



Pontificia Universidad
Católica del Ecuador

FACULTAD DE INGENIERÍA

MAESTRIA EN TECNOLOGÍAS DE LA INFORMACIÓN

MENCIÓN REDES DE COMUNICACIONES

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE MASTER EN
TECNOLOGÍAS DE LA INFORMACIÓN MENCIÓN REDES DE COMUNICACIONES**

**Tema: Propuesta de solución Internet of Things (IoT) con tecnología LoRaWAN,
para un Sistema de Parqueadero Inteligente (SPI).**

Caso de estudio Conjunto Habitacional Parque Real (CHPR).

AUTOR: FRANCISCO XAVIER MEJÍA TAMAYO

QUITO, 2020 – DICIEMBRE

Contenidos

1. Capítulo 1: Marco Teórico.....	5
1.1 Dispositivos Smart.....	5
1.2 IoT.....	5
1.2.1 Arquitectura IoT por capas	5
1.2.2 Requerimientos de una Arquitectura IoT	7
1.3 Servidores en la nube.....	8
1.4 API.....	8
1.5 LPWAN	9
1.6 Principales tecnologías LPWAN	11
1.7 LoRaWAN.....	13
1.8 Microcontrolador	14
1.9 Hardware para desarrollar proyectos IoT	15
1.9.1 Tarjetas de desarrollo basadas en Microcontroladores	15
1.9.2 SBC.....	16
1.10 Sensores	16
1.10.1 Sensor ultrasónico	17
1.11 Parqueadero.....	17
1.12 SPI.....	17
2. Capítulo 2: Situación Actual del Conjunto Habitacional Parque Real.....	18
2.1. Antecedentes y ubicación del CHPR.....	18
2.2. Planos arquitectónicos de los parqueaderos de visita del CHPR.....	18
2.3. Planos de puntos de energía eléctrica en los parqueaderos de visita	19
2.4. Planos de acceso de comunicaciones para los parqueaderos de visita	20
2.5. Problemática actual de los parqueaderos de visita del CHPR	20
3. Capítulo 3: Propuesta de Arquitectura IoT para un SPI.....	24
3.1. Diseño de capas IoT	25
3.1.1 Capa de ambiente (Environment Layer)	26
3.1.2 Capa de Hardware (Hardware Layer).....	26
3.1.3 Capa de Red (Network Layer)	27
3.1.4 Capa de comunicación (Communication Layer)	28
3.1.5 Capa de servicios (Services Layer)	28

3.1.6	Capa de soporte de aplicaciones y administración (Application Support & Management Layer)	28
3.1.7	Capa de aplicación (Application Layer)	29
3.2.	Requerimientos de arquitectura IoT	29
3.2.1	Número de dispositivos	29
3.2.2	Conectividad y Comunicación	29
3.2.3	Gestión de dispositivos	29
3.2.4	Recolección, Análisis y Acción sobre la información	30
3.2.5	Escalabilidad	30
3.2.6	Seguridad	30
3.2.7	Energía	30
3.2.8	Visualización de la información:	31
3.3.	Requerimientos de cobertura	31
3.4.	Selección de componentes	31
3.4.1	Selección de módulos de desarrollo	31
3.4.2	Selección sensores ultrasónicos	34
3.4.3	Selección de solución LoRaWAN	35
3.4.4	Selección de servidor IoT en la nube	38
4.	Capítulo 4: Prototipo funcional	41
4.1	Definición del número de Gateways LoRa necesarios	41
4.2	Diseño de atributos y funcionalidades del Gateway LoRa	44
4.3	Diseño de atributos y funcionalidades de los Nodos LoRa	45
4.3.1	Nodo LoRa para parqueaderos	45
4.3.2	Nodo LoRa para señalización	48
4.4	Interconexión del Gateway LG-02 con el servidor IoT M2MLight	51
4.4.1	Validación de Información con respecto a parqueaderos en servidor IoT M2MLight	54
4.5	Diseño de atributos y funcionalidades del servidor IoT	54
4.5.1	Diseño de la base de datos para la interacción con el servidor IoT M2MLight	55
4.5.2	Diseño de mensajes MQTT para la interacción entre el servidor IoT M2MLight y el Gateway Dragino LG02	56
4.5.3	Procesos diseñados para la interacción entre el servidor IoT M2MLight con la base de datos y con el Gateway Dragino LG02	58

4.6 Pruebas de funcionamiento del SPI propuesto	61
4.6.1 Calibración de los sensores de presencia	61
4.6.2 Calibración de parámetros de transmisión LoRa	62
4.6.3 Instalación de leds adicionales para señalización de los parqueaderos de visita	62
4.6.4 Capacitación del personal de recepción y guardias con respecto a la operación del SPI.....	63
4.7 Resultados del SPI implementado	64
5. Capítulo 5: Conclusiones y Recomendaciones	66
5.1 Conclusiones	66
5.2 Recomendaciones.....	67
6. Bibliografía	70
7. Glosario.....	73

1. Capítulo 1: Marco Teórico

Este capítulo presenta los conceptos, tecnológicas y/o metodologías utilizados para el desarrollo del Trabajo de Titulación.

1.1 Dispositivos Smart

Los dispositivos Smart o dispositivos inteligentes dentro del contexto electrónico, hacen referencia a que son dispositivos que se encuentran conectados hacia una red, cuyo objetivo primordial es recopilar y procesar información, basados en múltiples tipos de sensores, pudiendo llegar a ser cada vez más autónomos haciendo uso de lo denominada inteligencia artificial (Vermeulen, 2017).

1.2 IoT¹

El IoT, esta definido por la ITU² en la recomendación ITU-T Y.2060, como: *“Infraestructura mundial para la sociedad de la información que propicia la prestación de servicios avanzados mediante la interconexión de objetos (físicos y virtuales) gracias a la interoperatividad de tecnologías de la información y la comunicación presentes y futuras”* (UIT-T, 2012).

El principal objetivo de la tecnología IoT es convertir a objetos de uso diario como lámparas, licuadoras, termostatos, etc., en dispositivos inteligentes, esto se realiza dotándoles de conexión a internet e interconectándolos a servidores en la nube.

Para conseguir esto, la tecnología IoT se basa en 4 pilares fundamentales (Guru99, 2020):

- 1- **Sensores o dispositivos:** son los dispositivos que se encargan de recopilar información del entorno. De acuerdo con la aplicación un dispositivo puede tener uno o múltiples sensores conectados.
- 2- **Conectividad:** toda la información recolectada es enviada a través de internet hacia una infraestructura en la nube.
- 3- **Procesamiento de información:** una vez que la información llega al servidor en la nube, ésta es procesada y se toman las respectivas medidas correctivas en caso de ser necesario.
- 4- **Interfaz de usuario:** una vez que la información ha sido procesada, ésta puede ser mostrada en alguna aplicación o a su vez enviada al usuario ya sea por mensajes de texto, email, etc.

1.2.1 Arquitectura IoT por capas

Para facilitar el diseño de aplicaciones IoT, los expertos sugieren seguir un diseño en capas, mediante la recomendación ITU-T Y.2060, la misma que propone un esquema de capas, igual lo hace (Darwish, 2015), el cual pretende ser una mejora del recomendado por la ITU y que puede visualizarse en la figura 1 – 01.

¹ IoT Internet of Things - Internet de las Cosas.

² ITU International Telecommunications Union - Unión Internacional de Telecomunicaciones.

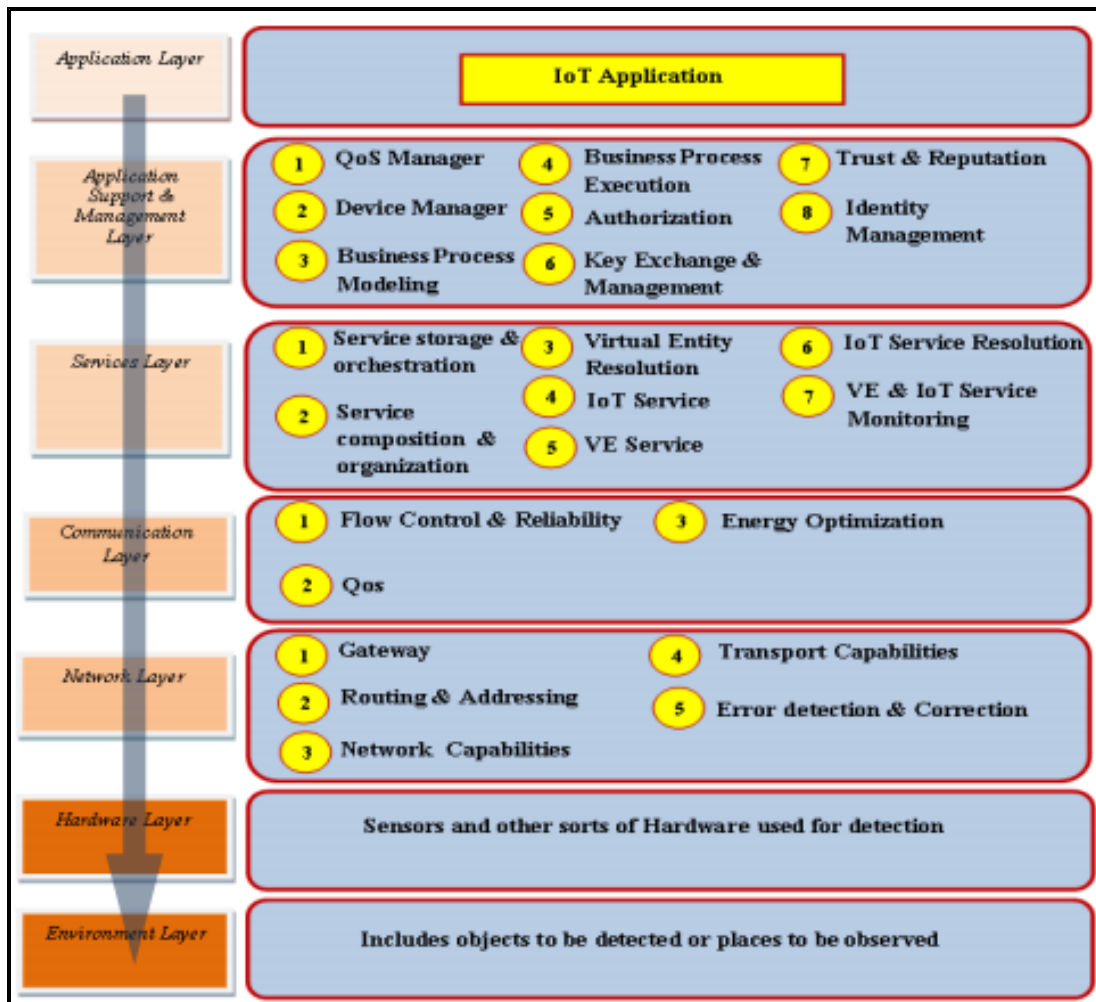


Figura 1 – 01 Diseño de aplicaciones IoT mediante arquitectura de capas recomendado por (Darwish, 2015).

- 1- **Capa de ambiente (Environment Layer):** constituyen los objetos o lugares a ser observados o monitoreados, incluyendo sus características como temperatura, humedad, etc.
- 2- **Capa de Hardware (Hardware Layer):** esta capa incluye los sensores que permiten la interconexión entre el mundo físico con el digital, así como también con cualquier otro hardware necesario como dispositivos embebidos, etc.
- 3- **Capa de Red (Network Layer):** se encarga de realizar las funciones de puerta de enlace, ruteo, direccionamiento, capacidades de red y transporte, detección y corrección de errores.
- 4- **Capa de comunicación (Communication Layer):** se encarga de ejecutar funciones de control de flujo, confiabilidad, calidad de servicio y optimización de energía. De ser necesario realiza también la comunicación con el portal web IoT.
- 5- **Capa de servicios (Services Layer):** se encarga de las funciones de monitoreo, almacenamiento, organización y visualización de la información recibida y resolver entidades virtuales creadas.

- 6- **Capa de soporte de aplicaciones y administración (Application Support & Management Layer):** en ella se ejecutan todas las acciones relacionadas con el control, seguridad y manejo de la aplicación.
- 7- **Capa de aplicación (Application Layer):** es la capa al tope de la arquitectura, y es la responsable de entregar las aplicaciones a los diferentes usuarios del IoT.

1.2.2 Requerimientos de una Arquitectura IoT

Al momento de diseñar una solución IoT nos vamos a apoyar en lo expuesto por (Chakray, 2016), se deben considerar ciertos requerimientos que son específicos para una arquitectura IoT, ya que al participar dispositivos que generalmente tienen capacidades limitadas, se deben tomar las respectivas características en consideración. A continuación, se detallan los requerimientos que siempre deben ser tomados en cuenta:

- **Definir el número de dispositivos:** de acuerdo con la naturaleza de la aplicación, deberán definirse la cantidad de nodos y en consecuencia también el número de sensores requeridos. De ser necesario también deberá tomarse en cuenta una posible expansión o incremento de dispositivos a futuro.
- **Conectividad y Comunicación:** se debe considerar los aspectos relacionados a la conectividad desde los nodos con los servidores en la nube, ya sea esta comunicación con o sin Gateways. Considerando los distintos protocolos que se vayan a utilizar para transmitir la información, ya que de acuerdo con estos, los dispositivos requerirán mayor o menor capacidad de memoria y procesamiento. Así como también se deberán definir los medios por los cuales se realizará esta comunicación, ya sea a través de redes de fibra óptica, telefonía celular, etc.
- **Gestión de dispositivos:** esto hace referencia a la capacidad de controlar los dispositivos remotamente, esto con la finalidad de realizar ciertas acciones sobre los mismos, pudiendo por ejemplo efectuar actualizaciones en el firmware, en las credenciales de seguridad, reconfigurar ciertos parámetros, etc.
- **Recolección, Análisis y Acción sobre la información:** el objetivo principal de un sistema IoT es el de recopilar información desde múltiples dispositivos, almacenar y analizar esta información y de acuerdo con ella gestionar acciones, ya sean correctivas o meramente en forma de alertas.
- **Escalabilidad:** la capacidad de escalar un sistema IoT es fundamental, el sistema deberá ser capaz de iniciar con unos pocos dispositivos y continuar ampliando la cantidad de estos conforme se requiera.
- **Seguridad:** es uno de los principales factores a tomar en cuenta en el diseño de un sistema IoT, puesto que dependiendo de la naturaleza de la aplicación, al encontrarse los dispositivos IoT recolectando información, ésta puede ser de tipo personal.
- **Energía:** se debe considerar las fuentes o suministros de energía de los cuales va a hacer uso el sistema IoT, se debe dimensionar los requerimientos de energía del sistema, incluyendo el consumo de leds, motores, etc. y de acuerdo con esto escoger una fuente

de energía apropiada, sea está alimentada con paneles solares o directamente del suministro de energía de la empresa eléctrica del lugar.

- **Visualización de la información:** de acuerdo con la naturaleza de la aplicación, se debe definir cómo será presentada la información al usuario final, ya sea esta mediante un buscador o alguna aplicación en el celular o inclusive mediante envío automático de mensajes vía Twitter o correo electrónico.

1.3 Servidores en la nube

Los servicios de computación en la nube tienen una estrecha relación con las aplicaciones IoT, una correcta complementación entre ambos ayuda a mejorar la eficiencia de un sistema que se desea implementar. Dada la gran cantidad de datos generados por las aplicaciones IoT, se requieren de servidores especializados, que permitan trazar un camino para el tráfico de información. Entre los principales beneficios de los servidores en la nube para aplicaciones IoT se pueden mencionar: la capacidad de contratar solo los recursos necesarios, no se requiere preocuparse de las necesidades de la infraestructura de red, se puede ahorrar dinero en la implementación de centros de datos y se puede desplegar aplicaciones a nivel mundial en cuestión de minutos (Intelligence, 2019).

1.4 API³

Como lo menciona (Becerril, 2020). *“Las API son interfaces que permiten conectar una infraestructura digital con otro software a través de soluciones tecnológicas nativas de la nube. Estos desarrollos posibilitan consumir y compartir datos específicos sin revelar información confidencial de clientes, sistemas y otros elementos.*

El uso de aplicaciones API ayuda a simplificar tiempos y recursos de desarrollo para las empresas. Utilizar una API abierta permite que los desarrolladores web puedan enfocar sus energías en otras tareas de desarrollo y montarse sobre una infraestructura existente.

Una API habilita el intercambio de información entre dos plataformas, sin revelar cómo están implementadas sus infraestructuras. Esto facilita el desarrollo de servicios y genera ahorros de tiempo y dinero. Además de la flexibilidad que brindan, las API ayudan a simplificar el diseño, administración y uso de las aplicaciones.

Existen API que permiten acceder a datos o a funciones de un dispositivo o un sistema operativo local, éstas se conocen como API local. Las API remotas permiten acceder a datos o servicios de un proveedor a través de una red de comunicaciones.

Con las API se puede consumir una cantidad limitada de datos, para los que se otorga un permiso exclusivo, y no las bases de datos completas”.

En resumen, las APIs se pueden considerar como subrutinas que permiten al programador hacer uso de códigos de programación previamente desarrollados, los cuales le permitirán enfocar sus

³ API Application Programming Interface – Interfaz de Programación de Aplicación.

esfuerzos en el desarrollo de nuevas aplicaciones que se fundamenten en una determinada API o que permitan mejorar las prestaciones originales.

Para el estudio planteado, se procederá a hacer uso de alguna de las APIs desarrolladas por las empresas que prestan servicios de alojamiento de servidores IoT como por ejemplo Thinkspeak, Thingierio, Cayenne, The Things Network o inclusive Arduino Cloud.

1.5 LPWAN⁴

Las LPWAN, son tecnologías que se encuentran en constante evolución, esto por sus múltiples beneficios como son: comunicaciones de larga distancia, bajo consumo de energía (dependiendo del estado en el que se encuentre el dispositivo, varía entre las unidades y decenas de miliamperios)(Eriksen 2019) y poca complejidad en la implementación, comunicación bidireccional con protocolos de seguridad para aplicaciones IoT. Por estas características este tipo de tecnología es ampliamente utilizada en aplicaciones IoT como son agricultura de precisión, monitoreo de flotas, parqueadero inteligente y muchas otras más (LinkLabs, 2016).

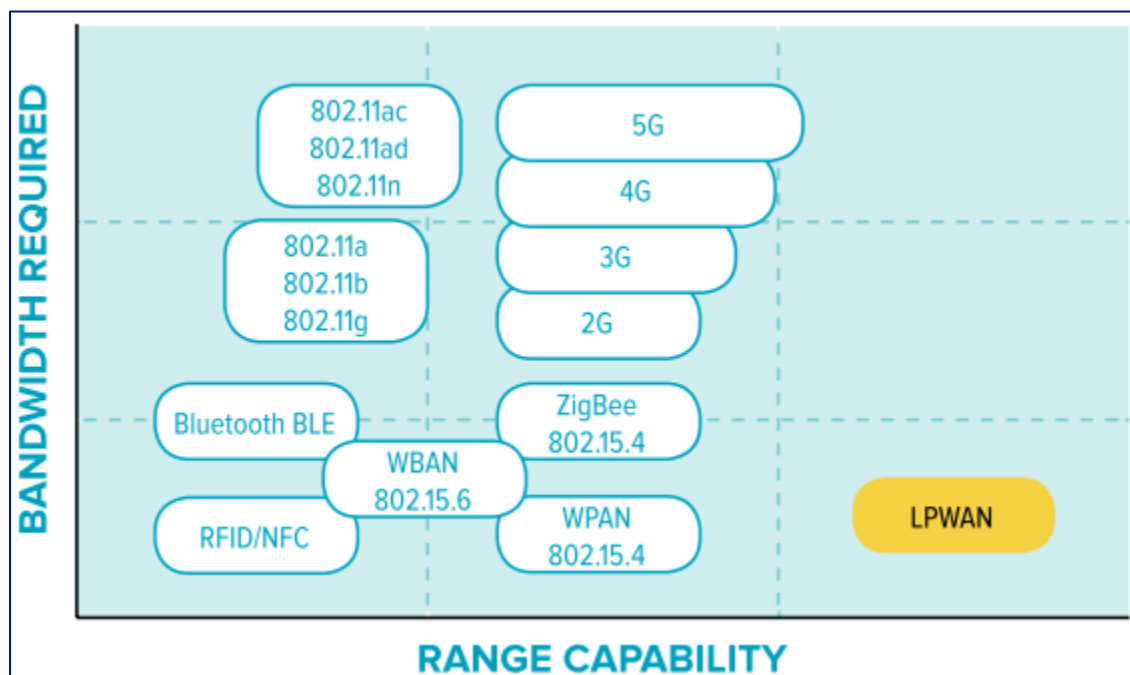


Figura 1 – 02 Comparación entre LPWAN con algunos tipos de tecnologías inalámbricas (Egli, 2015).

Entre las ventajas de LPWAN destacan:

- 1- **Larga duración de las baterías:** de acuerdo con lo mostrado en la tabla 1 – 01, dependiendo del tipo de batería utilizada, la duración y la frecuencia de las transmisiones, se estima una duración de las baterías de múltiples años.

⁴ LPWAN Low Power Wide Area Network - Red de Área Extensa de Bajo Consumo de Energía.

ESTIMACIÓN DEL TIEMPO DE VIDA DE UNA BATERÍA PARA UN DISPOSITIVO LoRaWAN⁵	
Proceso	Valor
Intervalo de transmisión	300 segundos
Número de paquetes por día	288
Capacidad de la batería	10.000 mAh (133,200 J)
Energía utilizada en una transmisión	39,73 mJ
Energía total utilizada en las 288 transmisiones de un día	11,44 J
Corriente utilizada en modo de reposo	0,08 mA
Energía utilizada en todo un día en modo de reposo	25,37 J
Consumo de energía utilizada en un día	36,82 J
Tiempo de expectativa de vida del dispositivo	9.9 años

Tabla 1 – 01 Estimación del tiempo de vida de una batería para un dispositivo LoRaWAN. Tomado de (Singh et al. 2020).

- 2- **Extraordinaria sensibilidad:** presentan una sensibilidad en la recepción de alrededor de 130 dBm, permitiendo tener enlaces con presupuestos de potencia entre 140 a 160 dB.
- 3- **Largo alcance:** debido a su extraordinaria sensibilidad permiten llegar a tener enlaces en espacios abiertos desde 5 hasta 40 Km.
- 4- **Uso de banda ISM⁶:** gracias a que hacen uso de la banda ISM, la puesta en marcha de estos sistemas no requiere de licenciamiento para ser utilizados, lo que facilita el despliegue de nuevos sistemas.

Entre las desventajas de LPWAN destacan:

- 1- **Baja tasa de datos:** se puede apreciar en la figura 1 – 02 que esta tecnología a diferencia de otras no está diseñada para transmitir grandes cantidades de información generalmente máximo 5000 bps.
- 2- **Otros dispositivos operando en banda ISM:** dado que funcionan con frecuencias no licenciadas, coexisten con múltiples tipos de tecnologías inalámbricas, lo que puede ocasionar interferencias o fallos en la comunicación especialmente en entornos urbanos como ciudades.

⁵ LoRaWAN Long Range Wide Area Network – Redes de largo Alcance de Area Extendida.

⁶ ISM Industrial Scientific and Medical, frecuencias para aplicaciones industriales, científicas y médicas.

1.6 Principales tecnologías LPWAN

Con la incesante evolución de la tecnología, hoy en día se tienen múltiples representantes de la tecnología LPWAN, cada uno con sus respectivas ventajas y desventajas, entre los cuales destacan LoRaWAN y Sigfox, cuyas características son detalladas en la tabla 1 – 02.

COMPARACIÓN ENTRE PRINCIPALES TECNOLOGÍAS LPWAN: LoRaWAN VS Sigfox		
Características	LoRaWAN	Sigfox
Tecnología	Utiliza modulación CSS ⁷ y puede también utilizar modulación GFSK ⁸ .	Utiliza UNB ⁹ junto con modulación DBPSK ¹⁰ .
Frecuencias de operación	433 / 868 / 780 / 915 MHz	868 / 902 MHz
Ancho de canal	Estados Unidos: 64 + 8 x 125 KHz Europa: 8 x 125 KHz	UNB: 100 Hz
Alcance	Campo abierto: hasta 15 Km Entorno urbano: 2 - 5 Km	Campo abierto: 20 - 50 Km Entorno urbano: 10 Km
Potencia de TX del nodo final	Estados Unidos: + 30 dBm Europa: + 14 dBm	Desde 10 uW hasta 100 mW
Tasa de datos del enlace de subida	Estados Unidos: 900 bps hasta 100 kbps Europa: 300 bps hasta 50 Kbps	Máximo 140 mensajes por día de 100 bps
Tasa de datos del enlace de bajada	Estados Unidos: 900 bps hasta 100 kbps Europa: 300 bps hasta 50 Kbps	Máximo 4 mensajes por día de 8 bytes
Número máximo de dispositivos conectados al radio base	Subida: > 1M Bajada: < 100K	1 M
Topología	Estrella extendida	Estrella
Presupuesto del enlace	> 150 dB	> 150 dB

⁷ CSS – Chirp Spread Spectrum.

⁸ GFSK – Gaussian Frequency Shift Keying.

⁹ UNB – Ultra Narrow Band.

¹⁰ BPSK – Binary Phase Shift Keying.

COMPARACIÓN ENTRE PRINCIPALES TECNOLOGÍAS LPWAN: LoRaWAN VS Sigfox		
Características	LoRaWAN	Sigfox
Interferencia	Dado su ancho espectral, es más susceptible a interferencias. Sin embargo, el tipo de modulación permite mitigar los efectos del ruido.	Dado el rango de frecuencia de operación en Estados Unidos, hay más operadores que pueden ocasionar interferencia. Sin embargo, el tipo de modulación permite mitigar los efectos del ruido.
Radio base	Dadas las capacidades de procesamiento son más costosos que los radios finales.	Requiere de mayor sofisticación en cuanto a capacidad de procesamiento y sensibilidad.
Radio receptor final	Son más baratos que los radios base, dado que no requieren de gran capacidad de procesamiento.	Son más baratos que los radios base, dado que no requieren de gran capacidad de procesamiento.
Comunicación bidireccional	Si, dado que los radios en ambos extremos son iguales, las capacidades de transmisión son similares en ambos sentidos.	Si, aunque funciona mejor en el sentido de transmisión desde el radio final hacia la radio base, por cuanto la sensibilidad del radio final es menor que el radio base.
Fabricantes de radios	Fabricados únicamente por la compañía Semtech.	Libre apertura para que fabricantes obtengan las especificaciones y fabriquen los radios.
Modelo de negocio	Permite el relativamente libre acceso a sus especificaciones y el potencial de ser flexibles, sin que el usuario deba atarse a una red en específico, o inclusive permite montarse su propia red.	Al permitir la libre construcción de sus equipos de radio finales, hace dinero al obtener regalías por la utilización de su software y/o red desplegada en conjunto con operadores de telefonía móvil.
Consideraciones	Carece de ciertas funcionalidades que pueden ser útiles para los usuarios como: roaming, paquetización y reintentos, operaciones de desconexión, calidad de servicio, actualizaciones de firmware por el aire y funcionalidad de repetidores. Para la operación requieren de servidores en la nube.	El uso esta limitado a la cobertura provista por los operadores móviles o empresas con quienes Sigfox tenga convenios. Para una comunicación bidireccional se requiere de una red más densa. Tiene limitaciones de tráfico para el usuario final a 15 bytes con 140 mensajes y 4 confirmaciones por día.

COMPARACIÓN ENTRE PRINCIPALES TECNOLOGÍAS LPWAN: LoRaWAN VS Sigfox		
Características	LoRaWAN	Sigfox
	Semtech es el único fabricante de los chips de comunicación, aunque se encuentran en conversaciones para permitir otros.	FCC ¹¹ apartado 15, limita la transmisión a 0,4 segundos lo que implica una modificación de su típica transmisión de 3 segundos.
Aplicaciones	Ideal para aplicaciones que se requiera comunicación bidireccional, por su enlace simétrico.	Mejor para aplicaciones que requieren el envío de información pequeña poco frecuente.

Tabla 1 – 02 Comparación entre las principales tecnologías LPWAN: LoRaWAN Vs. Sigfox. Tomado de (Ray 2018) y (Pardal 2017).

1.7 LoRaWAN

La tecnología LoRaWAN, es una especificación dentro del tipo de red LPWAN, fue desarrollado y es soportado por la denomina LoRa Alliance (asociación abierta sin fines de lucro, a la fecha está conformado por más de 500 miembros que se han seguido sumando desde su creación en marzo del 2015), la cual define la arquitectura del sistema y los protocolos de red para los dispositivos LoRa¹².

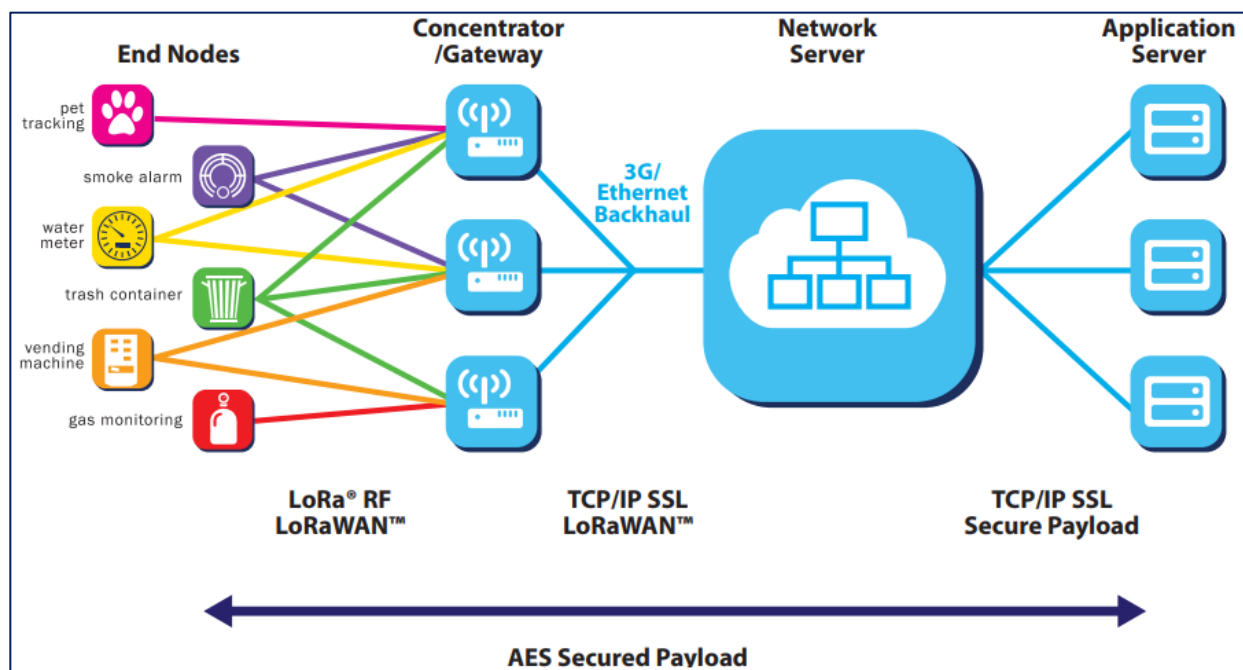


Figura 1 – 03 Interconexión entre dispositivos LoRaWAN con un servidor de aplicaciones IoT (LoRa-Alliance, 2015).

¹¹ FCC – Federal Communications Commission, es la agencia federal responsable de implementar y hacer cumplir ...

¹² LoRa especificación de capa física que emplea modulación similar a CSS (Chirp Spread Spectrum).

Se puede apreciar en la figura 1-03, que la tecnología LoRaWAN se basa en una topología de estrella de largo alcance, en donde los nodos LoRa envían sus paquetes de datos hacia los Gateways LoRa, y son estos últimos los que se conectan con los servidores mediante otras redes ya sean de telefonía celular o ethernet. Independientemente si más de un Gateway recibió la información, todos aquellos que la recibieron pasan la misma al servidor en la nube, el cual va a manejar los aspectos relativos a inteligencia y complejidad para realizar el respectivo filtrado de paquetes duplicados, tras lo cual los usuarios pueden acceder a la información a través de los servidores de aplicaciones mediante una conexión a internet (Islam et al., 2017).

Principales características de LoRaWAN (LoRa-Alliance, 2015):

- 1- **Largo alcance:** gracias a la tecnología empleada en la capa física se puede permitir una amplia cobertura normalmente entre 5 y 15 Km, haciendo factible que prácticamente un Gateway pueda proporcionar suficiente cobertura a varios cientos de kilómetros cuadrados e inclusive a una ciudad entera, dependiendo exclusivamente de los obstáculos existentes.
- 2- **Tasas de transferencia:** se puede tener tasas de transferencia desde 250 bps hasta 50 Kbps.
- 3- **Gran tiempo de vida de las baterías:** se estima un tiempo de vida de las baterías entre 10 y 20 años, al funcionar los nodos de una manera asíncrona, estos transmiten información únicamente cuando sucede algún cambio en las condiciones físicas o ambientales en donde se encuentra el nodo instalado, con lo cual se puede ahorrar considerablemente energía respecto a sistemas síncronos.
- 4- **Gran capacidad de dispositivos en la red:** esto se logra al disponer de tasas de transmisión de datos adaptativas y transceptores multicanal en el Gateway, permitiendo que se puedan recibir mensajes simultáneos desde los diferentes nodos.
- 5- **Escalabilidad:** dados los requisitos mínimos de infraestructura, en el caso de requerirse más capacidad, se puede instalar un nuevo Gateway y de igual manera aprovechar las tasas de datos adaptativas y mejorar así la conexión tanto como con dispositivos antiguos (incrementando las tasas de transferencia) y con los nuevos.
- 6- **Seguridad:** se implementan dos capas de seguridad, la primera en la capa de red permitiendo comprobar la autenticidad del nodo dentro de la red y la segunda implementando el protocolo de inscripción AES¹³. para proteger los datos del usuario.

1.8 Microcontrolador

Es un circuito electrónico que permite realizar procesamiento y almacenamiento de información, para lograr esto se basa en uno o varios procesadores, memoria RAM¹⁴ para realizar el

¹³ AES: Advanced Encryption Estandar, es un esquema de cifrado por bloques.

¹⁴ RAM Random Access Memory – Memoria de Acceso Aleatorio

procesamiento de la información y memoria EPROM¹⁵ para almacenar el programa o rutina programada a correr en el microcontrolador.

1.9 Hardware para desarrollar proyectos IoT

El desarrollo de aplicaciones IoT se fundamenta en herramientas de hardware modular o kits de desarrollo que han tomado un gran impulso y que han mejorado tanto en funcionalidad, costo y disponibilidad comercial, los cuales están basados en circuitos electrónicos. Se destacan dos tipos de módulos de desarrollo, los primeros denominados tarjetas de desarrollo apoyados en microcontroladores y los segundos denominados SBC¹⁶.

1.9.1 Tarjetas de desarrollo basadas en Microcontroladores

Son tarjetas PCB¹⁷, con circuitería adicional para facilitar la programación y el prototipado de soluciones a través de microcontroladores, las mismas permiten la interconexión con módulos externos (sensores o actuadores) mediante la utilización de sus pines GPIO¹⁸ y/o a través de protocolos de comunicación como son I2C¹⁹ o SPI²⁰, además se permite la conexión con módulos interconectados a la misma tarjeta PCB o a soluciones externas.

Entre los principales referentes dentro de este tipo de soluciones tenemos a Arduino, la misma que constituye una plataforma electrónica de código abierto basada en un software y hardware. Las tarjetas de desarrollo Arduino, disponen de múltiples puertos que pueden ser configurados como entradas y salidas, que permiten al Arduino interactuar con el mundo exterior (Arduino.cc, 2019).

Las funcionalidades de las tarjetas de desarrollo Arduino pueden ser incrementadas o mejoradas a través de la utilización de los denominados SHIELDS, estas son tarjetas PCB adicionales, que facilitan el incremento de funcionalidades de un sistema. Por ejemplo, si a un proyecto queremos dotarle de conectividad Ethernet, se puede utilizar el denominado Ethernet Shield, para el cual se tienen múltiples usos, y con el cual en cuestión de minutos se puede tener un servidor web funcional.

Se utiliza el software IDE²¹, para escribir el código de programación de las tarjetas de desarrollo Arduino y también se emplea para la conexión, programación e interacción con el mismo (Arduino.cc, 2015).

¹⁵ EPROM Erasable Programmable Read Only Memory – Memoria de Lectura Borrable Programable.

¹⁶ SBC Single Board Computers – Computador en una sola tarjeta.

¹⁷ PCB Printed Circuit Board – Placa de Circuito Impreso.

¹⁸ GPIO General Purpose Input Output – Pines de Entrada/Salida de Propósito General.

¹⁹ I2C: Inter Integrated Circuits, tipo de bus que se utilizaba para interconectar circuitos integrados.

²⁰ SPI: Serial Peripheral Interface, estándar para comunicaciones sincrónicas.

²¹ IDE Integrated Development Environment – Ambiente de Desarrollo Integrado para tarjetas Arduino.

1.9.2 SBC

Estos dispositivos constituyen un paso adicional a los microcontroladores, son diseños que en una sola tarjeta disponen de todos los circuitos esenciales como pueden ser CPU²², GPU²³, periféricos de entrada como teclado y ratón e inclusive lo necesario para conectar una o más pantallas, al igual que ofrecen mayores capacidades de memoria y capacidad de procesamiento que los otorgados por un microcontrolador. De igual manera permiten tener pines GPIO y ya vienen embebidos con muchas funcionalidades adicionales como son módulos de memoria externa, Bluetooth, Ethernet y Wifi. En resumen estos dispositivos son como mini computadoras, con su propio sistema operativo, generalmente basados en distribuciones Linux, lo cual implica que las soluciones de este tipo sean más complejas y a la par esto ocasiona que estos dispositivos consuman mucha más energía y que sean soluciones mucho más costosas (Yamanoor & Yamanoor, 2017).

Al igual que con las tarjetas de desarrollo basadas en microcontroladores, las características de los dispositivos SBC pueden ser incrementadas mediante los denominados HAT²⁴.

El principal representante de este tipo de tecnologías es el denominado Raspberry Pi, el cual se encuentra apoyado por la Raspberry Pi Foundation, una organización sin fines de lucro, cuyo objetivo es llevar educación digital de bajo costo a las aulas y escuelas de países en vías de desarrollo.

1.10 Sensores

Los sensores constituyen una parte fundamental al momento de automatizar algún proceso, estos se encargan de detectar los cambios en las condiciones físicas. Siempre que exista algún cambio en las condiciones físicas se producirá una respuesta medible (Sehrawat & Gill, 2019).



Figura 1 – 04 Sensor ultrasónico HC-SR04 (ELEC-FREAKS, 2013).

²² CPU Central Processing Unit – Unidad de Procesamiento Central.

²³ GPU Graphics Processing Unit – Unidad de Procesamiento Gráfico.

²⁴ HAT Hardware Attached on Top – Hardware Montado en el Tope.

1.10.1 Sensor ultrasónico

Los sensores ultrasónicos son dispositivos electrónicos que como su nombre lo indica, utilizan señales superiores a los del sonido audible por el ser humano, son ampliamente utilizados para medir distancias hacia objetos de interés. Un ejemplo de este tipo de sensores se observa en la figura 1 – 04.

Disponen de dos componentes fundamentales, el primero denominado transmisor que emite ondas ultrasónicas las cuales viajan a la velocidad del sonido (aproximadamente 343 m/s) y el segundo denominado receptor que se encarga de recibir la señal del sonido reflejada y convertirla en una señal eléctrica. Con la finalidad de calcular la distancia entre el sensor y el objeto, se utiliza uno de los principios básicos de física:

$$distancia = velocidad * tiempo$$

Dado que este tiempo medido es el tiempo total que toma la señal en viajar desde el transmisor, impactar el objeto y posteriormente regresar al receptor, únicamente nos interesa la mitad del tiempo medido, quedando la fórmula para el cálculo de la distancia de la siguiente manera:

$$distancia = \frac{(tiempo\ medido) * (velocidad\ del\ sonido)}{2}$$

Entre las principales aplicaciones de los sensores ultrasónicos se destacan la asistencia para parqueo de automóviles, sistemas de seguridad, prevenir colisiones, detección de obstáculos en la robótica y detección del nivel de líquidos en las empresas de manufactura. A diferencia de los sensores de infrarrojo estos son menos susceptibles al humo, al gas y otras partículas en el aire lo que les hace más convenientes para esta clase de aplicaciones.

1.11 Parqueadero

La Real Academia Española, define a un parqueadero o aparcamiento como: “Lugar destinado para aparcar o estacionar vehículos”(RAE, 2014).

1.12 SPI²⁵

Un SPI es el conjunto de elementos electrónicos que permiten convertir a un parqueadero convencional en un parqueadero inteligente. Un parqueadero inteligente es aquel que se caracteriza por permitir a los usuarios (conductores) encontrar plazas donde estacionar más eficientemente, a través de la aplicación de tecnologías de la información y comunicación (Jeon et al., 2018).

²⁵ SPI Sistema de Parqueadero Inteligente.

2. Capítulo 2: Situación Actual del Conjunto Habitacional Parque Real

En este capítulo se presentarán los antecedentes y ubicación geográfica, planos estructurales y situación actual con respecto a los parqueaderos de visita del CHPR²⁶.

2.1. Antecedentes y ubicación del CHPR

Como se puede apreciar en la figura 2 – 01, el CHPR está ubicado en el centro norte de Quito, específicamente en la avenida 6 de Diciembre, entre las avenidas Gaspar de Villarroel y Granados.



Figura 2 – 01 Ubicación geográfica del CHPR (Google-Maps, n.d.).

El conjunto está conformado por 12 torres residenciales, de las cuales, 5 cuentan con locales comerciales y 62 departamentos cada una. Las otras 7 torres están compuestas por 72 departamentos cada una. Esto da un estimado de 814 familias habitando el CHPR, motivo por el cual es de suma importancia aprovechar de una mejor manera los 150 parqueaderos de visita disponibles en el conjunto.

2.2. Planos arquitectónicos de los parqueaderos de visita del CHPR

En la figura 2 – 02 se expone una parte del plano arquitectónico del subsuelo uno, en donde se pueden visualizar los parqueaderos tanto privados como de visita más próximos a la Torre

²⁶ CHPR Conjunto Habitacional Parque Real.

Almendo 4, en la misma se puede visualizar sombreados en color cian los parqueaderos de visita. Para el detalle completo de los parqueaderos destinados para visitantes favor referirse a los documentos descritos en el anexo A.

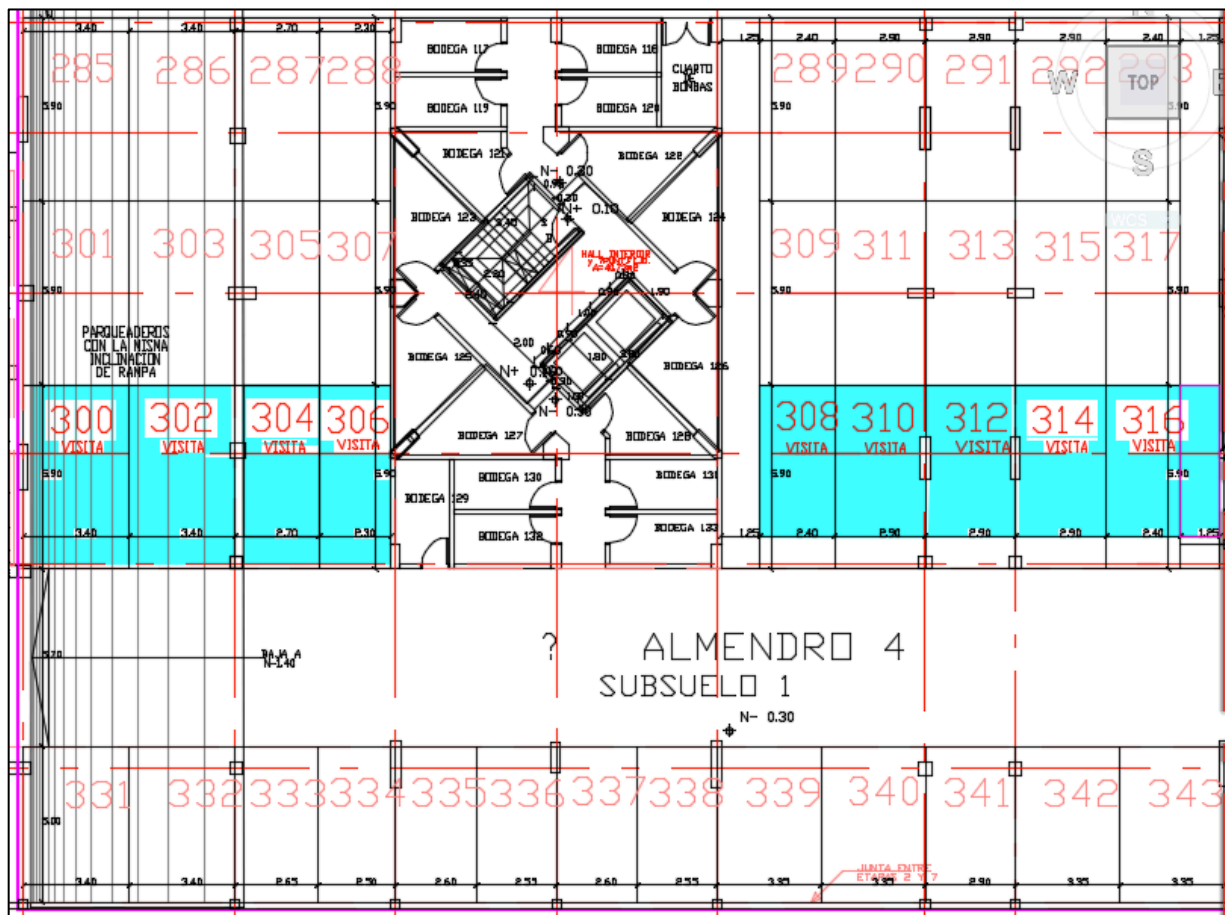


Figura 2 – 02 Plano arquitectónico de los parqueaderos del subsuelo 1 más próximos a la Torre Almendo 4. Los parqueaderos de visita están coloreados de color cian. (CHPR, 2010).

2.3. Planos de puntos de energía eléctrica en los parqueaderos de visita

En la figura 2 – 03 se expone una parte del plano arquitectónico del subsuelo uno, en el cual se pueden visualizar los parqueaderos tanto privados como de visita que están más próximos a la Torre Nogal 2, en la misma se puede visualizar sombreados en color verde los puntos de energía eléctrica más próximos. Para el detalle completo de las tomas de energía favor referirse a los documentos descritos en el anexo B.

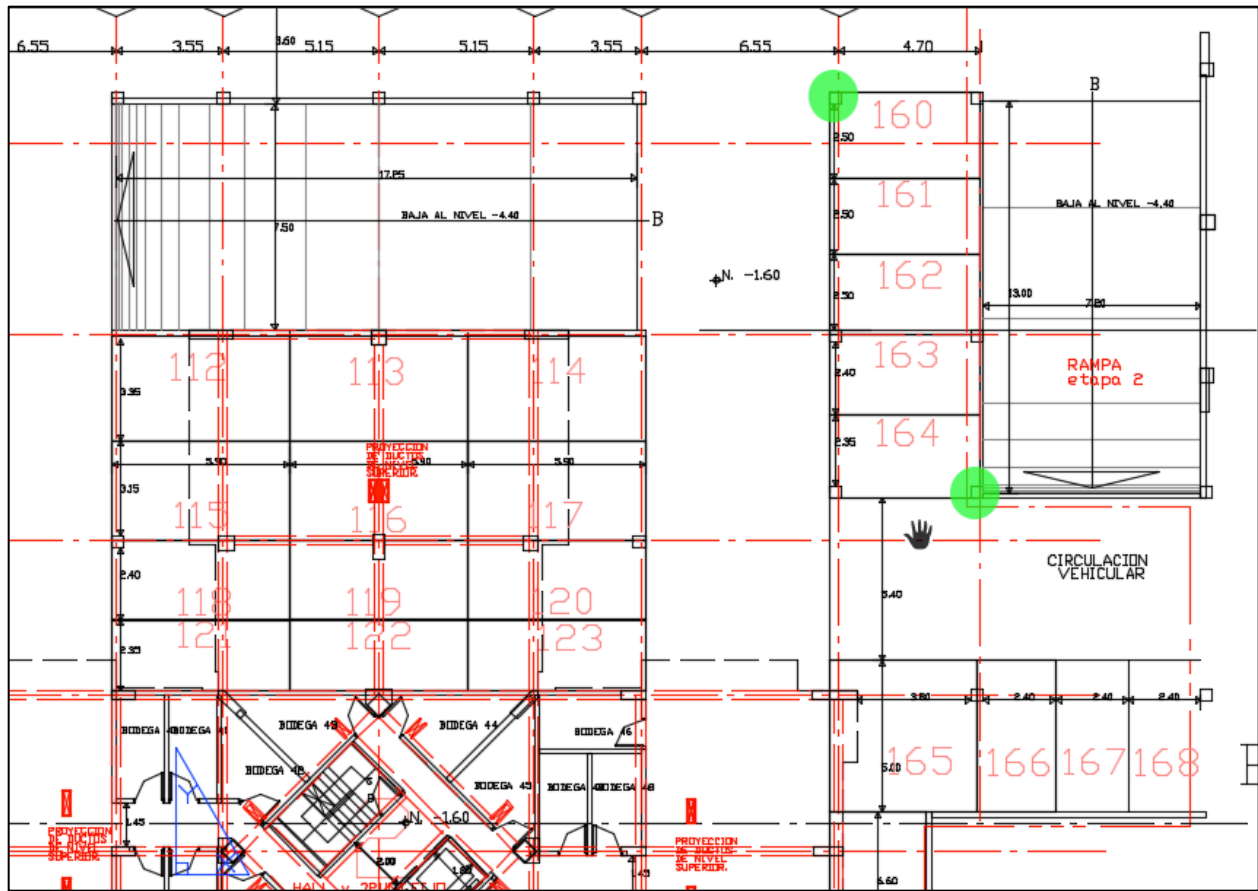


Figura 2 – 03 Plano arquitectónico de los parqueaderos del subsuelo 2 más próximos a la Torre Nogal 2. Los puntos de energía eléctrica más próximos están coloreados de color verde (CHPR, 2010).

2.4. Planos de acceso de comunicaciones para los parqueaderos de visita

En la figura 2 – 04 se expone una parte del plano arquitectónico del subsuelo uno, en el cual se pueden visualizar los parqueaderos tanto privados como de visita más próximos a la Torre Abedul 10, en la misma también se puede visualizar, destacada como una estrella de color rojo, el punto de acceso de comunicaciones más próximo en el subsuelo 1. Para el detalle completo de los puntos de comunicaciones favor referirse a los documentos descritos en el anexo C.

2.5. Problemática actual de los parqueaderos de visita del CHPR

De acuerdo con las estadísticas del INEC²⁷, la provincia de Pichincha tiene el mayor número de vehículos matriculados con 540 827, seguido por Guayas con 529 603 unidades, esto debido a que el parque automotor del país ha crecido de manera desproporcionada en los últimos años, tanto es así que para la última década se tuvo un incremento de más de 1.4 millones de vehículos. (EFE, 2019).

²⁷ INEC Instituto Nacional de Estadísticas y Censos.

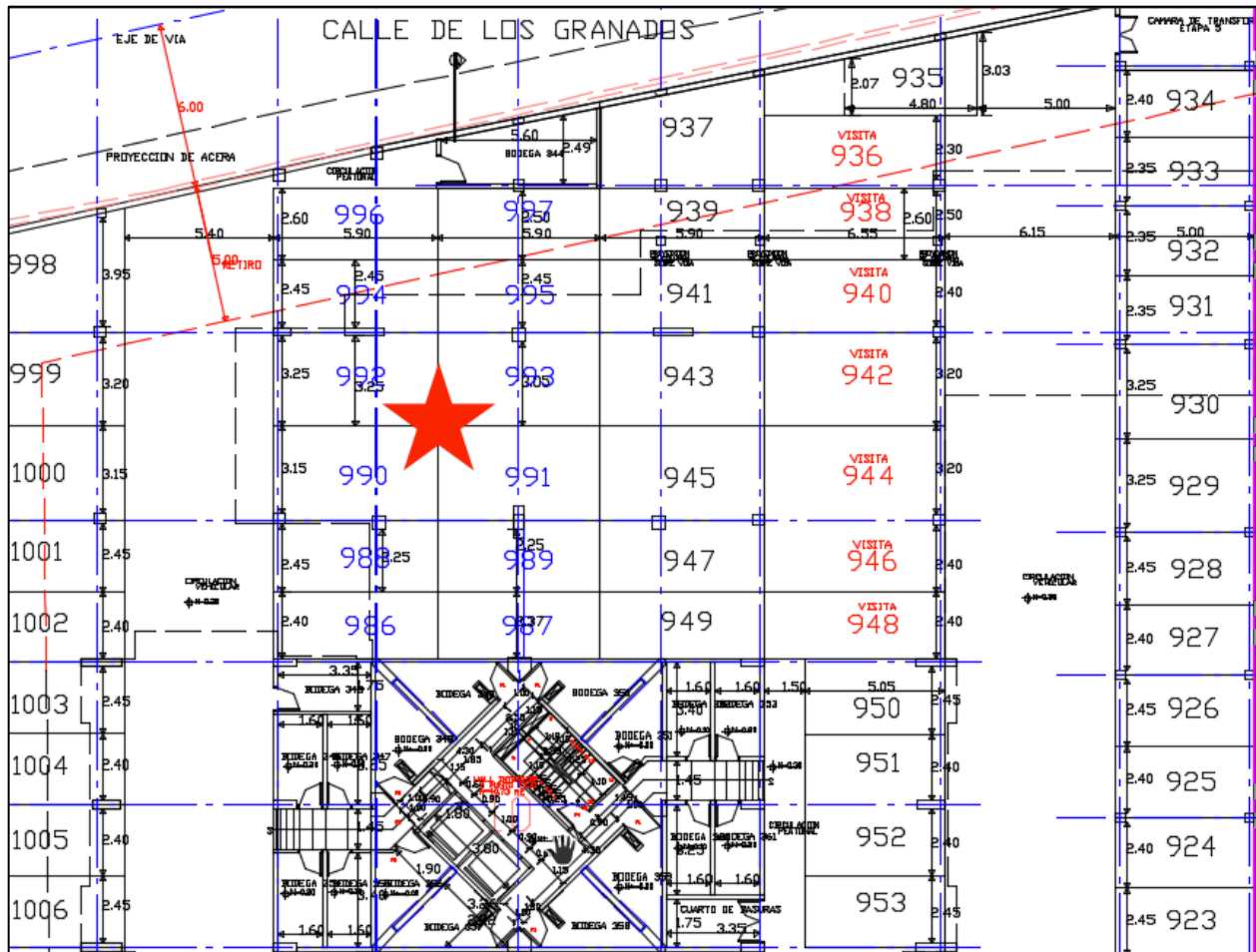


Figura 2 – 04 Plano arquitectónico de los parqueaderos del subsuelo 1 más próximos a la Torre Abedul 10. El punto de acceso de comunicaciones esta indicado como una estrella de color rojo (CHPR, 2010).

Conforme aumenta el parque automotor en la ciudad de Quito, el encontrar plazas de parqueo en lugares muy concurridos como son los centros comerciales, las universidades, los estadios y las entidades gubernamentales, resulta cada vez más complicado. Por esta razón la ubicación privilegiada del CHPR constituye un gran inconveniente cuando personas ajenas al CHPR hacen uso indebido de los parqueaderos de visita.

Como se puede apreciar en la figura 2 – 05, dado que el CHPR se encuentra en un sector privilegiado, muy próximo a entidades Gubernamentales como el Ministerio del Deporte, Agencia Metropolitana de Control, Policía Judicial y La Defensoría del Pueblo, así como también a centros educativos como la Universidad de las Américas, Unidad Educativa Santa María Eufrasia y el Instituto Tecnológico Superior Central Técnico, así como también lugares de espectáculos públicos como son el Estadio Olímpico Atahualpa, entre otros.

Lo anterior ocasiona que muchas personas decidan parquear sus automotores en los alrededores e inclusive algunas de ellas ingresan al CHPR y dejan sus vehículos estacionados en los parqueaderos de visita durante largos períodos. Por tanto es de vital importancia identificar a

estas personas que manifiestan ir a visitar a alguno de los condóminos, pero que en realidad usufructúan de los parqueaderos de visita, los mismos que podrían ser efectivamente utilizados por familiares o amistades de los verdaderos condóminos durante un tiempo razonable.



Figura 2 – 05 Entidades Gubernamentales, centros educativos y de espectáculos próximos al CHPR (Google-Maps, n.d.).

En la actualidad, al momento que entran visitantes al CHPR, la persona de la recepción simplemente pregunta: ¿A qué torre? y ¿A cuál departamento se dirige a visitar? Después si la persona ya conoce la ubicación de la torre a la cual ha manifestado se va a dirigir, simplemente avanza con su vehículo y busca lugar en el parqueadero de visitas más cercano a la mencionada torre.

Actualmente la información con respecto a quien ingresa al CHPR, a quien visita, desde que hora y hasta que hora va a permanecer, todo esto se registra en una bitácora que es llenada a mano. Debido a que existe un elevado movimiento de vehículos que ingresan y salen del interior del parqueadero, el trabajo de la persona encargada de la recepción se complica, resultando prácticamente imposible llevar un correcto control del tiempo de uso de los espacios disponibles en los parqueaderos de visita.

Como se mencionó, si la persona ya conoce la ubicación de la torre, ésta se dirige hacia la torre por su cuenta y busca un parqueadero disponible. En el caso de que sea la primera vez que la persona llega de visita, el encargado de la recepción contacta a un guardia y este guía a la persona hacia un parqueadero cercano a la torre que va a visitar. Esto no es lo más adecuado porque

distrae al guardia de sus actividades propias de guardianía e inclusive puede ocasionar la acumulación de vehículos en la entrada del CHPR.

A la fecha los parqueaderos solo disponen de una señalización vertical, que resulta ser ineficiente dado que los números y los colores ya se encuentran deteriorados tanto por el uso como por el transcurso del tiempo.

3. Capítulo 3: Propuesta de Arquitectura IoT²⁸ para un SPI²⁹

El presente estudio tiene como finalidad el diseñar e implementar un prototipo de parqueadero inteligente, basado en tecnología IoT, el cual va a utilizar como medio de transmisión inalámbrica la tecnología LoRaWAN³⁰ entre los nodos y el Gateway, tal cual se muestra en la figura 3 – 01.

Cabe destacar que para el presente estudio, no se van a considerar los efectos adversos sobre el sistema relacionados a interferencias, ya que dada la naturaleza del lugar de instalación, es decir al interior de los subsuelos de los parqueaderos residenciales, en donde prácticamente no llegan señales de tecnologías inalámbricas que operan en bandas de frecuencias similares, como la utilizada por la telefonía celular o en este caso por la tecnología LoRa³¹ a ser utilizada para el desarrollo de este proyecto.

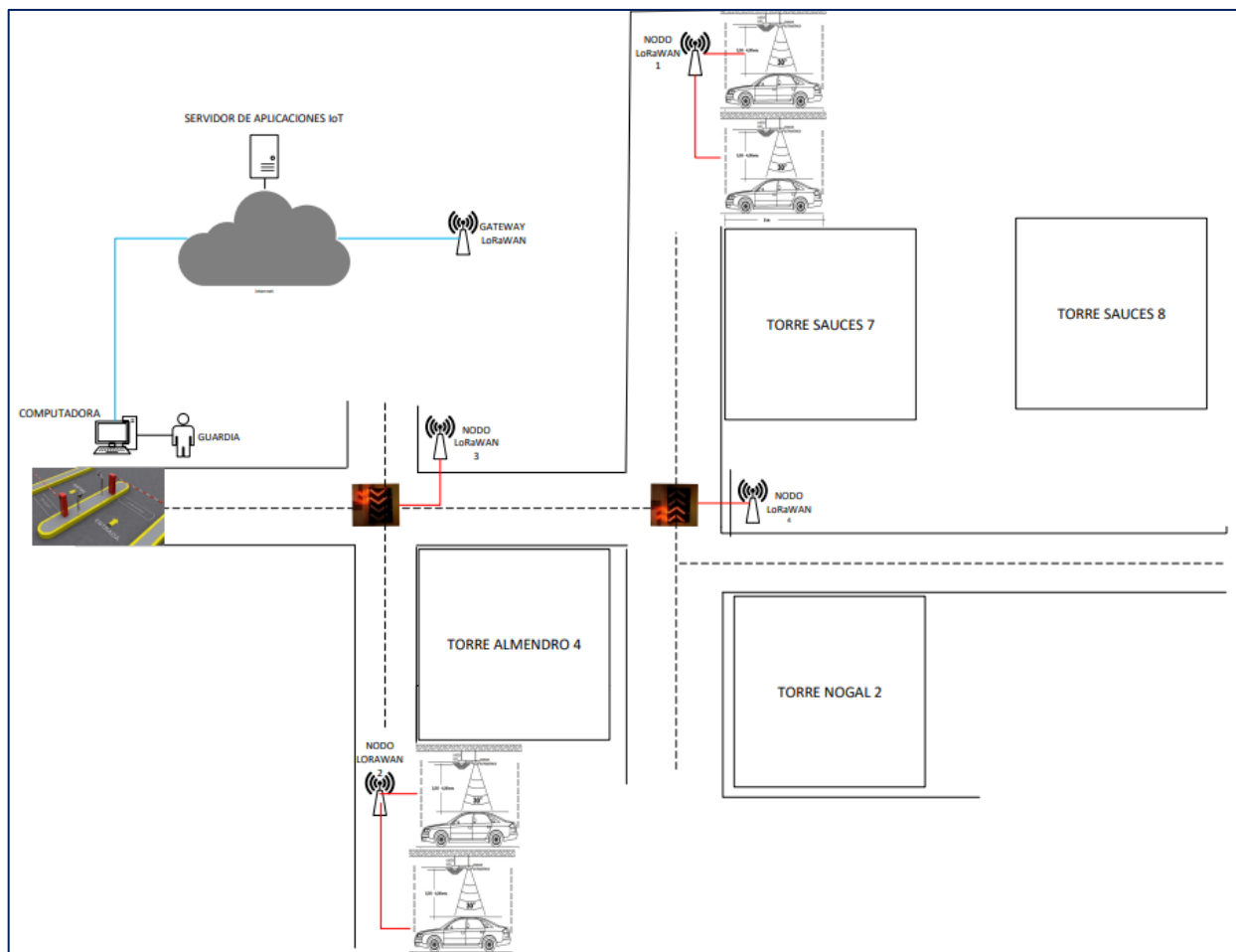


Figura 3 – 01 Funcionamiento del sistema planteado en este estudio (Mejía, 2020).

²⁸ IoT Internet Of Things – Internet de las Cosas.

²⁹ SPI Sistema de Parqueadero Inteligente.

³⁰ LoRaWAN Long Range Wide Area Network – Redes de largo Alcance de Area Extendida.

³¹ LoRa especificación de capa física que emplea modulación similar a CSS (Chirp Spread Spectrum).

La solución propuesta pretende dotar de un mayor control sobre los parqueaderos de visita, ya que se podrá disponer de una pantalla en la entrada a los parqueaderos, en la cual se podrá visualizar los parqueaderos disponibles más cercanos a cada una de las torres. Luego la persona de recepción seleccionará uno de estos parqueaderos e impartirá las respectivas indicaciones para llegar al parqueadero asignado, tomando en cuenta el sistema de señalización lumínica que permitirá guiar a los vehículos hasta llegar al parqueadero de visitas previamente asignado.

Con la solución propuesta se dispondrá de una base de datos, que permitirá realizar un control de tiempos de asignación de parqueaderos de visita en tiempo real, que adicionalmente permitirá la visualización de alertas de colores con respecto al uso de cada uno de los 150 parqueaderos de visita. En la pantalla de visualización se mantendrán en color verde los parqueaderos disponibles, en color azul mientras la visita se encuentre entre las 0 y las 4 horas, en color amarillo de 4 a 6 horas, en color naranja de 6 a 8 horas y en color rojo una vez que hayan excedido las 8 horas.

Como se mencionó anteriormente se pretende tener un sistema de señalización lumínica, el cual permitirá guiar a los vehículos de las visitas hacia el parqueadero designado por la persona de recepción. Se podrá contar en el trayecto con flechas luminosas y una vez que el vehículo se encuentre próximo al parqueadero, quien conduce podrá fácilmente identificar el parqueadero asignado ya que éste se encontrará con las luces verde y roja parpadeando secuencialmente. Las señales lumínicas de colores verde y rojo encima de cada uno de los parqueaderos para visitantes permanecerán encendidos en color verde cuando estén disponibles y en color rojo si se encuentran ocupados. Adicionalmente si un usuario que no ha sido asignado a parquearse en un determinado parqueadero utiliza uno de éstos por más de 5 minutos, se procederá a emitir una alerta de color negro en la pantalla de la persona de recepción, quien se encargará de enviar un guardia al lugar y verificar el mal uso del parqueadero.

3.1. Diseño de capas IoT

Se empezará definiendo los atributos que se deberán tener en cada una de las capas a ser desarrolladas para llegar a alcanzar la solución IoT.

PARQUEADEROS DE VISITA		
NIVEL	SUBSUELO 1	SUBSUELO 2
TORRE 1	171, 132, 135, 138	85, 43, 46, 49
TORRE 2	169, 170, 126, 129	83, 84, 37, 40
TORRE 3	310, 312, 314, 316	214, 216, 218, 220, 222
TORRE 4	300, 302, 304, 306, 308	206, 208, 210, 212
TORRE 5	560, 562, 564, 566, 568, 570, 514, 516, 518, 520	454, 456, 458, 460, 462, 464, 410, 412, 414, 316
TORRE 6	492, 594, 596, 598, 500, 502, 504, 506, 508, 510, 512	388, 390, 392, 394, 396, 398, 400, 402, 404, 406, 408
TORRE 7	719, 721, 723, 715, 717	607, 609, 611, 613, 615, 617, 619
TORRE 8	750, 748, 746, 744, 708, 709, 711, 713	640, 639, 638, 637, 636, 603, 605
TORRE 9	954, 956, 958, 960, 962, 1140, 1143, 1146	846, 848, 850, 852, 854
TORRE 10	936, 938, 940, 942, 944, 946, 948	828, 830, 832, 834, 836, 838, 840
TORRE 11	1149, 1152, 1155, 1158, 1161, 1164	1046, 1049, 1052, 1055, 1058, 1061
TORRE 12	1167, 1170, 1173	1067, 1070, 1073, 1076, 1079

Tabla 3 – 01 Distribución de los 150 parqueaderos de visita de acuerdo con su proximidad con los accesos a cada una de las torres (Mejía, 2020)

3.1.1 Capa de ambiente (Environment Layer)

El sistema se encargará de monitorizar los 150 parqueaderos de visita, los cuales al momento no son considerados como propiedad de una determinada torre, sino que constituyen en conjunto un bien comunal de todas las torres. Sin embargo, para los propósitos de este estudio se los va a asignar a cada una de las 12 torres con el afán de permitir que los parqueaderos en cuestión se encuentren lo más próximo posible a los accesos de cada una de las torres, tal cual se detalla en la tabla 3 – 01.

3.1.2 Capa de Hardware (Hardware Layer)

Como Hardware para el monitoreo de los parqueaderos, se utilizarán los denominados módulos de desarrollo, los mismos que para realizar la monitorización de las plazas de parqueadero de visitas, se interconectarán con sensores ultrasónicos, así como también con dispositivos LED³² para señalar el estado de ocupación de los parqueaderos. Finalmente, estos módulos de desarrollo en conjunto con algún tipo de solución LoRaWAN constituirán un nodo LoRa, los cuales se comunicarán con su respectivo Gateway y así podrán transmitir la información generada hacia el servidor en la nube, en donde propiamente se realizará el procesamiento de la información. En la figura 3 – 01, se puede apreciar un bosquejo de la instalación del hardware en la parte superior de los parqueaderos.

³² LED Light Emitting Diode – Diodos emisores de luz.

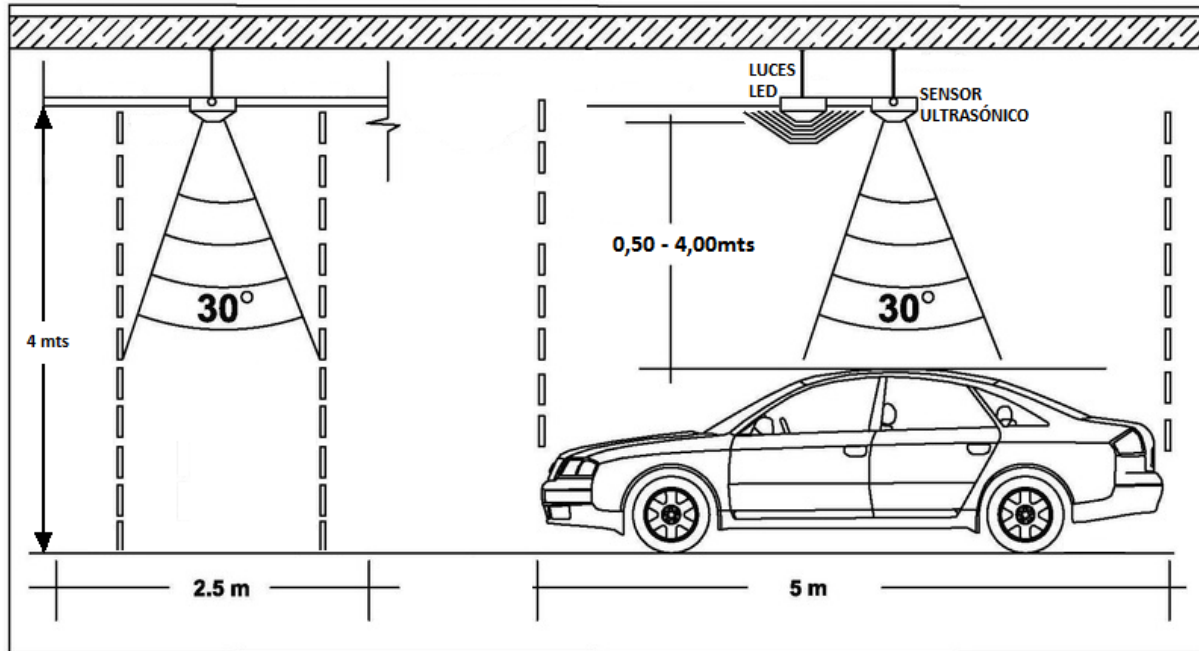


Figura 3 – 01 Funcionamiento del parqueadero planteado en este estudio. Modificado de (Chasoy, 2015).

De igual manera se utilizarán otro tipo de nodos LoRa, los cuales van a cumplir con el objetivo de señalización, para ello los módulos de desarrollo se interconectarán con matrices de leds para señalar el camino más conveniente desde la entrada hasta el parqueadero de visitas designado por la persona de recepción.

3.1.3 Capa de Red (Network Layer)

Esta capa se encarga de realizar las funciones de puerta de enlace, es decir la conexión hacia el internet y por ende debe cumplir con funcionalidades de ruteo, direccionamiento, capacidades de red, transporte, detección y corrección de errores, por lo tanto estas actividades constituyen las principales funciones que deben realizar el o los Gateways a ser utilizados por la solución LoRaWAN seleccionada, tal cual se muestra en la figura 3 – 02.

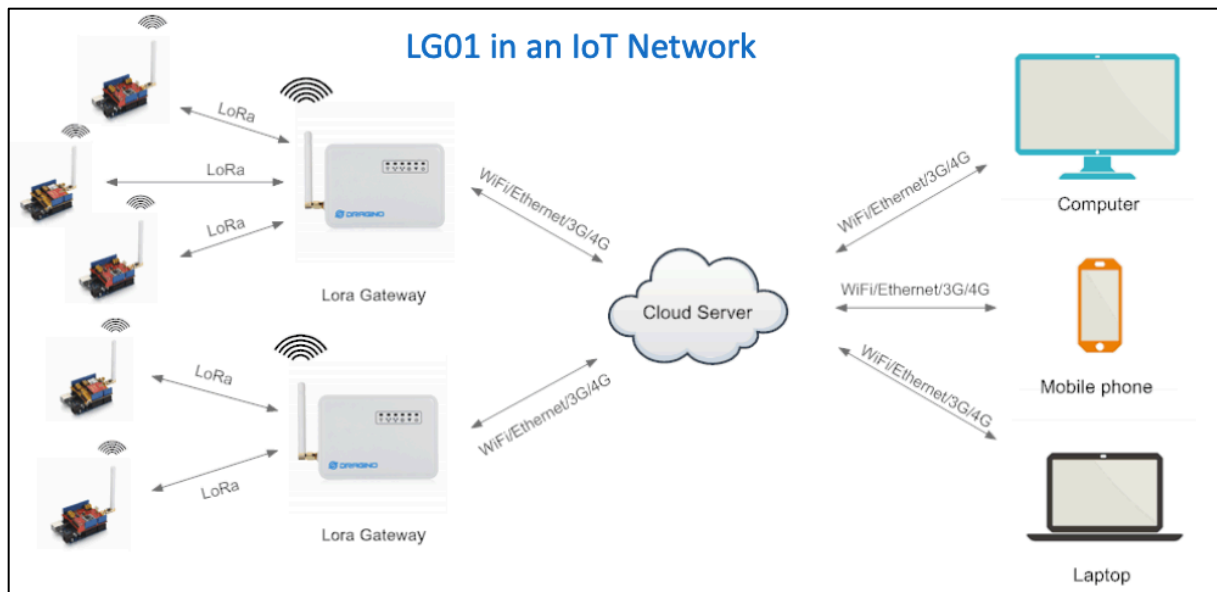


Figura 3 – 02 Representación gráfica del funcionamiento de la comunicación LoRaWAN (Dragino, 2019).

3.1.4 Capa de comunicación (Communication Layer)

Estas funcionalidades serán realizadas por las APIs³³ diseñadas por el servidor IoT en la nube, con las cuales se podrá proporcionar la interconexión entre el Gateway y el servidor IoT. Estas APIs se encargarán de permitir la transmisión de información tanto desde el Gateway hacia el servidor IoT, así como también desde el servidor IoT hacia el Gateway. Para lograr esto se deberán apoyar en la utilización de las denominadas API Keys tanto de escritura como de lectura.

En esta parte se realizará una comparación entre los distintos proveedores de servicios IoT en la nube, luego de lo cual se elegirá aquel que esté acorde de acuerdo con los requerimientos previamente establecidos.

3.1.5 Capa de servicios (Services Layer)

Estas funcionalidades serán proporcionadas por el servidor IoT en la nube, el cual realizará las funciones de monitoreo, almacenamiento, organización y visualización básica de la información recibida por parte del Gateway y de cada uno de los nodos LoRa.

3.1.6 Capa de soporte de aplicaciones y administración (Application Support & Management Layer)

Estas funcionalidades serán efectuadas por el servidor en la nube, en donde ahora sí se van a realizar y tomar acciones con la información recopilada, si la solución requiere de más de un Gateway, aquí se va a realizar el respectivo procedimiento para evitar información duplicada, ejecutando adicionalmente acciones relacionadas con el tiempo de uso de los parqueaderos, señalización para poder llegar a los parqueaderos, actualización del estado de los parqueaderos y en fin todo lo relacionado con el manejo de la aplicación.

³³ API Application Programming Interface – Interfaz de Programación de Aplicación.

3.1.7 Capa de aplicación (Application Layer)

Esta capa hace referencia a la forma en cómo se va a presentar la información al usuario, en este caso el objetivo está en disponer de una pantalla de visualización a la entrada del parqueadero, con la cual la persona de recepción podrá interactuar para conocer la situación en la que se encuentran en ese instante los parqueaderos de visita con respecto a su disponibilidad y ocupación. También al momento en que una visita arribe al conjunto, recepción estará en capacidad de utilizar la pantalla para asignar los sitios de parqueaderos disponibles más próximos a la torre que se pretende visitar. Adicionalmente en la pantalla se mostrarán las alertas en caso de presentarse un uso incorrecto de los parqueaderos de visita, tras lo cual la persona de recepción procederá a enviar un guardia a verificar el inconveniente presentado.

3.2. Requerimientos de arquitectura IoT

Con la finalidad de garantizar una correcta implementación de una arquitectura IoT ésta debe cumplir con ciertos requerimientos tanto funcionales como no funcionales, entre los cuales se pueden mencionar los siguientes:

3.2.1 Número de dispositivos

El objetivo del presente estudio es permitir la monitorización de 150 parqueaderos de visita, por lo tanto este es el número máximo de sensores requeridos, sin embargo, dada la naturaleza del estudio, y por tratarse de una propuesta, ésta será instalada únicamente en 4 parqueaderos distribuidos dos en la Torre Almendro 4 y dos en la Torre Sauces 8. Al número de nodos antes descrito se deben sumar los nodos para señalización. Dado que se desea optimizar las capacidades de los denominados kits de desarrollo, se buscará utilizar al máximo su capacidad de pines GPIO³⁴. Por este motivo y de acuerdo con las pruebas a ser realizadas, se pretende que cada nodo permita por lo menos monitorizar cinco parqueaderos de visita y así maximizar la cobertura y al mismo tiempo minimizar los costos de implementación.

3.2.2 Conectividad y Comunicación

Por cuanto se propone emplear la tecnología LoRaWAN, y dependiendo de los resultados de las pruebas de cobertura que se realicen más adelante, se utilizará una comunicación desde los nodos LoRa con uno o varios Gateways. A los Gateways se les dotará de conexión a internet mediante la interconexión con la red de comunal del CHPR³⁵. Para ello se va a utilizar el plano de acceso a comunicaciones detallado previamente en el numeral 2.4.

3.2.3 Gestión de dispositivos

Dadas las características de la propuesta, se requerirá tener la capacidad de controlar los dispositivos remotamente únicamente al momento de presentarse una visita. Una vez que la persona de recepción selecciona un parqueadero de visita, el servidor IoT procederá a realizar las respectivas acciones para configurar los nodos de visualización para trazar el mejor camino hacia el parqueadero asignado. También enviará la respectiva notificación al nodo que controla

³⁴ GPIO General Purpose Input Output – Pines de Entrada/Salida de Propósito General.

³⁵ CHPR Conjunto Habitacional Parque Real.

el respectivo parqueadero para que éste entre en modo de espera por el vehículo, haciendo parpadear sus luces verde y roja secuencialmente.

3.2.4 Recolección, Análisis y Acción sobre la información

De acuerdo con lo propuesto, se desea que el servidor se encargue de mantener el registro del estado de ocupación de los parqueaderos de visita, proporcionando las respectivas alertas en caso de que un vehículo se encuentre estacionado por un tiempo mayor al permitido, luego de lo cual y una vez cumplido el plazo establecido, se procederá a alertar a la persona de recepción con el fin de que ella se encargue de tomar las respectivas medidas correctivas.

3.2.5 Escalabilidad

El sistema va a ser diseñado para cumplir con el monitoreo de los 150 parqueaderos de visita, sin embargo, dado que este estudio es una propuesta se instalará el sistema solamente en 4 parqueaderos, sin embargo, el sistema estará listo para que en el futuro en caso de requerirlo, pueda ser escalado incrementando su capacidad de control y monitorización de todos los parqueaderos.

3.2.6 Seguridad

La seguridad es uno de los aspectos fundamentales a tomar en cuenta al momento de desarrollar aplicaciones IoT, por lo cual se deben tomar consideraciones para la protección de la información en caso de ser ésta de índole personal. Para el sistema propuesto, dado que el mismo se va a encontrar únicamente monitorizando los parqueaderos de visita, éstos no constituyen ningún riesgo para las personas que habitan el conjunto, sin embargo, para precautelar la información generada se tomarán las siguientes precauciones para evitar inconvenientes:

- Cambiar las contraseñas predeterminadas y ajustar la configuración de seguridad para satisfacer las necesidades específicas.
- Desactivar o deshabilitar cualquier función que no se necesite.
- Mantener siempre actualizado el firmware del Gateway para que éste se encuentre protegido contra vulnerabilidades de seguridad conocidas.
- En términos de configuración de aplicaciones en dispositivos, revisar los permisos que requieren y limitar el acceso otorgado a estas apps.
- Modificación de las API keys del sistema por lo menos una vez al mes para garantizar que éstas no puedan ser vulneradas por personas ajenas al CHPR.

3.2.7 Energía

Para el desarrollo del presente estudio, se utilizarán las tomas de energía eléctrica comunal descritas en los planos de puntos de energía eléctrica detallados en el numeral 2.3. A partir de estas tomas será posible disponer de la energía de 110 V y mediante un transformador se tendrá la energía en voltaje continuo, necesario para el módulo de desarrollo y demás componentes electrónicos.

3.2.8 Visualización de la información:

La información recopilada y procesada por el servidor IoT, será visualizada mediante cualquier dispositivo con conexión a internet y con un navegador que disponga de las apropiadas credenciales para el acceso al portal.

Desde el portal web a ser diseñado para la aplicación, será posible visualizar en tiempo real la información relacionada al estado actual de los parqueaderos de visita, en donde se podrá conocer información con respecto a los parqueaderos de visita vacíos y ocupados. Para estos últimos se podrá visualizar información con respecto al tiempo restante disponible para el uso de los parqueaderos asignados, el departamento al cual se encuentra visitando. En el caso de detectarse un inapropiado uso del parqueadero, se mostrará como multado y empezará a registrar el tiempo desde el cual se encuentra haciendo mal uso del mismo para poder calcular la respectiva multa.

También de acuerdo con el tiempo que haya transcurrido desde que un vehículo se encuentra haciendo mal uso del parqueadero de visita, se enviarán alertas mediante Twitter hacia los principales encargados de la administración del CHPR.

3.3. Requerimientos de cobertura

Para proporcionar el mejor funcionamiento posible de la solución de SPI planteada en el presente estudio, y con la finalidad de maximizar la cobertura del sistema propuesto dentro de ambos subsuelos que conforman parte de los parqueaderos privados y de visita del CHPR, se procederán a realizar varias pruebas de RSSI³⁶, con lo cual se podrá definir el mejor lugar para instalar el o los Gateways Lora necesarios y así garantizar una correcta cobertura del sistema para los 150 parqueaderos de visita disponibles en el CHPR.

3.4. Selección de componentes

Como se mencionó anteriormente, para lograr la implementación del presente estudio se utilizarán algunos tipos de tecnologías y sensores, para los cuales a continuación se realizará el respectivo análisis costo beneficio y se seleccionaran los dispositivos más adecuados para cumplir con los requerimientos de arquitectura IoT antes descritos.

3.4.1 Selección de módulos de desarrollo

Para seleccionar el tipo de módulo de desarrollo a ser utilizado, se realizará una comparación entre tarjetas de desarrollo basadas en microcontroladores, teniendo como principal representante a las tarjetas de desarrollo Arduino y las denominados SBC³⁷, teniendo como principal representante a las tarjetas de desarrollo Raspberry Pi.

³⁶ RSSI Received Signal Strength Indicator – Indicador de fuerza de señal Recibida.

³⁷ SBC Single Board Computers – Computador en una sola tarjeta.

COMPARACIÓN ENTRE DISPOSITIVOS ARDUINO VS RASPBERRY PI		
Características	Arduino	Raspberry Pi
Capacidad de procesamiento	Es un microcontrolador, constituye una parte de un computador. Puede correr solo un programa o rutina una y otra vez.	Es un minicomputador, utiliza un sistema operativo llamado Raspbian OS. Puede correr múltiples programas al mismo tiempo.
Complejidad para interconectar sensores y otros componentes	Se pueden interconectar sensores y actuadores de una manera simple.	Requiere de actividades complejas como la instalación de librerías y software para poder interactuar con sensores y otros componentes.
Costo	Está disponible a bajo costo. Aproximadamente en las unidades y decenas de dólares.	Más costosos. Aproximadamente en las varias decenas a centenas de dólares.
Facilidad de conexión a red	Requiere de hardware adicional, y se debe de programar este software adicional correctamente.	Puede conectarse al internet utilizando componentes embebidos.
Puertos USB	Dispone de un único puerto para conectarse a la computadora.	Dispone de 4 puertos para conectarse con otros dispositivos.
Procesador	Utiliza procesador de la familia AVR ³⁸ Atmega328P.	Utiliza procesador de la familia ARM ³⁹ .
Memoria RAM	Alrededor de los KB	Alrededor de los GB.
Problemas por fallos de energía	Ninguno. Si se apaga deja de ejecutar el programa, al restablecerse la energía reinicia el programa.	Puede corromperse el sistema operativo y el software. Siempre debe apagarse correctamente como un computador.
Lenguajes de programación	Utiliza C/C++	Se recomienda utilizar Python, pero viene preinstalado con C/C++ y Ruby.
Dispositivos con mayor cantidad de pines GPIO	Arduino MEGA 2560: 54 pines	Raspberry Pi 2, 3 Y 4: 26 pines

Tabla 3 – 02 Comparación entre las características proporcionadas por los dispositivos Arduino Vs. Raspberry Pi (ELECTRONICS-HUB, 2017).

Para el diseño del estudio propuesto, las características proporcionadas por la tecnología Arduino se acoplan de una mejor manera a los requisitos necesarios para la implementación de la solución

³⁸ AVR se refiere a la familia de microcontroladores de la 8- bits del fabricante Atmel.

³⁹ ARM es una familia de procesadores de computadora con un set de instrucciones reducido.

IoT planteada. En primera instancia éstos son más económicos, adicionalmente para la monitorización de los parqueaderos no se requiere de mucha capacidad de procesamiento, así como también la interconexión de los sensores ultrasónicos es más simple. La tecnología Arduino proporciona una mayor cantidad de pines de entrada y salida, lo cual permite usar una menor cantidad de módulos de desarrollo para cubrir una mayor cantidad de parqueaderos, además permitiría a futuro economizar en la cantidad de dispositivos LoRaWAN necesarios, a esto se agrega que los repuestos son más económicos.

Tomando en cuenta que se desea utilizar los módulos de desarrollo con la mayor cantidad de pines GPIO posibles, se utilizarán las tarjetas de desarrollo Arduino Mega 2560, que puede apreciarse en la figura 3 – 03 y cuyas características son descritas en la Tabla No. 3 – 03.

TARJETA DE DESARROLLO ARDUINO MEGA2560	
Características	ATmega2560
Voltaje de operación	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada Input (limite)	6-20V
Pines de entrada o salida digitales	54 (de los cuales 15 permiten salidas PWM)
Pines de entrada analógicos	16
Corriente DC por cada Puerto de entrada o salida	20 mA
Corriente DC para pines a 3.3V	50 mA
Memoria Flash	256 KB (8 KB son usados por el bootloader)
SRAM	8 KB
EEPROM	4 KB
Velocidad del reloj	16 MHz
Pin del led montado en la placa	13
Largo	101.52 mm
Ancho	53.3 mm
Peso	37 g

Tabla 3 – 03 Características de la tarjeta de desarrollo Arduino Mega 2560 (Arduino, 2020).

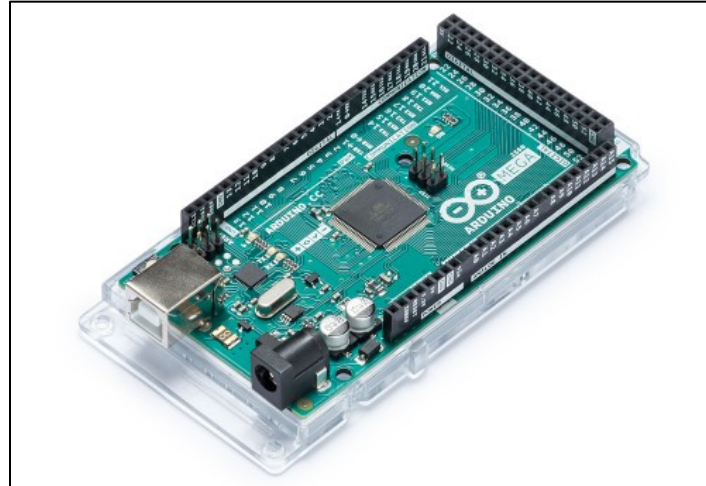


Figura 3 – 03 Tarjeta de desarrollo Arduino Mega2560 (Arduino, 2020).

3.4.2 Selección sensores ultrasónicos

Para seleccionar el tipo de sensor ultrasónico apropiado, se realizará una búsqueda en el mercado local a fin de comparar los mismos y determinar el más adecuado para los propósitos del presente estudio.

COMPARACIÓN ENTRE SENSORES ULTRASÓNICOS DISPONIBLES EN EL MERCADO LOCAL				
Características	US-100/Y401	HY-SRF05	HC-SR04	JSN-SR04T
Voltaje de operación	2.4 – 5 VDC	5VDC	5VDC	5VDC
Corriente de operación	15 mA	15 mA	15 mA	30 mA
Frecuencia de operación	40 Hz	40 Hz	20 Hz	20 Hz
Mínimo rango de medición	2 cm	2 cm	2 cm	25 cm
Máximo rango de medición	400 cm	400 cm	400 cm	450 cm
Rango de error	3 mm		Hasta 3 mm	Hasta 5 mm
Ángulo de medición	15 grados	15 grados	15 grados	< 50 grados
Costo por unidad (tomado de mercadolibre.com.ec)	\$5,00	\$4,99	\$2,50	\$19,99

Tabla 3 – 04 Comparación entre las características técnicas de los sensores ultrasónicos US-100/Y401 vs HY-SRF05 vs HC-SR04 vs JSN-SR04T, tomado de (Naylamp-Mechatronics, 2017).

Analizados algunos tipos de sensores ultrasónicos disponibles en el mercado local, se considera que el sensor ultrasónico HC-SR04, cumple con todas las características necesarias para el presente estudio, además que éste presenta las mejores prestaciones en cuanto a la relación costo - beneficio.

3.4.3 Selección de solución LoRaWAN

Para seleccionar el tipo de solución LoRaWAN a ser utilizado, se investigará en el mercado local las distintas posibilidades que se presentan a fin de comparar los mismos y determinar el más adecuado para los propósitos del presente estudio. En la tabla 3 – 05 se muestran algunas soluciones disponibles en el mercado local.

COMPARACIÓN ENTRE SOLUCIONES LoRa DISPONIBLES EN EL MERCADO LOCAL		
Características	LoRa ESP-32	LoRa Dragino Shield
Voltaje de operación	3.3 – 7 VDC	3.3 o 5 VDC
Frecuencia de operación	868 – 915 MHz	915/868/433 MHz
Sensibilidad	-148 dBm	-148 dBm
Potencia de TX	20 dBm	20 dBm
Tasa de transferencia		Hasta 300 kbps
Distancia máxima de transmisión en espacio abierto	2.8 Km	Hasta 15 Km
Compatible con Arduino	Si	Si
Tiene la implementación LoRaWAN desarrollada	No, solamente el protocolo LoRa. Se requiere circuitería adicional para hacer de Gateway e interconectarse a internet.	Si, a través de diferentes tipos de módulos como el LG-01 o el LG-02, los cuales permiten la conexión directa al internet.
Características adicionales	Dispone de módulo para comunicación Wifi, dispone de circuitería para carga de batería de litio y tiene una pantalla OLED embebida de 0,96 pulgadas	Existe otra versión que incluye la funcionalidad de GPS.
Costo por unidad en el exterior (tomado de amazon.com)	\$24,99	\$27,99

Tabla 3 – 05 Comparación entre soluciones LoRa disponibles en el mercado local: ESP Lora 32 vs LoRa Dragino Shield, tomado de (Dragino-Technology-Co-Limited, n.d.) y de (Amazon, n.d.).

Dadas las condiciones actuales del mercado nacional, el cual se encuentra seriamente afectado por la situación de emergencia sanitaria y el estado de Excepción decretado a nivel nacional, la adquisición de componentes electrónicos de este tipo va a requerir de procedimientos de importación y de desaduanización, los mismos que no van a lograr ser cumplidos hasta algunos meses después de que finalice el período de estado excepción.

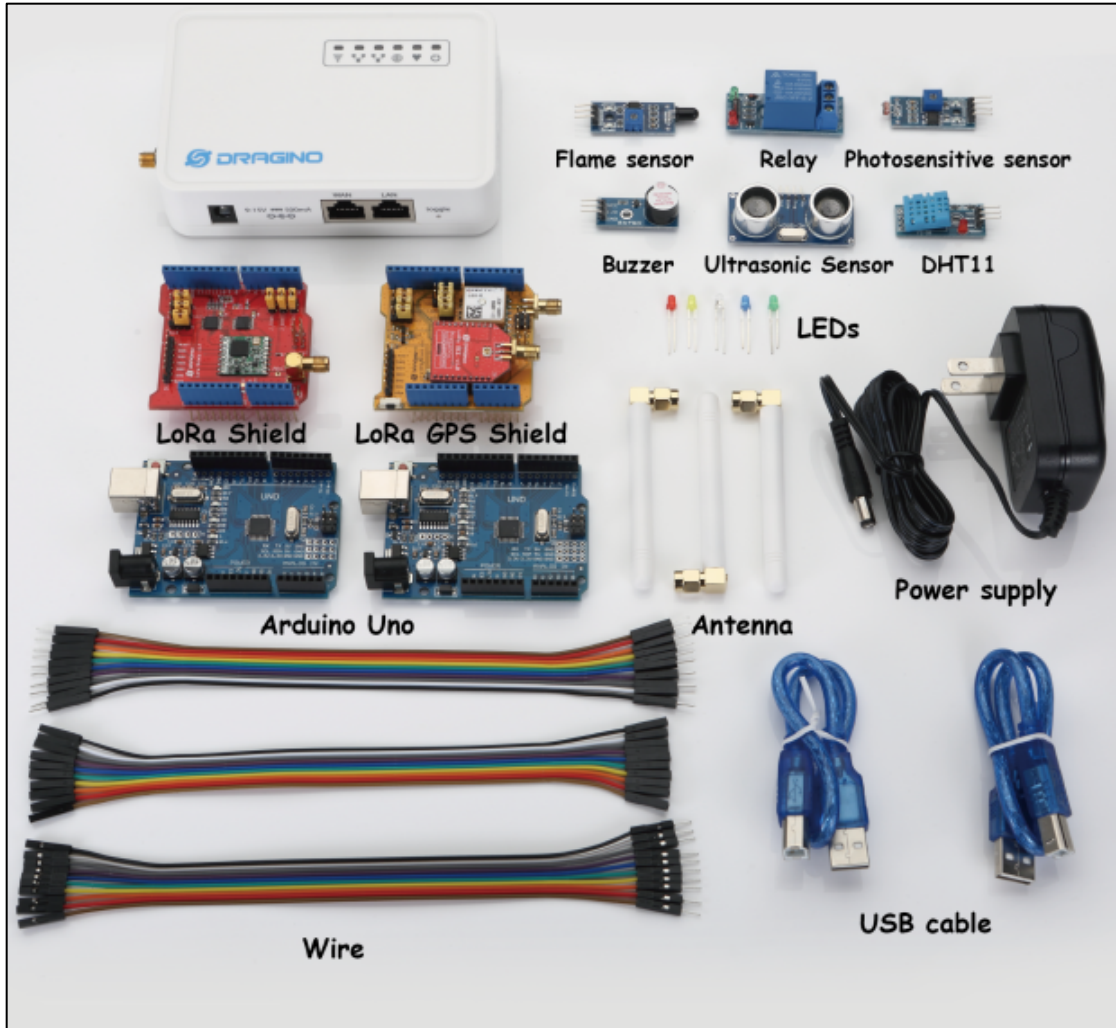


Figura 3 – 03 Kit de desarrollo LoRaWAN Dragino (Dragino, 2019).

Por la situación antes descrita, fue necesario contactar con un proveedor de dispositivos electrónicos a nivel local, para coordinar la adquisición de un kit de desarrollo de los diseñados por la empresa Dragino Technology Corporation Limited, el cual permite el aprendizaje e incursión hacia la tecnología LoRa, tanto para desarrolladores nuevos como para experimentados, tal cual se muestra en la figura 3 – 03. El kit se compone de un Gateway con su antena, dos nodos LoRa con sus antenas, dos tarjetas de desarrollo Arduino Uno, cables USB y para la interconexión de las tarjetas Arduino con dispositivos electrónicos externos, algunos tipos de sensores, fuente de energía y un manual con ejemplos para empezar el prototipado de aplicaciones del tipo IoT. (Dragino, 2019).

Se utilizará el LoRa Gateway LG-02 que viene incorporado en el kit de desarrollo de Dragino, el cual cuenta con la suficiente capacidad de procesamiento para pasar la información recopilada por la tarjeta de desarrollo Arduino Mega2560 hacia el servidor de aplicaciones IoT en la nube, mediante la utilización de ciertas APIs de comunicación y cuyas características se detallan en la tabla 3 – 06.

CARACTERÍSTICAS TÉCNICAS DEL LoRa GATEWAY LG-02	
Hardware: Parte Linux	Procesador: ar9331 400 MHz Memoria RAM: 64 MB Memoria Flash: 16 MB
Interfaces	Voltaje de entrada: 12 VDC Puertos RJ45 10M/100M: 2 Puerto USB Host: 1 LoRa: 2
Especificaciones WIFI	IEEE 802.11 b/g/n Banda de frecuencia: 2.4 a 2.462 GHz Potencia de transmisión: <ul style="list-style-type: none"> - 11n: mcs7/15: 11db mcs0: 17dB - 11b: 18db - 11g 54M: 12dB - 11g 6M: 18db Sensibilidad: <ul style="list-style-type: none"> - 11g 54M: -71dbm - 11n 20M: -67dbm
Especificaciones LoRa	Rango de frecuencia: <ul style="list-style-type: none"> - Banda 1 (HF): 862 - 1020 MHz - Banda 2 (LF): 410 - 528 MHz Máximo presupuesto de potencia: 168 dB Tasa de datos programable: hasta 300 kbps Alta sensibilidad: hasta -148 dBm. Modulaciones soportadas: <ul style="list-style-type: none"> - FSK, GFSK, MSK, GMSK, LoRaTM and OOK Construido con sensores de temperatura e indicadores de batería baja

Tabla 3 – 06 Características técnicas del LoRa GATEWAY LG-02 (Dragino, 2019).

3.4.4 Selección de servidor IoT en la nube

Para la definición del servidor IoT en la nube que se empleará en el presente estudio, se procederá a analizar las características ofrecidas por algunos de servidores IoT en la nube y las bondades ofrecidas por cada uno de ellos. Estas características son descritas en la tabla 3 – 07.

COMPARACIÓN ENTRE SERVIDORES IoT EN LA NUBE				
Características	Thing Speak	Kaa Project	Thinger.io Cloud	M2MLight
Hardware compatible	Arduino, Raspberry Pi, entre otros.	Arduino, Raspberry Pi, entre otros.	Arduino, Raspberry Pi, entre otros.	Arduino, Raspberry Pi, entre otros.
Ámbito de utilización	Smart Home y prototipos	Desarrolladores y empresariales	Desarrolladores y Empresariales	Aficionados al IoT y prototipos
Ventajas	Interfaz amigable, Dispone de widgets, Integración con redes sociales.	Integración con redes sociales, integración con redes sociales	En servicio de pago permite la configuración de dominios propios.	Interfaz amigable, Integración con aplicaciones.
Infraestructura de código abierto	Si	Si	Si	Si
Método de escritura de información en los canales	APIS ⁴⁰ REST ⁴¹ y MQTT ⁴²	APIS REST y MQTT en servicio de pago	APIS REST y MQTT	APIS REST y MQTT
Método de lectura de información en los canales	APIS REST y MQTT	APIS REST y MQTT en servicio de pago	APIS REST y MQTT	APIS REST y MQTT
Visualización de información	Información en pantalla principal del sitio web y mediante widgets.	Información en pantalla principal del sitio web y mediante widgets.	Información en pantalla principal del sitio web y mediante widgets.	Información en pantalla principal del sitio web.
Procesamiento de información adicional	Se puede complementar con la utilización de herramientas de MATLAB. Funciones adicionales	Disponible en servicio de pago.	Disponible en servicio de pago.	Disponible en servicio de pago.

⁴⁰ APIS subrutinas que permiten al programador hacer uso de códigos de programación previamente desarrollados.

⁴¹ REST interfaz entre sistemas mediante Hyper Text Transfer Protocol para intercambio de datos.

⁴² MQTT protocolo liviano para publicar y recuperar información de máquina a máquina.

COMPARACIÓN ENTRE SERVIDORES IoT EN LA NUBE				
Características	Thing Speak	Kaa Project	Thinger.io Cloud	M2MLight
	disponibles en servicio de pago.			
Aplicativos para reaccionar ante cambios en la información	APIs para envío automático de correos electrónicos o interacción con Twitter o aplicaciones para automatizar respuestas ante eventos preconfigurados.	APIs para envío automático de correos electrónicos o interacción con Twitter o aplicaciones para automatizar respuestas ante eventos preconfigurados.	APIs para envío automático de correos electrónicos o interacción con Twitter o aplicaciones para automatizar respuestas ante eventos preconfigurados.	Se puede conectar con aplicaciones desarrolladas para dispositivos Android. Permite automatizar respuestas ante eventos preconfigurados.
Limite de dispositivos configurables	En modo gratis limitado a 4 canales de 8 campos cada uno. Limitado a 3 millones de mensajes por año. Y limitado a 800 alertas vía correo electrónico por año.	Hasta 5 dispositivos con cuenta gratis.	Ilimitados, con servicio de pago.	Ilimitados
Facilidad para interacción con base de datos	Permite la descarga manual de la base de datos, pero no una interacción y procesamiento directamente. Se requeriría de otro servidor en la nube para el procesamiento de la información y recuperación de información enviada hacia su servidor.	Permite la descarga manual de la base de datos, pero no una interacción y procesamiento directamente. Se requeriría de otro servidor en la nube para el procesamiento de la información y recuperación de información enviada hacia su servidor.	Permite la descarga manual de la base de datos, pero no una interacción y procesamiento directamente. Se requeriría de otro servidor en la nube para el procesamiento de la información y recuperación de información enviada hacia su servidor.	Al ser un servidor IoT ofrecido por una empresa local, el proveedor permite la gestión y administración de la base de datos, así como también la interacción, procesamiento, utilización de otros servicios como servidor web y dominio de red y salida a internet.

Tabla 3 – 07 Características ofrecidas por varios servidores IoT en la nube. Tomado de: (MathWorks 2019), (Kaalot-Technologies 2020), (Thinger.io 2018) y (Albuja 2020).

Se seleccionará el servicio proporcionado por M2MLight, dado que para el presente estudio se requiere en mayor medida de la utilización e interacción con bases de datos como las ofrecidas por el sitio web M2MLight, así como también por su completo apoyo tanto logístico como operativo para el desarrollo del presente estudio y ya que por defecto permite la utilización de una manera gratis su sistema IoT. Además, se permitirá el libre acceso a su servidor y base de datos e interactuar directamente con la información y posteriormente facilitará la incorporación de una aplicación de servidor web, mediante la cual se va a interactuar con respecto al estado en tiempo real de los parqueaderos de visitas del CHPR.

4. Capítulo 4: Prototipo funcional.

En este capítulo se presenta el desarrollo de todo el equipamiento tecnológico tanto de hardware como de software necesario para poder cumplir con los requerimientos necesarios del SPI⁴³ propuesto en el presente estudio.

4.1 Definición del número de Gateways LoRa⁴⁴ necesarios

Considerando que los módulos LoRa Shield disponen del mismo tipo de radio Transmisor/Receptor que el de los Gateways, para definir el mínimo número de Gateways necesarios para proporcionar una correcta cobertura del SPI para los parqueaderos de visita de ambos subsuelos del CHPR⁴⁵, se procedió a instalar un módulo LoRa Shield en los lugares descritos en las figuras 4 – 01 y 4 – 02, tras lo cual se efectuaron pruebas de RSSI⁴⁶ al interior de los parqueaderos de visita del CHPR con un segundo módulo LoRa Shield, como se puede apreciar en la Figura 4 – 03. En la tabla 4 – 01 se puede apreciar una muestra de los resultados obtenidos, para el detalle completo de las pruebas realizadas, favor referirse al anexo D.



Figura 4 – 01 Lugar No. 1 para la instalación provisional del módulo LoRa shield en el subsuelo 1 del CHPR para pruebas de RSSI al interior de los parqueaderos de visita del CHPR (Mejía, 2020).

⁴³ SPI Sistema de Parqueadero Inteligente.

⁴⁴ LoRa especificación de capa física que emplea modulación similar a CSS (Chirp Spread Spectrum).

⁴⁵ CHPR Conjunto Habitacional Parque Real.

⁴⁶ RSSI Received Signal Strength Indicator – Indicador de fuerza de señal Recibida.

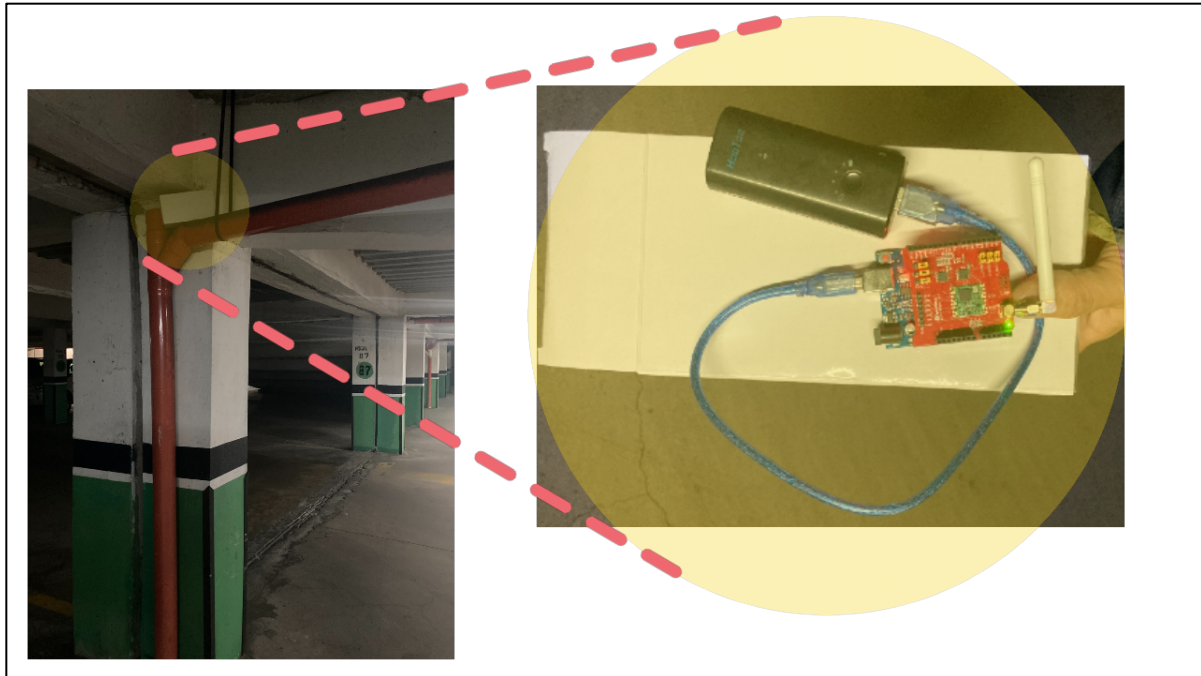


Figura 4 – 02 Lugar No. 2 para la instalación provisional del módulo LoRa shield en el subsuelo 1 del CHPR para pruebas de RSSI al interior de los parqueaderos de visita del CHPR (Mejía, 2020).

PRUEBAS CON INDICADOR DE FUERZA DE SEÑAL RECIBIDA (RSSI)				
TORRE	SUBSUELO	No. De Parqueadero	PRUEBA 1	PRUEBA 2
TORRE 4	S1	300	-52	-65
		302	-56	-65
		304	-59	-66
		306	-64	-75
		308	-77	-93
		310	-79	-95
	S2	206	-66	-67
		208	-67	-55
		210	-70	-52
		212	-69	-58
TORRE 8	S1	750	-94	Sin señal
		748	-94	Sin señal
		746	-93	Sin señal
		744	-89	Sin señal
		708	-76	Sin señal
		709	-74	Sin señal
		711	-75	-93
		713	-75	-94

PRUEBAS CON INDICADOR DE FUERZA DE SEÑAL RECIBIDA (RSSI)				
TORRE	SUBSUELO	No. De Parqueadero	PRUEBA 1	PRUEBA 2
	S2	640	-95	-84
		639	Sin señal	-84
		637	-95	-87
		603	-96	-70
		605	-96	-70
		607	-94	-68

Tabla 4 – 01 Muestra a modo de ejemplo de los valores de RSSI medidos en los parqueaderos de visita, de acuerdo con 2 de los lugares en donde se instaló provisionalmente los módulos LoRa Shield (Mejía, 2020).

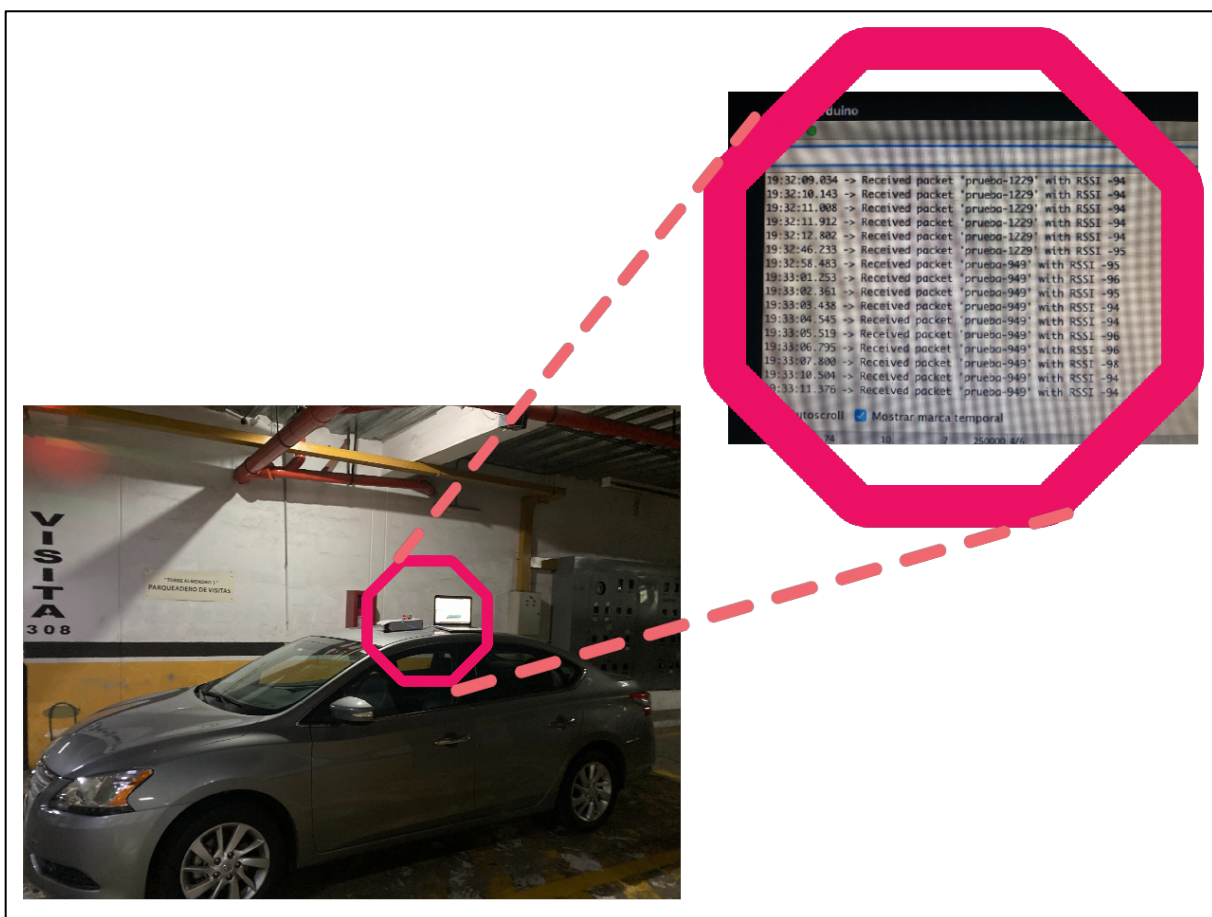


Figura 4 – 03 Pruebas de RSSI al interior de los parqueaderos del CHPR (Mejía, 2020).

De acuerdo con los resultados mostrados en la tabla 4 – 01 y en el anexo D, se determina que mínimo se requiere de 2 Gateways Dragino para poder dotar de un RSSI adecuado a todos los parqueaderos de visita. Dado que en el presente estudio únicamente se va a centrar en proporcionar la funcionalidad de parqueaderos inteligentes a dos parqueaderos de la Torre Almendro 4 y dos parqueaderos de la Torre Sauces 8, se va a optar por instalar un único Gateway

a pocos metros de la ubicación No. 1 descrita previamente en la figura 4 – 01 y cuya instalación se muestra en la figura 4 – 04.



Figura 4 – 04 Instalación del Gateway LG-01 en la ubicación No. 1 (Mejía, 2020).

4.2 Diseño de atributos y funcionalidades del Gateway LoRa

Como se describió anteriormente, el Gateway se va a encargar de recopilar la información generada por los nodos LoRa tras lo cual mediante su interconexión hacia internet, transmitirá esta información hacia el servidor IoT⁴⁷ M2MLight, en donde se realizará el respectivo procesamiento de la información generada por los mismos. A continuación, el servidor procederá a controlar la visualización del estado actual de los parqueaderos de visita y posteriormente se enviarán los respectivos comandos de regreso hacia los nodos para controlar el estado actual de los parqueaderos de visita y de ser el caso activar las distintas matrices de leds y leds de visualización para señalar los parqueaderos asignados a los visitantes.

Para cumplir con los requerimientos antes descritos, se va a definir un funcionamiento del Gateway de acuerdo con el diagrama de flujo mostrado en la figura 4 – 05.

⁴⁷ IoT Internet of Things - Internet de las Cosas.

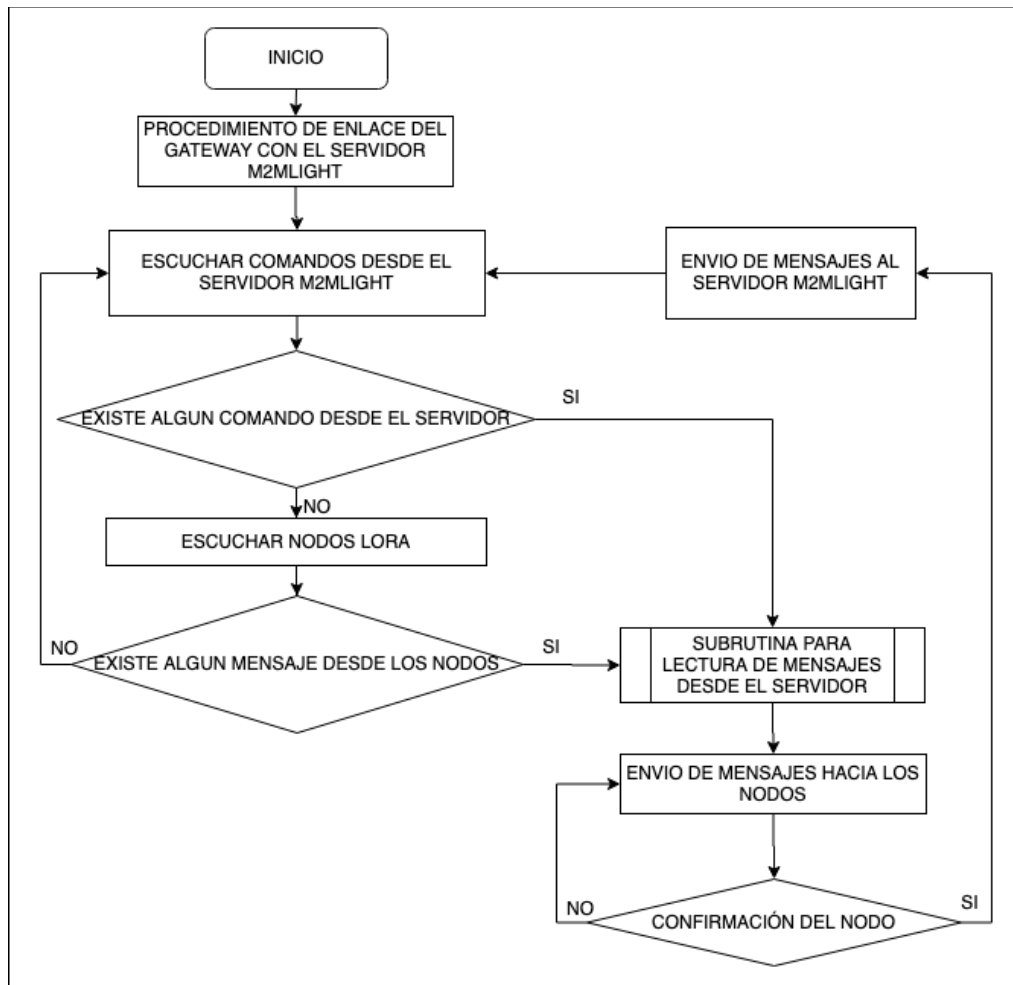


Figura 4 – 05 Diagrama de flujo para el funcionamiento del Gateway LoRa (Mejía, 2020).

4.3 Diseño de atributos y funcionalidades de los Nodos LoRa

Como se definió anteriormente, se va a disponer de dos tipos de nodos LoRa, los del primer tipo se van a encargar de monitorizar y de señalar el estado de ocupación actual de los parqueaderos de visita, mientras que los del segundo tipo de van a encargar de señalar el camino que debe tomar la visita para llegar al parqueadero de visita asignado por la persona de recepción.

4.3.1 Nodo LoRa para parqueaderos

Para cumplir con los requerimientos del presente estudio, se va a disponer de dos de estos nodos, los cuales en adelante se denominarán nodo No. 1000 y nodo No. 2000 respectivamente, los mismos que se encargarán de monitorear y señalar la ocupación de 2 parqueaderos de visita cada uno. Así pues, cada nodo se conformará de una tarjeta de desarrollo al cual se acoplarán los elementos detallados según el diagrama de bloques mostrado en la figura 4 – 06 y que van a cumplir con el funcionamiento descrito en el diagrama de flujo mostrado en la figura 4 – 07.

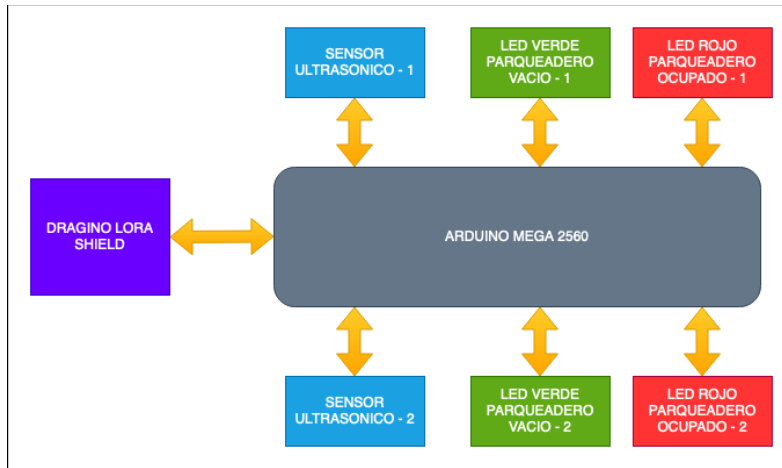


Figura 4 – 06 Diagrama de bloques del funcionamiento de los Nodos LoRa para monitoreo de parqueaderos (Mejía, 2020).

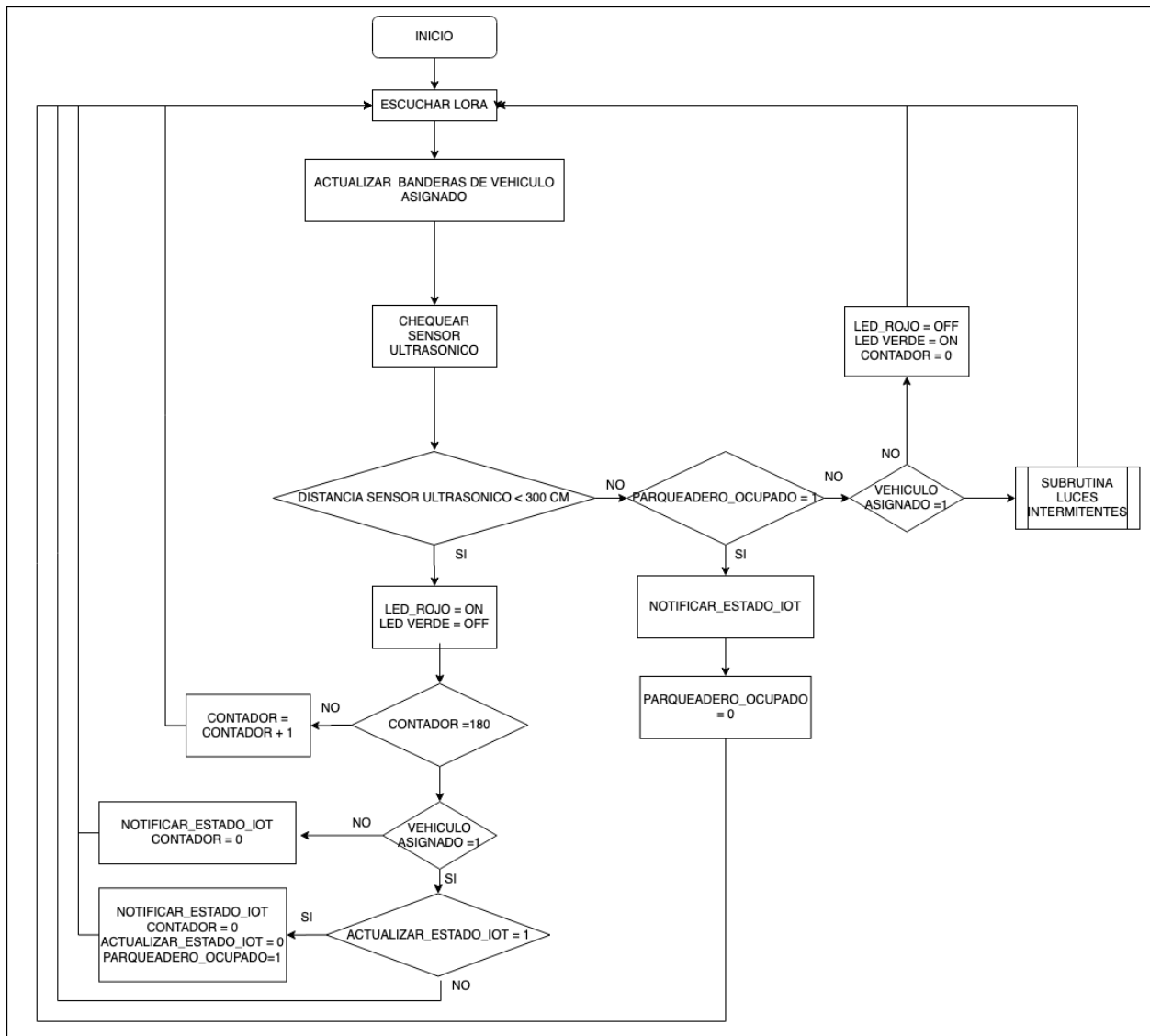


Figura 4 – 07 Diagrama de flujo del funcionamiento de los Nodos LoRa para monitoreo de parqueaderos (Mejía, 2020).

4.3.1.1 Programación de los Nodos LoRa para parqueaderos

La programación de la tarjeta de desarrollo es realizada mediante el software IDE⁴⁸, propio de las tarjetas de desarrollo Arduino, el cual gracias a su implementación de un interfaz serial virtual, permite la depuración del código de programación.

En la figura 4 – 08, se puede apreciar parte del código de programación desarrollado así como también su comprobación en tiempo real mediante la utilización de la interfaz serial del programador, para mayores detalles respecto al código de programación implementado, favor referirse al anexo E.

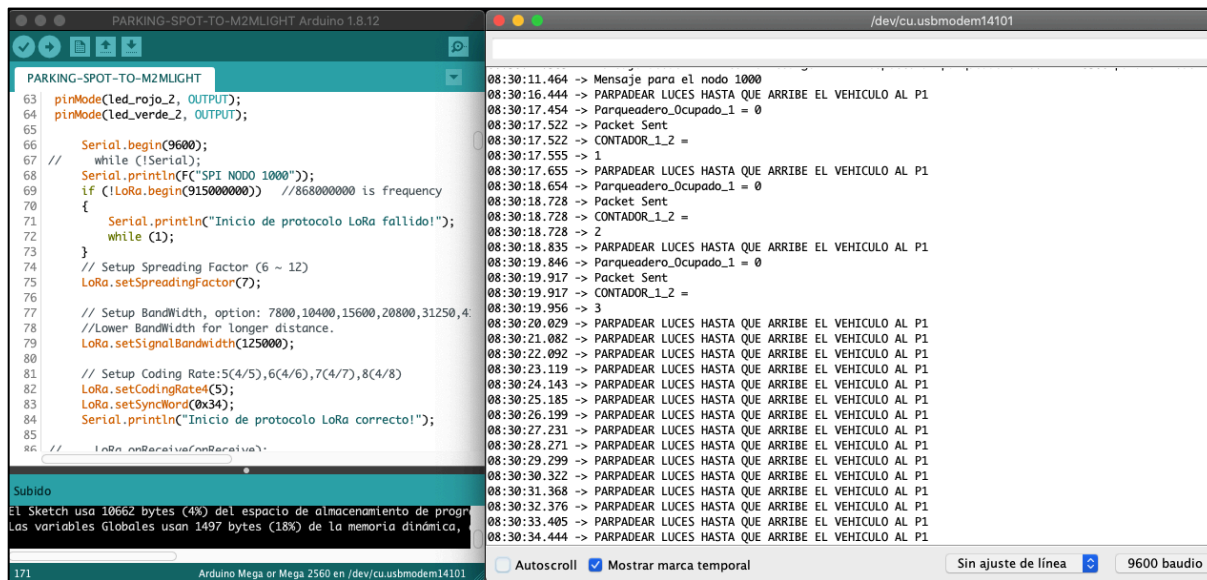


Figura 4 – 08 Ventana del software IDE con una parte del código implementado junto con ventana de la interfaz serial para depuración en tiempo real del código de programación de los Nodos LoRa para monitoreo de parqueaderos (Mejía, 2020).

4.3.1.2 Esquema de conexión de los Nodos LoRa para parqueaderos

En la figura 4 – 09, se puede apreciar el esquema de conexiones utilizado para el desarrollo de este tipo de nodos. Como se puede observar, para el monitoreo y señalización de cada parqueadero se utilizan únicamente 4 pines de la tarjeta de desarrollo, lo que permitirá en el futuro controlar 10 parqueaderos adicionales con cada tarjeta de desarrollo, dados los más de 40 pines disponibles.

⁴⁸ IDE Integrated Development Environment – Ambiente de Desarrollo Integrado para tarjetas Arduino.

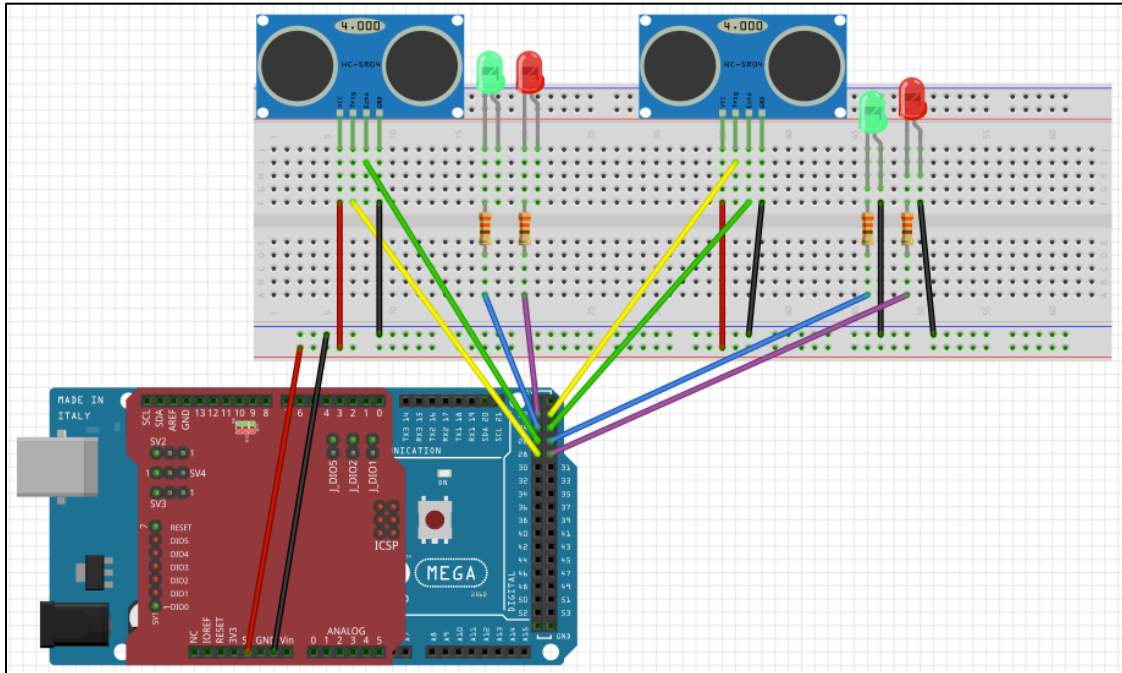


Figura 4 – 09 Esquema de conexiones necesarias para pruebas iniciales en el laboratorio entre la tarjeta de desarrollo Arduino Mega con el módulo Dragino LoRa Shield y el resto de los elementos electrónicos (Mejía, 2020).

4.3.2 Nodo LoRa para señalización

Para cumplir con los requerimientos definidos, dado que el control de las matrices de leds demanda de gran velocidad de procesamiento, no es adecuado que el mismo microcontrolador, que se encuentra escuchando el protocolo LoRa, realice a la vez el control de las matrices ya que durante la secuencia de control de las mismas, puede ocurrir que se deje de escuchar los mensajes MQTT provenientes desde el servidor IoT o desde los nodos.

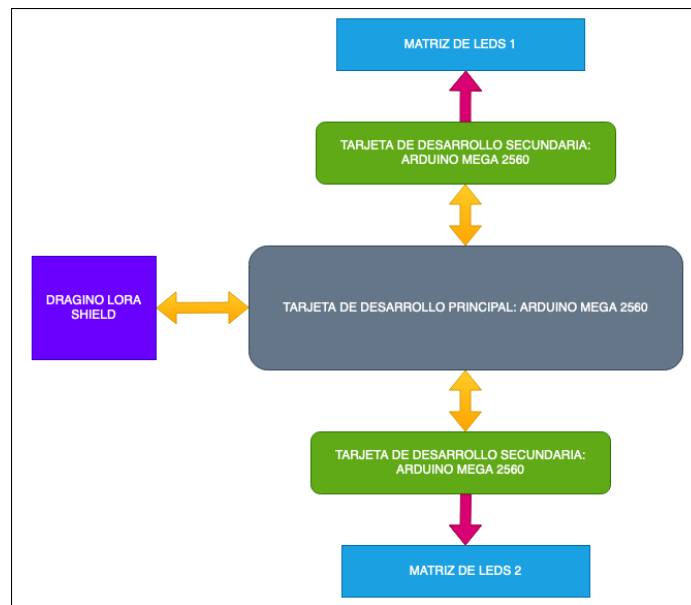


Figura 4 – 10 Diagrama de bloques del funcionamiento de los Nodos LoRa para señalización de ruta (Mejía, 2020).

Es así que de acuerdo con lo mostrado en el diagrama de bloques de la figura 4 – 10, se requerirá de una tarjeta de desarrollo principal la cual se encontrará constantemente escuchando los mensajes provenientes desde el servidor IoT. Una vez que haya sido detectado un mensaje de señalización, la tarjeta de desarrollo principal procederá a notificar al respecto a las tarjetas de desarrollo secundarias, las cuales se encontrarán controlando propiamente los arreglos de matrices de leds para la señalización del respectivo trayecto, y que van a cumplir con el funcionamiento descrito en el diagrama de flujo mostrado en la figura 4 – 11.

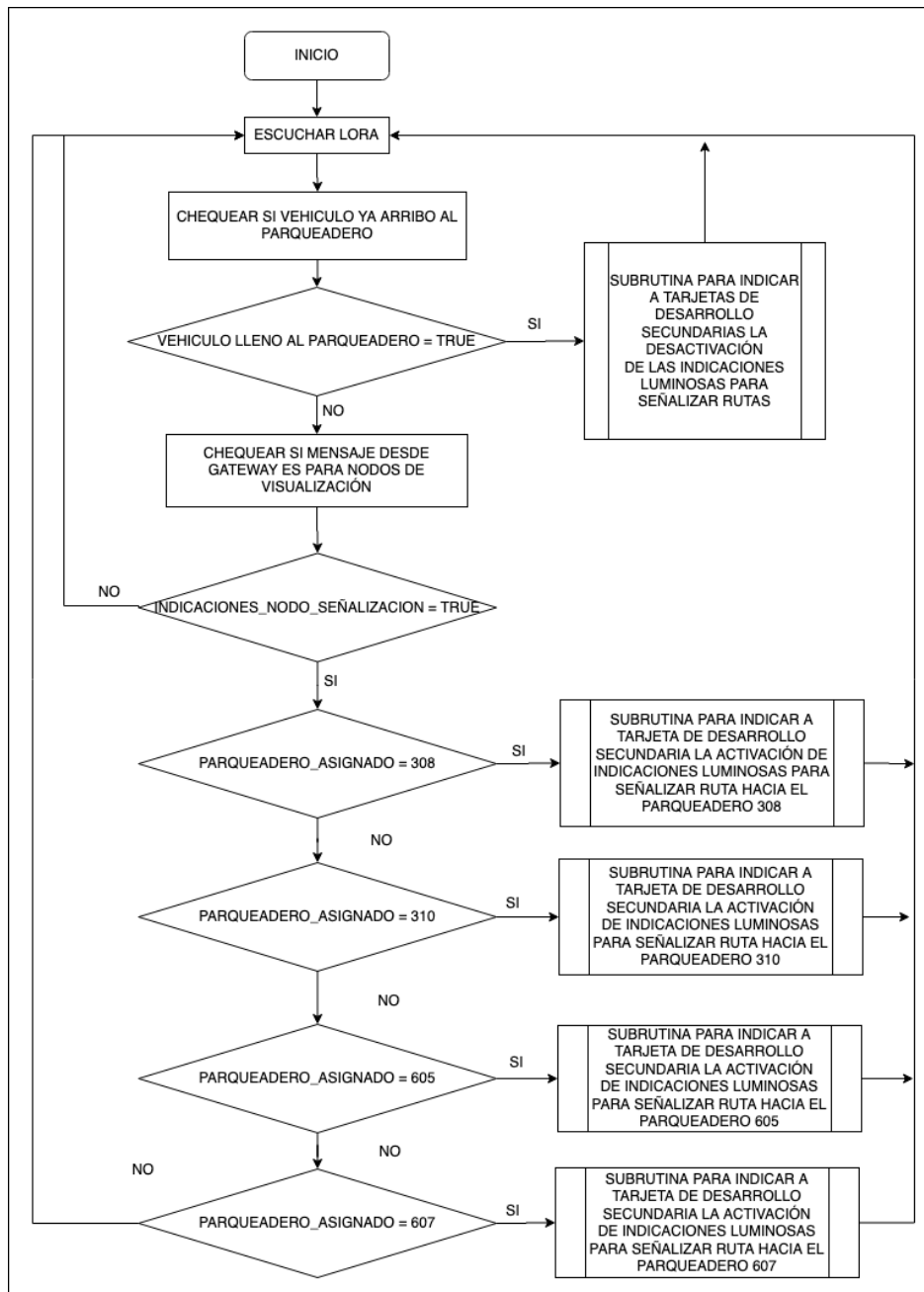


Figura 4 – 11 Diagrama de flujo del funcionamiento de la tarjeta de desarrollo principal de los Nodos LoRa para señalización de ruta (Mejía, 2020).

4.3.2.1 Programación de los Nodos LoRa para señalización de ruta

En la figura 4 – 12, se puede apreciar parte del código de programación desarrollado, así como también su comprobación en tiempo real mediante la utilización de la interfaz serial del programador, para mayores detalles respecto al código de programación implementado, favor referirse al anexo F.

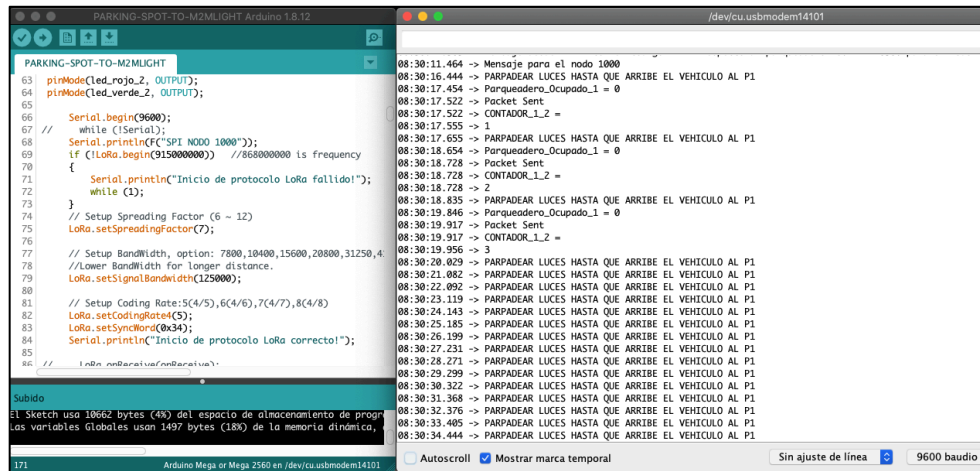


Figura 4 – 12 Ventana del software IDE con una parte del código implementado junto con ventana de la interfaz serial para depuración en tiempo real del código de programación de los Nodos LoRa para señalización de ruta (Mejía, 2020).

4.3.2.2 Esquema de conexión de los Nodos LoRa para señalización de ruta

En la figura 4 – 13, se puede apreciar el esquema de conexiones utilizado para el desarrollo de este tipo de nodos. Como se puede observar, para aspectos de señalización de ruta todavía quedan disponibles dos interfaces físicas seriales (RX1/TX1 y RX2/TX2) de la tarjeta de desarrollo principal, lo cual permitirá habilitar en el futuro dos arreglos de matrices de leds adicionales en el caso de que se desee mejorar los aspectos de señalización de ruta, esto solo si se considera las interfaces físicas. Si se emplean interfaces seriales, entonces por software será factible controlar muchas más.

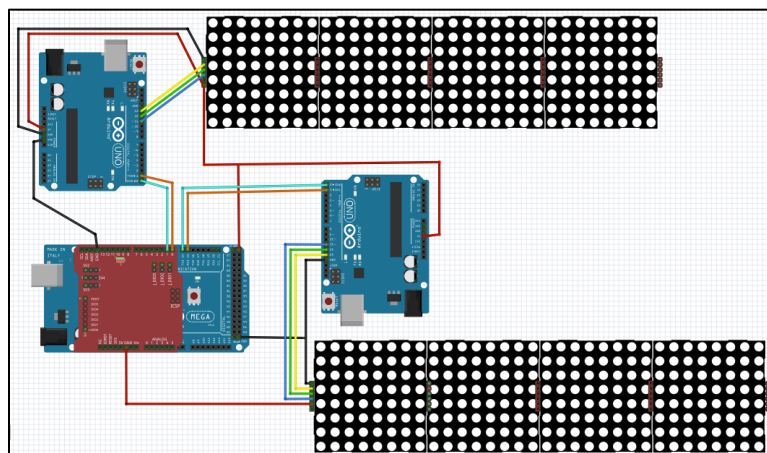


Figura 4 – 13 Diagrama de flujo del funcionamiento de la tarjeta de desarrollo principal de los Nodos LoRa para señalización de ruta (Mejía, 2020).

4.4 Interconexión del Gateway LG-02 con el servidor IoT M2MLight

Después de crear la respectiva cuenta gratuita en la página web del servidor M2MLight.com, como se puede apreciar en la figura 4 – 14, se accede a la sección “ACCOUNT”, desde donde se obtiene el respectivo API-KEY⁴⁹ de usuario. A continuación se accede a la sección SENSORS, en donde se procede a registrar los cuatro sensores, que servirán para configurar y monitorear el estado de los cuatro parqueaderos de visita. Como se puede apreciar en la figura 4 – 15, una vez creados éstos, se tienen las respectivas API-KEYS que deben ser configuradas a continuación en el Gateway.

The screenshot shows the 'Account details' page on the M2MLight website. The page includes a navigation menu with options like Home, Ip Cameras, Sensors, Actuators, Alerts, Subdomains, and Related Links. The account details form contains the following information:

- *Email Address: xvrmj@live.com
- *Name: Francisco Mejía
- *Password: [Redacted]
- *Re-enter Password: [Redacted]
- *Time Zone: America/Bogota -05:00:00
- Domain: [Redacted]
- Town/City: Quito
- Address: 6 de diciembre and Gaspar de Villaroel
- Phone: 593999233702
- Telegram Id: xvrmj
- Api_key: jfDr1njci7

Figura 4 – 14 Pantalla “Account” dentro del servidor M2MLight (Mejía, 2020).

The screenshot shows the 'Sensors' page on the M2MLight website. It features a table with columns for Sensor Name, Sensor Type, Control Type, Measure, Email msg, Interval, Set Point, Action Under Set Point, Action Over Set Point, Sample Time, Api_key, and Status. Below the table are links for 'Live Sensor Panel' and 'Live Sensor Map'.

Sensor Name	Sensor Type	Control Type	Measure	Email msg	Interval	Set Point	Action Under Set Point	Action Over Set Point	Sample Time	Api_key	Status
T4-0308	Single Value	Manual	PARQUEADERO-A4-308	Disable					0	rku51njcnq	✓
T4-0310	Single Value	Manual	PARQUEADERO-A4-310	Disable					0	tx8o1njcnt	✓
T8-0605	Single Value	Manual	PARQUEADERO-S8-605	Disable					0	sg4k1njcnu	✓
T8-0607	Single Value	Manual	PARQUEADERO-S8-607	Disable					0	f1lQ1njcnv	✓

Figura 4 – 15 Pantalla “Sensors” dentro del servidor M2MLight (Mejía, 2020).

⁴⁹ API-KEY – Clave de interfaz que se utiliza para autenticar a un usuario o aplicación.

Figura 4 – 16 Configuración de Gateway Dragino LG-02, de acuerdo con los parámetros del servidor M2MLight (Mejía, 2020).

Para la interconexión entre el Gateway LG-02 y el servidor IoT M2MLight se utilizará el protocolo MQTT⁵⁰, el cual es configurado en el Gateway dentro de la pestaña “SERVICE”, seleccionando la opción “MQTT” de acuerdo con lo mostrado en la figura 4 – 16. Aquí se empleará el API-KEY de usuario generado previamente al crear la cuenta en el servidor M2MLight. Adicionalmente de acuerdo con la figura 4 – 17, en la sección “MQTT CHANNELS” se registrarán los sensores con sus respectivas API-KEYS generadas previamente.

Finalmente también se debe registrar la información en la pestaña “SERVICE” seleccionando la opción “LORAWAN GATEWAY” de acuerdo con lo mostrado en la figura 4 – 18.

Local Channel in /var/iot/channels/	Remote Channel in IoT Server	Write API Key		
1000	1	rku51nrcnq	Edit	Delete
2000	2	tx8o1nrcnt	Edit	Delete
3000	3	sg4K1nrcnu	Edit	Delete
4000	4	f1lQ1nrcnv	Edit	Delete

Figura 4 – 17 Configuración de Gateway Dragino Lg-02, de acuerdo con los parámetros del servidor M2MLight (Mejía, 2020).

⁵⁰ MQTT – Es un protocolo de comunicación M2M (machine-to-machine) de tipo message queue.

dragino-1d9458 Status System Network Service Logout

LoRa Gateway Settings

Configuration to communicate with LoRa devices and LoRaWAN server

LoRaWAN Server Settings

IoT Service: LoRaRAW forward to MQTT serv

Debug Level: Many verbose output

Service Provider: --custom--

Server Address: m2mlight.com

Server Port: 1883

Gateway ID: jFDr1nrci7

Figura 4 – 18 Configuración del Gateway Dragino LG-02, de acuerdo con los parámetros del servidor M2MLight (Mejía, 2020).

Dentro de la misma pestaña “SERVICE” seleccionando la opción “LORAWAN GATEWAY”, se procede a configurar los principales parámetros de funcionamiento del protocolo LoRa⁵¹ como son frecuencia, codificación, ancho de banda, entre otros, tal cual se puede apreciar en la figura 4 – 18.

dragino-1d9458 Status System Network Service Logout

Channel 1 Radio Settings

Radio settings for Channel 1

RadioA Frequency (Unit:Hz): 915000000

RadioA Spreading Factor: SF7

RadioA Coding Rate: 4/5

RadioA Signal Bandwidth: 125 kHz

RadioA Preamble Length: 8
Length range: 6 ~ 65536

RadioA LoRa Sync Word: 52
Value 52(0x34) for LoRaWAN

Encryption Key: Encryption Key

Figura 4 – 19 Configuración de los parámetros LoRa en el Gateway Dragino LG-02 (Mejía, 2020).

⁵¹ LoRa especificación de capa física que emplea modulación similar a CSS (Chirp Spread Spectrum).

4.4.1 Validación de Información con respecto a parqueaderos en servidor IoT M2MLight

Dentro del servidor M2MLight en la sección “Sensors”, al hacer clic dentro de los sensores registrados se puede visualizar información histórica y en tiempo real del estado de los parqueaderos. Ésta puede ser en forma gráfica o tabulada como se muestra en la figuras 4 – 20 y 4 – 21 respectivamente.

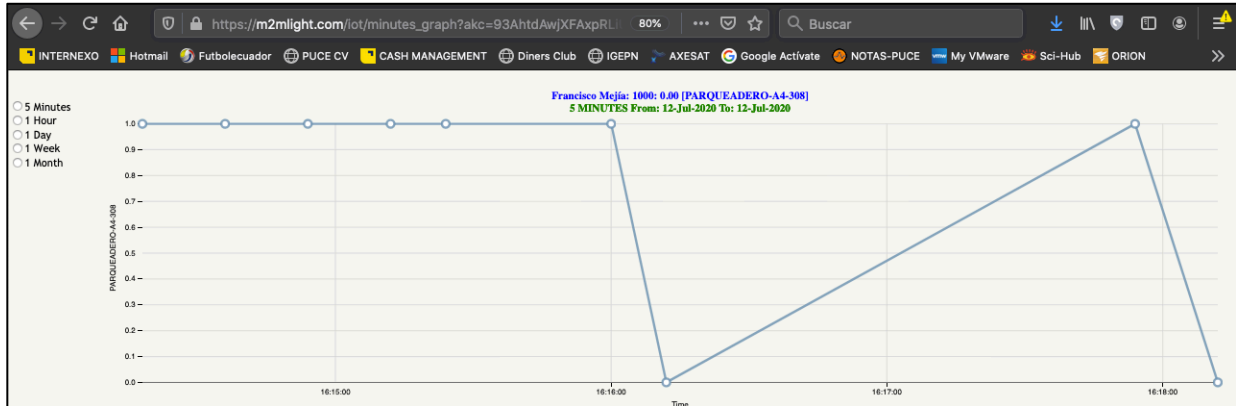


Figura 4 – 20 Información gráfica del estado del parqueadero No. 308 del Almendro 4 (Mejía, 2020).

m2mlight data
User: Francisco Mejía
Sensor Name: T4-0308

Show the most recent **SENSOR values**

Show all **SENSOR values between** **and**

Show the most recent **ALERT values**

Show all **ALERT values between** **and**

Date	Value
2020-07-26 11:15:05	0
2020-07-26 11:14:53	1
2020-07-26 11:13:26	0
2020-07-26 11:13:09	1
2020-07-26 11:08:42	0
2020-07-26 11:07:23	1
2020-07-26 11:07:07	1
2020-07-26 11:06:13	0

Figura 4 – 21 Información tabulada del estado del parqueadero No. T4-0308 (Mejía, 2020).

4.5 Diseño de atributos y funcionalidades del servidor IoT

Esta parte del sistema fue desarrollada con la ayuda del Ing. Mario Albuja de la empresa Minkafab. Cia. Ltda., la misma que es creada como un sistema personalizado ligado a la plataforma de IoT M2MLight. En la figura 4 – 22 se puede apreciar la ventana principal de la interfaz web desarrollada para el control del SPI.

Control de parqueaderos

2020/08/05 23:01:03

Id sensor	Nombre	Fecha y hora	Valor Sensor	Estado	Dpto. Visitado	Comentarios
214	T4-0308	2020-08-05 22:55:32	0.00			Parqueo Disponible <input type="button" value="Asignar Dpto."/>
217	T4-0310	2020-08-05 22:55:42	0.00			Parqueo Disponible <input type="button" value="Asignar Dpto."/>
218	T8-0605	2020-08-05 22:55:55	0.00			Parqueo Disponible <input type="button" value="Asignar Dpto."/>
219	T8-0607	2020-08-05 22:55:52	0.00			Parqueo Disponible <input type="button" value="Asignar Dpto."/>

Figura 4 – 22 Ventana principal del Sistema de Control de Parqueaderos (Mejía, 2020).

Uno de los principales atributos del SPI consiste en dotar de alertas en tiempo real lo relacionado con el estado de los parqueaderos de visita, como se puede apreciar en la figura 4-23. Se dispone de alertas visuales representadas por vehículos de diferentes colores, es así como, de acuerdo con el estado actual del parqueadero, estos van a tener los siguientes significados:

- Parqueadero disponible = Vehículo de color verde
- Vehículo parqueado menos de 4 horas = Vehículo de color azul.
- Vehículo parqueado más de 4 horas = Vehículo de color amarillo.
- Vehículo parqueado más de 6 horas = Vehículo de color anaranjado.
- Vehículo parqueado más de 8 horas = Vehículo de color negro.
- Vehículo estacionado sin autorización = Vehículo de color rojo.

Control de parqueaderos

2020/08/05 23:01:03

Id sensor	Nombre	Fecha y hora	Valor Sensor	Estado	Dpto. Visitado	Comentarios
214	T4-0308	2020-08-05 22:55:32	0.00			Parqueo Disponible <input type="button" value="Asignar Dpto."/>
217	T4-0310	2020-08-05 22:55:42	0.00			Parqueo Disponible <input type="button" value="Asignar Dpto."/>
218	T8-0605	2020-08-05 22:55:55	0.00			Parqueo Disponible <input type="button" value="Asignar Dpto."/>
219	T8-0607	2020-08-05 22:55:52	0.00			Parqueo Disponible <input type="button" value="Asignar Dpto."/>

Figura 4 – 23 Alertas visuales mostradas en la ventana principal del Sistema de Control de Parqueaderos (Mejía, 2020).

4.5.1 Diseño de la base de datos para la interacción con el servidor IoT M2MLight

Para cumplir con los requerimientos antes descritos del SPI, se va a crear una tabla dentro de la base de datos del servidor IoT, la misma que dispondrá de las siguientes columnas:

- **sensor_id:** Identificador del sensor
- **name:** Nombre del sensor
- **api_key:** Clave del sensor
- **status_sensor:** Estado del sensor en m2milight
- **tiempo:** Hora en segundos en formato Unix
- **valor:** Valor de la lectura del sensor: 0 o 1; 0=parqueadero disponible
- **departamento:** Departamento al que va a dirigirse la visita
- **comments:** Comentarios del estado del parqueadero
- **estado:** Estado del parqueadero: 0, 1, 2, 3, 4 o 5
- **mensaje_sent:** 0 o 1; 1=mensaje a celular enviado

Así pues cada vez que el Gateway envía un mensaje al servidor m2milight, sobre el cambio del estado del sensor de presencia (0 o 1), se ingresa un registro en esta tabla y gracias a un trigger ligado a la tabla sensor valor, propia de m2milight. El Trigger actualiza la columna “estado” con los valores de la columna “valor”, 0 o 1.

Para más información referirse al anexo G en el cual están definidos éste y otros triggers o funciones empleadas dentro del servidor IoT. La definición del trigger (insert_val_parking1), está en el /var/www/vhosts/m2milight.com/httpdocs/scripts/parqueadero.sql

4.5.2 Diseño de mensajes MQTT para la interacción entre el servidor IoT M2MLight y el Gateway Dragino LG02

Se utilizará el siguiente formato para los mensajes MQTT:

```
YYYXXX&c=CCC&p=PPPPPP&n=NNNN
```

Donde:

- YYY: Indica el origen del mensaje, pudiendo este ser m2m (servidor IoT) o gws (gateway).
- XXXX: Número aleatorio indicador de los mensajes. Ej. 1234 (no obligatorio)
- CCC: Código del mensaje. Ej. 211
- PPPPPP: Identificación del parqueadero. Ej. T4-0310
- NNNN: Identificación del nodo. Ej. 1000

Por ejemplo:

```
m2m3476&c=311&p=T4-0310&n=1000
```

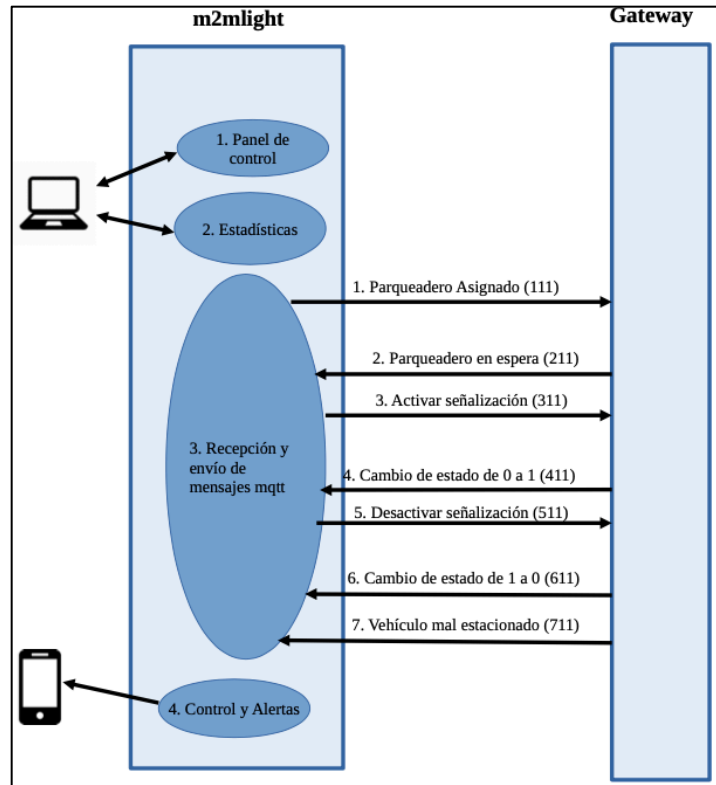


Figura 4 – 24 Alertas visuales mostradas en la ventana principal del Sistema de Control de Parqueaderos (Mejía, 2020).

Dado que el Gateway requiere de un código único para cada nodo, se relacionarán los parqueaderos T4-0308 y T4-0310 al nodo 1000, y los parqueaderos T8-0605 y T8-0607 al nodo 2000.

Para optimizar el funcionamiento del SPI, se emplearán ciertos códigos en los mensajes MQTT, los cuales se pueden apreciar en la figura 4 – 24, y que son definidos con mayor detalle en la tabla 4 – 02, los que servirán para una correcta interacción entre el servidor IoT M2MLight y el Gateway Dragino LG02.

Código	Acciones ejecutadas
111	Servidor M2M envía al Gateway asignación de Parqueadero asignado). El estado del parqueadero inicialmente vale 0.
211	Como respuesta al código 111, el respectivo Nodo de Parqueadero confirma al servidor M2M que se encuentra esperando el arribo del vehículo. Y empieza a parpadear secuencialmente los leds verde y rojo del parqueadero asignado hasta que arribe el vehículo.
311	Como respuesta al código 211, el servidor envía notificación a Nodos de Señalización, para que estos señalicen la ruta configurada hacia el parqueadero asignado.
411	Nodo de parqueadero confirma el arribo del vehículo al parqueadero asignado. Servidor cambia el estado del parqueadero de 0 a 1.

Código	Acciones ejecutadas
511	Como respuesta al código 411, el servidor envía notificación a Nodos de Señalización, para que estos desactiven la señalización de ruta hacia el parqueadero previamente asignado.
611	Nodo de parqueadero notifica al servidor IoT que el vehículo ha dejado el parqueadero asignado. Servidor cambia el estado del parqueadero de 1 a 0.
711	Nodo de parqueadero notifica al servidor IoT que un vehículo esta estacionado sin la respectiva asignación o autorización. Servidor envía un mensaje de alerta al celular del usuario.

Tabla 4 – 02 Códigos establecidos para la interacción entre el servidor IoT y los Nodos LoRa (Mejía, 2020).

4.5.3 Procesos diseñados para la interacción entre el servidor IoT M2MLight con la base de datos y con el Gateway Dragino LG02

De acuerdo con los atributos definidos anteriormente para la solución propuesta, se define la necesidad de implementar 4 procesos o programas que van a correr en el servidor M2MLight.

4.5.3.1 Proceso para el envío y recepción de mensajes MQTT

Este es un proceso que siempre se halla ejecutándose en background. Se puede considerar como un canal privado para cada usuario de m2mlight, en donde se está escuchando los mensajes MQTT que llegan y los mensajes MQTT que envía.

Este proceso actualiza en la base de datos el estado de los sensores (0 o 1) en base a los mensajes MQTT recibidos desde el Gateway como se mencionó anteriormente.

Para más información respecto al código fuente referirse al anexo G en la sección: /var/www/vhosts/m2mlight.com/httpdocs/iot/mqtt_sub_jfDr1njci7.php

4.5.3.2 Proceso para el panel de control

Este programa consiste en la interfaz web para que el usuario pueda gestionar el SPI. Permite visualizar el estado de los parqueaderos en tiempo real; y, asignar a un visitante un parqueadero de visita cuando el mismo está disponible (estado = 0 y alerta = vehículo de color verde).

Como se definió anteriormente, al asignar el parqueadero a un departamento se envía un mensaje MQTT con el código 111 al Gateway, para que este reenvíe a todos los nodos, y que éstos puedan interpretar si el mensaje es para ellos y que identifiquen el parqueadero al cual está haciendo referencia el servidor IoT.

Para mas información del código fuente favor referirse al anexo G en la sección: /var/www/vhosts/m2mlight.com/httpdocs/iot/parqueos_panel.php y [update_dpto.php](#)

Para acceder al SPI, se debe hacer una llamada desde un navegador con salida a internet con la siguiente dirección:

http://m2milight.com/iot/parqueos_panel.php?api_key=jfDr1njci7

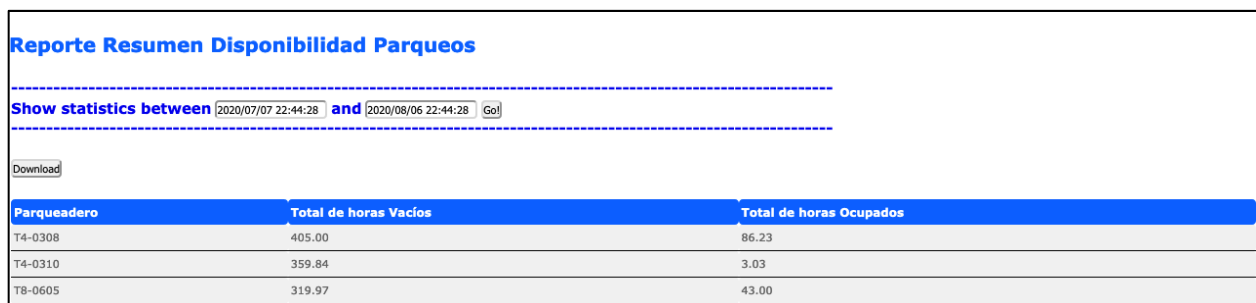
4.5.3.3 Proceso para la obtención de estadísticas de uso de los parqueaderos de visita

Se disponen de 2 tipos de reportes, el primero genera un resumen de la disponibilidad y el segundo genera un reporte con respecto a los parqueaderos de visita y cuales departamentos han hecho uso de éstos. Ambos tipos de reportes pueden ser exportados para ser editados con una hoja electrónica.

4.5.3.3.1 Reporte Resumen de Disponibilidad

Este programa presenta una interfaz al usuario, para generar un reporte de estadísticas de uso de los parqueaderos. Se escoge el período y se muestra el número de horas totales ocupadas y el número total de horas vacías.

Para acceder a éste desde el panel de control se debe hacer clic en la opción “Reporte Resumen Disponibilidad”, en donde se procede a definir el período y finalmente se da clic en “Go”, obteniéndose una tabla similar a la mostrada en la figura 4 – 25. Si se desea, se puede dar clic en “Download” para descargar esta información y editarla en una hoja electrónica.



Parqueadero	Total de horas Vacíos	Total de horas Ocupados
T4-0308	405.00	86.23
T4-0310	359.84	3.03
T8-0605	319.97	43.00

Figura 4 – 25 Interfaz para la generación del resumen de disponibilidad de uso de los parqueaderos de visita (Mejía, 2020).

El código fuente de este proceso está dividido en 3 archivos, para más información referirse al anexo G en las secciones:

```
/var/www/vhosts/m2milight.com/httpdocs/iot/parks_list.php  
/var/www/vhosts/m2milight.com/httpdocs/presentation/smarty_plugins/function.load_parks_list.php  
/var/www/vhosts/m2milight.com/httpdocs/presentation/templates/parks_list.tpl
```

4.5.3.3.1 Reporte de uso de parqueaderos de visita respecto a los departamentos

Este programa presenta una interfaz al usuario, para generar un reporte de la base de datos de la entrada (1) y salida (0) de los parqueaderos. Se escoge el período y se muestran los registros de los parqueaderos, las entradas, salidas y el departamento asignado

Parqueadero	Acción	Fecha y Hora	Departamento
T4-0308	Entrada	2020-08-05 22:25:40	t3-23
T4-0308	Salida	2020-08-05 22:29:22	t3-23
T4-0308	Entrada	2020-08-05 22:33:08	T4-94
T4-0308	Salida	2020-08-05 22:41:30	T4-94
T4-0308	Entrada	2020-08-05 22:44:11	T4-94
T4-0308	Salida	2020-08-05 22:53:01	
T4-0308	Entrada	2020-08-05 22:54:29	T4-94
T4-0308	Salida	2020-08-05 22:55:32	T4-33
T4-0308	Entrada	2020-08-05 23:18:09	T4-33
T4-0308	Salida	2020-08-06 08:25:05	
T4-0308	Entrada	2020-08-06 08:26:00	T4-33
T4-0308	Salida	2020-08-06 08:26:40	t4-98
T4-0308	Entrada	2020-08-06 08:27:48	T3-21
T4-0308	Salida	2020-08-06 08:28:39	T3-21
T4-0308	Entrada	2020-08-06 08:34:55	T3-21
T4-0308	Salida	2020-08-06 18:05:59	T4-94

Figura 4 – 26 Interfaz para la generación del Reporte de parqueaderos de visita utilizados por los departamentos (Mejía, 2020).

Para acceder a esta función, desde el panel de control se debe dar clic en la opción “Reporte Parques y Dptos”, en donde se procede a definir el período y se hace clic en “Go”, obteniéndose una tabla similar a la mostrada en la figura 4 – 26. Desde donde si se desea, se hace clic en “Download” para descargar esta información y editarla en una hoja electrónica.

El código fuente de este proceso se halla dividido en 3 archivos, para más información referirse al anexo G en las secciones:

```
/var/www/vhosts/m2milight.com/httpdocs/iot/dptos_list.php
/var/www/vhosts/m2milight.com/httpdocs/presentation/smarty_plugins/function.load_dptos_list.php
/var/www/vhosts/m2milight.com/httpdocs/presentation/templates/dptos_list.tpl
```

4.5.3.4 Proceso para el control y alertas

Este es un proceso que siempre está ejecutándose en background. Éste se encuentra verificando constantemente el tiempo de permanencia de las visitas en los parqueaderos asignados, para cambiar su estado entre 0 y 5.

Adicionalmente envía un mensaje de alerta a la aplicación Telegram del equipo celular del usuario, una vez que han transcurrido las 4 horas de uso de algún parqueadero o si existe un vehículo haciendo uso de un parqueadero sin la respectiva autorización.

Para mayor información del código fuente, favor referirse al anexo G en la sección: var/www/vhosts/m2mlight.com/httpdocs/iot/control_parqueos.php

4.6 Pruebas de funcionamiento del SPI propuesto

Una vez instalado el SPI propuesto, fue necesario considerar algunos aspectos adicionales para permitir que el sistema funcione correctamente, entre estos aspectos se destacan los siguientes:

4.6.1 Calibración de los sensores de presencia

Una vez instalados cada uno de los respectivos sensores en los parqueaderos de visita considerados, como se muestra en la figura 4 – 27, se requirió medir la distancia de éstos respecto al suelo, para fijar los umbrales de detección a ser utilizados dentro de la programación de las respectivas tarjetas de desarrollo encargadas de controlar los respectivos parqueaderos de visita. Los valores medidos se encuentran descritos en la tabla 4 – 03.



Figura 4 – 27 Medición de distancia desde los sensores instalados hasta el suelo. (Mejía, 2020).

Parqueadero	Distancia desde el sensor hasta el suelo
T4-308	210 cm
T4-310	215 cm
T8-605	195 cm
T8-607	190 cm

Tabla 4 – 03 Valores de distancia medidos desde los sensores de presencia hasta el suelo (Mejía, 2020).

4.6.2 Calibración de parámetros de transmisión LoRa

De acuerdo a los resultados obtenidos en el literal 4.1, se evidenció la necesidad de tener 2 Gateways LoRa para proporcionar una correcta cobertura a ambos subsuelos de parqueaderos, dado que la solución planteada únicamente contempla 4 parqueaderos de visita, se optó por instalar un único Gateway en el subsuelo 1. Es así que para mejorar los valores de RSSI obtenidos inicialmente para los 4 parqueaderos a ser utilizados, se eligió realizar varias pruebas modificando ciertos parámetros propios de la tecnología de transmisión LoRa como son: potencia de transmisión, factor de separación, tasa de datos y tasa de codificación. En la tabla 4 – 04 se puede apreciar una muestra de los principales resultados obtenidos. Para el detalle completo de las pruebas realizadas favor referirse al anexo H. Los parámetros utilizados son los que se encuentran sombreados de color azul.

Potencia de transmisión	Factor de separación	Tasa de datos	Tasa de codificación	RSSI	
				Nodo 1000: T4-308 Y T4-310	Nodo 2000: T8-605 Y T8-607
16	8	41700	4/5	-79	-90
16	8	41700	4/6	-78	-90
16	8	41700	4/7	-78	-90
16	8	41700	4/8	-78	-89
18	8	41700	4/5	-77	-90
18	8	41700	4/6	-77	-91
18	8	41700	4/7	-77	-89
18	8	41700	4/8	-78	-90

Tabla 4 – 04 Muestra de los principales resultados obtenidos para los parámetros de transmisión LoRa (Mejía, 2020).

4.6.3 Instalación de leds adicionales para señalización de los parqueaderos de visita

Para facilitar la identificación de los parqueaderos de visita a distancias superiores a los 30 metros, se debió adicionar para cada uno de los parqueaderos, leds de alta intensidad tanto de color verde como de color rojo. Esto con el afán de mejorar el reconocimiento y distinción de los parqueaderos desde el interior del vehículo, cuando se encuentra a distancias superiores a los 30 metros. En la figura 4 – 27, se puede identificar claramente los parqueaderos de visita T8-605 y T8-607 a una distancia superior a los 30 metros.

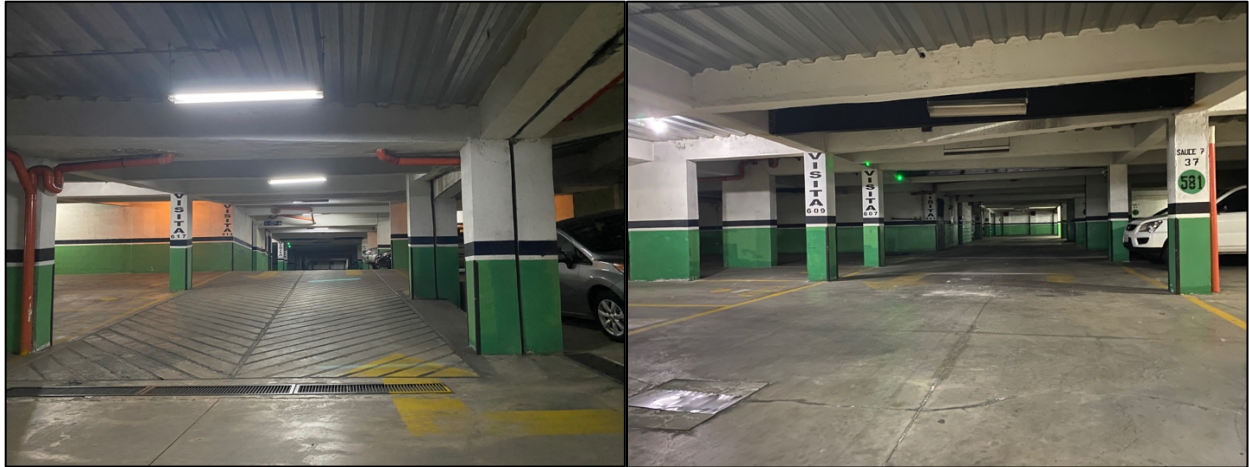


Figura 4 – 27 Luces de señalización de los parqueaderos T8-605 y T8-607, la imagen izquierda fue tomada a 40 metros de distancia mientras que la imagen de la derecha fue tomada a 15 metros de distancia. (Mejía, 2020).

4.6.4 Capacitación del personal de recepción y guardias con respecto a la operación del SPI

A continuación, como se puede apreciar en la figura 4 – 28, se brindó una explicación a la persona de recepción y al Jefe de los guardias para que ellos conozcan en mayor detalle el propósito del sistema y las funcionalidades del mismo. Luego se les capacitó en como verificar el estado de los parqueaderos y de ser el caso como asignar los mismos a las visitas. Finalmente se les instruyó respecto al significado de las alertas visuales para los parqueaderos de visita y como interpretar estos colores con respecto a la buena o mala utilización de los mismos.

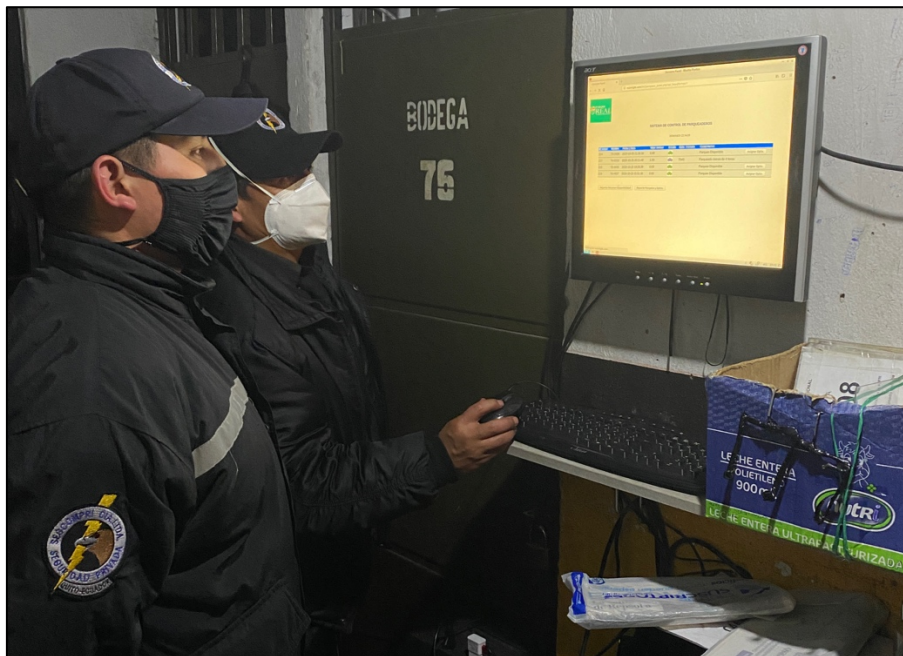


Figura 4 – 28 Capacitación a personal de recepción con respecto a las funcionalidades del SPI propuesto (Mejía, 2020).

4.7 Resultados del SPI implementado

Después de cumplir con la instalación del prototipo y de pulir los aspectos detallados en el literal 4.6, se comprobó el funcionamiento del sistema durante aproximadamente un mes, tiempo en el cual, se pudo evidenciar las bondades de diseñar un sistema IoT aplicando una arquitectura por capas. Además se logró cumplir con los atributos planteados originalmente para el SPI propuesto, es así que se facilitó la asignación de los parqueaderos disponibles por parte de la persona de recepción, así como también de acuerdo a lo mostrado en la figura 4-29, se ayudó a los usuarios a familiarizarse con la utilización y localización de los parqueaderos asignados a los visitantes del conjunto habitacional.

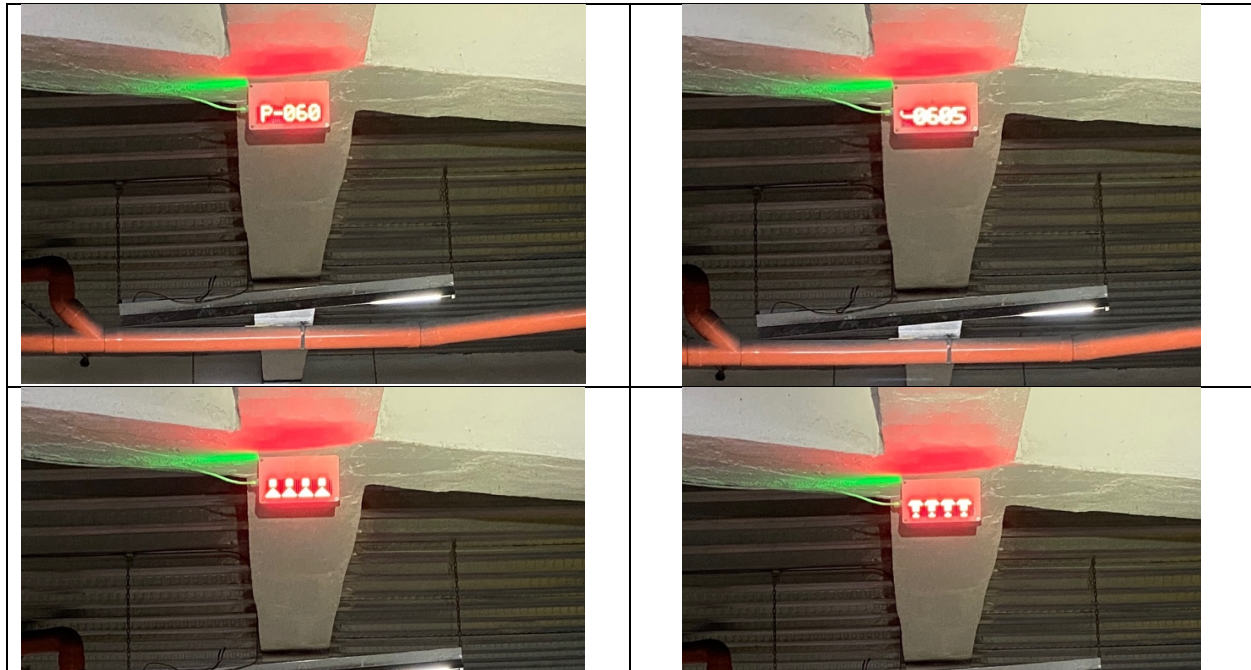


Figura 4 – 29 Señalización de ruta con matrices de leds instaladas para guiar a los visitantes hacia su parqueadero asignado (Mejía, 2020).

De igual manera, como se puede apreciar en la figura 4-30, el sistema permitió a la persona de recepción visualizar en tiempo real el estado de ocupación de los 4 parqueaderos de visita que disponen del SPI y conocer el estado de estos mediante alertas visuales en la pantalla a la entrada de los parqueaderos. Adicionalmente se permitió optimizar las labores inherentes de los guardias del conjunto, evitando que ellos tengan que encargarse de guiar a los visitantes hasta los parqueaderos asignados.

Id sensor	Nombre	Fecha y hora	Valor Sensor	Estado	Dpto. Visitado	Comentarios
214	T4-0308	2020-10-29 20:07:02	1.00		T422	Vehículo estacionado sin autorización
217	T4-0310	2020-10-29 17:15:35	1.00		T543	Parqueado más de 6 horas
218	T8-0605	2020-10-29 23:03:52	0.00			Parqueo Disponible Asignar Dpto.
219	T8-0607	2020-10-29 23:05:48	0.00			Parqueo Disponible Asignar Dpto.

Figura 4 – 30 Visualización en tiempo real del estado de ocupación de los parqueaderos de visita asignados al SPI propuesto (Mejía, 2020).

Finalmente, de acuerdo con la figura 4-31, se puede apreciar de qué manera el sistema permitió tener alertas mediante mensajes hacia la aplicación Telegram del celular del usuario principal, informándole respecto al mal uso de los parqueaderos de visita que disponen del sistema instalado.



Figura 4 – 31 Alerta visual en aplicación Telegram del celular del usuario principal (Mejía, 2020).

5. Capítulo 5: Conclusiones y Recomendaciones

5.1 Conclusiones

- La utilización de un modelo de arquitectura IoT⁵² facilita enormemente el diseño de soluciones IoT, pues permite identificar los elementos, atributos y requerimientos necesarios para cumplir con los propósitos y/o objetivos planteados que se desean monitorear o controlar con la solución IoT planteada.
- Al utilizar las tecnologías de transmisión en banda libre como aquella utilizada para este estudio LoRaWAN⁵³, es fundamental el realizar análisis espectrales para determinar o corregir ciertos aspectos que pueden ser propios de esta tecnología o que se relacionan con la banda de frecuencia de la que hace uso una determinada tecnología. Por ejemplo, si se utiliza una tecnología en la banda de 2.4 GHz como es el WiFi⁵⁴ de los hogares, esto podría acarrear múltiples inconvenientes, principalmente derivados por la interferencia existente por el amplio uso de este tipo de dispositivos en hogares y en sitios urbanos.
- La selección del tipo de hardware para el desarrollo de un prototipo es fundamental ya que de acuerdo con las características o atributos propios de éstos se puede solventar de una mejor manera los diferentes requerimientos que demanda el proyecto. Es así como en este estudio, sus necesidades con respecto a la cantidad de puertos de entrada y salida pudieron ser cubiertos de una mejor manera con una determinada tarjeta de desarrollo, así como también estos permitirán acoplarnos a nuestros presupuestos y también de acuerdo a la experiencia del programador permitirán que el desarrollo sea más ágil o que este se convierta en uno de los principales inconvenientes del diseño de la solución.
- De igual manera, la selección del proveedor del servidor IoT en la nube es una decisión crucial al momento de desarrollar este tipo de soluciones. Si la selección es la correcta esto puede ayudar en sobremanera el diseño del sistema, caso contrario puede convertirse en un dolor de cabeza y ser la razón por la cual el proyecto no salga de acuerdo con lo planificado. Siempre se debe tomar en cuenta los atributos necesarios o que se requieren que sean implementados por parte del servidor en la nube, ya que de acuerdo con estos requisitos se deberá escoger entre soluciones gratuitas o pagadas.
- Posiblemente para una completa implementación del SPI⁵⁵ dentro del CHPR⁵⁶, se deberá considerar la utilización de una solución tecnológica LoRaWAN completa, es decir una que permita la utilización de más de un solo canal de transmisión simultaneo, afín de evitar

⁵² IoT Internet of Things - Internet de las Cosas.

⁵³ LoRaWAN - Long Range Wide Area Network – Redes de largo Alcance de Area Extendida.

⁵⁴ WIFI - Tecnología inalámbrica que permite la conexión de dispositivos electrónicos entre si o hacia el internet.

⁵⁵ SPI - Sistema de Parqueadero Inteligente.

⁵⁶ CHPR - Conjunto Habitacional Parque Real.

posibles interferencias entre los propios nodos LoRa⁵⁷, necesarios para una correcta implementación del proyecto.

- Cuando se utiliza este tipo de arquitecturas IoT para parqueaderos inteligentes, es necesario la realización de pruebas IN-SITU⁵⁸, con las cuales se puede validar y corregir ciertos aspectos no considerados como pueden ser las distintas alturas de un parqueadero respecto a otro, o inclusive en el caso de que se tengan problemas de cobertura o interferencia, el tener este espacio o proceso en el proyecto, permitirá realizar una mayor cantidad de pruebas y permitirá pulir detalles que pueden haber sido omitidos en la fase de pruebas en laboratorio, pero que al ser la solución implementada pueden constituir en un serio problema como podría llegar a ser por ejemplo la interferencia al implementar la solución en un determinado ambiente urbano o de ciudad.
- Un aspecto fundamental para considerar es el mantener una constante comunicación con el personal de recepción, ya que ellos son quienes estarán a cargo de manejar directamente el SPI, y son ellos quienes están en contacto directo con las visitas, por lo cual ellos constituirán la principal referencia con respecto a información para poder solucionar posibles errores que puedan llegar a presentarse ya sea esto con respecto a aspectos de software o hardware durante la operación del SPI instalado.
- Sin duda la elaboración de este tipo de trabajos de titulación permite a los maestrantes crecer profesionalmente, permitiéndoles adquirir nuevos conocimientos respecto a las últimas tecnologías que se encuentran conquistando el mundo y que constituyen indudablemente un paradigma que revoluciona el estilo de vida de la humanidad.

5.2 Recomendaciones

- Se recomienda para futuros estudios relacionados a sistemas IoT, la utilización de un diseño mediante una arquitectura de capas IoT, ya sea el recomendado por la ITU⁵⁹ o algún otro similar, como el utilizado para el presente estudio propuesto por (Darwish, 2015), ya que al utilizar estos correctamente, nos permiten determinar los requerimientos tanto de hardware como de software, necesarios para poder llegar a presentar una propuesta a los problemas que pretendamos solventar mediante la utilización de la tecnología IoT.
- Dada la constante evolución de las tecnologías, se sugiere siempre intentar mantener los conocimientos al día a día, dado que de acuerdo con la naturaleza de los problemas que se pretendan solventar, siempre una tecnología puede ser mejor que otra, y pues que mejor que siempre intentar aprovechar al máximo las mismas y poder así obtener los mejores réditos tanto funcionales como económicos.

⁵⁷ LoRa - especificación de capa física que emplea modulación similar a CSS (Chirp Spread Spectrum).

⁵⁸ IN-SITU - Se refiere a un fenómeno que debe ser observado en el lugar.

⁵⁹ ITU International Telecommunications Union - Unión Internacional de Telecomunicaciones.

- Se recomienda al momento de conceptualizar estudios similares, el considerar un proceso de validación y/o calibración al momento de la instalación de soluciones análogas que utilicen una arquitectura de capas IoT, ya que el tener este proceso en mente puede ayudar significativamente a mejorar las funcionalidades proporcionadas por los sistemas propuestos y podría ayudar a reducir el tiempo de desarrollo e implementación, como fue el caso en este estudio, que se necesitó de cierto tiempo para poder cumplir con la calibración de los sensores ultrasónicos y la realización de pruebas para determinar las mejores características de transmisión LoRa al interior de ambos subsuelos de parqueadero, que sin lugar a duda tomaron más tiempo del planificado inicialmente.
- Se sugiere como un próximo tema de estudio para este tipo de arquitecturas, el desarrollo de aplicaciones o de sistemas de información para los usuarios potenciales de este tipo de arquitecturas, es decir como por ejemplo, la facultad de poder utilizar la información recabada por el servidor IoT en la nube y permitir la interacción de esta mediante aplicaciones que podrían permitir a los usuarios finales hacer la reservación de una plaza de parqueadero en un centro comercial en la ubicación más próxima a la tienda que vaya el cliente a visitar.
- Se recomienda para futuros estudios dentro de la Universidad, el continuar analizando los múltiples problemas sociales de la sociedad ecuatoriana y como poder solventar éstos desde el paradigma IoT, ya que como se mencionó anteriormente, las aplicaciones y usos de esta tecnología está únicamente limitada por nuestra imaginación.
- El SPI puede ser implementado en parqueaderos similares sin mayores inconvenientes, lo principal a considerar serían aspectos de cobertura, cantidad de parqueaderos, aspectos de protección contra agua del equipamiento en caso de ser instalado al aire libre y aspectos relacionados con los puntos para tomar energía eléctrica.
- Si bien el SPI propuesto, únicamente abarca aspectos relacionados al monitoreo del estado de los parqueaderos de visita, al momento de instalar los nodos de monitoreo en los parqueaderos los cuales hacen uso de tarjetas de desarrollo, se podría continuar ampliando y mejorando los atributos proporcionados por el SPI, por ejemplo, se podría incorporar sensores acústicos, los cuales permitirían conocer si las alarmas de los vehículos están siendo o han sido vulneradas, lo que permitiría a los guardias identificar y actuar inmediatamente ante un posible intento de robo de los automóviles. En sí el límite de atributos está limitado únicamente por nuestra imaginación.
- Para la completa implementación del sistema, se sugiere considerar todos los requerimientos de energía, tomando en cuenta los aspectos energéticos requeridos por los sensores, leds y demás elementos electrónicos a fin de poder instalar baterías recargables que permitan dotar de un adecuado tiempo de respaldo ante una eventual falla del fluido eléctrico convencional.

- Al momento como medida de seguridad con respecto a la información que proporciona el SPI, únicamente se contempla la periódica modificación de las API-KEYS⁶⁰ de los canales de datos correspondiente a cada uno de los sensores de los parqueaderos, así como también la periódica modificación del API-KEY de usuario. Por lo cual se sugiere también para la mejora del sistema a futuro, la respectiva implementación de un portal de acceso a la página web, a la cual se pueda únicamente acceder con las respectivas credenciales y las cuales también deben ser modificadas periódicamente.

⁶⁰ API-KEY – Clave de interfaz que se utiliza para autenticar a un usuario o aplicación.

6. Bibliografía

- Albuja, Mario. 2020. "M2MLight Internet of Things Services."
- Amazon. n.d. "MakerFocus ESP32 Junta de Desarrollo WIFI Bluetooth LoRa Dual Core 240MHz CP2102 Con Pantalla OLED de 0,96 Pulgadas y Antena 433/470 MHz Para Arduino." Retrieved (<https://www.amazon.com/-/es/MakerFocus-Desarrollo-Bluetooth-pantalla-pulgadas/dp/B076MSLFC9>).
- Arduino.cc. 2015. "Arduino Software - IDE." Retrieved (<http://arduino.cc/en/Guide/Environment>).
- Arduino.cc. 2019. "Arduino - Introduction." *Arduinio.Cc*. Retrieved (<https://www.arduino.cc/en/guide/introduction>).
- Arduino. 2020. "Arduino Mega 2560 Rev3 | Arduino Official Store." *Arduino 1*. Retrieved (<https://store.arduino.cc/arduino-mega-2560-rev3>).
- Becerril, Antonio. 2020. "¿Qué Es Una API?" *El Economista*. Retrieved (<https://www.eleconomista.com.mx/tecnologia/que-es-una-api-20200130-0078.html>).
- Chakray. 2016. "5 Requerimientos de Una Arquitectura IOT." Retrieved (<https://www.chakray.com/es/5-requisitos-de-una-arquitectura-iot/>).
- Chasoy, Iván Felipe. 2015. "REGISTRO DE VEHICULOS PARA UN PARQUEADERO | Ivanfelipechasoy." Retrieved (<https://ivanfelipechasoy.wordpress.com/2015/08/13/registro-de-vehiculos-para-un-parqueadero/>).
- Darwish, Dina Gamal. 2015. "Improved Layered Architecture for Internet of Things." *International Journal of Computing Academic Research* 4(4):214–23.
- Dragino-Technology-Co-Limited. n.d. "LoRa Shield - Long Range Wireless Transceiver for Arduino." 1.
- Dragino. 2019. "LoRa IoT Development Kit." Retrieved (<http://www.dragino.com/products/lora/item/120-lora-iot-kit.html>).
- EFE, AGENCIA. 2019. "Parque Automotor de Ecuador Creció En 1,4 Millones de Vehículos En Una Década." *El Comercio*, November 1.
- Egli, P. R. 2015. "LPWAN Technologies for Internet of Things (IoT) and M2M Scenarios."
- ELEC-FREAKS. 2013. *Ultrasonic Ranging Module HC - SR04*.
- ELECTRONICS-HUB. 2017. "What Are the Differences between Raspberry Pi and Arduino?" *ElectronicsHub.Org*. Retrieved (<https://www.electronicshub.org/raspberry-pi-vs-arduino/>).
- Eriksen, Rúni. 2019. "Energy Consumption of Low Power Wide Area Network Node Devices in the Industrial ,." KTH Royal Institute of Technology.

- Google-Maps. n.d. "Google Maps-Ubicacion-CHPR." Retrieved (<https://www.google.com/maps/@-0.170688,-78.4767896,784m/data=!3m1!1e3>).
- Guru99. 2020. "Internet of Things (IoT) Tutorial for Beginners: Introduction, Basics, Applications." Retrieved (<https://www.guru99.com/iot-tutorial.html>).
- Intelligence, Insider. 2019. "IoT & Cloud Computing Differences, Providers & Services in 2020 - Business Insider." *Business Insider*.
- Islam, Bashima, Tamzeed Islam, and Shahriar Nirjon. 2017. "Feasibility of LoRa for Indoor Localization." 1–11.
- Jeon, Yongsung, Hong-il Ju, and Seungyong Yoon. 2018. "Design of an LPWAN Communication Module Based on Secure Element for Smart Parking Application." 0–1.
- KaaloT-Technologies. 2020. "Enterprise IoT Platform with Free Plan | Kaa." Retrieved (<https://www.kaaproject.org/>).
- LinkLabs. 2016. "A COMPREHENSIVE LOOK AT Low Power, Wide Area Networks For 'Internet of Things' Engineers and Decision Makers." 38–41.
- LoRa-Alliance. 2015. *A Technical Overview of LoRa ® and LoRaWAN™ What Is It?*
- MathWorks. 2019. "IoT Analytics - ThingSpeak Internet of Things." Retrieved (<https://thingspeak.com/>).
- Naylamp-Mechatronics. 2017. "Sensor Ultrasonido US-100 - Naylamp Mechatronics - Perú." Retrieved (<https://naylampmechatronics.com/sensores-proximidad/283-sensor-ultrasonico-us-100.html>).
- Pardal, Pablo. 2017. "Redes de Área Extensa Para Aplicaciones de IoT: Modelado de Comunicaciones Sigfox." Universidad Politécnica de Valencia.
- RAE, Real Academia Española. 2014. "Diccionario de La Lengua Española." Retrieved (<https://dle.rae.es/parqueadero>).
- Ray, Brian. 2018. "SigFox Vs. LoRa: A Comparison Between Technologies & Business Models." *Link-Labs.Com*.
- Sehrawat, Deepti, and Nasib Singh Gill. 2019. "Smart Sensors: Analysis of Different Types of IoT Sensors." *Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019* (Icoei):523–28. doi: 10.1109/ICOEI.2019.8862778.
- Singh, Ritesh Kumar, Priyesh Pappinisseri Puluckul, Rafael Berkvens, and Maarten Weyn. 2020. "Energy Consumption Analysis of LPWAN Technologies and Lifetime Estimation for IoT Application." *Sensors (Switzerland)* 20(17):1–22. doi: 10.3390/s20174794.
- Thingier.io. 2018. "Thingier.io Platform." Retrieved (<https://thingier.io/>).
- UIT-T. 2012. "UIT-T Rec. Y.2060 Descripción General de Internet de Los Objetos." *Sector*

de Normalización de Las Telecomunicaciones de La UIT (2012-06–15):20.

Vermeulen, Erik. 2017. "What Is 'Smart' in Our New Digital World." Retrieved (<https://hackernoon.com/what-is-smart-in-our-new-digital-world-87e6426398>).

Yamanoor, Narasimha Saii, and Srihari Yamanoor. 2017. "High Quality, Low Cost Education with the Raspberry Pi." *GHTC 2017 - IEEE Global Humanitarian Technology Conference, Proceedings* 2017-Janua:1–5. doi: 10.1109/GHTC.2017.8239274.

7. Glosario

FCC – Federal Communications Commission, es la agencia federal responsable de implementar y hacer cumplir las leyes y regulaciones de comunicaciones de los Estados Unidos.