



UNIVERSIDAD TECNOLÓGICA NACIONAL

Facultad Regional Córdoba

Ingeniería en Sistemas de Información

PastAR

Plataforma de Monitoreo Ganadero de Precisión y Trazabilidad Individual

Curso: 5K3

Docentes: Ing. Quinteros, Sergio (Profesor Adjunto) y Ing. Martinez, Gabriel (JTP asignado)

Alumnos:

- Salvador gibert - 94181
- Agustin Rey Laje - 95535
- Tiziana Carrizo - 94506
- Marco Figueroa - 94359
- Ignacio Maspero - 94304



ALUMNOS

CARRIZO, Tiziana Frontend
FIGUEROA, Marco Backend
GIBERT, Salvador Product Owner
MASPERO CASTRO, Jose IoT & Firmware
REY LAJE, Agustin Scrum Master & QA

Plataforma de Monitoreo Ganadero de Precisión y Trazabilidad Individual

Monitorea cada animal las 24 horas —ubicación, actividad y estrés térmico— para que cabañas y productores ganaderos detecten celo, parto y problemas sanitarios sin recorrer el campo.

Usuarios y accesos
Roles y permisos

Trazabilidad
Ficha individual y sanidad

Monitoreo en vivo
Mapa georreferenciado

Analisis predictivo
ITH, celo y alertas

Biblioteca de razas
Enciclopedia + recomendador

Collar sensor
ESP32, GPS, solar

EL PROBLEMA

En la cría se maneja la información más crítica a papel y a ojo.

68 %

de las vacas desteta un ternero al año en la Argentina

+80%

el mismo índice en ambientes comparables del exterior

Fuente: INTA — Producción Animal.

Sin registro por animal

La información existe, pero dispersa: cuadernos, planillas y memoria.

Eventos que nadie ve

El celo dura horas y el parto no avisa. Lo que no se ve, no se registra.

Decisiones sin evidencia

Nadie puede probar con datos qué reproductores rinden en su ambiente

Donde no llega la infraestructura

Sin cobertura ni electricidad, lo que sirve en un tambo no sirve acá.

LA SOLUCION

Un collar en cada animal y una plataforma que traduce el dato en decisión.

24 h

de telemetría continua por animal

≥ 3 dias

de autonomía sin son ni red.

Ficha digital por animal

Sanidad, peso y reproducción, del nacimiento a la venta.

Detección automática de eventos

El acelerómetro marca celo, parto y anomalías. La alerta llega sola.

Métricas objetivas de adaptación

Correlaciona ITH y comportamiento por raza.

Collar autónomo sin red

ESP32 con GPS, buffer local y carga solar.

PastAR

ARQUITECTURA DEL SISTEMA

CAPTURA - EDGE

Collar Sensor – ESP32

Acelerómetro · GPS · temperatura
Muestreo periódico sobre el animal

Firmware – C++

Buffer local ante falta de enlace
Deep Sleep · autonomía ≥ 3 días

Enlace de campo

Gateway LoRa hacia el backend
Contrato de API definido

PROCESAMIENTO Y ENTREGA

API REST

Django + DRF · OAuth2
JWT · roles

Persistencia

PostgreSQL
serie temporal por animal

Dashboard web

Next.js + Tailwind
mapa georreferenciado

COMO FUNCIONA



- 1 Mide

Actividad, posición y temperatura, cada N minutos.
- 2 Compara

Contra la línea base del animal y la de su raza.
- 3 Clasifica

Pastoreo, rumia, desplazamiento o reposo.
- 4 Avisa y aprende

Alerta. Su confirmación recalibra el umbral.

LA PLATAFORMA

PastAR

- Dashboard
- Traceability
- Alerts & Monitoring
- Virtual Fencing
- Predictive Analysis
- Breed Library

Notificaciones

Configuración

Cerrar sesión

Gestión de Animales

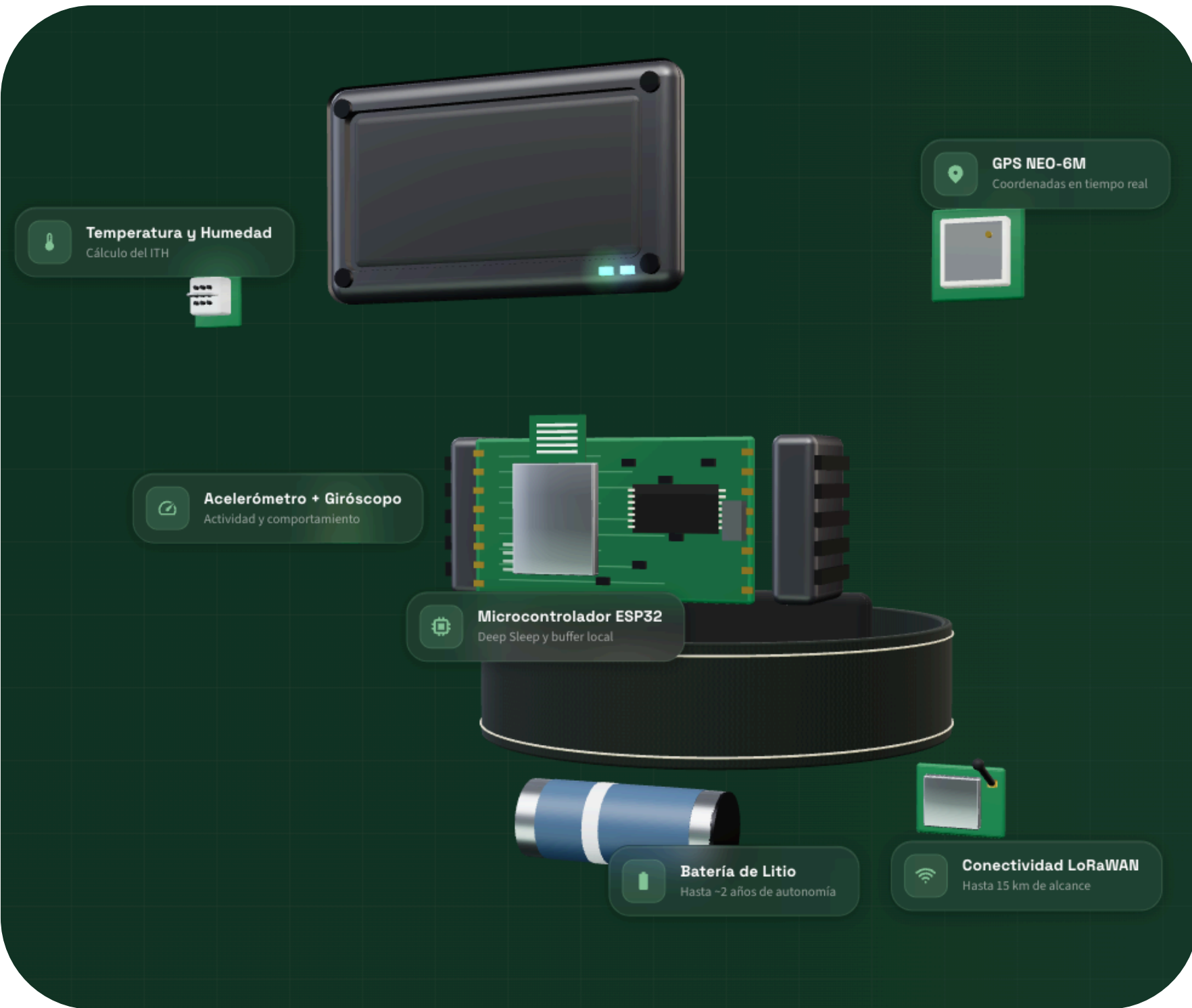
administra el registro y estado de tu rodeo.

Buscar por ID / caravana

Todas las razas

ID / CARAVANA	RAZA	SEXO	EDAD (MESES)	PESO ACTUAL	ESTADO
AR-0001	Aberdeen Angus	Hembra	36	432,5 kg	Activo
AR-0002	Brangus	Macho	28	465 kg	Activo
AR-0003	Hereford	Hembra	48	395 kg	Activo
AR-0004	Brasford	Hembra	18	—	Activo
AR-0005	Limousin	Macho	59	575 kg	Activo
AR-0006	Aberdeen Angus	Hembra	24	350 kg	Activo
AR-0007	Hereford	Macho	12	—	Activo
AR-0008	Brangus	Hembra	40	430 kg	Activo
ARG-2026-2202	—	Macho	30	—	Activo

EL DISPOSITIVO



RESULTADOS ESPERADOS

EN EL RODEO
Sanidad y estrés detectados antes de que afecten el peso.

EN LA REPRODUCCIÓN
Menos servicios perdidos por celos no detectados.

EN LA CABAÑA
Reproductores valorizados con datos de adaptación.

EN LA GESTIÓN
Historial digital por animal. Se termina el cuaderno.

EN EL NEGOCIO
SaaS desde USD 250/mes. Break-even: 5 clientes Pro.



PastAR: Plataforma de Monitoreo Ganadero de Precisión y Trazabilidad Individual Basada en IoT

Carrizo, Tiziana; Figueroa, Marco; Gibert, Salvador; Maspero Castro, José Ignacio; Rey Laje, Agustín

Universidad Tecnológica Nacional, Facultad Regional Córdoba

Abstract

Los establecimientos de cría bovina gestionaron históricamente la trazabilidad individual del rodeo en papel y dependieron de la recorrida visual a campo para detectar animales enfermos, en celo o extraviados, lo que generó demoras, errores de carga y pérdida del historial productivo. Para atender este problema se desarrolló PastAR, una plataforma que combinó un collar sensor IoT con un backend en Django REST Framework y un dashboard web en Next.js. El equipo trabajó bajo Scrum, con sprints de dos semanas, y adoptó un esquema de trabajo desfasado: el backend de cada historia se implementó en un sprint y el frontend correspondiente recién en el siguiente, para evitar los conflictos de integración detectados al desarrollar ambas capas en paralelo. Al momento de este reporte se encontraban operativos el módulo de Trazabilidad (alta, consulta, modificación y baja de animales, ficha sanitaria, control ponderal, árbol genealógico y confirmación de partos), el de Monitoreo en tiempo real (mapa georreferenciado e historial de trayectoria) y la Biblioteca de Razas, mientras el resto del backlog continuaba en curso. La arquitectura monolítica con separación lógica interna, elegida según el tamaño del equipo, sostuvo el ritmo de desarrollo académico sin comprometer la calidad del código entregado.

Palabras Clave

IoT ganadero, trazabilidad individual, monitoreo en tiempo real, ganadería de precisión, Django REST Framework, Next.js, LoRaWAN, Scrum. Categoría: Producto.

I. Introducción

La ganadería de cría bovina enfrenta un problema de información: la identificación individual del animal ya existe a través de la caravana electrónica del SENASA, pero el registro de los eventos sanitarios, reproductivos y productivos de cada animal se realiza en papel, y el estado del rodeo se conoce únicamente mediante la recorrida visual a campo varias veces al día. Esta dependencia genera demoras en la detección de animales enfermos, en celo o extraviados, y dispersa la información histórica de cada animal entre distintos cuadernos de campo, dificultando su consolidación y análisis.

PastAR es una plataforma de monitoreo ganadero de precisión y trazabilidad individual que busca resolver este problema aprovechando la identificación electrónica ya existente. El objetivo del proyecto es centralizar el historial productivo de cada animal y sumarle un monitoreo continuo mediante un collar sensor de bajo costo, para anticipar problemas sanitarios, reproductivos y de bienestar sin depender exclusivamente de la observación humana.

Este artículo describe el trabajo realizado durante los primeros tres sprints del desarrollo. La sección II detalla las herramientas, la metodología y las decisiones de arquitectura adoptadas. La sección III presenta los resultados obtenidos a la fecha. La sección IV discute esos resultados en relación con el objetivo planteado, y la sección V concluye con el estado del proyecto y el trabajo futuro.

II. Elementos del Trabajo y Metodología

El equipo adoptó Scrum adaptado al contexto académico, con sprints de dos semanas, roles rotativos entre los cinco integrantes y un asesor agropecuario externo que participó únicamente en las Sprint Reviews. El trabajo se organizó en Jira (backlog y tablero Kanban), Confluence (documentación técnica) y GitHub (código y revisión de pull requests), con una rama por historia y aprobación obligatoria de al menos un integrante antes de cada merge.

La arquitectura de la solución se definió mediante el modelo C4 y un conjunto de registros de decisión de arquitectura (ADR). Se optó por un monolito con separación lógica interna en lugar de microservicios, dado el tamaño del equipo y la necesidad de garantizar que la ingesta de datos del collar no compitiera por recursos con el tráfico del dashboard. El backend se implementó en Django y Django REST Framework, por su ecosistema maduro para ciencia de datos (pandas, scikit-learn,

SHAP, Prophet), mientras que el procesamiento periódico de los datos crudos del collar se delegó a un worker de Celery con Redis como broker, disparado cada 30 minutos por Celery Beat, para no competir por CPU con los endpoints de la API. El frontend se desarrolló en Next.js con Tailwind CSS. La conectividad de campo se resolvió con una red LoRaWAN de antena fija, elegida por su alcance de hasta 15 km y bajo consumo energético frente a la ausencia de cobertura celular en establecimientos rurales extensos. El hosting del MVP se concentró en Railway, que permite correr los tres procesos (web, worker, beat) desde un mismo repositorio.

El equipo adoptó un esquema de trabajo desfasado entre sprints para una misma funcionalidad: el backend de una historia se implementó en un sprint, y el frontend que lo consume se construyó recién en el sprint siguiente, ya con el endpoint real disponible. Este esquema se adoptó después de que, en el Sprint 2, desarrollar frontend y backend de las mismas historias en paralelo generará conflictos de integración entre los integrantes del equipo.

Nota sobre uso de IA: la redacción de este artículo se asistió con un modelo de lenguaje (Claude, Anthropic) para estructurar el texto según el formato IMRYD solicitado. Los datos, métricas y referencias fueron verificados por el equipo contra la documentación interna del proyecto en Jira y Confluence antes de su publicación.

III. Resultados

Al cierre del período reportado (Sprint 3), el módulo de Trazabilidad quedó operativo de punta a punta: alta, consulta, modificación y baja lógica con motivo de un animal; ficha sanitaria con historial de vacunaciones, tratamientos y desparasitaciones; control ponderal con cálculo automático de la Ganancia Diaria de Peso; árbol genealógico de hasta tres generaciones, con la regla de que un antecesor con descendencia registrada no puede eliminarse; y confirmación de parto, que genera automáticamente la ficha y el vínculo genealógico de cada cría, incluyendo mellizos.

El módulo de Monitoreo en tiempo real incorporó un mapa georreferenciado del rodeo

con filtro por animal, lote y categoría, y un historial de trayectoria por animal con reproducción de su recorrido en un rango de fechas. La Biblioteca de Razas expuso una enciclopedia de 16 razas bovinas de aptitud cárnica, agrupadas por grupo genético, con búsqueda y filtro.

Los módulos de Análisis Predictivo y de Recomendador de Razas, planificados para releases posteriores al MVP, todavía no se iniciaron.

IV. Discusión

Los resultados muestran que el objetivo de centralizar la trazabilidad individual se cumple para el subconjunto de eventos ya cubierto por el módulo de Trazabilidad, y que el monitoreo en tiempo real alcanza un primer nivel funcional. El esquema de trabajo desfasado entre sprints resulta efectivo para reducir los conflictos de integración detectados en el Sprint 2: al no tocar frontend y backend de una misma historia en paralelo, cada integrante trabaja sobre una base más estable.

Esta misma estrategia introduce una limitación: una historia con componente de interfaz no queda demostrable de punta a punta hasta, como mínimo, el sprint siguiente a aquel en el que se implementó su backend, lo que retrasa la validación completa frente al Product Owner y complica la medición de la velocidad en el sprint en el que solo se entrega una de las dos capas.

El mayor riesgo detectado no es técnico sino de gestión: la retrospectiva del Sprint 2 identifica que el tablero de Jira no siempre refleja el estado real del trabajo, lo que dificulta medir la velocidad del equipo y la visibilidad del avance para el Product Owner y el JTP.

V. Conclusión

PastAR alcanza, al cierre del período reportado, un producto funcional para los módulos de Trazabilidad, Monitoreo en tiempo real y Biblioteca de Razas, validando la arquitectura monolítica con separación lógica interna y el esquema de trabajo desfasado entre sprints —backend primero, frontend en el sprint siguiente— como estrategias adecuadas para un equipo reducido con

restricciones de tiempo académico, adoptadas luego de los conflictos de integración detectados en el Sprint 2. La principal lección del proyecto es que la sincronización del equipo —mantener el tablero actualizado y coordinar el orden de trabajo entre capas— pesa tanto como las decisiones técnicas de arquitectura.

El trabajo futuro incluye integrar el collar sensor físico, e iniciar los módulos de Análisis Predictivo y Recomendador de Razas planificados para releases posteriores al MVP.

Agradecimientos

Los autores agradecen a Nicolás Schmidt, asesor agropecuario externo del proyecto, por su aporte de conocimiento del dominio ganadero, y al JTP Gabriel Martínez Ocampo y al Profesor Adjunto Sergio Quinteros por su acompañamiento durante la cátedra.

Referencias

- [1] Django Software Foundation, "Django REST Framework Documentation". [En línea]. Disponible: <https://www.django-rest-framework.org/>. [Último acceso: jul. 2026].
- [2] Vercel, "Next.js Documentation". [En línea]. Disponible: <https://nextjs.org/docs>. [Último acceso: jul. 2026].
- [3] SENASA, "Sistema de identificación individual de ganado bovino y bubalino". [En línea]. Disponible: <https://www.argentina.gob.ar/senasa>. [Último acceso: jul. 2026].
- [4] Open-Meteo, "Weather API". [En línea]. Disponible: <https://open-meteo.com/>. [Último acceso: jul. 2026].
- [5] R. A. Day, "How to Write and Publish a Scientific Paper", 3ra ed., Phoenix, AZ: Oryx Press, 1991.
- [6] Equipo PastAR, "Plan de Proyecto y Arquitectura y Stack Tecnológico — Instancia 1 y 2", documentación interna del proyecto, UTN FRC, 2026.
- [7] Anthropic, "Claude" (modelo de lenguaje, asistencia de redacción). [En línea]. Disponible: <https://claude.ai/>. [Último acceso: jul. 2026].

Datos de Contacto

Salvador gibert - salvigibert@gmail.com

Agustin Rey Laje - agusreylaje@gmail.com

Tiziana Carrizo - tizianacarrizo123@gmail.com

Marco Figueroa - marcofiguero3@gmail.com

Ignacio Maspero - nachomaspero2@gmail.com

Anexo II Planilla para Catalogación de Proyecto

PLANILLA PARA CATALOGAR EL PROYECTO FINAL

AÑO	2026	CURSO Y NRO. DE GRUPO	5K3 - G23
NOMBRE DEL SISTEMA / PROYECTO			
PastAr			
CATEGORÍA (Solución De Negocio / Producto / Proyecto De Impacto Social)			
Producto			
HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS			
ÁMBITO DE APLICACIÓN		NOMBRE Y VERSIÓN	
ENTORNO DE DESARROLLO		VS Code · PlatformIO (para el ESP32)	
REPOSITORIOS Y VERSIONADO		Git + GitHub (estrategia GitFlow: main / develop / sprint/x / PAST-x)	
PROGRAMACIÓN		Python 3 + Django + Django REST Framework (backend) · TypeScript + Next.js + Tailwind CSS (frontend) · C++ (firmware ESP32)	
BASE DE DATOS		PostgreSQL	
COMUNICACIÓN INTERNA		WhatsApp (equipo), Jira, Google Meet	
CAPACITACIÓN		No aplica	
PRUEBAS DE SISTEMA		Django TestCase / pytest	
GESTIÓN DEL PROYECTO		Jira (tickets PAST-x, metodología Scrum) · GitHub Actions (CI/CD)	
DOCUMENTACIÓN		Confluence	
MODELOS		Draw.io - Miro	