

PROJECT, DESIGN AND MANAGEMENT

ISSN: 2683-1597



Cómo citar este artículo:

Toledo, M. A. (2022). Seguimiento y control de ganado vacuno mediante geoposicionamiento. *Project, Design and Management*, 4(1), 91-113. doi: 10.35992/pdm.4vi1.828

SEGUIMIENTO Y CONTROL DE GANADO VACUNO MEDIANTE GEOPOSICIONAMIENTO

Marcelo Alejandro Toledo

Universidad de la Cuenca del Plata (Argentina)

toledomarcelo_for@ucp.edu.ar · <https://orcid.org/0000-0001-6389-1590>

Resumen. El desarrollo del presente trabajo de investigación denominado *seguimiento y control de ganado vacuno mediante geoposicionamiento*, cuya línea prioritaria es innovación y tecnología y es del tipo aplicada. Con el mismo se logró desarrollar, un prototipo que permite determinar la ubicación en tiempo real el ganado vacuno. Permite recibir coordenadas de ubicación en tiempo real para luego enviarlas mediante la red celular a una plataforma web de Internet de las Cosas que opera con esta información y alimenta diferentes widgets en los cuales se puede observar la ubicación del ganado. Para lograr esto se utiliza un Sistema de Posicionamiento Global, cuyo objetivo es determinar las coordenadas espaciales de puntos respecto de un sistema de referencia mundial. El sistema está basado en la placa Arduino conectado a un módulo SIM808 que ofrece la tecnología GPS y permite solicitar a los satélites coordenadas de posicionamiento. En un futuro, se estima, que todas las ganaderas deberían contar con un sistema de control de animales, debido a que es primordial realizar seguimiento y control en tiempo real, para conocer la ubicación, combatir el robo y controlar la salud del mismo. La metodología utilizada para el desarrollo del sistema es la del prototipado.

Palabras clave: Geoposicionamiento, IoT, Ganado, Prototipo.

MONITORING AND CONTROL OF CATTLE BY GEOPOSITIONING

Abstract. The development of this research work called *monitoring and control of cattle by geopositioning*, whose priority line is innovation and technology and is of the applied type. With it, it will be possible to develop a prototype that will allow determining the location of the cattle in real time. It allows receiving location coordinates in real time and then sending them through the cellular network to an Internet of Things web platform that operates with this information and feeds different widgets in which the location of the cattle can be observed. To achieve this, a Global Positioning System is used, whose objective is to determine the spatial coordinates of points with respect to a world reference system. The system is based on the Arduino board connected to a SIM808 module that offers GPS technology and allows the satellites to request positioning coordinates. In the future, it is estimated that all farmers must have an animal control

system, because it is essential to monitor and control in real time, to know the location, combat theft and control the health of the same. The methodology used for the development of the system is that of prototyping.

Keywords: Geopositioning, IoT, Cattle, Prototype.

Introducción

Argentina ha sido históricamente un país agrícola-ganadero y, dada su extensión y sus volúmenes de producción, es considerado uno de los grandes proveedores de alimentos del mundo. En los últimos años debido, entre otras cosas, a los precios de los commodities (en primer lugar, el precio de la soja y en segundo lugar el precio del trigo) y a factores políticos, la ganadería ha perdido terreno por sobre la agricultura. (Agrositio, 2008)

El noreste argentino (NEA) está integrado por las provincias de Corrientes, Misiones, Chaco y Formosa en su totalidad y parte de las provincias de Entre Ríos y Santa Fe, las mismas comparten características agroeconómicas que le confieren identidad propia a la región. Cuenta con 19,46 millones de cabezas vacunas, que representan el 40% del stock ganadero nacional, según la primera campaña de vacunación del año 2010 del SENASA. Por orden de importancia, Santa Fe cuenta con 6,03 millones de cabezas, Corrientes 4,87 mill, Entre Ríos 3,98 millones, Chaco 2,38 millones, Formosa 1,79 millones y finalmente Misiones sólo 410 mil cabezas. (Acosta, F. y otros, 2012)

Los productores ganaderos se agrupan en estratos de acuerdo a la cantidad de hacienda que poseen, según el SENASA (2010). De esta manera, se destacan las provincias de: Santa Fe con el 44% y Misiones con el 35% de los vacunos en manos de los productores más pequeño (poseen hasta 100 cabezas) y Corrientes con el 19% de los vacunos en propiedad de los establecimientos más grandes (poseen más de 5.000 cabezas).

Según el INTA (2015), existen claras evidencias que la producción ganadera de la Provincia de Formosa ha crecido cuantitativamente en las últimas décadas. Son varias las causas que motorizaron esta intensificación productiva local. En especial, el proceso de agriculturización que viene sufriendo el país hace más de una década el cual provocó un crecimiento de la ganadería en las regiones como el NOA y el NEA. Es así que la ganadería en Formosa, al igual de lo que ocurre en el país, ha experimentado cambios estructurales y geográficos como consecuencia de la transformación del sector agropecuario.

En una visión a largo plazo, independientemente de los ciclos de corto plazo, se pueden identificar dos etapas en la evolución del stock ganadero en Formosa durante los últimos 12 años. La primera comprendida entre los años 2003 y 2008, en la cual se registró un significativo incremento del stock, que se expandió a una tasa anual acumulativa del 5%. Y en la segunda etapa comprendida entre 2009 y 2015 no se produjo una tasa de variación notable del stock, pero luego de una leve caída en el mismo, se inició una nueva fase de retención al punto que en 2013 se alcanzaron niveles superiores al registrado en 2008. Por lo tanto, se podría sostener que los valores mínimos alcanzados durante 2008 constituyeron un piso a partir del cual evolucionó el ciclo ganadero Formoseño en los últimos años, datos difundidos por el INTA en el año 2015.

Sin embargo, el “abigeato” sigue siendo en nuestro país uno de los flagelos que más ha golpeado al sector ganadero, conocido en la jerga del campo como “cuatrerismo”; es una conducta delictual difícil de controlar, a ello debemos sumar la faena clandestina

que implica un riesgo para la salud humana y animal, con la correspondiente evasión impositiva, el transporte y comercialización sin cumplimiento de las normas sanitarias y bromatológicas, comprometiendo seriamente el aspecto sanitario humano y animal, existiendo serios riesgos de la propagación de diversas enfermedades.

Este accionar delictivo se ve favorecido por las grandes extensiones rurales de nuestro país y de nuestra provincia, que impiden muchas veces al momento de la prevención de un control adecuado en el cuidado de la propiedad del ganado, de las reses, etc., sumado a las deficiencias en la infraestructura de las policías rurales del país y de la legislación que ordenaba una mayor acción represiva, establecida en la Ley N° 25.890 del Código Penal de la Provincia de Formosa.

La pérdida del ganado vacuno por parte de los ganaderos de la Provincia de Formosa es un tema muy importante a resolver, en la actualidad los mismos no tienen forma de controlar de manera eficaz la ubicación de sus animales y solo pueden contratar gente que cuide sus ganados, pero eso implica un alto costo económico. (AM990formosa, 2017)

Según el presidente de la Sociedad Rural Formosa, Carlos Montoya, en el último año aumentaron los casos de abigeato en la zona de frontera y expresó la preocupación de los ruralistas por la violencia y la agresividad de los cuatrereros. (Agrositio, 2008)

Esto se evidencia en la nota realizada a una ganadera situada al norte de la capital formoseña, quien comentó con mucha angustia, los graves daños que produce a la economía y al trabajo, la acción sostenida y cada vez más sofisticada de los cuatrereros. El abigeato depreda la economía ganadera de la provincia y hackea la fuente laboral. (Panorama Regional, 2018)

Según miembros de la Asociación Ganadera de Formosa, un 80% de las pérdidas de los animales en la Provincia de Formosa es por el Abigeato, un 15% por extravíos en otros campos y un 5% por enfermedades; razón por la cual, plantearon la factibilidad de utilizar las TIC para poder controlar y conocer la ubicación de sus animales en un tiempo determinado.

En la actualidad las tecnologías están emigrando a todos los campos de la economía, proveyendo innovación y progreso constante. Cada vez más nos encontramos en un ambiente mayormente interconectado, donde el fin es el desarrollo y mejoramiento de los actuales sistemas, buscando mayor viabilidad y factibilidad en los procesos.

De esta necesidad surge la idea del presente proyecto de investigación, con el fin de satisfacer esta necesidad básica para los ganaderos utilizando las nuevas tecnologías combinando los conceptos de Hardware y Software Libre, Internet de las Cosas y el geoposicionamiento satelital.

Por lo expresado anteriormente, se desarrolló un prototipo que permite determinar la ubicación en tiempo real del ganado vacuno. El sistema permite recibir coordenadas de ubicación en tiempo real para luego enviarles mediante señal satelital a una plataforma web de Internet de las cosas (Internet Of Things, IOT) que opera con esta información y alimenta diferentes widgets en los cuales se puede observar la ubicación del ganado vacuno, fecha y hora de los mismos.

Para lograr esto se utilizará un sistema de posicionamiento global (Global Positioning System, GPS), que cuyo objetivo es determinar las coordenadas espaciales de puntos respecto de un sistema de referencia mundial.

Según la Fundación de la Innovación Bankinter, Internet de las cosas o IoT se fundamenta por entrelazar objetos de uso cotidiano con el Internet, permitiendo así mayor facilidad para la obtención y manejo de la información. En esto consisten las reconocidas redes de sensores, las cuales están compuestas por una cantidad específica de dispositivos

y cuyos datos va a un nodo coordinador o router y de allí a un servidor web donde es posible almacenar y mantener la información de mayor relevancia.

Este tipo de tecnologías no solo se está viendo en el campo de intercomunicación y la milicia; también se está viendo cómo, este tipo de aspectos han migrado a sectores en los cuales muy poco se creía su necesidad, como es el campo de la agronomía en todas sus subdivisiones. El cual aún sigue teniendo un manejo con poca afección tecnológica en la Provincia de Formosa.

El objetivo general de la investigación fue determinar la aplicabilidad de un sistema tecnológico de hardware y software libre para el seguimiento del ganado vacuno en la provincia de Formosa en el periodo 2020 - 2021.

Y sus objetivos específicos fueron:

- Analizar alternativas de control y seguimiento de ubicación del ganado vacuno.
- Desarrollar el prototipo y la interfaz para el sistema de seguimiento.
- Simular el circuito del dispositivo para verificar el buen funcionamiento del prototipo.

Marco de referencia

Para la elaboración de este proyecto se ha llevado a cabo una recopilación de varias fuentes de investigaciones bibliográficas, tales como: repositorios digitales de varias universidades, artículos científicos, revistas científicas, etc., en busca de temas similares al propuesto los cuales han desarrollado trabajos que son esenciales para la elaboración de esta investigación.

En el año 2013, Ángel René Canché UC y Jonathan Ismael Mukul Chi desarrollaron un “LocaPet. Localizador Satelital para Mascotas”, donde describe el desarrollo de un sistema que se encarga del monitoreo de una mascota dentro de un área específica. El hardware que se utiliza en este sistema consta de una placa Arduino UNO 328 modelo MIC-06664, GPS GY-GPS6MV1 con el chip GPS NEO-6M, Placa GSM/GPRS SIM900 modelo WGW-06633. El software utilizado es de código abierto mediante la plataforma Arduino y códigos AT. Este sistema emite una alerta mediante un mensaje de texto con la ubicación geográfica, hora y fecha que es dirigida al dueño en el caso de que su mascota abandone el área indicada por un tiempo mayor al establecido, lo que permite saber la ubicación de la mascota en cualquier lugar siempre que exista cobertura de la red celular GSM.

En el 2015, Ezequiel Gorandi, Nicolás Clemares y Andrés Moltoni, elaboraron un “Collar con tecnología GPS para monitoreo animal”, donde dicho sistema permite obtener las coordenadas instantáneas de la posición y se comunica a través del protocolo NMEA 0183 con un microcontrolador de 8 bits para procesar y almacenar la información en memoria no volátil, el lenguaje de programación utilizado es Python y dicha información es cargada a un servidor y se puede acceder mediante una aplicación web en la cual se puede modificar los tiempos entre muestras. Este sistema fue utilizado en el ganado bovino, caprino, ovino y también en perros pastores.

En el 2016, Carlos Andrade Parreño elabora el “Diseño e implementación de un sistema cliente-servidor para el envío de posición y signos vitales de mascotas sobre dispositivos móviles en la plataforma Arduino”. Este proyecto se basa en la elaboración de un prototipo que permite evitar la pérdida de mascotas, enviando los signos vitales y su ubicación mediante un hardware que se encuentra ubicado en el pecho del animal con la ayuda de un arnés, este envía información a un servidor web y base de datos las cuales son procesadas y almacenadas para su visualización en un dispositivo móvil o interfaz web. El Hardware utilizado consta de una placa de Arduino Uno, Módulo SIM 908C,

Módulo Transductor Acelerómetro MMA7361, Sensor de Temperatura Infrarrojo MLX90614. El Software para el desarrollo de la aplicación Android se lo hace mediante el lenguaje Java y en ciertas partes hacen uso del lenguaje XML, lenguaje Arduino.

En el 2016, Guido Buscetti Castro, Matías Prieto, Joaquín Muguerza y Martín Ríos; retomaron un proyecto que iniciaron en la universidad: “Hacer algo para evitar el robo del ganado”, investigando llegaron a la conclusión de que era necesario saber “qué le pasaba al animal cuando se está por enfermar o cuando está entrando en celo, todos datos de mucha importancia para el productor”. A fines del año pasado, diseñaron un prototipo funcional de un collar que recopila datos y comenzaron a probar asociándose con la Facultad de Veterinaria de la Universidad Nacional de La Plata. “El hardware se conecta a Internet y envía la información a una plataforma capaz de alertar al productor cuando algo le pasa al animal que requiere una acción de su parte”.

El desarrollo es importante porque permite, mediante la detección temprana, ayudar al productor a ahorrar mucho dinero en fármacos o, incluso, evitar el sacrificio del animal. Además, mediante la detección del celo, cuya alerta se envía vía SMS, el trabajo de la persona de campo se simplifica y permite optimizar la ventana corta de tiempo para inseminar a los animales.

El avance incontenible de Internet ha tenido un impacto radical en la economía y en la sociedad del siglo XXI. La convergencia de las Tecnologías de la Información y las Comunicaciones han transformado la Internet tradicional en la Internet de las Cosas (IoT), hasta tal punto que ha logrado insertarse en el sistema productivo. El resultado en la industria ha sido un nuevo modelo organizativo y un cambio disruptivo que ha sido llamado la «cuarta revolución industrial» o «Industria 4.0»; término éste acuñado por el gobierno alemán para describir la fábrica inteligente (Smart Factory, en inglés), es decir, “una visión de la fabricación informatizada con todos los procesos interconectados por IoT”, (Romero y otros, 2017).

Partiendo de la problemática de la pérdida del ganado vacuno por parte de los ganaderos de la Provincia de Formosa, la cual necesita una pronta respuesta, se pretende con esta investigación dar una solución rápida a través de un dispositivo electrónico ubicado en las caravanas de las vacas, utilizando los conceptos de Internet de las cosas, el geoposicionamiento y las tecnologías de hardware y software libre.

Según Karen Rose, Scott Eldridge, Lyman Chapin, (2015) el concepto de Internet de las cosas es un tema emergente de importancia técnica, social y económica. En este momento se están combinando productos de consumo, bienes duraderos, automóviles y camiones, componentes industriales y de servicios públicos, sensores y otros objetos de uso cotidiano con conectividad a Internet y potentes capacidades de análisis de datos que prometen transformar el modo en que trabajamos, vivimos y jugamos. Las proyecciones del impacto de la IoT sobre Internet y la economía son impresionantes: hay quienes anticipan que en el año 2025 habrá hasta cien mil millones de dispositivos conectados a la IoT.

Por lo general, el término Internet de las Cosas se refiere a escenarios en los que la conectividad de red y la capacidad de cómputo se extienden a objetos, sensores y artículos de uso diario que habitualmente no se consideran computadoras, permitiendo que estos dispositivos generen, intercambien y consuman datos con una mínima intervención humana. Sin embargo, no existe ninguna definición única y universal. (Karen Rose, Scott Eldridge, Lyman Chapin, 2015)

Según Ghio M. Gina (2008), El Sistema de Posicionamiento Global o GPS, aunque su nombre correcto es NAVSTAR-GPS1, es un sistema global de navegación por satélite que permite determinar en todo el mundo la posición de un objeto, una persona,

un vehículo o una nave. Podemos alcanzar una precisión hasta de centímetros, usando el GPS diferencial, pero lo habitual son unos pocos metros.

El GPS funciona mediante una red de 27 satélites (24 operativos y 3 de respaldo) en órbita a 20.200 km sobre el globo terráqueo, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra. Cuando se desea determinar una posición, el receptor que se utiliza para ello localiza automáticamente como mínimo tres satélites de la red, de los que recibe unas señales indicando la posición y el reloj de cada uno de ellos. Con base en estas señales, el aparato sincroniza el reloj del Sistema de Posicionamiento y calcula el retraso de las señales; es decir, la distancia al satélite. Por "triangulación" los tres satélites calculan la posición en que el GPS se encuentra. La triangulación en el caso del Sistema de Posicionamiento Global se basa en determinar la distancia de cada satélite respecto al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los tres satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene la posición absoluta o coordenada reales del punto y medición. También se consigue una exactitud extrema en el reloj del GPS, similar a la de los relojes atómicos que llevan a bordo cada uno de los satélites. (Ghio M. Gina, 2008).

Cuando se habla de Hardware libre se hace referencia a la libertad que existe al utilizar cualquier dispositivo junto con toda su documentación. Según Delgado (2018), el hardware se considera libre cuando toda la información de su hardware, diseños, medidas y herramientas utilizadas para la creación de dichos dispositivos son compartidas públicamente, de esta manera se ayuda a los desarrolladores para que puedan mejorar los diseños y aportar mucho más a este tipo de proyectos. (Delgado, 2018).

El software libre hace referencia a la libertad que poseen los usuarios para poder modificar, copiar, ejecutar, estudiar, distribuir y mejorar el software. Sin embargo, según Free Software Foundation *“un programa se considera software libre si los usuarios tienen acceso a las cuatro libertades esenciales como lo son:*

- Libertad para poder ejecutar el programa como el usuario lo desee.
- Libertad para estudiar cómo funciona el programa y modificarlo de acuerdo a las necesidades del usuario.
- Libertad para distribuir copias del software al prójimo.
- Libertad de distribuir copias de versiones mejoradas de terceras personas”.

Metodología

A partir de los conceptos mencionados anteriormente el proyecto dividió en etapas teniendo en cuenta los objetivos propuestos al presente.

En lo que respecta al diseño del dispositivo y desarrollo del software se utilizará el Modelo Prototipo y para el cumplimiento de las etapas del proyecto utilizaremos la metodología ágil Kanban.

Según Sommerville (2011): *“Prototipo de sistema, donde rápidamente se desarrolla una versión del sistema o una parte del mismo, para comprobar los requerimientos del cliente y la factibilidad de algunas decisiones de diseño. Esto apoya el hecho de evitar el cambio, al permitir que los usuarios experimenten con el sistema antes de entregarlo y así refinar sus requerimientos. Como resultado, es probable que se reduzca el número de propuestas de cambio de requerimientos posterior a la entrega.”* (Sommerville, 2011, p. 44)

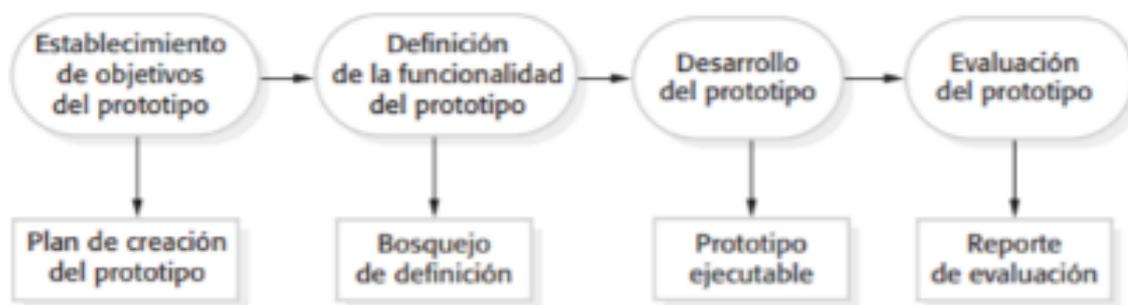


Figura 1. Proceso de desarrollo del prototipo

Nota: Fuente: (Sommerville, 2005, p. 375)

Durante el desarrollo se realizó el ensamble del módulo SIM 808 a la placa Arduino, para luego llevar a cabo la programación de estas placas. Se realizaron diferentes pruebas hasta lograr el correcto funcionamiento del sistema.

Como etapa final se realizó la evaluación del prototipo, donde se utilizaron los objetivos establecidos al principio, para obtener un plan de evaluación; dicho plan consistió en:

- Realizar pruebas de recorrido del animal con el GPS configurado y funcionando.
- Controlar mediante la plataforma IoT los datos que envía nuestro prototipo en tiempo real.
- Una vez finalizado el recorrido, controlar los valores obtenidos para luego definir el desempeño del prototipo y poder descubrir errores y omisiones en el mismo.

Definición y Desarrollo del sistema

En esta etapa se realizó en primera medida la definición de los requerimientos y el análisis de riesgos, posteriormente se explica cómo se llevó a cabo la elección de la placa base y los módulos necesarios para llevar adelante el desarrollo del prototipo y lograr que este cumpla con los objetivos del proyecto, se detallan las características del IDE de desarrollo del software, necesario para realizar la programación de la placa base.

El sistema propuesto en el presente trabajo de investigación, está basado en un prototipo desarrollado en hardware y software libre, más precisamente basado en la placa Arduino UNO como núcleo central, la cual trabajara de manera conjunta con módulos externos como lo es el shield SIM808, que permitirá, a partir de las funcionalidades integradas de GSM, GPRS Y GPS, obtener los datos necesarios para llevar a cabo la identificación periódica de la localización de una flota de vehículos, poder ver las rutas que realizaron los mismos y saber en tiempo real su ubicación, dirección y velocidad.

Requerimientos del sistema

Lo que respecta a la obtención de los requerimientos, los mismos fueron establecidos a partir de entrevistas, pero al tratarse de un proyecto que utiliza la metodología del prototipado, estos requerimientos pueden modificarse en cualquier segmento de desarrollo, como así también agregarse otros nuevos.

La utilización de esta metodología permite presentar a los usuarios avances de forma tal que les permiten ver el funcionamiento y su implicancia en las actividades

requeridas, como así también encontrar aspectos fuertes y débiles en el sistema, ya que ellos son los que lo van a utilizar en la cotidianeidad.

Los requerimientos iniciales que se obtuvieron a partir del análisis de la información obtenida en la entrevista para el sistema son los detallados a continuación:

- El sistema GPS debe solicitar, de manera periódica, información a los satélites sobre sus coordenadas de posicionamientos.
- Los datos entregados por el satélite deben ser enviados a un servidor web.
- Los datos enviados deben ser utilizados por una plataforma web, para poder representar en el mapa la posición real del móvil.

Estos requerimientos se vieron modificados mediante pruebas realizadas al prototipo y se pudieron obtener los siguientes cambios en los mismos:

- El sistema GPS debe solicitar, de manera periódica, información a los satélites como ser las coordenadas de posicionamiento (longitud, latitud, altitud), dirección, velocidad, fecha y hora.
- Los datos enviados deben ser utilizados por una plataforma de IoT, la cual pueda procesar los datos y mostrar sus valores en diferente widget (agregar al glosario).
- Dicha plataforma debe poder mostrar un historial con los recorridos realizados.

Elección del hardware adecuado

Se analizaron diferentes alternativas, las cuales nos ofrece el mercado en relación al hardware de libre programación, estas opciones fueron comparadas teniendo en cuenta sus características y prestaciones con respecto a las necesidades que presenta este proyecto.

Distintas alternativas en tecnologías hardware libre

Para la selección y elección final del hardware utilizado para este proyecto se investigó las características principales de diferentes hardware libres que se encuentran actualmente en el mercado, teniendo en cuenta los componentes, sus conexiones, sus prestaciones y costo.

A continuación, se especifican diferentes placas analizadas con sus correspondientes características, funcionalidades y valor de mercado:

Raspberry pi

Raspberry Pi es una maravilla en miniatura, aguarda en su interior un importante poder de cómputo en un tamaño no más grande que el de una tarjeta de crédito. (Upton, Halfacree, 2016, p. 2)

El procesador en el interior del Raspberry Pi es un procesador multimedia Broadcom BCM2835 system-on-chip (SoC). Esto quiere decir que la mayor parte de los componentes del sistema, incluidos la unidad central de procesamiento y la de gráficos junto con el audio y el hardware de comunicaciones, se encuentran integrados dentro de aquel único componente oculto ubicado justo debajo del chip de la memoria de 256 MB en el centro de la placa. (Upton, Halfacree, 2016, p. 6)

No es sólo el diseño del SoC lo que hace al BCM2835 diferente del procesador de su PC o laptop, lo que lo hace también diferente es que utiliza una arquitectura de conjunto

de instrucciones (Instruction Set Architecture, ISA) distinta, conocida como ARM. (Upton, Halfacree, 2016, p. 6)

Desarrollada por Acorn Computers años atrás a finales de 1980, la arquitectura ARM es relativamente poco conocida en el mundo de las computadoras de escritorio. En donde destaca, sin embargo, es en los dispositivos móviles: el teléfono en su bolsillo es casi seguro que cuente con al menos un núcleo de procesamiento basado en ARM escondido en su interior.

La combinación de la arquitectura RISC (Simple Reduced Instruction Set) y su bajo consumo energético lo convierten en la opción perfecta frente a los chips de computadoras de escritorio que demandan altos consumos y arquitecturas CISC (Complex Instruction Set). (Upton, Halfacree, 2016, p. 6)

Esto, sin embargo, significa que la Raspberry Pi no es compatible con el software de las PC tradicionales. La mayoría del software para computadoras de escritorio y laptops se construye teniendo en cuenta la arquitectura de conjunto de instrucciones x86, presente en los procesadores como AMD, Intel y VIA. Por consiguiente, este software no funciona en la Raspberry Pi que se basa en la arquitectura ARM. (Upton, Halfacree, 2016, p. 7)

Raspberry Pi, está diseñada para ejecutar el sistema operativo llamado GNU/Linux. A diferencia de Windows u OS X, Linux es de código abierto, esto quiere decir que es posible descargar el código fuente del sistema operativo por completo y hacer los cambios que uno desee, nada es ocultado, y todos los cambios hechos están a la vista del público.

Este espíritu de desarrollo de código abierto ha permitido a Linux rápidamente ser modificado para poder ejecutarse sobre la Raspberry Pi, un proceso conocido como portabilidad, varias versiones de Linux (conocidas como distribuciones) han sido portadas al chip BCM2835 de la Raspberry Pi, incluyendo Debian, Fedora Remix y Arch Linux. (Upton, Halfacree, 2016, p. 7)

BeagleBoard

Los Beagles son pequeños hardware abierto, computadoras de software abierto que se pueden conectar a lo que se tenga alrededor de la casa. (<https://uk.farnell.com/b/beagleboard>)

Beagles significa una gran funcionalidad en pequeños paquetes, porque estas pequeñas PC pueden ser usadas para todo tipo de aplicaciones, y pueden manejar muchas de las mismas tareas que una PC de escritorio. (<https://uk.farnell.com/b/beagleboard>)

Son diminutos, accesibles y de código abierto para Android, Ubuntu y diferentes versiones de Linux a su alcance, de alto rendimiento y bajo consumo.

La familia Beagle tiene como objetivo primordial ayudar a los estudiantes a aprender programación y a los desarrolladores que para que produzcan más rápidamente sin exceso de ruido y gastos. (<https://uk.farnell.com/b/beagleboard>)

Las características principales de esta placa son las siguientes:

El BeagleBoard es alimentado por USB y cuenta con un procesador OMAP3530 ARM Cortex-A8 de 720 MHz, extensiones NEON y VFP para aceleración adicional, video de alta resolución y la capacidad de transmitir con un reproductor multimedia portátil, permite trabajar con toda la funcionalidad de una computadora portátil en un solo paquete pequeño. (<https://beagleboard.org/beagleboard>)

La variedad de conectividad que presenta la placa es la siguiente:

- Puerto USB 2.0 en movimiento (OTG)
- Conecte periféricos estándar a USB utilizando cualquiera de los siguientes:
 - Un adaptador de cable mini-A a estándar A
 - DVI-D utilizando un adaptador de HDMI a DVI-D
 - Conector MMC / SD / SDIO que permite una experiencia de escritorio completa.

Intel

Intel ha diseñado dos modelos de placas, el Galileo® y el Edison® acordes para proyectos de hardware y software libre, pero actualmente se ha discontinuado en su desarrollo, existen otros kits, pero por los costos no se aplican, a continuación, se da más información técnica de estas placas.

Intel galileo

La placa Intel Galileo de segunda generación proporciona un controlador de placa única para la comunidad creadora, estudiantes y desarrolladores profesionales. Basado en el Intel Quark SoC X1000, un sistema de clase de procesador Intel Pentium de 32 bits en un chip (SoC), el procesador Intel original y las capacidades de entrada / salidas nativas de La placa Intel Galileo (Gen 2) ofrece una función completa para una amplia gama de aplicaciones Arduino-Certificado y diseñado para ser hardware y software compatible con pines de una amplia gama de shields del Arduino Uno R3.

La placa Intel Galileo Gen 2 también ofrece un entorno de desarrollo más sencillo y rentable en comparación con el procesador Intel Atom y los diseños basados en el procesador Intel Core.

Utilizan el entorno de desarrollo de software Arduino para crear programas para Galileo denominados "sketches".

Intel Edison®

La plataforma de desarrollo Intel Edison está diseñada para reducir las barreras de entrada para una gama de inventores, empresarios y diseñadores de productos de consumo para prototipos y producir rápidamente "Internet de las cosas" (IoT) y productos de computación portátiles.

El módulo de cómputo Intel Edison está diseñado para usarse con tableros de circuitos impresos personalizados.

Arduino

Antes que nada, es importante definir que cuando se habla de Arduino se habla de tres cosas:

- Una placa hardware libre que incorpora un microcontrolador reprogramable y una serie de pines-hembra (los cuales están unidos internamente a las patillas de entrada/salida del microcontrolador) que permiten conectar allí de forma muy sencilla y cómoda diferentes sensores y actuadores. (Torrente Artero, 2013, p. 65).
- Un software (más en concreto, un "entorno de desarrollo") gratis, libre y multiplataforma (ya que funciona en Linux, MacOS y Windows) que debemos instalar en nuestro ordenador y que nos permite escribir, verificar y guardar("cargar") en la memoria del microcontrolador de la placa Arduino el

conjunto de instrucciones que deseamos que este empiece a ejecutar. Es decir: nos permite programarlo.

La manera estándar de conectar nuestro computador con la placa Arduino para poder enviarle y grabarle dichas instrucciones es mediante un simple cable USB, gracias a que la mayoría de placas Arduino incorporan un conector de este tipo. (Torrente Artero, 2013, p. 66).

- Un lenguaje de programación libre, por “lenguaje de programación” se entiende cualquier idioma artificial diseñado para expresar instrucciones (siguiendo unas determinadas reglas sintácticas) que pueden ser llevadas a cabo por máquinas.

Concretamente dentro del lenguaje Arduino, encontramos elementos parecidos a muchos otros lenguajes de programación existentes (como los bloques condicionales, los bloques repetitivos, las variables, etc.), así como también diferentes comandos –asimismo llamados “órdenes” o “funciones” – que nos permiten especificar de una forma coherente y sin errores las instrucciones exactas que queremos programar en el microcontrolador de la placa. Estos comandos los escribimos mediante el entorno de desarrollo Arduino. (Torrente Artero, 2013, p. 66)

Existe una gran variedad de placas Arduino, pero la cual se toma para esta investigación es el Arduino UNO R3, el cual utiliza el microcontrolador ATmega328, en adición a todas las características de las tarjetas anteriores, el Arduino Uno utiliza el ATmega16U2 para el manejo de USB en lugar del 8U2, esto permite flujos de transferencia más rápidos y más memoria. No se necesitan drivers para Linux o Mac.

Se añade pins SDA y SCL cercanos al AREF, es más, hay dos nuevos pines cerca del pin RESET, uno es el IOREF, que permite a los shields adaptarse al voltaje brindado por la tarjeta, el otro pin no se encuentra conectado y está reservado para propósitos futuros. La tarjeta trabaja con todos los shields existentes y podrá adaptarse con los nuevos shields utilizando esos pines adicionales.

El IDE open-source puede ser descargado gratuitamente, actualmente está disponible para Mac OS X, Windows y Linux.

Las características de la placa son:

- Microcontrolador ATmega328.
- Voltaje de entrada 7-12V.
- 14 pines digitales de I/O (6 salidas PWM).
- 6 entradas análogas.
- 32k de memoria Flash.
- Reloj de 16MHz de velocidad.

Especificación técnica del hardware utilizado

El enfoque del proyecto está orientado hacia el Arduino UNO por lo que el resto del hardware o módulos fueron seleccionados en función a esta tarjeta de desarrollo.

Anteriormente se detallaron características de la placa Arduino, por ese motivo a continuación se brindará información específica sobre el módulo que se integra a la placa base para poder lograr la funcionalidad requerida para este proyecto, el mismo es el shield eMGing SIM808.

Shield eMGing SIM808

El shield eMGing SIM808 es una placa de gran utilidad para este proyecto, debido a que el mismo nos permite conectividad GSM, GPS y GPRS, mediante la integración de múltiples módulos en un solo shield, haciendo así muy sencilla su implementación a nuestra placa base Arduino UNO.

Resultados

En cuanto a los resultados, se llevaron a cabo varias pruebas al dispositivo; como primera medida se realizó la conexión de una fuente de energía de 12 voltios y 2 Amper al Arduino UNO, ya sea con un cargador o con una batería portátil, hay que asegurarse que encienda correctamente y comience a enviar datos a la plataforma web.

Una vez que se verificó que el dispositivo encendió de manera correcta, se encuentra habilitado para poder realizar las pruebas pertinentes en un recorrido simple en el campo y observar cómo se va dibujando en el mapa la ruta que estamos recorriendo en tiempo real.

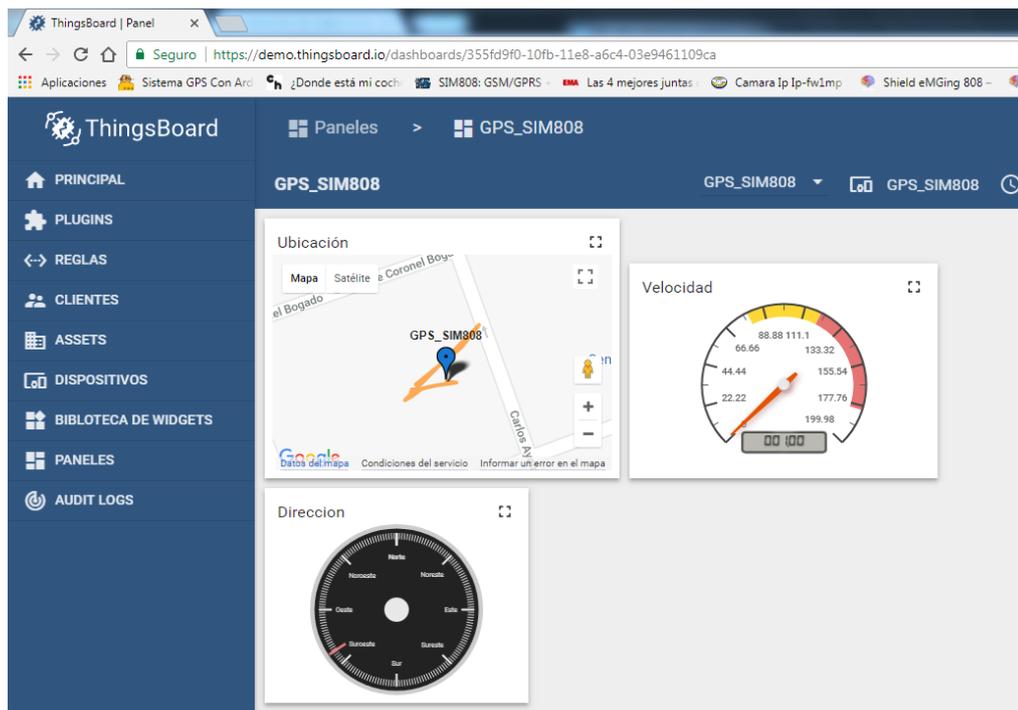


Figura 2. Simulación.

Nota: Fuente: Elaboración propia.

Como se puede ver en la ilustración anterior, el mapa supero todas las expectativas y se logró recopilar datos realmente precisos sobre los senderos recorridos, además, los datos de velocidad y dirección fueron considerablemente aceptables, cumpliendo así con los objetivos de este proyecto.

En general, el dispositivo puede ser instalado en cualquier tipo de animal que se desee monitorear y se podrán observar los resultados en la plataforma web ThingsBoard, obteniendo así un control preciso sobre los mismos.

En resumen, se realizó la programación del rastreador GPS para enviar datos directamente a ThingsBoard a través de solicitudes HTTP POST y administramos los datos en un tablero de instrumentos, en el mismo se pueden agregar múltiples dispositivos y paneles, cada uno con múltiples widgets brindan una muy buena impresión visual y tienen muchas opciones de personalización.

ThingsBoard ha demostrado ser una herramienta muy poderosa para observar datos de IoT, posee una interfaz muy simple, así como lo es su configuración, permitiendo una conexión sencilla y rápida para los dispositivos.

Conclusiones

El objetivo general de esta investigación consistió en el desarrollo de un sistema capaz de solicitar y recibir coordenadas de posicionamiento de los satélites dedicados al geo posicionamiento global, y poder enviar la información obtenida a una plataforma WEB para poder visualizar allí esos datos, para realizar el seguimiento al animal y tratar de evitar su robo. Esto se llevó a cabo mediante la implementación de hardware y software de libre programación, los elementos utilizados fueron el Arduino UNO, el shield SIM808, el entorno de desarrollo Arduino y la plataforma de internet de las cosas ThingsBoard.

El Arduino UNO R3 es una plataforma computacional física de código abierto que se basa en una tarjeta simple de entrada/salida que posee su propia IDE de desarrollo, dicha placa se puede integrar con diferentes módulos para así poder realizar infinitudes de funciones y acciones.

Se utilizó, junto al Arduino UNO, el shield SIM808 desarrollado por una empresa argentina, el cual tiene integrado en su placa los módulos GSM, GPRS Y GPS, necesarios para poder cumplir con los objetivos propuestos para esta investigación. Esta placa está diseñada para ir encastrado perfectamente sobre la placa Arduino, facilitando así su ensamble.

La plataforma web ThingsBoard, está dedicada al Internet de las cosas, es decir que cualquier dispositivo que envíe información a través de Internet pueda conectarse con ella, su objetivo principal es poder lograr administrar y visualizar la información que estos envían.

La implementación de hardware libre dedicado al campo del geo posicionamiento, permite la creación de nuevos servicios sobre la red, enfocados en el seguimiento y permitiendo su geo referencia en tiempo real, así también como el respectivo envío de información a través de la red móvil, lo que permite a plataformas de IoT una lectura en tiempo real de las coordenadas recibidas, y poder plasmar esos datos en diferentes widgets.

La investigación realizada fue muy satisfactoria, debido a que llevo a conocer varias opciones de hardware libre que podían ser utilizadas de diferentes maneras y con diferentes características, pero que llevarían a un resultado similar.

Para finalizar, la investigación cumplió con los objetivos propuestos al principio y los datos obtenidos fueron bien aceptados por las ganaderas.

Referencias bibliográficas

- Anderson, D. J., Carmichael, A., (2016). *Kanban Esencial Condensado*. (5ta Ed.). Washington: Lean Kanban University Press.
- Albanés, A. J. (21 de 05 de 2011). *Evolución en el Desarrollo Web*. <http://www.Isi.us.es/docencia>
- Barsky, O. (1992), Políticas agrícolas en la Argentina en el contexto del ajuste. *Ruralia*, 3, 7-34.
- Barsky, O.; Gelman, J. (2002). *Historia del agro argentino. Desde la conquista hasta fines del siglo XX*. Grijalbo-Mondadori. Argentina.
- Basualdo, E. & N. Arceo (2006). Evolución y situación actual del ciclo ganadero en la Argentina. *Revista Realidad Económica*, 221.
- Ghio M. Gina. (2008). Sistema de Posicionamiento Global (GPS): Sociedad de Especialistas Latinoamericanos. *Percepción Remota y Sistemas de Información Espacial Sesión*, 94.
- Held, Gilbert. (1997). *Diccionario de tecnología de las comunicaciones*. Paraninfo.
- Hernández Sampieri R., Fernández Collado C., Lucio P., (2010). *Metodología de la Investigación*. McGraw-Hill.
- Lattuada M. (1997). *El sector agropecuario Argentino hacia fines del milenio*. Transición e incertidumbre.
- Pallás Areny, Ramón. (2004). *Sensores y acondicionadores de señal*. Maracombo.
- Passaniti, M. V. (2011). *Estudio del sector de ganado y carne vacunos argentino y políticas públicas*. [Trabajo Final]. Universidad Católica Argentina.
- Pressman, R. (2002). *Ingeniería de Software. Un enfoque práctico*. Mc. Graw Hill.
- Rey, Eugenio. (1995). *Telecomunicaciones móviles*. Alfaomega.
- Sierra Pérez, M., Galocha Iragüen, B., Fernández Jambrina, J. I., & Sierra Castañer, M. (2003). *Electrónica de Comunicaciones*. Pearson Educación, S. A.
- Sommerville, I. (2011). *Ingeniería de Software*. Pearson.

Fecha de recepción: 29/09/2021

Fecha de revisión: 12/12/2021

Fecha de aceptación: 27/02/2022

Anexos

Anexo 1 Ensamble

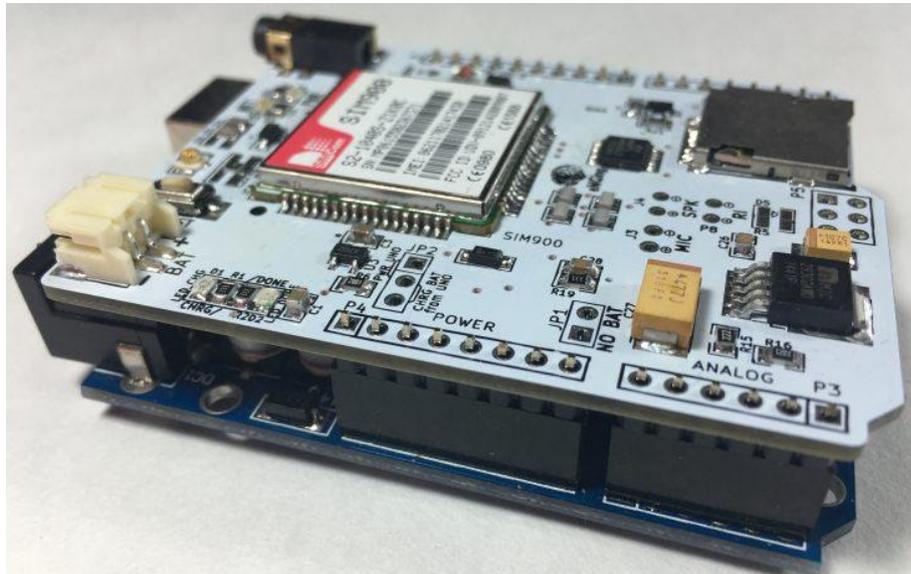


Figura 3. Ensamble paso 1.
Nota: Fuente: Elaboración propia.

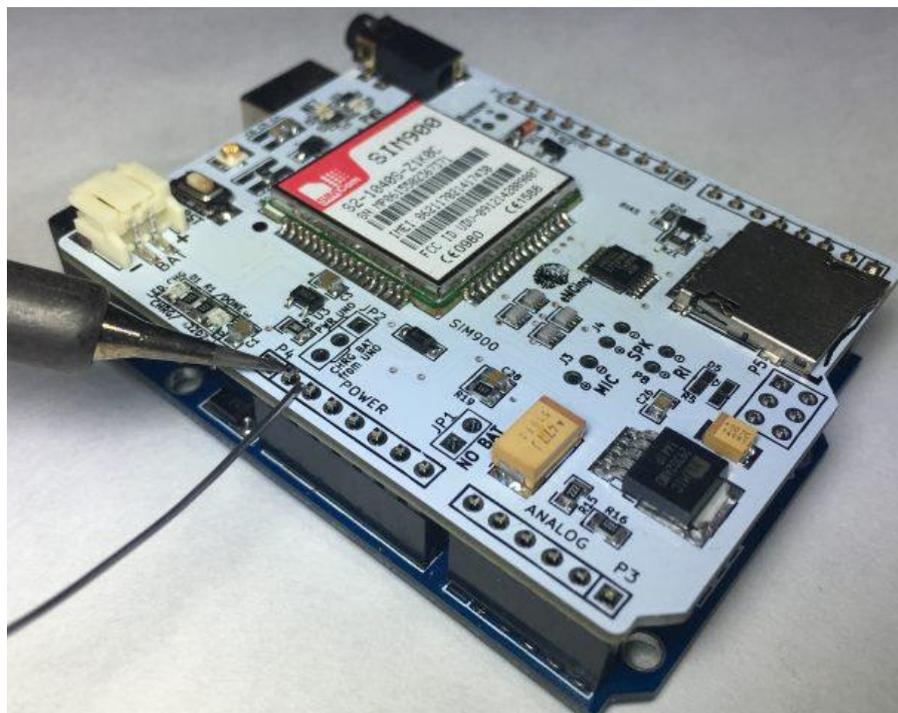
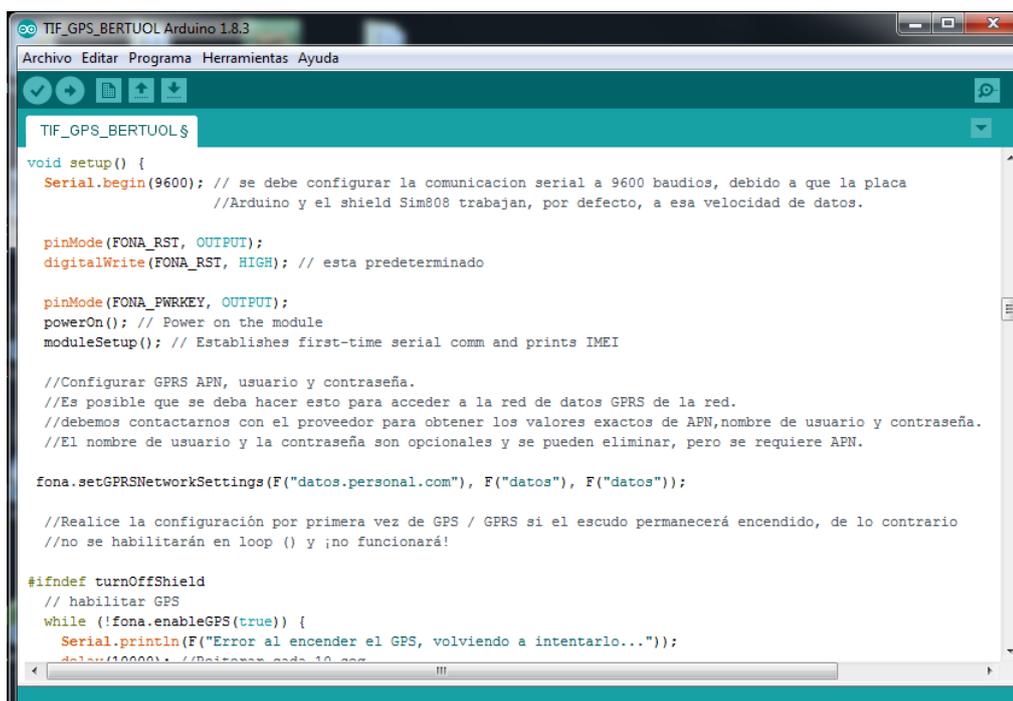


Figura 4. Ensamble paso 2.
Nota: Fuente: Elaboración propia.

Anexo 2

Desarrollo del Software GPS en Arduino



```

TIF_GPS_BERTUOL Arduino 1.8.3
Archivo Editar Programa Herramientas Ayuda

TIF_GPS_BERTUOL $

void setup() {
  Serial.begin(9600); // se debe configurar la comunicacion serial a 9600 baudios, debido a que la placa
                    //Arduino y el shield Sim808 trabajan, por defecto, a esa velocidad de datos.

  pinMode(FONA_RST, OUTPUT);
  digitalWrite(FONA_RST, HIGH); // esta predeterminado

  pinMode(FONA_PWRKEY, OUTPUT);
  powerOn(); // Power on the module
  moduleSetup(); // Establishes first-time serial comm and prints IMEI

  //Configurar GPRS APN, usuario y contraseña.
  //Es posible que se deba hacer esto para acceder a la red de datos GPRS de la red.
  //debemos contactarnos con el proveedor para obtener los valores exactos de APN,nombre de usuario y contraseña.
  //El nombre de usuario y la contraseña son opcionales y se pueden eliminar, pero se requiere APN.

  fona.setGPRSNetworkSettings(F("datos.personal.com"), F("datos"), F("datos"));

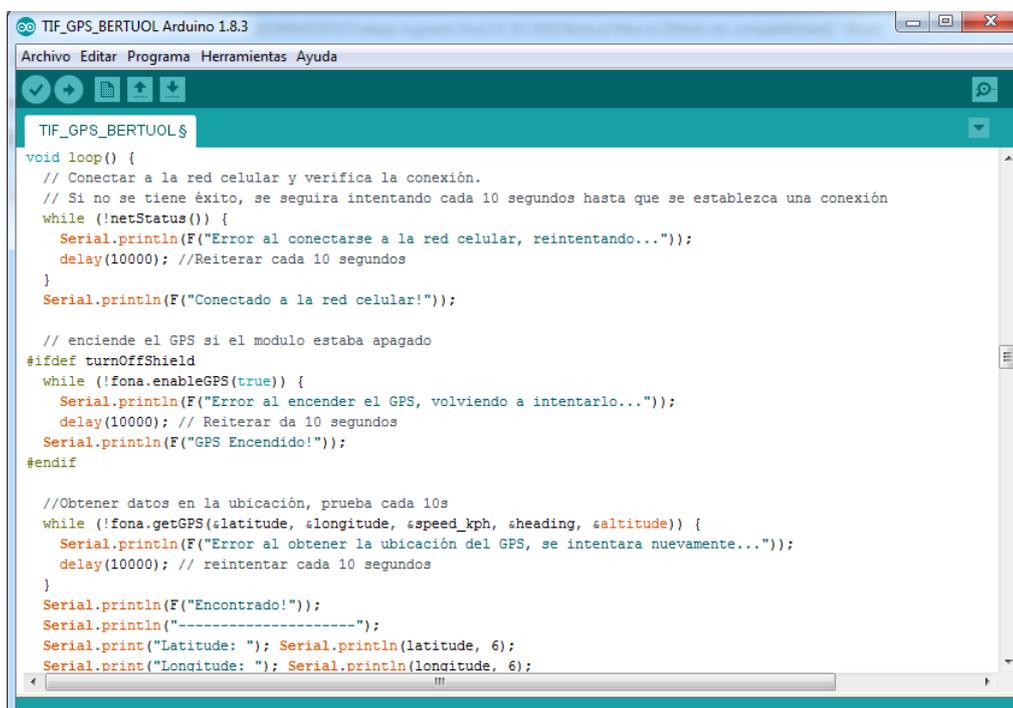
  //Realice la configuración por primera vez de GPS / GPRS si el escudo permanecerá encendido, de lo contrario
  //no se habilitarán en loop () y ¡no funcionará!

#ifdef turnOffShield
  // habilitar GPS
  while (!fona.enableGPS(true)) {
    Serial.println(F("Error al encender el GPS, volviendo a intentarlo..."));
    delay(10000); //Reiterar cada 10 seg
  }
}

```

Figura 6. Subrutina Setup.

Nota: Fuente: Elaboración propia.



```

TIF_GPS_BERTUOL Arduino 1.8.3
Archivo Editar Programa Herramientas Ayuda

TIF_GPS_BERTUOL $

void loop() {
  // Conectar a la red celular y verifica la conexión.
  // Si no se tiene éxito, se seguira intentando cada 10 segundos hasta que se establezca una conexión
  while (!netStatus()) {
    Serial.println(F("Error al conectarse a la red celular, reintentando..."));
    delay(10000); //Reiterar cada 10 segundos
  }
  Serial.println(F("Conectado a la red celular!"));

  // enciende el GPS si el modulo estaba apagado
#ifdef turnOffShield
  while (!fona.enableGPS(true)) {
    Serial.println(F("Error al encender el GPS, volviendo a intentarlo..."));
    delay(10000); // Reiterar da 10 segundos
    Serial.println(F("GPS Encendido!"));
  }
#endif

  //Obtener datos en la ubicación, prueba cada 10s
  while (!fona.getGPS(&latitude, &longitude, &speed_kph, &heading, &altitude)) {
    Serial.println(F("Error al obtener la ubicación del GPS, se intentara nuevamente..."));
    delay(10000); // reintentar cada 10 segundos
  }
  Serial.println(F("Encontrado!"));
  Serial.println("-----");
  Serial.print("Latitude: "); Serial.println(latitude, 6);
  Serial.print("Longitude: "); Serial.println(longitude, 6);
}

```

Figura 7. Subrutina Loop.

Nota: Fuente: Elaboración propia.

Anexo 3

Configuración de la plataforma Plataforma de IoT ThingsBoard

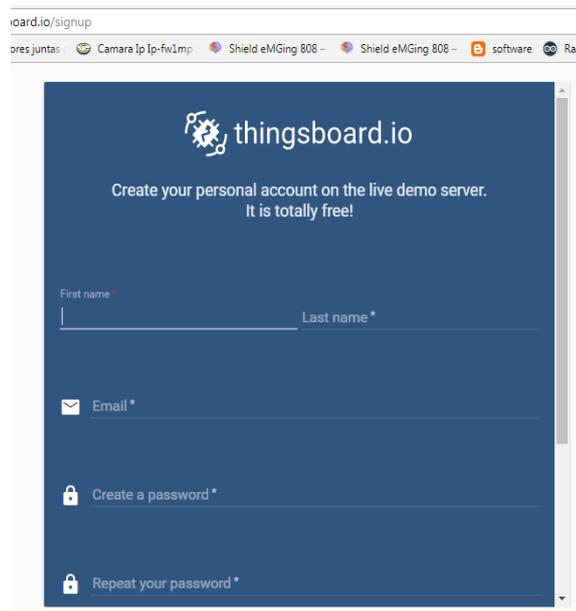


Figura 7. Configuración de la plataforma IoT paso 1.

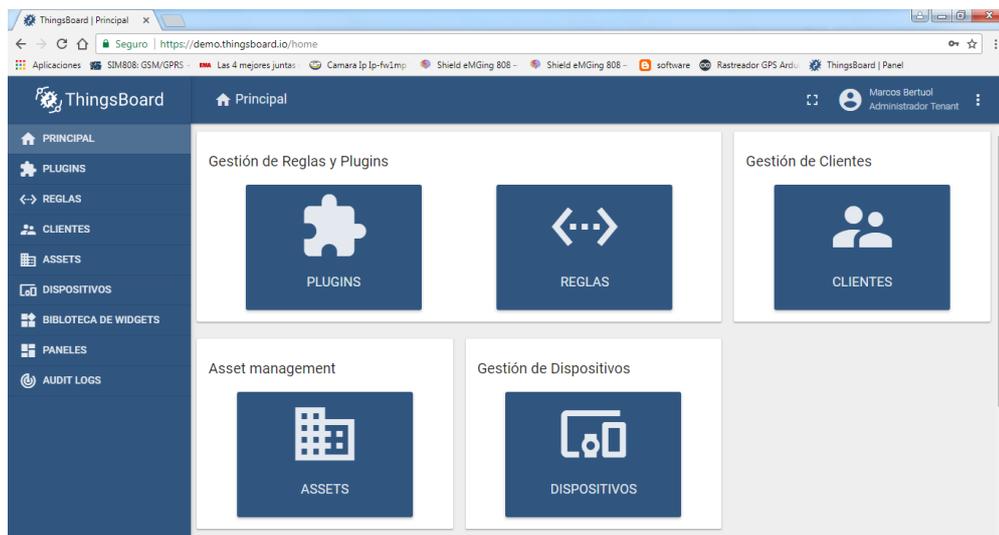


Figura 8. Configuración de la plataforma IoT paso 2.
Nota: Se seleccionó la pestaña "Dispositivos" en el lado izquierdo.

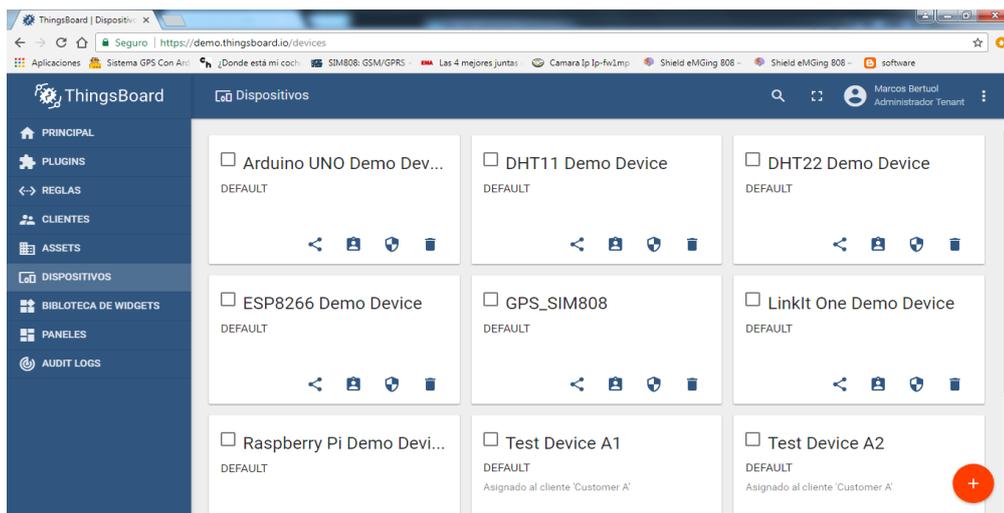


Figura 9. Configuración de la plataforma IoT paso 3.
Nota: Fuente: Elaboración propia.



Figura 10. Configuración de la plataforma IoT paso 4.
Nota: Fuente: Elaboración propia.

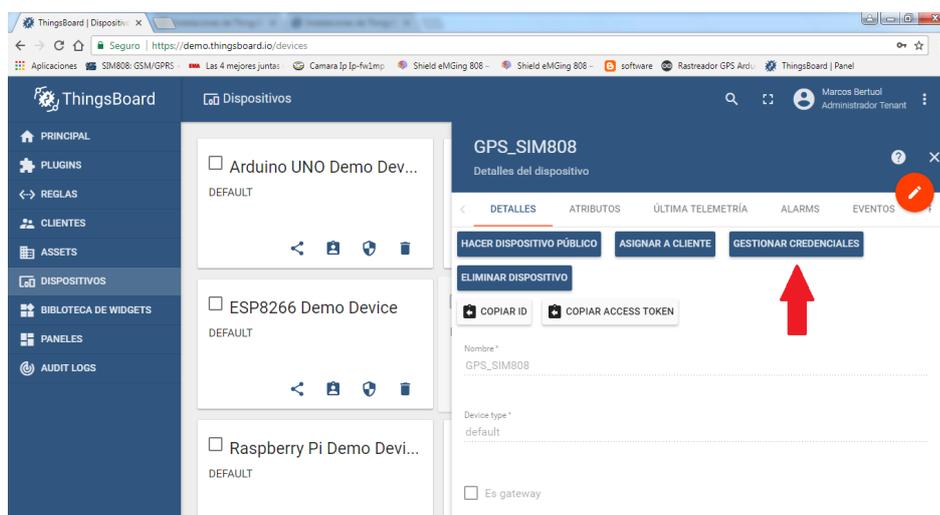
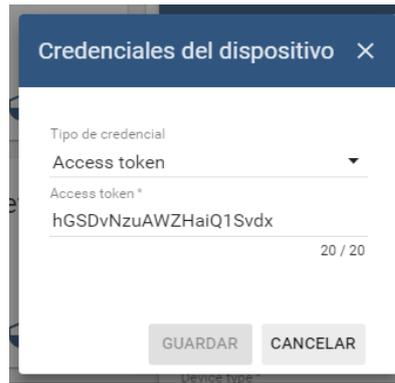


Figura 11. Configuración de la plataforma IoT paso 5.
Nota: Fuente: Elaboración propia.



Credenciales del dispositivo X

Tipo de credencial
Access token

Access token*
hGSDvNzuAWZHaiQ1Svdx
20 / 20

GUARDAR CANCELAR

Figura 12. Configuración de la plataforma IoT paso 6.

Nota: Fuente: Elaboración propia.

Anexo 4

Verificación la recepción de datos

Se ingresará en la misma página de detalles del dispositivo, en la pestaña "Última telemetría".

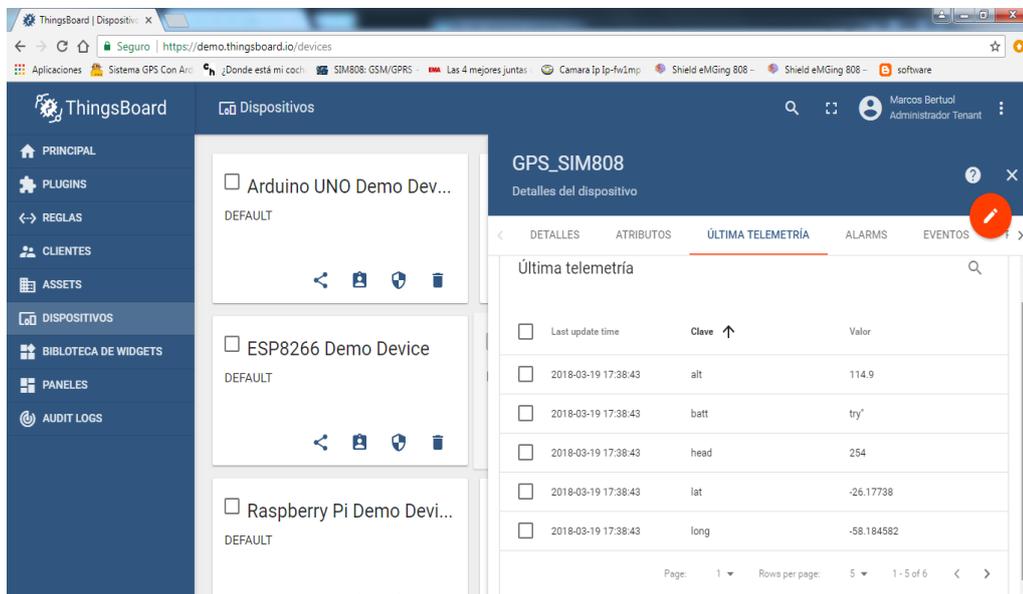


Figura 13. Verificación de recepción de datos.

Nota: Fuente: Elaboración propia.

Anexo 5 Configuración del tablero

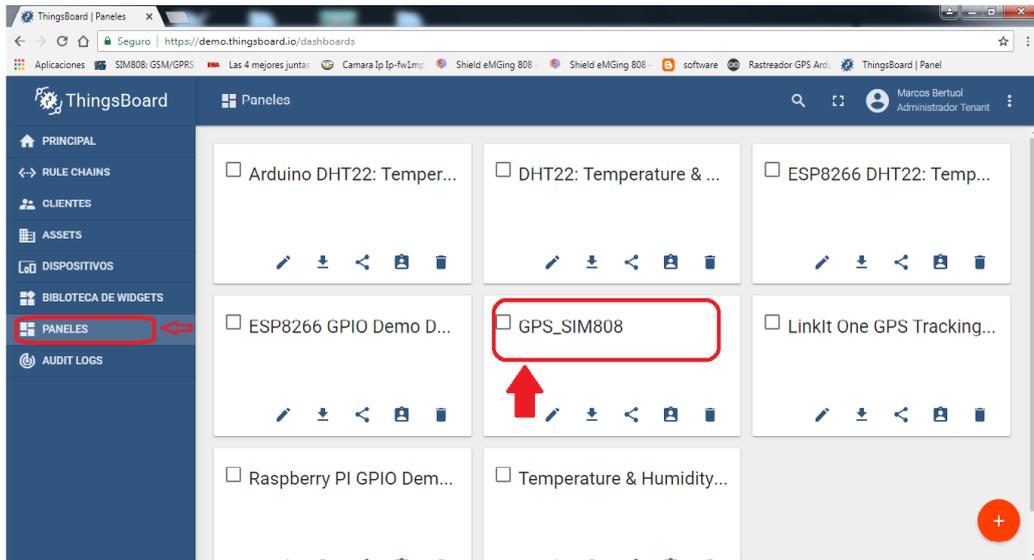


Figura 14. Configuración del tablero paso 1.
Nota: Fuente: Elaboración propia.

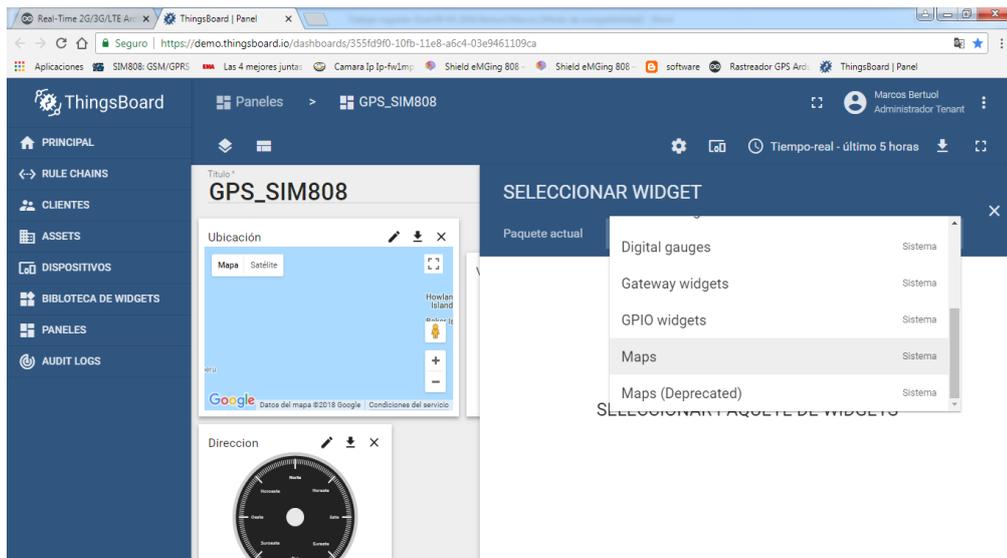


Figura 15. Configuración del tablero paso 2.
Nota: Fuente: Elaboración propia. Al Seleccionar esto se cargaron las vistas previas para todos los diferentes tipos de mapas que se puede elegir.

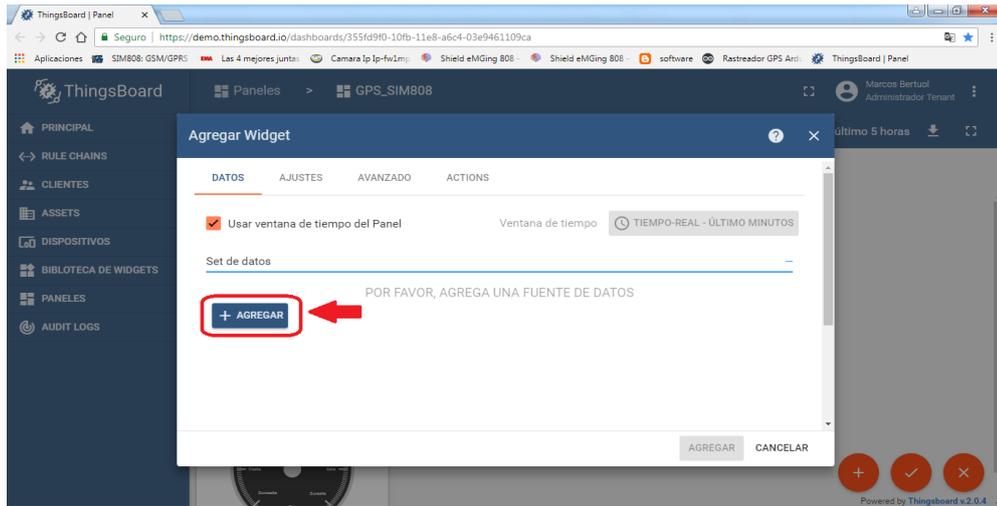


Figura 16. Configuración del tablero paso 3.

Nota: Fuente: Elaboración propia. Una vez seleccionado y configurado el widget se puede comenzar a utilizar el prototipo.