



UNIVERSIDAD NACIONAL DE ROSARIO  
FACULTAD DE CIENCIAS EXACTAS, INGENIERÍA Y AGRIMENSURA  
ESCUELA DE INGENIERÍA ELECTRÓNICA

# SASE 2025

## Prevención de Robo de Ganado con IoT: Sistema de monitoreo y sensado de signos vitales en campo abierto

### **Autores**

González, Tomás F.  
Morris Esquivel, Brian

### **Directora**

Dra. Verónica Miro

### **Resumen**

Se presenta un sistema IoT de bajo consumo para el rastreo y monitoreo del bienestar bovino. Se desarrolla un dispositivo de rastreo y sensado de signos vitales, basado en un STM32L0 integrándolo con sensores de temperatura y ritmo cardíaco, y módulos GPS y LoRa para la geolocalización y transmisión de datos.

Se desarrolla también un dispositivo central, compuesto por un microcontrolador ESP32 y dos módulos LoRa para distintos canales de transmisión, este recibe los datos y los publica por MQTT hacia un servidor web central.

# 1. Introducción

La ganadería argentina afronta dos retos crecientes: el robo sistemático de animales (*abigeato*) y la necesidad de mejorar indicadores de bienestar para acceder a mercados cada vez más exigentes. Este trabajo parte de la hipótesis de que un sistema IoT de muy bajo consumo, desplegado directamente sobre el animal, puede suministrar alertas tempranas y métricas objetivas a un costo asumible por productores medianos. El proyecto se desarrolla como Proyecto Final de Carrera en la Escuela de Ingeniería Electrónica de la UNR, con un enfoque integral que abarca desde la integración digital hasta la exposición de datos al usuario.

Además del aporte tecnológico, el proyecto busca demostrar la viabilidad económica de un esquema escalable: múltiples nodos de sensado (LSU) coordinados por una unidad central (CU) que consolida datos y los publica en un servidor central. Este planteo apunta a facilitar la adopción por parte de productores y la integración con plataformas existentes de gestión ganadera.

## Objetivo y alcance

Desarrollar un sistema IoT autónomo para monitoreo de signos vitales y ubicación en bovinos, capaz de operar más de 6 meses sin intervención y generar alertas ante eventos de robo o estrés animal, y cubrir un campo de 3 kilómetros de diámetro.

- Diseñar una LSU con sensores de temperatura, ritmo cardíaco y GPS, y enlace LoRa de largo alcance
- Implementar una CU capaz de orquestar hasta 100 LSU y publicar datos a un servidor mediante MQTT.
- Montar un prototipo del producto y validar precisión y cobertura de enlace.

## 2. Estado del arte

Las soluciones comerciales de trazabilidad bovina se basan típicamente en collares GSM+GPS, con costos superiores a USD 80 por unidad, lo cual restringe su adopción masiva en establecimientos medianos. En el ámbito académico y de I+D, se observa un creciente interés por LPWAN (LoRa, Sigfox, NB-IoT) para monitoreo de fauna y ganado debido a su largo alcance y eficiencia energética; sin embargo, muchos trabajos se enfocan exclusivamente en geolocalización, omitiendo variables fisiológicas clave para el bienestar animal.

En particular, estudios recientes exploran la integración de LoRa con redes de sensores distribuidas, pero suelen apoyarse en topologías simples (uno a uno) o protocolos MAC genéricos que no optimizan el uso del canal cuando la cantidad de nodos escala. Este trabajo intenta combinar prácticas de diseño *long-range low-power* con un conjunto mínimo pero útil de variables fisiológicas y un protocolo de acceso al medio ajustado a las necesidades.

### 3. Desarrollo

#### 3.1. Diagrama del Sistema

A continuación se muestra un diagrama conceptual del sistema propuesto, el desarrollo parte desde el diseño del LSU, pasando por el procesado y culmina con la entrega de los datos cargados a el servidor local.

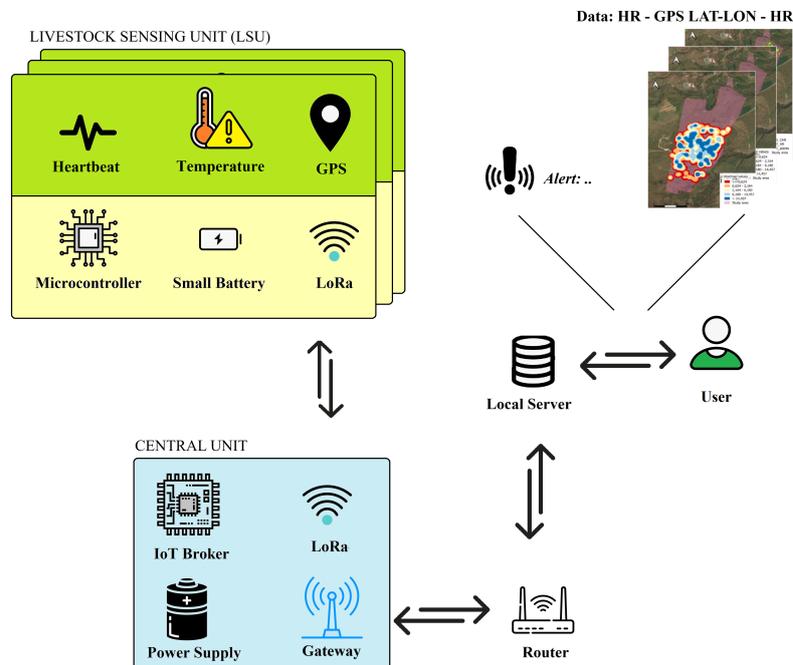


Figura 1: Diagrama ilustrativo del sistema

#### 3.2. Metodología y enfoque

Se adoptó una metodología iterativa: prototipado rápido en protobard, pruebas controladas y posterior ajuste de hardware y firmware. Se gestionó el proyecto con control de versiones (*git*), mientras que la planificación y seguimiento de tareas se realizaron con JIRA. Se planificó el sistema separando distintas features, diseñando el código de manera jerárquica y modular.

- [LSU - Github Repository](#)
- [CU - Github Repository](#)
- [CU WEB Page - Github](#)

#### 3.3. Selección de componentes y presupuesto energético

Se desarrollaron dos dispositivos: Livestock Sensing Unit (LSU) y Central Unit (CU), para cada uno de ellos se determinaron los módulos y componentes que se necesitaban integrar, y se seleccionaron los más adecuados a nuestras necesidades. Los componentes se eligieron priorizando, en orden: (i) bajo consumo en reposo, (ii) disponibilidad y costo razonable, y (iii) simplicidad de integración (drivers, librerías).

Componente / Modelo	Cant.	Proveedor sugerido
MCU STM32L031K6 (Nucleo-32)	1	Digi-Key (NUCLEO-L031K6)
Sensor de Temperatura MCP9700A-E/TO	2	Digi-Key
Sensor PPG MAX30102	1	Mercado Libre AR
GPS NEO-6M	1	Mercado Libre AR
Módulo LoRa RYLR998	1	Digi-Key
Batería Li-SOCl <sub>2</sub> EEMB ER14505	1	Mercado Libre AR
PCB	1	JLCPCB
R,C,Diodos,Headers, Sockets	-	Compra Local

Cuadro 1: Componentes principales de la LSU

Componente / Modelo	Cant.	Proveedor sugerido
DevKit ESP32-WROOM-32	1	Digi-Key
Módulo LoRa RYLR998	2	Digi-Key
Fuente SMPS 5 V / 2.5 A "Gralf SW-5V2.5USB"	1	Mercado Libre AR
Headers, Sockets	-	Compra Local
PCB	1	JLCPCB

Cuadro 2: Componentes principales de la CU

### Estimación de consumo de LSU en condiciones ideales de funcionamiento

Para la LSU se estimó un consumo total de 570 mAh para 6 meses de operación, con un período de muestreo/transmisión de 30 min. De forma simplificada:

$$I_{avg} = \frac{I_{active} \cdot t_{active} + I_{sleep} \cdot t_{sleep}}{T} \quad (1)$$

donde  $T$  es el período total (30 min),  $t_{active}$  el tiempo energizado de sensores y módulos, y  $t_{sleep}$  el tiempo que el dispositivo está en bajo consumo. El consumo en la LSU se puede condensar en la siguiente tabla y es importante para dimensionar la energización por baterías:

Componente	Corriente Media ( $I_{avg}$ )	Corriente máxima	Rango de funcionamiento	Capacidad Requerida
MCP9700	1 $\mu$ A	6 $\mu$ A	2.3V a 5.5 V	4,32 mAh
MCP9700	1 $\mu$ A	6 $\mu$ A	2.3V a 5.5 V	4,32 mAh
MAX30102	4 $\mu$ A	0.6 mA	1.8 V a 3.3V	17.41 mAh
NEO-6M	125 $\mu$ A	37 mA	3V a 4.3V	539 mAh
RYLR998	15 $\mu$ A	130 mA	2.3V a 3.6V	6,47 mAh
STM32L0	1 $\mu$ A	140 $\mu$ A	1.65 V a 3.6 V	4,84 mAh
<b>Total</b>	0.147 mA	-	-	570,36 mAh

Cuadro 3: Consumo de energía de los componentes de la LSU

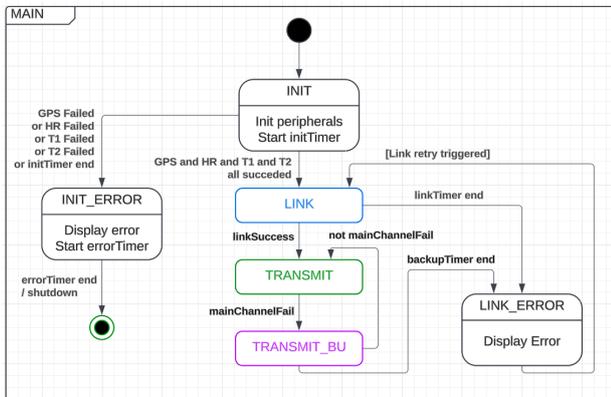
### 3.4. Implementación de la LSU

#### Arquitectura de software

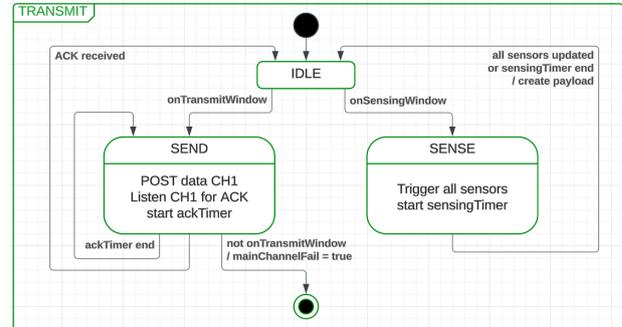
Sobre STM32CubeIDE se implementó una **máquina de estados jerárquica** (HFSM) que organiza el ciclo de vida del nodo, donde podemos destacar tres modos de funcionamiento principales:

- LINK: sincronización inicial con la CU, recepción de parámetros (ID, período, ventana TDMA).
- TRANSMIT: adquisición de sensores, empaquetado y transmisión de datos en su *slot*.
- TRANSMIT\_BU: reintento de transmisión mediante un canal auxiliar, cuando el canal principal falla.

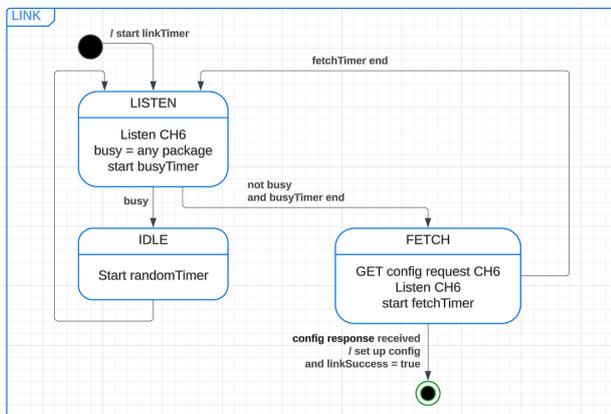
Las transiciones se disparan por eventos (timeout RTC, confirmación de enlace, error de sensor). Un *driver* genérico abstrae la interfaz LoRa (AT commands del RYLR998), mientras que los sensores se encapsulan en módulos independientes (`sensor_temperature.c`, `sensor_hearttrate.c`, etc.), minimizando dependencias.



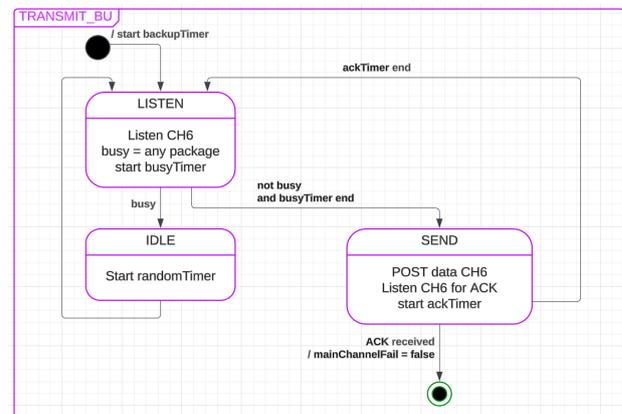
(a) HFSM Main



(b) FSM Transmit



(c) FSM Link



(d) FSM Transmit Backup

Figura 2: Máquinas de Estados Finitas utilizadas

### 3.5. Pruebas de Sensado LSU

A continuación se desarrolla brevemente lo implementado en cada uno de los sensores del nodo bovino.

#### Sensado de Ritmo Cardíaco

El módulo se conecta al microcontrolador mediante I2C usando los pines PB6 y PB7, mientras que el pin PA8 se empleó para recibir interrupciones generadas cuando la memoria FIFO del sensor se llenaba con datos.

Se trabajó con una resolución de 16 bits y se evaluaron distintas tasas de muestreo para capturar correctamente la señal del pulso. Tras capturar los datos, se aplicó un *downsample* y se utilizaron ventanas deslizantes para detectar máximos locales, permitiendo así calcular un estimado del ritmo cardíaco en BPM.

### Estimación de la frecuencia cardíaca.

Se estimó el cálculo de la siguiente manera:

$$\text{bpm} = \frac{T_{\text{peaks}} \cdot n_{\text{peaks}}}{60 \text{ s}}$$

donde  $T_{\text{peaks}}$  es el tiempo entre picos y  $n_{\text{peaks}}$  es el número de picos en el buffer.

El objetivo es evitar truncar picos durante el cálculo.

Para las pruebas, dado que no se dispone de un bovino, se opta por utilizar el pulso de una persona.

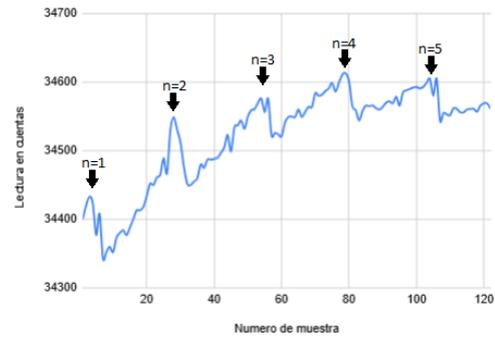


Figura 3: Ejemplo de buffer de datos de HR

### Pruebas de máxima distancia módulo LoRa

Si bien el protocolo admite funcionar hasta 15 km en campo abierto, existen pruebas independientes que reportan hasta 22 km, siempre que las antenas mantengan visión directa. Esto resalta la importancia de ubicar la unidad central en una posición elevada y sin obstrucciones para maximizar el alcance.

En nuestras prueba práctica, realizada en un entorno semiabierto y sin altura adicional, se configuraron dos escenarios con los siguientes parámetros:

Parámetros	Prueba 1	Prueba 2
Central Freq	915 MHz	915 MHz
Spreading Factor	9	11
Bandwidth	125 kHz	500 kHz
CR	4/5	4/7*
Preamble programado	12	12
RF Output Power	22 dBm	22 dBm

Cuadro 4: Configuraciones utilizadas

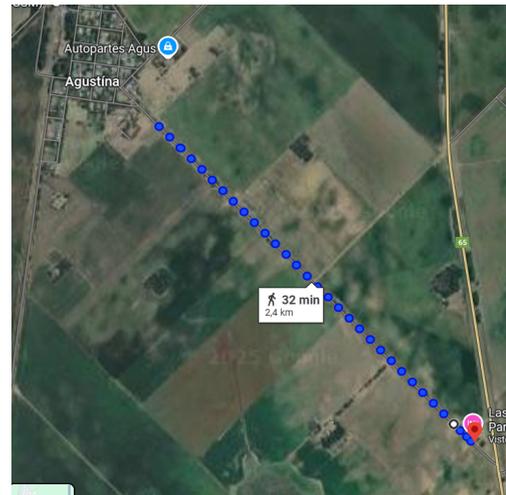


Figura 4: Distancia máxima de comunicación entre dos transmisores

\* Se modificó el Coding Rate en la Prueba 2 dado que la mayoría de los paquetes generaban un error +ERR=12, el cual corresponde a una falla con el CRC.

De estas pruebas, se encontró que el mensaje recibido a la distancia máxima se dio en la Prueba 1 a 2.4 km. En cambio, para la Prueba 2 se encontró el máximo en 2 km.

Si bien el resultado es menos satisfactorio del esperado, entendemos que las circunstancias del medio generan este comportamiento y consideramos que el rango resultó suficiente para el alcance de este trabajo.

### Sensores GPS y temperatura

En cuanto al módulo GPS, la comunicación se estableció mediante el protocolo UART. Si bien el dispositivo, una vez energizado, ya comienza a transmitir datos, fue necesario realizar una preconfiguración utilizando un módulo USB-to-TTL. Esto se debe a que el mismo transmite tramas con formato NMEA y resulta costoso procesar.

```

..... R -> NMEA GPGLL, Size 20, 'Geographic Position - Latitude/Longitude'
??:?:?? R -> NMEA GPRMC, Size 25, 'Recommended Minimum Specific GNSS Data'
??:?:?? R -> NMEA GPVTG, Size 21, 'Course Over Ground and Ground Speed'
??:?:?? R -> NMEA GPGGA, Size 33, 'Global Positioning System Fix Data'
??:?:?? R -> NMEA GPGSA, Size 45, 'GNSS DOP and Active Satellites'
??:?:?? R -> NMEA GPGSV, Size 18, 'GNSS Satellites in View'

```

Figura 5: Mensajes NMEA

Entonces, dado que es posible, se configura el GPS de manera que mande únicamente dos tramas: POSLLH y STATUS, las cuales contienen las variables de interés: LAT, LON, y Fixed.

```

??:?:?? R -> UBX NAV-STATUS, Size 24, 'Navigation Status'
??:?:?? R -> UBX NAV-POSLLH, Size 36, 'Geodetic Position'
??:?:?? R -> UBX NAV-STATUS, Size 24, 'Navigation Status'
??:?:?? R -> UBX NAV-POSLLH, Size 36, 'Geodetic Position'
??:?:?? R -> UBX NAV-STATUS, Size 24, 'Navigation Status'
??:?:?? R -> UBX NAV-POSLLH, Size 36, 'Geodetic Position'

```

Figura 6: Mensajes Configurados

Sin entrar en más detalle, se utiliza el entero Fixed, para verificar que el dato de latitud y longitud es válido.

En cuanto al sensor de temperatura, se utilizó el ADC para transformar la señal analógica en cantidad de cuentas. Se optó por implementar dos sensores, uno estaría ubicado en contacto con el animal y el otro mide la temperatura del entorno, con el objetivo de detectar una diferencia entre ambos.

### 3.6. Implementación de la CU

#### Arquitectura con FreeRTOS

La CU se implementó sobre un ESP32 (ESP-IDF + FreeRTOS) con una arquitectura orientada a tareas y paso de mensajes. El flujo general es:

1. Inicialización de periféricos y conectividad,
2. Recepción de tramas LoRa en dos canales UART independientes,
3. Encolado de solicitudes en una *request queue* centralizada,
4. Orquestado de solicitudes de enlace de LSU
5. Procesamiento y reenvío de datos a un servidor vía MQTT.

#### Tareas y responsabilidades

Tarea	Función
server_connection_task	Conecta a la WLAN, abre sesión MQTT y mantiene el enlace (retries, watchdog)
rx_channel_task (main)	Lee tramas del LoRa en el canal principal, valida y las encola como Request
rx_channel_task (aux)	Igual al anterior, pero para el puerto auxiliar (back-up, alta nodos, etc.)
process_requests_task	Desencola las Request y las procesa, arma payload y publica vía MQTT

Cuadro 5: Tareas FreeRTOS de la CU y responsabilidades principales

#### Capas de comunicación

**LoRa (uplink/downlink).** Cada `rx_channel_task` gestiona un RYLR998 por UART. Las tramas entrantes se parsean y se transforman en Request antes de encolarlas. Para *downlink* (acks, configuración) se aprovecha la misma capa `rylr998`, invocada desde `process_requests_task` o tareas auxiliares.

**Wi-Fi + MQTT.** La CU se conecta a una WLAN, utilizando el módulo Wi-Fi integrado en la ESP32. En esa misma red corren el broker MQTT y el servidor web. La `server_connection_task` inicializa Wi-Fi, establece la sesión MQTT y expone una API interna (`mqtt_api_publish()`) para publicar los datos procesados. Se contemplan reconexiones automáticas ante pérdida de enlace.

### Visualización local y diagnóstico

Se integró adicionalmente un OLED monocromo, que provee *feedback* en el sitio: mensajes de arranque, estado de conexión y cantidad de unidades conectadas. Los logs (`ESP_LOGI/W/E`) facilitan la depuración en desarrollo.

### Resumen del flujo de datos

1. LSU transmite su paquete LoRa en el *slot* asignado.
2. El RYLR998 (main/aux) lo recibe y lo entrega por UART.
3. `rx_channel_task` lee la UART, valida y `post_request()` a la cola.
4. `process_requests_task` extrae la solicitud, clasifica (*SYNC* o *DATA*) y genera el payload.
5. `mqtt_api_publish()` envía a la nube; el servidor web consume y visualiza.
6. `process_requests_task` envía la respuesta correspondiente al LSU mediante el mismo canal en que recibió la solicitud

Esta arquitectura modular facilita el escalado (nuevos tipos de mensajes o más canales LoRa) y simplifica el mantenimiento: cada capa está delimitada por interfaces claras y una cola centralizada garantiza el desacoplamiento temporal entre recepción y procesamiento.

## 3.7. Protocolo de comunicación

### Arquitectura de comunicación

Se utilizan dos canales LoRa para la comunicación. En el canal principal (CH-1) se transmiten los paquetes de datos sensados desde cada LSU a la CU, se utiliza TDMA para la multiplexación del canal. El canal auxiliar (CH-AUX) se utiliza para el enlace de nuevos LSU al sistema y para la retransmisión de datos, en caso de que el canal principal falle, para ello se configura CH-AUX para entregar una mayor potencia de señal, a coste de mayor consumo de energía

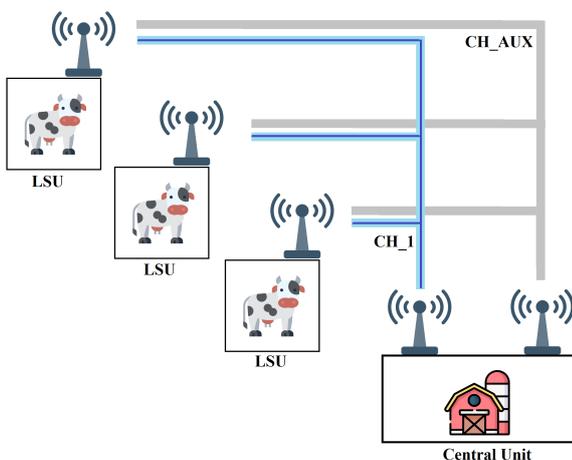


Figura 7: Comunicación via 2 canales

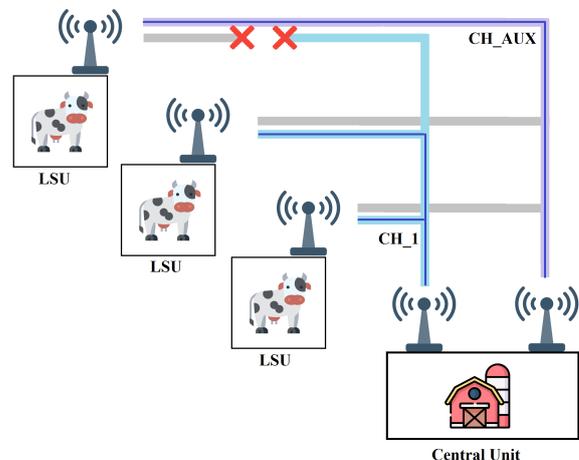


Figura 8: Fallo en una de las líneas

En este proyecto se implementó un único canal de datos (CH-1), pero la solución se puede escalar hacia varios canales para dar soporte a un mayor número de LSU.

## Enlace de LSU

Cuando se enciende un nuevo LSU, este envía un pedido de sincronización a la CU, al cual esta responde con un paquete de configuración, notificando acerca del período de sensado y asignándole un identificador y una ventana de tiempo de transmisión.

## TDMA en CH-1

La CU asigna a cada LSU un **slot temporal** dentro de una ventana TDMA global de duración 30 minutos. Cada LSU despierta, transmite en su slot y vuelve a dormir. Esto elimina colisiones entre nodos registrados y, **teóricamente**, permite escalar hasta 100 dispositivos si se le asigna una ventana de 18 segundos a cada LSU.

## CSMA en CH-AUX

Para mensajes fuera de agenda (alta de nuevos nodos, alertas o retransmisiones fuera de período), se utiliza el canal auxiliar, en el cual se implementa **CSMA/CA**: el nodo escucha el canal y transmite si detecta el medio libre, si el medio está ocupado, calcula un *backoff* aleatorio y espera para retransmitir.

## Análisis de espectro

Con el objetivo de verificar que el módulo transmite dentro de la banda de frecuencia correspondiente, sin distorsionar ni generar interferencias en otras bandas, se utilizó un analizador de espectro. Mediante este instrumento, se realizó un barrido en distintas frecuencias centrales.

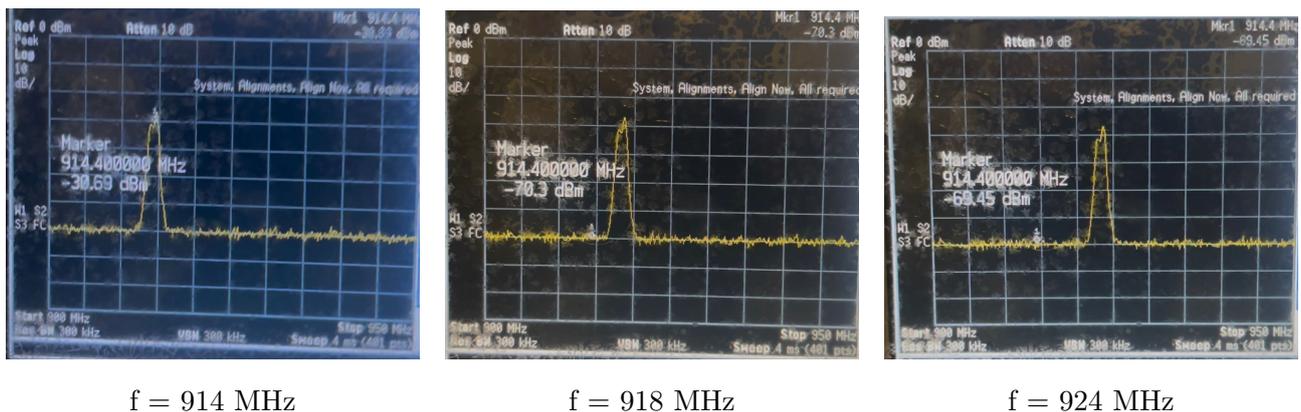


Figura 9: Algunas frecuencias utilizadas durante la verificación de transmisión

Los resultados obtenidos permiten confirmar que el módulo opera dentro del rango especificado, sin emitir señales fuera de banda. Esto valida su capacidad para transmitir en múltiples canales sin interferir entre ellos.

## 3.8. Herramienta web de visualización

Se desarrolló una herramienta web que muestra la posición y el mediciones recibidas de cada animal. Está implementado en **React/TypeScript**, y suscripción al broker MQTT.

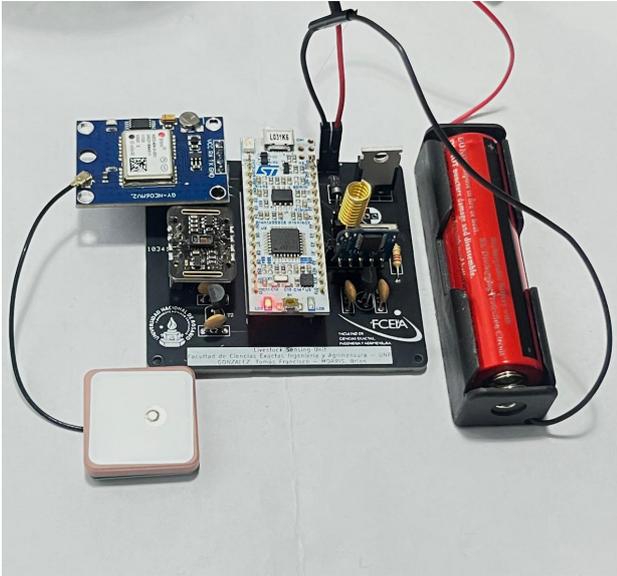


Figura 10: Simulación del dashboard de eventos recibidos

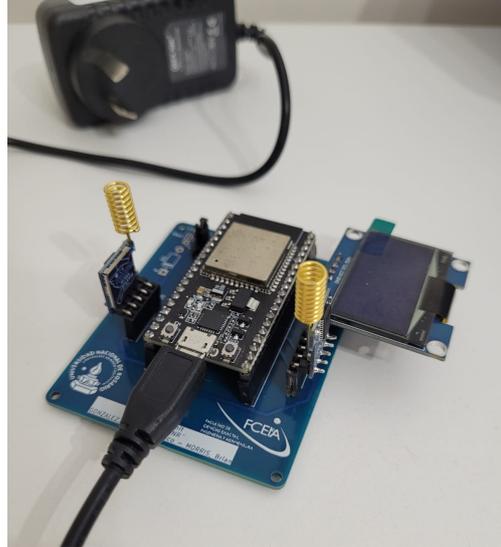
## 4. Resultados

### 4.1. Prototipos implementados

A continuación se muestran los prototipos utilizados para la implementación.



(a) Livestock Sensing Unit (LSU) – prototipo



(b) Central Unit (CU) – prototipo

### 4.2. Análisis y conclusiones

Tras validar previamente el correcto funcionamiento de los sensores (temperatura, PPG y GPS) y del enlace LoRa punto a punto, se avanzó con pruebas de escalamiento y coordinación temporal:

- **Multiplicación de nodos:** Se conectaron simultáneamente hasta **tres LSU**, verificando que cada uno recibiera su *slot* TDMA asignado y que los paquetes fueran recibidos sin colisiones. Los datos se visualizaron correctamente en la interfaz web de administración.
- **CSMA en ventana de contención:** Se probó el comportamiento del mecanismo CSMA/CA bajo canal ocupado, observando (en modo *debug*) el *backoff* aleatorio y la retransmisión posterior cuando el medio quedó libre.
- **Integridad de la cadena completa:** Desde el sensado local hasta la publicación en MQTT/TLS, el flujo funcionó de extremo a extremo. La CU procesó las tramas entrantes, las encoló y publicó sin pérdidas significativas.

Según las estimaciones energéticas y el diseño del protocolo, el sistema es **escalable teóricamente hasta ~100 LSU** con el mismo mecanismo de slots y ventana de contención. No obstante, las pruebas a gran escala no se realizaron por limitaciones de recursos (cantidad de nodos y hardware adicional).

**Limitaciones actuales.** El prototipo usa placas no optimizadas para producción. Tampoco se efectuaron ensayos prolongados (meses) en campo abierto que confirmen la autonomía de seis meses bajo condiciones reales (temperatura, movimiento, pérdidas de enlace).

### 4.3. Trabajo futuro

- **Integración en PCB final:** Si bien se intentó realizar el diseño de un PCB, no sentimos oportuno realizarlo dadas las dificultades que esto lleva, por lo que consolidar la LSU en una única placa sería el siguiente paso.

- **Diseño e implementación del empaquetado del LSU:** De la mano con la propuesta anterior, restaría diseñar el equipo final, integrando la placa con piezas físicas. De esta manera se podrá realizar una prueba sobre ganado.
- **Validación de autonomía a largo plazo:** pruebas continuas de meses con baterías para verificar la duración real y ajustar duty cycles.
- **Escalamiento de la red:** evaluar el desempeño con decenas de nodos ( $\geq 100$ ) y medir colisiones, latencias y uso efectivo del canal.
- **Mejoras de la plataforma web:** gráficos comparativos, alertas configurables, exportación de datos y autenticación robusta multiusuario.
- **Certificaciones y análisis económico:** evaluar normativas (radio, seguridad eléctrica) y elaborar un costo objetivo de producción por unidad.

## Referencias

- [1] INDEC. “CNA - Censo Nacional Agropecuario 2018.” (2018), dirección: <https://www.indec.gov.ar/indec/web/Nivel4-Tema-3-8-87>.
- [2] W. Boonsong, A. Sri-Ngernyuang y S. Boonsong, “Animal Identification and Performance Analysis Based on RFID-IoT Smart Farming Applications,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, n.º 2, págs. 152-158, 2022.
- [3] Contexto Ganadero. “¿Es necesario controlar la temperatura de los animales?” (2024), dirección: <https://www.contextoganadero.com/ganaderia-sostenible/es-necesario-controlar-la-temperatura-de-los-animales>.
- [4] R. E. Galarza Aguirre y F. G. Gutiérrez Muela, “Relación entre la temperatura corporal posparto y el comportamiento reproductivo en vacas,” Escuela Agrícola Panamericana, Zamorano, inf. téc., 2015.
- [5] C. Aubron, *Smart dairy farming: The impact of innovation on productivity and environment*, <https://hal.science/hal-01512238/document>, 2017.
- [6] S. Neethirajan, “Review: Use of infrared thermography in livestock animals and its applications,” *Asian-Australasian Journal of Animal Sciences*, vol. 32, n.º 7, págs. 1097-1110, 2019. DOI: [10.5713/ajas.18.0743](https://doi.org/10.5713/ajas.18.0743).
- [7] F. Diaz. “Strategies for Managing Heat Stress in Ruminants.” (2023), dirección: <https://www.agproud.com/articles/57303-strategies-for-managing-heat-stress-in-ruminants>.
- [8] A. M. K. Yoshida y M. M. C. Vincent, “Infrared thermography: A non-invasive window into the emotional valence of animals,” *Physiology & Behavior*, vol. 92, n.º 3, págs. 372-379, 2007. DOI: [10.1016/j.physbeh.2007.01.015](https://doi.org/10.1016/j.physbeh.2007.01.015).
- [9] D. M. Broom, “Stress and animal welfare,” *Physiology & Behavior*, vol. 73, n.º 1-2, págs. 207-214, 2001. DOI: [10.1016/S0031-9384\(01\)00651-5](https://doi.org/10.1016/S0031-9384(01)00651-5).
- [10] República Argentina. “Ley N.º 14.346 — Malos Tratos y Actos de Crueldad a los Animales.” (1954), dirección: <https://www.argentina.gob.ar/normativa/nacional/ley-14346-153011/texto>.
- [11] Servicio Nacional de Sanidad y Calidad Agroalimentaria (SENASA). “Normativa - SENASA.” (2025), dirección: <https://www.argentina.gob.ar/senasa/normativa>.
- [12] Ente Nacional de Comunicaciones (ENACOM). “Bandas de uso compartido sin autorización.” (2025), dirección: [https://www.enacom.gob.ar/bandas-de-uso-compartido-sin-autorizacion\\_p680](https://www.enacom.gob.ar/bandas-de-uso-compartido-sin-autorizacion_p680).
- [13] O. Kullinger, “Use of RFID Technology and Information Systems for Cattle Identification and Tracking,” Bachelor’s thesis, Linköping University, 2015. dirección: <https://www.diva-portal.org/smash/get/diva2:831645/FULLTEXT01.pdf>.
- [14] KU Leuven Researchers, *Heart Rate Monitoring in Pigs Using Photo Plethysmography Technology*, <https://lirias.kuleuven.be/retrieve/547904>, 2015.
- [15] S. Mnguni, P. Mudali, A. M. Abu-Mahfouz y M. Adigun, *Performance evaluation of Spreading Factors in LoRa Networks*, <https://researchspace.csir.co.za/server/api/core/bitstreams/116d8b1d-a4c0-4ec3-8356-73434f058dd6/content>.
- [16] J. Cheng, W. Tang, C. Li, X. Luo y H. Wang, “A Review of Wireless Sensor Networks and Applications in Smart Cities,” *Sensors*, vol. 20, n.º 8, pág. 2291, 2020. DOI: [10.3390/s20082291](https://doi.org/10.3390/s20082291).
- [17] K. Betke, *The NMEA 0183 Protocol*, <https://www.tronico.fi/OH6NT/docs/NMEA0183.pdf>, Compilación no oficial de la especificación NMEA 0183, ago. de 2001. dirección: <https://www.tronico.fi/OH6NT/docs/NMEA0183.pdf>.
- [18] G. Callebaut, G. Leenders, J. V. Mulders, G. Ottoy, L. D. Strycker y L. V. der Perre, “The Art of Designing Remote IoT Devices—Technologies and Strategies for a Long Battery Life,” *Sensors*, vol. 21, n.º 3, pág. 913, 2021. DOI: [10.3390/s21030913](https://doi.org/10.3390/s21030913).
- [19] Digitanimal. “Livestock GPS Tracker and Monitoring System.” (2025), dirección: <https://digitanimal.com/>.
- [20] Queclink Wireless Solutions. “Livestock Monitoring Solutions.” (2025), dirección: <https://www.queclink.com/ganado/>.
- [21] Basto. “Basto - IoT Livestock Monitoring.” (2025), dirección: <https://www.basto.io/>.
- [22] K. Kolobe y P. Phatshwane, “Cattle Monitoring and Identification System Using Radio Frequency Identification (RFID) Technology,” *International Journal of Electronics and Communication Engineering (IJECE)*, vol. 14, n.º 4, págs. 134-138, 2020.
- [23] S. Serwin, *Radio Propagation and Antennas: A Non-Mathematical Treatment of Radio and Antennas*. Independently published, 2018. dirección: <https://books.google.com.ar/books?id=L-ildWAAQBAJ>.
- [24] Maxim Integrated, *MAX30100: Pulse Oximeter and Heart-Rate Sensor IC for Wearable Health*, <https://www.analog.com/media/en/technical-documentation/data-sheets/MAX30100.pdf>, 2014.
- [25] u-blox AG, *NEO-6 GPS Modules Hardware Integration Manual*, [https://content.u-blox.com/sites/default/files/products/documents/LEA-NEO-MAX-6\\_HIM\\_%28UBX-14054794%29\\_1.pdf](https://content.u-blox.com/sites/default/files/products/documents/LEA-NEO-MAX-6_HIM_%28UBX-14054794%29_1.pdf), 2011.
- [26] STMicroelectronics, *STM32L0 Series Ultra-low-power 32-bit MCUs*, <https://www.st.com/en/microcontrollers-microprocessors/stm32l0-series/documentation.html>.
- [27] *LoRa AT Command RYLR998 RYLR498*, [https://reyax.com/upload/products\\_download/download\\_file/LoRa\\_AT\\_Command\\_RYLR998\\_RYLR498\\_EN.pdf](https://reyax.com/upload/products_download/download_file/LoRa_AT_Command_RYLR998_RYLR498_EN.pdf), 2023.
- [28] S. Martiradonna, G. Acciani, A. Fornarelli, M. De Giorgi y A. Fasanella, “The Art of Designing Remote IoT Devices—Technologies and Strategies for a Long Battery Life,” *Sensors*, vol. 21, n.º 3, pág. 913, 2021. DOI: [10.3390/s21030913](https://doi.org/10.3390/s21030913). dirección: <https://www.mdpi.com/1424-8220/21/3/913>.
- [29] Semtech Corporation. “SX127x Datasheet.” (2025), dirección: <https://www.semtech.com/products/wireless-rf/lora-connect/sx1272>.