

# CMS: Cow monitoring system 2.0

Mateo Scozzina, Pablo Á. Toledo, Santiago E. Roatta, Alejandro R. Sartorio

Facultad de Tecnología Informática, Universidad Abierta Interamericana (UAI),  
Rosario, Argentina

{Scozzina}mateoscozzina@outlook.com  
{Toledo}pablotoledo1994@hotmail.com  
{Roatta}santiago.roatta@gmail.com  
{Sartorio}alejandro.sartorio@gmail.com

**Resumen.** Este trabajo presenta una solución mejorada a un problema de Ingeniería llevado a cabo en el marco de un ejercicio con temática optativa de la asignatura Seguridad Informática de la carrera Ingeniería en Sistemas Informáticos de la Universidad Abierta Interamericana (UAI). Se continuó el desarrollo del primer proyecto abordando en esta versión 2.0 la resolución a la problemática planteada en las conclusiones del mismo. Si bien se trata de un trabajo de cátedra, se ha decidido respetar el formato habitual de los “papers científicos” con las secciones tradicionales que componen a los mismos. En primer lugar, se presenta una breve introducción sobre el rubro agropecuario, más precisamente sobre el sector de la ganadería bovina, seguida de una problemática dada en el mismo, la cual es tratada aplicando soluciones con sistemas embebidos e Internet de las Cosas. A continuación, se describe el protocolo LoRaWAN que utilizamos en el proyecto. Y finalmente, se muestra un prototipo de nuestro desarrollo: un collar electrónico que permite el monitoreo tanto de la geolocalización como de la temperatura del ganado situado dentro de un campo, logrando una mayor durabilidad de batería que la expuesta en la primera versión.

**Palabras clave.** Ganado, Localización, Temperatura, IoT, LoRaWAN.

## 1 Introducción

El origen de la cría ganadera y el gusto por el consumo de carne de los argentinos es anterior incluso a la conformación del Estado Argentino. Las primeras vacas descendieron desde el epicentro del Virreinato del Perú hacia el actual territorio argentino a mediados del siglo XVI, expandiéndose en las amplias llanuras pampeanas gracias a las abundantes pasturas dadas en un clima bondadoso. El crecimiento de la hacienda bovina en Argentina permitió que, durante muchos años, el consumo de carne vacuna fuera muy barato, arraigándose así en la dieta de la población. Fueron los gauchos quienes en la pausa de su labor diaria dieron inicio al rito del asado, al reunirse para cocinar la carne al fuego y compartir la comida. Las grandes olas migratorias que re-

cibió posteriormente el país hicieron de la milanesa el plato de carne más consumido en la actualidad, seguido por el bife y la tira de asado.

Actualmente las cadenas bovina, avícola y porcina combinadas generan cerca del 30% del VBP (Valor Bruto de Producción) de las cadenas agroalimentarias en Argentina, y aportan el 23% de su Valor Agregado. Presentes en las 23 provincias, emplean a más de 400 mil personas [1].

Reconociendo el importante impacto en las economías regionales que tiene la cría de animales en el país, vale la pena analizar dónde se origina el VBP detallado. Según los datos del último CNA18 (Censo Nacional Agropecuario 2018), más de la mitad de las cabezas de ganado bovino destinadas a la producción de carne se encuentran en los territorios bonaerense, santafesino y correntino. Si a estas existencias se le suma la cría de bovinos en Córdoba, Entre Ríos, La Pampa, Chaco, San Luis y Formosa, se tiene el 90% de las cabezas bovinas argentinas [2].

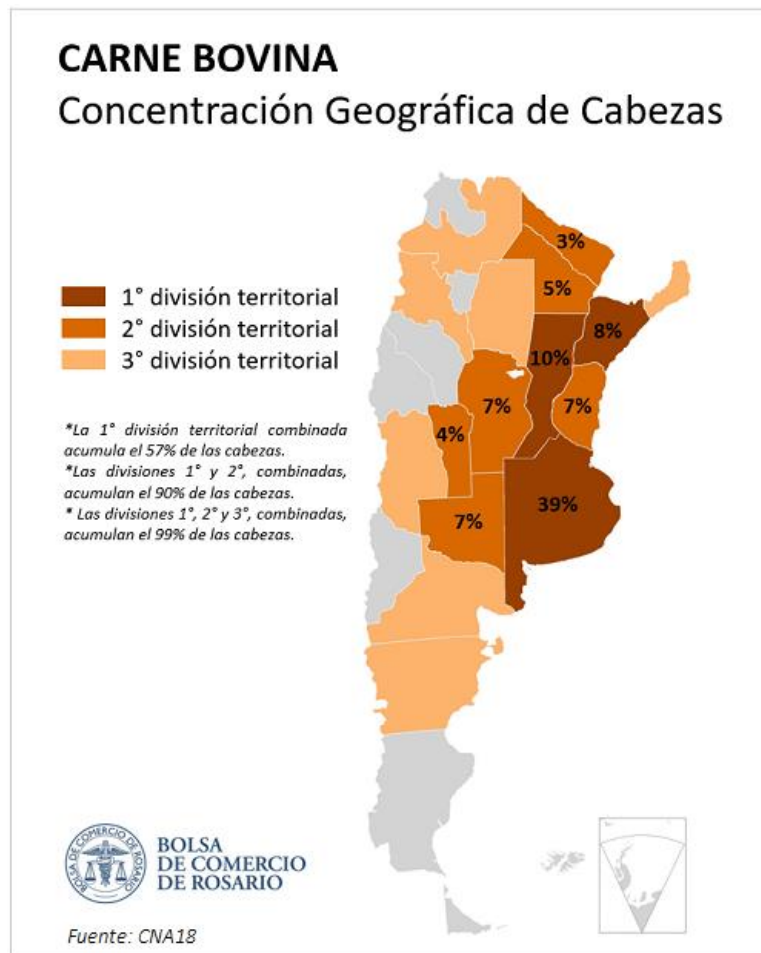


Fig. 1. Concentración geográfica de cabezas de ganado bovino en Argentina.

El MAGyP (Ministerio de Agricultura, Ganadería y Pesca de la Nación) informó que el stock bovino en Argentina al 31 de Diciembre de 2022 cerró en 54.242.595 cabezas totales, registrando un incremento anual de más de 826.000 cabezas respecto al cierre de 2021, lo que muestra un crecimiento del 1,6% del rodeo nacional [3].

Dados estos números e identificando a la ganadería bovina como un pilar elemental en la economía nacional, resulta la necesidad de numerosos propietarios de campos de poder controlar sus vacas. Generalmente ocurre que el dueño del campo reside a gran distancia geográfica del mismo, pero desea tener conocimiento continuo de dos principales situaciones que se pueden dar con su ganado: que fallezcan (por alguna razón desconocida) o que salgan de su perímetro (ya sea por alguna posibilidad eventual que encuentre el animal de salir o bien porque sea robado).

En este marco, se plantea la posibilidad de que el propietario de un establo pueda monitorear por remoto la **temperatura** y la **localización** del ganado, para poder chequear cuando lo desee si estos animales en pie para faena siguen vivos y dentro del perímetro establecido.

A través del desarrollo de una aplicación se muestra la relevancia que tienen actualmente los **SE (sistemas embebidos)** en nuestra vida cotidiana, los cuales son sistemas basados en un microprocesador que contienen hardware y software diseñados y optimizados específicamente para resolver un problema de una manera eficiente, buscando administrar el tamaño, costo, rendimiento y el uso de la energía (principalmente en sistemas que funcionen a batería sin estar conectados a la red eléctrica). En su arquitectura, los SE contienen un microprocesador que ejecuta las operaciones a cierta velocidad, controladas por una señal de reloj. Esta señal de reloj está condicionada por los recursos internos y la máxima frecuencia de operación, y se mide en MHz (megahertz).

En cuanto a la definición de un **microcontrolador**, el mismo se considera un SE, tratándose de un chip que contiene un sistema computarizado completo (microprocesador, unidades de memoria, unidades de entrada/salida y periféricos). Es importante también entender que los SE no son equivalentes a las computadoras de propósito general como notebooks o PCs de escritorio, ya que los primeros tienen recursos limitados y están diseñados exclusivamente para resolver una tarea específica, como son los casos por ejemplo de los sistemas de frenos ABS de los automóviles, los controles de climatización o las alarmas contra robos.

Este proyecto se sitúa dentro del marco de un concepto fundamental, que es el de **IoT (Internet of Things, “Internet de las Cosas”)**, el cual basa su significado en la esencia de la posibilidad de conectar los objetos que utilizamos en la vida cotidiana a Internet, siendo de gran importancia para marcar la primera evolución real de Internet, logrando que sea sensorial. IoT consiste en desarrollar un SE que tenga la capacidad de conectar el dispositivo a Internet a través de una dirección IP que le permitirá interactuar con el mundo exterior enviando y recibiendo datos [4].

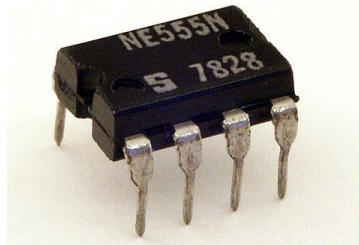
## 2 Análisis

Al pensar en las posibles complicaciones que podría generarnos el desarrollo del sistema de monitoreo, situamos dos grandes desafíos para el proyecto: la capacidad de duración de la batería de los dispositivos y el alcance de la señal de los mismos.

### Capacidad de duración de la batería de los dispositivos

Para lograr que el collar logre encenderse y apagarse por un determinado tiempo, se debe utilizar un dispositivo electrónico que nos facilite tal operación. Dentro de las opciones disponibles en el mercado, una muy buena alternativa resultó ser el circuito integrado 555, cuya principal función consiste en ser un temporizador configurable normalmente mediante dos principales maneras [5]:

- Modo monoestable: en este caso el pin de salida se pondrá en modo alto, es decir, enviará tensión y luego de un tiempo, dependiendo de la combinación de resistencias y capacitores, se pondrá en modo bajo (dejará de circular corriente a través de él). Este proceso no se repite, sino que requerirá que físicamente reciba un pulso eléctrico en la terminal trigger (disparador) para que vuelva a actuar como temporizador.
- Modo astable: si bien aquí el principio de funcionamiento es similar al caso anterior donde con una correcta configuración de componentes electrónicos pasivos (resistencias y capacitores) se logra determinar los tiempos de flanco alto (tensión positiva) y flanco bajo (sin tensión) en el pin de salida, a diferencia del modo monoestable, cuando termina el tiempo de flanco bajo el ciclo se reinicia solo, es decir, no necesita de la intervención de una acción externa.



**Fig. 2.** Circuito integrado 555.

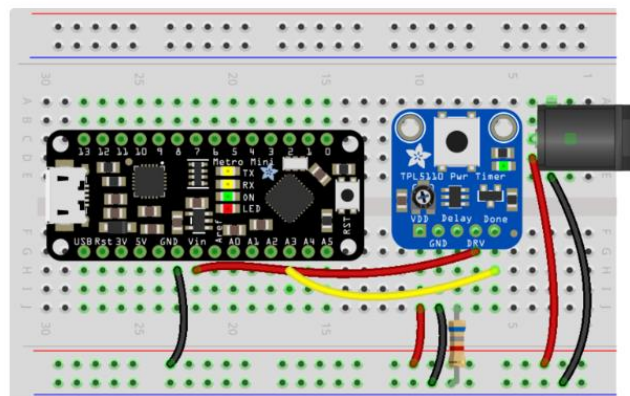
En principio se consideró para el proyecto optar por el modo astable ya que el mismo permite encender y apagar de manera autónoma el dispositivo que llevará cada animal. Las fórmulas que se utilizan para determinar los valores de resistencia y capacitor necesarios para parametrizar los tiempos de flanco alto y bajo son [6]:

- Tiempo encendido:  $0,693 * (\text{Resistencia A} + \text{Resistencia B}) * \text{Capacitor}$
- Tiempo apagado:  $0,693 * \text{Resistencia B} * \text{Capacitor}$

Analizando la ecuación, se llega a la conclusión de que no es posible avanzar con este circuito integrado ya que el valor del flanco alto siempre será mayor o igual que el flanco bajo, incluso hasta cuando se coloque 0 como valor de Resistencia A (utilizando por ejemplo un cable directo), y se necesita que ocurra lo contrario, es decir, que el flanco banco bajo sea menor que el alto.

En consecuencia, se investigaron otras alternativas y se descubrió el dispositivo TPL5110, que consiste es un temporizador electrónico de muy bajo consumo que permite establecer intervalos de salida en alto y bajo configurables entre 0,1 y 7200 segundos [7]. Estos intervalos se pueden configurar dependiendo del valor de resistencia que se le coloque entre el pin “Delay” y el negativo de la batería, que en nuestro caso será de 170K $\Omega$ .

Para apagarse, este temporizador necesita recibir un pulso eléctrico positivo en su terminal “Done”. Una vez que se hayan recolectado los datos de los sensores se envía la información al servidor, este confirma la recepción, y desde Arduino se envía un 1 hacia el “Done” para que el temporizador entre en modo reposo y se active nuevamente a las 2 horas [8].



**Fig. 3.** Esquema de conexiones del temporizador TPL5110 [9].

Finalmente se escogió la solución del temporizador TPL5110, ya que el mismo permite aumentar el rendimiento de la batería notablemente y con ello evitar el contacto constante con el ganado, eludiendo así la complejidad de la tarea y el estrés que genera en el animal tener que contenerlo, por ejemplo, para hacerle un cambio de batería al collar.

En una primera versión del proyecto se planteó esta solución basándose en la plataforma Arduino debido a que para las primeras pruebas de desarrollo la misma resulta muy amigable y de baja complejidad, por lo que cumplió los objetivos referentes a los testeos iniciales. Pero el problema que conlleva continuar el proyecto con la placa Arduino, lo cual fue mencionado en las conclusiones del proyecto inicial como un tema a tratar a futuro, radica en que la misma no es del todo eficiente en términos de consumo eléctrico.

En base a ello, en esta versión 2.0 se continuó con la investigación y se descubrió el modo de optimizar aún más la energía para que las baterías duren más tiempo, habiéndose adquirido conocimientos sobre cómo crear una placa electrónica a medida con el fin de ahorrar energía quitando componentes que trae Arduino por defecto que son innecesarios para los fines del proyecto, como por ejemplo leds, puertos USB y otros puertos que no se requieren para el caso.

Para poder lograr el posterior desarrollo de la placa, se analizó la asignación de pines del microcontrolador que utiliza Arduino UNO, el Atmel ATmega328, sobre el cual se basó el circuito electrónico optimizado que permite lograr mayor duración de la batería. A continuación, se ilustra en la figura 4 el pinout del microcontrolador:

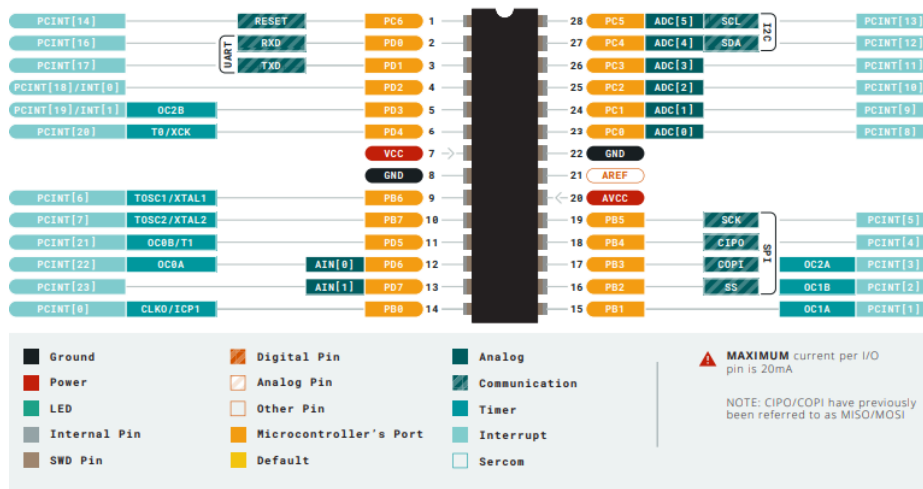


Fig. 4. Distribución de pines del microcontrolador Atmel ATmega328 [10].

### Alcance de la señal de los dispositivos

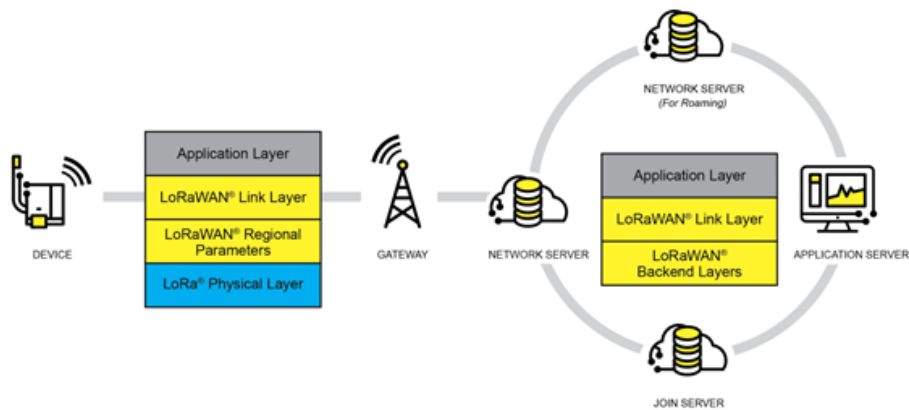
En cuanto al tema de la señal, se optó por la utilización de LoRaWAN debido a que se trata de una tecnología de transmisión inalámbrica a través de radiofrecuencia, óptima para el ámbito de IoT. Cuenta con grandes beneficios respecto al bajo consumo de energía eléctrica y al gran alcance de comunicación en comparación con otras tecnologías similares como 3G, WiFi, GPRS, Bluetooth, entre otras [11], [12]. El principal problema de esta tecnología radica en el bajo ancho de banda, aunque en la aplicación del proyecto no representa problema alguno.

LoRaWAN es un protocolo de comunicación basado en LoRa, creado por la asociación sin fines de lucro LoRa Alliance, cuyos miembros colaboran constantemente en su evolución. El estándar destinado para el proyecto se está adoptando exitosamente alrededor del mundo ya que es abierto, explotando todos los beneficios de LoRa, ideales para el ámbito de IoT. Esta tecnología está siendo transmitida por grandes operadores de redes móviles alrededor del mundo para ampliar el alcance de la misma. Otro importante beneficio que tiene radica en que trabaja con frecuencias abiertas, sin necesidad de abonar un licenciamiento por su utilización [13].

En cuanto a su funcionalidad, los principales casos de uso de aplicaciones que se pueden desarrollar con LoRaWAN y que muestran relevancia en nuestra vida cotidiana son: agricultura, edificios inteligentes, ciudades, industrias, logística [14].

### 3 Desarrollo

El sistema propuesto consiste en nodos (collares con GPS y sensor de temperatura integrados) colocados en los cuellos de los animales, que envían la información a un gateway y luego este la retransmite a un servidor centralizado para su posterior tratamiento. A continuación, se muestra en la figura 5 cómo se efectúa la comunicación entre los componentes:



**Fig. 5.** Arquitectura de red LoRaWAN [15].

Para el proyecto, el rol de “device” lo adquiere cada dispositivo colocado en los animales, los cuales enviarán los datos a través de una antena de RF mediante tecnología LoRa. El gateway recibe la información en un lugar del campo que cuente con energía eléctrica y conexión a Internet (puede ser el lugar de alojamiento del cuidador del establo) y desde allí la envía al servidor local para el procesamiento de los datos y posterior envío a un servidor web para poder levantarlo desde una aplicación móvil.

La función principal del dispositivo consiste en reportar cada 2 horas la temperatura y geoposicionamiento del ganado, y en esta versión 2.0 el desafío consistió en lograr que se pueda contar con una mayor durabilidad de la batería. Para ello, se logró diagramar un circuito electrónico basado en el microcontrolador de una placa Arduino UNO (un Atmel ATmega328) que pueda funcionar y cumplir los objetivos con los recursos electrónicos básicos necesarios [16].

A continuación, se ilustra en la figura 6 el diagrama que representa las conexiones del circuito electrónico que permite optimizar los recursos energéticos:

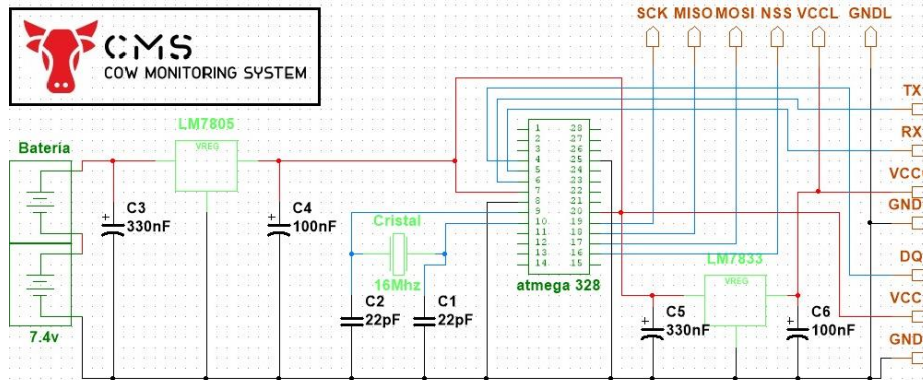


Fig. 6. Diagrama de conexiones electrónicas del circuito optimizado.

Este nuevo controlador 2.0 reemplaza al controlador Arduino, y para el desarrollo del prototipo se requieren dos de ellos, tal como en el primer proyecto. Ambos son necesarios para actuar de controladores de los sensores y de los módulos transceptores Lora; por ende, debe haber uno del lado del nodo y otro del lado del gateway.

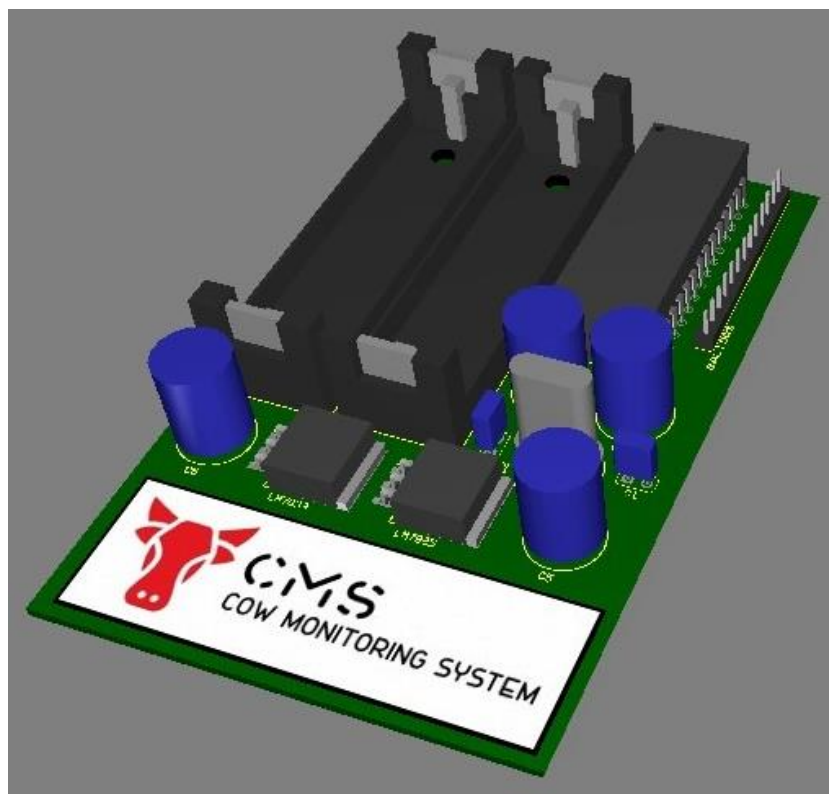


Fig. 7. Diseño en 3D del circuito electrónico optimizado.



En la figura 7 se pueden apreciar los componentes necesarios para el correcto funcionamiento de la placa, los cuales se explican en detalle a continuación:

- Placa PCB virgen: necesaria para la construcción del circuito impreso, tanto del nodo como del receptor.
- Baterías: para la confección del prototipo se pensó en baterías de Litio, ideales para este tipo de proyectos y comercialmente fáciles de conseguir. Poseen la limitación de que son de 3,7v, insuficientes para la alimentación del microcontrolador ATmega328, por lo que se deben colocar dos en serie (por ley de Kirchhoff, lo cual suma un total de 7,4v) para poder cumplir con la alimentación mínima requerida por el microcontrolador (5v).
- Microcontrolador ATmega328: para el desarrollo del prototipo se utilizaron estos microcontroladores ya que son robustos y muy conocidos en el mercado debido a ser utilizados por Arduino, lo que los hace ideales para este tipo de aplicación por la gigante comunidad que lo utiliza y la gran cantidad de información que hay al respecto [17].
- Regulador de tensión LM7805 y LM7833: son componentes electrónicos que garantizan que la tensión de salida sea siempre estable de 5v positivo, en el caso del LM7805 para el microcontrolador, o 3,3v positivo, en el caso del LM7833 para el sensor GPS y el módulo LoRa. Permiten garantizar que el microcontrolador y los demás componentes electrónicos reciban la tensión justa para poder trabajar correctamente y de esta manera evitar que se dañen los componentes [18]. En este proyecto necesitamos uno de cada uno; y, por otro lado, el fabricante del regulador indica en su Datasheet que para poder operarlo en un circuito necesita 2 capacitores electrolíticos (uno de 330 nF y otro de 100 nF), tal como se ilustra en el diagrama de conexiones en la figura 6.
- Capacitores (cerámicos y electrolíticos): componentes electrónicos pasivos que se utilizan para almacenar carga eléctrica en un campo eléctrico interno, y se aplican en circuitos electrónicos, digitales y analógicos. Para la placa 2.0 se utilizan del tipo cerámico (C1 y C2), para operar sobre el cristal, y del tipo electrolítico (C3, C4, C5 y C6), para actuar sobre los reguladores.
- Oscilador de cristal: elemento electrónico pasivo de un circuito electrónico generalmente utilizado en electrónica digital para generar una señal eléctrica con una frecuencia estable. El mismo se conecta en la placa a los pines 9 y 10 a través de los capacitores cerámicos.

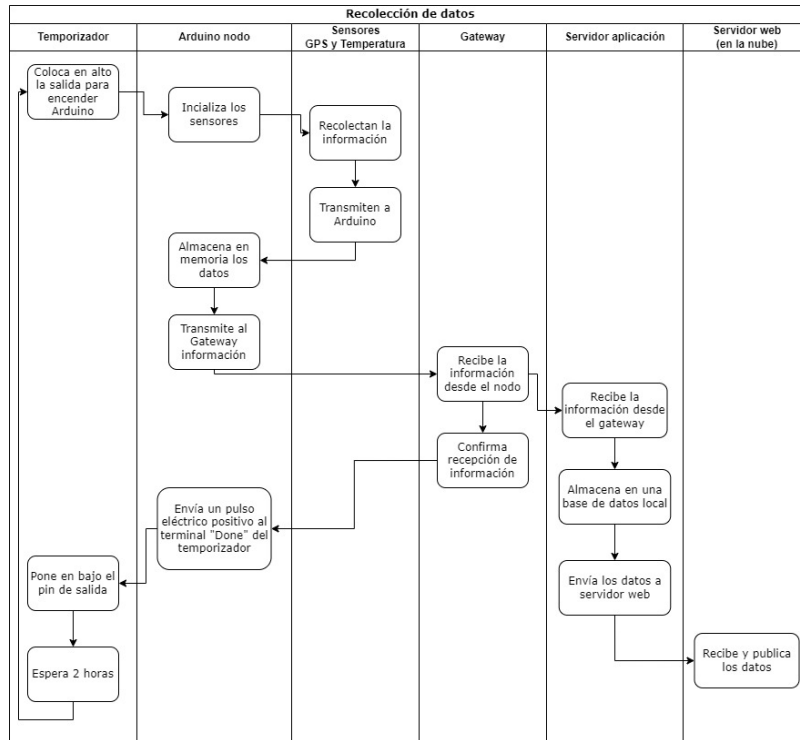
En complemento, el resto de los elementos a utilizar son los mismos que se han considerado en la primera versión del proyecto, donde el circuito estaba basado en la placa Arduino:

- Sensor de temperatura: para el desarrollo del prototipo se opta por el sensor DS18B20 ya que opera correctamente en el rango de temperatura necesario y es totalmente compatible con la electrónica utilizada para el desarrollo. Posee 2 pines de alimentación más uno para el envío de datos por serie [19]. En el

circuito electrónico, este se conecta a los pines de salida DQ (datos), VCCS (positivo) y GNDS (negativo).

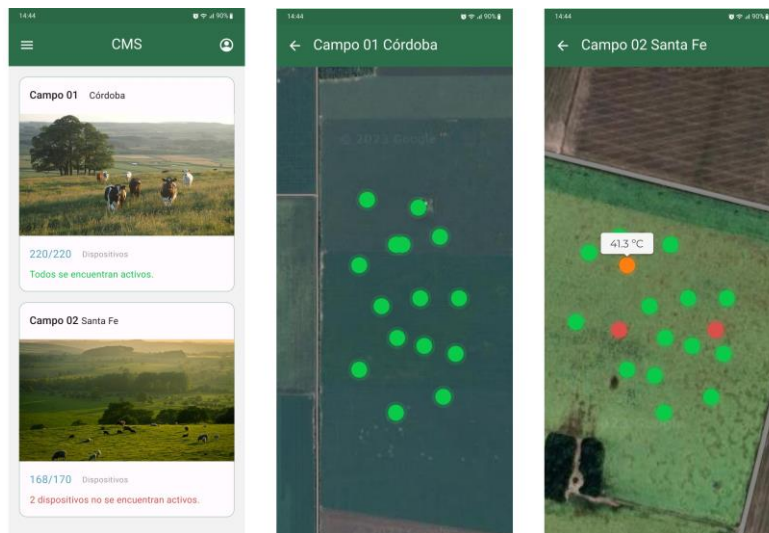
- GPS: se utiliza el módulo GY-NEO6MV2, el cual permite determinar cuál es la localización actual del animal, utilizado por ejemplo para el posicionamiento de dispositivos de vuelo como es el caso de drones o aviones RC (radio control). Es ideal para eficientizar la autonomía de las baterías debido a su bajo consumo y fina precisión, ya que puede comunicarse con hasta 22 satélites para determinar con exactitud la posición actual [20]. Este componente se conecta en la placa a través de los pines de salida TX (transmisión), RX (recepción), VCCG (positivo) y GNDG (negativo).
- Módulo transceptor LoRa: para poder establecer la comunicación entre el gateway y los nodos finales, se requiere de un transceptor LoRa que sea capaz de establecer esta comunicación a través del protocolo LoRaWAN. Para ello se eligió el transceptor SX1278 con una frecuencia de 433MHz, con gran inmunidad a la interferencia y bajo costo. Junto con el módulo, se adquiere una antena de RF helicoidal para los 433MHz necesaria para cubrir un buen rango de distancia. Será necesario contar con uno tanto para el lado del nodo como así también para el lado del gateway [21]. Este módulo se conecta al circuito mediando los pines de salida SCK (datos), MISO (datos), MOSI (datos), NSS (datos), VCCL (positivo) y GNDL (negativo).
- Temporizador: para el proyecto se emplea el temporizador TPL5110, debido a las grandes ventajas que brinda en cuanto al aumento de rendimiento de la batería.
- Servidor aplicación: se ubica detrás del gateway y es el encargado de transmitir hacia el servidor web alojado en la nube toda la información recolectada de los animales. Para el mismo es necesario disponer de una PC con Windows 10 o Linux, 4GB de RAM, procesador de 2 núcleos, 10GB de espacio en disco disponibles y conexión a Internet. Es necesario que contenga un motor de base de datos relacional para el almacenamiento de los datos.

A continuación, se muestra gráficamente en la figura 8 la lógica de funcionamiento de la recolección de datos:



**Fig. 8.** Lógica de funcionamiento de recolección de datos.

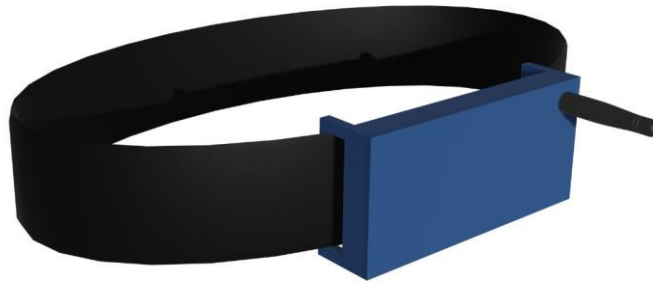
Por último, en la figura 9 pueden verse las pantallas de la aplicación móvil que utilizará el propietario del establo para poder monitorear el ganado:



**Fig. 9.** Vistas de la aplicación móvil.

Cuando el usuario ingresa en la aplicación, en primer lugar el mismo debe seleccionar el campo que desea monitorear y acto seguido se despliega una imagen satelital de Google Maps del campo. Con puntos verdes se identifica el ganado cuyo nodo responde correctamente, es decir, aquellos que no presentan anomalías en materia de temperatura y localización. En el caso que presente problemas de temperatura, el nodo se representará de color anaranjado. Y por último, si el dispositivo no responde en un rango de 6 horas (no reportó durante 3 ciclos), el mismo se pondrá de color rojo en su última ubicación reportada.

Para simular el proyecto inicial, se muestra en la figura 10 un modelo del collar diseñado en 3D:



**Fig. 10.** Imagen del prototipo diseñado en 3D.

## 4 Conclusiones

Para los dueños de establos resulta de máxima importancia poder tener un control de sus activos, principalmente cuando residen lejos del campo, incluso hasta en otra provincia. Los propietarios necesitan poder detectar rápidamente si el ganado fue robado como también tener información relevante actualizada acerca de posibles enfermedades que pueda tener el animal, como por ejemplo fiebre.

Con el avance de la tecnología y las posibilidades que brindan las aplicaciones de IoT, se puede lograr hacer llegar al propietario del campo esta valiosa información a través de un sistema de monitoreo y una aplicación móvil, permitiéndole al mismo poder eficientizar el cuidado y gestión de su ganado.

En cuanto a la complejidad del proyecto presentamos dos desafíos a resolver (duración de la batería y alcance de la señal), para los cuales resultó muy complejo encontrar alternativas viables que puedan resolverlos, pero luego de investigar se pudieron hallar soluciones realizables.

En la primera versión del proyecto, para poder realizar el prototipo y las primeras pruebas de funcionamiento, se había considerado que Arduino era ideal para la tarea ya que posee una gran comunidad que da soporte y una amplia cantidad de librerías que facilitan el desarrollo. De todos modos, si bien Arduino fue viable para la etapa de

pruebas del primer proyecto, el mismo resultó muy poco eficiente para utilizarlo en la práctica ya que consume demasiada energía, principalmente en componentes innecesarios para el sistema (por ejemplo, la interfaz USB, leds testigos de funcionamiento, etc.), y en este tipo de proyectos que requieren reducir al extremo el consumo de corriente, perdería eficiencia.

Por ello, en las conclusiones de la versión anterior se planteó resolver a futuro el problema de la duración de la batería y en esta versión 2.0 se logró optimizar la electrónica empleada utilizando un microcontrolador Atmel ATmega328 que permitió administrar más eficientemente los recursos electrónicos permitiendo así obtener una mayor durabilidad de la batería del dispositivo, e incluso también permitiendo reducir su costo y su tamaño.

## Referencias

1. BCR (Bolsa de Comercio de Rosario): La importancia de la ganadería para la economía argentina. Sitio web de BCR (2021).
2. INDEC (Instituto Nacional de Estadística y Censos de la República Argentina): CNA18 (Censo Nacional Agropecuario 2018). Portal oficial del INDEC (2019).
3. MAGyP (Ministerio de Agricultura, Ganadería y Pesca de la Nación): Existencias de bovinos al 31 de Diciembre de 2022. Portal oficial del Estado Argentino (2023).
4. Dave Evans: Internet de las Cosas. Cómo la próxima evolución de Internet lo cambia todo. Informe técnico de Cisco (2011) 5-7.
5. Md. Moyeed Abrar: Design and Implementation of Astable Multivibrator using 555 Timer. Trabajo de investigación publicado por Department of Computer Science & Engineering, Khaja Banda Nawaz College of Engineering, Kalaburagi, Karnataka, India (2017) 1-4.
6. Diodes Incorporated: NE555 PRECISION TIMER. Datasheet publicado por Diodes Incorporated (2021) 7-8.
7. Adafruit: Adafruit TPL5110 Low Power Timer Breakout. Sitio web de Adafruit (2022).
8. Texas Instruments Incorporated: TPL5110 Nano-Power System Timer for Power Gating. Datasheet publicado por Texas Instruments Incorporated (2018) 11-14.
9. Adafruit: Usage Adafruit TPL5110 Power Timer Breakout. Sitio web de Adafruit (2022).
10. Arduino: ATmega328P Pin Mapping. Sitio web de Arduino (2023).
11. Semtech: Why LoRa?. Sitio web de Semtech (2023).
12. Semtech: Ultimate Network Comparison. Sitio web de Semtech (2023).
13. Catsensors: Tecnología LoRa y LoRaWAN. Sitio web de Catsensors (2023).
14. CABASE: LoRaWAN Casos de uso. Sitio web de CABASE (2019).
15. LoRa Alliance: ¿Qué es la especificación LoRaWAN?. Sitio web de LoRa Alliance (2023).
16. Arduino: UNO-TH\_Rev3e. Sitio web de Arduino (2019).
17. ATMEL Corporation: Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash. Datasheet publicado por Alldatasheet (2010) 2-7.
18. Thinki Semiconductor Co. (Shanghai): (Shanghai). Datasheet publicado por Datasheet4U (2006) 1-2.
19. Dallas Semiconductor: DS18B20 Programmable Resolution 1-Wire Digital Thermometer. Datasheet publicado por Dallas Semiconductor (2008) 1-2.
20. Datasheethub: GY-NEO6MV2 Flight Control GPS Module. Sitio web de Datasheethub (2022).
21. Semtech: SX1276/77/78/79. Datasheet publicado por Semtech (2015) 9-10.