

**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**  
**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE**  
**COMUNICACIÓN**



**TEMA:**

“RED MESH LPWAN MEDIANTE EL USO DEL ESTÁNDAR IEEE 802.15.4G  
PARA LA RECOLECCIÓN DE DATOS SOBRE VARIABLES EN LOS  
CONTENEDORES DE RESIDUOS SÓLIDOS DEL SECTOR YACUCALLE DE LA  
CIUDAD DE IBARRA PROVINCIA DE IMBABURA”

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

**AUTORA:**

Andrea Elizabeth Cuasquer Irua

**DIRECTOR:**

Ing. Edgar Alberto Maya Olalla, MSc

Ibarra, 2024



# UNIVERSIDAD TÉCNICA DEL NORTE

## BIBLIOTECA UNIVERSITARIA

### AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

#### 1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
<b>CÉDULA DE IDENTIDAD:</b>	<b>DE</b>	0401938832	
<b>APELLIDOS Y NOMBRES:</b>	<b>Y</b>	Cuasquer Irua Andrea Elizabeth	
<b>DIRECCIÓN:</b>		El Olivo, Ibarra	
<b>EMAIL:</b>		<a href="mailto:aecuasqueri@utn.edu.ec">aecuasqueri@utn.edu.ec</a>	
<b>TELÉFONO FIJO:</b>	2204006	<b>TELÉFONO MÓVIL:</b>	0992020038

DATOS DE LA OBRA	
<b>TÍTULO:</b>	“Red mesh LPWAN mediante el uso del estándar IEEE 802.15.4g para la recolección de datos sobre variables en los contenedores de residuos sólidos del sector Yacucalle de la ciudad de Ibarra provincia de Imbabura”
<b>AUTORA:</b>	Cuasquer Irua Andrea Elizabeth
<b>FECHA DE APROBACIÓN:</b> DD/MM/AAAA	2024/02/16
<b>PROGRAMA:</b>	<input checked="" type="checkbox"/> <b>PREGRADO</b> <input type="checkbox"/> <b>POSGRADO</b>
<b>TÍTULO POR EL QUE OPTA:</b>	Ingeniería en Electrónica y Redes de Comunicación
<b>ASESOR /DIRECTOR:</b>	Ing. Fabián Geovanny Cuzme Rodríguez, MSc Ing. Edgar Alberto Maya Olalla, MSc

## CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 16 días, del mes de febrero de 2024.

### EL AUTOR:

Firma.....

Nombre: Cuasquer Irua Andrea Elizabeth

## CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 15 de febrero de 2024.

Ing. Edgar Alberto Maya Olalla, MSc  
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.



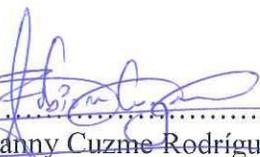
(f) .....  
ING. EDGAR ALBERTO MAYA OLALLA, MSc

C.C.: 1002702197

## APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificado del trabajo de Integración Curricular “Red Mesh LPWAN mediante el uso del estándar IEEE 802.15.4g para la recolección de datos sobre variables en los contenedores de residuos sólidos del sector Yacucalle de la ciudad de Ibarra provincia de Imbabura” elaborado por Cuasquer Irua Andrea Elizabeth, previo a la obtención del título de Ingeniería en Electrónica y Redes de Comunicación, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:

(f):   
Ing. Edgar Alberto Maya Olalla, MSc  
C.C.: 1002702197

(f):   
Ing. Fabián Geovanny Cuzme Rodríguez, MSc  
C.C.: 1311527012

## **DEDICATORIA**

Dedico el presente trabajo de titulación primeramente a mi Familia que ha estado apoyándome en todo momento a lo largo de mi vida estudiantil, agradecida con el esfuerzo por parte de mis padres y hermanos, quienes me han sabido demostrar que el esfuerzo y dedicación conlleva al éxito. Para ti hermano Aquiles que fuiste como un papá, gracias por tu amor incondicional, donde quiera que estés este logro es para ti mi Ángel.

*Andrea Elizabeth Cuásquer Irua*

## AGRADECIMIENTO

Agradezco primeramente a Dios, por brindarme la salud y la vida, por permitirme compartir a lo largo de esta vida con mi familia, a mis papas Blanca y Servio, que se han esforzado por brindarme el estudio y apoyo en toda ocasión, a mis hermanos, Silvio, Aquiles, Filimon, Marcelo, Danilo y Maricela, quienes han aportado con un granito en lo que fue necesario, brindándome su apoyo incondicional e inspirándome a ser una mejor persona cada día en el ámbito educativo, personal y moral.

Quiero extender un agradecimiento a mi tutor Ing. Edgar Maya y a mi asesor Ing. Fabian Cuzme, quienes me proporcionaron el conocimiento adquirido en su momento, y de igual manera por permitirme concretar este trabajo con sus consejos y sugerencias, lo cuales me estuvieron encaminando a la mejora de mi tesis y a la culminación de la misma.

A todos quienes más que compañeros se llegaron a convertir en amigos, con los cuales nos apoyamos en todo momento, gracias infinitas por permitirme contar con su amistad en el transcurso de nuestra vida estudiantil.

*Andrea Elizabeth Cuasquer Irua*

## RESUMEN

El presente trabajo de tesis titulado “Red mesh LPWAN mediante el uso del estándar IEEE 802.15.4g para la recolección de datos sobre variables en los contenedores de residuos sólidos del sector Yacucalle de la ciudad de Ibarra provincia de Imbabura”, consta de una primera parte denominada nodo recolector conformada por dos microcontroladores conectados de manera serial y un arreglo de sensores, los cuales van a tomar datos de las variables: volumen peso, CO2 y ubicación, esta se coloca dentro de un contenedor, en el cual dependiendo del estado de llenado va a enviar los datos de los sensores hacia el nodo coordinador el cual va a estar en comunicación mediante el protocolo MQTT con una página Web donde se mostrarán en primera instancia la ubicación del contenedor que se encuentra lleno y los datos de las demás variables. Se hace las pruebas colocando materiales dentro del contenedor para que se pueda evidenciar la actualización de datos en la página Web, con ello se busca optimizar rutas, estableciendo la posición del contenedor que ya se encuentra lleno, de esa manera el vehículo recolector pasa primero a desocupar los contenedores que están llenos. Esto permite que disminuya la contaminación generada por el vehículo, así como también disminuye las molestias a la población debido la descomposición de los residuos y por el desbordamiento de los basureros ocasionando muchas veces la presencia de fauna urbana.

**Palabras Clave:** 802.15.4g, malla, lpwan, rutas, recolección, cc1352R1.

## ABSTRACT

The present thesis work with de name " Mesh network LPWAN through the use of the IEEE 802.15.4g standard for the collection of data of variables in the solid waste containers of the Yacucalle sector of Ibarra city of Imbabura province", it consists of a first part called collector node made up of two microcontrollers with a serially connection and a sensors array , which will take data from the variables: volume, weight, CO2 and location, it is located inside a container, in which, depending on the state of full, will send the data from the sensors to the coordinator node, it will be in communication through the MQTT protocol with a Web page where shows the location of the container that is full and the data for the variables. This allows the pollution generated by the vehicle to decrease, as well as this allows to decrease the inconvenience to the population for the degradation of waste and the overflow of garbage cans, often causing the presence of urban fauna.

**Keywords:** 802.15.4g, mesh, lpwan, route, collection, cc1352R1.

## **LISTA DE SIGLAS**

**VRR.** Vehículo Recolector de Residuos.

**INEC.** Instituto Nacional de Estadística y Censos.

**IEEE.** Instituto de Ingenieros Eléctricos y Electrónicos.

**PNVB.** Plan Nacional para el Buen Vivir.

**LPWAN.** Low Power Wide Area Network.

**PAN.** Personal Area Network.

**MAC.** Media Access Control.

**PHY.** Physical layer.

**WPAN.** Wireless Personal Area Network.

**FFD.** Dispositivo de Función Completa.

**RFD.** Dispositivo Final de la Rama.

**LR-WPAN.** Low-Rate Wireless Personal Area Network.

**LQI.** Indicador de Calidad de Enlace.

**CCA.** Clear Channel Assessment.

**MPDU.** Unidades de datos de protocolo.

**MLME-SAP.** Punto de acceso al servicio (SAP) de la entidad de gestión de la subcapa MAC.

**GTS.** Gestión del intervalo de tiempo garantizado.

**PDU.** Protocol Data Unit.

**SHR.** Synchronization Header.

**SDF.** Start-of-Frame Delimiter.

**GFSK.** Gaussian Frequency Shift Keying.

**O-QPSK PHY.** Offset Quadrature Phase Shift Keying.

**FSK.** Frequency Shift Keying.

**MPM.** Multi-Phased Modeling.

**CSM.** Common mode signaling.

**FCS .** Frame Check Sequence.

**RSNI.** Radio Signal-to-Noise Indicator.

**RCPI.** Received Channel Power Indicator.

**CSMA-CA.** Carrier Sense Multiple Access with Collision Avoidance.

**LLC.** Logical Link Control.

**RPL.** Routing Protocol for Low-Power and Lossy Networks.

**DODAG.** Destination Oriented Directed Acyclic Graph.

**LLNs.** Low-Power and Lossy Networks.

**DIO.** DODAG Information Object.

**6LoWPAN.** IPv6 over Low-Power Wireless Personal Area Networks.

**IPHC-NHC.** IP Header Compression - Next Header Compression.

**CID.** Context Identifier Extension.

**SAC.** Source Address Compression.

**SAM.** Source Address Mode.

**DAC.** Destination Address Compression.

**DAM.** Destination Address Mode.

**UDP.** User Datagram Protocol.

**TCP.** Transmission Control Protocol.

**CoAP.** Constrained Application Protocol.

**MQTT.** Message Queuing Telemetry Transport.

**AMPQ.** Advanced Message Queuing Protocol.

**QoS.** Quality of Service.

**PDR.** Packet Delivery Rate.

## INDICE GENERAL

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE.....	i.
CONSTANCIAS.....	ii
CERTIFICACIÓN.....	iii
APROBACIÓN DEL COMITÉ CALIFICADOR.....	iv
DEDICATORIA.....	v
AGRADECIMIENTO.....	vi
RESUMEN.....	vii
ABSTRACT.....	viii
INTRODUCCIÓN.....	26
1.1.    Planteamiento De Problema.....	26
1.2.    Justificación.....	29
1.3.    Objetivos.....	31
1.3.1.    Objetivo General.....	31
1.3.2.    Objetivos Específicos.....	31
2.    Capitulo II.....	32
Fundamentación Teórica.....	32
2.1.1.    Topologías de Red.....	33
2.1.1.1.    Estructura de la topología peer-to-peer.....	34
2.1.2.    Arquitectura IEEE 802.15.4.....	36
2.1.2.1.    PHY.....	36
2.1.2.2.    Subcapa MAC.....	37
2.1.3.    Modelo de transferencia de datos.....	37
2.1.3.1.    Transferencia de datos a un coordinador.....	37
2.1.3.2.    Transferencia de datos de un coordinador.....	38
2.1.3.3.    Transferencia de datos peer-to-peer.....	38
2.1.4.    Estructura de la Trama.....	38

2.1.5.	Capa PHY .....	39
2.1.5.1.	Formato General de la Trama PHY. ....	39
2.1.5.2.	Especificaciones de Radio. ....	41
a.	Nivel de entrada máximo del receptor de la señal deseada .....	41
b.	Receptor ED .....	41
c.	Indicador de calidad de enlace (LQI) .....	41
d.	Clear channel assessment (CCA).....	41
2.1.5.3.	Modulaciones 802.15.4g.....	42
a.	Modulación GFSK.....	42
b.	Modulación O-QPSK PHY .....	45
c.	Modulación SUN FSK PHY .....	49
d.	Modulación SUN OFDM PHY .....	60
2.1.6.	Descripción funcional MAC .....	64
2.1.6.1.	Formato General de la Trama MAC. ....	64
a.	Campo Control de Tramas.....	65
b.	Campo Número de secuencia. ....	69
c.	Campo ID de PAN de destino. ....	69
d.	Campo Dirección de destino.....	69
e.	Campo ID PAN de origen. ....	70
f.	Campo Dirección de origen. ....	70
g.	Campo Encabezado de seguridad auxiliar.....	70
h.	El campo IE .....	70
i.	Campo Carga útil de tramas .....	71
j.	Campo FCS.....	71
2.1.6.2.	Medición de Recursos del Espectro .....	72
a.	Energy Detection (ED) .....	72
b.	Indicador de potencia del canal recibido (RCPI).....	72
c.	Indicador de ruido de señal recibida (RSNI) .....	73
d.	Indicador de intensidad de señal recibida (RSSI).....	73
e.	Evaluación de Canal Libre (CCA).....	73
f.	Utilización del canal .....	75

2.1.6.3.	Métodos de acceso al medio. ....	75
a.	Algoritmo CSMA-CA.....	75
2.1.7.	Subcapa de Control de Enlace Lógico (LLC) .....	79
2.1.8.	Protocolo de enrutamiento para baja potencia RPL .....	80
2.1.8.1.	Formación de un GRAFO RPL.....	80
	Para la formación del grafo RPL se sigue los siguientes pasos: .....	80
2.1.9.	Protocolo 6LOWPAN .....	82
2.1.9.1.	Capa de Adaptación de 6LoWPAN .....	83
a.	Tareas de la Capa de Adaptación.....	83
2.1.9.2.	Capa de red. ....	89
2.1.9.3.	Capa transporte. ....	90
2.1.9.4.	Capa Aplicación.....	91
2.1.10.	Sistema Operativo Contiki .....	92
2.1.10.1.	Capas en Contiki. ....	92
2.1.11.	Protocolo MQTT (Message Queue Telemetry Transport).....	94
2.1.12.	Métricas en el enlace inalámbrico.....	97
3.	Capítulo III .....	100
3.1.	Metodología de Investigación .....	100
3.1.1.	Requerimientos.....	101
3.1.1.1.	Situación Actual .....	101
3.1.1.2.	Técnicas de Investigación .....	102
3.1.1.3.	Análisis de Resultados. ....	104
3.1.1.4.	Propósito y Ámbito del Sistema.....	104
3.1.1.5.	Descripción General del Sistema. ....	105
3.1.1.6.	Determinación de Requerimientos .....	106
3.1.1.7.	Nomenclatura de requerimientos.....	107
3.1.1.8.	Requerimientos Operacionales y de Usuario. ....	108
3.1.1.9.	Requerimientos Funcionales del Sistema.....	109
3.1.1.10.	Requerimientos iniciales de arquitectura. ....	110

3.1.2.	Diseño del Prototipo .....	111
3.1.2.1.	Construcción de modelos de Hardware y Software. ....	112
3.1.2.2.	Elección del Hardware .....	115
3.1.2.3.	Estudio de Benchmark de sensores. ....	117
3.1.2.4.	Elección del Software.....	131
3.1.2.5.	Plataforma para visualización de datos. ....	132
3.1.2.6.	Diseño del Sistema. ....	132
3.1.2.7.	Diagrama de bloques del sistema. ....	132
3.1.2.8.	Diagrama de Flujo del Sistema .....	134
3.1.2.9.	Arquitectura del Sistema .....	135
3.1.2.10.	Diseño Fase 1 .....	136
a.	Diagrama de Pines del microcontrolador Arduino Mega 2560 R3 .....	137
b.	Diagrama de Pines del módulo MCU CC1352R1 .....	138
c.	Alimentación de Energía del Nodo Recolector .....	141
d.	Diagrama de Conexión del Nodo Recolector.....	143
e.	Configuración de los Nodos Recolectores .....	143
f.	Configuración del Nodo coordinador.....	146
g.	Configuración de la Plataforma IOT .....	147
h.	Diagrama de flujo de la plataforma IOT .....	149
i.	Medición de variables .....	150
j.	Ubicación del Sistema Embebido.....	153
3.1.2.11.	Diseño Fase 2 .....	153
a.	Diseño de la comunicación Inalámbrica. ....	154
b.	Análisis Inalámbrico .....	156
c.	Perdidas en el Espacio Libre .....	157
d.	Cálculo de parámetros haciendo uso de Software de Simulación Radio Mobile.....	164
e.	Construcción de la red Mesh .....	170
4.	Capitulo IV: Implementación y Pruebas de funcionamiento .....	176
4.1.	Implementación.....	176
4.1.1.	Instalación del Nodo Recolector.....	177
4.1.2.	Ubicación y calibración de sensores en el contenedor .....	177
4.1.3.	Instalación del Nodo Coordinador.....	181

4.1.4. Simulación de la red mesh.....	182
4.2. Pruebas de funcionamiento .....	201
4.2.1. Pruebas según el estado del contenedor.....	202
4.2.2. Cronograma de pruebas de funcionamiento del sistema .....	204
4.2.3. Pruebas de alcance de la red mesh en un ambiente real con línea de vista .....	215
4.2.4. Evaluación del desempeño de la Red Mesh .....	222
4.3. Beneficios del Sistema .....	225
CONCLUSIONES Y RECOMENDACIONES.....	227
Referencias Bibliograficas .....	229
Anexos.....	231
ANEXO A: Entrevista realizada al encargado del Departamento de Gestión de Residuos Sólidos de la Ciudad de Ibarra. ....	231
ANEXO B. Ficha de Observación aplicada al lugar de estudio.....	233
ANEXO C: Encuesta aplicada a personas que residen en el sector Yacucalle de la ciudad de Ibarra. ....	235
ANEXO D: Tabulación de las preguntas contenidas en las encuestas aplicadas al sector Yacucalle.....	239
ANEXO E: Datasheet Arduino MEGA 2560.....	245
ANEXO F: Datasheet Modulo CC13523R1 .....	252
ANEXO G: Datasheet Modulo HX711 .....	261
ANEXO H: Datasheet Modulo GPS NEO-7M.....	265
ANEXO I: Datasheet Sensor MQ-135 .....	273
ANEXO J: Datasheet Sensor HC-SR04.....	275
ANEXO K: Datasheet Sensor celda de carga (50kg).....	277
ANEXO L: Código Nodo Recolector- Arduino.....	278
ANEXO M: Código Nodo Recolector- CC1352R1 .....	281
ANEXO N: Código Nodo coordinador- CC1352R1.....	285
ANEXO O: Código Phyton.....	288
ANEXO P: Código Phyton ejecutable .....	289
ANEXO Q: MANUAL DE USUARIO .....	290

## INDICE DE TABLAS

Tabla 1. Condiciones de Sensibilidad en el Receptor.....	43
Tabla 2. Valores de SUN FSK PHY SFD para 2-FSK.....	51
Tabla 3. Valores de SUN FSK PHY SFD para 4-FSK.....	51
Tabla 4 .Valores del campo del esquema de modulación.....	54
Tabla 5. Modulación SUN FSK y parámetros de canal.....	55
Tabla 6. Patrón de bits de cola para los codificadores RSC y NRNSC.....	58
Tabla 7. Símbolos PHYR para las opciones SUN OFDM PHY.....	62
Tabla 8. Descripción Tipos de Trama.....	66
Tabla 9. Valores válidos del modo de direccionamiento de destino y Campos del modo de direccionamiento de origen.....	68
Tabla 10. Lista de Stakeholders involucrados.....	107
Tabla 11. Abreviaturas de Requerimientos.....	107
Tabla 12. Requerimientos de Stakeholders .....	108
Tabla 13. Requerimientos Funcionales del Sistema.....	109
Tabla 14. Requerimientos iniciales de arquitectura.....	110
Tabla 15. Descripción Cumple o No cumple.....	112
Tabla 16. Elección de Hardware del sistema .....	115
Tabla 17. Características del microcontrolador Arduino Mega.....	116
Tabla 18. Comparativa Sensores Ultrasónicos.....	119
Tabla 19. Elección sensor para medición de volumen.....	120
Tabla 20. Comparativa Sensores de peso.....	123
Tabla 21. Elección de sensor para medición de peso.....	124
Tabla 22. Comparación sensores de CO2.....	127

Tabla 23. Elección de sensor para medición de CO2.....	128
Tabla 24. Comparación Sistemas Operativos.....	131
Tabla 25. Descripción de entradas del microcontrolador.....	138
Tabla 26. Consumo de corriente de cada componente del sistema embebido.....	142
Tabla 27. Coordenadas Nodos red Mesh.....	157
Tabla 28. Distancia de enlaces.....	158
Tabla 29. Comparativas de valores calculados y valores simulados de los enlaces.....	169
Tabla 30. Cronograma de pruebas.....	204

## INDICE DE FIGURAS

Figura 1. <i>Topología en estrella y Topología peer-to-peer</i> .....	33
Figura 2. <i>Red de Árbol de Clústeres</i> .....	35
Figura 3. <i>Arquitectura del dispositivo LR-WPAN</i> .....	36
Figura 4. <i>Unidad de Datos PHY</i> .....	38
Figura 5. <i>Formato general de la trama PHY</i> .....	39
Figura 6. <i>Formato campo SHR</i> .....	40
Figura 7. <i>Funciones de blanqueamiento de datos y modulación GFSK</i> .....	42
Figura 8. <i>Formato PPDU</i> .....	45
Figura 9. <i>Formato SHR</i> .....	45
Figura 10. <i>Formato del campo SFD</i> .....	45
Figura 11. <i>Campo PHR</i> .....	46
Figura 12. <i>Funciones de modulación y propagación para las PHY de O-QPSK</i> .....	47
Figura 13. <i>Muestra de secuencias de chips de banda base con modelado de pulsos</i> .....	48
Figura 14. <i>Formato del SUN FSK PPDU (sin conmutador de modo)</i> .....	50
Figura 15. <i>Formato del PPDU del conmutador de modo SUN FSK</i> .....	50
Figura 16. <i>Formato del SHR</i> .....	50
Figura 17. <i>Formato del PHR para SUN FSK</i> .....	52
Figura 18. <i>Formato del conmutador de modo PHR para SUN FSK</i> .....	53
Figura 19. <i>Formato del campo New Mode</i> .....	54
Figura 20. <i>Modulador de Referencia SUN FSK</i> .....	57
Figura 21. <i>Extensión del bloque de datos con bits de cola antes de la codificación</i> .....	58
Figura 22. <i>Formato del SUN OFDM PPDU</i> .....	61

Figura 23. Formato del campo de carga útil PHY.....	61
Figura 24. Campos de encabezado PHY para SUN OFDM .....	61
Figura 25. Diagrama del modulador de referencia para SUN OFDM.....	63
Figura 26. Codificador convolucional de frecuencia.....	64
Figura 27. Formato General de trama MAC.....	65
Figura 28. Formato de la Trama Control .....	65
Figura 29. Formato del campo IE .....	71
Figura 30. Utilización del Canal .....	75
Figura 31. Algoritmo CSMA-CA .....	78
Figura 32. Creación del grafo DODAG.....	81
Figura 33. Modelos de Capas.....	82
Figura 34. Proceso de Fragmentación de un paquete IPV6 sobre IEEE 802.15.4.....	84
Figura 35. Encapsulación sin comprensión de una trama IEEE 802.15.4 .....	85
Figura 36. Comprensión de cabeceras IPHC-NHC .....	86
Figura 37. Formato de cabecera IPHC.....	86
Figura 38. Formato de cabecera UDP NHC .....	87
Figura 39. Modelo de Enrutamiento IP.....	88
Figura 40. Modelo de enrutamiento “mesh-under” de paquetes en 6LOWPAN .....	89
Figura 41. Modelo de enrutamiento “route-over” de paquetes 6LoWPAN.....	90
Figura 42. 6LoWPAN utiliza un socket como interfaz .....	91
Figura 43. MPDU del estándar 6LoWPAN.....	91
Figura 44. Netstack Contiki OS.....	92
Figura 45. Stack de 6LoWPAN de Contiki .....	94

Figura 46. <i>Estructura del mensaje MQTT</i> .....	96
Figura 47. <i>Mensajes intercambiados MQTT</i> .....	97
Figura 48. <i>Metodología en Cascada</i> .....	100
Figura 49. <i>Vista general del sistema</i> .....	105
Figura 50. <i>Modelo 1 Hardware</i> .....	112
Figura 51. <i>Modelo 2 Hardware</i> .....	113
Figura 52. <i>Modelo 3 Hardware</i> .....	114
Figura 53. <i>Sensor Ultrasónico HC-SR04</i> .....	118
Figura 54. <i>Sensor ultrasónico a prueba de agua 3-450cm</i> .....	118
Figura 55. <i>Sensor de escala corporal, 50kg de carga, resistencia de celda, sensor de medio puente, 50kg</i> .....	122
Figura 56. <i>Celda de carga 50 Kg</i> .....	123
Figura 57. <i>Sensor Gas MQ-135</i> .....	126
Figura 58. <i>Sensor de Dióxido de Carbono MG-811</i> .....	126
Figura 59. <i>Modelo1 Software</i> .....	129
Figura 60. <i>Modelo 2 Software</i> .....	130
Figura 61. <i>Modelo 3 Software</i> .....	130
Figura 62. <i>Diagrama de Bloques del Sistema</i> .....	133
Figura 63. <i>Diagrama de Flujo del Sistema</i> .....	134
Figura 64. <i>Diagrama de Arquitectura del Sistema</i> .....	136
Figura 65. <i>Diagrama de pines Arduino Mega</i> .....	137
Figura 66. <i>Diagrama de Pines MCU CC1352R1</i> .....	140
Figura 67. <i>Diagrama de conexión del sistema embebido</i> .....	143
Figura 68. <i>Librerías usadas en el código de Arduino</i> .....	144

Figura 69. <i>Identificación de la dirección IPV6 del nodo cliente</i> .....	145
Figura 70. <i>Establecimiento de conectividad entre Cliente UDP (derecha) y Servidor UDP (izquierda)</i> .....	145
Figura 71. <i>Información de nodo raíz y tabla de vecinos</i> .....	146
Figura 72. <i>Interfaz gráfica de representación de datos al usuario final</i> .....	147
Figura 73. <i>Estructura de los objetos y sus variables a medir</i> .....	148
Figura 74. <i>Diagrama de Flujo del funcionamiento de la Plataforma IOT</i> .....	149
Figura 75. <i>Datos obtenidos de los sensores en tiempo real</i> .....	150
Figura 76. <i>Plataforma para cliente MQTT Hive MQTT Broker</i> .....	151
Figura 77. <i>Plataforma Iot para la visualización de datos para el usuario final</i> .....	152
Figura 78. <i>Ubicación del sistema en el contenedor de residuos sólidos</i> .....	153
Figura 79. <i>Delimitación sector Yacucalle</i> .....	154
Figura 80. <i>Ubicación real de los contenedores de residuos sólidos</i> .....	155
Figura 81. <i>Red Mesh en base a ubicación de contenedores</i> .....	156
Figura 82. <i>Simulación de Enlace de nodos C1-C2</i> .....	164
Figura 83. <i>Simulación enlace de nodos C1-C2</i> .....	165
Figura 84. <i>Simulación de Enlace de nodos C1-NC</i> .....	166
Figura 85. <i>Simulación de Enlace de nodos C1-NC</i> .....	166
Figura 86. <i>Simulación de Enlace de nodos C2-NC</i> .....	167
Figura 87. <i>Simulación de Enlace de nodos C2- NC</i> .....	168
Figura 88. <i>Red Mesh, clientes y nodo coordinador</i> .....	170
Figura 89. <i>Librerías utilizadas en contiki</i> .....	171
Figura 90. <i>Variables a almacenar las métricas</i> .....	171
Figura 91. <i>Código de impresión de métricas</i> .....	171

Figura 92. <i>Código de respuesta de ACK desde el nodo cliente</i> .....	172
Figura 93. <i>Impresión valores PDR y PER</i> .....	172
Figura 94. <i>Verificación de direcciones nodo coordinador derecha y nodo cliente izquierda</i> .....	173
Figura 95. <i>Información de métricas de un paquete de 8 bytes enviado desde el cliente al servidor UDP</i> .....	174
Figura 96. <i>Valor correspondiente a CLOCK SECONDS</i> .....	175
Figura 97. <i>Montaje de los elementos del sistema embebido para la toma de datos</i> .....	177
Figura 98. <i>Ubicación de los sensores para la determinación de volumen del contenedor</i> .....	178
Figura 99. <i>Ubicación de las celdas de peso de 50kg para la toma de datos del peso de los residuos</i> .....	178
Figura 100. <i>Ubicación del sensor de CO2 en el contenedor</i> .....	179
Figura 101. <i>Ubicación del módulo GPS para determinar la ubicación del contenedor</i> .....	180
Figura 102. <i>Caja protección nodo coordinador</i> .....	181
Figura 103. <i>Interfaz de simulación</i> .....	183
Figura 104. <i>Tipos de nodos dentro de la red mesh</i> .....	183
Figura 105. <i>Envío de datos al nodo 3</i> .....	184
Figura 106. <i>Envío de datos al nodo 1</i> .....	185
Figura 107. <i>Envío de datos al nodo 3 desde el nodo 1</i> .....	185
Figura 108. <i>Llegada de datos al nodo 2</i> .....	186
Figura 109. <i>Intercambio de mensajes entre nodo 1 y nodo 2</i> .....	187
Figura 110. <i>Identificación de nodo coordinador y nodos clientes</i> .....	187
Figura 111. <i>Nodos cliente reconociendo los nodos vecinos</i> .....	188
Figura 112. <i>Establecimiento de rutas desde los nodos 2 y 3 hacia el nodo 1</i> .....	189
Figura 113. <i>Inicialización de todos los nodos de la mesh</i> .....	190

Figura 114. <i>Establecimiento de la comunicación entre el nodo 1 y nodo 2</i> .....	191
Figura 115. <i>Primer salto del paquete desde el nodo 2</i> .....	191
Figura 116. <i>Segundo salto del paquete desde el nodo 2</i> .....	192
Figura 117. <i>Tercer salto del paquete desde el nodo 2</i> .....	193
Figura 118. <i>Cuarto salto del paquete desde el nodo 2</i> .....	193
Figura 119. <i>Quinto salto del paquete desde el nodo 2</i> .....	194
Figura 120. <i>Llegada del paquete al nodo destino</i> .....	195
Figura 121. <i>Verificación de los próximos saltos desde el Nodo coordinador</i> .....	195
Figura 122. <i>Primer salto desde el nodo 3</i> .....	196
Figura 123. <i>Segundo salto desde el nodo 3</i> .....	197
Figura 124. <i>Tercer salto desde el nodo 3</i> .....	197
Figura 125. <i>Cuarto salto desde el nodo 3</i> .....	198
Figura 126. <i>Quinto salto desde el nodo 3</i> .....	198
Figura 127. <i>Sexto salto desde el nodo 3</i> .....	199
Figura 128. <i>Primer salto del paquete de respuesta desde el nodo 1</i> .....	199
Figura 129. <i>Plataforma sin ninguna alerta por contenedor vacío</i> .....	200
Figura 130. <i>Datos y posición mostrados en la página web</i> .....	201
Figura 131. <i>Contenedor lleno de objetos</i> .....	202
Figura 132. <i>Datos obtenidos por los sensores</i> .....	203
Figura 133. <i>Habilitación del tunslip</i> .....	206
Figura 134. <i>Identificación de interfaz a asignar ip</i> .....	207
Figura 135. <i>Comando configuración de ip a la interfaz</i> .....	207
Figura 136. <i>Verificación de interfaz configurada</i> .....	209
Figura 137. <i>Configuración de archivo bridge.sh con credenciales</i> .....	210

Figura 138. <i>Configuración de dirección ip y de puerto</i> .....	211
Figura 139. <i>Puente se encuentra a la escucha de un envío de datos</i> .....	212
Figura 140. <i>Datos y estado del contenedor en formato json</i> .....	212
Figura 141. <i>Publicación de datos en la página HiveMQTT</i> .....	213
Figura 142. <i>Página actualizada con los datos de un contenedor en tiempo real</i> .....	214
Figura 143. <i>Información sobre el nodo coordinador</i> .....	216
Figura 144. <i>Datos enviados desde el nodo servidor hacia el cliente</i> .....	216
Figura 145. <i>Datos enviados en la distancia con un valor aproximadamente 30 m</i> .....	217
Figura 146. <i>Datos en distancia 65 m aproximadamente, donde se presenta un mensaje</i> .....	218
Figura 147. <i>Información del nodo recolector</i> .....	219
Figura 148. <i>Datos de métricas en el instante 0</i> .....	220
Figura 149. <i>Datos de métricas en distancia 30m aproximadamente</i> .....	221
Figura 150. <i>Datos en distancia igual a 65m aproximadamente, se tiene mayor latencia.</i> .....	221
Figura 151. <i>Valor de PDR en la transmisión</i> .....	222
Figura 152. <i>Valor de PER en la transmisión</i> .....	223
Figura 153. <i>Grafica RSSI en función del tiempo</i> .....	223
Figura 154. <i>Valor de LQI para la transmisión</i> .....	224
Figura 155. <i>Valores de latencia para cada paquete enviado</i> .....	224

# INTRODUCCIÓN

El presente trabajo de integración curricular corresponde al desarrollo de una red mesh LPWAN utilizando el estándar IEEE 802.15.4g para poder recolectar variables de medición de contenedores de residuos sólidos con el objetivo de mejorar rutas de recolección, mitigando la generación de CO<sub>2</sub> por parte de los vehículos recolectores, así como también el malestar en la población debido a la existencia de contenedores que se encuentran sobrecargados donde se puede presenciar fauna urbana. Por todo ello se hace el planteamiento de construir el sistema, el cual se puede implementar en una zona urbana en este caso en el sector Yacucalle de la ciudad de Ibarra, en los siguientes apartados se indica el desarrollo de dicho sistema, desde la construcción del sistema embebido, levantamiento de la red mesh así como también se hace una simulación con el número real de contenedores que se encuentran en el sector de estudio, de igual manera se hace la evaluación de algunas métricas en el enlace inalámbrico. Por último, se realiza la presentación de resultados y conclusiones.

## 1.1. Planteamiento De Problema

En la actualidad, la generación de desechos de todo tipo se ha convertido en un aspecto problemático para la sociedad que ha venido incrementando con los años, por lo que la gestión de los mismos constituye un proceso que debe ser ejecutado con responsabilidad para asegurar el bienestar público.

Para dar cumplimiento a la gestión integral de residuos domiciliarios existe una reglamentación indicada por la Ley 25916, que establece las siguientes etapas: generación, disposición inicial, recolección, transferencia, transporte, tratamiento y disposición final. El

Municipio del Cantón Ibarra en coordinación con la Dirección de Gestión Ambiental son los encargados de dar servicio de recolección de residuos sólidos a la población. Este proceso consiste en recoger los residuos orgánicos e inorgánicos todos los días, de acuerdo con el horario establecido para el Vehículo Recolector de Residuos (VRR).

La ciudad de Ibarra recolecta 135 toneladas diarias de basura, es decir 297 624 libras cada día, que son depositadas en la estación de transferencia Socapamba y trasladada hasta el relleno sanitario San Alfonso, en Ambuquí. Además, según el departamento Gestión Ambiental del GAD Municipal de Ibarra, existen 2 300 contenedores instalados en el 98 % de la población urbana y 25 % en el área rural (el Norte, 2018), lo que contribuye al mejoramiento de la calidad de vida de los ciudadanos.

El cantón Ibarra presenta graves problemas ambientales relacionados con los residuos sólidos, debido al crecimiento poblacional. “Según el VII censo nacional de población realizado en el año 2010, de los 14' 483 499 habitantes que tiene el Ecuador, la población total del cantón Ibarra alcanza a 181.175 habitantes de los cuales 93.389 corresponden a mujeres y 87.786 a hombres” (INEC, 2010), esto a su vez conlleva al aumento de la generación de residuos sólidos poblacional lo que involucra el aumento de generación de basura por parte de los habitantes ya que no existe un adecuado control de los mismos.

Además ”Según la Reforma El Libro VI Del Texto Unificado de Legislación Secundaria del Ministerio del Ambiente en su PARÁGRAFO III, indica que los contenedores para el

almacenamiento temporal de residuos sólidos no peligrosos deberán cumplir como mínimo con: estar cubiertos y adecuadamente ubicados, capacidad adecuada acorde con el volumen generado, contruidos con materiales resistentes y tener identificación de acuerdo al tipo de residuo (Ministerio del Ambiente, 2015). En cuanto a la terminología de volumen generado se refiere a la cantidad de residuos sólidos que se encuentran dentro del contenedor. El incumplimiento de la normativa con respecto al control de llenado de los residuos dentro de los contenedores conlleva al apareamiento de efectos colaterales que pueden ser corregidos entre los que se encuentran la emisión excesiva de gases efecto invernadero por parte de los vehículos recolectores de residuos, la capacidad limitada de los contenedores, así como también los gases producidos por la descomposición de diferentes tipos de desechos lo que involucra malestar de la población y del medio ambiente.

En la actualidad existen contenedores de capacidad limitada ubicados en puntos específicos dentro de la ciudad de Ibarra, los cuales, al no tener control del volumen generado, atraen la presencia de fauna urbana, así como también existe la producción de gases tanto de efecto invernadero por parte de los VRR, como aquellos generados por los mismos desechos que pueden ser nocivos para la salud humana y el medio ambiente. Según datos proporcionados por el Ing. Manuel Enríquez encargado de la Gestión de Residuos Sólidos del Municipio de Ibarra. La recolección se lleva a cabo por diferentes sectores, el sector de estudio corresponde a la ruta Yacucalle, Pílanqui P1 y Los Ceibos P.2, que se lleva a cabo de lunes a domingo en horario de 07h00 a 13h00. Es importante mencionar que el VRR se desplaza calle por calle sin tomar consideración el volumen de cada contenedor.

Mediante una RED MESH LPWAN usando en estándar IEEE 802.15.4g, se busca obtener información sobre el volumen, peso, nivel de CO<sub>2</sub> y ubicación de los diferentes contenedores distribuidos geográficamente en el sector Yacucalle. La red estará formada por Nodos principales que trabajarán como coordinadores. Los nodos recolectores serán ubicados en los distintos contenedores los cuales estarán formados por un sistema embebido compuesto por sensores de volumen, peso, CO<sub>2</sub> y ubicación. Una vez recolectados los datos, estos serán enviados a una estación base para su almacenamiento y posterior uso.

La mala gestión de los desechos sólidos por parte de los organismos como lo son los gobiernos descentralizados en consecuencia del incumplimiento a cabalidad de la normativa proporcionada por el Estado ecuatoriano corresponde un problema de tipo social, el cual afecta en gran magnitud a la población Ibarreña. Por otro lado, no existen proyectos de desarrollo o implementación tecnológica aplicada al control del llenado de los contenedores que contrarreste los inconvenientes. Además, es importante mencionar el aspecto ambiental el cual está afectado en gran magnitud al existir contaminación por parte de los camiones recolectores al emanar CO<sub>2</sub>. De igual forma el apareamiento de fauna urbana lo que causa malestar en la población y por su puesto sin dejar de lado la producción de gases propios de los desechos que puede afectar a la salud de la ciudadanía.

## **1.2. Justificación**

El proyecto busca mejorar la calidad de vida de los ciudadanos así como habla el Plan Nacional de Buen Vivir en su objetivo 7: “Garantizar los derechos de la naturaleza y promover un ambiente sano y sustentable” (PNVB, 2017), mediante el correcto llenado de los contenedores de basura; la cual la hace la estructura institucional del municipio; en donde la población es la principal

afectada y que conviven a diario con esta problemática, son los principales afectados por los impactos ambientales y nocivos para la salud.

Actualmente en la ciudad de Ibarra el manejo inadecuado de los residuos sólidos crea un serio problema ambiental en las calles de la ciudad, además que daña el ornato y la estabilidad ecológica; ya que, al no tener un control adecuado del llenado, puede generar los problemas antes mencionados, además como es de notar la carencia de una cultura organizacional y ambiental provoca poca relevancia en el correcto control del llenado de los contenedores. Es importante tomar en consideración que la generación de estos residuos es debido a una actividad directa del ser humano tanto en los hogares, oficinas, mercados, industrias, entre otros, y como buenos ciudadanos es nuestra obligación aportar para que se tenga una correcta gestión de los desechos sólidos.

Los residuos que se generen en una población o localidad, no como residuo propio, sino, además, como resultado de las distintas ocupaciones, se debe gestionar con responsabilidad. Por otro lado, es relevante intentar disminuir de alguna manera la contaminación generada por la acumulación de los residuos sólidos en sitios denominados públicos del sector, como lo son los parques, aceras, esquinas y principalmente que se haga un adecuado control del volumen de los contenedores ubicados a lo largo de las distintas calles de la localidad.

Hoy en día no existe un control de llenado que pueda tratar y manejar los residuos sólidos que permita realizar acciones para ayudar a la administración encargada o designada en la recolección de los desechos de los contenedores, por lo que, es necesario emprender acciones inmediatas, para atenuar los impactos socioambientales y de salud de los habitantes.

La presente investigación se sustenta mediante el objetivo 2 del programa Nacional para la gestión Integral de desechos sólidos que dice: “Diseñar e implementar un modelo estandarizado de gestión integral de Residuos Sólidos en base a parámetros geográficos, poblacionales, generación y caracterización de los residuos sólidos” (PNGIDS,2010).

Este proyecto se verá orientado por el Anexo 6 del Texto Unificado de Legislación Secundaria del Ministerio del Ambiente: “Esta Norma establece los criterios para el manejo de los desechos sólidos no peligrosos, desde su generación hasta su disposición final” (TULSMA,2003)

### **1.3. Objetivos**

#### **1.3.1. Objetivo General**

Diseñar una red Mesh LPWAN mediante el uso del estándar 802.15.4g para la recolección de datos sobre variables presentes en los contenedores de residuos sólidos que permita un control de los mismos.

#### **1.3.2. Objetivos Específicos**

Recolectar información desde diferentes recursos informativos que permita la construcción de un marco teórico para que aporte a la ejecución del proyecto.

Construir modelos de Hardware y Software mediante la investigación que apoyen a la realización del diseño para tener un modelo funcional.

Diseñar la red Mesh LPWAN en base al cumplimiento de los requerimientos establecidos para obtener información de diferentes variables de los contenedores de residuos sólidos.

Realizar pruebas de funcionamiento de la red ubicando el sistema embebido en los contenedores de residuos sólidos para asegurar su correcto funcionamiento.

## 2. Capítulo II

### Fundamentación Teórica

En el segundo capítulo se realiza la recopilación del fundamento teórico para sustentar los argumentos presentados en el proyecto. Se abarcarán diferentes temáticas relacionadas con el diseño del sistema, entre las cuales se tiene: Red Mesh, Red de Área Amplia de Baja Potencia, Estándar IEEE 802.15.4g, en base al uso de la metodología en cascada se hace referencia a requerimientos necesarios de hardware y software para la realización del diseño de la red mallada.

El estándar IEEE 802.15.4 proporciona las capas de red más baja fundamentales para una red de área personal inalámbrica (WPAN). Los principales requisitos son de bajo costo, la comunicación ubicua de baja velocidad entre dispositivos, bajo consumo de batería y un largo rango de operación, por ejemplo 1 Km. (IEEE Comsoc, 2020).

La versión más actual del Estándar IEEE 802.15.4g provee en su estructura mejoras como:

- Soporte para una base de tiempo compartida con un mecanismo de marca de tiempo de datos.
- Soporte para programación de balizas.
- Sincronización de mensajes de transmisión en redes de área personal (PAN) habilitadas para balizas: seguridad de capa MAC mejorada.

En cuanto a los componentes de IEEE 802.15.4g se hace referencia a que posee un dispositivo con una única interfaz de radio que implementa un IEEE Std 802.15.4 MAC y PHY.

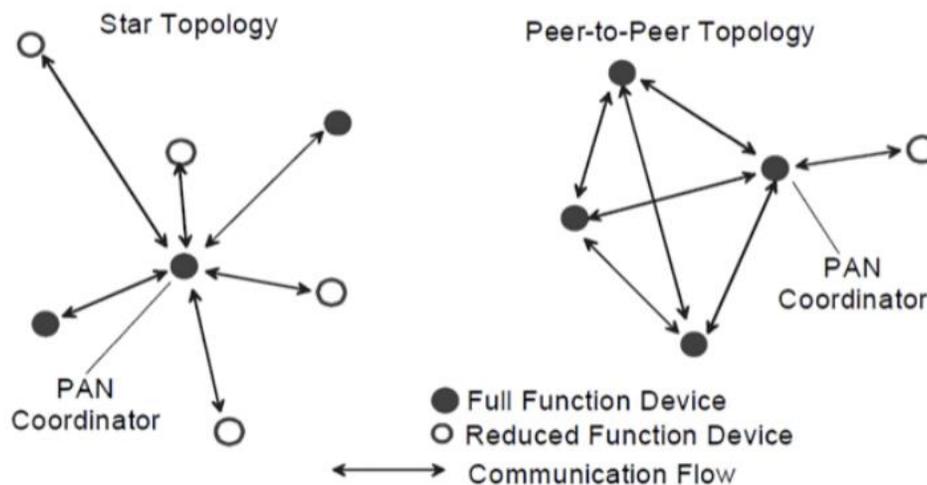
Dos o más dispositivos que se comunican en el mismo canal físico constituyen una red de área personal inalámbrica (WPAN). Un WPAN incluye al menos un Dispositivo de Función Completa (FFD), que opera como coordinador del PAN.

### 2.1.1. Topologías de Red

Las topologías de aplicabilidad del estándar IEEE 802.15.4g dependen de los propósitos de uso, un IEEE 802.15.4 LR-WPAN opera en cualquiera de las dos topologías: la topología en estrella o la topología peer-to-peer.

**Figura 1**

*Topología en estrella y Topología peer-to-peer*



Fuente: (IEEE Comsoc, 2020)

Como se muestra en la Figura 1, en la topología en estrella, la comunicación se establece entre los dispositivos y un único controlador central, denominado coordinador PAN. Todos los dispositivos que operan en una red de cualquier topología tienen direcciones únicas, denominadas direcciones extendidas, por otro lado, se usan direcciones cortas al momento de la asociación. Un dispositivo utilizará la dirección

extendida o la dirección corta para la comunicación dentro del PAN. Las aplicaciones que se benefician de una topología en estrella se relacionan con la automatización del hogar, periféricos de computadoras personales, juegos y atención médica personal.

Al igual que la topología en estrella en la red peer-to-peer también se tiene un coordinador PAN; no obstante, la diferencia está en que cualquier dispositivo puede comunicarse con cualquier otro siempre que estén dentro del alcance de los demás, lo que constituye la denominada red en malla.

Cada PAN independiente selecciona una ID única que permite la comunicación entre dispositivos dentro de una red utilizando direcciones cortas y habilita transmisiones entre dispositivos a través de redes independientes.

Las aplicaciones de este tipo de red incluyen control y monitoreo industrial, redes de sensores inalámbricos, rastreo de activos e inventarios, agricultura inteligente y seguridad.

#### **2.1.1.1. Estructura de la topología peer-to-peer.**

En una topología peer-to-peer, cada dispositivo es capaz de comunicarse con cualquier otro dispositivo dentro de su rango de radiocomunicaciones. Un dispositivo se considera como coordinador PAN, cuando ha sido el primero en comunicarse en el canal. La red de árbol de clúster es un ejemplo de red peer-to-peer, es un caso especial en la que la mayoría de los dispositivos son FFD. Un RFD se conecta a una red de árbol de clúster como un dispositivo hoja al final de una rama porque los RFD no permiten que otros dispositivos se asocien.

Un dispositivo FFD se puede considerar como coordinador PAN y es aquel que posee mayores recursos computacionales que cualquier otro dispositivo del PAN. Asimismo, se puede

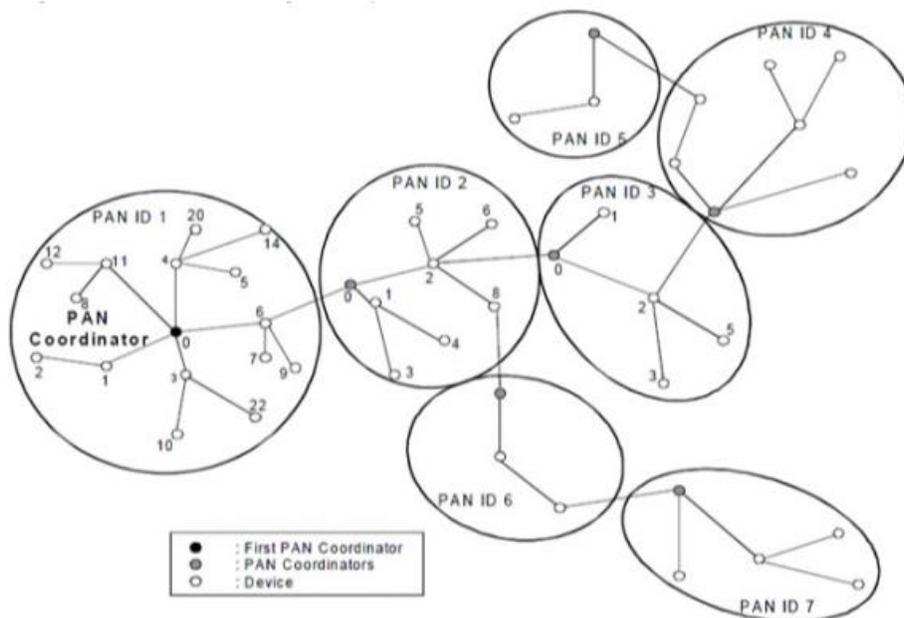
comunicar con otros coordinadores sean FFD o RFD como es el caso de un dispositivo final de la rama, los cuales no participan del enrutamiento de mensajes.

Un dispositivo coordinador PAN crea un grupo eligiendo un ID PAN únicamente, es decir de ningún otro tipo de comunicación con los demás coordinadores, en el caso de existir una disputa entre coordinadores para tomar el control principal de la red se realizará un proceso en la capa superior. Si un dispositivo quiere unirse a una red, el coordinador de dicha red debe autorizar que forme parte de la lista de vecinos, de lo contrario el dispositivo rechazado buscara otra red a la que pueda integrarse.

Ya formado el clúster, el primer coordinador le indica a uno de sus vecinos que se convierta en coordinador de un nuevo clúster para que exista comunicación entre clústeres y de esa manera ir formando una red multiclúster como se indica en la Figura 2 donde se puede apreciar la relación entre padres e hijos.

**Figura 2**

*Red de Árbol de Clústeres*



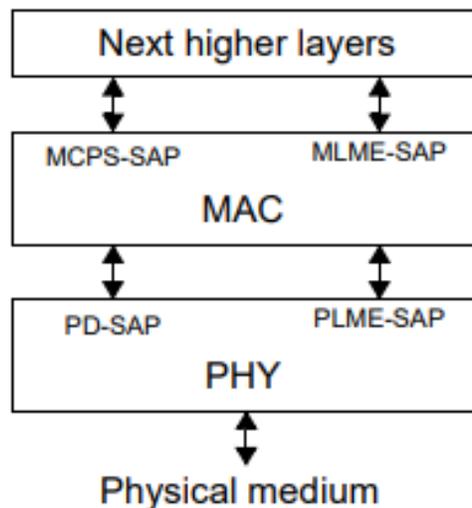
Fuente: (IEEE Comsoc, 2020).

### 2.1.2. Arquitectura IEEE 802.15.4

La arquitectura IEEE 802.15.4 se define en varias capas, donde cada una es responsable de una parte del estándar y ofrece servicios a las capas superiores. Las interfaces entre las capas sirven para definir los enlaces lógicos que se describen en este estándar. Las siguientes capas superiores, que se muestran en la Figura 3, consisten en una capa de red, que proporciona configuración de red, manipulación y enrutamiento de mensajes, y una capa de aplicación, que proporciona la función anunciada del dispositivo.

**Figura 3**

*Arquitectura del dispositivo LR-WPAN*



Fuente: (IEEE Comsoc, 2020).

#### 2.1.2.1.PHY.

Entre muchas características PHY permite la activación y desactivación del transceptor de radio, detección de energía (ED), selección de canal, indicación de calidad de enlace

(LQI), CCA, rango y transmisión, así como recepción de paquetes a través del medio físico.

### **2.1.2.2.Subcapa MAC.**

La subcapa MAC ofrece dos servicios: el servicio de datos MAC que permite la transmisión y recepción de unidades de datos de protocolo MAC (MPDU) a través del servicio de datos PHY.

El segundo es el servicio de gestión MAC que se interconecta con el punto de acceso al servicio (SAP) de la entidad de gestión de la subcapa MAC (MLME) (conocido como MLME-SAP). La subcapa MAC admite la gestión de balizas, el acceso al canal, la gestión del intervalo de tiempo garantizado (GTS), la validación de tramas, la entrega de tramas reconocidas, la asociación y la disociación.

### **2.1.3. *Modelo de transferencia de datos***

#### **2.1.3.1.Transferencia de datos a un coordinador.**

En el momento en que un dispositivo quiere transferir datos a un coordinador en un PAN habilitado para baliza, primero escucha la baliza. Al encontrar la baliza, el dispositivo se sincroniza con la estructura de supertrama. En el momento adecuado, el dispositivo transmite su trama de datos al coordinador.

Por otro lado, cuando un dispositivo desea transferir datos en un PAN no habilitado para balizas, únicamente transmite su trama de datos al coordinador.

### **2.1.3.2. Transferencia de datos de un coordinador.**

En el modo de transferencia de datos de un coordinador a un dispositivo en un PAN habilitado para baliza, en la baliza indica que el mensaje de datos está pendiente. Luego el dispositivo escucha habitualmente la baliza y, si existe un mensaje pendiente, transmitirá un comando de solicitud de datos. El coordinador envía la trama de datos pendiente. Una vez completada con éxito la transferencia de datos, se elimina el mensaje de la lista de mensajes pendientes en la baliza.

Al contrario, cuando un coordinador desea transferir datos a un dispositivo en un PAN no habilitado para balizas, almacena los datos para que el dispositivo apropiado haga contacto y solicite los datos.

### **2.1.3.3. Transferencia de datos peer-to-peer**

Para las transferencias peer-to-peer, un dispositivo de destino recibirá constantemente o se sincronizará con el dispositivo de origen. En el primer caso, el dispositivo originador intenta transmitir los datos cuando obtiene acceso al canal.

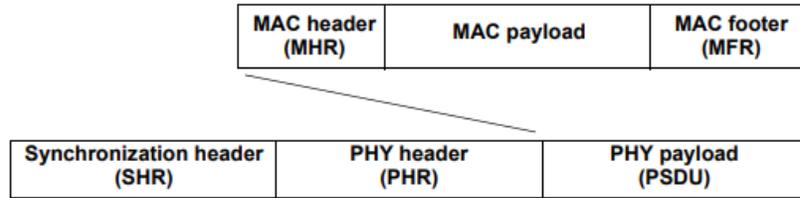
### **2.1.4. Estructura de la Trama**

Las estructuras de la trama poseen un diseño complejo y robusto al momento de transmitir en un canal con ruido.

Las tramas MAC se pasan a la PHY como PSDU, que se convierte en la carga útil de la PHY. En la Figura 4 se muestra la unidad de datos del protocolo PHY (PPDU).

## **Figura 4**

*Unidad de Datos PHY*



Fuente: (IEEE Comsoc, 2020)

### 2.1.5. Capa PHY

La capa PHY es la responsable de las siguientes funciones:

- Activación y desactivación del transceptor de radio
- ED dentro del canal actual
- LQI para paquetes recibidos
- CCA para CSMA-CA
- Selección de frecuencia de canal
- Transmisión y recepción de datos

#### 2.1.5.1. Formato General de la Trama PHY.

En la capa física la PDU se compone de tres espacios divididos en cuatro campos, como se ilustra en la Figura 5.

### Figura 5

*Formato general de la trama PHY*



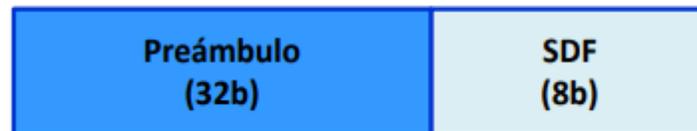
Fuente: (Acosta, 2006)

**SHR (Synchronization Header):** Permite que un dispositivo receptor se sincronice y bloquee mediante un flujo de bits. Como se indica en la Figura 6 el campo a su vez se subdivide en los siguientes campos:

- Preámbulo: El transceptor utiliza este campo para obtener el chip y el símbolo de sincronización de un mensaje entrante.
- SDF (Start-of-Frame Delimiter): Indica el final del SHR y el inicio de datos de la trama.

### Figura 6

*Formato campo SHR*



Fuente: (Acosta, 2006)

**PHR (PHY Header):** Este campo indica la longitud de la información de la trama.

**Carga Útil:** Campo de longitud variable que lleva el contenido de capas superiores, la misma que no debe exceder los 127 bytes (Acosta, 2006).

### **2.1.5.2. Especificaciones de Radio.**

#### *a. Nivel de entrada máximo del receptor de la señal deseada*

Corresponde al nivel máximo de potencia de la señal deseada en la entrada del receptor.

Considerando como sensibilidad del receptor, a la potencia de entrada más baja para la que se cumplen las condiciones de la Tasa de error del paquete PER.

PER: Fracción promedio de paquetes transmitidos que no se reciben correctamente.

#### *b. Receptor ED*

La medición de ED del receptor es una estimación de la potencia de la señal recibida dentro del ancho de banda del canal. El tiempo de medición de ED, para promediar, será igual a 8 períodos de símbolo.

#### *c. Indicador de calidad de enlace (LQI)*

Es una caracterización de la fuerza y / o calidad de un paquete recibido. La medición LQI se puede implementar usando el receptor ED, una estimación de la relación señal / ruido o una combinación de estos métodos. El parámetro de LQI se medirá para cada paquete recibido.

#### *d. Clear channel assessment (CCA)*

CCA trabaja con 3 modos:

CCA Modo 1: Energía por encima del umbral. CCA informará un medio ocupado al detectar cualquier energía por encima del umbral de ED.

Modo CCA 2: Solo detección de portadora. CCA informará un medio ocupado solo tras la detección de una señal que cumpla con este estándar con las mismas características de modulación y propagación de la PHY que está actualmente en uso por el dispositivo.

Modo CCA 3: detección de portadora con energía por encima del umbral. CCA notificará un medio ocupado utilizando una combinación lógica de:

- Detección de una señal con las características de modulación y propagación de este modo.
- Energía por encima del umbral ED, donde el operador lógico puede ser AND u OR.

Para cualquiera de los modos CCA, si la PHY recibe una solicitud para realizar CCA durante la recepción de una PPDU, CCA informará un medio ocupado.

### 2.1.5.3. Modulaciones 802.15.4g.

#### a. Modulación GFSK

La velocidad de datos de la modulación GFSK PHY será de 100 kb/s.

En la Figura 7, se ilustra un diagrama de bloques del proceso de blanqueamiento de datos y la modulación GFSK.

#### Figura 7

*Funciones de blanqueamiento de datos y modulación GFSK*



Fuente: (IEEE Comsoc, 2020)

- Blanqueamiento de datos

El blanqueamiento de datos viene a ser el OR exclusivo del campo de carga útil PHY con la secuencia PN9. Esto será realizado por el transmisor y se indica en la Ecuación 1:

$$E_n = R_n \oplus PN9_n$$

Ec. 1

Donde:

$E_n$  es el bit de datos que se blanquea

$R_n$  es el bit blanqueado

$PN9_n$  es el bit de secuencia PN9

Para cada paquete transmitido con blanqueamiento de datos,  $R_0$  es el primer bit de la PSDU y el índice  $n$  se incrementa para los bits subsiguientes de la PSDU.

Las secuencias de bits serán moduladas en la portadora utilizando GFSK con un índice de modulación de uno, donde el filtro gaussiano BT es 0.5, y un valor de bit de uno se transmite mediante el cambio de la frecuencia por encima de la frecuencia central del canal y el valor de bit de cero se transmite con el cambio de la frecuencia por debajo de la frecuencia central del canal actual.

Es importante mencionar que la modulación GFSK PHY opera en la banda de frecuencia de 920,8 a 928 MHz.

Por otro lado, la velocidad de símbolo de GFSK PHY será de 100 ksymbol / s con una precisión de  $\pm 40 \times 10^{-6}$ .

Un dispositivo GFSK PHY compatible deberá ser capaz de alcanzar una sensibilidad de – 85 dBm o mejor, parámetro especificado en la Tabla 1.

## Tabla 1

## Condiciones de Sensibilidad en el Receptor

<b>Término</b>	<b>Definición del término</b>	<b>Condiciones</b>
<b>Tasa de error de paquete (PER)</b>	Fracción promedio de paquetes transmitidos que no se reciben correctamente.	Promedio medido sobre datos PSDU aleatorios.
		Longitud de PSDU = 250 octetos para SUN PHY con velocidades de datos de 50 kb / s y superiores, 20 octetos para todas las demás PHY.
<b>Sensibilidad del receptor</b>	Potencia de entrada más baja para la que se cumplen las condiciones PER.	PER <10% para SUN PHY. PER <1% para todos los demás PHY. Potencia medida en los terminales de la antena sin interferencia.

Fuente: (IEEE Comsoc, 2020)

Considerando los niveles mínimos de rechazo de interferencia, se tiene los canales adyacentes, canales deseados y canales alternativos, es así como el canal adyacente es uno a cada lado del canal deseado que tiene la frecuencia más cercana al canal deseado, y el canal alternativo es uno más retirado del canal adyacente. Entonces los niveles mínimos de rechazo de interferencia del receptor se dan:

Rechazo del canal adyacente: 0 dB

Rechazo del canal alternativo: 24 dB

La tolerancia de la frecuencia central de transmisión de GFSK PHY será de  $\pm 40 \times 10^{-6}$ .

Además, el GFSK PHY deberá poder transmitir a un nivel de potencia de al menos  $-3$  dBm.

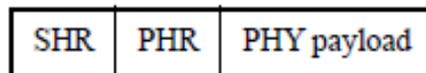
El GFSK PHY deberá tener un nivel de entrada máximo del receptor superior o igual a  $-20$  dBm.

*b. Modulación O-QPSK PHY*

En la modulación O-QPSK PHY el formato del PPDU será el de la Figura 8.

**Figura 8**

*Formato PPDU*

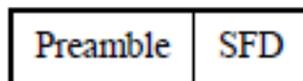


Fuente: (IEEE Comsoc, 2020)

Entonces, considerando cada campo se inicia por el campo **SHR** indicado en la Figura 9, el cual a su vez consta de un campo Preámbulo y un campo SFD.

**Figura 9**

*Formato SHR*



Fuente: (IEEE Comsoc, 2020)

La longitud del campo de preámbulo para las PHY de O-QPSK consiste en 8 símbolos (4 octetos), y los bits del campo de preámbulo serán ceros binarios. En cambio, el campo SFD tendrá el formato indicado en la Figura 10.

**Figura 10**

*Formato del campo SFD*

<b>Bits: 0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
1	1	1	0	0	1	0	1

Fuente: (IEEE Comsoc, 2020)

El SFD es un campo que indica el final del SHR y el comienzo del paquete de datos.

La Figura 11, muestra el formato del campo PHR:

**Figura 11**

*Campo PHR*

<b>Bits: 0-6</b>	<b>7</b>
Frame Length	Reserved

Fuente: (IEEE Comsoc, 2020)

El cual especifica el número total de octetos contenidos en la PSDU (carga útil PHY).

Por último, se tiene el campo **Payload PHY** que lleva la PSDU.

- Modulación y Propagación.

La modulación O-QPSK PHY emplea un tipo de modulación cuasi ortogonal 16-ary. Durante cada período de símbolo de datos, se utilizan cuatro bits de información para seleccionar 1 de 16 secuencias de ruido pseudoaleatorio (PN) casi ortogonales a transmitir. Las secuencias PN para los símbolos de datos sucesivos se concatenan, y la secuencia de chips agregada se modula en la portadora utilizando codificación por desplazamiento de fase en cuadratura desplazada (O-QPSK).

En cuanto a la velocidad de datos del O-QPSK PHY será de 250 kb / s cuando opere en las bandas de 2450 MHz, 915 MHz, 780 MHz o 2380 MHz y será de 100 kb / s cuando opere en la banda de 868 MHz.

El soporte para la PHY O-QPSK de 2450 MHz es obligatorio cuando se opera en la banda de 2450 MHz. El soporte para O-QPSK PHY de 2380 MHz es obligatorio cuando se opera en la banda de 2380 MHz.

El O-QPSK PHY no es obligatorio en la banda de 868 MHz o 915 MHz. Si el O-QPSK PHY se usa en la banda de 868 MHz o 915 MHz, entonces el mismo dispositivo será capaz de señalizar usando el BPSK PHY también.

En la Figura 12, se muestra un diagrama de bloques de la modulación O-QPSK.

## Figura 12

*Funciones de modulación y propagación para las PHY de O-QPSK*



Fuente: (IEEE Comsoc, 2020)

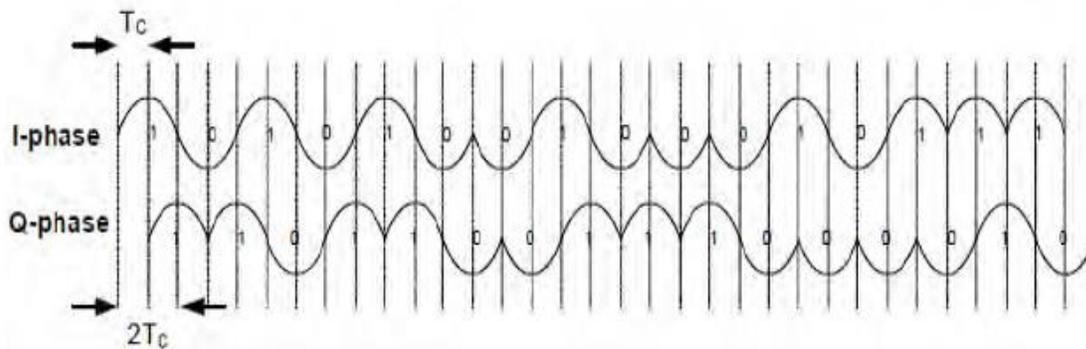
Las secuencias de chips que representan cada símbolo de datos se modulan en la portadora utilizando O-QPSK con forma de pulso de medio seno. Los chips de índice par se modulan en la portadora en fase (I) y los chips de índice impar se modulan en la portadora de fase en cuadratura (Q). En las bandas de 2450 MHz y 2380 MHz, cada símbolo de datos está representado por una secuencia de 32 chips, por lo que la tasa de chip es 32 veces la tasa de

símbolos. En las bandas de 915 MHz, 868 MHz y 780 MHz, cada símbolo de datos está representado por una secuencia de 16 chips, por lo que la tasa de chip es 16 veces la tasa de símbolos.

En la Figura 13, se ilustra una secuencia de chip de banda base de muestra con conformación de pulso de medio seno.

### Figura 13

*Muestra de secuencias de chips de banda base con modelado de pulsos*



Fuente: (IEEE Comsoc, 2020)

La modulación O-QPSK opera en las siguientes bandas:

- 779–787 MHz
- 868.0–868.6 MHz
- 902–928 MHz
- 2360–2400 MHz
- 2400.0–2483.5 MHz

En condiciones favorables un dispositivo es capaz de lograr una sensibilidad de -85 dBm o mejor.

En cuanto a los requisitos mínimos de rechazo de interferencia de receptor para las bandas, 780 MHz, 915 MHz, 2380 MHz y 2450 MHz.

Rechazo del canal adyacente: 0dB

Rechazo del canal alternativo: 30dB

En O-QPSK la tolerancia de frecuencia central de transmisión O-QPSK PHY será de  $\pm 40 \times 10^{-6}$  máximo.

Además, O-QPSK PHY deberá poder transmitir a un nivel de potencia de al menos  $-3$  dBm.

- Recepción ED

El valor ED mínimo (cero) indicará la potencia recibida inferior a 10 dB por encima de la sensibilidad del receptor más baja especificada, en dBm, para la PHY.

- Medición LQI

LQI se puede implementar utilizando el receptor ED, una estimación de la relación señal / ruido o una combinación de estos métodos. La medición de LQI se realizará para cada paquete recibido.

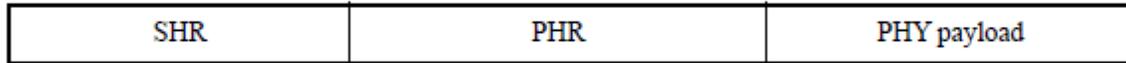
### *c. Modulación SUN FSK PHY*

Se hace uso del esquema MPM, el cual se introduce para facilitar la interoperabilidad y la negociación entre los posibles coordinadores de PAN que trabajan con diferentes PHY. En este proceso, el MPM permitirá que un coordinador detecte una red en funcionamiento existente durante su fase de descubrimiento, utilizando el modo de señalización común (CSM). El CSM que se analiza aquí es un modo de capa física común, que se utiliza entre los dispositivos SUN que implementan el esquema MPM (Xue, 2015).

Para la presente modulación se tendrá el formato de trama de la Figura 14:

## Figura 14

*Formato del SUN FSK PDU (sin conmutador de modo)*



Fuente: (IEEE Comsoc, 2020)

Aunque puede adoptar el formato de la Figura 15 si el cambio de modo está habilitado.

## Figura 15

*Formato del PDU del conmutador de modo SUN FSK*

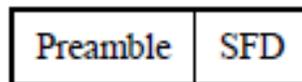


Fuente: (IEEE Comsoc, 2020)

El campo **SHR** tendrá el formato ilustrado en la Figura 16.

## Figura 16

*Formato del SHR*



Fuente: (IEEE Comsoc, 2020)

Es así como el campo **preámbulo** debe contener phyFskPreambleLength múltiplos de la secuencia de 8 bits “01010101” para 2-FSK y phyFskPreambleLength múltiplos de la secuencia de 16 bits “0111 0111 0111 0111” para 4-FSK.

El **SFD** para 2-FSK consta de una secuencia de 2 octetos seleccionada de los valores mostrados en la Tabla 2. El SFD para 4-FSK será una secuencia de 4 octetos seleccionada de los valores indicados en la Tabla 2.

**Tabla 2**

*Valores de SUN FSK PHY SFD para 2-FSK*

	<b>Valor SFD para formato codificado (b0 – b15)</b>	<b>Valor SFD para formato no codificado (b0 – b15)</b>
<b>phySunFskSfd = 0</b>	0110 1111 0100 1110	1001 0000 0100 1110
<b>phySunFskSfd = 1</b>	0110 0011 0010 1101	0111 1010 0000 1110

Fuente: (IEEE Comsoc, 2020)

**Tabla 3**

*Valores de SUN FSK PHY SFD para 4-FSK*

	<b>Valor SFD para formato codificado (b0 – b31)</b>	<b>Valor SFD para formato no codificado (b0 – b31)</b>
<b>phySunFskSfd = 0</b>	0111 1101 1111 1111 0111	1101 0111 0101 0101 0111
	0101 1111 1101	0101 1111 1101

---

<b>phySunFskSfd = 1</b>	0111 1101 0101 1111 0101	0111 1111 1101 1101 0101
	1101 1111 0111	0101 1111 1101

---

Fuente: (IEEE Comsoc, 2020)

En el formato del **PHR** que se ilustra en la Figura 17, donde el campo de cambio de modo se pondrá a cero, lo que indica que todo el paquete será transmitido a una única velocidad de datos y utilizando un único esquema de modulación.

El campo de tipo FCS indica la longitud del campo FCS que se incluye en la MPDU. Por otro lado, el campo Tipo de FCS se pondrá a cero para una FCS de 4 octetos y se pondrá a uno para una FCS de 2 octetos.

El campo Blanqueamiento de datos si se establece en uno indica que se usa el blanqueamiento de datos de la PSDU durante la transmisión, caso contrario se establecerá en cero. Por último, el campo Longitud de trama es un entero sin signo que se fijará al número total de octetos contenidos en la PSDU y se transmitirá primero a MSB.

### Figura 17

*Formato del PHR para SUN FSK*

Bits: 0	1-2	3	4	5-15
Mode Switch	Reserved	FCS Type	Data Whitening	Frame Length

Fuente: (IEEE Comsoc, 2020)

El campo **Modo Switch PHR** ilustrado en la Figura 18, posee un campo llamado cambio de modo el cual, al establecerse en uno, indicará que se va a producir un cambio de modo. El modo de la siguiente PPDU transmitida (es decir, la PPDU de nuevo modo) será como se describe en los campos restantes contenidos en el PHR en la Figura 18. Si el nuevo modo es SUN FSK, el nuevo modo PPDU es el mostrado en la Figura 14, excepto que el preámbulo y SFD son opcionales.

El campo Parámetro de cambio de modo, es el índice que define los parámetros de cambio de modo que se utilizarán. Si el campo Parámetro de cambio de modo indica una entrada no admitida, el receptor descartará el paquete y permanecerá en el modo PHY actual.

En la Figura 18, se indica el formato de trama del campo Modo Switch PHR.

### Figura 18

*Formato del conmutador de modo PHR para SUN FSK*

Bits: 0	1-2	3	4-10	11-14	15
Mode Switch	Mode Switch Parameter	New Mode FEC	New Mode	Checksum	Parity Check

Fuente: (IEEE Comsoc, 2020)

El campo New Mode FEC indica si el paquete que sigue al conmutador de modo PPDU se transmite utilizando FEC. Un valor de cero especifica que el paquete de modo nuevo se transmite sin FEC y un valor de uno indicará lo contrario. Si el paquete de nuevo modo tiene un SFD y, por lo tanto, información de codificación de paquete, el SFD anulará el valor del campo FEC de nuevo modo.

Por otro lado, el formato de New Mode, mostrado en la Figura 19, donde a su vez posee un campo llamado Página que se establecerá en cero para indicar la página del canal nueve o en uno para indicar la página del canal diez.

**Figura 19**

*Formato del campo New Mode*

<b>Bits: 0</b>	<b>1–2</b>	<b>3–6</b>
<b>Page</b>	<b>Modulation Scheme</b>	<b>Mode</b>

Fuente: (IEEE Comsoc, 2020)

El campo Esquema de modulación como su nombre lo indica, establecerá el parámetro de acuerdo con la información proporcionada en la Tabla 4.

**Tabla 4**

*Valores del campo del esquema de modulación*

<b>Valor de campo</b>	<b>Descripción</b>
<b>0</b>	SUN FSK
<b>1</b>	SUN OFM
<b>2</b>	SUN O-QPSK
<b>3</b>	Modos Adicionales

Fuente: (IEEE Comsoc, 2020)

El campo Modo, indica el nuevo modo de funcionamiento. Cuando el campo Página es cero (canal página nueve), la interpretación del campo Modo dependerá del esquema de modulación.

Cuando el campo Página es uno (canal página diez), el nuevo modo PHY se define mediante el mecanismo PHY genérico de SUN FSK.

En la Tabla 5, se presentan los parámetros de modulación y canal para los modos de operación PHY definidos de manera estándar para 867 MHz, 870 MHz, 915 MHz-a, 915 MHz-b, 915 MHz-c, 915 MHz-d, 915 MHz-e, Bandas de 919 MHz, 920 MHz, 920 MHz-a y 920 MHz-b. Para estas bandas, un dispositivo debe admitir ambos modos de funcionamiento n.º 1 y n.º 2 y, además, puede admitir los modos de funcionamiento n.º 3, n.º 4 y n.º 5.

**Tabla 5**

*Modulación SUN FSK y parámetros de canal*

Band designation	Parameter	Operating mode	Operating mode #2	Operating mode #3	Operating mode #4	Operating mode #5
<b>867</b>	Data rate (kb/s)	50	100	150	200	300
	Modulation	2-FSK	2-FSK	2-FSK	2-FSK	2-FSK
	Modulation index	0.5	0.5	0.5	0.5	0.5
	Channel spacing (kHz)	100	200	200	400	400
<b>870</b>	Data rate (kb/s)	50	100	150	-	-
	Modulation	2-FSK	2-FSK	2-FSK	2-FSK	2-FSK
	Modulation index	0.5	0.5	0.5	-	-
	Channel spacing (kHz)	100	200	200	-	-
<b>915-a</b>	Data rate (kb/s)	50	100	150	200	300
	Modulation	2-FSK	2-FSK	2-FSK	2-FSK	2-FSK
	Modulation index	1.0	0.5	0.5	0.5	0.5

	Channel spacing (kHz)	100	200	200	400	400
	Data rate (kb/s)	50	100	150	200	300
	Modulation	2-FSK	2-FSK	2-FSK	2-FSK	2-FSK
<b>915-b</b>	Modulation index	1.0	0.5	0.5	0.5	0.5
	Channel spacing (kHz)	200	200	200	400	400
	Data rate (kb/s)	50	100	150	200	300
	Modulation	2-FSK	2-FSK	2-FSK	2-FSK	2-FSK
<b>915-c</b>	Modulation index	1.0	0.5	0.5	0.5	0.5
	Channel spacing (kHz)	200	200	200	400	400
	Data rate (kb/s)	50	100	150	200	300
	Modulation	2-FSK	2-FSK	2-FSK	2-FSK	2-FSK
<b>915-d</b>	Modulation index	1.0	0.5	0.5	0.5	0.5
	Channel spacing (kHz)	200	200	200	400	400
	Data rate (kb/s)	50	100	150	200	300
	Modulation	2-FSK	2-FSK	2-FSK	2-FSK	2-FSK
<b>915-e</b>	Modulation index	1.0	0.5	0.5	0.5	0.5
	Channel spacing (kHz)	200	200	200	400	400
	Data rate (kb/s)	50	100	150	200	300
	Modulation	2-FSK	2-FSK	2-FSK	2-FSK	2-FSK
<b>919</b>	Modulation index	1.0	0.5	0.5	0.5	0.5
	Channel spacing (kHz)	200	200	200	400	-
	Data rate (kb/s)	50	100	200	400	-
<b>920</b>	Modulation	2-FSK	2-FSK	2-FSK	4-FSK	-
	Modulation index	1.0	1.0	1.0	0.33	-

	Channel spacing (kHz)	200	400	600	-	
	Data rate (kb/s)	50	100	150	-	-
	Modulation	2-FSK	2-FSK	2-FSK	-	-
<b>920-a</b>	Modulation index	1.0	0.5	0.5	-	-
	<b>Channel spacing (kHz)</b>	200	200	200	-	-
	Data rate (kb/s)	50	100	150	200	300
	Modulation	2-FSK	2-FSK	2-FSK	2-FSK	2-FSK
<b>920-b</b>	Modulation index	1.0	0.5	0.5	0.5	0.5
	Channel spacing (kHz)	200	200	200	400	400

Fuente: (IEEE Comsoc, 2020)

- Modulador de referencia

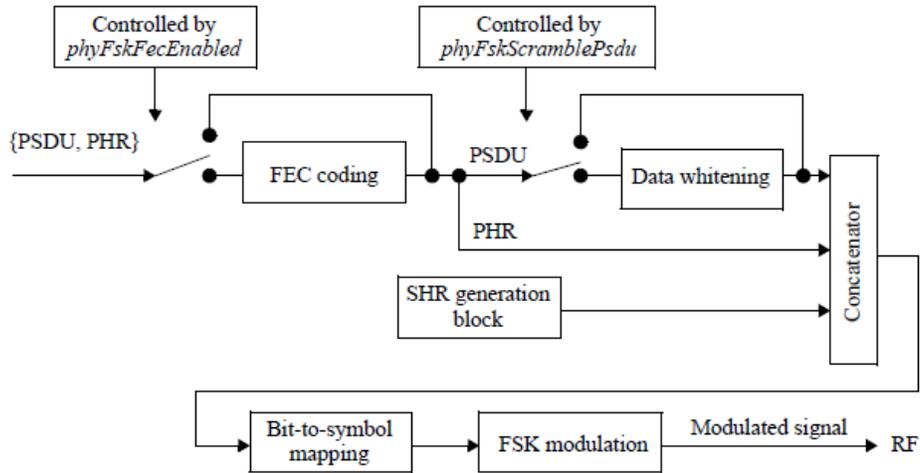
El diagrama de bloques de la modulación SUN FSK PHY se ilustra en la Figura 20, utilizado como referencia para la especificación de funciones de procesamiento de flujo de datos SUN FSK PHY.

Cuando FEC se encuentra habilitado, el PHR y PSDU se procesarán para codificar como un solo bloque de datos. La codificación solo será aplicada sobre la PSDU cuando el blanqueamiento de datos está habilitado.

Todos los campos de la PPDU deben utilizar la misma velocidad de símbolo y orden de modulación.

## Figura 20

*Modulador de Referencia SUN FSK*



Fuente: (IEEE Comsoc, 2020)

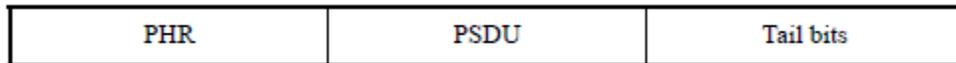
- FEC

El valor de PSDU es cero en el caso de un paquete de cambio de modo.

Inmediatamente después de codificar el PHR y PSDU, se insertará en el codificador una secuencia de terminación con longitud  $L_{TAIL} = 3$  bits, como se muestra en la Figura 21. Los bits de cola son necesarios para devolver el codificador al estado cero.

**Figura 21**

*Extensión del bloque de datos con bits de cola antes de la codificación*



Fuente: (IEEE Comsoc, 2020)

El valor de los bits de cola depende del esquema de codificación y son determinados como se muestra en la Tabla 6.

**Tabla 6**

*Patrón de bits de cola para los codificadores RSC y NRNSC*

<b>Memory State</b> <b>(<math>M_0 - M_2</math>)</b>	<b>Tail bits</b>	
	<b>RSC (b0 b1 b2)</b>	<b>NRNSC (b0 b1 b2)</b>
<b>000</b>	000	000
<b>001</b>	100	000
<b>010</b>	110	000
<b>011</b>	010	000
<b>100</b>	111	000
<b>101</b>	011	000
<b>110</b>	001	000
<b>111</b>	101	000

Fuente: (IEEE Comsoc, 2020)

Con respecto al cambio de Canal, al establecer una comunicación se basa en un tiempo específico, el cual debe ser menor o igual a 500  $\mu$ s. El tiempo de cambio de canal se define como el tiempo transcurrido al cambiar a un nuevo canal, incluido el tiempo de estabilización.

Por otro lado, la tolerancia de la velocidad de símbolo del transmisor es un factor importante el cual será menor o igual a  $\pm 300 \times 10^{-6}$ . La fluctuación máxima de la velocidad de símbolo del transmisor será menor o igual a  $\pm 40 \times 10^{-6}$ . Los paquetes transmitidos deberán tener velocidades de símbolo acordes a la tolerancia de velocidad de símbolo especificada, y todos los símbolos dentro del paquete deberán estar dentro de la tolerancia de velocidad de símbolo en relación con la velocidad de símbolo promedio de todos los símbolos del paquete.

- Rechazo de interferencias del receptor

El rechazo de canal adyacente se medirá de la siguiente manera: la señal deseada será una señal SUN FSK PHY compatible de datos pseudoaleatorios en la frecuencia central del canal deseado. La señal deseada se introduce en el receptor a un nivel 3 dB por encima de la sensibilidad del receptor.

En el canal adyacente o en el alternativo, se introduce una portadora no modulada en el centro de ese canal en el siguiente nivel referente al nivel de la señal deseada:

- El rechazo del canal adyacente será superior o igual a 10 dB.
- El rechazo del canal alternativo será mayor o igual a 30 dB.

Un transmisor deberá poder transmitir a un nivel de potencia de al menos  $-3$  dBm. Se considera necesario que cualquier dispositivo transmita menor potencia cuando sea posible para reducir la interferencia con otros equipos y sistemas.

El SUN FSK PHY deberá tener un nivel de entrada máximo del receptor superior o igual a  $-20$  dBm.

#### *d. Modulación SUN OFDM PHY*

El SUN OFDM PHY trabaja con rangos de velocidades de datos desde  $12,5$  kb / s a  $2400$  kb / s. El espaciado de las subportadoras es constante y es igual a  $10416 \cdot 2 / 3$  Hz.

La velocidad de símbolo es de  $8 \cdot 1 / 3$  ksymbol / s, que corresponde a  $120$   $\mu$ s por símbolo.

El ancho de banda total de la señal para cada opción varía desde  $1,2$  MHz hasta menos de  $200$  kHz.

En la modulación SUN OFDM PHY el formato del **PPDU** es el mostrado en la Figura 22:

## Figura 22

*Formato del SUN OFDM PPDU*

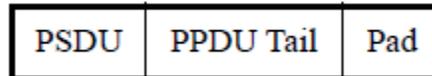


Fuente: (IEEE Comsoc, 2020)

El campo de carga útil PHY tendrá en siguiente formato:

## Figura 23

*Formato del campo de carga útil PHY*



Fuente: (IEEE Comsoc, 2020)

El STF hace uso de un número menor de tonos que el campo de carga útil PHY. Por lo que, se requiere la normalización del STF en el dominio de la frecuencia para garantizar que la potencia del STF sea la misma que la del resto del paquete.

Ahora se analizará el campo PHR cuyo formato se lo muestra en la Figura 24.

## Figura 24

*Campos de encabezado PHY para SUN OFDM*

Bits: 0–4	5	6–16	17–18	19–20	21	22–29	30–35
Rate	Reserved	Frame Length	Reserved	Scrambler	Reserved	HCS	Tail

Fuente: (IEEE Comsoc, 2020)

El PHR ocupa símbolos para cada opción OFDM y se transmitirá utilizando el nivel MCS admitido más bajo, como se indica en la Tabla 7, para la opción que se utiliza, excepto para la

opción OFDM 3 y la opción OFDM 4, el PHR se transmitirá utilizando MCS1 y MCS2 respectivamente. Se envía al codificador convolucional empezando desde el bit más a la izquierda hasta el bit más a la derecha.

**Tabla 7**

*Símbolos PHYR para las opciones SUN OFDM PHY*

Opción OFDM	Nivel MSC	phyOfdmInterleaving	PHR Symbols
<b>1</b>	0-6	0	3
<b>2</b>	0-6	0	6
<b>3</b>	1-6	0	6
	0	0	12
<b>4</b>	2-6	0	6
	0-1	0	24
<b>1</b>	0-6	1	4
<b>2</b>	0-6	1	8
<b>3</b>	1-6	1	6
	0	1	12
<b>4</b>	2-6	1	6
	0-1	1	24

Fuente: (IEEE Comsoc, 2020)

El campo Longitud de trama es un entero sin signo que se establecerá al número total de octetos contenidos en la PSDU (antes de la codificación FEC). Por otro lado, el campo Codificador, especifica la semilla de codificación. El campo de secuencia de verificación de encabezado (HCS) es un CRC de 8 bits tomado de los campos PHR.

El HCS se calculará utilizando los primeros 22 bits del PHR.

Por último, en campo Cola de la Figura 23, consta de todos ceros, que está destinado a la descarga del decodificador Viterbi.

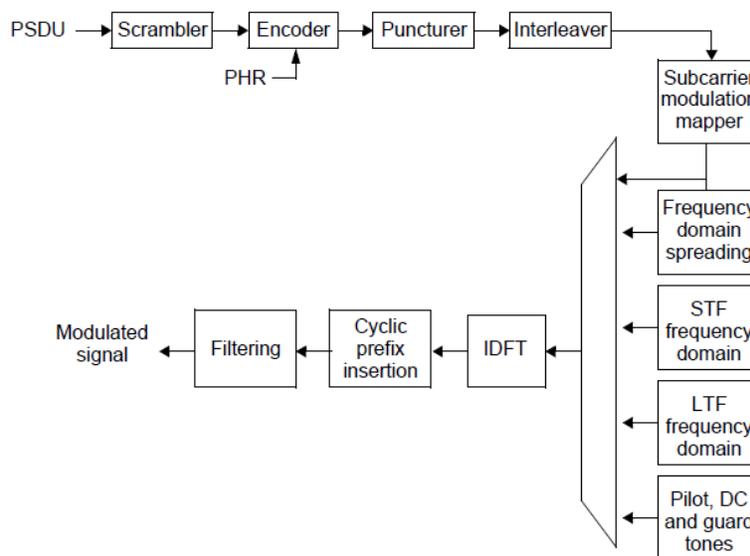
El campo **PSDU** de la Figura 23 contiene la PSDU codificada.

- **Diagrama del modulador SUN OFDM PHY de referencia**

La Figura 25, muestra el modelo de referencia del codificador para la modulación SUN OFDM PHY.

**Figura 25**

*Diagrama del modulador de referencia para SUN OFDM*



Fuente: (IEEE Comsoc, 2020)

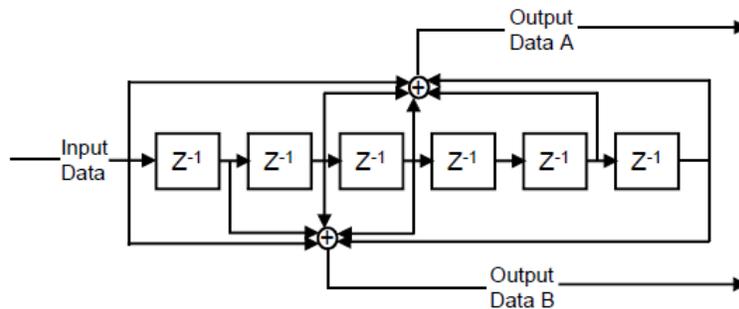
- **FEC**

El campo de carga útil PHY se codificará con un codificador convolucional de velocidad de codificación  $R = 1/2$  o  $3/4$ , correspondiente a la velocidad de datos deseada.

El codificador convolucional se inicializará al estado todos ceros antes de codificar el PHR y luego se reiniciará al estado todos ceros antes de codificar la PSDU.

**Figura 26**

*Codificador convolucional de frecuencia*



Fuente: (IEEE Comsoc, 2020)

- **Rango de frecuencia de funcionamiento.**

El SUN OFDM PHY opera en las siguientes bandas: 470–510 MHz, 779–787 MHz, 863–870 MHz, 865–867 MHz, 866–869 MHz, 870–876 MHz, 902–928 MHz, 902–928 (alternativo) MHz, 902–907,5 MHz y 915–928 MHz, 915–928 MHz, 915–921 MHz, 915–918 MHz, 917–923,5 MHz, 919–923 MHz, 920–928 MHz, 920,5–924,5 MHz, 920–925 MHz, 2400–2483,5 MHz.

### **2.1.6. Descripción funcional MAC**

#### **2.1.6.1. Formato General de la Trama MAC.**

El formato de la trama MAC general, se encuentra ilustrada en la Figura 27. Los campos del MHR aparecen en un orden fijo, y es posible que algunos campos no se incluyan en todas las tramas.

## Figura 27

### Formato General de trama MAC

	Octets: 1/2	0/1	0/2	0/2/8	0/2	0/2/8	variable	variable	variable	2/4
Frame Control	Sequence Number	Destination PAN ID	Destination Address	Source PAN ID	Source Address	Auxiliary Security Header	IE		Frame Payload	FCS
							Addressing fields			
							Header IEs	Payload IEs		
MHR								MAC Payload		MFR

Fuente:(IEEE Comsoc, 2020)

#### a. Campo Control de Tramas

El campo de control de tramas se ilustra en la Figura 28.

## Figura 28

### Formato de la Trama Control

Bits: 0-2	3	4	5	6	7	8	9	10-11	12-13	14-15
Frame Type	Security Enabled	Frame Pending	AR	PAN ID Compression	Reserved	Sequence Number Suppression	IE Present	Destination Addressing Mode	Frame Version	Source Addressing Mode

Fuente: (IEEE Comsoc, 2020)

El campo **Tipo de trama** se configurará como se define en la Tabla 8. Su tamaño corresponde a 3 bytes de longitud.

**Tabla 8***Descripción Tipos de Trama*

<b>Frame Type value b2 b1 b0</b>	<b>Descripción</b>
<b>000</b>	Beacon
<b>001</b>	Data
<b>010</b>	Acknowledgment
<b>011</b>	MAC command
<b>100</b>	Reserved
<b>101</b>	Multipurpose
<b>110</b>	Fragmento o Frank
<b>111</b>	Extended

Fuente: (IEEE Comsoc, 2020)

El **campo Security Enable** se setea en uno si la trama está resguardada por la subcapa MAC y se establecerá a cero en caso contrario. El campo de encabezado de seguridad auxiliar del MHR estará presente solo si el campo Security Enable se establece en uno. Su tamaño es de 1 byte.

El **campo Frame Pending field** se establecerá en uno si el dispositivo que envía la trama tiene más datos para el destinatario y se colocará a cero en caso contrario. Su tamaño es de 1 byte.

Por otro lado, el campo **AR** indica si se requiere de un acuse de recibo del dispositivo receptor al recibir una trama de datos o un comando MAC. Si este campo se establece en uno, el dispositivo receptor enviará una trama Ack, de lo contrario si se setea en cero no se enviara Ack.

El campo de **compresión PAN ID** tendrá distintas consideraciones, cuando el valor del campo de versión de trama es 0b00 o 0b01, el campo de compresión de ID de PAN se interpretará:

- Si la información de direccionamiento de origen y destino está presente, la subcapa MAC comparará los ID PAN de origen y destino, si son idénticos, el campo de compresión de ID de PAN se establecerá en uno y el campo ID de PAN de origen se omitirá de la trama transmitida. Por el contrario, si son diferentes, el campo de compresión de ID de PAN se establecerá en cero, y el campo de ID de PAN de origen y destino se incluirán en la trama transmitida.

- Si únicamente está presente la información de direccionamiento de origen o de destino, el campo de compresión de ID PAN se pondrá en cero y el campo ID PAN de la dirección única se incluirá en la trama transmitida. Su tamaño es de 1 byte.

El campo **Número de Secuencia de Supresión** al establecerse en uno, indica la supresión del campo Número de secuencia en la trama y se omitirá el número de secuencia, pero cuando se establece en cero, el campo Número de secuencia está presente. Si el campo Versión de trama es 0b00 o 0b01, el campo Supresión del número de secuencia será cero.

El campo **IE Present** se establece en uno si la trama contiene IE. Este campo se pondrá a cero en caso contrario. Si el campo Versión de trama es 0b00 o 0b01, el campo IE Present será cero.

El campo de **modo de direccionamiento de destino** tomará uno de los valores enumerados en la Tabla 9.

En caso de que este campo sea igual a cero y el campo Tipo de trama especifica una trama de datos o comando MAC y el campo Versión de trama se establece en 0b00 o 0b01, el campo Modo de direccionamiento de origen será distinto de cero, lo que quiere decir que la trama se dirige al coordinador PAN con su respectivo ID de PAN como se indica en el campo ID de PAN de origen. Su tamaño es de 2 bytes.

**Tabla 9**

*Valores válidos del modo de direccionamiento de destino y Campos del modo de direccionamiento de origen*

<b>Valor del modo de direccionamiento b1 b0</b>	<b>Descripción</b>
<b>00</b>	Los campos de ID y dirección de PAN no están presentes.
<b>01</b>	Reservado
<b>10</b>	El campo de dirección contiene una dirección corta (16 bits).
<b>11</b>	El campo de dirección contiene una dirección extendida (64 bits).

El campo **Versión de trama** es un entero sin signo que especifica el número de versión correspondiente a la trama. Su tamaño es de 2 bytes.

El campo **Modo de direccionamiento de origen** se establecerá en uno de los valores enumerados. Su tamaño es de 2 bytes.

Si este campo es igual a cero y el campo Tipo de trama especifica una trama de datos o comando MAC, y el campo Versión de trama se establece en 0b00 o 0b01, el campo Modo de direccionamiento de destino será distinto de cero, lo que implica que la trama se originó en el PAN coordinador con el ID de PAN como se especifica en el campo ID de PAN de destino.

En la trama de la Figura 28, se tiene además los siguientes campos:

*b. Campo Número de secuencia.*

Este campo especifica el identificador de secuencia para la trama.

*c. Campo ID de PAN de destino.*

Es un número entero sin signo que especifica el ID de PAN único del destinatario conocido de la trama.

Un valor de la ID de PAN de transmisión, en el campo ID de PAN de destino será aceptado como una ID de PAN válida por todos los dispositivos que escuchan actualmente el canal.

*d. Campo Dirección de destino.*

Este campo con una longitud especificada en el campo Modo de direccionamiento de destino del campo Control de trama, indica la dirección del destinatario previsto de la trama.

Además, este campo se incluirá en la trama MAC solo si el campo Modo de direccionamiento de destino es distinto de cero.

*e. Campo ID PAN de origen.*

Cuando está presente, este campo especifica el ID PAN único del originador de la trama. EL ID de PAN de un dispositivo se determina inicialmente durante la asociación en un PAN, pero puede cambiar después de una resolución de conflicto de ID de PAN.

*f. Campo Dirección de origen.*

Especifica la dirección del inventor de la trama. Este campo se incluirá en la trama MAC solo si el campo Modo de direccionamiento de origen es distinto de cero.

*g. Campo Encabezado de seguridad auxiliar.*

Indica la información necesaria para el procesamiento de seguridad. Este campo estará presente solo si el campo Seguridad habilitada se establece en uno.

*h. El campo IE*

Es de longitud variable y contiene uno o más IE. Este campo se compone de los subcampos de IE de encabezado y de IE de carga útil. Este campo estará presente sólo si el campo IE Present en el campo Frame Control está en uno. Posee un formato indicado en la Figura 29.

Los IE de encabezado, si están presentes, siguen el encabezado de seguridad auxiliar y son parte del MHR. En cambio, los IE de carga útil, si están presentes, siguen el MHR y se consideran parte de la carga útil de MAC, es decir, pueden estar cifrados.

## Figura 29

### Formato del campo IE

Octets: variable	...	variable	0/2	variable	...	variable	0/2
Header IE #1	...	Header IE #n	Header Termination IE	Payload IE #1	...	Payload IE #m	Payload Termination IE

#### i. Campo Carga útil de tramas

Contiene información específica para tipos de tramas individuales. Si el campo Seguridad habilitada se establece en uno, la carga útil de la trama puede estar protegida criptográficamente. La longitud es variable, sin embargo, el tamaño de la trama MAC completa no debe exceder los 127 bytes de información. Los datos que lleva la carga útil dependen del tipo de trama. El estándar IEEE 802.15.4 tiene cuatro diferentes tipos de tramas: de guía, de datos, tramas de enterados y tramas de comandos MAC. Solo las tramas de datos y de guía contienen información de capas superiores; las tramas de mensajes de enterado y la de comandos MAC originados en el MAC se usan para comunicaciones MAC peer-to-peer (Archundia, 2007).

#### j. Campo FCS.

El campo FCS se calcula sobre las partes de carga útil MHR y MAC de la trama; estas partes juntas se conocen como el campo de cálculo.

Los dispositivos que cumplen con MSK PHY que implementan la velocidad de datos de 2000 kb / s implementarán FCS de 2 y 4 octetos. Si el campo PHR extendido se pone a uno, se utilizará el FCS de 4 octetos; si el campo PHR ampliado se pone a cero, se utilizará el FCS de 2 octetos.

### 2.1.6.2. Medición de Recursos del Espectro

#### a. *Energy Detection (ED)*

Es una estimación de la potencia de la señal recibida dentro del ancho de banda del canal y se utiliza para detectar tipo de transmisiones de RF. El valor ED mínimo (cero) mostrará la potencia recibida inferior a 10 dB por encima de la sensibilidad del receptor más baja especificada, en dBm, para la PHY.

Las tecnologías PHY cuentan con capacidad ED (Receptor ED). La ED medida en dBm se puede calcular de la siguiente manera:

$$\text{macEd} = \text{ED medida [dBm]} - (\text{la sensibilidad del receptor más baja [dBm]} + 10).$$

#### b. *Indicador de potencia del canal recibido (RCPI)*

El indicador RCPI, es una medida de la potencia de RF recibida en el canal seleccionado para una trama recibida. El valor de RCPI será equivalente a la potencia de RF recibida promediada durante toda la trama recibida. El rango válido de valores es de 0 a 150 y se redondeará al valor entero más próximo en unidades de dBm.

- 0: Power  $\leq -150$  dBm
- 1: Power =  $-149$  dBm
- 2: Power =  $-148$  dBm

Y así hasta

- 150: Power  $\geq -0$  dBm

- 151–254: Reservado
- 255: Medida no disponible

Donde:

$RCPI = \text{Integer} [\text{Power in dBm} + 150]$  for  $0 \text{ dBm} > \text{Power} > -150 \text{ dBm}$

*c. Indicador de ruido de señal recibida (RSNI)*

RSNI es una indicación de la relación señal / ruido más la interferencia de una trama recibida, que se define por la relación entre la potencia de la señal recibida (RCPI) y la potencia de ruido más interferencia (ANPI) medida en el canal y en el conector de la antena utilizado para recibir la trama.

*d. Indicador de intensidad de señal recibida (RSSI)*

El indicador de intensidad de señal recibida está diseñado para que lo utilice en un entorno de red inalámbrica múltiple, con el fin de evaluar la utilización del recurso del espectro junto con RCPI y RSNI. RSSI es una medida de la potencia de RF en dBm para el paquete recibido. Posee distintos valores donde los valores mínimo y máximo son 0 (–174 dBm) y 254 (80 dBm), respectivamente. 255 está reservado. Si algún valor medido es inferior a –174 dBm, el valor notificado se redondeará a –174 dBm.

*e. Evaluación de Canal Libre (CCA)*

Al hacer uso del mecanismo de acceso al medio CSMA-CA, el MAC solicita a la PHY que realice una evaluación de Canal Despejado CCA para asegurar que otro dispositivo no está haciendo del uso del canal. Si el CCA indica que otro nodo está transmitiendo actualmente, la capa MAC se abstiene de enviar su propio paquete. En cambio, la capa MAC espera un tiempo especificado y luego vuelve a intentar enviar el paquete.

Existen tres modos CCA, y una PHY compatible con IEEE 802.15.4 debe poder funcionar en cualquiera de ellos:

*Modo CCA 1.* En este modo, solo se tiene en cuenta el resultado de ED. Si el nivel de energía está por encima del umbral ED, el canal se considera ocupado. El nivel de umbral de ED puede ser establecido por el fabricante.

*Modo CCA 2.* El modo 2 usa solo el resultado CS, y el canal se considera ocupado solo si la señal de ocupación cumple con la PHY del dispositivo que está realizando la CCA.

*Modo CCA 3.* Este modo es una combinación lógica (Y / O) del modo 1 y el modo 2. En otras palabras, en el modo 3, la PHY puede utilizar uno de los siguientes como indicación de un canal ocupado:

El nivel de energía detectado está por encima del umbral y se detecta una portadora compatible.

El nivel de energía detectado está por encima del umbral o se detecta una portadora compatible (Farahani, 2008).

*f. Utilización del canal*

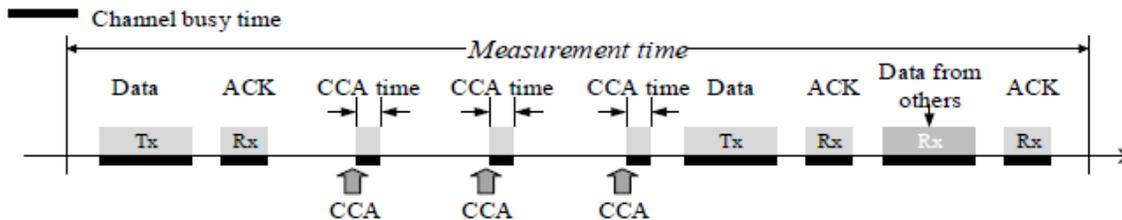
El tiempo total de canal utilizado para Tx y Rx durante el tiempo de medición se escala linealmente con 0xff que representa el 100% de utilización del canal. El tiempo de ocupación del canal se define de la siguiente manera:

- El tiempo total de Tx y Rx de todas las tramas, incluidas las balizas y los ACK para y desde este dispositivo.

- También se agrega el tiempo de CCA durante el período diferido. La precisión para el tiempo utilizado será de  $\pm 100 \mu\text{s}$ .

**Figura 30**

*Utilización del Canal*



Fuente: (IEEE Comsoc, 2020)

**2.1.6.3. Métodos de acceso al medio.**

**a. Algoritmo CSMA-CA**

El algoritmo CSMA-CA se utilizará antes de la transmisión de datos, a menos que la trama se pueda transmitir rápidamente después del acuse de recibo de un comando de solicitud de datos. El algoritmo se implementa utilizando unidades de tiempo denominadas períodos de

retroceso que es una cantidad de tiempo aleatorio. En CSMA-CA sin ranuras, cada dispositivo posee variables para cada intento de transmisión: NB, CW, BE, aMaxBE y macMaxCSMABackoffs.

NB corresponde al número de veces que se solicitó que el algoritmo CSMA-CA retrocediera al intentar la transmisión actual; este valor se inicializará a cero antes de cada nuevo intento de transmisión. CW es la longitud de la ventana de contención, que define el número de períodos de interrupción que deben estar libres de la actividad del canal antes de que pueda comenzar la transmisión. La variable CW solo se usa para CSMA-CA ranurado la cual no sería necesario para este caso. BE es el exponente de retroceso, en otras palabras, es el periodo de retroceso que un dispositivo debe esperar antes de intentar evaluar un canal. BE se inicializará con el valor de macMinBe. Considerar si macMinBe se establece en cero, la prevención de colisiones se desactivará durante la primera insistencia de este algoritmo. aMaxBE es el máximo valor permitido para la variable BE, especifica el máximo tiempo de *backoff* que el nodo espera antes de evaluar el canal. macMaxCSMABackoffs es el máximo valor permitido para el valor de la variable NB, es decir, es el máximo número de intentos de transmisión.

Aunque el receptor del dispositivo está habilitado durante la parte de análisis CCA de este algoritmo, el dispositivo puede descartar cualquier trama recibida durante este tiempo. Si el algoritmo termina en "Success", se permite que el MAC comience la transmisión de la trama, de lo contrario, el algoritmo termina con una negación o falla en el proceso de acceder al canal. Para una mejor comprensión del algoritmo se hace uso de valores establecidos en el estándar IEEE 802.15.4g.

Se inicia determinando el tiempo de retroceso o backoff que se tiene que esperar antes de comprobar si el canal está ocupado o no. Este tiempo es igual al número de slots (medidas de

tiempo), y se lo calcula como un número entero aleatorio en el intervalo  $(0, 2^{BE} - 1)$ , para el valor de BE se le asigna el valor de establecido en el estándar IEEE 802.15.4g que corresponde a  $macMinBe$  con un valor igual a 3, y por otro lado se tiene el valor de las variables  $aMaxBE$  y  $macMaxCSMABackoffs$  con los valores de 5 y 4 respectivamente, entonces se tiene el rango de retroceso  $(0, 7)$ , además se tiene el valor NB establecido en 0 inicialmente.

El tiempo de slot es equivalente a la variable  $aUnitBackoffPeriod$  con un valor de 20 símbolos. Para una frecuencia de 2.4GHz la tasa de símbolo es igual a 62,5ksym/s, entonces se tiene la Ec. 2.

$$Tiempo\ de\ Slot = aUnitBackoffPeriod = \frac{1}{tasa\ de\ simbolo} \quad Ec. 2$$

y con los valores reemplazados:

$$Tiempo\ de\ Slot = 20sym * \frac{1}{62,5\ ksym/s} = 320\mu s$$

Como segundo paso se realiza la función CCA, considerando el nivel de energía, si se tiene un valor de umbral predeterminado, por ejemplo -44dBm, y que el valor que se estime sea -40 dBm entonces se considera que está por encima del umbral lo que ocasionaría un bloqueo en el intento de transmisión, por el contrario, si se tiene un valor de -60 dBm al estar por debajo del umbral la transmisión no será bloqueada.

Si se tiene el primer caso en que la transmisión se bloquea debido a que el canal está ocupado las variables BE y NB se incrementarán en 1 como se indica en la Ecuación 3.

$$BE = BE + 1; NB = NB + 1 \quad Ec. 3$$

$$BE = \min (BE + 1, macMaxBE) \quad Ec. 4$$

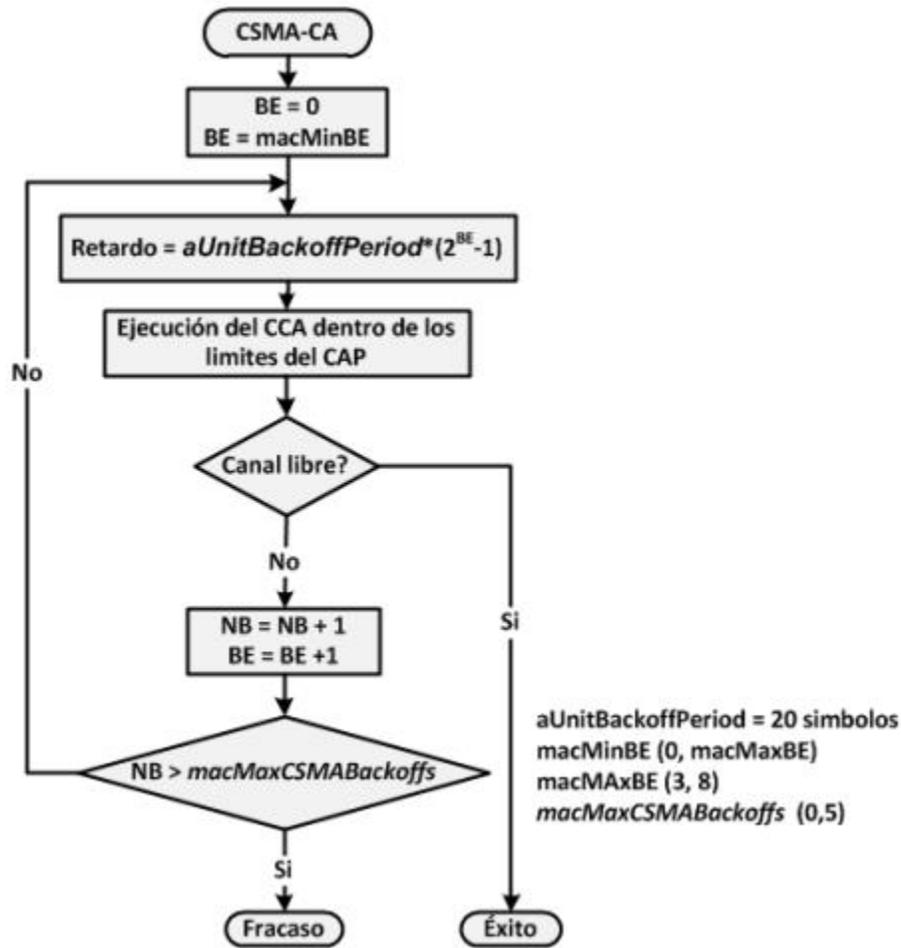
En la Ecuación 4,  $BE = \min(4,5)$  entonces al realizar la comparación con la condición  $BE \leq aMaxBE$ , donde BE tiene el valor 4 para el tiempo de backoff para evaluar el canal el cual es menor a 5, por lo que se procede a verificar el valor de NB:

$$NB > MacMaxCSMABackoffs$$

Y para el valor de  $NB = 1$  por lo que es menor al valor de MacMaxCSMABackoffs igual a 4, entonces en este caso se volverá a calcular el tiempo de backoff con el nuevo valor de BE y el proceso se vuelve a reiniciar hasta que el número de intentos fallidos sea menor al número de intentos permitidos.

### **Figura 31**

*Algoritmo CSMA-CA*



Fuente: (Cali Francisco, 2018)

### 2.1.7. Subcapa de Control de Enlace Lógico (LLC)

El control de enlace lógico (LLC) coloca información en la trama que identifica qué protocolo de capa de red está siendo utilizado por la trama. Esta información permite que varios protocolos de la Capa 3, tales como IP e IPX, utilicen la misma interfaz de red y los mismos medios, para el caso de estudio se hace uso del estándar IEEE 802.15.4 siendo su tamaño de 8 bytes.

### **2.1.8. Protocolo de enrutamiento para baja potencia RPL**

RPL es un protocolo de enrutamiento de vector de distancia IPv6 para LLNs (Low-Power and Lossy Networks), que trabaja en la capa de red e indica cómo construir un gráfico acíclico orientado a destino (DODAG Destination Oriented Directed Acyclic Graph) utilizando una función objetivo y un conjunto de métricas / restricciones. Los DODAGS poseen un nodo raíz que es accesible para cualquier nodo y no pueden tener un ciclo en su estructura. La función objetivo opera en una combinación de métricas y restricciones para calcular la mejor ruta. El gráfico construido por RPL es una topología de enrutamiento lógico construida sobre una red física, para cumplir un criterio específico y el administrador de red puede decidir tener múltiples topologías de enrutamiento (gráficos) activas al mismo tiempo, utilizadas para transportar tráfico con diferentes requisitos.

RPL puede ser usado con los sistemas operativos de redes de sensores inalámbricos más difundidos, TinyOS y Contiki OS, con los nombres de TinyRPL y ContikiRPL, respectivamente (Murillo, 2017).

#### **2.1.8.1. Formación de un GRAFO RPL.**

Para la formación del grafo RPL se sigue los siguientes pasos:

- 1) Se escoge los nodos que se configurarán como DODAG roots y poseerán la configuración del DODAG.
- 2) Los nodos deberán anunciar su configuración, su afiliación con un DODAG, y su coste de encaminamiento a través de la difusión de mensajes DIO (DODAG Information Object).

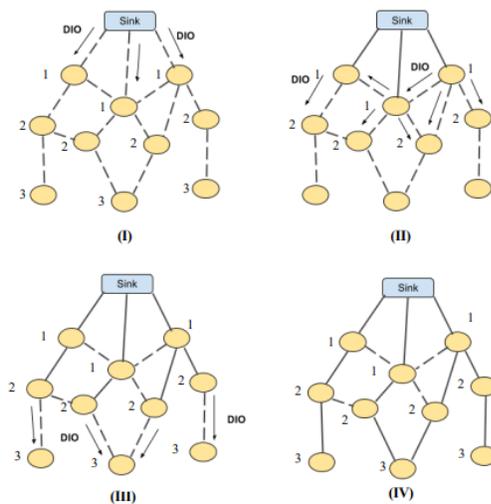
El mensaje DIO poseerá información tal como: el identificador único para la instancia, el DODAGID, versión de DODAG, y la posición del nodo, además este parámetro se

calcula mediante el OF (Función Objetivo ) que permite conocer cuál es el próximo nodo que conviene para llegar al nodo root.

- 3) Los demás nodos estarán a la escucha de estos mensajes DIOs y los usarán para unirse al nuevo DODAG e iniciar el proceso de selección para elegir al mejor nodo padre o seguir perteneciendo al DODAG anteriormente elegido. Cuando un nodo de la red reciba el mensaje DIO, éste mantendrá a un conjunto de nodos vecinos candidatos eligiendo aquellos que tengan un valor de posición igual o inferior al suyo propio, para que se conviertan en “nodos padres”.
- 4) Cada entrada de la tabla de encaminamiento de los nodos registrará los destinos especificados según el mensaje DIO, así como el DODAG padre del DODAG versión. Los nodos que decidan unirse a este DODAG deben elegir uno o más DODAG padres como siguientes saltos para la ruta por defecto de esa instancia. Una vez calculado la posición de un nodo, éste propagará el mensaje DIO actualizado al resto de sus vecinos. Todo el proceso se puede apreciar en la Figura 32.

### **Figura 32**

*Creación del grafo DODAG*



Fuente: (Murillo, 2017)

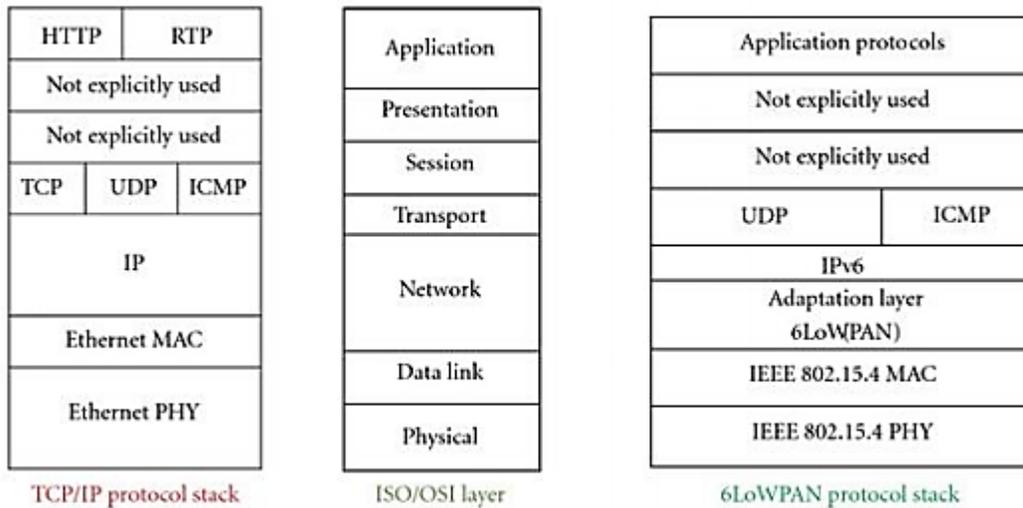
### 2.1.9. Protocolo 6LOWPAN

IEEE 802.15.4 permite el uso de direcciones extendidas IEEE de 64 bits o direcciones de 16 bits únicas dentro del PAN, dichas direcciones son distribuidas por la función de coordinador del PAN en el proceso de asociación. Sin embargo, es posible que las direcciones extendidas IEEE de 64 bits presenten inconvenientes de escalabilidad referentes al enrutamiento, el descubrimiento, la configuración, entre otros.

6LoWPAN utiliza un modelo de capas similar a OSI, pero con algunas modificaciones. En la Figura 33, se indica una comparación entre los tres modelos de pila: TCP/IP, OSI y la pila de 6LoWPAN, con lo que se puede apreciar la similitud entre el modelo OSI y el modelo 6LoWPAN, con la diferencia de que este último cuenta con una capa adicional llamada adaptación, y que hay dos capas, las capas de sesión y presentación en el modelo OSI, que no se utilizan en 6LoWPAN. (Murillo, 2017)

**Figura 33**

## Modelos de Capas



Fuente: (Murillo, 2017)

### 2.1.9.1. Capa de Adaptación de 6LoWPAN

El objetivo principal de 6LoWPAN es permitir la transmisión de información entre dispositivos IP, sin que sea necesario traducir los protocolos que se utilizan desde una red origen a una red destino. Es así como el estándar 6LoWPAN ingresa dentro de su pila de protocolos una capa de adaptación la cual se ubica entre la capa de red y la capa de enlace de datos para poder permitir la transmisión de los datagramas IPv6 sobre los enlaces de radio IEEE 802.15.4 (Murillo, 2017).

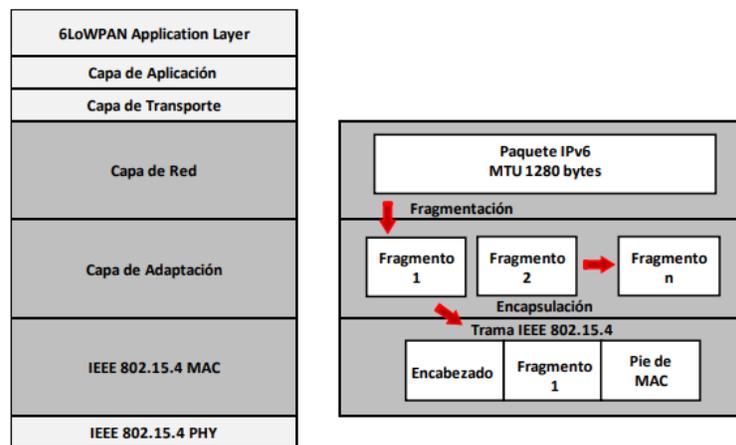
#### a. Tareas de la Capa de Adaptación

**Fragmentación y reensamblado:** El estándar de IPv6 establece como longitud mínima MTU 1280 bytes, sin embargo, es difícil transportar dicho tamaño de trama dentro la MSDU de IEEE 802.15.4, ya que solo se disponen 127 bytes por lo que conviene dividir en varios segmentos el paquete IPv6 que se quiere transmitir sobre IEEE 802.15.4. Consecuentemente, la capa de adaptación debe encargarse de fragmentar y reensamblar el paquete según sea el caso, para luego

ser encapsulado dentro de la MSDU dentro del formato MPDU indicado en la Figura 34. En caso de que sea necesaria la fragmentación se agregará un encabezado con tamaño de 5 bytes en cada segmento y con ello poder realizar una reconstrucción exacta del paquete original en el destino (Murillo, 2017).

### Figura 34

*Proceso de Fragmentación de un paquete IPV6 sobre IEEE 802.15.4*



Fuente: (Murillo, 2017)

**Compresión de encabezado:** Consiste en reducir la sobrecarga que aporta al paquete que se está transmitiendo mediante la compresión el encabezado de IPv6. Una vez que el paquete ha sido fragmentado y transmitido sobre tramas IEEE 802.15.4 se tendrá fragmentos que contienen parte del paquete original de IPv6. Además, es importante mencionar que, sin la compresión, sería imposible transmitir el paquete con efectividad (Murillo, 2017). Considerando que la trama IEEE 802.15.4 no debe exceder los 127 bytes se propone el uso de mecanismos de compresión para paquetes IPV6, ya que es evidente que IPv6 no está diseñado para trabajar en redes IEEE 802.15.4. En el mejor de los casos (sin seguridad a nivel de enlace y usando UDP como protocolo

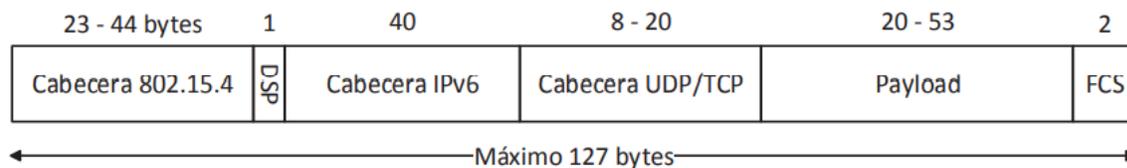
de transporte) solo el 41.7 % de los bytes transmitidos son información de aplicación.

Fundamentalmente, se basan en la compresión de la cabecera de red IPv6 y la de transporte UDP.

En la Figura 35, se muestra una trama IEEE 802.15.4 sin compresión (Vadillo Gutiérrez, 2014).

### Figura 35

*Encapsulación sin compresión de una trama IEEE 802.15.4*



Fuente: (Vadillo Gutiérrez, 2014)

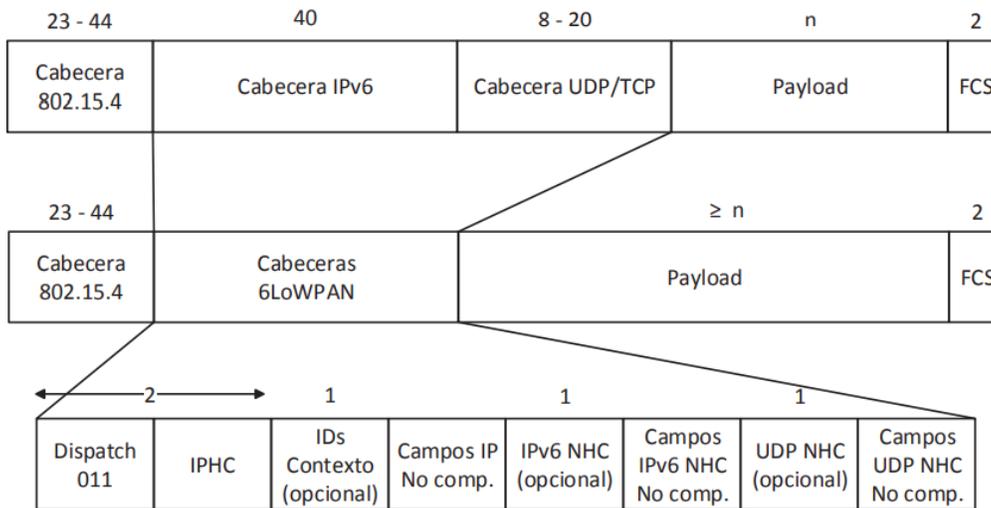
Existen distintos mecanismos de compresión de cabeceras, uno de ellos es la compresión basada en contexto: IPHC-NHC.

Definido en el RFC 6282 el cual reemplaza al HC1-HC2 (mecanismo obsoleto) incluido en el RFC 4944. Aunque ambos pueden funcionar de forma paralela, el estándar recomienda que todas las implementaciones nuevas usen el formato de compresión IPHC-NHC para la compresión y transmisión de paquetes y solo el HC1-HC2 para la descompresión, favoreciendo la interoperabilidad y el traspaso hacia el nuevo mecanismo.

Para implementar el proceso de compresión, los últimos 5 bits del campo del dispatch byte se aprovechan para la propia cabecera IPHC. De igual manera, a la suma de distintas cabeceras le continúan los campos no comprimidos de la misma, a diferencia de HC1-HC2 que juntaba todas las cabeceras y a continuación todos los campos no comprimidos como se indica en la Figura 36.

### Figura 36

#### Comprensión de cabeceras IPHC-NHC



Fuente: (Vadillo Gutiérrez, 2014)

La cabecera IPHC es la que contiene la información del proceso de compresión de la cabecera IPv6. El formato de la cabecera se muestra en la Figura 37.

### Figura 37

#### Formato de cabecera IPHC

0	1	2	3	4	5	6	7
0	1	1	TF		NH	HLIM	
CID	SAC	SAM		M	DAC	DAM	

Fuente: (Vadillo Gutiérrez, 2014)

TF (Traffic Class, Flow Label): Indican si se suprime el campo Clase de tráfico y/o Etiqueta de flujo de la cabecera IPv6.

NH (Next Header ): Indica si se suprime el campo NH de la cabecera IPv6 y usar NHC.

HLIM (Hop Limit): Indica si el campo Hops de la cabecera IPv6 se comprime además de limitar el contador a un valor máximo de 1, 64 o 255.

CID (Context Identifier Extension): Informa sobre el uso opcional del campo de extensión de identificador de contexto.

SAC (Source Address Compression): Indica si la dirección IPv6 de la fuente va a usar compresión stateful (basada en contexto) o compresión stateless.

SAM (Source Address Mode): En función del bit comentado anteriormente, se tienen 4 modos de compresión para cada tipo (stateless o stateful). Las direcciones IPv6 pueden comprimirse a 64 bits, 16 bits, comprimirse completamente o no comprimirse.

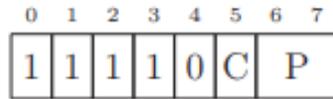
M (Multicast Compression): Indica si la dirección IPv6 destino es multicast. DAC (Destination Address Compression): Similar al SAC, pero para el destino.

DAM (Destination Address Mode): Similar al SAM, pero para el destino. Si el bit M está activado se sigue una compresión idónea (48, 32 u 8 bits) para direcciones multicast (Vadillo Gutiérrez, 2014).

Por último, la cabecera UDP NHC permite la compresión de la cabecera UDP. El formato de la cabecera se muestra en la Figura 38.

### **Figura 38**

*Formato de cabecera UDP NHC*



Fuente: (Vadillo Gutiérrez, 2014)

C (Checksum): Indica si el campo de Checksum de la cabecera UDP se comprime.

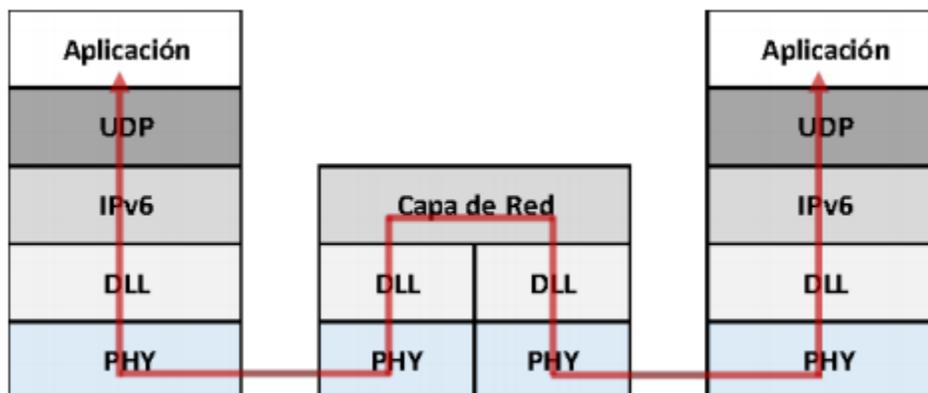
P (Ports): Indica si los puertos origen/destino se comprimen de 16 bits a 4 bits, de forma idéntica a como se hacía en la compresión HC2. A diferencia con la compresión HC2, el campo longitud debe ser siempre comprimido.

**Enrutamiento:** Por lo general la capa de red es en donde se realiza el enrutamiento IP, independientemente de las capas inferiores.

Una ventaja que posee 6LoWPAN es que, debido a su capa de adaptación, puede soportar el enrutamiento en la capa de enlace y en la capa de red, “mesh-under” y “routeover”, respectivamente. En la Figura 39, se muestra el modelo de enrutamiento IP en la capa de red.

**Figura 39**

*Modelo de Enrutamiento IP*

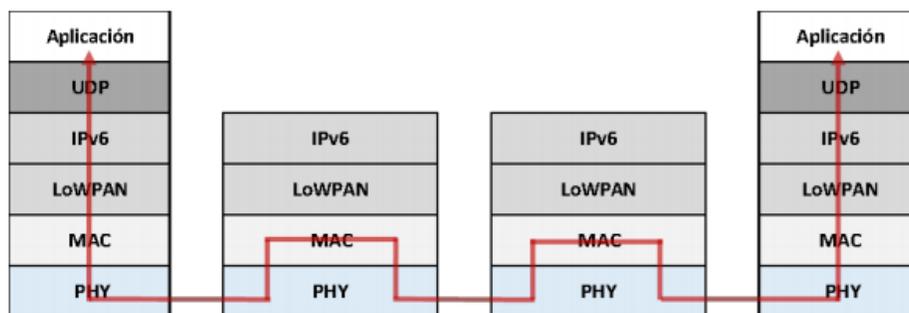


Fuente: (Vadillo Gutiérrez, 2014)

En la Figura 40, se puede apreciar el tipo de enrutamiento mesh-under el cual radica en que la capa de adaptación direcciona los paquetes al destino en múltiples saltos radio (multi-hop). Para lograr el direccionamiento multi-hop, la organización IETF 6LoWPAN ha definido un encabezado “mesh-header”, que codifica el límite de saltos, así como la fuente y destino de los paquetes en el nivel de enlace, con la ventaja de que no se realice la fragmentación y reensamblado en cada salto, lo cual evita que la transmisión sea lenta. No obstante, al perderse algún fragmento el paquete debe ser enviado de nuevo desde el nodo original.

#### Figura 40

*Modelo de enrutamiento “mesh-under” de paquetes en 6LoWPAN*



Fuente: (Vadillo Gutiérrez, 2014)

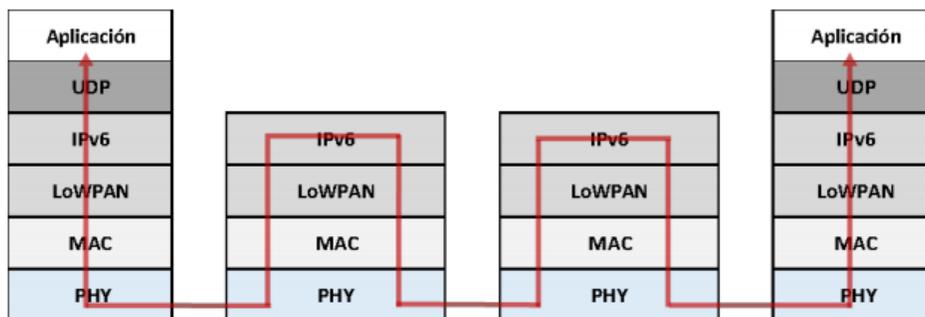
#### 2.1.9.2. Capa de red.

La capa de red de 6LoWPAN tiene la función de direccionamiento, así como la definición de los protocolos de enrutamiento utilizados para transportar los paquetes hacia su destino. En esta capa también se lleva a cabo el enrutamiento “route-over”, que proporciona la capa de adaptación, el mismo que se diferencia del enrutamiento “mesh-under”, por el proceso de envío del paquete (fragmentado/reensamblado), esto se debe a que un salto en la capa de enlace corresponde a un salto en IP y cada nodo actúa como router IP. Luego del proceso de

fragmentación, los segmentos son enviados hacia el siguiente salto donde se tiene la condición en que, si la capa de adaptación recibe todos los segmentos exitosamente en el siguiente salto, se creará un paquete IP y será enviado hacia la capa de red la cual envía el paquete hacia capas superiores, si ese es el caso. O por el contrario se considera el uso de una tabla de enrutamiento para el envío del paquete. En caso de existir fragmentos perdidos únicamente son retransmitidos desde un salto anterior (Murillo, 2017).

**Figura 41**

*Modelo de enrutamiento “route-over” de paquetes 6LoWPAN*



Fuente: (Vadillo Gutiérrez, 2014)

### **2.1.9.3. Capa transporte.**

Permite el proceso de entrega de los segmentos de información a la capa de aplicación.

Existen dos tipos de protocolos de transporte:

UDP (User Datagram Protocol). Es el protocolo no orientado a conexión. Se caracteriza por la entrega de paquetes los cuales pueden no llegar en orden, por lo que es un protocolo poco fiable al no contar con un mecanismo de confirmación de entrega o recepción del paquete.

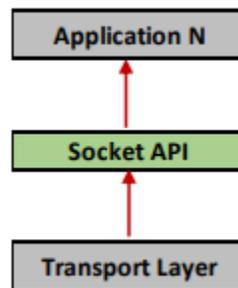
TCP (Transmission Control Protocol). A diferencia de UDP este es un protocolo orientado a conexión, ya que está enfocado a brindar una comunicación confiable entre dos hosts, gracias al establecimiento de conexión, conocida como negociación en tres pasos (Murillo, 2017).

#### 2.1.9.4. Capa Aplicación.

Como se ilustra en la Figura 43, en 6LoWPAN, la capa aplicación utiliza un socket como interfaz, donde cada aplicación habilita un socket con el cual se puede recibir y enviar paquetes.

#### Figura 42

*6LoWPAN utiliza un socket como interfaz*



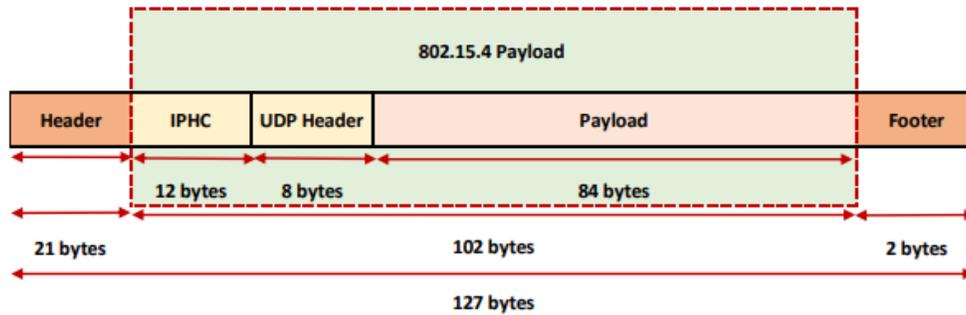
Fuente: (Murillo, 2017)

Existen distintos protocolos que se pueden implementar para el caso de redes con dispositivos de recursos limitados, algunos de ellos son: REST, CoAP, MQTT, AMPQ.

El formato de la trama MPDU del estándar 6LoWPAN se compone: 23 bytes que constituyen el encabezado y footer de la capa MAC del estándar IEEE 802.15.4, 12 bytes adheridos por IPv6, enseguida se tienen los 8 bytes agregados por el protocolo UDP y 84 bytes disponibles para carga útil de datos, lo descrito se puede apreciar en la Figura 43.

#### Figura 43

*MPDU del estándar 6LoWPAN*



Fuente: (Murillo, 2017)

### 2.1.10. Sistema Operativo Contiki

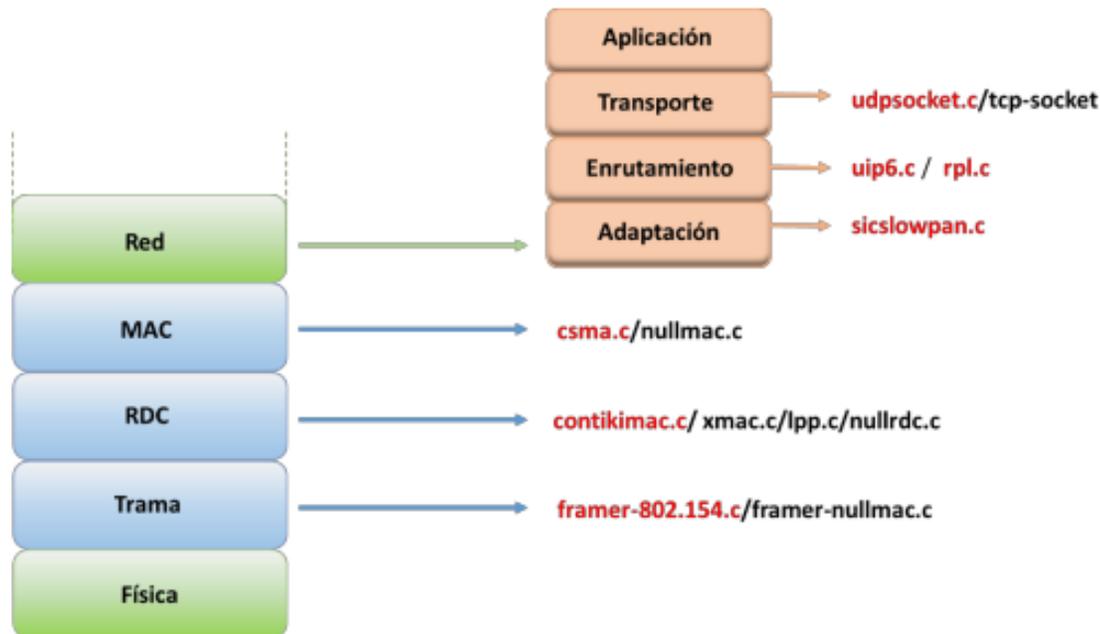
Contiki es un sistema operativo de código abierto empleado en redes de sensores, implementado en lenguaje C y soporta distintas arquitecturas de microcontroladores, como Texas Instruments MSP430 o Atmel AVR; Contiki cuenta con una pila de red conocida como  $\mu$ IP de baja potencia para poder dar soporte a IPv6. (Trinidad, 2020):

#### 2.1.10.1. Capas en Contiki.

A diferencia del modelo de 6LoWPAN, el “netstack” (pila de protocolos de Contiki OS) implementado en Contiki cuenta con tres capas diferentes entre la capa de red y la capa física: Trama (“Framer”), Ciclo de Trabajo del Radio (RDC, “Radio Duty-Cycle”) y mecanismo de acceso al medio (MAC, “Medium Access Control”). La Figura 45 muestra la organización de las capas en Contiki OS (Murillo, 2017).

### Figura 44

*Netstack Contiki OS*



Fuente: (Murillo, 2017)

**Capa Radio:** Es la encargada de la transmisión y recepción de tramas a través del medio inalámbrico por medio de los drivers de la plataforma especificada.

**Capa Framer:** Se encarga de dar formato al paquete mediante el estándar IEEE 802.15.4 en caso de que fuese necesario a través del protocolo *framer\_802154* en un escenario 6LoWPAN.

**Capa RDC:** En redes de baja potencia, el transceptor radio es recomendable que se apague siempre que sea posible con el fin de ahorrar energía e incrementar la autonomía de las baterías. En Contiki esto se consigue por medio de la capa Radio Duty Cycling (RDC) (Vadillo Gutiérrez, 2014).

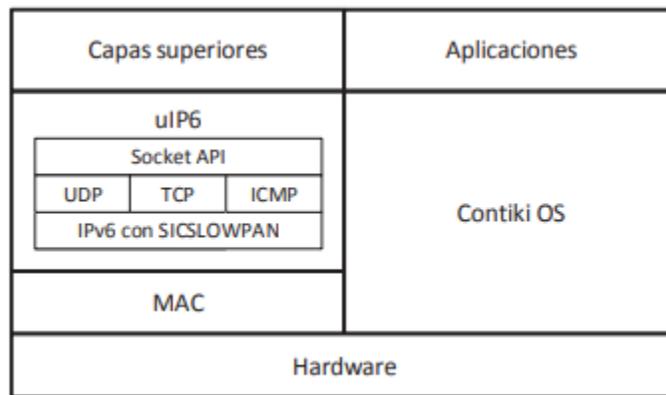
**Capa MAC:** La capa MAC de Contiki es la responsable de evitar las colisiones en el medio inalámbrico y retransmitir las tramas en caso de que existan colisiones. Contiki facilita dos tipos de driver: uno que emplea el mecanismo CSMA/CA y otro NullMAC. De forma análoga a

lo que pasa con la capa RDC, no todas las radios son compatibles con el driver, luego no siempre es posible utilizar CSMA/CA.

**Capa de red:** Contiki proporciona dos tipos de arquitecturas, Rime o uIP. La implementación de 6LoWPAN que incorpora uIP recibe el nombre de SICSLOWPAN, como se muestra en la Figura 46. Contiki tiene implementado el protocolo de enrutamiento RPL, enfocado esencialmente para el mundo de las redes LoWPAN.

**Figura 45**

*Stack de 6LoWPAN de Contiki*



Fuente: (Vadillo Gutiérrez, 2014)

**2.1.11. Protocolo MQTT (Message Queue Telemetry Transport)**

Es un protocolo de mensajería asíncrona, el cual usa el tipo de comunicaciones machine-to-machine (M2M), en el aspecto de Internet de las Cosas. El funcionamiento de este protocolo se da mediante el intercambio de mensajes haciendo uso del modelo de publicación y suscripción.

**Modelo publicación/suscripción**

Este tipo de modelo se basa en la comunicación de los clientes con un agente llamado broker. Para el envío y recepción de mensajes, tanto cliente como bróker se suscriben a un topic donde se acomodan los mensajes, convirtiendo así este modelo en un modelo no-bloqueante.

### **Cliente MQTT**

Tanto publicadores como subscriptores son clientes MQTT, aunque también un cliente puede ser ambos a la vez.

### **Broker MQTT**

Es el servidor cuya función es la distribución de los mensajes a los receptores. Antes de encaminar los mensajes hacia los clientes verifica el topic al cual están suscritos los clientes.

Además, el broker permite que los mensajes sean persistentes, lo cual indica que los mensajes se quedaran guardados hasta que el cliente se encuentre conectado. Por otro lado, brinda la posibilidad de que se autentique la identidad de cada cliente.

La conexión entre los clientes y el broker se da a través del puerto 1883 y con el 8883 si la conexión funciona sobre TLS.

### **QoS (Quality of Service)**

MQTT posee un mecanismo para la gestión el envío de mensajes al cliente ante fallos. De esa manera se tiene niveles de calidad:

QoS 0: El mensaje es enviado sola vez y no se recibe confirmación de entrega.

QoS 1: El mensaje es enviado hasta que el receptor lo recibe, con lo que se puede tener mensajes duplicados.

QoS 2: Se garantiza la entrega del mensaje y solo una vez, Sin embargo, se aumenta la sobrecarga y baja el rendimiento del sistema

### Intercambio de mensajes MQTT

MQTT se basa en el modelo TCP/IP y sus mensajes se componen de las siguientes cabeceras:

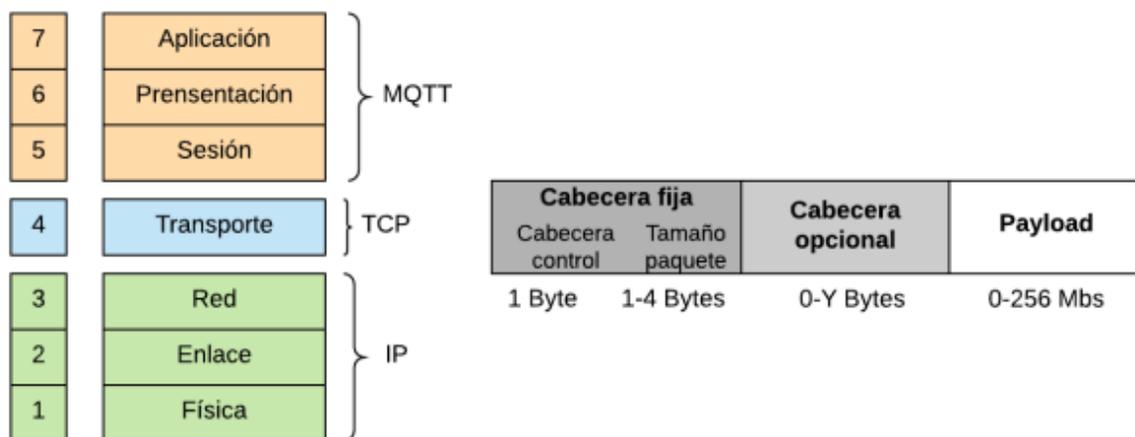
Cabecera fija: Consta de 2 a 5 bytes. Estos forman la cabecera de control y la longitud del mensaje. Esta longitud está codificada de 1 a 4 bytes, en los que se usan los 7 primeros bits y el restante se trata de un bit de continuidad.

Cabecera opcional: Esta cabecera es variable

Payload: Corresponde al contenido real del mensaje. Su tamaño máximo es de 256 MB aunque en las implementaciones reales el máximo se encuentra entre 2 y 4 kB (Mahedero, 2020)

**Figura 46**

*Estructura del mensaje MQTT*



Fuente: (Mahedero, 2020)

MQTT pose distintos tipos de mensajes que se describen el Figura 47.

**Figura 47**

*Mensajes intercambiados MQTT*

Tipo de mensaje	Tipo de mensaje	Código	Descripción
CONNECT		0X10	Petición del cliente para conectarse al servidor
CONNACK		0X20	Reconocimiento de la conexión
PUBLISH		0X30	Edición del mensaje
PUBACK		0X40	Reconocimiento de la edición
PUBREC		0X50	Edición recibida
PUBREL		0X60	Edición liberada
PUBCOMP		0X70	Edición completa
SUSCRIBE		0X80	Petición de suscripción del cliente
SUBACK		0X90	Reconocimiento de suscripción
UNSUBSCRIBE		0XA0	Petición de desuscripción del cliente
UNSUBACK		0XB0	Reconocimiento de desuscripción
PINGREQ		0XC0	Petición de ping
PINGRESP		0XD0	Respuesta de ping
DISCONNECT		0XE0	Cliente desconectado

(Mahedero, 2020)

**2.1.12. Métricas en el enlace inalámbrico**

**PDR (Packet Delivery Rate o Tasa de Paquetes Entregados)**

De acuerdo con el estudio de Godoy (Godoy et al., 2010) indica que el PDR debe estar por encima del 80%, lo que involucraría tener un buen enlace.

Para el cálculo del PDR, independiente de la distancia, se hace una transmisión de un determinado número de paquetes. Es así como el PDR se calcula de la siguiente manera:

$$PDR = \frac{\text{Paquetes Recibidos}}{\text{Paquetes Enviados}} * 100$$

*Ec. 5*

### **PER (Tasa de Error de Paquete)**

De acuerdo con la recomendación UIT-T E.800, se puede definir al PER como la relación entre el número de paquetes erróneos y el número total de paquetes transmitidos durante un intervalo determinado (López Vilos & Valencia Cordero, 2017).

$$PER = \frac{\text{Paquetes Erróneos}}{\text{Paquetes Totales}} \quad \text{Ec. 6}$$

### **LQI (Indicador de la calidad de enlace)**

Corresponde a la indicación de la calidad de los paquetes recibidos por el receptor. Puede usarse la intensidad de señal de recepción ó la relación señal ruido la cual entre mayor sea su valor se tendrá una mayor garantía de que el paquete llegue a su destino. Para la elección de rutas dentro de una red malla se considerará que sean valores altos de LQI más alto (Dignani & Tinetti, 2011).

### **RSSI (Indicador de la Potencia de la Señal Recibida)**

Corresponde a una escala de referencia utilizado para medir la potencia en la antena del receptor asociado al último paquete de información recibido. Esta métrica es de importancia ya que no solo se usa en tecnologías Wi SUN, sino que también es utilizado en tecnologías como WiFi y Bluetooth. En el caso particular de los módulos de comunicación que utilizan el protocolo Zigbee, la referencia que se usa es de 1mW y la unidad es el dBm (Nahuel Vara, 2010).

## **Retardo**

La sumatoria del retardo desde un extremo hacia otro de la comunicación se llama latencia. Cuyos contribuyentes a la latencia se tiene origen, la red y el destino dentro de una comunicación. El origen ayuda con el tiempo que se tiene desde que muestrea la señal hasta que envía los datos muestreados y el destino contribuye con el tiempo que se demora antes de analizarlo(Strauch, 2000)

### **3. Capítulo III**

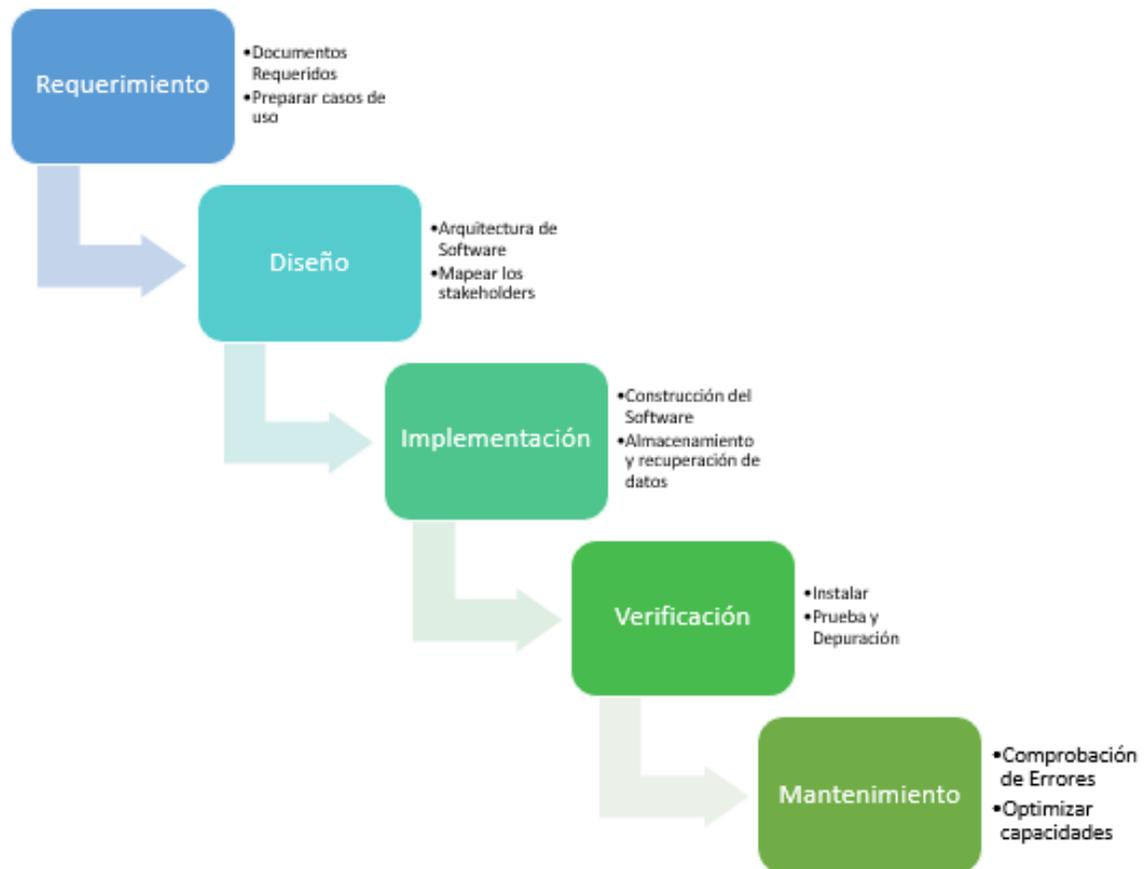
El presente capítulo describe la metodología a utilizar para el desarrollo del diseño, la misma que es conocida como metodología en cascada, la cual consta de diferentes fases que permiten el correcto funcionamiento del sistema, además se analiza la situación actual y se acota los diferentes requerimientos tanto en hardware como el software, que permitirán la construcción del sistema como tal.

#### **3.1. Metodología de Investigación**

Para el trabajo propuesto se hace uso de la Metodología en Cascada, la cual trabaja siguiendo una secuencia lineal, lo que permite identificar diferentes etapas específicas a las cuales se les debe dar cumplimiento en orden, al momento que se van implementando las adecuaciones oportunas y las pruebas para mitigar los errores que se puedan presentar, se debe tener en cuenta que es un procesos sistémico, analítico, disciplinado y técnico para el desarrollo y mantenimiento de softwares, lo cual involucra un trabajo amplio y enfatizado en la solución de las problemáticas particulares que se hayan detectado al momento de empezar a desarrollar el sistema (González et al., 2019). En la Figura 48, se representa de manera gráfica, con el fin de exponer de qué se trata y así dar cuenta de sus características.

#### **Figura 48**

*Metodología en Cascada*



Fuente: (González et al., 2019). Editado por: Cuasquer Andrea

### 3.1.1. Requerimientos

Para la determinación de esta etapa se utiliza el modelo de Benchmark que permite hacer una comparativa del hardware a utilizar. Además, es importante hacer un estudio previo de la situación actual del sitio donde se realizará las pruebas, lo que permitirá considerar distintos factores que consentirán realizar una correcta elección de los elementos a utilizar.

#### 3.1.1.1. Situación Actual

El lugar físico donde se ha analizado la problemática corresponde en primera instancia a una zona urbana, sector Yacucalle de la Ciudad de Ibarra, donde al igual que la mayoría de la

urbanización cuenta con el servicio de recolección de desechos sólidos en diferentes horarios y días, además se cuenta con contenedores que permiten almacenar los desechos a cualquier hora y día. Sin duda el hecho de que se tenga esta ventaja conlleva a una problemática mucho mayor, y radica en que no se tiene un control del llenado de los contenedores, y eso a su vez acarrea con situaciones desagradables para la población en sí, como es la presencia de fauna urbana, contaminación ambiental y mal aspecto a la ciudad.

Es importante mencionar que a pesar de que se tengan horarios establecidos por los vehículos recolectores de desechos sólidos, no es suficiente ya que, a más de no tener un control del volumen de los contenedores, las rutas establecidas tampoco son una solución a esta problemática, y con ello se tiene mayor consumo de combustible al no tener una idea en qué lugar exacto se encuentra el contenedor lleno a su capacidad máxima o lo que resulta peor sobrepasando la misma.

Es así como la propuesta de diseñar una red mesh LPWAN IEEE 802.15.4g con diferentes sensores permite tener un control del volumen de un contenedor de desechos sólidos, mejora en gran porcentaje a mitigar los problemas de saneamiento del sitio donde están ubicados, y en segundo ayudar a optimizar rutas específicas que indiquen con exactitud el lugar donde se encuentre el contenedor lleno.

### **3.1.1.2. Técnicas de Investigación**

Para el caso de estudio se emplea la técnica de investigación de campo, debido a que permite tener una observación directa del objeto de estudio en su elemento, adaptando herramientas tales como la encuesta y entrevista con las cuales se busca extraer la mayor cantidad de información en el sector Yacucalle.

Como primer método de recolección de información se hace una visita de campo donde se puede observar la distribución física de los contenedores en el sector de estudio, esto con el fin de determinar la ubicación y el número de contenedores.

Es así como se tiene un total de 38 contenedores distribuidos en diferentes zonas del sector, dicha información se encuentra en el *ANEXO B*.

Por otro lado, se aplica una encuesta a las familias que residen en el sector, para ello es necesario un cálculo muestral, donde el número total de habitantes en el sector es 5940 (Cadena & Moran, 2018), de los cuales al tomar como referencia que en cada hogar existen promedio 4 miembros, se hace una división por estos y se obtiene como resultado 1485 hogares.

Aplicando la fórmula para el cálculo de la muestra:

$$n = \frac{N * Z^2 * \sigma}{(N-1) * e^2 + Z^2 * \sigma^2} \quad \text{Ec. 7}$$

Donde:

$n$ : tamaño de la muestra extraída de la población.

$N$ : tamaño de la población residente en el sector Yacucalle.

$Z$ : valor constante correspondiente al nivel de confianza.

$\sigma$ : desviación estándar.

$e$ : límite aceptable de error muestral.

Al remplazar los datos se tiene:

$N$ : 1485

$Z$ : 1,96 con un nivel de confianza del 95%

$\sigma$ : 0,5

$e$ : Error muestral del 10% (0,1)

$$n = \frac{1485 * (1,96)^2 * 0,5}{(1485 - 1) * (0,1)^2 + (1,96)^2 * (0,5)^2} = 180$$

De esa manera se aplica la encuesta a 180 hogares tomando en cuenta que la persona encuestada sea el jefe de familia.

### **3.1.1.3. Análisis de Resultados.**

Después de haber aplicado las encuestas a las familias del sector Yacucalle se pudo determinar información que sustenta la problemática sobre la recolección de residuos sólidos, así como también poder establecer requerimientos para poder diseñar el sistema que se propone de mejor manera. El formato de la encuesta aplicada *ANEXO C* y las respectivas tabulaciones de las preguntas contenidas en las encuestas que se aplicaron se pueden evidenciar en el *ANEXO D*.

### **3.1.1.4. Propósito y Ámbito del Sistema.**

El sistema propuesto busca brindar un control del llenado del contenedor de desechos sólidos, para ello se toma en consideración diferentes variables que permiten determinar cuándo un contenedor esta al tope de su capacidad máxima tales como: volumen, peso, nivel de CO<sub>2</sub>, así como también la ubicación de los contenedores distribuidos geográficamente en el sector Yacucalle.

La aplicación el sistema permite tener una recolección controlada debido a que se tiene un conocimiento previo del estado del contenedor de desechos sólidos y la ubicación, siendo información destacada para los encargados de manejar la recolección de desechos sólidos, en primera instancia se tiene priorización de rutas y en segunda se logra mitigar la contaminación ambiental y daños en la salud de la población por el exceso de basura al igual que por la

generación se gases contaminantes generados por los VRR al recorrer innecesariamente sitios donde los contenedores aún no están llenos totalmente.

### **3.1.1.5. Descripción General del Sistema.**

El sistema consta de un arreglo de sensores que sirven en la obtención de datos sobre volumen, peso, nivel de CO<sub>2</sub>, además de un dispositivo que indica la ubicación del contenedor de desechos sólidos. El arreglo se conecta a Arduino, el cual a su vez mediante comunicación serial envía los datos al módulo con tecnología IEEE 802.15.4 configurado como cliente, el mismo que sirve de Gateway y establecerá conexión con otros nodos de iguales características que cumplen la misma función, de igual manera estará conectado otro módulo IEEE 802.15.4 que cumplirá funciones de Nodo Coordinador el cual establecerá un control de todos los nodos, de esta manera se tendrá una topología mesh.

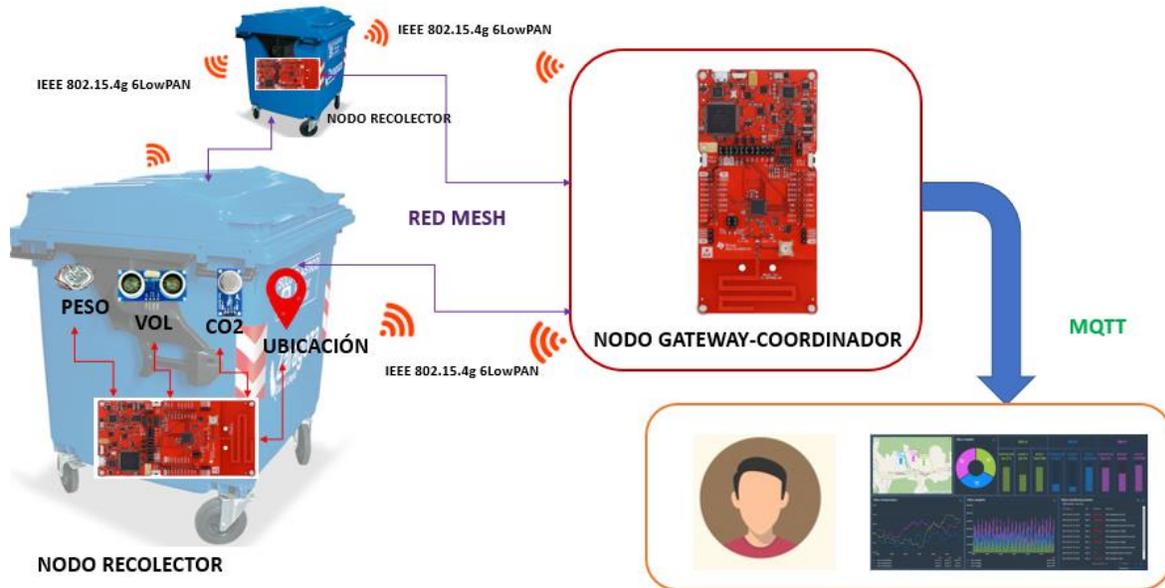
Además, se trabajará en conjunto con software embebido y el uso del estándar 6lowpan que sea compatible con los dispositivos a emplearse independientemente de la marca que estos tengan. Los datos obtenidos se enviarán en tiempo real a una plataforma web mediante el protocolo MQTT.

También se emplea un método de representación de datos para que el usuario pueda hacer uso de la información.

Por otro lado, es necesario evaluar el desempeño de la red mesh en el escenario propuesto considerando métricas como: PDR (Proporción de entrega de paquetes), PER (Tasa de error de paquete), RSSI (Indicador de la Fuerza de la Señal Recibida), LQI (Calidad del Enlace) y Retardo. En la Figura 49, se refleja lo mencionado en los párrafos anteriores.

### **Figura 49**

### *Vista general del sistema*



#### **3.1.1.6. Determinación de Requerimientos**

Para poder establecer los requerimientos necesarios para la construcción de sistema se toma como base al estándar ISO/IEC/IEEE 29148:2011, que brinda normas para el establecimiento de requisitos y procesos involucrados en la implementación de sistemas, software y servicios.

De esa manera para el caso de estudio, se tiene requerimientos involucrados con el usuario o partes interesadas (Stakeholders), con la operabilidad del sistema y con la arquitectura del sistema.

Para la determinación de stakeholders, es necesario determinar cuáles son los elementos que se encuentran vinculados al proyecto ya sea directa o indirectamente, así como también aquellos individuos que presenten interés en el desarrollo del trabajo, en la Tabla 10, se especifican los stakeholders relacionados con la elaboración del sistema.

Lo que se busca es establecer aquellas partes que estarán vinculadas en el diseño, implementación, así como también en la presentación de sugerencias para brindar un sistema optimo que permita dar solución a los inconvenientes planteados.

**Tabla 10**

*Lista de Stakeholders involucrados*

<b>STAKEHOLDERS</b>	
<b>1</b>	Personal encargado de la recolección directa de desechos sólidos.
<b>2</b>	Encargado de la Gestión de Residuos Sólidos del Municipio de Ibarra
<b>3</b>	Ciudadanos del Sector Yacucalle
<b>4</b>	Director de Tesis (Ing. Edgar Maya, MSc)
<b>6</b>	Asesor (Ing. MSc. Fabián Cuzme, MSc)
<b>7</b>	Responsable del Proyecto (Andrea Cuásquer)

### **3.1.1.7. Nomenclatura de requerimientos.**

Es importante abreviar los requerimientos que van a permitir comprender de mejor manera tales como requerimientos de stakeholdders, funcionales, y de arquitectura. En la Tabla 11, se presenta las abreviaturas de los requerimientos mencionados.

**Tabla 11**

*Abreviaturas de Requerimientos*

<b>Tipo de Requerimiento</b>	<b>Abreviatura</b>
<b>Requerimientos de Stakeholders del Sistema</b>	StSR
<b>Requerimientos Funcional del sistema</b>	SySR

---

**3.1.1.8. Requerimientos Operacionales y de Usuario.**

Para determinar los requerimientos se tiene que establecer los requisitos de stakeholders involucrados para el correcto funcionamiento del sistema, los cuales buscan satisfacer objetivos a cumplir que permiten evaluar al mismo, los cuales están sustentados en la entrevista y las encuestas realizadas (véase el ANEXO A , ANEXO B).

**Tabla 12***Requerimientos de Stakeholders*

StSR				
#	Requerimiento	Prioridad		
		Alta	Media	Baja
<b>Requerimientos Operacionales</b>				
StSR 1	Se debe hacer pruebas del funcionamiento del sistema en los contenedores del sector Yacucalle	X		
StSR 2	El sistema debe brindar un bajo consumo energético		X	
StSR 3	El sistema debe indicar que el contenedor se encuentra lleno.	X		
StSR 4	Protección del Sistema Embebido	X		
StSR 5	Conectividad a Internet	X		
<b>Requerimientos de Usuario</b>				
StSR 6	Sistema de fácil uso para el usuario	X		
StSR 7	Visualización de datos obtenidos amigable con el usuario	X		

### 3.1.1.9. Requerimientos Funcionales del Sistema.

Constituyen requerimientos que responden a las funcionalidades específicas del sistema, tales como su uso, modos, estado entre otros. En la tabla 13 se especifica este tipo de requerimientos.

**Tabla 13**

#### *Requerimientos Funcionales del Sistema*

SySR				
#	Requerimiento	Prioridad		
		Alta	Media	Baja
<b>Requerimientos de Uso</b>				
SySR 1	Alimentación por batería	X		
SySR 2	Alimentación por tomacorriente			X
SySR 3	Recolección de datos ininterrumpida		X	
<b>Requerimientos de Interface</b>				
SySR 4	Los sensores de volumen, peso y CO2 y ubicación se conectan a sistema embebido.	X		
SySR 5	Pines de entrada y salida del sistema embebido.	X		
<b>Requerimientos de Modos/Estado</b>				
SySR6	Modo bajo consumo de energía	X		
<b>Requerimientos Físicos</b>				
SySR7	Sistema embebido de tamaño reducido			X
SySR8	Protección física Sistema Embebido	X		
SySR9	Ubicación óptima de los diferentes tipos de Nodos	X		
SySR9	Protección física de los diferentes tipos de Nodos	X		
SySR10	Material de cajas de protección de nodos resistente a condiciones ambientales.	X		

### 3.1.1.10. Requerimientos iniciales de arquitectura.

Corresponde a la selección de requerimientos tanto en hardware como en software que son necesarios para el desarrollo del sistema. Los requerimientos satisfacen el cumplimiento de los puntos establecidos en el alcance, sección 1.4, y se toma como un apoyo el trabajo: (Huilca Salcedo & Sichiqli Velecela, 2019), para entender el funcionamiento de un sistema embebido y como se da la comunicación entre nodos. La tabla 15 muestra este tipo de requerimientos.

**Tabla 14**

*Requerimientos iniciales de arquitectura*

SRSH				
#	Requerimiento	Prioridad		
		Alta	Media	Baja
<b>Requerimientos Lógicos</b>				
SRSH 1	Comunicación de sensores con el Nodo Recolector	X		
SRSH 2	Calibración de sensores previo al funcionamiento en el entorno de estudio	X		
<b>Requerimientos de Software</b>				
SRSH 3	Software de compatibilidad con los módulos a trabajar	X		
SRSH 4	Lenguaje de Programación sencillo	X		
SRSH 5	Software libre para programación de nodos	X		
SRSH 6	Interfaz de programación sencilla	X		
SRSH 7	Representación de los datos amigable con el usuario	X		
<b>Requerimientos de Hardware</b>				
SRSH 8	Entradas analógicas y digitales disponibles para la conexión de sensores.	X		
SRSH 9	Disponibilidad de los dispositivos en el mercado		X	
SRSH 10	Lectura en tiempo real de los diferentes sensores	X		
SRSH 11	Tecnología IEEE 802.15.4g	X		

SRSH 12	Compatibilidad de sensores con microcontrolador	X	
SRSH 13	Tamaño reducido		X
SRSH 14	Bajo costo	X	
SRSH 15	Disponibilidad de librerías para microcontrolador	X	
SRSH 16	Fácil calibración	X	
SRSH 17	Precisión en las tomas de datos	X	
SRSH 18	Funcionamiento a prueba de agua		X
<b>Requerimientos Eléctricos</b>			
SRSH 19	Bajo consumo de energía del sistema	X	
SRSH 20	Fuente de alimentación para el sistema embebido y nodos.	X	
SRSH 21	Valor de sensibilidad del receptor considerable		X
<b>Requerimientos de Seguridad</b>			
SRSH 22	Protección del sistema embebido y red mesh que evite la manipulación externa.	X	
SRSH 23	Protección del sistema a nivel lógico evitando que se tenga acceso no autorizado a la información manejada.	X	
<b>Requerimientos Inalámbricos</b>			
SRSH 24	Frecuencia de trabajo (Sub-1 GHz)	X	
SRSH 25	Alcance mínimo de distancia entre nodos 50 m		X

### 3.1.2. Diseño del Prototipo

Para el diseño del sistema se analizó los requerimientos que permiten satisfacer su óptimo funcionamiento, también se considera la situación actual que conlleva a adaptar el sistema de acuerdo con las necesidades presentes. Posteriormente se determina los componentes en hardware y software aptos, así como también se realizará un diagrama de bloques que describe su completo funcionamiento.

### 3.1.2.1. Construcción de modelos de Hardware y Software.

Para este procedimiento es necesario hacer un estudio de los diferentes elementos que permitan dar cumplimiento a los requerimientos que se establecen para la construcción del sistema. Se realiza una comparación mediante tablas que buscan satisfacer los requerimientos como se señala en el apartado 3.2.1; se asigna una valoración de “1” para cumplimiento y “0” para no cumplimiento, tal como se muestra en la Tabla 15.

**Tabla 15**

*Descripción Cumple o No cumple*

<b>Descripción</b>	<b>Valor asignado</b>
<b>Cumplimiento</b>	1
<b>No Cumplimiento</b>	0

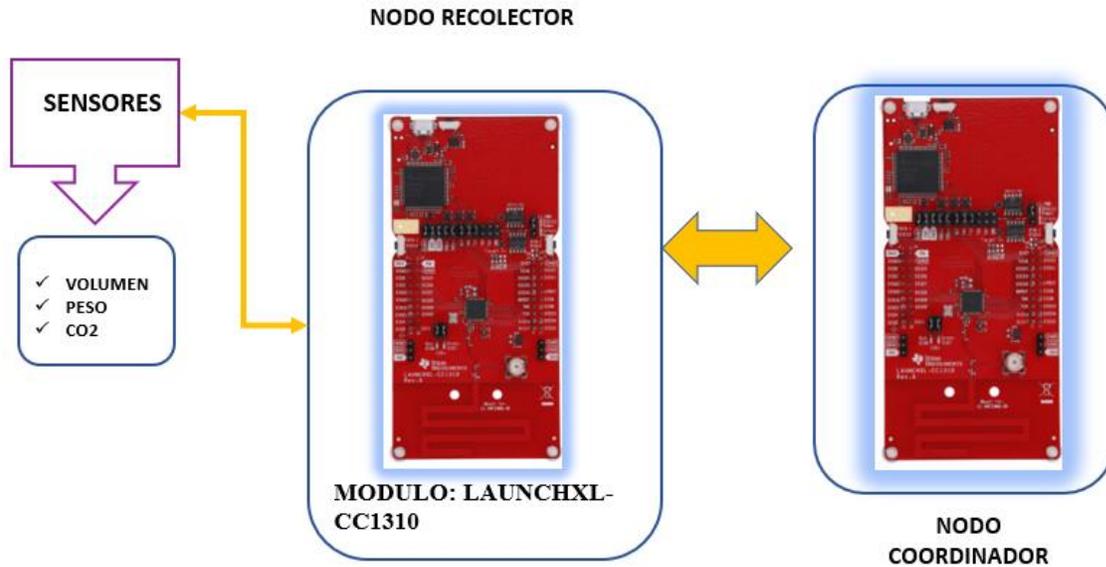
#### **a. Modelos de Hardware**

##### **MODELO 1 DE HARDWARE**

Consta de un arreglo de sensores con conexión directa al módulo LAUNCHXL-CC1310 configurado como Nodo Recolector, un segundo dispositivo configurado como Nodo Coordinador, se basa en el estándar IEEE 802.15.4g que para este modelo trabaja en la frecuencia de 2.4Ghz y frecuencias Sub-1 GHz.

##### **Figura 50**

*Modelo 1 Hardware*

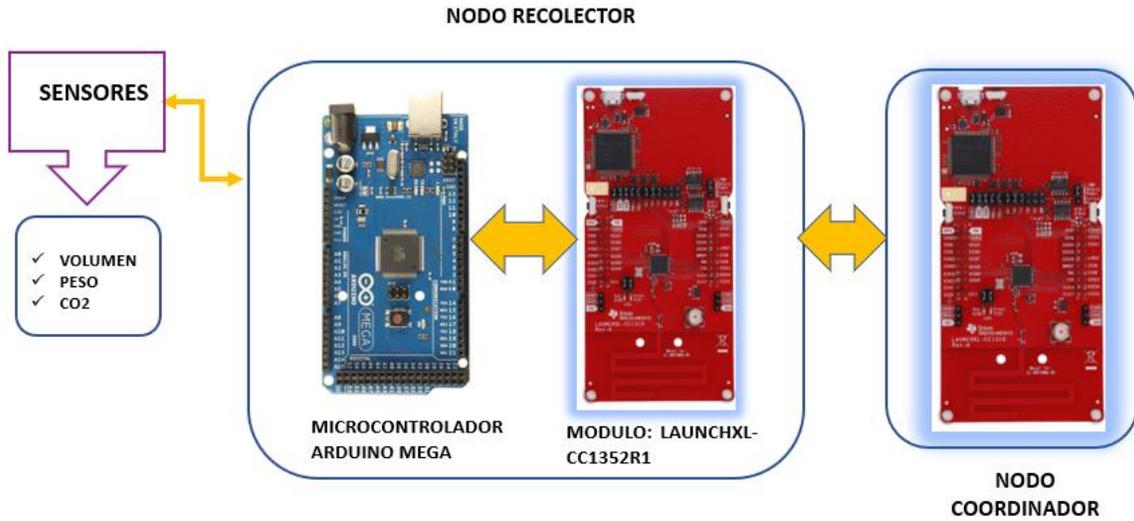


## MODELO 2 DE HARDWARE

El segundo modelo consta de un arreglo de sensores en conjunto con un microcontrolador Arduino Mega para la toma de datos, se conecta al módulo LAUNCHXL-CC1352R1 con la configuración de Nodo Recolector, un segundo módulo como nodo coordinador, la frecuencia de trabajo IEEE 802.15.4g se da por debajo de 1 GHz (Sub-1GHZ), donde el presente módulo permite la configuración deseada.

### Figura 51

*Modelo 2 Hardware*

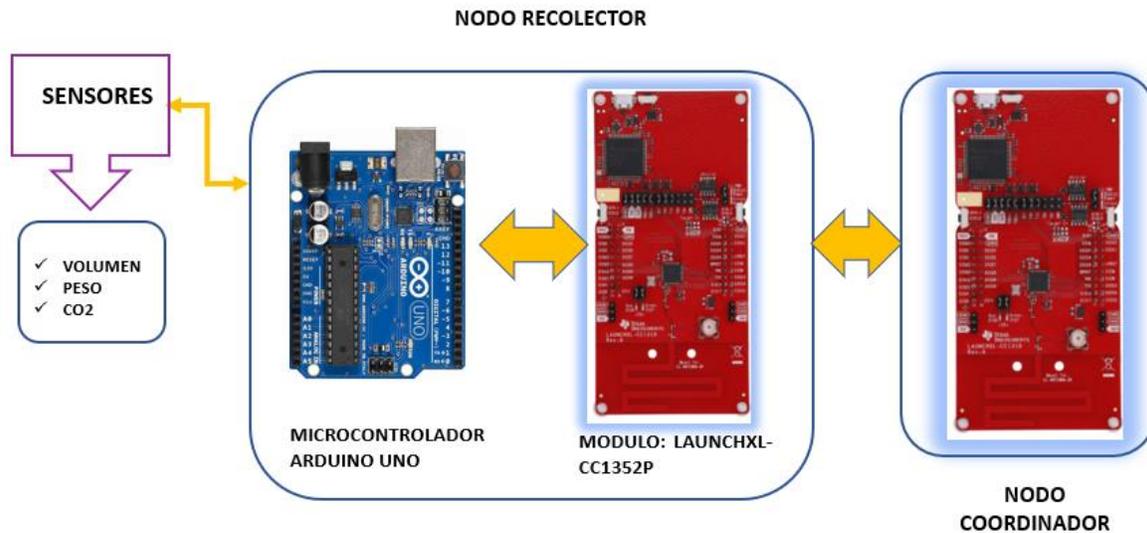


### MODELO 3 DE HARDWARE

El tercer modelo que se plantea consta del arreglo de sensores con un microcontrolador Arduino UNO, el cual a su vez se conecta al módulo LAUNCHXL- CC1352P configurados como Nodo Recolectores y de igual manera otro dispositivo de la misma familia con configuración de Nodo Coordinador formando la red mesh. Posee propiedades similares a los anteriores módulos, sin embargo, su costo es mayor.

#### Figura 52

*Modelo 3 Hardware*



### 3.1.2.2. Elección del Hardware

Como primera elección se hace referencia a los módulos que tiene las funciones de Nodo Recolector y Coordinador, teniendo en cuenta que se trabajará con el mismo módulo para cada tipo de nodo dentro de la mesh. La Tabla 16, propone los módulos que cumplen con la mayoría de los requerimientos establecidos.

**Tabla 16**

*Elección de Hardware del sistema*

Hardware	Requerimientos							Valoración Total
	SySR	SRSH	SRSH	SRSH	SRSH	SRSH	SRSH	
	1	8	11	12	19	20	22	
<b>LAUNCHXL- CC1310</b>	1	1	1	1	0	1	0	5

<b>LAUNCHXL- CC1352R</b>	1	1	1	1	0	1	1	6
<b>LAUNCHXL- CC1352P</b>	1	1	1	1	0	1	0	5
1-Cumple								
0-No cumple								
Elección: LAUNCHXL-CC1352R1								

En consideración con la Tabla 17, se puede evidenciar que el módulo **LAUNCHXL-CC1352R1** cumple con la mayoría de los requerimientos establecidos, sin embargo, su precio es menor a comparación de los demás. De esa manera se hace la selección del modelo 2 apoyado por el microcontrolador Arduino Mega y resto de elementos tales como los sensores que son los mismos para cada uno de los modelos, se trabaja en conjunto con Arduino para tener la toma de datos fiable. Las especificaciones de Arduino Mega se indican en la tabla 17. Cabe mencionar además que el kit de desarrollo del módulo seleccionado posee un paquete de software idóneo para el desarrollo de aplicaciones que trabajan con frecuencias por debajo de 1 GHz y 2,4 GHz, incluido el soporte para Bluetooth Low Energy, Zigbee, Thread, basado en 802.15.4.

**Tabla 17**

*Características del microcontrolador Arduino Mega*

<b>Características</b>	<b>Valores</b>
<b>Microcontrolador</b>	ATmega2560
<b>Voltaje Operativo</b>	5v
<b>Voltaje de Entrada</b>	7-12 v
<b>Voltaje de Entrada(límites)</b>	6-20v
<b>Pines digitales de E/S</b>	54 (15 proveen salida PWM)
<b>Pines análogos</b>	16
<b>Corriente DC por cada pin E/S</b>	40 Ma

### **3.1.2.3. Estudio de Benchmark de sensores.**

El estudio benchmark permite determinar los diferentes sensores que se van a utilizar para realizar las respectivas mediciones del sistema y complementaran el funcionamiento del modelo de Hardware seleccionado. Se analizan características lo que permite la elección del sensor más idóneo, además para poder encontrar los posibles sensores se hace referencia a parámetros que se van a medir tales como:

- ✓ Volumen
- ✓ Peso
- ✓ CO2
- ✓ Ubicación

Posteriormente se presentan los sensores con las características que ofrecen:

#### *a. Sensores de Volumen*

Los sensores de volumen disponibles en el mercado están diseñados para diferentes aplicaciones dependiendo del elemento, es decir si se requiere hacer la medición en líquidos o sólidos así mismo de la cantidad de elemento que se quiere verificar.

En este caso se opta por la medición de volumen para sólidos debido a que la mayoría de los desechos presentes en los contenedores son de tipo sólido. En el mercado se encuentran distintos tipos de sensores en este caso se ha elegido un sensor de tipo ultrasónico y se tiene como candidatos los siguientes sensores con sus características específicas:

## **Ultrasónico HC-SR04**

Este sensor permite medir distancia mediante ultrasonido. Se encuentra disponible en el mercado local a un costo de 2,00 USD, además el rango de operación corresponde al establecido para el caso de estudio.

### **Figura 53**

*Sensor Ultrasónico HC-SR04*



## **Sensor ultrasónico a prueba de agua 3-450cm**

El sensor de ultrasonido determina la distancia midiendo los lapsos de tiempo entre el envío y recepción del pulso. Posee precisión, es estable, de bajo consumo y rápida respuesta. Usado en aplicaciones de evasión de obstáculos, indicadores de parqueo, detección de objetos, etc. Se encuentra disponible en el mercado local a un precio de 24,50 USD.

### **Figura 54**

*Sensor ultrasónico a prueba de agua 3-450cm*



### Selección del Sensor de Volumen

Para poder seleccionar el sensor adecuado para la determinación de volumen. En la tabla 18, se detalla las especificaciones de cada sensor al igual que el precio y disponibilidad.

**Tabla 18**

*Comparativa Sensores Ultrasónicos*

Sensor	Ultrasónico HC-SR04	Sensor ultrasónico a prueba de agua 3-450cm
<b>Parámetros</b>		
<b>Voltaje de Operación</b>	5V DC.	3.3~5V
<b>Rango de Medición</b>	2cm a 400cm.	3-450cm
<b>Precisión</b>	+/- 3mm.	+/-2mm
<b>Angulo de Apertura</b>	15°	100°
<b>Corriente de reposo:</b>	< 2mA.	≤5Ma
<b>Corriente de trabajo:</b>	15mA.	≤8mA
<b>Temperatura de operación:</b>	N/A	-15~60°C

<b>Tiempo de respuesta:</b>	N/A	100ms
<b>Precio</b>	2 USD	24,50
<b>Disponibilidad</b>	Quito	Medellín

Fuente: Autoría

### **Elección del sensor para medición de volumen en base al cumplimiento de requerimientos**

A continuación, se hace una comparativa en base a los requerimientos de Hardware de los sensores ultrasónicos en este caso, lo cual se puede evidenciar en la Tabla 19.

**Tabla 19**

*Elección sensor para medición de volumen*

<b>Hardware</b>	<b>Requerimientos</b>								<b>Valoración Total</b>
	<b>SRSH 9</b>	<b>SRS H 12</b>	<b>SRS H 13</b>	<b>SRS H 14</b>	<b>SRS H 15</b>	<b>SRS H 16</b>	<b>SRS H 17</b>	<b>SRS H 18</b>	
<b>Ultrasónico</b>									
<b>HC-SR04</b>	1	1	1	1	1	1	1	0	7
<b>Sensor ultrasónico a prueba de agua 3-450cm</b>	1	1	1	0	1	0	1	1	6

<b>1</b>	Cumple
<b>0</b>	No cumple
<b>Elección</b>	Ultrasónico
<b>n</b>	o HC-SR04

Según la Tabla 19, al analizar los requerimientos de los dispositivos se puede evidenciar que el sensor que cumple con la mayoría de los requerimientos es el **Ultrasónico HC-SR04** a pesar de que el otro candidato tiene un mayor rango de censado y es a prueba de agua, sin embargo, se puede hacer una distribución de más de un sensor que cubra un mayor rango de medición de igual manera es un sensor más económico y accesible

*b. Sensores de Peso*

Según las especificaciones de mercado del contenedor que se ha implementado en el sistema de contenedores en el sector de estudio, el peso ideal de desechos dentro del contenedor de 1100 litros es equivalente a 250kg. Sin embargo, no se tiene la certeza que dicho valor corresponda a un contenedor lleno, debido a la variedad de desechos sólidos que las personas depositan, entonces para propósitos de prueba se opta por un peso menor, es decir 50kg como base.

**Sensor de escala corporal, 50kg de carga, resistencia de celda, sensor de medio puente, 50kg**

El interior es un medidor de tensión de medio puente de 1000 ohm, con un rango de celda de carga de 50kg y una estructura de medio puente.

Al medir, el lateral exterior aplica correctamente la fuerza externa, la parte del haz de tensión del sensor en forma de E. Y el lateral exterior debe formar una fuerza de corte en la dirección opuesta, es decir, el medio del haz de tensión debe ser capaz de sufrir cambios de flexión bajo estrés, y no debe haber obstrucciones en el otro lado de la superficie de tensión del haz de tensión. El precio en el mercado es 1,53 USD.

Dentro del sensor hay un conjunto de medidores de tensión de medio puente, que se pueden utilizar de las siguientes tres formas:

1. Utiliza un sensor con una resistencia externa para formar una medición de puente completo, el rango es el rango de un sensor: 50kg. Mayores requisitos para resistencia externa.

2. Utiliza dos sensores para formar una medición de puente completo, el rango es la suma de los rangos de los dos sensores:  $50\text{kg} \times 2 = 100\text{kg}$ .

3. Utiliza cuatro sensores para formar una medición de puente completo, el rango es la suma de los rangos de los cuatro sensores:  $50\text{kg} \times 4 = 200\text{kg}$  (Electrónicas Didácticas, 2020).

### **Figura 55**

*Sensor de escala corporal, 50kg de carga, resistencia de celda, sensor de medio puente, 50kg*



**Celda de carga 50 Kg**

La celda de carga puede medir el peso o la fuerza aplicada hasta alrededor de los 50 Kg. Está disponible en el mercado con un precio de 13,51 USD.

Las mediciones de la celda de carga pueden tener una diferencia de +/- 5% debido a una variedad de factores que incluyen temperatura, fluencia, vibración, deriva y otras interferencias eléctricas y mecánicas.

### **Figura 56**

*Celda de carga 50 Kg*



### **Selección del Sensor de Peso**

Para poder seleccionar el sensor adecuado para la determinación del peso de desechos sólidos se presenta en la Tabla 20, la comparativa con las especificaciones de cada sensor al igual que el precio y disponibilidad.

### **Tabla 20**

*Comparativa Sensores de peso.*

<b>Sensor</b>	<b>Sensor de escala corporal, 50kg de carga, resistencia de celda, sensor de medio puente, 50kg</b>		<b>Celda de carga 50 Kg</b>
<b>Parámetros</b>			
<b>Capacidad</b>	0-50kg		40-50
<b>Voltaje de Operación</b>	3V ~ 10V		≤10 V
<b>Precisión</b>	0.03		0.03
<b>Fluencia</b>	0.02%FS/30min		0.03 (3min)%FS
<b>Salida nominal</b>	2 +/- 0.1mv/v		N/A
<b>Sobrecarga segura</b>	120%FS		N/A
<b>Sobrecarga Definitiva</b>	150%FS		150%FS
<b>Disponibilidad</b>	Medellín		Quito
<b>Precio</b>	1.54 USD		13.51

**Elección del sensor para medición de peso en base al cumplimiento de requerimientos**

Para el caso de la variable peso se hace una comparativa en base a los requerimientos de Hardware de los sensores de peso que se ha elegido, lo mencionado se evidencia en la Tabla 21.

**Tabla 21**

*Elección de sensor para medición de peso*

<b>Hardware</b>	<b>Requerimientos</b>								<b>Valoración Total</b>
	<b>SRSH</b>	<b>SRSH</b>	<b>SRSH</b>	<b>SRSH</b>	<b>SRSH</b>	<b>SRSH</b>	<b>SRSH</b>	<b>SRSH</b>	
	<b>9</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	

---



---

<b>Sensor de escala corporal, 50kg de carga, resistencia de celda, sensor de medio puente, 50kg</b>	1	1	1	1	1	1	1	1	0	7
---	---	---	---	---	---	---	---	---	---	---

---

<b>Celda de carga 50 Kg</b>	1	1	1	0	1	0	1	0	5
-------------------------------------	---	---	---	---	---	---	---	---	---

---

**1** Cumple

---

**0** No  
cumple

---

**Elección**  
Celda de  
carga 50  
Kg

---

Según la comparativa de la tabla 19, se puede evidenciar que el dispositivo que ofrece mayor prestación tanto en precisión, y precio es la Celda de carga 50kg.

*c. Sensores de CO2*

El elemento CO2 está dentro de los gases nocivos que se presentan en un ambiente contaminado por la presencia de residuos sólidos, en proceso de descomposición. Es así como la medición de este

factor se convierte en un elemento de gran importancia en el presente trabajo. En el mercado existen diferentes tipos de sensores que permiten la medición de CO<sub>2</sub>, para la elección se considera los siguientes sensores:

### **Sensor Gas MQ-135**

Este sensor permite la detección de concentración de gas en diversos porcentajes, tal y como los son sus análogos MQ-3/4/5. La señal de salida que proporciona el MQ-135 es dual, de carácter analógico y digital. Respecto a la señal analógica proporcionada, esta viene a ser directamente proporcional al incremento de voltaje. La señal digital presenta niveles TTL por lo que esta señal puede ser procesada por un microcontrolador.

### **Figura 57**

*Sensor Gas MQ-135*



### **Sensor de Dióxido de Carbono MG-811**

Este tipo de sensor permite la detección de CO<sub>2</sub>, se encuentra disponible en el mercado local a un precio de 25,13 USD.

Posee dependencia de baja humedad y temperatura, larga estabilidad y reproducibilidad.

### **Figura 58**

## *Sensor de Dióxido de Carbono MG-811*



### **Selección del sensor de CO2**

Según las especificaciones de los dos sensores se hace una tabla comparativa para poder verificar si cumple con los requerimientos descritos, además de considerar su disponibilidad y precio.

**Tabla 22**

### *Comparación sensores de CO2*

<b>Sensor</b>	<b>Sensor Gas MQ-135</b>	<b>Sensor de Dióxido de Carbono MG811</b>
<b>Parámetro</b>		
<b>Voltaje de Operación</b>	5V	<5.5V
<b>Detección de partes por millón</b>	10ppm~1000ppm.	350-10000ppm de CO2
<b>Tiempo de precalentamiento:</b>	20 segundos.	N/A
<b>Corriente de operación:</b>	150mA.	N/A
<b>Resistencia de carga:</b>	Potenciómetro (Ajustable).	Ajustable
<b>Temperatura de operación:</b>	20°C~70°C	N/A

<b>Disponibilidad</b>	Quito	Quito
<b>Precio</b>	4,02 USD	25.13 USD

**Elección del sensor para medición de CO2 en base al cumplimiento de requerimientos**

Para la medición de CO2 se hace una comparativa en base a los requerimientos de Hardware de los sensores candidatos mostrados en la Tabla 23 que se ha elegido.

**Tabla 23**

*Elección de sensor para medición de CO2*

Hardware	Requerimientos								Valoración Total
	SRSH 9	SRSH 12	SRSH 13	SRSH 14	SRSH 15	SRSH 16	SRSH 17	SRSH 18	
Sensor	1	1	1	1	1	1	1	0	7
Gas MQ-135									
Sensor de Dióxido de Carbono MG-811	1	1	1	0	1	0	1	0	5
1	Cumple								

---

<b>0</b>	No cumple
----------	--------------

---

<b>Elección</b>	Sensor de Dióxido de Carbono MG-811
-----------------	--

---

Según las especificaciones el sensor que ofrece mayor prestación es el MQ-135 por tener mayor detección de partes por millón del gas que se va a medir. Además de tener un precio accesible, así como la disponibilidad en el país.

### **Módulo GPS**

Para el presente proyecto se hace uso del **MÓDULO GPS NEO6VM2 con antena** debido a que ofrece compatibilidad con Arduino y cumple con los requerimientos establecidos de igual manera su costo es accesible de 14,00 USD y se puede adquirir localmente. No se hace una comparativa ya que solo depende de que permita saber la ubicación del contenedor.

#### **a. Modelos de Software**

A continuación, se hace la construcción de los modelos de Software idóneos para trabajar con el modelo de hardware seleccionado previamente en el apartado 3.2.2.2.

### **MODELO DE SOFTWARE 1**

El sistema operativo Tiny OS es de código abierto y posee compatibilidad con protocolos 6LowPAN, RPL, CoAP. Sin embargo, no es compatible con el protocolo Thread y no trabaja con lenguajes de programación C, C++.

### **Figura 59**

### Modelo1 Software

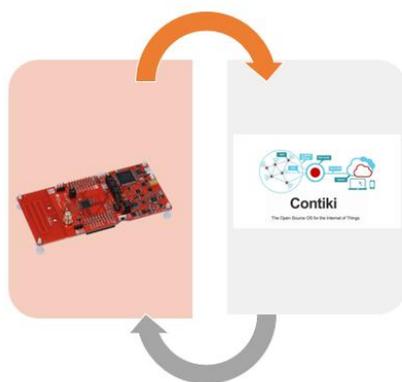


### MODELO DE SOFTWARE 2

Contiky OS posee compatibilidad con 6LowPAN, RPL, CoAP, MQTT. Al igual que Tiny OS no es compatible con Thread, y su lenguaje de programación el C.

### Figura 60

#### Modelo 2 Software

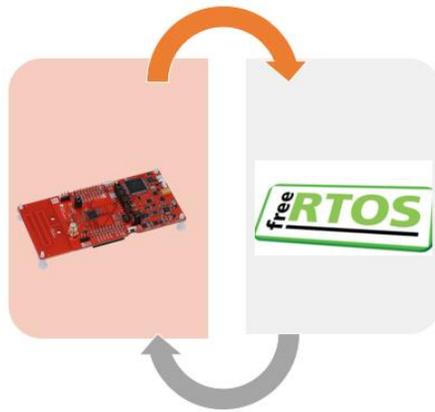


### MODELO SOFTWARE 3

FreeRTOS posee poca compatibilidad con los protocolos requeridos para el presente diseño, sus lenguajes de programación C, posee seguridad SSL/TLS.

### Figura 61

#### Modelo 3 Software



### 3.1.2.4. Elección del Software

Para la selección del sistema operativo se toman en cuenta diferentes requerimientos como compatibilidad y soporte con los equipos, tipo de distribución entre otros. Es así que en la Tabla 24, se presenta los candidatos para este caso de uso.

**Tabla 24**

*Comparación Sistemas Operativos*

Software	Requerimientos			Valoración
	SRSH 3	SRSH 4	SRSH 5	Total
<b>Tiny OS</b>	1	0	1	2
<b>Contiki</b>	1	1	1	3
<b>FreeRTOS</b>	1	0	1	2
			1-Cumple	
			0-No cumple	
			Elección: Contiki	

En base a los requerimientos de Software para ser más exactos se selecciona el Sistema operativo Contiki que tiene mayores prestaciones para el cumplimiento de objetivos del sistema en general.

### **3.1.2.5. Plataforma para visualización de datos.**

Para el proceso de visualización de la información existen diferentes plataformas que ofrecen servicios de manera gratuita o con un cargo adicional. Sin embargo, aquellas disponibles no brindan todas las opciones requeridas donde se pueda evidenciar la ubicación de los contenedores que se encuentran llenos, por lo que se procede a realizar una plataforma web personalizada que se ajuste al cumplimiento puntual de objetivos con respecto a la presentación de la información para el usuario final.

### **3.1.2.6. Diseño del Sistema.**

Luego de establecer los modelos, analizar los tipos de requerimientos y ver su cumplimiento se procede a realizar el diseño del sistema que corresponde al segundo paso del modelo en cascada, donde para una mejor comprensión se dividirá en dos fases; la primera se basa en el funcionamiento del nodo recolector conformado por los sensores de volumen, peso, CO<sub>2</sub>, módulo GPS y el launchpad cc1352r1. En la segunda etapa se describe la funcionalidad de la red en malla, los nodos que están involucrados, y la plataforma que servirá para la visualización gráfica de la información. A continuación, se desarrolla el diseño de las fases:

### **3.1.2.7. Diagrama de bloques del sistema.**

El diagrama de bloques explica de forma gráfica las etapas mencionadas con anterioridad con el fin de brindar un mejor entendimiento del sistema completo.

En el primer gráfico se tiene la primera etapa que consiste en el funcionamiento del sistema embebido, los sensores toman datos de las variables: volumen, peso, CO2 y ubicación, al tomar los datos el sistema embebido realiza un procesamiento de dicha información y luego es enviada por el microcontrolador Arduino hacia el Nodo recolector.

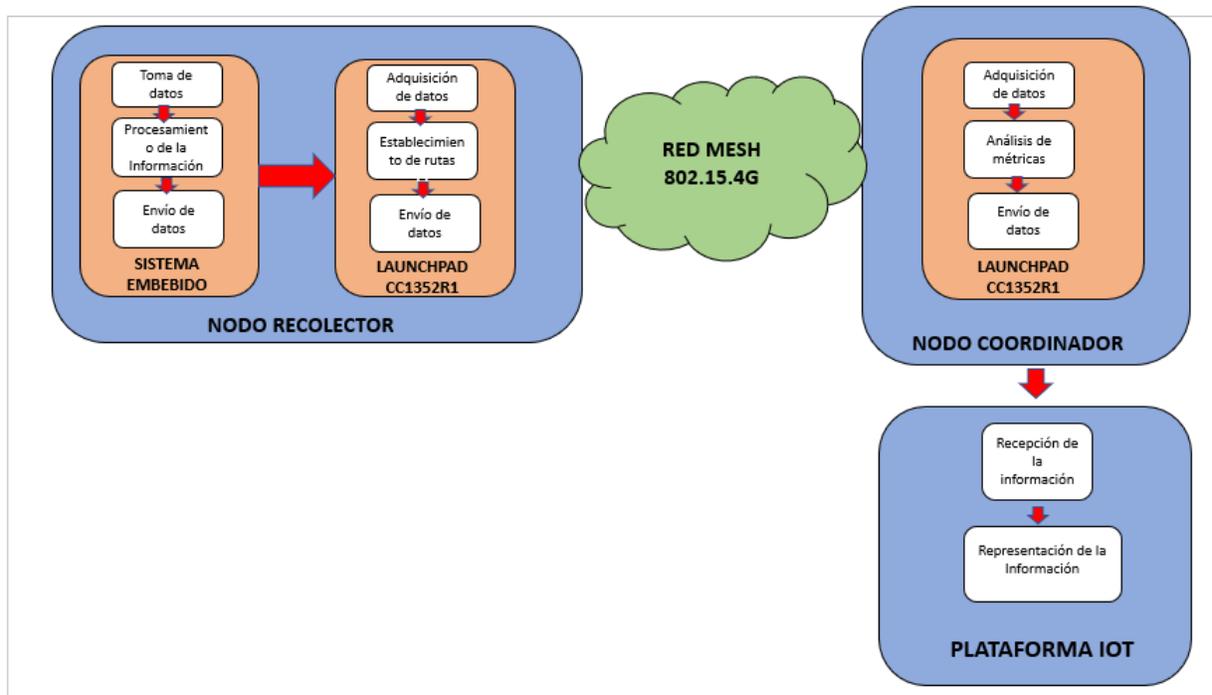
El segundo bloque corresponde al Nodo Recolector el cual recibe la información desde el sistema embebido y establece una ruta dentro de la red mesh hacia donde se va a enviar la información, en este caso hacia el Nodo coordinador.

El tercer gráfico corresponde a las funciones del Nodo coordinador que, a más de cumplir las funciones específicas de nodo principal como es el control de la red mesh, adquiere la información que envían los nodos recolectores y la envía hacia la plataforma IOT, además es en donde se analiza las métricas correspondientes al desempeño de la red mesh.

Por último, se tiene el bloque de la plataforma IOT donde se puede visualizar la información obtenida desde los nodos recolectores. La representación de la información ayuda al usuario a conocer la ubicación del contenedor que se encuentra lleno.

## **Figura 62**

*Diagrama de Bloques del Sistema*

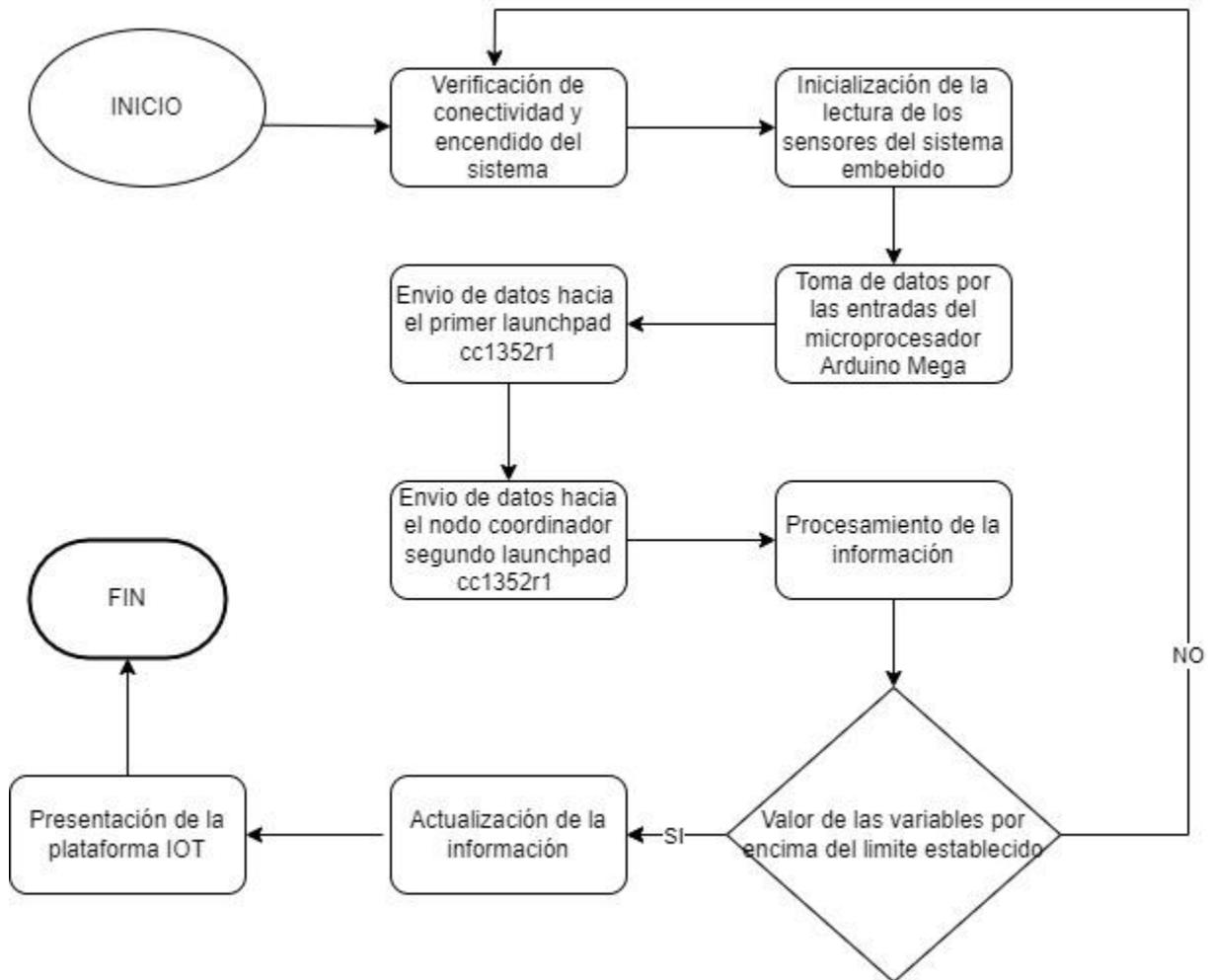


### 3.1.2.8. Diagrama de Flujo del Sistema

El proceso empieza con la inicialización del censado de los sensores: MQ-135, celda de carga y ultrasónico HC-SR04, los cuales entregan datos a través de los pines de Arduino el cual se comunica con el Nodo CC1352R1 que actúa como nodo recolector, este a su vez envía la información a través de la red usando el protocolo IPv6 hacia el Nodo coordinador el cual realiza el procesamiento de los datos y verifica el estado de las variables las cuales notificarán el estado del contenedor de residuos sólidos lleno y su ubicación, información que será mostrada en la plataforma IOT.

#### Figura 63

*Diagrama de Flujo del Sistema*



### 3.1.2.9.Arquitectura del Sistema

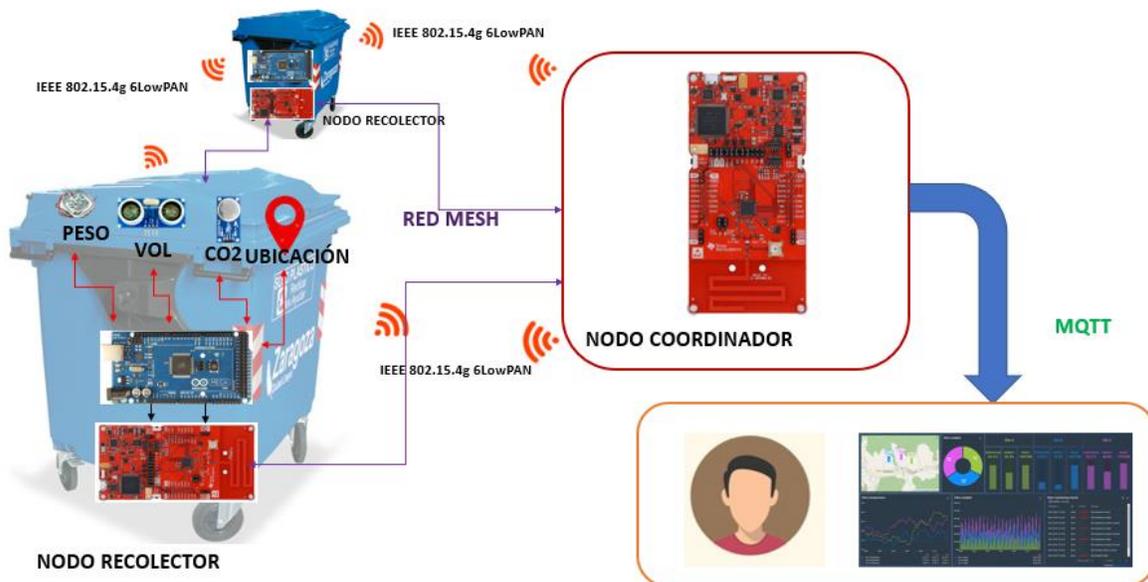
El sistema cuenta con un arreglo de sensores que está conectado a un microcontrolador Arduino para la toma de datos, este se comunica con un nodo recolector IEEE 802.15.4g. Dicho sistema está ubicado dentro del contenedor de residuos sólidos y permite la toma de información por medio de los sensores que lo componen.

El Nodo recolector se comunica a su vez con un Nodo coordinador inalámbricamente a través del protocolo IEEE 802.15.4g, el cual se encuentra configurado para que procese la información

obtenida de los nodos recolectores y envíe la información hacia plataforma IOT en la cual se puede apreciar la visualización de la información de los contenedores que se encuentran llenos y su respectiva ubicación.

### Figura 64

#### Diagrama de Arquitectura del Sistema



#### 3.1.2.10. Diseño Fase 1

La fase 1 corresponde al Nodo Recolector en el cual se encuentra el sistema embebido que consta de tres sensores, el primero, un sensor ultrasónico que permite determinar el volumen de los desechos sólidos a través de la medición de distancia entre la base superior del contenedor y los residuos e indica si el contenedor está lleno, el segundo sensor consta de una celda de peso con la que se puede conocer la cantidad de residuos sólidos basándose en un determinado peso, se tiene además el sensor de CO2 que permite medir la concentración del gas debido a que algunos residuos al entrar en estado de descomposición generan este gas y por último un módulo

GPS que indica la ubicación del contenedor que ya está lleno. Los sensores se conectan con Arduino y este por medio de comunicación serial envía la información a un launchpad cc1352r1 con tecnología IEEE 802.15.4g.

Para el diseño del proyecto, en la etapa 1 que corresponde al desarrollo del sistema embebido en conjunto con el módulo MCU CC1352R1 que cumple la función del Nodo recolector.

#### **a. Diagrama de Pines del microcontrolador Arduino Mega 2560 R3**

El microcontrolador Arduino mega posee las siguientes entradas y salidas:

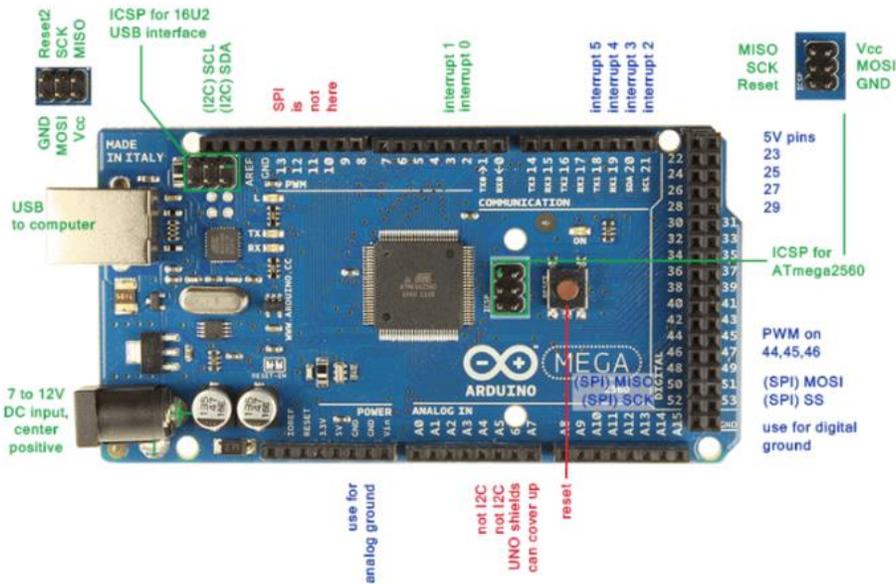
Posee 54 pines digitales que funcionan como entrada/salida.

Además, dispone 16 entradas análogas, un cristal oscilador de 16 MHz, una conexión USB, un boton de reset y una entrada para la alimentación de la placa.

La comunicación entre la computadora y Arduino se da mediante el Puerto Serie. (Panama Hitek, n.d.)

#### **Figura 65**

*Diagrama de pines Arduino Mega*



Fuente: (Panama Hitek, n.d.)

### b. Diagrama de Pines del módulo MCU CC1352R1

El MCU CC1352R1 posee diferentes entradas que permiten la comunicación. A continuación, se realiza una explicación de las entradas y sus características:

**Tabla 25**

*Descripción de entradas del microcontrolador*

ENTRADAS	DESCRIPCIÓN
SSI	Las SSI son interfaces seriales síncronas que son compatibles con las interfaces seriales síncronas de SPI, MICROWIRE y TI. Los SSI admiten tanto el maestro como el esclavo SPI hasta 4 MHz.

---

<b>UART</b>	Los UART posee funciones de transmisor y receptor asíncrono universal. Permiten la generación de velocidad en baudios flexible hasta un máximo de 3 Mbps.
<b>I2S</b>	La interfaz I2S se usa para el manejo de audio digital, así como también para interconectar micrófonos de modulación de densidad de pulso (PDM).
<b>I2C</b>	La interfaz I2C se utiliza para la comunicación con dispositivos compatibles con el estándar I2C. Usa operaciones de 100 kHz y 400 kHz, y puede utilizarse como maestra y esclava.
<b>El controlador de E / S (IOC)</b>	El controlador de E / S (IOC) controla los pines de E / S digitales y contiene circuitos multiplexores permitiendo que se asigne un conjunto de periféricos a los pines de E / S de manera flexible. Todas las E / S digitales tienen capacidad de interrupción y activación, tienen una función pullup y pulldown que se puede programar y pueden generar una interrupción en un borde positivo o negativo. Cuando se configura como una salida, los

---

pins pueden funcionar como push-pull u open-drain.

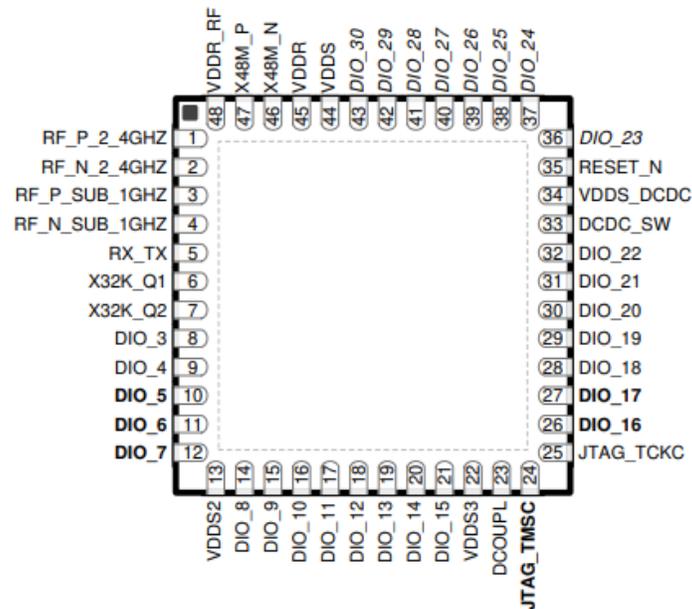
### Cinco GPIO

Cinco GPIO tienen capacidades de alta velocidad, que están marcadas en negrita en la Figura 64. Todos los periféricos digitales se pueden conectar a cualquier pin digital del dispositivo

(Instruments, 2019)

**Figura 66**

*Diagrama de Pines MCU CC1352R1*



Fuente: (Instruments, 2019)

### **c. Alimentación de Energía del Nodo Recolector**

Para que exista un correcto funcionamiento de los elementos que conforman en sistema embebido del Nodo Recolector se considera el consumo de energía de todos los elementos cada uno con sus procesos que permiten determinar en conjunto energía necesaria para que el sistema funcione de manera correcta.

Para el cálculo de la corriente total se hace uso de la Ecuación:

$$I_t = I_c * \sqrt{2} \quad \cdot \quad \text{Ec. 8}$$

Donde:

$I_t$ : corriente total

$I_c$ : corriente consumida

$\sqrt{2}$ : constante pitagórica.

Remplazando la ecuación por los valores obtenidos:

$I_c = 2,8363 \text{ A}$

$I_t = 4,011133927 \text{ A}$

El voltaje se escoge el mayor:

$V_t = 5\text{V}$

Potencia:  $P = 20,05566963 \text{ W}$

En la Tabla 26, se evidencia los parámetros de consumo de los elementos como son el voltaje, corriente además los procesos denominados de esa manera ya que representan a cada una de las entradas que posee cada dispositivo que conforma el sistema embebido. Se hace una sumatoria de los valores tanto en Amperios como en voltios y se saca el valor que corresponderán a la fuente de alimentación que va a permitir la alimentación general. De esa manera se tiene una fuente de de alimentación de 5V a 6 A.

**Tabla 26***Consumo de corriente de cada componente del sistema embebido*

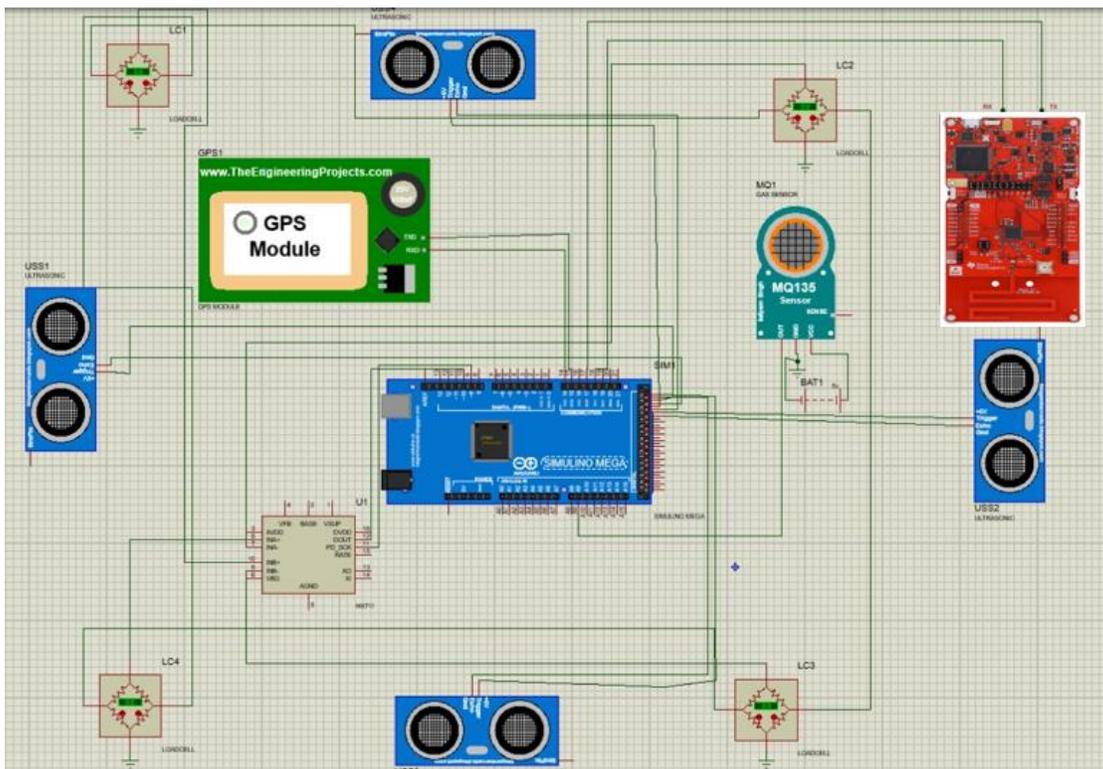
Componente	Unidades	Voltaje (V)	Consumo Max (A)	Procesos	Consumo Total (A)	Detalle
<b>Arduino Mega 2560</b>	1	5	0,04	23	<b>0,92</b>	DC Current per I/O Pin
		3,3	0,05	5	<b>0,25</b>	DC Current for 3.3V Pin
<b>Texas Ins. CC1352R1</b>	1	5	0,0069	2	<b>0,0138</b>	Active mode Rx 3,0V, 2,4GHz
		3,3	0,0071	1	<b>0,0071</b>	Active mode TX 3,0V, 2,4GHz
		3	0,0029	1	<b>0,0029</b>	Active mode MCU 48 MHz
		3,3	0,0096	2	<b>0,0192</b>	Radio transmit current 2.4 GHz PA (BLE)
		3,3	0,0249	2	<b>0,0498</b>	Radio transmit current Boost mode, Sub-1 GHz PA
		3,3	0,0223	5	<b>0,1115</b>	Radio transmit current Sub-1 GHz PA
<b>Modulo Gps Ublox Neo-6m-0-001</b>	1	5	0,05	1	<b>0,05</b>	DC current trough any digital I/O pin (except supplies)
		3,3	0,1	2	<b>0,2</b>	VCC_RF output current
		3,3	0,067	1	<b>0,067</b>	Max. supply current
<b>Sensor De Peso Galga 50Kg</b>	4	5	0,2	4	<b>0,8</b>	Resistivo
<b>Modulo Hx711 + Celda Carga</b>	1	5	0,05	1	<b>0,05</b>	Analog supply current (including regulator)
			0,035	1	<b>0,035</b>	Analog supply current Power down
<b>Sensor Ultrasonido HCSR04</b>	4	5	0,015	4	<b>0,06</b>	Corriente de trabajo
<b>Mg811 - Módulo Sensor De Dioxido De Carbono Co2</b>	1	6,0±0,1	0,2	1	<b>0,2</b>	Heating Current
					<b>2,8363</b>	

#### d. Diagrama de Conexión del Nodo Recolector

Como se muestra en la Figura 67, se tiene 4 sensores ultrasónicos, 4 celdas de carga para el peso y un módulo GPS los cuales se conectan a los pines digitales del microcontrolador Arduino Mega a excepción del sensor MQ-135 para medición de CO<sub>2</sub> que se conecta de manera analógica para la toma de datos. El Arduino a su vez se conecta a través de una conexión serial con el módulo CC1352R1 el cual podrá hacer envío de la información inalámbricamente a otro modulo CC1352R1.

**Figura 67**

*Diagrama de conexión del sistema embebido*



#### e. Configuración de los Nodos Recolectores

Para la toma de datos desde los sensores se hace uso de la placa Arduino Mega, que mediante el uso de pines específicos se realiza la lectura de los sensores de volumen, peso, co2 y el módulo GPS para establecer la ubicación del contenedor. De esa manera se tiene el código en placa Arduino detallado en el *ANEXO L*.

En Arduino se hace uso de las siguientes librerías:

### **Figura 68**

*Librerías usadas en el código de Arduino*

```
//////////Librería//////////  
#include "HX711.h"  
#include <TinyGPSPlus.h>  
#include <SoftwareSerial.h>  
#include <HCSR04.h>
```

Las cuales permiten trabajar con los sensores y módulos utilizados, así como también definir puertos en serie.

Los datos obtenidos mediante Arduino se envían a través de comunicación serial hacia el Nodo CC1352R1.

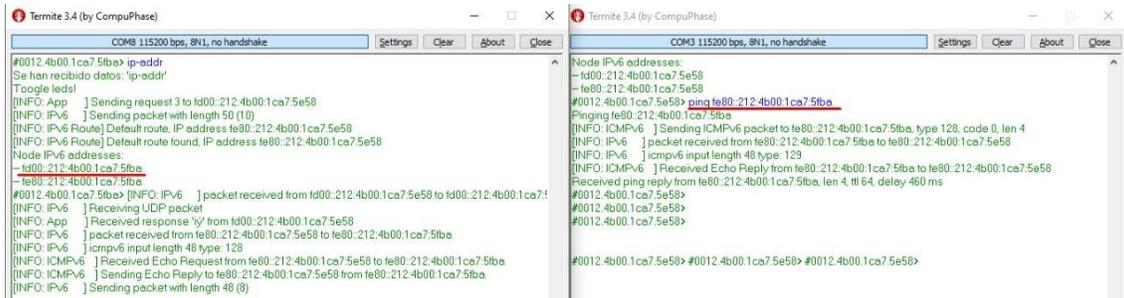
En el *ANEXO M* se tiene el código detallado del archivo Cliente UDP, el cual lo que hace es establecer la comunicación con el Servidor UPD, en contiki se denominará al Nodo recolector como cliente UDP y al Nodo Coordinador como servidor UDP o border router, el cliente UDP receipta los datos obtenidos mediante la placa Arduino por comunicación serial y luego los envía hacia el border router. Para saber que los datos están llegando a la placa, se activa el encendido de leds en la placa del nodo cliente.

### **Pruebas Iniciales**

En la Figura 69, se identifica la dirección IPV6 del cliente para poder verificar la conectividad mediante el comando **ip-addr**.

**Figura 69**

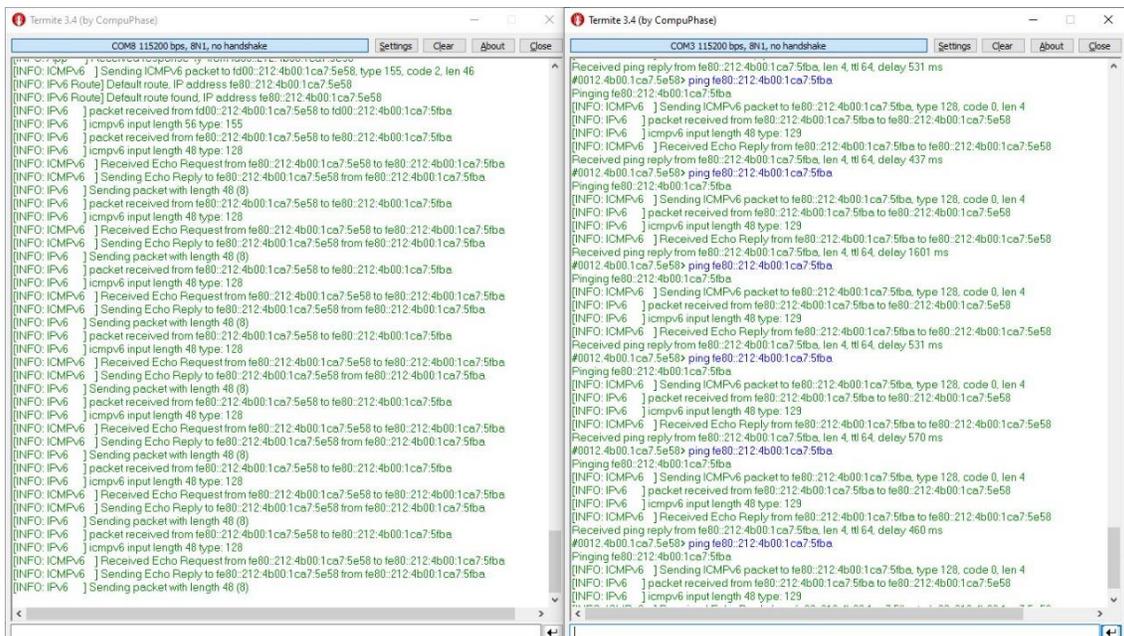
*Identificación de la dirección IPV6 del nodo cliente*



En la Figura 70 se puede evidenciar la conectividad mediante un ping entre el Cliente UDP y Servidor UDP.

**Figura 70**

*Establecimiento de conectividad entre Cliente UDP (derecha) y Servidor UDP (izquierda)*





hacia el bróker , en conjunto con el phyton bridge el cual permite tener la comunicación con la plataforma web.

Se asigna a la interfaz de la máquina virtual ens33 una dirección ipv6 global la cual va a estar combinada con un puerto para de esa manera cualquier mensaje que le llegue al border router va a ser reenviado hacia el phyton bridge en la dirección ipv6 global.

Luego se inicia el puente con el comando python3 nombre del archivo phyton y se asignan los parámetros como son la dirección ipv6 global y el puerto 4444.

### **g. Configuración de la Plataforma IOT**

Para la presentación de la información se tiene la Plataforma Web apegada a los requisitos establecidos, donde se tiene el valor de las variables, volumen, peso, CO2, además se presenta un mapa donde se puede evidenciar la ubicación del contenedor que se encuentra ocupado en un valor máximo de su capacidad. De esa manera el vehículo recolector podrá determinar la ruta más conveniente basada en los contenedores llenos. En la Figura 72, se muestra la interfaz en donde se observa los parámetros para poder determinar la ubicación de los contenedores en el mapa, así como también sus valores.

### **Figura 72**

*Interfaz gráfica de representación de datos al usuario final*



Al momento de realizar el envío mediante el simulador se puede evidenciar los valores que se han modificado de igual manera de visualiza en el mapa donde se encuentran los contenedores gracias a los datos de latitud y longitud.

Para poder representar los datos, se debe crear el objeto el cual va a tener como atributos las variables que se va a medir, en la Figura 73, se muestra la estructura de dichos objetos.

**Figura 73**

*Estructura de los objetos y sus variables a medir*

```

{} estructura.json > ...
1  {
2  |   "contenedores":
3  |   [
4  |     {
5  |       "name": "1",
6  |       "volumen": "2000",
7  |       "co2": "234",
8  |       "peso": "23",
9  |       "ubicacion": {
10 |         "lat": "0.3382770",
11 |         "long": "-78.1267512"
12 |       }
13 |     },
14 |     {
15 |       "name": "2",
16 |       "volumen": "12",
17 |       "co2": "33",
18 |       "peso": "144",
19 |       "ubicacion": {
20 |         "lat": "0.3380000",
21 |         "long": "-78.1267512"
22 |       }
23 |     }
24 |   ]
25 | }
26

```

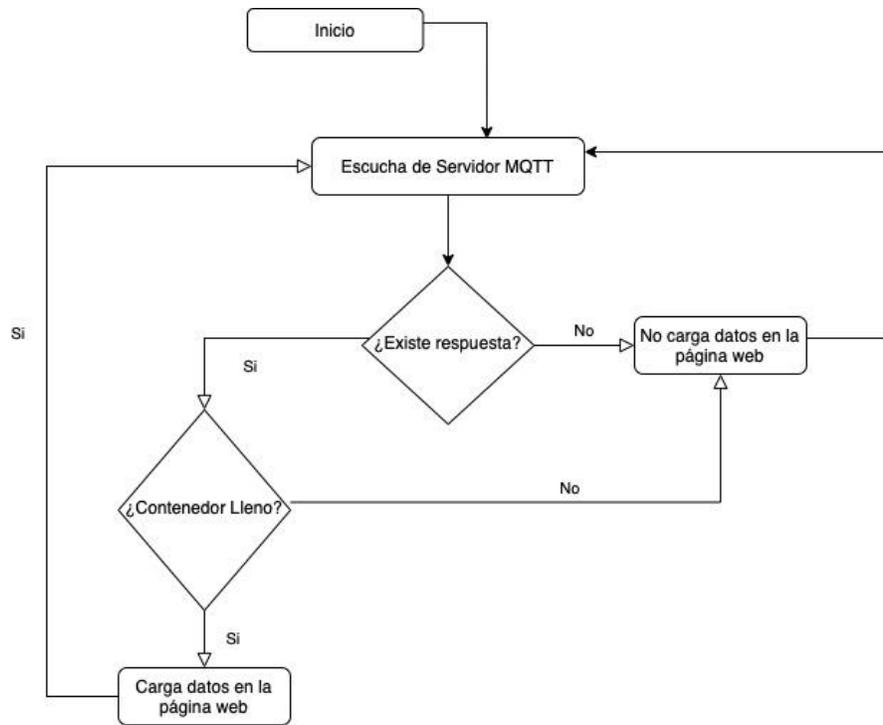
## h. Diagrama de flujo de la plataforma IOT

Para poder visualizar la ubicación de los contenedores es necesario tener conocimiento de cómo opera la plataforma IOT, en este caso se habilita la escucha del servidor MQTT para la recepción de datos desde el sistema embebido.

Donde existe un control desde el sistema embebido que me indica si es que el contenedor está lleno o no lo está. De esa manera si el contenedor está lleno se procede a cargar los datos en la plataforma de lo contrario seguirá en escucha hasta obtener el dato que se necesita.

### Figura 74

*Diagrama de Flujo del funcionamiento de la Plataforma IOT*



### i. Medición de variables

Mediante Arduino se hace la toma de valores de los sensores, el microcontrolador a su vez se comunica de manera serial con el Módulo CC1352R1 el cual hace el envío de la información al coordinador. En la Figura 75, se puede notar que la información de los sensores llega hasta el coordinador y de la misma manera hacia el python bridge.

**Figura 75**

*Datos obtenidos de los sensores en tiempo real*

```

contiki@ubuntu: ~/contiki-ng/examples/rpl-border-pers
[INFO: RPL_BR ] desde fd00::212:4b00:1ca7:5fba
[INFO: RPL_BR ] Enviando peticion hacia servidor UDP.
LEDS!
[INFO: RPL_BR ] Mensaje recibido: [{"contenedores":[{"name":"1","volumen":"64","co2":"82","peso":"0.02","ubicacion":{"lat":"0.338000","long":"-78.126751"},"state":"vacio"}]}] desde fd00::212:4b00:1ca7:5fba
[INFO: RPL_BR ] Enviando peticion hacia servidor UDP.
LEDS!
[INFO: RPL_BR ] Mensaje recibido: [{"contenedores":[{"name":"1","volumen":"63","co2":"82","peso":"2.35","ubicacion":{"lat":"0.338000","long":"-78.126751"},"state":"vacio"}]}] desde fd00::212:4b00:1ca7:5fba
[INFO: RPL_BR ] Enviando peticion hacia servidor UDP.
LEDS!
[INFO: RPL_BR ] Mensaje recibido: [{"contenedores":[{"name":"1","volumen":"64","co2":"82","peso":"5.10","ubicacion":{"lat":"0.338000","long":"-78.126751"},"state":"vacio"}]}] desde fd00::212:4b00:1ca7:5fba
[INFO: RPL_BR ] Enviando peticion hacia servidor UDP.
LEDS!
[INFO: RPL_BR ] Mensaje recibido: [{"contenedores":[{"name":"1","volumen":"62","co2":"84","peso":"4.02","ubicacion":{"lat":"0.338000","long":"-78.126751"},"state":"vacio"}]}] desde fd00::212:4b00:1ca7:5fba
[INFO: RPL_BR ] Enviando peticion hacia servidor UDP.
LEDS!
[INFO: RPL_BR ] Mensaje recibido: [{"contenedores":[{"name":"1","volumen":"62","co2":"84","peso":"4.02","ubicacion":{"lat":"0.338000","long":"-78.126751"},"state":"vacio"}]}] desde fd00::212:4b00:1ca7:5fba
[INFO: RPL_BR ] Enviando peticion hacia servidor UDP.
LEDS!

contiki@ubuntu: ~/python-bridge
{"lat":"0.338000","long":"-78.126751"},"state":"vacio"}}
Script ejecutado...
{"contenedores":[{"name":"1","volumen":"64","co2":"82","peso":"0.02","ubicacion":{"lat":"0.338000","long":"-78.126751"},"state":"vacio"}]}
{"lat":"0.338000","long":"-78.126751"},"state":"vacio"}}
{"contenedores":[{"name":"1","volumen":"61","co2":"82","peso":"2.35","ubicacion":{"lat":"0.338000","long":"-78.126751"},"state":"vacio"}]}
Script ejecutado...
{"contenedores":[{"name":"1","volumen":"61","co2":"82","peso":"2.35","ubicacion":{"lat":"0.338000","long":"-78.126751"},"state":"vacio"}]}
{"contenedores":[{"name":"1","volumen":"64","co2":"83","peso":"5.10","ubicacion":{"lat":"0.338000","long":"-78.126751"},"state":"vacio"}]}
{"contenedores":[{"name":"1","volumen":"62","co2":"84","peso":"4.02","ubicacion":{"lat":"0.338000","long":"-78.126751"},"state":"vacio"}]}
Script ejecutado...
{"contenedores":[{"name":"1","volumen":"62","co2":"84","peso":"4.02","ubicacion":{"lat":"0.338000","long":"-78.126751"},"state":"vacio"}]}

```

```

2022-12-29 15:54:12 Topic: testEc2021 QoS: 0
{"contenedores": [{"name": "1", "volumen": "62", "co2": "84", "peso": "4.02", "ubicacion": {"lat": "0.338000", "long": "-78.126751", "state": "vacio"}}]}

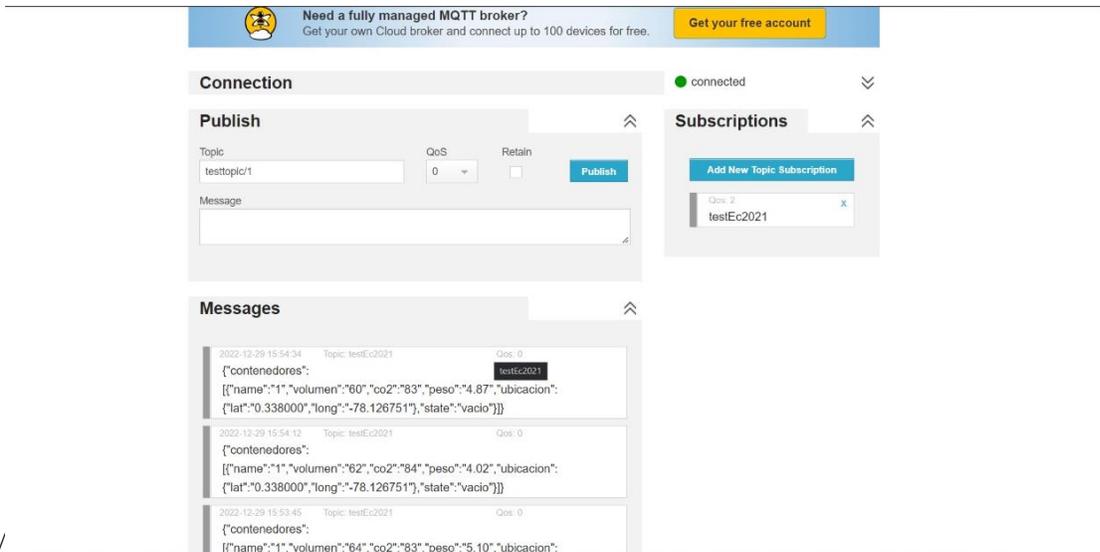
2022-12-29 15:53:45 Topic: testEc2021 QoS: 0
{"contenedores": [{"name": "1", "volumen": "64", "co2": "83", "peso": "5.10", "ubicacion": {"lat": "0.338000", "long": "-78.126751", "state": "vacio"}}]}

2022-12-29 15:53:25 Topic: testEc2021 QoS: 0
{"contenedores": [{"name": "1", "volumen": "61", "co2": "82", "peso": "2.35", "ubicacion": {"lat": "0.338000", "long": "-78.126751", "state": "vacio"}}]}

```

De igual manera se debe crear un usuario para el Cliente MQTT, en este caso se opta por la plataforma Hive MQTT que permite la creación del Broker el cual va a recibir los datos provenientes desde el coordinador, los cuales en primera instancia fueron obtenidos desde el nodo recolector. En la Figura 76, se muestra los mensajes que provienen desde los sensores.

**Figura 76**  
*Plataforma para cliente MQTT Hive MQTT Broker*



Por último, se puede verificar que los datos llegan a la plataforma Web donde se tiene los valores reflejados de manera gráfica, de igual manera se tiene la ubicación en el mapa de la posición del contenedor como se ilustra en la Figura 77.

**Figura 77**

*Plataforma Iot para la visualización de datos para el usuario final*

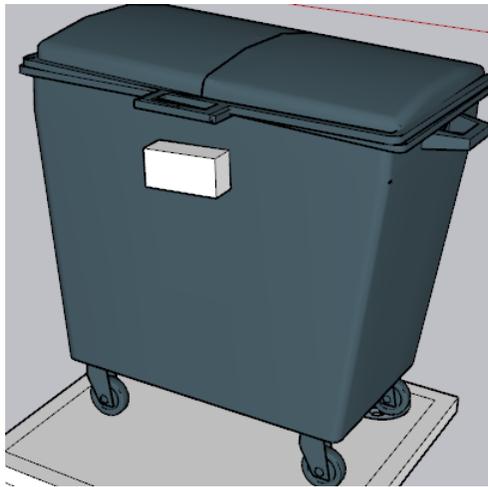


## **j. Ubicación del Sistema Embebido**

Para que exista la protección adecuada de los sensores, se opta por la utilización de una caja que evita que el sistema embebido tenga contacto con algún factor externo tales como polvo, humedad que comúnmente ocasionan deterioro a los elementos.

### **Figura 78**

*Ubicación del sistema en el contenedor de residuos sólidos*



### **3.1.2.11. Diseño Fase 2**

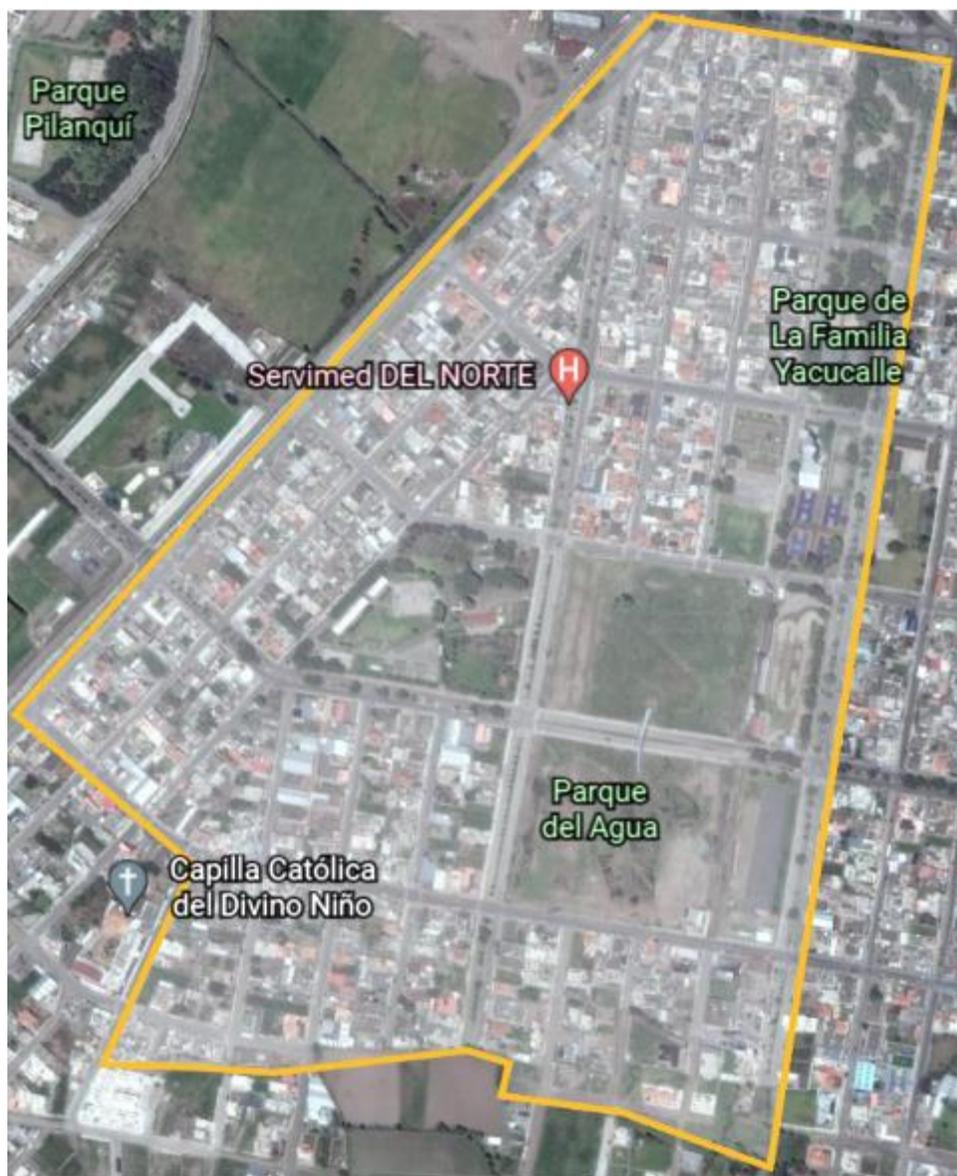
La fase 2 constituye una red tipo malla donde entran en funcionamiento los nodos tanto recolectores como coordinador, todos con tecnología IEEE 802.15.4g los cuales estarán conectados inalámbricamente, el nodo coordinador ejercerá control de la red mesh así como también recepta la información originada en los contenedores y la envía a una plataforma IOT mediante el protocolo MQTT.

**a. Diseño de la comunicación Inalámbrica.**

El barrio Yacucalle pertenece a la parroquia San Francisco del cantón Ibarra y está ubicado en las siguientes coordenadas Latitud: 0.333333, Longitud: -78.1167, y con una superficie aproximada de 537.874,48 m<sup>2</sup>. Se trata de una zona urbana con poca vegetación.

**Figura 79**

*Delimitación sector Yacucalle*



Para la ubicación y configuración de los nodos router se considera la ubicación real de los contenedores que albergarán a los nodos recolectores, de esa manera se realizó un recorrido por el sector donde se tienen 38 contenedores ubicados en diferentes puntos, además se agrega una etiqueta a cada contenedor con el fin de diferenciarlo del resto como se muestra en la Figura 80.

### Figura 80

*Ubicación real de los contenedores de residuos sólidos*



## **b. Análisis Inalámbrico**

Para el presente análisis se considera los factores involucrados en un enlace de comunicación inalámbrica, así como también las especificaciones de los equipos que se utilizan al momento de establecer la comunicación. En la Figura 81, se presenta una tentativa de topología mesh en la cual se tiene el nodo coordinador y nodos recolectores.

Para establecer los enlaces presentes en la red malla se hace referencia a la Ecuación 9.

$$e = \frac{n(n-1)}{2} \quad \text{Ec. 9}$$

Donde:

e: número de enlaces

n: número de nodos.

Entonces al hacer el remplazo de los datos se tiene el siguiente resultado:

$$e = \frac{3(3 - 1)}{2}$$
$$e = 3$$

### **Figura 81**

*Red Mesh en base a ubicación de contenedores*



### c. Pérdidas en el Espacio Libre

Al momento de establecer la comunicación entre transmisor y receptor se toma en cuenta las pérdidas que están determinadas por factores tales como la presencia de objetos que evitan que exista línea de vista, frecuencia a las que se está trabajando, así como también la potencia de los equipos y la sensibilidad y ganancia de las antenas.

**Tabla 27**

*Coordenadas Nodos red Mesh*

NODO	LATITUD	LONGITUD	ALTURA (m)
C1	0,342914	-78,1258	2213
C8	0,338081	-78,13025	2215
CN	0,335456	-78,12639	2222

Tomando la información recolectada con respecto a la ubicación de los contenedores se realiza el cálculo de la distancia entre cada uno de ellos haciendo uso de las coordenadas descritas en la tabla 22 y la Ecuación 10.

$$d = \sqrt{[111.32 * (LatA - latB)]^2 + [111.32 * (LonA - LonB)]^2 + \left(\frac{HA-HB}{1000}\right)^2} \quad Ec. 10$$

Donde:

d = Longitud del trayecto radioeléctrico [ km]

LatA = Latitud A

LatB = Latitud B

LoA = Longitud A

LoB = Longitud B

HA = Altitud A

HB = Altitud B

En base a la fórmula general del cálculo de la distancia se tiene los datos presentes en la tabla 28:

**Tabla 28**

*Distancia de enlaces*

<b>Punto Inicial – Punto Final</b>	<b>Distancia (m)</b>	<b>Distancia (km)</b>
<b>C1-C8</b>	731	0.7313
<b>C8-NC</b>	519	0.5196
<b>C1-NC</b>	832	0.8328

### **Atenuación en el Espacio Libre Enlace 1**

Para el cálculo de la Atenuación en espacio libre se hace uso de la Ecuación 11.

$$Lp = 32.45 + 20 \log(f) + 20 \log(d) \quad Ec. 11$$

Lp=Perdidas en el espacio libre (dB)

K=32.45, cte de atenuación

F=Frecuencia (MHz)

d= Distancia (km)

**Para el caso de la frecuencia 915 MHz, y considerando el primer enlace se tiene:**

$$Lp = 32.45 + 20 \log(915) + 20 \log(0.7313)$$

$$Lp = 88.96 \text{ dB}$$

Pérdidas totales considerando factores ambientales:

*Pérdida en el espacio libre + pérdidas debido a la lluvia (Losses rain) + neblina (Los fog) y gases atmosféricos (losses as).*

$$Lr = \frac{1\text{dB}}{\text{KM}} * 0.7313 = 0.7313 \text{ [dB]}$$

$$Lf = 0.01 \frac{1\text{dB}}{\text{KM}} * 0.7313 = 0.007313 \text{ [dB]}$$

$$Lg = 0.01 \frac{1\text{dB}}{\text{KM}} * 0.7313 = 0.007313 \text{ [dB]}$$

$$Lrfg = 0.745926 \text{ [dB]}$$

De esa manera se puede tener un total de perdidas añadiendo el valor de las perdidas por espacio libre calculado anteriormente, lo cual se resume en la Ecuación 12.

$$LT = Lp + Lrfg = 88.96 + 0.745926 = 89.70 \text{ [dB]} \quad \text{Ec. 12}$$

### **Ángulo de azimuth**

Es aquel que indica la orientación en el que se debe fijar la antena con respecto al plano Horizontal, y es representado por la Ecuación 13.

$$\text{azimut} = \tan^{-1} \theta \quad \text{Ec. 13}$$

En la Ecuación 14,  $\theta$  se define como la variación tanto de longitud sobre la variación de latitud entre los puntos inicial y final.

$$\theta = \frac{\text{lon}A - \text{lon}B}{\text{lat}A - \text{lat}B} \quad \text{Ec. 14}$$

Haciendo el cálculo para el primer enlace se tiene:

$$\theta = \frac{-78,1258 + 78,13025}{0,342914 - 0,338081}$$

$$\theta = 0.920$$

$$\text{azimut} = \tan^{-1}(0.953286)$$

$$\text{azimut} = 223.63$$

### Ángulo de Elevación

Corresponde a la inclinación de acuerdo con el plano vertical y se calcula mediante la Ecuación 15.

$$\text{elevación} = \tan^{-1} \alpha \quad \text{Ec. 15}$$

Donde para el cálculo de  $\alpha$  se lo realiza mediante la Ecuación 16.

$$\alpha = \frac{HA - HB}{d(m)} \quad \text{Ec. 16}$$

Remplazando valores del primero enlace se tiene:

$$\alpha = \frac{2213 - 2215}{731}$$

$$\text{elevación} = \tan^{-1}(-2.735 \times 10^{-3})$$

$$\text{elevación} = -0.156^\circ$$

### Atenuación en el Espacio Libre Enlace 2

$$L_p = 32.45 + 20 \log(f) + 20 \log(d)$$

$L_p$  = Pérdidas en el espacio libre (dB)

$K$  = 32.45, cte de atenuación

$F$  = Frecuencia (MHz)

d= Distancia (km)

**Para el caso de la frecuencia 915 MHz, y considerando el primer enlace se tiene:**

$$Lp = 32.45 + 20 \log(915) + 20 \log(0.05196)$$

$$Lp = 66.99 \text{ dB}$$

Pérdidas totales considerando factores ambientales:

*Pérdida en el espacio libre + pérdidas debido a la lluvia (Losses rain) + neblina (Los fog) y gases atmosféricos (losses as).*

$$Lr = \frac{1dB}{KM} * 0.05196 = 0.05196 \text{ [dB]}$$

$$Lf = 0.01 \frac{1dB}{KM} * 0.05196 = 0.0005196 \text{ [dB]}$$

$$Lg = 0.01 \frac{1dB}{KM} * 0.05196 = 0.0005196 \text{ [dB]}$$

$$Lrfg = 0.0529992 \text{ [dB]}$$

De esa manera se puede tener un total de perdidas añadiendo el valor de las perdidas por espacio libre calculado anteriormente.

$$LT = Lp + Lrfg = 69.99 + 0.0529992 = 67.04 \text{ [dB]}$$

### **Ángulo de azimuth**

Es aquel que indica la orientación en el que se debe fijar la antena con respecto al plano

horizontal.

$$\text{azimut} = \tan^{-1} \theta$$

Donde  $\theta$  se define como la variación tanto de longitud sobre la variación de latitud entre los puntos inicial y final.

$$\theta = \frac{\text{lonA} - \text{lonB}}{\text{latA} - \text{latB}}$$

Haciendo el cálculo para el segundo enlace se tiene:

$$\theta = \frac{-78,13025 + 78.12639}{0,338081 - 0.335456}$$

$$\theta = -1.470$$

$$\text{azimut} = \tan^{-1}(-1.470)$$

$$\text{azimut} = 123,78$$

### Ángulo de Elevación

Corresponde a la inclinación de acuerdo con el plano vertical

$$\text{elevación} = \tan^{-1} \alpha$$

$$\alpha = \frac{HA - HB}{d(m)}$$

Remplazando valores del primero enlace se tiene:

$$\alpha = \frac{2215 - 2222}{519}$$

$$\text{elevación} = -0.013487475$$

$$\text{elevación} = -0.772^\circ$$

### Atenuación en el Espacio Libre Enlace 3

$$L_p = 32.45 + 20 \log(f) + 20 \log(d)$$

$L_p$ =Perdidas en el espacio libre (dB)

$K=32.45$ , cte de atenuación

$F$ =Frecuencia (MHz)

$d$ = Distancia (km)

**Para el caso de la frecuencia 915 MHz, y considerando el primer enlace se tiene:**

$$Lp = 32.45 + 20 \log(915) + 20 \log(0.8328)$$

$$Lp = 90.08 \text{ dB}$$

Pérdidas totales considerando factores ambientales:

*Pérdida en el espacio libre + pérdidas debido a la lluvia (Losses rain) + neblina (Los fog) y gases atmosféricos (losses as).*

$$Lr = \frac{1\text{dB}}{\text{KM}} * 0.8328 = 0.8328 \text{ [dB]}$$

$$Lf = 0.01 \frac{1\text{dB}}{\text{KM}} * 0.8328 = 0.008328 \text{ [dB]}$$

$$Lg = 0.01 \frac{1\text{dB}}{\text{KM}} * 0.8328 = 0.008328 \text{ [dB]}$$

$$Lrfg = 0.849456 \text{ [dB]}$$

De esa manera se puede tener un total de pérdidas añadiendo el valor de las pérdidas por espacio libre calculado anteriormente.

$$LT = Lp + Lrfg = 90.08 + 0.849456 = 90.92 \text{ [dB]}$$

### **Ángulo de azimuth**

Es aquel que indica la orientación en el que se debe fijar la antena con respecto al plano horizontal.

$$\text{azimut} = \tan^{-1} \theta$$

Donde  $\theta$  se define como la variación tanto de longitud sobre la variación de latitud entre los puntos inicial y final.

$$\theta = \frac{\text{lonA} - \text{lonB}}{\text{latA} - \text{latB}}$$

Haciendo el cálculo para el tercer enlace se tiene:

$$\theta = \frac{-78,1258 + 78,12639}{0,342914 - 0,335456}$$

$$\theta = 0.07910$$

$$azimut = \tan^{-1}(0.07910)$$

$$azimut = 182,52^\circ$$

### Ángulo de Elevación

Corresponde a la inclinación de acuerdo con el plano vertical

$$elevación = \tan^{-1} \alpha$$

$$\alpha = \frac{HA - HB}{d(m)}$$

Remplazando valores del primero enlace se tiene:

$$\alpha = \frac{2213 - 2222}{832}$$

$$elevación = \tan^{-1} -0.0108$$

$$elevación = -0.619^\circ$$

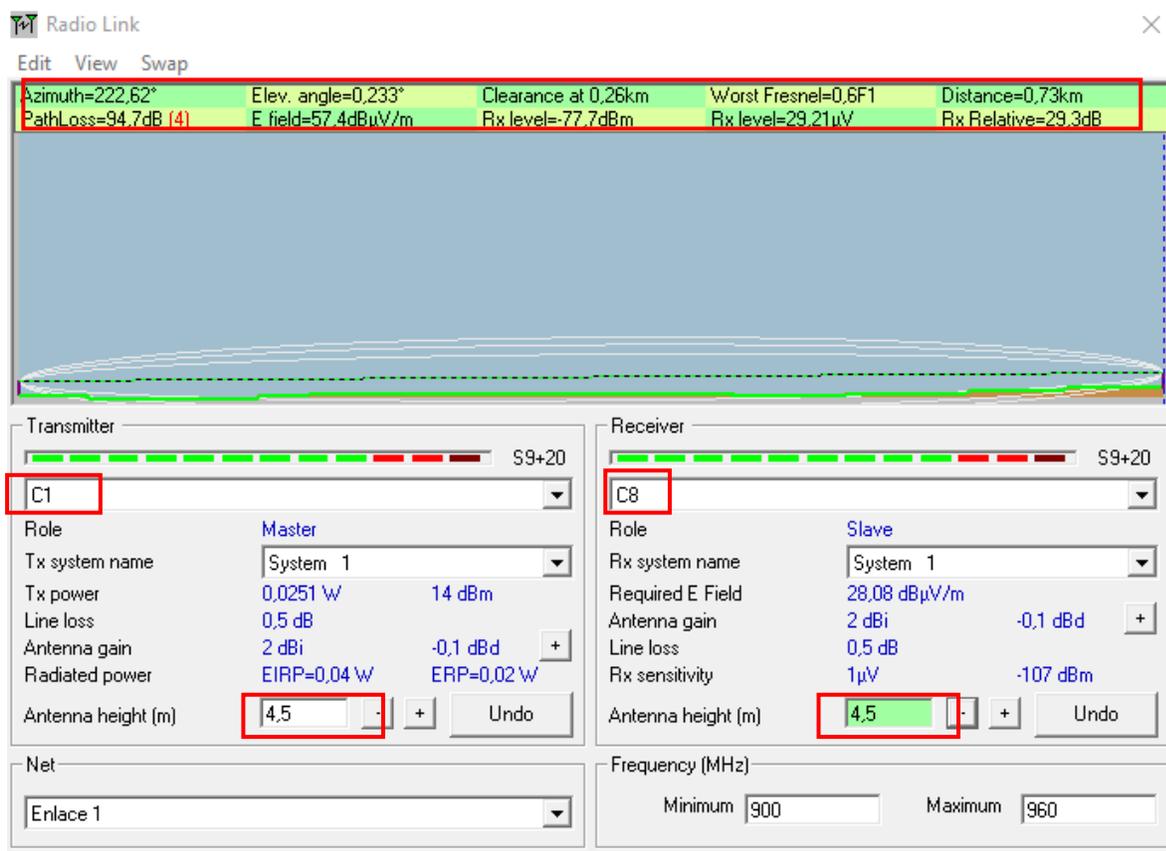
#### d. Cálculo de parámetros haciendo uso de Software de Simulación Radio Mobile

##### Enlace 1

En la Figura 82, se tiene la simulación del enlace entre los nodos clientes, donde se tiene los parámetros señalados del enlace, de igual manera la altura que se tiene de los nodos para este caso 4,5m en la cual se tiene una zona de fresnel libre e idónea para que se tenga un enlace estable.

##### Figura 82

*Simulación de Enlace de nodos C1-C2*



De igual manera se hace la presentación del enlace en radio mobile para poder verificar que la zona de fresnel del enlace se encuentra despejada, y evidenciar que el enlace se encuentra establecido en las coordenadas señaladas para el caso de estudio. Lo mencionado se indica que la

**Figura 83**

*Simulación enlace de nodos C1-C2*

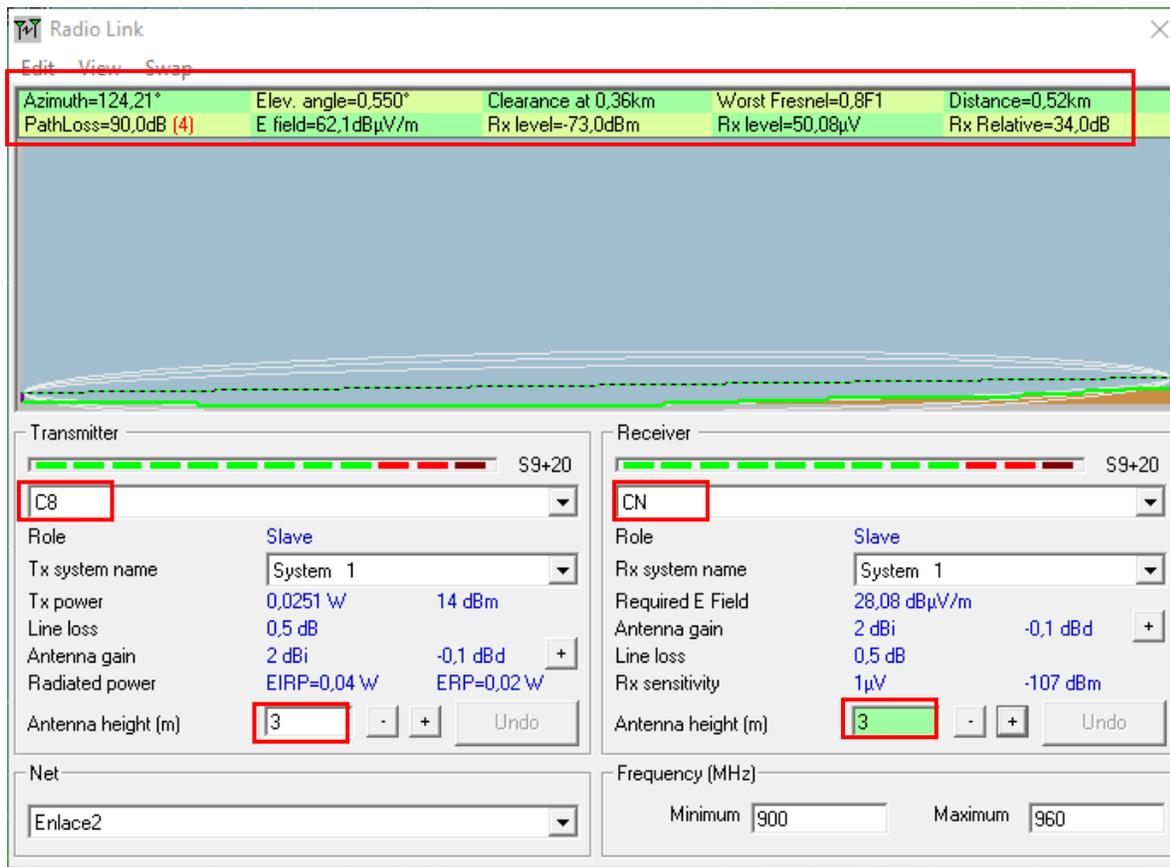


## Enlace 2

Para el segundo enlace se tiene el nodo cliente C1 y el nodo coordinador CN, los nodos se encuentran a una distancia de 0,36 km, además se presenta algunos parámetros relevantes dentro de un enlace, de igual manera la altura a la que están cada nodo.

### Figura 84

Simulación de Enlace de nodos C1-NC



Para el caso de la simulación en google earth se tiene el enlace entre los nodos coordinador y el nodo cliente 1 en esta ocasión, donde se puede verificar que la zona de fresnel se encuentra despejada a una distancia 0,36km. Se puede verificar en la Figura 85, lo mencionado anteriormente.

### Figura 85

### *Simulación de Enlace de nodos C1-NC*

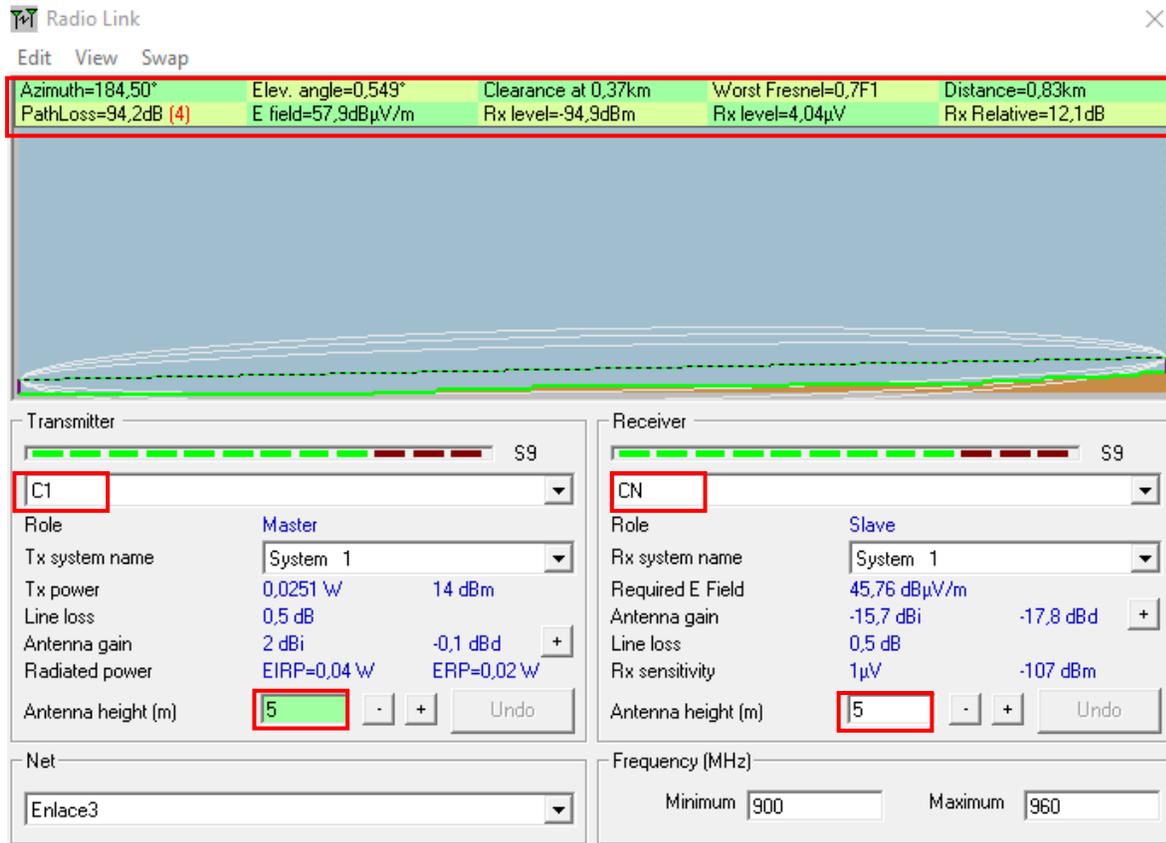


### **Enlace 3**

Por último, el enlace final está conformado por el coordinador y el nodo cliente 2, donde al igual que lo mencionado anteriormente en la simulación de radio mobile, se tiene los parámetros de la comunicación entre dichos nodos, al igual que la altura de ubicación de los nodos. En la Figura 86, se puede evidenciar lo descrito.

### **Figura 86**

*Simulación de Enlace de nodos C2-NC*



Al igual que las gráficas anteriores en la Figura 87, se muestra el enlace establecido del nodo coordinador y de nodo cliente 2, donde se puede visualizar que se tiene un enlace estable, que no afecta a zona de fresnel.

**Figura 87**

*Simulación de Enlace de nodos C2- NC*



Considerando los valores obtenidos se puede determinar que la altura de las antenas influye en gran mayoría para tener un enlace favorable lo que se pretende de esa manera es colocar los puntos tanto transmisión y recepción a una altura mayor.

**Tabla 29**

*Comparativas de valores calculados y valores simulados de los enlaces*

PARAMETRO	CALCULADO	SIMULADO
<b>ENLACE 1</b>		
Path Loss	89,70dB	97.7dB
Angulo Azimut	223.63°	222,62
Angulo Elevación	0,1560	0.233
<b>ENLACE 2</b>		
Path Loss	67,04dB	90,0dB
Angulo Azimut	123,78	124,21
Angulo Elevación	0,772°	0.550
<b>ENLACE 3</b>		
Path Loss	90,92 dB	94.2 dB

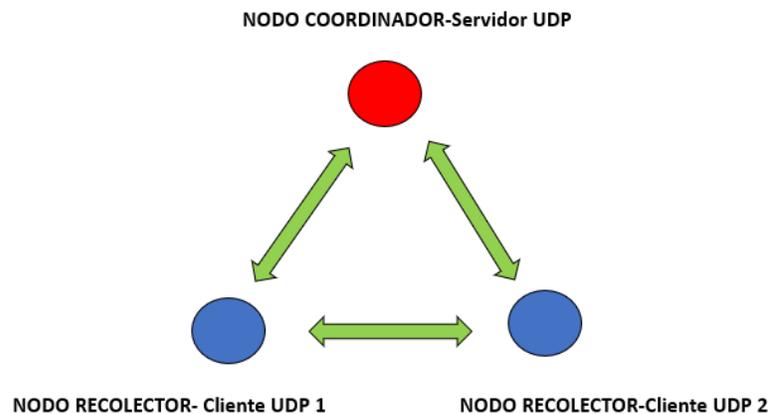
<b>Angulo Azimut</b>	182,52°	184,50
<b>Angulo Elevación</b>	0,619	0.549

### e. Construcción de la red Mesh

La red mesh se construye mediante dos clientes UDP los cuales se los ha venido conociendo como nodos recolectores y un Servidor UDP al cual hemos llamado coordinador dentro de la Mesh, al converger la red el protocolo RPL permite que se conozcan los vecinos y se establezcan las rutas posibles para el caso de comunicación entre clientes, y el servidor UDP.

#### Figura 88

*Red Mesh, clientes y nodo coordinador*



En contiki se hace uso de las siguientes librerías indicadas en la Figura 89, para obtener las métricas que van a permitir evaluar la red malla como son PDR, PER, RSSI, LQI y retardo. Permitiendo de esa manera que se tenga información sobre el enlace establecido, y poder verificar que tan eficiente es la red.

## Figura 89

*Librerías utilizadas en contiki*

```
58  /*----- PRUEBAS -----*/
59  #include "net/ipv6/uip.h"
60  #include "net/ipv6/uip-icmp6.h"
61  #include "net/ipv6/sicslowpan.h"
62  #define MODO_METRICAS 1
63
```

En la Figura 90, se tiene establecido las variables en las cuales se van a ir almacenando los valores que se obtienen de cada métrica en tiempo real.

## Figura 90

*Variables a almacenar las métricas*

```
73  static int def_rt_rssi = 0;
74  static int rx_count = 1;
```

En la Figura 91, se tiene el código para la impresión de los valores, tanto del RSSI, LQI y Retardo, para el caso de los PDR, PER se hace un análisis del número de paquetes enviados y recibidos es decir que cada 20 paquetes enviados se tendría un resumen de cuantos llegaron a su destino y cuantos se perdieron.

## Figura 91

*Código de impresión de métricas*

```

92  #if MODO_METRICAS
93  /*----- RSSI -----*/
94  def_rt_rssi = (int16_t)uipbuf_get_attr(UIPBUF_ATTR_RSSI);
95  printf("\nRSSI (dBm)\":%d", def_rt_rssi);
96  /*-----*/
97
98  /*----- LQI -----*/
99  printf("\nLQI %u",uipbuf_get_attr(UIPBUF_ATTR_LINK_QUALITY));
100 /*-----*/
101
102 /*----- DELAY -----*/
103 printf("\nTiempo de respuesta (sec):%lu\n", clock_seconds());
104 /*-----*/
105 #endif

```

En la Figura 92, se muestra la configuración de la respuesta del ACK que se va a reflejar en el lado del nodo coordinador, ya que dicha respuesta va a ser enviada por parte del nodo cliente hacia el nodo coordinador.

### Figura 92

*Código de respuesta de ACK desde el nodo cliente*

```

121  char ACK[32];
122  sprintf(ACK, "ACK N. %d", rx_count);
123  simple_udp_sendto(&udp_conn_client1, ACK, strlen(ACK), sender_addr);
124  rx_count++;
125  }

```

Para las métricas de PER y PDR se tiene la impresión de las estadísticas como se ilustra en la Figura 93, se hace un resumen del número de paquetes que se han enviado de manera exitosa, así como los que se han perdido, lo que permite calcular el desempeño de estas métricas en base a los paquetes que se han entregado o no.

### Figura 93

*Impresión valores PDR y PER*

```
148 /* Print statistics every 10th TX */
149 if(tx_count % 10 == 0) {
150     LOG_INFO("=====");
151     LOG_INFO("Paquetes Transmidos: %d PRIu32 "Paquetes Recibidos: %d PRIu32 "Paquetes Perdidos: %d PRIu32 "\n",
152             tx_count, rx_count, missed_tx_count);
153
154     printf("\nPER: %d PRIu32 ", missed_tx_count);
155
156     printf("\nPKR: %d PRIu32 "\n", rx_count/tx_count);
157 }
```

Para poder evidenciar las métricas se tiene dos nodos de los cuales el de la izquierda es el nodo cliente y el de la derecha es el nodo coordinador en RPL conocido como DAG root. Se envía los mensajes cada 10 segundos, cabe mencionar que son paquetes de 8 bytes

El nodo cliente envía cada 10 segundos los paquetes, el nodo coordinador, guarda los datos necesarios, y responde con un ACK, de igual manera tendrá las métricas correspondientes a los ACK.

En la Figura 94, se puede verificar la dirección tanto de origen **fd00::212:4b00:1ca7:5fba** como de destino **fd00::212:4b00:1ca7:5e58**

### Figura 94

*Verificación de direcciones nodo coordinador derecha y nodo cliente izquierda*

```

*Tiempo actual: 21908 [ticks]
*Latencia: 22 [ticks]
*RSSI: -20 [dBm]
*LQI: 0
*Latencia convertida: 171 [ms]
[INFO: App ] Enviando paquete con timestamp 23053 to fd00::212:4b00:1ca7:5e58
[INFO: App ]
[INFO: App ]
*ACK recibido de: fd00::212:4b00:1ca7:5e58[INFO: App ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 23053 [ticks]
*Tiempo actual: 23075 [ticks]
*Latencia: 22 [ticks]
*RSSI: -18 [dBm]
*LQI: 0
*Latencia convertida: 171 [ms]
[INFO: App ] Enviando paquete con timestamp 24403 to fd00::212:4b00:1ca7:5e58
[INFO: App ]
[INFO: App ]
*ACK recibido de: fd00::212:4b00:1ca7:5e58[INFO: App ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 24403 [ticks]
*Tiempo actual: 24427 [ticks]
*Latencia: 24 [ticks]
*RSSI: -21 [dBm]
*LQI: 0
*Latencia convertida: 187 [ms]
[INFO: App ] Enviando paquete con timestamp 25777 to fd00::212:4b00:1ca7:5e58
[INFO: App ]
[INFO: App ]
*ACK recibido de: fd00::212:4b00:1ca7:5e58[INFO: App ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 25777 [ticks]
*Tiempo actual: 25794 [ticks]
*Latencia: 17 [ticks]
*RSSI: -21 [dBm]
*LQI: 0
*Latencia convertida: 132 [ms]

*Paquete recibido de: fd00::212:4b00:1ca7:5fba[INFO: App ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 15428 [ticks]
*Tiempo actual: 15440 [ticks]
*Latencia: 12 [ticks]
*RSSI: -18 [dBm]
*LQI: 0
*Latencia convertida: 93 [ms]
[INFO: App ]
[INFO: App ]
*Paquete recibido de: fd00::212:4b00:1ca7:5fba[INFO: App ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 16713 [ticks]
*Tiempo actual: 16745 [ticks]
*Latencia: 32 [ticks]
*RSSI: -18 [dBm]
*LQI: 0
*Latencia convertida: 250 [ms]
[INFO: App ]
[INFO: App ]
*Paquete recibido de: fd00::212:4b00:1ca7:5fba[INFO: App ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 17939 [ticks]
*Tiempo actual: 17944 [ticks]
*Latencia: 5 [ticks]
*RSSI: -18 [dBm]
*LQI: 0
*Latencia convertida: 39 [ms]
[INFO: App ]
[INFO: App ]
*Paquete recibido de: fd00::212:4b00:1ca7:5fba[INFO: App ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 19227 [ticks]
*Tiempo actual: 19234 [ticks]
*Latencia: 7 [ticks]
*RSSI: -21 [dBm]
*LQI: 0
*Latencia convertida: 54 [ms]
[INFO: App ]

```

Las placas, manejan un tiempo de sincronización los cuales se determinan por los llamados ticks que son una medida del tiempo de sincronización entre nodos, o una unidad de tiempo entre los nodos, con los cuales a partir de la unidad se determina la latencia en ms. Es importante conocer que 128 ticks son equivalentes a 1 segundo. En la Figura 95, se tiene la información con cada métrica que se va a mostrar en cada envío de paquete.

**Figura 95**

*Información de métricas de un paquete de 8 bytes enviado desde el cliente al servidor UDP*

```

[INFO: App ]
*Paquete recibido de: fd00::212:4b00:1ca7:5fba[INFO: App ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 9665 [ticks]
*Tiempo actual: 9678 [ticks]
*Latencia: 13 [ticks]
*RSSI: -14 [dBm]
*LQI: 0
*Latencia convertida: 101 [ms]
[INFO: App ]

```

Para el cálculo de la latencia, se hace el siguiente cálculo tomando un ejemplo al azar de un paquete enviado. Ejemplo el valor de tiempo creado 24403(ticks)- 24427 (ticks) del tiempo actual, da un valor de 24 (ticks) da el valor de latencia en ticks, la cual mediante las formula, se divide 24 (ticks) sobre 128 y se multiplica por 1000 para tener un valor de latencia en milisegundo para este caso sería igual a 187.

En la Figura 96, se tiene el valor de CLOCK SECOND, cada que se reinicie el servidor muestra el valor.

**Figura 96**

*Valor correspondiente a CLOCK SECONDS*



```
Disconnected - click to connect. Settings Clear About Close
[INFO: Main ] Starting Contiki-NG-release/v4.8-305-g2998a0f5f-dirty
[INFO: Main ] - Routing: RPL Lite
[INFO: Main ] - Net: sicslowpan
[INFO: Main ] - MAC: TSCH
[INFO: Main ] - 802.15.4 PANID: 0x81e5
[INFO: Main ] - 802.15.4 TSCH default hopping sequence length: 4
[INFO: Main ] Node ID: 24152
[INFO: Main ] Link-layer address: 0012.4b00.1ca7.5e58
[INFO: Main ] Tentative link-local IPv6 address: fe80::212:4b00:1ca7:5e58
[INFO: CC13xx/CC26xx] Operating frequency on Sub-1 GHz
[INFO: CC13xx/CC26xx] RF: Channel 15
[INFO: App ]
CLOCK_SECOND: 128
[INFO: Main ] Starting Contiki-NG-release/v4.8-305-g2998a0f5f-dirty
[INFO: Main ] - Routing: RPL Lite
[INFO: Main ] - Net: sicslowpan
[INFO: Main ] - MAC: TSCH
[INFO: Main ] - 802.15.4 PANID: 0x81e5
[INFO: Main ] - 802.15.4 TSCH default hopping sequence length: 4
[INFO: Main ] Node ID: 24152
[INFO: Main ] Link-layer address: 0012.4b00.1ca7.5e58
[INFO: Main ] Tentative link-local IPv6 address: fe80::212:4b00:1ca7:5e58
[INFO: CC13xx/CC26xx] Operating frequency on Sub-1 GHz
[INFO: CC13xx/CC26xx] RF: Channel 15
[INFO: App ]
CLOCK_SECOND: 128
[INFO: App ]
[INFO: App ]
*Paquete recibido de: fd00::212:4b00:1ca7:5fba[INFO: App ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 3429 [ticks]
*Tiempo actual: 3441 [ticks]
*Latencia: 12 [ticks]
*RSSI: -19 [dBm]
*LOI: 0
*Latencia convertida: 93 [ms]
[INFO: App ]
[INFO: App ]
*Paquete recibido de: fd00::212:4b00:1ca7:5fba[INFO: App ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 3370 [ticks]
```

## **4. Capítulo IV: Implementación y Pruebas de funcionamiento**

Párrafo, de lo que se aborda en este capítulo

El Capítulo IV aborda la implementación del sistema en los contenedores de residuos sólidos y se realizan las respectivas pruebas de funcionamiento tanto del nodo recolector, así como también la funcionalidad de la red mesh, de igual manera se hace una simulación donde se puede verificar el funcionamiento de la red mesh con el número real de contenedores que se encuentran en el lugar de estudio. Además, se incluye las métricas de evaluación del desempeño de la red mesh y por último se indican los beneficios que conlleva a la aplicación del sistema en una localidad cualquiera como apartado final del capítulo.

### **4.1. Implementación**

En el apartado de la Implementación se hace la instalación de los nodos recolectores y del nodo coordinador, de igual manera se hace una simulación que permite evidenciar la funcionalidad de la red mesh para dos nodos recolectores y un nodo coordinador con la finalidad de hacer una explicación de como se hace el envío de los datos desde el nodo recolector al nodo coordinador De igual manera se maneja el escenario donde existen los 38 contenedores que hacen referencia al número total de tachos presentes en el sector Yacucalle.

#### 4.1.1. Instalación del Nodo Recolector

Para la instalación es necesario asegurar la protección contra polvo, humedad o cualquier elemento ambiental que pueda causar deterioro en los dispositivos empleados. Se coloca los dispositivos cada uno con una carcasa tanto para el Microcontrolador Arduino, así como también para el módulo CC1352R1, los cuales a su vez están protegido en la caja de protección, así como también se tiene una fuente de 5V conectada para su respectiva alimentación. El montaje se evidencia en la Figura 97.

#### Figura 97

*Montaje de los elementos del sistema embebido para la toma de datos*



#### 4.1.2. Ubicación y calibración de sensores en el contenedor

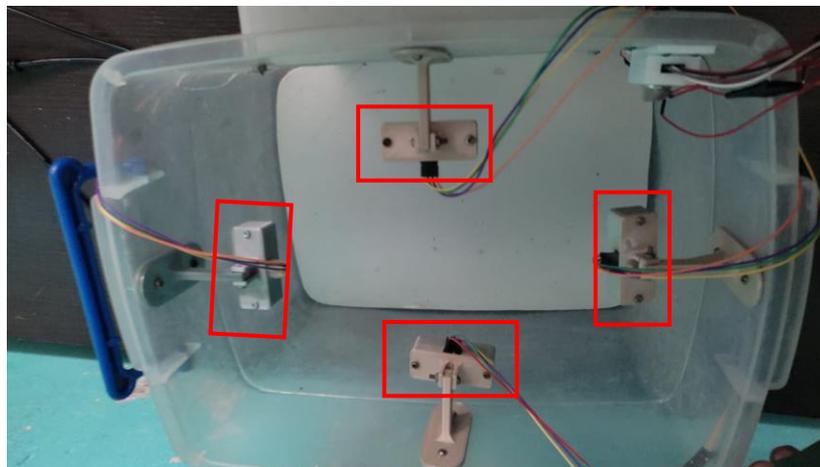
##### Volumen

Para la determinación del volumen se hace el uso de cuatro sensores HC-SR04 ubicados en cada lado del contenedor como se ilustra en la Figura 98. La lectura se da mediante la proximidad de los objetos dentro del contenedor, es así que se emplea un cálculo con el cual se

determina el porcentaje de llenado del contenedor, para ello se asigna un 25 % a cada sensor es decir que, si los cuatro sensores detectan un objeto a cierta distancia, se tendrá un volumen de 100%, lo que significa que el contenedor se encuentra lleno, de igual manera se puede considerar que se encuentra lleno si 3 de los sensores detectan objetos, se tendría un 75 % que sería más de la mitad de residuos dentro del contenedor.

### **Figura 98**

*Ubicación de los sensores para la determinación de volumen del contenedor*

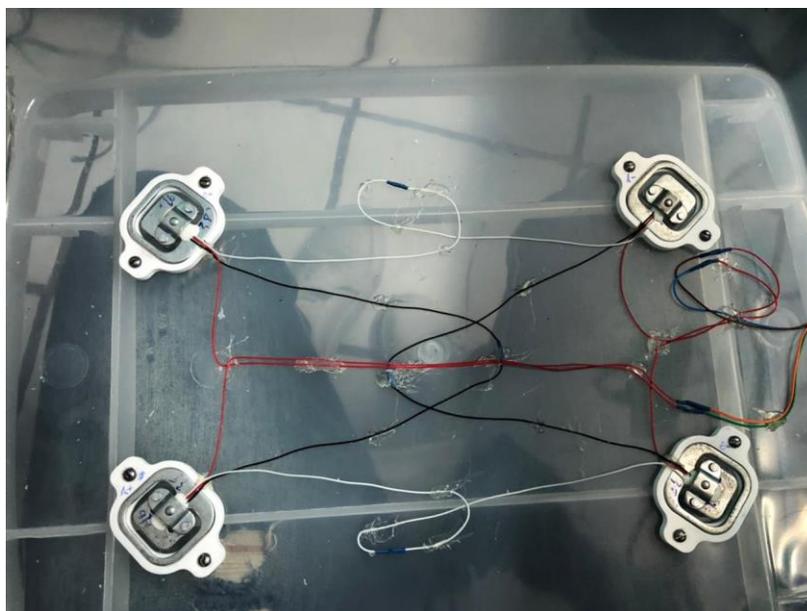


### **Peso**

De igual manera para la determinación del peso se hace uso de cuatro celdas de carga las cuales están distribuidas en las esquinas del contenedor de manera equidistante lo que permite la lectura del peso distribuido en todos los sensores como se puede apreciar en la Figura 99, sobre ellos se coloca una plancha de un material de aluminio y policarbonato.

### **Figura 99**

*Ubicación de las celdas de peso de 50kg para la toma de datos del peso de los residuos*

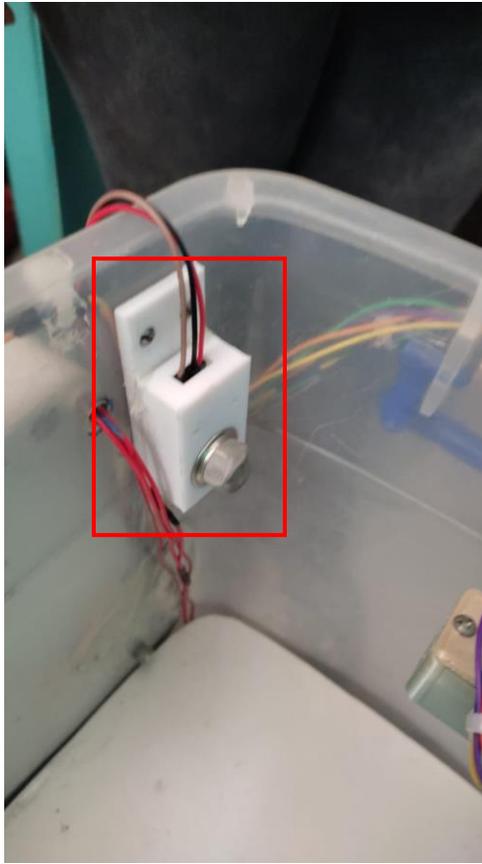


## **CO2**

Para la lectura del nivel de CO2 se usa el sensor MQ-135 el cual está ubicado de igual manera dentro del contenedor como se aprecia en la Figura 100, cabe mencionar que este sensor no siempre presentara un nivel alto de concentración ya que el gas se presenta solo en algunos casos, lo que dependerá de los residuos que se encuentren dentro del contenedor, sin embargo, se está considerando su lectura para cualquiera que sea el caso.

### **Figura 100**

*Ubicación del sensor de CO2 en el contenedor*

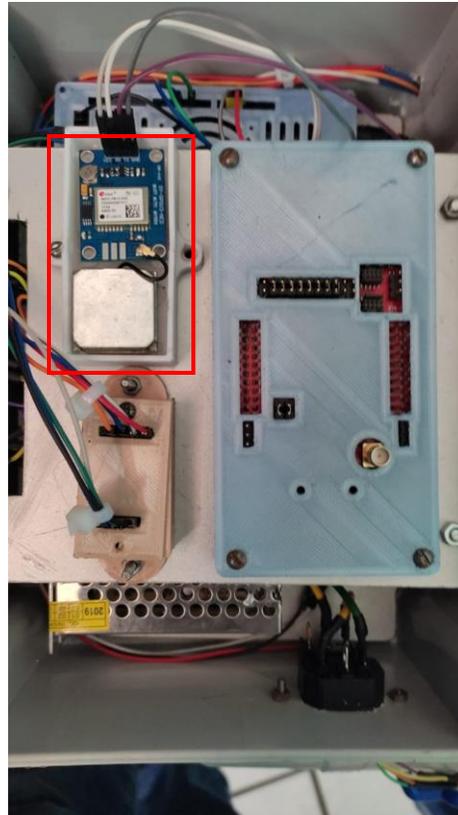


### **Ubicación**

Para el caso de esta variable se usa el Módulo GPS NEO-7M, ubicado en la caja de protección ya que solo se necesita saber la ubicación del contenedor lleno. Es así como se tendrá las lecturas únicamente cuando el contenedor está a su capacidad máxima establecida para el caso de estudio.

### **Figura 101**

*Ubicación del módulo GPS para determinar la ubicación del contenedor*

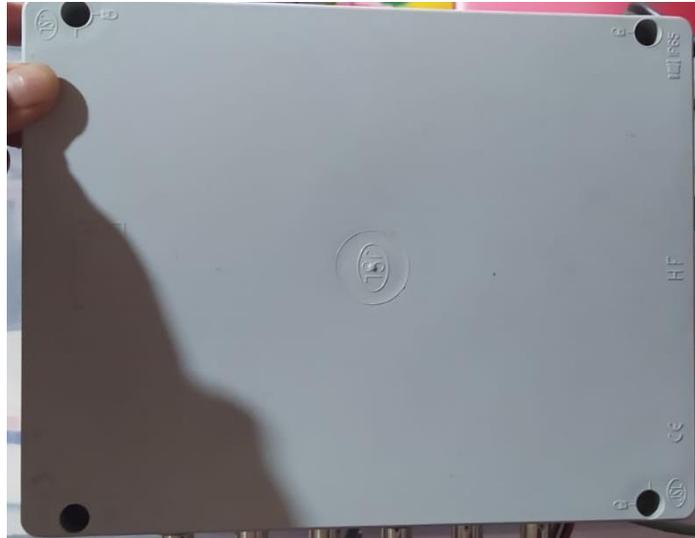


### 4.1.3. Instalación del Nodo Coordinador

El nodo coordinador va a estar ubicado de forma centralizada donde se tenga alcance hacia los nodos clientes para que puedan hacer el envío de todos los datos. De igual manera se coloca en un lugar dentro de una caja de protección que evite que se tenga contacto físico con el ambiente.

#### **Figura 102**

*Caja protección nodo coordinador*



#### **4.1.4. Simulación de la red mesh**

Para poder verificar el comportamiento de los nodos dentro de una red mesh en un ambiente simulado se opta por el uso del simulador COOJA que permite la colocación diferentes nodos y establecer algunos escenarios donde se puede probar como es el envío de información dentro de una red mesh.

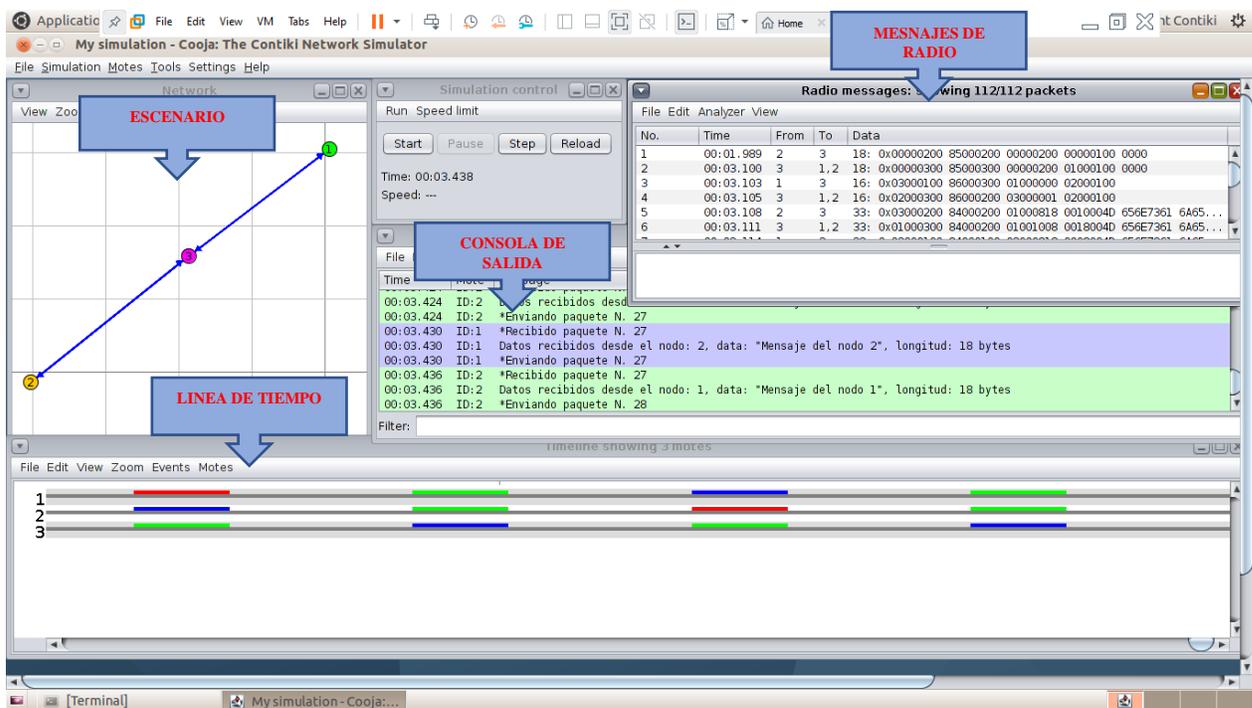
Para poder entender como se hace el envío de paquetes y como se presenta la información en el simulador se opta por un escenario básico con 3 nodos un coordinador, un nodo cliente y un nodo router, el cual enrutará los datos hacia el coordinador. En la Figura 104, se tiene tres tipos de nodos con los que se va a trabajar para mostrar como es el envío de datos entre nodos, de igual manera de tiene el panel de radio donde se muestran los mensajes de intercambio en capa física de los nodos y de igual manera los datos contenidos en cada paquete que se envía.

En el segundo campo se muestran la consola de salida donde indican los paquetes que se han enviado o recibido, así como también el número de paquete el contenido del campo de datos y longitud de los datos.

En la parte de abajo se tiene una línea de tiempo donde se puede analizar la comunicación RF de los nodos dentro de la mesh.

**Figura 103**

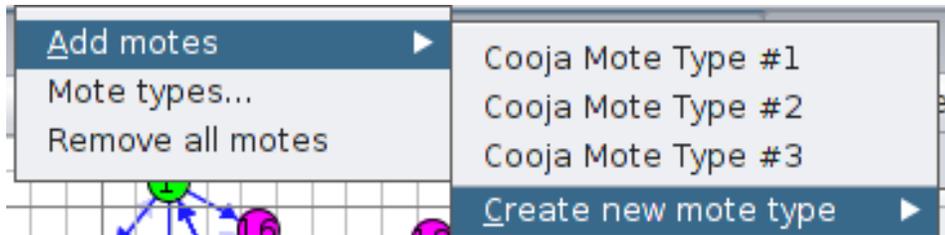
*Interfaz de simulación*



De igual manera se tiene configurado los tres tipos de nodos, para lo cual cada uno tendrá las funciones específicas dentro de la red mesh. Se tiene el tipo 1 que sería para un nodo coordinador, el tipo 2 que sería para los nodos clientes y el tipo 3 para los nodos router.

**Figura 104**

*Tipos de nodos dentro de la red mesh*



Al iniciar la simulación se tiene una convergencia donde los nodos se inicializan e intercambian los primeros mensajes que permiten armar la red como tal.

Para el escenario de la Figura 105, se tiene que el nodo 2 va a enviar un mensaje al nodo 1 el cual se ha configurado como nodo coordinador, al no estar cerca va a enviar el mensaje primero al nodo 3 ya que a través de este va a lograr comunicarse con el nodo 1, de igual manera se tiene el contenido y tamaño del mensaje enviado desde el nodo 2 al 3.

**Figura 105**

*Envío de datos al nodo 3*

No.	Time	From	To	Data
4	00:03.105	3	1,2	16: 0x02000300 86000200 03000001 02000100
5	00:03.108	2	3	33: 0x03000200 84000200 01000818 0010004D 656E7361 6A65...
6	00:03.111	3	1,2	33: 0x01000300 84000200 01001008 0018004D 656E7361 6A65...
7	00:03.114	1	3	33: 0x03000100 84000100 02000818 0008004D 656E7361 6A65...
8	00:03.117	3	1,2	33: 0x02000300 84000100 02001010 0018004D 656E7361 6A65...
9	00:03.120	2	3	33: 0x03000200 84000200 01000818 0010004D 656E7361 6A65...

**Packet data (33 bytes)**  
 03000200 84000200 01000818 0010004D 656E7361 ..... Mensa  
 6A652064 656C206E 6F646F20 32 je del nodo 2

Luego de hacer el envío de los datos desde el nodo 3 hacia el nodo 1, donde se puede apreciar de igual manera en la Figura 106 que se hace el envío hacia el nodo 1, de igual manera se aprecia como candidato a destino el nodo 2 debido a que se encuentra dentro del alcance del nodo 3.

**Figura 106**

*Envío de datos al nodo 1*

No.	Time	From	To	Data
4	00:03.105	3	1,2	16: 0x02000300 86000200 03000001 02000100
5	00:03.108	2	3	33: 0x03000200 84000200 01000818 0010004D 656E7361 6A65...
6	00:03.111	3	1,2	33: 0x01000300 84000200 01001008 0018004D 656E7361 6A65...
7	00:03.114	1	3	33: 0x03000100 84000100 02000818 0008004D 656E7361 6A65...
8	00:03.117	3	1,2	33: 0x02000300 84000100 02001010 0018004D 656E7361 6A65...
9	00:03.120	2	3	33: 0x03000200 84000200 01000818 0010004D 656E7361 6A65...

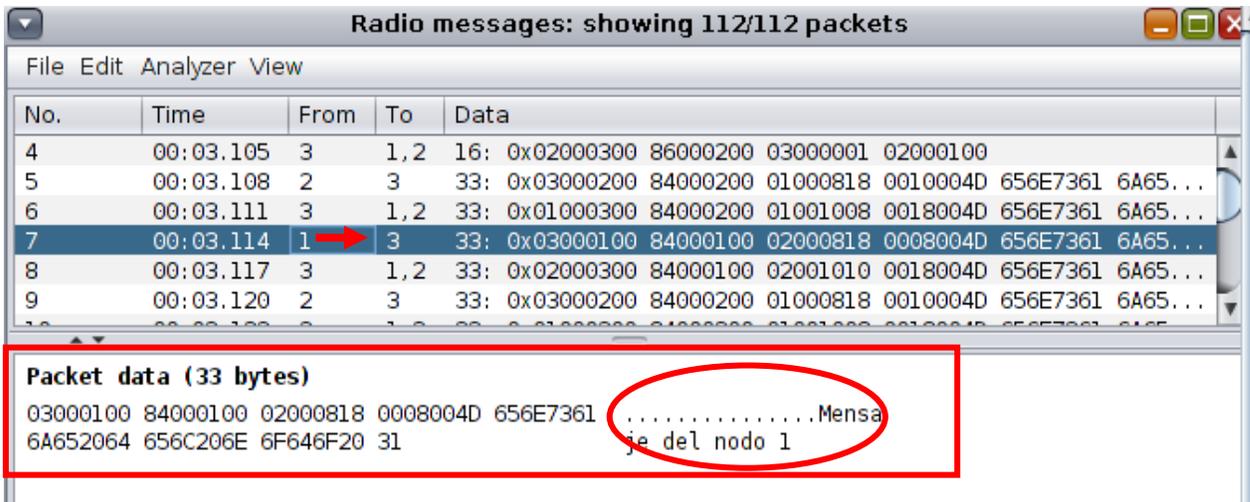
  

Packet data (33 bytes)	
01000300 84000200 01001008 0018004D 656E7361	.....Mensaje
6A652064 656C206E 6F646F20 32	je del nodo 2

De esa manera llegaría el mensaje desde el nodo 2 hacia el nodo 1, de igual manera se tiene una respuesta desde el nodo 1 hacia el nodo 2. En la Figura 107, se aprecia el mensaje de respuesta.

**Figura 107**

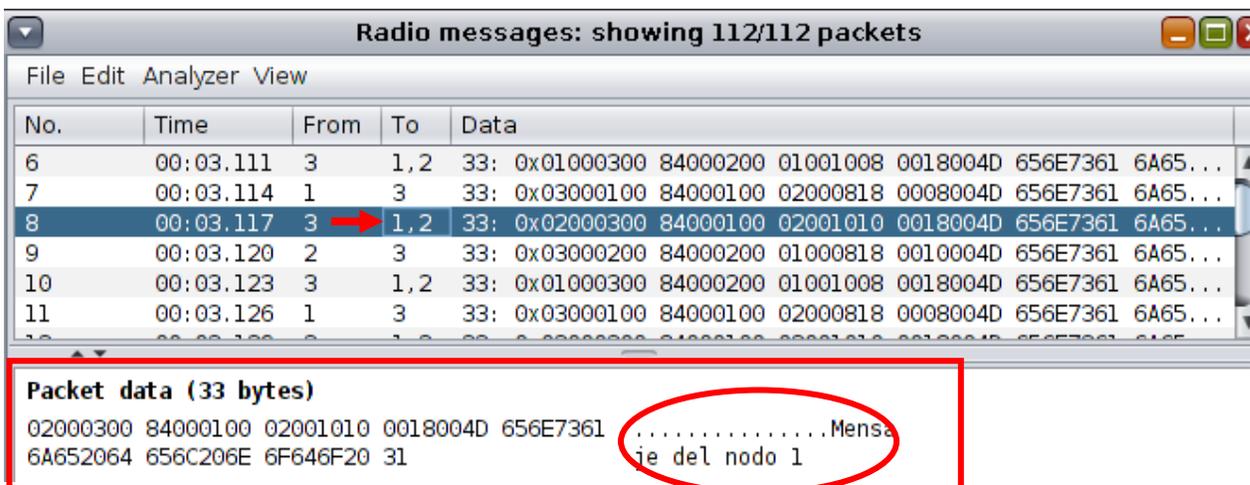
*Envío de datos al nodo 3 desde el nodo 1*



Y finalmente el mensaje de respuesta desde el nodo 1 hacia el nodo 2 llega al destino como se puede verificar en la Figura 108.

**Figura 108**

*Llegada de datos al nodo 2*



En la parte de la consola se tiene los mensajes que se envían desde el nodo origen hacia el nodo destino en este caso nodo 2 y nodo 1 respectivamente, de igual manera el mensaje de respuesta en la Figura 109, se puede visualizar lo explicado.

**Figura 109**

*Intercambio de mensajes entre nodo 1 y nodo 2*

The screenshot shows a 'Mote output' window with a table of messages. The table has three columns: 'Time', 'Mote', and 'Message'. The messages are as follows:

Time	Mote	Message
00:01.987	ID:2	Empezando transmision...
00:03.106	ID:2	*Enviando paquete N. 1
00:03.112	ID:1	*Recibido paquete N. 1
00:03.112	ID:1	Datos recibidos desde el nodo: 2, data: "Mensaie del nodo 2", longitud: 18 bytes
00:03.112	ID:1	*Enviando paquete N. 1
00:03.118	ID:2	*Recibido paquete N. 1
00:03.118	ID:2	Datos recibidos desde el nodo: 1, data: "Mensaje del nodo 1", longitud: 18 bytes
00:03.118	ID:2	*Enviando paquete N. 2

Annotations in the image include:

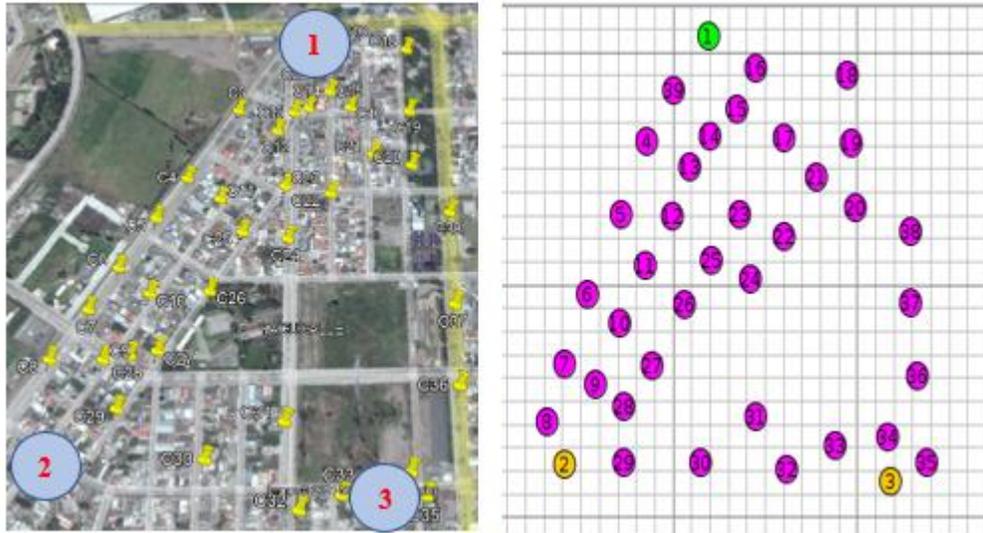
- A blue arrow pointing left from the message at 00:03.106 labeled 'NODO 2 ENVIA'.
- A blue arrow pointing left from the message at 00:03.112 labeled 'NODO 1 RECIBE'.
- A blue arrow pointing left from the message at 00:03.112 labeled 'NODO 1 ENVIA'.
- A blue arrow pointing left from the message at 00:03.118 labeled 'NODO 2 RECIBE'.

a. *Red mesh nodos ubicados distribuidamente*

Se tiene 38 nodos que se encuentran ubicados según el caso de estudio en el sector Yacucalle.

**Figura 110**

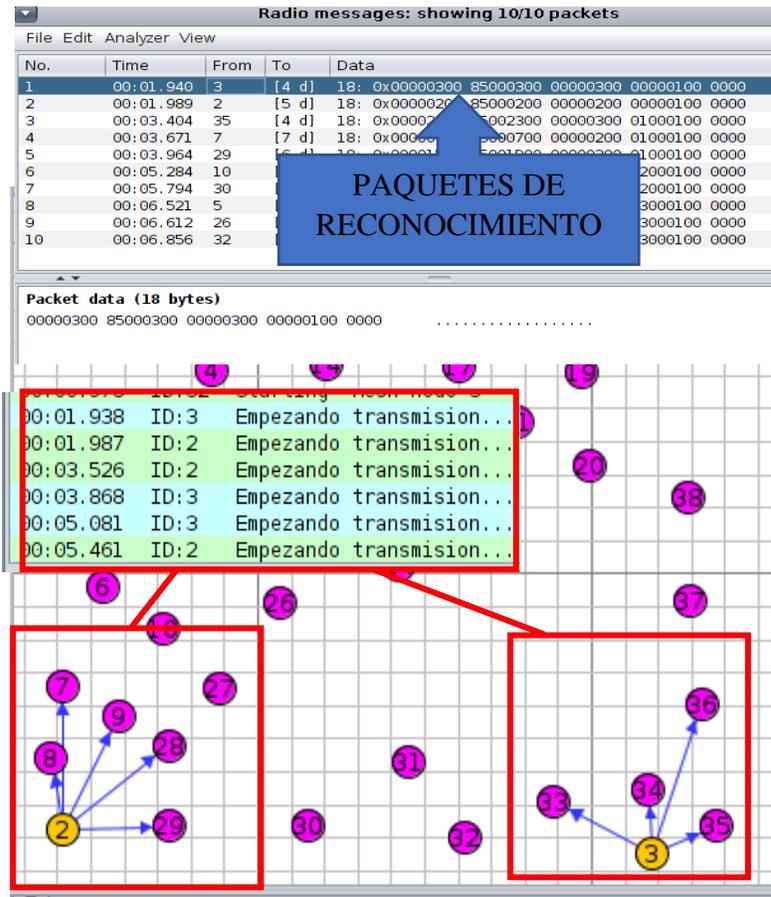
*Identificación de nodo coordinador y nodos clientes*



Se tiene un nodo coordinador y dos nodos clientes, los cuales van a ser ubicados en los lugares extremos para poder evidenciar como se hace el reenvío de información a través de los nodos router lo cuales van a permitir que el mensaje llegue al destino en este caso al nodo coordinador que es el número 1, con lo que se puede verificar el objetivo de una red mesh como tal. En la Figura 111, se puede observar que se hace un reconocimiento desde los nodos cliente hacia los nodos más cercanos para ir estableciéndolos posibles saltos para crear la ruta de envío de los datos.

**Figura 111**

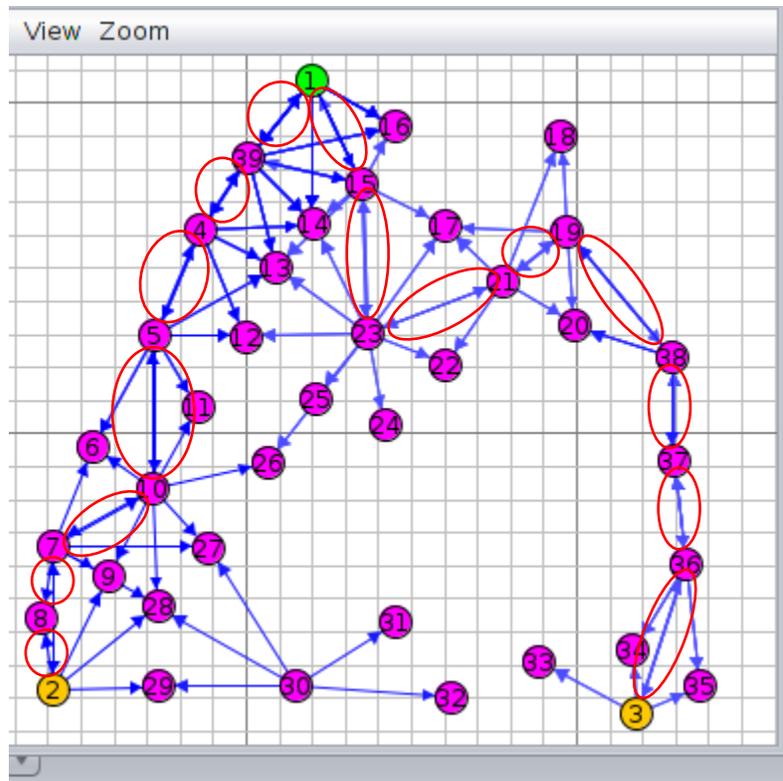
*Nodos cliente reconociendo los nodos vecinos*



En la figura 112, se puede observar que para el establecimiento de rutas se va haciendo salto a salto y reconociendo a los nodos más cercanos, para verificar las rutas establecidas se las puede observar que están con una flecha de doble lado, ya que se tiene una petición y una respuesta tanto de nodo origen como del nodo destino.

**Figura 112**

*Establecimiento de rutas desde los nodos 2 y 3 hacia el nodo 1*



A continuación, se procede a verificar el envío de mensajes desde los nodos origen hacia el nodo final. Primeramente, se inicializan todos los nodos para que luego sean detectados por los nodos clientes como elementos de la ruta a seguir como se indica en la Figura 113.

**Figura 113**

*Inicialización de todos los nodos de la mesh*

Time	Note	Message
00:00.881	ID:36	Starting 'Mesh nodo 3'
00:00.884	ID:13	Contiki-2.6-900-ga6227e1 started. Node id is set to 13.
00:00.884	ID:13	Rime started with address 13.0
00:00.884	ID:13	MAC 0d:00:00:00:00:00:00:00:00 Rime/CSMA/nullrdc, channel check rate 1000 Hz
00:00.884	ID:13	Starting 'Mesh nodo 3'
00:00.899	ID:3	Contiki-2.6-900-ga6227e1 started. Node id is set to 3.
00:00.899	ID:3	Rime started with address 3.0
00:00.899	ID:3	MAC 03:00:00:00:00:00:00:00:00 Rime/CSMA/nullrdc, channel check rate 1000 Hz
00:00.899	ID:3	Starting 'Mesh nodo 2'
00:00.906	ID:21	Contiki-2.6-900-ga6227e1 started. Node id is set to 21.
00:00.906	ID:21	Rime started with address 21.0
00:00.906	ID:21	MAC 15:00:00:00:00:00:00:00:00 Rime/CSMA/nullrdc, channel check rate 1000 Hz
00:00.906	ID:21	Starting 'Mesh nodo 3'
00:00.945	ID:23	Contiki-2.6-900-ga6227e1 started. Node id is set to 23.
00:00.945	ID:23	Rime started with address 23.0
00:00.945	ID:23	MAC 17:00:00:00:00:00:00:00:00 Rime/CSMA/nullrdc, channel check rate 1000 Hz
00:00.945	ID:23	Starting 'Mesh nodo 3'
00:00.978	ID:32	Contiki-2.6-900-ga6227e1 started. Node id is set to 32.
00:00.978	ID:32	Rime started with address 32.0
00:00.978	ID:32	MAC 20:00:00:00:00:00:00:00:00 Rime/CSMA/nullrdc, channel check rate 1000 Hz

Filter:

Luego de ello el nodo 2 es el que primero envía la información hacia el nodo 1. En la consola de salida se puede verificar que ya se están comunicando simultáneamente los nodos 1 y 2.

**Figura 114**

*Establecimiento de la comunicación entre el nodo 1 y nodo 2*

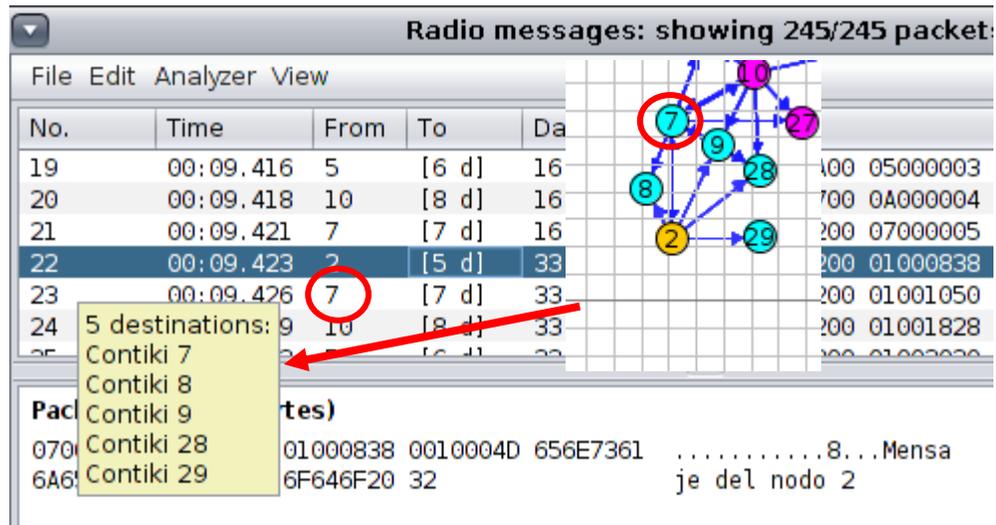
00:09.898	ID:2	*Enviando paquete N. 14
00:09.916	ID:1	*Recibido paquete N. 14
00:09.916	ID:1	Datos recibidos desde el nodo: 2, data: "Mensaje del nodo 2", longitud: 18 bytes
00:09.916	ID:1	*Enviando paquete N. 14
00:09.935	ID:2	*Recibido paquete N. 14
00:09.935	ID:2	Datos recibidos desde el nodo: 1, data: "Mensaje del nodo 1", longitud: 18 bytes
00:09.935	ID:2	*Enviando paquete N. 15

Ahora, lo que se va a mostrar es cada uno de los saltos que se han establecido para poder corroborar la información mostrada en la Figura 112. Se va a enumerar los saltos que dan los paquetes para mejor entendimiento.

En la Figura 115, el nodo 2 analiza los vecinos alcanzables más cercanos, es así que opta por el nodo 7 como siguiente salto.

**Figura 115**

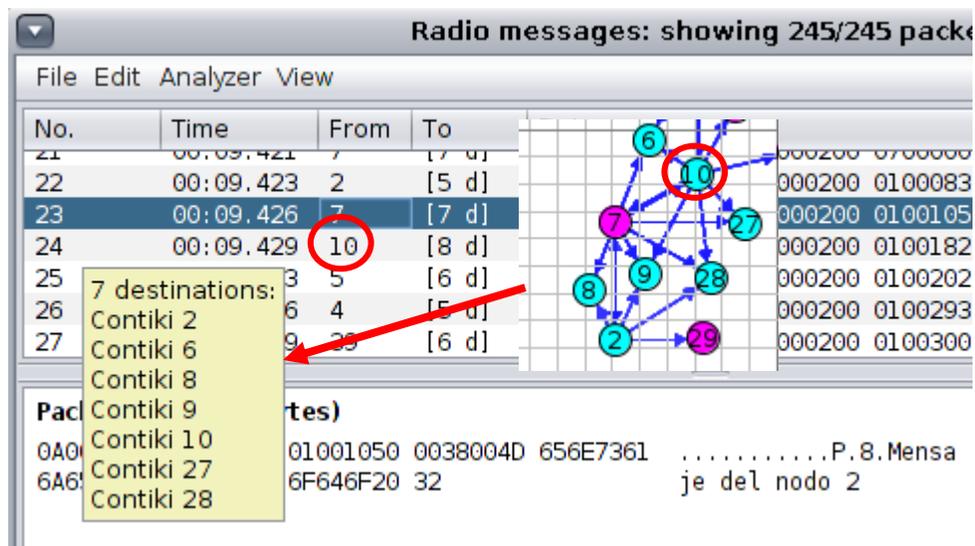
*Primer salto del paquete desde el nodo 2*



En la Figura 116, el nodo 7 va a enviar al siguiente salto entre los posibles que se muestran, es así como considera el más idóneo el nodo 10.

**Figura 116**

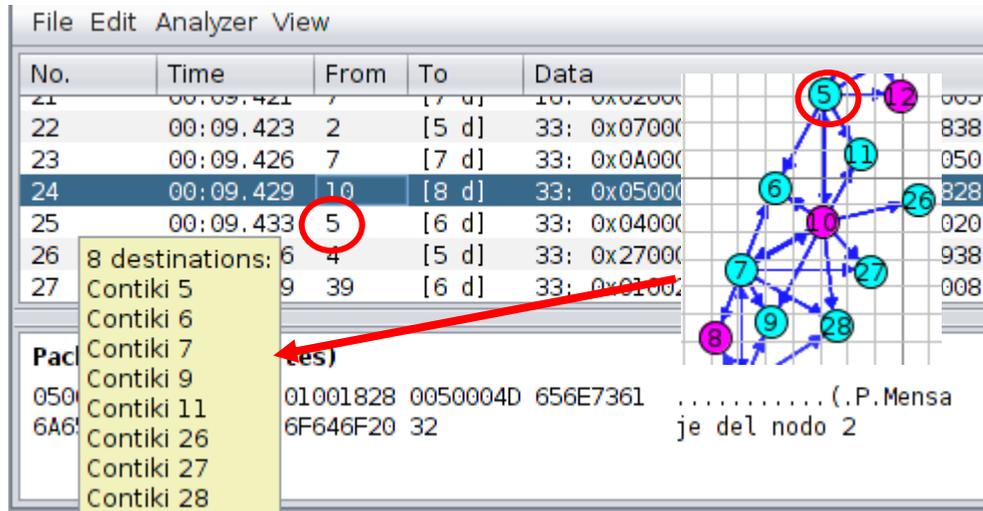
*Segundo salto del paquete desde el nodo 2*



Al siguiente salto al cual va a enviar es el nodo 5 desde el nodo 10 tal y como se muestra en la Figura 117.

**Figura 117**

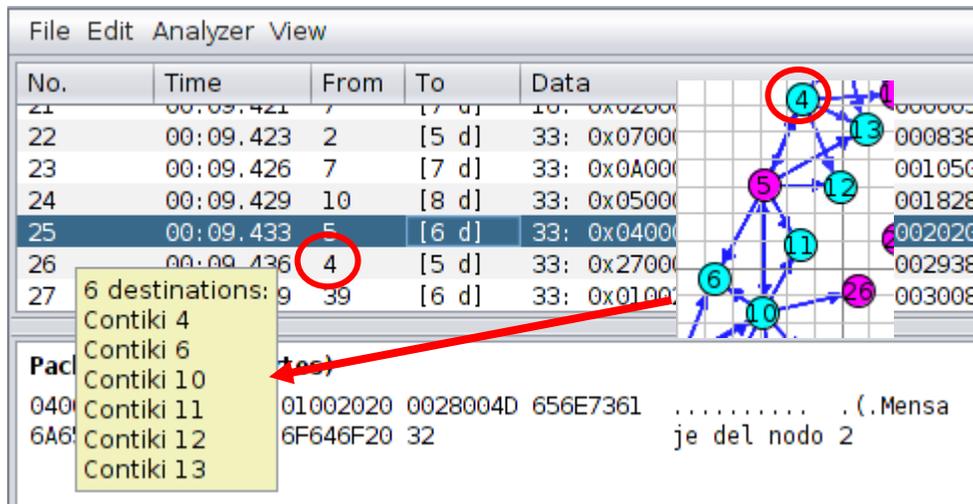
*Tercer salto del paquete desde el nodo 2*



El nodo 5 elije al nodo 4 como siguiente salto entre aquellos alcanzables como se muestra en la Figura 118.

**Figura 118**

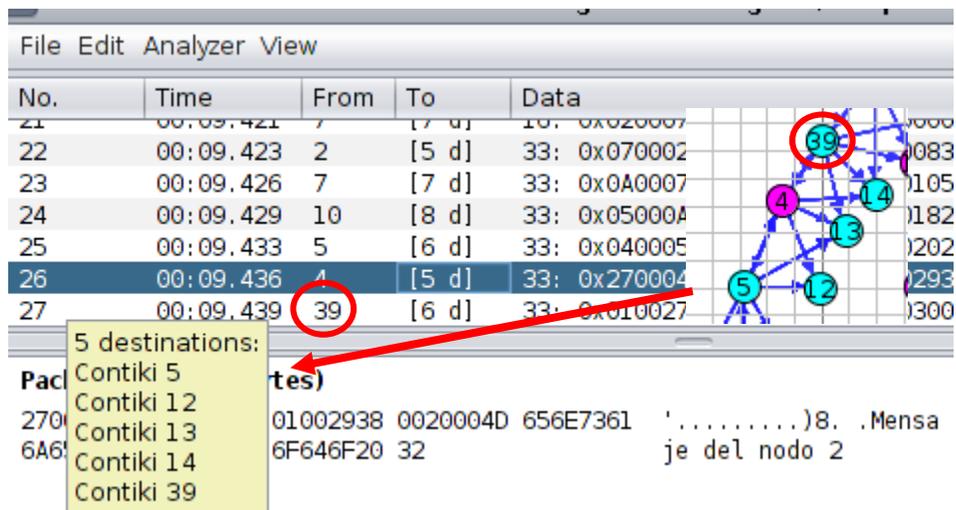
*Cuarto salto del paquete desde el nodo 2*



El próximo salto que tomará el paquete es el nodo 39 el cual ya se encuentra ceca del nodo destino como se indica en la Figura 119.

**Figura 119**

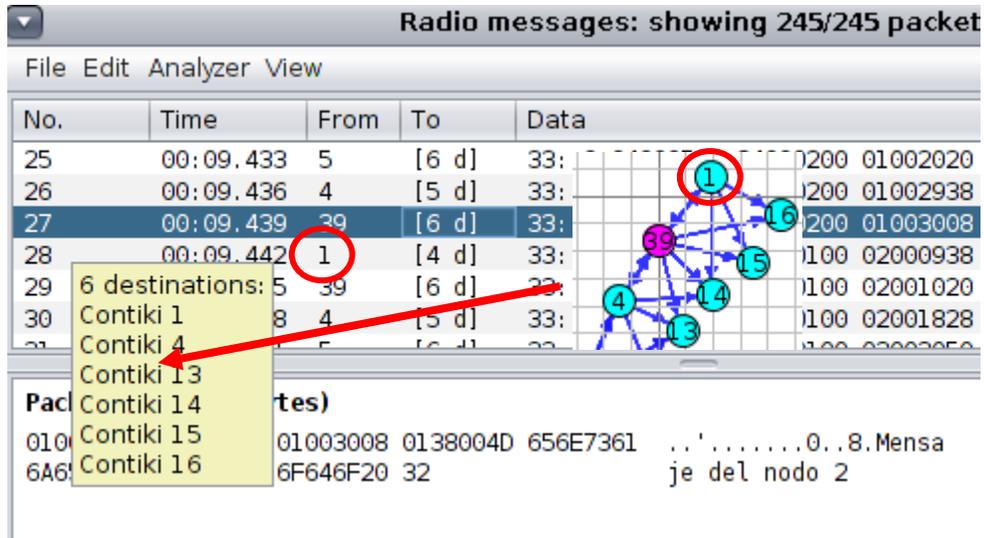
*Quinto salto del paquete desde el nodo 2*



Por último, se tiene ya como siguiente salto al nodo 1 el mismo que es el nodo destino, es así que de igual manera se tienen los posibles saltos, pero en este caso se concretaría el mensaje de ida como se indica en la Figura 120.

**Figura 120**

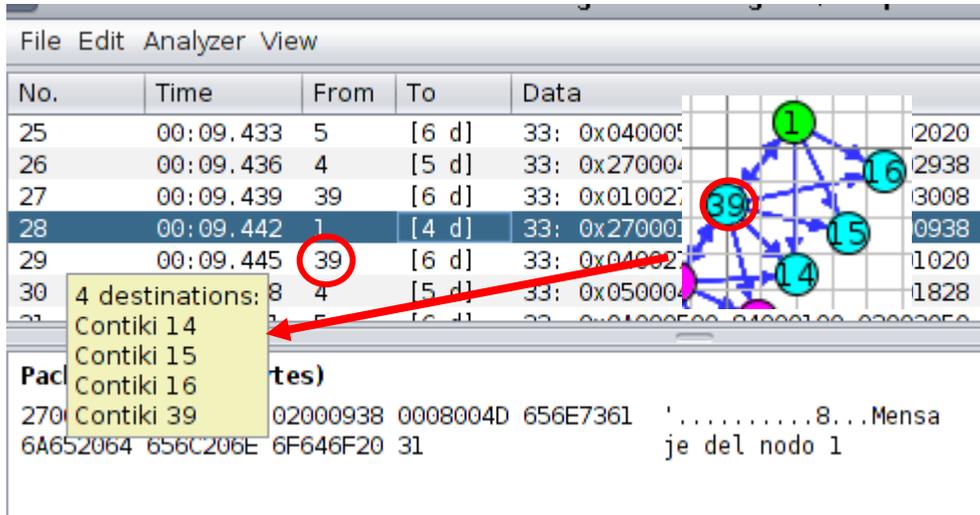
*Llegada del paquete al nodo destino*



Para el proceso de respuesta el paquete va a tomar la misma ruta que se tomó el paquete al llegar al nodo coordinador, en la Figura 121, se mira que el nodo 1 toma como primer salto al nodo 39 por el cual fue que le llegó el mensaje de origen.

**Figura 121**

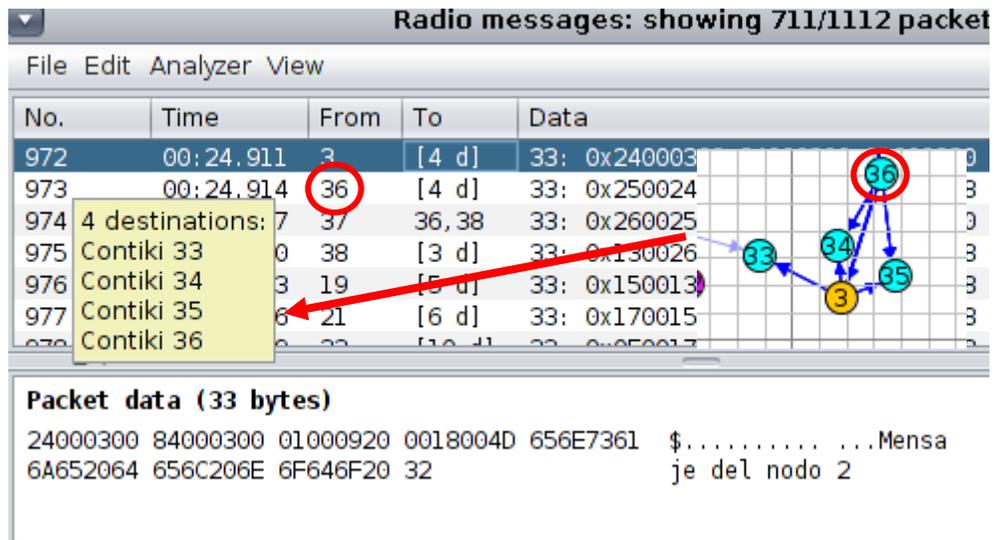
*Verificación de los próximos saltos desde el Nodo coordinador*



Ahora se puede verificar el envío de datos desde el nodo 3 hacia el nodo 1, en la Figura 122, se muestra el primer salto que da el paquete.

**Figura 122**

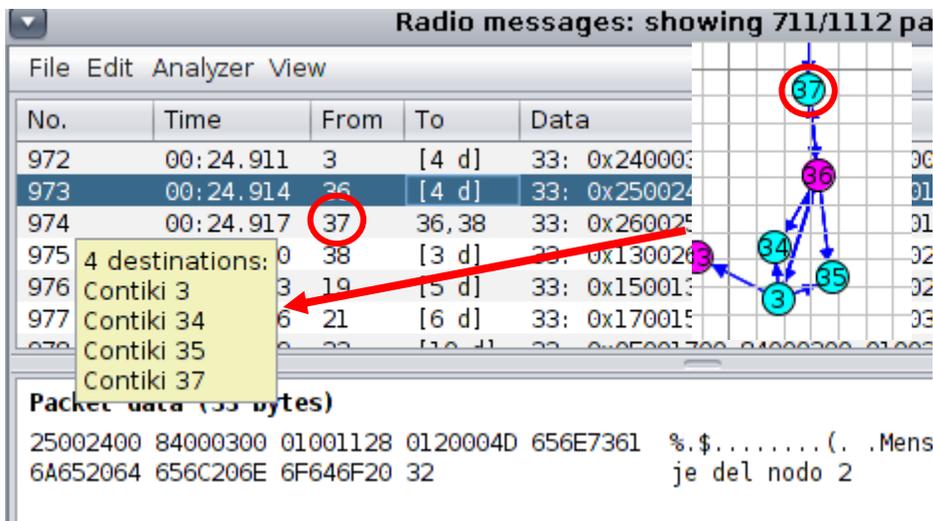
*Primer salto desde el nodo 3*



En la Figura 123, se ilustra los posibles saltos para el envío del mensaje desde el nodo 36, el cual toma como siguiente nodo al 37.

**Figura 123**

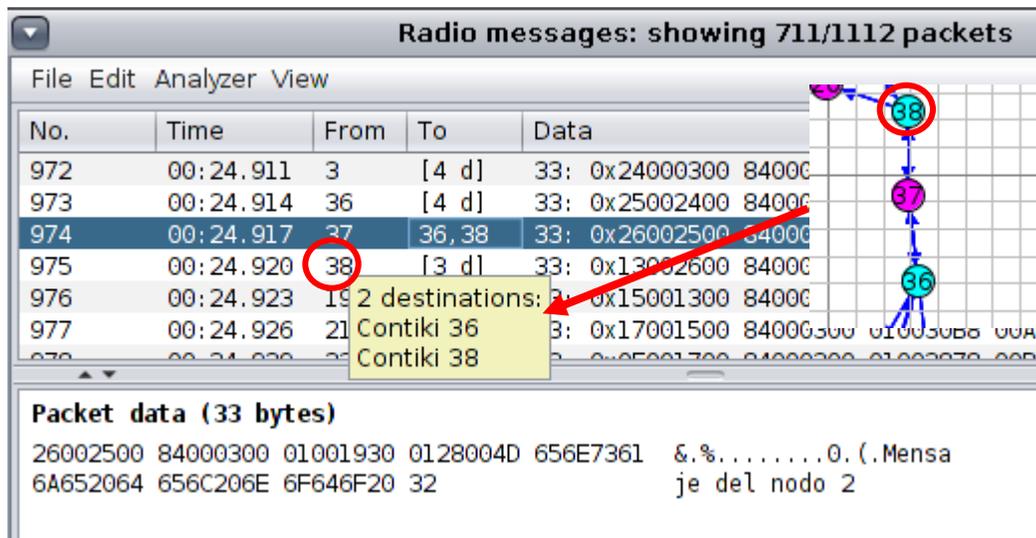
*Segundo salto desde el nodo 3*



El próximo salto que tomara el paquete es del nodo 38 como se ilustra en la Figura 124.

**Figura 124**

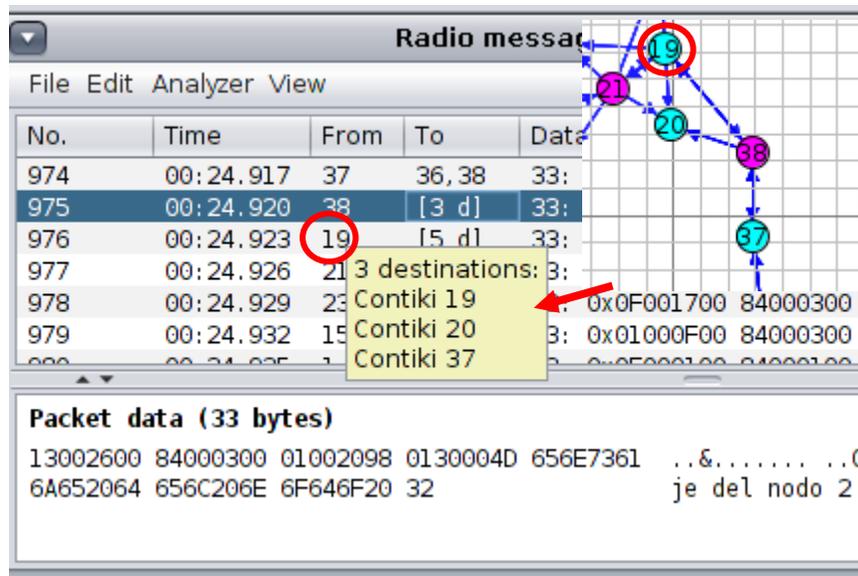
*Tercer salto desde el nodo 3*



En la Figura 125, se muestra que el próximo salto que dará el paquete es el nodo 19, entre todos los posibles saltos.

**Figura 125**

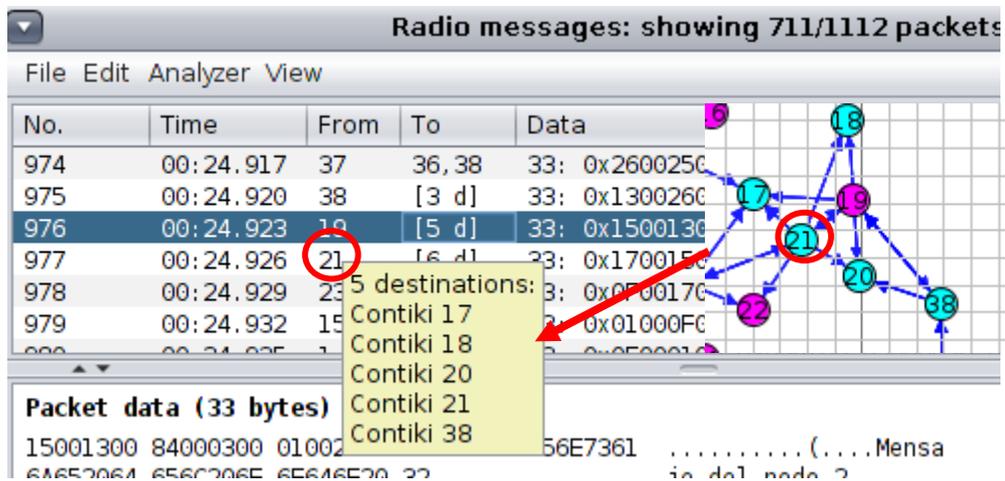
*Cuarto salto desde el nodo 3*



Ahora se tiene que el siguiente salto se da desde el nodo 19 hasta el nodo 21 como se muestra en la Figura 126.

**Figura 126**

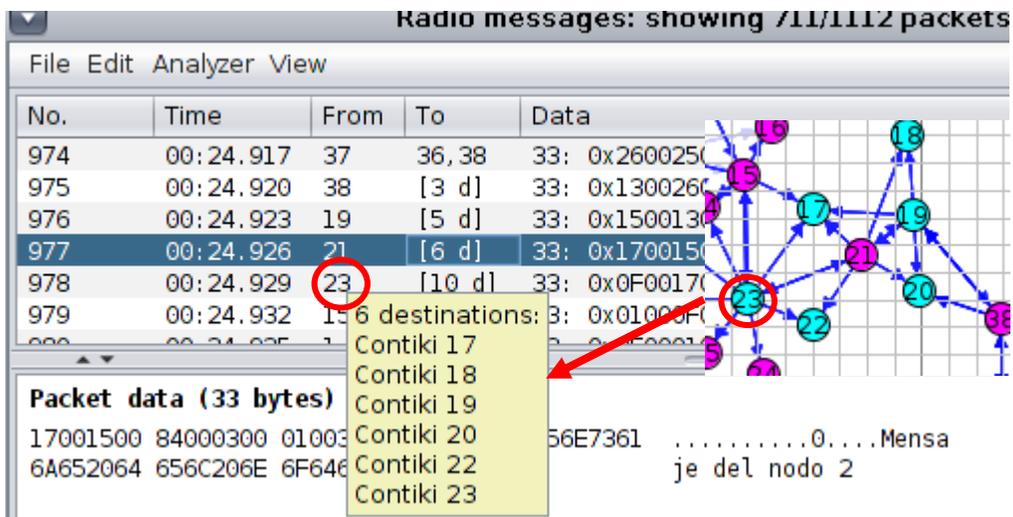
*Quinto salto desde el nodo 3*



En este punto el paquete ya se encuentra en el nodo 21, y toma como siguiente salto al nodo 23 entre los posibles saltos.

**Figura 127**

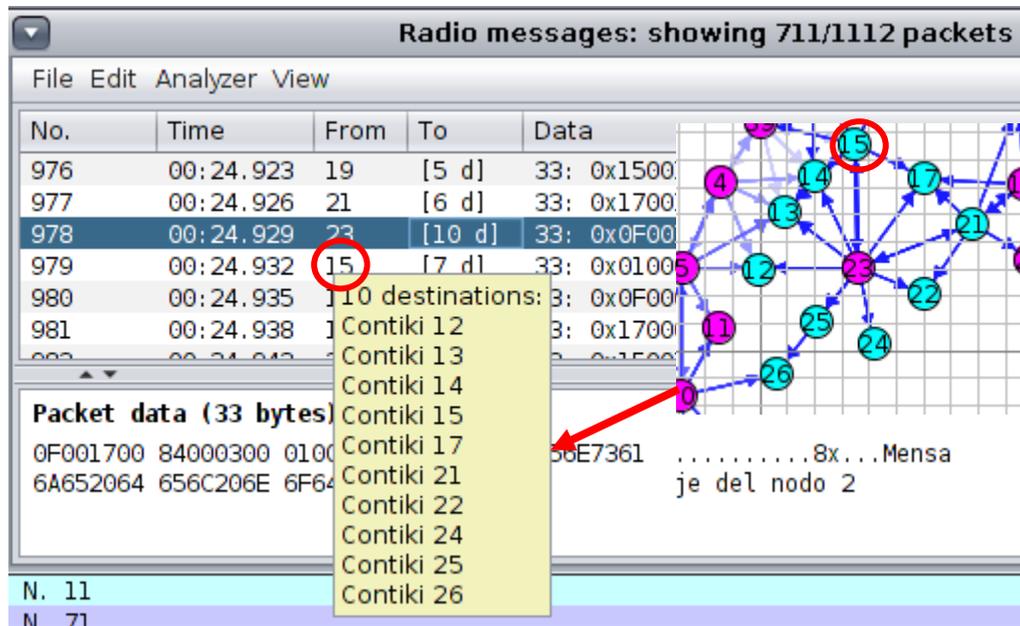
*Sexto salto desde el nodo 3*



El siguiente salto que tomará el paquete es el nodo 15, se puede evidenciar en la Figura 128.

**Figura 128**

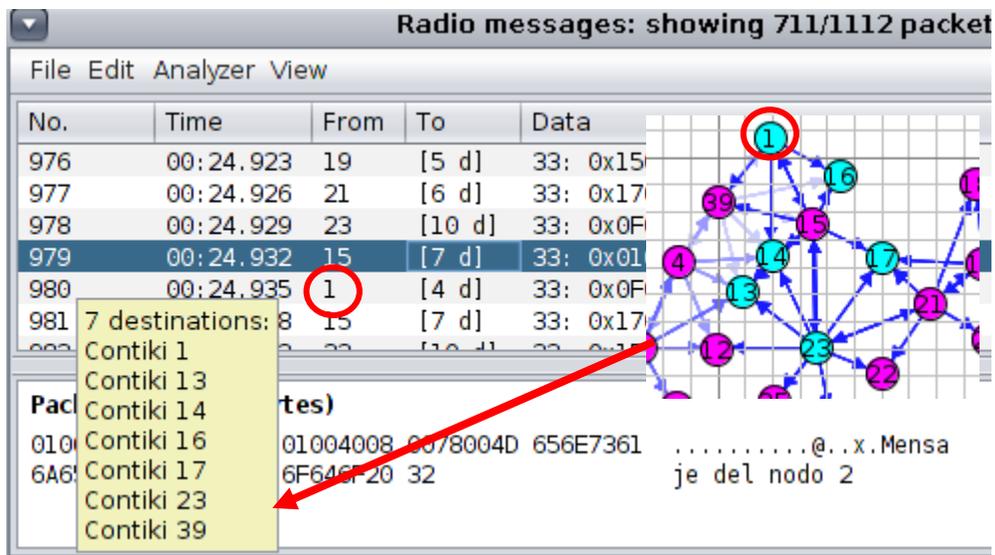
*Sexto salto desde el nodo 3*



Como ultimo salto y destino es el nodo 1, el cual fue elegido entre los posibles saltos, a partir del nodo 15 como se ilustra en la Figura 129.

**Figura 129**

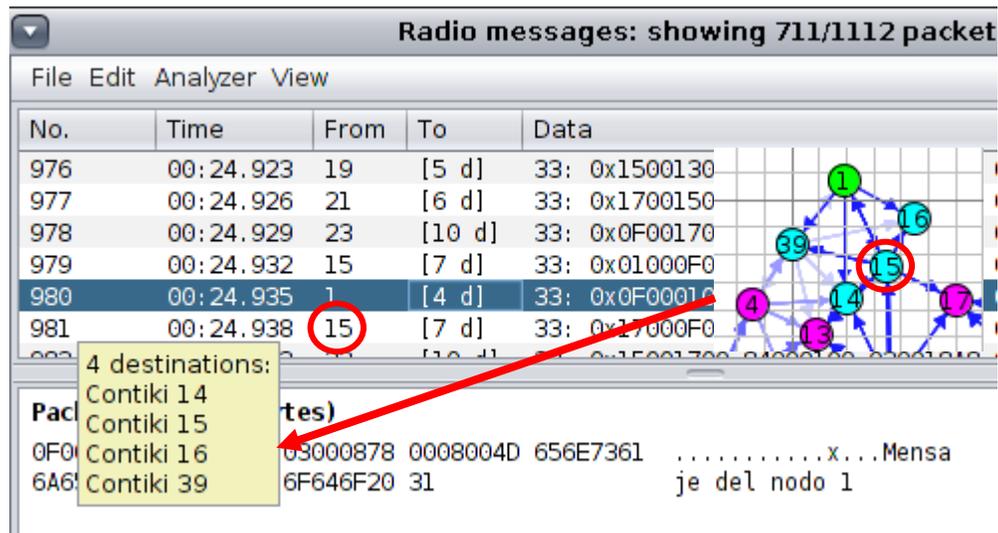
*Séptimo salto desde el nodo 3*



Al igual que en la respuesta que dio el nodo 1 hacia el nodo 2 se tiene de la misma manera hacia el nodo 3, tomando la misma ruta que se eligió para ida como vuelta del paquete en este caso la respuesta desde el nodo 1, lo mencionado se indica en la Figura 130.

**Figura 130**

*Primer salto del paquete de respuesta desde el nodo 1*



## 4.2. Pruebas de funcionamiento

Este apartado corresponde a las pruebas que se realiza haciendo uso del sistema donde se puede evidencia que está funcionando de manera correcta, se hace la pruebas dependiendo del estado de contenedor, y luego en base a un contenedor lleno se realiza un cronograma de pruebas donde se va verificando cada una de las etapas desde la toma de datos por parte de los sensores hasta la publicación de la información en la plataforma web.

### 4.2.1. Pruebas según el estado del contenedor

Para la realización de las pruebas iniciales del sistema embebido se adaptan los contenedores a diferentes estados físicos lo que permite determinar la correcta lectura de los sensores dentro de cada contenedor.

#### Contenedor Vacío

Como se puede apreciar en la Figura 131, el contenedor no dispone de ningún elemento en su interior por lo que no se va a tener un dato de volumen, peso, y en este caso tampoco de CO<sub>2</sub>, de igual manera la ubicación no se genera ya que únicamente se mostrará cuando el contenedor esté con un volumen considerable.

#### Figura 131

*Plataforma sin ninguna alerta por contenedor vacío*



# UBICACION

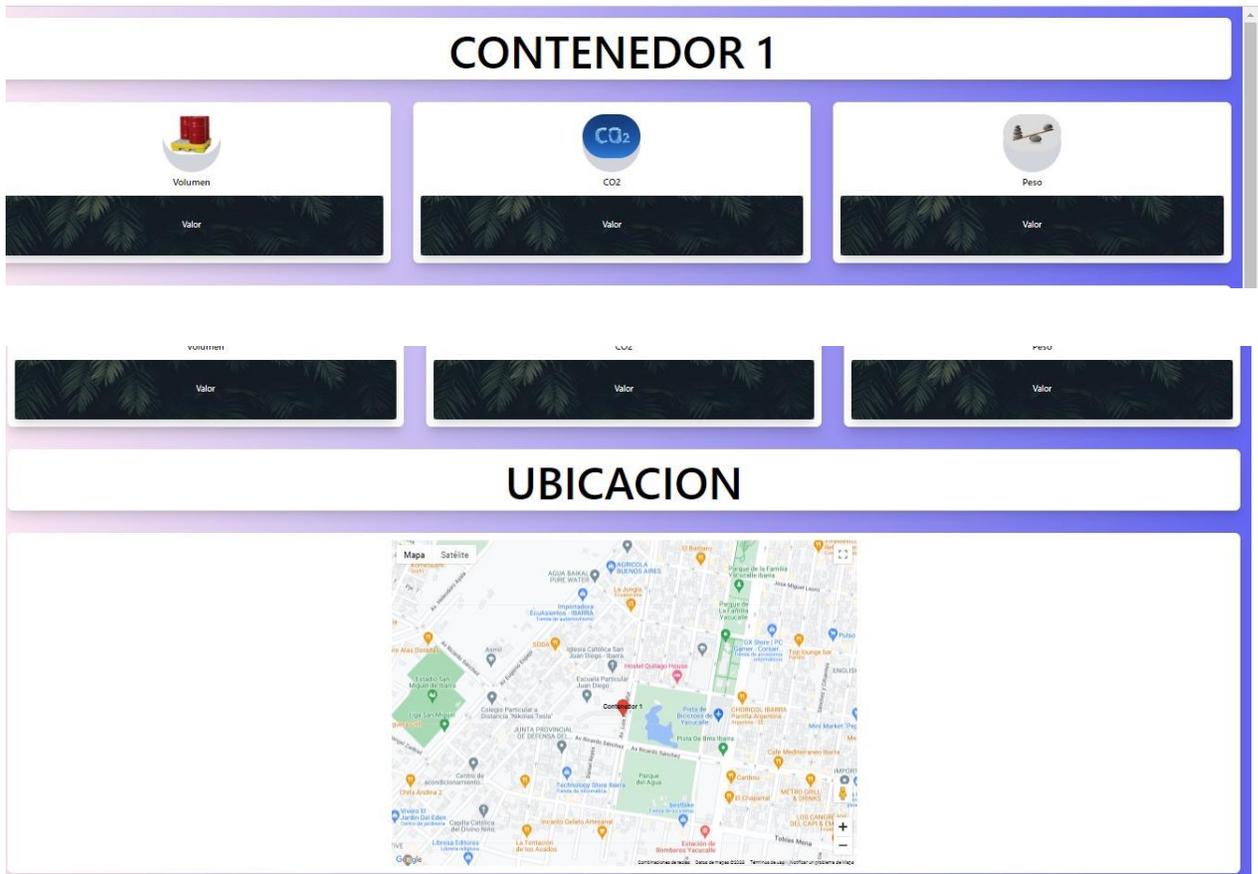


## Contenedor lleno a un 75% o más de su capacidad

En la programación de Arduino se establece que si el contenedor se encuentra lleno un 75% o es decir tres de los sensores de volumen marcan valores, se entiende que el contenedor está lleno, a pesar de que no tenga un peso considerable o el nivel de CO<sub>2</sub> sea nulo, se tendrá la actualización de la ubicación del contenedor en las coordenadas que esta en este momento como se puede apreciar en la Figura 132.

### Figura 132

*Datos y posición mostrados en la página web*



#### 4.2.2. Cronograma de pruebas de funcionamiento del sistema

Para asegurar el correcto funcionamiento del sistema es necesario realizar pruebas en un ambiente real con las cuales se puede realizar algún tipo de corrección en el caso de que se presente algún inconveniente en el proceso de ejecución. Para ellos se hace un cronograma de pruebas como se indica en la Tabla 30, en el cual se determina el tipo de prueba, lugar y condiciones del contenedor y de igual manera la duración de cada prueba para este caso de estudio.

**Tabla 30**

*Cronograma de pruebas*

---

## CONOGRAMA DE PRUEBAS

---

<b>Prueba</b>	<b>Lugar</b>	<b>Resultado</b>	<b>Duración</b>
<b>1. Funcionamiento del sistema embebido en conjunto con el nodo recolector</b>	Ambiente urbano, haciendo uso de alimentación eléctrica, ya que no se dispone de baterías.	Se tiene la recepción de datos desde los nodos sensores, se puede evidenciar conectando el módulo recolector al puerto serial para monitorear la llegada de los datos	Se hace las pruebas durante un día, para poder verificar que los datos enviados son los correctos dependiendo de estado del contenedor
<b>2. Recepción de datos en el nodo coordinador</b>	Se hace la ubicación en un ambiente externo, de igual manera se hace el uso de un cargador portátil para la alimentación del dispositivo	Se verifica conectando el módulo coordinador al puerto serial de la computadora donde mediante software idóneo se verifica la llegada de los datos hasta este punto del sistema	Se hace las pruebas por un día, lo que permite evidenciar que los datos están llegando de manera correcta en el nodo coordinador

---

---

<b>3. Publicación de datos en la plataforma Web IOT</b>	Se verifica en el domicilio donde se tiene conexión a internet y se disponga de un computador para acceder a la página web	Se puede verificar las publicaciones de los datos en la plataforma web, dependiendo del estado del contenedor	Se hace por un día donde se hace distintos cambios en el estado del contenedor para asegurar que los datos que llegar a la plataforma web son verdaderos.
---	--	---	---

---

**1. Funcionamiento del sistema embebido en conjunto con el nodo recolector**

Para asegurar que los datos estén tomados de manera correcta se hace pruebas del contenedor colocando elementos para que los sensores marquen un llenado, de los contarios no se tendrá datos que mostrar, en la Figura 133.

**Figura 133**

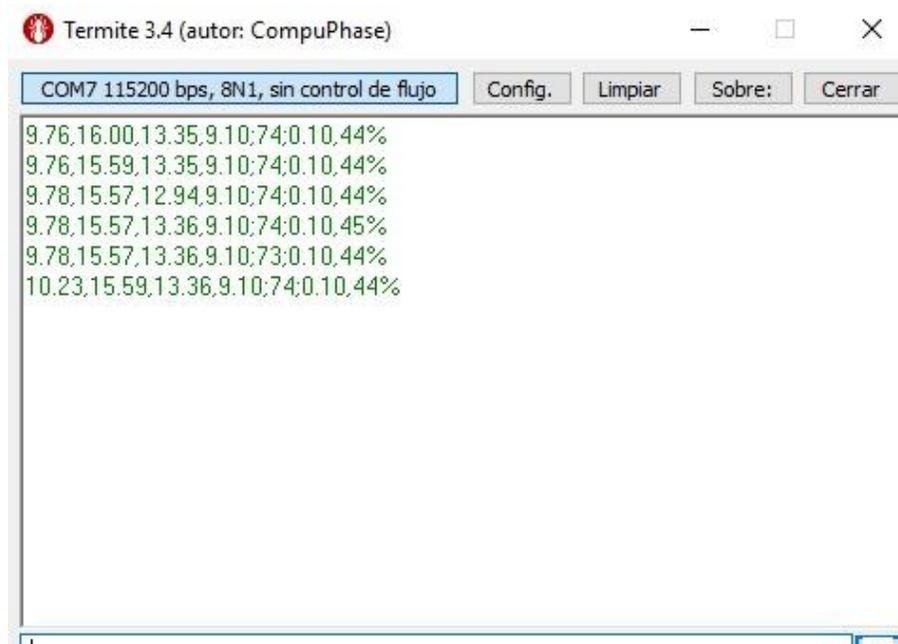
*Contenedor lleno de objetos*



De igual manera se tiene los datos obtenidos desde los sensores como se ilustra en la Figura 134, donde los datos se están obteniendo desde el contenedor, los datos se reciben en formato json. En este caso se puede verificar los datos que llegan a la terminal conectada al nodo cliente.

**Figura 134**

*Datos obtenidos por los sensores*



## 2. Recepción de datos en el nodo coordinador

Para verificar que los datos están llegando al nodo coordinador se hace el siguiente procedimiento, se habilita el tunslip, para que se establezca la comunicación del nodo coordinador y la máquina virtual.

**Figura 135**

*Habilitación del tunslip*

```
contiki@ubuntu: ~/contiki-ng/examples/rpl-border-pers
contiki@ubuntu:~/contiki-ng/examples/rpl-border-pers$ sudo ../../tools/serial-io
/tunslip6 -s /dev/ttyACM0 fd00::1/64
tunslip6: can't open /dev: ``/dev/ttyACM0'': No such file or directory
contiki@ubuntu:~/contiki-ng/examples/rpl-border-pers$ sudo ../../tools/serial-io
/tunslip6 -s /dev/ttyACM0 fd00::1/64
*****SLIP started on ``/dev/ttyACM0''
opened tun device ``/dev/tun0''
ifconfig tun0 inet `hostname` mtu 1500 up
ifconfig tun0 add fd00::1/64
ifconfig tun0 add fe80::0:0:0:1/64
ifconfig tun0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 127.0.1.1 netmask 255.255.255.255 destination 127.0.1.1
    inet6 fe80::1 prefixlen 64 scopeid 0x20<link>
    inet6 fe80::7109:2a71:fba6:603f prefixlen 64 scopeid 0x20<link>
    inet6 fd00::1 prefixlen 64 scopeid 0x0<global>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 500
(UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[INFO: BR      ] Waiting for prefix
*** Address:fd00::1 => fd00:0000:0000:0000
[INFO: BR      ] Waiting for prefix
[INFO: BR      ] Server IPv6 addresses:
[INFO: BR      ]   fd00::212:4b00:1ca7:5e58
[INFO: BR      ]   fe80::212:4b00:1ca7:5e58
```

Para poder verificar las interfaces de la máquina virtual se ejecuta el siguiente comando ipconfig, donde se puede verificar en este caso todas las interfaces, en este caso la interfaz ens33 es la que se va a configurar con la dirección ipv6, como se indica en la Figura 134.

**Figura 134**

*Identificación de interfaz a asignar ip*

```
contiki@ubuntu: ~  
contiki@ubuntu:~$ ifconfig  
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 192.168.129.128 netmask 255.255.255.0 broadcast 192.168.129.255  
inet6 fe80::20c:29ff:fed1:6db6 prefixlen 64 scopeid 0x20<link>  
ether 00:0c:29:d1:6d:b6 txqueuelen 1000 (Ethernet)  
RX packets 2999 bytes 245854 (245.8 KB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 1017 bytes 130233 (130.2 KB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
inet 127.0.0.1 netmask 255.0.0.0  
inet6 ::1 prefixlen 128 scopeid 0x10<host>  
loop txqueuelen 1000 (Bucle local)  
RX packets 104 bytes 8656 (8.6 KB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 104 bytes 8656 (8.6 KB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500  
inet 127.0.1.1 netmask 255.255.255.255 destination 127.0.1.1  
inet6 fe80::1 prefixlen 64 scopeid 0x20<link>  
inet6 fe80::7109:2a71:fba6:603f prefixlen 64 scopeid 0x20<link>  
inet6 fd00::1 prefixlen 64 scopeid 0x0<global>  
unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 500  
(UNSPEC)  
RX packets 0 bytes 0 (0.0 B)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 5 bytes 240 (240.0 B)
```

En la Figura 135, se muestra el comando para poder configurar la interfaz. De igual manera se puede verificar que se está asignando la dirección.

**Figura 135**

*Comando configuración de ip a la interfaz*

```
contiki@ubuntu:~$  
contiki@ubuntu:~$ sudo ip a a fe00::1/64 dev ens33
```

De igual manera se puede verificar como queda establecida la configuración de la interfaz con la dirección ipv6.

**Figura 136**

*Verificación de interfaz configurada*

```
contiki@ubuntu: ~  
[sudo] contraseña para contiki:  
contiki@ubuntu:~$ ifconfig  
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.129.128 netmask 255.255.255.0 broadcast 192.168.129.255  
    inet6 fe80::20c:28ff:fed1:6db5 prefixlen 64 scopeid 0x20<link>  
    inet6 fe00::1 prefixlen 64 scopeid 0x0<global>  
    ether 08:00:20:c2:8f:fd txqueuelen 1000 (Ethernet)  
    RX packets 3143 bytes 257874 (257.8 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 1088 bytes 139889 (139.8 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Bucle local)  
    RX packets 104 bytes 8656 (8.6 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 104 bytes 8656 (8.6 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
vun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500  
    inet 127.0.1.1 netmask 255.255.255.255 destination 127.0.1.1  
    inet6 fe80::7109:2a71:fba6:603f prefixlen 64 scopeid 0x20<link>  
    inet6 fd00::1 prefixlen 64 scopeid 0x0<global>  
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 500  
(UNSPEC)  
    RX packets 0 bytes 0 (0.0 B)
```

Para que se establezca la comunicación mediante el protocolo MQTT se hace la creación del cliente MQTT el cual se hace creando un usuario en la página Hive MQTT, en el cual se tendrá que colocar las credenciales que van a ir en el archivo Python.sh como se indica en la Figura 137.

**Figura 137**

*Configuración de archivo bridge.sh con credenciales*

```
contiki@ubuntu: ~/python-bridge
GNU nano 4.8 bridge.sh Modificado
#!/bin/sh
# Utiliza el cliente MOSQUITTO-MQTT para publicar en HiveMQ
echo "Script ejecutado...\n"
echo $1

sudo mosquitto_pub -h broker.mqttdashboard.com -t testEc2021 -m $1

^G Ver ayuda ^C Guardar ^W Buscar ^K Cortar Tex ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^T Ortografía ^_ Ir a línea
```

Luego de ellos se para poder ejecutar el archivo de python bridge se coloca el comando en la ruta especifica como se muestra en la Figura 138, donde se especifica la dirección que se configuro anteriormente y el puerto de comunicación en este caso el 4444.

**Figura 138**

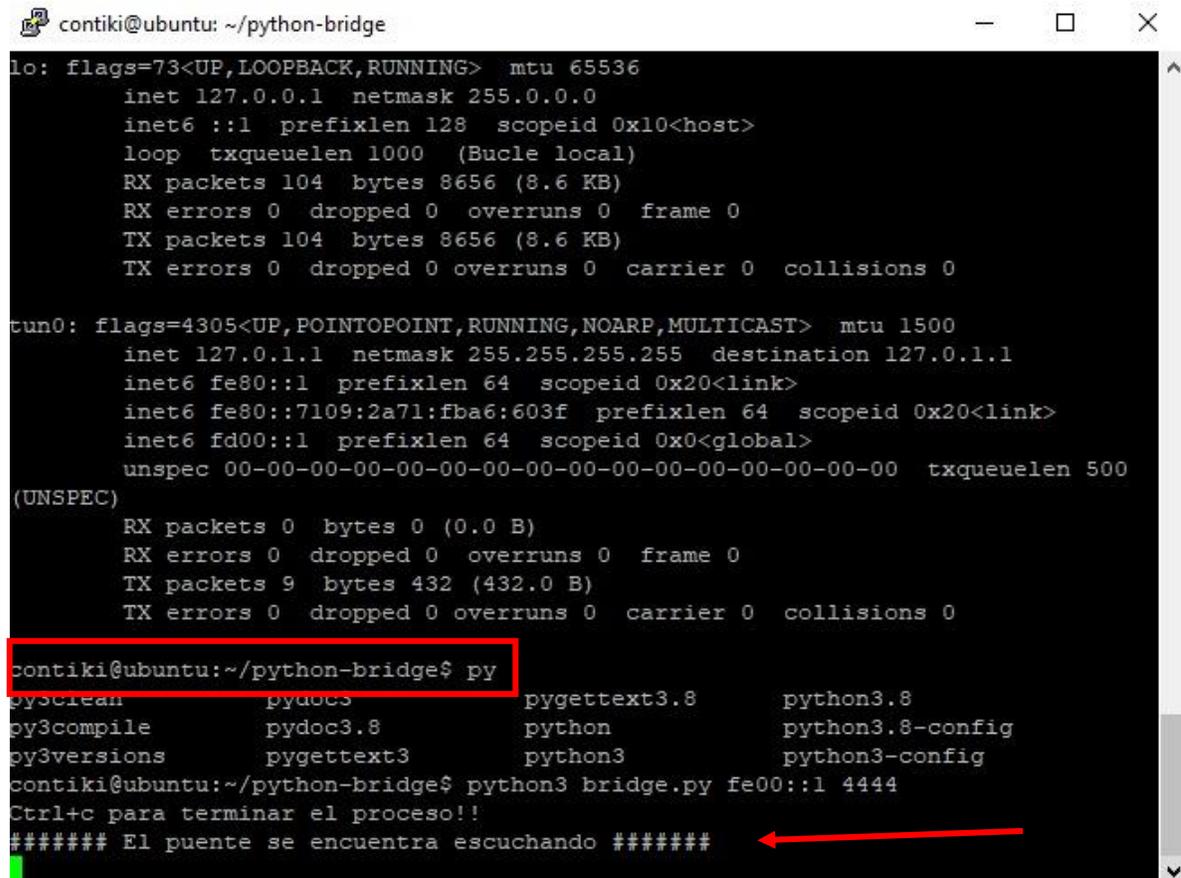
*Configuración de dirección ip y de puerto*

```
contiki@ubuntu:~/python-bridge$ py
py3clean          pydoc3            pygettext3.8     python3.8
py3compile        pydoc3.8         python           python3.8-config
py3versions      pygettext3       python3          python3-config
contiki@ubuntu:~/python-bridge$ python3 bridge.py fe00::1 4444
```

A partir de ahí se muestra que el bridge está a la escucha para poder recibir la información proveniente desde el nodo recolector

**Figura 139**

*Puente se encuentra a la escucha de un envío de datos*



```
contiki@ubuntu: ~/python-bridge
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Bucle local)
    RX packets 104 bytes 8656 (8.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 104 bytes 8656 (8.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 127.0.1.1 netmask 255.255.255.255 destination 127.0.1.1
    inet6 fe80::1 prefixlen 64 scopeid 0x20<link>
    inet6 fe80::7109:2a71:fba6:603f prefixlen 64 scopeid 0x20<link>
    inet6 fd00::1 prefixlen 64 scopeid 0x0<global>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 500
(UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 9 bytes 432 (432.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

contiki@ubuntu:~/python-bridge$ py
py3clean      pydoc3      pygettext3.8 python3.8
py3compile   pydoc3.8   python      python3.8-config
py3versions  pygettext3 python3     python3-config
contiki@ubuntu:~/python-bridge$ python3 bridge.py fe00::1 4444
Ctrl+c para terminar el proceso!!
##### El puente se encuentra escuchando #####
```

Se puede evidenciar en la Figura 140, donde se tiene ya los datos, los cuales va a ser enviado hacia la plataforma Web, donde indican si el contenedor está lleno o vacío, en esta ocasión para que haya constancia se coloca el contenedor lleno, y su estado full lo corrobora.

**Figura 140**

*Datos y estado del contenedor en formato json*



Facebook mail www.bidi.uson.mx/... Ver The Walking De... ute utn biblio Usando el MPU605... Giroscopio y Aceler... paper Kick Ass

ogle.com/mail/u/0/#inbox

**¿Necesita un corredor MQTT completamente administrado?**  
Obtenga su propio agente de la nube y conecte hasta 100 dispositivos de forma gratuita.

Consigue tu cuenta gratis

**Conexión** connected

**Publicar**

Tema: testtopic/1 QoS: 0 Retener:  **Publicar**

Mensaje

**Suscripciones**

Agregar nueva suscripción de tema

Preguntas frecuentes: 2 pruebaEc2021

**Mensajes**

```
{\"contenedores\":  
  [{\"name\": \"1\", \"volumen\": \"84\", \"co2\": \"202\", \"peso\": \"0.88\", \"ubicacion\":  
    {\"lat\": \"0.338000\", \"largo\": \"-78.126751\", \"estado\": \"vacio\"}}]
```

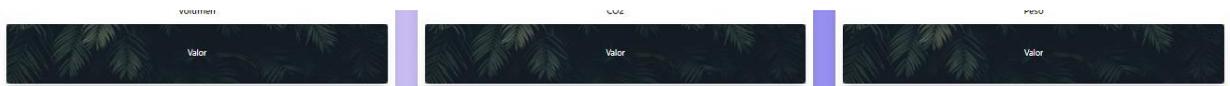
### 3. Publicación de datos en la plataforma Web IOT

Al encontrarse lleno el contenedor se puede evidenciar que los datos se publican en la página, indicando las variables que se estableció como objetivo, donde se muestra el volumen, co2, peso, y la ubicación, la última reflejada en el mapa, como se indica en la Figura 142.

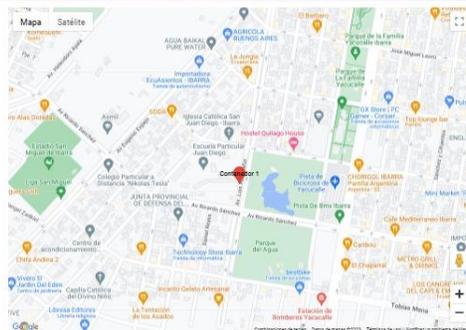
**Figura 142**

*Página actualizada con los datos de un contenedor en tiempo real*

# CONTENEDOR 1



## UBICACION



### 4.2.3. Pruebas de alcance de la red mesh en un ambiente real con línea de vista

Las pruebas de alcance se hacen en un ambiente real, con línea de vista, haciendo uso de las antenas acopladas por defecto en las placas. Como se indica en la Figura 143, se tiene la información del nodo servidor y la dirección IP.

## ESTACION NODO COORDINADOR

Para este caso se toma como estación al nodo coordinador el cual se lo va a ubicar en un punto. En la Figura 143, se indica la información del nodo coordinador y se resalta la dirección ipv6 que lo identifica dentro de la red.

### Figura 143

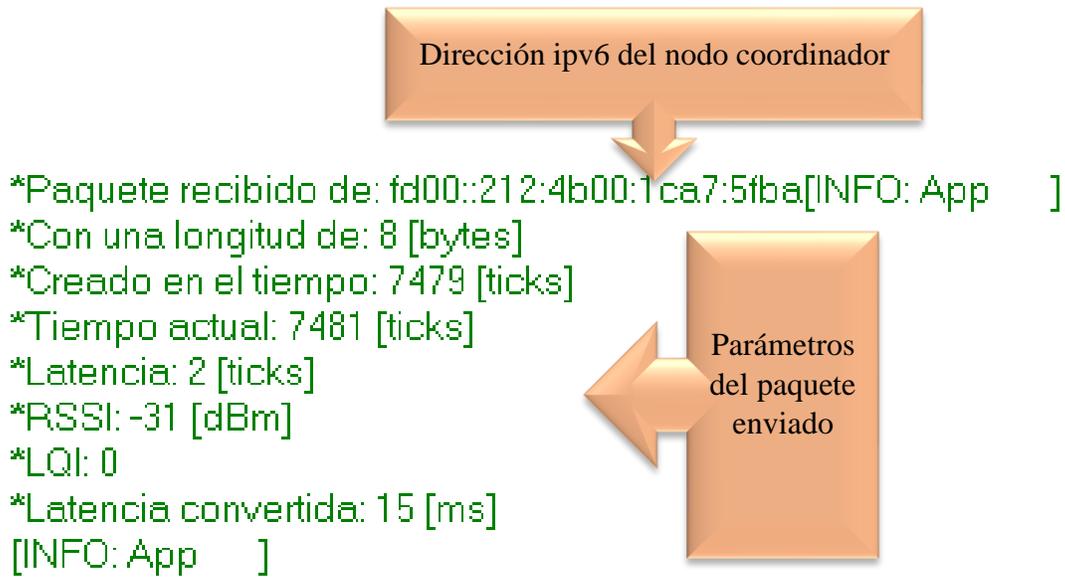
*Información sobre el nodo coordinador*

```
[INFO: Main ] Starting Contiki-NG-release/v4.8-305-g2998a0f5f-dirty
[INFO: Main ] - Routing: RPL Lite
[INFO: Main ] - Net: sicslowpan
[INFO: Main ] - MAC: TSCH
[INFO: Main ] - 802.15.4 PANID: 0x81a5
[INFO: Main ] - 802.15.4 TSCH default hopping sequence length: 4
[INFO: Main ] Node ID: 24152
[INFO: Main ] Link-layer address: 0012.4b00.1ca7.5e58
[INFO: Main ] Tentative link-local IPv6 address: fe80::212:4b00:1ca7:5e58
[INFO: CC13xx/CC26xx] Operating frequency on Sub-1 GHz
[INFO: CC13xx/CC26xx] RF: Channel 15
[INFO: App ]
CLOCK_SECOND: 128
```

En la Figura 144, se muestra la estructura de la presentación de las métricas en este caso en el nodo coordinador, de igual manera se puede observar la dirección ip origen de los datos, la longitud del paquete en este caso 8 bytes, el valor de ticks con el cual se saca el valor de latencia para cada mensaje.

### Figura 144

*Datos enviados desde el nodo servidor hacia el cliente*



Se hace una medición de una distancia aproximadamente 30 m, como se ilustra en la Figura 145.

**Figura 145**

*Datos enviados en la distancia con un valor aproximadamente 30 m*

```

[INFO: App    ]
*Paquete recibido de: fd00::212:4b00:1ca7:5fba[INFO: App    ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 26416 [ticks]
*Tiempo actual: 26424 [ticks]
*Latencia: 8 [ticks]
*RSSI: -71 [dBm]
*LQI: 0
*Latencia convertida: 62 [ms]
[INFO: App    ]
*****
[INFO: App    ]
*Paquete recibido de: fd00::212:4b00:1ca7:5fba[INFO: App    ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 27806 [ticks]
*Tiempo actual: 27822 [ticks]
*Latencia: 16 [ticks]
*RSSI: -72 [dBm]
*LQI: 0
*Latencia convertida: 125 [ms]
[INFO: App    ]
*****
[INFO: App    ]
*Paquete recibido de: fd00::212:4b00:1ca7:5fba[INFO: App    ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 29023 [ticks]
*Tiempo actual: 29036 [ticks]
*Latencia: 13 [ticks]
*RSSI: -85 [dBm]
*LQI: 0
*Latencia convertida: 101 [ms]

```

Se hace una medición hasta que no se dé el envío de datos normal, como se indica en la Figura 146.

### Figura 146

*Datos en distancia 65 m aproximadamente, donde se presenta un mensaje.*

---

```
*Latencia: 16 [ticks]
*RSSI: -72 [dBm]
*LQI: 0
*Latencia convertida: 125 [ms]
]INFO: App [ ]
*****
]INFO: App [ ]
*Paquete recibido de: fd00::212:4b00:1ca7:5fba[INFO: App [ ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 29023 [ticks]
*Tiempo actual: 29036 [ticks]
*Latencia: 13 [ticks]
*RSSI: -85 [dBm]
*LQI: 0
*Latencia convertida: 101 [ms]
]INFO: App [ ]
*****
]INFO: App [ ]
*Paquete recibido de: fd00::212:4b00:1ca7:5fba[INFO: App [ ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 30304 [ticks]
*Tiempo actual: 30311 [ticks]
*Latencia: 7 [ticks]
*RSSI: -85 [dBm]
*LQI: 0
*Latencia convertida: 54 [ms]
WARN: TSCH [ ]! drop dup II from 0012.4b00.1ca7.5fba seqno 253
WARN: TSCH [ ]! drop dup II from 0012.4b00.1ca7.5fba seqno 253
]INFO: App [ ]
*****
]INFO: App [ ]
*Paquete recibido de: fd00::212:4b00:1ca7:5fba[INFO: App [ ]
```

---

## ESTACION RECOLECTOR

Para este caso se toma como estación al nodo recolector el cual se lo va a ubicar en el mismo punto que se encontraba en el nodo coordinador. En la Figura 147, se indica la información del nodo recolector y se resalta la dirección ipv6 que lo identifica dentro de la red.

### Figura 147

*Información del nodo recolector*

```

[INFO: App ] Enviando paquete con timestamp 2875 to fd00::212:4b00:1ca7:5e58
[INFO: Main ] Starting Contiki-NG-release/v4.8-305-g2998a0f5f-dirty
[INFO: Main ] - Routing: RPL Lite
[INFO: Main ] - Net: sicslowpan
[INFO: Main ] - MAC: TSCH
[INFO: Main ] - 802.15.4 PANID: 0x81a5
[INFO: Main ] - 802.15.4 TSCH default hopping sequence length: 4
[INFO: Main ] Node ID: 24506
[INFO: Main ] Link-layer address: 0012.4b00.1ca7.5fba
[INFO: Main ] Tentative link-[INFO: App ] NO alcanzable aun
[WARN: RPL ] just joined, no parent yet, setting timer for leaving

```

Se hace las mismas pruebas de alcance, en este caso el nodo servidor va a variar su posición.

En la Figura 148 se tiene la información en el tiempo 0.

### Figura 148

*Datos de métricas en el instante 0*

```

[INFO: App ]
*ACK recibido de: fd00::212:4b00:1ca7:5e58[INFO: App ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 7468 [ticks]
*Tiempo actual: 7497 [ticks]
*Latencia: 29 [ticks]
*RSSI: -32 [dBm]
*LQI: 0
*Latencia convertida: 226 [ms]
[INFO: App ] Enviando paquete con timestamp 8658 to fd00::212:4b00:1ca7:5e58
[INFO: App ]
*****
[INFO: App ]
*ACK recibido de: fd00::212:4b00:1ca7:5e58[INFO: App ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 8658 [ticks]
*Tiempo actual: 8680 [ticks]
*Latencia: 22 [ticks]
*RSSI: -32 [dBm]
*LQI: 0
*Latencia convertida: 171 [ms]
[INFO: App ] Enviando paquete con timestamp 10057 to fd00::212:4b00:1ca7:5e58
[INFO: App ]

```

En la Figura 149, se ilustra el envío de datos en la distancia 30 m aproximadamente.

## Figura 149

*Datos de métricas en distancia 30m aproximadamente*

```
*****
[INFO: App  ]
*ACK recibido de: fd00::212:4b00:1ca7:5e58[INFO: App  ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 19977 [ticks]
*Tiempo actual: 20002 [ticks]
*Latencia: 25 [ticks]
*RSSI: -83 [dBm]
*LQI: 0
*Latencia convertida: 195 [ms]
[INFO: App  ] Enviando paquete con timestamp 21184 to fd00::212:4b00:1ca7:5e58
[INFO: App  ]
*****
[INFO: App  ]
*ACK recibido de: fd00::212:4b00:1ca7:5e58[INFO: App  ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 21184 [ticks]
*Tiempo actual: 21385 [ticks]
*Latencia: 201 [ticks]
*RSSI: -89 [dBm]
*LQI: 0
*Latencia convertida: 1570 [ms]
[INFO: App  ] Enviando paquete con timestamp 22381 to fd00::212:4b00:1ca7:5e58
[INFO: App  ]
*****
```

Por último se tiene la medición en 65 m aproximadamente, donde también se evidencia que hay valores mayores de latencia como se indica en la Figura 150.

## Figura 150

*Datos en distancia igual a 65m aproximadamente, se tiene mayor latencia.*

```

.....
[INFO: App  ]
*ACK recibido de: fd00::212:4b00:1ca7:5e58[INFO: App  ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 27675 [ticks]
*Tiempo actual: 28530 [ticks]
*Latencia: 855 [ticks]
*RSSI: -86 [dBm]
*LQI: 0
*Latencia convertida: 6679 [ms]
[INFO: App  ] Enviando paquete con timestamp 28849 to fd00::212:4b00:1ca7:5e58
[INFO: App  ]
.....

[INFO: App  ]
*ACK recibido de: fd00::212:4b00:1ca7:5e58[INFO: App  ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 28849 [ticks]
*Tiempo actual: 29420 [ticks]
*Latencia: 571 [ticks]
*RSSI: -78 [dBm]
*LQI: 0
*Latencia convertida: 4460 [ms]
[INFO: App  ] Enviando paquete con timestamp 30047 to fd00::212:4b00:1ca7:5e58
[INFO: App  ]
.....

```

#### 4.2.4. Evaluación del desempeño de la Red Mesh

Para la evaluación del desempeño de la red mesh, se considera las siguientes métricas:

**PDR:** Esta métrica es independiente de la distancia, se hace una transmisión de un determinado número de paquetes. Es así como para el PDR se hace un cálculo basado en la fórmula, en este caso para un total de 70 paquetes transmitidos, se tiene el valor de 1, ya que no hay diferencia entre la relación que indica la fórmula de cálculo, su valor es 1.

#### Figura 151

*Valor de PDR en la transmisión*

```
#####
[INFO: App ]
Paquetes Transmitidos: 70
Paquetes Recibidos: 70
Paquetes Perdidos: 0

PER: 0
PDR: 1
[INFO: App ] Enviando paquete con timestamp 94595 to fd00::212:4b00:1ca7:5e58
[INFO: App ]
*****
```

**PER:** el cual corresponde a la tasa de entrega de paquetes la cual es calculada basándose en la relación de los paquetes enviados y los paquetes recibidos dentro de la red, por lo que se tendrá el porcentaje de fiabilidad en la entrega de paquetes, para una cantidad de 60 paquetes enviados se tiene un PER de 0 en este caso ya que se entregaron y recibieron todos los datos. En la Figura 152, se muestra el valor de PER.

### Figura 152

*Valor de PER en la transmisión*

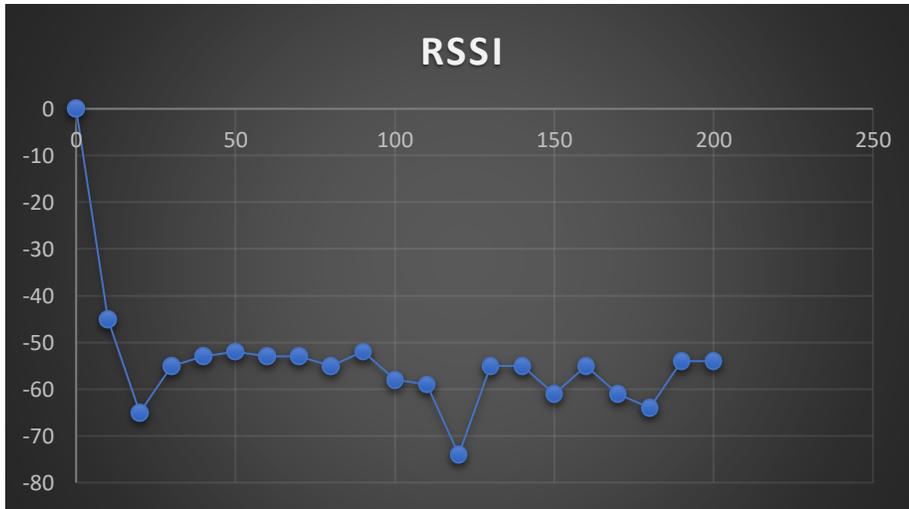
```
#####
[INFO: App ]
Paquetes Transmitidos: 60
Paquetes Recibidos: 60
Paquetes Perdidos: 0

PER: 0
PDR: 1
[INFO: App ] Enviando paquete con timestamp 81511 to fd00::212:4b00:1ca7:5e58
[INFO: App ]
*****
```

**Potencia RSSI.** La potencia RSSI corresponde a la fuerza de la señal recibida, o bien conocida como la potencia. En la Figura 153, se muestra un gráfico donde indica que para un total de 20 paquetes enviados se tiene un RSSI variable dependiendo del escenario donde se desarrolla, en este caso se tiene los nodos a una distancia de 1 m aproximadamente.

### Figura 153

*Grafica RSSI en función del tiempo*



**LQI:** Esta métrica representa el número de retransmisiones requeridas para recibir con éxito un paquete. Permite determinar la probabilidad de entrega exitosa de paquetes. En la Figura 154, se evidencia que nos brinda un valor de cero, su valor no varía.

**Figura 154**

*Valor de LQI para la transmisión*

```

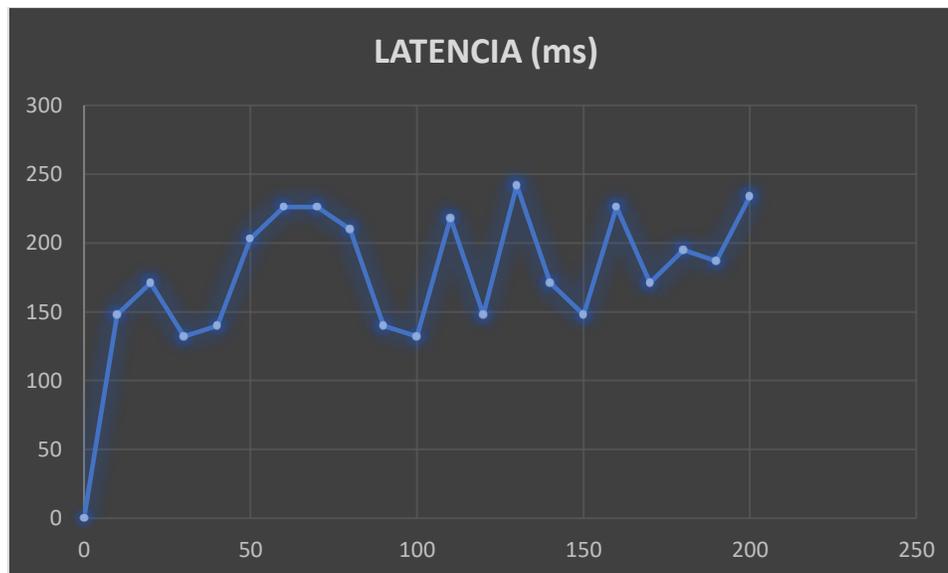
*****
[INFO: App ]
*ACK recibido de: fd00::212:4b00:1ca7:5e58[INFO: App ]
*Con una longitud de: 8 [bytes]
*Creado en el tiempo: 81511 [ticks]
*Tiempo actual: 81532 [ticks]
*Latencia: 21 [ticks]
*RSSI: -56 [dBm]
*LQI: 0
  Latencia convertida: 164 [ms]
[INFO: App ] Enviando paquete con timestamp 82686 to fd00::212:4b00:1ca7:5e58
[INFO: App ]
*****

```

**Retardo.** Corresponde en la medida del tiempo que se demora un paquete en llegar desde el origen hasta el destino, en la Figura 155, se evidencia que, para un total de 20 paquetes enviados, cada 10 segundos, se tiene una gráfica de la latencia presentada en estas transmisiones.

**Figura 155**

*Valores de latencia para cada paquete enviado*



### **4.3. Beneficios del Sistema**

A continuación, se menciona los beneficios que trae la aplicación del sistema en el entorno.

La aplicación del sistema en un entorno real permite tener un control del volumen de los contenedores evitando el desbordamiento de los desechos, de igual manera permitiendo que se establezcan rutas de recolección, reduciendo costos y haciendo efectivo el proceso de recolección.

Con la implementación del sistema en los contenedores de residuos se minimiza la probabilidad de que se genere algún gas u olores desagradables que molesten a la población. Así como también mejorar el saneamiento público.

El sistema brinda la ubicación exacta del contenedor lleno con lo que la persona encargada del vehículo recolector establece nuevas rutas en base a la posición de los

contenedores, esto conlleva a tener un ahorro en el combustible y a la vez disminuir la generación de dióxido de carbono y otras emisiones.

Mediante el uso del sistema se reduce la frecuencia de recolección, y se aporta a disminuir el tráfico, al menos en ciudades grandes con gran cantidad de vehículos.

# CONCLUSIONES Y RECOMENDACIONES

## Conclusiones

Las redes de baja potencia permiten economizar energía que se puede implementar en ambientes IOT donde se puede recibir datos que no necesariamente va a estar enviándose a cada segundo, sino más bien tener un control del sistema, para la toma de decisiones.

Las placas seleccionadas para el trabajo permiten trabajar a una distancia aproximada de 65 m con las antenas acopladas por defecto, sin embargo, si se quiere alcanzar mayores distancias, es necesario colocar antenas de mayor alcance, pero hay que tener las herramientas y conocimientos adecuados para poder modificar las placas.

Una red mesh permite tener un control en la convergencia, y el alcance de todos los nodos, permitiendo de esa manera que se tenga un enrutamiento de la información confiable sin pérdida de datos.

Al trabajar en ambientes reales es necesario que se tenga los recursos necesarios para poder implementar un sistema como el de estudio debido a que muchas veces no se tiene el apoyo de autoridades competentes en esos aspectos, lo cual puede llegar a ser un inconveniente al querer realizar pruebas en lugares específicos y con recursos públicos.

## **Recomendaciones**

Se recomienda tener claro el material teórico que se va a necesitar para el desarrollo del trabajo, debido a que muchas veces por falta de fuentes bibliográficas, no se puede avanzar de manera significativa.

Al trabajar con placas nuevas es necesario que se tenga bases que permitan entender el funcionamiento básico de los microcontroladores, así como también conocimiento en el manejo tanto de la electrónica y redes como tal.

Al ensamblar el prototipo tener cuidado con las conexiones porque se puede llegar a generar algún error en las mediciones por mala manipulación del mismo.

## Referencias Bibliograficas

- Acosta, M. (2006). *Estudio del estándar IEEE 802.15.4 "ZIGBEE" para comunicaciones inalámbricas de área personal de bajo consumo de energía y su comparación con el estándar IEEE 802.15.1 "BLUETOOTH."* www.epn.edu.ec
- Archundia, F. M. (2007). El estándar IEEE 802.15.4. *WPAN and Home Networking*, 1–18.
- Cadena, E., & Moran, M. (2018). *"AMPLIACIÓN E INNOVACIÓN DEL MICROMERCADO PANADERÍA Y VÍVERES EN EL BARRIO YACUCALLE 5º ETAPA DE LA CIUDAD DE IBARRA, PROVINCIA DE IMBABURA. PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA COMERCIAL.*
- Cali Francisco. (2018). *IMPLEMENTACIÓN DEL ALGORITMO DE PROTOCOLO DE DIRECCIONAMIENTO PARA REDES DE SENSORES INALÁMBRICOS CON EL ESTANDAR IEEE 802.15.4.*
- Dignani, J. P., & Tinetti, F. G. (2011). *ANÁLISIS DEL PROTOCOLO ZIGBEE Alumno.*
- el Norte. (2018). *Ibarra produce 135 toneladas de basura a diario.* <https://www.elnorte.ec/ibarra/ibarra-produce-135-toneladas-de-basura-a-diario-JH87081>
- Electrónicas Didácticas. (2020). *Fuerza y Vibracion: Celda de carga de 50K.* <https://www.didacticaselectronicas.com/index.php/sensores/fuerza-y-vibracion/celda-de-carga-de-50k-celdas-de-carga-galgas-sensores-de-peso-fuerza-tension-con-estructura-de-medio-puente-de-50kg-detail>
- Farahani, S. (2008). *Clear Channel Assessment MulteFire Alliance IoT technologies.*
- Godoy, P., Iacono, L., Marianetti, O., & García Garino, C. (2010). *Segundo Congreso Virtual, Microcontroladores y sus Aplicaciones.*
- González, F., Calero, S., & Loaiza, D. (2019). *Comparación de las metodologías cascada y ágil para el aumento de la productividad en el desarrollo de software.*
- Huilca Salcedo, J. G., & SichiQUI Velecela, P. F. (2019). *Diseño e implementación de un sistema embebido de monitoreo de las variables climáticas para plantaciones de maíz.* 134. <http://dspace.ups.edu.ec/handle/123456789/17338>
- IEEE Comsoc. (2020). *IEEE Standard for Low - Rate Wireless Networks - IEEE Std 802.15.4 - 2020* (Vol. 2020). [https://standards.ieee.org/standard/802\\_15\\_4-2020.html](https://standards.ieee.org/standard/802_15_4-2020.html)
- INEC. (2010). *Población y Demografía.* <https://www.ecuadorencifras.gob.ec/censo-de-poblacion-y-vivienda/>
- Instruments, T. (2019). *CC1312R SimpleLink™ High-Performance Sub-1 GHz Wireless MCU* (Issue January 2018).

- López Vilos, N. A., & Valencia Cordero, C. (2017). *Constant Jamming Experimental Results on Indoor Wireless Sensor Network*.
- Mahedero, F. (2020). *DESARROLLO DE UNA APLICACIÓN IoT PARA EL ENVÍO DE IMÁGENES MEDIANTE EL PROTOCOLO MQTT*.
- Ministerio del Ambiente. (2015). *Reforme El Libro VI del Texto Unificado de Legislación Secundaria del Ministerio del Ambiente*.
- Murillo, A. (2017). *Evaluación del desempeño en la transmisión de señales biomédicas en un ambiente inalámbrico en redes 6LoWPAN*.  
<https://cicese.repositorioinstitucional.mx/jspui/bitstream/1007/899/1/TesisVersionFinal.pdf>
- Nahuel Vara, G. A. P. Dr. M. C. Dr. A. J. B. (2010). *CALCULO DE DISTANCIA ENTRE LOS NODOS DE UNA RE D INALÁMBRICA ZIGBEE EN FUNCIÓN DEL PARÁMETRO RSSI*.
- Panama Hitek. (n.d.). *Características Arduino Mega 2560*. Retrieved April 2, 2022, from <http://panamahitek.com/arduino-mega-caracteristicas-capacidades-y-donde-conseguirlo-en-panama/>
- Strauch, F. M. (2000). *COMUNICACIÓN EN TIEMPO REAL SOBRE INTERNET*.
- Trinidad, O. O. (2020). *Estudio de la viabilidad de RPL y Contiki para un entorno de sensorización en ganadería* (Vol. 5).
- Vadillo Gutiérrez, Ó. (2014). *Provisión de servicios de la Internet de las Cosas sobre redes de sensores basadas en 6LoWPAN*. <http://repositorio.unican.es/xmlui/handle/10902/6023>
- Xue, L. (2015). *Analysis, Simulations and Improvement Proposals of IEEE 802.15.4g-2012*.

## **Anexos**

### **ANEXO A: Entrevista realizada al encargado del Departamento de Gestión de Residuos Sólidos de la Ciudad de Ibarra.**

**UNIVERSIDAD TÉCNICA DEL NORTE**

**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN**

**El presente documento corresponde a la entrevista realizada al Ing. Manuel Enríquez encargado de la Gestión de Residuos Sólidos del Municipio de Ibarra.**

**1. Como se realiza la recolección de los residuos sólidos.**

Los operarios tienen establecido el sector, el horario y días de recolección, visitan cada sector recogiendo cada contenedor que se encuentra en el sitio para posteriormente transportarlo al centro de transferencia socapamba.

**2. ¿Existen rutas establecidas para la recolección? Si la respuesta es Si en base a que fundamento se las establece.**

Si, las rutas las establece los encargados del vehículo recolector de residuos sólidos, únicamente tiene conocimiento del sector, y el horario mas no de una ruta concreta a seguir para la recolección.

**3. ¿El sector Yacucalle a que ruta corresponde y cuál es su horario y días de recolección?**

Corresponde a la ruta Yacucalle, Pílanqui P1 y Los Ceibos P.2, que se lleva a cabo de lunes a domingo en horario de 07h00 a 13h00.

4. ¿Para seguir las rutas consideran los contenedores que ya están llenos?

El VRR se desplaza calle por calle sin tomar en consideración el volumen de cada contenedor. Eso depende del encargado del VRR.

**5. ¿Cuán importante cree usted que se establezcan las rutas en base al llenado de contenedores?**

Sería de gran ayuda ya que el VRR tiene la posibilidad de obviar sectores en donde no se encuentran los contenedores llenos, y con ellos se ahorraría en combustible, lo que beneficiaría de manera económica.

**ANEXO B. Ficha de Observación aplicada al lugar de estudio.**

<b>UNIVERSIDAD TÉCNICA DEL NORTE</b>				
<b>FACULTAS DE INGENIERÍA EN CIENCIAS APLICADAS</b>				
<b>CARRERA DE INGENIERÍA EN ELECTRÓNICA Y</b>				
<b>REDES DE COMUNICACIÓN</b>				
<b>RESPONSABLE:</b> Andrea Cuásquer				
La presente información es recolectada con el propósito de determinar las condiciones en las que se maneja la recolección de residuos sólidos, así como también la distribución de contenedores de residuos sólidos en el sector Yacucalle de la ciudad de Ibarra.				
<b>REGISTRO DIARIO DE OBSERVACIÓN</b>				
<b>FECHA:</b>	<b>SITUACIONES A OBSERVAR</b>			
	Distribución de contenedores de residuos sólidos en el sector de estudio.			
<b>SECTOR:</b>	Número de contenedores por cuadra			
Yacucalle - Ibarra	Número total de contenedores en el sector Yacucalle			
<b>PAISAJE</b>	<b>CRONOLOGIA</b>	<b>OBSERVACIÓN</b>	<b>SI</b>	<b>NO</b>
Yacucalle	11h00 am	Contenedores distribuidos a igual distancia unos de otros		X
Yacucalle	11h00 am	Presencia de un contenedor por cuadra		X

Yacucalle	11h00 am	Presencia de dos o más contenedores en zonas con mayor concentración de personas	X	
Yacucalle	11h00 am	Distancia aproximada entre contenedores aledaños mayor a 300 m		X
Yacucalle	11h00 am	Distancia aproximada entre contenedores aledaños menor a 20 m		X
Yacucalle	11h00 am	Todos los contenedores están en perfecto estado físico		X

**ANEXO C: Encuesta aplicada a personas que residen en el sector Yacucalle de la ciudad de Ibarra.**

**UNIVERSIDAD TÉCNICA DEL NORTE**

**FACULTAS DE INGENIERÍA EN CIENCIAS APLICADAS**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN**

**ENCUESTA**

La información a obtener es de uso confidencial y tiene como objetivo recolectar datos sobre el proceso de depósito de los residuos sólidos en el sector Yacucalle de la ciudad de Ibarra. Agradezco su colaboración en beneficio de la presente investigación.

N°	Pregunta
1.	Por lo general que días deposita los residuos sólidos en el contenedor más cercano.  <input type="radio"/> Lunes  <input type="radio"/> Martes  <input type="radio"/> Miércoles  <input type="radio"/> Jueves  <input type="radio"/> Viernes
2.	Qué hora del día hace el depósito de los residuos sólidos en el contenedor  <input type="radio"/> 6h00 am-9h00 am

	<ul style="list-style-type: none"> <li>○ 9h00 am-12h00 am</li> <li>○ 12h00 am-15h00</li> <li>○ 15h00-18h00</li> <li>○ Después de las 18h00</li> </ul>		
3.	<p>Con que frecuencia debería pasar el carro recolector de desechos en su barrio.</p> <ul style="list-style-type: none"> <li>○ Una vez por semana</li> <li>○ Dos-tres veces por semana</li> <li>○ Mas de tres veces por semana</li> </ul>		
	Pregunta	SI	NO
4.	<p>Le afecta en gran magnitud que los contenedores de residuos sólidos estén demasiado llenos.</p>		
5.	<p>Los contenedores aledaños con frecuencia se encuentran al límite de su capacidad.</p>		

6.	Se encuentra conforme con el proceso de recolección de los residuos sólidos en su barrio		
7.	Le gustaría que se haga un control del llenado de los contenedores de residuos sólidos para evitar el desbordamiento		
8.	Le gustaría que el vehículo recolector vacié los contenedores antes de que estos sobrepasen el límite.		
9.	Sabía usted que con ayuda de la tecnología se puede		

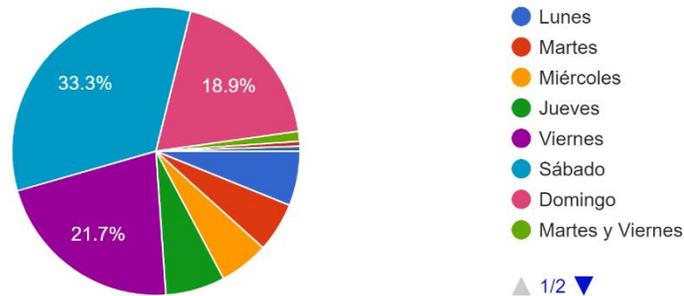
	tener un control del volumen de los contenedores de residuos sólidos.		
10.	Piensa usted que al tener un control del llenado de los contenedores de residuos sólidos pueda tener beneficios en la comunidad.		

## ANEXO D: Tabulación de las preguntas contenidas en las encuestas aplicadas al sector Yacucalle.

### 1. Por lo general que días deposita los residuos sólidos en el contenedor más cercano.

Por lo general que días deposita los residuos sólidos en el contenedor más cercano.

180&nbsp;respuestas

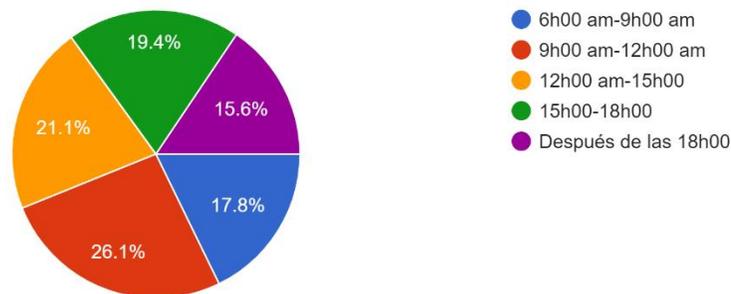


Como se puede observar en el resultado las personas no tienen un día específico para hacer el depósito, aunque la mayoría radica en los fines de semana con un 21.7% los viernes, un 33.3% los sábados y un 18.9% los días domingo.

### 2. Qué hora del día hace el depósito de los residuos sólidos en el contenedor

Qué hora del día hace el depósito de los residuos sólidos en el contenedor

180&nbsp;respuestas

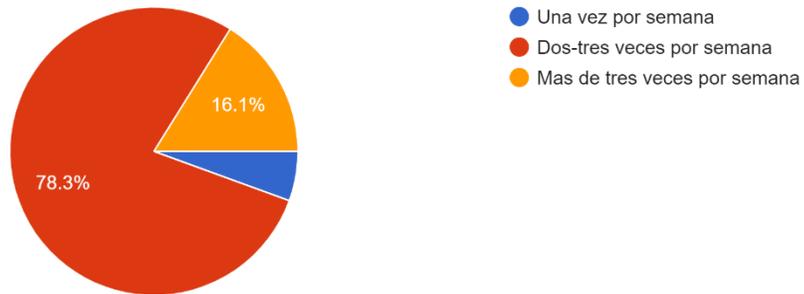


Como se puede observar las horas de depósito varían, sin embargo la mayoría está entre las horas de la mañana de 9h00 a 12h00 que vendrían de la mano con fines de semana.

3. Con que frecuencia debería pasar el carro recolector de desechos en su barrio.

Con que frecuencia debería pasar el carro recolector de desechos en su barrio.

180 respuestas

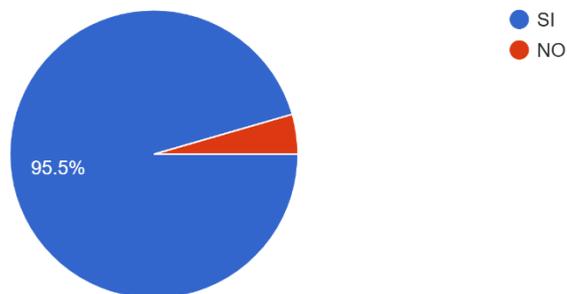


En los resultados de la presente pregunta se nota que un 78.3% opta por responder que el vehículo recolector de residuos pase por la zona de dos a tres veces por semana.

4. Le afecta en gran magnitud que los contenedores de residuos sólidos estén demasiado llenos.

Le afecta en gran magnitud que los contenedores de residuos sólidos estén demasiado llenos.

179 respuestas

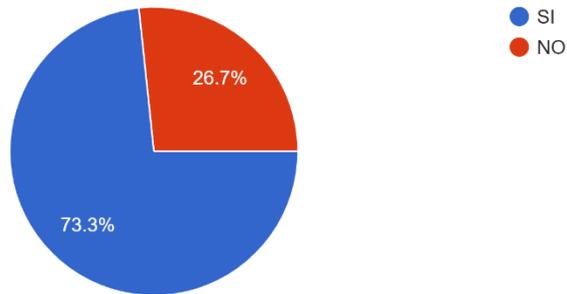


Como se aprecia en el gráfico una mayoría que corresponde al 95.5% se siente afectado que los contenedores se encuentren al límite de su capacidad.

5. Los contenedores aledaños con frecuencia se encuentran al límite de su capacidad.

Los contenedores aledaños con frecuencia se encuentran al límite de su capacidad.

180 respuestas

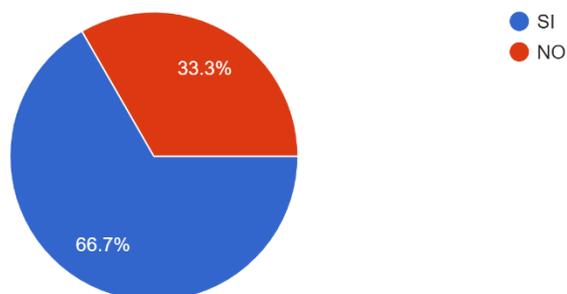


Según el resultado se puede evidenciar que los contenedores de la zona con frecuencia se encuentran llenos, sin embargo, existen zona en las cuales hay mayor numero de contenedores que hace que exista una distribución de residuos en todos los contenedores lo que hace que los encuestados no tengan inconveniente.

6. Se encuentra conforme con el proceso de recolección de los residuos sólidos en su barrio.

Se encuentra conforme con el proceso de recolección de los residuos sólidos en su barrio

180 respuestas

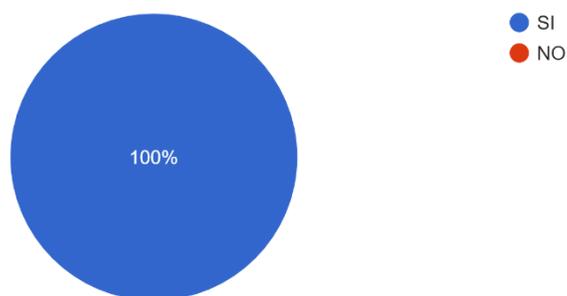


La mayoría que corresponde 66.7% esta conforme con la recolección de residuos sólidos. Sin embargo, el otro porcentaje asume que se puede mejorar con la propuesta realizada.

7. Le gustaría que se haga un control del llenado de los contenedores de residuos sólidos para evitar el desbordamiento.

Le gustaría que se haga un control del llenado de los contenedores de residuos sólidos para evitar el desbordamiento

180 respuestas

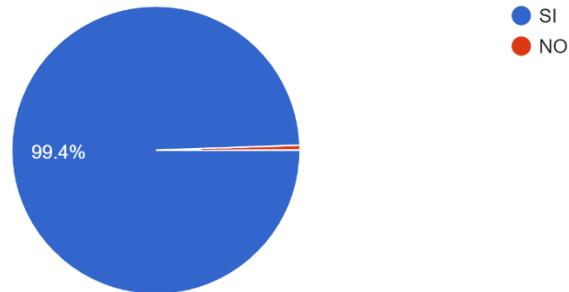


El total de los encuestados esta de acuerdo que de tenga un control del llenado de los residuos sólidos.

8. Le gustaría que el vehículo recolector vacié los contenedores antes de que estos sobrepasen el límite.

Le gustaría que el vehículo recolector vacíe los contenedores antes de que estos sobrepasen el límite.

180 respuestas

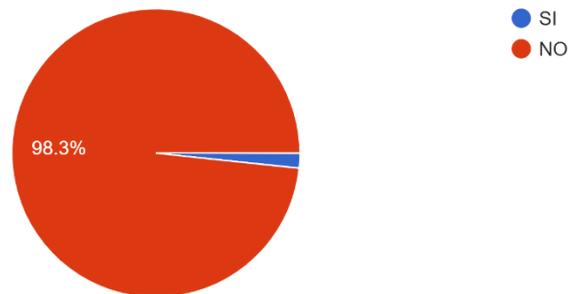


En esta ocasión la mayoría que responde está de acuerdo con que el vehículo recolector vacíe los contenedores ya llenos para que no exista contaminación y presencia de animales en la zona debido a este inconveniente.

9. Sabía usted que con ayuda de la tecnología se puede tener un control del volumen de los contenedores de residuos sólidos.

Sabía usted que con ayuda de la tecnología se puede tener un control del volumen de los contenedores de residuos sólidos.

180 respuestas

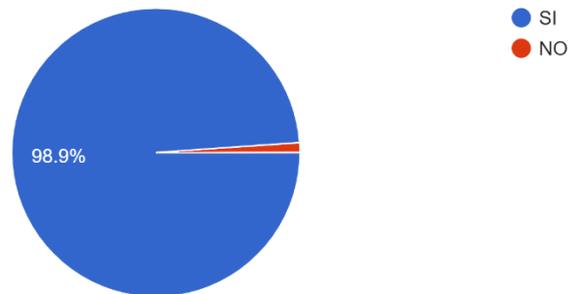


Un 98.3% no tiene conocimiento se que se puede aplicar herramientas tecnológicas orientadas al control del llenado de los contenedores de residuos sólidos.

10. Piensa usted que al tener un control del llenado de los contenedores de residuos sólidos pueda tener beneficios en la comunidad.

Piensa usted que al tener un control del llenado de los contenedores de residuos sólidos pueda tener beneficios en la comunidad.

180 respuestas



Un 98.9 % asume que gracias al control del volumen de los contenedores de residuos sólidos pueden evitar el malestar de la comunidad, evitando la contaminación, presencia de fauna urbana.

## ANEXO E: Datasheet Arduino MEGA 2560



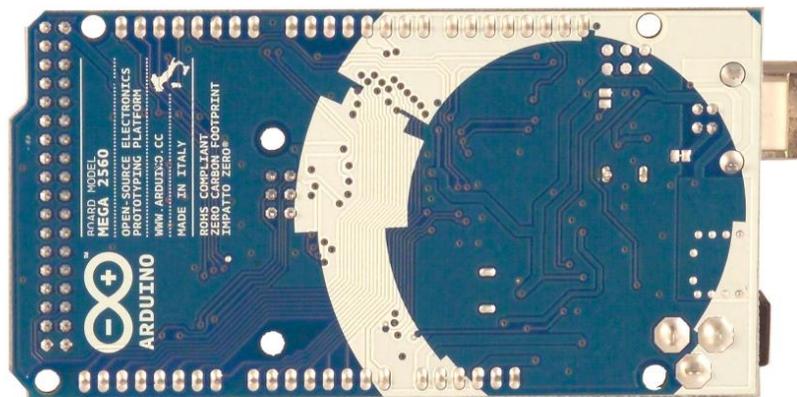
## Arduino Mega 2560 Datasheet





www.robotshop.com

La robotique à votre service! - Robotics at your service!



## Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

## Schematic & Reference Design

EAGLE files: [arduino-mega2560-reference-design.zip](#)



www.robotshop.com

La robotique à votre service! - Robotics at your service!



Schematic: [arduino-mega2560-schematic.pdf](#)

## Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

## Power

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

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.



www.robotshop.com

La robotique à votre service! - Robotics at your service!



The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

## Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

## Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH



www.robotshop.com



La robotique à votre service! - Robotics at your service!

- value, the LED is on, when the pin is LOW, it's off.
- **I<sup>2</sup>C: 20 (SDA) and 21 (SCL).** Support I<sup>2</sup>C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I<sup>2</sup>C pins on the Duemilanove or Decimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and `analogReference()` function.

There are a couple of other pins on the board:

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

## Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I<sup>2</sup>C (TWI) and SPI communication. The Arduino software includes a `Wire` library to simplify use of the I<sup>2</sup>C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, use the [SPI library](#).

## Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It



www.robotshop.com

La robotique à votre service! - Robotics at your service!



communicates using the original STK500 protocol ([reference](#), [C header files](#)). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

## Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

## USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

## Physical Characteristics and Shield Compatibility



[www.robotshop.com](http://www.robotshop.com)



**La robotique à votre service! - Robotics at your service!**

The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and Duemilanove / Diecimila. *Please note that I<sub>2</sub>C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).*

# ANEXO F: Datasheet Modulo CC13523R1



CC1352R

SWRS196I – JANUARY 2018 – REVISED FEBRUARY 2021

## CC1352R SimpleLink™ High-Performance Multi-Band Wireless MCU

### 1 Features

- Microcontroller
  - Powerful 48-MHz Arm® Cortex®-M4F processor
  - EEMBC CoreMark® score: 148
  - 352KB of in-system programmable flash
  - 256KB of ROM for protocols and library functions
  - 8KB of cache SRAM (alternatively available as general-purpose RAM)
  - 80KB of ultra-low leakage SRAM. The SRAM is protected by parity to ensure high reliability of operation.
  - 2-pin cJTAG and JTAG debugging
  - Supports over-the-air (OTA) update
- Ultra-low power sensor controller with 4KB of SRAM
  - Sample, store, and process sensor data
  - Operation independent from system CPU
  - Fast wake-up for low-power operation
- TI-RTOS, drivers, bootloader, *Bluetooth*® 5.2 low energy controller, and IEEE 802.15.4 MAC in ROM for optimized application size
- RoHS-compliant package
  - 7-mm × 7-mm RGZ VQFN48 (28 GPIOs)
- Peripherals
  - Digital peripherals can be routed to any GPIO
  - 4× 32-bit or 8× 16-bit general-purpose timers
  - 12-bit ADC, 200 kSamples/s, 8 channels
  - 2× comparators with internal reference DAC (1× continuous time, 1× ultra-low power)
  - Programmable current source
  - 2× UART
  - 2× SSI (SPI, MICROWIRE, TI)
  - I<sup>2</sup>C and I<sup>2</sup>S
  - Real-time clock (RTC)
  - AES 128- and 256-bit cryptographic accelerator
  - ECC and RSA public key hardware accelerator
  - SHA2 accelerator (full suite up to SHA-512)
  - True random number generator (TRNG)
  - Capacitive sensing, up to 8 channels
  - Integrated temperature and battery monitor
- External system
  - On-chip buck DC/DC converter
  - TCXO support
- Low power
  - Wide supply voltage range: 1.8 V to 3.8 V
  - Active mode RX: 5.8 mA (3.6 V, 868 MHz), 6.9 mA (3.0 V, 2.4 GHz)
  - Active mode TX 0 dBm: 8.0 mA (3.6 V, 868 MHz), 7.1 mA (3.0 V, 2.4 GHz)
  - Active mode TX at +14 dBm: 24.9 mA (868 MHz)
  - Active mode MCU 48 MHz (CoreMark): 2.9 mA (60 µA/MHz)
  - Sensor controller, low power-mode, 2 MHz, running infinite loop: 30.1 µA
  - Sensor controller, active mode, 24 MHz, running infinite loop: 808 µA
  - Standby: 0.85 µA (RTC on, 80KB RAM and CPU retention)
  - Shutdown: 150 nA (wake-up on external events)
- Radio section
  - Multi-band sub-1 GHz and 2.4 GHz RF transceiver compatible with Bluetooth 5.2 Low Energy and earlier LE specifications, and IEEE 802.15.4 PHY and MAC
  - 3-wire, 2-wire, 1-wire PTA coexistence mechanisms
  - Excellent receiver sensitivity:
    - 121 dBm for SimpleLink long-range mode
    - 110 dBm at 50 kbps, -105 dBm for Bluetooth 125-kbps (LE Coded PHY)
  - Output power up to +14 dBm (Sub-1 GHz) and +5 dBm (2.4 GHz) with temperature compensation
  - Suitable for systems targeting compliance with worldwide radio frequency regulations
    - ETSI EN 300 220 Receiver Category 1.5 and 2, EN 300 328, EN 303 131, EN 303 204 (Europe)
    - EN 300 440 Category 2
    - FCC CFR47 Part 15
    - ARIB STD-T108 and STD-T66
  - Wide standard support
- Wireless protocols
  - *Thread*, *Zigbee*®, *Bluetooth*® 5.2 Low Energy, IEEE 802.15.4g, IPv6-enabled smart objects (6LoWPAN), MIOTY®, Wireless M-Bus, Wi-SUN®, KNX RF, Amazon Sidewalk, proprietary systems, SimpleLink™ TI 15.4-Stack (Sub-1 GHz), and dynamic multiprotocol manager (DMM) driver.
- Development *Tools and Software*



An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

- CC1352R LaunchPad™ Development Kit
- SimpleLink™ CC13x2 and CC26x2 Software Development Kit (SDK)
- SmartRF™ Studio for simple radio configuration
- Sensor Controller Studio for building low-power sensing applications
- Grid communications – [wireless communications](#) – long-range sensor applications
- Other alternative energy – [energy harvesting](#)
- Industrial transport – [asset tracking](#)
- Factory automation and control
- Medical
- Electronic point of sale (EPOS) – [Electronic Shelf Label \(ESL\)](#)
- Communication equipment
  - [Wired networking](#) – [wireless LAN or Wi-Fi access points](#), [edge router](#)
- Personal electronics
  - [Portable electronics](#) – [RF smart remote control](#)
  - [Home theater & entertainment](#) – [smart speakers](#), [smart display](#), [set-top box](#)
  - [Connected peripherals](#) – [consumer wireless module](#), [pointing devices](#), [keyboards and keypads](#)
  - [Gaming](#) – [electronic and robotic toys](#)
  - [Wearables \(non-medical\)](#) – [smart trackers](#), [smart clothing](#)

## 2 Applications

- 169, 433, 470 to 510, 868, 902 to 928, and 2400 to 2480 MHz ISM and SRD systems<sup>1</sup> with down to 4 kHz of receive bandwidth
- [Building automation](#)
  - [Building security systems](#) – [motion detector](#), [electronic smart lock](#), [door and window sensor](#), [garage door system](#), [gateway](#)
  - [HVAC](#) – [thermostat](#), [wireless environmental sensor](#), [HVAC system controller](#), [gateway](#)
  - [Fire safety system](#) – [smoke and heat detector](#), [fire alarm control panel \(FACP\)](#)
  - [Video surveillance](#) – [IP network camera](#)
  - [Elevators and escalators](#) – [elevator main control panel for elevators and escalators](#)
- [Grid infrastructure](#)
  - [Smart meters](#) – [water meter](#), [gas meter](#), [electricity meter](#), and [heat cost allocators](#)

## 3 Description

The SimpleLink™ CC1352R device is a multiprotocol and multi-band Sub-1 GHz and 2.4-GHz wireless microcontroller (MCU) supporting [Thread](#), [Zigbee®](#), [Bluetooth® 5.2 Low Energy](#), IEEE 802.15.4g, IPv6-enabled smart objects (6LoWPAN), MIOTY®, Wi-SUN®, proprietary systems, including the TI 15.4-Stack (Sub-1 GHz and 2.4 GHz), and [concurrent multiprotocol](#) through a Dynamic Multiprotocol Manager (DMM) driver. The device is optimized for low-power wireless communication and advanced sensing in [building security systems](#), [HVAC](#), [smart meters](#), [medical](#), [wired networking](#), [portable electronics](#), [home theater & entertainment](#), and [connected peripherals](#) markets. The highlighted features of this device include:

- Multi-band device supporting concurrent multiprotocol for both Sub-1 GHz and 2.4 GHz through a DMM driver.
- Wide flexibility of protocol stack support in the [SimpleLink™ CC13x2 and CC26x2 Software Development Kit \(SDK\)](#).
- Maximum transmit power of +14 dBm at Sub-1 GHz with 24.9 mA and +5 dBm at 2.4 GHz with 9.6 mA current consumption.
- Longer battery life wireless applications with low standby current of 0.85 µA and full RAM retention.
- Industrial temperature ready with lowest standby current of 5 µA at 85°C.
- Advanced sensing with a programmable, autonomous ultra-low power [Sensor Controller CPU](#) with fast wake-up capability. As an example, the sensor controller is capable of 1-Hz ADC sampling at 1 µA system current.
- Low [SER \(Soft Error Rate\) FIT](#) (Failure-in-time) for long operation lifetime with no disruption for industrial markets with always-on SRAM parity against corruption due to potential radiation events.
- Dedicated software controlled radio controller (Arm® Cortex®-M0) providing flexible low-power RF transceiver capability to support multiple physical layers and RF standards.
- Excellent radio sensitivity (-121 dBm) and robustness (selectivity and blocking) performance for SimpleLink™ long-range mode.

<sup>1</sup> See [RF Core](#) for additional details on supported protocol standards, modulation formats, and data rates.

The CC1352R device is part of the SimpleLink™ MCU platform, which consists of Wi-Fi®, *Bluetooth* Low Energy, Thread, Zigbee, Sub-1 GHz MCUs, and host MCUs that all share a common, easy-to-use development environment with a single core software development kit (SDK) and rich tool set. A one-time integration of the SimpleLink™ platform enables you to add any combination of the portfolio's devices into your design, allowing 100 percent code reuse when your design requirements change. For more information, visit [SimpleLink™ MCU platform](#).

**Device Information**

<b>PART NUMBER<sup>(1)</sup></b>	<b>PACKAGE</b>	<b>BODY SIZE (NOM)</b>
CC1352R1F3RGZ	VQFN (48)	7.00 mm × 7.00 mm

(1) For the most current part, package, and ordering information for all available devices, see the Package Option Addendum in [Section 12](#), or see the [TI website](#).

## 4 Functional Block Diagram

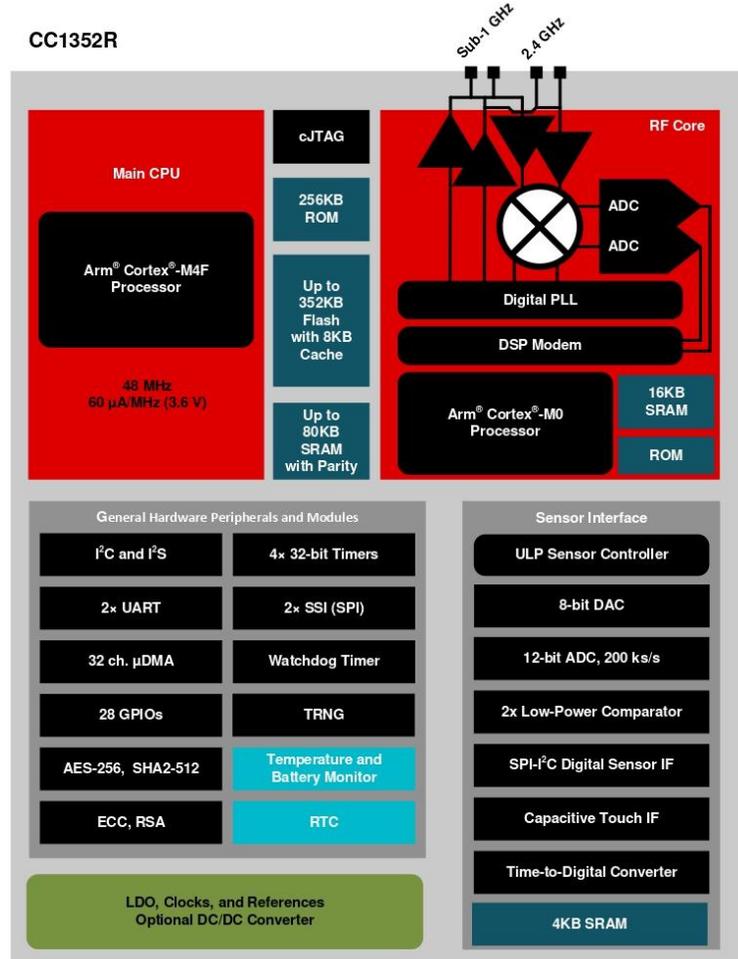


Figure 4-1. CC1352R Block Diagram

## 7 Terminal Configuration and Functions

### 7.1 Pin Diagram – RGZ Package (Top View)

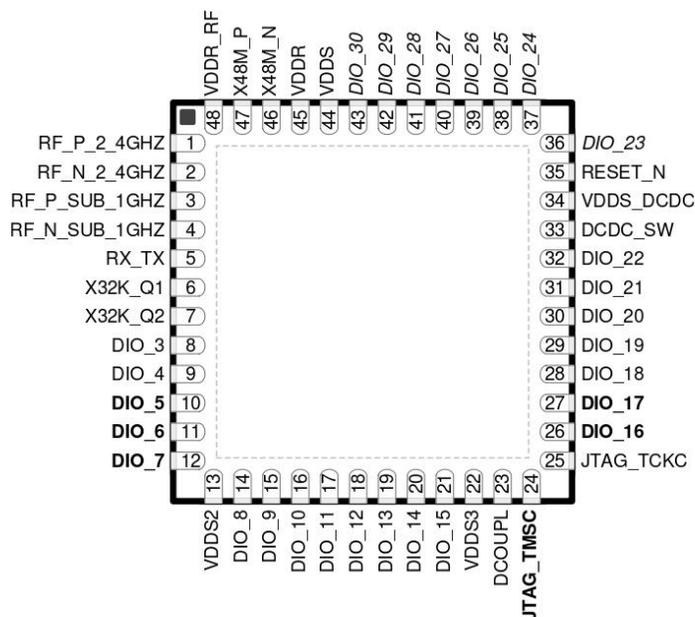


Figure 7-1. RGZ (7-mm × 7-mm) Pinout, 0.5-mm Pitch (Top View)

The following I/O pins marked in Figure 7-1 in **bold** have high-drive capabilities:

- Pin 10, **DIO\_5**
- Pin 11, **DIO\_6**
- Pin 12, **DIO\_7**
- Pin 24, **JTAG\_TMSC**
- Pin 26, **DIO\_16**
- Pin 27, **DIO\_17**

The following I/O pins marked in Figure 7-1 in *italics* have analog capabilities:

- Pin 36, *DIO\_23*
- Pin 37, *DIO\_24*
- Pin 38, *DIO\_25*
- Pin 39, *DIO\_26*
- Pin 40, *DIO\_27*
- Pin 41, *DIO\_28*
- Pin 42, *DIO\_29*
- Pin 43, *DIO\_30*

## 7.2 Signal Descriptions – RGZ Package

**Table 7-1. Signal Descriptions – RGZ Package**

PIN		I/O	TYPE	DESCRIPTION
NAME	NO.			
DCDC_SW	33	—	Power	Output from internal DC/DC converter <sup>(1)</sup>
DCOUP	23	—	Power	For decoupling of internal 1.27 V regulated digital-supply <sup>(2)</sup>
DIO_3	8	I/O	Digital	GPIO
DIO_4	9	I/O	Digital	GPIO
DIO_5	10	I/O	Digital	GPIO, high-drive capability
DIO_6	11	I/O	Digital	GPIO, high-drive capability
DIO_7	12	I/O	Digital	GPIO, high-drive capability
DIO_8	14	I/O	Digital	GPIO
DIO_9	15	I/O	Digital	GPIO
DIO_10	16	I/O	Digital	GPIO
DIO_11	17	I/O	Digital	GPIO
DIO_12	18	I/O	Digital	GPIO
DIO_13	19	I/O	Digital	GPIO
DIO_14	20	I/O	Digital	GPIO
DIO_15	21	I/O	Digital	GPIO
DIO_16	26	I/O	Digital	GPIO, JTAG_TDO, high-drive capability
DIO_17	27	I/O	Digital	GPIO, JTAG_TDI, high-drive capability
DIO_18	28	I/O	Digital	GPIO
DIO_19	29	I/O	Digital	GPIO
DIO_20	30	I/O	Digital	GPIO
DIO_21	31	I/O	Digital	GPIO
DIO_22	32	I/O	Digital	GPIO
DIO_23	36	I/O	Digital or Analog	GPIO, analog capability
DIO_24	37	I/O	Digital or Analog	GPIO, analog capability
DIO_25	38	I/O	Digital or Analog	GPIO, analog capability
DIO_26	39	I/O	Digital or Analog	GPIO, analog capability
DIO_27	40	I/O	Digital or Analog	GPIO, analog capability
DIO_28	41	I/O	Digital or Analog	GPIO, analog capability
DIO_29	42	I/O	Digital or Analog	GPIO, analog capability
DIO_30	43	I/O	Digital or Analog	GPIO, analog capability
EGP	—	—	GND	Ground – exposed ground pad <sup>(3)</sup>
JTAG_TMSC	24	I/O	Digital	JTAG TMSC, high-drive capability
JTAG_TCKC	25	I	Digital	JTAG TCKC
RESET_N	35	I	Digital	Reset, active low. No internal pullup resistor
RF_P_2_4GHZ	1	—	RF	Positive 2.4-GHz RF input signal to LNA during RX Positive 2.4-GHz RF output signal from PA during TX
RF_N_2_4GHZ	2	—	RF	Negative 2.4-GHz RF input signal to LNA during RX Negative 2.4-GHz RF output signal from PA during TX
RF_P_SUB_1GHZ	3	—	RF	Positive Sub-1 GHz RF input signal to LNA during RX Positive Sub-1 GHz RF output signal from PA during TX
RF_N_SUB_1GHZ	4	—	RF	Negative Sub-1 GHz RF input signal to LNA during RX Negative Sub-1 GHz RF output signal from PA during TX
RX_TX	5	—	RF	Optional bias pin for the RF LNA

**Table 7-1. Signal Descriptions – RGZ Package (continued)**

PIN		I/O	TYPE	DESCRIPTION
NAME	NO.			
VDDR	45	—	Power	Internal supply, must be powered from the internal DC/DC converter or the internal LDO <sup>(2)</sup> (4) (6)
VDDR_RF	48	—	Power	Internal supply, must be powered from the internal DC/DC converter or the internal LDO <sup>(2)</sup> (5) (6)
VDDS	44	—	Power	1.8-V to 3.8-V main chip supply <sup>(1)</sup>
VDDS2	13	—	Power	1.8-V to 3.8-V DIO supply <sup>(1)</sup>
VDDS3	22	—	Power	1.8-V to 3.8-V DIO supply <sup>(1)</sup>
VDDS_DCDC	34	—	Power	1.8-V to 3.8-V DC/DC converter supply
X48M_N	46	—	Analog	48-MHz crystal oscillator pin 1
X48M_P	47	—	Analog	48-MHz crystal oscillator pin 2
X32K_Q1	6	—	Analog	32-kHz crystal oscillator pin 1
X32K_Q2	7	—	Analog	32-kHz crystal oscillator pin 2

- (1) For more details, see technical reference manual listed in [Section 11.2](#).
- (2) Do not supply external circuitry from this pin.
- (3) EGP is the only ground connection for the device. Good electrical connection to device ground on printed circuit board (PCB) is imperative for proper device operation.
- (4) If internal DC/DC converter is not used, this pin is supplied internally from the main LDO.
- (5) If internal DC/DC converter is not used, this pin must be connected to VDDR for supply from the main LDO.
- (6) Output from internal DC/DC and LDO is trimmed to 1.68 V.

### 7.3 Connections for Unused Pins and Modules

**Table 7-2. Connections for Unused Pins**

FUNCTION	SIGNAL NAME	PIN NUMBER	ACCEPTABLE PRACTICE <sup>(1)</sup>	PREFERRED PRACTICE <sup>(1)</sup>
GPIO	DIO_n	8–12 14–21 26–32 36–43	NC or GND	NC
		6		
32.768-kHz crystal	X32K_Q1	6	NC or GND	NC
	X32K_Q2	7		
DC/DC converter <sup>(2)</sup>	DCDC_SW	33	NC	NC
	VDDS_DCDC	34	VDDS	VDDS

- (1) NC = No connect
- (2) When the DC/DC converter is not used, the inductor between DCDC\_SW and VDDR can be removed. VDDR and VDDR\_RF must still be connected and the 22 µF DCDC capacitor must be kept on the VDDR net.

## 8 Specifications

### 8.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)<sup>(1) (2)</sup>

		MIN	MAX	UNIT
V <sub>DD</sub> S <sup>(3)</sup>	Supply voltage	-0.3	4.1	V
	Voltage on any digital pin <sup>(4)</sup>	-0.3	V <sub>DD</sub> S + 0.3, max 4.1	V
	Voltage on crystal oscillator pins, X32K_Q1, X32K_Q2, X48M_N and X48M_P	-0.3	V <sub>DD</sub> R + 0.3, max 2.25	V
V <sub>in</sub>	Voltage on ADC input	Voltage scaling enabled	V <sub>DD</sub> S	
		Voltage scaling disabled, internal reference	1.49	V
		Voltage scaling disabled, V <sub>DD</sub> S as reference	V <sub>DD</sub> S / 2.9	
	Input level, Sub-1 GHz RF pins		10	dBm
	Input level, 2.4 GHz RF pins		5	dBm
T <sub>stg</sub>	Storage temperature	-40	150	°C

- (1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
- (2) All voltage values are with respect to ground, unless otherwise noted.
- (3) V<sub>DD</sub>S\_DCDC, V<sub>DD</sub>S2 and V<sub>DD</sub>S3 must be at the same potential as V<sub>DD</sub>S.
- (4) Including analog capable DIOs.

### 8.2 ESD Ratings

			VALUE	UNIT	
V <sub>ESD</sub>	Electrostatic discharge	Human body model (HBM), per ANSI/ESDA/JEDEC JS-001 <sup>(1)</sup>	All pins	±2000	V
		Charged device model (CDM), per ANSI/ESDA/JEDEC JS-002 <sup>(2)</sup>	All pins	±500	V

- (1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process
- (2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process

### 8.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	MAX	UNIT
Operating junction temperature <sup>(2)</sup>		-40	105	°C
Operating supply voltage (V <sub>DD</sub> S)		1.8	3.8	V
Operating supply voltage (V <sub>DD</sub> S), boost mode	V <sub>DD</sub> R = 1.95 V +14 dBm RF output power	2.1	3.8	V
Rising supply voltage slew rate		0	100	mV/μs
Falling supply voltage slew rate <sup>(1)</sup>		0	20	mV/μs

- (1) For small coin-cell batteries, with high worst-case end-of-life equivalent source resistance, a 22-μF V<sub>DD</sub>S input capacitor must be used to ensure compliance with this slew rate.
- (2) For thermal resistance characteristics refer to [Thermal Resistance Characteristics](#). For application considerations, refer to [Junction Temperature](#).

## 8.4 Power Supply and Modules

over operating free-air temperature range (unless otherwise noted)

PARAMETER		MIN	TYP	MAX	UNIT
VDDS Power-on-Reset (POR) threshold			1.1 - 1.55		V
VDDS Brown-out Detector (BOD) <sup>(1)</sup>	Rising threshold		1.77		V
VDDS Brown-out Detector (BOD), before initial boot <sup>(2)</sup>	Rising threshold		1.70		V
VDDS Brown-out Detector (BOD) <sup>(1)</sup>	Falling threshold		1.75		V

(1) For boost mode (VDDR = 1.95 V), TI drivers software initialization will trim VDDS BOD limits to maximum (approximately 2.0 V)

(2) Brown-out Detector is trimmed at initial boot, value is kept until device is reset by a POR reset or the RESET\_N pin

# ANEXO G: Datasheet Modulo HX711



HX711

## 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales

### DESCRIPTION

Based on Avia Semiconductor's patented technology, HX711 is a precision 24-bit analog-to-digital converter (ADC) designed for weigh scales and industrial control applications to interface directly with a bridge sensor.

The input multiplexer selects either Channel A or B differential input to the low-noise programmable gain amplifier (PGA). Channel A can be programmed with a gain of 128 or 64, corresponding to a full-scale differential input voltage of  $\pm 20\text{mV}$  or  $\pm 40\text{mV}$  respectively, when a 5V supply is connected to AVDD analog power supply pin. Channel B has a fixed gain of 32. On-chip power supply regulator eliminates the need for an external supply regulator to provide analog power for the ADC and the sensor. Clock input is flexible. It can be from an external clock source, a crystal, or the on-chip oscillator that does not require any external component. On-chip power-on-reset circuitry simplifies digital interface initialization.

There is no programming needed for the internal registers. All controls to the HX711 are through the pins.

### FEATURES

- Two selectable differential input channels
- On-chip active low noise PGA with selectable gain of 32, 64 and 128
- On-chip power supply regulator for load-cell and ADC analog power supply
- On-chip oscillator requiring no external component with optional external crystal
- On-chip power-on-reset
- Simple digital control and serial interface: pin-driven controls, no programming needed
- Selectable 10SPS or 80SPS output data rate
- Simultaneous 50 and 60Hz supply rejection
- Current consumption including on-chip analog power supply regulator:
  - normal operation  $< 1.5\text{mA}$ , power down  $< 1\mu\text{A}$
- Operation supply voltage range: 2.6 ~ 5.5V
- Operation temperature range:  $-40 \sim +85^\circ\text{C}$
- 16 pin SOP-16 package

### APPLICATIONS

- Weigh Scales
- Industrial Process Control

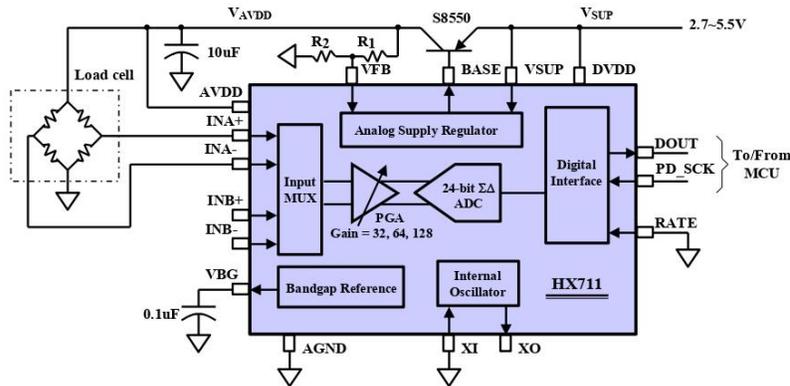
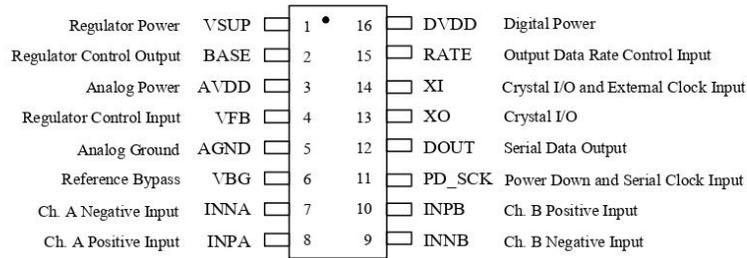


Fig. 1 Typical weigh scale application block diagram

**Pin Description**


SOP-16L Package

Pin #	Name	Function	Description
1	VSUP	Power	Regulator supply: 2.7 ~ 5.5V
2	BASE	Analog Output	Regulator control output (NC when not used)
3	AVDD	Power	Analog supply: 2.6 ~ 5.5V
4	VFB	Analog Input	Regulator control input (connect to AGND when not used)
5	AGND	Ground	Analog Ground
6	VBG	Analog Output	Reference bypass output
7	INA-	Analog Input	Channel A negative input
8	INA+	Analog Input	Channel A positive input
9	INB-	Analog Input	Channel B negative input
10	INB+	Analog Input	Channel B positive input
11	PD_SCK	Digital Input	Power down control (high active) and serial clock input
12	DOUT	Digital Output	Serial data output
13	XO	Digital I/O	Crystal I/O (NC when not used)
14	XI	Digital Input	Crystal I/O or external clock input, 0: use on-chip oscillator
15	RATE	Digital Input	Output data rate control, 0: 10Hz; 1: 80Hz
16	DVDD	Power	Digital supply: 2.6 ~ 5.5V

**Table 1 Pin Description**

**KEY ELECTRICAL CHARACTERISTICS**

Parameter	Notes	MIN	TYP	MAX	UNIT
Full scale differential input range	V(inp)-V(inn)	$\pm 0.5(AVDD/GAIN)$			V
Common mode input		AGND+1.2		AVDD-1.3	V
Output data rate	Internal Oscillator, RATE = 0	10			Hz
	Internal Oscillator, RATE = DVDD	80			
	Crystal or external clock, RATE = 0	$f_{clk}/1,105,920$			
	Crystal or external clock, RATE = DVDD	$f_{clk}/138,240$			
Output data coding	2's complement	800000		7FFFFFFF	HEX
Output settling time <sup>(1)</sup>	RATE = 0	400			ms
	RATE = DVDD	50			
Input offset drift	Gain = 128	0.2			mV
	Gain = 64	0.4			
Input noise	Gain = 128, RATE = 0	50			nV(rms)
	Gain = 128, RATE = DVDD	90			
Temperature drift	Input offset (Gain = 128)	±6			nV/°C
	Gain (Gain = 128)	±5			ppm/°C
Input common mode rejection	Gain = 128, RATE = 0	100			dB
Power supply rejection	Gain = 128, RATE = 0	100			dB
Reference bypass (V <sub>BG</sub> )		1.25			V
Crystal or external clock frequency		1	11.0592	20	MHz
Power supply voltage	DVDD	2.6		5.5	V
	AVDD, VSUP	2.6		5.5	
Analog supply current (including regulator)	Normal	1400			μA
	Power down	0.3			
Digital supply current	Normal	100			μA
	Power down	0.2			

(1) Settling time refers to the time from power up, reset, input channel change and gain change to valid stable output data.

**Table 2 Key Electrical Characteristics**

Reference PCB Board (Single Layer)

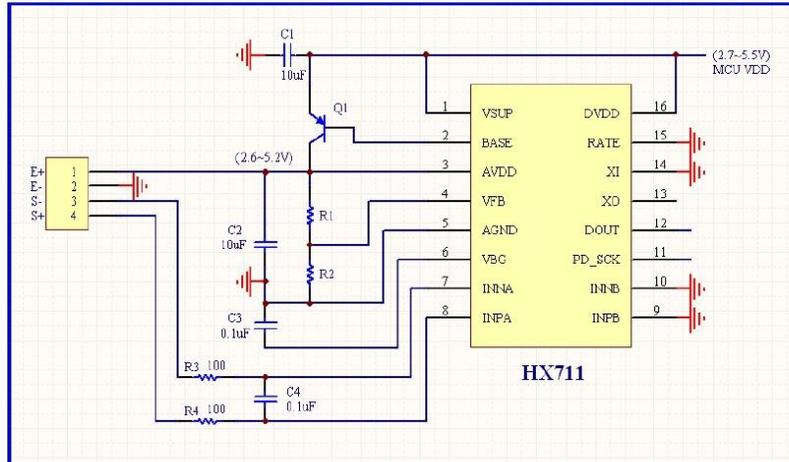


Fig.4 Reference PCB board schematic

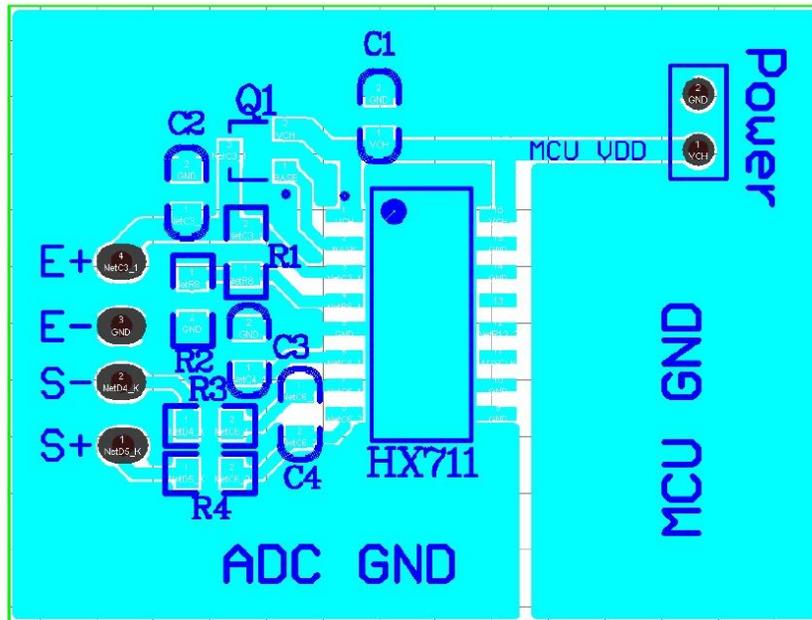


Fig.5 Reference PCB board layout

## NEO-7

### u-blox 7 GNSS modules

#### Data Sheet

#### Highlights:

- GNSS engine for GPS/QZSS, GLONASS
- Product variants to meet performance and cost requirements
- Combines low power consumption and high sensitivity
- Backward compatible with NEO-6 and NEO-5 families



[www.u-blox.com](http://www.u-blox.com)

UBX-13003830 - R07

**Document Information**

<b>Title</b>	<b>NEO-7</b>		
<b>Subtitle</b>	u-blox 7 GNSS modules		
<b>Document type</b>	Data Sheet		
<b>Document number</b>	UBX-13003830		
<b>Revision, date</b>	R07	11-Nov-2014	
<b>Document status</b>	Production Information		

**Document status explanation**

Objective Specification	Document contains target values. Revised and supplementary data will be published later.		
Advance Information	Document contains data based on early testing. Revised and supplementary data will be published later.		
Early Production Information	Document contains data from product verification. Revised and supplementary data may be published later.		
Production Information	Document contains the final product specification.		

**This document applies to the following products:**

<b>Name</b>	<b>Type number</b>	<b>ROM/FLASH version</b>	<b>PCN reference</b>
NEO-7M-0	NEO-7M-0-000	ROM1.00	N/A
NEO-7N-0	NEO-7N-0-002	FW1.00	N/A

u-blox reserves all rights to this document and the information contained herein. Products, names, logos and designs described herein may in whole or in part be subject to intellectual property rights. Reproduction, use, modification or disclosure to third parties of this document or any part thereof without the express permission of u-blox is strictly prohibited.

The information contained herein is provided "as is" and u-blox assumes no liability for the use of the information. No warranty, either express or implied, is given, including but not limited, with respect to the accuracy, correctness, reliability and fitness for a particular purpose of the information. This document may be revised by u-blox at any time. For most recent documents, visit [www.u-blox.com](http://www.u-blox.com).  
Copyright © 2014, u-blox AG.

u-blox® is a registered trademark of u-blox Holding AG in the EU and other countries. ARM® is the registered trademark of ARM Limited in the EU and other countries.

## 1.3 GNSS performance

### 1.3.1 GPS performance

Parameter	Specification		
Receiver type	56 Channels GPS L1C/A SBAS L1C/A QZSS L1C/A Galileo E1B/C <sup>1</sup>		
Time-To-First-Fix <sup>2</sup>		<b>NEO-7N</b>	<b>NEO-7M</b>
	Cold Start	29 s	30 s
	Warm Start	28 s	28 s
	Hot Start	1 s	1 s
	Aided Starts <sup>3</sup>	5 s	5 s
Sensitivity <sup>4</sup>		<b>NEO-7N</b>	<b>NEO-7M</b>
	Tracking & Navigation	-162 dBm	-161 dBm
	Reacquisition	-160 dBm	-160 dBm
	Cold Start	-148 dBm	-147 dBm
	Warm Start	-148 dBm	-148 dBm
	Hot Start	-156 dBm	-155 dBm
Horizontal position accuracy <sup>5</sup>	Autonomous	2.5 m	
	SBAS	2.0 m	
Accuracy of time pulse signal	RMS	30 ns	
	99%	60 ns	
Frequency of time pulse signal		0.25 Hz ... 10 MHz (configurable)	
Max navigation update rate		10 Hz	
Velocity accuracy <sup>6</sup>		0.1 m/s	
Heading accuracy <sup>6</sup>		0.5 degrees	
Operational limits <sup>7</sup>	Dynamics	≤ 4 g	
	Altitude	50,000 m	
	Velocity	500 m/s	

**Table 1: GPS performance**

<sup>1</sup> Ready to support Galileo E1B/C when available (NEO-7N)

<sup>2</sup> All satellites at -130 dBm

<sup>3</sup> Dependent on aiding data connection speed and latency

<sup>4</sup> Demonstrated with a good external LNA

<sup>5</sup> CEP, 50%, 24 hours static, -130 dBm, > 6 SVs

<sup>6</sup> 50% @ 30 m/s

<sup>7</sup> Assuming Airborne < 4 g platform

## 1.4 Block diagram

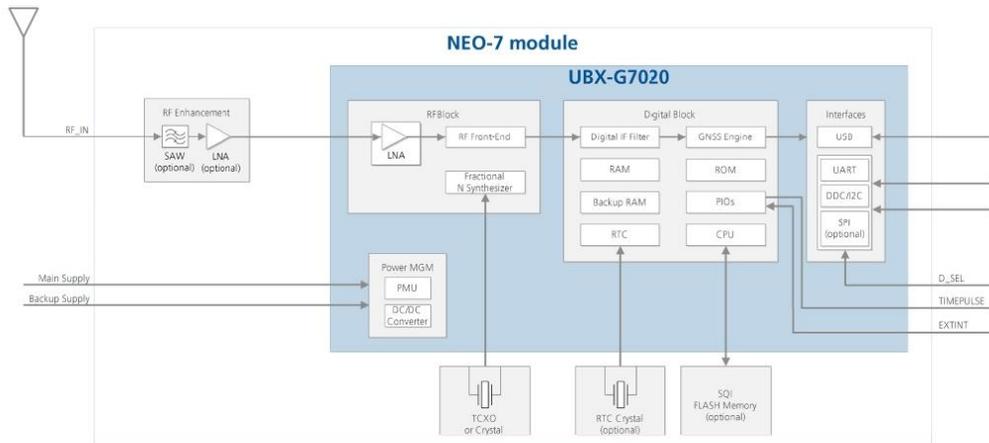


Figure 1: NEO-7 block diagram

## 1.5 GNSS

u-blox 7 positioning modules are GNSS receivers and can singly receive and track GPS, GLONASS, or Galileo signals. QZSS signals may be received concurrently with GPS signals.

### 1.5.1 GPS

u-blox 7 receivers are designed to receive and track the L1C/A signals provided at 1575.42 MHz by the Global Positioning System (GPS).

### 1.5.2 GLONASS

The Russian GLONASS satellite system is an alternative system to the US-based Global Positioning System (GPS). The u-blox 7 module is capable of receiving and processing GLONASS signals using the same hardware and provides the lowest power GLONASS functionality in the industry at low cost and with minimal integration effort. In order to take advantage of GPS and GLONASS, dedicated hardware preparation must be taken during the design-in phase, see the *MAX-7 / NEO-7 Hardware Integration Manual* [1] for u-blox design recommendations.

The ability to receive and track GLONASS L1OF satellite signals with the same hardware results in an optimized hardware BOM and allows design of GLONASS ready receivers where required by regulations.

Note: GLONASS and GPS signals cannot be received and tracked simultaneously by u-blox 7 modules.

### 1.5.3 Galileo

When Galileo-L1 signals become available, u-blox 7 flash-based receivers will be capable of receiving and processing them via a firmware upgrade. The ability to receive and track Galileo satellite signals will result in higher coverage, improved reliability and better accuracy.

### 1.5.4 QZSS

The Quasi-Zenith Satellite System (QZSS) is a navigation satellite overlay system for the Pacific region covering Japan and Australia which transmits additional GPS L1C/A signals. u-blox 7 positioning modules are able to

## 1.11 Interfaces

A number of interfaces are provided either for data communication or memory access. The embedded firmware uses these interfaces according to their respective protocol specifications.

### 1.11.1 UART

NEO-7 modules include one UART interface, which can be used for communication to a host. It supports configurable baud rates. For supported baud rates see the *u-blox 7 Receiver Description Including Protocol Specification* [2].

### 1.11.2 USB

A USB version 2.0 FS compatible interface can be used for communication as an alternative to the UART. The pull-up resistor on pin USB\_DP is integrated to signal a full-speed device to the host. The VDD\_USB pin supplies the USB interface.

u-blox provides a Microsoft® certified USB driver for Windows XP, Windows Vista and Windows 7 operating systems.

### 1.11.3 SPI

The SPI interface is designed to allow communication to a host CPU. The interface can be operated in slave mode only. The maximum transfer rate using SPI is 1 Mb/s and the maximum SPI clock frequency is 5.5 MHz. Note that SPI is not available in the default configuration, because its pins are shared with the UART and DDC interfaces. The SPI interface can be enabled by connecting D\_SEL (Pin 2) to ground (see section 3.1).

### 1.11.4 Display Data Channel (DDC)

An I<sup>2</sup>C compliant DDC interface is available for communication with an external host CPU or u-blox cellular modules. The interface can be operated in slave mode only. The DDC protocol and electrical interface are fully compatible with Fast-Mode of the I<sup>2</sup>C industry standard. Since the maximum SCL clock frequency is 400 kHz, the maximum transfer rate is 400 kb/s.

## 1.12 Clock generation

### 1.12.1 Oscillators

NEO-7 GNSS modules are available in Crystal and TCXO versions. The TCXO allows accelerated weak signal acquisition, enabling faster start and reacquisition times.

### 1.12.2 Real-Time Clock (RTC)

The RTC is driven by a 32 kHz oscillator, which makes use of an RTC crystal. If the main supply voltage fails, and a battery is connected to V\_BCKP, parts of the receiver switch off, but the RTC still runs providing a timing reference for the receiver. This operating mode is called Hardware Backup Mode, which enables all relevant data to be saved in the backup RAM to later allow a hot or warm start.

## 1.13 Power management

u-blox 7 technology offers a power-optimized architecture with built-in autonomous power saving functions to minimize power consumption at any given time. An integrated high efficiency DC/DC converter allows low power consumption even for higher main supply voltages. The receiver can be operated in two operating modes: Continuous Mode or Power Save Mode.

 For more information about power management strategies, see the *u-blox 7 Receiver Description Including Protocol Specification* [2].

### 1.13.1 DC/DC converter

NEO-7 Modules integrate a DC/DC converter, allowing reduced power consumption especially when using a main supply voltage above 2.5 V.

 For more information see the *MAX-7 / NEO-7 Hardware Integration Manual* [1]

### 1.13.2 Operating modes

u-blox 7 modules have two operating modes:

- Continuous Mode for best GNSS performance
- Power Save Mode to optimize power consumption

#### 1.13.2.1 Continuous Mode

Continuous Mode uses the acquisition engine at full performance resulting in the shortest possible TTFF and the highest sensitivity. It searches for all possible satellites until the Almanac is completely downloaded. The receiver then switches to the tracking engine to lower power consumption.

Thus, a lower tracking current consumption level will be achieved when:

- A valid GNSS position is obtained
- The entire Almanac has been downloaded
- The Ephemeris for each satellite in view is valid

#### 1.13.2.2 Power Save Mode

For power sensitive applications, u-blox 7 receivers provide a Power Save Mode for reduced power consumption. Power Save Mode uses two dedicated operations called ON/OFF and Cyclic tracking, that reduce average current consumption in different ways to match the needs of the specific application. These operations can be set by using a specific ubx message.

 Power Save Mode is not available in GLONASS mode.

## 1.14 Antenna

NEO-7 modules are designed for use with passive<sup>13</sup> and active<sup>14</sup> antennas.

Parameter	Specification	
Antenna Type	Passive and active antenna	
Active Antenna Recommendations	Minimum gain	15 dB (to compensate signal loss in RF cable)
	Maximum gain	50 <sup>15</sup> dB / 30 <sup>16</sup> dB
	Maximum noise figure	1.5 dB

**Table 4: Antenna Specifications for all NEO-7 modules**

<sup>13</sup> For integration NEO-7 modules with Cellular products, see the *MAX-7 / NEO-7 Hardware Integration Manual* [1].

<sup>14</sup> For information on using active antennas with NEO-7 modules, see the *MAX-7 / NEO-7 Hardware Integration Manual* [1].

<sup>15</sup> NEO-7M

<sup>16</sup> NEO-7N

## 2 Pin Definition

### 2.1 Pin assignment



Figure 2: Pin Assignment

No	Module	Name	I/O	Description
1	All	Reserved	I	Reserved
2	All	D_SEL	I	Interface select
3	All	TIMEPULSE	O	Time pulse (1PPS)
4	All	EXTINT	I	External Interrupt Pin
5	All	USB_DM	I/O	USB Data
6	All	USB_DP	I/O	USB Data
7	All	VDD_USB	I	USB Supply
8	All	RESET_N	I	RESET_N
9	All	VCC_RF	O	Output Voltage RF section
10	All	GND	I	Ground
11	All	RF_IN	I	GNSS signal input
12	All	GND	I	Ground
13	All	GND	I	Ground
14	NEO-7N	ANT_ON	O	Antenna control
	NEO-7M	Reserved	-	Reserved
15	All	Reserved	-	Reserved
16	All	Reserved	-	Reserved
17	All	Reserved	-	Reserved
18	All	SDA SPI CS_N	I/O	DDC Data if D_SEL = 1 (or open) SPI Chip Select if D_SEL = 0
19	All	SCL SPI CLK	I/O	DDC Clock if D_SEL = 1 (or open) SPI Clock if D_SEL = 0
20	All	TxD SPI MISO	O	Serial Port if D_SEL = 1 (or open) SPI MISO if D_SEL = 0
21	All	RxD SPI MOSI	I	Serial Port if D_SEL = 1 (or open) SPI MOSI if D_SEL = 0
22	All	V_BCKP	I	Backup voltage supply
23	All	VCC	I	Supply voltage
24	All	GND	I	Ground

Table 5: Pinout



Pins designated Reserved should not be used. For more information about Pinouts see the *MAX-7 / NEO-7 Hardware Integration Manual* [1].

## 4 Electrical specification

- 

The limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to these limits for extended periods may affect device reliability.
- 

Where application information is given, it is advisory only and does not form part of the specification. For more information see the *MAX-7 / NEO-7 Hardware Integration Manual* [1].

### 4.1 Absolute maximum rating

Parameter	Symbol	Module	Condition	Min	Max	Units
Power supply voltage	VCC	All		-0.5	3.6	V
Backup battery voltage	V_BCKP	All		-0.5	3.6	V
USB supply voltage	VDD_USB	All		-0.5	3.6	V
Input pin voltage	Vin	All		-0.5	3.6	V
	Vin_usb	All		-0.5	VDD_USB	V
DC current through any digital I/O pin (except supplies)	Ipin				10	mA
VCC_RF output current	ICC_RF	All			100	mA
Input power at RF_IN	Prfin	All	source impedance = 50 Ω, continuous wave		13	dBm
Storage temperature	Tstg	All		-40	85	°C

**Table 7: Absolute maximum ratings**

- 

**Stressing the device beyond the “Absolute Maximum Ratings” may cause permanent damage. These are stress ratings only. The product is not protected against overvoltage or reversed voltages. If necessary, voltage spikes exceeding the power supply voltage specification, given in table above, must be limited to values within the specified boundaries by using appropriate protection diodes.**

# ANEXO I: Datasheet Sensor MQ-135



炜盛科技 Zhengzhou Winsen Electronics Technology Co., Ltd

[www.winsen-sensor.com](http://www.winsen-sensor.com)

## MQ135 Semiconductor Sensor for Air Quality

### Profile

Sensitive material of MQ135 gas sensor is SnO<sub>2</sub>, which with lower conductivity in clean air. When target pollution gas exists, the sensor's conductivity gets higher along with the gas concentration rising. Users can convert the change of conductivity to correspond output signal of gas concentration through a simple circuit.

MQ135 gas sensor has high sensitivity to ammonia gas, sulfide, benzene series steam, also can monitor smoke and other toxic gases well. It can detect kinds of toxic gases and is a kind of low-cost sensor for kinds of applications.

### Features

It has good sensitivity to toxic gas in wide range, and has advantages such as long lifespan, low cost and simple drive circuit &etc.

### Main Applications

It is widely used in domestic gas alarm, industrial gas alarm and portable gas detector.

### Technical Parameters Stable.1

Model		MQ135	
Sensor Type		Semiconductor	
Standard Encapsulation		Bakelite, Metal cap	
Target Gas		ammonia gas, sulfide, benzene series steam	
Detection range		10~1000ppm( ammonia gas, toluene, hydrogen, smoke)	
Standard Circuit Conditions	Loop Voltage	V <sub>c</sub>	≤24V DC
	Heater Voltage	V <sub>H</sub>	5.0V±0.1V AC or DC
	Load Resistance	R <sub>L</sub>	Adjustable
Sensor character under standard test conditions	Heater Resistance	R <sub>H</sub>	29Ω±3Ω (room tem.)
	Heater consumption	P <sub>H</sub>	≤950mW
	Sensitivity	S	R <sub>s</sub> (in air)/R <sub>s</sub> (in 400ppm H <sub>2</sub> )≥5
	Output Voltage	V <sub>s</sub>	2.0V~4.0V (in 400ppm H <sub>2</sub> )
Standard test conditions	Tem. Humidity	20°C±2°C; 55%±5%RH	
	Standard test circuit	V <sub>c</sub> : 5.0V±0.1V; V <sub>H</sub> : 5.0V±0.1V	
	Preheat time	Over 48 hours	

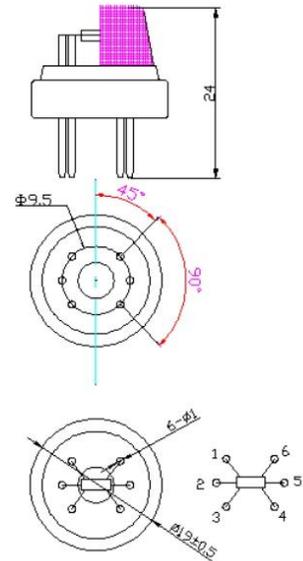
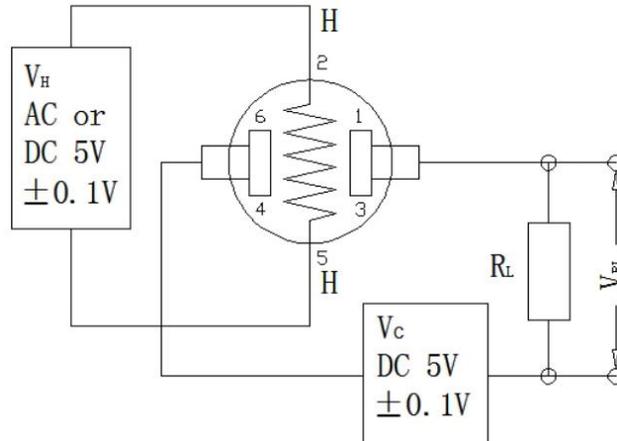


Fig1.Sensor Structure  
Unit: mm

NOTE: Output voltage (V<sub>s</sub>) is V<sub>RL</sub> in test environment.

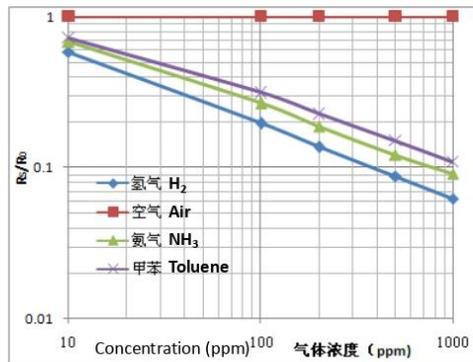
**Basic Circuit**



**Fig2. MQ135 Test Circuit**

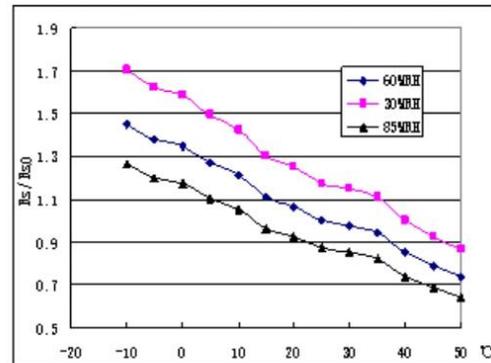
**Instructions:** The above fig is the basic test circuit of MQ135. The sensor requires two voltage inputs: heater voltage ( $V_H$ ) and circuit voltage ( $V_C$ ).  $V_H$  is used to supply standard working temperature to the sensor and it can adopt DC or AC power, while  $V_{RL}$  is the voltage of load resistance  $R_L$  which is in series with sensor.  $V_C$  supplies the detect voltage to load resistance  $R_L$  and it should adopt DC power.

**Description of Sensor Characters**



**Fig3. Typical Sensitivity Curve**

The ordinate is resistance ratio of the sensor ( $R_s/R_0$ ), the abscissa is concentration of gases.  $R_s$  means resistance in target gas with different concentration,  $R_0$  means resistance of sensor in clean air. All tests are finished under standard test conditions.



**Fig4. Typical temperature/humidity characteristics**

The ordinate is resistance ratio of the sensor ( $R_s/R_{s0}$ ).  $R_s$  means resistance of sensor in 400ppm  $H_2$  gas under different tem. and humidity.  $R_{s0}$  means resistance of the sensor in 400ppm  $H_2$  gas under 20°C/55%RH.

## ANEXO J: Datasheet Sensor HC-SR04



Datasheet **HC-SR04**

### HC-SR04



#### Descripción:

El sensor **HC-SR04** esta compuesto por un emisor y un receptor de ultrasonidos. Estos nos ayuda a medir la distancia a la que se encuentra un objeto justo frente a el, enviando un pulso de ultrasonidos y midiendo el tiempo que transcurre hasta que vuelve dicho pulso.

#### Descripción pines del sensor:



**-Vcc:** Pin de alimentación. (5V)

**-Trigger:** Pin de disparo. Este pin es una entrada, por lo que en el sistema de control, por ejemplo Arduino, se tiene que conectar a una salida.

**-Echo:** Este pin es una salida del sensor, por lo que ha de ser conectado a una entrada del sistema de control.

**Gnd:** Pin negativo de alimentación.

### Características eléctricas:

- Voltaje de trabajo: 5V.
- Corriente de trabajo: 15mA.
- Frecuencia de trabajo: 40KHz
- Rango de funcionamiento: 2 a 500 cm
- Ángulo de detección: 15 a 20 grados.

### Funcionamiento y Diagrama de temporización:

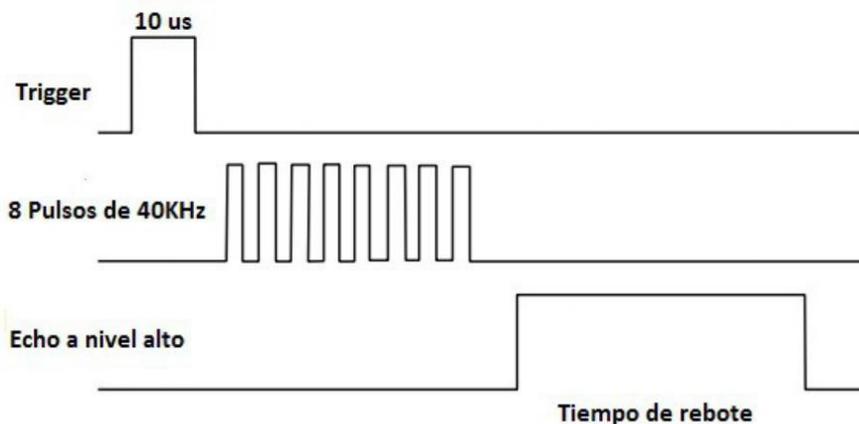
En el diagrama de temporización se aprecia como solo es necesario aplicar un pulso de 10uS en el pin trigger para comenzar con la medición.

A continuación el sensor envía una serie de 8 pulsos de 40KHz y pone el pin de Echo a nivel alto.

El pin Echo permanecerá a nivel alto hasta que se reciba el eco de los pulsos de 40KHz.

Para saber a la distancia a la que se encuentra el objeto, solo hay que medir el tiempo al que está el pin Echo a nivel alto y aplicar la siguiente formula.

$$\text{Centimetros} = \mu\text{S} * 0,01715$$



## ANEXO K: Datasheet Sensor celda de carga (50kg)

Especificaciones: <http://robot.com.ve/product/celda-de-carga-50kg-sensor-de-fuerza-peso-mas-modulo-hx711/>

Capacity	kg	40-50	
Comprehensive Error	mv/v	0.05	
Output Sensitivity	mv/v	1.0±0.1	
Nonlinearity	%FS	0.03	
Repeatability	%FS	0.03	
Hysteresis	%FS	0.03	
Creep	(3min)%FS	0.03	
Zero Drift	(1min)%FS	0.03	
Temp. Effect on Zero	%FS/10°C	1	
Temp. Effect on Output	%FS/10°C	0.05	
Zero Output	mV/V	±0.1	
Input Resistance	Ω	1000±20	
Output Resistance	Ω	1000±20	
Insulation Resistance	MΩ	≥5000	
Excitation Voltage	V	≤10	
Operation Temp. Range	°C	0--+50	
Overload Capacity	%FS	150	

## ANEXO L: Código Nodo Recolector- Arduino

```
#include <SoftwareSerial.h>

SoftwareSerial gps(4, 3);

char dato = '';

//////////Variables//////////

//Sensor Ultrasónico Norte
int TN = 24;    // trigger en pin 24
int EN = 25;    // echo en pin 25
int DUN;
int DIN;

//Sensor Ultrasónico Oeste
int TO = 26;    // trigger en pin 26
int EO = 27;    // echo en pin 27
int DUO;
int DIO;

//Sensor Ultrasónico Este
int TE = 28;    // trigger en pin 28
int EE = 29;    // echo en pin 29
int DUE;
int DIE;

//Sensor Ultrasónico Sur
int TS = 30;    // trigger en pin 30
int ES = 31;    // echo en pin 31
int DUS;
int DIS;

//CO2
int Gas;

//GPS
String Lat = " ";
String Lon = " ";

//Alarma
bool Alarm;

void setup()
{
  //Sensor Ultrasónico Norte
  pinMode(TN, OUTPUT);    // Trigger como salida
```

```

pinMode(EN, INPUT);      // Echo como entrada
//Sensor Ultrasonico Oeste
pinMode(TO, OUTPUT);    // Trigger como salida
pinMode(EO, INPUT);    // Echo como entrada
//Sensor Ultrasonico Este
pinMode(TE, OUTPUT);    // Trigger como salida
pinMode(EE, INPUT);    // Echo como entrada
//Sensor Ultrasonico Sur
pinMode(TS, OUTPUT);    // Trigger como salida
pinMode(ES, INPUT);    // Echo como entrada

```

```

Serial.begin(9600); // Inicialización de comunicacion serial a 9600 bps
gps.begin(9600); //inicializando comunicacion serial al gps
gps()
}

```

```

void loop()
{
  Ultrasonico();
  CO2();
  Alarma();
  Enviar();
  delay(500);
}

```

```

void gps()
{
  if (gps.available())
  {
    char data;
    data = gps.read();
    Serial.print(data);
    if (newData)
    {
      gps.f_get_position(&flat, &flon, &age);
      Lat=(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);
      Serial.print(" LON=");
      Lon=(flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);
    }
  }
}

```

```

void Ultrasonico() {
  //Sensor Ultrasonico Norte
  digitalWrite(TN, HIGH); // generacion del pulso a enviar
  delay(5); // al pin conectado al trigger
  digitalWrite(TN, LOW); // del sensor
}

```

```

DUN = pulseIn(EN, HIGH);    // con funcion pulseIn se espera un pulso alto en Echo
DIN = DUN / 58.2;          // distancia medida en centimetros
//Sensor Ultrasonico Oeste
digitalWrite(TO, HIGH);    // generacion del pulso a enviar
delay(5);                  // al pin conectado al trigger
digitalWrite(TO, LOW);     // del sensor
DUO = pulseIn(EO, HIGH);   // con funcion pulseIn se espera un pulso alto en Echo
DIO = DUO / 58.2;          // distancia medida en centimetros
//Sensor Ultrasonico Este
digitalWrite(TE, HIGH);    // generacion del pulso a enviar
delay(5);                  // al pin conectado al trigger
digitalWrite(TE, LOW);     // del sensor
DUE = pulseIn(EE, HIGH);   // con funcion pulseIn se espera un pulso alto en Echo
DIE = DUE / 58.2;          // distancia medida en centimetros
//Sensor Ultrasonico Sur
digitalWrite(TS, HIGH);    // generacion del pulso a enviar
delay(5);                  // al pin conectado al trigger
digitalWrite(TS, LOW);     // del sensor
DUS = pulseIn(ES, HIGH);   // con funcion pulseIn se espera un pulso alto en Echo
DIS = DUS / 58.2;          // distancia medida en centimetros
}

void CO2() {
  Gas = analogRead(A0);
}

void Alarma() {
  if (DIN < 11 && DIE < 11 && DIO < 11 && DIS < 11) {
    Alarm = 1;
    Serial.println("LLENO");
  } else {
    Alarm = 0;
    Serial.println("VACIO");
  }
}

void Enviar() {
  if (Alarm == 1) {
    Serial.print(Lat);
    Serial.print(";");
    Serial.print(Lon);
    Serial.print(";");
    Serial.println(Gas);
  }
}

```

## ANEXO M: Código Nodo Recolector- CC1352R1

```
#include "contiki.h"
#include "net/routing/routing.h"
#include "net/netstack.h"
#include "net/ipv6/simple-udp.h"
#include <stdint.h>
#include <inttypes.h>

#include "sys/log.h"
#define LOG_MODULE "App"
#define LOG_LEVEL LOG_LEVEL_INFO

//UDP CLIENTE 1
#define UDP_CLIENT_PORT 8764 //PUERTO LOCAL
#define UDP_SERVER_PORT 8765 //PUERTO REMOTO

//Serial arduino
#include "dev/serial-line.h"
#include "dev/leds.h"
#include <stdio.h> /* For printf() */
#include <string.h>

static struct simple_udp_connection udp_conn;
static uint32_t rx_count = 0;

/*-----
*/
PROCESS(udp_client_process, "UDP client");
AUTOSTART_PROCESSES(&udp_client_process);
/*-----
*/
//Callback para recibir (NO usado)
static void
udp_rx_callback(struct simple_udp_connection *c,
                const uip_ipaddr_t *sender_addr,
                uint16_t sender_port,
                const uip_ipaddr_t *receiver_addr,
                uint16_t receiver_port,
                const uint8_t *data,
                uint16_t datalen)
{
    LOG_INFO("Received response '%.*s' from ", datalen, (char *) data);
    LOG_INFO_6ADDR(sender_addr);
#ifdef LLSEC802154_CONF_ENABLED
    LOG_INFO_(" LLSEC LV:%d", uipbuf_get_attr(UIPBUF_ATTR_LLSEC_LEVEL));
#endif
#endif
```



```

    }
    //Sensor2-<datos>
    if (strstr(dataIn,"sensor2") != NULL){
        strncpy(sensor2, dataIn + 8, strlen(dataIn) -
8);
        sensor2[strlen(dataIn) - 8] = '\0'; //Null
        printf("sensor2: %s\n", sensor2);
    }
    //Sensor3-<datos>
    if (strstr(dataIn,"sensor3") != NULL){
        strncpy(sensor3, dataIn + 8, strlen(dataIn) -
8);
        sensor3[strlen(dataIn) - 8] = '\0'; //Null
        printf("sensor3: %s\n", sensor3);
    }
    //Sensor4-<datos>
    if (strstr(dataIn,"sensor4") != NULL){
        strncpy(sensor4, dataIn + 8, strlen(dataIn) -
8);
        sensor4[strlen(dataIn) - 8] = '\0'; //Null
        printf("sensor4: %s\n", sensor4);
    }
    //Sensor5-<datos>
    if (strstr(dataIn,"sensor5") != NULL){
        strncpy(sensor5, dataIn + 8, strlen(dataIn) -
8);
        sensor5[strlen(dataIn) - 8] = '\0'; //Null
        printf("sensor5: %s\n", sensor5);
    }
    //Sensor6-<datos>
    if (strstr(dataIn,"sensor6") != NULL){
        strncpy(sensor6, dataIn + 8, strlen(dataIn) -
8);
        sensor6[strlen(dataIn) - 8] = '\0'; //Null
        printf("sensor6: %s\n", sensor6);
    }
    //Comando <enviar>
    if (strstr(dataIn,"enviar") != NULL){
        if(NETSTACK_ROUTING.node_is_reachable() &&
NETSTACK_ROUTING.get_root_ipaddr(&dest_ipaddr)) {
            /* Envia hacia el DAG root */
            LOG_INFO("Enviando UDP hacia ");
            LOG_INFO_6ADDR(&dest_ipaddr);
            LOG_INFO_("\n");

            char json[240]; //Char array para armar el
json

```

```

        sprintf(json,
"{\"contenedores\": [{\"name\": \"1\", \"volumen\": \"%s\", \"co2\": \"%s\", \"peso\": \"%s\", \"ubicacion\": {\"lat\": \"%s\", \"long\": \"%s\"}, \"state\": \"%s\"}]}",
        sensor1, sensor2,
sensor3, sensor4, sensor5, sensor6); //Reemplaza los valores recibidos en el
json siguiendo el formato

        printf("json: %s\n", json); //Imprime el
json armado para comprobar

        snprintf(str, sizeof(str), json); //Copia
el json y su length en

        simple_udp_sendto(&udp_conn, str,
strlen(str), &dest_ipaddr); //Envia el json por UDP

    } else {
        LOG_INFO("Todavia no hay conexion\n");
    }
}
}
}

PROCESS_END();
}
/*-----
*/

```

## ANEXO N: Código Nodo coordinador- CC1352R1

```
#include "contiki.h"

//UDP-RPL
#include "net/routing/routing.h"
#include "net/netstack.h"
#include "net/ipv6/simple-udp.h"
#include <stdint.h>
#include <inttypes.h>

//PUERTOS UDP - CLIENTE 1
#define UDP_LOCAL_PORT1 8765 //puerto local
#define UDP_REMOTE_PORT1 8764 //puerto remoto
//PUERTOS UDP - CLIENTE 2
#define UDP_LOCAL_PORT2 9765 //puerto local
#define UDP_REMOTE_PORT2 9764 //puerto remoto
//PUERTOS UDP - MV
#define UDP_MV_LOCAL_PORT      3333 //puerto remoto
#define UDP_MV_REMOTE_PORT     4444 //puerto remoto

//Cx serial con MV
#include "dev/serial-line.h"
#include "dev/leds.h"
#include <stdio.h> /* For printf() */
#include <string.h>

/* Logs */
#include "sys/log.h"
#define LOG_MODULE "RPL BR"
#define LOG_LEVEL LOG_LEVEL_INFO

static struct simple_udp_connection udp_conn_client1;
static struct simple_udp_connection udp_conn_client2;
static struct simple_udp_connection udp_connMV;

/* Declare and auto-start this file's process */
PROCESS(contiki_ng_br, "Contiki-NG Border Router");
AUTOSTART_PROCESSES(&contiki_ng_br);

/*-----
*/
//Callback al recibir mensajes UDP
static void
udp_rx_callback(struct simple_udp_connection *c,
                const uip_ipaddr_t *sender_addr,
                uint16_t sender_port,
                const uip_ipaddr_t *receiver_addr,
```

```

    uint16_t receiver_port,
    const uint8_t *data,
    uint16_t datalen)
{
    LOG_INFO("Mensaje recibido: '%.*s' desde ", datalen, (char *) data);
    LOG_INFO_6ADDR(sender_addr);
    LOG_INFO_("\n");

    /* Reenvia hacia el servidor UDP en la MV */
    LOG_INFO("Reenviando hacia Python-Bridge en fe00::1.\n");
    uip_ipaddr_t dest_ipaddr;
    uip_ip6addr(&dest_ipaddr, 0xfe00, 0, 0, 0, 0, 0, 0, 0x1);
    simple_udp_sendto(&udp_connMV, data, datalen, &dest_ipaddr);

    printf("LEDS!\n");
    leds_single_toggle(LED1);
    leds_single_toggle(LED2);
}
/*-----*/
*/

PROCESS_THREAD(contiki_ng_br, ev, data)
{
    PROCESS_BEGIN();

    leds_set(LED1); //Enciende leds al iniciar

    /* Inicializa el nodo como DAG root */
    NETSTACK_ROUTING.root_start();

    //WEBSERVER para visualizar vecinos RPL
#ifdef BORDER_ROUTER_CONF_WEBSERVER
    PROCESS_NAME(webserver_nogui_process);
    process_start(&webserver_nogui_process, NULL);
#endif /* BORDER_ROUTER_CONF_WEBSERVER */

    LOG_INFO("Contiki-NG Border Router iniciado!!\n");

    /* Inicializa las conexiones UDP */
    simple_udp_register(&udp_conn_client1, UDP_LOCAL_PORT1, NULL,
                      UDP_REMOTE_PORT1, udp_rx_callback);
//Cliente 1

    simple_udp_register(&udp_conn_client2, UDP_LOCAL_PORT2, NULL,
                      UDP_REMOTE_PORT2, udp_rx_callback);
//Cliente 2

    simple_udp_register(&udp_connMV, UDP_MV_LOCAL_PORT, NULL,

```

```
                                UDP_MV_REMOTE_PORT, udp_rx_callback);  
//MV  
    PROCESS_END();  
}  
/*-----  
*/
```

## ANEXO O: Código Phyton

```
#!/usr/bin/env python3
# Python Bridge
# Recibe mensajes UDP al puerto definido por el usuario y envÃa a HiveMQ a
travÃs del cliente de MOSQUITTO-MQTT
import socket
import sys
import subprocess

if len(sys.argv) == 3:
    # Obtiene los argumentos: 1) IPv6 2) Puerto
    ip = sys.argv[1]
    port = int(sys.argv[2])
else:
    print("Utilizar como: python3 bridge.py <direccion IPv6 - fe00::1>
<puerto local - 4444 >")
    exit(1)

# Se crea el socket UDP para IPv6 (INET6)
s = socket.socket(socket.AF_INET6, socket.SOCK_DGRAM)
# Se enlaza el puerto con la direccion ingresados por teclado
server_address = (ip, port)
s.bind(server_address)
print("Ctrl+c para terminar el proceso!!")
print("##### El puente se encuentra escuchando #####")
while True:
    # Se reciben mensajes UDP y se imprimen en pantalla
    data, address = s.recvfrom(4096)
    msg = data.decode('utf-8')
    print(msg)
    # Se llama al ejecutable
    subprocess.call(['sh', './bridge.sh',msg])
#FIN
```

## **ANEXO P: Código Phyton ejecutable**

```
#!/bin/sh
```

```
# Utiliza el cliente MOSQUITTO-MQTT para publicar en HiveMQ
```

```
echo "Script ejecutado...\n"
```

```
echo $1
```

```
sudo mosquitto_pub -h f2b1609840084b779ce38ba266e6eb8d.s1.eu.hivemq.cloud -p  
8883 -u acuasquer -P admin -t 'my/test/topic' -m $1
```

## ANEXO Q: MANUAL DE USUARIO



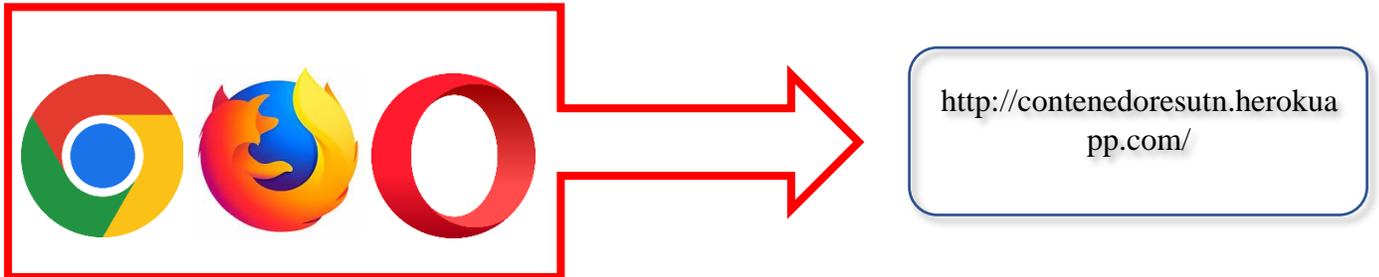
**“RED MESH LPWAN MEDIANTE EL USO DEL ESTÁNDAR IEEE 802.15.4G PARA LA RECOLECCIÓN DE DATOS SOBRE VARIABLES EN LOS CONTENEDORES DE RESIDUOS SÓLIDOS DEL SECTOR YACUCALLE DE LA CIUDAD DE IBARRA PROVINCIA DE IMBABURA”**

**MANUAL DE USUARIO**

**AUTOR: ANDREA ELIZABETH CUÁSQUER IRUA**

**Ibarra-Ecuador**

1. Para poder acceder a la información proporcionada debe ingresar a cualquier navegador que disponga en su dispositivo móvil o en su computadora. Ingresar al siguiente enlace.



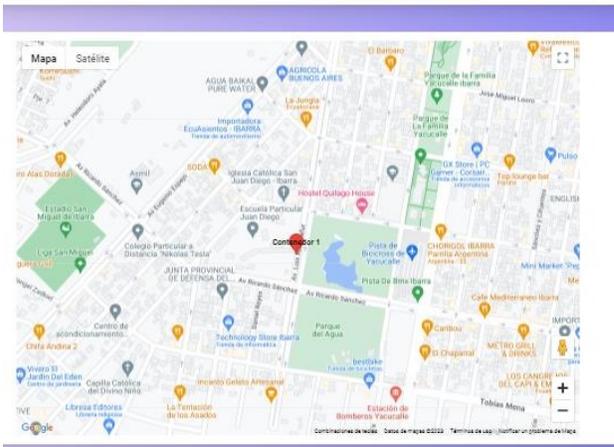
2. Se desplegará la información en tiempo real sobre los contenedores que ya están en estado lleno, se los podrá identificar con el símbolo de ubicación dentro del mapa.

INFORMACIÓN DE LAS VARIABLES DEL  
CONTENERO QUE ESTA LLENO



UBICACIÓN DEL  
CONTENEDOR  
QUE ESTA LLENO  
DENTRO DEL  
MAPA

# UBICACION



En base a las ubicaciones de los contenedores el vehículo podrá establecer las rutas de recolección para visitar únicamente los sectores que contengan contenedores ya llenos.