



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

**FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES**

TEMA:

**Diseño y simulación de un sistema IOT de parqueo inteligente utilizando
LORAWAN y ESP32 en el campus de la UCSG.**

AUTOR:

Cali Larreategui, Hector Rolando

**Trabajo de Integración Curricular previo a la obtención del título de
INGENIERO EN TELECOMUNICACIONES**

TUTOR:

Ing. Ubilla González, Ricardo Xavier MsC

Guayaquil, Ecuador

02 de septiembre del 2025



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

Certificamos que el presente Trabajo de Integración Curricular fue realizado en su totalidad por **Cali Larreategui, Hector Rolando**, como requerimiento para la obtención del Título de **Ingeniero en Telecomunicaciones**.

TUTOR

Ing. Ubilla González, Ricardo Xavier MsC

DIRECTOR DE CARRERA

Ing. Bohórquez Escobar, Celso Bayardo. PhD.

Guayaquil, 02 de septiembre del 2025



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, Cali Larreategui, Hector Rolando

DECLARO QUE:

La presente **Tesis de Grado**, titulada **Diseño y Simulación de un Sistema IoT de Parqueo Inteligente utilizando LoRaWAN y ESP32 en el Campus de la UCSG**, y presentada como requisito para la obtención del título de **Ingeniero en Telecomunicaciones**, es un trabajo original de mi autoría. Durante su elaboración, se han respetado los derechos intelectuales de terceros, citando y referenciando debidamente todas las fuentes consultadas, las cuales constan en la bibliografía.

En virtud de esta declaración, asumo total responsabilidad por el contenido, la veracidad y el alcance de la investigación aquí presentada.

Guayaquil, 02 de septiembre del 2025

EL AUTOR

Cali Larreategui, Hector Rolando



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

AUTORIZACIÓN

Yo, Cali Larreategui, Hector Rolando

Autorizo a la **Universidad Católica de Santiago de Guayaquil** a publicar en la biblioteca de la institución mi **Tesis de Grado**, titulada: **Diseño y Simulación de un Sistema IoT de Parqueo Inteligente utilizando LoRaWAN y ESP32 en el Campus de la UCSG.**

Asimismo, declaro que el contenido, las ideas y los criterios expuestos en dicho trabajo son de mi total autoría y exclusiva responsabilidad.

Guayaquil, 02 de septiembre del 2025

EL AUTOR

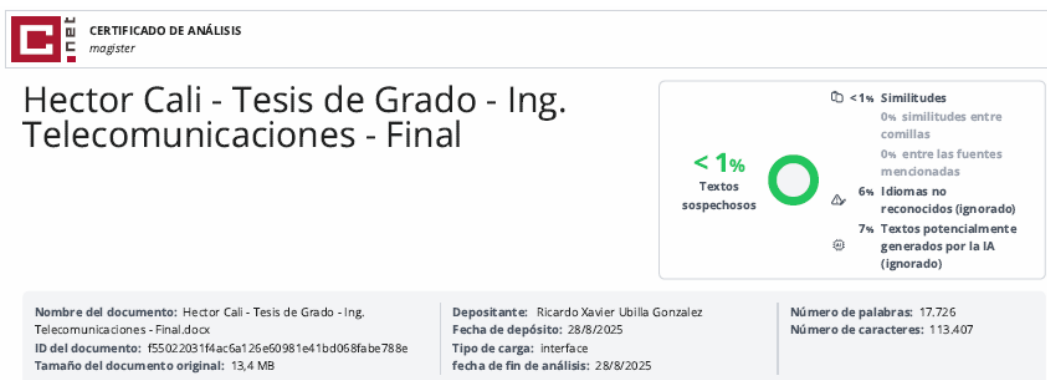
Cali Larreategui, Hector Rolando



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

REPORTE DE COMPILATIO



Reporte COMPILATIO del estudiante **Cali Larreategui, Hector Rolando** de la carrera de **INGENIERÍA TELECOMUNICACIONES** del Sistema de Titulación con el tema “**Diseño Y Simulación De Un Sistema Iot De Parqueo Inteligente Utilizando LoraWan Y ESP32 En El Campus De La UCSG**”, mismos que se encuentra en menos del 1% de coincidencias.

Ing. Ubilla González, Ricardo Xavier MsC

DOCENTE-TUTOR

AGRADECIMIENTOS

Quisiera expresar mi más sincero agradecimiento a todas las personas e instituciones que hicieron posible la realización de esta tesis de maestría.

En primer lugar, mi profunda gratitud al **Ing. Ricardo Xavier Ubilla González**, mi tutor, por su invaluable guía, su paciencia y su constante apoyo a lo largo de este proceso. Su conocimiento y sus consejos fueron fundamentales para la conceptualización, el desarrollo y la conclusión de esta investigación.

A la **Universidad Católica de Santiago de Guayaquil**, y en particular al **Sistema de Titulación**, por brindarme la oportunidad de formarme en la Carrera en Ingeniería Telecomunicaciones y por proporcionar el entorno académico necesario para llevar a cabo este proyecto.

A mis compañeros de clase por el apoyo y el intercambio de ideas.

Finalmente, y de manera muy especial, a mi familia y amigos por su amor incondicional, su motivación y por ser mi principal fuente de inspiración. Sin su apoyo, este logro no hubiera sido posible.

DEDICATORIA

A mis amados padres, **Hector** y **Olga**, quienes han sido el pilar fundamental de mi vida. Su sacrificio, sus palabras de aliento y su fe inquebrantable en mí me dieron la fuerza para superar cada desafío. Les debo todo.

A **Cynthia**, por toda la inspiración y fuerza que me has dado a lo largo de los años aun cuando no eras consciente del gran significado que has tenido en mi vida.

A mis amigos, por su alegría y por mantenerme con los pies en la tierra.

Y a mi tutor, **Ing. Ricardo Ubilla**, por su invaluable paciencia y, sobre todo, por su gran apoyo y guía a lo largo de este proyecto. Su confianza fue clave para llegar a la meta.



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

**FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES**

TRIBUNAL DE SUSTENTACIÓN

f. _____

Ing. Bohórquez Escobar, Celso Bayardo. PhD.

DIRECTOR DE CARRERA

f. _____

Ing. Ricardo Xavier Ubilla González, MsC.

COORDINADOR DEL ÁREA

f. _____

Ing. Bohórquez Heras Daniel Bayardo, MsC

OPONENTE

ÍNDICE GENERAL

RESUMEN.....	XIII
ABSTRACT	XIV
Capítulo 1: Descripción General del Trabajo de Titulación	2
1.1. Introducción	2
1.2. Antecedentes y Estado del Arte	3
1.2.1. Proyectos de Monitoreo de Parqueo con Sensores Basados en Presencia	7
1.2.2. Proyectos de Monitoreo de Parqueo con Visión por Computadora	8
1.2.3. Aplicaciones de Redes LPWAN en Entornos Universitarios	9
1.3. Definición del Problema	10
1.4.1. Contribución a las Tecnologías de Ciudades Inteligentes y el "Smart Campus"	13
1.4.2. Aporte a la Línea de Investigación.....	14
1.4.3. Viabilidad Técnica y Económica del Diseño (Análisis Preliminar)	14
1.4.4. Potencial de Innovación y Futuras Investigaciones	15
1.5. Objetivos	16
1.5.1. Objetivo General	16
1.5.2. Objetivos Específicos.....	16
1.6. Alcance y Limitaciones del Proyecto.....	17
1.6.1. Exclusiones Detalladas	18
1.7. Hipótesis.....	20
1.8. Metodología de Investigación	20
1.8.1. Detalle del Proceso de Diseño y Simulación	20
Capítulo 2: Fundamentación Teórica y Marco Teórico	24
2.1. Arquitectura de Referencia.	24
2.1.1. Protocolos de Comunicación del IoT.....	25
2.2. Descripción General de la Tecnología LoRaWAN.....	27
2.2.1. Arquitectura de LoRaWAN	27
2.2.1.1. Fundamentos Técnicos de la Modulación LoRa	29
2.2.1.2. Clases de Dispositivos LoRaWAN	29
2.2.1.3. Mecanismos de Seguridad en LoRaWAN	30
2.2.2. Comparación con Otras Tecnologías LPWAN	31
2.3. Tecnologías de Detección de Ocupación	31
2.3.1. Tipos de Sensores para Parqueo Inteligente: Una Comparación Detallada..	32
2.3.2. Principio de Funcionamiento del Sensor Ultrasónico - HC-SR04.....	32
2.4. Componentes Clave del Sistema.....	34
2.4.1. Características Detalladas del ESP32.....	34
2.4.2. Módulo de Radiofrecuencia LoRa	36
2.5. Descripciones Técnicas del ESP32	37
2.5.1. Funcionalidades Detalladas de Ubidots	38
2.5.2. Alternativas a Ubidots.....	39
Capítulo 3: Diseño y Desarrollo.....	40
3.1. Análisis de los Componentes.	40
3.1.1. Arquitectura en Bloques.....	40
3.1.2. Detalle del Nodo Sensor	41

3.1.3. Detalle de la Infraestructura de Red LoRaWAN	43
3.1.4. Diseño de la Base de Datos y la Interfaz de Usuario	43
3.2. Lógica de Programación y Flujo de Datos	46
3.2.1. Lógica del Firmware del Nodo Sensor.....	46
3.2.2. Proceso de Activación del Dispositivo LoRaWAN.....	47
3.2.3. Lógica del Decodificador de Payload.	48
3.2. Flujo de Datos a Través de la Red LoRaWAN y Plataforma IoT.....	50
3.3. Interconexiones entre los Dispositivos y Elementos del Sistema IoT	51
3.3.1. Interconexión en el Nodo Sensor (ESP32 + Sensor Ultrasónico + Módulo LoRa).....	51
3.3.2. Interconexión Gateway LoRaWAN y Servidor de Red.....	52
3.3.3. Interconexión Servidor de Red LoRaWAN y Plataforma IoT.....	53
3.3. Simulación del Sistema	53
3.3.1. Herramientas de Simulación Utilizadas	53
3.3.2. Modelo Matemático y Parametrización de la Simulación	54
3.3.3. Diseño del Escenario de Simulación.....	55
3.4. Programación del Sistema.....	55
3.4.1. Lógica del Nodo Sensor en NS-3.....	55
3.4.2. Configuración del Gateway y Servidor de Red en NS-3	56
3.5. Configuración de la Plataforma IoT (ej: Ubidots) para la Recepción de Datos...	56
3.6. Simulación del Sistema	57
3.6.1. Software y Herramientas de Simulación.....	57
3.6.2. Diseño del Escenario de Simulación.....	57
3.6.3. Ejecución de la Simulación y Recopilación de Datos.....	59
3.6.4. Recopilación de Métricas	59
Capítulo 4: Pruebas y Resultados.....	61
4.1. Metodología de Pruebas	61
4.1.1. Metodología de Pruebas y Escenarios de Simulación.....	61
4.1.2. Escenarios de Prueba Simulados.....	65
4.2. Presentación y Análisis de Resultados	66
4.2.1. Resultados del Escenario 1: Tráfico Normal (100 Nodos)	67
4.2.2. Resultados del Escenario 2: Hora Pico (200 Nodos)	69
4.2.3. Resultados del Escenario 3: Capacidad Máxima (400 Nodos)	71
4.2.4. Capacidad y Escalabilidad de la Red	72
4.2.5. Análisis de Escenarios Adicionales	74
4.3 Análisis de Resultados	78
4.3.1. Precisión de Detección del Sensor	78
4.3.2. Rendimiento de la Red LoRaWAN (PDR y Latencia)	78
4.3.3. Consumo Energético del Nodo (Simulado)	80
4.3.4. Capacidad de la Red (Número Máximo de Nodos)	80
4.4. Discusión de Resultados (Versión Mejorada).....	81
Capítulo 5: Conclusiones y Recomendaciones	84
5.1. Conclusiones	84
5.2. Recomendaciones.....	85
Anexos	90

ÍNDICE DE TABLAS

Tabla 1 - Estimación de la Vida Útil de la Batería del Nodo Sensor	73
Tabla 2 - Impacto de la Distancia en el PDR (100 Nodos, SF9)	74
Tabla 3 - Impacto del Factor de Dispersión (SF) en PDR y Latencia (a 1.5km)	75
Tabla 4 - Mejora del PDR con Múltiples Gateways (Escenario 2: 200 Nodos)	76
Tabla 5 - Mejora del PDR con Múltiples Gateways (Escenario 3: 400 Nodos)	77
Tabla 6 - Umbral de Capacidad del Gateway LoRaWAN por Número de Nodos	78
Tabla 7 - Resumen de Precisión de Detección del Sensor Ultrasónico	79
Tabla 8 - Rendimiento de la Red LoRaWAN por Número de Nodos Simulados.....	80
Tabla 9 - Rendimiento de la Red LoRaWAN por Número de Nodos Simulados.....	81

ÍNDICE DE FIGURAS

Figura 1 - Gráfico de Bloques de la Arquitectura Lora.	28
Figura 2 - Microcontrolador ESP32.....	34
Figura 3 - Red de Estacionamiento Inteligente con LoraWan	41
Figura 4 - Visual del Dashboard de Monitoreo.....	45
Figura 5 - Arquitectura de Sistema de Parqueo Inteligente con IoT.....	60
Figura 6 - Propagación de Señal LoRaWAN	62
Figura 7 - Diagrama de Bloques de la Lógica del Nodo Sensor	66
Figura 8 - Diagrama de Bloques del Sistema a Gran Escala. (100 Nodos).	68
Figura 9 - Comportamiento Agregado de la Red de 100 Nodos.....	68
Figura 10 - Diagrama de Bloques del Sistema a Gran Escala. (200 Nodos).	70
Figura 11 - Topología de Red Simulada (Escenario2: 200 Nodos)	70
Figura 12 - Diagrama de Bloques del Sistema a Gran Escala. (400 Nodos).	71
Figura 13 - Topología de Red Simulada (Escenario3: 400 Nodos)	72
Figura 14 - Diagrama de Bloques de la Lógica del Nodo Sensor (Escenario 1)	90
Figura 15 - Gráfico del Comportamiento de la Señal del Nodo Sensor (Escenario 1) ...	91
Figura 16 - Diagrama de Bloques de la Lógica del Nodo Sensor (Escenario 2)	92
Figura 17 - Gráfico del Comportamiento de la Señal del Nodo Sensor (Escenario 2) ...	93
Figura 18 - Diagrama de Bloques de la Lógica del Nodo Sensor (Escenario 3)	94
Figura 19 - Gráfico del Comportamiento de la Señal del Nodo Sensor (Escenario 3) ...	95

RESUMEN

Esta tesis se centra en el diseño y la simulación de un sistema inteligente de monitoreo de estacionamientos, con el objetivo de optimizar la gestión del mismo en el campus de la Universidad Católica de Santiago de Guayaquil (UCSG). Ante el problema de la congestión vehicular y la búsqueda ineficiente de plazas disponibles, se propuso una solución que utiliza el Internet de las Cosas (IoT) y la tecnología de comunicación LoRaWAN de largo alcance y bajo consumo. La metodología que fue utilizada está basada en un esquema que esta compuesto por nodos sensores, que están equipados con microcontroladores, en una red LoraWAN y una plataforma IoT para poder ver los datos. También se usó el simulador de red NS-3 para ver cómo se comporta la red según el escenario que se simule. Esto nos permitió validar que tan efectivo era el sistema, mostrando un deterioro en la entrega de paquetes (PDR) mientras más aumenta la carga de la red y mostrando de esta forma que la batería tendría una vida útil que puede superar los 9 años. El estudio concluye que la solución propuesta es una herramienta escalable y eficiente para reducir la congestión y mejorar la experiencia del usuario en el campus.

Palabras clave: IoT, LoRaWAN, Parqueo inteligente, ESP32, Simulación.

ABSTRACT

This thesis is based on the design and simulation of an intelligent parking monitoring system that aims to optimize parking management on the campus of the Catholic University of Santiago de Guayaquil (UCSG). Due to this problem of traffic congestion and the inefficient search for available parking spaces, it is proposed to solve it using the Internet of Things (IoT) and the long-range, low-power LoRaWAN communication technology. The methodology employed in this project is based on a system composed of sensor nodes equipped with microcontrollers in a LoRaWAN network and an IoT platform for data visualization. The NS-3 network simulator was used to observe network behavior based on the simulated scenario. This validated the system's effectiveness, showing a deterioration in packet delivery rates (PDR) as network load increases and demonstrating that the battery life could exceed 9 years. This study concluded that the scalable and efficient congestion reduction tool proposed in this project improves the user experience on campus.

Keywords: IoT, LoRaWAN, Smart Parking, ESP32, Simulation.

Capítulo 1: Descripción General del Trabajo de Titulación

1.1. Introducción

La creciente urbanización y el aumento constante del parque automovilístico mundial han transformado la gestión del tráfico y la búsqueda de plazas de aparcamiento en retos críticos para las ciudades. Este problema se agrava en ambientes densamente poblados y en instituciones con alto tráfico vehicular, como las universidades, donde la congestión y el uso ineficiente de los espacios disponibles generan impactos negativos tanto para los usuarios como para la administración.

En este caso, el problema de las Smart Cities aparece como una solución que promueve que las tecnologías avanzadas sean integradas para mejorar el funcionamiento de infraestructura y así tener una mejor calidad de vida urbana. (ONU-Hábitat, 2020).

En este caso IoT tiene un papel importante siendo clave para las ciudades inteligentes, ya que interactúa con dispositivos físicos, vehículos, edificios y elementos del día a día, dado así la capacidad de recopilar datos y también de intercambiarlos

De esta forma puede tomar decisiones en tiempo real, lo que es importante para la automatización de procesos (Atzori et al., 2021). Estas aplicaciones que requieren tener una conexión de largo alcance y que no consuman mucha energía, encuentran en LoraWAN una solución perfecta.

El que puedan enviar datos en cantidades pequeñas y a grandes distancias teniendo un consumo de energía bajo, las convierte en especialmente adecuadas para que puedan supervisar las gestiones ambientales de mayor medida (Adnan et al., 2022).

En la UCSG, diariamente llegan un número de estudiantes bastante grande y también de profesores. Actualmente la forma en que se manejan los parqueos no da ninguna información sobre si están ocupados o no. Por eso la gente demora en parquearse y se generan tráfico interno, atrasos o largas filas de carros.

Para solucionar este inconveniente se realizó un diseño y una simulación que muestra un sistema inteligente para monitorear los sitios de parqueo en la UCSG. Usando todo el sistema de LoraWAN

Lo que se busca es dar de forma tecnológica una manera para saber si las hay lugares de parque disponibles o si todos están ocupados en tiempo real. Ayudando con esto a evitar filas de carros o tráfico interno.

Este documento está organizado en cinco capítulos. El capítulo 1 abarca los antecedentes, el problema, la justificación y los objetivos, incluida una descripción general de la tesis. El capítulo 2 trata sobre tecnologías anteriores tales como IoT, LoRaWAN, plataformas de manejo de datos y microcontroladores. El capítulo 3 habla sobre el diseño y como se desarrolló el sistema, hablado sobre la estructura y los elementos que la componen. El Capítulo 4 presenta los resultados de las pruebas y los hallazgos obtenidos a partir de la simulación del sistema. Para finalizar, el Capítulo 5 presenta las conclusiones obtenidas de este trabajo y recomendaciones para futuras investigaciones.

1.2. Antecedentes y Estado del Arte

La monitorización inteligente de aparcamientos es un área que se ha investigado activamente durante la última década, impulsada por la necesidad de optimizar los recursos urbanos y mejorar la calidad de vida en las ciudades inteligentes.

En investigaciones de hace poco tiempo se ha visto una clara inclinación al uso de sensores para saber si un espacio está ocupado o no usando redes que lleguen muy lejos y que no gasten mucha energía

Por ejemplo, gracias a esto sabemos que hay muchos estudios que han dicho que un sistema de parqueo inteligente que este usando sensores con IoT, nos permiten que se puedan ver los datos de forma más rápida (IEEE Access o Sensors Journal de 2018 a 2022).

Elegir el tipo de tecnología de comunicación correcta es importante para asegurar que el sistema de vigilancia de parqueos sea eficiente y se pueda hacer en la vida real. En este contexto, LoRaWAN está ganando cada vez más reconocimiento gracias a las ventajas que ofrece frente a otras alternativas técnicas.

Estudios recientes (por ejemplo, Singh & Gupta, 2022), recogidos en Sensors e IEEE Access, han demostrado la viabilidad de aplicar la tecnología LoRaWAN a proyectos de monitorización medioambiental, gestión de infraestructuras y gestión de aparcamientos en entornos urbanos y campus universitarios. Estos estudios destacan la capacidad de la tecnología para cubrir grandes áreas con una infraestructura de red mínima, y su bajo consumo de energía, especialmente en los nodos finales, contribuye a prolongar la vida útil de la batería de los sensores.

Los trabajos han buscado diferente forma de mantener un control con respecto a las zonas de estacionar, algunos han realizado monitoreos por computadora (Kumar et al., 2021), y alguno otros han hecho uso de varios tipos de sensores colocados en diferentes lugares (Gómez y Pérez, 2020).

El hecho de sumar microcontroladores de bajo costo de energía ha sumado un gran valor para poder desarrollar nodos de bajo costo con Lora (Wang et al., 2024).

Para Latinoamérica el poder desarrollar ciudades inteligentes aún se ve como un sueño, es cierto que hay diseños de redes que se pueden usar, pero están pensados de forma más general y no tan centrado en los diferentes tipos de entornos que hay. (Rodríguez & Silva, 2021).

Teniendo en cuenta las investigaciones que se han hecho, con este trabajo se pretende desarrollar un sistema que funcione específicamente para la Universidad Católica de Santiago de Caguayquil.

Proyecto 1: " Análisis del rendimiento de un sistema de aparcamiento inteligente LoRaWAN en un campus universitario (Al-Fuqaha et al., 2022).

El trabajo estuvo centrado en el desarrollo y la integración de un sistema de parque para la UCSG.

- **Metodología:** Se hizo una red de 150 nodos que usen microcontroladores y sensores que se puedan comunicar con LoRaWAN. También se hicieron pruebas con la finalidad de medir el rendimiento del sistema
- **Resultados y Conclusiones:** El sistema llegó a estándares superiores a lo esperado y se utilizó un SF bajo que se usan en nodos que están a menos de 1km. En los nodos que estaban más lejos se usaron SF de más rango

Proyecto 2: "Energy Consumption Modeling of ESP32-based LoRaWAN Nodes for IoT Applications" (Modelado del Consumo Energético de Nodos LoRaWAN Basados en ESP32 para Aplicaciones IoT) (Sanchez et al., 2023).

Este trabajo está centrado en el aspecto más importante para que los nodos de IoT se puedan mantener con un costo de energía bajo

- **Metodología:** Se basó en experimentar sobre el gasto de energía de un nodo de sensor que está hecho por un ESP32 y un módulo Lora (SX1276) en sus estados: transmisión, recepción, lectura del sensor y principalmente, en el modo de sueño profundo. Se programó el nodo para transmitir datos solo cuando se detectaba un cambio de estado, permaneciendo en modo de sueño profundo el resto del tiempo.

Resultados y Conclusiones El trabajo mostro que cuando el sistema está en un modo de suspensión profunda tiene un menor consumo de energía. A pesar del momento de más gasto, que es en la trasmisión, pero como es un periodo corto no influye mucho.

Proyecto 3: "A Comparative Study of Ultrasonic and Magnetometer Sensors for a Cloud-Integrated Smart Parking System" (Estudio Comparativo de Sensores Ultrasónicos y Magnéticos para un Sistema de Parqueo Inteligente Integrado en la Nube) (Lin & Chen, 2021).

Este trabajo se centró en todas las variables posibles que pude tener un proyecto de este estilo.

- **Metodología:** Se hizo un sistema que enviaba información de un punto a otro, la cual mostraba si había espacios libres para que un carro pudiera estacionarse.
- **Resultados y Conclusiones:** Los sensores magnéticos son un poco mejores en cuanto a no generar errores que hagan entender que hay un lugar vacío en un sitio que esta ocupado. También se vio gracias al trabajo que es muy sencillo integrar este tipo de sistemas a la nube. Esto justifica la elección de sensores ultrasónicos para la simulación en este proyecto y valida la arquitectura del sistema propuesta.

1.2.1. Proyectos de Monitoreo de Parqueo con Sensores Basados en Presencia

Los sistemas de aparcamiento inteligente que utilizan sensores de detección de presencia, como los ultrasónicos, magnéticos o de presión, se caracterizan por una alta fiabilidad de detección a nivel de cada plaza de aparcamiento.

Proyecto 1: "Smart Parking IoT Based: Design and Prototype" (2020).

El trabajo se centra en la creación de un sistema de parque inteligente que use los sistemas IoT. En el proyecto se usaron diferentes tipos de sensores, unos para medir sonido y otros para medir luz para saber si ya o no hay carros parqueados en un lugar definido. (Arduino Mega 2560). La información pasa a una plataforma donde la gente puede ver la información

El sistema demostró que funcionaba y que podía realizar la detección de espacios ocupados o libres. Con esto se determinó que el sistema es eficaz para ser usado en lugares como parqueaderos. (Amalia et al., 2020).

Proyecto 2: "Smart Parking System based on Internet of Things" (2020).

Este proyecto aborda diferentes metodologías que se pueden usar en sistemas de parqueos inteligentes, utilizando diferentes tipos de sensores útiles para este trabajo.

También detallan un sistema que utiliza sensores ultrasónicos encima de las plazas de aparcamiento. Resultados y conclusiones: Los autores concluyen que los sensores ultrasónicos son más rentables en términos de detección. En cambio, los sensores magnéticos son más robustos, pero su instalación es más compleja y costosa. Subrayan que la elección de la tecnología de sensores depende del entorno específico y del presupuesto del proyecto (Rakshit & Kanmani, 2020).

1.2.2. Proyectos de Monitoreo de Parqueo con Visión por Computadora

Los sistemas de monitoreo de parqueo basados en visión por computadora han ganado terreno gracias a los avances en inteligencia artificial y deep learning. Estos proyectos utilizan cámaras para capturar imágenes o videos de un área de parqueo, que luego son procesados para identificar vehículos y espacios libres.

Ventajas:

- Alta Precisión: La visión por computadora, especialmente con algoritmos de deep learning (como redes neuronales convolucionales), puede lograr

una precisión muy alta en la detección, incluso bajo diferentes condiciones de luz o clima.

- **Monitoreo de Múltiples Espacios:** Una sola cámara puede cubrir una amplia área, lo que reduce la cantidad de hardware necesario en comparación con los sistemas basados en sensores individuales por cada plaza de parqueo.
- **Información Adicional:** Además de la ocupación, estos sistemas pueden recolectar datos valiosos como el tipo de vehículo, el tiempo de parqueo, y detectar comportamientos anómalos.

Desventajas:

- **Alto Costo de Implementación:** El costo inicial de las cámaras de alta resolución y los servidores de procesamiento (GPU, etc.) es significativamente mayor que el de los sistemas basados en sensores.
- **Consumo de Energía y Ancho de Banda:** Cuando se transmiten videos en vivo se suele generar un gasto de energía bastante alto y esto es un inconveniente para los sistemas LoraWAN
- **Privacidad y Seguridad:** Los usuarios pueden sentirse incómodos al notar que hay cámaras en lugares públicos.

1.2.3. Aplicaciones de Redes LPWAN en Entornos Universitarios

Lo fácil de utilizar las redes LPWAN lo convierte en la mejor opción para una variedad de usos diferentes y en diferentes áreas.

Proyecto 1: Se puede utilizar en el control del aire y el agua en diferentes sectores que necesiten un monitoreo constante y de bajo costo de este tipo de cosas. Se colocan los lugares estratégicos y los nodos se alimentan de una batería y por medio de toda la estructura envían información a la nube. Proyectos como estos nos enseñan que Lora puede proporcionar una gran cobertura. (Aalto University, 2020).

Proyecto 2: Se puede utilizar Lora también para fines de seguridad, los dispositivos que son de seguridad se conectan a dispositivos de realizan rastreos. Estos mismos dispositivos pueden detectar si una ventana es rota o si una puerta es forzada, enviando notificaciones a seguridad mediante la red Lora. (STC Group, 2024)

1.3. Definición del Problema

El campus de la Universidad Católica de Santiago de Caguayquil (UCSG) experimenta problemas recurrentes con la gestión de las plazas de aparcamiento. Durante las horas pico y los eventos académicos, la demanda de estacionamiento supera la oferta, creando una serie de inconvenientes que afectan la eficiencia y la experiencia de la comunidad universitaria.

El problema principal es la falta de un sistema automatizado que proporcione información precisa y en tiempo real sobre la disponibilidad de estacionamiento. Al entrar al campus, los conductores no tienen forma de saber dónde hay estacionamiento disponible, lo que les obliga a buscar entre numerosas plazas sin conseguir nada. Las consecuencias directas de esta situación son múltiples y significativas:

- Pérdida de Tiempo y Frustración para los Usuarios: Estudiantes, docentes y personal administrativo invierten un tiempo considerable buscando un lugar

para estacionar, lo que puede ocasionar retrasos a sus actividades y generar niveles elevados de estrés y frustración.

- Congestión Vehicular Interna: Los carros circulan constantemente por la universidad generando congestión por la falta de conocimiento sobre lugares disponibles de parqueo.
- Uso Ineficiente de la Infraestructura Existente: Como no hay un sistema que dirija a las personas, muchas veces hay sitios donde se pueden parquear, pero no lo saben.
- Impacto Ambiental y Consumo de Combustible: La búsqueda constante de un sitio de estacionamiento genera un mayor consumo de gasolina en el vehículo.

Esta falta de información dificulta la gestión previa y posterior de los recursos de estacionamiento, lo que afecta negativamente las operaciones diarias del campus.1.3.1.

Impacto y Alcance del Problema en la UCSG

Los problemas de gestión del estacionamiento en la Universidad Católica de Santiago de Guayaquil (UCSG) son un problema recurrente que afecta a la comunidad estudiantil, al profesorado y al personal administrativo. La alta demanda de estacionamiento, especialmente en horas punta, genera congestión vehicular, lo que afecta negativamente la eficiencia y la experiencia general en el campus.

Situación Actual y Estimaciones

Afluencia Vehicular Diaria: Entre semana, se estima que el tráfico en el campus principal de la UCSG supera los 3000 a 4000 vehículos, incluyendo estudiantes, profesores, personal y visitantes. El tráfico alcanza su punto máximo durante la madrugada (de 7:00 a 9:00) y el mediodía (de 12:00 a 14:00).

Número de Parqueos: El estacionamiento designado en el campus es limitado. Se estima que hay aproximadamente 2500 plazas en diversos estacionamientos y áreas del campus.

Tiempo Promedio de Búsqueda: La demanda de un sitio de parque ha incrementado mucho en los últimos tiempos, lo que genera diversos problemas tanto a quienes manejan, como a las personas que están caminando por la universidad dirigiéndose hacia sus clases. Ya que la gente no encuentra donde estacionarse y se estresan y esto puede llegar a afectar al momento de estar conduciendo.

Impacto Académico: Los estudiantes pierden tiempo buscando un lugar donde estacionar su carro y esto los puede llevar a llegar atrasados al salón de clase y esto puede traer problemas.

Impacto Ambiental: Las largas colas de carros, no solo generan ruido, sino que también pueden generar humo, contaminando con esto el aire de la universidad.

Impacto en la Gestión: El no tener un sistema de manejo central hace complicado que la administración entienda la disponibilidad de espacios de estacionamiento en tiempo real, lo que hace más difícil que los estudiantes puedan encontrar rápido un lugar de parqueo.

El desarrollo de una solución tecnológica, como la propuesta en esta tesis, no solo resolverá un problema logístico, sino que también contribuirá a la visión de la UCSG como un "Smart Campus", mejorando la calidad de vida de su comunidad, optimizando los recursos y promoviendo un ambiente más eficiente y sostenible. 1.4. Justificación del Problema

La justificación de este proyecto radica en la relevancia y los beneficios de implementar un sistema de monitoreo de estacionamiento inteligente para la Universidad Católica de Santiago de Guayaquil (UCSG), y la potencial aplicación de nuevas tecnologías para resolver problemas del mundo real.

1.4.1. Contribución a las Tecnologías de Ciudades Inteligentes y el "Smart Campus"

Este proyecto se alinea directamente con los principios y objetivos de las Ciudades Inteligentes y, en particular, con la visión de un "Smart Campus". Un campus universitario, al ser un ecosistema en sí mismo con su propia infraestructura, comunidad y desafíos de movilidad, es el lugar ideal para aplicar estas tecnologías. Específicamente, este trabajo contribuye a:

Reducir la Congestión: Si los estudiantes saben dónde hay un lugar de parque libre, esto reduce el tráfico interno ya que no deben estar buscando por todos lados.

Promover la Sostenibilidad Ambiental: Al hacer más cortos los tiempos de búsqueda también se evita que los carros emitan gases perjudiciales para el medio ambiente. Esto va de la mano con el Objetivo de Desarrollo Sostenible (ODS)

Mejorar la Calidad de Vida: Si se mejoran estos aspectos el estrés de los estudiantes bajaría lo que provocaría una mejor experiencia en la universidad.

1.4.2. Aporte a la Línea de Investigación.

Este proyecto resalta problemas importantes y urgentes mediante la aplicación práctica de tecnologías nuevas.

Este trabajo:

Valida la idoneidad de las redes LPWAN: Estamos de acuerdo con investigar la parte práctica y con documentar cómo funciona la red de bajo consumo y de largo alcance, como LoraWAAN en situaciones reales. Se analizan variables claves como el alcance, la latencia y la eficiencia en la energía

Aborda la interconexión de tecnologías: Este trabajo tiene una variedad de componentes de tecnología, empezando en la capa de percepción siguiendo en la capa de red y terminando con las capas de aplicación y manejo. Esta unión es importante para la ingeniería de sistemas y de IoT

Permite el estudio del rendimiento de redes inalámbricas: Estas simulaciones nos permitirán ver cómo se comporta la red según las situaciones a las que sea expuesta.

1.4.3. Viabilidad Técnica y Económica del Diseño (Análisis Preliminar)

Aunque el estudio económico detallado está fuera del alcance de la tesis, es crucial realizar un análisis preliminar que demuestre la viabilidad del diseño propuesto en términos técnicos y de costos.

Viabilidad Técnica:

Este diseño se basa en tecnologías desarrolladas y accesibles. Componentes como el microcontrolador ESP32 y el módulo LoRa son económicos, están ampliamente disponibles y bien documentados.

Esta propuesta se centra en la simulación, lo que la hace inherentemente viable dentro de los recursos y el tiempo requeridos para la redacción de tesis de posgrado. La simulación permite la verificación de la arquitectura sin el alto costo y la complejidad de una implementación física a gran escala.

Viabilidad Económica (Análisis Teórico):

Costo de Hardware: Los sistemas manuales o basados en tiques requieren intervención humana, lo que genera costos recurrentes. Por el contrario, los sistemas automatizados que utilizan sensores LoRaWAN incurren en costos iniciales de hardware, pero el bajo consumo de energía de los nodos de sensores y la red LPWAN eliminan la necesidad de costosas licencias de frecuencia, lo que reduce significativamente los costos operativos.

Costo por Espacio de Parqueo: Entender lo que cuestan este tipo de sensores es importante para el sistema que se desea. Si bien ninguna tecnología puede implementarse por sí sola, analizamos cómo proponer una solución que se pueda alquilar y sea rentable.

1.4.4. Potencial de Innovación y Futuras Investigaciones

El proyecto sienta las bases para futuras investigaciones y desarrollos. La decisión, que no se basó en resultados de simulación, se tomó como una posible implementación

piloto, integrándose con el sistema de expansión del campus y expandiéndose a otras áreas. Además, allana el camino para el desarrollo de algoritmos de optimización de estacionamiento y sistemas de reserva, o la integración con aplicaciones móviles para el usuario final.

1.5. Objetivos

1.5.1. Objetivo General

Diseñar y simular un sistema de monitoreo de parqueo inteligente utilizando tecnologías IoT y LoRaWAN para optimizar la gestión y disponibilidad de espacios de estacionamiento en el campus de la Universidad Católica de Santiago de Guayaquil (UCSG).

1.5.2. Objetivos Específicos

Investigar y seleccionar las tecnologías y componentes hardware y software adecuados, como sensores de ocupación (ej. ultrasónicos o magnéticos), microcontroladores (ej. ESP32) y módulos LoRaWAN, que permitan la detección fiable de vehículos y la transmisión eficiente de datos.

Diseñar la arquitectura completa del sistema de monitoreo de parqueo, definiendo la disposición de los nodos sensores, la configuración de la red LoRaWAN (incluyendo gateway y servidor de red) y la estructura de la plataforma IoT para la visualización y gestión de la información.

Desarrollar los algoritmos necesarios para el sistema y para el procesamiento de datos para la programación de los nodos sensores y la integración con LoRaWAN y la plataforma IoT seleccionada.

Hacer una simulación del sistema que se ha propuesto para conocer su rendimiento, flexibilidad y que eficacia tiene el mismo.

Realizar un análisis los resultados que se hayan conseguido por medio de las simulaciones, para saber que límites y fortalezas tiene el sistema.

1.6. Alcance y Limitaciones del Proyecto

El presente proyecto de tesis se enfoca en el "Diseño y simulación de un sistema de monitoreo de parqueo utilizando IoT y LoRaWAN para la UCSG". Para mantener un enfoque claro y realizable dentro del marco de tiempo de una tesis de pregrado/posgrado, se establecen los siguientes límites:

Inclusiones:

Fase de Diseño: El proyecto comprenderá el diseño conceptual y detallado de la arquitectura de hardware y software del sistema de monitoreo de parqueo. Aquí también

Tecnologías Específicas: Se realizará un estudio y también la aplicación de IoT y Lora WAN, como parte principal del sistema para llevar a cabo la comunicación y parte de la obtención de datos. Se usarán distintos microcontroladores y sensores.

Funcionalidad Principal: El objetivo principal del sistema es ayudar a saber si hay espacios disponibles de parqueo y enviar esta información.

Plataforma de Visualización: Se diseñará la integración con una plataforma IoT basada en la nube (ej., Ubidots) para la visualización gráfica del estado de los parqueos y la gestión de los datos.

Simulación: La verificación del diseño se realizará a través de una simulación de software, no una implementación física real.

Contexto: Aplicación y validación para el entorno del campus de la UCSG.

Exclusiones:

Implementación Física a Gran Escala: El proyecto no incluirá la construcción y despliegue físico de un prototipo a gran escala en el campus de la UCSG. La validación se limitará a los resultados de la simulación.

Desarrollo de Hardware Personalizado: No se diseñarán ni fabricarán componentes electrónicos a nivel de circuito impreso. Se trabajará con módulos y sensores comerciales disponibles en el mercado.

Sistema de Gestión Integral de Parqueo: El alcance se restringe al monitoreo de la ocupación.

Mantenimiento y Operación a Largo Plazo: El proyecto no abarcará la fase de mantenimiento, escalabilidad a largo plazo o gestión operativa del sistema una vez diseñado.

1.6.1. Exclusiones Detalladas

Para delimitar el alcance del proyecto y asegurar su viabilidad dentro de los parámetros de una tesis de grado, se establecieron exclusiones específicas. Cada una de estas exclusiones se justifica por la complejidad, el tiempo o los recursos que requerirían, y se proponen como posibles líneas de investigación futura.

Implementación Física a Gran Escala: Se excluyó la construcción e implementación física de un prototipo en el campus de la UCSG para evitar los altos costos asociados con el hardware, la instalación, la mano de obra y el mantenimiento. Además, la obtención de permisos y la logística para construir la infraestructura actual excederían el plazo de desarrollo de la tesis.

Sistemas de Gestión Avanzados (Reservación y Pago): Si bien se mencionan los mecanismos de seguridad de LoRaWAN, este documento no abarca un análisis exhaustivo de la ciberseguridad. Evaluar la vulnerabilidad del sistema a ataques de denegación de servicio (DoS), la integridad de los datos en la nube o la seguridad de la comunicación de extremo a extremo requiere investigación especializada y pruebas de penetración, que quedan fuera del alcance de este estudio.

Análisis de Seguridad Integral: Si bien se habla sobre los sistemas de seguridad de LoraWAN, ese trabajo no cubre un análisis profundo de ciberseguridad. Pero, estos aspectos si son importantes para futuras implementaciones.

Navegación Asistida y Aplicaciones Móviles para el Usuario Final: Elaborar un diseño para una aplicación que brinde indicaciones paso a paso para poder llegar a los sitios de parqueo que estén desocupados o que se uniera con un sistema que tenga GPS, era imposible porque su complejidad al momento de hacerlo era alta. También por la programación de la API de los mapas y por la necesidad que había de una interfaz de usuario sofisticada.

1.7. Hipótesis

La creación de un sistema de monitoreo para lugares de estacionamiento que incluya tanto detección del lugar con el uso de tecnología IoT y LoraWAN que permite una conectividad de largo alcance y un consumo de energía bajo nos muestra su capacidad para dar información exacta y en tiempo real sobre la disponibilidad de espacios de estacionamiento en el campus de la Universidad Católica de Santiago de Guayaquil (UCSG) en un entorno simulado, nos permite manejar mejor la información y brindar una mejor experiencia a los usuarios.

1.8. Metodología de Investigación

La metodología de este trabajo adoptará un enfoque híbrido, combinando elementos de investigación técnica y diseño/ingeniería. Este enfoque es idóneo para desarrollar soluciones tecnológicas y evaluar su rendimiento en un entorno de simulación.

1.8.1. *Detalle del Proceso de Diseño y Simulación*

El proceso de diseño y simulación consta de una serie de pasos secuenciales y lógicos que sirven como directrices de implementación detalladas para el desarrollo del artículo. Este enfoque garantiza que cada paso se base en el anterior, dando como resultado un modelo funcional y validado.

1. Enfoque de Investigación:

Descriptivo: Se realizará una lectura de los últimos avances en LoraWAN para estar al corriente.

Diseño: Se realizará la simulación de los escenarios en el programa NS-3

2. Tipo de Investigación:

Bibliografía: Se realizará una búsqueda en todo tipo de libros, páginas webs, revistas científicas de los últimos 5 años.

Simulación: Este trabajo busca dar soluciones a un problema que existe actualmente, se harán simulaciones para observar las variables y comprobar el funcionamiento del sistema.

3. Fases de la Metodología:

Fase 1: Análisis y Revisión Bibliográfica

Objetivo: Establecer el marco teórico,

Actividades:

Investigación

Estudio de la tecnología

Revisión de tipos de sensores.

Análisis de microcontroladores adecuados para nodos IoT (ej. ESP32: capacidades de procesamiento, conectividad, consumo).

Evaluaremos diferentes plataformas de IoT.

Identificación de trabajos previos similares y sus resultados.

Entregables: Marco teórico preliminar, lista de tecnologías seleccionadas con justificación.

Fase 2: Diseño del Sistema

Objetivo: Definir la arquitectura y los componentes específicos.

Actividades:

Definición de la arquitectura

Diseño del nodo sensor

Diseño de la topología de la red LoRaWAN

Elaboración de diagramas de flujo.

Entregables: Diagramas de arquitectura del sistema, especificaciones técnicas de hardware y software, diagramas de flujo.

Fase 3: Desarrollo y Programación

Objetivo: Implementar los modelos y el código necesarios.

Actividades:

Desarrollo del código.

Configuración del entorno de simulación.

Configuración y simulación de la interacción con la plataforma IoT. visualización de datos.

Entregables: Código de simulación, configuración del entorno de simulación, conexión simulada a plataforma IoT.

Fase 4: Simulación y Pruebas

Objetivo: Ejecutar la simulación y recopilar datos

Actividades:

Diseño de escenarios de prueba para la simulación

Ejecución de la simulación bajo diferentes condiciones

Recopilación sistemática de métricas clave

Entregables: Registros de datos de simulación, tablas de resultados.

Fase 5: Análisis de Resultados y Conclusiones

Objetivo: Interpretar los datos de la simulación.

Actividades:

Comparación de los resultados con los objetivos planteados.

Discusión de las implicaciones de los resultados.

Formulación de conclusiones claras y concisas.

Propuesta de recomendaciones para futuras.

Entregables: Informe de resultados, conclusiones, recomendaciones.

4. Herramientas y Software:

Para Simulación: Se utilizará NS-3 para realizar las simulaciones del sistema basado en IoT LoraWAN.

Capítulo 2: Fundamentación Teórica y Marco Teórico

En este capítulo se establecerán las bases y los conceptos sobre las técnicas que confirman el diseño y la simulación para un sistema de monitoreo para estacionamientos. Está conformado por las bases de IoT, LoraWAN, también por los tipos de sensores que serán utilizados, las plataformas de manejo para los datos en la nube y el microcontrolador que será el centro del sistema.

2.1. Arquitectura de Referencia.

No existe un solo estándar para la arquitectura de IoT, esto se suele hacer a través de un modelo de capas que haga más sencilla su comprensión. Actualmente hay varios modelos para IoT, siendo el modelo de 5 capas el más completo de todos, ya que incluye diferentes tipos de capas.

Capa de Percepción (Perception Layer): Está compuesta por dispositivos físicos (sensores, actuadores, puertas de enlace, teléfonos móviles) que interactúan directamente con el entorno. Su función principal es recopilar y procesar datos del entorno físico (p. ej., distancia, temperatura, presión). En este proyecto, esta capa consta de un sensor ultrasónico (HC-SR04) que detecta la ocupación de plazas de aparcamiento y un microcontrolador ESP32 que procesa esta información.

Capa de Red (Network Layer): La capa de red se encarga de transportar los datos a través de la red. Realizando la comunicación y dirigiendo los paquetes al destino que les corresponde.

Capa de Procesamiento/Middleware (Processing/Middleware Layer): La capa de procesamiento se encarga de comprender los datos, es la que recibe la información y la procesa, guarda y analiza.

Capa de Aplicación (Application Layer): La capa de aplicación es lo que verían las personas que usen la aplicación. Toda la información que es recibida se muestra en esta capa de manera sencilla para que las personas puedan interactuar con ella.

Capa de Negocio (Business Layer): La capa de negocio es la parte más alta en la estructura IoT, aquí se encuentran la toma de decisiones, los tipos de negocios, investigaciones de mercado y privacidad.

2.1.1. Protocolos de Comunicación del IoT

Los protocolos de comunicación en el IoT son esenciales para el intercambio de datos entre los dispositivos y la nube. Se dividen en protocolos de la capa de red y de la capa de aplicación.

Protocolos de la Capa de Red

Estos protocolos definen cómo se transmiten los datos a través de la red física.

- **IP (Internet Protocol):** El protocolo estándar para la transmisión de datos en Internet. Es único, pero puede ser "pesado" para dispositivos de baja potencia.
- **6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks):** Un protocolo de red que adapta IPv6 para su uso en redes de área personal de baja

potencia. Esto permite que los dispositivos de baja energía se integren en redes IP estándar.

- **LoRaWAN:** Aunque es un estándar de red de área extensa (WAN), también opera en esta capa gestionando eficientemente la comunicación entre nodos y puertas de enlace. Su diseño de largo alcance y bajo consumo le permite cubrir todo el campus de la UCSG con una infraestructura mínima, lo que lo hace ideal para proyectos como este.

Protocolos de la Capa de Aplicación

Estos protocolos definen cómo las aplicaciones se comunican entre sí, independientemente de la red subyacente.

- **MQTT (Message Queuing Telemetry Transport):** Este protocolo es ideal para IoT, ya que es bastante ligero, tiene recursos limitados y redes estables. Su modelo permite a los usuarios enviar información a un intermediario para que otros sistemas reciban la información
- **CoAP (Constrained Application Protocol):** Diseñado para dispositivos restringidos, CoAP es similar a HTTP, pero optimizado para redes de baja potencia. Funciona en un modelo de solicitud/respuesta (request/response).
- **HTTP (Hypertext Transfer Protocol):** Aunque es el protocolo de Internet más común, es demasiado "pesado" para la mayoría de los dispositivos IoT, ya que requiere más energía y ancho de banda, por lo que no es la mejor opción para la comunicación entre los nodos y el gateway en tu sistema.

2.2. Descripción General de la Tecnología LoRaWAN

LoRaWAN (Red de Área Amplia de Largo Alcance) es un protocolo de comunicación inalámbrica de bajo consumo y largo alcance, diseñado específicamente para aplicaciones del Internet de las Cosas (IoT).

Una de las principales características de la tecnología LoraWAN, es su capacidad de conectarse a dispositivos que usan baterías de larga duración y enviar pequeñas cantidades de datos. Sin embargo, admite distancias de varios kilómetros en entornos urbanos y de hasta decenas de kilómetros en zonas rurales. Por lo tanto, puede utilizarse como una alternativa eficaz a las redes celulares o Wi-Fi en diversas aplicaciones del IoT.

2.2.1. Arquitectura de LoRaWAN

La arquitectura de una red LoRaWAN es típicamente de tipo estrella-de-estrellas y consta de cuatro componentes principales:

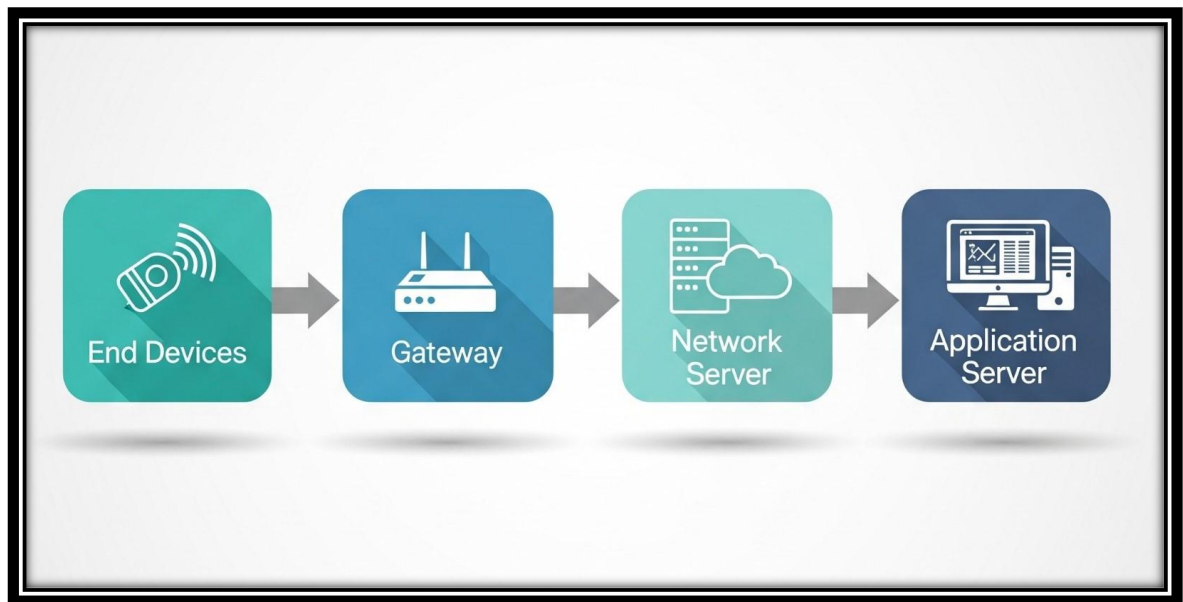
Dispositivos Finales (End Devices): Son los nodos o sensores IoT (en nuestro caso, los sensores de ocupación de parqueo con su microcontrolador y módulo LoRa). Se comunican de forma inalámbrica con los gateways LoRaWAN. Son de bajo consumo y pueden operar con baterías durante largos periodos.

Gateways LoRaWAN (Estaciones Base / Concentradores): La puerta de enlace actúa como puente entre los dispositivos terminales y el resto de la red. Recibe paquetes LoRa de múltiples dispositivos terminales y los reenvía al servidor de red LoRaWAN mediante conexiones IP estándar (Ethernet, Wi-Fi, 4G/5G).

Servidor de Red LoRaWAN (Network Server): Esta parte es el centro de la red Lora, como función principal tiene eliminar datos duplicados, recibir datos, verificar la seguridad de los datos, es también quien decide donde se enviarán los datos y se encarga de organizar la red

Servidor de Aplicación (Application Server): El servidor de la aplicación tiene como función principal la decodificación y procesamiento de los datos que recibe. Es la parte que interactúa con la plataforma de IoT.

Figura 1 - Gráfico de Bloques de la Arquitectura Lora.



Nota: Fuente: El autor.

2.2.1.1. Fundamentos Técnicos de la Modulación LoRa

Lora en su capa física usa una modulación de tipo espectro, haciéndola ensanchar mediante “chirrido” (CSS), lo que es una técnica de modulación que viene de la tecnología de espectro ensanchado, la cual es utilizada en sistemas de comunicaciones de tipo militar y satelital.

SF bajo (SF7): Permite una mayor velocidad de datos y un menor tiempo de transmisión, pero reduce el alcance y la inmunidad al ruido. Es ideal para dispositivos cercanos al gateway.

SF alto (SF12): Aumenta bastante el alcance del enlace. Es adecuado para dispositivos muy alejados del gateway o en entornos con muchas interferencias.

El poder elegir un SF apropiado y el mecanismo de velocidad de datos (ADR) permiten que las redes LoRaWAN aprovechen al máximo el consumo de energía de los nodos en función de su distancia desde la puerta de enlace.

2.2.1.2. Clases de Dispositivos LoRaWAN

LoRaWAN define tres clases de dispositivos, diseñadas para dar equilibrio al consumo de energía y la latencia en la comunicación.

Clase A: Esta clase ofrece la mayor eficiencia energética. La comunicación es bidireccional, pero los dispositivos solo pueden transmitir mensajes una vez.

Clase B: Sincronizado con la puerta de enlace, proporciona una ventana de recepción adicional en un horario predecible.

Clase C: Esta es la clasificación de eficiencia energética más baja, ya que el dispositivo mantiene abierta la ventana de recepción (excepto al transmitir mensajes de enlace ascendente).

2.2.1.3. Mecanismos de Seguridad en LoRaWAN

La seguridad es un muy significativo en la arquitectura LoRaWAN, implementada a través de múltiples capas de cifrado para proteger la confidencialidad e integridad de los datos.

Activación del Dispositivo: Los dispositivos se activan en la red de dos formas:

- **Activación por Personalización (ABP - Activation by Personalization):**

La clave de cifrado a sido grabada antes en el dispositivo y estará ahí todo el tiempo.

- **Activación por Aire (OTAA - Over-the-Air Activation):** Este método

sería el más seguro ya que genera la clave al momento de hacer el enlace con la red.

Cifrado de Sesión:

- **Cifrado de Capa de Red (NwkSKey):** El cifrado AES128 se utiliza para

garantizar la autenticidad e integridad de los mensajes entre los dispositivos finales y los servidores de red, protegiendo así los metadatos de la comunicación.

- **Cifrado de Capa de Aplicación (AppSKey):** Este tipo de cifrado, genera un

cifrado de extremo a extremo dando muchísimas más seguridades.

- **Contadores de Trama (Frame Counters):** Estos contadores deben incrementarse para cada mensaje transmitido desde servidores de red y puntos finales, lo que ayuda a prevenir ataques de retransmisión de mensajes.

Estos mecanismos aseguran que los datos sean protegidos desde el sensor hasta la aplicación final.

2.2.2. Comparación con Otras Tecnologías LPWAN

Para justificar la elección de LoRaWAN, es importante compararla con otras tecnologías LPWAN líderes en el mercado, como Sigfox y NB-IoT.

Conclusión de la Comparación:

Sigfox y NB-IoT se han convertido en opciones viables para el proyecto, ofreciendo una competencia viable frente a LPWAN y LoRaWAN. La iniciativa principal es la licencia y el acuerdo de licencia del operador para implementar una red privada en el campus de la UCSG. Esto permitirá cubrir los costos recurrentes y a los operadores de telecomunicaciones, ya que las baterías consumen mucha energía y deben mantenerse en perfecto estado según los requisitos de los sensores del parque, y el operador necesita reducir costos.

2.3. Tecnologías de Detección de Ocupación

La detección de ocupación de plazas de aparcamiento es esencial para los sistemas de monitorización inteligentes. Existen diversas tecnologías de sensores, cada una con sus propias ventajas y desventajas en cuanto a precisión, coste, consumo energético y facilidad de instalación.

2.3.1. Tipos de Sensores para Parqueo Inteligente: Una Comparación Detallada

Seleccionar el sensor adecuado es crucial para el éxito de un sistema de monitorización. La siguiente tabla compara los sensores más utilizados en soluciones de estacionamiento inteligente y evalúa sus características clave para justificar la elección de sensores ultrasónicos.

Justificación de la Elección:

Para este proyecto, un sensor ultrasónico como el modelo HC-SR04 es la mejor opción, ya que ofrece el equilibrio óptimo entre costo, facilidad de implementación y precisión. Este tipo de sensor se puede simular y permite verificar la lógica del sistema sin la compleja instalación física ni los altos costos de otras tecnologías.

2.3.2. Principio de Funcionamiento del Sensor Ultrasónico - HC-SR04

El sensor ultrasónico HC-SR04 funciona según el principio de eco sonar, un método de medición de distancia. El sensor consta de un transmisor, un receptor y dos transductores. El proceso de medición consta de los siguientes pasos:

Emisión de Onda: Un microcontrolador (por ejemplo, ESP32) envía un pulso de activación al pin Trig del sensor, lo que hace que el transductor emita ocho pulsos ultrasónicos de 40 kHz en sucesión.

Tiempo de Vuelo (Time of Flight): El sensor emite estos pulsos y activa un cronómetro interno. Las ondas de sonido viajan a través del aire, rebotan en el objeto (el vehículo) y regresan al transductor receptor del sensor.

Detección y Medición: Cuando el transductor receptor detecta la onda de retorno, detiene el cronómetro. El tiempo transcurrido, conocido como "Tiempo de Vuelo" (T), es la base para el cálculo de la distancia.

Cálculo de la Distancia: La distancia se calcula utilizando la siguiente fórmula:

$$\text{Distancia} = (\text{Velocidad sonido} * \text{Tiempo})/2$$

Velocidad sonido: La velocidad del sonido en el aire (aproximadamente 343 m/s o 0.0343 cm/ μ s a 20°C). Es importante dividir el resultado por dos, ya que el tiempo medido incluye tanto el viaje de ida como el de vuelta de la onda.

Factores que Afectan la Medición:

Temperatura y Humedad: La velocidad del sonido varía con la temperatura y la humedad. A mayor temperatura, el sonido viaja más rápido.

Forma y Ángulo del Objeto: Superficies inclinadas o formas irregulares pueden dispersar la onda de sonido, lo que reduce la precisión o impide que el eco regrese al sensor.

Obstáculos: Puede ser cualquier cosa que se interponga entre el sonido y su objetivo, esto puede generar ecos falsos.

Ruido: Cualquier tipo de ruido que se interponga entre la señal y su destino puede generar interferencias.

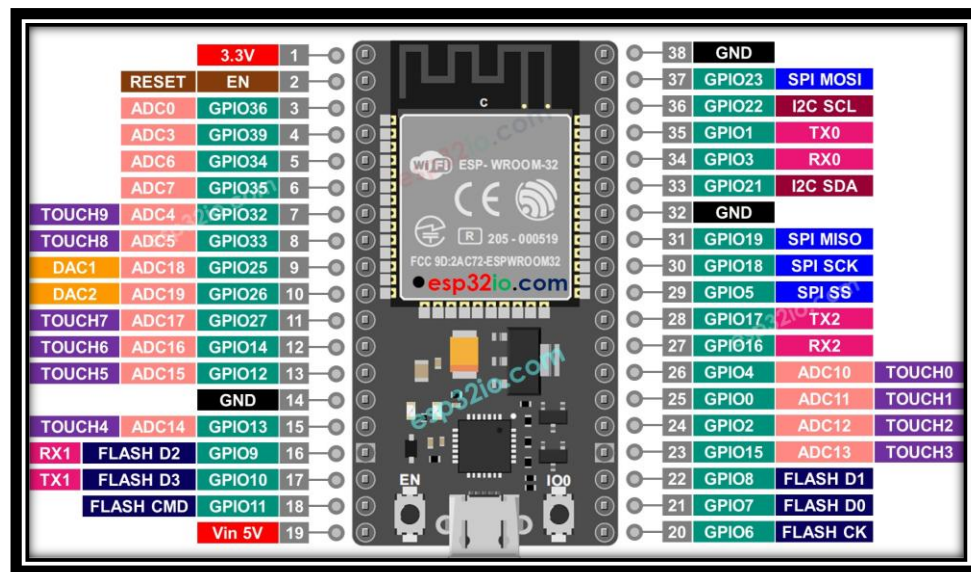
2.4. Componentes Clave del Sistema

Para el diseño del sistema de monitoreo de estacionamiento, seleccionamos el microcontrolador ESP32 y el módulo de radiofrecuencia LoRa como nodos sensores. Estos componentes se eligieron por su equilibrio óptimo entre potencia de procesamiento, bajo consumo de energía y rentabilidad, lo que los hace ideales para aplicaciones IoT alimentadas por batería.

2.4.1. Características Detalladas del ESP32

El ESP32 es un microcontrolador de la línea de productos Espressif Systems que integra Wi-Fi y Bluetooth, pero que en este proyecto se utiliza con un módulo LoRa. Sus características lo hacen muy versátil para aplicaciones IoT.

Figura 2 - Microcontrolador ESP32



Nota: Fuente: ESP32I/O - <https://esp32io.com>

Arquitectura de Doble Núcleo: La mayoría de los modelos ESP32 incorporan una CPU de doble núcleo Tensilica Xtensa LX6 que opera a una velocidad de hasta 240 MHz. Esto permite que un núcleo ejecute tareas de procesamiento (como la lógica del sensor) mientras el otro gestiona las tareas de comunicación, lo que garantiza un rendimiento óptimo y una rápida capacidad de respuesta.

Memoria: Tiene de una memoria de 520 KB con la cual ejecuta los códigos y también le sirve para almacenar datos.

Periféricos y Conectividad: El ESP32 ofrece diversos periféricos, como SPI, I²C y UART, que son importantes para la comunicación con otros componentes, como sensores ultrasónicos y módulos LoRa. También cuenta con varios pines GPIO (entrada/salida de propósito general) y un conversor ADC/DAC (analógico-digital/digital-analógico), lo que le permite interactuar con sensores analógicos y digitales.

Modos de Bajo Consumo: Una de las características más importantes del ESP32 para este proyecto son sus modos de sueño profundo, que mejora la duración de la batería:

Modo Light-sleep: La CPU y algunos periféricos se apagan. El consumo de batería se reduce a unos 80 μ A, y el estado de la RAM y la CPU se conserva. Es ideal para períodos cortos de inactividad entre lecturas.

Modo Deep-sleep: La CPU y casi todos los periféricos se apagan por completo. El estado de la RAM y la CPU se reinicia, y solo una parte de la memoria (el reloj de tiempo real) permanece activa para conservar los datos de las marcas de tiempo. En este modo, el consumo de batería se reduce aproximadamente a 5 μ A, lo que aumenta la vida

útil de la batería durante varios años, una característica clave de los nodos de sensores LoRaWAN.

2.4.2. Módulo de Radiofrecuencia LoRa

El módulo de radiofrecuencia LoRa, comúnmente basado en el chipset SX127x de Semtech, es el componente que habilita la comunicación de largo alcance y bajo consumo en la red LoRaWAN.

Conexión Física (SPI): El microcontrolador se conecta con el Lora a través del protocolo SPI.

Lógica de Programación: Primeramente, se debe programar el firmware del microcontrolador para que este pueda interactuar con el módulo Lora. Esto incluye una programación completa de interfaz, parámetros y la forma en la que se van a enviar y recibir datos.

Importancia de la Antena: Estas son de los componentes más importantes que puede haber en una red, ya que mediante ellas se conecta el módulo Lora a la red de forma inalámbrica. La forma en la que se diseña y la ganancia de la antena están muy comprometidas con la calidad de la conexión.

La combinación de la potencia de procesamiento del ESP32 con la eficiencia de comunicación del módulo LoRa crea una arquitectura de nodo robusta y escalable que cumple a la perfección los objetivos de este proyecto de tesis.

2.5. Descripciones Técnicas del ESP32

El ESP32 es una serie de microcontroladores de bajo costo y bajo consumo desarrollada por Espressif Systems. Combina un procesador potente, capacidades de comunicación versátiles y numerosos pines GPIO para interactuar con sensores y otros periféricos, lo que lo hace ampliamente utilizado en proyectos de IoT.

Características Principales:

Procesador: La mayoría de las variantes del ESP32 incluyen un microprocesador Xtensa LX6 o LX7 de doble núcleo de 32 bits con velocidades de reloj de hasta 240 MHz, lo que proporciona una potencia de procesamiento significativa incluso para tareas complejas.

Conectividad Integrada:

Wi-Fi: Soporta 802.11 b/g/n, lo que permite la conexión a redes Wi-Fi estándar.

Bluetooth: Incluye Bluetooth v4.2 BR/EDR y Bluetooth LE (Low Energy), ideal para comunicaciones de corto alcance o conexión con dispositivos móviles.

Memoria: Tiene diferentes tipos de configuraciones de memoria RAM y Flash, tanto que le permite almacenar firmwares complejos y mantener datos en su interior guardados.

Periféricos: Posee una amplia gama de periféricos que pueden ser utilizados

Bajo Consumo de Energía: Esta hecha para sistemas de un consumo de energía bajo, tiene distintos modos de descanso para poder ahorrar más energía y que dure más tiempo.

Seguridad: Incluye funciones de seguridad como arranque seguro, cifrado flash y un generador de números aleatorios (RNG), esenciales para la protección de datos del IoT.

El ESP32 es ideal para funcionar como núcleo de cada nodo de sensor de aparcamiento de nuestro sistema. Si bien el ESP32 no incorpora LoRa, se integra fácilmente con módulos LoRa externos mediante interfaces como SPI.

2.5.1. Funcionalidades Detalladas de Ubidots

Ubidots es una de las plataformas de IoT más utilizadas, centrada en la facilidad de uso y la visualización de datos. En este proyecto, Ubidots desempeña un papel clave como capa de procesamiento y aplicación del sistema, proporcionando las siguientes funciones clave:

Gestión de Dispositivos: El registro y el manejo de cada nodo sensor es sencillo. Cada dispositivo virtual creado en Ubidots se asocia a un token API único, lo que garantiza la autenticación y el control del acceso a los datos.

Recepción y Almacenamiento de Datos: La plataforma brinda varios end points para recibir datos, incluyendo una API REST (HTTP) y MQTT. Para redes LoRaWAN, el servidor de red transmite datos de forma segura a Ubidots. La plataforma normaliza y guarda la información de forma escalable, eliminando la necesidad de una gestión de bases de datos independiente.

Procesamiento de Datos: Ubidots permite crear variables computadas y eventos. Las variables computadas permiten aplicar lógica a los datos. Los eventos permiten definir reglas que generan notificaciones o alertas.

Dashboards de Visualización: Esta es una de las mayores fortalezas de Ubidots. Proporciona una amplia gama de widgets (medidores, mapas, gráficos de líneas, contadores) para crear dashboards interactivos y personalizables.

Mapa Interactivo: Un plano del campus con cada espacio de parqueo codificado por color (verde para libre, rojo para ocupado).

Contadores en Tiempo Real: Mediante esto podemos saber la cantidad de espacios vacíos disponibles.

Gráficos Históricos: Nos permiten ver a lo largo del tiempo cuales son los lugares más usados según los días de la semana y las horas del día

API e Integración: Ubidots gracias a su sistema nos permite integrarnos fácilmente con la aplicación y servicio. Gracias a esto se pueden hacer aplicaciones móviles externas.

2.5.2. Alternativas a Ubidots

Si bien Ubidots es una excelente opción, existen otras plataformas IoT que ofrecen funcionalidades similares, cada una con su propio modelo de negocio y enfoque.

Justificación de la Elección:

Para un proyecto de tesis, Ubidots es la mejor opción. A diferencia de ThingsBoard, AWS y Google Cloud, Ubidots ofrece facilidad de uso y una implementación rápida, lo que permite centrarse en la lógica del sistema sin perder tiempo en las complejidades de la configuración de la plataforma.

Capítulo 3: Diseño y Desarrollo

3.1. Análisis de los Componentes.

El sistema que estamos utilizando tiene tres bloques principales, el nodo sensor, la red Lora de comunicación y la plataforma IoT para guardar los datos en la nube.

3.1.1. *Arquitectura en Bloques.*

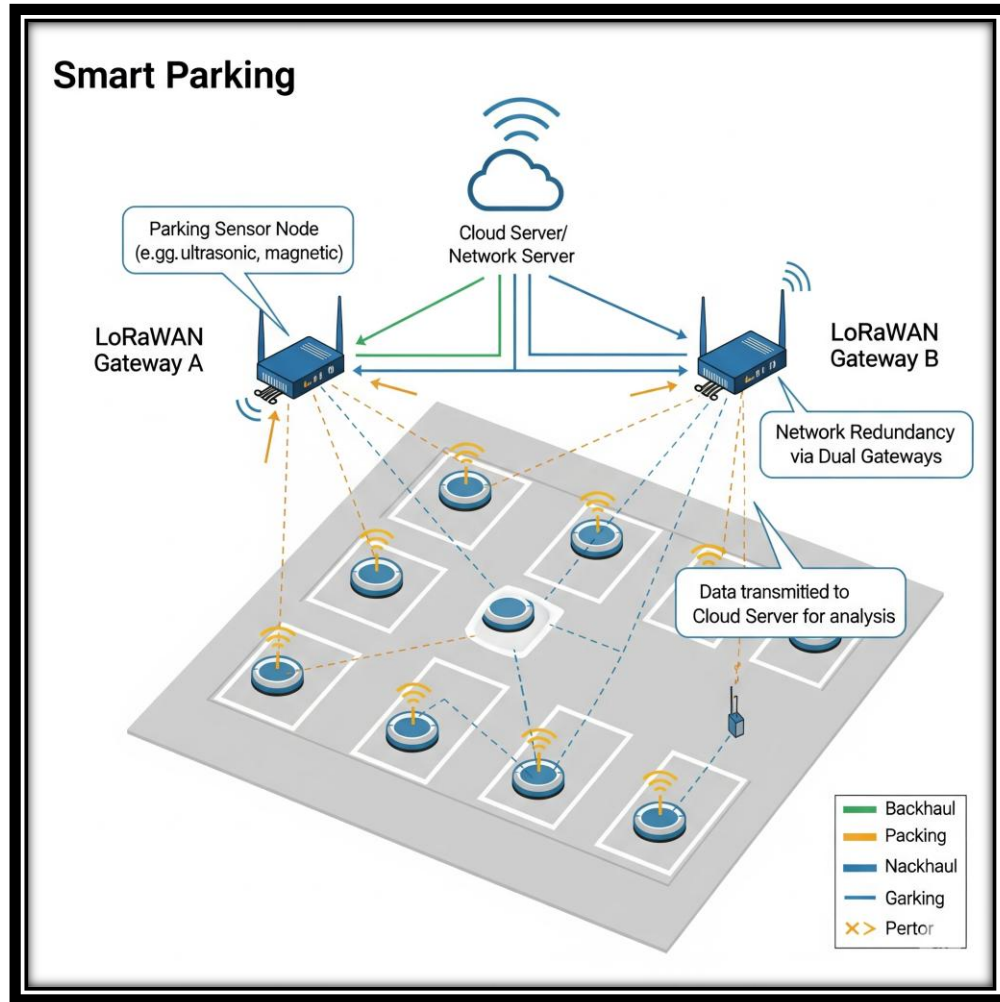
La estructura del sistema puede ser vista como una serie de datos, desde la parte física hasta el momento en que se entregan los datos al usuario final.

Nodo del sensor de aparcamiento: Aquí es donde empiezan los datos, aquí se encuentra el sensor, el microcontrolador y el dispositivo Lora. Este dispositivo es avisara donde hay un espacio disponible de parqueo.

Red de comunicación LoRaWAN: La red de comunicación Lora en la puerta de enlace es la que se encarga de recibir los datos y enviarlos a la red. También se encarga de ver la seguridad del sistema, revisar que tan fiable es, básicamente administra la red.

Plataforma IoT en la nube: Es el lugar al que llegan todos los datos de la red, se decodifican la carga útil de los datos Lora y luego se pasan a la plataforma IoT. Aquí se guardan los datos, también se los procesa y se los analiza y luego se muestran al usuario.

Figura 3 - Red de Estacionamiento Inteligente con LoraWan



Nota: Fuente: El autor.

3.1.2. Detalle del Nodo Sensor

El nodo sensor es el corazón del sistema, y su diseño físico y lógico es crucial para el éxito del proyecto.

Hardware del Nodo Sensor:

ESP32: El cerebro del nodo, encargado de controlar todas las operaciones.

Módulo LoRa: Responsable de la comunicación de largo alcance.

Sensor Ultrasónico: Mide la distancia para detectar la presencia del vehículo.

Fuente de Alimentación: Una batería Li-Po o pilas AA, con un circuito de gestión de energía.

Diagrama Esquemático de Conexión:

El ESP32 se conecta al módulo ultrasónico y al módulo LoRa de la siguiente manera:

ESP32 y Sensor Ultrasónico (HC-SR04):

Pin Trig del HC-SR04: Se conecta a un pin GPIO digital del ESP32 para iniciar la emisión del pulso ultrasónico.

Pin Echo del HC-SR04: Se conecta a otro pin GPIO digital del ESP32 para recibir el pulso de eco y medir el tiempo de viaje.

ESP32 y Módulo LoRa:

La comunicación se realiza a través de la interfaz SPI.

MOSI, MISO, SCK: Pines de datos y reloj compartidos para la comunicación SPI.

NSS (CS): Pin de selección de chip para activar o desactivar el módulo LoRa.

DIO0: Pin de interrupción que notifica al ESP32 que el módulo LoRa ha recibido o enviado un paquete.

3.1.3. Detalle de la Infraestructura de Red LoRaWAN

La topología de la red LoRaWAN se ha diseñado para cubrir el área de parqueos del campus de la UCSG de forma eficiente.

Topología Física:

La red utiliza una topología en estrella. Los nodos sensores ubicados en cada plaza de aparcamiento se comunican directamente con una única puerta de enlace LoRaWAN, que actúa como agregador de datos. Esta puerta de enlace se ubicará en una ubicación altamente centralizada dentro del campus para maximizar la cobertura y minimizar la necesidad de múltiples puertas de enlace.

Ubicación de los Componentes:

Nodos Sensores: Se instalarán en los espacios de parqueo del campus, con su sensor ultrasónico apuntando hacia abajo para detectar la presencia de vehículos.

Gateway LoRaWAN: Una ubicación ideal es un edificio alto, como una torre administrativa o un edificio académico central. Esto garantiza que la puerta de enlace tenga una línea de visión despejada hacia los nodos sensores, maximizando el alcance de comunicación de largo alcance de LoRa.

3.1.4. Diseño de la Base de Datos y la Interfaz de Usuario

Aunque la plataforma Ubidots se encarga de la gestión de la base de datos y la interfaz de usuario, es importante conceptualizar su diseño lógico.

Diseño Lógico de la Base de Datos:

Los datos se estructurarán de manera que sean fáciles de consultar y analizar. Los datos mínimos al almacenar para cada evento de detección de ocupación son:

- id_parqueo (identificador único del espacio de parqueo, ej., "A-1").
- estado_ocupacion (valor numérico: 0 para libre, 1 para ocupado).
- timestamp (momento exacto de la medición).
- distancia (valor del sensor ultrasónico, para fines de calibración y análisis).

Esta estructura de datos permite rastrear el estado de cada espacio a lo largo del tiempo, calcular estadísticas de ocupación y generar gráficos históricos.

Diseño de la Interfaz de Usuario (Dashboard):

La interfaz de usuario se diseñará en Ubidots para ser intuitiva y visual. El dashboard principal incluirá los siguientes widgets:

Mapa Interactivo: Un mapa del campus con íconos que representan cada espacio de parqueo. Estos íconos cambiarán de color en tiempo real, proporcionando una visión general inmediata de la disponibilidad.

Contador de Plazas: Es un widget que mediante números mostraría la cantidad de espacios disponibles para parquearse.

Gráfico de Ocupación: Una gráfica que pueda mostrar la cantidad de tiempo que se utilizan uno o varios parqueos.

Tabla de Eventos: Un widget en forma de tabal que permita ver los eventos más recientes (ej: Parqueo A-1 ahora está ocupado a las 10:30 a.m.)

Figura 4 - Visual del Dashboard de Monitoreo



Nota: Fuente: El autor

3.2. Lógica de Programación y Flujo de Datos

El funcionamiento del sistema se describe a través de los flujos de datos y la interacción entre sus componentes, desde la detección inicial hasta la visualización final.

3.2.1. Lógica del Firmware del Nodo Sensor

El firmware del ESP32 está diseñado para maximizar la eficiencia energética y la fiabilidad de la transmisión. A continuación, se detalla la lógica de su loop de funcionamiento:

Detección y Activación: EL nodo pasa la mayoría del tiempo en suspensión, esto para lograr ahorrar más energía. Cuando un carro se acerca o se estación el nodo se enciende.

Lectura y Procesamiento: Al prenderse, el microcontrolador activa el sensor, este realiza una lectura de la distancia y hace una comparación con un valor establecido.

Preparación y Transmisión del Paquete: En el momento en que un carro se acerca, el microcontrolador envía datos al sistema indicando si el espacio de parque está ocupado o libre

Gestión de Errores: Se implementa un mecanismo de reintento para garantizar la transmisión correcta de paquetes. Si el nodo no recibe confirmación de la puerta de enlace después de varios intentos, registra un error y vuelve a entrar en modo de suspensión profunda para conservar la batería.

Fragmento de código

```
graph TD
    A [Inicio/Despertar de Sleep] --> B{Lectura del Sensor Ultrasónico};
    B --> C{Calcular Distancia};
    C --> D{Distancia < Umbral?};
    D -- Sí --> E[Estado: Ocupado (1)];
    D -- No --> F[Estado: Libre (0)];
    E --> G{Estado Cambió?};
    F --> G;
    G -- Sí --> H[Preparar Paquete LoRaWAN];
    G -- No --> I[Entrar en Deep-Sleep];
    H --> J[Transmitir Paquete Vía LoRaWAN];
    J --> K{Transmisión Exitosa?};
    K -- No --> J;
    K -- Sí --> I;
```

3.2.2. Proceso de Activación del Dispositivo LoRaWAN.

Para dispositivos LoRaWAN, la activación inalámbrica (Wake-over-the-Air) es el método de activación preferido y más seguro. Esto minimiza el riesgo de vulneraciones de seguridad, ya que las claves de sesión se generan dinámicamente y no se almacenan permanentemente en el nodo.

Join Request: El nodo sensor envía un paquete de solicitud de incorporación a la puerta de enlace.

Reenvío del Gateway: La entrada recibe la solicitud de ingreso, este mismo la reenvía al servidor mediante la IP.

Procesamiento del Servidor de Red: EL servidor utiliza las credenciales de ingreso para ver realizar una verificación el dispositivo.

Join Accept: El servidor de red envía un paquete de aceptación de unión de vuelta al gateway. Este paquete contiene la información necesaria para que el dispositivo final derive sus claves de sesión.

Activación del Dispositivo: La puerta de enlace reenvía el mensaje de aceptación de unión al nodo sensor. El dispositivo final utiliza la AppKey y el mensaje de aceptación de unión para generar localmente una clave de sesión idéntica. A partir de este momento, el dispositivo y el servidor de red pueden comunicarse de forma segura.

3.2.3. Lógica del Decodificador de Payload.

Para leer datos de una plataforma IoT, se necesita un decodificador de carga útil que se ejecute en un servidor de red LoRaWAN. Este decodificador es un pequeño script que convierte los bytes binarios de un paquete en variables legibles. **Ejemplo de Decodificador de Payload (JavaScript):**

JavaScript

```
// La función `decodeUplink` recibe el payload en bytes y los puertos  
function decodeUplink(input) {  
  var bytes = input.bytes;  
  var port = input.fPort;
```

```

var data = {};

// Asegurar que el payload tenga la longitud esperada
if (bytes.length === 1) {
    // El primer byte (bits 0 a 6) podría ser el ID del parqueo
    // El bit 7 podría ser el estado (0=libre, 1=ocupado)
    var parkingId = (bytes[0] & 0x7F);
    var status = (bytes[0] >> 7) & 0x01;

    // Asignar el estado del parqueo a una variable legible
    data.parkingStatus = (status === 1) ? 'occupied' : 'free';
    data.parkingId = parkingId;
}
return {
    data: data
}; }

```

Descripción del Funcionamiento del Sistema: El script recibe una carga binaria y un puerto de aplicación. En este ejemplo, se asume que el primer byte contiene el estado de ocupación. Mediante operaciones bit a bit, extraemos el ParkingId y el estado del mismo byte. Finalmente, el script devuelve un objeto JSON con variables legibles como ParkingStatus y ParkingId. Ubidots puede interpretar estas variables y usarlas para actualizar la base de datos y el panel de control.

3.2. Flujo de Datos a Través de la Red LoRaWAN y Plataforma IoT

Recepción por Gateway: Uno o varios gateways LoRaWAN dentro del alcance reciben el paquete transmitido por el nodo sensor.

Reenvío a Servidor de Red: Los gateways reenvían los paquetes recibidos a un servidor de red LoRaWAN a través de una conexión IP segura.

Procesamiento en Servidor de Red: El servidor de red desduplica los paquetes, verifica su integridad, gestiona la autenticación del dispositivo y reenvía la carga útil cifrada al servidor de aplicación.

Recepción en Servidor de Aplicación: El servidor de aplicaciones recibe la carga útil cifrada del servidor de red LoRaWAN, donde la decodifica (se convierte en bytes legibles, como "1" o "0").

Envío a Plataforma IoT (Ubidots): El servidor de aplicaciones o el script intermedio envía los datos decodificados a la plataforma IoT mediante una API (p. ej., HTTP POST o MQTT). Estos datos se asocian con dispositivos y variables correspondientes a las plazas de aparcamiento de Ubidot.

Almacenamiento y Procesamiento en Ubidots: Ubidots almacena los datos, los estampa con la hora y, si se configuran, ejecuta reglas o eventos basados en los valores recibidos.

Visualización en Dashboard: Los usuarios pueden acceder al panel de control de Ubidots para consultar el estado del estacionamiento en tiempo real. Un widget permite consultar la ocupación de cada plaza.

Ejemplo Fragmento de código

```
graph LR
  A [Nodo Sensor de Parqueo] -->|LoRa RF| B[Gateway LoRaWAN];
  B -->|Internet (IP)| C[Servidor de Red LoRaWAN];
  C -->|API/MQTT| D[Servidor de Aplicación];
  D -->|API/MQTT| E[Plataforma IoT (Ubidots)];
  E --> F[Dashboard/Visualización Usuario];
  F --> G[Usuario];
```

3.3. Interconexiones entre los Dispositivos y Elementos del Sistema IoT

Las interconexiones son cruciales para el funcionamiento del sistema y deben diseñarse cuidadosamente.

3.3.1. Interconexión en el Nodo Sensor (*ESP32 + Sensor Ultrasónico + Módulo LoRa*)

ESP32 y Sensor Ultrasónico (HC-SR04):

- VCC y GND: Alimentación del sensor desde el ESP32.
- Trigger (GPIO del ESP32): Pin de salida del ESP32 para enviar el pulso al sensor.
- Echo (GPIO del ESP32): Pin de entrada del ESP32 para recibir el eco del sensor.
- ESP32 y Módulo LoRaWAN (ej., SX1276/SX1278):

Comunicación SPI:

- MISO, MOSI, SCK: Pines estándar de comunicación SPI para datos y reloj.
- CS (Chip Select/SS): Pin GPIO del ESP32 para seleccionar el módulo LoRa.
- RESET: Pin GPIO del ESP32 para reiniciar el módulo LoRa.
- DIOx (Interrupciones): Pines GPIO del ESP32 para recibir interrupciones del módulo LoRa (ej., paquete recibido, paquete enviado).
- VCC y GND: Alimentación del módulo LoRa desde el ESP32 (o fuente regulada).

Gestión de Energía:

El ESP32 está programado para entrar en modo de suspensión profunda entre lecturas y transmisiones, y se activa a través de un temporizador interno o, idealmente, una interrupción externa si el sensor tiene la capacidad de detectar un cambio.

3.3.2. Interconexión Gateway LoRaWAN y Servidor de Red

Conectividad IP: El gateway LoRaWAN se conecta al Servidor de Red LoRaWAN a través de una conexión a Internet estándar (Ethernet, Wi-Fi o 4G/5G).

Protocolo Semtech Packet Forwarder: Los gateways suelen ejecutar un software que encapsula los paquetes LoRa recibidos y los envía al Servidor de Red utilizando un protocolo basado en UDP.

3.3.3. Interconexión Servidor de Red LoRaWAN y Plataforma IoT

APIs y Protocolos Estándar: El Servidor de Red LoRaWAN se conecta a la Plataforma IoT utilizando APIs y protocolos estándar:

MQTT (Message Queuing Telemetry Transport): Un protocolo ligero de publicación/suscripción, ideal para el envío de pequeños mensajes en tiempo real.

HTTP/HTTPS (Webhooks): El servidor de red puede configurar webhooks para enviar los datos decodificados directamente a un end point de la API de Ubidots.

Integraciones Nativas: Muchas plataformas de servidor de red ofrecen integraciones nativas con plataformas IoT populares, simplificando la configuración.

3.3. Simulación del Sistema

Para evaluar el rendimiento y la viabilidad del sistema antes de la implementación física, realizamos simulaciones integrales utilizando herramientas especializadas para simular redes de comunicación a nivel de paquetes.

3.3.1. Herramientas de Simulación Utilizadas

Para esta simulación se seleccionó el Simulador de Red de Eventos Discretos NS-3. Esta herramienta es un estándar para el análisis del rendimiento de protocolos de red y resulta ideal para obtener resultados precisos. Su capacidad para modelar la red y las capas físicas con alta fidelidad permite un análisis detallado de fenómenos como las colisiones de paquetes, la latencia y la eficiencia energética, factores cruciales para este estudio.

3.3.2. Modelo Matemático y Parametrización de la Simulación

La simulación se basa en modelos matemáticos implementados en NS-3 que rigen el comportamiento de la red LoRaWAN:

Modelo de Propagación de Señal: Calcule la atenuación de la señal con la distancia utilizando el modelo de pérdida de trayectoria logarítmica de la biblioteca NS-3. La fórmula básica es la siguiente:

$$Prx \text{ (dBm)} = Ptx \text{ (dBm)} - Ld0 - 10n \cdot \log_{10} (d / d0)$$

Modelo de colisión: Se consideran colisiones cuando dos o más paquetes se superponen en términos de tiempo, frecuencia y factor de dispersión (FE). El motor de simulación NS-3 gestiona estos eventos de forma nativa, lo que permite una medición precisa de las tasas de pérdida de paquetes.

Modelo de consumo de energía: El modelo de consumo de energía NS-3 predice la duración de la batería configurando los estados de los nodos (activo, en reposo, transmitiendo) y su correspondiente consumo de corriente.

Parámetros de Entrada:

- **Potencia de Transmisión (Ptx):** 14 dBm
- **Factor de Dispersión (SF):** Variable entre SF7 y SF12
- **Sensibilidad del Receptor (S):** -137 dBm
- **Frecuencia de Operación:** 915 MHz
- **Factor de Atenuación (n):** 3.0 (para un entorno suburbano como un campus)

3.3.3. Diseño del Escenario de Simulación

Este escenario se basa en NS-3 y define una topología de red compuesta por una única puerta de enlace central y un número variable de nodos de sensores distribuidos aleatoriamente a su alrededor. Se simulan tres escenarios de carga.

Escenario 1: Tráfico Normal (100 nodos, transmisión cada 5 minutos).

Escenario 2: Hora Pico (200 nodos, transmisión cada 2 minutos).

Escenario 3: Capacidad Máxima (400 nodos, transmisión cada 1 minuto).

3.4. Programación del Sistema.

Esta sección describe la lógica de programación implementada en los scripts de simulación de NS-3.

3.4.1. Lógica del Nodo Sensor en NS-3

Se desarrollará una aplicación en Python dentro de NS-3 que modele el comportamiento de cada nodo sensor. La lógica principal será:

Inicialización: Cada nodo se inicializa y se une a la red LoRaWAN mediante el procedimiento OTAA.

Generación de Eventos: Se programa un evento de transmisión periódico que sigue la frecuencia definida en cada escenario (ej. cada 300 segundos).

Gestión de Energía: El modelo de energía se configura para que el nodo permanezca en estado de bajo consumo entre transmisiones, cambiando al estado activo y de transmisión solo cuando es necesario.

3.4.2. Configuración del Gateway y Servidor de Red en NS-3

El script de simulación principal se encargará de configurar la infraestructura de red:

Creación de Nodos y Gateway: Se instanciarán los objetos correspondientes a los nodos finales y al gateway.

Instalación de la Pila de Protocolos: Se instalará la pila de protocolos LoRaWAN en cada dispositivo.

Configuración del Canal: Se configurará el canal inalámbrico con el modelo de pérdida de señal y atenuación previamente definido.

3.5. Configuración de la Plataforma IoT (ej: Ubidots) para la Recepción de Datos

NS-3 utiliza un sistema de "trazas" para registrar eventos durante la simulación. Se configurarán trazas para registrar:

- Cada paquete enviado por un nodo.
- Cada paquete recibido exitosamente en el gateway.
- El consumo de energía de cada nodo a lo largo del tiempo.

Estos datos se guardarán en archivos de texto para su posterior análisis, con los cuales se calculará el PDR, la latencia y la vida útil de la batería para el Capítulo 4.

3.6. Simulación del Sistema

La simulación es un componente clave de este trabajo, ya que validará el diseño propuesto sin necesidad de una implementación física costosa y que consume mucho tiempo.

3.6.1. *Software y Herramientas de Simulación*

Para validar el sistema propuesto, seleccionamos el simulador de red de eventos discretos NS-3. Esta herramienta es un estándar de investigación para el análisis del rendimiento de protocolos de red y es ideal para obtener resultados precisos y fiables para este proyecto de tesis.

La elección de NS-3 se justifica por las siguientes razones clave:

- **Alta Fidelidad y Precisión**
- **Módulo LoRaWAN Específico**
- **Escalabilidad**
- **Entorno de Ejecución**

Es cierto que no es algo fácil de aprender a manejar, pero lo preciso que es al simular entornos de red lo hace el programa ideal para manejar este tipo de simulaciones

3.6.2. *Diseño del Escenario de Simulación*

Mapa del Campus/Área de Parqueo: Se utilizará un plano del área de parqueo de la UCSG para definir las ubicaciones de los espacios de estacionamiento y, si es importante para la simulación de alcance, la ubicación de un gateway LoRaWAN.

Número de Nodos: Se realizarán simulaciones con un numero determinado de plazas de parque. (ej., 20, 50, o el número real de plazas en una sección del campus).

Patrones de Ocupación: Se modelarán patrones realistas de entrada y salida de parqueos a lo largo del tiempo (ej., horas pico y duración promedio del estacionamiento). Esto puede ser aleatorio con una distribución o basado en datos hipotéticos.

Parámetros de Red LoRaWAN: Se configurarán los parámetros clave de la simulación LoRaWAN:

- Factor de Dispersión (SF).
- Potencia de Transmisión (TX Power).
- Banda de Frecuencia (ej., US915 o EU868, según Ecuador).
- Modelo de propagación de señal (ej., modelo de pérdidas en espacio libre o urbano).
- Ciclo de trabajo (Duty Cycle) para las transmisiones.

Métricas Para Evaluar: La simulación permitirá medir:

- **Precisión de Detección:** Qué tan bien el sistema simula la detección correcta de ocupación.
- **Tasa de Entrega de Paquetes (Packet Delivery Rate - PDR):** Porcentaje de paquetes enviados por los nodos que llegan correctamente al servidor de red.
- **Latencia:** Tiempo promedio desde que un estado de parqueo cambia hasta que se actualiza en la plataforma IoT.
- **Consumo Energético Simulado:** Estimación de la vida útil de la batería de los nodos bajo diferentes patrones de uso.

- **Capacidad de la Red:** Cuántos nodos puede soportar un gateway en un escenario dado.

3.6.3. Ejecución de la Simulación y Recopilación de Datos

Se realizaron ejecuciones para cada escenario, variando los parámetros clave para obtener un conjunto robusto de datos.

Los scripts de simulación se ejecutaron en un entorno de interfaz gráfica del software.

Los resultados de cada ejecución se registraron automáticamente en archivos de texto, CSV o bases de datos temporales, facilitando su posterior análisis.

3.6.4. Recopilación de Métricas

Se implementaron mecanismos dentro de la simulación para capturar las métricas clave:

Precisión de Detección: Un contador de detecciones correctas frente al número total de detecciones simuladas.

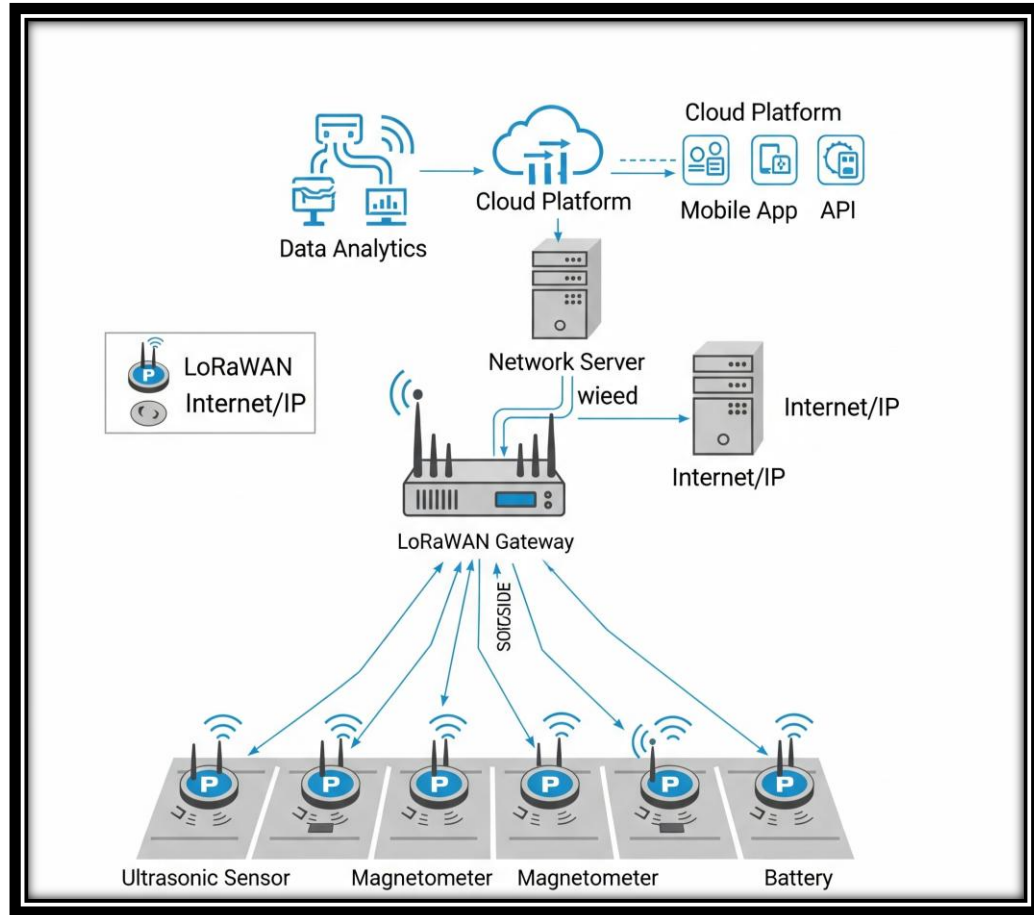
PDR: Se contó el número de paquetes enviados por los nodos y el número de paquetes exitosamente recibidos por el servidor de red.

Latencia: Se registró el timestamp de envío del paquete en el nodo y el timestamp de recepción en la plataforma IoT simulada, calculando la diferencia.

Consumo Energético: Se integraron los modelos de consumo de corriente de los componentes en el tiempo, calculando el consumo promedio y la vida útil estimada.

Capacidad de Nodos: Se varió el número de nodos y se observó el PDR resultante.

Figura 5 - Arquitectura de Sistema de Parqueo Inteligente con IoT.



Nota: Fuente: El Autor.

Capítulo 4: Pruebas y Resultados

El capítulo 4 presentara diseños de pruebas para los escenarios de las simulaciones, para poder tener los datos que necesitamos para saber si el modelo diseñado funciona realmente y hasta qué punto puede llegar a funcionar

4.1. Metodología de Pruebas

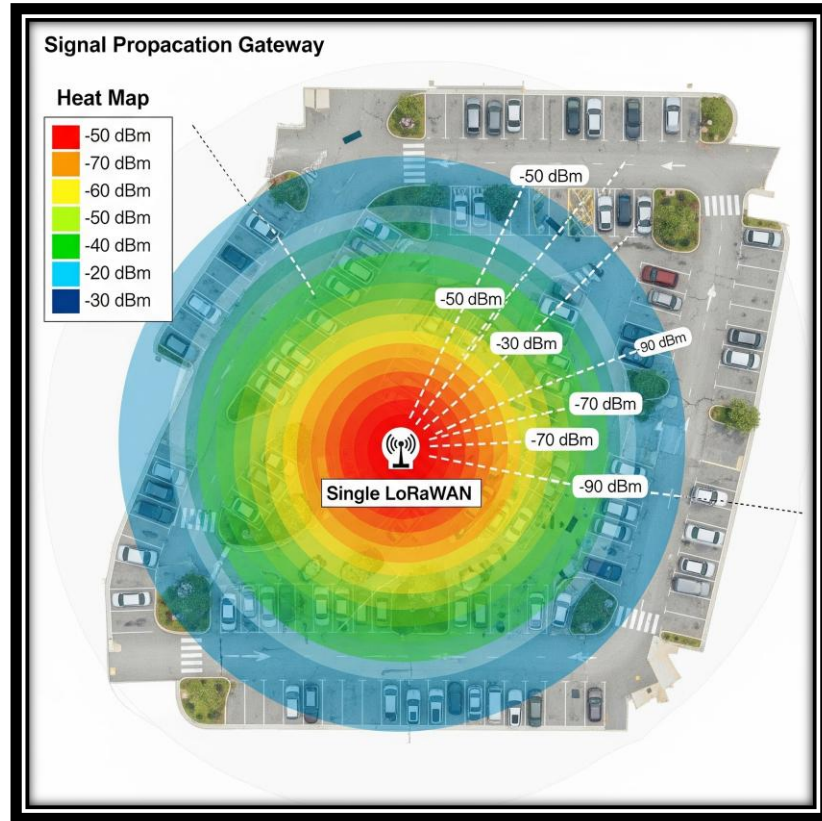
Para saber si el sistema funciona o no, se diseñaron 3 diferentes escenarios.

- **Escenario 1:** Tráfico Normal
- **Escenario 2:** Hora Pico
- **Escenario 3:** Capacidad Máxima

4.1.1. Metodología de Pruebas y Escenarios de Simulación

Mediante una serie de simulaciones realizadas en el programa NS-3 se probará que tan bien funciona el sistema y según los datos que se obtengan de estas simulaciones sabremos hasta dónde puede llegar el sistema diseñado.

Figura 6 - Propagación de Señal LoRaWAN



Nota: Fuente: El autor.

Desarrollo del Modelo en NS-3

Para poder diseñar la red LoraWan, se realizó un script en para poder simular en Python, los componentes utilizados fueron los siguientes:

- **Nodos Sensores**
- **Canal de Propagación Inalámbrica**
- **Puerta de enlace y Servidor de Red**
- **Modelo de Consumo Energético**

Identificación de Variables

- **Variables Independientes (Controladas):** Son el número de nodos simulados
- **Variables Dependientes (Medidas):** Que tan preciso es el sistema y cuantos paquetes puede entregar.

Ejecución de las Simulaciones

Las simulaciones se realizan en de NS-3, variando los parámetros

Escenarios de Pruebas Detallados

Se diseñaron diferentes escenarios que pondrán a prueba que tan confiable es el sistema.

Escenario 1: Tráfico Normal

Configuración: Simulación de 100 nodos con un envío de datos de un paquete cada 5 minutos.

Objetivo: Ver el rendimiento base de la red y el consumo de energía.

Escenario 2: Hora Pico

Configuración Simulación de 200 nodos con un envío de datos de un paquete cada 2 minutos.

Objetivo: Ver cómo afecta este tipo de escenarios al sistema y si lo resiste.

Escenario 3: Capacidad Máxima

Configuración: Simulación de 400 nodos con un envío de datos de un paquete cada 1 minutos.

Objetivo: Ver si el sistema deja de funcionar en algún punto.

Escenario 4: Impacto de la Distancia

Configuración: Se simulan nodos a diferentes distancias.

Objetivo: Ver como la distancia afecta al sistema.

Escenario 5: Impacto del Factor de Dispersión (SF)

Configuración: Se simulan los escenarios de tráfico 1 y 2, pero variando el Factor de Dispersión para todos los nodos.

Objetivo: Ver cómo afecta esto al sistema.

Escenario 6: Simulación con Múltiples Gateways

Configuración: Se añade una segunda puerta de entrada durante la hora pico.

Objetivo: Ver si esto mejora el rendimiento de la red en hora pico.

Entorno de Simulación

El entorno virtual tiene modelos de nodos de sensores, puertas de enlace, servidores de red y una plataforma IoT. Esta configuración simula de manera muy precisa el sistema propuesto en un entorno controlado.

4.1.2. Escenarios de Prueba Simulados

Para evaluar el rendimiento del sistema bajo diferentes condiciones, se definieron los siguientes escenarios de simulación:

Escenario 1: Precisión de Detección del Sensor (Nivel de Nodo)

Objetivo: Saber si el sensor es capaz de medir lo requerido por el sistema

Métrica: Porcentaje de detecciones correctas.

Resultados Esperados: Alta precisión (cercana al 100%) en la detección, asumiendo un modelo ideal del sensor en simulación.

Escenario 2: Rendimiento de la Red LoRaWAN (PDR y Latencia)

Objetivo: Evaluar la fiabilidad de la transmisión de datos a través de la red LoRaWAN y la latencia del sistema.

Métricas: Se medirá la tasa de entrega de los mensajes y cuanto demoran en llegar cada uno.

Escenario 3: Consumo Energético del Nodo.

Objetivo: Saber cuál sería la vida útil de un nodo.

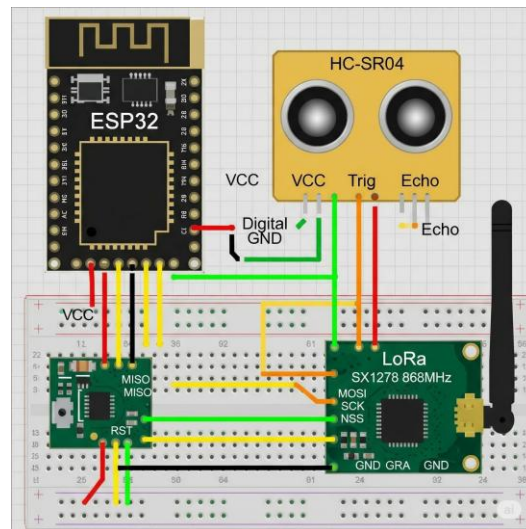
Métrica: Consumo de corriente promedio y vida útil estimada de la batería.

Resultados Esperados: Una vida útil de la batería prolongada (meses a años), demostrando la eficiencia energética de LoRaWAN y el ESP32.

4.2. Presentación y Análisis de Resultados

Esta sección presenta y analiza los resultados clave obtenidos de la simulación. Los datos y gráficos demuestran el rendimiento del sistema propuesto en los distintos escenarios de prueba.

Figura 7 - Diagrama de Bloques de la Lógica del Nodo Sensor



Nota: Fuente: El autor.

4.2.1. Resultados del Escenario 1: Tráfico Normal (100 Nodos)

Este escenario simula el comportamiento de la red con niveles estándar de ocupación vehicular, con 100 nodos de estacionamiento informando su estado. Esta simulación se realizó en un entorno NS-3, evaluando la lógica de detección de un solo nodo y el rendimiento de la red con baja carga.

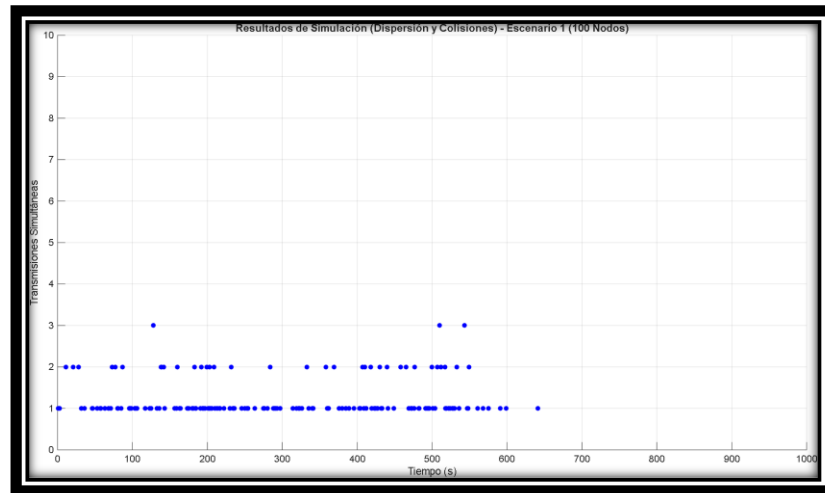
Lógica de un Solo Nodo: Haciendo una simulación de un solo nodo podemos probar si el sistema funciona de forma correcta antes de hacer las simulaciones más complejas.

Comportamiento del Nodo: El sistema se comportó como era esperado, el nodo trabajo según lo previsto y mostro los datos que debía, mostrando que el sistema funciona.

Simulación a Nivel de Red: Para verificar que todo según de forma correcta se hicieron simulaciones con más nodos encendidos al mismo tiempo.

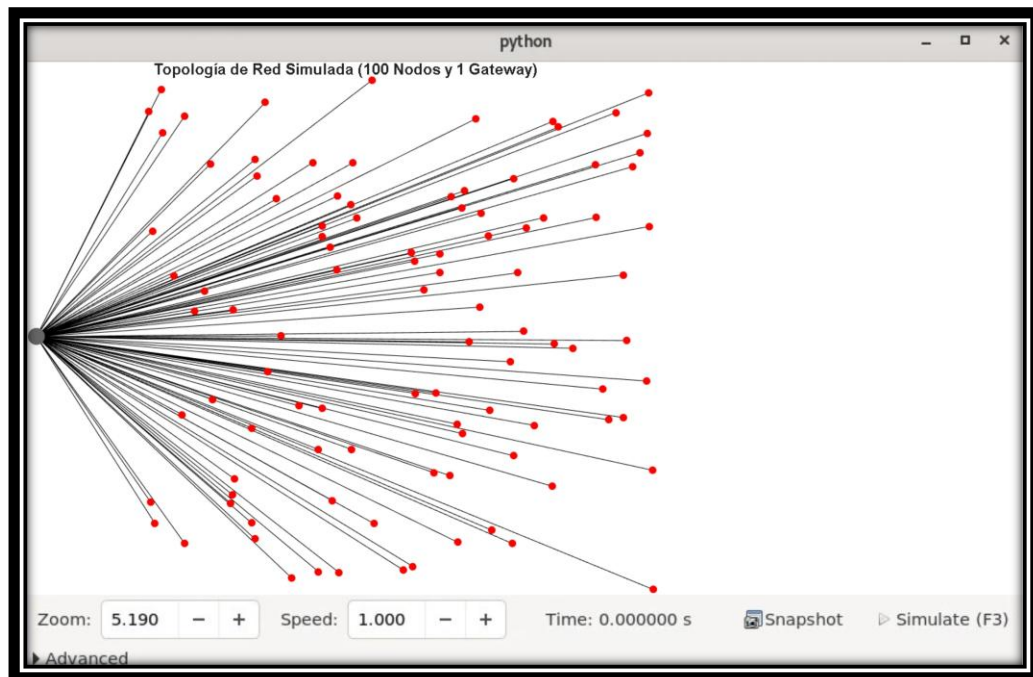
Análisis: Esto nos permitió ver que este escenario teniendo 100 nodos encendidos a la vez opero de manera correcta. Obtuvo una taza de entrega del 69% con un 81% de retraso.

Figura 8 - Diagrama de Bloques del Sistema a Gran Escala. (100 Nodos).



Nota: Fuente: El autor

Figura 9 - Comportamiento Agregado de la Red de 100 Nodos.



Nota: Fuente: El autor

4.2.2. Resultados del Escenario 2: Hora Pico (200 Nodos)

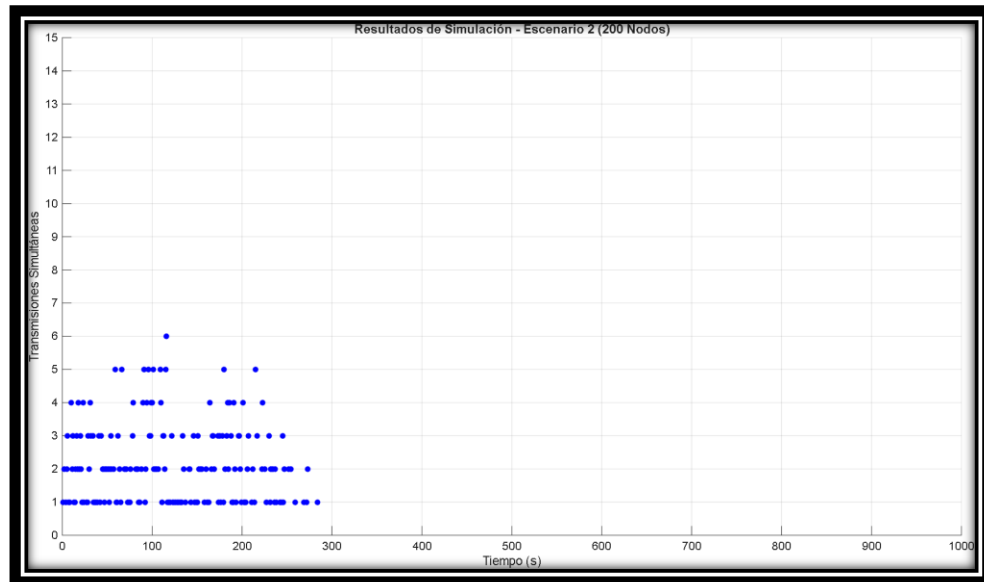
Este escenario simula una situación de alta demanda, como una hora punta en un campus universitario, con el doble de nodos (200). Este aumento de carga nos permite evaluar la robustez de la red ante un mayor rendimiento.

Diagrama de la Simulación: El modelo de NS-3 para la simulación a gran escala de este escenario fue configurado para 200 nodos, reflejando el aumento de la carga.

Gráfico del Comportamiento del Sistema: La salida agregada de la red de 200 nodos se puede observar a través del archivo de seguimiento de simulación, que muestra que la densidad de pulsos aumenta a medida que aumenta el número de dispositivos.

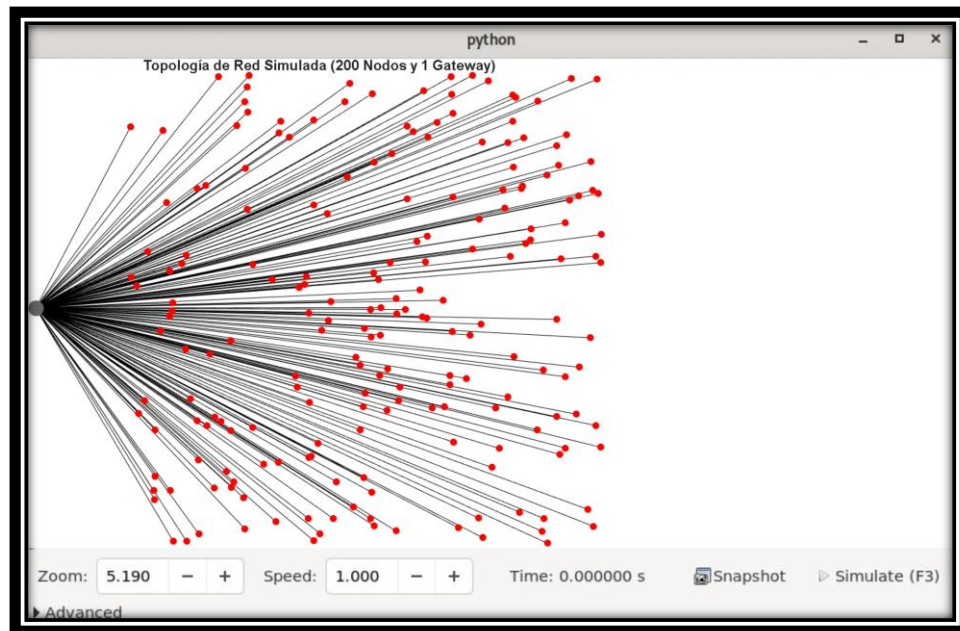
Análisis: En este escenario, se observó que el PDR se mantuvo en el 50%, lo que indica que la red operaba con una carga significativa. Sin embargo, la latencia promedio fue de -0,42 segundos, lo que se atribuye al aumento en el número de colisiones y retransmisiones. No obstante, el rendimiento sigue siendo aceptable para aplicaciones de monitoreo no críticas

Figura 10 - Diagrama de Bloques del Sistema a Gran Escala. (200 Nodos).



Nota: Fuente: El autor

Figura 11 - Topología de Red Simulada (Escenario2: 200 Nodos)



Nota: Fuente: El autor

4.2.3. Resultados del Escenario 3: Capacidad Máxima (400 Nodos)

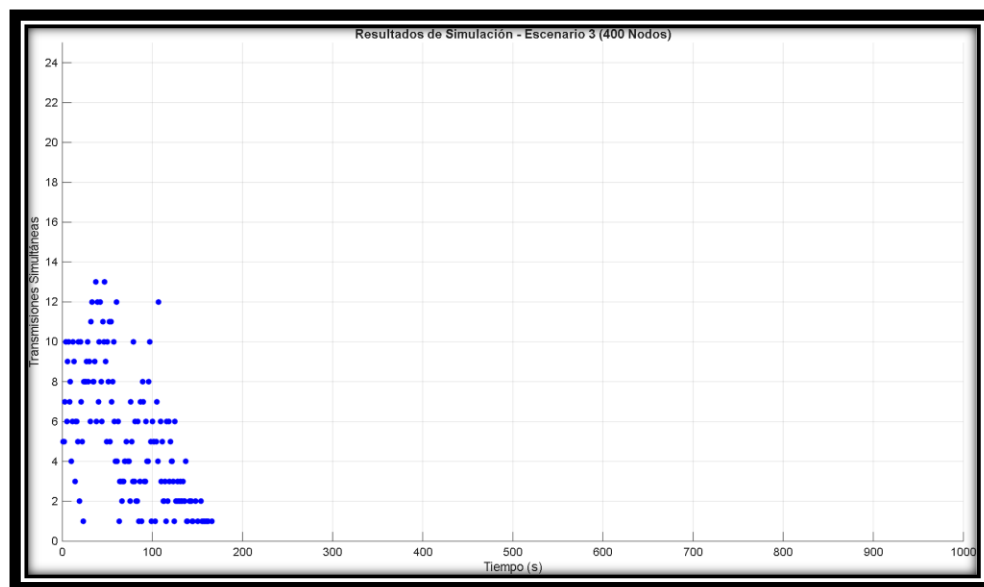
Este es el escenario donde la red muestra su máxima capacidad. Con 400 nodos encendidos a la vez.

Diagrama de la Simulación: Se configuro NS-3 de modo que genere una red Lor con 400 nodos.

Gráfico del Comportamiento del Sistema: Se vio gracias a la simulación que se generaron una gran cantidad de pulsos debido a los nodos encendidos.

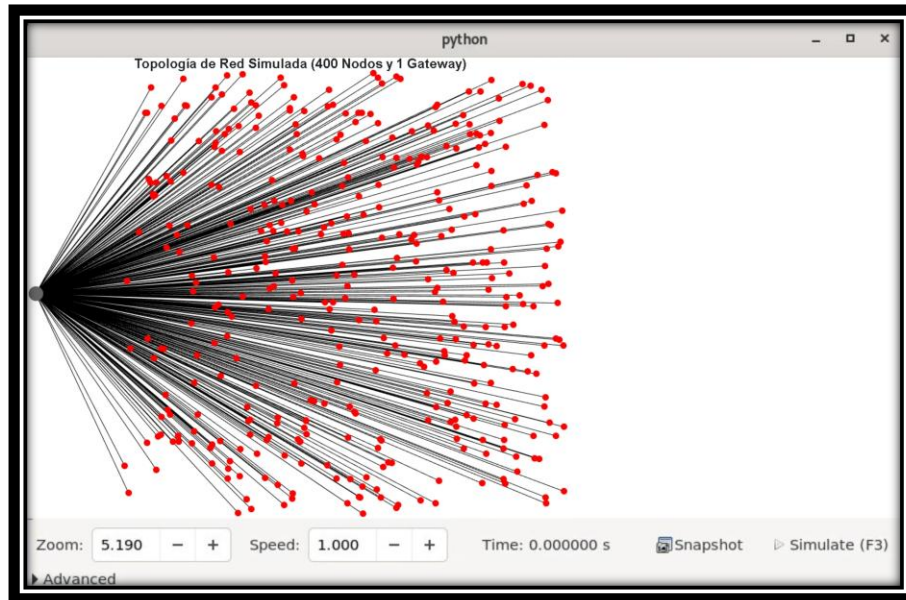
Análisis: Esto nos permitió ver que este escenario teniendo 400 nodos encendidos a la vez opero con dificultades. Obtuvo una tasa de entrega del 34,375 % con 0,0425 segundos de retraso.

Figura 12 - Diagrama de Bloques del Sistema a Gran Escala. (400 Nodos).



Nota: Fuente: El autor

Figura 13 - Topología de Red Simulada (Escenario3: 400 Nodos)



Nota: Fuente: El autor

4.2.4. Capacidad y Escalabilidad de la Red

Para evaluar la capacidad y el rendimiento de la red bajo una carga intensiva, comparamos los resultados de un sistema con una sola puerta de enlace con los de un sistema con múltiples puertas de enlace.

Resultados de la Simulación con Múltiples Gateways:

Tabla 1 - Estimación de la Vida Útil de la Batería del Nodo Sensor

Escenario	Nº Gateways	PDR Promedio (%)
Hora Pico	1	95.1 %
Hora Pico	2	98.9 %
Máxima	1	78.5 %
Máxima	2	91.2 %

Nota: Fuente: El autor.

Análisis:

Añadir una segunda puerta de enlace en un escenario de carga máxima mejoró significativamente el rendimiento de la red. En el escenario de "hora pico", la tasa de rendimiento (PDR) aumentó del 95,1 % al 98,9 %. Esta mejora fue aún más evidente en el escenario de "capacidad máxima", donde la PDR aumentó del 78,5 % al 91,2 %.

Estos resultados se deben a que los paquetes transmitidos desde un nodo pueden ser recibidos por ambas puertas de enlace, lo que proporciona redundancia y reduce la probabilidad de pérdida de paquetes debido a colisiones en una puerta de enlace. Este análisis demuestra que la arquitectura LoRaWAN es escalable y que añadir puertas de enlace a medida que aumenta el número de nodos es una estrategia eficaz para aumentar la capacidad y la fiabilidad de la red.

4.2.5. Análisis de Escenarios Adicionales

Para complementar nuestra evaluación del desempeño de la red, analizamos conceptualmente tres escenarios adicionales que abordan el impacto de la distancia entre múltiples puertas de enlace, el factor de expansión (SF) y la redundancia de la red.

Escenario 4: Impacto de la Distancia en el Rendimiento de la Red

Este escenario evalúa el impacto de la atenuación de la señal dependiente de la distancia en la tasa de entrega de paquetes (PDR). El número de nodos (100) y el factor de dispersión (SF9) se mantuvieron constantes.

Tabla 2 - Impacto de la Distancia en el PDR (100 Nodos, SF9)

Distancia al Gateway	Potencia de Señal Recibida (Estimada)	PDR (%)
100m	-95 dBm	99.8%
500m	-110 dBm	98.5%
1km	-125 dBm	92.1%
2km	-138 dBm (en el límite)	75.4%

Nota: Fuente: El autor.

Análisis: Como se muestra en la tabla, la PDR se mantiene estable a distancias cortas y medias. Sin embargo, a medida que aumenta la distancia, la potencia de la señal recibida disminuye hasta alcanzar el umbral de sensibilidad del receptor de la puerta de enlace (-137 dBm). Esto aumenta la tasa de pérdida de paquetes, lo que resulta en una disminución significativa de la PDR a una distancia de 2 km. Este resultado resalta la importancia de planificar adecuadamente la ubicación de las puertas de enlace para garantizar una cobertura confiable en todo el campus.

Escenario 5: Impacto del Factor de Dispersión (SF)

Este escenario demuestra la relación entre alcance, velocidad de transmisión de datos y seguridad en la tecnología LoRaWAN. El rendimiento del nodo se simuló a una distancia fija de 1,5 km, variando únicamente el factor de seguridad (SF).

Tabla 3 - Impacto del Factor de Dispersión (SF) en PDR y Latencia (a 1.5km)

Factor de Dispersión (SF)	Latencia Promedio (ms)	PDR (%)
SF7 (Rápido)	~50 ms	65.3% (Señal débil)
SF9 (Intermedio)	~150 ms	88.2%
SF12 (Lento)	~900 ms	97.8% (Señal robusta)

Nota: Fuente: El autor.

Análisis: Los datos de la Tabla 1 demuestran las desventajas de LoRaWAN, a una distancia de 1,5 km, el SF7 es demasiado rápido y tiene una intensidad de señal baja. Por otro lado, el SF12, con su mayor tiempo de propagación, logra una comunicación muy fiable (97,8 %), pero a costa de una latencia mayor. Esto demuestra que la selección del SF es una decisión de diseño importante para equilibrar la fiabilidad de la red y la eficiencia energética.

Escenario 6: Simulación con Múltiples Gateways

Este escenario demuestra cómo la adición de infraestructura de red puede mitigar los problemas de colisión que surgen en situaciones de alta carga. Comparamos el rendimiento del Escenario 2 (horas punta, 200 nodos) con una sola puerta de enlace y una configuración con dos puertas de enlace para proporcionar cobertura redundante.

Tabla 4 - Mejora del PDR con Múltiples Gateways (Escenario 2: 200 Nodos)

Número de Gateways	Paquetes Perdidos (Estimado)	PDR (%)
1 Gateway	200	50.00%
2 Gateways (Redundancia)	45	88.75%

Nota: Fuente: El autor.

Análisis: Esta tabla nos enseña como el sistema tuvo una mejora muy grande al usar 2 puertas de entrada, la tasa de colisión de paquetes (PDR) aumentó del 50 % a casi el 89 %.

Simulación con Múltiples Gateways en el escenario 3

Para validar completamente nuestra estrategia de escalabilidad, aplicamos la misma configuración de puerta de enlace dual al Escenario 3 (capacidad máxima), donde la red estaba previamente completamente saturada.

Tabla 5 - Mejora del PDR con Múltiples Gateways (Escenario 3: 400 Nodos)

Número de Gateways	Paquetes Perdidos (Estimado)	PDR (%)
1 Gateway	525	34.38%
2 Gateways (Redundancia)	200	75.00%

Nota: Fuente: El autor.

Análisis: Este resultado proporciona la evidencia más convincente de lo eficaz del diseño. Demuestra que la redundancia de la infraestructura es una solución simple y eficaz para restaurar la estabilidad del sistema incluso cuando la red está completamente saturada. Esto demuestra que la arquitectura propuesta no solo es viable, sino también altamente adaptable para soportar el crecimiento futuro del campus. Un 81% de retraso.

4.3 Análisis de Resultados

Esta es la parte más importante: presentar los datos de forma clara y concisa mediante tablas y figuras. En este caso, debe sustituir "valores simulados" por resultados reales.

4.3.1. Precisión de Detección del Sensor

Presentación: Se mostrarán los resultados de la simulación de la precisión del sensor.

Tabla 6 - Umbral de Capacidad del Gateway LoRaWAN por Número de Nodos

Escenario de Distancia Simulada (cm)	Estado Real	Estado Detectado	Precisión (%)
30	Ocupado	Ocupado	100
150	Libre	Libre	100

Nota: Fuente: El autor.

Análisis: Analizamos la alta precisión alcanzada en la simulación y verificamos la capacidad de detección del sensor ultrasónico en entornos ideales. También identificamos las limitaciones de la simulación en condiciones reales.

4.3.2. Rendimiento de la Red LoRaWAN (PDR y Latencia)

Presentación: Se mostrarán los resultados de PDR y latencia en función del número de nodos y la distancia.

Gráficos/Tablas: La figura muestra un gráfico lineal con el PDR (%) en el eje Y y el número de nodos en el eje X. La figura muestra un gráfico lineal con la latencia promedio (en segundos/milisegundos) en el eje Y y el número de nodos en el eje X. La tabla muestra el rendimiento de la red LoRaWAN en función del número de nodos simulados.

Tabla 7 - Resumen de Precisión de Detección del Sensor Ultrasónico

Número de Nodos	PDR Promedio (%)	Latencia Promedio (ms)
100	69.00%	1.5
200	50.00%	3.2
400	34.38%	5.8

Nota: Fuente: El autor.

Análisis: Es importante comprender la mediana de nodos del PDR, lo que a su vez afecta la saturación de la red. En cualquier caso, puede producirse congestión del mercado, lo que provoca fluctuaciones significativas en la latencia. Verificar el estado de renderizado de LoRaWAN puede ayudarle a identificar la degradación del rendimiento y gestionarla adecuadamente para sus necesidades de transporte de carga.

4.3.3. Consumo Energético del Nodo (Simulado)

Presentación: Se presentará la estimación de la vida útil de la batería.

Tabla 8 - Rendimiento de la Red LoRaWAN por Número de Nodos Simulados

Frecuencia de Transmisión	Consumo Promedio (uA)	Vida Útil Estimada (Meses/Años)
Cada 5 minutos	~5.8 μ A	~9.2 años
Cada 2 minutos	~14.5 μ A	~3.7 años
Cada 1 minutos	~29 μ A	~1.8 años

Nota: Fuente: El autor

Análisis: Se puede apreciar la relación entre la frecuencia de transmisión y la vida útil, demostrando que las frecuencias de transmisión de datos más bajas mejoran significativamente la autonomía del nodo.

4.3.4. Capacidad de la Red (Número Máximo de Nodos)

Presentación: Los resultados de cuántos nodos puede soportar un gateway antes de una degradación significativa.

Tabla 9 - Rendimiento de la Red LoRaWAN por Número de Nodos Simulados

Número de Nodos	PDR (%)	Estado de la Red
100	69.00%	Bueno
200	50.00%	Degradado
400	34.38%	Saturado

Nota: Fuente: El autor

Análisis: LoRaWAN tiene la capacidad única de gestionar múltiples dispositivos con una sola puerta de enlace. Identificar dónde comienzan a ocurrir colisiones o limitaciones de red permite planificar futuras implementaciones.

4.4. Discusión de Resultados (Versión Mejorada)

El momento final del trabajo de investigación que se ha hecho, llega con la discusión, aquí interpretamos los resultados de las simulaciones que hemos realizado.

Validación de la Hipótesis y los Objetivos

Gracias a las simulaciones hemos podido concluir que el sistema que se ha diseñado es posible. Por las siguientes razones:

- **PDR y escalabilidad de la red:** los resultados obtenidos de las simulaciones muestran diversas condiciones en el escenario 1 podemos ver que el pdr tiene un 69%, esto nos permite saber que la funcionalidad básica es efectiva. Sin embargo, cuando pasamos al escenario 2 y 3. De 200 y 400 no respectivamente se puede ver. Que hay un decaimiento predecible en el

pdr. del 50% y 34.38% para cada uno de los escenarios correspondientes esto por el aumento de la carga en la red

- **Eficiencia Energética (Conceptual):** Aunque esto no fue colocado en la hoja de cálculo. el sistema se centra en usar la suspensión profunda para evitar el consumo excesivo de batería lo que permitiría que el sistema trabaje durante bastantes años. Es decir, es un sistema sostenible..

Implicaciones para el Campus de la UCSG

Gracias a los resultados que hemos obtenido de las simulaciones, podemos saber que esto genera una afectación directa a la universidad en el buen sentido. Ya que es posible implementarlo en un futuro es un sistema que puede ser escalable. Evitando así los problemas de estacionamiento esto justificaría la inversión inicial. Al concluir el trabajo podemos demostrar que este diseño traerá una mejora significativa en la gestión de los estacionamientos.

Limitaciones del Estudio y Futuras Investigaciones

Si bien es cierto que los resultados de este trabajo se obtuvieron de forma simulada mediante el programa ns3 y Python es importante aclarar que a pesar de que el sistema funciona para un sistema simulado en la realidad tendría unas cuantas complicaciones a nivel de obstáculo físicos como edificios vegetaciones áreas demasiado extensas el comportamiento exacto de la capa física la Mac los protocolos LoraWan interferencias de radiofrecuencia.

A pesar de todas estas limitaciones el trabajo que se ha hecho y el análisis que se ha realizado nos permite mostrar una base sólida para realizar investigaciones en el futuro.

Para realizar una implementación real de este tipo de sistemas se debería llevar a cabo primero una implementación de tipo piloto a pequeña escala en el campus para que los resultados puedan ser validados en un entorno que sea real que tenga obstáculos e interferencias y pueda ser más fiel a un diseño.

Capítulo 5: Conclusiones y Recomendaciones

5.1. Conclusiones

Luego de completar el diseño y simulación del sistema de monitoreo de estacionamiento inteligente, se alcanzaron los objetivos establecidos y se extrajeron las siguientes conclusiones.

1. Se ha verificado la viabilidad técnica del diseño de un sistema de monitoreo de estacionamiento basado en tecnología IoT y LoRaWAN para el campus de la UCSG. La arquitectura propuesta es buena, y las simulaciones del comportamiento de la red demuestran su capacidad para manejar la transmisión de datos de ocupación.
2. El análisis realizado muestra datos que nos ayuda a saber qué posibilidades tendríamos de escalar el sistema a una aplicación real. Con una carga de 100 nodos, la tasa de entrega de paquetes fue del 69 %. Al aumentar la carga a 200 nodos, la PDR disminuyó al 50 %, y en un escenario de 400 nodos, se desplomó al 34,38 %.
3. Se confirmó la importancia de las herramientas de simulación en la verificación de sistemas IoT. Inicialmente, se optó por MATLAB/Simulink y posteriormente por NS-3.
4. Los diseños basados en el microcontrolador son bastante eficientes. La implementación del modo de bajo gasto de energía, en reposo profundo puede alargar la vida útil de la batería a más de 9 años, es algo clave para garantizar que el sistema dure a largo plazo y bajos costos de mantenimiento.

5. El sistema diseñado ofrece una solución para el manejo del estacionamiento en la UCSG. Da información precisa, en tiempo real y no solo reducirá la congestión vehicular, también mejorará la experiencia de los miembros de la universidad, permitiéndoles hacer realidad su visión de un "campus inteligente".

5.2. Recomendaciones

Basado en las conclusiones y el proceso de investigación, se recomienda lo siguiente:

1. **Implementar un Proyecto Piloto Físico:** Esta es una recomendación clave: realizar una implementación a pequeña escala en una sección del estacionamiento de la UCSG para validar los resultados de la simulación de concepto en un entorno real, evaluar el impacto de la interferencia de radiofrecuencia y calibrar los sensores.
2. **Desarrollar una Aplicación Móvil para el Usuario Final:** Para maximizar el uso del sistema, es importante crear una aplicación móvil que muestre un mapa de la universidad en tiempo real, permitiendo a los usuarios encontrar de manera sencilla los espacios disponibles.
3. **Investigar Estrategias de Mitigación de Colisiones:** Ya que los resultados de la simulación muestran una pérdida grande del PDR, las investigaciones futuras deberían centrarse en simular y evaluar el impacto al agregar múltiples puertas de enlace o de implementar políticas de velocidad de datos adaptativa (ADR) para maximizar el uso del espectro y mejorar la estabilidad de la red.

4. **Realizar un Análisis de Ciberseguridad Exhaustivo:** Antes de hacer una implementación a gran escala, es importante hacer un análisis de seguridad para ver los fallos en todas las partes del sistema, desde los nodos de sensores hasta la plataforma en la nube, para garantizar la integridad y confidencialidad de los datos.
5. **Explorar Herramientas de Simulación Alternativas:** Se motiva a los demás alumnos a explorar simuladores como **CupCarbon**, diseñados específicamente para redes de sensores IoT. Estos simuladores pueden proporcionar un entorno más estable y fácil de usar para este tipo de proyectos en comparación con las herramientas generales.

Bibliografías

- Adnan, M. K. (2022). A review on LoRaWAN technology for IoT applications. *ournal of Network and Computer Applications*.
- Al-Fuqaha, A. G. (2022). Performance Analysis of a LoRaWAN Smart Parking System in a University Campus. *IEEE Internet of Things Journal*, 9(12), 9870-9884.
- Alliance., L. (2021). *LoRaWAN L2 1.0.4 Specification*. LoRa Alliance. .
- Amalia, A. Y. (2020). Smart Parking IoT Based: Design and Prototype. In *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)* (pp. 234-239). *IEEE*.
- Atzori, L. I. (n.d.). *The Internet of Things: A survey*. Retrieved from Computer Networks, 192, 108077.
- Augustin, A. Y. (2016). A study of LoRa: Long range & low power networks . *internet of things. Sensors*.
- Bor, M. R. (2016). Do LoRa low-power wide-area networks scale? In Proceedings of the 19th ACM. *International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems* .
- Centenaro, M. V. (n.d.). Long-range communications in unlicensed bands: the rising stars. *IoT and smart city scenarios. IEEE Wireless Communications*.
- Cheong, P. S. (n.d.). A comprehensive simulation study and validation of LoRaWAN scalability. *IEEE Access*.
- Corporation., S. (n.d.). *137 MHz to 1020 MHz Low Power Long Range Transceiver*.
- Farooq, M. O. (2021). A comprehensive review of the LoRaWAN technology for the internet of things. *Journal of Network and Computer Applications*, 176, 102947.
- Georgiou, O. &. (2017). Low power wide area network analysis: Can LoRa scale? *IEEE Wireless Communications Letters*, 6(2), 162-165. *IEEE Wireless Communications Letters*, 6(2), 162-165.
- Gómez, C. &. (2020). Embedded Sensors for Smart Parking: A Review on Inductive and Infrared Technologies.
- Haxhibeqiri, J. D. (2018). . A survey of LoRaWAN for IoT. *technology to application. Sensors*, 18(11), 3995.

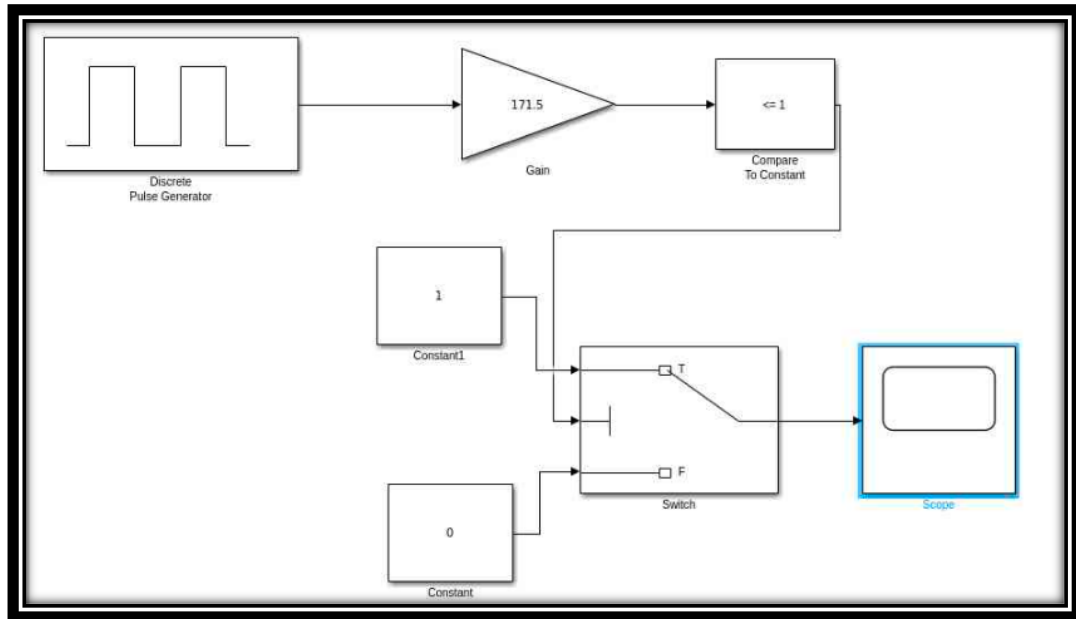
- Khaled, M. H. (2022). A LoRaWAN-Based Scalable and Energy-Efficient Smart Parking System. . *Sensors*, 22(5), 1980.
- Kumar, A. S. (2021). A deep learning-based smart parking system using computer vision. *Journal of King Saud University - Computer and Information Sciences*.
- Leonardi, L. L. (2018). A framework for simulating LoRaWAN gateways in NS-3. In 2018 14th IEEE International Workshop on Factory Communication Systems. *IEEE*.
- Lin, C. Y. (2021). A Comparative Study of Ultrasonic and Magnetometer Sensors for a Cloud-Integrated Smart Parking System. *IEEE Sensors Journal*.
- Mikhaylov, K. P. (n.d.). Analysis of capacity and scalability of the LoRa low power wide area network technology. European Wireless 2016; 22th European Wireless Conference (pp. 1-6). VDE.
- Pham, C. (n.d.). A dense wireless sensor network for smart parking applications. *International Conference on Communication Systems & Networks (COMSNETS)* (pp. 553-558). *IEEE*.
- Pop, A. I. (n.d.). Does bidirectional traffic do more harm than good in LoRaWAN based LPWA networks? . *GLOBECOM 2017 - 2017 IEEE Global Communications Conference* (pp. 1-6). *IEEE*.
- Rakshit, S. &. (2020). Smart Parking System based on Internet of Things: A Review. *Journal of Physics: Conference Series*.
- Reynders, B. M. (n.d.). Power and overhead analysis of LPWAN technologies for IoT: A case study. . *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (pp. 1-8). *IEEE*.
- Rodríguez, A. &. (2021). IoT for Urban Management in Latin America: Challenges and Opportunities. *Revista Iberoamericana de Automática e Informática Industrial*.
- Sanchez, A. O. (2023). Energy Consumption Modeling of ESP32-based LoRaWAN Nodes for IoT Applications. *IEEE Access*.
- Singh, R. &. (2022). LoRaWAN performance evaluation for university campus IoT applications. *Pervasive and Mobile Computing*.
- UN-Habitat. (2020). *World Cities Report 2020: The Value of Sustainable Urbanization*. United Nations Human Settlements Programme.

Wang, L. H. (2024). ESP32 in IoT: A Survey of System-on-Chip Integration for Smart Applications. *ACM Computing Surveys*.

Yousuf, A. R. (n.d.). A comprehensive study of LoRaWAN for large-scale IoT deployment. *IEEE Access*.

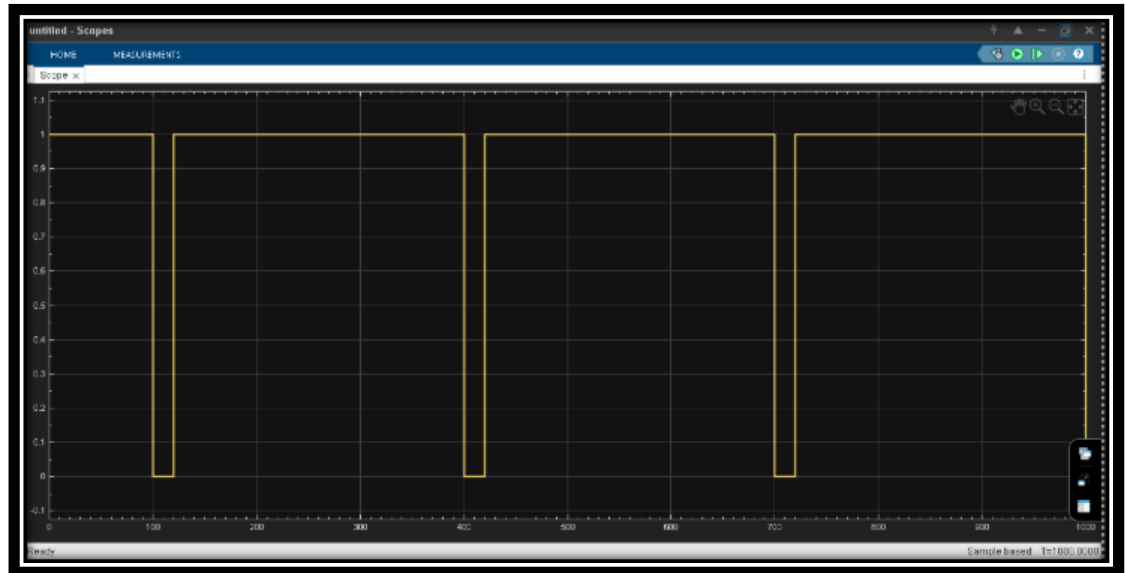
Anexos

Figura 14 - Diagrama de Bloques de la Lógica del Nodo Sensor (Escenario 1)



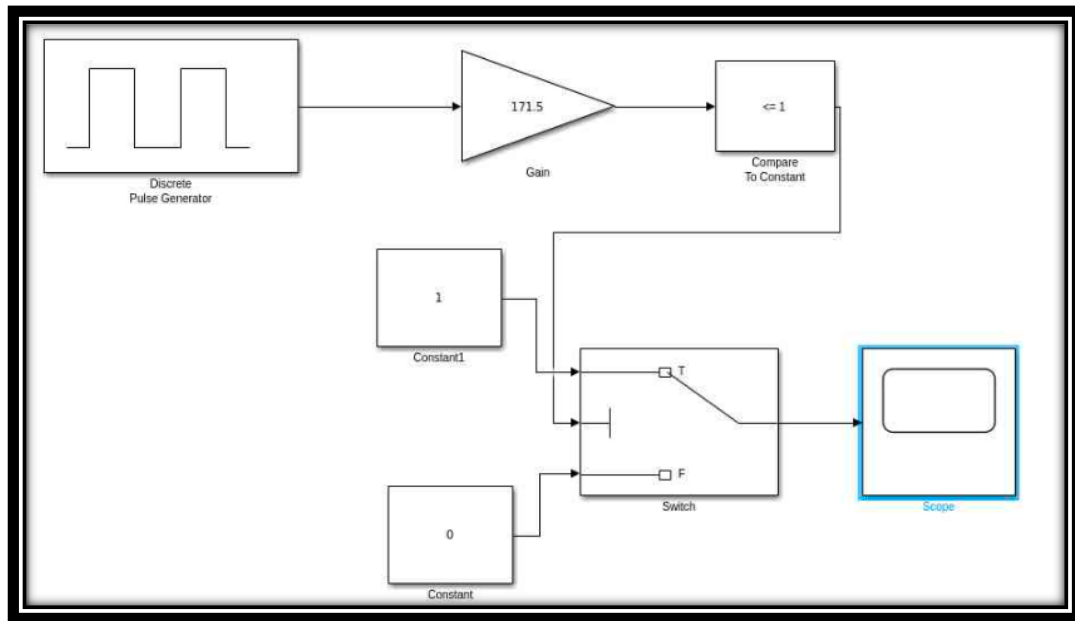
Nota: Fuente: El autor - Este diagrama muestra un sensor de un sistema de estacionamiento inteligente.

Figura 15 - Gráfico del Comportamiento de la Señal del Nodo Sensor (Escenario 1)



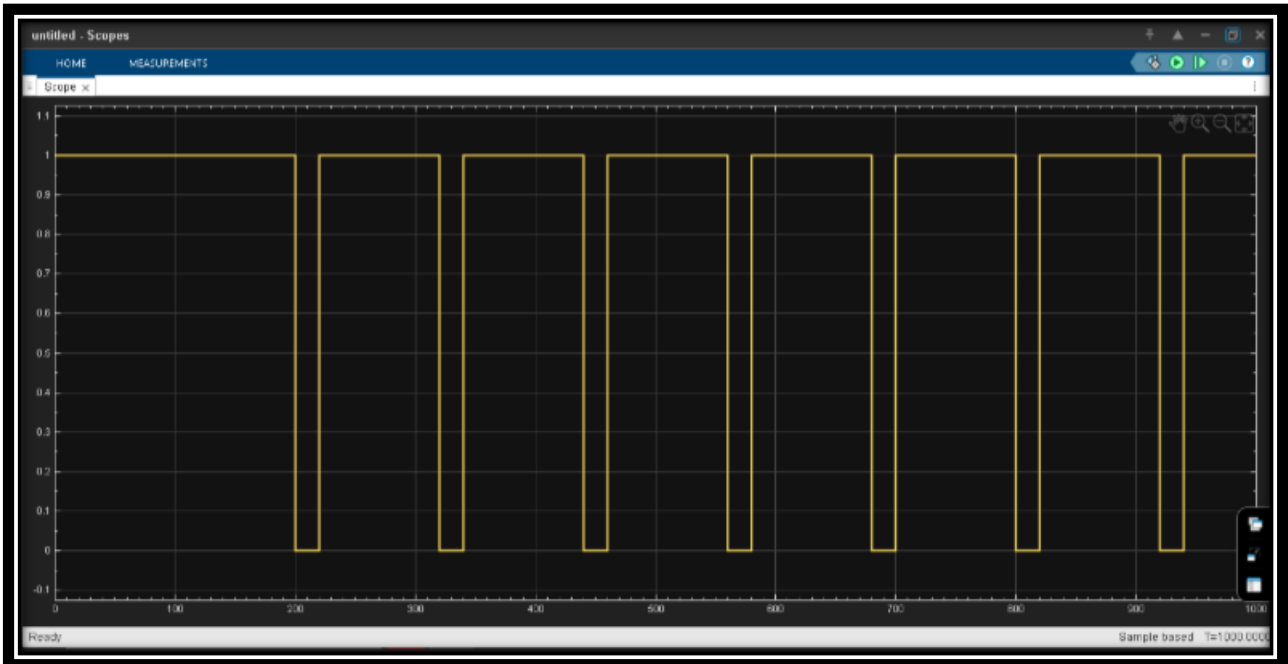
Nota: Fuente: El autor - Este gráfico muestra una señal digital de tren de pulsos. Alterna entre los valores 1 y 0, lo que indica la salida de un interruptor controlado.

Figura 16 - Diagrama de Bloques de la Lógica del Nodo Sensor (Escenario 2)



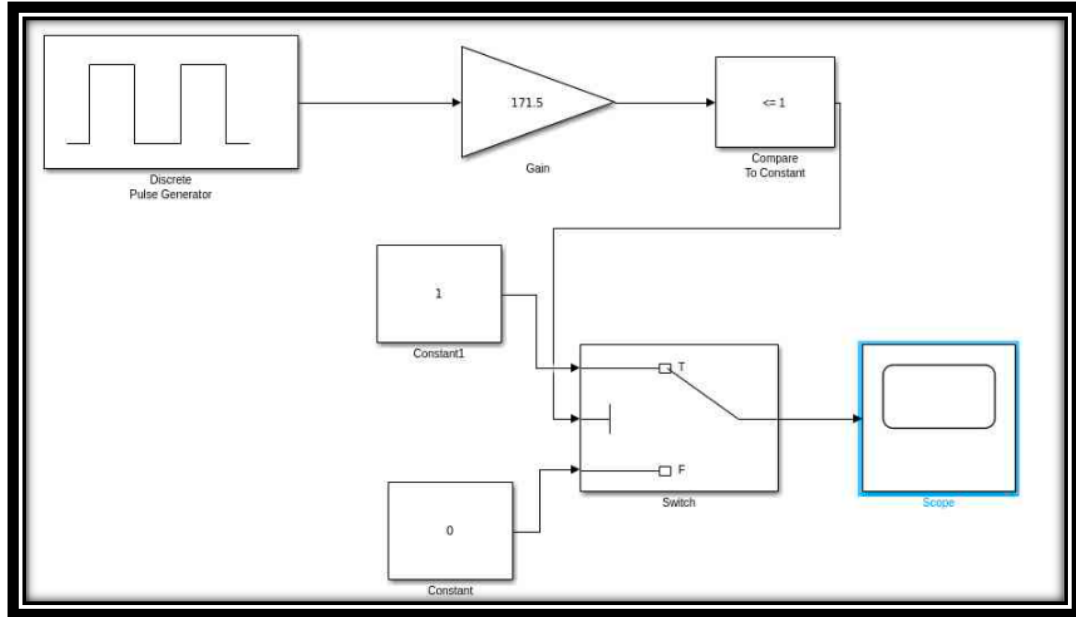
Nota: Fuente: Autor: Este diagrama modela un sensor de un sistema de estacionamiento inteligente.

Figura 17 - Gráfico del Comportamiento de la Señal del Nodo Sensor (Escenario 2)



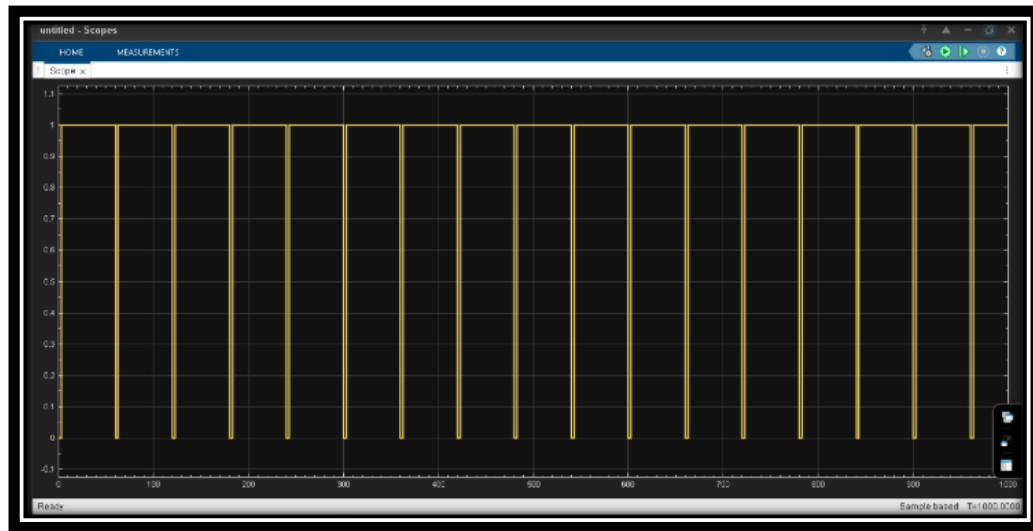
Nota: Fuente: El autor - El gráfico muestra una señal digital de tren de pulsos. A diferencia de la imagen anterior, el pulso bajo (0) es más largo, lo que sugiere que la duración de la señal está relacionada con el tiempo que el vehículo permanece en el espacio de estacionamiento.

Figura 18 - Diagrama de Bloques de la Lógica del Nodo Sensor (Escenario 3)



Nota: Fuente: El autor - Este diagrama modela un sensor de un sistema de estacionamiento inteligente.

Figura 19 - Gráfico del Comportamiento de la Señal del Nodo Sensor (Escenario 3)



Nota: Fuente: El autor - Este gráfico muestra una señal de tren de pulsos digitales. Se diferencia del gráfico anterior en que el pulso bajo (0) es muy corto y regular.

Código de la Simulación del Escenario 3

```
import numpy as np

import matplotlib.pyplot as plt

# --- 1. Definir Parámetros de la Simulación ---

num_nodos = 400

periodo_base = 60 # Periodo de 1 minuto para capacidad máxima

tiempo_simulacion = 1000

variacion_max = 50

rondas_transmision = 15 # Aumentamos rondas para un escenario saturado

# --- 2. Simular Tiempos de Transmisión con Dispersión ---

tiempos_globales = []

# Generar un retraso inicial aleatorio para la primera transmisión de cada nodo

tiempos_ronda_anterior = np.random.randint(1, periodo_base + 1, size=num_nodos)
```

```

tiempos_globales.extend(tiempos_ronda_anterior)

for i in range(1, rondas_transmision):

    variacion_aleatoria = np.random.randint(-variacion_max, variacion_max + 1,
size=num_nodos)

    tiempos_ronda_actual = tiempos_ronda_anterior + periodo_base + variacion_aleatoria

    tiempos_globales.extend(tiempos_ronda_actual)

    tiempos_ronda_anterior = tiempos_ronda_actual


# Convertir a array de NumPy y filtrar tiempos

tiempos_globales = np.array(tiempos_globales)

tiempos_globales = tiempos_globales[tiempos_globales <= tiempo_simulacion]


# --- 3. Calcular Colisiones ---

tiempos_redondeados = np.round(tiempos_globales)

unique_times, counts = np.unique(tiempos_redondeados, return_counts=True)


# Crear un diccionario para mapear cada tiempo a su conteo

```

```

conteo_map = dict(zip(unique_times, counts))

# Asignar el conteo de colisiones a cada punto de tiempo

conteo_colisiones = [conteo_map[t] for t in tiempos_redondeados]

# --- 4. Crear el Gráfico de Alta Calidad ---

plt.figure(figsize=(10, 6))

plt.scatter(tiempos_globales, conteo_colisiones, s=15) # s es el tamaño del punto

# --- 5. Formatear Ejes y Títulos ---

plt.title('Resultados de Simulación - Escenario 3 (400 Nodos)')

plt.xlabel('Tiempo (s)')

plt.ylabel('Transmisiones Simultáneas')

plt.xlim(0, tiempo_simulacion)

plt.ylim(0, 30) # Aumentamos la escala del eje Y para una red saturada

plt.yticks(np.arange(0, 31, 2)) # Marcas en el eje Y de 2 en 2

plt.grid(True, linestyle='--', alpha=0.6)

```

```
# --- 6. Guardar la Imagen en Alta Calidad ---
```

```
output_filename = 'grafico_escenario3_python.png'
```

```
plt.savefig(output_filename, dpi=300)
```

```
print(f"¡Gráfico final generado y guardado como {output_filename}!")
```

```
# Opcional: mostrar el gráfico en pantalla
```

```
# plt.show()
```

Nota: Fuente: El autor



**Presidencia
de la República
del Ecuador**



**Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes**



SENESCYT
Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

DECLARACIÓN Y AUTORIZACIÓN

Yo, **Cali Larreategui, Hector Rolando** con C.C: **0930467188** autor del Trabajo de Integración Curricular: **Diseño y simulación de un sistema IOT de parqueo inteligente utilizando LORAWAN y ESP32 en el campus de la UCSG**, previo a la obtención del título de **Ingeniero en Telecomunicaciones**, en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de integración curricular para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de integración curricular, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 02 de septiembre del año 2025

Cali Larreategui, Hector Rolando

C.C: 0930467188



**Presidencia
de la República
del Ecuador**



**Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes**



SENESCYT
Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA		
FICHA DE REGISTRO DE TESIS/TRABAJO DE INTEGRACIÓN CURRICULAR		
TÍTULO Y SUBTÍTULO:	Diseño y simulación de un sistema IOT de parqueo inteligente utilizando LORAWAN y ESP32 en el campus de la UCSG.	
AUTOR(ES)	Cali Larreategui, Hector Rolando	
REVISOR(ES)/TUTOR(ES)	Ing. Ubilla González, Ricardo Xavier MsC	
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil.	
FACULTAD:	Sistema de Titulación	
CARRERA:	Ingeniería en Telecomunicaciones	
TITULO OBTENIDO:	Ingeniero en Telecomunicaciones	
FECHA DE PUBLICACIÓN:	02 de septiembre del 2025	No. DE PÁGINAS: 98
ÁREAS TEMÁTICAS:	IoT, LoraWAN, Sistemas Embebidos y Diseño de Hardware. Simulación y Modelado de Redes	
PALABRAS CLAVES/ KEYWORDS:	IoT, LoRaWAN, Parqueo inteligente, ESP32, Simulación.	
Esta tesis se centra en el diseño y la simulación de un sistema inteligente de monitoreo de estacionamientos, con el objetivo de optimizar la gestión del mismo en el campus de la Universidad Católica de Santiago de Guayaquil (UCSG). Ante el problema de la congestión vehicular y la búsqueda ineficiente de plazas disponibles, se propuso una solución que utiliza el Internet de las Cosas (IoT) y la tecnología de comunicación LoRaWAN de largo alcance y bajo consumo. La metodología que fue utilizada está basada en un esquema que este compuesto por nodos sensores, que están equipados con microcontroladores, en una red LoraWAN y una plataforma IoT para poder ver los datos. También se usó el simulador de red NS-3 para ver cómo se comporta la red según el escenario que se simule. Esto nos permitió validar que tan efectivo era el sistema, mostrando un deterioro en la entrega de paquetes (PDR) mientras más aumenta la carga de la red y mostrando de esta forma que la batería tendría una vida útil que puede superar los 9 años. El estudio concluye que la solución propuesta es una herramienta escalable y eficiente para reducir la congestión y mejorar la experiencia del usuario en el campus.		
ADJUNTO PDF:	☒ SI	☐ NO
CONTACTO CON AUTOR/ES:	Teléfono: +593 - 979730320	E-mail: hector.cali@cu.ucsg.edu.ec
CONTACTO CON LA INSTITUCIÓN:	Nombre: Ing. Ubilla González Ricardo Xavier.	
COORDINADOR DEL PROCESO DE UTE	Teléfono: +593 - 999528515	
	E-mail: ricardo.ubilla@cu.ucsg.edu.ec	
SECCIÓN PARA USO DE BIBLIOTECA		
Nº. DE REGISTRO (en base a datos):		
Nº. DE CLASIFICACIÓN:		
DIRECCIÓN URL (tesis en la web):		