



UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CARRERA DE TELECOMUNICACIONES

TRABAJO DE INTEGRACIÓN CURRICULAR

previo a la obtención del título de

INGENIERO EN TELECOMUNICACIONES

**“DESARROLLO Y DISEÑO DE UN PROTOTIPO AUTOSOSTENIBLE
GEOLOCALIZADO IMPLEMENTANDO UNA RED WSN PARA EL RASTREO
DE CAPRINOS EN UN SECTOR AGRARIO DE LA COMUNA ZAPOTAL”**

AUTORES

PÉREZ ALTAMIRANO CRISTIAN GEOVANNY

VILLÓN TORRES VÍCTOR ANTONIO

DOCENTE TUTOR

ING. LUIS MIGUEL AMAYA FARIÑO, MGTR.

LA LIBERTAD - ECUADOR

2022-2

DECLARACIÓN DE DOCENTE TUTOR

En mi calidad de Docente Tutor del Trabajo de Integración Curricular, “**Desarrollo y diseño de un prototipo autosostenible geolocalizado implementando una red WSN para el rastreo de caprinos en un sector agrario de la comuna Zapotal**”, elaborado por los señores **Pérez Altamirano Cristian Geovanny** y **Villón Torres Victor Antonio**, estudiantes de la Carrera de Telecomunicaciones, Facultad de Sistemas y Telecomunicaciones de la Universidad Península de Santa Elena, previo a la obtención del título de Ingenieros en Telecomunicaciones, me permito declarar que luego de haber dirigido su desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por la cual, apruebo en todas sus partes, encontrándose apto para la evaluación del docente especialista.

Atentamente

A handwritten signature in blue ink, written over a horizontal line. The signature is stylized and appears to read 'Luis Amaya Fariño'.

Ing. Luis Amaya Fariño, Mgtr.

DOCENTE TUTOR

DECLARACIÓN AUTORÍA DEL ESTUDIANTE

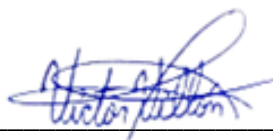
El presente Trabajo de Integración Curricular, con el título “**Desarrollo y diseño de un prototipo autosostenible geolocalizado implementando una red WSN para el rastreo de caprinos en un sector agrario de la comuna Zapotal**”, declaro que la concepción, análisis y resultados son originales y aportan a la actividad educativa en el área de Telecomunicaciones.

Atentamente,



Sr. Pérez Altamirano Cristian Geovanny

C.I. 1804376356



Sr. Villón Torres Víctor Antonio

C.I. 2450038506

DECLARACIÓN DE DOCENTE ESPECIALISTA

En mi calidad de Docente Especialista, del Trabajo de Integración Curricular “**Desarrollo y diseño de un prototipo autosostenible geolocalizado implementando una red WSN para el rastreo de caprinos en un sector agrario de la comuna Zapotal**”, elaborado por los señores **Pérez Altamirano Cristian Geovanny** y **Villón Torres Víctor Antonio**, estudiantes de la Carrera de Telecomunicaciones, Facultad de Sistemas y Telecomunicaciones de la Universidad Península de Santa Elena, previo a la obtención del título de Ingenieros en Telecomunicaciones, me permito declarar que luego de haber evaluado el desarrollo y estructura final del trabajo, éste cumple y se ajusta a los estándares académicos, razón por la cual, declaro que se encuentra apto para su sustentación.

Atentamente



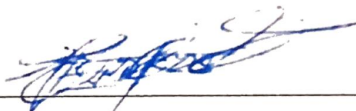
Ing. Vladimir García Santos, Mgtr.

DOCENTE ESPECIALISTA

TRIBUNAL DE GRADO



Ing. Ronald Rovira Jurado, Ph. D.
DIRECTOR DE LA CARRERA
TELCOMUNICACIONES



Ing. Vladimir García Santos, Mgtr.
DOCENTE ESPECIALISTA



Ing. Luis Amaya Fariño, Mgtr.
DOCENTE TUTOR_GUÍA UIC



Ing. Corina Gonzabay De la A, Mgtr.
SECRETARIA

DECLARATORIA DE RESPONSABILIDAD

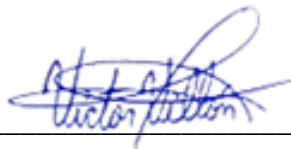
Quienes suscriben, **PÉREZ ALTAMIRANO CRISTIAN GEOVANNY** con C.I. 1804376356 y **VILLÓN TORRES VÍCTOR ANTONIO** con C.I. 2450038506, estudiantes de la carrera de Telecomunicaciones declaramos que el Trabajo de Titulación presentado a la unidad de Integración Curricular cuyo tema es “**Desarrollo y diseño de un prototipo autosostenible geolocalizado implementando una red WSN para el rastreo de caprinos en un sector agrario de la comuna Zapotal**” Corresponde y es de exclusiva responsabilidad de los autores y pertenece al patrimonio intelectual de la Universidad Estatal península de Santa Elena.

Atentamente,



Sr. Pérez Altamirano Cristian Geovanny

C.I. 1804376356



Sr. Villón Torres Víctor Antonio

C.I. 2450038506

AGRADECIMIENTO

Agradezco este proyecto a Dios que me ha dado las fortalezas necesarias para cumplir este proceso, también a mi tutor de tesis, Luis Miguel Amaya Fariño que ha invertido su tiempo y esfuerzos para que este proyecto culmine con satisfacción, sus guías han permitido que logremos obtener los resultados pertinentes de este proceso. Además, a mis mejores amigos Erick y especialmente a Cristian quien ha estado desde el principio hasta el término de este gran proyecto, sin duda este trabajo en equipo es un logro del que ambos estaremos orgullosos, ya que invertimos tiempo, esfuerzo y sobre todo constancia. De la misma manera, agradezco este escrito a la Universidad Estatal Península de Santa Elena y su cuerpo docente que me ha permitido llegar hasta donde estoy hoy en día académica y profesionalmente.

Víctor Antonio Villón Torres

Agradezco a Dios por permitirme llegar hasta este punto y brindarme de salud para alcanzar mis objetivos, a los docentes que han contribuido en mi formación académica, especialmente a mi tutor Ing. Luis Miguel Amaya por el aporte de conocimientos en el desarrollo de este proyecto y por la confianza para poderlo finalizar con éxito. Quería agradecer a mis mejores amigos Erick Suarez y Víctor Villon por estar siempre ahí para mí, especialmente durante los momentos difíciles sin importar el día o la hora, esto es una pequeña muestra de agradecimiento por todo el amor y apoyo que me han brindado a lo largo de los años.

Cristian Geovanny Pérez Altamirano

DEDICATORIA

En primer lugar, quisiera dedicarle a Dios que me ha brindado la oportunidad de llegar hasta donde estoy hoy en día, de igual forma a mis padres Alexandra Torres y Antonio Villon que han dado todo de sí mismos para convertirme en el profesional que soy actualmente, también a mi pareja Josselyn Cantos que me ha apoyado en este proceso tan importante. Todos y cada uno de ellos han influido en mi vida, por ello este proyecto es dedicado a ellos y sobre todo a mí que no me he rendido a pesar de los obstáculos que se me han presentado, dándome a entender que con esfuerzo y pasión todo es posible.

Víctor Antonio Villón Torres

Cuando miro hacia atrás en el transcurso de la vida universitaria me lleno de gratitud por los que me han apoyado en este camino, es por aquello que, le dedico este logro a quien ha sido el pilar de fortaleza mi madre Rosario Altamirano con su amor y apoyo me han ayudado en momentos más difíciles de mi vida, le tengo la mayor deuda de gratitud, a mi pareja Karen Gonzabay por ser comprensiva durante este tiempo y brindarme apoyo emocional y moral cuando fue necesario, de igual forma le dedicó a mis mejores amigos Víctor y Erick, a mis hermanas Clara y Mónica que me han apoyado a lo largo de este proceso, ofreciéndome palabras de aliento cuando fue necesario, sin su ayuda, la realización de esta tesis no hubiera sido posible.

Cristian Geovanny Pérez Altamirano

RESUMEN

El presente trabajo de integración curricular planteó el desarrollo y diseño de un prototipo autosostenible geolocalizado que implemente una red WSN para el seguimiento de caprinos. El prototipo se fundamentó en la tecnología Zigbee y ser un módulo autosostenible capaz de no requerir ninguna fuente de energía externa. Además, el sistema incluirá capacidades de rastreo GPS e integración con Ubidots para la visualización de datos. Adicionalmente, el sistema emite notificaciones vía SMS y correo electrónico, así como a la ubicación en mapa de cada nodo supervisor a través de una aplicación móvil.

Finalmente, el proyecto incluye una fase de implementación para probar el prototipo en escenarios del mundo real considerando que tiene una potencia autónoma de 12,96 horas sin la ayuda de paneles solares, tras lo cual procede a recargar autónomamente la batería cuando se requiera, en consecuencia, se concluyó que el prototipo implementado es de utilidad para los microempresarios que laboran en la industria ganadera. Se recomienda implementar e investigar nuevas tarjetas de desarrollo que permitan reducir el tamaño del prototipo y agregar sensores para medir otras variables.

Palabras Clave: WSN, Geolocalización, Sistema de Notificaciones, Rastreo de caprinos

ABSTRACT

The present work of curricular integration proposed the development and design of a geolocated self-sustaining prototype that implements a WSN network for the monitoring of goats. The prototype was based on Zigbee technology and being a self-sustaining module capable of not requiring any external power source. Additionally, the system will include GPS tracking capabilities and integration with Ubidots for data visualization. Additionally, the system issues notifications via SMS and email, as well as the map location of each supervisor node through a mobile application.

Finally, the project includes an implementation phase to test the prototype in real world scenarios considering that it has an autonomous power of 12.96 hours without the help of solar panels, after which it proceeds to autonomously recharge the battery when required, in Consequently, it was concluded that the implemented prototype is useful for microentrepreneurs who work in the livestock industry. It is recommended to implement and investigate new development cards that allow reducing the size of the prototype and adding sensors to measure other variables.

Keywords: WSN, geolocation, notification system, goat tracking.

ÍNDICE DE CONTENIDO

DECLARACIÓN DE DOCENTE TUTOR.....	ii
DECLARACIÓN AUTORÍA DEL ESTUDIANTE	iii
DECLARACIÓN DE DOCENTE ESPECIALISTA.....	iv
TRIBUNAL DE GRADO	v
DECLARATORIA DE RESPONSABILIDAD.....	ii
AGRADECIMIENTO	iii
DEDICATORIA	iv
RESUMEN.....	v
ABSTRACT.....	vi
ÍNDICE DE CONTENIDO	vii
ÍNDICE DE FIGURAS.....	x
ÍNDICE DE TABLAS.....	xii
DEFINICIONES Y ACRÓNIMOS	xiv
Definiciones	xiv
Acrónimos.....	xv
ÍNDICE DE ANEXOS	xvii
INTRODUCCIÓN	1
CAPÍTULO I EL PROBLEMA.....	2
Planteamiento del problema.....	2
Descripción del proyecto.....	3
Objetivos del proyecto	4
Objetivo general	4
Objetivos específicos	4
Justificación.....	5
Alcance.....	7
Resultados esperados	8
Capítulo II MARCO TEÓRICO	10
Antecedentes	10
Estado del arte sobre el comportamiento individual de caprinos.....	11
Introducción del estado del arte	11
Desarrollo del estado del arte.....	12
Conclusiones del estado del arte	18

Análisis de tecnologías inalámbricas de baja frecuencia	18
Tecnología Bluetooth.....	19
Tecnología Zigbee.....	21
Tecnología Wi-Fi	22
Tecnología LoRaWAN	23
Comparación entre tecnologías inalámbricas de bajas frecuencias	24
Geolocalización.....	26
Usos de la geolocalización	27
Geolocalización en dispositivos móviles	27
Entornos de desarrollo integrado para las plataformas móviles.....	28
Android Studio.....	28
Microsoft Visual Studio	31
Java Development kit.....	32
Softwares para el desarrollo del prototipo autosostenible geolocalizado	33
Ubidots	33
XCTU.....	34
Arduino IDE.....	35
Red de sensores inalámbricos	36
Tarjetas de Desarrollo	38
Arduino UNO.....	38
NodeMCU.....	38
Raspberry Pi.....	40
Comparativa de las tarjetas de desarrollo	41
Módulos complementarios	42
Módulos Xbee	42
Xbee Explorer	44
Módulo GPS.....	44
Panel solar	45
Batería de Litio.....	45
Capítulo III MARCO METODOLÓGICO	47
Tipo de investigación	47
Investigación exploratoria.....	47
Investigación aplicada.....	47
Población y muestra	48

Simulación Desarrollada en Sketchup	48
Diseño de implementación de prototipo	50
FASE 1: Estudio del comportamiento de una especie gregaria (caprinos)...	50
FASE 2: Simulación de la ubicación de los materiales utilizados para el desarrollo del prototipo	51
FASE 3: Verificación de la red de sensores inalámbricos	51
Diagrama de concepción general	52
Diagrama de Bloque perteneciente al nodo Supervisor	53
Diagrama de Bloque perteneciente al nodo Principal	54
Esquema de conexión de los nodos.....	55
Especificaciones de Arduino UNO	55
Especificaciones de NodeMCU	56
Especificaciones de Shield de Xbee para Arduino	58
Especificaciones de módulo de carga tp4056	59
Especificaciones de Batería de Litio 18650.....	60
Especificaciones del módulo Xbee S2C	62
Especificaciones GPS Ublox NEO-6M	63
Especificaciones del panel solar	65
Esquema de conexión de nodo Principal	67
Esquema de conexión del nodo Supervisor	67
Configuración de los módulos Xbee S2C	68
Configuración del Nodo Coordinador.....	69
Configuración del Nodo Supervisor	70
Fundamentos del código Arduino para los nodos	70
Desarrollo del código para el nodo coordinador	71
Desarrollo del código para el nodo supervisor.....	71
Ensamblaje del prototipo.....	72
Pasos para ensamblar el nodo coordinador	73
Pasos para ensamblar el nodo supervisor.....	73
Diseño estructural del prototipo.....	75
Conexión por software	76
Conexión con Ubidots.....	77
Conexión con aplicación móvil.....	82
Costos de implementación	83
Capítulo IV PRUEBAS Y RESULTADOS DE LA PROPUESTA	87

Pruebas de funcionamiento	87
Prueba de ubicación del prototipo.....	87
Prueba de alcance y geocerca.....	88
Consumo de Energía	92
Margen de error en el Almacenamiento de los datos	94
CONCLUSIONES.....	98
RECOMENDACIONES.....	100
ANEXOS	101
REFERENCIAS BIBLIOGRÁFICAS.....	112

ÍNDICE DE FIGURAS

Figura 1 Visualización de datos en un área delimitada por software.....	26
Figura 2 Logo de Android Studio	29
Figura 3 Tablero de paneles en Ubidots	34
Figura 4 Interfaz de XCTU	35
Figura 5 Gestor de bibliotecas de Arduino IDE.....	36
Figura 6 Ejemplo de una red inalámbrica de sensores.....	37
Figura 7 Arduino UNO	38
Figura 8 <i>ESP32</i>	39
Figura 9 Raspberry Pi	40
Figura 10 Módulo Xbee S2C	43
Figura 11 Módulo Xbee S2B	43
Figura 12 Módulo Xbee S2.....	44
Figura 13 Baterías de Litio	46
Figura 14 Diseño tentativo en 3D de nodo supervisor.....	49
Figura 15 Ubicación del nodo supervisor en el caprino.....	49

Figura 16	Diagrama de proceso para el diseño del prototipo	52
Figura 17	Diagrama de la topología general del prototipo	53
Figura 18	Diagrama del nodo supervisor.....	54
Figura 19	Diagrama de Bloque del nodo Principal.....	54
Figura 20	Arduino UNO	55
Figura 21	Esp32	57
Figura 22	Xbee Shield.....	58
Figura 23	Módulo de carga	59
Figura 24	Baterías 18650.....	61
Figura 25	Módulo Xbee S2C	62
Figura 26	GPS Ublox NEO-6M.....	64
Figura 27	Panel solar.....	66
Figura 28	Nodo coordinador	67
Figura 29	Nodo supervisor.....	68
Figura 30	Opciones de configuración del Xbee coordinador	69
Figura 31	Opciones de configuración del Xbee supervisor	70
Figura 32	Código del nodo coordinador	71
Figura 33	Código del nodo supervisor.....	72
Figura 34	Creación de PCB para el nodo coordinador	73
Figura 35	Parte 1 del ensamblaje del nodo supervisor.....	74
Figura 36	Parte 2 del ensamblaje del nodo supervisor.....	74
Figura 37	Diseño de estructura de nodos en Sketchup	75
Figura 38	Presentación final del prototipo.....	76

Figura 39 Credenciales del API Ubidots.....	77
Figura 40 Configuración del token en el ESP32.....	78
Figura 41 Dispositivo ESP32 en Ubidots	78
Figura 42 Condiciones de Geocercas.....	79
Figura 43 Delimitación de georreferencia	80
Figura 44 Modalidad de alerta por correo electrónico.....	81
Figura 45 Modalidad de alerta por SMS	81
Figura 46 ID del nodo supervisor 1	82
Figura 47 Tokens en código de la aplicación móvil	83
Figura 48 Primera prueba de ubicación	87
Figura 49 Segunda prueba de ubicación	88
Figura 50 Conectividad de los nodos con el servidor de Ubidots.....	89
Figura 51 Supervisor1 en mapa	90
Figura 52 Notificaciones de alerta del supervisor1	90
Figura 53 Supervisor2 en mapa	91
Figura 54 Notificaciones de alerta del supervisor2.....	91
Figura 55 Gráfica de continuidad para la obtención de datos.....	95
Figura 56 Gráfica de discontinuidad al obtener datos	95
Figura 57 Datos perdidos en la base de datos de Ubidots.....	96

ÍNDICE DE TABLAS

Tabla 1 Recopilación de la información	16
Tabla 2 Características de transmisión para la tecnología Wi-Fi.....	23

Tabla 3 Cuadro comparativo de tecnologías inalámbricas	24
Tabla 4 Desarrollo en Android Studio	30
Tabla 5 Cuadro comparativo de tarjetas de desarrollo.....	41
Tabla 6 Comportamiento de los caprinos.....	50
Tabla 7 Especificaciones de Arduino UNO	56
Tabla 8 Especificaciones ESP32	57
Tabla 9 Especificaciones de Xbee Shield	58
Tabla 10 Especificaciones del módulo de carga	60
Tabla 11 Especificaciones de Batería de Litio 18650.....	61
Tabla 12 Especificaciones de Xbee S2C.....	62
Tabla 13 Especificaciones de módulo GPS	64
Tabla 14 Especificaciones del panel solar.....	66
Tabla 15 Costo del nodo Supervisor	84
Tabla 16 Costo del nodo coordinador	85
Tabla 17 Costo total de implementación del prototipo	86

DEFINICIONES Y ACRÓNIMOS

Definiciones

Arduino UNO: Es una placa de microcontrolador que posee código abierto el cual está basado en el microchip ATmega328P y es desarrollado por Arduino.cc.

Datasheet: Es una ficha técnica elaborada a partir de un documento en forma de sumario que contiene la descripción de un objeto, material, proceso o programa de forma detallada.

ESP32: Se denomina así a la familia de chips SoC de bajo coste y consumo de energía, incorporada con tecnología Wifi y Bluetooth de modo dual integrada.

Explorer Xbee: Es un dispositivo que permite conectar y utilizar cualquiera de los módulos XBee directamente a través de un puerto USB.

Módulo XBee S2C: Es un módulo de bajo costo que permite la conexión inalámbrica entre dispositivos electrónicos. Permitirá crear redes punto a punto, punto a multipunto, broadcast e interconexión.

NodeMCU: Es una placa de desarrollo totalmente abierta, a nivel de software y de hardware.

Nodo principal: El nodo general administra el clúster y, por lo general, ejecuta los componentes principales de las aplicaciones distribuidas.

Nodo supervisor: Es aquel equipo que en las redes DLR (Device Level Ring), se encarga de fijar las condiciones para la comunicación entre los nodos.

Raspberry Pi: Es una serie de ordenadores de placa reducida de bajo costo, desarrollado en el Reino Unido por la Fundación Raspberry Pi, con el objetivo de poner el poder de cómputo en manos de personas de todo el mundo.

Sensor GPS: Estos sensores usan con el fin de determinar la posición desde tres segmentos para determinar la posición como satélites, estaciones terrestres, y sensores.

Shield: Es una placa de circuito impreso colocada encima de la placa Arduino y acoplada a ella conectando sus pines sin ninguna otra conexión externa.

Sketchup: Es un software de modelado y diseño gráfico 3D basado en rostros. Se utiliza para el modelado ambiental de planificación urbana, entre otros.

Ubidots: Es una plataforma de Internet que permite a las empresas crear aplicaciones IoT para transformar los datos de los sensores en conocimiento procesable.

Zigbee: Es la denominación de un conjunto de protocolos de alto nivel de comunicación inalámbrica que se utiliza dentro de la radiodifusión digital de bajo consumo.

Piconet: Es un ciberespacio en donde cuyos nodos están conectados vía Bluetooth.

Wi-Fi: Tecnología que permite conectar diferentes dispositivos informáticos a través de una red inalámbrica de banda ancha.

Bluestar: Arquitectura de red que mejora la estabilidad de los enlaces Wifi y Bluetooth sin interferencias.

Acrónimos

GPS: del inglés Global Positioning Satellites

mA: miliamperios

WSN: del inglés Wireless Sensor Networks

WPAN: Red inalámbrica de área personal.

BWG: del inglés Gateway Inalámbricos Bluetooth.

WLAN: del inglés Wireless Local Area Network.

IOT: del inglés Internet Of Things.

ISM: del inglés Industrial, Scientific and Medical frequency band

ÍNDICE DE ANEXOS

Anexo A Datasheet de Xbee/XBee-PRO S2C Zigbee.....	101
Anexo B Datasheet de Arduino UNO.....	102
Anexo C Datasheet de GPS NEO6M	103
Anexo D Datasheet de ESP32 Series.....	104
Anexo E Código del nodo coordinador	105
Anexo F Código del nodo supervisor 1	107
Anexo G Código del nodo supervisor 2	108
Anexo H Código del nodo supervisor 3	109
Anexo I Certificado de análisis antiplagio	110
Anexo J Código de la aplicación móvil localdots	111

INTRODUCCIÓN

Este prototipo autosostenible de red sensorial inalámbrica para geolocalizar especies caprinas en Zapotal pretende brindar una alternativa para mantener segura la inversión de los campesinos a través de mediciones en tiempo real que le permita al propietario de la parcela monitorear el ganado caprino desde cualquier punto. El desarrollo del presente prototipo se efectuó en cuatro capítulos, los cuales se detallan a continuación:

En el primer capítulo se plantea el problema que se necesita solucionar, la justificación, la cual indica las razones por las que se necesita desarrollar el prototipo, se proponen los objetivos que guiarán la creación de los prototipos autosostenibles geolocalizados que están siendo implementados por una red WSN.

En el segundo capítulo, se presentan los antecedentes investigativos relacionados con la temática a desarrollar; así como la fundamentación teórica involucrada de forma profunda en el tema propio, para finalmente dar paso a la propuesta de solución, en la que se narra brevemente la solución al problema de investigación.

De la misma forma el tercer capítulo, presenta la metodología básica de investigación utilizada para desarrollar este trabajo, describiendo su diseño, método, técnicas, costos en el equipamiento, población y muestra.

El cuarto capítulo comprende la discusión de los resultados, conclusiones y recomendaciones obtenidos a partir de la creación del prototipo autosostenible geolocalizado implementando una red WSN para el rastreo de caprinos en un sector agrario de la comuna Zapotal.

CAPÍTULO I EL PROBLEMA

Planteamiento del problema

El aumento constante y progresivo de la delincuencia que azota a el Ecuador se ha visto presente, no solo en zonas urbanas, sino también en zonas rurales, esto ha significado que aquellos sectores agrarios se vean afectados, debido a que, al tener grandes expansiones territoriales, resulta difícil mantener un control de los rebaños, ya que, estos caminan libres en sus parcelas.

El robo de estas especies resulta ser una pérdida significativa para los dueños de estos rebaños, ya que el precio aproximado de cada uno ronda por los \$450, debido a que en la práctica real no solo se pierde un solo espécimen sino varios estos valores aumentan. La desventaja que produce esta situación en los moradores de la comuna es muy alta, y tomando en cuenta que los propietarios de este tipo de parcelas solo se lucran de la producción de sus animales, esto indica que un déficit en su rebaño no es opción dentro de este tipo de negocio.

En zonas rurales han existido varias medidas contra la desaparición de sus animales, siendo los líderes comunales quienes emiten proyectos de seguridad, no obstante, no logran obtener los resultados esperados, debido a que fueron métodos arcaicos, por lo que no se conseguía una recopilación de información en tiempo real dentro de los límites territoriales en sus parcelas, sino que el tiempo en que estos datos eran receptados por los administradores de las propiedades resultaban tardíos, dificultando los protocolos necesarios para emprender la búsqueda o rescate del rebaño.

El declive de la economía tras una pandemia obligó a que los propietarios de estas parcelas tengan menos empleados al cuidado y vigilancia de los animales, lo cual ha tenido como consecuencia que los animales se extravíen y que muy pocos sean recuperados, debido a que las parcelas no tiene un cerramiento a lo largo de toda su lindero, por lo cual el rebaño sale a recorrer terrenos aledaños del lugar, y se dan ocasiones en las cuales se alejan en busca de alimento, de tal forma que no encuentran el camino de regreso.

Descripción del proyecto

Dentro de la provincia de Santa Elena se encuentra la comuna Zapotal, en el recorrido por esta es posible evidenciar varios restaurantes que tienen como especialidad el seco de chivo, es por esta razón que varios dueños de estos negocios tienen su propio rebaño de chivos, pero debido al abigeato que existe en la zona, provoca pérdidas significativas a la economía de estas personas, es por esta razón que se ve necesario la implementación de un prototipo capaz de monitorear y rastrear al ganado o animales de granja, ya que, mediante este proyecto además de tener la posición del animal en tiempo real, se pueden añadir sensores y aplicaciones con las cuales se obtendrán datos para la mejorar de atención hacia el animal.

La forma en la que se realizará este proceso es por medio de una red WSN, es decir, una red de sensores inalámbricos, su función principal dentro del proyecto es la comunicación entre los nodos que se emplean en el diseño del hardware, que son nodo principal y nodo supervisor.

El nodo principal será el encargado de almacenar la información recolectada y subirla a la nube con que se trabajará, para así mostrarla en una interfaz final al administrador. Este módulo estará compuesto de placas de desarrollo, las mismas que contarán con la programación necesaria para el funcionamiento, las tarjetas son: NodeMCU y módulo Xbee S2C.

El diseño del nodo supervisor está enfocado en la geolocalización y autosostenibilidad, este recolecta los datos de longitud y latitud de un solo ejemplar para así obtener su posición en tiempo real, para la elección de las tarjetas del módulo se debe tener en cuenta su bajo consumo de energía, la relación calidad-precio, cantidad de pines disponibles y su compatibilidad con otros componentes, por esta razón hará uso de: Arduino UNO, módulo Xbee, Shield Xbee, módulo GPS, panel solar, módulo de carga y baterías recargables.

Objetivos del proyecto

Objetivo general

Diseñar e implementar un prototipo autosostenible direccionado al sector agrario que permita la geolocalización de especies caprinas basado en redes de sensores inalámbricos a través de una aplicación móvil cuya función sea monitorear las especies.

Objetivos específicos

Realizar el estado del arte del comportamiento individual de caprinos el cual brinde la información necesaria para una ubicación adecuada del nodo supervisor.

Comparar mediante la revisión literaria los tipos de tecnología inalámbrica más utilizadas que abarquen la comunicación de bajas frecuencias considerando el consumo de energía para definir los tiempos de duración y cargas de la batería.

Realizar el diseño final del prototipo colocado en la especie caprina utilizando el software de simulación en 3D Sketchup.

Configurar las tarjetas electrónicas que permitirán el acceso a los datos del prototipo a la nube implementando la tecnología inalámbrica que más se adecue a la zona de estudio como medio inalámbrico de transmisión.

Diseñar una aplicación para el sistema operativo Android que permita la visualización de los datos recolectados por los nodos supervisor y principal con el fin de conseguir vigilancia remota de caprinos.

Implementar la modalidad de alerta y actualización continua de la ubicación cuando la especie caprina salga del área delimitada.

Implementar un prototipo con tecnología inalámbrica en especies caprinas que permita su geolocalización a través de una red de sensores.

Justificación

Considerando que en el Ecuador no existe ninguna empresa que brinde servicios de rastreo de ganado, a pesar de que la desaparición de estos animales es un problema que afecta a todo el país, se considera necesario el desarrollo de un prototipo autosostenible para la geolocalización basado en redes de sensores inalámbricos y tecnologías web.

La sistematización de los procesos permite incorporar herramientas para la toma de decisiones basadas en la información recopilada y procesada. Esto permite reemplazar a operadores humanos en tareas que requieren precisión y son de carácter rutinario. La automatización influye en la disminución de costos asociados a la contratación de personal y tiempos cortos de recolección de información permitiendo una mejor toma de decisiones.

En diferentes investigaciones se ha propuesto el desarrollo de elementos tecnológicos como lo son las redes de sensores inalámbricos o sus siglas en inglés WSN (Wireless Sensor Network) para el monitoreo de caprinos, puesto que en las comunas donde se encuentran las parcelas no cuentan con las estructuras necesarias para la comunicación. Se requiere un bajo consumo de energía y las redes de sensores son una adecuada elección tomando en cuenta estos aspectos.

En contexto con lo antes mencionado, en el proyecto de titulación surge la pregunta ¿Será posible generar un prototipo autosostenible para la geolocalización basado en redes de sensores inalámbricos y tecnologías web?

La implementación de un proyecto prototipo que emplee la tecnología WSN en la comuna Zapotal, se clasifica dentro de las tecnologías del futuro, ya que, todavía no se alcanza el máximo desarrollo en trabajos de ingeniería a bajas frecuencias, en los próximos años, tanto por el uso de dispositivos móviles en la vida cotidiana como el uso de nuevas tecnologías inalámbricas.

El desarrollo del presente proyecto permitirá a la comuna Zapotal contar con un Prototipo de geolocalización basado en redes de sensores inalámbricos y tecnologías web para la administración de datos en tiempo real.

Alcance

Este proyecto tiene como finalidad diseñar e implementar un prototipo autosostenible geolocalizado implementando una red de sensores inalámbricos para el rastreo de caprinos en un sector agrario, esto será desarrollado por medio de tarjetas electrónicas como lo son el NodeMCU, raspberry pi y módulo GPS, siendo estas las de mayor importancia, que en conjunto formarán los nodos principal y supervisor, además de implementar tecnologías web para la visualización de datos y gestión de manera remota.

Para la administración de datos se desarrollará un aplicativo capaz de mostrar la ubicación en tiempo real de los caprinos y alertar si alguno de los animales ha salido del área delimitada, para ello se emplea el software Ubidots, la cual es una plataforma direccionada a la gestión de datos y al manejo de una gran cantidad de sensores.

Con este sistema se pretende combatir la problemática de extravío del rebaño, facilitar la búsqueda del animal por abigeato y a la reducción de empleados que se generaron en los sectores agrarios debido al declive de la economía tras una pandemia.

El proyecto ejecutará los procesos acordes con los objetivos propuestos, sin embargo, existen funcionalidades con las que se puede ampliar el objeto de estudio, por ejemplo, el estado del animal por medio del ritmo cardiaco, nodos especiales para hembras en las que monitoreen el periodo de gestación o llevar el control del periodo de

vida del caprino, pero todos estos escenarios exceden a la problemática inicial a la que se pretende dar solución.

Resultados esperados

En el presente trabajo se busca ampliar las alternativas que se puedan utilizar con respecto a la desaparición y abigeato de caprinos en la comuna zapotal, por medio de una red inalámbrica de sensores, en donde al trabajar con bajas frecuencias y transductores permite una forma de comunicación para el seguimiento, almacenamiento de datos y ubicaciones, además de utilizar nodos sensoriales siendo estos pequeños, livianos y portátiles.

Con el análisis de comportamiento de caprinos, se pretende entender las posiciones en las que el animal comúnmente se encuentra para de esta forma encontrar un sitio adecuado para la colocación del nodo supervisor.

La comparación entre los tipos de tecnología inalámbrica permitirá elegir una que se adecue a la zona de estudio y de esta manera que la conexión sea óptima entre el nodo supervisor y nodo principal.

El diseño de una aplicación móvil permitirá al usuario final tener acceso a los datos recolectados mediante los nodos, y de esta forma saber si su rebaño se encuentra dentro de su propiedad.

Al seleccionar correctamente la tecnología inalámbrica a utilizar se obtiene un mayor rendimiento, ya que, todas las tarjetas electrónicas serán compatibles y debido a que se realiza un análisis del consumo de energía de las tarjetas se consigue una mayor autonomía al momento de su funcionamiento debido a que en general tiene un consumo

mínimo en mA por cada actualización de ubicación, haciendo que la duración del prototipo sea mayor.

El diseño del prototipo realizado en Sketchup deberá dar una idea clara de la ubicación óptima en la especie caprina, además de brindar información acerca del tamaño del prototipo.

La configuración es decir el ingreso de código a las tarjetas electrónicas tiene la finalidad de ejecutar una serie de comandos de programación dirigido a los puertos de cada placa para decirle cómo deben actuar entre microcontroladores.

La serie de pruebas deberán corroborar el funcionamiento y precisión del prototipo geo localizador, para posteriormente en su respectivo análisis brindar los resultados de su margen de error.

Capítulo II MARCO TEÓRICO

Antecedentes

Como resultado de la revisión literaria sobre el posicionamiento y monitoreo de caprinos en el sector agrario se pudo sintetizar la información en base a investigación científica que ponen en evidencia los resultados y las observaciones de estudios que guardan relación con las variables, es así, que se tomara en cuenta diversos puntos de vista a nivel global y nacional, estas investigaciones proporcionan técnicas, métodos de última generación enmarcado en el rastreo de caprinos, reconociendo que la creación de formas de posicionamiento es fundamental para llevar un control eficiente. A nivel internacional una de las investigaciones más relevantes es la rastreabilidad de caprinos. De acuerdo con Jorner et al. (2021):

La integración de las tecnologías WSN con GPS ha permitido el desarrollo de sistemas de seguimiento en tiempo real para el seguimiento de animales y el control sanitario, que permiten una gestión más eficaz de los rebaños caprinos. Estos sistemas utilizan dispositivos de identificación electrónica que, a través de sensores y comunicación inalámbrica, brindan información sobre la ubicación, estado fisiológico y comportamiento de los animales, facilitando la detección de problemas de salud, la prevención de enfermedades y la mejora de la productividad.

Esta investigación sugiere que la utilización de tecnología de monitoreo de animales, como las redes inalámbricas de sensores, puede ser una herramienta útil para la gestión efectiva de rebaños de caprinos. Además, la cita resalta la importancia de

estas tecnologías para la detección y prevención de enfermedades, lo que puede contribuir a mejorar la salud de los animales y la productividad de los rebaños. Desde el punto de vista de Llario et al. (2013):

El sistema propuesto es capaz de detectar las actividades sospechosas en los corrales de ovejas y cabras mediante el uso de una red de sensores inalámbricos inteligentes. Los sensores están colocados en las cercas de los corrales, y envían alertas a un sistema central si se detecta alguna actividad sospechosa. Esto permite a los productores tomar medidas preventivas para proteger a sus animales y reducir los daños causados por los ataques de animales depredadores.

Este estudio es relevante porque destaca la importancia de la tecnología en la protección de los rebaños de ovejas y cabras. Los ataques de animales depredadores son una amenaza constante para estos animales, y la detección temprana puede ser crucial para protegerlos. El uso de una red de sensores inalámbricos inteligentes, como se describe en este estudio, puede ayudar a detectar estos ataques de manera oportuna y reducir los daños causados.

A nivel nacional se puede evidenciar que no existen estudios del desarrollo y diseño de un prototipo autosostenible geolocalizado implementando una red WSN para el rastreo de caprinos en zonas rurales, sin embargo, el avance de la tecnología ha permitido que se le dé un uso semejante para otras especies como las domésticas.

Estado del arte sobre el comportamiento individual de caprinos

Introducción del estado del arte

Pocas son las especies animales domésticas que llegan a obtener potencial a nivel regional y local en temas de nutrición e ingresos económicos, como los caprinos. La adaptabilidad de esta especie a climas variados, aunados en su capacidad de producción de leche, carne y su fácil manejo debido a su docilidad, hacen del chivo, un animal de gran futuro y valor que mejora el nivel de vida del pequeño productor.

El estudio del comportamiento de los animales es realizado mediante la etología, la cual es la ciencia que se basa en el análisis del comportamiento tanto de animales domésticos como silvestres. La conducta de los caprinos es el reflejo de establecer una serie de relaciones entre ellos y el ambiente que los rodea.

El comportamiento de una especie tiene en su origen dos conductas básicas: la conducta innata, que se encuentra constituida diversos patrones instintivos pertenecientes a una especie, que es heredada y desarrollada de igual forma por todos los miembros de una especie determinada, es común verla presente en conductas de alimentación, reproducción, entre otros, mientras que la conducta adquirida es aquella que comprende comportamientos obtenidos en el transcurso de la vida, a través del autoaprendizaje por medio de experiencias tanto positivas como negativas y que varía entre individuos de la misma especie.

Desarrollo del estado del arte

Las especies caprinas hace aproximadamente diez mil años fueron domesticadas, existe constancia de vestimenta usada hace más de cuatro mil años en la que se aprecia fibras caprinas. Se puede observar cabras con vellon tipo Angora en el Estandarte de la antigua ciudad de Ur (Irak) lo cual nos remonta al siglo XXVI a.C., adicionalmente se encontraron tablillas de arcilla que constan del siglo XXI

a.C., en Umma (Irak) que contenían escrituras cuneiformes sumerias. (Mueller et al., 2018)

Existe evidencia que afirma que la cabra se origina en el neolítico, sin embargo, esta fue relacionada con la humanidad hasta los años 8 000 A.C. aunque a pesar de la antigüedad los caprinos no han tenido tanta evolución a comparación de otras especies domésticas, por lo que esto implica que sus características anatómicas y de comportamiento no han variado con el paso del tiempo. Desde el punto de Ducoing Watty (2016):

Actualmente debido a los avances que se han tenido en ciertas investigaciones se ha podido aclarar el papel de la cabra, planteado como el verdadero destructor al ser humano siendo la causal principal el situar al caprino en ciertos lugares no desarrollado por otras especies.

Los caprinos debido a su agresividad y capacidad de supervivencia en lugares inhóspitas, fue considerado como el principal factor causante de la erosión, sin embargo investigaciones han demostrado que el destructor principal es el ser humano, por otro lado los caprinos cuentan con la capacidad de ramonear constantemente para de esta forma poder consumir sus alimentos, adicionalmente, aquello le otorga un lugar en los rumiantes, dado que accede a porciones vegetales diferenciándose así del bovino y ovino. En un estudio realizado por Hernández Acosta (2014) indica:

Se pudo concluir que gran parte del rebaño realizan rumia echados, sin embargo, depende de la seguridad que encuentren en el medio este mantiene la posición,

de lo contrario aumenta la cantidad de animales que rumean parados y con estado de alerta.

Dado a que los caprinos son especies con un alto grado de supervivencia, esto influye en sus hábitos al momento de rumiar, debido a que la postura y la tranquilidad con la que este se alimente dependerá de si el sitio en el que se encuentren lo consideran seguro.

Por otra parte Helguero & Correa (2005) manifiestan “Comúnmente los caprinos reposan echados, sin embargo, el tiempo de descanso varia en torno a la estación climática y el lugar que seleccione para descansar, normalmente se agrupan dentro del corral durante la noche”.

Para entender el comportamiento cotidiano de una especie caprina, el docente de Fisiología Animal Santiago Helguero da a entender que son gregarios y se caracterizan por estar alerta en la mayoría de los ambientes en que se encuentres e indistintamente de la actividad que esté realizando, considerando que la gran parte del tiempo se encuentran echados.

“Si la especie está bien cuidada, tal vez podrían comenzar a ser sexualmente activos después de los 5 meses de edad. Sin embargo, las cabras hembras comenzarán a aparearse cuando alcancen el 75% de su peso adulto” (Solíz et al., 2014)

En los caprinos machos la pubertad también varía según las condiciones en las que se desarrolla, ya que es el factor nutricional el más importante para el inicio temprano de la pubertad. Cuando la cría tiene entre 3 y 4 meses, ya está produciendo esperma, pero el deseo sexual llega más tarde, junto con la erección de su órgano

reproductor. Desde otro punto de vista acerca del comportamiento de los caprinos es expresado por Rodríguez & Rodríguez (2017):

La raza caprina tiene el mayor porcentaje de concepción de junio a octubre y durante la temporada de apareamiento los machos se vuelven más agresivos, golpeando cabeza a cabeza o en cualquier lugar, el contacto puede variar desde un suave empujón hasta un audible golpe en la cabeza para evitar que otro macho invada su territorio. Por otro lado, la presencia de una hembra en celo está marcada por el estiramiento del cuello del macho en un intento de captar el olor de la hembra en el aire.

Tal como Rodríguez & Rodríguez menciona, los caprinos son muy agresivos en época de reproducción, dado que usan sus cuernos para golpear, su cabeza es la que sufre el daño del impacto al dar un empujón o un golpe entre sus cabezas.

Piedra (2022) plantea que “La reacción de Flehmen es otro comportamiento sexual en las cabras donde la boca está abierta, el labio superior se encuentra levantado, las encías están expuestas y la cabeza y el cuello generalmente está extendido”.

Las pautas comportamentales expresadas por la Dra. Piedra Aguirre mencionan que pese a la expresión facial que el caprino tiene al estar cerca de una hembra en época de apareamiento, su cuello generalmente esta extendido,

La recopilación del estado del arte presentado a continuación utiliza el modelo propuesto por Ana Saldaña, el cual compila la información más relevante de las fuentes de información utilizadas:

Tabla 1*Recopilación de la información*

Estado del arte				
Institución	Autor	Año	Título	Resumen
Revista de Investigaciones Agropecuarias	Mueller, J. P., Taddeo, H. R., Abad, M. I., & Debenedetti, S.	2018	Revisión sobre el origen y el desarrollo de la producción de caprinos de Angora en Argentina	Las especies caprinas hace aproximadamente diez mil años fueron domesticadas, existe constancia de vestimenta usada hace más de cuatro mil años en la que se aprecia fibras caprinas.
XVI Semana de la Capricultura	Ducoing Watty, Andres	2016	Introducción a La Caprino Cultura	Los caprinos debido a su agresividad y capacidad de supervivencia en lugares inhóspitas, fue considerado como el principal factor causante de la erosión.
Revista de la Facultad de Agronomía	Hernández Acosta, Iván	2014	Nota sobre observaciones hechas a rebaños de cabras criollas ramoneando	Gran parte del rebaño realizan rumia echados, sin embargo, depende de la seguridad que encuentren

			un bosque seco tropical	en el medio este mantiene la posición.
REDVET. Revista Electrónica de Veterinaria	Helguero, Santiago Pedro Correa, Julio	2005	Pastoreo caprino en el monte formoseño (Argentina)	Comúnmente los caprinos reposan echados, sin embargo, el tiempo de descanso varia en torno a la estación climática.
Entorno Ganadero	Solíz Pablo, Estrada Jesús, Rodríguez Kurt	2014	Manejo Reproductivo De La Cabra Características Particulares De Las Cabras	Si la especie está bien cuidada, tal vez podrían comenzar a ser sexualmente activos después de los 5 meses de edad.
UPR – Colegio de Ciencias Agrícolas	Rodríguez Abner, Rodríguez Héctor	2017	Comportamiento de las cabras en pastoreo	Durante la temporada de apareamiento los machos se vuelven más agresivos, golpeando cabeza a cabeza o en cualquier lugar.
UNIVERSIDAD TÉCNICA DE BABAHOYO	Piedra Angela	2022	Estudio bibliográfico del uso de vitaminas A, C, E y su influencia en el Comportamiento Sexual en especies	La reacción de Flehmen es otro comportamiento sexual en las cabras donde la boca está abierta, el labio superior se encuentra

menores productivas (ovinos, caprinos y conejos)	levantado, las encías están expuestas y la cabeza y el cuello generalmente está extendido
--	--

Conclusiones del estado del arte

De acuerdo con la realización del estado del arte se pudo comprender el comportamiento de los caprinos, de esta forma el diseño del prototipo se adecuará a sus características fisiológicas y comportamentales, esto para asegurar su funcionalidad y practicidad, sin interrumpir sus hábitos generales el mejor sitio para colocar el prototipo es el cuello (nuca) lo cual garantiza un óptimo funcionamiento del sistema autosostenible desarrollado con paneles solares.

Análisis de tecnologías inalámbricas de baja frecuencia

Para la revisión literaria se deben consultar varias tecnologías inalámbricas que trabajan a bajas frecuencias, ya que en la actualidad los sistemas de monitoreo remoto con nodos sensores utilizan tecnologías inalámbricas como Bluetooth, Zigbee, WI-FI y LoRaWAN, lo cual ha permitido expandirse a diversos campos de desarrollo tecnológico de software y hardware que permiten a los dispositivos electrónicos enviar información por internet, esto se conoce como el “internet de las cosas” (IOT).

Debido a la importancia que han tomado las redes inalámbricas, ha surgido el interés de estudiar las redes inalámbricas de sensores. (WSN), en este apartado se pretende dar un análisis comparativo sobre las tecnologías Wireless de baja frecuencia.

Para hablar de las redes de sensores inalámbricos o WSN es necesario tener clara su conceptualización y según Zennaro (2010):

Se definen las WSN como una red de característica autónoma integrada por una pequeña cantidad de nodos de sensores distribuidos espacialmente que se comunican entre sí a través de señales de radio para monitorear y comprender el mundo físico.

Tecnología Bluetooth

En los últimos años la necesidad de intercambiar información ha llegado a ser de suma importancia desde de inicio de las vastas tecnologías de telecomunicaciones. Las WPAN brindan la posibilidad de interconectar dispositivos de manera sencilla, sin impactar en el consumo de energía, rapidez y seguridad. Esto es lo que ha popularizado en los dispositivos móviles a tecnologías como el Bluetooth, Topología Física y Topología Virtual. Bluetooth es un método para la comunicación que usa señales de radio de corto alcance a fin de eliminar los cables a la hora de interconectar equipos de cómputo y dispositivos en general.

Cuando se trata de redes Bluetooth, directamente se la vincula a palabra Piconet, que no es más que el nombre de una red que consta de dispositivos que comparten información a través de Bluetooth. Según Avila & Reyes (2017):

El modelo de red de una Piconet es lo comprensible, ya que se tiene un dispositivo maestro y los restantes serán esclavos. La función del dispositivo maestro abarca escoger el hop adecuado para mantener el enlace, controlar el tráfico del canal y reservar los slots para los archivos que son transmitidos.

Esto significa que un dispositivo solo puede ser maestro en una piconet, pero puede formar parte de otras piconets como esclavo. Por lo tanto, se puede argumentar que cualquier dispositivo puede pertenecer a más de una piconet.

Dado que los dispositivos Wi-Fi y Bluetooth funcionan en la misma banda de 2,4 GHz, debe haber una manera de que coexistan sin interferir entre sí. Existe una arquitectura llamada Bluestar que garantiza una conexión Wi-Fi y Bluetooth estable y sin interferencias.

Dicha arquitectura funciona de la siguiente manera: Existen unos dispositivos Bluetooth llamados Gateway Inalámbricos Bluetooth (BWG) también miembros de la WLAN, habilitando así a dispositivos de bajo costo y corto alcance acceder a la infraestructura global de Internet mediante el uso de transmisores Wifi de alta potencia. Cabe decir que es posible que dispositivos Bluetooth accedan a internet a través de redes WAN como LTE, WiMAX, entre otras.

Bluetooth es una tecnología bastante completa y eficiente en cuanto a transferencia de información en cuanto a redes de área personal (WPAN). Debido a sus características intrínsecas de bajo coste de producción, sencillez de implementación del estándar, entre otras se perfila como la solución de conectividad predilecta para dispositivos móviles.

En cuanto a la carga energética de bluetooth esta se desprende dependiendo de su tipo, puesto que, según los datos específicos de Bluetooth Smart, el consumo máximo de esta tecnología es inferior a 15 mA y el consumo de energía es de 0,01 W a 0,05 W. En comparación con la conexión Bluetooth clásica, tiene el consumo más alto de 30mA,

que es el doble y el consumo alcanza 1W. El bluetooth consume alrededor de un 4% de la batería. En ese sentido, apagarlo puede agregar otros 10 a 15 minutos para un dispositivo que normalmente permitiría 5 horas de uso.

Tecnología Zigbee

ZigBee es un protocolo de comunicación inalámbrico avanzado de baja potencia basado en el estándar IEEE 802.15.4 (red de área personal inalámbrica, WPAN). El nombre dado a este tipo de dispositivos cumple con la finalidad de indicar en una sola palabra dos de los conceptos principales: su pequeño tamaño y la idea de que pueden estar distribuidos en cualquier lugar, cosa que es posible gracias a que son dispositivos autónomos que funcionan con baterías muy similares a las de los teléfonos celulares y que permiten ser cargados por paneles solares en el momento que así se requiera. De acuerdo con Sikora & Groza (2005):

Zigbee se desarrolló para satisfacer la creciente demanda de capacidades de redes inalámbricas en una variedad de dispositivos de bajo consumo. En esta industria, se utiliza en la próxima generación de fabricación automatizada con pequeños transmisores en cada dispositivo, lo que permite que los dispositivos se comuniquen con una computadora central.

Es así, como la tecnología Zigbee se ha ido ampliando a los demás sectores de las industrias gracias a las nuevas generaciones de automatización, y el estudio realizado por Sikora & Groza se aprecia como resulta ventajosa su aplicación por su variada cantidad de dispositivos emplear.

En cuanto al uso energético es correcto decir que Zigbee consume 30mA en transmisión y 3 μ A en reposo, frente a los 40mA en transmisión y 200 μ A en reposo que consume Bluetooth. Este menor consumo se debe a que el sistema Zigbee está en modo reposo la mayor parte del tiempo, mientras que con la comunicación Bluetooth esto no es posible y siempre está transmitiendo y/o recibiendo.

Tipos de dispositivos Zigbee

En lo que respecta a las redes de dispositivos Zigbee se puede evidenciar una clasificación delimitada por el coordinado, router caracterizados por ser dispositivos de función completa, además, una tercer clasificación denominada dispositivo final posee la característica de ser un dispositivo de función reducida. Fernández (2019):

Una red ZigBee consta de tres tipos de dispositivos, a saber, coordinadores, enrutadores y dispositivos finales. El coordinador es la unidad más compleja y toda red debería tener uno. Su función es gestionar la red y los caminos que deben tomar los dispositivos para conectarse entre sí.

Es decir, estos dispositivos de coordinación deben tener memoria y poder de procesamiento porque almacenarán y administrarán los datos de la red. Por otra parte, en la misma investigación se habla del modo router, en donde se menciona que se configura así cuando se requiere el mismo código de usuario para todos los nodos conectados, así mismo, se dice que el modo dispositivo final es aquel que solo recibe información y mantiene conexión con el nodo principal.

Tecnología Wi-Fi

Esta tecnología surge con la idea y la necesidad de comunicación mediante la conexión inalámbrica y que pueda ser compatible con diversos dispositivos móviles. Wireless Fidelity una marca comercial que hace referencia a la abreviación de Wi-Fi o conocida como WLAN soportan el estándar IEEE 802.11., estas proporcionan un rango de acceso inalámbrico hasta 100 metros.

Los estándares Wi-Fi 4 y Wi-Fi 6 utilizan la banda de frecuencia de 2,4 GHz. Tiene 3 canales de 20 MHz o 1 canal de 40 MHz, en cuyo caso parte del espectro asignado se reservará para esa banda si nuestro enrutador elige usar solo canales de 40 MHz, pero no afectará nuestra conexión. Actualmente consta de varios criterios, que se describen en la tabla 2.

Tabla 2

Características de transmisión para la tecnología Wi-Fi

IEEE	Banda	Tasa de transferencia
802.11.a	5GHz	48 Mbps
802.11. b	2,4 GHz	11 Mbps
802.11. g	2,4 GHz	54 Mbps
802.11n	2,4 GHz y 5GHz	600 Mbps
802.11.ac	5 GHz	1.3 Gbps

Nota: Obtenido de (Salazar, 2012)

Tecnología LoRaWAN

LoRaWAN es una especificación para redes de área amplia y baja potencia, LPWAN, que no solo proporciona interconexión entre objetos inteligentes sin necesidad

de instalaciones locales complicadas, sino que también ofrece una amplia libertad de uso para usuarios finales, desarrolladores y empresas que desean instalar sus propias redes IoT.

El estándar de red LoRaWAN cumple con las necesidades inherentes del Internet de las cosas, como una conexión bidireccional segura, bajo consumo de energía, largo alcance de comunicación, baja frecuencia de transmisión, baja velocidad de datos y servicios de ubicación.

Comparación entre tecnologías inalámbricas de bajas frecuencias

Tabla 3

Cuadro comparativo de tecnologías inalámbricas

Especificaciones	Bluetooth	Zigbee	Wi-Fi	LoRaWAN
Modulación	GFSK, HFSS	O-QPSK, BPSK, DSSS	DSSS, 64QAM, 256QAM	CSS
Rango	10-100 metros	10-100 metros	50-100 metros	10-20 Km
Banda Ancha	80 MHz	2Mhz	20, 40 MHz	125, 500 MHz
Tasa de transferencia	32 Mbps	802.15.4 25-250 kbps	802.11.b 11 Mbps 802.11n 600 Mbps	0.3-50 kbps

		Ad-Hoc			
Topología de Red	Punto a punto, punto a multipunto	punto a punto, estrella, árbol o malla.	Ad-Hoc, BSS, ESS	Red estrella, doble estrella	
Número de nodos por maestro	32000	64770	8191 en un mismo AP	-	
Frecuencia de Operación	2,402 GHz-2,480 GHz	868 MHz, 900-928 MHz, 2,4 GHz	2.4 GHz y 5GHz	868 MHz en Europa, 915 MHz en América, y 433 MHz en Asia	

Nota: Obtenido de (Adeva, 2021; Mannion, 2017; Pascual, 2012)

La comparación de tecnologías se muestra en la tabla 3, donde se puede ver que Bluetooth, ZigBee y Wi-Fi comparten el mismo rango de frecuencia de 2.4 GHz y, por otro lado, LoRaWAN y ZigBee también comparten el mismo rango de frecuencia de 868 MHz 900-928 MHz. Según la tabla comparativa, la elección de la tecnología ZigBee se debe a que tiene más nodos que Bluetooth y Wi-Fi, lo que nos permite consumir menos energía durante la transmisión de datos y alargar la vida de la batería.

LoRaWAN ha recibido más atención de los medios que Zigbee en los últimos años; sin embargo, ambas son redes IoT de baja potencia, y Zigbee también tiene un largo alcance que puede igualar o incluso superar el de LoRaWAN. Zigbee también es un protocolo más confiable y tiene una historia más larga en la industria de monitoreo y control, mientras que LoRaWAN se limita a sensores de bajo costo y bajo consumo.

Geolocalización

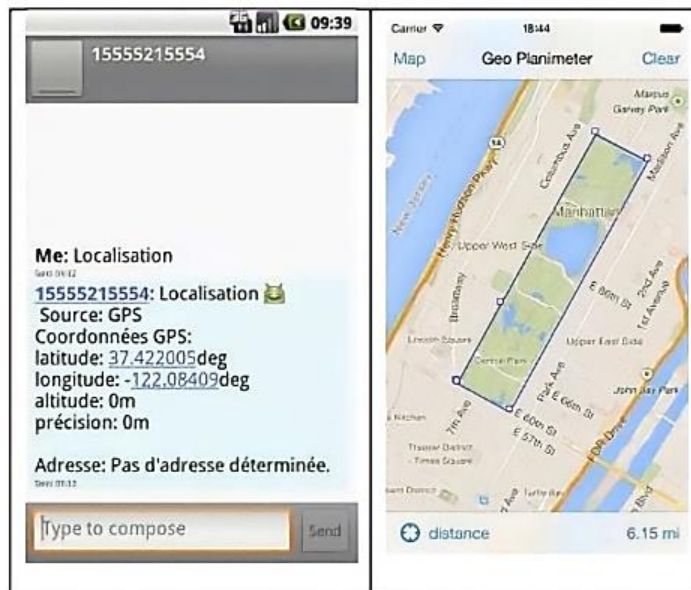
Se considera que la geolocalización juega un papel importante en el campo de la tecnología en la actualidad debido a la creciente necesidad de rastrear mercancías a diario, ya que, como señala Huerta et al. (2005):

El propósito de la geolocalización es determinar un punto en el espacio en el sistema de referencia mundial, obteniendo coordenadas de latitud y longitud, lo cual indica si está en cualquier lugar de la tierra, si está estacionario o en movimiento, y determina la ubicación de un objeto en cualquier momento.

Cuando se aclara la definición de geolocalización y se complementa con la Figura 1, podemos comprender la importancia y el impacto de este tema en dispositivos como teléfonos inteligentes, relojes, etc.

Figura 1

Visualización de datos en un área delimitada por software



Nota: (Fombona & Del Valle Mejias, 2018)

Usos de la geolocalización

El deseo humano de lograr la ubicación exacta de casi cualquier característica física para uso personal o diario ha hecho que la geolocalización sea más que una simple ubicación, debido a que ofrece diferentes aplicaciones, como menciona Fernández Lobo (2021) “El desarrollo de las ciudades inteligentes ha encontrado su camino en la geolocalización, que conecta la seguridad y el marketing en igual medida, todo por la capacidad de visualizar el movimiento de los usuarios en su espacio físico.”.

La geolocalización es una herramienta que se mejora día a día y que, gracias a sus ventajas, ofrece nuevas posibilidades de aplicación, *la* geolocalización le permite rastrear personas, animales u objetos, *las* aplicaciones reales se reflejan en los servicios de *delivery*.

Geolocalización en dispositivos móviles

La geolocalización en smartphones permite ubicar el dispositivo en cualquier lugar de la superficie terrestre en función de las coordenadas geográficas, que se pueden obtener de señales de Wi-Fi, GPS, torres celulares.

Es por ello que varios campos de desarrollo, incluidos los de la ecología y la biología, han sido de gran interés, ya que permite crear rutas habituales para una especie, por lo que es recomendable utilizar estas bases como punto de referencia, esto se plantea utilizar estas bases para aplicarlos a las especies caprinas en las zonas rurales para evitar su extinción y así promover la cría y cría segura “La información sobre la distribución geográfica del ganado, las preferencias de hábitat y los sitios preferidos abre nuevas oportunidades para el manejo de las poblaciones de ganado y vida silvestre con fines de conservación o producción”. (Lomillos et al., 2017)

Entornos de desarrollo integrado para las plataformas móviles

El desarrollo de software enfocados a dispositivos móviles presenta desafíos debido a las características únicas que se implementan. La necesidad de abordar diferentes plataformas de red, estándares, protocolos y tecnologías; aunque está en constante evolución, las capacidades limitadas de los dispositivos y los requisitos de adopción del mercado son solo algunos de los desafíos que deben abordarse. Esta es la razón por la que el desarrollo de software móvil es muy diferente del desarrollo de software tradicional.

Android Studio

Es un entorno de desarrollo integrado (IDE) se basa en el motor IntelliJ IDEA, ofrece un potente editor de código y muchas funciones para aumentar la productividad

del desarrollo de aplicaciones. Además, proporciona un sistema de compilación flexible, emuladores muy rápidos y herramientas para solucionar problemas de rendimiento y compatibilidad, estos detalles han sido representados por medio de su logo que se puede ver en la figura 2.

Figura 2

Logo de Android Studio



Nota: Obtenido de (Santaella, 2022)

De acuerdo con Esteban (2020) “El entorno de desarrollo de Android studio, se encuentran las herramientas que necesitamos para construir la aplicación. Esto incluye el diseño desde el código hasta la interfaz de usuario de la aplicación”. Es decir, con todo esto permite que las compilaciones se ejecuten rápidamente para que pueda verificar si hay errores y mejorar la aplicación final.

Debido a que realiza acciones en tiempo real constantes copias de seguridad o respaldos de las modificaciones que se realizan en el código, permite trabajar en proyectos reutilizando códigos y recursos de versiones pasadas de aplicaciones, por esta

y otras razones que se describen en la tabla 4 se posiciona de entre las mejores aplicaciones para el desarrollo de aplicaciones.

Tabla 4

Desarrollo en Android Studio

Ventajas	Desventajas
<p>Android Studio se ha convertido en el entorno de desarrollo recomendado para aplicaciones de Android, ya que es el entorno de desarrollo oficial de Google en asociación con JetBrains.</p>	<p>Aunque se lanzó la primera versión estable, v1.0, se encuentra en las primeras etapas y siempre es fácil introducir más cambios que pueden causar inestabilidad entre las diferentes versiones del proyecto.</p>
<p>Android Studio le permite crear nuevos módulos en el mismo proyecto sin cambiar su espacio de trabajo de gestión de proyectos.</p>	<p>Los nuevos desarrolladores de Android tienen una curva de aprendizaje lenta.</p>
<p>Con una simple descarga desde su página oficial se obtiene Android Studio, contando con todas las herramientas que se necesita para el desarrollo de aplicaciones.</p>	<p>El constructor de proyector en su sistema Gradle resulta complicado para quien inicia a desarrollar aplicaciones.</p>

Nota: Obtenido de (Android Studio, 2014)

Una de las características más poderosas de Android Studio es su compatibilidad con Flutter, el marco de desarrollo móvil de código abierto de Google. Flutter permite a los desarrolladores crear aplicaciones multiplataforma con rendimiento y diseño nativos en dispositivos iOS y Android. Con Flutter, los desarrolladores pueden crear hermosas interfaces de usuario de forma rápida y sencilla, al mismo tiempo que aprovechan funciones potentes como la aceleración de hardware, la recarga en caliente y más. Para cualquier desarrollador que busque crear aplicaciones en la plataforma Android, usar Flutter en Android Studio es esencial para crear aplicaciones de alta calidad que les encantarán a los usuarios.

Microsoft Visual Studio

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE) de Microsoft compatible con Windows y macOS, es capaz de admitir lenguajes como C, C#, Visual Basic .NET, PHP, F#, Python, Ruby, Java entre otros lenguajes de programación y otros entornos de desarrollo web.

De acuerdo con Ródenas (2012) “Visual Studio facilita a los desarrolladores crear aplicaciones web, sitios web y servicios web en cualquier entorno que admita la plataforma .NET. Esto hace posible la creación de aplicaciones que se comunican entre escritorios, páginas web y dispositivos móviles”.

La plataforma .NET es la base de Visual Studio, lo que permite a los desarrolladores crear e implementar rápidamente aplicaciones potentes con facilidad. Con funciones como herramientas de depuración, sistemas de control de fuente integrados y compatibilidad con varios idiomas, Visual Studio facilita el desarrollo de

aplicaciones web sólidas que se pueden implementar en varias plataformas. Además, Visual Studio proporciona una gran cantidad de recursos para que los desarrolladores los aprovechen al crear sus aplicaciones.

Aunque es posible programar utilizando solo el SDK de .NET Framework en Visual Studio, el SDK de .NET Framework consta de herramientas separadas lo que lo hace más engorroso. de trabajar, como entorno de desarrollo integrado, Visual Studio .NET facilita mucho la creación de programas.

Java Development kit

Java Development Kit (JDK) es un entorno de desarrollo de software utilizado para escribir y ejecutar aplicaciones Java. Contiene un conjunto de herramientas, bibliotecas y complementos que permiten a los desarrolladores crear aplicaciones sólidas en el lenguaje Java. El JDK también proporciona características de seguridad que ayudan a proteger contra código malicioso y malware, como lo menciona Martínez (2013):

La plataforma de desarrollo de Java conocida como Java Development Kit (JDK) se ha expandido y atraído a programadores de todo el mundo, de hecho, Java más que un lenguaje es una plataforma de desarrollo, un entorno de tiempo de ejecución y un conjunto de bibliotecas para desarrollar programas complejos.

Con la ayuda de JDK, los desarrolladores pueden crear aplicaciones de escritorio, móviles y basadas en web para Android y otras plataformas, además, permite a los desarrolladores utilizar la API de Ubidots para crear aplicaciones IoT con facilidad, con su amplia gama de funciones y complementos, JDK es uno de los kits de desarrollo

de software más populares utilizados por profesionales de todo el mundo debido a que también proporciona funciones de seguridad como cifrado y autenticación para aplicaciones.

En conclusión, Java Development Kit es un poderoso conjunto de herramientas que ayuda a los desarrolladores a crear fácilmente aplicaciones de software de alta calidad. Proporciona los componentes necesarios como API, compilador, depurador y complementos para facilitar el proceso de desarrollo.

Softwares para el desarrollo del prototipo autosostenible geolocalizado

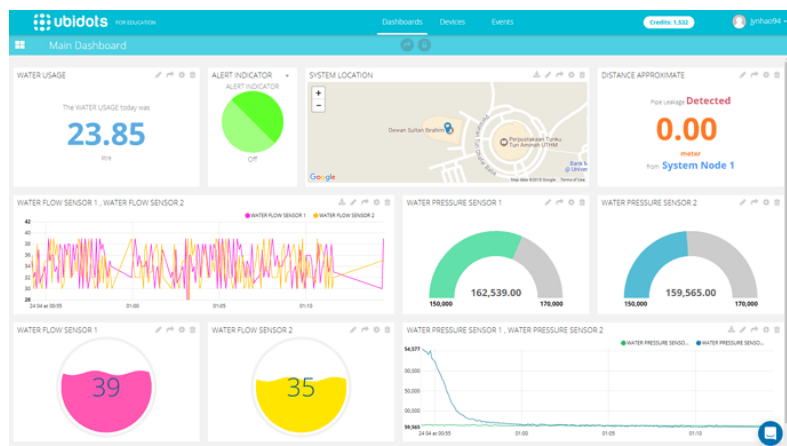
A fin de configurar el prototipo es necesario la implementación de software encargado de modelar la parte física en una red inalámbrica de sensores y que además tenga conexión a una nube que administre los datos obtenidos por medio del código de programación. Las herramientas que se proponen para la conectividad de los nodos son tres: Ubidots, XCTU y Arduino IDE.

Ubidots

Ubidots es una plataforma basada en la nube que ayuda a los usuarios a recopilar, almacenar, analizar y visualizar datos de diferentes tipos de sensores. Permite a los usuarios conectar múltiples nodos y sensores para monitorear y controlar su entorno desde paneles como muestra la figura 3. Con la ayuda de Ubidots, los usuarios pueden realizar un seguimiento de los cambios en su entorno en tiempo real y tomar decisiones basadas en los datos recopilados.

Figura 3

Tablero de paneles en Ubidots



Nota: Obtenido de (Ubidots Community, 2018)

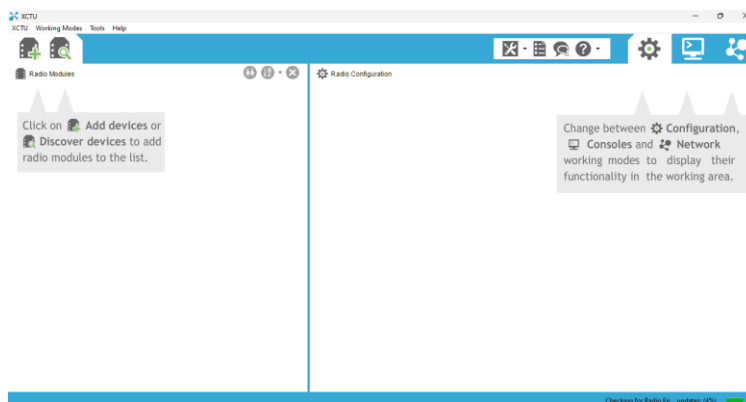
Es una solución ideal para aquellos usuarios que desean realizar un seguimiento de sus activos o necesitan monitorear procesos de forma remota. Ubidots también ofrece una amplia gama de funciones, como la programación de tareas, notificaciones, paneles, herramientas de análisis, entre otros, lo que la convierte en una excelente opción para el desarrollo de proyectos que buscan una plataforma potente y fácil de usar.

XCTU

XCTU es una potente aplicación multiplataforma que permite a los usuarios configurar, probar y depurar sus módulos Digi RF. De acuerdo con Rosendo et al. (2016) “El programa es de fácil comprensión y posibilita a los clientes de Digi probar sus módems de radio en el entorno actual usando solo un ordenador y los elementos que vienen con el módem de radio”. Es decir, permite a los usuarios modelar de forma rápida y sencilla una comunicación a bajas frecuencias, debido a la simplicidad de la interfaz que se ve en la figura 4 lo que les permite enviar y recibir datos de forma rápida y fiable.

Figura 4

Interfaz de XCTU



Con XCTU, los usuarios pueden configurar sus módulos para varios casos de uso, como detección inalámbrica, monitoreo, automatización y control. Además, proporciona una interfaz intuitiva con funciones avanzadas, como capacidades de secuencias de comandos que permiten a los usuarios automatizar tareas complejas. XCTU es una gran herramienta para los desarrolladores que necesitan configurar rápidamente sus módulos de RF para diversas aplicaciones.

Arduino IDE

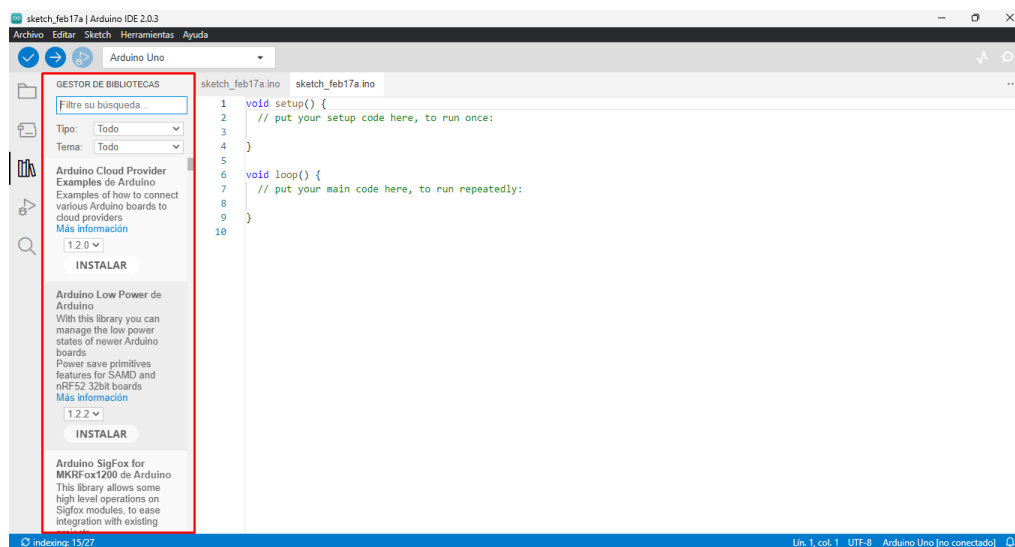
Arduino IDE es una plataforma de código abierto utilizada tanto por aficionados como por profesionales para crear proyectos electrónicos interactivos. De acuerdo con Peña (2020) “Se puede usar el software para escribir y cargar programas en placas Arduino, así como en placas compatibles”. Es decir, se puede usar para programar microcontroladores, como Arduino Uno, para controlar una variedad de componentes, desde LED y motores hasta sensores y más.

Con la ayuda de las librerías, los entusiastas de Arduino pueden encontrar recursos que los ayudarán a comprender los conceptos básicos de trabajar con el IDE y

enseñarles cómo crear sus propios proyectos. Los libros disponibles en las librerías que se encuentran organizadas dentro del layout como se puede ver en la figura 5 brindan información detallada sobre la programación para Arduino IDE, así como consejos útiles para solucionar problemas comunes. Además, pueden incluir tutoriales sobre proyectos específicos que los lectores pueden probar por sí mismos.

Figura 5

Gestor de bibliotecas de Arduino IDE



Red de sensores inalámbricos

Las redes inalámbricas de sensores son redes autónomas de nodos que se utilizan para monitorear condiciones físicas o ambientales como temperatura, presión, humedad y otras variables. Han ganado popularidad en los últimos años debido a su capacidad para proporcionar datos en tiempo real desde ubicaciones remotas. Esto los hace ideales para aplicaciones como vigilancia, seguimiento de activos y mantenimiento predictivo. Desde el punto de vista de Egas et al. (2019):

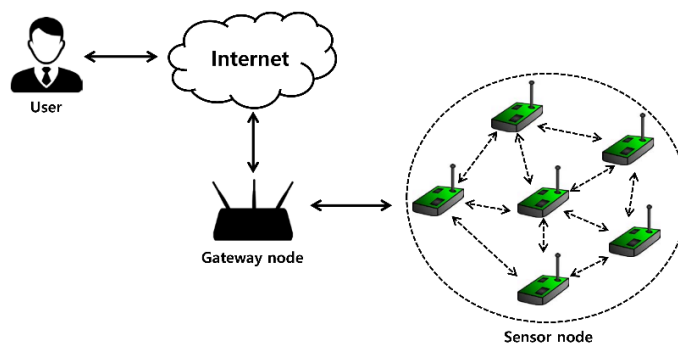
Una de las tecnologías utilizadas en el Internet de las Cosas es una red inalámbrica de sensores, la cual se caracteriza porque los nodos se construyen utilizando microcontroladores de baja potencia de cómputo, alimentados por baterías, de alta velocidad de transmisión y diseñados para funcionar en un largo tiempo.

El uso de estas redes aplicadas para el IoT es lo que ha permitido el desarrollo de sistemas más eficientes para recopilar y analizar datos de múltiples fuentes, además, de ser utilizar en una variedad de industrias, incluidas la atención médica, la agricultura, el transporte y la fabricación.

Las redes de sensores inalámbricos se basan en receptores de señales que intercambian datos entre sí para formar una red como se puede ver en la figura 6. El intercambio de receptor de salida de señal permite que los nodos de la red se comuniquen entre sí y compartan información sobre su entorno. Es así, que a partir de esta premisa los nodos pueden llegar a ser transmisores, receptores, escáneres y controladores.

Figura 6

Ejemplo de una red inalámbrica de sensores



Nota: Obtenido de (Yu & Park, 2020)

Tarjetas de Desarrollo

Arduino UNO

Arduino UNO es una popular placa de microcontrolador de código abierto que permite a los usuarios crear proyectos electrónicos interactivos. En la figura 7 se puede observar un procesador Atmel AVR, lo que lo hace compatible con muchas librerías y escudos.

Figura 7

Arduino UNO



Nota: Obtenido de (Grupo Velasco, 2018)

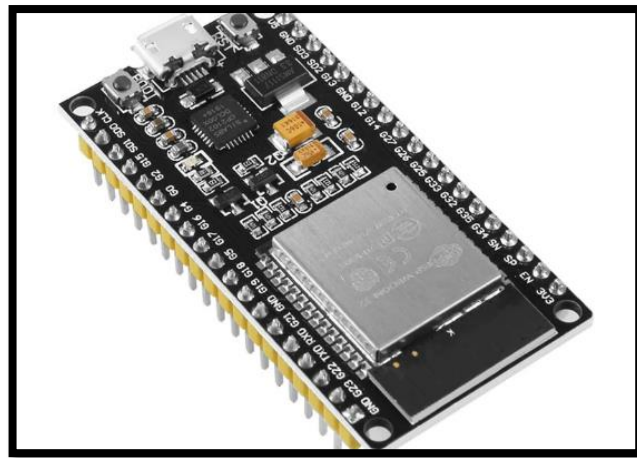
Una de las bibliotecas más populares utilizadas con Arduino UNO es TinyGPS, que permite a los usuarios rastrear su ubicación mediante tecnología GPS. Además de esta biblioteca, Arduino UNO también tiene compatibilidad con otras bibliotecas como ESP8266 Wifi, ESP32 e I2C LCD. Con estas bibliotecas, los usuarios pueden crear proyectos más complejos que pueden conectarse a Internet o mostrar datos conectados a una aplicación.

NodeMCU

La placa NodeMCU que se aprecia en la figura 8, usa LUA como lenguaje de programación para aplicaciones que requieren una conexión Wifi en el menor tiempo posible, y la placa tiene las capacidades de procesamiento y memoria para acomodar sensores y aplicaciones con tiempos de ejecución cortos. NodeMCU es compatible con Arduino para realizar aplicaciones IoT (Internet de las cosas)

Figura 8

ESP32



Nota: Obtenido de (Grupo ElectroStore, s.f.)

NodeMCU es una plataforma IoT de código abierto basada en los microcontroladores ESP8266 y ESP32, comúnmente se utiliza para conectar dispositivos a Internet y crear aplicaciones de Internet de las cosas (IoT). NodeMCU es una excelente plataforma para crear aplicaciones IoT de bajo costo y bajo consumo.

Tiene una amplia gama de funciones, como conectividad Wi-Fi, soporte para múltiples protocolos e integración con plataformas en la nube como Ubidots. NodeMCU permite a los desarrolladores desarrollar rápidamente aplicaciones conectadas con un mínimo esfuerzo. Con su entorno de desarrollo fácil de usar y su amplia gama de

funciones, NodeMCU es la opción perfecta para cualquiera que busque crear una aplicación IoT.

Raspberry Pi

Raspberry Pi es una computadora de placa única de bajo costo diseñada para fines educativos y aficionados. Es una plataforma de código abierto que tiene la capacidad de ejecutar múltiples lenguajes de programación como Python, C++, Java y más.

Su pequeño tamaño como se lo puede evidenciar en la figura 9 lo hace perfecto para su uso en proyectos de automatización del hogar, proyectos de robótica y otras aplicaciones donde el espacio es primordial. El Raspberry Pi también cuenta con un potente procesador Broadcom, RAM, GPU, USB, HDMI, Ethernet y un conector de cámara, con lo que puede manejar la mayoría de las tareas informáticas con facilidad.

Figura 9

Raspberry Pi



Nota: Obtenido de (Adafruit, 2018)

Comparativa de las tarjetas de desarrollo

En el mundo de las tarjetas de desarrollo, hay una variedad de opciones disponibles. Desde Arduino Uno hasta ESP32 y Raspberry Pi, cada tarjeta tiene sus propias características y capacidades únicas. En la Tabla 5 se compara las tres tarjetas de desarrollo en donde se analiza las especificaciones de hardware, pines disponibles, consumo y otras características de cada tarjeta para decidir cuál es la mejor opción para el prototipo.

Tabla 5

Cuadro comparativo de tarjetas de desarrollo

Características	Arduino UNO	Raspberry pi	NodeMCU
Procesador	ATMega328	Arm11	ESP32
Velocidad reloj	16 MHz	700 MHz	80-160 MHz
Voltaje	5 V	5 V	3.3 V
Voltaje in	7-12 V	5 V	5 V
Voltaje máximo	6-20 V	6 V	5 V
RAM	2 kb	512 Mb	50 kb
USB	No dispone	2	No dispone
Audio	No dispone	HDMI, Analógico	No dispone
Video	No dispone	HDMI, Analógico	No dispone
Ethernet	No dispone	1 gigabit Ethernet	No dispone

Pines in/out digitales	14	28 GPIO	9 GPIO
Pines in analógicos	6	No dispone	1
Alto	68.6 mm	57 mm	51 mm
Ancho	53.4 mm	86 mm	23 mm
Peso	25gr	45 gr	6.8 gr

Nota: Obtenido de (Gascón, 2019; González, 2015)

Como se muestra en la Tabla 5, NodeMCU supera a el resto de las tarjetas en casi todos los aspectos, la única desventaja de la placa mencionada, en comparación con las otras placas es el número limitado de pines tanto en analógico como en digital. Es por aquello que para el desarrollo de este prototipo se planea usar el NodeMCU y Arduino UNO.

Módulos complementarios

Módulos Xbee

Los módulos Xbee son una opción popular para la comunicación inalámbrica, ya que, proporcionan una conexión fiable y segura entre dos o más dispositivos. Los módulos Xbee vienen en tres tipos principales: S2C, S2B y S2. Cada tipo tiene sus propias ventajas que lo hacen ideal para diferentes aplicaciones.

Con base a lo expresado por Faludi (2021) “Muchas líneas Digi XBee están disponibles en versiones estándar y PRO. La principal distinción entre los dos es la potencia de transmisión y, como resultado, el rango del XBee”. Es decir, los módulos regulares de Xbee se utilizan para la comunicación básica, mientras que la versión pro tiene características adicionales como encriptación y velocidades de datos más altas.

En la figura 10 se puede apreciar el módulo S2C, este se encuentra diseñado para el tipo comunicación punto a punto y es perfecto para conectar dos dispositivos entre sí, además, proporciona comunicación de largo alcance a grandes distancias con un consumo mínimo de energía.

Figura 10

Módulo Xbee S2C



Nota: Obtenido de (AV Electronics, s.f.)

El módulo S2B está diseñado para proporcionar transmisión de datos de alta velocidad en distancias más cortas, es decir, para redes de malla en donde se desea conectar varios dispositivos entre sí, este se puede evidenciar en la figura 11.

Figura 11

Módulo Xbee S2B



Nota: Obtenido de (SOLID STRTE SOPPLIES, s.f.)

En la figura 12 se puede evidenciar el módulo S2, siendo este la última versión de la serie Xbee y ha mejorado el rendimiento con respecto a sus predecesores,

ofreciendo una combinación de capacidades de transmisión de largo alcance y alta velocidad.

Figura 12

Módulo Xbee S2



Nota: Obtenido de (DIGI, s.f.)

XBee Explorer

XBee Explorer es un dispositivo utilizado para configurar módulos XBee. Se utiliza para conectar y emparejar módulos XBee S2C con otros dispositivos. El terminal se utiliza en combinación con el software XCTU para configurar los ajustes del módulo XBee. XBee Explorer también permite dar solución fácilmente a cualquier problema de conexión que pueda tener algún módulo XBee dentro de la red inalámbrica.

Módulo GPS

El módulo GPS es un componente esencial para cualquier dispositivo que necesite acceder a datos de ubicación, es especialmente importante para las aplicaciones de navegación y seguimiento, debido a que es capaz de proporcionar datos en tiempo real. Hay muchos tipos diferentes de módulos GPS disponibles en la actualidad, como NEO-6M, GT-U7 y XA1110, cada uno de estos módulos tiene características y capacidades únicas que los hacen adecuados para diferentes casos de uso.

Los módulos GPS NEO-6M y GT-U7 son dos de las opciones más populares para los usuarios que desean integrar la funcionalidad GPS en sus proyectos, ambos son compatibles con Arduino y otros microcontroladores, lo que proporciona información de posicionamiento precisa. El modelo XA1110 es una opción ideal para quienes buscan una solución de módulo GPS fiable y rentable. Con su alta precisión y bajo consumo de energía, es perfecto para una variedad de aplicaciones como servicios de navegación, seguimiento o ubicación.

Panel solar

El panel solar es una forma eficiente y rentable de generar electricidad a partir del sol. Se puede utilizar para una amplia variedad de aplicaciones, como alimentar hogares, negocios e incluso vehículos. Para aprovechar al máximo los paneles solares, se debe desarrollar sistemas complejos como el presente prototipo compuesto por Arduino y XBee los cuales conforman una red de sensores inalámbricos, adicionalmente es posible ayudar a optimizar su eficiencia y prolongar su vida útil mediante el uso de baterías de litio para el almacenamiento de energía.

Batería de Litio

Las baterías de litio son una de las mejores opciones para almacenar energía, ya que, tienen la capacidad de almacenar una gran cantidad de energía a través de sus celdas con un voltaje nominal de 3.7 V y corriente de carga de 2500 mA. Esto permite que sean usadas en diferentes aplicaciones con la respectiva configuración de serie o paralelo para aumentar su capacidad y duración. Además, estas baterías son muy versátiles y se pueden encontrar en diferentes tamaños para adaptarse a cualquier necesidad.

Figura 13

Baterías de Litio



Nota: Obtenido de (TETTSA, s.f.)

Capítulo III MARCO METODOLÓGICO

Para realizar el levantamiento de la información, se procede a definir el proceso, diagramas de referencia, las placas de desarrollo y módulos complementarios fundamentales para la construcción del prototipo, ya que, basándose en el estudio de Pérez et al. (2015) “Es la ciencia que suministra al investigador una colección de conceptos, principios y leyes que le permite regir el proceso de investigación científica de manera efectiva y tienden a ser excelentes”. Es por esa razón que se considera acertados los temas a desarrollar.

Tipo de investigación

Investigación exploratoria

De acuerdo con (Ortega, 2017) “La investigación exploratoria tiene como finalidad la recolección de datos en diferentes fuentes como libros, sitios web, personas o experiencias”. Este proyecto pretende dar una nueva alternativa que solucione la pérdida de estos animales a través de geolocalización, por lo tanto, se utilizarán manuales de uso y datasheet que permita describir de forma detallada la funcionalidad de los equipos utilizados.

Investigación aplicada

Para el desarrollo de este prototipo se usará la técnica de prueba y error de los equipos implementados en la propuesta tecnológica, esto permitirá realizar ensayos y corregir los problemas de la red de sensores. Teniendo en cuenta la opinión de Vargas (2009):

Las prácticas, entendidas como investigación aplicada, son experimentos de investigación destinados a resolver o mejorar una situación particular o específica, para verificar un procedimiento o modelo mediante la aplicación de la innovación y una proposición de intervención, en este caso de carácter educativo, en un grupo, individuo, organización o empresa lo solicite.

Esta investigación permitirá llevar la teoría a la parte práctica, en el caso del desarrollo y diseño de un prototipo autosostenible geolocalizado implementando una red WSN para el rastreo de caprinos se tomará en cuenta las diferentes fases.

Población y muestra

Se entiende como población al foco primordial de sujetos a quienes va dirigida la investigación con base en Ardila et al. (2004) “Una población es una colección de personas o animales que viven en una ubicación geográfica particular.”. Mientras que, la muestra permite al investigador estudiar cierta parte de la población para que los resultados se puedan utilizar para hacer conclusiones sobre toda la población.

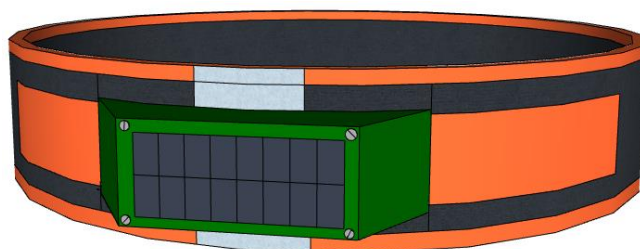
En la investigación se toma como población a un grupo de 20 caprinos con un tamaño promedio a 63.1cm y 65.8cm con un peso promedio de 120 kg. La muestra tomada abarca el 15% de la población que será a quienes se le implementará el prototipo autosostenible geolocalizado implementando una red WSN. La técnica de muestreo está dada por conveniencia debido a la facilidad de acceso a la población de caprinos.

Simulación Desarrollada en Sketchup

Para cumplir la propuesta de la fase 2, mediante la plataforma de simulación Sketchup se realizó la elaboración de un boceto en 3D del diseño tentativo para el collar como se puede apreciar en la figura 14.

Figura 14

Diseño tentativo en 3D de nodo supervisor



De igual forma, se elabora una simulación que represente el diseño final del nodo supervisor, en donde se visualiza la colocación respectiva del prototipo en la nuca del caprino como se ve en la figura 15.

Figura 15

Ubicación del nodo supervisor en el caprino



Mediante las imágenes mostradas anteriormente se puede evidenciar el cumplimiento de la fase 2 propuesta para la construcción del prototipo.

Diseño de implementación de prototipo

El diseño de los prototipos comprende tres fases fundamentales, entre estas la primera fase comprende del análisis de estudio, en la segunda fase se realizan las simulaciones pertinentes y finalmente se entra en la etapa de verificación del funcionamiento del prototipo.

FASE 1: Estudio del comportamiento de una especie gregaria (caprinos)

Se analizaron y se estudiaron los comportamientos de los caprinos con el fin de ampliar el conocimiento sobre su conducta alimenticia, hábitos de consumo y tiempo de pastoreo, tiempos de descanso; en base a los datos analizados se simulará el prototipo que se adapte a las necesidades de esta especie gregaria, es así como en la tabla 6 mostrada a continuación se describen los comportamientos propios de esta especie:

Tabla 6

Comportamiento de los caprinos

Generalidades de los caprinos	
Conducta alimenticia	Los caprinos ramonean echados para consumir sus alimentos, sin embargo, depende de la seguridad del ambiente este mantiene la posición, de lo contrario ramonean parados y con estado de alerta.

Hábitos sexuales	Los caprinos resultan muy agresivos en época reproductiva, dado que usan sus cuernos para golpear, su cabeza es la que sufre el daño del impacto al dar un golpe.
<hr/>	
Tiempo de pastoreo	Los caprinos cuentan con un alto grado de supervivencia e influye en sus hábitos al momento de rumiar debido a que la postura con la que este se alimente dependerá de si el sitio en el que se encuentren lo consideran seguro.
<hr/>	
Tiempo de descanso	Es común en especies caprinas reposar echados, sin embargo, el tiempo de descanso varia en torno a la estación climática y el lugar que seleccione para descansar.

FASE 2: Simulación de la ubicación de los materiales utilizados para el desarrollo del prototipo

El software de simulación Sketchup permite tener una idea clara acerca de la ubicación de los accesorios implementados en la caprino, considerando esto para la fase de implementación del prototipo.

FASE 3: Verificación de la red de sensores inalámbricos

Terminada la fase de investigación e implementación, se procede a la verificación efectiva entre los nodos prototipos de geolocalización que se encuentran inmersos en la red WSN, se analiza la exactitud y se evidencia el cumplimiento de los objetivos.

La comprensión de estas fases se puede interpretar de una mejor manera describiéndola de manera gráfica, se seleccionó el diagrama de embudo como se puede

apreciar en la figura 16 para representar el proceso que se debe llevar para lograr el propósito de la propuesta tecnológica.

Figura 16

Diagrama de proceso para el diseño del prototipo

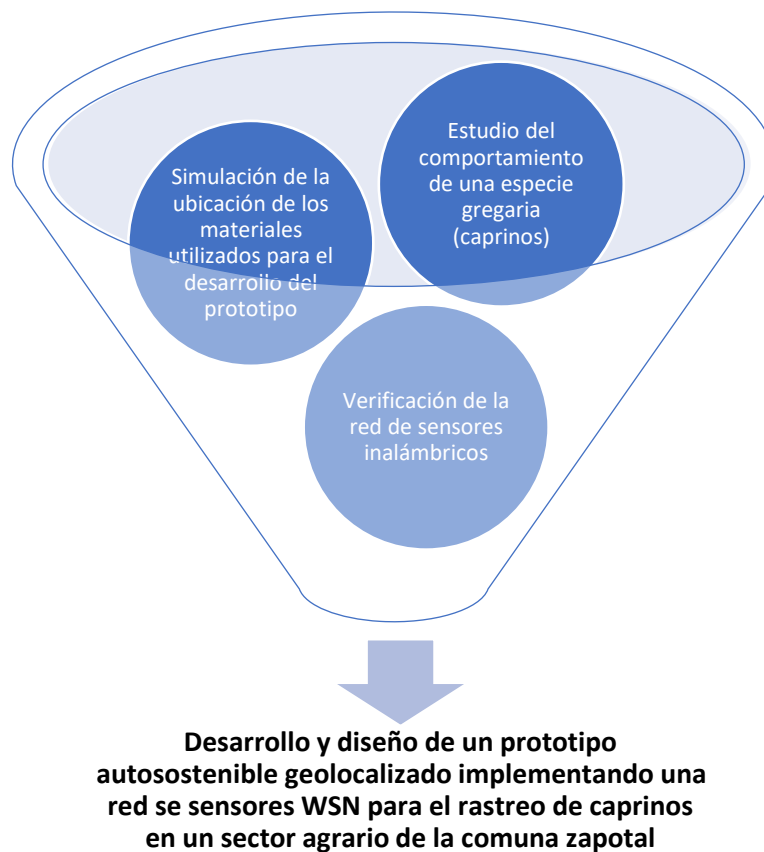
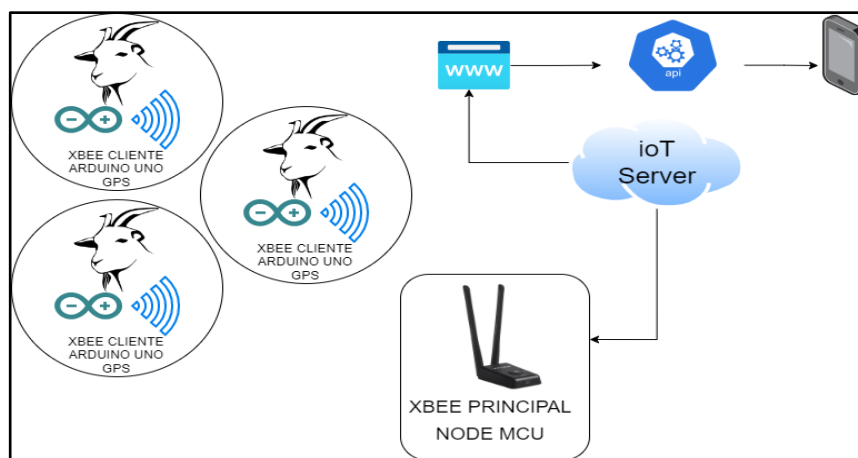


Diagrama de concepción general

La concepción y los requerimientos para el prototipo a utilizarse se detallan en la figura 17, donde, se observa está conformado por dos bloques, el nodo principal y el nodo supervisor y cada uno de estos puede ser desglosado en diagramas de bloque para mejorar su comprensión y análisis.

Figura 17

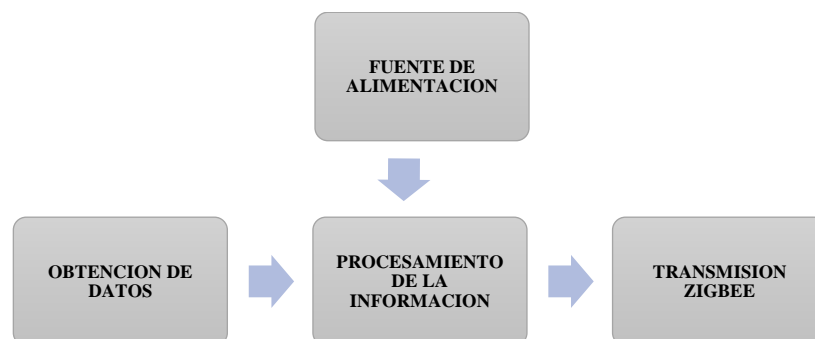
Diagrama de la topología general del prototipo



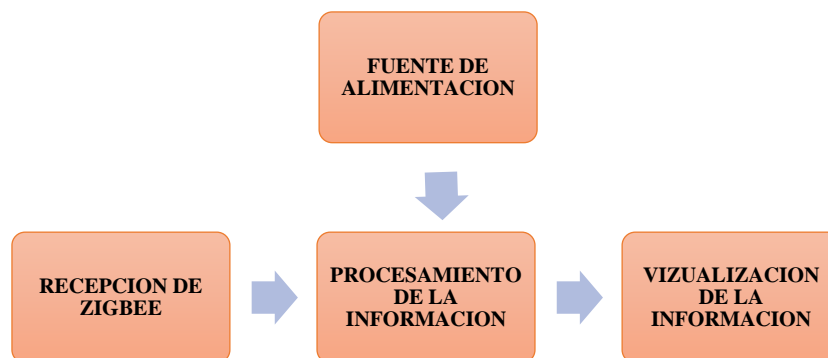
Este diagrama proporciona una representación visual de la topología general del prototipo desarrollado para un sistema IoT basado en Zigbee, describiendo la conexión entre NodeMCU y Ubidots, así como también su conexión a la aplicación móvil. La red de malla también se muestra en este diagrama para ilustrar cómo se transfieren los datos de un dispositivo a otro, además, esta topología sirve como una referencia para comprender el funcionamiento de este prototipo.

Diagrama de Bloque perteneciente al nodo Supervisor

El diagrama sigue el proceso de 4 bloques que se muestra en la figura 18, el primer bloque corresponde a la lectura del GPS desde el puerto serial para adquirir las coordenadas de latitud y longitud, el segundo bloque procesa los datos utilizando la salida digital del Arduino UNO, el tercero representa los paneles solares y la batería recargable usados como fuente de alimentación del prototipo, mientras que, el cuarto es la unidad de transmisión Zigbee, que envía los datos obtenidos del nodo supervisor al nodo maestro.

Figura 18*Diagrama del nodo supervisor***Diagrama de Bloque perteneciente al nodo Principal**

De igual forma que el nodo supervisor, el diagrama del nodo maestro de la figura 19 también consta de 4 bloques, siendo el primero Zigbee Receiver, el cual receipta las tramas enviadas por el nodo supervisor, el segundo corresponde al procesamiento de los datos enviados por el NodeMCU ESP32 hacia la unidad de visualización, el tercero hace referencia a la fuente de energía que suministran electricidad a toda la unidad del nodo coordinador, mientras que el cuarto permite mostrar fecha, hora, latitud y longitud del caprino usando Ubidots.

Figura 19*Diagrama de Bloque del nodo Principal*

Esquema de conexión de los nodos

Teniendo en cuenta las especificaciones y tablas comparativas recopiladas en el marco teórico, se tomó la decisión de trabajar con los componentes que cumplan con los requisitos que se necesita para el desarrollo del prototipo y la función de ser autosostenible. A continuación, se procede a detallar cada componente a utilizarlo, mencionando el rol específico que cumplirá en el desarrollo de la tesis.

Especificaciones de Arduino UNO

En la figura 20 se observa el Arduino UNO, esta es una placa basada en el microcontrolador ATmega328P, consta de 14 pines digitales distribuidos para entrada y salida, tal que, 6 se configuran como salidas PWM y otros 6 funcionan como entradas analógicas, además, incluye un cristal oscilador de 16 MHz, conexión a USB, un puerto de alimentación y un pulsador para reseteo. Algunas características adicionales a esta versión de Arduino son las mencionadas en la tabla 7.

Figura 20

Arduino UNO



Tabla 7*Especificaciones de Arduino UNO*

Especificaciones	Descripción
Voltaje de alimentación	5 VDC
Voltaje de entrada	7.5 – 12 VDC
Puerto serial	1
Memoria flash	32 KB
Memoria RAM	2 KB
Memoria EEPROM	1 KB

La placa de desarrollo Arduino UNO cumple el rol de ser el cerebro dentro del nodo supervisor, siendo el encargado de portar el código de desarrollo en el lenguaje de programación, lo cual le permite emitir notificaciones al nodo coordinador indicando de forma constante la ubicación del prototipo.

Especificaciones de NodeMCU

El ESP32 de la figura 21 es la evolución del ESP8266 con mayor capacidad de comunicación y computación, en la conectividad permite utilizar varios protocolos inalámbricos como: Wifi, Bluetooth, por otro lado, el procesamiento, su CPU de 32 bits cuenta con hasta 240 MHz que son aportados por dos núcleos, los cuales se pueden controlar de forma independiente.

Figura 21*Esp32*

ESP32 permite desarrollar aplicaciones utilizando diversos lenguajes de programación y bibliotecas, al trabajar en el IDE Arduino, además de usar un lenguaje de programación familiar y un IDE fácil de usar, Algunas características adicionales a esta versión de ESP32 son las mencionadas en la tabla 8.

Tabla 8*Especificaciones ESP32*

Especificaciones	Descripción
Voltaje de alimentación	5 VDC
Voltaje de entrada	3.3 VDC
Pines de salida	30
Puerto serial	1
Memoria flash	520 KB

Memoria ROM	248 KB
Wifi:	802.11 b/g/n/e/i (2.4 GHz hasta 150 Mbit/s)

Especificaciones de Shield de Xbee para Arduino

En la figura 22 se observa el XBee Shield, el cual es un adaptador popular para placas Arduino que permite a los usuarios conectar dichas placas a módulos inalámbricos XBee, permitiendo la comunicación de forma inalámbrica con otros dispositivos Xbee, creando redes inalámbricas con la posibilidad de controlarlos de forma remota. El Shield también proporciona el regulador de voltaje, la fuente de alimentación y el botón de reinicio necesarios para el módulo XBee. Las especificaciones de esta placa son las detalladas en la tabla 9.

Figura 22

Xbee Shield

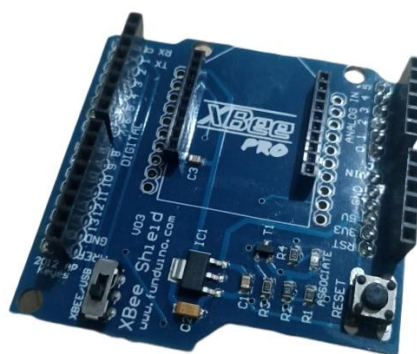


Tabla 9

Especificaciones de Xbee Shield

Especificaciones	Descripción
-------------------------	--------------------

Voltaje de salida dual	3.3 – 5 V
IO compatible	3.3 – 5 V
Versión de USB	2.0
Bitbang	Habilitado
Software compatible	XCTU

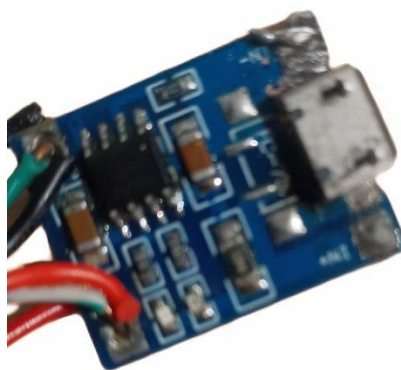
Nota: Obtenido de (Techmake, 2023)

Especificaciones de módulo de carga tp4056

El módulo de carga tp4056 de la figura 23 es una solución rentable para proteger contra la sobrecarga, la descarga excesiva, fuga térmica y los cortocircuitos de la batería, convirtiéndola en una opción ideal para alimentar dispositivos electrónicos, por lo tanto, es imprescindible para cualquiera que trabaje con baterías de litio.

Figura 23

Módulo de carga



Dicho módulo al detectar cuando hay una sobrecarga o un cortocircuito en el sistema del nodo supervisor este desconectará la alimentación de la placa Arduino para

proteger los componentes de daños. Algunas de sus características son las descritas en la tabla 10.

Tabla 10

Especificaciones del módulo de carga

Especificaciones	Descripción
Voltaje de entrada	5 V
Intensidad máxima	1000 mA
Corte de carga	4.1 – 4.3 V
Puerto de entrada	Micro USB

Nota: Obtenido de (SDR, 2022)

Especificaciones de Batería de Litio 18650

La batería de litio 18650 de la figura 24 es una opción popular para alimentar dispositivos pequeños como placas Arduino y otros microcontroladores. Ha sido diseñada para proporcionar altas tasas de carga y descarga y baja tasa de autodescarga, además este modelo se convierte en una opción para los sistemas de almacenamiento de energía solar debido a su capacidad para almacenar grandes cantidades de energía y su vida útil relativamente larga.

Figura 24

Baterías 18650



También es ligero y compacto, por lo que es fácil de transportar, y cuando se usa junto con paneles solares, la batería de litio 18650 puede proporcionar energía duradera durante largos períodos de tiempo. Con su alta densidad de energía, la batería de litio 18650 se puede utilizar en una variedad de aplicaciones que van desde la electrónica de consumo hasta los vehículos eléctricos. Algunas características adicionales a esta batería de litio son las mencionadas en la tabla 11.

Tabla 11

Especificaciones de Batería de Litio 18650

Especificaciones	Descripción
Tipo de Batería	Li-ion
Voltaje nominal	3.7V DC
Voltaje de carga completa	4.2V DC
Voltaje de descarga de corte	2.75V DC
Potencia	8.14Wh

Corriente	2100 - 2300 mAh
Peso	43g

Especificaciones del módulo Xbee S2C

Xbee S2C que se observa en la figura 25 es un módulo transceptor de radio con características como bajo consumo de energía, largo alcance de transmisión y alta velocidad de datos, diseñado para proporcionar una comunicación inalámbrica confiable, esto convierte a la placa en la mejor opción para la implementación del prototipo, ya que, proporciona características de control y detección remota al crear la red inalámbrica de sensores, otras características que aportan a la implementación del prototipo son las detalladas en la tabla 12.

Figura 25

Módulo Xbee S2C

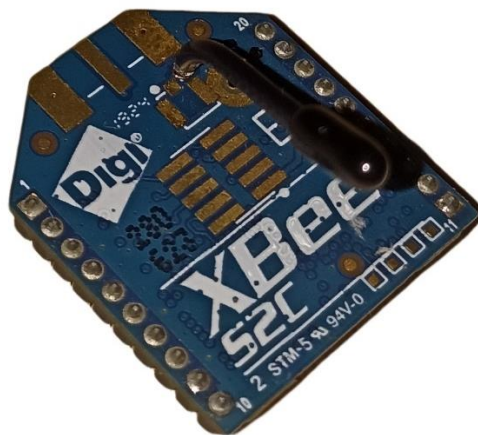


Tabla 12

Especificaciones de Xbee S2C

Especificaciones	Descripción
------------------	-------------

Frecuencia ISM	2.4 GHz
Velocidad de transferencia	250 kb/s
Radio de señal	1.2 km
Sensibilidad	-100 a -102 dBm
Voltaje de entrada	2.1 – 3.6 v
Modulación	DSSS
Memoria Flash	32 KB
Memoria RAM	2kB
Temperatura de operación	-40°C – 85°C

Especificaciones GPS Ublox NEO-6M

El módulo Ublox NEO 6M de la figura 26 es un potencial módulo GPS que se puede utilizar en todo tipo de proyectos, debido a la variedad de características, que incluyen alta precisión y bajo consumo de energía, por lo que se convierte en una opción ideal para aplicaciones de seguimiento y navegación.

Figura 26*GPS Ublox NEO-6M*

El módulo GPS compatible con Arduino necesita un voltaje de entrada de 3 a 5 volts, en tanto, los voltajes de salida y entrada serán de 3.3 volts, es decir, que para la placa de desarrollo transmita tramas de señal hacia el GPS, será necesario un convertidor para regular los niveles de voltaje. Para el caso del nodo supervisor al solo requerir que se reciba los datos solo se necesita conectar los pines de tx y rx. Otras características necesarias para el prototipo son las descritas en la tabla 13.

Tabla 13*Especificaciones de módulo GPS*

Especificaciones	Descripción
Interfaz	Serial UART 5V
Sistema geoposicionamiento	WGS-84
Sensibilidad recepción	-148 dBm
Sensibilidad rastreo	-161 dBm

Altura máxima	18000
Velocidad máxima	515 m/s
Exactitud	1microsegundo
Baudrate	9600
Tamaño de la antena	23mm x 23mm
Tamaño del módulo	23mm x 32mm
Antena	Cerámica activa

Nota: Obtenido de (Electronilab, s.f.)

Especificaciones del panel solar

Las celdas solares como se aprecia en la figura 27 son una excelente opción para cualquier proyecto autosostenible, se pueden usar para generar electricidad a partir de la energía del sol. La celda solar de 6 voltios y 200 mA con un tamaño de 110 x 60 mm es una opción ideal para este tipo de proyectos, ya que tiene suficiente potencia de salida para la mayoría de las aplicaciones. Este tipo de celda solar también es relativamente fácil de instalar y mantener, lo que la convierte en una excelente opción para desarrollar este proyecto autosostenible.

Figura 27*Panel solar*

Las celdas solares son una parte importante de los proyectos para encender dispositivos Arduino y Xbee, proporcionando una fuente de energía confiable de hasta 6 voltios de electricidad, suficiente para alimentar pequeños dispositivos electrónicos eliminando la necesidad de costosas facturas de electricidad u otras fuentes de energía. Otras características necesarias del panel solar son las descritas en la tabla 14.

Tabla 14*Especificaciones del panel solar*

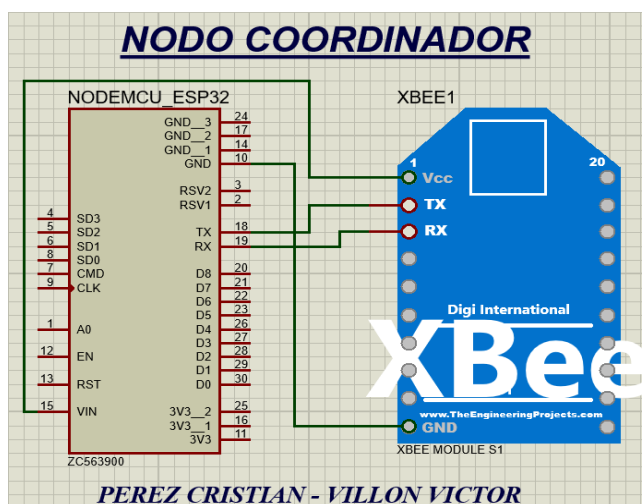
Especificaciones	Descripción
Voltaje	6 V
Corriente nominal	200 mA
Potencia entregada	1.2 W
Dimensiones	110*60mm
Placa trasera	Metálica
Cubierta	Fibra de epoxy

Esquema de conexión de nodo Principal

Mediante un software de simulación se realizó el esquemático para las conexiones del nodo coordinador de la figura 28, los componentes necesarios para la correcta ejecución son las respectivas librerías de la tarjeta de desarrollo esp32 y del módulo Xbee S2C.

Figura 28

Nodo coordinador



La lógica de conexión del nodo coordinador debe ser capaz de tener comunicación con todos los nodos configurados en modo supervisor, además de transferir los datos de posicionamiento al servidor Ubidots en donde se podrá visualizar las coordenadas obtenidos para posteriormente realizar un análisis de los datos obtenidos.

Esquema de conexión del nodo Supervisor

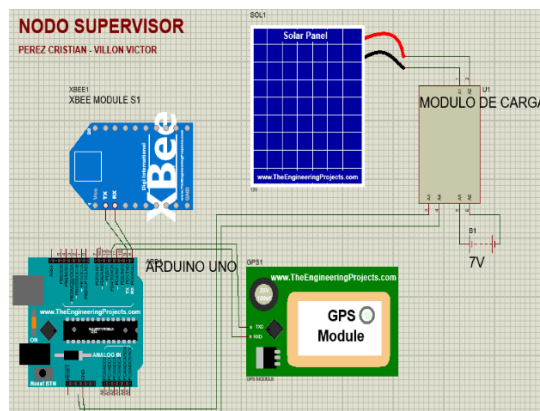
Mediante el mismo software de simulación se realiza el esquemático de conexión del nodo supervisor de la figura 29, los componentes que lo conforman son: Arduino UNO, Shield Xbee, Xbee S2C, módulo GPS, módulo de carga, panel solar y baterías de litio. Se conectan de tal forma que el prototipo sea autosostenible y pueda tanto receptor como enviar la trama de coordenadas hacia el nodo coordinador.

La placa de desarrollo Arduino UNO se conecta al módulo Xbee y al módulo GPS a través de los terminales tx y rx respectivamente. Los 5 volts y tierra que suministran los pines analógicos de la tarjeta Arduino se utilizan para suministrar energía tanto al Xbee como al módulo GPS.

Por otro lado, el panel solar se conecta al módulo de carga de las baterías de litio a través de los terminales positivo y negativo respectivamente. De esta forma el nodo puede ser autosostenible, ya que, el módulo de carga al mantener cargadas las baterías por medio del panel solar abastece al Arduino UNO con energía mediante los pines de 5 volts y tierra.

Figura 29

Nodo supervisor



Configuración de los módulos Xbee S2C

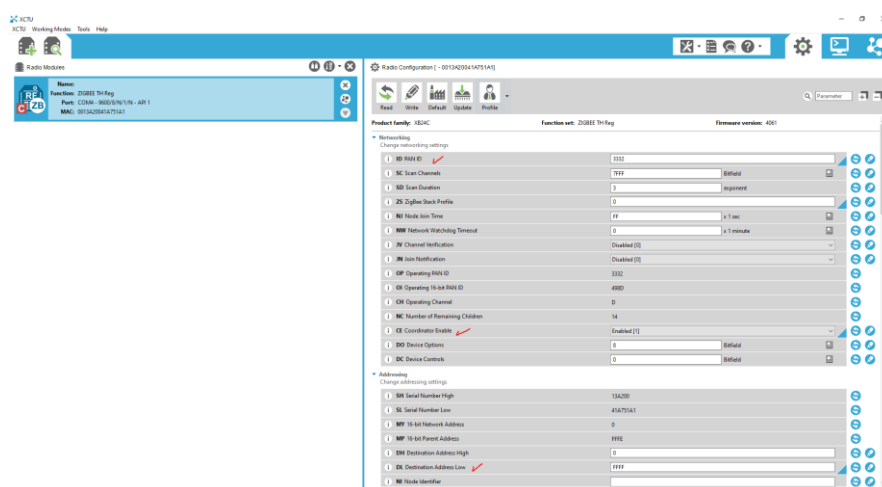
Para la respectiva configuración de modos en cada módulo Xbee es necesario el complemento Xbee Explorer para el uso del software XCTU debido a que esta plataforma es esencial y nos proporciona herramientas para un inicio rápido de los módulos de comunicación, ya que, permite la visualización por medio de gráficos niveles de conexión, intensidad de señal y gestión de Apis.

Configuración del Nodo Coordinador

En la figura 30 se observan los aspectos que se debe configurar para que el módulo Xbee se comporte como coordinador, siendo el primero la opción ID, el cual, proporciona un identificador para la autenticación de la red, de igual forma, se habilita la opción coordinador cambiando su valor lógico de 0 a 1, también, se asigna a la opción de dirección de destino alta el valor de 0 y destino baja FFFF, este proceso se realiza para el envío de mensajes broadcast a los nodos que conformen la misma red. La última configuración consiste en habilitar la opción Api, el cual define el modo de transmisión. El resto de las opciones son el inicio rápido por defecto del software.

Figura 30

Opciones de configuración del Xbee coordinador

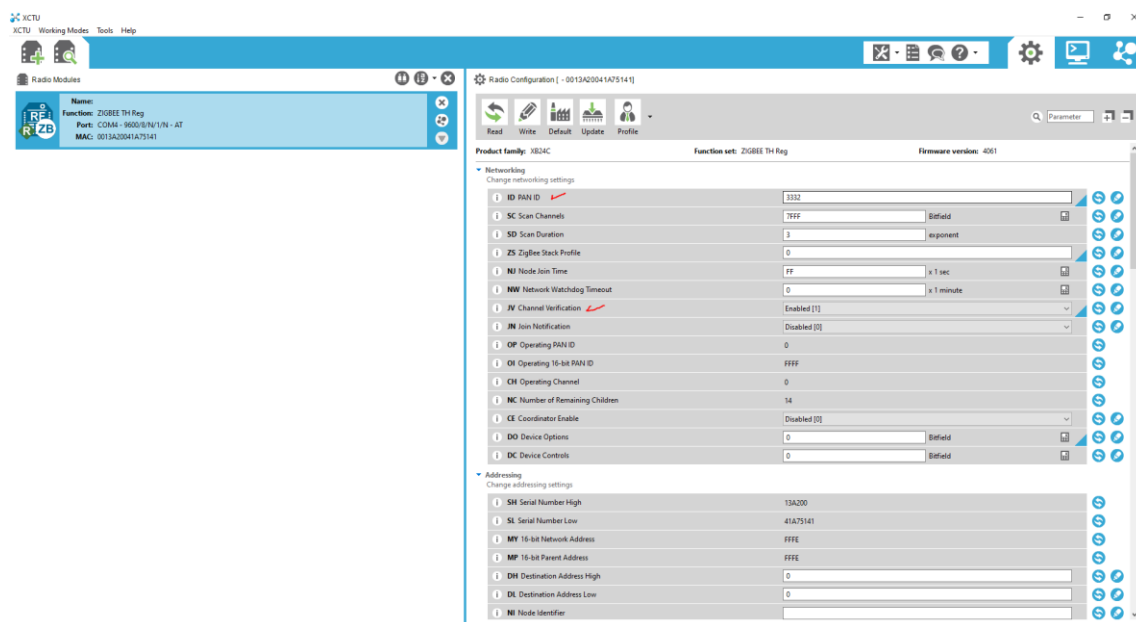


Configuración del Nodo Supervisor

Para la configuración del nodo coordinador se usó el software XCTU, el cual está diseñado para configurar, probar y depurar dispositivos XBee S2C con su interfaz de usuario intuitiva y funciones integrales como se puede apreciar a la figura 31, se procedió a configurar fácilmente la ID de PAN asignándole el valor de 3332, posterior se modifica la verificación del canal y se le asigna el valor en 1 lo que le permite afirmar que tanto el dispositivo coordinador y el supervisor estén usando el mismo canal.

Figura 31

Opciones de configuración del Xbee supervisor



Fundamentos del código Arduino para los nodos

Arduino es una plataforma de código abierto para construir proyectos interactivos, se ejecuta con el lenguaje de programación Arduino, este lenguaje permite

a los usuarios crear programas con instrucciones que pueden controlar los componentes de hardware e interactuar entre sí.

Desarrollo del código para el nodo coordinador

En la figura 32 se puede observar el código que se asigna al nodo coordinador debe cumplir con dos principios fundamentales, el primero consiste en la recepción de coordenadas que transmitan todos los nodos supervisores por medio del protocolo de comunicación inalámbrica Zigbee, para la segunda condición se debe cumplir el envío de datos desde la placa de desarrollo al servidor en la nube por medio del protocolo de red MQTT, para poder visualizar los datos desde la interfaz gráfica de Ubidots en tiempo real. Se puede visualizar el código completo en el anexo E.

Figura 32

Código del nodo coordinador

```

ESP32_version2.ino
1  /*****
2  * Include Libraries
3  *****/
4  #include "UbidotsEsp32Mqtt.h"
5
6  /*****
7  * Define Constants
8  *****/
9  const char *UBIDOTS_TOKEN = "BBFF-XSgWqLmtRwDECuXEMLfpjzTtUAeT"; // Put here your Ubidots TOKEN
10 const char *WIFI_SSID = "chivos"; // Put here your Wi-Fi SSID
11 const char *WIFI_PASS = "12345678"; // Put here your Wi-Fi password
12 const char *DEVICE_LABEL = "coordinador"; // Put here your Device label to which data will be published
13 char *VARIABLE_LABEL = "position1"; // Put here your Variable label to which data will be published
14
15 const int PUBLISH_FREQUENCY = 10000; // Update rate in milliseconds
16
17 unsigned long timer;
18
19 Ubidots ubidots(UBIDOTS_TOKEN);
20 /*
21 mis variables
22 */
23
24
25 bool started= false;//True: Message is strated
26 bool ended = false;//True: Message is finished
27 char incomingByte ; //Variable to store the incoming byte
28 char msg[23]; //Message - array from 0 to 2 (3 values - PWM - e.g. 240)

```

Desarrollo del código para el nodo supervisor

En la figura 33 se puede observar el código que se asigna al nodo supervisor el cual está diseñado para adquirir las coordenadas brindadas por el módulo GPS en tiempo real y a su vez transmitir los datos obtenidos del GPS al nodo coordinador mediante la conexión establecida por los dispositivos Zigbee. Es posible visualizar el código completo en el anexo F.

Figura 33

Código del nodo supervisor

```

Supervisor1.ino
1  #include <SoftwareSerial.h>//incluimos SoftwareSerial
2  #include <TinyGPS.h>//incluimos TinyGPS
3
4  TinyGPS gps;//Declaramos el objeto gps
5  SoftwareSerial serialgps(4,3);//Declaramos el pin 4 Tx y 3 Rx
6
7  //Declaramos la variables para la obtención de datos
8  int identificador=50;
9  float latcero=0.0;
10 float lngcero=0.0;
11
12 void setup()
13 {
14   Serial.begin(9600);//Iniciamos el puerto serie
15   serialgps.begin(9600);//Iniciamos el puerto serie del gps
16   //delay(30000);
17   //Serial.println("E1 LISTO");
18 }
19
20 void loop()
21 {
22   while(serialgps.available())
23   {
24     int c = serialgps.read();
25
26     if(gps.encode(c))
27     {
28       float latitude, longitude;

```

Ensamblaje del prototipo

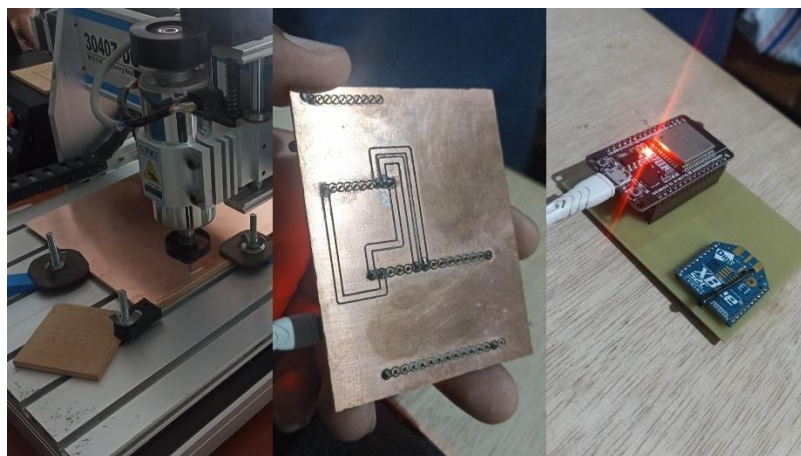
El ensamblaje de prototipos es un paso importante en el proceso de desarrollo de cualquier dispositivo electrónico, para este prototipo se requiere el uso de herramientas y componentes especializados como Arduino, Xbee y Shield de Arduino. Estos componentes se utilizan para crear el prototipo de geolocalización y permiten probar y evaluar antes de entrar en producción.

Pasos para ensamblar el nodo coordinador

Para el ensambaje del nodo coordinador existen dos alternativas posibles, la primera es conectar directamente los pines de alimentación, transmisión y recepción por medio de cables hembra – hembra, sin embargo, con el fin de reducir inconvenientes durante la ejecución del proceso del nodo, la opción con mayor factibilidad es la segunda, ya que, al tener la disponibilidad de una impresora PCB y previamente haber realizado el diseño del esquemático del nodo, se puede proceder con el proceso que se observa en la figura 34.

Figura 34

Creación de PCB para el nodo coordinador



Pasos para ensamblar el nodo supervisor

El nodo supervisor consta inicialmente de un Arduino UNO, un Shield xbee de Arduino y un módulo Xbee s2c, la conexión de estos se lo realizó utilizando espadines hembra – macho, lo cual ha permitido evitar contactos falsos o desconexión de cables. Cada componente fue superpuesto uno encima de otro como se aprecia en la figura 35.

Figura 35

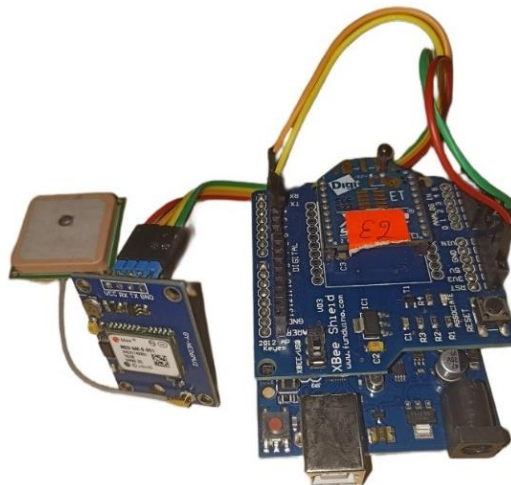
Parte 1 del ensamblaje del nodo supervisor



Una vez verificado que los módulos Xbee s2c de toda la red de sensores se puede conectar, procedemos a conectar el GPS el cual brindara los datos que se transmitirán dentro de la red hacia el nodo coordinador, cabe mencionar que el módulo GPS se encuentra conectado mediante cable con el Shield Xbee de Arduino debido a que cuenta con pines de alimentación y transmisión de datos (rx, tx), los cuales se conectan en pines opuestos, como se aprecia en la figura 36.

Figura 36

Parte 2 del ensamblaje del nodo supervisor



Diseño estructural del prototipo

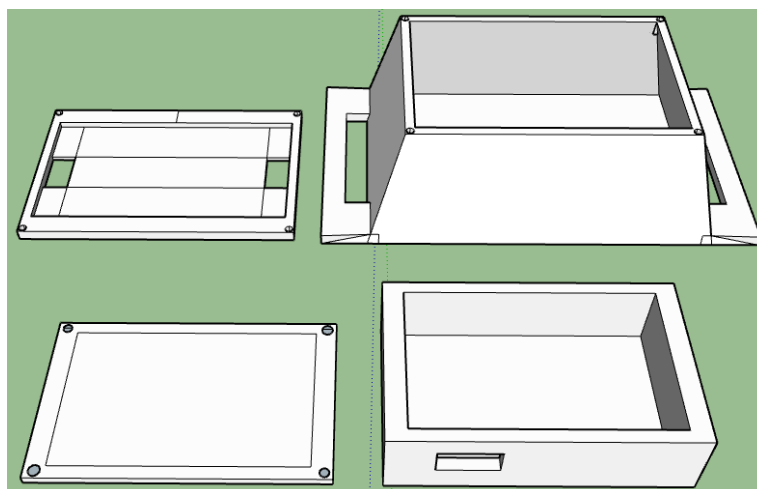
La presentación de un prototipo es un paso crucial en el desarrollo de cualquier proyecto, ya que, es importante asegurarse de que el prototipo cumpla con todos los requisitos y se vea estéticamente agradable. Esta sección se centrará en el diseño final de un prototipo autosostenible geolocalizado con diseño 3D.

La impresión 3D es una tecnología revolucionaria que tiene el potencial de cambiar la forma en que fabricamos productos, ya que, nos permite crear formas y estructuras complejas con facilidad a partir de un archivo digital, con aplicaciones son casi ilimitadas.

Por medio del programa Sketchup, a fin de encontrar la mejor estructura de cada prototipo en escenarios de la vida real, se revisó cada componente del modelo de tal forma que optimice todo el espacio posible y ningún elemento pueda sufrir desperfectos durante la ejecución. Por lo que, en la figura 37 se puede observar los resultados de la investigación realizada.

Figura 37

Diseño de estructura de nodos en Sketchup



Montar un prototipo autosostenible no es tarea fácil, requiere la combinación correcta de componentes como panel solar, Arduino y Xbee para crear un sistema de geolocalización autosostenible, por lo que al culminar los pasos previos a la figura 38, se puede apreciar la culminación del ensamblaje tanto del nodo supervisor como del coordinador.

Figura 38

Presentación final del prototipo



En la figura 38 es posible apreciar ambos nodos ensamblados por completo, permitiendo visualizar la organización de los componentes dentro de los nodos, convirtiendo a este prototipo en un dispositivo funcional que pueda usarse en la vida cotidiana.

Conexión por software

Conectar el software al hardware ahora es más fácil que nunca con la ayuda de Ubidots, ya que, permite a los usuarios conectar su microcontrolador ESP32 con una aplicación de Android por medio de las Apis. Con esta conexión, los usuarios pueden controlar sus dispositivos de forma remota, monitorear datos en tiempo real y crear

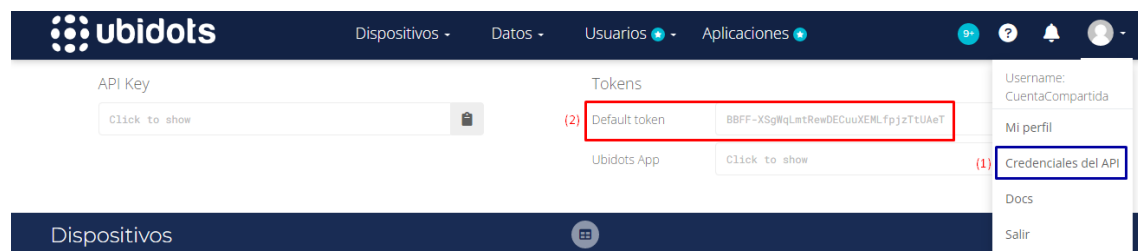
aplicaciones personalizadas para cualquier caso de uso imaginable. Además, la plataforma también proporciona acceso a potentes herramientas de análisis que permiten a los usuarios obtener una visión más profunda de sus datos.

Conexión con Ubidots

Ubidots es una plataforma IoT que permite a los usuarios conectarse y administrar fácilmente sus dispositivos, proporciona una amplia gama de funciones, como visualización de datos, análisis e integración con otros servicios. Al conectar una placa ESP32 a Ubidots, los usuarios pueden crear rápida y fácilmente potentes aplicaciones de IoT.

Figura 39

Credenciales del API Ubidots



En la figura 39 es posible visualizar la información necesaria para enlazar el dispositivo ESP32 con la plataforma Ubidots, para lo cual se tomará el token predeterminado que genera el servidor, lo cual permitirá monitorear el rendimiento del dispositivo en tiempo real, almacenar datos para análisis futuros e incluso desencadenar eventos en función de ciertas condiciones

Figura 40

Configuración del token en el ESP32

```

...../
const char *UBIDOTS_TOKEN = "BBFF-XSgWqLmtRwDEcUuXEMLfjzTtUAeT"; // Put here your Ubidots TOKEN
const char *WIFI_SSID = "chivos"; // Put here your Wi-Fi SSID
const char *WIFI_PASS = "12345678"; // Put here your Wi-Fi password
const char *DEVICE_LABEL = "coordinador"; // Put here your Device label to which data will be published
char *VARIABLE_LABEL = "position1"; // Put here your Variable label to which data will be published

```

La configuración del ESP32 se ve reflejada en la figura 40, donde mediante línea de código se declara el tipo de variable y es identificada como “UBIDOTS_TOKEN”, posterior a ello es necesario conformar el enlace de red mediante Wi-Fi y designar un nombre que identificara al dispositivo, en la plataforma Ubidots se agregara el dispositivo de forma automática como se aprecia en la figura 41.

Figura 41

Dispositivo ESP32 en Ubidots



Eventos de Ubidots

Ubidots Events es una poderosa herramienta que facilita la generación de notificaciones basadas en la ubicación debido a que permite configurar geocercas y crear

disparadores personalizados para que pueda enviar notificaciones por SMS o correo electrónico cuando algo suceda dentro de un área determinada, lo que la convierte en una herramienta ideal monitorear fácilmente cualquier ubicación y recibir alertas cuando algo cambie en esa área.

Un arreglo básico de programación es el si – entonces, este se aplica en diversas áreas en donde se requiere automatizar acciones sencillas, en Ubidots se emplea dicha regla como se observa en la figura 42, para determinar el área segura en donde se encuentra el caprino. Para esto se debe crear una condición para cada variable del nodo supervisor, en donde se selecciona que estará en función de una posición, y que además detecte cuando la variable salga de la geocerca que se delimita en la figura 43, por último, se define el tiempo en que la variable emitirá una notificación al salir del rango.

Figura 42

Condiciones de Geocercas

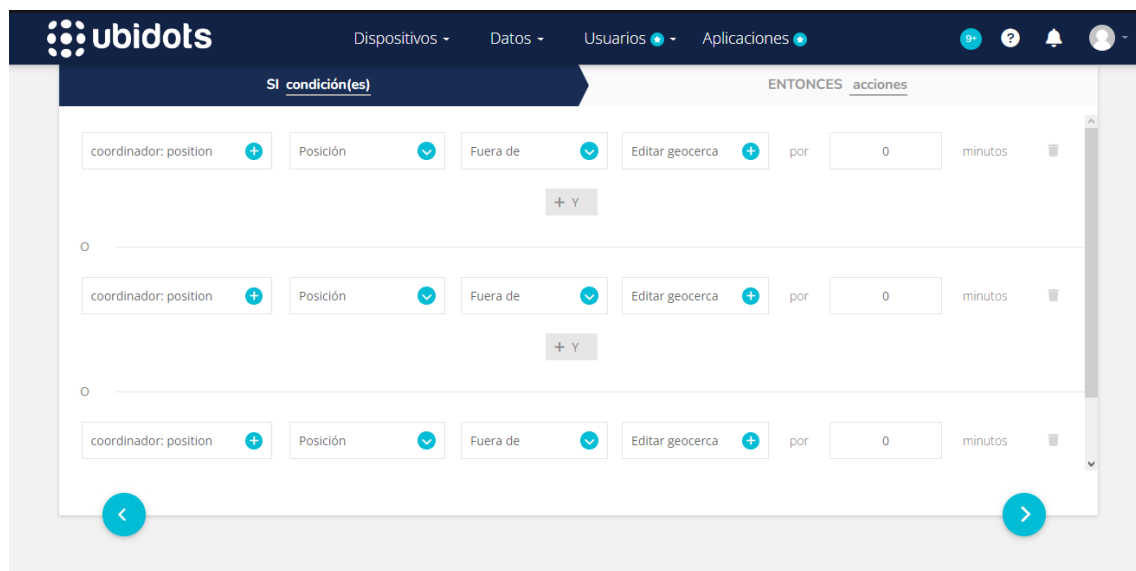
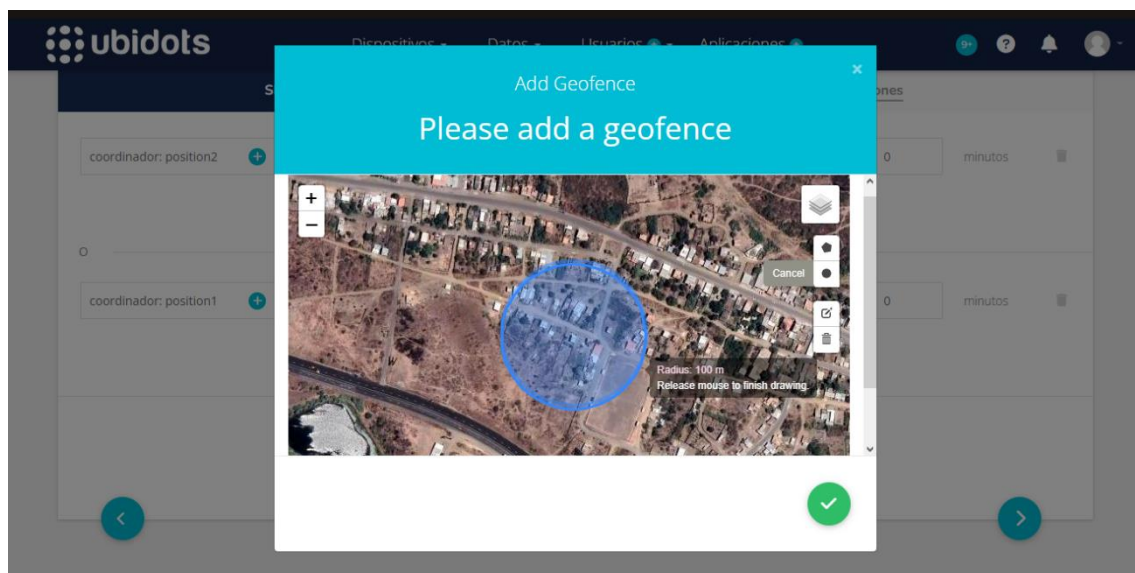


Figura 43

Delimitación de georreferencia



Para crear notificaciones automáticas a través de correo electrónico y SMS. Mediante el uso de geocercas basadas en la ubicación, la plataforma Ubidots puede determinar cuándo los dispositivos ingresan o salen de un área determinada y enviar notificaciones en consecuencia, sin tener que enviar mensajes manualmente cada vez para llegar a las personas adecuadas en el momento adecuado.

Las notificaciones se pueden personalizar en su totalidad, indicando el nodo que se está saliendo del área, además, de emitir por mensaje las coordenadas, la fecha y hora del evento y enviarlas a un número celular o correo electrónico predefinido. Es necesario mencionar que las notificaciones pueden emitirse varias veces en un tiempo predeterminado como se observa en la figura 44 y figura 45.

Figura 44

Modalidad de alerta por correo electrónico

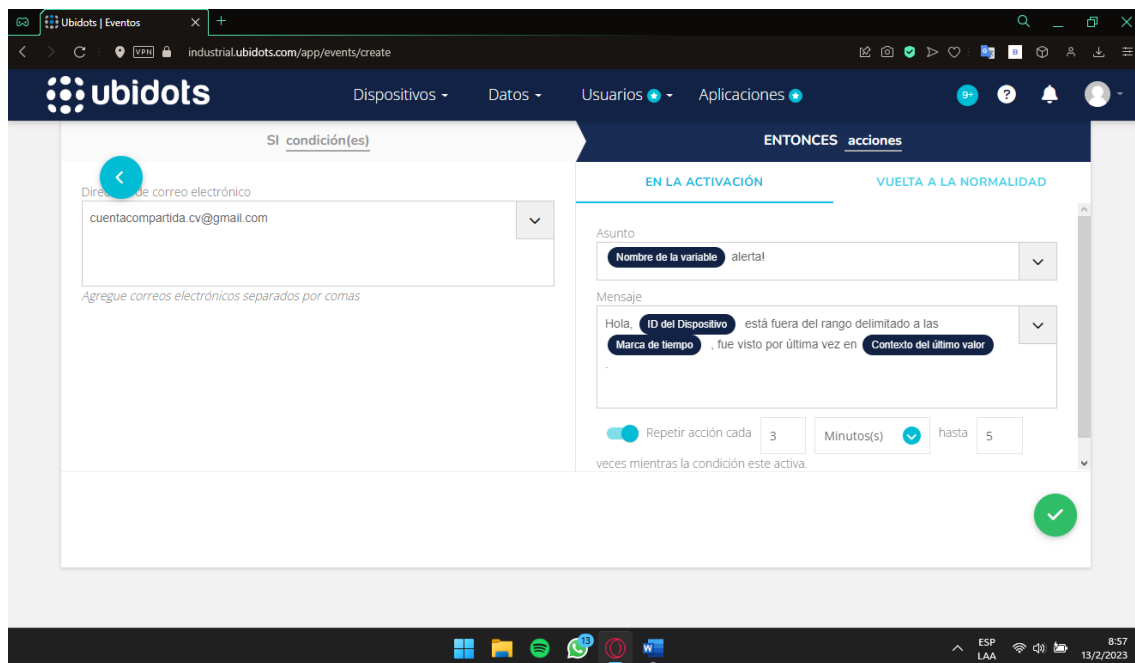
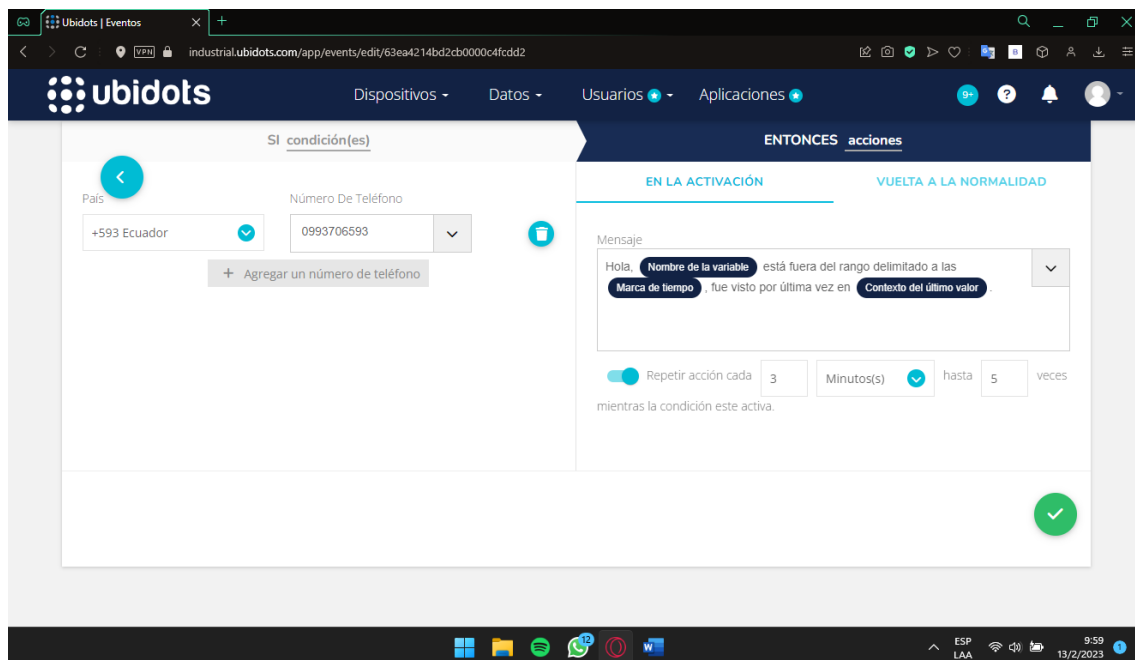


Figura 45

Modalidad de alerta por SMS



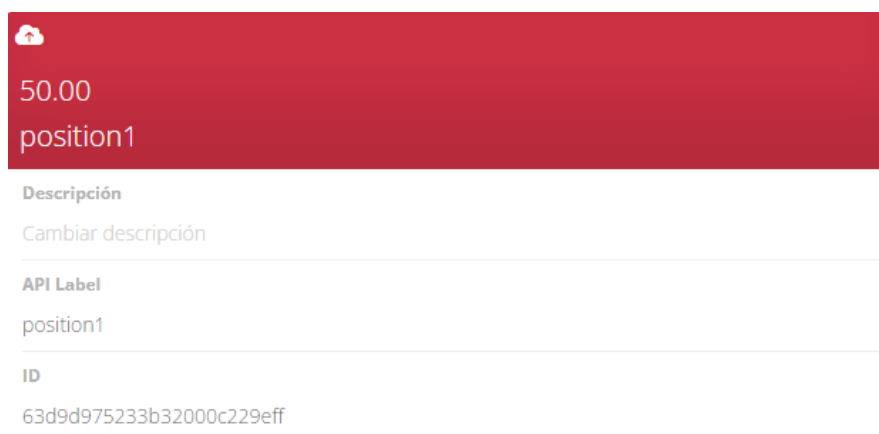
Conexión con aplicación móvil

El desarrollo de aplicaciones móviles es un campo en constante evolución que requiere la combinación correcta de software y funciones para obtener resultados con beneficios en áreas desde entretenimiento hasta de negocios. Android Studio y Ubidots son dos herramientas populares utilizadas por los desarrolladores para lograr dichos beneficios.

La aplicación móvil que se desarrolló para la visualización de las ubicaciones de cada nodo supervisor necesita poder extraer datos de la plataforma Ubidots, esto se logra por medio del token que se visualiza en la figura 39, también, se necesita de cada identificador del nodo asignado automáticamente por el servidor en la figura 46 se puede visualizar el ID del supervisor variable position1.

Figura 46

ID del nodo supervisor 1



Las credenciales son agregadas por código respectivamente a la sección en donde se ha configurado la recepción de datos enviadas por Ubidots, mismas que deberán ser gestionadas por las Api por medio del protocolo de red que se ha empleado,

en la figura 47 se observan los comandos necesarios para la implantación de los tokens al código de la aplicación

Figura 47

Tokens en código de la aplicación móvil

```

Future<ResponseCoord?> getVariable(String id) async
{
  try{
    Uri url=Uri.parse(_url+'api/v1.6/variables/$id/values/?page=1&page_size=1');
    Map<String,String> headers={
      'Content-type':'application/json',
      'X-Auth-Token':'BBFF-XSgWqLmtRewDECuuXEMLfpjzTtUAeT'
    };
  }
}

Future<Set<Marker>> _getMarkers() async{
  _responseCoord= await _variableProvider.getVariable('63d9d975233b32000c229eff');
  _supervisor2= await _variableProvider.getVariable('63d9d5b35d61c3000b735300');
  _supervisor3= await _variableProvider.getVariable('63c0659992c0e6000ce99bf5');

  final Set<Marker> markers={};

  markers.add(Marker(markerId: MarkerId("1"),
    position: LatLng(double.parse(_responseCoord!.lat!),double.parse(_responseCoord!.lng!)),
    infoWindow: InfoWindow(title: "Supervisor 1"),
    icon: BitmapDescriptor.defaultMarker
  )); // Marker

  markers.add(Marker(markerId: MarkerId("2"),
    position: LatLng(double.parse(_supervisor2!.lat!),double.parse(_supervisor2!.lng!)),
    infoWindow: InfoWindow(title: "Supervisor 2"),
    icon: BitmapDescriptor.defaultMarkerWithHue(23)
  )); // Marker
}

```

Costos de implementación

La implementación de un proyecto de IoT es un proceso complejo que requiere una cuidadosa planificación y consideración de la viabilidad económica. Ubidots es una gran herramienta para ayudar a crear proyectos autosuficientes sin tener que preocuparse por los costos de implementar un servidor propio y así crear fácilmente soluciones IoT que sean rentables y brinden valor a largo plazo.

La presente propuesta tecnológica se basa en la implementación de un prototipo a un 15% de la población de estudio, por lo que, es necesario recalcar que dentro del presupuesto a cada especie caprina se le asigna un nodo supervisor y un nodo coordinador por todos los supervisores, por lo que se procede a presentar un presupuesto de cada nodo y un presupuesto total de la ejecución del proyecto.

Tabla 15

Costo del nodo Supervisor

ítem	Artículo	Precio unitario	Cantidad	Precio total
1	Tarjeta de desarrollo	\$27,00	3	\$81,00
2	Arduino UNO	\$30,00	3	\$90,00
3	Sensor GPS	\$18,00	3	\$54,00
4	Shield Xbee para Arduino	\$20,00	3	\$60,00
5	Explorer Xbee	\$80,00	3	\$240,00
6	Módulo XBee S2C	\$12,00	3	\$36,00
7	Panel solar	\$8,00	3	\$24,00
8	Módulo de carga	\$20,00	3	\$60,00
	Par de baterías de litio 3.7V			

9	Impresión 3D	\$25,00	3	\$75,00
10	Faja Ant. XL Reflectiva	\$15,00	3	\$45,00
11	Misceláneos	\$20,00	3	\$60,00
Totales		\$275,00		\$825,00

Tabla 16*Costo del nodo coordinador*

Ítem	Artículo	Precio unitario	Cantidad	Precio total
1	NodeMCU- ESP32	\$18,00	1	\$18,00
2	Módulo XBee S2C	\$80,00	1	\$80,00
3	Explorer Xbee	\$20,00	1	\$20,00
4	Circuito impreso	\$35,00	1	\$35,00
5	Impresión 3D	\$20,00	1	\$20,00
6	Misceláneos	\$10,00	1	\$10,00
Totales		\$183,00		\$183,00

Tabla 17*Costo total de implementación del prototipo*

Ítem	Artículo	Precio unitario	Cantidad	Precio total
1	Nodo coordinador	\$183,00	1	\$178,00
2	Nodo supervisor	\$275,00	3	\$825,00
	Membresía			
3	Ubidots Professional	\$199,00/mes	1	\$199,00
Total				\$1 202,00

El presupuesto es un paso importante para determinar la viabilidad de implementar el proyecto y nos ha ayudado a evaluar el costo por cada nodo como se aprecia en la tabla 17 y se encuentra detallado con mayor profundidad en la tabla 15 y tabla 16. Con esta información, se vuelve más fácil identificar áreas potenciales donde se pueden reducir los costos y explorar alternativas para una mejor rentabilidad.

Capítulo IV PRUEBAS Y RESULTADOS DE LA PROPUESTA

Pruebas de funcionamiento

Las pruebas de funcionalidad son una parte importante del proceso de desarrollo ya que ayuda a garantizar que el dispositivo funcione como se espera y mediante una cuidadosa observación y análisis, las pruebas de funcionalidad identificarán posibles problemas con el diseño del prototipo y ayudar a garantizar que cumpla con los requisitos del usuario.

Prueba de ubicación del prototipo

En la primera prueba de ubicación del prototipo autosostenible geolocalizado se coloca el nodo en la zona ventral del caprino como se observa en la figura 48, al colocarlo en esta posición y realizar una observación detallada, observamos que el nodo dificulta y obstruye las actividades del animal, confirmando el estudio hecho en el estado del arte del capítulo 2.

Figura 48

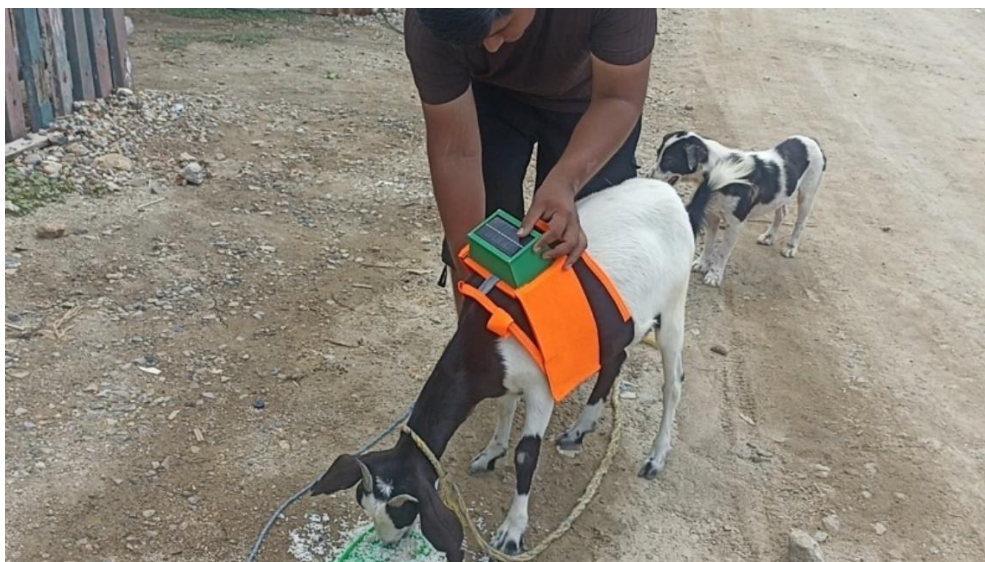
Primera prueba de ubicación



Debido al tamaño final del prototipo, el nodo quedara ubicado en la segunda posición optima detallada por el estado del arte, esta se encuentra ubicada en la zona dorsal del caprino como se visualiza en la figura 49. Dicha ubicación permite al caprino realizar actividades habituales sin inconvenientes.

Figura 49

Segunda prueba de ubicación



Prueba de alcance y geocerca

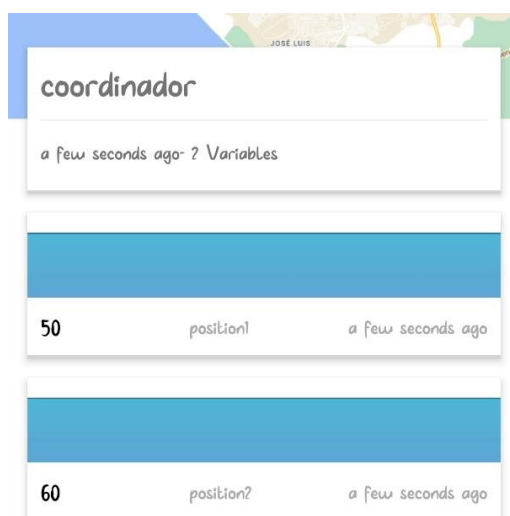
La prueba de rango es un paso crítico en el proceso de configuración de una red inalámbrica. Implica probar los nodos coordinador, supervisor y enlaces para asegurarse de que todos funcionan correctamente. La prueba de rango también incluye probar la aplicación móvil para asegurarse de que pueda comunicarse con la red. Las pruebas de rango son necesarias para garantizar que la red funcione de manera óptima y para identificar cualquier problema potencial que exista durante la ejecución del prototipo.

Al iniciar la prueba de alcance se debe verificar que tanto el nodo coordinador como el nodo supervisor tengan conexión entre sí y salida a internet para poder

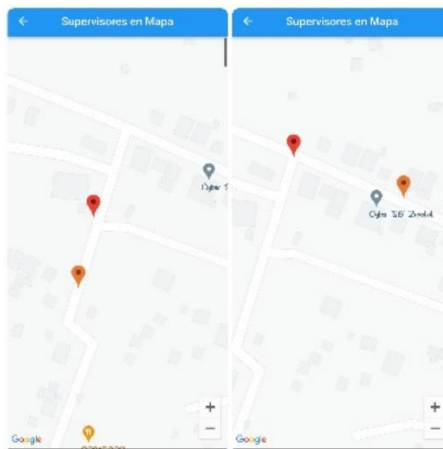
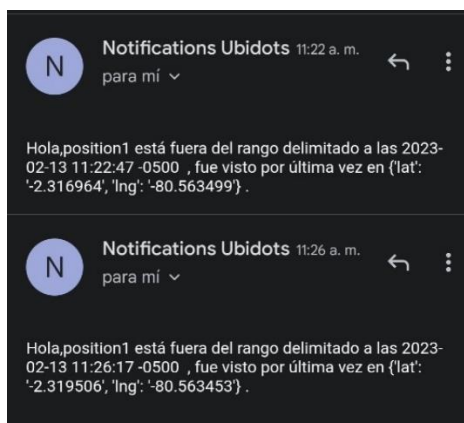
transmitir datos hacia la aplicación explicada en el capítulo 3. En la figura 50 se puede observar cómo ambos nodos presentan la etiqueta “a few seconds ago” haciendo referencia a que hace poco segundos obtuvo la última actualización de ubicación proporcionada por cada nodo respectivamente.

Figura 50

Conectividad de los nodos con el servidor de Ubidots



Una vez ubicado cada nodo en el caprino que corresponde, el animal procede con las acciones que realiza normalmente, al pastorear se alejan del nodo coordinador como se observa en la figura 51, específicamente el supervisor 1 con la etiqueta de color naranja. En la figura 52 se visualizan las notificaciones de alerta que genera Ubidots al detectar que el nodo sobrepasa el límite de la geocerca, la cual corresponde a un valor de 100 metros, y dejando de emitir notificaciones a partir de los 110 metros por obstáculos en la línea de vista.

Figura 51*Supervisor1 en mapa***Figura 52***Notificaciones de alerta del supervisor1*

Con respecto al nodo supervisor 2, el cual corresponde a la etiqueta roja de la figura 53 la ruta del caprino es distinta, sin embargo, al igual que el nodo supervisor 1 empezó a enviar alertas por correo como se aprecia en la figura 54 una vez sobrepasada la geocerca de 100 metros. Al tener menos obstaculos entre los nodos la línea de vista resulta tener una mayor efectividad, por lo que, en el presente escenario la conexión

empezo a presentar intermitencias a una distancia aproximada de 150 metros del nodo coordinador.

Figura 53

Supervisor2 en mapa



Figura 54

Notificaciones de alerta del supervisor2



Consumo de Energía

A medida que el mundo depende cada vez más de la tecnología, es importante comprender el consumo de energía de los diferentes dispositivos, con la ayuda de pruebas y análisis de prototipos, podemos medir la duración de funcionamiento de un dispositivo con una sola carga de batería.

Digi-Key Electronics ha desarrollado una ecuación para calcular el consumo de energía de varios tipos de dispositivos, en esta fórmula se tiene en cuenta factores como la capacidad de la batería utilizada y corriente consumida durante el funcionamiento con una sola carga. Al usar la expresión detallada en la ecuación 1 es posible medir con precisión el consumo de energía de cualquier dispositivo

Ecuación 1

Ecuación vida de la batería

$$Vida\ de\ la\ Bateria = \frac{Capacidad\ de\ la\ Bateria\ (mAh)}{Corriente\ Consumida\ (mA)} * 0.70 \quad (1)$$

Las baterías utilizadas en el prototipo son 18650 compuestas por Ion de litio (Li-Ion) y cuentan con 3.7 voltios, 2500 mAh cada una mientras que el prototipo cuenta con un consumo por parte del Arduino UNO es de 50 mA, el GPS consume 100 mA y el Xbee S2C 120 mA de consumo. Es necesario mencionar que estos se obtuvieron de sus datasheets respectivos. Para obtener el consumo exacto del nodo supervisor se debe sumar cada uno de los datos previamente mencionados.

Ecuación 2

Ecuación corriente de carga

$$\text{Corriente Consumida} = I_{\text{Arduino UNO}} + I_{\text{Xbee S2C}} + I_{\text{GPS NEO6M}} \quad (2)$$

$$\text{Corriente Consumida} = 50 \text{ mA} + 120 \text{ mA} + 100 \text{ mA}$$

$$\text{Corriente Consumida} = 270 \text{ mA}$$

Posterior al cálculo teórico de la corriente consumida por el nodo supervisor se obtuvo como resultado de la ecuación 2 el consumo asciende a 270 mA, después de este cálculo, es posible resolver la ecuación 1 por lo cual se procede a reemplazar los valores requeridos.

$$\text{Vida de la Bateria} = \frac{5000 \text{ (mAh)}}{270 \text{ (mA)}} * 0.70$$

$$\text{Vida de la Bateria} = 18.52 * 0.70$$

$$\text{Vida de la Bateria} = 12.96 \text{ [horas]}$$

El resultado obtenido de la ecuación 1 indica que la vida útil de las baterías que forman parte del nodo supervisor es de 12,96 horas antes de que estas se descarguen por completo, en la práctica las baterías dieron una autonomía de 11.30 horas. Como ya se obtuvo los valores correspondientes a la vida de la batería se puede calcular el porcentaje de eficiencia del nodo, para ello se utiliza la ecuación 3 (error relativo) con los valores obtenidos anteriormente.

Ecuación 3

Ecuación eficiencia del nodo

$$Eficiencia\ del\ nodo = \frac{Valor\ Practico}{Valor\ Teorico} * 100 \quad (3)$$

$$Eficiencia\ del\ nodo = \frac{11.30\ [h]}{12.96\ [h]} * 100$$

$$Eficiencia\ del\ nodo = 0.87 * 100$$

$$Eficiencia\ del\ nodo = 87\%$$

La eficiencia del dispositivo es un factor importante para considerar al desarrollar este prototipo, debido a que la batería de litio ha proporcionado hasta un 87% de eficiencia lo que la convierte en una excelente opción, ya que, tiene una larga vida útil y se puede utilizar en una variedad de aplicaciones, como navegación y rastreo.

Margen de error en el Almacenamiento de los datos

El rango de error es un concepto fundamental al momento de almacenar datos y bases de datos. Es la medida de la precisión de un punto determinado en relación con su valor real. Ubidots, una plataforma basada en la nube para la adquisición y el análisis de datos permite al usuario definir rangos de fechas para monitorear la precisión de sus datos almacenados en intervalos específicos.

Los datos que se obtienen por medio del nodo coordinador almacenados en el servidor de Ubidots, se pueden verificar que llegan correctamente por la gráfica de la figura 55, ya que, al tener valores constantes en la etiqueta del supervisor la recta que cruza por la gráfica debe ser una línea, de lo contrario aparecerán valores aleatorios

como se observa en la figura 56 en el eje de identificador, denotando que existe un error en la trama de datos.

Figura 55

Gráfica de continuidad para la obtención de datos

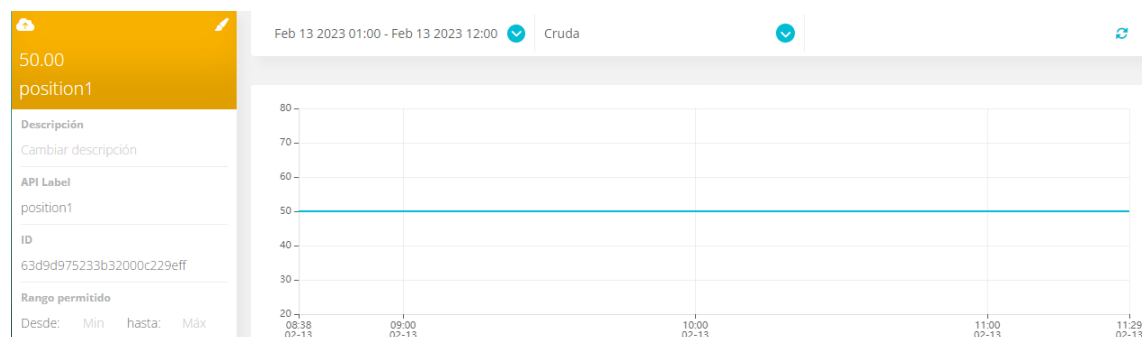
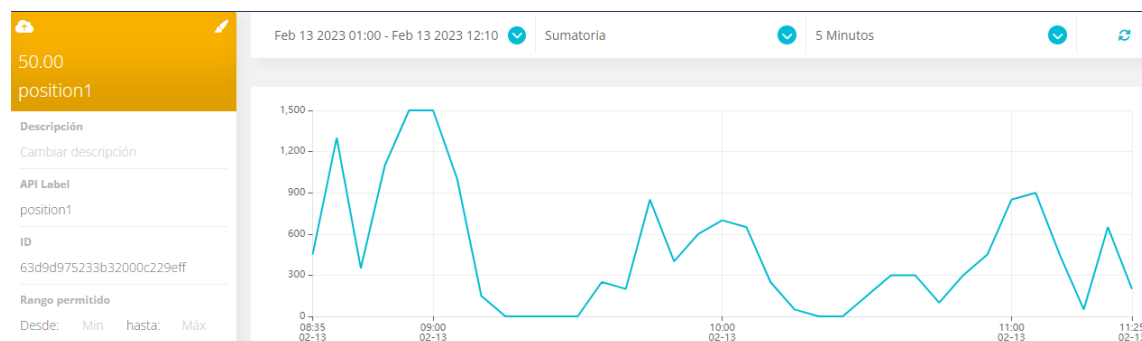


Figura 56

Gráfica de discontinuidad al obtener datos



Con el fin de determinar el porcentaje de pérdida entre coordenadas se empleó la ecuación 4, la cual es la razón entre los errores registrados y el total de muestras obtenidas durante la fase de pruebas que corresponde a un total de 173 minutos, en este lapso se obtuvo 20 datos perdidos en la base de datos del servidor, esto se lo puede observar en la figura 57, donde se visualiza los intervalos de tiempo en donde se considera una pérdida de datos por minutos.

Figura 57

Datos perdidos en la base de datos de Ubidots

2023-02-13 10:42:58 -05:00	50.00	{"lat": "-2.318087", "lng": "-80.564186"}
2023-02-13 10:42:48 -05:00	50.00	{"lat": "-2.318087", "lng": "-80.564193"}
2023-02-13 10:41:58 -05:00	50.00	{"lat": "-2.318092", "lng": "-80.564186"}
2023-02-13 10:39:58 -05:00	50.00	{"lng": "-80.564193", "lat": "-2.318076"}
2023-02-13 10:39:48 -05:00	50.00	{"lng": "-80.564193", "lat": "-2.318075"}
2023-02-13 10:38:58 -05:00	50.00	{"lng": "-80.564186", "lat": "-2.318079"}
2023-02-13 10:37:58 -05:00	50.00	{"lat": "-2.318083", "lng": "-80.564186"}
2023-02-13 10:36:57 -05:00	50.00	{"lat": "-2.318077", "lng": "-80.564208"}
2023-02-13 10:35:58 -05:00	50.00	{"lng": "-80.564201", "lat": "-2.318065"}
2023-02-13 10:33:27 -05:00	50.00	{"lng": "-80.564193", "lat": "-2.318068"}
2023-02-13 10:33:17 -05:00	50.00	{"lng": "-80.564186", "lat": "-2.318067"}

Ecuación 4

Ecuación del error en el almacenamiento

$$Error_{Almacenamiento} = \frac{\text{Errores encontrados}}{\text{Muestra total}} * 100 \quad (4)$$

El análisis que se realizó para determinar la cantidad de elementos perdidos consiste en la visualización de los intervalos en donde no existe trama, por ejemplo, del minuto 10:41 a 10:39 hubo una pérdida de 1 dato, de igual forma en lapso de 10:33 a 10:35 existe un dato perdido. Una vez determinada la cantidad de datos al emplear en la fórmula de la ecuación 4, se procede a realizar un remplazo de datos en la expresión detallada en la ecuación 5.

$$Error_{Almacenamiento} = \frac{20}{173} * 100 \quad (5)$$

$$Error_{Almacenamiento} = 0.116 * 100$$

$$Error_{Almacenamiento} = 11.6 \%$$

Después de analizar los errores de almacenamiento, resulta una tasa de pérdida de datos del 11,6% mientras el dispositivo envía datos al servidor, por consecuencia el dispositivo final tiene una eficiencia del 88,4% al momento de almacenar las coordenadas en la base de datos de la nube.

CONCLUSIONES

El presente prototipo responde a la necesidad del extravío de estas especies ya sea en circunstancias normales o el abigeato de estas mismas. Por este motivo el diseño de este prototipo permitió evitar la pérdida de los caprinos, ya que a través de las redes de sensores se enviaron los datos en tiempo real mientras que el caprino se alejaba de la zona delimitada.

Mediante la realización del estado del arte se obtuvo el suficiente sustento teórico en base a los comportamientos individuales de los caprinos, de esta manera los materiales seleccionados para el diseño de prototipo se realizaron de acuerdo a las necesidades fisiológicas de estas especies, por este motivo se optó por un diseño rectangular anudado por un sistema reflector que se colocó en el torso del caprino, ubicado en la parte superior para que el espécimen realice sus actividades diarias sin dificultad alguna.

A través de la revisión literaria se pudo obtener que una de las mejores tecnologías de redes inalámbricas corresponde a Zigbee, ya que, cuenta con una larga historia en la industria de monitoreo y control, lo que le ha permitido mantenerse a la vanguardia como un protocolo seguro a diferencia del resto de su competencia.

Con ayuda del software de simulación Sketchup se pudo obtener una idea previa del diseño del prototipo, tamaño y además de brindar datos que sirvan de guía para ubicar el nodo supervisor en el caprino según lo investigado en el estado del arte realizado en el capítulo 2.

Por medio de la correcta configuración de las placas de desarrollo y cada módulo complementario respectivamente, se logró la implementación de una red inalámbrica de sensores, misma que por medio del desarrollo de instrucciones en plataformas de código abierto se llevó a cabo la conexión al servidor por medio del protocolo de red mqtt.

El diseño de la aplicación móvil a tenido un uso fundamental en la interacción con el usuario final, ya que, es el medio por el cual se están presentando los datos recopilados sobre los nodos supervisores hacia la plataforma de Ubidots indicando si la especie caprina aún se encuentra dentro del área delimitada

Mediante la implementación de la modalidad de alerta por correo electrónico y mensaje de texto, se logró definir el rango máximo de conexión que existe entre los nodos del prototipo. Además, por medio de la función de actualización continua de datos recolectados por el nodo coordinador y enviados hacia el servidor se puede mantener una visualización de movimiento en tiempo real del caprino en la aplicación móvil.

La implementación del presente prototipo autosostenible geolocalizado basado en una red de sensores inalámbricos permitió establecer una geocerca que delimite un área y tener datos en tiempo real de cualquier anomalía que provoque que las especies caprinas abandonen la propiedad donde residen, todo esto mediante notificaciones SMS o emails.

RECOMENDACIONES

Se recomienda comprar la licencia de la plataforma Ubidots, para poder disfrutar de todos los beneficios que facilitan el monitoreo y almacenamiento de datos generados por los nodos sensores a implementar, además, de habilitar la sección para generar eventos en caso de requerir establecer una condicional que genere una acción.

Para poder mejorar el alcance de la comunicación inalámbrica Zigbee y la exactitud de la ubicación es recomendable implementar tecnologías adicionales como la GPRS para transmitir datos de geoposicionamiento con una menor latencia.

La aplicación móvil actualmente tiene un gran margen de mejora, ya que, al funcionar por medio de las Apis que proporciona la plataforma Ubidots son limitadas las funcionalidades que se pueden implementar dentro del aplicativo, por lo que, se recomienda el desarrollo de una nueva propuesta tecnológica titulada “Desarrollo y diseño de un servidor implementando funciones de almacenamiento y gestión de recursos para la conexión a internet de nodos inalámbricos de sensores en un escenario controlado de los laboratorios de la Universidad Estatal Península de Santa Elena” para la carrera de Tecnologías de la información.

El prototipo se puede mejorar por medio de la implementación de módulos sensoriales complementarios para la obtención de datos que faciliten conocer el estado actual del caprino, por ejemplo, un módulo de detección de pulso cardiaco serviría para saber si el prototipo se encuentra en el cuerpo de caprino, además, permitiría la implementación de nuevas funcionalidades en el aplicativo móvil como actividad física, niveles de pánico, entre otros, todos estos datos serán en beneficio del cliente.

ANEXOS

Anexo A

Datasheet de Xbee/XBee-PRO S2C Zigbee

Technical specifications		Performance specifications	
Performance specifications			
The following table describes the performance specifications for the devices.			
<i>Note</i> Range figure estimates are based on free-air terrain with limited sources of interference. Actual range will vary based on transmitting power, orientation of transmitter and receiver, height of transmitting antenna, height of receiving antenna, weather conditions, interference sources in the area, and terrain between receiver and transmitter, including indoor and outdoor structures such as walls, trees, buildings, hills, and mountains.			
Specification	XBee Zigbee S2C	XBee-PRO Zigbee S2C	
Indoor/urban range	Up to 60 m (200 ft)	Up to 90 m (300 ft)	
Outdoor RF line-of-sight range	Up to 1200 m (4000 ft)	Up to 3200 m (2 mi)	
Transmit power output (maximum)	6.3 mW (+8 dBm), boost mode 3.1 mW (+5 dBm), normal mode channel 26 max power (+3 dBm)	63 mW (+18 dBm)	
RF data rate	250,000 b/s		
Receiver sensitivity	-102 dBm, boost mode -100 dBm, normal mode	-101 dBm	
Power requirements			
The following table describes the power requirements for the XBee/XBee-PRO Zigbee RF Module.			
Specification	XBee	XBee-PRO	
Supply voltage	2.1 - 3.6 V	2.7 - 3.6 V	
Transmit current (typical, VCC = 3.3 V)	45 mA (8 dBm, Boost mode) 33 mA (5 dBm, Normal mode)	120 mA (18 dBm)	
Idle / receive current (typical, VCC = 3.3 V)	31 mA (Boost mode) 28 mA (Normal mode)	31 mA	
Power-down current	<1 uA @ 25C	<1 uA @ 25C	
General specifications			
The following table describes the general specifications for the devices.			
Specification	XBee Zigbee S2C	XBee-PRO Zigbee S2C	
Operating frequency band	ISM 2.4 - 2.5 GHz		

XBee/XBee-PRO S2C Zigbee RF Module User Guide

Technical specifications		Networking and security specifications	
Specification	XBee Zigbee S2C	XBee-PRO Zigbee S2C	
Form factor	through-hole, surface-mount		
Dimensions	through-hole: 2.438 x 2.761 cm (0.960 x 1.087 in) surface-mount: 2.199 x 3.4 x 0.305 cm (0.866 x 1.33 x 0.120 in)	through-hole: 2.438 x 3.294 cm (0.960 x 1.297 in) surface-mount: 2.199 x 3.4 x 0.305 cm (0.866 x 1.33 x 0.120 in)	
Operating temperature	-40 to 85 °C (industrial)		
Antenna options	through-hole: PCB antenna, UFL connector, RP SMA connector, or integrated wire surface-mount: RF pad, PCB antenna, or UFL connector		
Networking and security specifications			
The following table describes the networking and security specifications for the devices.			
Specification	XBee Zigbee S2C	XBee-PRO Zigbee S2C	
Supported network topologies	Point-to-point, point-to-multipoint, peer-to-peer, and DigiMesh		
Number of channels	16 Direct sequence channels	15 Direct sequence channels	
Interface immunity	Direct Sequence Spread Spectrum (DSSS)		
Channels	11 to 26		
Addressing options	PAN ID and addresses, cluster IDs and endpoints (optional)		
Communication interface specifications			
The following table provides the device's communication interface specifications.			
Interface options			
UART	250 Kb/s maximum		
SPI	5 Mb/s maximum (burst)		

Regulatory conformity summary

This table describes the agency approvals for the devices.

Note Legacy XBee-PRO SMT (model: PRO S2C; hardware version 21xx) has different FCC and IC IDs. For more information, see [Regulatory information](#).

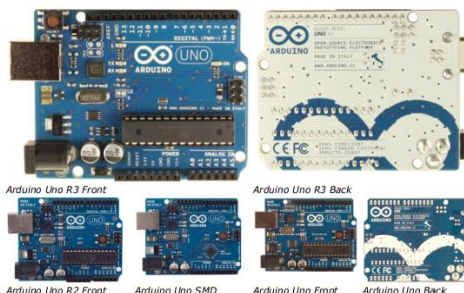
Approval	XBee (surface-mount)	XBee-PRO (surface-mount)	XBee (through-hole)	XBee-PRO (through-hole)	XBee S2D (surface-mount)
United States (FCC Part 15.247)	FCC ID: MCQ-XBS2C	FCC ID: MCQ-XBPS2C (revision K and earlier) FCC ID: MCQ-PS2CSM (revision L and later) FCC ID: MCQ-PS2C5 (for model PS2C5)	FCC ID: MCQ-S2C7H	FCC ID: MCQ-PS2C7H	FCC ID: MCQ-S2DSM
Innovation, Science and Economic Development Canada (ISED)	IC: 1846A-XBS2C	IC: 1846A-XBPS2C (revision K and earlier) IC: 1846A-PS2CSM (revision L and later) IC: 1846A-PS2C5 (for model PS2C5)	IC: 1846A-S2C7H	IC: 1846A-PS2C7H	IC: 1846A-S2DSM
FCC/IC Test Transmit Power Output range	-26 to +8 dBm	-0.7 to +19.4 dBm	-26 to +8 dBm	+1 to +19 dBm	-10 to +8 dBm
Europe (CE)	Yes		Yes		Yes
Australia	RCM	RCM	RCM	RCM	
Japan	R201WW10215369		R210-105563		
Brazil	ANATEL: 0616-15-1209	ANATEL: 07440-19-01209 (XBP24CZ7PIS-004, XBP24CZ7RIS-004, XBP24CZ7UIS-004)	ANATEL: 4556-15-1209	ANATEL: 4077-15-1209	
Mexico	RCPDIXB19-1821	RCPDIPS19-1817-A1	RCPDIS219-	RCPDIPS19-	

Approval	XBee (surface-mount)	XBee-PRO (surface-mount)	XBee (through-hole)	XBee-PRO (through-hole)	XBee S2D (surface-mount)
			1821-A1	1817	
South Korea	MSIP-CRM-DIG-XBee-S2C		MSIP-CRM-DIG-XBee-S2C-TH		
RoHS	Compliant				

Anexo B

Datasheet de Arduino UNO

Arduino Uno



Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega16U2 (ATmega8U2 up to version R2) programmed as a USB-to-serial converter.

Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

Revision 3 of the board has the following new features:

- **1.0 pinout:** added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the JOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- ATmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V

- **SP1: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the [SPI library](#).
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and ATmega328 ports](#). The mapping for the ATmega8, 168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, [on Windows, a .inf file is required](#). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the I2C bus; see the [documentation](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [header file](#)). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available . The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Unix) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Automatic (Software) Reset

Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: [arduino-uno-Rev3-reference-design.zip](#) (NOTE: works with Eagle 6.0 and newer)

Schematic: [arduino-uno-Rev3-schematic.pdf](#)

Note: The Arduino reference design can use an ATmega8, 168, or 328. Current models use an ATmega328, but an ATmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection


The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

Anexo C

Datasheet de GPS NEO6M



NEO-6 - Data Sheet

1 Functional description

1.1 Overview

The NEO-6 module series is a family of stand-alone GPS receivers featuring the high performance u-blox 6 positioning engine. These flexible and cost effective receivers offer numerous connectivity options in a miniature 16 x 12.2 x 2.4 mm package. Their compact architecture and power and memory options make NEO-6 modules ideal for battery operated mobile devices with very strict cost and space constraints.

The 50-channel u-blox 6 positioning engine boasts a Time-To-First-Fix (TTFF) of under 1 second. The dedicated acquisition engine, with 2 million correlators, is capable of massive parallel time/frequency space searches, enabling it to find satellites instantly. Innovative design and technology suppresses jamming sources and mitigates multipath effects, giving NEO-6 GPS receivers excellent navigation performance even in the most challenging environments.


1.2 Product features

Model	Type	Supply	Interface	Features
NEO-6G	GPS PPP Trimble Real Time Dead Reckoning	1.75 V _A -2.0 V _B 2.7 V _A -3.0 V _B	UART USB SPI	DCDC-GPS-compliant Programmable I/O RTK RTK-Correlator Assisted A-GPS ¹ Assisted SBAS ² Configurable pins Temperature External interrupt ³ Watchdog
NEO-6H	•	•	•	•
NEO-6I	•	•	•	•
NEO-6J	•	•	•	•
NEO-6K	•	•	•	•
NEO-6L	•	•	•	•
NEO-6M	•	•	•	•
NEO-6N	•	•	•	•
NEO-6P	•	•	•	•
NEO-6Q	•	•	•	•
NEO-6R	•	•	•	•
NEO-6S	•	•	•	•
NEO-6T	•	•	•	•

• Requires external components and integration on application processor

Table 1: Features of the NEO-6 Series

¹ All NEO-6 modules are based on GPS chips qualified according to AEC-Q100. See Chapter 5.1 for further information.



NEO-6 - Data Sheet

1.3 GPS performance

Parameter	Specification
Receiver type	50 Channels GPS L1 frequency, CIA Code SBAS: WAAS, EGNOS, MSAS
Time-To-First-Fix ¹	NEO-6GPT NEO-6MV NEO-6P
Cold Start ²	26 s 27 s 32 s
Warm Start ²	26 s 27 s 32 s
Hot Start ²	1 s 1 s 1 s
Aided Starts ³	1 s <3 s <3 s
Sensitivity ⁴	NEO-6GPT NEO-6MV NEO-6P
Tracking & Navigation	-162 dBm -161 dBm -160 dBm
Reacquisition ⁵	-160 dBm -160 dBm -160 dBm
Cold Start (without aiding)	-148 dBm -147 dBm -146 dBm
Hot Start	-157 dBm -156 dBm -155 dBm
Maximum Navigation update rate	NEO-6GPT NEO-6MV
5Hz 1 Hz	
Horizontal position accuracy ⁶	GPS 2.5 m SBAS 2.0 m SBAS + PPP <1 m (2D, R50) ⁷ SBAS + PPP <2 m (3D, R50) ⁷
Configurable Timepulse frequency range	NEO-6GMV NEO-6T
0.25 Hz to 1 kHz 0.25 Hz to 10 MHz	
Accuracy for Timepulse signal	RMS 30 ns 99% <60 ns Granularity 21 ns Compensated ⁸ 15 ns
Velocity accuracy ⁹	0.1 m/s
Heading accuracy ⁹	0.5 degrees
Operational Limits	Dynamic ≤4 g Altitude ¹⁰ 50,000 m Velocity ¹⁰ 500 m/s

Table 2: NEO-6 GPS performance

1.4 Block diagram

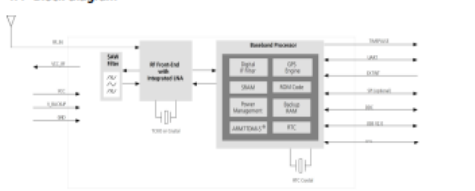


Figure 1: Block diagram (For available options refer to the product features table in section 1.2).

1.5 Assisted GPS (A-GPS)

Supply of aiding information like ephemeris, almanac, rough last position and time and satellite status and an optional time synchronization signal will reduce time to first fix significantly and improve the acquisition sensitivity. All NEO-6 modules support the u-blox AssistNow Online and AssistNow Offline A-GPS services¹ and are OMA SUPL compliant.

1.6 AssistNow Autonomous

AssistNow Autonomous provides functionality similar to Assisted-GPS without the need for a host or external network connection. Based on previously broadcast satellite ephemeris data downloaded to and stored by the GPS receiver, AssistNow Autonomous automatically generates accurate satellite orbital data ("AssistNow Autonomous data") that is usable for future GPS position fixes. AssistNow Autonomous data is reliable for up to 3 days after initial capture.

u-blox² AssistNow Autonomous benefits are:

- Faster position fix
- No connectivity required
- Complementary with AssistNow Online and Offline services
- No integration effort, calculations are done in the background

¹ For more details see the u-blox 6 Receiver Description including Protocol Specification [2].

² AssistNow Offline requires external memory.

1.7 Precision Timing

1.7.1 Time mode

NEO-6T provides a special Time Mode to provide higher timing accuracy. The NEO-6T is designed for use with stationary antenna setups. The Time Mode features three different settings described in Table 3: Disabled, Survey-In and Fixed Mode. For optimal performance entering the position of the antenna (when known) is recommended as potential source of errors will be reduced.

Time Mode Settings	Description
Disabled	Standard PVT operation
Survey-In	The GPS receiver computes the average position over an extended time period until a predefined maximum standard deviation has been reached. Afterwards the receiver will be automatically set to Fixed Mode and the timing features will be activated.
Fixed Mode	In this mode, a fixed 3D position and known standard deviation is assumed and the timing features are activated. Fixed Mode can either be activated directly by sending pre-defined position coordinates (ICDF - Earth Center Earth Fixed format) or by performing a Survey-In. In Fixed mode, the timing errors in the "TIMEPULSE" signal which otherwise result from positioning errors are eliminated. Single-satellite operation is supported. For details, please refer to the u-blox 6 Receiver Description including Protocol Specification [2].

Table 3: Time mode settings

1.7.2 Timepulse and frequency reference

NEO-6T comes with a timepulse output which can be configured from 0.25 Hz up to 10 MHz. The timepulse can either be used for time synchronization (i.e. 1 pulse per second) or as a reference frequency in the MHz range. A timepulse in the MHz range provides excellent long-term frequency accuracy and stability.

1.7.3 Time mark

NEO-6T can be used for precise time measurements with sub-microsecond resolution using the external interrupt (EXTINT). Rising and falling edges of these signals are time-stamped to the GPS or UTC time and counted. The Time Mark functionality can be enabled with the UBX-CFG-TM2 message.

For details, please refer to the u-blox 6 Receiver Description including Protocol Specification [2].

1.8 Raw data

Raw data output is supported at an update rate of 5 Hz on the NEO-6T and NEO-6P. The UBX-RXM-RAW message includes carrier phase with half-cycle ambiguity resolved, code phase and Doppler measurements, which can be used in external applications that offer precision positioning, real-time kinematics (RTK) and attitude sensing.

1.9 Automotive Dead Reckoning

Automotive Dead Reckoning (ADR) is u-blox' industry proven off-the-shelf Dead Reckoning solution for tier-one automotive customers. u-blox' ADR solution combines GPS and sensor data using a tightly coupled Kalman filter. This improves position accuracy during periods of no or degraded GPS signal.

The NEO-6V provides ADR functionality over its software sensor interface. A variety of sensors (such as wheel ticks and gyroscope) are supported, with the sensor data received via UBX messages from the application processor. This allows for easy integration and a simple hardware interface, lowering costs. By using digital sensor data available on the vehicle bus, hardware costs are minimized since no extra sensors are required for Dead Reckoning functionality. ADR is designed for simple integration and easy configuration of different sensor options (e.g. with or without gyroscope) and vehicle variants, and is completely self-calibrating.

Anexo D

Datasheet de ESP32 Series

1. Overview

1. Overview

ESP32 is a single 2.4 GHz Wi-Fi-and-Bluetooth combo chip designed with the TSMC ultra-low-power 40 nm technology. It is designed to achieve the best power and RF performance, showing robustness, versatility and reliability in a wide variety of applications and power scenarios.

The ESP32 series of chips includes ESP32-D0WDG6, ESP32-D0WD, ESP32-D2WD, and ESP32-S0WD. For details on part numbers and ordering information, please refer to Part Number and Ordering Information.

1.1 Featured Solutions

1.1.1 Ultra-Low-Power Solution

ESP32 is designed for mobile, wearable electronics, and Internet-of-Things (IoT) applications. It features all the state-of-the-art characteristics of low-power chips, including fine-grained clock gating, multiple power modes, and dynamic power scaling. For instance, in a low-power IoT sensor hub application scenario, ESP32 is woken up periodically and only when a specified condition is detected. Low-duty cycle is used to minimize the amount of energy that the chip expends. The output of the power amplifier is also adjustable, thus contributing to an optimal trade-off between communication range, data rate and power consumption.

Note:

For more information, refer to Section 3.7 RTC and Low-Power Management.

1.1.2 Complete Integration Solution

ESP32 is a highly-integrated solution for Wi-Fi-and-Bluetooth IoT applications, with around 20 external components. ESP32 integrates an antenna switch, RF balun, power amplifier, low-noise receiver amplifier, filters, and power management modules. As such, the entire solution occupies minimal Printed Circuit Board (PCB) area.

ESP32 uses CMOS for single-chip fully-integrated radio and baseband, while also integrating advanced calibration circuitries that allow the solution to remove external circuit imperfections or adjust to changes in external conditions. As such, the mass production of ESP32 solutions does not require expensive and specialized Wi-Fi testing equipment.

1.2 Wi-Fi Key Features

- 802.11 b/g/n
- 802.11 n (2.4 GHz), up to 150 Mbps
- WMM
- TX/RX A-MPDU, RX A-MSDU
- Immediate Block ACK
- Defragmentation
- Automatic Beacon monitoring (hardware TSP)
- 4 × virtual Wi-Fi interfaces

Espressif Systems

1

ESP32 Datasheet V3.0

1. Overview

- Simultaneous support for Infrastructure Station, SoftAP, and Promiscuous modes
Note that when ESP32 is in Station mode, performing a scan, the SoftAP channel will be changed.
- Antenna diversity

Note:

For more information, please refer to Section 3.5 Wi-Fi.

1.3 BT Key Features

- Compliant with Bluetooth v4.2 BR/EDR and BLE specifications
- Class-1, class-2 and class-3 transmitter without external power amplifier
- Enhanced Power Control
- +12 dBm transmitting power
- NZF receiver with -97 dBm BLE sensitivity
- Adaptive Frequency Hopping (AFH)
- Standard HCI based on SDIO/SPI/UART
- High-speed UART HCI, up to 4 Mbps
- Bluetooth 4.2 BR/EDR BLE dual mode controller
- Synchronous Connection-Oriented/Extended (SCO/eSCO)
- CVSD and SBC for audio codec
- Bluetooth Piconet and Scatternet
- Multi-connections in Classic BT and BLE
- Simultaneous advertising and scanning

1.4 MCU and Advanced Features

1.4.1 CPU and Memory

- Xtensa® single-/dual-core 32-bit LX6 microprocessor(s), up to 600 MIPS (200 MIPS for ESP32-S0WD, 400 MIPS for ESP32-D2WD)
- 448 KB ROM
- 520 KB SRAM
- 16 KB SRAM in RTC
- QSPI supports multiple flash/SRAM chips

1. Overview

1.4.2 Clocks and Timers

- Internal 8 MHz oscillator with calibration
- Internal RC oscillator with calibration
- External 2 MHz ~ 60 MHz crystal oscillator (40 MHz only for Wi-Fi/BT functionality)
- External 32 kHz crystal oscillator for RTC with calibration
- Two timer groups, including 2 × 64-bit timers and 1 × main watchdog in each group
- One RTC timer
- RTC watchdog

1.4.3 Advanced Peripheral Interfaces

- 34 × programmable GPIOs
- 12-bit SAR ADC up to 18 channels
- 2 × 8-bit DAC
- 10 × touch sensors
- 4 × SPI
- 2 × I²S
- 2 × I²C
- 3 × UART
- 1 host (SD/eMMC/SDIO)
- 1 slave (SDIO/SPI)
- Ethernet MAC interface with dedicated DMA and IEEE 1588 support
- CAN 2.0
- IR (TX/RX)
- Motor PWM
- LED PWM up to 16 channels
- Hall sensor

1.4.4 Security

- Secure boot
- Flash encryption
- 1024-bit OTP, up to 768-bit for customers
- Cryptographic hardware acceleration:
 - AES
 - Hash (SHA-2)
 - RSA
 - ECC

Espressif Systems

3

ESP32 Datasheet V3

1. Overview

- Random Number Generator (RNG)

1.5 Applications (A Non-exhaustive List)

- Generic Low-power IoT Sensor Hub
- Generic Low-power IoT Data Loggers
- Cameras for Video Streaming
- Over-the-top (OTT) Devices
- Speech Recognition
- Image Recognition
- Mesh Network
- Home Automation
 - Light control
 - Smart plugs
 - Smart door locks
- Smart Building
 - Smart lighting
 - Energy monitoring
- Industrial Automation
 - Industrial wireless control
 - Industrial robotics
- Smart Agriculture
 - Smart greenhouses
 - Smart irrigation
- Agriculture robotics
- Audio Applications
 - Internet music players
 - Live streaming devices
 - Internet radio players
 - Audio headsets
- Health Care Applications
 - Health monitoring
 - Baby monitors
- Wi-Fi-enabled Toys
 - Remote control toys
 - Proximity sensing toys
 - Educational toys
- Wearable Electronics
 - Smart watches
 - Smart bracelets
- Retail & Catering Applications
 - POS machines
 - Service robots

Anexo E

Código del nodo coordinador

```

ESP32_version2.ino
1  /*****
2  * Include Libraries
3  *****/
4  #include "UbidotsEsp32Mqtt.h"
5
6  /*****
7  * Define Constants
8  *****/
9  const char *UBIDOTS_TOKEN = "BFF-XSgWqLmtRwDEcuuXEMLfpjzTtUAeT"; // Put here your Ubidots TOKEN
10 const char *WIFI_SSID = "chivos"; // Put here your Wi-Fi SSID
11 const char *WIFI_PASS = "12345678"; // Put here your Wi-Fi password
12 const char *DEVICE_LABEL = "coordinador"; // Put here your Device label to which data will be published
13 char *VARIABLE_LABEL = "position1"; // Put here your Variable label to which data will be published
14
15 const int PUBLISH_FREQUENCY = 10000; // Update rate in milliseconds
16
17 unsigned long timer;
18
19 Ubidots ubidots(UBIDOTS_TOKEN);
20 /*
21 mis variables
22 */
23
24
25 bool started= false;//True: Message is strated
26 bool ended = false;//True: Message is finished
27 char incomingByte ; //Variable to store the incoming byte
28 char msg[23]; //Message - array from 0 to 2 (3 values - PWM - e.g. 240)
29 byte mi_indice;
30
31 /*****
32 * Auxiliar Functions
33 *****/
34
35 void callback(char *topic, byte *payload, unsigned int length)
36 {
37   Serial.print("Message arrived [");
38   Serial.print(topic);
39   Serial.print("] ");
40   for (int i = 0; i < length; i++)
41   {
42     Serial.print((char)payload[i]);
43   }
44   Serial.println();
45 }
46
47 /*****
48 * Main Functions
49 *****/
50
51 void setup()
52 {
53   // put your setup code here, to run once:
54   Serial.begin(9600);
55   Serial2.begin(9600);
56   //ubidots.setDebug(true); // uncomment this to make debug messages available
57   ubidots.connectToWifi(WIFI_SSID, WIFI_PASS);
58   ubidots.setCallback(callback);
59   ubidots.setup();
60   ubidots.reconnect();
61   timer = millis();
62 }
63
64 void loop()
65 {
66
67
68   // put your main code here, to run repeatedly:
69
70
71   while (Serial2.available()>0){
72     //Read the incoming byte
73     incomingByte = Serial2.read();
74     //Serial.println(incomingByte);
75     //Start the message when the '<' symbol is received
76     if(incomingByte == '<')
77     {
78       started = true;
79       mi_indice = 0;
80       msg[mi_indice] = '\0'; // Throw away any incomplete packet
81     }
82     //End the message when the '>' symbol is received
83     else if(incomingByte == '>')
84     {

```



```

85     ended = true;
86     break; // Done reading - exit from while loop!
87 }
88 //Read the message!
89 else
90 {
91     if(mi_indice < 22) // Make sure there is room
92     {
93         if (isdigit(incomingByte) || incomingByte=='.' || incomingByte=='-')
94         {
95             msg[mi_indice] = incomingByte; // Add char to array
96             //Serial.println(msg);
97             mi_indice++;
98             msg[mi_indice] = '\0'; // Add NULL to end
99         }
100     }
101 }
102 }
103
104 if (!ubidots.connected())
105 {
106     | ubidots.reconnect();
107 }
108
109 if(started && ended && (millis() - timer) > PUBLISH_FREQUENCY)
110 {
111     if (mi_indice>20)
112     {
113         String mensaje="";
114         mensaje=msg;
115         char* str_lat = (char*)malloc(sizeof(char) * 10);
116         char* str_lng = (char*)malloc(sizeof(char) * 12);
117         char* context = (char*)malloc(sizeof(char) * 60);
118
119         sprintf(str_lat, "%s", mensaje.substring(2,11));
120         sprintf(str_lng, "%s", mensaje.substring(11,23));
121
122
123         int v_etiqueta=mensaje.substring(0,2).toInt();
124         if (v_etiqueta==50)
125         {
126             VARIABLE_LABEL="position1";
127         }
128         else if (v_etiqueta==60)
129         {
130             VARIABLE_LABEL="position2";
131         }
132         else if (v_etiqueta==70)
133         {
134             VARIABLE_LABEL="position3";
135         }
136
137         ubidots.addContext("lat",str_lat);
138         ubidots.addContext("lng",str_lng);
139         ubidots.getContext(context);
140         ubidots.add(VARIABLE_LABEL, v_etiqueta, context);
141
142         ubidots.publish(DEVICE_LABEL);
143         Serial.println("Dato enviado");
144         free(str_lat);
145         free(str_lng);
146         free(context);
147
148         timer=millis();
149         //mensaje="";
150         mi_indice = 0;
151         msg[mi_indice] = '\0';
152         started = false;
153         ended = false;
154     }
155 }
156
157 if (ubidots.connected())
158 {
159     | ubidots.loop();
160 }
161 }

```

Anexo F

Código del nodo supervisor 1

```

Supervisor1.ino
1  #include <SoftwareSerial.h>//incluimos SoftwareSerial
2  #include <TinyGPS.h>//incluimos TinyGPS
3
4  TinyGPS gps;//Declaramos el objeto gps
5  SoftwareSerial serialgps(4,3);//Declaramos el pin 4 Tx y 3 Rx
6
7  //Declaramos la variables para la obtención de datos
8  int identificador=50;
9  float latcero=0.0;
10 float lngcero=0.0;
11
12 void setup()
13 {
14   Serial.begin(9600);//Iniciamos el puerto serie
15   serialgps.begin(9600);//Iniciamos el puerto serie del gps
16   //delay(30000);
17   //Serial.println("E1 LISTO");
18 }
19
20 void loop()
21 {
22   while(serialgps.available())
23   {
24     int c = serialgps.read();
25
26     if(gps.encode(c))
27     {
28       float latitude, longitude;
29       gps.f_get_position(&latitude, &longitude);
30       Serial.print('<');
31       Serial.print(identificador);
32       Serial.print(latitude,6);
33       Serial.print(longitude,6);
34       Serial.println('>');
35       delay(1000);
36     }
37   }
38 }
39
40 }

```


Anexo G*Código del nodo supervisor 2*

```
Supervisor2.ino
1  #include <SoftwareSerial.h>//incluimos SoftwareSerial
2  #include <TinyGPSPlus.h>//incluimos TinyGPS
3
4  TinyGPSPlus gps;//Declaramos el objeto gps
5  SoftwareSerial serialgps(4, 3); //Declaramos el pin 4 Tx y 3 Rx
6
7  //Declaramos la variables para la obtención de datos
8  int identificador = 60;
9  float latcero = 0.0;
10 float lngcero = 0.0;
11
12 void setup()
13 {
14   Serial.begin(9600);//Iniciamos el puerto serie
15   serialgps.begin(9600);//Iniciamos el puerto serie del gps
16   //delay(30000);
17   //Serial.println("E2 LISTO");
18 }
19
20 void loop()
21 {
22   while (serialgps.available())
23   {
24     int c = serialgps.read();
25
26     if (gps.encode(c))
27     {
28       if (gps.location.isValid())
29       {
30         Serial.print('<');
31         Serial.print(identificador);
32         Serial.print(gps.location.lat(), 6);
33         Serial.print(gps.location.lng(), 6);
34         Serial.println('>');
35         delay(1000);
36       }
37     }
38   }
39 }
```


Anexo H

Código del nodo supervisor 3

```
Supervisor3.ino
1  #include <SoftwareSerial.h>//incluimos SoftwareSerial
2  #include <TinyGPSPlus.h>//incluimos TinyGPS
3
4  TinyGPSPlus gps;//Declaramos el objeto gps
5  SoftwareSerial serialgps(4, 3); //Declaramos el pin 4 Tx y 3 Rx
6
7  //Declaramos la variables para la obtención de datos
8  int identificador = 70;
9  float latcero = 0.0;
10 float lngcero = 0.0;
11
12 void setup()
13 {
14   Serial.begin(9600);//Iniciamos el puerto serie
15   serialgps.begin(9600);//Iniciamos el puerto serie del gps
16   //delay(30000);
17   //Serial.println("E3 LISTO");
18 }
19
20 void loop()
21 {
22   while (serialgps.available())
23   {
24     int c = serialgps.read();
25
26     if (gps.encode(c))
27     {
28       if (gps.location.isValid())
29       {
30         Serial.print('<');
31         Serial.print(identificador);
32         Serial.print(gps.location.lat(), 6);
33         Serial.print(gps.location.lng(), 6);
34         Serial.println('>');
35         delay(1000);
36       }
37     }
38   }
39 }
```

Anexo I

Certificado de análisis antiplagio



CERTIFICADO DE ANÁLISIS
original

PROPUESTA TECNOLÓGICA-PEREZ CRISTIAN_VILLON VICTOR

4% Similitudes

3% Texto entre comillas

0% Similitudes entre comillas

1% Tablas no reconocidas

Nombre del documento: PROPUESTA TECNOLÓGICA PEREZ CRISTIAN_VILLON VICTOR.docx

ID del documento: 8f77c028ba6c788029f1270a0915a01810

Tamaño del documento original: 8,77 Mb

Depositar: JIMENA PAREJO LUIS MIGUEL

Fecha de depósito: 2023/02/03


Tipo de carga: VicioFicha

Fecha de fin de análisis: 2023/02/03





Número de palabras: 20.708

Número de caracteres: 133.703





Ubicación de las similitudes en el documento:









Fuentes principales detectadas



Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	 hull.hull.ac.uk Creación de un efecto apantallador de los LEDs frente a la ex... http://hull.hull.ac.uk/	< 1%		Palabras idénticas = 116 (10 palabras)
2	 repositorio.unillanos.edu.co https://repositorio.unillanos.edu.co/handle/2018-1/4496 Indicador de Salud PRODIGO BA...	< 1%		Palabras idénticas = 116 (10 palabras)
3	 repositorio.unillanos.edu.co El juego simbólico y su influencia en el desarrollo socio-af... https://repositorio.unillanos.edu.co/handle/2018-1/4496 114.PDF 131.2023.0014.pdf	< 1%		Palabras idénticas = 116 (10 palabras)
4	 repositorio.unillanos.edu.co Implementación de un prototipo tecnológico híbrido para ... https://repositorio.unillanos.edu.co/handle/2018-1/4496 114.PDF 131.2023.0014.pdf	< 1%		Palabras idénticas = 116 (10 palabras)
5	 repositorio.unillanos.edu.co Implementación de un prototipo electrónico automatizado... https://repositorio.unillanos.edu.co/handle/2018-1/4496 114.PDF 131.2023.0014.pdf	< 1%		Palabras idénticas = 116 (10 palabras)

Fuentes con similitudes fortuitas






Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	 analisis.fisica.unam.mx https://analisis.fisica.unam.mx/ Analisis de introducción a la configuración PAPM.pdf	< 1%		Palabras idénticas = 116 (10 palabras)
2	 Documento de otro usuario ver más El documento proviene de otro grupo	< 1%		Palabras idénticas = 116 (10 palabras)
3	 mail.profesor@ceadefiticas.org Nota sobre observaciones hechas a rebufo de C... https://mail.profesor@ceadefiticas.org/ index.php?agencia=arbitraje&id=201802	< 1%		Palabras idénticas = 116 (10 palabras)
4	 Documento de otro usuario ver más El documento proviene de otro grupo	< 1%		Palabras idénticas = 116 (10 palabras)
5	 repositorio.unillanos.edu.co Análisis y estudio de impacto de la tecnología ZIGBEE aplicada a... https://repositorio.unillanos.edu.co/handle/2018-1/4496 114.PDF 131.2023.0014.pdf	< 1%		Palabras idénticas = 116 (10 palabras)

Fuentes ignoradas Estas fuentes han sido ignoradas del cálculo del porcentaje de similitud por el propietario del documento.

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	 Documento de otro usuario ver más El documento proviene de otro grupo	7%		Palabras idénticas = 76 (7.600 palabras)
2	 Documento de otro usuario ver más El documento proviene de otro grupo	7%		Palabras idénticas = 76 (7.600 palabras)
3	 Documento de otro usuario ver más El documento proviene de otro grupo	3%		Palabras idénticas = 36.308 palabras)
4	 Documento de otro usuario ver más El documento proviene de otro grupo	3%		Palabras idénticas = 36.308 palabras)
5	 www.academyofart.com https://www.academyofart.com/ art/series/2011.pdf	1%		Palabras idénticas = 16.208 palabras)
6	 Documento de otro usuario ver más El documento proviene de otro grupo	< 1%		Palabras idénticas = 116 (10 palabras)
7	 TikTok Creación de una empresa de formación y capacitación para jóvenes en... https://www.tiktok.com/@tiktok o-son-empresas-formacion-empresarios-jovenes-empresario...	< 1%		Palabras idénticas = 116 (10 palabras)

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
8	 Documento de otro usuario ver más El documento proviene de otro grupo	< 1%		Palabras idénticas = 116 (10 palabras)
9	 Documento de otro usuario ver más El documento proviene de otro grupo	< 1%		Palabras idénticas = 116 (10 palabras)

Fuentes mencionadas (sin similitudes detectadas) Estas fuentes han sido citadas en el documento sin encontrar similitudes.

-  <https://www.adfruit.com/category/105>
-  <https://www.electronic.com/producto/bee-2m-wire-antenna/>
-  <https://www.acompra.com/cuador/item/energia-electronica-panel-solar-celda-5v-180ma-107-42-5235-637/>
-  <https://www.bee.com/bee-serie-2-configuracion/>
-  <https://doi.org/10.29019/enfoque.v10n1.552>

Firmado digitalmente por Miguel Amaya



Anexo J

Código de la aplicación móvil localdots

```

1  import 'dart:async';
2
3  import 'package:flutter/material.dart';
4  import 'package:google_maps_flutter/google_maps_flutter.dart';
5  import 'package:localdots/models/response_coord.dart';
6  import 'package:localdots/provider/variable_provider.dart';
7  class MapPage extends StatefulWidget {
8    const MapPage({Key? key}) : super(key: key);
9
10
11  @override
12  State<MapPage> createState() => _MapPageState();
13
14  CameraPosition _initialPosition=CameraPosition(target: Latlng(-2.2261335, -80.9505463 ), zoom: 14);
15  Completer<GoogleMapController> _mapController=Completer();
16
17
18  void _onMapCreated(GoogleMapController controller)
19  {
20    _mapController.complete(controller);
21  }
22
23  class _MapPageState extends State<MapPage> {
24    VariableProvider _variableProvider=new VariableProvider();
25    ResponseCoord? _responseCoord;
26    ResponseCoord? _supervisor2;
27    ResponseCoord? _supervisor3;
28
29    future: _markersFuture,
30    builder: (context, AsyncSnapshot<Set<Marker>> snapshot){
31      if (snapshot.hasData)
32      {
33        return GoogleMap(
34          myLocationEnabled: true,
35          markers: snapshot.data!,
36          mapType: MapType.normal,
37          onCameraMove: (position){
38            _initialPosition=position;
39          },
40          onMapCreated: _onMapCreated,
41          initialCameraPosition: _initialPosition,
42        ); // GoogleMap
43      }else if (snapshot.hasError){
44        return Center(child: Text("Error al cargar los marcadores"),);
45      }
46      else{
47        return Center(child: CircularProgressIndicator(),);
48      }
49    },
50  ) // FutureBuilder
51  ],
52  ), // Stack
53  ); // Scaffold
54  }
55  }

```

REFERENCIAS BIBLIOGRÁFICAS

- adafruit. (2018). *Raspberry Pi*. <https://www.adafruit.com/category/105>
- Adeva, R. (2021, April 14). *Tecnologías inalámbricas: diferencias y usos de WiFi, Bluetooth, Zigbee y Z-Wave*. ADSL ZONE.
- Android Studio. (2014, December 11). *Android Studio v1.0: características y comparativa con Eclipse*. Academia Android.
- Ardila, J., Rodríguez, N., & Gil, F. (2004). Población y muestreo. In *Epidemiología clínica: investigación clínica* (Vol. 09, Issue 08, pp. 129–139). Carrera de Ciencias de la Comunicación Social -UCB.
- AV Electronics. (n.d.). *Xbee 2mw Wire Antenna serie S2C*. AV Electronics. Retrieved February 17, 2023, from <https://avelectronics.cc/producto/xbee-2mw-wire-antenna/>
- Avila, L., & Reyes, C. (2017). *Revisión estado del Arte de la tecnología Bluetooth*. <https://core.ac.uk/download/pdf/267928921.pdf>
- DIGI. (n.d.). *Configuración XBEE Serie 2*. DIGI. Retrieved February 17, 2023, from <https://xbee.cl/xbee-serie-2-configuracion/>
- Ducoing Watty, A. (2016). Introducción a La Caprino Cultura. *XVI Semana de La Capricultura, 0*, 1–8.
- Egas, C. R., Viracocha, D., & Rivera, J. (2019). Implementación de una red inalámbrica de sensores para la gestión de luminarias utilizando IPv6. *Enfoque UTE, 10*(4), 45–56. <https://doi.org/10.29019/enfoque.v10n4.553>

Electronilab. (n.d.). *Módulo GPS Ublox NEO-6M v2 con memoria EEPROM*.

Electronilab. Retrieved February 19, 2023, from

<https://electronilab.co/tienda/modulo-gps-ublox-neo-6m-v2-con-memoria-eeeprom/>

Esteban. (2020, January 17). *Android Studio: esto es lo que puedes hacer con este programa*.

Faludi, R. (2021, June 29). *Guía de compra de XBee*. DIGI.

Fernández Lobo, A. E. (2021). *Detección de Patrones en Datos Espaciales para Aplicaciones con Geolocalización en Smart Cities*.

Fernández, Y. (2019, January 9). *Conectividad ZigBee de Amazon Echo Plus*. Xataka.

Fombona, J., & Del Valle Mejias, M. (2018). Análisi de la geolocalización y realidad en dispositivos móviles, propuestas socioeducativas relacionadas con el entorno y las salidas de campo. *Profesorado*, 22(4), 197–222.
<https://doi.org/10.30827/profesorado.v22i4.8413>

Gascón, J. (2019). DISEÑO DE UNA PLACA ELECTRÓNICA MODULAR PARA EL MICROCONTROLADOR NODEMCU V3. *Universidad Politécnica de Valencia*.

González, E. (2015). Red de Sensores y el Internet de las cosas. *Universidad Politécnica de Valencia*.

Grupo ElectroStore. (n.d.). *MÓDULO ESP32 ESP-32 WIFI BLUETOOTH*.

Grupo Velasco. (2018). *ARDUINO UNO R3 - NETO*.

[http://www.velasco.com.ec/velasco/producto.php?id=3862:](http://www.velasco.com.ec/velasco/producto.php?id=3862)

Helguero, S. P., & Correa, J. (2005). Pastoreo caprino en el monte formoseño (Argentina). *REDVET. Revista Electrónica de Veterinaria*, VI(11), 1–14.

Hernández Acosta, I. (2014). Nota sobre observaciones hechas a rebaños de cabras criollas ramoneando un bosque seco tropical. *Revista de La Facultad de Agronomía*, 7(3), 180–184.

Huerta, E., Mangiaterra, A., & Noguera, G. (2005). GPS Posicionamiento Satelital. In *UNR*.

Jornet Monteverde, J. A., Galiana Merino, J. J., & Soler Llorens, J. L. (2021).

Design and implementation of a wireless sensor network for seismic monitoring of buildings. *Sensors*, 21(11). <https://doi.org/10.3390/s21113875>

Llarío, F., Sendra, S., Parra, L., & Lloret, J. (2013). Detection and protection of the attacks to the sheep and goats using an intelligent wireless sensor network. *2013 IEEE International Conference on Communications Workshops, ICC 2013, June*, 1015–1019. <https://doi.org/10.1109/ICCW.2013.6649385>

Lomillos, A., Varga, A. de la, Garcia, J., & Gaudioso, L. (2017). Monitoring lidia cattle and spatial distribution. *Veterinaria Mexico*, 17(4), 1–17.

<https://doi.org/10.21753/vmoa.4.4.405>.http

Mannion, P. (2017, October 26). *Comparación de tecnologías inalámbricas de baja potencia*. Digi-Key.

- Martinez, J. (2013). Fundamentos de programación en Java. In *G-TeC* (Vol. 8, Issue 2).
- Mueller, J. P., Taddeo, H. R., Abad, M. I., & Debenedetti, S. (2018). Revision sobre el origen y el desarrollo de la producción de caprinos de Angora en Argentina. *Revista de Investigaciones Agropecuarias*, 44(3), 286–300.
- Ortega, J. G. (2017). Cómo se genera una investigación científica que luego sea motivo de publicación. *Journal of the Selva Andina Research Society*, 8(2), 145–146.
- Pascual, C. M. (2012). Bluetooth : criterios de selección y comparativa con otras tecnologías inalámbricas. *Revista Técnica Industrial*, 76–80.
- Peña, C. (2020). *Arduino IDE Domina la programación y controla la placa.* RedUsers.
- Pérez Vera, M. G., Ocampo Botello, F., & Sánchez Pérez, K. R. (2015). Aplicación de la metodología de la investigación para identificar las emociones / Application of research methodology to identify emotions. *RIDE Revista Iberoamericana Para La Investigación y El Desarrollo Educativo*, 6(11), 796. <https://doi.org/10.23913/ride.v6i11.130>
- Piedra, A. (2022). Estudio bibliográfico del uso de vitaminas A, C, E y su influencia en el Comportamiento Sexual en especies menores productivas (ovinos, caprinos y conejos). *UNIVERSIDAD TÉCNICA DE BABAHOYO*, 36.

Ródenas, M. (2012). Diseño e implementación de servicios web multiplataforma utilizando el framework .NET ". *UNIVERSIDAD POLITÉCNICA DE CARTAGENA*.

Rodríguez, A., & Rodríguez, H. (2017). Comportamiento de las cabras en pastoreo. *Uprm, 1*(1), 1–9.

<https://www.uprm.edu/cms/index.php?a=file&fid=16083#:~:text=Comportamiento Ingestivo,-Los cabros se&text=Bajo condiciones naturales no destruyen,poca variedad de fuentes forrajeras>.

Rosendo, V., Cárdenas, M., Serrano, R., Beatriz, K., & Sánchez, O. (2016). *Revista Aristas: Ciencia Básica y Aplicada* (Vol. 5, Issue 9).

<http://fcqi.tij.uabc.mx/usuarios/revistaaristas/>

Salazar, J. (2012). Redes Inalámbricas. In *Articulo* (Vol. 2).

<https://upcommons.upc.edu/handle/2117/100918>

Santaella, J. (2022, April 26). *¿Qué es Android Studio?* TALENTLY BLOG.

SDR. (2022). *Módulo de Carga con Circuito de Protección – TP4056*.

SanDoRobotics. [https://sandorobotics.com/producto/hr0140-](https://sandorobotics.com/producto/hr0140-1/#:~:text=Corriente%20de%20carga%20máxima%3A%201000,de%20la%20batería%3A%203%20A)

[1/#:~:text=Corriente%20de%20carga%20máxima%3A%201000,de%20la%20batería%3A%203%20A](https://sandorobotics.com/producto/hr0140-1/#:~:text=Corriente%20de%20carga%20máxima%3A%201000,de%20la%20batería%3A%203%20A)

Sikora, A., & Groza, V. F. (2005). Coexistence of IEEE802.15.4 with other

Systems in the 2.4 GHz-ISM-Band. *Techology Conference*, 1786–1791.

SOLID STRTE SOPPLIES. (n.d.). *Digi Embedded – XBee Zigbee S2B Series*.

SOLID STRTE SOPPLIES. Retrieved February 17, 2023, from

<https://www.sssltd.com/product/digi-embedded-xbee-zigbee-s2b-series/>

Solíz, P., Fuentes, E., & Rodríguez, K. (2014). Manejo Reproductivo De La Cabra

Características Particulares De Las Cabras. *Entorno Ganadero*, 37(844), 6.

Techmake. (2023). *XBee Shield para Arduino*. Techmake.

<https://techmake.com/products/wrlxbe00422>

TET TSA. (n.d.). *TH 18650 5000mAh 3.7V UltraFire BLISTER 2 UNIDADES*

AZUL. Tecnología T y T Electrónicos. S.A. Retrieved February 18, 2023,

from <https://tienda.tettsa.gt/producto/th-18650-5000mah-3-7v-ultrafire-blister-2-unidades-azul/>

Ubidots Community. (2018, April). *Ubidots Dashboards require refresh to see the*

latest and event not function. <https://ubidots.com/community/t/solved-ubidots-dashboards-require-refresh-to-see-the-latest-and-event-not-function/1550>

Vargas Cordero, Z. R. (2009). La Investigación aplicada: Una forma de conocer las

realidades con evidencia científica. *Revista Educación*, 33(1), 155.

<https://doi.org/10.15517/revedu.v33i1.538>

Yu, S., & Park, Y. (2020). SLUA-WSN: Secure and Lightweight Three-Factor-

Based User Authentication Protocol for Wireless Sensor Networks. *Sensors*, 20(15), 4143. <https://doi.org/10.3390/s20154143>

Zennaro, M. (2010). Introducción a las Redes de Sensores Inalámbricos. *ICTP Trieste-Italy*, 56. <https://docplayer.es/8261980-Introduction-to-wireless-sensor-networks-marco-zennaro-ictp-trieste-italy.html>