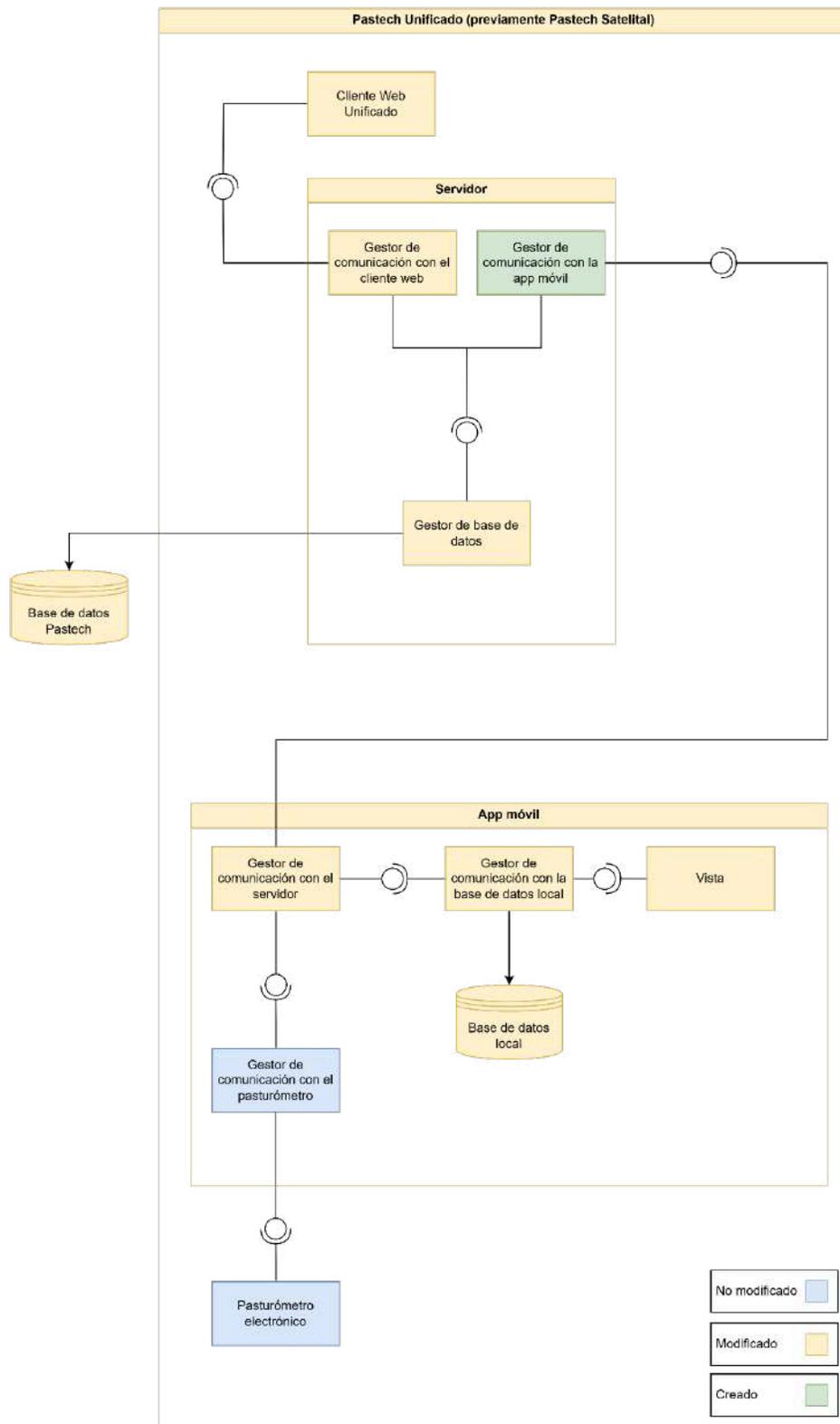


Figura 3: Arquitectura de Pastech Unificado con referencias sobre los componentes

Dependencias y Colaboraciones Externas

Durante la etapa de análisis del proyecto hubo interacción con los desarrolladores y responsables de cada sistema de Pastech para poder comprender los módulos existentes y establecer las bases del desarrollo.

La aplicación móvil fue desarrollada inicialmente por Nicolás Escude y Matías Veitch, quienes proporcionaron acceso al repositorio del código fuente y recursos clave como documentación de interfaces y guías para ejecutar la aplicación. Para trabajar de manera ordenada se utilizó un repositorio GIT. Por otro lado, el sistema basado en imágenes satelitales fue desarrollado por Lorenzo Longaretto y Bautista Rodríguez, con quienes hubo dos reuniones al inicio del proyecto para aclarar cuestiones clave sobre el funcionamiento actual del sistema.

Para profundizar en el funcionamiento del pasturómetro electrónico y su protocolo de comunicación, se realizó una reunión presencial con los encargados de su desarrollo: Melisa Kuzman, Juan Manuel López, y Rodrigo Russo, en el departamento de Electrónica de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata. La interacción presencial fue suficiente para comprender cómo se utiliza el dispositivo y cómo se integra con el resto de los sistemas.

Esta etapa fue de ayuda para comprender mejor cada subsistema que integraría al producto final y como sería su interacción.

Definición de Tareas y Estimación de Tiempos

La etapa de definición de tareas fue un componente importante dentro del desarrollo del proyecto, ya que permitió al equipo organizar y estructurar las actividades necesarias para alcanzar los objetivos establecidos. En esta fase se identificaron las tareas a realizar, su naturaleza, y su relación con el proyecto en su totalidad.

Definir e identificar cada tarea también estableció una hoja de ruta que guió al equipo durante todo el proceso. Además, esta etapa proporcionó una estimación de la envergadura del proyecto y de las horas requeridas para completarlo, lo que contribuyó a una planificación más realista en cuanto a estimación de tiempos.

A continuación, se detallan las tareas definidas por el equipo:

- **Análisis de objetivos y requerimientos:** Reuniones con el referente funcional para realizar el relevamiento de los distintos requerimientos y definir el alcance del proyecto, e investigación sobre el área de trabajo.
- **Estudio y análisis de código:** Análisis de código de las plataformas actuales en funcionamiento.
- **Conformidad del demandante:** Reuniones con el referente funcional informando avances en el desarrollo. Pueden existir modificaciones por parte del referente funcional.
- **Desarrollo e integración de plataformas:** Codificación e implementación de las nuevas funcionalidades, y modificaciones al sistema existente.
- **Testing:** Realización de pruebas para verificar el correcto funcionamiento del sistema garantizando su calidad, confiabilidad y robustez.
- **Modificaciones a la base de datos:** Modificaciones de la base de datos de los sistemas de desarrollo y producción.

- **Despliegue:** Despliegue de nuevas funcionalidades de la plataforma sin impactar en el funcionamiento de la plataforma actual en producción.
- **Elaboración de bitácora:** Documentación de las tareas realizadas cada día.
- **Elaboración del informe:** Elaboración del informe del Trabajo Final.

Una vez definidas las tareas a realizar y su duración estimada, se elaboró un diagrama de Gantt que permitió visualizar la disposición de las tareas en el tiempo. Este diagrama permitió obtener una visión clara de la duración estimada del proyecto, y de la relación y solapamiento entre las diferentes tareas.

En la Figura 4 se muestra el diagrama realizado:

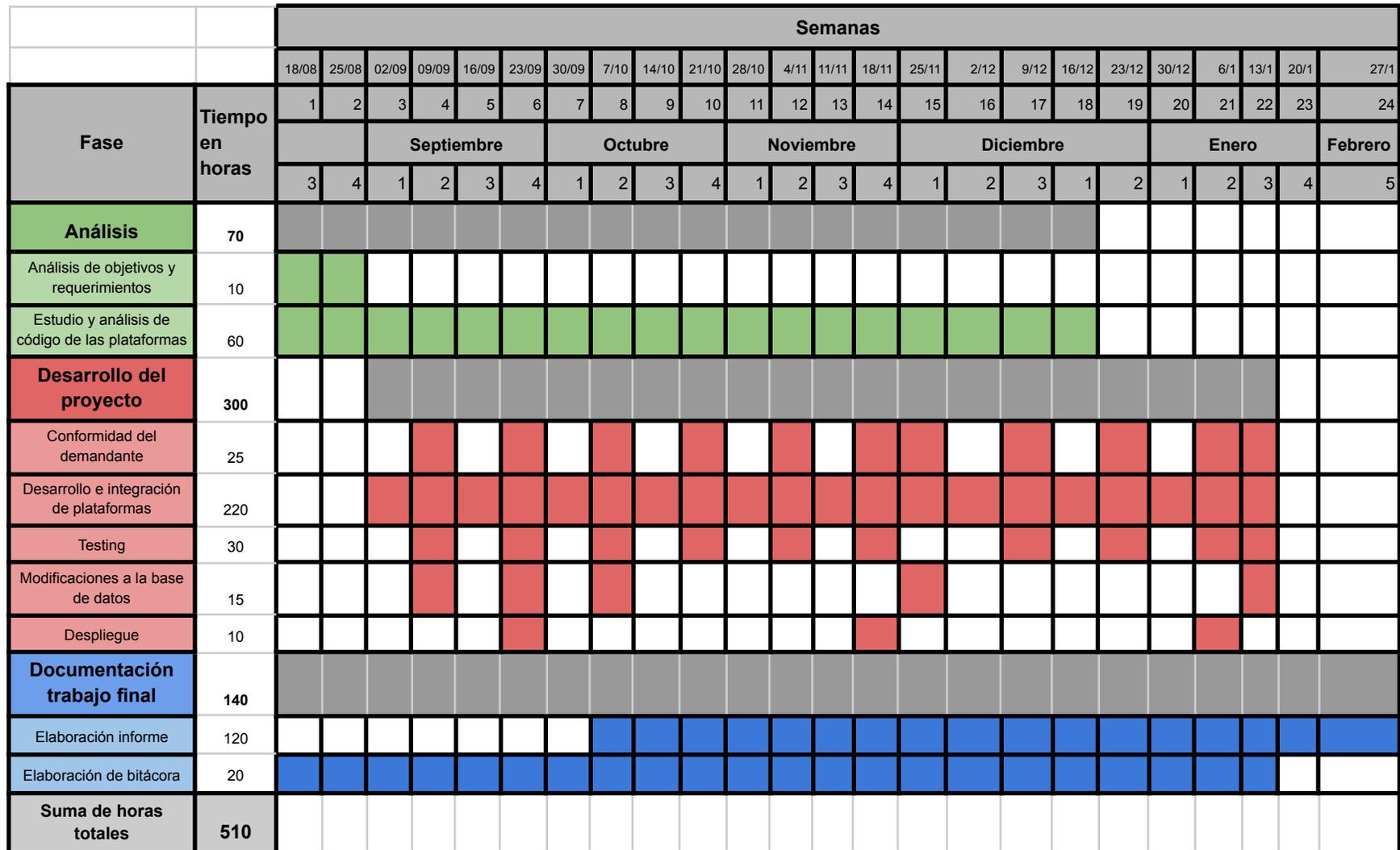


Figura 4: Diagrama de Gantt con la planificación inicial

Metodología de Trabajo

Para asegurar el éxito en la implementación de un proyecto de esta envergadura, fue necesario decidir qué metodología de trabajo resultaría adecuada para adaptarse a los requerimientos cambiantes y garantizar una validación frecuente con el demandante. Con el objetivo de satisfacer estas necesidades críticas y luego de un análisis comparativo de las principales metodologías de trabajo, se optó por el uso de metodologías ágiles. Estas metodologías se caracterizan por su enfoque en la entrega incremental de valor y su flexibilidad para gestionar cambios en los requisitos, y en la división del trabajo en iteraciones cortas y manejables, lo que facilita el control del progreso y la adaptación a nuevas necesidades.

Una de las metodologías ágiles más populares, y la elegida en este caso, es SCRUM. Esta metodología organiza el trabajo en sprints, ciclos cortos de tiempo predefinido, donde se planifica, desarrolla y revisa un conjunto de tareas específicas. La principal ventaja de este enfoque es que permite tener una validación constante con el referente funcional y adaptarse a requerimientos cambiantes de forma eficiente.

Para llevar a cabo esta metodología, se implementaron las siguientes prácticas:

- El trabajo se organizó en sprints de 1 semana, ya que esta duración era suficiente para completar las tareas asignadas y permitía mantener un flujo continuo de validación con el referente funcional.
- El referente funcional utilizaba un documento compartido para detallar los cambios requeridos, las nuevas funcionalidades y otras consideraciones. A medida que las tareas se completaban, el avance se reflejaba en este documento, permitiendo al referente funcional tener una visión clara del estado del proyecto. Inicialmente se contempló la idea de utilizar un tablero Kanban para gestionar las tareas a realizar, pero finalmente se decidió utilizar el documento compartido ya que representaba una opción más conveniente y

cómoda para que el cliente pudiera comunicar los requerimientos y ver su avance.

- Se mantuvo un registro continuo de los avances en una bitácora interna del equipo. Este documento colaborativo permitió registrar las tareas realizadas, los desafíos encontrados y las soluciones implementadas.

Ejecución del Proyecto

Contexto Inicial

Sistemas Existentes y Tecnologías Utilizadas

Como se mencionó anteriormente, Pastech está compuesta por dos sistemas principales: Pastech Satelital y Pasturómetro Electrónico. La implementación de estos sistemas había sido iniciada por equipos de trabajo previos, por lo que fue necesario realizar un análisis de las tecnologías elegidas y de la arquitectura definida para cada uno de ellos.

La aplicación web de Pastech Satelital, seleccionada por el demandante para ser ampliada, está compuesta por: un servidor desarrollado con Express, un cliente web construido con React, y una base de datos relacional MySQL. A su vez, tanto el servidor como la base de datos estaban alojados en Railway.

A continuación, se describen en mayor detalle estas tecnologías:

Express es un framework para Node.js, ampliamente reconocido por su capacidad para construir servidores web y desarrollar APIs REST de manera eficiente. Las APIs REST (siglas de Representational State Transfer) permiten que distintos sistemas, como el cliente web, el servidor y la base de datos, intercambien información siguiendo un conjunto de reglas estándar.

El protocolo REST presenta un enfoque stateless (sin estado) a la hora de pensar el servidor, lo que implica que el servidor no almacena información sobre el estado de las solicitudes anteriores del cliente. Este diseño reduce el uso de memoria en el servidor, ya que cada solicitud es independiente, y permite manejar un mayor número de solicitudes simultáneamente.

REST utiliza los verbos HTTP más comunes, como GET (obtener datos), POST (crear nuevos recursos), PUT (actualizar recursos existentes) y DELETE (eliminar recursos), para realizar operaciones específicas sobre los datos.

Por otro lado, React es una biblioteca de JavaScript diseñada para desarrollar interfaces de usuario dinámicas, interactivas y reactivas. Su enfoque se basa en dividir la interfaz en componentes reutilizables y parametrizables, cada uno responsable de una parte específica de la aplicación. Esto no solo facilita la reutilización de código, sino que también simplifica el desarrollo, el mantenimiento y la escalabilidad..

El motor de base de datos relacional elegido por el equipo previo para persistir la información de la aplicación es MySQL.

La infraestructura de la aplicación incluía dos bases de datos separadas: una utilizada para el entorno de desarrollo, donde se realizaban pruebas y ajustes sin afectar al funcionamiento del sistema en producción, y otra utilizada en el entorno de producción, que gestionaba los datos reales de la aplicación. Esta separación de entornos es una práctica usual para garantizar la estabilidad y seguridad del sistema, y permitir un desarrollo controlado y un despliegue de cambios de manera segura.

Tanto el servidor como la base de datos estaban alojados en Railway, un servicio de tipo Infrastructure as a Service (IaaS) diseñado para simplificar el despliegue y la gestión de aplicaciones web y bases de datos. Esta plataforma ofrece herramientas para monitorear métricas de uso, ver registros generados por la aplicación y gestionar configuraciones, lo que facilita el mantenimiento y la supervisión del sistema.

El flujo de CI/CD (Continuous Integration and Continuous Deployment) estaba configurado para automatizar el proceso de despliegue. Esto significaba que cualquier cambio enviado al repositorio Git utilizado por el equipo se desplegaba automáticamente. A la hora de desplegar modificaciones que requirieran cambios en la estructura de la base de datos, la aplicación experimentaría un tiempo de mantenimiento, en la que no sería utilizable por los usuarios finales hasta que el código de la aplicación haya sido desplegado y la estructura de la base de datos modificada. De esta manera se evitan inconsistencias a lo largo de la plataforma. El demandante estaba de acuerdo con esta estrategia, y no representaba un problema.

La infraestructura incluía dos entornos separados: uno para desarrollo, donde se probaban nuevas características y modificaciones, y otro para producción, que manejaba las operaciones en tiempo real. Considerando que al trabajar en un entorno de producción es sumamente importante no perjudicar la experiencia de los usuarios actuales de la plataforma, esta configuración permitió realizar modificaciones y pruebas sin comprometer la funcionalidad del sistema en uso por los usuarios finales.

Por otro lado, la aplicación móvil fue desarrollada utilizando React Native, una biblioteca basada en React que permite construir interfaces de usuario tanto para aplicaciones móviles como para aplicaciones web, manteniendo las principales ventajas de React como la creación de componentes reutilizables, la gestión sencilla del estado de la aplicación y una experiencia de desarrollo más eficiente gracias al uso compartido de lógica entre plataformas.

En cuanto al Pasturómetro Electrónico, se trata de un instrumento diseñado para medir la altura del pasto comprimido. Este dispositivo envía datos compuestos por la localización geográfica y la altura medida directamente a la aplicación móvil, permitiendo el registro y análisis de las mediciones en tiempo real.

La elección de las tecnologías realizadas por los equipos previos resultó oportuna, ya que los integrantes del presente equipo cuentan con experiencia trabajando con las herramientas más relevantes del sistema, como Express y React. Esto facilitó la adaptación al proyecto y la implementación de las funcionalidades necesarias.

Modificaciones Previas al Inicio del Desarrollo

El inicio del proyecto requirió realizar una serie de cambios en los proyectos existentes para facilitar la implementación de las funcionalidades solicitadas. A continuación, se detallan las principales modificaciones realizadas:

Optimización de Tiempos de Compilación

Inicialmente el equipo trabajó para reducir los tiempos de compilación para poder avanzar de un modo más rápido en el futuro. Para lograrlo, se modificó la configuración

de las librerías utilizadas para la compilación del proyecto e inicio del entorno de desarrollo, concretamente de Babel, la librería utilizada para transpilar la sintaxis JSX utilizada en React, y Webpack, utilizada para empaquetar la aplicación web. El resultado fue que los tiempos de compilación se redujeron drásticamente y se agilizó la experiencia de desarrollo.

```
PS C:\Users\Renzo\Desktop\Pastech\pastech-web\Plataforma-Web> npm run dev
> pag-web-loginfull@1.0.0 dev
> npx nodemon --exec "npm run build && node dist/index.js" --ignore dist/

[nodemon] 3.1.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `npm run build && node dist/index.js`

> pag-web-loginfull@1.0.0 build
> babel src --ignore src/Utils,src/public -d dist && copyfiles --up 1 src/**/*.json src/**/*.html src/**/*.css src/**/*.jpg src/**/*.png src/**/*.svg src/Utils/* dist && webpack

Successfully compiled 71 files with Babel (8743ms).
assets by status 18.7 MiB [compared for emit]
  asset admin.js 2.77 MiB [compared for emit] (name: admin)
  asset map.js 2.62 MiB [compared for emit] (name: map)
  asset boards.js 2.41 MiB [compared for emit] (name: boards)
  asset changePassword.js 2.33 MiB [compared for emit] (name: changePassword)
  asset cliente.js 2.09 MiB [compared for emit] (name: cliente)
  asset clientes.js 1.96 MiB [compared for emit] (name: clientes)
  asset register.js 1.15 MiB [compared for emit] (name: register)
  asset login.js 1.14 MiB [compared for emit] (name: login)
  asset newPass.js 1.14 MiB [compared for emit] (name: newPass)
  asset recuperar.js 1.14 MiB [compared for emit] (name: recuperar)
assets by status 6.44 MiB [emitted]
  asset pastures.js 3.66 MiB [emitted] (name: pastures)
  asset pasturometer.js 2.78 MiB [emitted] (name: pasturometer)
orphan modules 2.07 MiB [orphan] 1147 modules
runtime modules 13.1 KiB 64 modules
cacheable modules 3.44 MiB 606 modules
webpack 5.93.0 compiled successfully in 14059 ms
Servidor corriendo en puerto 3000
```

Figura 5: Salida de consola con tiempos de compilación de la Plataforma Web previos a las optimizaciones implementadas

```
PS C:\Users\Renzo\Desktop\Pastech\pastech-web\Plataforma-Web> npm run dev
> pag-web-loginfull@1.0.0 dev
> npx nodemon --exec "npm run build:babel && node dist/index.js" --ignore dist/

[nodemon] 3.1.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): src\**\*
[nodemon] watching extensions: js,jsx,json
[nodemon] starting `npm run build:babel && node dist/index.js`

> pag-web-loginfull@1.0.0 build:babel
> babel src --ignore src/utils,src/public -d dist

Successfully compiled 71 files with Babel (2290ms).
Servidor corriendo en puerto 3000
```

Figura 6: Salida de consola con tiempos de compilación de la Plataforma Web posteriores a las optimizaciones implementadas

Como se puede ver en las Figuras 2 y 3, la salida del proceso de compilación previo a las optimizaciones implementadas indica que tardó aproximadamente 23 segundos en iniciar el proyecto, mientras que luego de las optimizaciones el proceso completo tarda menos de 3 segundos, por lo que se redujo el tiempo de espera de compilación en un 87%.

Documentación de la API REST

El análisis de riesgos realizado antes de iniciar el proyecto había identificado la necesidad de documentación técnica como una posible problemática, lo que permitió ejecutar un plan de contingencia efectivo. La estrategia implementada consistió en coordinar reuniones con los desarrolladores previos, quienes brindaron explicaciones sobre cómo funcionaban los sistemas. Estas reuniones de transferencia de conocimiento fueron de gran ayuda para la elaboración de la documentación, y permitieron al equipo aclarar dudas y comprender mejor el estado actual de la plataforma.

Como medida adicional y siguiendo el análisis de riesgos previamente planteado, se decidió documentar todos los endpoints del servidor utilizando Swagger, una librería

para facilitar la documentación de APIs REST en base a la especificación OpenAPI. La documentación es accesible mediante el endpoint “/docs” y posee las medidas de seguridad adecuadas para evitar que usuarios no autorizados accedan a información. En el anexo se puede encontrar más información al respecto.

Reestructuración de los Endpoints REST

Para facilitar la implementación de las nuevas funcionalidades solicitadas por el demandante, que requerían la creación de nuevos endpoints REST y la modificación de otros existentes, y la elaboración de la documentación técnica, se decidió realizar una refactorización de la arquitectura del servidor, implementando una arquitectura más modular mediante el uso de routers de Express. Este cambio consistió en separar los endpoints en múltiples archivos, agrupándolos según su funcionalidad (por ejemplo, endpoints relacionados con "potreros", "calibraciones", "pasturómetro", etc.).

Esta refactorización facilitó el desarrollo del proyecto. Por un lado, hizo que el código sea más fácil de mantener, ya que ahora cada grupo de rutas está organizado en su propio archivo, lo que permite trabajar en una sección específica sin afectar otras partes. Además, la nueva estructura es más clara y fácil de leer, lo que ayuda no solo a los desarrolladores actuales, sino también a cualquiera que se sume al equipo en el futuro. Al organizar las rutas de esta forma, se simplificó la tarea de agregar nuevas funcionalidades.

En el anexo se incluye un diagrama actualizado de la nueva estructura del servidor, detallando cómo se organizaron los routers y proporcionando ejemplos de implementación.

Implementación de la Solución

Pasturómetro Electrónico

El pasturómetro, como se mencionó anteriormente, es un instrumento diseñado para medir la altura comprimida del pasto y en base a estos datos facilitar la toma de decisiones informadas. Este dispositivo realiza mediciones georreferenciadas que son enviadas a la aplicación móvil, desde donde se transmiten al servidor web para su almacenamiento en la base de datos.

Si bien más adelante se explicará en detalle el funcionamiento de las calibraciones y de la aplicación móvil, en esta sección se abordarán específicamente las mejoras implementadas en la plataforma web relacionadas con el Pasturómetro Electrónico.

El cliente solicitó incorporar una nueva funcionalidad en la plataforma web de Pastech Satelital. El objetivo principal de esta funcionalidad, que será detallada más adelante en los requerimientos específicos, consiste en ofrecer a los usuarios una vista que permita visualizar de manera centralizada todos los datos relevantes asociados a las mediciones realizadas con el pasturómetro.

El flujo pensado por el demandante para utilizar esta herramienta se puede apreciar en la Figura 7:

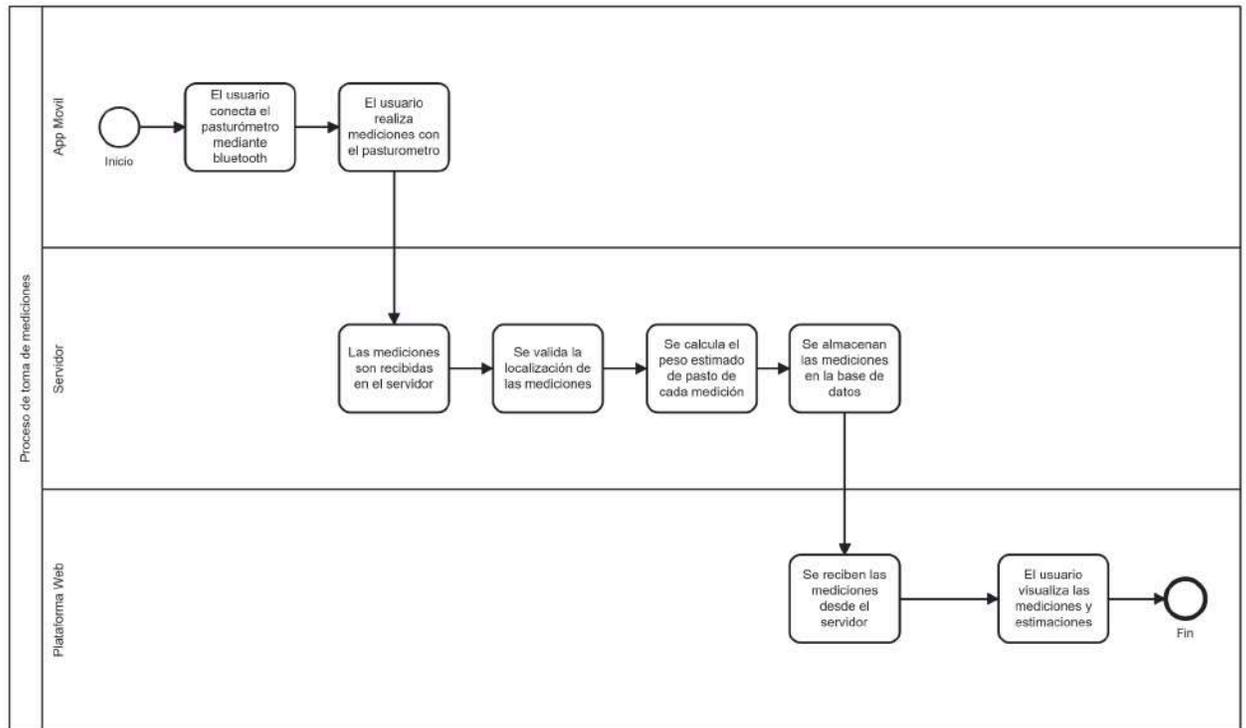


Figura 7: Diagrama de proceso de toma de mediciones

Requerimientos Funcionales

A continuación se detallan los requerimientos funcionales del sistema relacionados al Pasturómetro Electrónico:

- Visualización de mediciones en el mapa:
 - El sistema deberá permitir a los usuarios visualizar en un mapa la ubicación de las mediciones realizadas en un potrero específico en una fecha seleccionada.
- Stock de pasto por fecha:
 - El sistema deberá permitir a los usuarios visualizar el stock de pasto de un potrero en una fecha determinada y seleccionar fechas específicas para su análisis.
- Colores por sectores:
 - El sistema deberá permitir a los usuarios visualizar las mediciones realizadas en diferentes sectores representadas en el mapa con diferentes colores para facilitar la diferenciación.

- Resumen en tabla por sector:
 - El sistema deberá ofrecer a los usuarios una tabla que muestre, para cada sector, la cantidad de mediciones realizadas, su color de referencia en el mapa y el stock correspondiente.
- Agrupación de potreros por circuito:
 - El sistema deberá permitir a los usuarios visualizar en un mapa todos los potreros con mediciones tomadas con el pasturómetro, agrupados por circuito.
- Vista en tabla de potreros:
 - El sistema deberá ofrecer a los usuarios una tabla que muestre todos los potreros con mediciones tomadas con el pasturómetro. Para cada potrero, se deberá incluir:
 - Nombre del potrero
 - Cantidad de mediciones realizadas
 - Stock de pasto
 - Calibración utilizada
 - Circuito al que pertenece
 - Fecha de la última medición
- Gráfico de stock:
 - El sistema deberá ofrecer a los usuarios una visualización del stock de pasto mediante gráficos de barras que incluyan:
 - El stock de cada potrero
 - El stock agrupado por circuito de potreros
 - El stock agrupado por zona
- Selección de calibración:
 - El sistema deberá permitir a los usuarios seleccionar la calibración específica para un potrero determinado.
- Cambio de calibración retroactivo:

- El sistema deberá permitir a los usuarios cambiar la calibración utilizada para calcular el stock en base a mediciones realizadas en fechas previas, ajustando los cálculos retroactivamente según sea necesario.

Implementación

Para implementar la funcionalidad solicitada por el demandante, fue necesario realizar una serie de modificaciones en la plataforma web.

Modificaciones a la base de datos

En primer lugar, se modificó el esquema existente de la base de datos para incluir una nueva tabla destinada a almacenar los datos de las mediciones realizadas con el pasturómetro. Esta tabla fue integrada en la misma base de datos utilizada por la plataforma de Pastech Satelital y contiene los datos de cada medición: la altura medida, la ubicación geográfica (latitud y longitud), la fecha en la que se realizó, el potrero correspondiente y el cliente que la llevó a cabo.

Todos los cambios realizados en la base de datos se implementaron inicialmente en un ambiente de desarrollo para asegurar que la base de datos de producción, que contiene datos sensibles de los clientes, permaneciera intacta hasta que los cambios fueran probados y validados.

Adicionalmente, se creó una segunda tabla con una estructura idéntica a la anterior, pero con todos los campos configurados como opcionales y un atributo adicional de tipo lógico (verdadero o falso) llamado "aprobada". Esta tabla surgió como una medida de apoyo durante el desarrollo que ayudó a evitar la pérdida de datos provenientes de mediciones que no superaban las validaciones.

Esta segunda tabla resultó ser una herramienta muy útil ya que permitió identificar y corregir errores durante el proceso de toma y validación de mediciones, además de evitar la pérdida de datos importantes. La repetición de las mediciones puede implicar una inversión de tiempo significativa para los usuarios, por lo que esta estrategia

aseguró que la información crítica pudiera recuperarse, minimizando el impacto negativo en la experiencia del cliente.

Nuevos endpoints REST

Una vez que las mediciones fueron almacenadas en la base de datos, se procedió a implementar una serie de endpoints REST en el servidor Express. Estos endpoints son la forma en la que el cliente web puede acceder a la información almacenada, procesarla y mostrarla a los usuarios.

Cada endpoint fue diseñado para cumplir con los requerimientos funcionales del sistema, como la visualización de datos en tiempo real, la gestión de calibraciones específicas para diferentes fechas, y la exportación de datos en formatos compatibles con herramientas externas.

Los endpoints fueron estructurados bajo un esquema lógico y organizado en la ruta base /api/pasturómetro, lo que facilita su mantenimiento.

En la Tabla 2 se presentan los endpoints implementados, junto con sus especificaciones técnicas:

Endpoint	Entradas	Salida
GET /potreros	-	Lista de potreros que tienen mediciones asociadas.
GET /potreros/stock	-	Lista de potreros con su stock de pasto, en kg MS/ha.
GET /potreros/{id}/mediciones	id: Id del potrero.	Lista de mediciones del potrero, ordenadas por fecha.
PUT /potreros/{id}/calibracion	id: Id del potrero. Body: <ul style="list-style-type: none">- id_calibracion: id de la nueva calibración.	Actualizar la calibración utilizada para las mediciones de una fecha en un potrero.

	- fecha: fecha de las mediciones a ajustar.	
GET /circuitos/{id}	id: Id del circuito	Lista de mediciones del circuito.
GET /export/potreros	-	Datos de todos los potreros para exportar a Excel.
GET /export/mediciones	-	Datos de todas las mediciones para exportar a Excel.
GET /pasturometro/tasa?{fecha_inicio}	fecha_inicio: Fecha de inicio para filtrar las tasas de crecimiento.	Lista de potreros con sus tasas de crecimiento.

Tabla 2: Endpoints implementados para la ruta /api/pasturómetro

Estos endpoints son utilizados desde el cliente React para interactuar con el servidor y gestionar la información. A continuación, se describen las vistas implementadas en el cliente que habilitan a los usuarios a interactuar con las funcionalidades del Pasturómetro.

Vistas

En la Figura 8 se detallan las vistas implementadas para soportar la funcionalidad requerida.

Vista Integrada

Potrero	Calibración	Tipo de pasto	Cantidad de hectareas	Profundidad del suelo	Fertilidad del suelo	Circuito	Zona	Acciones
pot-1	Otoño	Fescue	3.96	medio	alto	Sierra	MdP	-
pot-2	-	Alfalfa	0.67	alto	alto	Sierra	MdP	Asignar calibración
lote-bal1	-	Ryegrass	13.43	medio	bajo	Loma	MdP	Asignar calibración
lote-bal2	-	Avena	5.96	medio	alto	Loma	MdP	Asignar calibración
laguna	-	Alfalfa	2.04	medio	alto	Laguna	MdP	Asignar calibración
lg	Primavera	Mezcla	0.46	medio	alto	Laguna	MdP	Ver Mediciones
casa	Otoño	Alfalfa	2.64	alto	alto	pinares	MdP	Ver Mediciones
vm1	Anual	Ryegrass	117.70	medio	medio	villa maria	Cordoba	-
vm2	Anual	Ryegrass	8.03	medio	medio	villa maria	Cordoba	-
vm3	Anual	Alfalfa	11.66	medio	medio	villa maria	Cordoba	-

Figura 8: Vista integrada del Pasturómetro

La vista principal, denominada “Integrada”, ofrece una tabla que muestra información sobre todos los potreros del usuario. Esta tabla incluye los siguientes detalles:

- Nombre: Identificación del potrero.
- Calibración actual: Calibración en uso, con la opción de cambiarla.
- Tipo de pasto: Clasificación del pasto predominante.
- Superficie: Tamaño del potrero en hectáreas.
- Profundidad del suelo: Características del suelo.
- Fertilidad del suelo: Indicador del nivel de fertilidad.
- Circuito: Circuito al que pertenece el potrero.
- Zona: Zona geográfica asociada.

Además, se habilitan las siguientes acciones:

- Asignar calibración: Disponible si el potrero no tiene calibración asignada.
- Ver mediciones: Habilitada para potreros con calibración asignada y mediciones registradas.

La vista también permite exportar los datos en formato Excel, con las siguientes opciones:

- Lista de potreros con su stock.
- Todas las mediciones realizadas.

Vista de Mediciones de cada Potrero

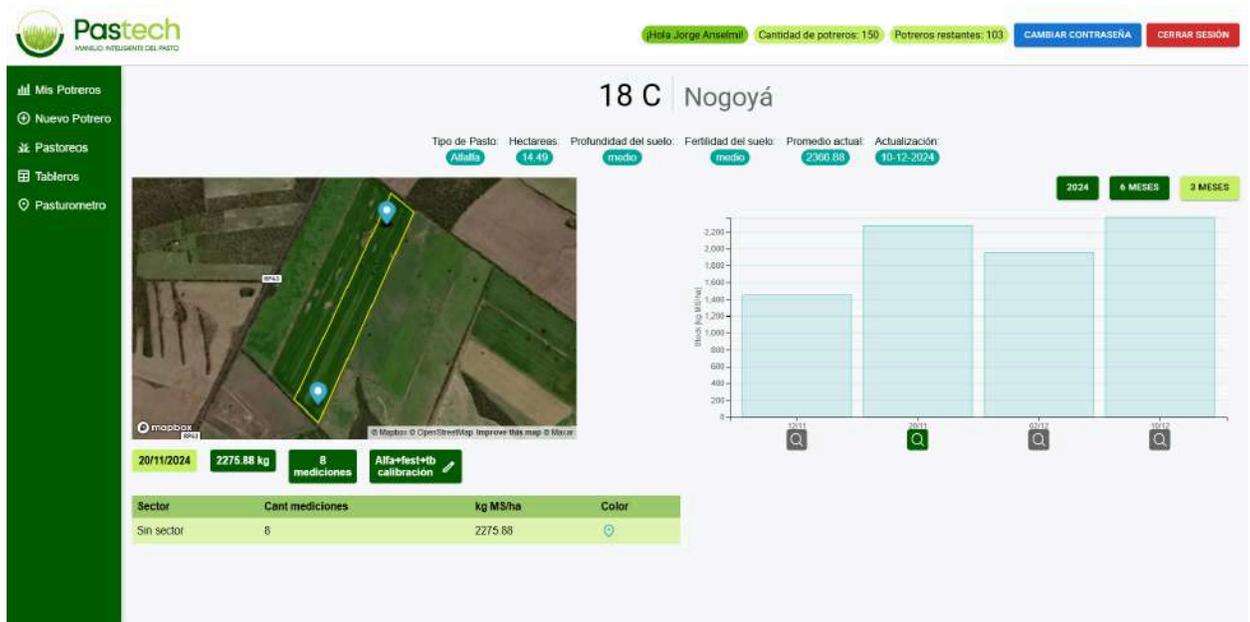


Figura 9: Vista de mediciones de cada potrero

Al seleccionar la opción “Ver Mediciones” desde la vista integrada, el usuario accede a una página detallada con los datos específicos del potrero seleccionado, incluyendo:

- Nombre del potrero.
- Superficie en hectáreas.
- Profundidad y fertilidad del suelo.
- Promedio de peso de pasto actual (kg de masa seca por hectárea).
- Fecha de la última medición.

Además, esta vista incluye:

- Gráfico de barras: Muestra el promedio de peso por fecha con mediciones asociadas, permitiendo seleccionar una fecha específica para visualizar sus mediciones en el mapa.
- Mapa satelital: Representa el potrero y las mediciones georreferenciadas de la fecha seleccionada.
- Tabla por sector: Contiene:
 - Identificador del sector.
 - Cantidad de mediciones asociadas.
 - Peso promedio (kg de masa seca por hectárea).
 - Color del marcador representado en el mapa.
 - Calibración asignada: Con la opción de cambiar la calibración para las mediciones de la fecha analizada.

Vista de Mapa

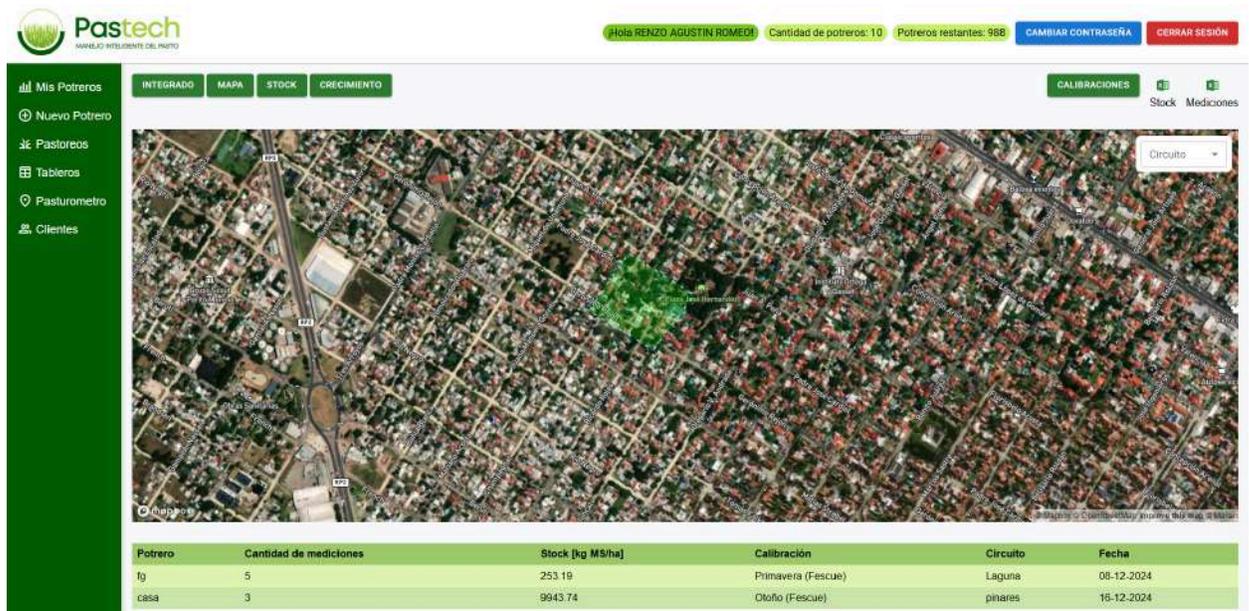


Figura 10: Vista de Mapa

La vista “Mapa” ofrece un mapa satelital que muestra todos los potreros. Estos se pueden agrupar por circuito. También se muestra una tabla que lista los potreros, indicando para cada uno:

- Nombre.

- Cantidad de mediciones.
- Stock actual (kg de masa seca por hectárea).
- Calibración asignada.
- Circuito asociado.
- Fecha de la última medición.

Vista de Stock



Figura 11: Vista de stock por potrero



Figura 12: Vista de stock por circuito



Figura 13: Vista de stock por zona

La vista “Stock” proporciona una representación visual del stock de pasto mediante gráficos de barras. Los gráficos muestran:

- El stock individual de cada potrero.
- El stock agrupado por circuitos y zonas.

Calibraciones

Uno de los principales módulos implementados dentro de la plataforma es el de calibraciones. Una calibración es una curva ajustada por regresión lineal que se utiliza para estimar el peso del pasto correspondiente a las mediciones realizadas con el pasturómetro.

Por defecto, el sistema cuenta con una serie de curvas de calibración definidas por el experto que cubren los casos más generales teniendo en cuenta el tipo de pasto y la época del año. Pero estas calibraciones estándar pueden no ser suficientes para usuarios que necesiten mayor precisión. Fundamentalmente, este módulo le otorga al usuario la posibilidad de realizar su propia curva de calibración basada en los datos de sus propios potreros, con el objetivo de obtener datos más precisos.

Implementación

Anteriormente el proceso para crear una calibración era muy complicado, constaba de muchos pasos y era poco intuitivo. Con el fin de mejorar la experiencia de usuario, el equipo decidió refactorizar el proceso de creación de calibraciones.

El proceso para crear una calibración se compone de distintas etapas que a continuación se detallan:

1. Crear una calibración desde la aplicación móvil indicando el nombre.
2. Tomar mediciones con el pasturómetro electrónico para la calibración creada y enviarlas al servidor.
3. Calcular el peso de cada medición realizada.
 - Por cada medición realizada con el pasturómetro, el usuario deberá cortar un trozo de pasto para luego medir su peso seco, y realizar una estimación del peso en kilogramos de masa seca por hectárea.
4. Cargar los pesos medidos en la plataforma, indicar el tipo de pasto y confirmar la creación de la calibración.

Una vez confirmada la creación de la calibración, el servidor se encarga de procesar los datos cargados por el usuario, y de almacenar la curva de calibración personalizada.

Cálculo de curva de calibración

Para determinar la mejor manera calcular los kilogramos de masa seca por hectárea a partir de la altura comprimida, se realizó una reunión con el ingeniero agrónomo Juan Insua. Según el experto existen tres maneras de relacionar la altura con el peso:

- Exponencial
- Cuadrática
- Lineal

La razón por la cual se optó por la regresión lineal es porque la curva de regresión lineal resulta la más precisa para relacionar ambas variables según el experto.

La regresión lineal es una técnica de análisis de datos que permite predecir el valor de datos desconocidos mediante el análisis de valores conocidos, modelando la función como una recta que minimice las distancias a todos los pares de valores conocidos. De esta forma, se puede utilizar la función lineal obtenida para estimar nuevos valores de la variable dependiente, partiendo de valores conocidos de la variable independiente.

La regresión lineal simple se define mediante la función lineal:

$$Y = \beta_0 \times X + \beta_1$$

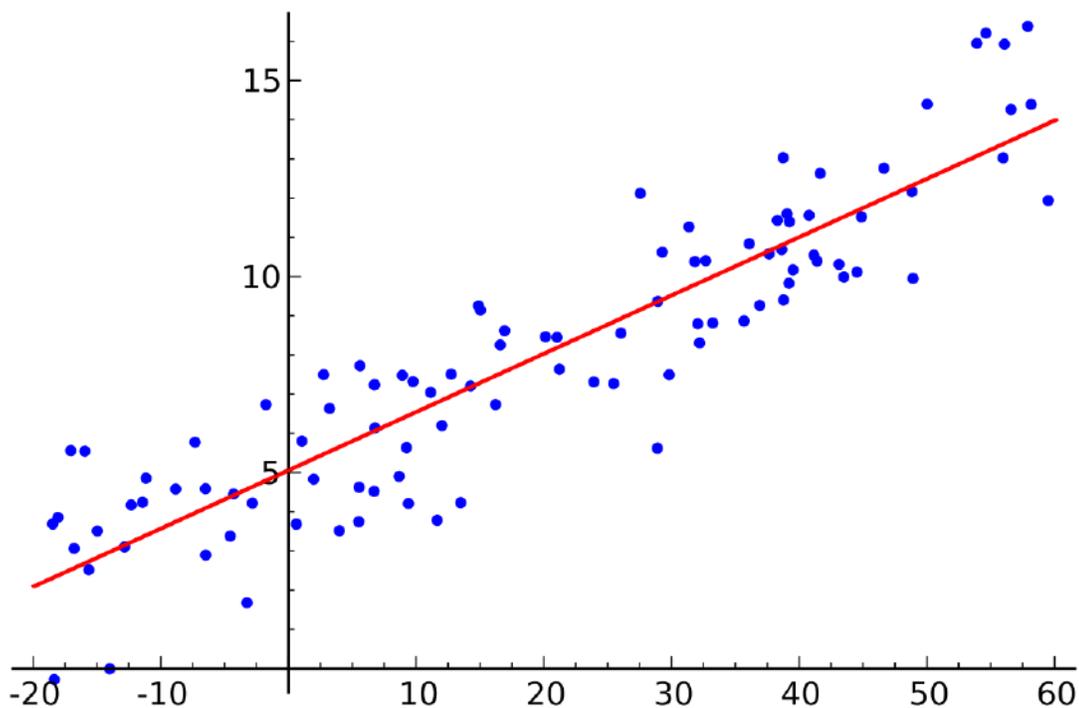


Figura 14: Ilustración de una regresión lineal sobre un conjunto de datos

β_0 representa la pendiente de regresión mientras que β_1 representa la ordenada al origen.

Para las calibraciones de cálculo del peso de pasto, las variables conocidas (X e Y) son la altura comprimida y el peso medido manualmente por el usuario. De esta manera, una vez obtenida la curva se puede utilizar para aproximar los kilogramos de masa seca por hectárea a partir de las mediciones realizadas. En otras palabras, le otorga al usuario la posibilidad de ajustar de manera más precisa la relación entre el la altura de su pasto y su peso.

Nuevos endpoints REST

Una vez almacenadas las mediciones relacionadas con las calibraciones en la base de datos, se procedió a implementar endpoints REST en el servidor Express para permitir el acceso a esta información desde el cliente web.

En la Tabla 3 se describen los endpoints creados, especificando las entradas que reciben y las salidas que generan:

Endpoint	Entrada	Salida
GET /api/calibraciones	-	Lista de calibraciones del usuario.
DELETE /api/calibraciones/{id_calibracion}	id_calibracion: id de la calibración a eliminar.	Mensaje de calibración eliminada.
GET /api/calibraciones/{id_calibracion}	id_calibracion: id de la calibración que se desea obtener.	Calibración solicitada
GET /api/calibraciones/{id_calibraciones}/mediciones	id_calibracion: id de la calibración para la cual se desea obtener las mediciones.	Mediciones de la calibración solicitada.
POST /api/calibraciones/{id_calibraciones}/mediciones	id_calibracion: id de la calibración para la cual se desea establecer las mediciones y el tipo de pasto.	Mensaje de mediciones y tipo de pasto establecidos.

	Body: <ul style="list-style-type: none">- mediciones []- especie	
--	---	--

Tabla 3: Endpoints implementados para la ruta /api/calibraciones

Vistas

En las Figuras 15 y 16 se muestran las vistas creadas en el cliente para permitir al usuario interactuar con las funcionalidades relacionadas a las calibraciones.

Vista principal

Calibracion	Tipo de Calibracion	Tipo de pasto	Acciones
test	Personalizada		Completar Calibración
test1	Personalizada		Completar Calibración
Calibracion de prueba	Personalizada	Alfalfa	-
Prueba 2010	Personalizada		-
Prueba10	Personalizada	Alfalfa	-
Prueba20	Personalizada		-
Anual	Estándar	Alfalfa	-
Otoño	Estándar	Fescue	-
Otoño	Estándar	Ryegrass	-
Primavera	Estándar	Fescue	-
Primavera	Estándar	Ryegrass	-

Figura 15: Vista principal calibraciones

La vista principal de calibraciones está disponible en la sección de Pasturómetro, accesible a través de la barra lateral del sistema. Dentro de esta sección, los usuarios pueden acceder a una barra de navegación que ofrece la opción de ver las calibraciones.

Esta vista presenta una tabla que proporciona al usuario un resumen completo de las calibraciones asociadas a su cuenta. La tabla incluye los siguientes detalles:

- Nombre de cada calibración: Para identificar rápidamente cada conjunto de datos de calibración.
- Tipo de calibración: Se clasifica como "Estándar" o "Personalizada".

- Calibraciones estándar: son las ofrecidas por la plataforma de forma predeterminada
- Personalizadas: son aquellas creadas por el usuario a partir de sus propias mediciones
- Tipo de pasto: Indica el tipo de pasto relacionado con cada calibración.
- Acciones: Contiene una opción denominada “Completar calibración” que redirige a una nueva vista para finalizar las calibraciones pendientes.

Vista completar calibración

The screenshot displays the 'completar calibración' view in the Pastech application. The top header shows the user 'Hola RENZO AGUSTIN ROMEO', 'Cantidad de potreros: 10', and 'Potreros restantes: 988'. The main content area features a table with the following data:

ID Medición	Fecha	Peso [kg MS/ha]
7	2/1/2025 16:24:47	0
10	2/1/2025 16:24:51	0
6	2/1/2025 16:24:45	0
9	2/1/2025 16:24:50	0
8	2/1/2025 16:24:48	0

Below the table, there is a 'Tipo de pasto:' dropdown menu set to 'Fascua' and a 'Guardar Pesos' button.

Figura 16: Vista para completar una calibración

Al seleccionar la opción "Completar Calibración", el usuario es redirigido a una vista donde puede finalizar el proceso de creación de la calibración completando el peso de cada medición asociada y el tipo de pasto. Esta vista proporciona al usuario una tabla con todas las mediciones asociadas a la calibración, mostrando los siguientes elementos:

- ID de las mediciones
- Fecha y hora
- Campo para completar el peso
- Campo para seleccionar el tipo de pasto

El botón de "Guardar Pesos" permite al usuario almacenar provisionalmente el estado de los pesos cargados, garantizando que los datos no se pierdan mientras se completan todas las mediciones. Esto permite que el proceso pueda ser realizado de

manera progresiva a lo largo del tiempo y que no sea necesario cargar todos los pesos en un mismo momento. Una vez que todas las mediciones están completadas y el tipo de pasto seleccionado, el botón cambia a "Finalizar Calibración". Al hacer clic en este botón, la calibración es finalizada y los datos se almacenan en la base de datos para su uso posterior.

Aplicación Móvil

La ejecución de este proyecto y su correcta integración con los módulos implementados requirió realizar modificaciones en la aplicación móvil. Este proceso implicó la incorporación de nuevas funcionalidades, la actualización de las vistas y la reestructuración de la base de datos para garantizar la compatibilidad con las funcionalidades implementadas en plataforma.

La aplicación fue desarrollada utilizando React Native y la plataforma Expo. React Native es una librería que permite utilizar React para desarrollar interfaces de usuario para múltiples plataformas, como Android, iOS y web, sin necesidad de escribir código nativo para cada una de ellas. Por otro lado, Expo es una plataforma de código abierto que facilita el desarrollo, la compilación y el despliegue de aplicaciones creadas con React Native.

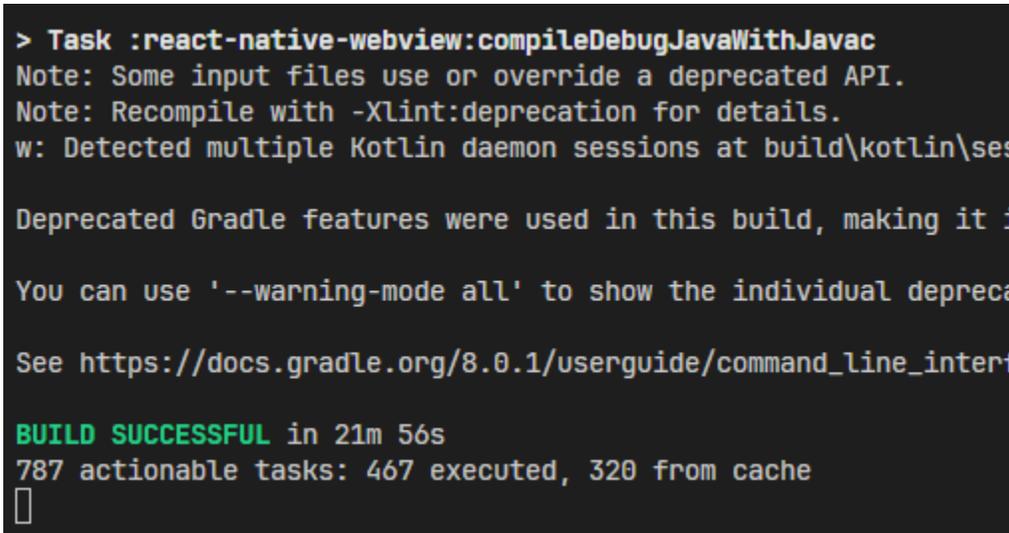
A su vez, usa una base de datos local para almacenar temporalmente los datos de las mediciones tomadas. Esta estrategia permite que los usuarios puedan tomar y guardar mediciones incluso cuando no tienen conexión. Cuando el dispositivo recupera la conexión, los datos locales se sincronizan con el servidor de la plataforma web.

Inicialmente, la aplicación móvil contaba con documentación sobre las APIs requeridas para comunicarse con el servidor web. Esta documentación sirvió como referencia durante el desarrollo de nuevos endpoints REST en el servidor express de la plataforma web que dieron soporte a las nuevas funcionalidades.

Problemas Iniciales

Inicialmente fue necesario determinar un flujo de desarrollo adecuado para minimizar los tiempos de espera a la hora de modificar la aplicación. Uno de los principales inconvenientes fue que el tiempo de inicio del entorno de desarrollo era muy elevado:

- Cada cambio en el código requería reconstruir el proyecto completo, un proceso que tomaba aproximadamente 10 minutos por iteración.
- Aunque se intentó utilizar un emulador para visualizar la aplicación en ejecución, esta opción resultó lenta e ineficiente. El emulador tardaba cerca de 20 minutos en iniciar, lo que generó una experiencia negativa para los desarrolladores y ralentizó el ritmo de trabajo.



```
> Task :react-native-webview:compileDebugJavaWithJavac
Note: Some input files use or override a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
w: Detected multiple Kotlin daemon sessions at build\kotlin\ses

Deprecated Gradle features were used in this build, making it i
You can use '--warning-mode all' to show the individual depreca
See https://docs.gradle.org/8.0.1/userguide/command_line_interf

BUILD SUCCESSFUL in 21m 56s
787 actionable tasks: 467 executed, 320 from cache
█
```

Figura 17: Salida de la consola con los tiempos de inicio del emulador

Como solución a estos inconvenientes se decidió conectar un teléfono móvil directamente a la computadora y usarlo como entorno de prueba en tiempo real.

- Con este método, solo fue necesario realizar la compilación inicial de la aplicación una vez.
- Los cambios en el código se reflejaban de forma inmediata en el dispositivo conectado, eliminando la necesidad de compilar todo el proyecto y mejorando la eficiencia del flujo de trabajo.

Mejoras Realizadas

Creación de Calibración

Se refactorizó por completo el proceso de creación de calibraciones para hacerlo más intuitivo y mejorar la experiencia del usuario.

En base a la retroalimentación recibida por los usuarios finales de la aplicación, el demandante determinó que el flujo para crear nuevas calibraciones era complejo y demasiado extenso, lo que generaba confusión. El proceso anterior incluía numerosos pasos, tales como navegar entre diferentes vistas, seleccionar opciones de manera redundante y realizar sincronizaciones manuales para finalizar una calibración.

A continuación se detallan los pasos en su totalidad:

1. Ir a la vista de calibración.
2. Presionar la opción “lista de calibraciones”.
3. Hacer click en el botón “+”.
4. Ingresar el nombre de la nueva calibración.
5. Presionar la opción “a partir de mediciones”.
6. Presionar botón “crear”.
7. Ir a la vista de calibraciones.
8. Seleccionar la opción creada en el desplegable.
9. Tomar las mediciones con el Pasturómetro.
10. Ir a la vista de calibración.
11. Presionar botón “enviar calibración”.
12. Seleccionar la calibración recién creada.
13. Presionar botón “volver”.
14. Ir a la vista del usuario.
15. Presionar “sincronizar”.

Tras una reunión con el referente funcional, se concluyó que era fundamental simplificar este procedimiento para mejorar la experiencia de usuario. A partir de este análisis, se diseñó e implementó un nuevo sistema de creación de calibraciones con un

flujo más ágil y directo. El nuevo proceso de creación de calibraciones reduce drásticamente la cantidad de pasos necesarios y organiza las acciones de manera lógica, permitiendo que los usuarios creen calibraciones de forma rápida y sin necesidad de navegar excesivamente entre vistas.

A continuación se detallan los pasos del nuevo proceso para crear calibraciones:

1. Ir a la vista de calibraciones.



Figura 18: Vista de mediciones en aplicación móvil

2. Presionar el botón “+”.

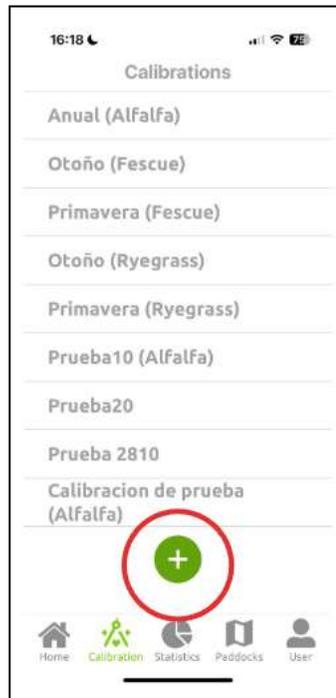


Figura 19: Vista de calibraciones en aplicación móvil

3. Ingresar el nombre de la nueva calibración y presionar el botón “crear calibración”.

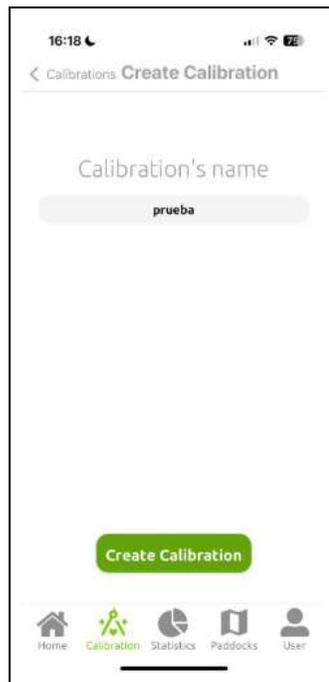


Figura 20: Vista para insertar nombre de la nueva calibración en aplicación móvil

4. Ingresar a la calibración mediante el icono “+” para la toma de mediciones con el Pasturómetro.



Figura 21: Vista de calibraciones en aplicación móvil

5. Tomar todas las mediciones deseadas y presionar el botón “enviar calibración” para finalizar.

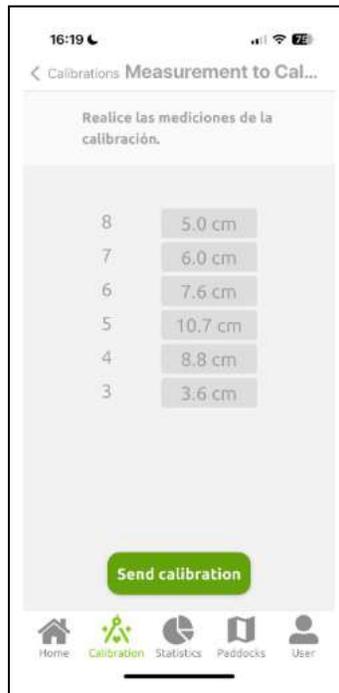


Figura 22: Vista de mediciones tomadas para crear la calibración

De esta manera, tras unos pocos pasos el usuario es capaz de crear una nueva calibración personalizada para poder completarla desde la plataforma web posteriormente.

Georeferenciación de las mediciones

Uno de los principales aspectos mencionado por los usuarios de la aplicación fue que algunas mediciones no se realizaban de forma correcta, debido que no se podía determinar la ubicación en forma precisa. Esto representaba una problemática grave ya que se perdían datos importantes y que son difíciles de recolectar, ya que requieren que el usuario se movilice hasta la localización del potrero, por lo que fue necesario refactorizar el método de obtención de la localización del dispositivo.

Inicialmente, la aplicación móvil utilizaba la biblioteca expo-location para obtener la ubicación del usuario en tiempo real al momento de registrar una medición con el pasturómetro y así asociar la altura medida con la ubicación en que fue tomada la medición. Este enfoque consistía en solicitar la ubicación directamente y asociarla a la medición en ese instante. Sin embargo, durante las pruebas y el uso en entornos

reales, se identificaron algunos problemas que afectan la funcionalidad y la experiencia del usuario de forma negativa.

El principal inconveniente era que el proceso de obtención de la ubicación a menudo resultaba lento o, en algunos casos, no se completaba. Esto hacía que la toma de la medición quedara bloqueada mientras esperaba la respuesta del servicio de localización, impidiendo que la toma de la medición se completara. Además, cuando la ubicación lograba ser capturada, su precisión era insuficiente para el nivel de detalle requerido, con desviaciones de hasta 600 metros, lo que hacía que los datos tomados no fueran tan útiles.

Para abordar estos problemas, se exploraron varias alternativas:

Inicialmente, se intentó cambiar los parámetros de precisión al solicitar la ubicación, ya que originalmente la librería estaba configurada para tomar la ubicación con una precisión baja. Sin embargo, esta solución no resolvió los problemas de lentitud ni mejoró la precisión de manera significativa.

Luego se intentó implementar otra estrategia que aprovechaba otra funcionalidad de expo-location, que consistía en capturar la ubicación en intervalos regulares y almacenarla en un estado global utilizando Redux, una biblioteca para gestionar el estado en aplicaciones React. Este estado global servía como referencia al momento de registrar una medición. A pesar de ser un avance, esta solución tampoco resultó efectiva, ya que las ubicaciones almacenadas no siempre estaban actualizadas, lo que significaba que continuaban los problemas de precisión.

Otra solución implementada consistió en utilizar una funcionalidad de expo-location que devolvía la última posición conocida del usuario. Si bien esta solución eliminaba el problema de la latencia al no requerir la obtención de la ubicación en tiempo real, la ubicación obtenida resultaba muy imprecisa, ya que no se actualizaba correctamente según el recorrido del usuario. Esto provocaba que algunas mediciones tuvieran una referencia geográfica inconsistente con el lugar donde realmente habían sido tomadas, utilizando en cambio una localización antigua del

usuario. Como consecuencia, las mediciones quedaban fuera del potrero correspondiente y no podían ser visualizadas correctamente en la plataforma web.

La solución definitiva, que permitió resolver todos los problemas identificados, fue el uso de otra funcionalidad de expo-location que permite establecer un callback que se activa cada vez que el servicio de ubicación detecta un cambio en la posición del usuario. Este enfoque permitió mantener un seguimiento constante y en tiempo real de la ubicación, almacenándola en Redux inmediatamente después de cada actualización. De esta manera, al tomar una medición, la aplicación siempre disponía de la última ubicación registrada, asegurando datos precisos y actualizados.

Esta implementación mejoró la experiencia del usuario y también permitió asegurar la calidad y confiabilidad de los datos recolectados.

Otro aspecto mencionado por los usuarios finales es que la aplicación debía permanecer encendida en primer plano para que las mediciones fueran tomadas correctamente, lo cual resultaba poco práctico. Para abordar este inconveniente, se actualizaron los permisos solicitados por la aplicación, habilitando el acceso a la ubicación en tiempo real y en segundo plano. Esto permitió que el sistema pudiera registrar las actualizaciones de ubicación incluso cuando la aplicación no estaba en primer plano.

Despliegue de la aplicación en Android e iOS

Inicialmente, la aplicación estaba compilada exclusivamente para dispositivos Android y no estaba publicada en la Play Store, ni existía una versión disponible para iOS. Sin embargo, el cliente solicitó que la aplicación estuviera disponible para su descarga tanto en la Play Store (Android) como en la App Store (iOS).

Para cumplir con este requisito, fue necesario tener en cuenta varias consideraciones:

Despliegue de la Aplicación en iOS

El despliegue de la aplicación en dispositivos iOS requirió la implementación de procesos específicos para cumplir con los lineamientos técnicos y regulatorios establecidos por Apple. A continuación, se detallan los aspectos más relevantes considerados durante esta etapa del proyecto:

Generalmente, para compilar y enviar una aplicación al portal de desarrollador de Apple, es necesario contar con una Mac, ya que el proceso de compilación para iOS requiere herramientas específicas de macOS. Sin embargo, este requisito fue evitado gracias al uso de EAS (Expo Application Services). EAS es un servicio ofrecido por Expo que permite compilar aplicaciones tanto para iOS como para Android de forma remota en la nube, lo que eliminó la necesidad de una Mac. Además, la herramienta permite la gestión automática de versiones de la aplicación, por lo que no fue necesario gestionarlas manualmente. A través de EAS, el equipo pudo compilar la aplicación y enviarla directamente al portal de desarrollo de Apple, facilitando el proceso de publicación sin depender de hardware específico. Esto simplificó la distribución de la aplicación en la plataforma iOS.

Para poder gestionar las compilaciones y publicaciones, así como para probar la aplicación en dispositivos reales, se necesitaba una cuenta de desarrollador de Apple, que fue proporcionada por el cliente, quien cubrió los costos asociados con la membresía anual de Apple Developer.

A pesar de contar con la cuenta de desarrollador, el equipo de trabajo encontró dificultades al intentar publicar la aplicación en la App Store. El equipo de revisión de Apple rechazó la publicación debido a varios problemas que se detallan a continuación:

- Guía 2.1 - Información Necesaria: Se solicitó un video demostrativo que mostrara la aplicación en uso en un dispositivo iOS físico, evidenciando la interacción con el hardware designado y documentando las funcionalidades relevantes, las solicitudes de permisos de usuario, y otros aspectos de la

aplicación. El equipo grabó y proporcionó el video requerido para cumplir con esta solicitud.

- Guía 2.5.4 - Rendimiento - Requisitos de Software: Apple observó que la aplicación había declarado soporte para la ubicación en segundo plano en el archivo de permisos, pero no encontró ninguna característica que justificara la necesidad de actualizaciones persistentes de la ubicación en tiempo real. Se resolvió aclarando cómo y cuándo la aplicación utiliza la ubicación, y para qué propósito.
- Guía 5.1.1 - Legal - Privacidad - Recolección y Almacenamiento de Datos: La solicitud de permisos para acceder a la ubicación y al Bluetooth no proporcionaba una explicación suficientemente clara sobre el uso de estos datos. El equipo actualizó el archivo de permisos, añadiendo descripciones detalladas de los permisos solicitados, explicando cómo y por qué la aplicación necesita acceso a la ubicación y al Bluetooth, con ejemplos específicos sobre el uso de estos datos.

Una vez resueltos estos problemas siguiendo las indicaciones del equipo de revisión de Apple, la aplicación fue enviada nuevamente para su revisión y publicación en la App Store.

Otros requerimientos solicitados fueron:

- Acuerdo de Privacidad: Se solicitó al cliente un documento detallado que describiera cómo se gestionan los datos personales de los usuarios, y se presentó junto con la solicitud de publicación.
- Capturas de Pantalla: Se tomaron imágenes representativas de la interfaz de la aplicación para su inclusión en la tienda.

Para la validación de la aplicación en dispositivos iOS, se utilizó TestFlight, la herramienta oficial de Apple para la distribución de versiones beta. Se definieron grupos de prueba conformados por usuarios con diferentes niveles de experiencia técnica, permitiendo que la aplicación fuera evaluada por diferentes tipos de usuarios.

Despliegue de la aplicación en Play Store

El proceso de compilación de la aplicación para Play Store resultó ser más simple en comparación con el de iOS, debido a que los requisitos de la tienda son menos estrictos. En primer lugar, no es necesario pagar una membresía anual como en el caso de Apple. En cambio, basta con una compra única para obtener una cuenta de desarrollador en Google. Este costo fue cubierto por el cliente, quien proporcionó la cuenta necesaria para realizar las publicaciones.

Para compilar la aplicación en el formato requerido por Play Store (AAB), se utilizó nuevamente EAS. Sin embargo, se presentó un inconveniente cuando la aplicación fue inicialmente publicada utilizando la cuenta personal de Expo de uno de los integrantes del equipo. Al cambiar la cuenta de Expo a una dedicada exclusivamente a Pastech, surgió un conflicto con la firma de la aplicación. Al intentar publicar una nueva versión, la consola de Play Store indicaba que se utilizó una firma diferente para compilarla. Para resolver esto, se extrajo la keystore de la cuenta personal en formato de archivo y se configuró el proyecto para utilizar esta keystore local en todas las compilaciones, lo que permitió hacer el proceso más portable.

Otro inconveniente fue la falta de un acuerdo de privacidad en la configuración de la aplicación. Este problema se resolvió añadiendo un vínculo al acuerdo de privacidad desde la consola de Play Store.

Después de realizar estas modificaciones, la aplicación fue finalmente publicada en Play Store sin más inconvenientes.

Validación del producto

Durante el desarrollo de la aplicación, se implementó un proceso de validación para garantizar que todas las funcionalidades se implementaran de manera efectiva y cumplieran con los requisitos definidos. Este proceso de validación se dividió en varias fases, comenzando con pruebas individuales de cada nueva funcionalidad a medida que se iban creando. Esto permitió identificar rápidamente posibles errores en las funcionalidades recién implementadas antes de integrarlas al sistema completo.

Una vez realizadas las pruebas individuales, se procedió a la fase de pruebas de integración, cuyo objetivo era garantizar que los nuevos requerimientos funcionaran correctamente en los escenarios de uso definidos y que las funcionalidades se integraran sin problemas con las existentes.

El proceso de validación se estructuró en tres niveles de pruebas:

- **Pruebas realizadas por el equipo de desarrollo:** En esta fase inicial, el equipo de desarrollo ejecutó pruebas para verificar que las funcionalidades implementadas funcionaran correctamente.
- **Pruebas realizadas por usuarios con perfil técnico:** El siguiente nivel de pruebas estuvo a cargo de usuarios internos de Pastech con experiencia en informática y desarrollo de software. Este grupo de usuarios realizó pruebas más detalladas desde una perspectiva técnica, lo que permitió identificar posibles errores y escenarios que podrían haber pasado desapercibidos para los desarrolladores.
- **Pruebas realizadas por usuarios con perfil no técnico:** El último nivel de pruebas involucró a usuarios finales de Pastech sin formación técnica. Este grupo fue clave para simular situaciones atípicas o imprevistas, como problemas de usabilidad o dificultades que podrían encontrar los usuarios sin conocimientos especializados. Estas pruebas fueron útiles para garantizar que la aplicación fuera intuitiva y fácil de usar, y para identificar cualquier área de mejora en términos de interfaz y experiencia de usuario.

Esta estrategia de pruebas en tres niveles proporcionó una validación completa del sistema, asegurando que la aplicación no solo cumpliera con los requisitos técnicos, sino que también fuera accesible y funcional para los usuarios finales, independientemente de su nivel de conocimiento técnico.

Herramientas utilizadas

Durante el proceso de testing, se utilizaron varias herramientas que facilitaron la validación de la aplicación y la resolución de problemas.

Postman es una herramienta utilizada para probar y verificar APIs REST. Durante el desarrollo de la aplicación, se utilizó Postman para realizar pruebas de las APIs implementadas, asegurando que todas las solicitudes HTTP, ya sea de tipo GET, POST, PUT o DELETE, funcionaran correctamente y devolvieran las respuestas esperadas. Con Postman, el equipo de desarrollo pudo enviar solicitudes al servidor y revisar las respuestas en tiempo real, facilitando la detección de posibles errores en el backend.

Por otro lado, DBeaver es una herramienta de administración de bases de datos que fue esencial para gestionar las interacciones con la base de datos de la aplicación. Utilizando esta herramienta, el equipo pudo conectarse a la base de datos, ejecutar consultas y realizar verificaciones directas de los datos almacenados. Además, DBeaver permitió insertar datos de prueba de manera sencilla, lo que facilitó la simulación de escenarios reales y garantizó que las implementaciones funcionaran correctamente. Esta funcionalidad fue importante para verificar que la integración entre la base de datos y la aplicación estuviera correcta y que las funcionalidades de la aplicación, como el almacenamiento y recuperación de datos, fueran efectivas.

Problemas encontrados y soluciones implementadas

Errores en el Registro de Mediciones

Durante el desarrollo de la aplicación, se detectó una falla en la funcionalidad de registro de mediciones. Mientras que las mediciones realizadas por un miembro del equipo eran almacenadas en la base de datos luego de llegar al servidor, las mediciones realizadas por otro usuario no eran registradas.

Para abordar esta problemática, se implementó una nueva tabla en la base de datos denominada "Pasturometro_Mediciones_Todas", diseñada para almacenar todas las mediciones que llegaban al servidor, ya fueran aceptadas o rechazadas. Esta tabla incluye detalles como la fecha en que la medición fue realizada, la fecha en que fue registrada en el log, y un campo booleano llamado "aceptada" para indicar si la medición cumplía con los criterios necesarios para ser procesada.

Las mediciones aceptadas se registran en su tabla correspondiente (Pasturometro_Mediciones o Mediciones_Calibraciones, dependiendo de su naturaleza), mientras que las rechazadas permanecen en el log para su análisis. Además, esta tabla permitió corregir un error que hacía que las mediciones utilizadas para crear una calibración personalizada se duplicarían en tablas erróneas.

Gracias a esta estrategia, se pudieron identificar y analizar mediciones rechazadas, descubriendo que el usuario afectado estaba realizando mediciones fuera de los límites de su potrero asignado.

Entre las ventajas de esta solución destacan:

- Ninguna medición se pierde, ya que todas se almacenan en el log, lo que permite recuperarlas si es necesario.
- Facilita la experiencia de desarrollo, al proporcionar una herramienta para identificar errores y analizar las causas de fallos en las mediciones.

Valores Negativos en el Cálculo de Peso del Pasto

Luego de que un usuario realizara una serie de mediciones se detectó un problema crítico: los valores promedio de peso del pasto en ciertos potreros resultaban negativos, lo cual es inconsistente con la realidad esperada.

123 id_calibracion	A-Z nombre	123 id_especie	123 a	123 b
1	Anual	2	181,38	-28,854
3	Otoño	1	362,75	-766,43
4	Primavera	1	237,59	-568,78
5	Otoño	3	290,54	-623,12
6	Primavera	3	228,95	-808,26

Figura 23: Vista de la tabla “Calibraciones” en la base de datos

Para abordar esta situación, se inició una investigación para identificar las posibles causas. Se evaluaron tres factores principales:

- Curva de calibración incorrecta: Se verificó que el cliente utilizó una de las curvas estándar proporcionadas por el sistema, descartando este como el origen del problema.
- Valores absurdos en las mediciones: Se detectó que las alturas registradas en la base de datos para las mediciones eran extremadamente bajas, en su mayoría entre 0 y 1 cm, lo que no refleja condiciones realistas en un potrero.
- Errores en los cálculos matemáticos: Se realizó una revisión de los cálculos asociados al peso y se corroboró que los valores obtenidos eran correctos para las mediciones ingresadas, confirmando que no existía un error en las fórmulas implementadas.

Al detectar que este problema era causado por mediciones con valores imposiblemente bajos, se implementaron validaciones para evitar que ingresaran al sistema mediciones con valores demasiado pequeños que no pueden ser utilizadas para realizar estimaciones de stock.

Ubicación y Asociación de Mediciones con Potreros

Un cliente reportó que había realizado 180 mediciones con el pasturómetro, las cuales ingresaron correctamente a la tabla de logs, pero ninguna fue registrada en la tabla Pasturometro_Mediciones. Tras un análisis, se descubrió que el problema radicaba en la obtención del potrero asociado a cada medición, es decir, en qué potrero definido por el cliente había sido realizada la medición. El sistema devolvía NULL como resultado debido a que las ubicaciones registradas correspondían a la ciudad de Maciá, Entre Ríos, mientras que los potreros del cliente estaban ubicados a varios kilómetros de distancia.



Figura 24: Imagen representativa de la distancia entre Maciá y el potrero

Para identificar la causa del problema, se ejecutó una consulta SQL que ordenaba los potreros del cliente según la distancia al punto de una de las mediciones almacenadas en la tabla de logs. La consulta confirmó que no existían potreros cercanos a las ubicaciones registradas, lo que indicaba que las mediciones no estaban siendo asociadas correctamente debido a datos de localización imprecisos.

```
SELECT
  p.id_potrero,
  p.nombre,
  p.geojson,
  p.metros_cuadrados,
  p.profundidad_del_suelo,
  p.fertilidad_del_suelo,
  ST_Distance(
    ST_SRID(ST_GeomFromGeoJSON(p.geojson), 4326),
    ST_SRID(ST_GeomFromText('POINT(-59.4068002 -32.1656823)'), 4326)
  ) AS distance
FROM
  Potrero p
WHERE id_cliente = 109
ORDER BY
  distance ASC;
```

El análisis reveló que la función de la biblioteca expo-location en la aplicación móvil, no obtenía la ubicación más reciente del usuario en tiempo real, sino que devolvía la última ubicación conocida del dispositivo, sin verificar su antigüedad. Esto ocasionaba que mediciones tomadas lejos de los potreros fueran consideradas válidas, pero no se asociaran a un potrero específico.

Para solucionar esta problemática se refactorizó el módulo de obtención de la localización del usuario desde la aplicación móvil como se mencionó previamente.

Mediciones no asignadas debido a incompatibilidad en coordenadas

Se detectó que las mediciones realizadas por un usuario de la aplicación ingresaban correctamente a la tabla Pasturometro_Mediciones_Todas pero no eran procesadas en la tabla Pasturometro_Mediciones. Esto sucedía porque la consulta encargada de verificar en qué potrero caía cada medición fallaba al trabajar con potreros cuyo perímetro estaba definido en coordenadas tridimensionales. En este caso, el potrero del usuario estaba configurado con coordenadas en 3D, donde la tercera dimensión tenía un valor constante de 0, mientras que el resto de potreros de los usuarios estaba configurado para usar coordenadas 2D. Se cree que esta inconsistencia se debe a que anteriormente se utilizaban coordenadas 3D para definir el perímetro de los potreros y que luego se cambió la implementación para utilizar coordenadas 2D, pero el potrero de este usuario, que fue uno de los primeros en utilizar la aplicación, mantuvo la definición previa.

Para resolver este problema, se modificó la llamada a la función ST_GeomFromGeoJSON en la consulta SQL, agregando un parámetro adicional que indica que las dimensiones de las coordenadas deben ser ignoradas a partir de la segunda. De este modo, la consulta puede procesar correctamente los potreros definidos en 3D.

Adicionalmente, se realizó un pasaje manual de las mediciones afectadas a la tabla Pasturometro_Mediciones, calculando nuevamente los pesos para garantizar la consistencia de los datos.

Memorias del Proyecto

Cumplimiento de objetivos

Se lograron cumplir con éxito tanto el objetivo general como los objetivos específicos planteados inicialmente, integrando los subsistemas existentes en una única plataforma y permitiendo a los usuarios gestionar de manera eficiente sus pasturas. A continuación, se detalla cómo cada uno de los objetivos fue alcanzado:

Objetivo General: Integración de subsistemas

El principal desafío consistía en combinar los subsistemas independientes de Pastech Satelital y Pasturómetro Electrónico en una única plataforma. Esta integración eliminó la necesidad de que los usuarios alternaran entre plataformas, lo que permite un acceso centralizado a toda la información necesaria para gestionar las pasturas.

Los aspectos considerados para alcanzar una gestión eficiente de las pasturas de los usuarios son los siguientes:

Mejora en la experiencia de usuario

La experiencia de usuario a la hora de interactuar con la aplicación móvil fue mejorada drásticamente, especialmente a la hora de crear nuevas calibraciones. Previamente el proceso para crear nuevas calibraciones constaba de 15 pasos en los que el usuario debía navegar entre varias vistas, lo que resultaba innecesariamente complejo y poco claro. Este proceso fue simplificado reduciendo la cantidad de pasos necesarios en un 66% a tan solo 5 pasos repartidos entre 3 vistas.

Anteriormente, a la hora de tomar mediciones era usual que la aplicación demorara varios segundos entre que el usuario presionaba el botón en el Pasturómetro y la medición aparecía en la aplicación, o peor aún, no se completaba y los datos de esa medición se perdían. Esto se traduce en tiempos de respuesta altos, del orden de 8~10 segundos, o en pérdida de información. Luego de las mejoras realizadas, se redujo el tiempo de respuesta de la aplicación en un ~90% (entre 0,5 y 1,5 segundos) y

se evitó la pérdida de mediciones tomadas por los usuarios. Esto último es de especial importancia ya que la toma de datos con el Pasturómetro requiere que el usuario vaya al campo y tome las mediciones en el lugar deseado, por lo que asegurar que los datos se tomen correctamente es fundamental.

A su vez, gracias a la retroalimentación de los usuarios finales, se incorporó una mejora importante en la experiencia de uso: la posibilidad de tomar mediciones incluso con la pantalla del teléfono apagada y la aplicación en segundo plano. Originalmente era necesario que la pantalla del teléfono permaneciera encendida y con la aplicación en primer plano para poder tomar mediciones, lo que resultaba poco práctico para los usuarios finales. En el producto final entregado, la aplicación puede permanecer en segundo plano y no es necesario que la pantalla esté encendida para poder tomar mediciones. De este modo, se solucionó una preocupación general expresada por los usuarios.

Optimización del acceso a la información

En el producto final entregado, el acceso a la información resulta mucho más práctico. Previamente era necesario que el usuario interactúe con dos sistemas por separado para poder acceder a toda la información que la plataforma tiene para ofrecer, por lo que este enfoque no era óptimo. Ahora el usuario puede acceder a toda esta información desde una única plataforma, lo que representa una clara mejora respecto al estado anterior.

Nuevas funcionalidades que facilitan la toma de decisiones basadas en datos precisos

El producto final ofrece nuevas funcionalidades que permiten tomar mediciones más precisas y mostrar la información de una forma clara y concisa, permitiendo tomar decisiones más fácilmente.

Por un lado, el usuario ahora tiene la opción de crear calibraciones personalizadas que permiten mejorar la precisión de las mediciones tomadas. De este modo, en lugar de usar las calibraciones predeterminadas ofrecidas por la plataforma el usuario puede crear calibraciones en base a datos de cada uno de sus potreros, por lo

que la estimación del stock se realiza con datos del mismo potrero. Gracias a esto, las estimaciones calculadas resultan mucho más precisas.

Además, se mejoró drásticamente la precisión en la ubicación de las mediciones tomadas. En base a reportes de los usuarios de que algunas de las mediciones tomadas no aparecían en la plataforma, se descubrió que había imprecisiones de hasta 600 metros en la ubicación obtenida por la aplicación. En el producto final entregado se resolvió esta problemática y ahora las mayores imprecisiones son del orden de ~15 metros, lo que se traduce en una mejora de precisión de ~98% en los casos de mayor imprecisión.

Por otro lado, el producto final ofrece una nueva forma de visualizar las estimaciones realizadas con datos provenientes del pasturómetro. Ahora se ofrece la posibilidad de visualizar, en forma de gráficos de barras, el stock de pasto tanto para potreros particulares como para circuitos y zonas, lo que permite tomar decisiones más fácilmente respecto a cómo mover a los animales entre los diferentes potreros según el stock que tienen disponible.

Objetivos Específicos

Gestión de Potreros

Se implementaron herramientas que permiten a los usuarios gestionar potreros de forma sencilla, incluyendo la creación, edición y eliminación. La interfaz visual, apoyada en mapas interactivos, facilita la identificación y manejo de los potreros en la plataforma.

Visualización de Mediciones Georreferenciadas

La integración de las mediciones tomadas con el Pasturómetro Electrónico se logró exitosamente. Los usuarios pueden seleccionar una fecha para visualizar las mediciones de ese día, las cuales se presentan en un mapa con georreferenciación precisa. Además, se incluyeron gráficos de barras que muestran el stock por fecha, tablas detalladas para distinguir las mediciones por sector y herramientas para

visualizar la calibración utilizada en cada caso. También se añadió la opción de cambiar la calibración para recalcular el stock de las mediciones, brindando mayor flexibilidad y control al usuario.

Disponibilidad de Pasto por Grupos

Se implementaron gráficos que muestran la disponibilidad de pasto por potrero, circuito y zona. Estos gráficos permiten a los usuarios tomar decisiones basadas en datos al momento de planificar el manejo de los recursos.

Consulta de Disponibilidad Estimada mediante Imágenes Satelitales

Se implementaron gráficos de stock por fecha y la visualización del perímetro del potrero en un mapa, junto con los valores del índice NDVI para la superficie del potrero. Estas herramientas permiten a los usuarios complementar las mediciones del Pasturómetro Electrónico con información satelital, para tener más información a la hora de tomar mediciones.

Gestión de Circuitos de Potreros

La creación y gestión de circuitos de potreros se habilitó en la plataforma, permitiendo a los usuarios organizar los potreros de acuerdo con sus prácticas productivas.

Gestión de Calibraciones

Se creó una interfaz específica para la gestión de calibraciones basadas en mediciones del Pasturómetro Electrónico. Los usuarios ahora pueden crear sus propias calibraciones personalizadas para tener datos más precisos a la hora de tomar mediciones con el Pasturómetro Electrónico.

Adaptación de la Aplicación Móvil de Pasturómetro Electrónico

La aplicación móvil fue modificada para soportar las nuevas funcionalidades de la plataforma unificada. Se realizaron ajustes en la interfaz de usuario para hacerla más intuitiva y fácil de usar. Además, se realizaron refactorizaciones para garantizar el

correcto funcionamiento de todas las funcionalidades del sistema, y que las mediciones fueran tomadas en forma correcta.

Aspecto Crítico para el Éxito del Proyecto

Trabajar con un sistema en producción fue un aspecto fundamental del proyecto que influyó las decisiones tomadas por el equipo desde el comienzo, ya que cualquier modificación podía afectar directamente a los usuarios finales y a las operaciones del sistema en tiempo real. En este contexto, fue necesario abordar cada tarea con especial cautela, priorizando preservar la experiencia de usuario, la estabilidad, la seguridad y la integridad de los datos.

Para evitar que las modificaciones afectaran directamente al sistema en funcionamiento, se decidió trabajar con un entorno de desarrollo. Esta elección fue fundamental ya que definió la metodología de trabajo: cada cambio fue implementado y probado primero en este entorno antes de ser llevado a producción. El equipo considera que esta práctica fue un gran acierto, ya que permitió incorporar las nuevas funcionalidades y modificaciones sin comprometer la estabilidad del sistema en uso.

Una ventaja importante de tratarse de un sistema en producción fue que ya existía una base de usuarios activa que aportó retroalimentación tanto sobre el estado previo del sistema como sobre los cambios introducidos por el equipo. Esta retroalimentación constante fue fundamental para asegurar un producto final satisfactorio, tanto para los usuarios como para el demandante.

Sin embargo, hubo una situación particular a tener en cuenta. Al aplicar cambios que requerían modificaciones en la estructura de la base de datos de producción, la aplicación debía entrar en un breve período de mantenimiento. Durante ese lapso, se desplegaba el nuevo código junto con las actualizaciones en la base de datos, lo que evitaba inconsistencias entre ambos componentes. Aunque esta estrategia fue acordada con el demandante, en retrospectiva el equipo considera que no fue la mejor solución, ya que implicaba que la aplicación quedara temporalmente fuera de servicio.

Para mitigar este aspecto, hubiera sido acertado evaluar alternativas con cero tiempo de inactividad para mejorar la experiencia del usuario.

Etapa de Análisis

La etapa de análisis fue fundamental para sentar las bases del proyecto y garantizar una transición exitosa hacia la nueva plataforma unificada. Durante esta etapa, se realizó un estudio de los sistemas existentes para comprender su alcance y definir los requerimientos clave de la solución propuesta. Sin embargo, en retrospectiva hubo desaciertos y desviaciones de la etapa de análisis inicial con respecto a lo acontecido durante el desarrollo del proyecto.

El análisis de riesgos inicial resultó muy útil para anticipar posibles problemas y definir estrategias efectivas para abordarlos. Sin embargo, también presentó ciertas limitaciones: no contempló la aparición de nuevos riesgos que terminaron siendo decisivos durante el desarrollo, y subestimó otros que tuvieron un alto impacto. Del mismo modo, el análisis FODA inicial no logró identificar completamente todas las debilidades que surgirían a lo largo del proyecto.

Uno de los aspectos mencionados como clave para el éxito del proyecto fue el hecho de trabajar sobre un sistema ya en producción. Esta condición representó un riesgo de alto impacto durante el desarrollo, aunque no fue identificado como tal en el análisis inicial. La estrategia de mitigación adoptada por el equipo, que consistió en utilizar un entorno de desarrollo, resultó ser un gran acierto ya que permitió reducir significativamente la probabilidad de pérdida de datos e inconsistencias y de generar un impacto negativo en la experiencia del usuario. Además, este factor no fue contemplado inicialmente como una debilidad en el análisis FODA, cuando en realidad sí lo es, ya que trabajar con un sistema en producción impone precondiciones y restricciones críticas al momento de implementar nuevas funcionalidades o realizar modificaciones.

Un desacierto dentro del análisis de riesgos fue el de menospreciar el riesgo “diferencias entre las expectativas del demandante y el producto entregado”.

Inicialmente este fue considerado como un riesgo de bajo impacto, por lo que no se hizo un plan de mitigación para el mismo, pero en la práctica este fue un aspecto sumamente importante para garantizar el éxito del proyecto, ya que la no satisfacción del demandante con el producto entregado impactaría de manera crítica en el éxito del proyecto dado que hubiera extendido los plazos de entrega a causa del retrabajo. Si bien no había un plan de mitigación predefinido para este riesgo, la metodología de trabajo seleccionada fue un acierto ya que permitió mitigarlo proporcionando una validación continua mediante las reuniones semanales con el demandante.

Otro riesgo que apareció durante el desarrollo del proyecto y no fue tenido en cuenta en la etapa de análisis inicial, fue el de trabajar con código ajeno. En la práctica, este fue un aspecto crítico y de alto impacto en el desarrollo del proyecto, e incluso llevó al equipo a considerar empezar de cero y reemplazar el código existente por completo. Aunque en la práctica esto no ocurrió, la decisión tomada por el equipo de navegar a ciegas fue un acierto dado que minimizó la dependencia de actores externos, conservó la lógica ya implementada, y evitó el costo elevado que suponía una reescritura completa.

A partir de estas reflexiones el equipo toma como principal aprendizaje que tanto el análisis de riesgos como el análisis FODA son herramientas dinámicas, y no estáticas. Los riesgos, fortalezas, oportunidades, amenazas y debilidades planteados inicialmente pueden sufrir cambios durante el desarrollo del proyecto, y esto no debe ser menospreciado.

Otro aprendizaje importante del equipo fue que es imprescindible no minimizar el posible impacto de los riesgos identificados durante el análisis, ya que esto podría causar que no haya un plan de contingencia preparado para afrontar el impacto que este podría generar.

Validación del Producto

La validación del producto final entregado es fundamental ya que determina si el proyecto fue un éxito, si cumplió con las expectativas del demandante, y si los usuarios

finales perciben una mejora real en la herramienta. Para llevar a cabo esta validación se decidió consultar tanto al demandante de forma directa como a los usuarios finales, de forma indirecta a través del propio demandante.

En un diálogo posterior a la entrega, se formularon una serie de preguntas clave orientadas a validar el producto entregado.

Desde el punto de vista de los usuarios finales, se identificaron mejoras concretas en la experiencia de uso. La existencia de una única plataforma unificada facilita considerablemente el trabajo diario. La toma de mediciones es ahora significativamente más rápida y precisa, gracias a un proceso de calibración que pasó de requerir 15 pasos a solo 5, siendo ahora más simple e intuitivo. La aplicación móvil fue destacada por su facilidad de uso, y los gráficos de stock de pasto se señalaron como una herramienta fundamental para la toma de decisiones, ya que permiten visualizar claramente en qué zonas conviene hacer pastorear a los animales.

Además, el demandante agregó que en base al aporte del proyecto se corrigieron errores presentes en versiones anteriores y los datos recolectados están ahora estructurados de manera tal que permiten establecer patrones y tendencias, como zonas productivas, tipos de pasto, y correlaciones con mediciones satelitales.

Desde el punto de vista del demandante, se expresó que el producto final cumple ampliamente con las expectativas definidas al inicio del proyecto. Se destacó una experiencia de usuario “superadora y muy satisfactoria”, validada mediante pruebas en establecimientos ganaderos de Argentina y Uruguay.

En función de las respuestas obtenidas, tanto por parte del demandante como de los usuarios finales, se concluye que el producto entregado fue un éxito. No solo resolvió las limitaciones de la versión anterior, sino que además introdujo mejoras valoradas por sus usuarios, cumpliendo con las expectativas y generando un impacto positivo en el uso diario de la herramienta.

Comparación entre los plazos estimados y los reales

A continuación se analiza cómo las estimaciones de tiempo realizadas al comienzo del proyecto se comparan con su ejecución real. Si bien algunas etapas se desarrollaron dentro de los márgenes esperados, otras presentaron desviaciones notorias.

Al iniciar el proyecto, se realizó una planificación de tiempos dividiendo el proyecto en distintas etapas. Para cada una se estimó la cantidad de horas necesarias considerando una dedicación de diez horas semanales por integrante. Esto permitió proyectar la duración total del proyecto en semanas, reflejada en el diagrama de Gantt presentado anteriormente.

Sin embargo, durante la ejecución real surgieron desvíos. La Figura 25 muestra la distribución real del tiempo según tareas, mientras que la Figura 26 compara los tiempos planificados con los finalmente ejecutados por etapa.

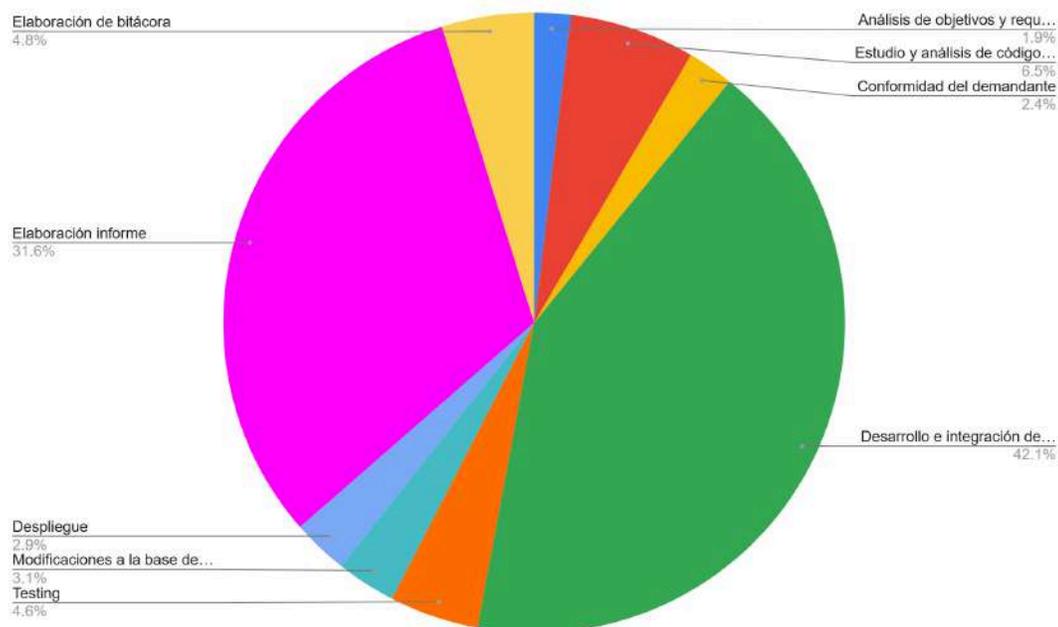


Figura 25: Diagrama de torta con las tareas realizadas y el porcentaje de tiempo que representan

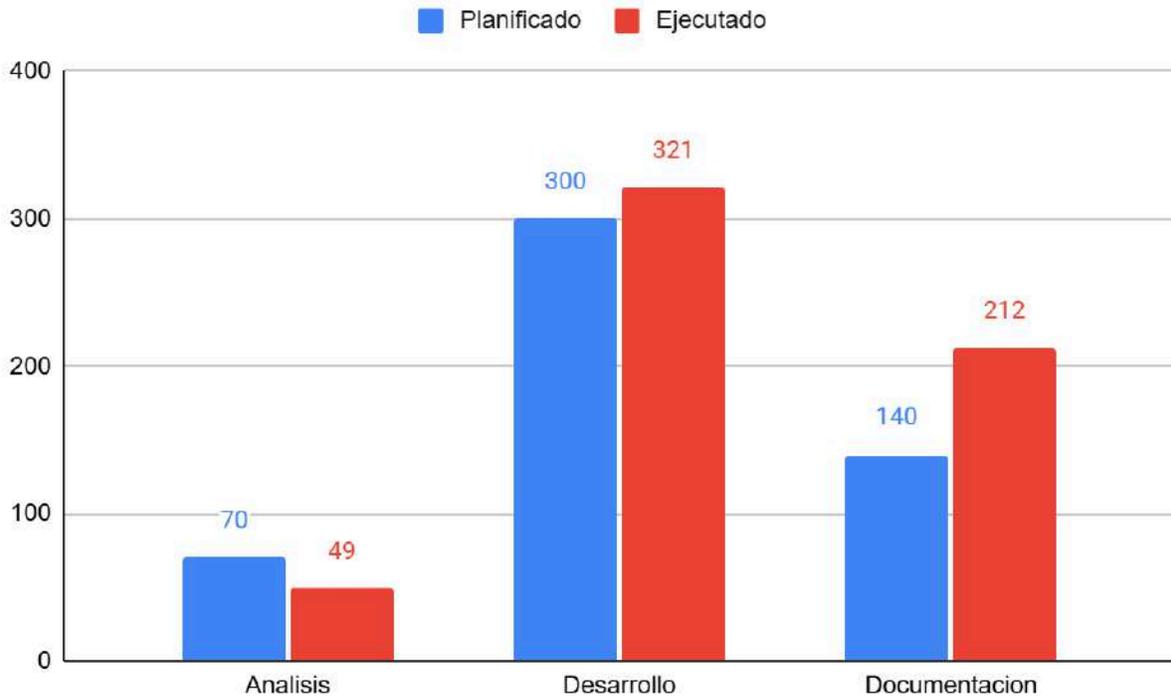


Figura 26: Gráfico de barras comparativo entre los tiempos planificados y ejecutados de cada etapa

La etapa de análisis se completó en mucho menos tiempo del estimado inicialmente. En retrospectiva, el equipo sobreestimó la duración de esta fase debido a la falta de experiencia en proyectos reales de esta magnitud. Se adoptó una postura conservadora por precaución, asumiendo que la comprensión del dominio y la definición de requerimientos llevarían más tiempo. Sin embargo, gracias a una organización clara, buena comunicación con el demandante y una rápida apropiación del problema, el análisis fue mucho más fluido de lo esperado.

Cabe aclarar que, aunque el análisis del código existente inicialmente formaba parte de esta etapa, el equipo decidió avanzar sin asistencia del equipo anterior y explorar el código de manera autónoma. Esto implicó que parte del proceso de análisis técnico, especialmente la comprensión del funcionamiento interno del sistema heredado, se realizara de forma progresiva y distribuida entre la etapa de análisis y la etapa de desarrollo, lo que permitió reducir la dependencia externa con el equipo anterior y evitó el costo elevado de la reescritura del código.

El desarrollo fue completado en un plazo ligeramente superior al planificado, y representó la estimación más acertada del proyecto. Aun así, se identificaron causas específicas del desvío: en las primeras etapas se implementaron funcionalidades que más adelante fueron descartadas o modificadas, lo que obligó a reestructurar parte del trabajo. Además, parte del desvío se debe a falta de conocimiento del código heredado y el tiempo dedicado a la comprensión del mismo, dado que, como se mencionó, el equipo invirtió tiempo en el estudio del código en la etapa de desarrollo. A pesar de estos factores, la estimación resultó bastante acertada ya que el equipo decidió realizar una estimación inicial conservadora, contemplando la aparición de posibles demoras.

La etapa de documentación fue la más desafiante y la que mayor desvío presentó respecto a lo planificado. El equipo tenía experiencia trabajando en conjunto y se conocía la manera de trabajar de cada integrante. Es por eso que inicialmente se decidió, desacertadamente, afrontar la escritura del informe de manera asíncrona dividiendo las diferentes secciones entre los integrantes para una escritura más rápida. Sin embargo, esta estrategia tuvo el efecto opuesto, ya que a la hora de realizar la puesta en común fue necesario realizar mucha reescritura para que las piezas logren formar un texto coherente. Muchas secciones fueron modificadas mientras que otras fueron reescritas por completo, lo que se tradujo en un alto costo de tiempo dedicado a la reescritura.

Hacia el final de la documentación del proyecto, el equipo tomó como aprendizaje que trabajar de manera asíncrona y dividida no es la mejor opción a la hora de documentar un proyecto de esta magnitud, y que es preferible dedicar más tiempo a la coordinación y escritura conjunta.

Conclusiones

El producto entregado no solo cumplió con los objetivos planteados al inicio del proyecto, sino que superó ampliamente las expectativas del demandante, quien destacó la nueva experiencia de usuario como “superadora y muy satisfactoria”. Esta validación fue respaldada por pruebas realizadas en distintos establecimientos ganaderos de Argentina y Uruguay, donde se pudo comprobar en un contexto real el valor del sistema implementado.

Por su parte, los usuarios finales también percibieron una mejora sustancial. A través de la retroalimentación recopilada por el demandante, se destacó que la plataforma es ahora más fácil de usar, rápida y precisa. El proceso de calibración fue simplificado, la aplicación móvil es más fácil de utilizar, y los gráficos incorporados se volvieron herramientas útiles para la toma de decisiones. Estas mejoras reflejan que se entregó un producto verdaderamente superador.

Además, el desarrollo del sistema fue una instancia clave de aprendizaje para el equipo. No solo se adquirieron nuevas competencias técnicas, sino que también se enfrentaron desafíos reales al trabajar sobre un sistema ya existente y en producción.

Un desacierto importante durante la etapa de análisis fue subestimar el impacto potencial de ciertos riesgos. Con el avance del proyecto surgieron riesgos y debilidades no contempladas que afectaron directamente los tiempos y la organización del trabajo. Esto permitió al equipo comprender que el análisis inicial no es estático, y que tanto el análisis FODA como el análisis de riesgos deben revisarse y ajustarse a lo largo del desarrollo.

Por otro lado, haber sido conservadores con las estimaciones de tiempo resultó ser una decisión acertada, mientras que la elección de trabajar de manera completamente asincrónica de forma dividida durante la documentación no fue efectiva, y se reconoció que una mayor coordinación habría agilizado la documentación del proyecto.

En base a los resultados obtenidos, la satisfacción del demandante, la recepción positiva por parte de los usuarios y el crecimiento profesional alcanzado por el equipo, puede afirmarse con total claridad que el proyecto fue un éxito.

Bibliografía

Recursos Web

1. Especificación OpenAPI. Recuperado de <https://learn.openapis.org/>
2. Generación de documentación de APIs con Swagger. Recuperado de <https://swagger.io/solutions/api-documentation/>
3. Guía introductoria de Postman. Recuperado de <https://learning.postman.com/docs/introduction/overview/>
4. Guía de rutas de Express. Recuperado de <https://expressjs.com/en/guide/routing.html>
5. API oficial de Node.js. Recuperado de <https://nodejs.org/docs/latest/api/>
6. Documentación de React. Recuperado de <https://react.dev/reference/react>
7. Documentación de Expo. Recuperado de <https://docs.expo.dev/>
8. Conceptos de Webpack. Recuperado de <https://webpack.js.org/concepts/>
9. Introducción a Material UI. Recuperado de <https://mui.com/material-ui/getting-started/>
10. Documentación de Railway. Recuperado de <https://docs.railway.com/>
11. Documentación oficial de React Native. Recuperado de <https://reactnative.dev/>
12. Uso del módulo de localización en Expo. Recuperado de <https://docs.expo.dev/versions/latest/sdk/location/>
13. Introducción al framework Scrum. Recuperado de <https://www.atlassian.com/agile/scrum>
14. Recursos sobre Scrum. Recuperado de <https://www.scrum.org/resources/what-scrum-module>

15. Guía de diseño RESTful. Recuperado de <https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design>
16. Artículo Regresión Lineal. Recuperado de https://en.wikipedia.org/wiki/Linear_regression
17. Artículo FODA. Recuperado de <https://asana.com/es/resources/swot-analysis>
18. Documentación GIT. Recuperado de <https://git-scm.com/doc>
19. Documentación SQLite. Recuperado de <https://www.sqlite.org/>

Libros

1. Pressman, R. (2010). *Ingeniería del Software. Un enfoque práctico*. 7ma Ed. McGraw Hill.
2. Kendall, K., & Kendall, J. (2005). *Análisis y diseño de sistemas*. 8va Ed. Pearson.
3. Sommerville, I. (2005). *Ingeniería del Software*. 7ma Ed. Pearson.

Anexo: Glosario

API: Interfaz de Programación de Aplicaciones; un conjunto de definiciones y protocolos que permiten la comunicación entre diferentes sistemas o aplicaciones.

Backend: Lógica y procesamiento detrás de una aplicación que gestiona datos y funcionalidades.

Calibración: Función utilizada para calcular el stock de pasto en base a alturas medidas. Puede ser estándar o creada por el usuario.

Circuito: Conjunto de potreros agrupados para facilitar su gestión en la plataforma.

Experiencia de Usuario (UX): Percepción y satisfacción del usuario al interactuar con el sistema.

Frontend: Parte visible de una aplicación con la que interactúa el usuario.

Mapa Georreferenciado: Representación visual de datos espaciales sobre un mapa.

Medición: Registro georreferenciado de datos de altura de pasto tomados en campo con el pasturómetro.

NDVI (Índice de Vegetación de Diferencia Normalizada): Indicador que mide la intensidad de la vegetación utilizando imágenes satelitales.

Pasturómetro Electrónico: Dispositivo utilizado para medir la altura de pasto en un potrero.

Potrero: Parcela de tierra delimitada y utilizada para el pastoreo del ganado.

REST: Estilo de arquitectura para APIs que utiliza operaciones estándar de HTTP como GET, POST, PUT y DELETE.

SQL: Lenguaje de consulta estructurado utilizado para gestionar bases de datos relacionales.

Stock: Cantidad de pasto disponible en un potrero o conjunto de potreros.

Zona: Área más amplia que incluye varios circuitos y potreros, utilizada para análisis a nivel macro.

Anexo: Documentación de Endpoints

Uno de los principales cambios implementados al iniciar el desarrollo fue la creación de documentación para los endpoints existentes. Originalmente no existía documentación y la única forma de comprender el funcionamiento de los endpoints era mediante la exploración manual del código. Esto implicaba que cada vez que se necesitaba trabajar con un endpoint, era necesario revisar el código nuevamente y esto representaba una tarea tediosa. Para optimizar este proceso y facilitar futuros desarrollos, se decidió documentar todos los endpoints que expone el servidor.

Para realizar la tarea de documentación del código durante el desarrollo se utilizó la especificación OpenAPI, un estándar diseñado para describir APIs REST, detallando todos los aspectos necesarios para su consumo, como rutas, métodos, parámetros y respuestas, lo que permite contar con una documentación clara, estructurada y fácil de comprender. Para generar documentación siguiendo este estándar se empleó la herramienta Swagger, utilizando comentarios incluidos en las funciones que implementan los endpoints. Esto simplifica considerablemente el mantenimiento y la actualización de la documentación, facilitando tanto el desarrollo como la comprensión de las APIs.

En la Figura 27 se puede ver el ejemplo de una función utilizada para implementar uno de los endpoints, en este caso relacionado al pasturómetro, y el comentario en base al cual se genera la documentación:

```
/**
 * @swagger
 * /api/pasturometro/potreros:
 *   get:
 *     summary: Obtener potreros con mediciones
 *     description: Obtiene todos los potreros activos que tienen mediciones asociadas para el cliente autenticado
 *     tags:
 *       - Pasturometro
 *     security:
 *       - bearerAuth: []
 *     responses:
 *       200:
 *         description: Lista de potreros con sus mediciones
 *       401:
 *         $ref: '#/components/responses/UnauthorizedError'
 *       500:
 *         $ref: '#/components/responses/ServerError'
 */
router.get("/potreros", obtenerPotrerosConMediciones);
```

Figura 27: Ejemplo de documentación de endpoints en el código con Swagger

Swagger permitió crear una página adicional alojada en el mismo dominio que el resto de la aplicación, que contiene la documentación completa de todos los endpoints. Dicha documentación es accesible desde el endpoint /api/docs, e incluye información como:

- Ruta
- Parámetros de entrada y su tipo
- Salida esperada y su tipo
- Requerimientos de autenticación
- Estados devueltos (éxito, error, etc.)
- Ejemplo de uso

Las Figuras 28 y 29 ilustran el contenido de la página de documentación:

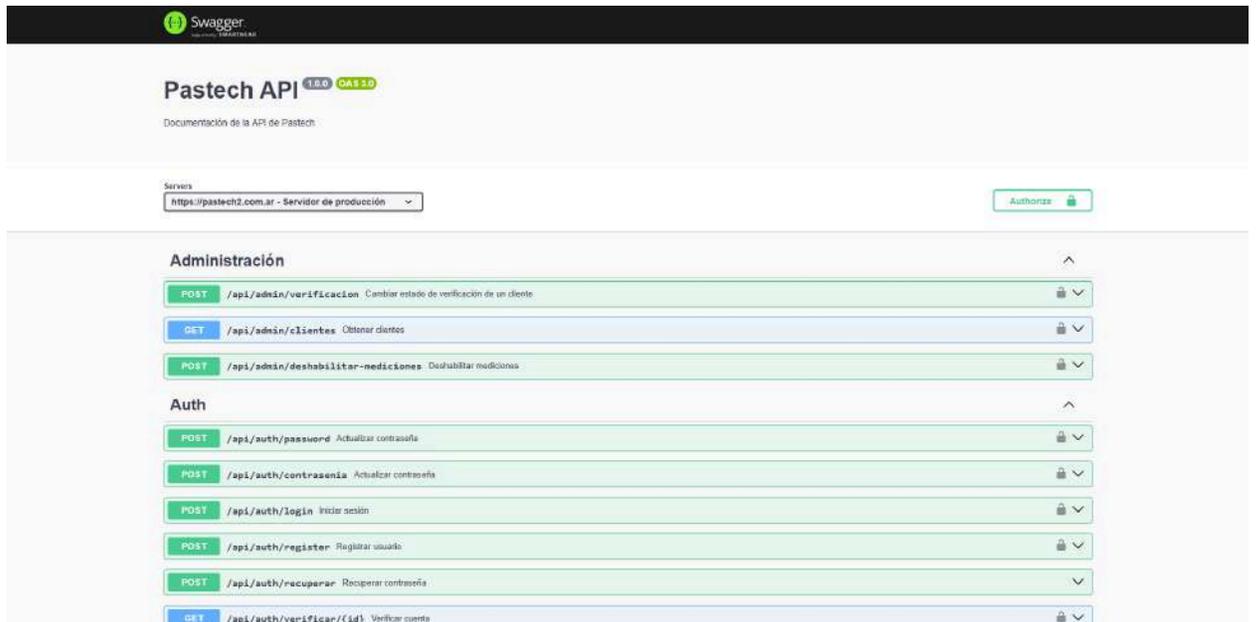


Figura 28: Vista general de la documentación de los endpoints

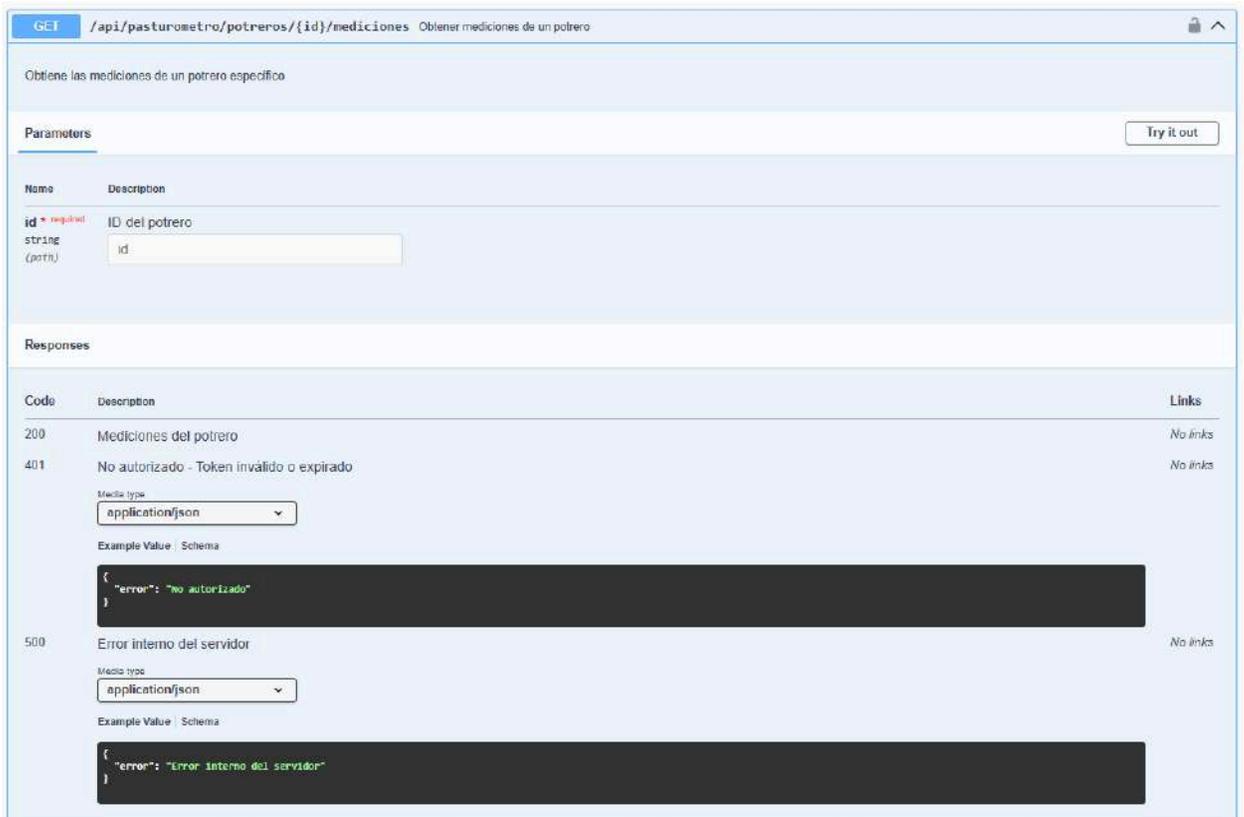


Figura 29: Vista particular de la documentación de un endpoint

Es importante destacar que esta página contiene información sensible, ya que expone las interfaces internas utilizadas por los módulos del sistema. Por este motivo, fue necesario implementar medidas de seguridad que aseguren el acceso únicamente a usuarios autorizados.

Anexo: Refactorización de Rutas

Una de las tareas que contribuyó significativamente a facilitar el desarrollo fue la refactorización del manejo de rutas de los endpoints REST. En su estado original, el proyecto se basaba en un único archivo de entrada (`index.js`) que centralizaba todas las vistas de la aplicación, los manejadores de rutas, los controladores y las configuraciones del servidor.

Se decidió realizar una refactorización inicial, enfocándose en lograr una mayor modularización de las rutas y los controladores responsables de implementar los endpoints de la API REST. Como primer paso, se organizó la API dividiendo sus endpoints según su funcionalidad (por ejemplo, pasturómetro, calibraciones, potreros, etc.). Con esta estructura de alto nivel definida, se procedió a crear un manejador de rutas independiente para cada funcionalidad de la API. Estos controladores se implementaron en archivos separados, lo que facilitó su mantenimiento y mejoró la modularidad del proyecto.

Por ejemplo, todos los controladores de rutas relacionados con el pasturómetro se consolidaron en un único archivo denominado `pastuometro.routes.js`. Posteriormente, los manejadores de rutas de todas las funcionalidades fueron unificados en un único manejador general bajo el endpoint `/api`, conformando así una estructura en forma de árbol para las rutas de la API.

En la Figura 30 se presenta un diagrama ilustrativo de la estructura de la API para la funcionalidad del pasturómetro. En este diagrama, los nodos hoja representan los endpoints junto con sus rutas correspondientes.

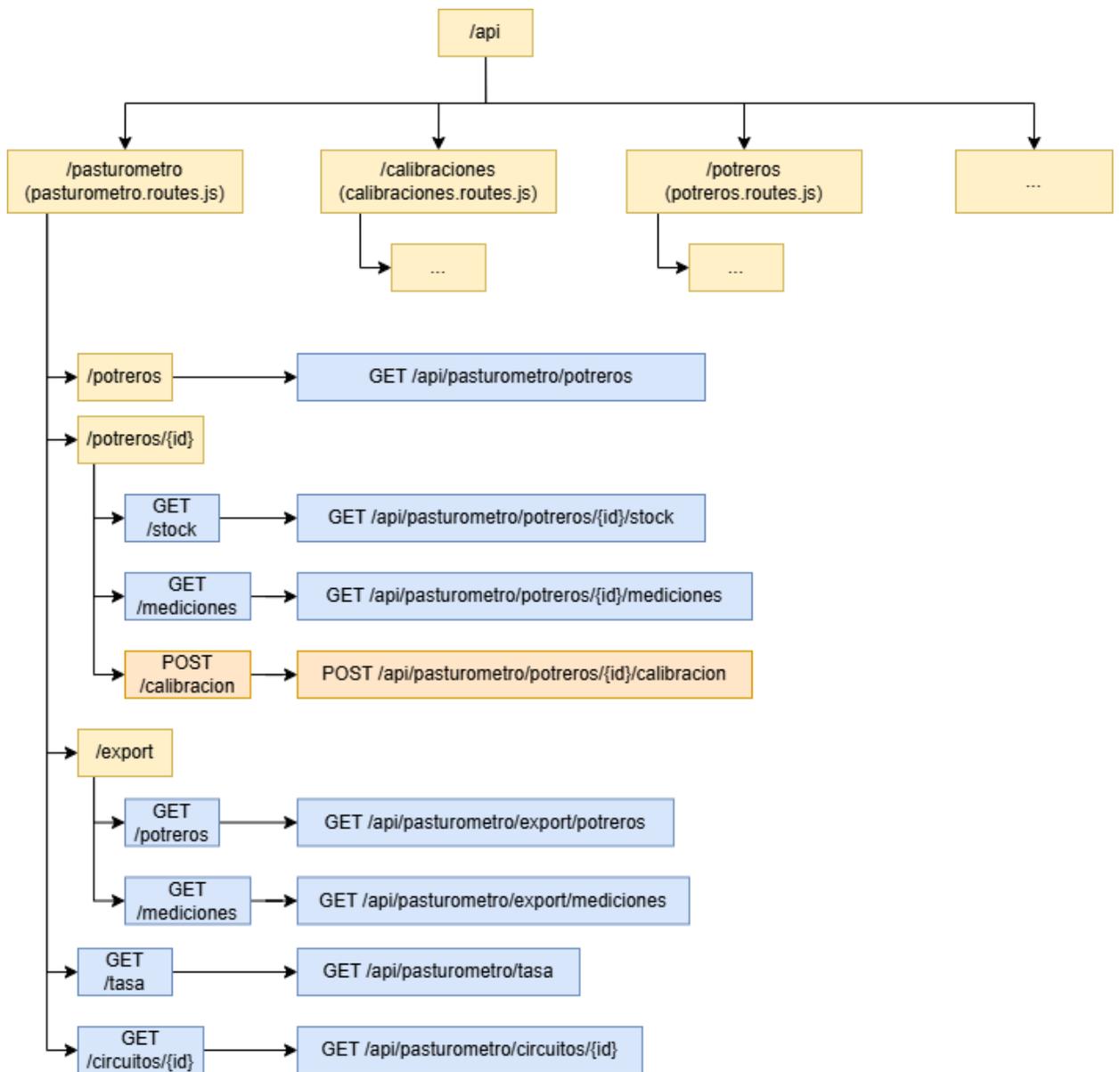


Figura 30: Diagrama ilustrativo de la estructura de la API para la funcionalidad del pasturómetro

Esta refactorización resultó fundamental para agilizar el desarrollo, al mismo tiempo que mejoró significativamente la legibilidad y la mantenibilidad del código.