



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

**DISEÑO E IMPLEMENTACIÓN DE TARJETA ELECTRÓNICA PARA PLUVIÓMETROS DE
BALANCÍN PARA COMUNICACIÓN EN RED LORAWAN Y VISUALIZACIÓN EN TIEMPO
REAL**

Enrique Manuel José Coloch Tahuico

Asesorado por el Ing. Julio César Solares Peñate

Guatemala, octubre de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO E IMPLEMENTACIÓN DE TARJETA ELECTRÓNICA PARA PLUVIÓMETROS DE
BALANCÍN PARA COMUNICACIÓN EN RED LORAWAN Y VISUALIZACIÓN EN TIEMPO
REAL**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

ENRIQUE MANUEL JOSÉ COLOCH TAHUICO
ASESORADO POR EL ING. JULIO CÉSAR SOLARES PEÑATE

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO ELECTRÓNICO

GUATEMALA, OCTUBRE DE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Kevin Vladimir armando Cruz
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Hugo Leonel Tiul Valenzuela
EXAMINADOR	Ing. Christian Antonio Orellana López
EXAMINADOR	Ing. Sergio Leonel Gómez Bravo
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**DISEÑO E IMPLEMENTACIÓN DE TARJETA ELECTRÓNICA PARA PLUVIÓMETROS DE
BALANCÍN PARA COMUNICACIÓN EN RED LORAWAN Y VISUALIZACIÓN EN TIEMPO
REAL**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica eléctrica, con fecha abril 2021.

Enrique Manuel José Coloch Tahuico

Guatemala, 19 de mayo de 2021

**Señor
Coordinador del Área de Electrónica
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.**

Estimado Ingeniero:

Por este medio me permito dar aprobación al trabajo de Graduación titulado: **DISEÑO E IMPLEMENTACIÓN DE TARJETA ELECTRÓNICA PARA PLUVIÓMETROS DE BALANCÍN PARA COMUNICACIÓN EN RED LORAWAN Y VISUALIZACIÓN EN TIEMPO REAL**, desarrollado por el estudiante **Enrique Manuel José Coloch Tahuico**, ya que considero que cumple con los requisitos establecidos.

Por lo tanto, el autor de este trabajo y yo como asesor, nos hacemos responsables del contenido y conclusiones del mismo.

Sin otro en particular, aprovecho la oportunidad para saludarlo.

ID Y ENSEÑAD A TODOS

A handwritten signature in blue ink, appearing to read 'Julio César Solares Peñate', is centered on the page.

**Ing. Julio César Solares Peñate
Asesor**



Guatemala, 25 de mayo de 2021

Señor Director
Armando Alonso Rivera Carrillo
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC

Estimado Señor director:

Por este medio me permito dar aprobación al Trabajo de Graduación titulado: **DISEÑO E IMPLEMENTACIÓN DE TARJETA ELECTRÓNICA PARA PLUVIÓMETROS DE BALANCÍN PARA COMUNICACIÓN EN RED LORAWAN Y VISUALIZACIÓN EN TIEMPO REAL**, desarrollado por el estudiante **Enrique Manuel José Coloch Tahuico**, ya que considero que cumple con los requisitos establecidos.

Sin otro particular, aprovecho la oportunidad para saludarlo.

Atentamente,

ID Y ENSEÑAD A TODOS

A handwritten signature in blue ink, appearing to read 'Julio César Solares Peñate'.

Ing. Julio César Solares Peñate
Coordinador de Electrónica



REF. EIME 151 2021.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; ENRIQUE MANUEL JOSÉ COLOCH TAHUICO, titulado: DISEÑO E IMPLEMENTACIÓN DE TARJETA ELECTRÓNICA PARA PLUVIÓMETROS DE BALANCÍN PARA COMUNICACIÓN EN RED LORAWAN Y VISUALIZACIÓN EN TIEMPO REAL, procede a la autorización del mismo.

Ing. Armando Alonso Rivera Carrillo



GUATEMALA, 19 DE OCTUBRE 2,021.



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

Decanato
Facultad de Ingeniería
24189101 - 24189102
secretariadecanato@ingenieria.usac.edu.gt

DTG. 531-2021

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **DISEÑO E IMPLEMENTACIÓN DE TARJETA ELECTRÓNICA PARA PLUVIÓMETROS DE BALANCÍN PARA COMUNICACIÓN EN RED LORAWAN Y VISUALIZACIÓN EN TIEMPO REAL**, presentado por el estudiante universitario: **Enrique Manuel José Coloch Tahuico**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:


Inga. Anabela Cordova Estrada
Decana



Guatemala, octubre de 2021

AACE/cc

ACTO QUE DEDICO A:

Dios	Por proveerme de sabiduría para alcanzar esta importante meta.
Mis padres	José Coloch y Alberta Tahuico, nunca habrá con qué pagarles el esfuerzo, amor y el apoyo incondicional, por haberme educado con excelentes valores y ser mi inspiración.
Mis hermanos	Sara y Alan Coloch, por su generosidad y amor, es una bendición haber crecido a su lado.
Mis abuelos	José León Coloch, María Xolop, Pedro Tahuico, Sara Raxcacó, por su cariño y consejos que me han servido en cada día de mi vida.
Mi novia	Maribel con especial cariño y con mucho amor por su importante influencia en mi carrera.
Mi familia	Por brindarme su afecto y estima siempre.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por proveerme de enseñanza y herramientas para mi vida profesional.
Facultad de Ingeniería	Por ser pilar importante de mi carrera y darme la formación profesional.
Mi asesor	Ing. Julio Solares, por su asesoría y su contribución en la elaboración de este proyecto de graduación.
Mis amigos	Por brindarme su amistad y aportar a mi desarrollo.
Mis compañeros	Por su aporte y trabajo en equipo en cada proyecto elaborado durante la carrera.
Ingenio Magdalena	Por permitirme utilizar instalaciones para construir este proyecto, en especial Gustavo Rosal por la confianza.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
LISTA DE SÍMBOLOS	IX
GLOSARIO	XI
RESUMEN	XXVII
OBJETIVOS.....	XXIX
INTRODUCCIÓN	XXXI
1. PLUVIÓMETRO DE BALANCÍN	1
1.1. Principio de funcionamiento.....	1
1.1.1. Medición de lluvia	3
1.1.2. Partes del pluviómetro de balancín.....	5
1.2. Pluviómetro de balancín a utilizar	8
1.2.1. Ampliación de área de captación.....	10
1.2.2. Adaptación de balancín	10
1.2.3. Cubierta	11
1.2.4. Construcción.....	12
1.2.5. Error del instrumento	13
1.3. Transmisión de datos	16
1.3.1. Esquema general.....	17
2. DISEÑO DE TARJETA ELECTRÓNICA	19
2.1. Evaluación de requerimientos	19
2.2. Microcontrolador.....	21
2.2.1. Microcontrolador ATMEGA328P.....	22
2.2.1.1. Memoria Flash	25

	2.2.1.2.	Memoria SRAM	25
	2.2.1.3.	Memoria EEPROM.....	26
	2.2.1.4.	Interfaz USART	27
		2.2.1.4.1. Software serial	27
	2.2.1.5.	Interfaz SPI.....	28
	2.2.1.6.	Unidad ADC	29
	2.2.1.7.	Interfaz TWI.....	31
	2.2.1.8.	Puertos GPIO	32
	2.2.1.9.	Interrupciones.....	32
	2.2.2.	Arduino Pro Mini.....	35
		2.2.2.1. Programación de Arduino Pro mini.....	37
		2.2.2.2. Arduino IDE	38
2.3.		Circuito de medición de lluvia.....	39
	2.3.1.	Circuito anti-rebote	40
	2.3.2.	Conexión al microcontrolador.....	45
	2.3.3.	Programación de interrupciones.....	46
2.4.		GPS.....	47
	2.4.1.	Funcionamiento de GPS	48
	2.4.2.	Modulo GPS NEO-6MV2.....	49
	2.4.3.	Circuito de conexión módulo GPS.....	50
	2.4.4.	Programación de GPS	51
		2.4.4.1. Librería TinyGPS.....	52
2.5.		LoRa.....	53
	2.5.1.	Modulación LoRa	53
		2.5.1.1. Spreading factor (SF)	55
		2.5.1.2. Coding Rate (CR).....	56
		2.5.1.3. Data Rate (DR).....	57
	2.5.2.	Módulo RFM95W	59
	2.5.3.	Conexión al microcontrolador.....	60

2.5.4.	Programación de LoRa	61
2.5.4.1.	Librería LMIC	63
2.5.4.2.	Payload pluviómetro de balancín.....	67
2.6.	Algoritmo de tarjeta electrónica para pluviómetro de balancín	69
2.7.	Energía.....	72
2.7.1.	Batería	73
2.7.2.	Módulo de carga TP4056	75
2.7.3.	Panel fotovoltaico	76
2.7.4.	Circuito de energía	77
2.8.	Implementación de tarjeta electrónica	78
2.8.1.	Circuito esquemático	78
2.8.2.	Circuito PCB	80
2.8.3.	Construcción de tarjeta y resultados.....	82
3.	IMPLEMENTACIÓN DE RED LORAWAN.....	85
3.1.	Redes LPWAN	86
3.2.	LoRaWAN.....	89
3.2.1.	Arquitectura	90
3.2.2.	Frecuencias	93
3.2.2.1.	Regulaciones de radiofrecuencia en Guatemala	94
3.2.2.2.	Banda 902-928MHz Guatemala	96
3.2.2.3.	RIABAP	98
3.3.	Servidor LoRaWAN	99
3.3.1.	Chirpstack.....	101
3.3.1.1.	Arquitectura de Chirpstack.io.....	101
3.3.1.2.	Broker MQTT	103
3.3.1.3.	Chirpstack Gateway Bridge	105

3.3.1.4.	Chirpstack Network Server.....	106
3.3.1.5.	Chirpstack Application Server	106
3.3.1.6.	Integraciones.....	107
3.3.2.	Implementación de servidor LoRaWAN	107
3.4.	Gateway LoRaWAN	114
3.4.1.	Conexión de gateway a la red LoRaWAN	116
3.5.	Nodos LoRaWAN.....	122
3.5.1.	Pluviómetro de balancín en red LoRaWAN.....	124
4.	VISUALIZACIÓN DE DATOS	131
4.1.	Thingsboard.io	131
4.2.	Instalación y configuración Thingsboard	133
4.3.	Integración de pluviómetros de red LoRaWAN	138
4.4.	Construcción de dashboard	143
4.5.	Alertas.....	146
4.5.1.	AWS SNS.....	147
4.5.1.1.	Configuración AWS SNS.....	148
4.5.1.2.	Creación de usuario AWS para Thingsboard	150
4.5.1.3.	Configuración SNS Thingsboard.....	151
4.5.2.	Alertas de lluvia en Thingsboard	152
4.6.	Escalabilidad	156
	CONCLUSIONES.....	161
	RECOMENDACIONES	163
	BIBLIOGRAFÍA.....	165
	APÉNDICES.....	173

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Balancín	2
2.	Funcionamiento de balancín	3
3.	Litro de agua en 1 m ²	4
4.	Sensor <i>Reed switch</i>	7
5.	Partes del pluviómetro de balancín	8
6.	Pluviómetro MISOL modelo WH-SP-RG	9
7.	Ampliación de área de captación para WH-SP-RG.....	10
8.	Adaptación embudo y WH-SP-RG	11
9.	Diseño final de pluviómetro	12
10.	Construcción de pluviómetro.....	13
11.	Gráfica Volumen y Volteos.....	15
12.	Esquema general del proyecto.....	17
13.	Esquema de tarjeta electrónica	20
14.	Arquitectura AVR ATMEGA328P	24
15.	Memoria SRAM ATMEGA328P	26
16.	Arquitectura SPI	29
17.	Muestreo señal analógica	30
18.	Arquitectura protocolo TWI.....	31
19.	Arduino pro mini	37
20.	PINOUT Arduino pro mini.....	37
21.	Conexión para programador FTDI.....	38
22.	Ancho de pulsos de pluviómetro de balancín.....	42
23.	Pulsos teóricos de récord de lluvia.....	43

24.	Circuito RC	44
25.	Circuito de medición de lluvia	46
26.	Signal Edge	47
27.	Trilateración	48
28.	Módulo GPS NEO6MV2	49
29.	Conexión GPS	51
30.	Up-Chirp y Down-Chirp	54
31.	Concepto de codificación	56
32.	Módulo LoRa RFM95W	59
33.	PINOUT RFM95W	60
34.	Conexión RFM95W.....	61
35.	Arquitectura de firmware LMIC	62
36.	Pinmap RFM95W.....	64
37.	Diagrama de flujo de programa tarjeta de pluviómetro	70
38.	Batería 18650	75
39.	Módulo TP4056.....	76
40.	Panel solar 5.5 V 160mA	77
41.	Circuito de energía.....	78
42.	Diagrama esquemático tarjeta de pluviómetro de balancín	79
43.	PCB tarjeta vista superior	80
44.	PCB tarjeta vista inferior	81
45.	Tarjeta electrónica para pluviómetro de balancín frontal	82
46.	Tarjeta electrónica para pluviómetro de balancín trasera	83
47.	Comparación tecnologías inalámbricas	87
48.	Comparación LPWAN costo y velocidad	88
49.	Arquitectura LoRaWAN.....	90
50.	Modelo LoRaWAN	92
51.	Canales LoRaWAN US915MHz.....	94
52.	Banda 902-928Mhz Cuadro TNAF.....	97

53.	Arquitectura Chirpstack.io	102
54.	Modelo MQTT	104
55.	Login Chirpstack	112
56.	Web Chirpstack.....	113
57.	Instalación de Gateway UG87.....	117
58.	Creación de perfil de Gateway UG87.....	118
59.	IP para configuración de Gateway	119
60.	Interfaz de configuración de Gateway UG87.....	120
61.	Configuración de Packet Forwarder Gateway UG87.....	120
62.	Configuración IP de Gateway UG87	121
63.	Habilitación de gateway UG87	122
64.	Clase de nodos LoRaWAN	123
65.	Creación de dispositivo en Chirpstack	127
66.	Autenticación de nodo en Chirpstack.....	128
67.	Llaves en programa de tarjeta de pluviómetro	128
68.	Dispositivo activo en red LoRaWAN.....	129
69.	Arquitectura Thingsboard	132
70.	Thingsboard WEB	136
71.	Crear organización Thingsboard WEB.....	137
72.	Crear usuario administrador.....	138
73.	Perfil de dispositivo pluviómetro.....	139
74.	Credenciales de dispositivo Thingsboard.....	140
75.	Credenciales en Chirpstack	141
76.	Integración Thingsboard.....	142
77.	Última telemetría Thingsboard	142
78.	Edición de dashboard.....	143
79.	Creación de alias	144
80.	Función de cálculo de milímetros.....	145
81.	Visualización en tiempo real.....	146

82.	Configuración tópico Amazon SNS.....	149
83.	Configuración tópico Amazon SNS.....	149
84.	Creación de usuario AWS para Thingsboard.....	151
85.	Configuración SMS Thingsboard.....	152
86.	Regla principal Thingsboard.....	153
87.	Regla de alarmas SMS Thingsboard.....	154
88.	Alarmas dashboard.....	155
89.	Alarma SMS.....	155
90.	Proyecto de 175 pluviómetros con LoRaWAN.....	159

TABLAS

I.	Muestra de volúmenes y volteos.....	14
II.	Interfaces módulos-microcontrolador.....	22
III.	Interrupciones externas ATMEGA328P.....	34
IV.	Activación de interrupciones.....	34
V.	Especificaciones NEOG6MV2.....	49
VI.	Trama NMEA de GPS NEO6MV2.....	52
VII.	Ejemplos de Data Rate.....	58
VIII.	Funciones de ejecución LMIC.....	65
IX.	Log de eventos LMIC.....	66
X.	Payload lluvia.....	69
XI.	Payload ubicación.....	69
XII.	Identificación de conectores tarjeta.....	81
XIII.	Parámetros técnicos RIABAP.....	98

LISTA DE SÍMBOLOS

Símbolo	Significado
A	Amperio
BW	Ancho de banda
bps	Bits por segundo
cm	Centímetro
cm²	Centímetro cuadrado
CR	Coding Rate
DR	Data Rate
dB	Decibeles
dB_i	Decibeles de ganancia
dB_m	Decibeles por metro
gls	Galones
Hz	Hertz
hrs	Horas
Kb	Kilo bits
KB	Kilo Bytes
KΩ	Kilo Ohm
Km	Kilómetro
L	Litro
MHz	Mega Hertz
m	Metro
m²	Metro cuadrado
μF	Microfaradio
mA	Miliamperio

mAh	Miliamperio por hora
ml	mililitro
mm	Milímetro
ms	Milisegundo
mW	Miliwatt
%	Porcentaje
RX	Receptor
s	Segundo
SF	Spreading Factor
TX	Transmisor
V	Voltio
W	Watt

GLOSARIO

ADC	Convertidor de señal analógico a digital.
ALU	Unidad aritmética lógica.
Analógico	Señal variante en el tiempo con valores no discretos.
API	Interfaz de programación de aplicaciones.
<i>Application Server</i>	Servidor de aplicación.
Baudrate	Tasa de baudios o bits por segundo.
Blockchain	Tecnología de servidores descentralizados.
Bootloader	Gestor de arranque de hardware.
Broker	Organizador central de transacciones.
Capacitancia	Capacidad de condensador electrónico.
Codificación	Consiste en representar de manera distinta un mensaje.
Dashboard	Pantalla que ordena de en varias gráfica datos.

Embudo	Instrumento para canalizar líquidos.
Firmware	Programa informático que establece la lógica de los circuitos.
Framework	Entorno de trabajo con estándares sobre el cual se construyen aplicaciones.
Full duplex	Comunicación capaz de mantener bidireccionalidad de forma simultánea.
Gateway	Dispositivo tipo puerta de enlace o pasarela de una red hacia internet.
HTTP	Protocolo de transferencia de hipertexto.
IDE	Entorno de desarrollo integrado.
Inteligencia Artificial	Concepto tecnológico para referirse a sistemas que imitan funciones cognitivas.
Interfaz	Conexión entre dos sistemas.
<i>Internet of Things</i>	Interconexión digital de objetos cotidianos con internet.
<i>IoT</i>	Internet of things.
Log	Registro secuencial de eventos.

LPWAN	Red de área amplia y de baja potencia.
<i>Machine learning</i>	Rama de la inteligencia artificial que utiliza algoritmos que aprenden a partir de datos.
Modulación	Técnica para transportar información sobre una onda portadora.
MOSFET	Transistor de efecto de campo metal-óxido-semiconductor.
Network Server	Servidor de red.
Networking	Interconexión de redes.
Open Source	Código de software con licencia pública.
Payload	Carga útil de datos.
PCB	Placa de Circuito impreso.
Pinout	Disposición de pines de un chip.
PoE	Alimentación a través de ethernet.
RF	Radiofrecuencia.

RESUMEN

La lluvia es un fenómeno natural esencial para la vida en nuestro planeta, desde tiempos antiguos la humanidad ha buscado conocer el comportamiento de estos eventos y sacar algún provecho de ello.

El proyecto desarrollado en este documento presenta una alternativa para la medición totalmente automatizada de la lluvia, altamente escalable y de bajo costo. Haciendo uso de un pluviómetro de balancín y para la transmisión de datos una red LoRaWAN.

Además de la medición y transmisión de datos se muestra un caso de uso para el aprovechamiento de estos, agregando el concepto de internet de las cosas con una plataforma de visualización de datos en tiempo real y generación de alertas.

Se compone de cuatro capítulos abarcando cada uno de los componentes que integran esta solución de internet de las cosas, iniciando con el instrumento de medición, luego el diseño y construcción de la tarjeta electrónica controladora del pluviómetro. Posteriormente se aborda la implementación de una red LoRaWAN. El proyecto finaliza entregando un portal web para usuarios finales con una pantalla de visualización de datos en tiempo real y envío de alertas vía SMS.

Este proyecto busca dar una herramienta sumamente útil a los usuarios para tomar decisiones basadas en datos y también coloca la plataforma sobre la

cual pueden integrarse otros dispositivos, sensores o datos para realizar análisis más complejos.

OBJETIVOS

General

Diseñar e implementar una tarjeta electrónica para pluviómetro de balancín para comunicación en red LoRaWAN y visualización en tiempo real.

Específicos

1. Describir el funcionamiento del pluviómetro de balancín y la metodología para medir la lluvia.
2. Construir una tarjeta electrónica capaz de medir y transmitir a largas distancias con poca energía, datos de lluvia.
3. Implementar una red LoRaWAN privada para la conexión de dispositivos a aplicaciones en internet.
4. Implementar una herramienta de visualización gráfica en tiempo real de la variable climática de precipitación para usuarios finales y programar alertas de detección de eventos de lluvia.

INTRODUCCIÓN

Todos hemos sido afectados alguna vez por el clima y su variabilidad para bien o para mal, son sucesos naturales incontrolables, pero con gran efecto en el transcurrir cotidiano de la humanidad. En la actualidad, existen distintas tecnologías para registrar, analizar y predecir con alcance limitado de tiempo los eventos meteorológicos.

Hoy en día se están desarrollando nuevas técnicas de predicción basados en *machine learning* e inteligencia artificial, los cuales necesitan de datos y registros históricos para alimentar sus modelos y entrenarlos, lo cual hace esencial la recolección de información de forma automática y en un lapso corto.

El desarrollo de estas herramientas y aplicaciones tecnológicas surge de la necesidad de tomar decisiones basadas en datos y acciones que se traduzcan en beneficios para los distintos procesos y áreas de la sociedad.

La lluvia es uno estos fenómenos físicos naturales, la cual se entiende como la caída de agua proveniente de las nubes hacia el suelo, este fenómeno tiene gran influencia en distintos campos laborales, principalmente en agricultura, generación de energía, en obras de ingeniería civil, marina, aviación, entre otros.

Es importante el estudio de la técnica para la medición del fenómeno para posteriormente diseñar y describir el funcionamiento de la tarjeta electrónica y del protocolo de comunicación para la transmisión de datos. Y posteriormente entregar a un cliente final una herramienta útil mediante una pantalla gráfica y alertas.

1. PLUVIÓMETRO DE BALANCÍN

En este capítulo se presenta un instrumento de medición denominada como el nombre del capítulo y sirve para registrar la cantidad de lluvia que cae en un punto geográfico determinado.

Se eligió presentar la siguiente estructura para que la lectura sea más entendible, conociendo inicialmente conceptos básicos y principio de funcionamiento del pluviómetro de balancín, luego la propuesta de diseño y construcción de un pluviómetro de bajo costo, así como el esquema general del proyecto a implementar para la transmisión y visualización de información en tiempo real.

1.1. Principio de funcionamiento

Un pluviómetro de balancín es un medidor de lluvia que utiliza un método directo para obtener el valor de volumen de agua. Se llama así debido a que su parte principal es un balancín o cazoleta colocado al centro del instrumento, esta parte del pluviómetro tiene la característica de tener una geometría simétrica y está asentado sobre un pivote como puede observarse en la figura 1.

Figura 1. **Balancín**



Fuente: elaboración propia.

El proceso de medición de lluvia inicia con la recolección de la muestra de agua, la cual ingresa al instrumento por medio del área de captación y es conducida por efecto de gravedad por medio de un embudo hacía el balancín, una parte de dicho balancín siempre está posicionada debajo del embudo, de tal manera que pueda ser llenado con el agua de lluvia recolectada.

Cuando este lado del balancín se llena o llega a cierto volumen de agua, el balancín es obligado a eliminar el agua por efecto del peso de esta, a esto se le conoce como volteo, lo cual implica que la segunda parte del balancín se posicione debajo del embudo repitiendo el procedimiento. Se muestra la forma de balancín basculante en la figura 2.¹

Para registrar los volteos se utiliza un sistema con aprovechamiento del efecto magnético el cual consiste en un imán permanente instalado en el centro del balancín y un sensor por lo regular un “reed switch” magnético posicionado

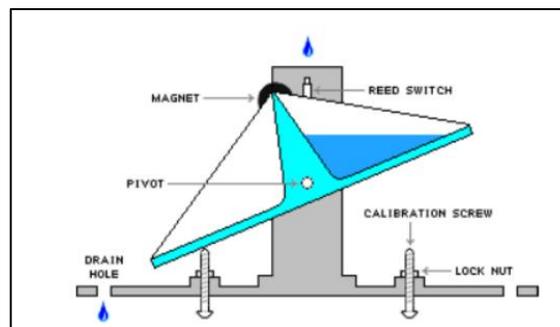
¹ AREIZA, Y.; VARGAS, G.; PÉREZ, C. E.; ARISTIZABAL, J. C.: SALAZAR, A. *Diseño, construcción y prueba de un pluviómetro de cazoletas con comunicación inalámbrica.* <http://revcolfis.org/ojs/index.php/rcf/article/download/440109/364>.

de tal manera que cada vez que se produzca un volteo cierre el circuito generando un pulso que será leído por una tarjeta electrónica o registrador.

Cada volteo representa determinado volumen de agua de lluvia caído en el área de captación. Este valor posteriormente se convertirá en mm, debido a que la lluvia se representa como una altura, como se explica en la sección siguiente.

Una de las ventajas del pluviómetro de balancín con respecto a otros sistemas de medición de lluvia, es la autonomía ya que no almacena agua. Es un sistema basculante que puede funcionar sin intervención humana más que limpieza periódica.

Figura 2. **Funcionamiento de balancín**



Fuente: ROJAS BARRET, Pedro. *Diseño estaciones meteorológicas en las cuencas hidrográficas de CONELECTRICAS R.L.* <https://docplayer.es/68639575-Instituto-tecnologico-de-costa-rica-escuela-de-ingenieria-electronica.html>. Consulta: marzo 2020.

1.1.1. **Medición de lluvia**

La Organización Meteorológica Mundial² define la lluvia como la precipitación de partículas líquidas de agua, de diámetro mayor a 0,5 mm. Su

² Organización Meteorológica Mundial. *Guía de Prácticas Hidrológicas*. p.1-3.

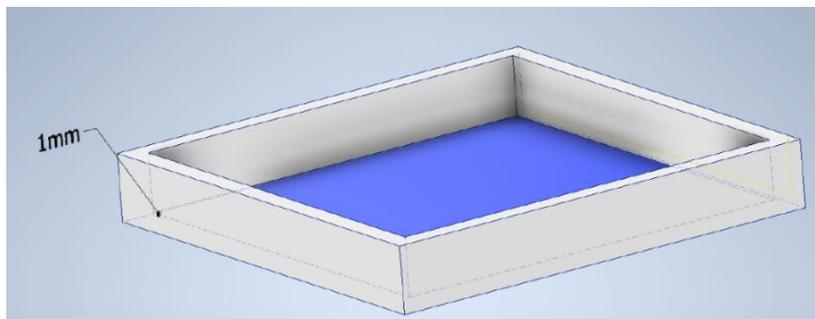
origen inicia con la condensación de vapor de agua en las nubes y esto se debe a cambios de presión, temperatura y radiación solar.

Para medir la precipitación se utiliza la lámina de agua caída en una superficie plana. Se mide como una altura en milímetros de agua y también se puede representar como litros caídos por unidad de área (L/m^2).

Una observación muy importante es la relación: Un milímetro de altura de agua de lluvia, equivale a 1 Litro de agua en un 1 metro cuadrado.

Es decir, si se vertiera un litro de agua en un tanque de 1 metro cuadrado, la lámina tendría un milímetro de altura. Se observa el ejemplo en la figura 3.

Figura 3. **Litro de agua en 1 m²**



Fuente: elaboración propia, empleando Autodesk Inventor 2018.

Esta relación nos permite realizar la conversión de volumen de agua a milímetros de lluvia, dado que está claro que en este caso es más fácil medir volumen que un nivel de agua.

La conversión de volumen en un área determinada a milímetros de lluvia es:

Ecuación 1:

$$h = \frac{V}{A}$$

Para un área de captación circular se tiene:

Ecuación 2:

$$h = \frac{V}{\pi\left(\frac{d}{2}\right)^2}$$

Donde la h representa la altura de la lámina, V se define como el volumen de agua captado en el área circular de diámetro d .

Teniendo en cuenta que, con el pluviómetro de balancín se detecta el volteo que se realiza tras superar el volumen necesario para que el balancín cambie de posición, se puede tener la información de cuantos mm representan cada volteo.

1.1.2. Partes del pluviómetro de balancín

Un pluviómetro de balancín se compone de las siguientes partes:

- Área de captación: es el área utilizada para recolectar el agua de lluvia, se encuentra en la parte superior del pluviómetro. “El diámetro de la boca del colector no es importante. En algunos países se utiliza una superficie receptora de unos 1 000 cm^2 , aunque entre 200 y 500 cm^2 será probablemente un tamaño más práctico”³. El área de captación puede ser variable, sin embargo, es necesario conocer el valor con precisión del 0,5 % para realizar la conversión de volumen de agua en m^3 a lámina de lluvia en mm .

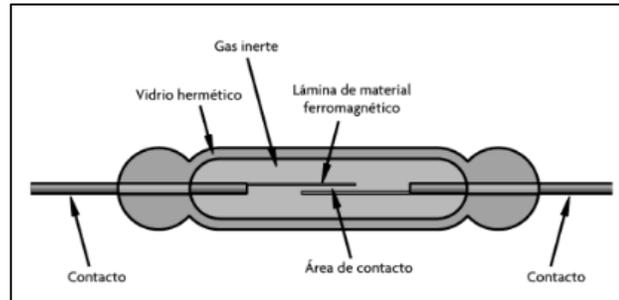
³ Organización Meteorológica Mundial. *Guía de Prácticas Hidrológicas*. p. 1-3.

El área generalmente es circular y el agua captada es conducida por medio de un embudo hacia el balancín. El material que lo compone debe ser rígido para no tener variaciones de área, regularmente se utiliza aluminio.

- **Balancín:** Es la parte principal del medidor, se trata de una pieza simétrica montada sobre un pivote, la geometría de esta pieza está completamente ligada a la resolución del pluviómetro, debido a que cada balanceo que realice sumará un valor en mm de precipitación.
- **Sensor *reed switch*:** Un *reed switch* o interruptor de lengüeta es un transductor entre un campo magnético y un contacto mecánico. Este sensor se compone de dos láminas ferromagnéticas que reaccionan uniéndose o separándose en presencia o ausencia de un campo magnético dependiendo de su diseño NC o NA (Normalmente Cerrado o Normalmente Abierto). Se aprovecha esta conmutación para permitir el paso o no de corriente entre sus contactos.

Estos elementos están en un encapsulado de vidrio e inmersos en un gas, típicamente nitrógeno con el objetivo de protegerlos de desgastes consecuentes del medio.

Figura 4. **Sensor Reed switch**



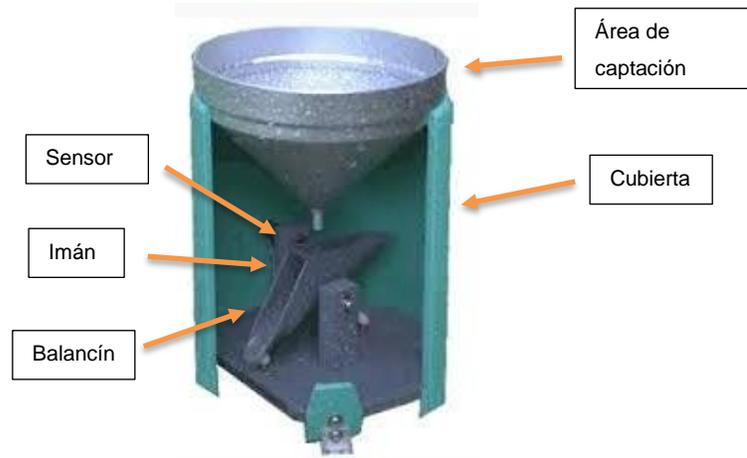
Fuente: CORONA RAMÍREZ, Leonel; ABARCA JIMÉNEZ, Griselda; MARES CARREÑO, Jesús.
Sensores y actuadores. p. 171.

El sensor de un pluviómetro es colocado en una posición en la cual se pueda detectar el giro del balancín, lo cual dará información de lluvia.

- **Imán:** se aprovecha el campo magnético generado por un imán permanente para excitar los contactos del *reed switch*, está colocado en el centro del balancín y pasará sobre el sensor cuando éste cambie de posición.
- **Cable:** el cable lleva los dos contactos del sensor del pluviómetro y generalmente se utiliza una interfaz RJ11 para la conexión a una tarjeta electrónica que registrará los cambios que se produzcan en el balancín y medir la lluvia.
- **Cubierta:** es la base y el aislamiento del ambiente que tiene el pluviómetro y donde se sostienen las partes, típicamente se utilizan materiales resistentes a la intemperie como aluminio y acero inoxidable. En la instalación, está cubierta tiene que quedar completamente nivelada para que el sistema de pivoteo del balancín funcione correctamente.⁴

⁴ CORONA RAMÍREZ, Leonel; ABARCA JIMÉNEZ, Griselda; MARES CARREÑO, Jesús.
Sensores y actuadores. p. 171.

Figura 5. Partes del pluviómetro de balancín



Fuente: LÓPEZ-REY, David. *Pluviómetros y pluviógrafos: Un afán histórico por medir la lluvia*. <https://www.tutiempo.net/meteorologia/articulos/pluviometro-y-pluviografo.html>. Consulta: 1 de marzo de 2020.

1.2. Pluviómetro de balancín a utilizar

Los pluviómetros de balancín existentes en el mercado varían de precio según la resolución del balancín, así como el material del cual están fabricados. Entre los materiales más utilizados se pueden encontrar aluminio y acero inoxidable, los cuales pueden estar a la intemperie sin afectar su estado en un tiempo considerable.

En el mercado también existen pluviómetros balancín que son de materiales menos resistentes, más pequeños y menor área de captación, pero de bajo costo. Una opción para aprovechar estos pluviómetros de bajo costo y utilizar en condiciones de exterior. Es construir una cubierta de material resistente para proteger el balancín.

Esto se traduce en un costo menor al conseguir un pluviómetro y con el mismo aprovechamiento del método de medición de lluvia.

Figura 6. **Pluviómetro MISOL modelo WH-SP-RG**



Fuente: mi sol. *Weather Station*. <http://www.misolie.net/misol-spare-part-for-weather-station-to-measure-the-rain-volume-for-rain-meter-for-rain-gauge-p-513.html>. Consulta: 2 de marzo de 2020.

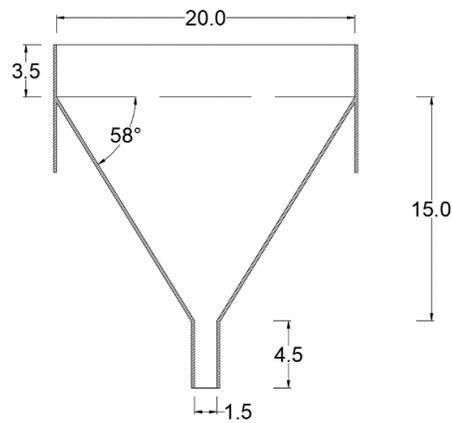
Este pluviómetro tiene tres desventajas, la primera es el material de plástico del cual está fabricado, lo cual lo hace frágil en un entorno rudo como la intemperie. La segunda y tercera desventaja se encuentran en el área de captación, la cual es de 55 cm^2 , mucho menor a la dicha como práctica por la OMM, también se considera que, ante lluvias torrenciales, al no tener un cono pronunciado o embudo, el agua tiende a salpicar, lo cual es malo para la medición.

A continuación, se presenta una solución para aprovechar el pluviómetro marca MISOL modelo WH-SP-RG de bajo costo en condiciones que requieren mayor resistencia a la intemperie, y también mejorando el área de captación para reducir los errores en la medición.

1.2.1. Ampliación de área de captación

Para cubrir las recomendaciones de la OMM se necesita ampliar el área de captación de 55 cm² a un área mayor a 1 000 cm² y ampliar el ángulo de inclinación del embudo.

Figura 7. **Ampliación de área de captación para WH-SP-RG**

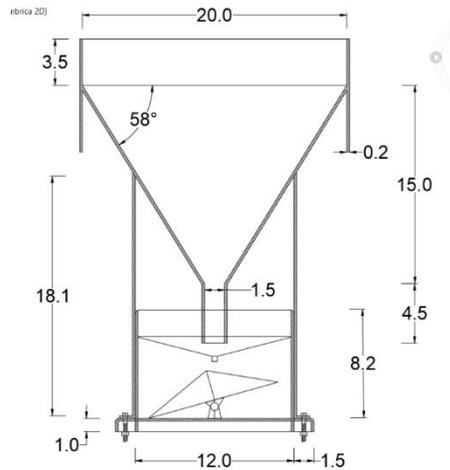


Fuente: elaboración propia, empleando AutoCAD 2018.

1.2.2. Adaptación de balancín

Para realizar la adaptación del medidor MISOL a la nueva área de captación se necesitará dos láminas de aluminio de 18.1 cm de largo para mantener sujeto el medidor con el área de captación. Este diseño nos garantiza que la desembocadura del área de captación quede alineada al balancín.

Figura 8. **Adaptación embudo y WH-SP-RG**

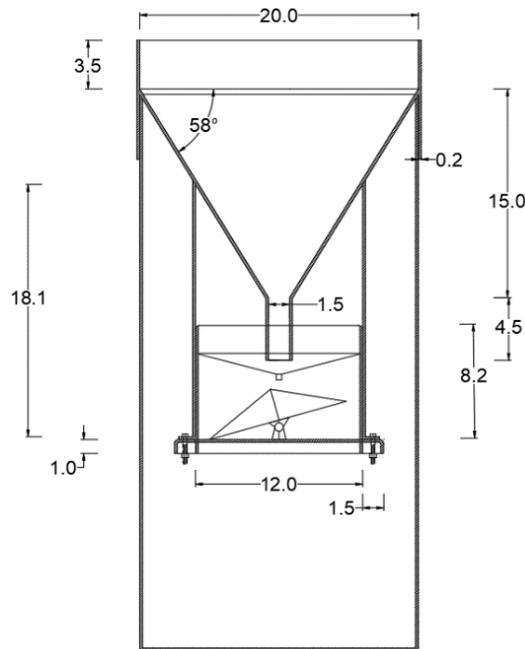


Fuente: elaboración propia, empleando AutoCAD 2018.

1.2.3. **Cubierta**

Debido a la resistencia ante intemperie, bajo costo y fácil manejo se optó por utilizar aluminio, la cubierta protegerá el balancín de la intemperie y le dará una base sólida y firme para que el área de captación siempre este nivelada. El diseño final del pluviómetro se muestra a continuación.

Figura 9. **Diseño final de pluviómetro**



Fuente: elaboración propia, empleando AutoCAD 2018.

1.2.4. **Construcción**

Llevando a cabo la adecuación del balancín con el área de captación ampliada y cubierta, se muestra el resultado del pluviómetro en aluminio. Uno de los problemas más frecuentes en este tipo de medidores es la acumulación de polvo y hojas en el área de captación, por lo cual se deja una malla que evitará que partículas sólidas obstruyan el flujo del agua hacia el balancín.

No obstante, esta solución no es a largo plazo, por lo que se recomienda una limpieza periódica tanto en el área de captación, balancín y un área de influencia libre de malezas alrededor de la base del pluviómetro.

Figura 10. **Construcción de pluviómetro**



Fuente: elaboración propia.

1.2.5. **Error del instrumento**

Como se explica en la sección 1.1.1. Medición de lluvia, se necesita conocer la altura en milímetros de la lámina de agua de una precipitación. En la sección anterior se definen las dimensiones del pluviómetro y a continuación se presentará la obtención de la resolución que tiene el instrumento, entendiéndose que la resolución se define como la cantidad de milímetros de lluvia que representa cada volteo del balancín.

La medición en milímetros es una medida indirecta que se realiza por medio de la ecuación 2 que nos relaciona el diámetro y volumen de agua para dar como resultado la lámina de agua.

La obtención del volumen de agua por cada volteo se podría hacer con la geometría, pero, el balancín no se llena completamente de agua cuando se voltea, por lo cual se recurre a una medida indirecta tomando muestras de un volumen de agua conocido y número de volteos.

Tabla I. **Muestra de volúmenes y volteos**

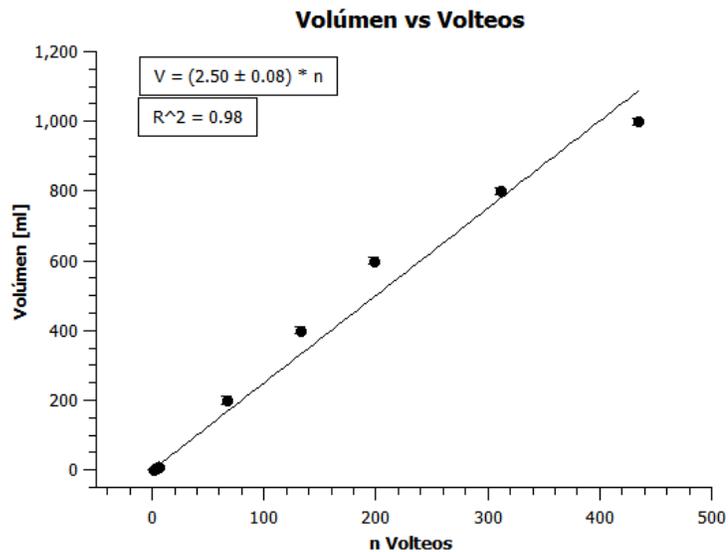
Muestra	V [ml]	Error [ml]	Volteos
1	1,6	0,1	1
2	3,4	0,1	2
3	4,9	0,1	3
4	6,6	0,1	4
5	8,6	0,1	5
6	10,5	0,1	6
7	200	10	67
8	400	10	132
9	600	10	198
10	800	10	312
11	1 000	10	434

Fuente: elaboración propia, empleando Microsoft Excel 365.

En la tabla I se observan los datos de los volúmenes de agua en mililitros con su incerteza y la cantidad de volteos que se produjeron al esparcir la muestra en el área de captación, al graficar queda de la siguiente forma.

Realizando un ajuste lineal de los datos, empleando el programa QtiPlot se obtuvo que, un volteo representa 2,50 mililitros de agua que a su vez relacionado con el área de captación presenta la siguiente resolución del pluviómetro.

Figura 11. Gráfica Volumen y Volteos



Fuente: elaboración propia, empleando QtiPlot.

Ecuación 3:

$$\Delta h = \frac{2,50\text{ml}}{\pi \left(\frac{0,2\text{m}}{2}\right)^2} * 0,001 \text{ m}^3 \approx 0,08 \text{ mm}$$

La medición de la precipitación en cualquier pluviómetro es afectada por factores diversos no controlables, como las corrientes de aire que se puedan generar en el área de captación, salpicaduras, gotas directas en el balancín, desnivel. Una de las desventajas que presenta este tipo de pluviómetro es la saturación ante una intensidad de lluvia muy alta.

1.3. Transmisión de datos

Las necesidades de tecnologías inalámbricas cada día son más urgentes y requieren de la implementación de nuevos métodos de traslado de información. *Internet of Things* es un concepto muy popular en las grandes ciudades y últimamente en hogares, empleando redes de datos tradicionales como WiFi o utilización de redes celulares, lo cual, a gran escala, por ejemplo, al medir la lluvia en cincuenta puntos en una finca representa un alto costo de implementación y consumo energético.

Es por esta razón la necesidad de buscar otras soluciones de conectividad, más versátiles y ligeras que sean fáciles de implementar, de bajo consumo energético y de largo alcance.

En este proyecto se utilizará la tecnología LoRaWAN. Es un protocolo de red de área amplia y baja potencia, esto debido a las grandes ventajas con respecto a otras tecnologías en el campo de la agricultura, campo donde se le da un mayor valor a la medición de lluvia.

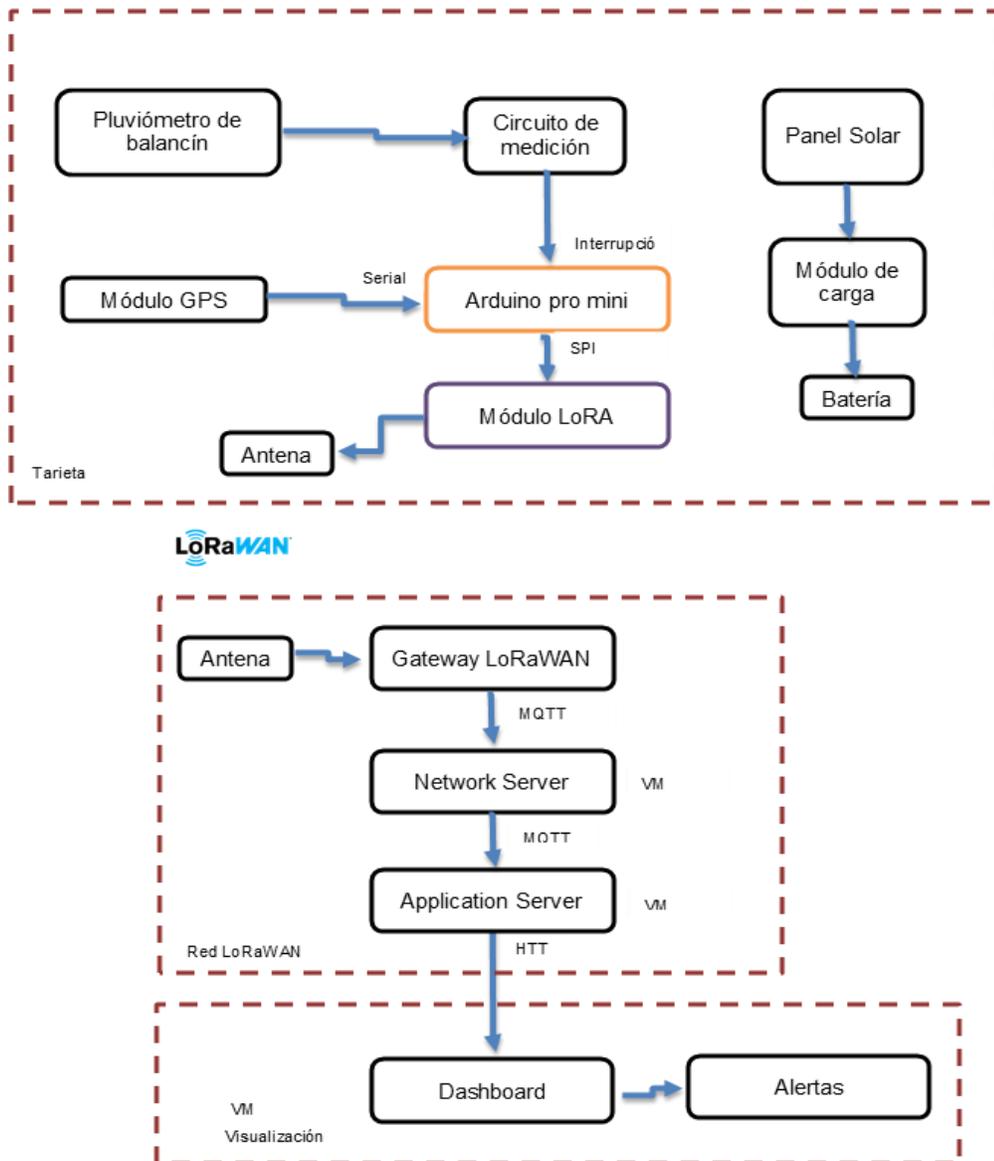
La tecnología que mayor se adapta a las necesidades de conectividad de dispositivos en un entorno rural es LoRaWAN.

Se ampliará con mayor detalle el funcionamiento de esta tecnología en el capítulo tres, pero se deja una breve explicación del funcionamiento general del proyecto mediante un esquema general que a continuación se presenta.

1.3.1. Esquema general

En la figura 12 se describe el esquema general del proyecto.

Figura 12. Esquema general del proyecto



Fuente: elaboración propia, empleando Microsoft Word 365.

Para cumplir el objetivo de tener a la mano siempre la información de lluvia de los pluviómetros son necesarios tres grandes elementos:

Tarjeta electrónica, es la encargada de recibir la señal del sensor del pluviómetro de balancín, procesar señal, adjuntar información de la ubicación y transmitir esa información mediante una modulación de radiofrecuencia LoRa.

Red LoRaWAN, se compone de tres elementos, los cuales son Gateways, Network Server, Application Server. Los Gateways o concentradores son los encargados de recibir las señales moduladas en LoRa de las tarjetas electrónicas conectadas y reenvían los paquetes hacia el servidor de Red, el cual se encarga de verificar duplicidad de paquetes, autenticación y *networking* entre dispositivos y *gateways*.

El servidor de red también envía la información a un servidor de aplicación en el cual se pueden manejar los equipos de la red en una interfaz gráfica y también es el encargado de reenviar la información a algún servicio externo de almacenamiento, *dashboard* u otro.

Para tener la información en una base de datos y visualizar gráficamente los datos se necesita un servicio para el usuario final. Existen muchos servicios de este estilo, en este proyecto se utilizará Thingsboard, servicio que ofrece una versión gratuita.

En los siguientes capítulos se abordarán estos tres pilares del proyecto. Esta solución es sumamente escalable, ya que una red LoRaWAN puede extenderse a un área extensa agregando gateways y dispositivos finales, en este caso en específico se utilizará pluviómetros digitales, pero podría haber muchas más aplicaciones de internet de las cosas.

2. DISEÑO DE TARJETA ELECTRÓNICA

En el capítulo anterior se plasmó un mapa general del proyecto (figura 12) dejando claro qué componentes engloba todo el sistema de medición de lluvia y visualización en tiempo real, desde el sensor de medición hasta la visualización del usuario final.

En este capítulo se hará énfasis en los elementos que componen la tarjeta electrónica o, en el ámbito de internet de las cosas conocidas como nodo o dispositivo final.

2.1. Evaluación de requerimientos

La tarjeta electrónica tiene por objetivo obtener las señales desde el pluviómetro, procesar, adjuntar información de la ubicación y transmitir los datos por medio del protocolo inalámbrico LoRa. Además, para garantizar un sistema completamente autónomo es necesario proveer de un método recurrente de carga, para lo cual se utilizará energía solar.

Teniendo clara las necesidades a cubrir, se iniciará con la parte principal del sistema, el microcontrolador y las características que tendrá. Luego se pasará a obtener datos a transmitir, lluvia y ubicación. Posteriormente se tratará el envío de los datos por radiofrecuencia LoRa.

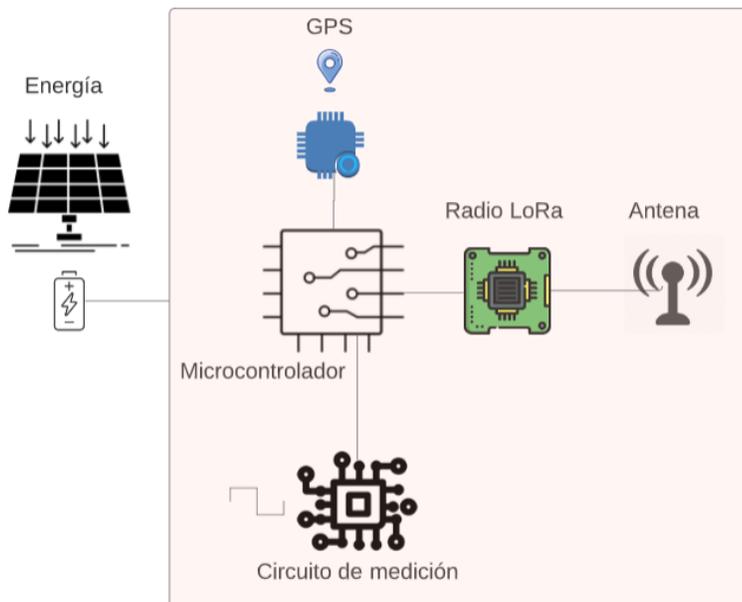
Cuando se tengan todos los componentes unidos se hará un análisis de cuanta energía es necesaria para alimentar los módulos y que especificaciones deben de cumplir los elementos que proveerán de potencia a los circuitos.

Se busca realizar un diseño que sea de bajo costo, fácil de configurar, armar y que sea eficiente energéticamente, para lo cual se buscará elegir los componentes adecuados para el fin.

Asimismo, en este capítulo se construirá el algoritmo con el que funcionará el medidor de lluvia, medición de GPS y transmisión de datos con el protocolo LoRa y se procederá a la construcción de dicha tarjeta electrónica.

Consolidando las necesidades en un gráfico, quedaría de la siguiente manera:

Figura 13. **Esquema de tarjeta electrónica**



Fuente: elaboración propia, empleando Draw.io.

2.2. Microcontrolador

Una idea general de un microcontrolador es: un circuito integrado programable que funciona con una arquitectura que contiene una unidad central de proceso, memorias, interfaces, periféricos, todo un sistema computacional en un solo chip cuyo fin es llevar a cabo procesos lógicos que se configuran en un programa escrito en un lenguaje de alto nivel y llevado posteriormente a instrucciones en lenguaje ensamblador entendibles por el microcontrolador.

Los microcontroladores son utilizados en todos los dispositivos digitales embebidos de hoy en día, por lo que existen gran cantidad modelos que cumplen características especiales para cada aplicación dependiendo de la complejidad de esta.

El papel de este elemento en la tarjeta electrónica para medición de lluvia es fundamental, ya que esta será el cerebro que manejará las señales de entrada y salida hacia los módulos que harán autónoma la medición de la precipitación.

Para elegir el microcontrolador adecuado para la aplicación, es necesario conocer los requerimientos en cuanto a ¿qué módulos del sistema estarán comandados por este? y basados en esto, definir cuantas entradas y salidas digitales, entradas analógicas, interfaces, memoria, frecuencia de reloj, debería tener el micro además considerar el consumo de energía que utiliza para el funcionamiento,

Como se describe en sección 2.1. Evaluación de requerimientos y en la figura 13, se construirá una tarjeta electrónica que tenga la capacidad de medir la cantidad de lluvia, medir la ubicación y transmitir lo datos mediante un chip de

radiofrecuencia LoRa. A continuación, se muestran las características para conectar los módulos requeridos al microcontrolador.

Tabla II. **Interfaces módulos-microcontrolador**

Módulo	Interfaz	Pines	Observación
Circuito de medición lluvia	GPIO	1	Interrupción
GPS	Serial	2	115 200 bps
Radio LoRa	SPI + GPIO	4 SPI+ 4 GPIO	RFM95W
Entrada analógica adicional	ADC	4	

Fuente: elaboración propia, empleando Microsoft Excel 365.

En las siguientes secciones se profundizará más en cada uno de los elementos del microcontrolador que servirán para hacer funcionar todo el sistema embebido.

2.2.1. Microcontrolador ATMEGA328P

El chip ATMEGA328P es un microcontrolador de la familia lógica CMOS (*Complementary Metal Oxide Semiconductor*), construido con transistores de efecto de campo MOSFET que en pares forman los transistores nMOS y pMOS los cuales tienen la ventaja de tener alta impedancia en su puerta de entrada y que estando en inactividad solamente tienen consumo energético por corrientes basuras.

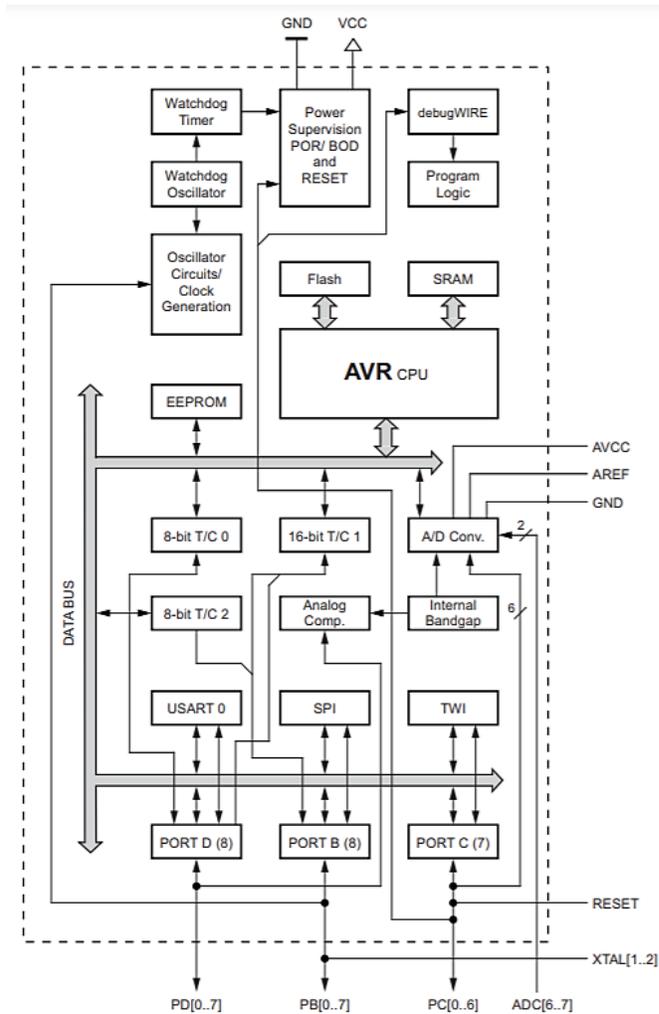
Esta es la principal ventaja para utilizarse en microprocesadores, memorias, y diversidad de circuitos integrados incluyendo los que integran dispositivos de internet de las cosas. (*SetUp*)

Este microcontrolador tiene un bus de 8 bits y forma parte de la familia AVR, la cual inicialmente fue diseñada para optimizar código en C, razón por la cual las placas Arduino se basan en esta familia. El chip tiene una arquitectura Harvard, esta composición separa los buses de instrucciones que se almacenan en memoria *flash* y datos provenientes de la memoria SRAM, esto para ejecutar instrucciones en cada ciclo de reloj empleando el concepto de *pipelining* que garantiza que cuando una instrucción se está ejecutando, la siguiente se está preparando para ser efectuada.

Es importante mencionar que este microcontrolador obedece 131 instrucciones RISC de las cuales la mayoría pueden ser ejecutadas en un ciclo de reloj cada una. Adicionalmente el microcontrolador contiene 32 registros de propósito general de 8 bits cada uno, los cuales también se pueden manejar con un solo ciclo de reloj, permitiendo que operaciones aritméticas en la ALU puedan realizarse en un pulso de reloj. Seis de estos registros pueden ser utilizados como 3 registros de direcciones de 16 bits, lo cual hace que se pueda computar operaciones entre direcciones.

A continuación, se muestra la arquitectura del microcontrolador ATMEGA328P.

Figura 14. **Arquitectura AVR ATMEGA328P**



Fuente: Atmel. *8-bit AVR Microcontroller with 32K Bytes In-System.*

https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf. Consulta: 1 de abril de 2020.

Observando la arquitectura del micro puede notarse que es un sistema muy optimizado que engloba lo necesario para hacer funcionar la tarjeta de medición de lluvia. A continuación, se expondrá lo más importante de cada elemento que conforma el microcontrolador.

2.2.1.1. Memoria Flash

Este tipo de memoria es no volátil, es decir que no pierde información al ser desenergizada y es ideal para almacenar registros de programa de instrucciones debido a la velocidad de lectura y la tolerancia de ciclos de escritura y borrado (10 000 ciclos en el ATMEGA328P), los cuales son característicos en actualizaciones de *firmware* en dispositivos electrónicos.

El microcontrolador ATMEGA328P integra una memoria *flash* de 32Kbytes para el almacenamiento del programa, cuyas instrucciones son de 16 o 32 bits. Esta memoria se divide en dos partes, una para almacenar el *bootloader* y otra para el algoritmo de aplicación. El *bootloader* es un software que ejecuta el microcontrolador para recibir nuevas instrucciones y reprogramar la sección de aplicación en donde se almacena el conjunto de órdenes para el manejo de periféricos, memorias, entradas y salidas del chip.⁵

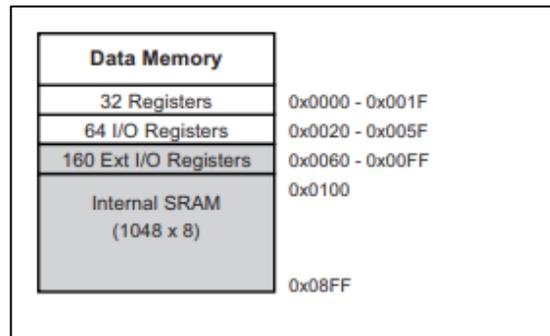
2.2.1.2. Memoria SRAM

La memoria SRAM (Static Random Access Memory) Es una memoria volátil que se caracteriza por la rapidez de lectoescritura en las direcciones. Almacena datos temporales de variables locales y es el segmento del microcontrolador donde el programa principal establece y maneja las variables cuando están en ejecución.

El microcontrolador ATMEGA328P contempla 2304 localidades de memoria de 8bits, estas localidades se dividen en cuatro segmentos dentro de la memoria, los cuales sirven para almacenar registros específicos.

⁵ GÜNTHER GRIDLING, Bettina Weiss. *Introduction to Microcontrollers*. <https://www.skylineuniversity.ac.ae/pdf/software-engineering/Microcontroller.pdf>.

Figura 15. **Memoria SRAM ATMEGA328P**



Fuente: Atmel. *8-bit AVR Microcontroller with 32K Bytes In-System*.

https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf. Consulta: 1 de abril de 2020.

La división de memoria está dada por 32 registros de acceso rápido y son utilizados como de propósito general, 64 registros I/O sirven para la configuración y control de recursos del propio microcontrolador, como banderas de estados y manejo de puertos de entrada y salida, 160 registros I/O extendido tienen la misma función que los registros I/O con variaciones en el tipo de instrucciones que pueden utilizarlos. Los 2048 registros restantes se utilizan para almacenar las variables y datos que ejecuten el programa principal.⁶

2.2.1.3. Memoria EEPROM

Esta memoria es no volátil y se utiliza para almacenar valores que se desean conservar después de un reinicio, estas tienen un número limitado de lectura/escritura para el ATMEGA328P es de 100,000 ciclos y solo es posible

⁶ RODRÍGUEZ SANCHEZ, Cristina; VAQUERO LÓPEZ, Joaquín. *Domótica y sistemas de embebidos para comunicaciones inalámbricas*. <https://pdfslide.tips/documents/tema-3-sistemas-embebidos-pic.html>.

realizar acciones por bytes individualmente. El tamaño de la EEPROM en el chip es de 1Kbyte.

2.2.1.4. Interfaz USART

Los microcontroladores AVR incluyen, por lo regular, uno o más bloques de *hardware* de interfaces seriales USART (*Universal Synchronous Asynchronous Serial Receiver and Transmitter*), este protocolo de comunicación por cable permite transmitir información de un módulo a otro de forma serial empleando dos cables: uno para recibir RX y otro para transmitir TX en cada dispositivo, lo cual representa que la comunicación puede ser *full duplex*.

Una comunicación serial representa que los datos serán enviados bit a bit separados por un período de tiempo. El protocolo empaqueta los datos en tramas que incluyen un bit de inicio, parada, paridad y el *payload* de 8 bits.

La velocidad de transmisión o baudrate depende del reloj del bloque USART y puede ser configurado en el microcontrolador para adecuarse a las velocidades del agente externo con el que se da la comunicación. El microcontrolador ATMEGA328P tiene una interfaz de este tipo.

2.2.1.4.1. Software serial

La única interfaz de comunicación del microcontrolador ATMEGA328P por lo regular es utilizada para realizar la comunicación hacia una computadora por medio de un conversor a USB, esto con el fin de trasladar el programa del chip y visualizar información importante en el ordenador cuando se están desarrollando sistemas embebidos.

En caso de tener la necesidad de conectar más módulos seriales al microcontrolador se realiza una réplica del bloque USART por medio de software, lo cual permite tener más pines del microcontrolador dedicados a esta comunicación. Software Serial es una librería construida para realizar esta acción, pero tiene algunas limitantes en comparación al bloque de *hardware*, como en el caso de las múltiples interfaces que no pueden recibir datos al mismo tiempo, solo es posible configurar pines con interrupciones.

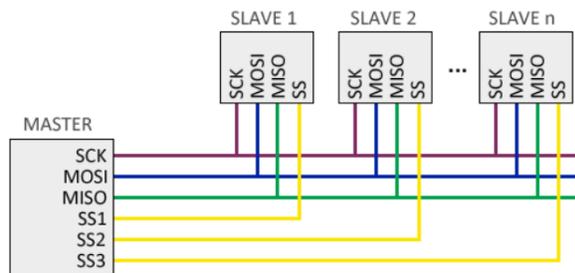
Para el Arduino pro-mini, los pines que pueden ser utilizados como RX con Software serial son los digitales 2 y 3.

2.2.1.5. Interfaz SPI

El microcontrolador ATMEGA328P envuelve un bloque de hardware para comunicación SPI (*Serial Peripheral Interface*) el cual sigue un protocolo de comunicación síncrona basado en un esquema maestro-esclavos. Decir que es síncrona representa que los dispositivos a comunicar tendrán una misma señal de reloj, la cual es manejada por el maestro.

El estándar es muy utilizado en electrónica y automatización para comunicaciones de corto alcance y por lo regular en tarjetas electrónicas para el manejo de varios módulos por una sola interfaz del microcontrolador.

Figura 16. **Arquitectura SPI**



Fuente: LLAMAS, Luis. *El bus SPI en Arduino*. <https://www.luisllamas.es/arduino-spi/>.

Consulta: 1 de abril de 2020.

Su funcionamiento se da con 4 cables, SCK, MOSI, MISO, SS siendo la señal de reloj, transmisión de maestro a esclavo, transmisión de esclavo a maestro y selección de esclavo respectivamente. El principio de comunicación es sencillo, el maestro controla la señal de reloj con SCK y selecciona con que esclavo se comunicará con SS, MOSI y MISO transmiten datos full duplex entre maestro-esclavo.⁷

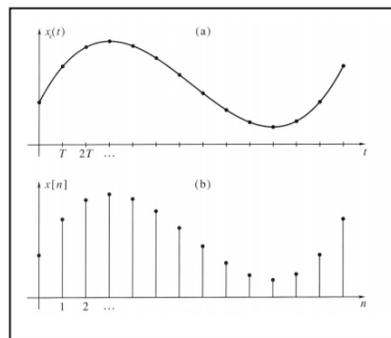
2.2.1.6. **Unidad ADC**

Las magnitudes físicas del mundo real no son señales eléctricas, sino se representan con otros tipos de energía. Variables como la temperatura, humedad, presión o luminancia son propiedades de la naturaleza que para ser entendidas por un aparato electrónico tienen que ser traducidos a su lenguaje binario.

⁷ LLAMAS, Luis. *El bus SPI en Arduino*. <https://www.luisllamas.es/arduino-spi/>.

Este proceso de traducción inicia con un sensor que es capaz de convertir alguna representación de energía a una señal eléctrica, pero estos valores aún están dados por la naturaleza y son analógicos, es ahí donde entra en juego el papel de las unidades ADC (*Analog to Digital Converter*) en los microcontroladores, su función es convertir estas señales continuas en valores discretos y con esto entender lo que sucede en el entorno físico.

Figura 17. **Muestreo señal analógica**



Fuente: SORIA, Emilio. *Introducción al procesamiento digital de señales. Conversión A/D, D/A.*
https://www.uv.es/soriae/tema_1_pds.pdf. Consulta: 2 de abril de 2020.

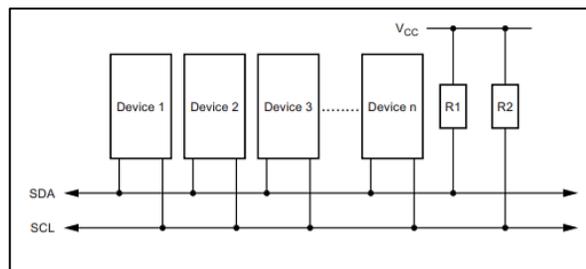
La unidad ADC recibe una señal continua en función del tiempo $x(t)$ la cual es muestreada $x[n]$ recabando información de las magnitudes, posteriormente se realiza una cuantización de la señal lo cual es definir niveles o rangos para clasificar las magnitudes muestreadas y asignarles una codificación binaria.

El microcontrolador ATMEGA328P integra una unidad ADC con entrada por medio de un multiplexor para 8 pines analógicos. La resolución es de 10 bits, esto representa que la señal de entrada será codificada en 1023 valores distintos dependiendo de la magnitud.

2.2.1.7. Interfaz TWI

El protocolo TWI (*Two Wires Interface*) también conocido como I2C es un estándar de comunicación por cable síncrono de poca distancia y es utilizado para comunicar módulos en una tarjeta electrónica por medio de 2 cables SCL y SDA uno para reloj y uno para datos.

Figura 18. Arquitectura protocolo TWI



Fuente: Atmel. *8-bit AVR Microcontroller with 32K Bytes In-System*.

https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf. Consulta: 1 de abril de 2020.

En su arquitectura existe un dispositivo maestro y varios esclavos, todos tienen una dirección y pueden conectarse hasta 128. La comunicación es *half-duplex* y la operatividad es manejada por el maestro, el cual decide si enviar o recibir y hacia qué dirección, el esclavo está sujeto a la petición del maestro y no puede iniciar la comunicación ni comunicarse directamente con otros esclavos.

Es necesario conectar dos resistencias *pull-up* en el bus debido a la configuración de hardware interna de la interfaz.

2.2.1.8. Puertos GPIO

Los microcontroladores, como bien se ha mencionado, recaban información de su entorno y tienen también la capacidad de controlar actuadores y enviar y recibir señales a otros componentes o circuitos. Estas acciones se realizan por medio de los puertos GPIO por sus siglas en inglés (*General Purpose Input Output*) son pines que pueden ser utilizados para recibir o enviar una señal digital y pueden ser programados como pines de entrada o salida dependiendo de las necesidades que se tengan.

La manipulación de estos puertos permite a los sistemas embebidos crear sistemas de control con señales de entrada que al ser interpretados por la CPU llevan a cabo acciones por medio de un actuador como un motor, indicador u otro equipo que traduzca una señal digital a otro tipo de energía.

El microcontrolador ATMEGA328P provee de 3 puertos de 8 bits cada uno, poniendo a disposición 24 pines de propósito general que muchos de ellos son entradas y salidas de las interfaces de comunicación por lo que al realizar un diseño se tiene que considerar qué conexiones tendrá el microcontrolador.

2.2.1.9. Interrupciones

Como bien se definió que un microcontrolador es un sistema en un chip que integra una unidad central de proceso y que está conducido por un programa con una serie de instrucciones que se ejecutan de manera secuencial (instrucción por instrucción).

En determinadas circunstancias surge la necesidad que la CPU dé prioridad a otras instrucciones que se dan a raíz de un evento que no se puede saber

cuándo pasará y no es posible esperar a que la secuencia del programa llegue a las instrucciones que deben ser atendidas de forma urgente.

Debido a esto los microcontroladores integran una unidad de interrupciones y son un mecanismo para responder a sucesos de bloques internos o externos de forma urgente, permitiendo detener temporalmente el programa principal y volver a la instrucción donde fue interrumpido luego de completar la subrutina de interrupción.

Generalmente estas interrupciones son generadas por las interfaces de comunicación que se trata anteriormente como SPI, TWI, USART cuando reciben una solicitud de recepción de datos de otros dispositivos. También existen interrupciones dadas por los TIMMERS, los cuales son segmentos del microcontrolador que sirven para realizar conteos de tiempo y pueden llegar a desbordarse o producir otro evento generando una interrupción. Existen también las interrupciones externas que están directamente ligados a los GPIO del chip los cuales pueden generar una interrupción al cambiar de estado lógico.

Esta herramienta es muy útil en aplicaciones donde se tiene que monitorear el estado de un pin de manera continua si perder un solo milisegundo de vista como, por ejemplo: al medir las revoluciones de un motor se tiene que realizar el conteo sin perder ninguna señal, para un medidor de frecuencias o la lectura del sensor del pluviómetro de balancín.⁸

⁸ REYES CORTÉS, Fernando; CID MONJARAZ, Jaime. *Arduino. Aplicaciones en robótica, mecatrónica e ingenierías*. p. 122.

Tabla III. **Interrupciones externas ATMEGA328P**

Vector No.	Localidad	Fuente	Característica	Identificador
1	0x00	RESET	Pin externo, reset al encender, apagón parcial, reset por watchdog	
2	0x02	INT0	Interrupción externa pin 2	INT0_vect
3	0x04	INT1	Interrupción externa pin 3	INT1_vect
4	0x06	PCINT0	Interrupción por cambio pines 8 al 13	PCINT0_vect
5	0x08	PCINT1	Interrupción por cambio pines A0 a A5	PCINT1_vect
6	0x00A	PCINT2	Interrupción por cambio pines 0 a 7	PCINT2_vect

Fuente: elaboración propia, empleando Microsoft Excel 365.

El microcontrolador ATMEGA328P tiene 6 interrupciones externas, la primera es utilizada para realizar un RESET del microcontrolador, INT0 e INT1 están directamente ligadas a los pines 2 y 3 del chip y se activan al haber un cambio de estado lógico en estos, con posibilidad de ser RISING, FALLING, LOW, CHANGE.

Los PCINT a diferencia de los INT se manejan en conjunto y solo pueden configurarse como CHANGE, es decir que si uno de los pines que integra el PCINT0 cambia de estado lógico se activa la interrupción.

Tabla IV. **Activación de interrupciones**

Evento	Descripción
RISING	Se da cuando el estado lógico pasa de 0 a 1
FALLING	Se da cuando el estado lógico pasa de 1 a 0

Continuación de la tabla IV.

LOW	Indica que se activa cuando tiene el estado lógico 0
CHANGE	Se da cuando el estado lógico cambia

Fuente: elaboración propia, empleando Microsoft Excel 365.

2.2.2. Arduino Pro Mini

Arduino consiste en una placa de circuito programable y un software instalado en una computadora (IDE) en donde se escriben códigos que le dirán al microcontrolador dentro de la placa como responder a las señales de entrada y salida, estas instrucciones se escriben con lenguaje C que serán compiladas y ejecutadas por el microcontrolador que contiene la placa física.⁹ Esta plataforma (placa y software) es Open-Source, lo cual representa que puede ser utilizada y modificada por cualquiera que desee hacerlo, sin costo.

La principal ventaja que ofrece esta plataforma es la facilidad de realizar un proyecto integrando distintos módulos y sensores con librerías desarrolladas por la comunidad.¹⁰ Por ejemplo, en un prototipo que involucre varios sensores con diferentes protocolos de comunicación, almacenamiento en memorias internas y externas, sin utilizar librerías el desarrollo sería muy complejo porque inicialmente se tendría que definir procedimientos de muy bajo nivel entendiendo los protocolos, registros, tiempos y secuencias. Arduino facilita esta tarea con el volumen de desarrollo previo que se tiene, concentrándose solamente en las funciones que hace cada módulo y llegar al objetivo de forma más rápida.

⁹ B.E.N. What is an Arduino? <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>.

¹⁰ ISHIKAWA, Masato; MARUTA, Ichiro. *Rapid Prototyping for Control Education using Arduino and Open-Source Technologies*.
<https://www.sciencedirect.com/science/article/pii/S1474667015316396>.

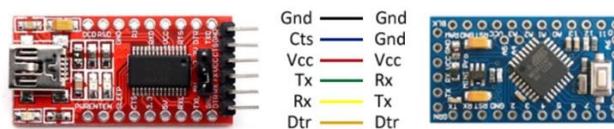
Un Arduino Pro mini es una de las versiones más pequeñas de Arduino y está pensado para aplicaciones de funcionamiento permanente ideal para aplicaciones de internet de las cosas. Contiene un microcontrolador ATMEGA328P y puede ser en versión de 3.3 V a 8 MHz o 5 V a 16 MHz. Cabe mencionar que este Arduino no incluye un puerto USB como otras versiones, para lo cual es necesario programarlo empleando convertor de interfaz serial a USB, pero debido a que el dispositivo de medición de lluvia será instalado en campo y programado una sola vez, no es necesario dejar el módulo convertor permanente.

Esta placa añade unos componentes conectados al microcontrolador ATMEGA328P para facilitar la fácil implementación de los proyectos, tal es el caso de un circuito regulador de voltaje, que permite alimentar la placa hasta con 12 V. También hay resistencias para la interfaz TWI, un botón de *reset*, cristal oscilador y leds indicadores de energía y otro al pin 13 para pruebas.

Los requerimientos de este proyecto indican que se busca realizar un diseño minimalista, de bajo costo y con bajo consumo de energía por lo tanto se utilizará el Arduino pro mini para el diseño y aprovechar las librerías existentes para el manejo de módulos GPS, radio LoRa y memorias reduciendo el tiempo de implementación.

Los pines pueden ser configurados como interfaz de comunicación o GPIO dependiendo de la necesidad y la interfaz USART del microcontrolador tiene pines en un extremo para realizar la comunicación con la computadora por medio de un conversor serial TTL a USB de la siguiente forma. Cabe mencionar que antes de utilizar el conversor deben estar instalados los *drivers* en el ordenador y tener instalado el IDE de arduino.

Figura 21. **Conexión para programador FTDI**



Fuente: LLAMAS, Luis. *Programar Arduino mini pro con conversor FTDI FT232R*
<https://www.luisllamas.es/programar-arduino-mini-pro-conversor-ftdi-ft232r/>. Consulta: 2 de mayo de 2020.

2.2.2.2. **Arduino IDE**

Arduino IDE es un *software open-source* y constituye un entorno completo de programación de *hardware* que incluye editor de código, compilador, depurador y las herramientas necesarias para cargar los programas directamente a las tarjetas arduino. Todo esto bajo un interfaz gráfica que hace muy fácil la utilización de los recursos y librerías.

El editor de código maneja el lenguaje C y C++, lo cual se pensaría que es complicado pero la dificultad es muy poca porque han añadido muchas funciones específicas, sentencias muy cortas y pocos atributos para los procedimientos. La interfaz además subraya muy bien las palabras reservadas para que el programador se concentre solamente en el algoritmo del proyecto.

Los proyectos de arduino se trabajan en un solo archivo llamado sketch y es un fichero con terminación “.ino”, por lo tanto la estructura y el inicio de un proyecto se simplifica solamente a tener instalado el paquete de arduino y creando una hoja nueva.

La estructura de un sketch tiene dos partes principales, el *void setup()* y *void loop()*. La primera es un procedimiento para realizar cualquier configuración inicial de interfaces, variables o librerías, esta solamente se ejecuta una vez al energizar la placa y la segunda es el bucle infinito donde estarán las sentencias y algoritmo del programa. Además pueden crearse variables, procedimientos y funciones globales fuera de estas dos partes.

Otras bondades que aporta al programar el arduino IDE es la versatilidad entre tarjetas arduino y hasta otras plataformas de *hardware* que utilizan el software para programar, también detecta automáticamente la conexión del dispositivo al ordenador, permite visualizar el espacio en la memoria flash y sram, autoguardado al compilar, cuenta con un monitor serial para escribir y leer datos de la tarjeta, también contiene un serial plotter que sirve para visualizar el comportamiento de alguna señal o variable analógica de manera gráfica.

Además de esto, el arduino IDE cuenta con un manejador de librerías desarrolladas por la comunidad a la que uno puede acceder y utilizar ejemplos proveídos por el autor. No solamente se puede utilizar estas librerías, en internet la comunidad es mucho mayor, compartiendo, depurando y mejorando el código.

2.3. Circuito de medición de lluvia

En el primer capítulo se abarcó el tema del pluviómetro de balancín, su funcionamiento y las partes que lo componen, detallando la metodología para

obtener pulsos por medio de un sensor de tipo *reed switch* cuando se detecta un evento de precipitación. Sin embargo, aún no se ha tratado el asunto de convertir ese acoplamiento magnético del *reed switch* a una señal digital entendible por el microcontrolador.

Este proceso de conexión sensor-microcontrolador debe tratarse con mucho análisis debido a las condiciones de la aplicación que se está construyendo. Existen lluvias muy torrenciales que harán que se genere una serie de pulsos con una frecuencia alta, también es posible que el balancín al volcarse rebote en la base produciendo micro pulsos que no deben ser considerados como una medición de lluvia adicional.

Además del rebote ocasionado por el golpeo mecánico del balancín, existe el producido propiamente por el *reed switch*, se recuerda que este sensor une dos placas ferromagnéticas a razón de un campo magnético externo producido por un imán permanente en el balancín, el cual tiene un movimiento angular cuando se produce un volteo. Esto como consecuencia puede producir corrientes muy pequeñas entre los contactos del sensor.

2.3.1. Circuito anti-rebote

Como bien se ha dicho, el efecto de rebote se produce cuando se utiliza un transductor mecánico o electromecánico como sensor, el efecto de cerrar o abrir el circuito no se produce en un solo movimiento, siempre hay movimiento mecánico que hace que las señales no sean impulsos puros y estén acompañados de ruido.

Para solucionar este problema se pueden implementar soluciones de *hardware* y de *software* siendo algunas técnicas más complejas que otras y deben seleccionarse dependiendo de las características de la señal de entrada.

Estos parámetros son la frecuencia, el tiempo en alto de los pulsos, el tipo de sensor y que tanto rebote produce. Por ejemplo, si la señal de entrada debe sus impulsos a las revoluciones de un motor, la frecuencia será muy alta y los pulsos muy cortos o en un pulsador para aumentar el volumen de algún aparato de sonido, la frecuencia de los pulsos es grande y una persona deja presionado el botón por un tiempo largo en comparación al ejemplo anterior.

Por lo tanto, la selección de la técnica antirrebote depende de la aplicación a utilizar y por su puesto del sensor. A continuación, se analizará las características del medidor de lluvia con sensor *reed switch*.

Ahora se analizará la frecuencia máxima que puede presentarse de pulsos, para lo cual se consultará el récord histórico de medición de lluvia, y según un artículo publicado por la Universidad de Arizona de Estados Unidos con datos de la Organización Meteorológica Mundial.¹¹

Existe un récord histórico de 31,2 mm en un minuto, suponiendo que la intensidad de lluvia fue constante y que la medición hubiera sido realizada por el pluviómetro de balancín de 0,08 mm por pulso. La frecuencia sería la siguiente:

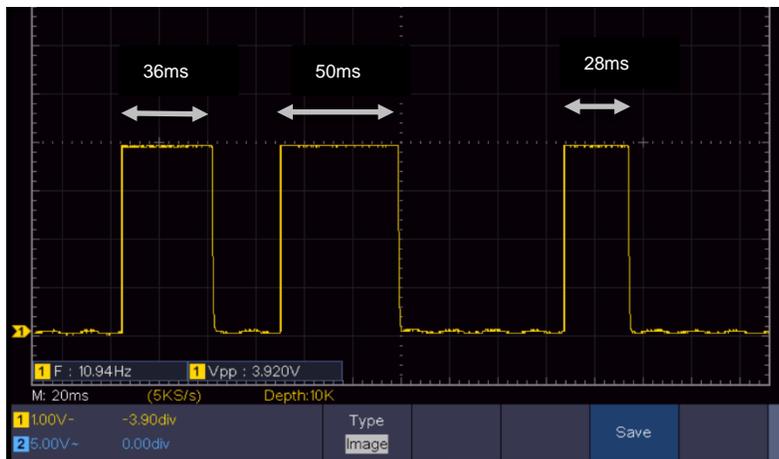
¹¹ Universidad Estatal de Arizona. *WMO Rapporteur of Weather & Climate Extremes*. <https://wmo.asu.edu/content/world-meteorological-organization-global-weather-climate-extremes-archive>.

Ecuación 4:

$$f = \frac{31,2mm}{min} * \frac{1 pulso}{0,08mm} * \frac{1 min}{3 600s} \approx 6,5 Hz^{12}$$

Según una medición de los pulsos del pluviómetro de balancín, realizada con un osciloscopio, la duración del pulso cuando hay un volteo es en promedio de 38 ms como se muestra en la fotografía.

Figura 22. **Ancho de pulsos de pluviómetro de balancín**

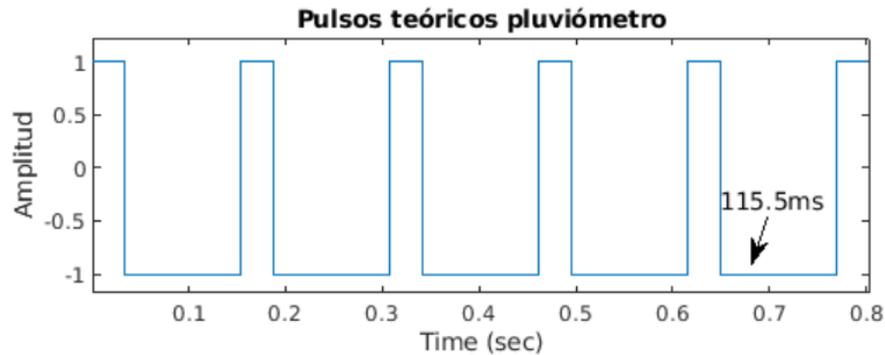


Fuente: elaboración propia, empleando Osciloscopio.

Teniendo esa información se puede realizar un bosquejo de la señal teórica de esa medición extrema, la cual sería una onda cuadrada con una frecuencia de 6,5 Hz, período de 154 ms, con un ciclo de trabajo de un 25 %, es decir 38 ms de pulso en alto, la que quedaría de la siguiente forma:

¹² Universidad Estatal de Arizona. *WMO Rapporteur of Weather & Climate Extremes*. <https://wmo.asu.edu/content/world-meteorological-organization-global-weather-climate-extremes-archive>.

Figura 23. Pulsos teóricos de récord de lluvia



Fuente: elaboración propia, empleando Matlab.

El dato importante que se obtiene de este análisis es que, si existen pulsos separados por tiempo menor a 115,5 ms debería interpretarse como una señal de ruido o rebote, debido a que, según el récord histórico de lluvia, los pulsos del pluviómetro no alcanzarían una frecuencia mayor.

Ahora se resuelve este problema, y existen 2 métodos usualmente aplicados a esta situación. El primero es aplicar una pausa al programa después de haber leído el primer pulso, con el tiempo necesario para que el microcontrolador no lea señales de ruido en la duración de la pausa.

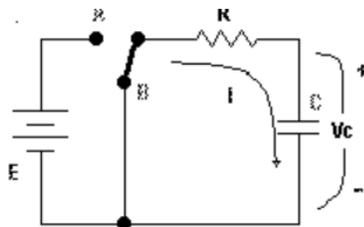
El otro método intenta retardar el tiempo en alto de los pulsos para que las señales de rebote queden en ese retardo y no sean interpretadas por el microcontrolador como pulsos adicionales. Este tratamiento se realiza por medio de un circuito RC aprovechando el tiempo de descarga de un capacitor.¹³

¹³ BOEMO, Eduardo. *Aspectos eléctricos del diseño digital*. p. 2.

En esta tarjeta electrónica se aplicará el segundo método, en virtud de que no es nada recomendable pausar el programa, porque se dejan de ejecutar otras tareas que también son importantes para el funcionamiento del sistema.

El proceso de descarga de un capacitor ocurre cuando éste tiene un camino de desagüe de corriente, y el tiempo de descarga está en función de la resistencia y la capacitancia.

Figura 24. **Circuito RC**



Fuente: Electrónica Fácil. Carga y descarga de un condensador.

<https://www.electronicafacil.net/tutoriales/Carga-descarga-condensador.html>. Consulta: 2 de mayo de 2020.

El tiempo de descarga del capacitor está dado por la constante de tiempo, y se estima que cinco veces este valor representa la descarga en un 99,3 % por lo tanto se dice que el capacitor esta descargado cuando se cumplieron cinco constantes de tiempo.

Ecuación 5:

$$Td = 5\tau = 5 * R * C [s]$$

En la aplicación se necesita un retardo de 115,5 ms, por lo tanto, se realiza el cálculo de los valores de resistencia y capacitancia para lograr este tiempo de descarga. Se coloca un valor de 0,1 μF para el capacitor, dado que es un valor comercial y nos quedaría calcular la resistencia.

Ecuación 6:

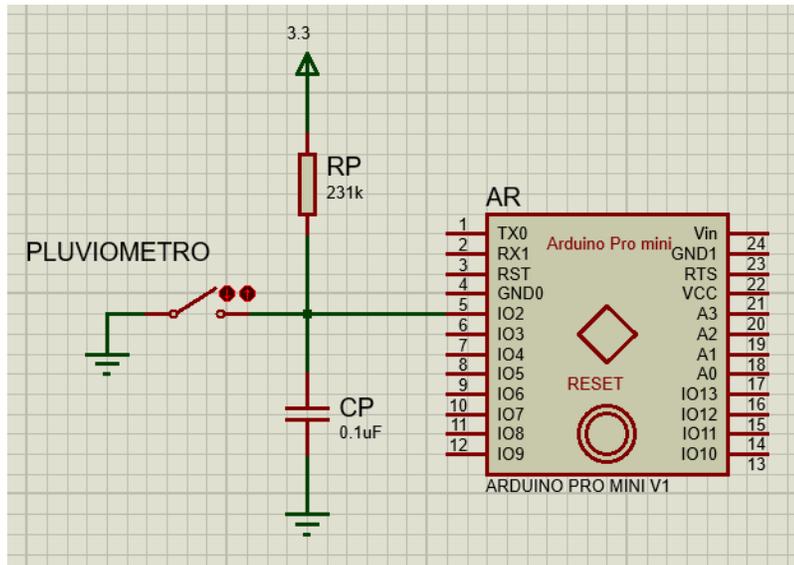
$$R = \frac{Td}{5 * C} = 231 \text{ k}\Omega$$

2.3.2. Conexión al microcontrolador

Para la conexión del circuito antirrebote al microcontrolador se utiliza un puerto GPIO con interrupción, debido a que el evento de lluvia es aleatorio y el programa siempre debe estar disponible para realizar la medición. El Arduino pro mini tiene el pin 2 y 3 como interrupción y se utilizará para la medición de lluvia el pin 2.

Además, otra técnica para evitar las lecturas incorrectas por interferencias electromagnéticas es colocar el circuito en *pull-up*, es decir que la señal permanecerá en estado alto y cuando el sensor se active, la señal se pondrá en estado bajo. El microcontrolador detectará esta transición más conocida como FALLING y es posible configurarlo en la interrupción. El circuito y conexión se muestran a continuación.

Figura 25. **Circuito de medición de lluvia**



Fuente: elaboración propia, empleando Proteus.

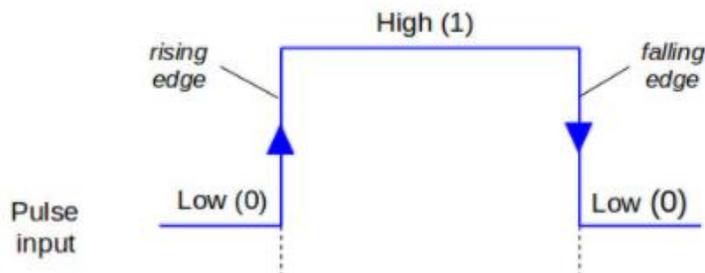
2.3.3. Programación de interrupciones

Como bien se ha dicho, la plataforma de Arduino facilita mucho la programación de interfaces del microcontrolador. Tal es el caso de las interrupciones externas, las cuales pueden ser definidas en una sola línea de código y enviar la secuencia del programa hacia un procedimiento prioritario. La sentencia de declaración de interrupción es `attachInterrupt(#interrupt, ISR, mode)`.

Donde `#interrupt` es el número de interrupción a configurar, `ISR` es el procedimiento a donde se dirigirá la secuencia del programa al ser activada la interrupción y `mode` es el criterio de estado de la señal para determinar cuándo se ingresa a la interrupción. Estos estados son llamados `signal edges` y son `LOW`,

RISING, HIGH, FALLING y existe otro también que representa al RISING y FALLING al mismo tiempo, su nombre es CHANGE.

Figura 26. **Signal Edge**



Fuente: Sci lab. *Signal edge detection*. <https://www.scilab.org/signal-edge-detection>. Consulta: 5 de mayo de 2020.

En este caso el objetivo es capturar el cambio de estado del interruptor del pluviómetro, y como bien se define en el circuito de medición, se detectarán los cambios a estado bajo, por lo cual, la interrupción debe ser configurada en modo FALLING.

2.4. **GPS**

El Sistema de Posicionamiento Global, por sus siglas en inglés GPS, es una tecnología inicialmente desarrollada para aplicaciones militares para realizar mediciones de la posición, velocidad y tiempo de un sistema electrónico. Hoy en día esta técnica es utilizada por cualquier ciudadano en el mundo con un dispositivo celular y ahora con un incremento de dispositivos de internet de las cosas para monitorear activos, su uso está siendo explotado para toda aplicación.

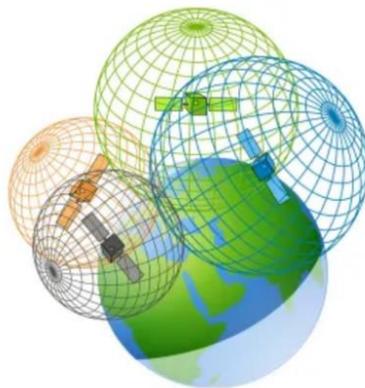
2.4.1. Funcionamiento de GPS

El GPS utiliza una técnica llamada trilateración, que consiste en saber la posición de un punto en función de la distancia a por lo menos 4 puntos de referencia. Las referencias son 30 satélites orbitando la tierra en órbitas de 20 000 kilómetros de altura y completan una vuelta a la tierra en un período de 12 horas. Los satélites son manejados por estaciones en tierra que corrigen la posición y su tiempo es manejado por relojes atómicos muy precisos.

Para conocer la posición de un determinado dispositivo, el mismo espera a recibir señal con la ubicación y tiempo de por lo menos 4 satélites, esto para calcular la distancia desde su punto a cada aparato espacial, aprovechándose del valor conocido de la velocidad de la luz. Con esa información realiza la trilateración y obtiene un valor de latitud y longitud.

La precisión de los valores de ubicación varía en función de la cantidad de satélites que tenga a la vista el dispositivo. Con mayor cantidad de referencias se logra una mejor medición que puede ser desde metros a unos pocos centímetros.

Figura 27. **Trilateración**



Fuente: ArcGeek. *¿Cómo funcionan los dispositivos GPS?* <https://acolita.com/como-funcionan-los-dispositivos-gps-trilateracion-vs-triangulacion/>. Consulta: 1 de junio de 2020.

2.4.2. Módulo GPS NEO-6MV2

El chip NEO-6M forma parte de la familia de circuitos integrados GPS de la marca U-BLOX. Este chip está integrado en el módulo NEO-6MV2, tarjeta que combina flexibilidad de integración mediante un protocolo serial y alta eficiencia energética además de ser un dispositivo compacto para aplicaciones en aparatos móviles.

Figura 28. Módulo GPS NEO6MV2



Fuente: La Electrónica. *Módulo GPS NEO6MV2*. <https://laelectronica.com.gt/modulo-gps-gy-neo6mv2>. Consulta: 2 de junio de 2020.

El módulo incluye una antena cerámica, una memoria EEPROM y un circuito RTC para hacer un inicio rápido y un regulador de voltaje.

Tabla V. Especificaciones NEOG6MV2

Voltaje alimentación	3V a 5V
Tipo de antena	Cerámica
Interfaz	UART 9600 bps por defecto

Continuación de la tabla V.

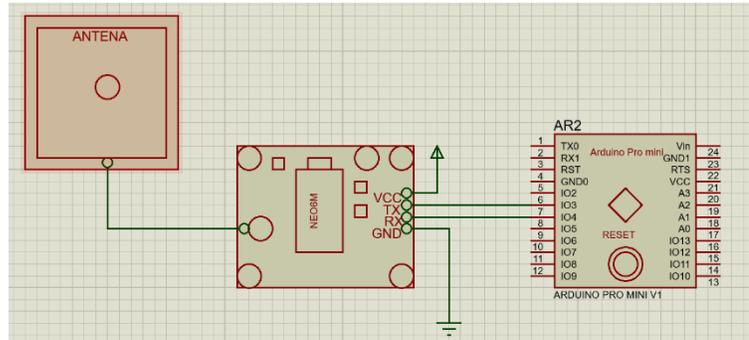
Corriente en modo continuo	37 mA
Precisión	2,5 m
Refresco de posición	5 Hz max
Tamaño de módulo	23 mmx30mm
Tamaño de antena	24 mmx24mm
Longitud cable antena	5 cm

Fuente: Synacorp. *Arduino GY-NEO6MV2 GPS Module c/w Antenna & Flight Control EEPROM*.
<https://www.epitran.it/ebayDrive/datasheet/NEO6MV2.pdf>. Consulta: 3 de junio de 2020.

2.4.3. Circuito de conexión módulo GPS

Para la conexión del módulo a la tarjeta controladora del pluviómetro se necesita una interfaz serial, debido a que así funciona el módulo. El Arduino pro-mini cuenta con uno de estos puertos, pero, es utilizado para la programación del chip cuando se conecta a una computadora. Para no interferir con esta operación y dejarla disponible siempre, se utilizan pines digitales para simular este bloque de *hardware* UART mediante la librería *software* serial mencionada anteriormente.

Figura 29. **Conexión GPS**



Fuente: elaboración propia, empleando Proteus.

Se utiliza el pin 3 como RX y el pin 4 como TX, cabe mencionar que la conexión es cruzada, es decir el RX módulo va al TX Arduino y TX módulo al RX Arduino.

2.4.4. Programación de GPS

Para programar el módulo NEO6MV2, inicialmente se necesita configurar una interfaz serial para la comunicación, dado que la interfaz del Arduino pro-mini se utiliza solamente para la comunicación con la computadora, se hace uso de la librería SoftwareSerial y se configura el pin 3 como RX y pin 4 como TX.

El módulo envía datos en formato NMEA (National Marine Electronics Association) por sus siglas en inglés y tiene la siguiente estructura:

\$GPRMC,hhmmss.ss,A,llll.ll,a,yyyy.yy,a,x.x,x.x,ddmmyy,x.x,a*hh

Tabla VI. **Trama NMEA de GPS NEO6MV2**

\$GPRMC	Identificador
hhmmss.ss	Hora UTC
A	Estado receptor (A = OK, V = warning)
llll.ll,a	Latitud (a = N o S)
yyyy.yy,a	Longitud (a = E o W)
x.x	Velocidad en nudos in knots
x.x	Curso en grados
ddmmyy	Fecha UT
x.x,a	variación magnética en grados (a = E o W)
*hh	Checsum

Fuente: LLAMAS, Luis. *Localización GPS con Arduino y los módulos GPS neo 6*.

<https://www.luisllamas.es/localizacion-gps-con-arduino-y-los-modulos-gps-neo-6/>. Consulta: 3 de junio de 2020.

Para darle significado a la trama \$GPRMC se utiliza la librería TinyGPS para obtenerlo de manera rápida.

2.4.4.1. Librería TinyGPS

Es una librería desarrollada por Mikal Hart para obtener de forma rápida los valores de latitud, longitud, altura, tiempo, velocidad y curso. Este código proporciona una función que extrae del formato NMEA los datos de ubicación que se necesitan y los asigna a las variables que se han designado para almacenar latitud y longitud, pudiendo después de esto manipular y utilizar a conveniencia los datos.

2.5. LoRa

LoRa es un protocolo de comunicación inalámbrica punto a punto de largo alcance y baja potencia. Su nombre es proveniente de las palabras *Long Range* en inglés, que significa largo alcance y se refiere precisamente a la alta distancia que alcanzan sus transmisiones con una potencia mínima. Está diseñado para comunicar dispositivos iot, los cuales transmiten pequeños volúmenes de información.

Estas comunicaciones alcanzan por lo regular hasta 10 km en línea vista, aunque con condiciones adecuadas de topografía y adaptando los valores de modulación se pueden conseguir mayores longitudes. La transmisión más larga registrada con LoRa fue de 702 km con un dispositivo montado en un globo aerostático y otro en tierra.

Este protocolo tiene sus inicios en una empresa francesa Cycleo y posteriormente fue adquirido por Semtech, compañía dedicada a la producción de chips. Semtech es la sociedad que fundó en 2015 la LoRa-Allience¹⁴, asociación cuya misión es estandarizar y promover el protocolo mediante las redes LoRaWAN que se explicará más adelante.

2.5.1. Modulación LoRa

Una modulación es la técnica de transportar información mediante una señal de radio conocida como portadora, existen las modulaciones análogas como AM, PM y FM en donde la portadora adopta infinitos valores de amplitud, fase o frecuencia respectivamente y modulaciones digitales como ASK, PSK,

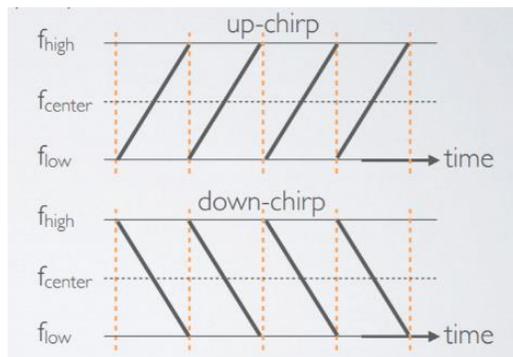
¹⁴ LoRA. *What are LoRa® and LoRaWAN®?* <https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/>.

FSK en donde el valor modulado de la señal solo mantiene dos valores que representan un bit.

De estos tipos de modulación digital se derivan otras técnicas que cambian la forma de codificar la información y una de ellas es LoRa que utiliza FSK (*Frequency Shift Keying*) o Modulación por desplazamiento de frecuencia para transportar información.

LoRa es una modulación de radiofrecuencia derivada de *chirp spread spectrum* cuya base de operatividad es la codificación de bits mediante el aumento o disminución de la frecuencia en periodos muy cortos, dando lugar a chirridos. Específicamente a un chirrido que aumenta su frecuencia se le conoce como *up-chirp* y a uno que disminuye como *down-chirp*

Figura 30. **Up-Chirp y Down-Chirp**



Fuente: Mobile Fish. *Lora / Lorawan tutorial 12*.

https://www.mobilefish.com/download/lora/lora_part12.pdf. Consulta: 1 de julio de 2020.

Cada uno de los cambios de frecuencia es llamado símbolo y representa un conjunto de bits de tamaño 2^{SF} , donde SF es conocido como *Spreading Factor*.

2.5.1.1. Spreading factor (SF)

Este factor indica el número de bits que se codifican en cada símbolo, en la modulación LoRa SF toma valores desde 7 a 12. Es decir que si el dispositivo LoRa está configurado con SF=7, en cada chirrido transmitirán 7 bits que pueden adoptar $2^7 = 128$ valores y con una configuración de SF=12 el número de valores que puede representar el símbolo será de $2^{12} = 4096$.

Para representar el dato a transmitir por medio de un chirrido se utiliza la frecuencia inicial, la cual toma un valor determinado dependiendo del dato a transmitir.

Para entender el concepto de codificación de manera sencilla, suponiendo que se quiere transmitir un número el decimal 64 con un SF=7.

Primero se establece que con el SF=7 la frecuencia inicial podrá tomar 128 valores y cada uno representará del 0 al 128. El número 64 en binario es 1 000 000 y estará dado por un chirrido que inicia en la frecuencia central, llega a la frecuencia superior y completa el período con un chirrido complementario, como se muestra en la siguiente figura. Se puede visualizar que representa una secuencia de transmisión del número 0,6432.

Donde n=1,2,3,4

2.5.1.3. Data Rate (DR)

Para realizar el cálculo de la velocidad de transmisión en bits/segundo se necesita saber cuántos símbolos por segundo se transmiten y considerar las variaciones de SF y CR en los paquetes que se envían.

La tasa de símbolos por segundo que es posible transmitir depende del ancho de banda, debido a que este da la pauta de cuantos bits puede contener un símbolo, a mayor ancho de banda se pueden transmitir más bits y viceversa por lo tanto la tasa de símbolos por segundo esta dado por:

Ecuación 8:

$$\text{Simbolos/seg} = \frac{BW}{2^{SF}}$$

Donde BW representa el ancho de banda, en LoRa se utilizan canales de 125 khz y 500 khz. Por lo tanto, el cálculo del *Data Rate* está dado por, la cantidad de bits por símbolo a transmitir (SF), la tasa de símbolos por segundo y la tasa de bits no redundantes (CR)

Ecuación 9:

$$DR = SF * \frac{BW}{2^{SF}} * CR \quad [\text{bits/s}]$$

Tabla VII. Ejemplos de *Data Rate*

<i>Spreading Factor (SF)</i>	Ancho de banda del canal BW	<i>Code Rate</i>	<i>Data Rate (DR)</i>
SF7	125000Hz	4/5	5 468,7 bits/s
SF8	125000Hz	4/5	3 125 bits/s
SF9	125000Hz	4/5	1 757,8 bits/s
SF10	125000Hz	4/5	976,5 bits/s
SF11	125000Hz	4/5	537,1 bits/s
SF12	125000Hz	4/5	292,9 bits/s
SF7	125000Hz	4/6	4 557,2 bits/s
SF8	125000Hz	4/6	2 604,1 bits/s
SF9	125000Hz	4/6	1 464,8 bits/s
SF10	125000Hz	4/6	813,8 bits/s
SF11	125000Hz	4/6	447,5 bits/s
SF12	125000Hz	4/6	244,1 bits/s

Fuente: LoRA. *What are LoRa® and LoRaWAN®?* <https://loradevelopers.semtech.com/library/tech-papers-and-guides/loralandlorawan/>. Consulta: 2 de julio de 2020.

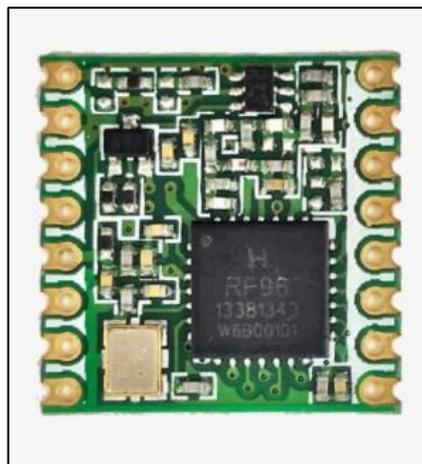
En la tabla se muestran ejemplos de valores de SF y CR y claramente se puede notar que, si se aumenta SF o CR la velocidad de transmisión disminuye, pero al realizar esta acción se puede brindar mayor garantía de que el receptor logre interpretar el mensaje. Con la consecuencia de gastar más energía debido a que el tiempo de transmisión aumenta.

2.5.2. Módulo RFM95W

Este módulo de radio LoRa es muy popular por sus características de larga distancia, baja potencia y alta inmunidad a interferencias y sobre todo bajo costo. Tiene una sensibilidad de -148dBm y potencia de transmisión de +14dBm, siendo alimentado tan solo con 3,3 V. El rango de frecuencias con el que se puede configurar este módulo es desde los 800 Mhz hasta los 928 Mhz pudiendo trabajar con las bandas ISM de Europa 868 MHz y Estados Unidos 915 MHz. La frecuencia de trabajo se configura por software a la ideal de cada país, más adelante se verá la banda de frecuencias en la cual funcionará el pluviómetro digital.

Además, una de las características más importantes de este chip es la interfaz de comunicación SPI de la cual ya se se platicó anteriormente y se utilizará para comunicar este módulo al microcontrolador.

Figura 32. **Módulo LoRa RFM95W**



Fuente: HOPERF. *RFM95W. LoRa Module*. <https://www.hoperf.com/modules/lora/RFM95.html>.

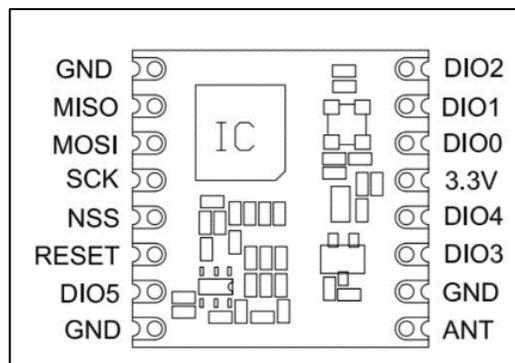
Consulta: 1 de agosto de 2020.

Otra de las características a destacar de este módulo es su diminuto tamaño pues solamente mide 16x16 milímetros, haciendo se utilice en dispositivos pequeños y de bajo costo.

2.5.3. Conexión al microcontrolador

El módulo de radio LoRa tiene una interfaz SPI con la cual puede comunicarse con el Arduino pro mini para enviar y recibir datos que son traducidos de ondas electromagnéticas a señales eléctricas y viceversa por medio de la antena.

Figura 33. **PINOUT RFM95W**



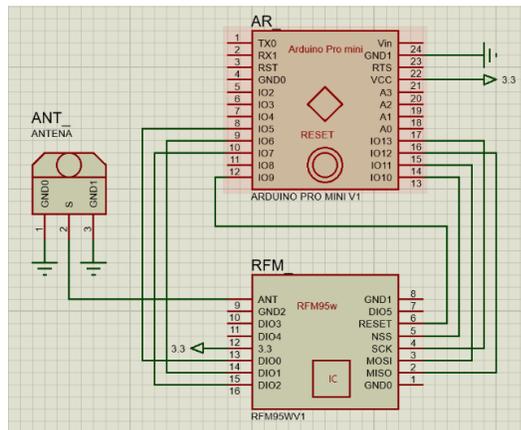
Fuente: HOPERF. *RFM95/96/97/98(W) - Low Power Long Range Transceiver Module V1.0.*

https://cdn.sparkfun.com/assets/learn_tutorials/8/0/4/RFM95_96_97_98W.pdf. Consulta: 2 de agosto de 2020.

Además de la interfaz SPI, antena y pines de alimentación, el módulo posee 6 pines GPIO que pueden ser configurados para distintas funciones, por lo regular se dejan para indicar estados del chip y recibir datos para entrar y salir de modo durmiente para ahorrar energía.

La conexión del módulo al microcontrolador en la tarjeta de pluviómetro quedaría de la siguiente manera.

Figura 34. **Conexión RFM95W**



Fuente: elaboración propia, empleando Proteus.

2.5.4. Programación de LoRa

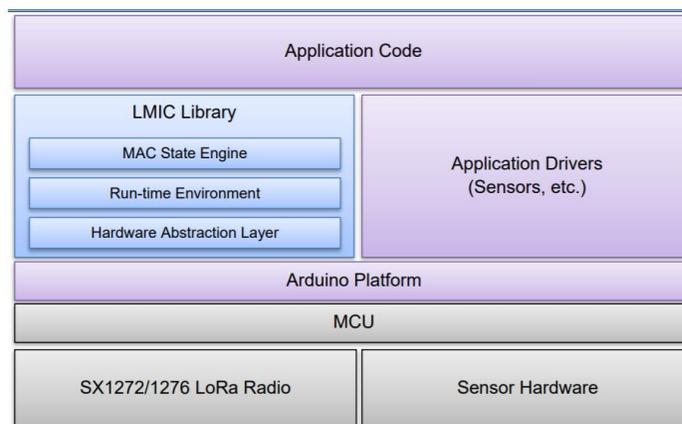
La recepción y envío de datos por medio de radiofrecuencia LoRa quizá sea de las tareas más complejas que realizará el microcontrolador en la tarjeta electrónica del pluviómetro de balancín y al ser este un dispositivo de internet de las cosas todo girará en torno a la conexión a la red.

Siendo LoRa una modulación de radio, se debe tener estricto cuidado en el manejo de tiempos mediante *timers* e interrupciones para el cumplimiento de las especificaciones del protocolo, afortunadamente para una rápida implementación, hay desarrolladores que han puesto a disposición de la comunidad de desarrolladores de dispositivos LoRa una librería que se encarga

de realizar todas las funciones que debe cumplir el protocolo LoRa y LoRaWAN del cual se hablará más adelante.

Esta librería se llama LMIC y simplifica el manejo de la comunicación a funciones y eventos. Dada esta ventaja se construirá la aplicación del pluviómetro de balancín tomando como base esta librería, poniendo en paralelo las tareas de medición y comunicación.

Figura 35. **Arquitectura de firmware LMIC**



Fuente: LAAS-CNRS Projects. *Arduino LoRaWAN MAC in C (LMIC)*<https://redmine.laas.fr/attachments/download/1505/LMIC-v2.3.pdf>. Consulta: 2 de agosto de 2020.

En las secciones subsiguientes se explicará cada una de las partes que conforman nuestra arquitectura de *firmware* para cumplir con el objetivo de medir lluvia y comunicación en tiempo real.

2.5.4.1. Librería LMIC

LMIC es una librería ligera escrita en lenguaje C, desarrollada por la empresa IBM para el uso fácil del protocolo LoRaWAN en las bandas de frecuencia ISM US915 Mhz, EU868 Mhz, AU921 Mhz, AS923 Mhz, admite dispositivos clase A y B, se entrará más a detalle sobre estas bandas de frecuencia y tipos de dispositivos en el capítulo de LoRaWAN.¹⁵

Se inicia explicando el funcionamiento de esta librería, partiendo de la parte de más bajo nivel, la que controla el *hardware* del microcontrolador. LMIC anexa un código portable llamado *Hardware Abstraction Layer* o HAL por sus siglas en inglés. Esta porción de código funciona como un *framework* multiplataforma que hace más sencilla la configuración de interrupciones, interfaces, GPIOs del microcontrolador para concentrarse solamente en la aplicación de alto nivel.

En otras palabras, esta porción de código nos deja la parte de declarar pines de interfaces, *timers*, interrupciones, y ahorrarse la parte de revisar la *datasheet* del micro y configurar uno a uno cada pin. Se llama multiplataforma porque el programa de aplicación o algoritmo no estará asociado directamente al *hardware*, sino HAL estará de intermediario para definir los puertos para los distintos chips para los cuales es compatible. Por lo tanto, será posible utilizar el mismo programa en un Arduino UNO, como en un Arduino Pro-mini o MEGA.

En el caso de la configuración para la tarjeta del pluviómetro de balancín, la librería LMIC permite definir las conexiones hacia el radio LoRa mediante un objeto llamado `Imic_pinmap`. Según la figura 34 de conexión del módulo RFM95W la declaración de pines quedaría de la siguiente manera:

¹⁵ KOOIJMAN, Mathijs; TELKAMP, Thomas. *Arduino LoraWan MAC in C (LMIC)*. <https://redmine.laas.fr/attachments/download/1505/LMIC-v2.3.pdf>.

Figura 36. Pinmap RFM95W

```
const lmic_pinmap lmic_pins = {
  .nss = 10,           // Chip Select SPI
  .rxtx = LMIC_UNUSED_PIN, // Switch de antena RXTX
  .rst = 9,           // Reset pin
  .dio = {5,6,7},    // DIO States
};
```

Fuente: elaboración propia, empleando Arduino IDE.

Donde *nss* es parte del protocolo SPI conocido como chip select, *rxtx* comanda la función de la antena como receptor o emisor, en el caso del pluviómetro no se controla esta conmutación. También se nombran los pines de *reset* para reiniciar el módulo de forma forzada y los DIO que son básicamente banderas que indican el estado del módulo para iniciar algún evento.

Como se muestra en la figura 35, la librería LMIC se divide en la capa de *hardware* HAL del cual ya se habló, el entorno de ejecución y el motor de estados.

El entorno en ejecución es el alma del programa, ya que aquí se manejan las funciones de autenticación, envío, recepción y las tareas preprogramadas del protocolo LoRaWAN, incluyendo la inicialización del protocolo con llaves de autenticación y configuración de parámetros de radiofrecuencia.

La librería permite utilizar estas funciones para definir el modo de funcionamiento del dispositivo y el período de transmisión. En caso del pluviómetro de balancín tendrá un funcionamiento de envío de datos por horario periódico. Por lo tanto, en el programa se define el período de transmisión de datos. Existen otras aplicaciones donde se especifica el envío de datos en caso se active una interrupción externa o el valor de un sensor llega a un valor crítico

no siendo el caso de la medición de lluvia donde el interés radica en la cantidad de lluvia en un período de tiempo.

Algunas de las funciones del entorno de ejecución más importantes y a utilizar en nuestra tarjeta son:

Tabla VIII. **Funciones de ejecución LMIC**

Función	Detalle
void os_init ()	Inicializa el entorno de la librería LMIC.
void os_runloop_once ()	Establece el inicio del ciclo de transmisión.
void os_setTimedCallback (osjob_t* job, ostime_t time, osjobcb_t cb)	Planifica la ejecución de una función en base al tiempo determinado.
LMIC_setSession (0x13, DEVADDR, NWKKEY, APPKEY);	Define paquete a enviar para autenticación.
void LMIC_setTxData ()	Coloca en la cola un dato a transmitir.
LMIC_setTxData2 (u1_t port, xref2u1_t data, u1_t dlen, u1_t confirmed)	Coloca en la cola un dato a transmitir esperando confirmación.
ostime_t os_getTime ()	Consulta la hora del sistema.
void os_setCallback (osjob_t* job, osjobcb_t cb)	Coloca una tarea dependiendo de una entrada o evento.
void os_clearCallback (osjob_t* job)	Limpia las tareas programadas.

Fuente: LAAS-CNRS Projects. *Arduino LoRaWAN MAC in C (LMIC)*.

<https://redmine.laas.fr/attachments/download/1505/LMIC-v2.3.pdf>. Consulta: 2 de agosto de 2020.

La librería LMIC tiene funcionalidades de llamada y devolución (*callbacks*) cuyo objetivo es consultar información específica de la aplicación y activar un evento o actualizar banderas de estados. Quizá una de las funciones más importantes para definir el flujo del programa es `onEvent(ev_t ev)`. Esta función no devuelve ningún valor, pero muestra un *log* de los procedimientos del protocolo dependiendo de la variable de entrada.

Tabla IX. **Log de eventos LMIC**

Evento	Descripción
EV_JOINING	Inicio de autenticación en la red.
EV_JOINED	Dispositivo autenticado exitosamente.
EV_JOIN_FAILED	No se pudo unir a la red.
EV_REJOIN_FAILED	No se pudo unir a una nueva red.
EV_TXCOMPLETE	El paquete ha sido enviado y espera confirmación si está configurada.
EV_RXCOMPLETE	Paquete recibido (Solo para dispositivos clase B).
EV_SCAN_TIMEOUT	No se tuvo respuesta para programación de envío <i>uplink</i> .
EV_BEACON_FOUND	Se recibió primera respuesta para programar próximo envío.
EV_BEACON_TRACKED	Se recibió el valor del próximo tiempo de envío.
EV_BEACON_MISSED	No se recibió el valor del próximo tiempo de envío.
EV_LOST_TSYNC	Se desincronizó el tiempo de respuestas de <i>beacon</i> .
EV_RESET	Se reinició la sesión por el desbordamiento del contador de paquetes.

Continuación de la tabla IX.

EV_LINK_DEAD	No se recibió respuesta del servidor en un largo período de tiempo.
EV_LINK_ALIVE	Se recibió noticias del servidor después de no responder en un largo período de tiempo.
EV_TXSTART	Se le enviarán los datos al módulo de radio para la transmisión.

Fuente: LAAS-CNRS Projects. *Arduino LoRaWAN MAC in C (LMIC)*.
<https://redmine.laas.fr/attachments/download/1505/LMIC-v2.3.pdf>. Consulta: 2 de agosto de 2020.

2.5.4.2. Payload pluviómetro de balancín

Hasta ahora se ha mostrado el funcionamiento del protocolo LoRa y la librería para utilizarlo y programar el módulo de radio. Pero, no se ha tratado aún el envío de los datos de lluvia, como se enviará la información de los volteos del pluviómetro. Pues estos son los datos que puramente nos interesan y se le llaman *payload* o carga útil.

Recapitulando acerca de cómo afecta el *Spreading Factor*, *Code Rate* en la velocidad Data Rate y a su vez en el tiempo en el aire de la transmisión. Se sabe que a mejores condiciones del ambiente se puede reducir el SF y CR teniendo mayor velocidad y menos tiempo de transmisión. Pero también otro factor importante que afecta el tiempo en el aire es el tamaño del *payload*. Dado a que, a mayor cantidad de datos a enviar, aumenta el tiempo en el aire.

LoRa es un protocolo de internet de las cosas y este concepto se entiende como una gran cantidad de dispositivos con sensores emitiendo ondas de radio

al espacio periódicamente por lo cual siempre se necesita un canal disponible para la comunicación.

Si el tiempo en el aire de cada dispositivo es muy largo, se tendrá un problema de indisponibilidad de la red y saturación. Entonces velar que el tiempo de las transmisiones sea muy corto es una prioridad. Incluso existen regulaciones internacionales y nacionales que limitan estos tiempos y se verá más adelante en el capítulo de LoRaWAN.

Para hacer que las transmisiones de los paquetes LoRa tengan una duración corta, se deben enviar puramente los datos que den más información de lo que se mide, en el caso del pluviómetro de balancín el número de volteos da el valor fundamental para el cálculo de lluvia según la ecuación 3 revisada en el capítulo 1. Considerando que un pluviómetro es instalado en un punto geográfico estático, por lo cual la ubicación no cambiará constantemente, se puede hacer que este dato no se envíe en todas las transmisiones sino enviar la ubicación solamente al encender el dispositivo.

Dada esta situación se tienen dos *payloads*, uno con el número de los volteos del balancín que se transmitirá periódicamente durante el día y otro de ubicación que se transmitirá al inicio. También se considera que existe la posibilidad de la pérdida de algunos paquetes de información y para asegurar que no se pierda información. El dato para transmitir será el número de volteos acumulado desde que el pluviómetro está instalado, siendo una variable que incrementa cada vez que se detecta lluvia.

Con esto se conoce que esta variable será positiva, entera y alcanzará un valor mayor al pasar el tiempo, por lo tanto, se utiliza un tipo de dato `uint32_t` el cual aparta 32 bits de memoria para un valor entero sin signo con un rango desde

0 hasta 4 294 967 295 equivalente a 343 597383,6 mm de lluvia. Este *payload* será trasladado por el fport 1 de LoRaWAN.

Tabla X. **Payload lluvia**

Acumulado de volteos [uint32_t]			
Byte 3	Byte 2	Byte 1	Byte 0

Fuente: elaboración propia, empleando Microsoft Excel 365.

Para el caso de la ubicación, se tienen los valores de latitud y longitud que tienen una parte entera y una parte con 6 decimales, es decir valores flotantes que también pueden ser negativos. Para simplificar la codificación de bytes es conveniente convertirlos a un dato entero multiplicando por una potencia de 10 con la cantidad de decimales. Y se guarda este valor en un tipo de dato *long* de 32 bits para luego ser transmitido por LoRa. Este *payload* será trasladado por el fport 5 de LoRaWAN.

Tabla XI. **Payload ubicación**

Longitud [Long]				Latitud [Long]			
Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0

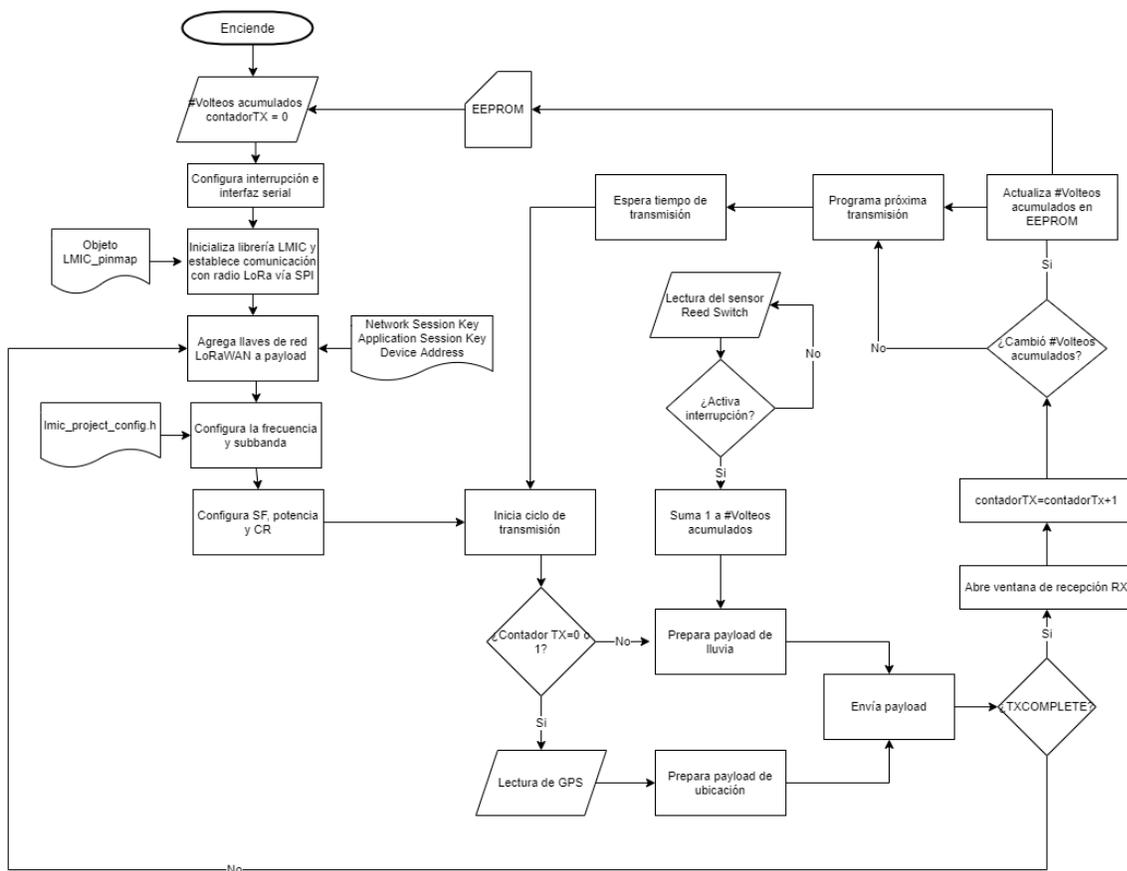
Fuente: elaboración propia, empleando Microsoft Excel 365.

2.6. Algoritmo de tarjeta electrónica para pluviómetro de balancín

La tarjeta de pluviómetro tendrá la capacidad de acumular un valor de lluvia detectada en una variable llamada *volteosAcumulados* y transmitir la información vía LoRa cada 10 minutos y la ubicación cada vez que se encienda el dispositivo.

A continuación, se muestra el flujo del programa situado en el microcontrolador para realizar la medición de lluvia y transmisión de datos vía LoRa empleando la librería LMIC que añade bytes para la autenticación en la red LoRaWAN.

Figura 37. Diagrama de flujo de programa tarjeta de pluviómetro



Fuente: elaboración propia, empleando Microsoft Visio 2019.

Al encender, el dispositivo busca en la memoria EEPROM el valor registrado de la variable volteosAcumulados, el cual representa la lluvia acumulada histórica.

El programa recoge los valores del objeto LMIC_pinmap que contiene los pines de conexión del módulo LoRa y configura la interfaz SPI. El pluviómetro de balancín funcionará en una red LoRaWAN lo que implica que en los paquetes transmitidos viajarán llaves para autenticarse a la red. Al iniciar el programa se preparan estas llaves en la trama a transmitir y se dejan esperando que se le agregue el payload. Estas llaves son Network Session Key y Application Session Key, además del Device Address que identifica al dispositivo en la red. Estos valores permiten al dispositivo entrar a la red y aplicación, se profundizará más en estas llaves en el capítulo de LoRaWAN.

Posteriormente, se realiza la configuración de radiofrecuencia LoRa, eligiendo la frecuencia y canales a utilizar, además del Spreading Factor, Code Rate y potencia de transmisión. Después de que el microcontrolador tenga todas las configuraciones que utilizará para las transmisiones, se iniciará el bucle infinito de transmisiones.

Como ya se definió anteriormente, para el pluviómetro de balancín, se enviará la ubicación cada vez que se encienda el dispositivo, para esto se tiene una condición que consulta si el contador de transmisiones tiene los valores 0 o 1, los cuales se definen como los 2 primeros paquetes que transmite la tarjeta.

En caso de que sean los primeros dos paquetes, el microcontrolador pasará a una función para leer los valores de GPS y codificarlos en bytes para ser transmitidos.

Cuando el paquete es puesto en cola y transmitido, el *log* de eventos de LMIC nos reflejará un estado TXCOMPLETE, lo cual indica que se abren dos ventanas de recepción de datos establecidas por el protocolo LoRaWAN e incrementa el contador de paquetes enviados.

Para una próxima transmisión, el flujo del programa actualiza el valor de `volteosAcumulados` en caso de incrementar y se establece el inicio de la próxima transmisión según `TX_INTERVAL`.

El dispositivo espera a que se cumpla el tiempo de espera para la siguiente transmisión, sin dejar de leer la interrupción del pluviómetro. En caso de detectar lluvia se activa la interrupción e incrementa el valor de `volteosAcumulados` dejándolo listo para cuando se dé la próxima transmisión.

En caso de haber detectado lluvia y haberse transmitido, se actualizará el valor de `volteosAcumulados` en la memoria EEPROM, como protección ante el corte de energía. Además, en caso de perderse un paquete entre transmisor y receptor LoRa se tendrá el valor de la lluvia acumulada en el próximo envío.

El protocolo LoRaWAN admite confirmación de recepción de paquetes entre dispositivo y servidor. Sin embargo, al utilizar esta opción se tendrían más transmisiones y tiempo en aire, lo cual no es conveniente si se tienen muchos dispositivos en una misma zona.

2.7. Energía

Cuando todos los componentes que harán el procedimiento de medir lluvia y transmitir los datos están conectados y funcionando, es necesario conocer el consumo energético para poder elegir una fuente de energía adecuada para el equipo. Realizando una medición del sistema a un voltaje de 3,7 V la corriente que consume el circuito es de 69 mA y en etapa de transmisión 80 mA, por consiguiente, se busca una solución que nos proporcione una corriente constante y redundante.

Existen técnicas para reducir el consumo de energía de los dispositivos LoRa, estas consisten en hacer ingresar al microcontrolador en un modo de hibernación, esperando tener el próximo período para envío de datos o cuando alguna interrupción externa se vea afectada. Esto reduciría considerablemente el consumo de energía y podría funcionar únicamente con una batería por un largo período de tiempo.

Esta opción abarca un entendimiento más profundo del trabajo del microcontrolador y una programación de más bajo nivel, para ser completamente óptimos energéticamente. Dado que la finalidad de este proyecto no es con fines de mostrar lo extremadamente eficiente que es energéticamente el dispositivo de medición de lluvia, sino la utilización de una red LoRaWAN para visualización de datos en tiempo real, se utilizará una fuente recurrente de energía con un sistema de panel solar y batería.

2.7.1. Batería

La batería es el elemento que proveerá de energía el circuito de la tarjeta de pluviómetro de balancín, es imprescindible ya que este aparato funcionará instalado en campo, sin acceso a una red eléctrica convencional.

Existen pilas convencionales desechables, pero en este caso se considera una batería recargable de Li-ion, comúnmente conocidas con el número 18650 la cual es muy similar en tamaño a una AA, pero tiene salida de voltaje de 3,7 V y existen en diferentes capacidades de mAh.

Este tipo de baterías se utilizan en equipos que requieren buena capacidad de carga y descarga como linternas, bancos de energía, computadoras portátiles, entre otros. La codificación 18650 es para identificar las dimensiones, los

primeros dos números representan el diámetro de 18 mm y los otros 3 representan la longitud con 65 mm.

Estas baterías tienen la característica de no tener efecto de memoria, efecto que reduce la capacidad de carga al no descargarse completamente. También tienen larga vida útil y con buena cantidad de ciclos entre los 500 y 1 000.

También es probable encontrar inconvenientes con estas baterías dado un mal manejo y un circuito no adecuado para la carga y descarga, pueden llegar a explotar en consecuencia a un calentamiento o sobrecarga. Tampoco es recomendable la descarga completa de la batería ya que reduce la vida útil considerablemente.

Para alimentar la tarjeta electrónica de pluviómetro de balancín es necesario un voltaje de 3,3 V y una corriente de 69 mA, por lo cual resulta conveniente utilizar este tipo de baterías, teniendo que buscar la capacidad correcta para esta aplicación.

Esta batería estará siendo cargada constantemente por un panel solar y un circuito de carga, por lo que se debe considerar cuanto tiempo se debería tener de respaldo en caso de haber días nublados, con poca captación de radiación solar y obviamente cubrir los períodos nocturnos, se considera que un período de 3 días.

Con una batería de 5 000 mAh el pluviómetro podría funcionar aproximadamente 72 horas si se considera un período de transmisión de 10 minutos, dándole un buen respaldo ante ocasiones con poca captación de energía solar.

Figura 38. **Batería 18650**



Fuente: elaboración propia.

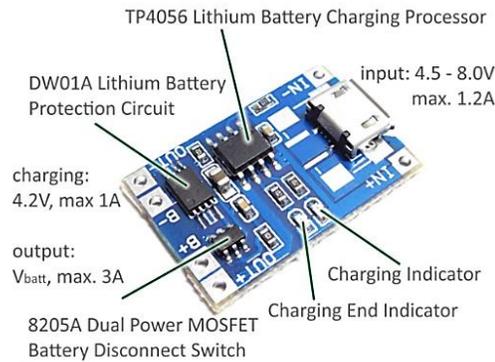
2.7.2. Módulo de carga TP4056

Las baterías 18650 deben mantener parámetros correctos de corriente y voltaje durante el período de carga y también tener protección ante eventos de sobrevoltajes o sobre corrientes que puedan afectar su vida útil o producir sobrecalentamientos llegando al punto de explosión. Por esta razón debe existir un circuito de control para la carga de la batería.

El chip TP4056 es un controlador de carga de voltaje y corriente constantes, que tiene una arquitectura que evita que se presenten corrientes negativas que tengan consecuencias de descargar la batería.

Este chip se utiliza en el módulo de carga con el mismo nombre TP4056, el cual contiene un circuito empleando este controlador junto a un chip de protección y un arreglo de transistores mosfet para realizar la conmutación entre la fuente de alimentación y la batería. Este circuito englobado en un módulo está diseñado para cumplir con los parámetros de carga de una batería 18650, por lo cual se utiliza en la tarjeta de pluviómetro de balancín.

Figura 39. **Módulo TP4056**



Fuente: IOE. *5pcs TP4056 Micro USB 5V 1A 18650 Lithium Battery Charging Dual Protection DW01A*. <https://www.importedelectronics.com/products/5pcs-5v-micro-usb-1a-18650-lithium-battery-charging-board-charger-module-tp4056>. Consulta: 1 de enero de 2021.

El voltaje de alimentación de este módulo es entre 4,5 V y 8,0 V, por lo que se busca una fuente de energía que provea un valor entre el rango.

2.7.3. Panel fotovoltaico

Un panel solar fotovoltaico es un dispositivo que aprovecha la energía percibida por luz directa del sol para generar electricidad, estos paneles contienen células semiconductoras que aprovechan el efecto fotovoltaico, que ocurre cuando fotones hacen desprender electrones de una unión PN generando una diferencia de potencial.

Esta diferencia de potencial puede ser aprovechada para alimentar aparatos electrónicos y como la tarjeta de pluviómetro de balancín. Para cargar la batería de la tarjeta se necesita un panel solar fotovoltaico que cumpla con los parámetros de alimentación del módulo de carga TP4056 los cuales son de 4,5 V a 8 V con una corriente máxima de 1,2 A.

Dados estos parámetros se elige un panel solar que proporcione el voltaje correcto y con la corriente suficiente para hacer funcionar el equipo electrónico, siendo estos valores 5,5 V y 160 mA.

Figura 40. **Panel solar 5.5V 160 mA**

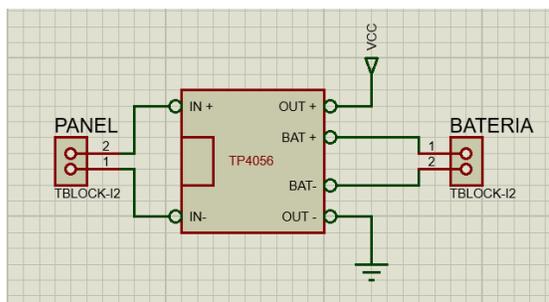


Fuente: Aliexpress. *Yucosolar*. <https://yucosolar.aliexpress.com/store/127080>. Consulta: 1 de enero de 2021.

2.7.4. Circuito de energía

Los tres elementos anteriores conformarán el circuito de energía, parte de la tarjeta electrónica que tiene el objetivo de proporcionar alimentación recurrente para el funcionamiento del equipo, el circuito queda conectado de la siguiente manera:

Figura 41. **Circuito de energía**



Fuente: elaboración propia, empleando Proteus.

2.8. Implementación de tarjeta electrónica

Teniendo definidos los componentes, conexiones y programación de los distintos módulos de la tarjeta para pluviómetro de balancín se define un circuito esquemático con todos los componentes conectados. Teniendo este circuito esquemático se construye un circuito PCB para fabricar la tarjeta PCB donde estarán montados los componentes. Finalmente, la construcción de la tarjeta y montaje en una caja con cierto grado de protección ante intemperie.

2.8.1. Circuito esquemático

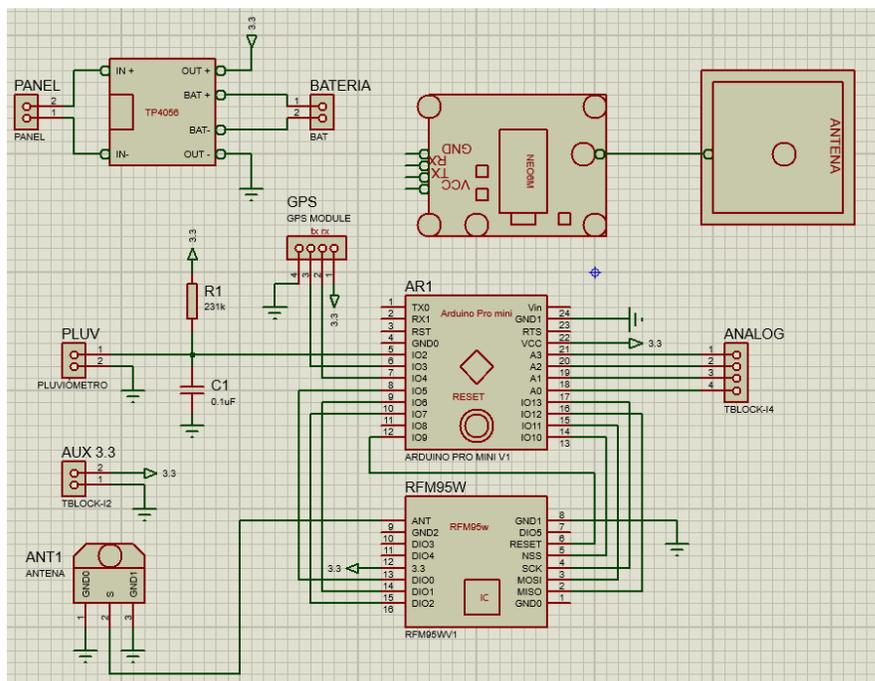
Un circuito esquemático es un diagrama que define las conexiones de los componentes representados con símbolos eléctricos, no tomando en cuenta la ubicación real de los componentes en la placa física. Existen diversos *softwares* para realizar estos diagramas, para el caso del circuito de la tarjeta del pluviómetro de balancín se utilizó Proteus.

En el circuito esquemático se dejan terminales para conectar componentes que no estarán montados sobre la tarjeta. Además, se agregaron terminales

como Aux3,3 y ANALOG las cuales no tienen ninguna funcionalidad actual, pero hay posibilidad de agregar más sensores en estas entradas analógicas. Por lo tanto, se podría conectar sensores analógicos de humedad, temperatura, velocidad o dirección del viento y radiación formando una estación meteorológica completa con transmisión de datos LoRa.

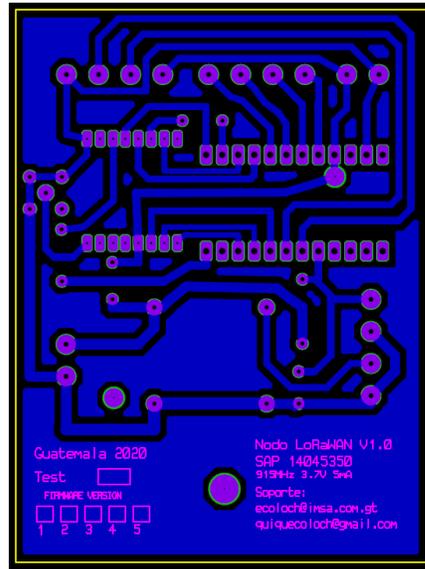
El módulo GPS no quedará montado sobre la tarjeta electrónica y tendrá conectores para que pueda ser conectado al instalar el equipo, esto a manera de posicionar correctamente la antena. A continuación, se muestra el resultado del circuito esquemático.

Figura 42. Diagrama esquemático tarjeta de pluviómetro de balancín



Fuente: elaboración propia, empleando Proteus.

Figura 44. PCB tarjeta vista inferior



Fuente: elaboración propia, empleando Proteus.

Para identificar fácilmente las conexiones del pluviómetro, GPS, batería y panel, se etiquetaron los terminales *block* de la siguiente manera.

Tabla XII. Identificación de conectores tarjeta

Terminal Block	Componente	Identificación de pines			
		Negativo: -		Positivo: +	
AUX 3.3	Voltaje auxiliar				
ANALOG	Sensores adicionales	Análogo: A3	Análogo: A2	Análogo: A1	Análogo A0
GPS	Módulo GPS	VCC: +	GPS RX: GRX	GPS TX: GTX	GND: -
PLUV	Pluviómetro de balancín	— — — — —			
PANEL	Panel solar	VCC: +		GND: -	
BATERIA	Batería	VCC: +		GND: -	
ANT1	Antena LoRa				

Fuente: elaboración propia, empleando Microsoft Excel 365.

2.8.3. Construcción de tarjeta y resultados

Posterior a tener el diseño PCB se procede a construir la tarjeta electrónica y soldar los componentes. Existen distintos métodos para trasladar el diseño a una tarjeta electrónica física, pero, para que se tengan acabados profesionales algunas empresas ubicadas en China tienen equipos especializados, que ponen a disposición por medio de páginas web.

En estos sitios solamente es necesario subir el diseño, configurar parámetros, personalizar, pagar y esperar para recibirlos. Los resultados son los siguientes.

Figura 45. **Tarjeta electrónica para pluviómetro de balancín frontal**



Fuente: elaboración propia.

Figura 46. **Tarjeta electrónica para pluviómetro de balancín trasera**



Fuente: elaboración propia.

3. IMPLEMENTACIÓN DE RED LORAWAN

En los últimos años se ha visto un crecimiento acelerado de los dispositivos conectados a internet, esto a raíz de la apertura de muchas industrias al aprovechamiento de las tecnologías de vanguardia para la reducción de costos y aumento de la productividad en sus procesos.

Se estima que para 2025 se tengan 64 mil millones equipos de IOT funcionando en infinidad de aplicaciones en distintos sectores como el manufacturero, agricultura, salud, movilidad, logística, inmuebles, urbanismo y muchos otros.

Existe una frase atribuida al físico y matemático británico (1824-1907) William Thomson Kelvin la cual dice: “Lo que no se define no se puede medir. Lo que no se mide, no se puede mejorar. Lo que no se mejora, se degrada siempre”¹⁶ muchas veces traída a nuestros tiempos como “Todo lo que se puede medir se puede mejorar”, bajo esa premisa se está en busca realizar mediciones de todo el entorno que nos rodea, pero muchas veces este entorno requiere del desarrollo de nuevas tecnologías que se adapten completamente a nuestras necesidades.

Conectar un pluviómetro digital a internet es una tarea que puede verse sencilla empleando protocolos convencionales de acceso como WIFI, LTE, Ethernet entre otros. Sin embargo, las mediciones de lluvia regularmente se

¹⁶ Aiteco Consultores. Desarrollo y gestion. *Lo que no se mide, no se puede mejorar.* <https://www.aiteco.com/lo-que-no-se-mide/#:~:text=Esta%20frase%2C%20atribuida%20frecuentemente%20a,mide%2C%20no%20se%20puede%20mejorar.>

realizan en lugares remotos para el monitoreo de cultivos o aplicaciones donde no se tiene un acceso fácil, por lo cual en este documento se presenta una alternativa completamente funcional para conseguir la conectividad en áreas amplias con baja potencia.

En este capítulo se detallará el funcionamiento y la aplicación del protocolo de internet de las cosas LoRaWAN, conociendo inicialmente la categoría en la que se encuentra entre las tecnologías de IOT y las características que la hacen una opción muy fuerte en el mercado.

Posteriormente se profundizará en los elementos principales que llevan a cabo la transmisión de información y cómo se utilizó esta tecnología para la medición de lluvia en tiempo real, el cual es el fin de este trabajo.

3.1. Redes LPWAN

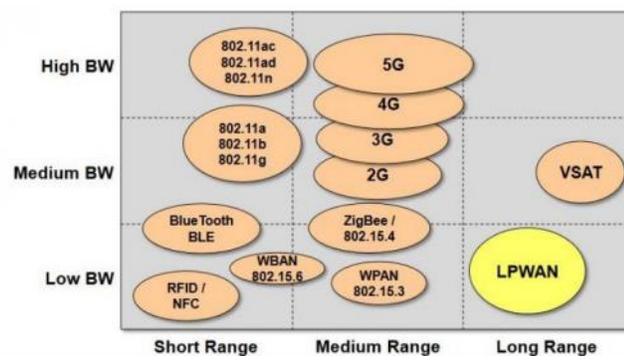
Una LPWAN (*Low Power Wide Area Network*) es una red de área amplia y baja potencia, cuya característica principal es proveer de cobertura de señal inalámbrica a dispositivos electrónicos con poca energía y a gran distancia. Este tipo de redes son ideales para equipos que solamente envían y reciben poco volumen de información y por lo general su fuente de energía es un panel solar o batería.

La Superintendencia de Telecomunicaciones de Guatemala categoriza algunas de estas tecnologías como RIABAP, categoría de la cual se hablará posterior a dejar el contexto de las aplicaciones y funcionamiento.¹⁷

¹⁷ Superintendencia de Telecomunicaciones. *Resolución SIT-DSI-349-2019*. https://sit.gob.gt/2019/09/11/resolucion-sit-dsi_349-2019/.

Dentro de los protocolos inalámbricos que existen, se puede posicionar a las LPWAN como de bajo ancho de banda, regularmente utilizado por sistemas embebidos y dispositivos de corto período de comunicación y de largo alcance para aplicaciones en entornos abiertos.

Figura 47. **Comparación tecnologías inalámbricas**



Fuente: HÉRNANDEZ CABALLERO, Silvia. *Estudio en detalle LoRaWAN*.

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/106369/7/shernandezc6TFM0120memoria.pdf>. Consulta: 2 de febrero de 2021.

En esta gráfica extraída de un estudio de comparación de LoRaWAN con otras tecnologías LPWAN realizado por Silvia Hernández en 2020. Se coloca a las LPWAN en el eje que coincide con el RFID, Bluetooth, ZigBee y redes personales, pero a diferencia de las habituales, LPWAN tiene largo alcance, no mayor a las comunicaciones satelitales, pero si más que las existentes para difusión celular y transmisiones de contenido digitales.

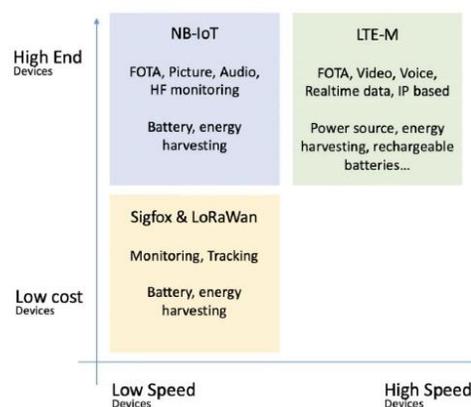
Estas redes se utilizan en aplicaciones donde se transfieren pequeñas cantidades de datos a largas distancias. Otra ventaja considerable de las LPWAN es el costo de implementación, ya que son protocolos livianos, lo cual reduce la complejidad del *hardware* para la comunicación de dispositivos.

Con respecto a la tecnología satelital, puede utilizarse para enviar o recibir información, pero se complica al requerir más complejidad en el hardware y no es tan eficiente en cuestión de energía.

Dentro de las LPWAN existe una subclasificación en función a la utilización de infraestructura celular, entre estas resaltan NB-IoT, LTE-M y con infraestructura independiente como LoRaWAN y Sigfox. Estas se utilizan en iguales aplicaciones, pero en diferentes entornos. Por ejemplo, en un entorno rural no existen antenas de telefonía para desplegar dispositivos con esta comunicación y una ciudad está casi cubierta por esta red, por lo cual si es conveniente utilizarla. Seleccionar una tecnología inalámbrica dependerá completamente de la aplicación y el entorno donde funcionará.

En el siguiente gráfico Silvia Hernández cataloga las LPWAN en función de la velocidad de transmisión y el costo del hardware.

Figura 48. **Comparación LPWAN costo y velocidad**



Fuente: HERNÁNDEZ CABALLERO, Silvia. *Estudio en detalle LoRaWAN*.
<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/106369/7/shernandezc6TFM0120memoria.pdf>. Consulta: 2 de febrero de 2021.

Uno de los objetivos también del proyecto de tarjeta para pluviómetro de balancín es el bajo costo en la implementación y teniendo en cuenta que la velocidad de LoRaWAN es suficiente para transmitir datos de lluvia. Se elegirá este estándar inalámbrico para el proyecto.

3.2. LoRaWAN

LoRaWAN es protocolo de red catalogado como LPWAN utilizado para aplicaciones de internet de las cosas. Utiliza como capa física la modulación inalámbrica LoRa y establece reglas para comunicar y administrar dispositivos desde internet.

Este estándar es manejado por la LoRa-Alliance creada en 2015 y tiene el objetivo de actualizar y proveer a desarrolladores de normas para interoperabilidad entre dispositivos y crecer el ecosistema de aplicaciones empleando LoRaWAN. Estas aplicaciones de crecimiento son en industrias de agricultura, edificios y ciudades inteligentes, fabricas, logística y empresas de servicios básicos como energía, agua, gas entre otros.

Una red LoRaWAN basa su funcionamiento en tres agentes de red, los dispositivos finales normalmente conocidos como nodos que conectan sensores o actuadores, concentradores y un servidor de red.

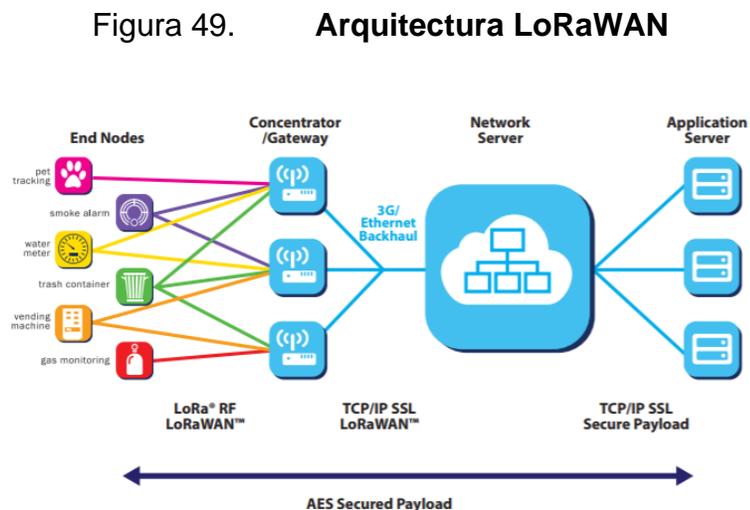
En los siguientes apartados se verá cómo interactúan estos tres elementos y a grandes rasgos las reglas de comunicación que siguen, así como las frecuencias que se utilizan y las regulaciones que existen en Guatemala para implementar una red de este tipo.

3.2.1. Arquitectura

Como bien se ha dicho, LoRaWAN es un protocolo de internet de las cosas, pero ¿cómo se hacen llegar los datos desde un sensor hacia internet?, en la arquitectura se definen como están interconectados los agentes para llevar a cabo este fin.

LoRaWAN define una topología tipo estrella, donde los dispositivos finales se conectan a uno o varios *gateways lora*, estos actúan como dispositivos transparentes convirtiendo los paquetes de radiofrecuencia a paquetes IP trasladándolos al servidor LoRa donde se realiza la autenticación y enrutamiento de los datos.

La comunicación en la red es de dos vías, ascendente y descendente, la transmisión nodo-servidor es conocida como *uplink* y servidor-nodo como *downlink*.



Fuente: LoRa Alliance. *LoRaWAN™ What is it?* <https://loro-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf>. Consulta: 2 de febrero de 2021.

Cada nodo de la red no está asociado a un *gateway* en específico, regularmente un paquete es recibido por varios *gateways* que solamente reenvían hacia el servidor donde se elimina la redundancia, se verifica autenticación y se enruta hacia aplicaciones de alto nivel para la utilización de los datos.

La comunicación entre *gateways* y servidor se realiza por métodos convencionales de acceso a internet con el estándar IP y forman también una topología estrella, por lo que muchos llaman a este esquema una estrella de estrellas, incluso existen redes globales de LoRaWAN donde se interconectan servidores ejemplificando la escalabilidad que tiene esta arquitectura.

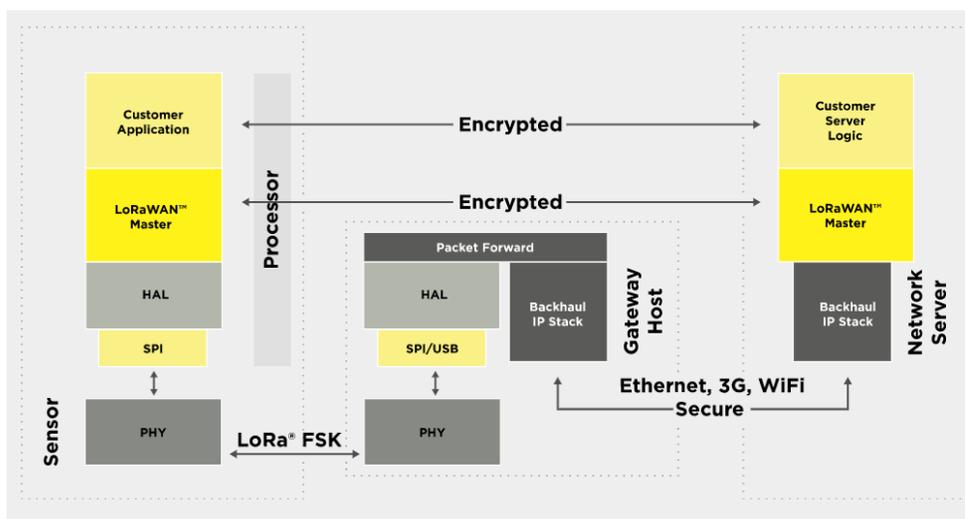
Los *gateways* LoRa tienen la capacidad de atender a un gran volumen de dispositivos gracias a la comunicación asíncrona de los nodos, los cuales solo están empleando un canal cuando tienen alguna medición lista o se lo indica un horario establecido. Esta facultad de escuchar muchos dispositivos al mismo tiempo se ve afectada por el tiempo en aire de los nodos, a su vez consecuencia del tamaño del *payload*, *code rate* y *spreading factor*, términos que se trató en el capítulo anterior.

Siempre en una red la seguridad es muy importante, por lo cual el estándar LoRaWAN tiene dos capas, una para la autenticación a la red y llegar al servidor y otra para autenticación en la aplicación en la que se encuentre el dispositivo, esto debido a que un servidor LoRa puede manejar distintos perfiles de sensores que están asociados a distintas aplicaciones y tienen diferentes usuarios finales. Se utiliza una encriptación AES con llaves de 128 bits. Para insertar estas llaves en los nodos existen dos tipos de autenticación en LoRaWAN, ABP (Activation By Personalisation) y OTAA (*Over The Air Activation*).

Para una autenticación ABP, las llaves son generadas en el servidor y trasladadas manualmente al nodo y ya podrá comunicarse. En una autenticación OTAA el nodo envía una solicitud al servidor con su identificador único y una llave de aplicación, si los valores son correctos, el servidor responderá con las dos llaves de servidor y aplicación. El pluviómetro de balancín tendrá una autenticación ABP.

En la siguiente figura se puede observar cómo están divididas las capas y en donde se realiza la encriptación en el nodo y descriptación en la aplicación del usuario.

Figura 50. **Modelo LoRaWAN**



Fuente: LoRa Alliance. *LoRaWAN™ What is it?* <https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf>. Consulta: 2 de febrero de 2021.

El modelo de capas del protocolo establece como capa física al protocolo LoRa, establece también una capa de transporte con la utilización de puertos, una capa de enlace a la red y otra de aplicación.

3.2.2. Frecuencias

El espectro radioeléctrico es un recurso físico limitado, por lo cual esta estrictamente regulado por los gobiernos de todo el mundo. Cada espacio de este espectro de frecuencias está dividido y asignado para la utilización en aplicaciones específicas en su mayoría siguiendo estándares internacionales.

Teniendo en cuenta estos estándares, LoRaWAN plantea la utilización de bandas ISM (Industrial Scientific Medic). Estas bandas han sido adoptadas internacionalmente para utilización de sistemas de comunicación inalámbrica sin licenciamiento formal. Actualmente algunos sistemas que las utilizan son microondas, wifi, bluetooth y otros.

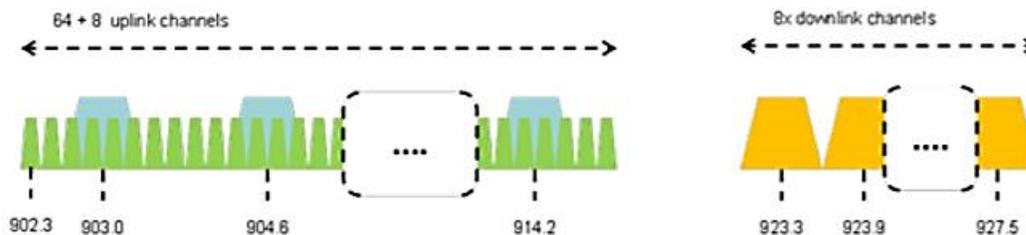
A pesar de que internacionalmente se definan estas bandas de frecuencia, no todos los países adoptan las mismas, en consecuencia, existe una división por regiones y LoRaWAN adopta estas regiones y recomienda para cada país una o varias bandas. Entre ellas están para Estados Unidos US915Mhz, Europa EU868, Australia AU921, Asia AS923.

En el caso de Guatemala es recomendada la región AU921 y US915 para la utilización de LoRaWAN. En consecuencia, a la cercanía con Estados Unidos y la fácil obtención de dispositivos se utilizará la banda US915 para la comunicación del pluviómetro de balancín.

La banda US915 define 64 canales de 125 Khz cada uno para transmisiones tipo *uplink* desde la frecuencia 902,3 a 914,9 MHz. Es posible también mantener 8 canales de 500 KHz desde los 903 MHz a los 904,9 MHz. Para las transmisiones tipo *dowlink* solo se tienen 8 canales de 500 KHz que van desde los 923,3 MHz hasta 927,5 MHz.

Los *gateways* en el mercado tienen la capacidad de atender 8 o 16 canales en simultáneo, por lo cual hacen una división en subbandas de 8 canales cada uno, teniendo 8 subbandas. Para el propósito del proyecto del pluviómetro de balancín se utilizará la subbanda 1.

Figura 51. **Canales LoRaWAN US915MHz**



Fuente: LoRa Alliance. *LoRaWAN™ What is it?* <https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf>. Consulta: 2 de febrero de 2021.

En Estados Unidos la potencia máxima en este espectro es de +30 dBm, sin embargo, la mayoría de los dispositivos se comunican con una potencia menor a esa.

3.2.2.1. Regulaciones de radiofrecuencia en Guatemala

En Guatemala el espectro radioeléctrico es regulado por la Superintendencia de Telecomunicaciones, organismo técnico del Ministerio de Comunicaciones, Infraestructura y Vivienda, y se encarga de velar el cumplimiento de la Ley General de Telecomunicaciones, administrando,

registrando y supervisando la utilización de las frecuencias, así como tomando un rol de juez entre operadores para resolver conflictos.¹⁸

La Ley General de Telecomunicaciones de Guatemala establece 3 tipos de bandas de frecuencia según el ente explotador, las bandas reservadas cuya utilización es solamente para entidades estatales, las bandas para radioaficionados y bandas reguladas para utilización de entidades privadas o individuales.

La autorización para el uso de las bandas reservadas y de radioaficionados no representa un título de usufructo, pero deben de regirse bajo normas internacionales y nacionales ratificadas y cumplir con el registro ante la SIT.

Por otra parte, las bandas reguladas pueden ser utilizados por entidades privadas nacionales e internacionales bajo un título de usufructo el cual es adquirido participando en una subasta pública tratada por la SIT.

La principal herramienta de la SIT para la distribución ordenada de frecuencias entre las distintas aplicaciones de comunicaciones inalámbricas es la Tabla Nacional de Atribución de Frecuencias conocida por sus siglas TNAF, documento actualizado y aprobado cada cuatro años por la SIT, la última versión se actualizó en 2020 y tiene vigencia de 2021 a 2024, estas mejoras se dan basándose en desarrollos tecnológicos y actualizaciones de organismos que regulan las radiocomunicaciones a nivel mundial.

Entre las organizaciones internacionales que influyen en la atribución de frecuencias en Guatemala están: la Unión Internacional de Telecomunicaciones

¹⁸ Superintendencia de Telecomunicaciones. *Registro de comercializadores de equipos terminales móviles y tarjetas SIM en línea*. <https://sit.gob.gt/>.

(UIT) anexo a las Naciones Unidas (NU), la Comisión Interamericana de Telecomunicaciones (CITEL) adscrita a la Organización de Estados Americanos (OEA).

La Tabla Nacional de Atribución de Frecuencias TNAF es integrada por tres documentos para la correcta interpretación: Gráfica, Cuadro y Pies de página.

La primera muestra de forma gráfica la distribución de frecuencias en el país colocando una codificación de colores y etiquetas en cada banda de frecuencias. El cuadro TNAF muestra la misma información que la gráfica, pero en un formato de tabla agregando la información de la UIT y los pies de página que aplican a la banda.

Los pies de página TNAF contienen descripción y detalles técnicos de tecnologías que pueden utilizarse dentro del territorio nacional, también los adoptados internacionalmente.

3.2.2.2. Banda 902-928MHz Guatemala

Anteriormente se acordó la utilización de la banda US915 para la implementación de la red LoRaWAN; Estados Unidos reconoce esta banda de frecuencia como ISM y de libre utilización. Sin embargo, para cumplir con las regulaciones de Guatemala se tiene que verificar que esta banda pueda utilizarse para este fin.

Figura 52. **Banda 902-928Mhz Cuadro TNAF**

<p>902 - 928 FIJO MÓVIL TERRESTRE Radiolocalización</p>	<p>Reg</p>	<p>RR 5.150 / 5.325A GTM-6 / GTM-14 / GTM-18 GTM-47</p>
---	------------	---

Fuente: Superintendencia de Telecomunicaciones. *Tabla nacional de atribución de frecuencias radioeléctricas (Cuadro)*. <https://bit.ly/34OHliy>. Consulta: 2 de febrero de 2021.

Según el Cuadro TNAF la banda de 902-928Mhz es una banda regulada para la utilización en dispositivos terrestres fijos y móviles o radiolocalización y dentro de los pies de página se presenta el identificador GTM-6 en cuyo apartado del documento Pies de Página TNAF se escribe literalmente.

Se reconoce la operación de sistemas y/o dispositivos designados para aplicaciones industriales, científicas y médicas (ICM). Estos sistemas y/o dispositivos realizan su operación en una base de No Interferencia / No Protección respecto a las autorizaciones hechas a estaciones a las cuales los rangos de frecuencia citados están atribuidos. La potencia máxima efectiva radiada de transmisión que se utilice en estas bandas de frecuencia por tales dispositivos no excederá de 250 mW (23,98 dBm). Así mismo, deberán adoptarse cuantas medidas prácticas sean necesarias para que la radiación de los equipos destinados a aplicaciones ICM sea mínima y para que, especialmente fuera de las bandas indicadas, el nivel de radiación sea tal que no cause interferencia perjudicial a otras estaciones, en particular, a un servicio de radionavegación o cualquier otro servicio de seguridad que funcione de acuerdo con el Reglamento de Radiocomunicaciones.¹⁹

El estado de Guatemala reconoce que este rango de frecuencias puede ser utilizada como ICM por lo tanto es adaptable a la red LoRaWAN US915MHz.

¹⁹ Superintendencia de Telecomunicaciones. *Tabla nacional de atribución de frecuencias radioeléctricas*. [https://sit.gob.gt/download/872/tabla-nacional-de-atribucion-de-frecuencias-frecuencias/1zKgbuwsQuK3ytp3hprvQcljXxX2WyCbd/4.%20Tabla%20Nacional%20Atribuci%C3%B3n%20Frecuencias%20\(Pies%20de%20P%C3%A1gina\).pdf](https://sit.gob.gt/download/872/tabla-nacional-de-atribucion-de-frecuencias-frecuencias/1zKgbuwsQuK3ytp3hprvQcljXxX2WyCbd/4.%20Tabla%20Nacional%20Atribuci%C3%B3n%20Frecuencias%20(Pies%20de%20P%C3%A1gina).pdf).

3.2.2.3. RIABAP

Como parte del avance tecnológico y tendencias actuales de internet de las cosas, la Superintendencia de Telecomunicaciones de Guatemala en diciembre de 2019 aprobó la resolución SIT-DSI-349-2019²⁰ en donde se establece un tipo de redes llamadas Redes Inalámbricas de Área Extensa Usando Baja o Muy Baja Potencia (RIABAP).

Según el documento de disposiciones técnicas de RIABAP se le define a esta red como una red inalámbrica que cubre áreas extensas que pueden ser superiores a los mil metros cuadrados donde se pueden tener una o más radio bases que recogen la información de dispositivos terminales en periodos de tiempo muy cortos con una potencia muy baja.

Según la definición de RIABAP, se puede catalogar como una LPWAN de la cual se habla anteriormente y donde una red LoRaWAN se adapta completamente a este concepto.

En este documento también se especifica que este tipo de red es únicamente para dispositivos de internet de las cosas y se deberá de cumplir con los siguientes parámetros técnicos.

Tabla XIII. **Parámetros técnicos RIABAP**

Parámetro	RIABAP
Ancho de banda máx.	500 KHz
<i>Payload</i> máximo	40 bytes

²⁰ Superintendencia de Telecomunicaciones. *Resolución SIT-DSI-349-2019.*
[https://sit.gob.gt/2019/09/11/resolucion-sit-dsi_349-2019/.](https://sit.gob.gt/2019/09/11/resolucion-sit-dsi_349-2019/)

Continuación de la tabla XIII.

Tiempo en aire máx.	400ms
Potencia de transmisión máx.	Uplink: 26,99 dBm Downlink: 36,00 dBm
Comunicación	M2M
Restricción	Voz, video

Fuente: Super Intendencia de Telecomunicaciones. *Aprobado por Resolución: SIT-DSI-349-2019.* <https://sit.gob.gt/download/878/google-driver/13JcH1uvYtEQgegwa5GbcccqXyjQPU7Hil/DISPOSICIONES%20TECNICAS%20-%20RIABAP.pdf>. Consulta: 3 de febrero de 2021.

Para la implementación de estas redes no es necesario pagar un título de usufructo de la frecuencia, pero si se establece que se deberá realizar una inscripción ante la SIT de las radio-bases (*gateways*) instaladas y realizar una actualización anual de la red, esto con costos administrativos.

Además, esta inscripción no garantiza la protección contra interferencias de parte de la Superintendencia de Telecomunicaciones. Entre las bandas que pueden utilizarse para RIABAP se encuentra la 902-928Mhz.

3.3. Servidor LoRaWAN

Dentro del modelo de red LoRaWAN (véase figura 45) se encuentra el Servidor LoRaWAN el cual es el núcleo de la red. Se trata de un servidor que integra servicios para el manejo de la red, autenticación y direccionamiento de los datos hacia otras aplicaciones en internet y enrutamiento de paquetes entre dispositivos.

En este servidor se encuentran instalados un conjunto de servicios que realizan las tareas anteriormente mencionadas, pero, los servicios con mayor relevancia para el protocolo LoRaWAN son: Servicio MQTT, Servicio Network Server, Servicio Application Server.

La LoRa-Alliance establece las reglas de comunicación entre dispositivos finales, *gateways* y servidor LoRaWAN, sin embargo, no desarrollan el software para poner en práctica estas normas. Para esto empresas privadas, comunidades o desarrolladores independientes implementan paquetes que contienen el *stack* del protocolo.

Estas entidades ponen al servicio del mercado soluciones en nube o licencias para la utilización de su *software*. También existen organizaciones que brindan este servicio de forma gratuita, The Things Network es la red global pública más grande del mundo y pone a disposición un servidor LoRaWAN al cual cualquiera puede conectar *gateways* y dispositivos finales y redirigirlos a otras aplicaciones en internet de forma gratuita. Una observación muy importante es que el servidor LoRaWAN no incluye el servicio de almacenamiento de datos enviados por los sensores. Pero si tiene la posibilidad de redireccionarlos hacia algún servicio de base de datos o cualquier otra aplicación. Este servidor es puramente de red, administración y autenticación de dispositivos finales y *gateways*.

Para la implementación de una red privada se utiliza el servicio llamado Chirpstack, el cual es de código abierto y con el cual se tiene el control total de la red haciendo que únicamente los *gateways* y nodos propios se conecten al servidor.

3.3.1. Chirpstack

Chirpstack es un conjunto de servicios con que interactúan para cumplir las reglas del protocolo LoRaWAN, este software es *open-source* y fue desarrollado por el ingeniero holandés independiente Orne Brocaar y la primera versión fue publicada en 2015 y ha ido evolucionando al ritmo del protocolo LoRaWAN.

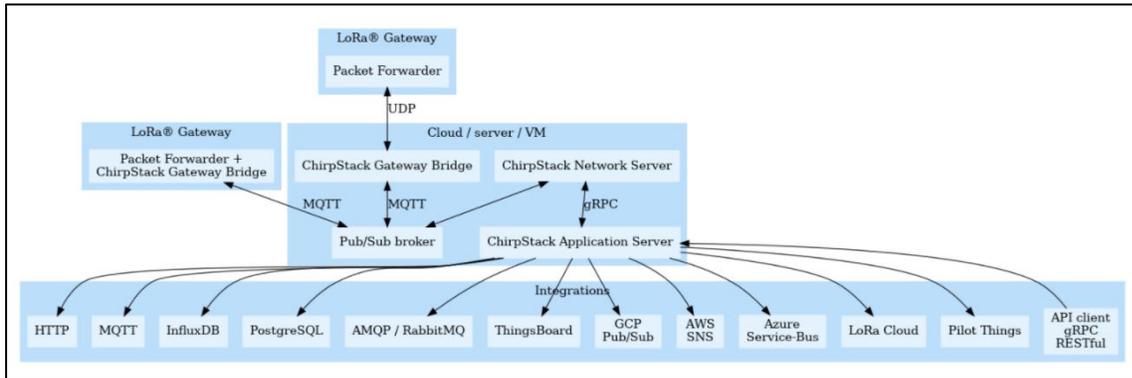
Chirpstack cuenta con una comunidad a inicios de 2021 de más de tres mil usuarios en todo el mundo que brindan soporte y colaboran para la mejora de los servicios.

Estos paquetes son para instalación en un sistema operativo Linux, es muy ligero y cuenta con una interfaz visual para la administración de la red. También tiene una API para manejar la red desde otras plataformas en internet. A continuación, se dará una explicación general de la interacción entre servicios de chirpstack.io que hay que destacar, no son muy diferentes a lo que ofrecen los demás en el mercado.

3.3.1.1. Arquitectura de Chirpstack.io

Como ya se ha establecido, la función principal del servidor LoRaWAN es administrar la red, dispositivos, *gateways*, también brindar seguridad a la red y transmitir los datos hacia otras aplicaciones en internet.

Figura 53. **Arquitectura Chirpstack.io**



Fuente: ChirpStack. *ChirpStack architecture*. <https://www.chirpstack.io/project/architecture/>.

Consulta: 4 de febrero de 2021.

En la figura 49 se observa cómo interactúan los distintos servicios del servidor teniendo como entrada los *gateways* y varias salidas para las integraciones a aplicaciones en internet. Es muy importante mencionar que la dirección es en dos vías haciendo de LoRaWAN una red poderosa que puede acoplarse a otras redes de iot o cualquier producto que funcione con internet.

Por ejemplo, un producto de un botón WiFi puesto en una fábrica en China que encienda y apague una bombilla LoRaWAN en un área rural en Guatemala. Obviamente no se sabe qué fines de negocio tendría este ejemplo, pero da una idea de qué tan poderoso es LoRaWAN como protocolo de IOT y de la capacidad de adaptación a otras tecnologías.

Tomando nuevamente el ejemplo anterior, el funcionamiento podría ser que la bombilla mande su estado actual cada minuto, esta información es transmitida vía LoRa hacia el *gateway*, este transforma el paquete RF a IP por medio del servicio Packet Forwarder.

En el servidor el servicio Chirpstack Gateway Bridge toma los datos del dispositivo y la metadata agregada por el *gateway* y lo publica en un bróker MQTT en donde está suscrito el servicio de Chirpstack Network Server donde se validan las llaves de cifrado de los datos y si todo es correcto muestran la información en la interfaz de administración ubicada en el servicio Chirpstack Application Server y redirigen los datos hacia las distintas integraciones en internet.

En el caso del botón, podría tener un *endpoint* HTTP que reciba los datos del estado de la bombilla y responder con un nuevo estado haciendo el envío de datos en dirección contraria.

Chirpstack utiliza una base de datos *postgresql* para almacenar perfiles de gateways, dispositivos, llaves, administradores de la red y algunos otros valores para el manejo de la red, se recuerda que se menciona anteriormente que este servidor no almacena mediciones realizadas por los sensores, este servicio es una capa de red.

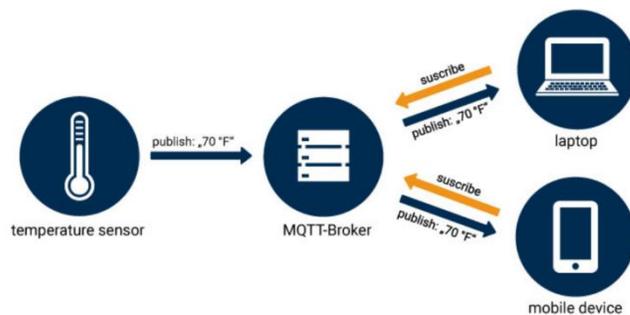
3.3.1.2. Broker MQTT

El protocolo MQTT Message Query Telemetry Transport por sus siglas en inglés, fue diseñado inicialmente para una aplicación petrolera de monitoreo remoto en 1999 por ingenieros de IBM, actualmente es el protocolo más utilizado en internet de las cosas por la ligereza y modelo de publicación suscripción. (HiveMQ)

Ser un protocolo ligero hace que sea ideal para dispositivos con poca capacidad de hardware y que por lo regular envían datos puntuales de una variable con necesidad de un ancho de banda reducido. Diferenciando con http

que tendría un encabezado e información del protocolo, MQTT solamente publica la carga útil bajo un tópico y es procesado por el bróker MQTT velozmente.

Figura 54. **Modelo MQTT**



Fuente: OPC Router. *What is MQTT? A practical introduction.* <https://www.opc-router.com/what-is-mqtt/>. Consulta: 6 de febrero de 2021.

Hay cinco conceptos importantes en el protocolo MQTT, el cliente, el bróker y el tópico, una publicación y una suscripción. El cliente MQTT es la entidad que recibe o transmite datos a otros clientes, estas transmisiones se hacen por medio del bróker MQTT quien es el ente central del modelo y quien distribuye la información entre los clientes.

El modelo funciona cuando un cliente publica un mensaje bajo un tópico en el bróker, inmediatamente todos los clientes que estén suscritos a ese tópico en específico recibirán el mensaje. Un cliente puede publicar y estar suscrito a uno o varios tópicos.

En el ejemplo de la figura 50 se tiene un sensor de temperatura que publica en el bróker el valor 70 grados bajo un tópico temperatura, la temperatura y el

celular tienen instalado un cliente MQTT que está suscrito a este mismo tópico, por lo tanto, reciben el valor de temperatura.

Hay desarrolladores que ponen a disposición *software* de bróker o cliente como servicio en la nube o para implementarse con recursos propios. El bróker más popular *open source* es Mosquitto, el cual fue desarrollado por Eclipse y puede instalarse en sistemas operativos Linux, IOS y Windows.

Los servicios de Chirpstack pueden utilizar cualquier bróker, pero por ser *open source* se utiliza Mosquitto y el puerto usado para este servicio es el 1883.

En el caso del servidor LoRaWAN cada *gateway* es un cliente MQTT que publica y está suscrito a tópicos de la red. En caso de que los *gateways* no tengan instalado un cliente MQTT, se comunicarán hacia el servicio Chirpstack Gateway Bridge el cual funcionará como puente entre el bróker y el *gateway*.

Otro de los clientes es el servicio Chirpstack Network Server quien está suscrito a todos los dispositivos registrados y recibirá todos los mensajes provenientes de estos. Después de la autenticación también el bróker puede ser utilizado para conexión hacia otras aplicaciones por el servicio Chirpstack Application Server.

3.3.1.3. Chirpstack Gateway Bridge

Este servicio como bien lo dice su nombre es un puente entre los *gateways* que envían datos con el formato de Packet Forwarder establecido por Semtech al bróker MQTT. Esto se da para que se tenga compatibilidad entre los tópicos que conoce el Network Server.

3.3.1.4. Chirpstack Network Server

Este componente es el principal de LoRaWAN, tiene inmerso un cliente MQTT que está suscrito a un tópico con un identificador único de cada dispositivo final y cada *gateway* en la red.

Para que los datos de un dispositivo final puedan procesarse por este servicio deben de contener la llave conocida como Network Session Key que se genera en este servicio y se traslada al dispositivo manualmente en la autenticación ABP o el dispositivo primero genera una petición JOIN al servidor y este le devuelve el Network Session Key para las transmisiones, a este último se le conoce como autenticación OTAA.

El servicio Chirpstack Network Server también elimina los duplicados de los paquetes que le llegan de varias antenas, otra de sus tareas es identificar qué *gateway* es mejor para devolverle un mensaje al dispositivo y mantiene el paquete en cola.

LoRaWAN es un protocolo de red de internet de las cosas por lo que pueden existir múltiples aplicaciones empleando la misma red. Para identificar a qué aplicación pertenece cada dispositivo, existe una capa de aplicación la cual es accesible por medio de la llave Application Session Key, también generada en este servidor por medio de la autenticación ABP u OTAA.

3.3.1.5. Chirpstack Application Server

Después de pasar la fase del Network Server, está el servicio de Chirpstack Application Server, el cual sirve para comunicar cada aplicación del servidor LoRa hacia una o varias aplicaciones en internet.

En este servicio también se integra una página web para el manejo de dispositivos finales, gateways, aplicaciones y también manejar distintas organizaciones y usuarios administradores.

Chirpstack también añade una API mediante este servicio para manejar toda la red desde una aplicación externa si se desea.

3.3.1.6. Integraciones

Chirpstack ha implementado software para conectarse a distintos servicios populares, se recuerda que el Internet of Things no solamente se limita a monitorear los dispositivos, también es posible controlarlos mediante decisiones basadas en análisis, por lo cual es necesario buscar la forma de conectar la red hacia estos servicios en nube o desarrollos propios.

Por defecto, el método más adecuado para conectar la red es HTTP y MQTT, además Chirpstack puede integrar con AMQP / RabbitMQ, Amazon SNS, Azure Service Bus, GCP Pub/Sub, InfluxDB, Kafka, LoRa Cloud, myDevices, Pilot Things, PostgreSQL y Thingsboard.

3.3.2. Implementación de servidor LoRaWAN

Para la instalación de los servicios en el servidor LoRaWAN se necesita un servidor con sistema operativo Linux, por lo que se utiliza la versión de Ubuntu 18.04.5 LTS sin interfaz gráfica y acceder vía ssh a la consola. Los servicios son muy ligeros por lo que 2 GB de RAM y 20 GB de disco son suficientes.

Estando en la consola del sistema operativo, se instalarán los prerequisites para el servidor, los cuales son: MQTT bróker, PostgreSQL, RedisServer y se logra con estos comandos:

```
sudo apt-get update
sudo apt install mosquitto
sudo apt-get install postgresql
sudo apt-get install redis-server
```

Posterior se tiene que configurar los roles de la base de datos PostgreSQL y crear 2 esquemas para el almacenamiento de perfiles de dispositivos y usuarios administradores de la red entrando a la consola de postgres con:

```
sudo -u postgres psql
```

Estando en la consola de PostgreSQL se crean los roles y contraseñas que se utilizaran, muy importante anotar estas credenciales. Posteriormente las bases de datos:

```
CREATE ROLE chirpstack_as WITH login password 'dbpassword';
CREATE ROLE chirpstack_ns WITH login password 'dbpassword';
CREATE DATABASE chirpstack_as WITH OWNER chirpstack_as;
CREATE DATABASE chirpstack_ns WITH OWNER chirpstack_ns;

\c chirpstack_as
CREATE extension pg_trgm;
CREATE extension hstore;
\q
```

Luego, se debe obtener el repositorio de Chirpstack para realizar la instalación, para esto se debe tener el complemento apt-transport-https en caso de no estar instalado, instalarlo con este comando.

```
sudo apt install apt-transport-https dirmngr
```

Se configura una llave para el nuevo repositorio y se descarga.

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys  
1CE2AFD36DBCCA00  
sudo echo "deb https://artifacts.chirpstack.io/packages/3.x/deb stable main" |  
sudo tee /etc/apt/sources.list.d/chirpstack.list  
sudo apt update
```

El primer servicio para instalar será el Chirpstack Gateway Bridge que se comunica con los gateways, posteriormente el Chirpstack Network Server y Chirpstack Application Server.

```
sudo apt install chirpstack-gateway-bridge  
sudo apt install chirpstack-network-server  
sudo apt install chirpstack-application-server
```

Estos servicios vienen configurados por defecto para instalarse en la misma instancia, así como la base de datos, bróker MQTT y REDIS. En caso de tenerlas instaladas en varias instancias y de añadir contraseñas entre servicios, los archivos de configuración se pueden encontrar en:

```
cd /etc/chirpstack-gateway-bridge/chirpstack-gateway-bridge.toml  
cd /etc/chirpstack-network-server/chirpstack-network-server.toml  
cd /etc/chirpstack-application-server/chirpstack-application-server.toml
```

El servicio de Chirpstack Gateway Bridge trae una configuración por defecto que es suficiente para el sistema. Chirpstack Network Server se tiene que

configurar con la banda de frecuencia que se utilizará en la red e ingresar las credenciales de la base de datos PostgreSQL creada para este servicio. Para abrir el archivo de configuración se utiliza:

```
nano /etc/chirpstack-network-server/chirpstack-network-server.toml
```

Se modifica el archivo con la siguiente configuración. Como bien se mencionó anteriormente, se utilizará la subbanda 1 de la región US915 MHz la cual comprende los canales del 8 al 15 y 65.

```
[general]  
log_level=4
```

```
[postgresql]  
dsn="postgres://chirpstack_ns:dbpassword@localhost/chirpstack_ns?sslmode=  
disable"
```

```
[network_server]  
net_id="000000"
```

```
[network_server.band]  
name="US_902_928"
```

```
[network_server.network_settings]  
enabled_uplink_channels=[8, 9, 10, 11, 12, 13, 14, 15, 65]
```

Se guarda y pasa a configurar el Chirpstack Application Server entrando al archivo de configuración, pero, antes se debe de generar una llave secreta para la API empleando openssl.

```
openssl rand -base64 32
```

Se copia la llave generada y se abre el archivo de configuración.

```
nano /etc/chirpstack-application-server/chirpstack-application-server.toml
```

Se edita con las credenciales de la base de datos creada para este servicio y la llave generada en el espacio "verysecret"

```
[general]  
log_level=4
```

```
[postgresql]  
dsn="postgres://chirpstack_as:dbpassword@localhost/chirpstack_as?sslmode=  
disable"
```

```
[application_server.external_api]  
jwt_secret="verysecret"
```

Después de la configuración se podrán encender los servicios para que empiece a funcionar la red LoRaWAN a frecuencia US915 MHz.

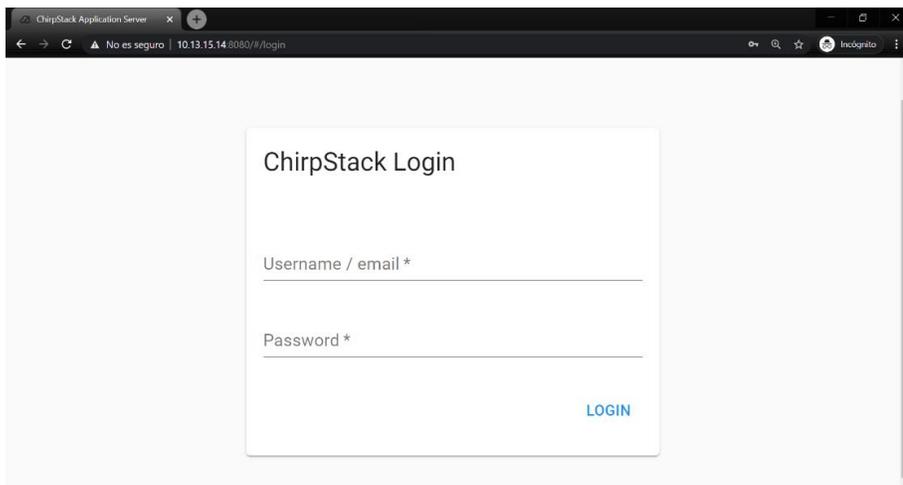
```
sudo systemctl start chirpstack-gateway-bridge  
sudo systemctl enable chirpstack-gateway-bridge  
sudo systemctl start chirpstack-network-server  
sudo systemctl enable chirpstack-network-server  
sudo systemctl start chirpstack-application-server  
sudo systemctl enable chirpstack-application-server
```

Para verificar el log de los servicios se puede utilizar el siguiente comando.

```
sudo journalctl -f -n 1 -u chirpstack-gateway-bridge  
sudo journalctl -f -n 1 -u chirpstack-network-server  
sudo journalctl -f -n 1 -u chirpstack-application-server
```

Al momento de encender los servicios la red estará lista para agregar dispositivos, *gateways*, aplicaciones, manejar usuarios y perfiles por medio de la página web. Se puede ingresar desde un equipo en la misma red con un explorador mediante la IP de la instancia, seguido del puerto 8080 de la forma `http://host:8080`. Es posible cambiar el puerto en el archivo de configuración del Chirpstack Application Server, también agregarle certificados de seguridad y publicar el servicio a una IP pública para que la red sea accesible desde cualquier parte del mundo. Si es una red pequeña se recomienda mantenerla local para no ser víctima de ataques que puedan dañar la aplicación.

Figura 55. **Login Chirpstack**



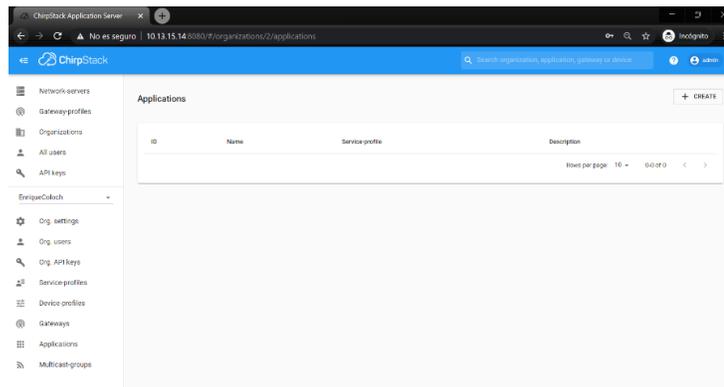
Fuente: elaboración propia.

Al ingresar solicitará las credenciales que por defecto son:

- Username: admin
- Password: admin

Al ingresar se debe cambiar la contraseña y realizar algunas configuraciones para agregar *gateways* y dispositivos finales, que en este caso son los pluviómetros.

Figura 56. **Web Chirpstack**



Fuente: elaboración propia.

Para cambiar la contraseña se posiciona en la parte superior derecha y se selecciona `admin > Change password > "Se escribe la contraseña nueva" > UPDATE PASSWORD`.

Posterior a eso se agrega el Chirpstack Network Server a nuestra interfaz, es importante mencionar que el servicio Chirpstack Application Server puede manejar varias redes, en caso de que se tengan dispositivos a distintas frecuencias. Para realizar esta acción se debe ir al menú de la izquierda y seleccionar `Network-servers > +ADD > Se ingresa nombre > Se ingresa dirección IP > UPDATE NETWORK-SERVER`

También en este servidor se puede manejar Organizaciones en caso de que se tengan varias redes y clientes con dispositivos y gateways, para crear una

Organización se debe posicionar nuevamente en el menú y seleccionar Organizations > CREATE > Se ingresan datos > CREATE ORGANIZATION.

Como último paso, se habilita un perfil de servicio, el cual servirá para crear los objetos json que se trasladarán a las integraciones y nuevamente se debe volver al menú y se selecciona Service-profiles > CREATE > Se ingresa el nombre > Se selecciona el Network Server > Se agregan datos que se quieran adjuntar al JSON > UPDATE SERVICE-PROFILE.

Con estos ajustes la red estará preparada para agregar *gateways* y dispositivos.

3.4. Gateway LoRaWAN

Un gateway LoRaWAN es el equipo intermediario entre los dispositivos finales y el servidor LoRaWAN. Este rol de mediador requiere que estos equipos cuenten con múltiples interfaces de acceso a internet y por su puesto una interfaz LoRa para comunicarse con los nodos.

Describiendo brevemente la estructura de *hardware* de estos equipos, se puede mencionar que contienen una interfaz de LoRa con capacidad de escuchar varios canales en simultáneo, actualmente los fabricantes tienen versiones con capacidad de 8 y 16 canales. El criterio para elegir la capacidad se basa en la cantidad de dispositivos que estará atendiendo, se debe recordar que un canal es ocupado por un nodo durante un tiempo corto y al finalizar la transmisión dejará el canal libre para que otros dispositivos lo puedan utilizar. Pero si la cantidad de nodos es muy grande, algunos pueden tener problemas en encontrar un canal disponible.

Con esta explicación se ve porque se le ha dado mucha importancia al tiempo que dura una transmisión y lo mucho que hay que considerar el espectro del tiempo cuando se implementa una red LoRaWAN para alguna aplicación. En el caso de los pluviómetros digitales, la concentración en un área es muy rara por lo cual un gateway de 8 canales será suficiente para atender a varios pluviómetros.

Para las interfaces que sirven para la comunicación con el servidor, los *gateways* tienen controladores ETHERNET, WIFI y LTE. No siempre tienen los tres y el diseñador de la red deberá escoger la adecuada, considerando bajo qué vía tiene acceso en el lugar donde será instalado.

La comunicación con el servidor funciona empleando el modelo de red TCP/IP por lo que el gateway deberá tener visibilidad con el servidor.

Los concentradores también incluyen un módulo GPS para determinar su ubicación, cabe mencionar que LoRaWAN hace compatible un servicio de geolocalización de nodos por medio de trilateración basada en la posición de varios gateways que reciben un mismo paquete, en este caso no se utilizará este servicio.

Los gateways realizan la conversión de datos por medio del servicio LoRa Packet Forwarder el cual es un estándar establecido por Semtech y como protocolo de transporte utiliza UDP y por lo regular los sistemas operativos son basados en Linux.

3.4.1. Conexión de *gateway* a la red LoRaWAN

Para este proyecto se utilizará un *gateway* de la marca URSALINK modelo UG87. Este aparato tiene grado de protección IP67 por lo que se podrá instalar en exteriores, para una mayor cobertura se instalará a 35 metros de altura.

Específicamente el modelo de *gateway* a utilizar es UG87-L00AU-G-P-915 con características de alimentación con PoE Power over Ethernet por sus siglas en inglés, con interfaces de acceso a internet vía Ethernet y LTE y con la banda de frecuencias 902 a 928 MHz.

Este modelo soporta 8 canales LoRa, con sensibilidad de -140 dBm y potencia de transmisión de +27 dBm. También cabe mencionar que, en caso de implementar soluciones a pequeña escala, con un solo *gateway* este modelo tiene integrado un *software* de servidor LoRaWAN. En este caso, pensando en una red amplia escalable con varios *gateways* y dispositivos, solo se utilizará como puente hacia el servidor ya configurado.

Este equipo incluye el servicio de Chirpstack Gateway Bridge por lo que es posible conectarlo directamente al servicio de Chirpstack Network Server que instaló en nuestra instancia.

Figura 57. **Instalación de Gateway UG87**

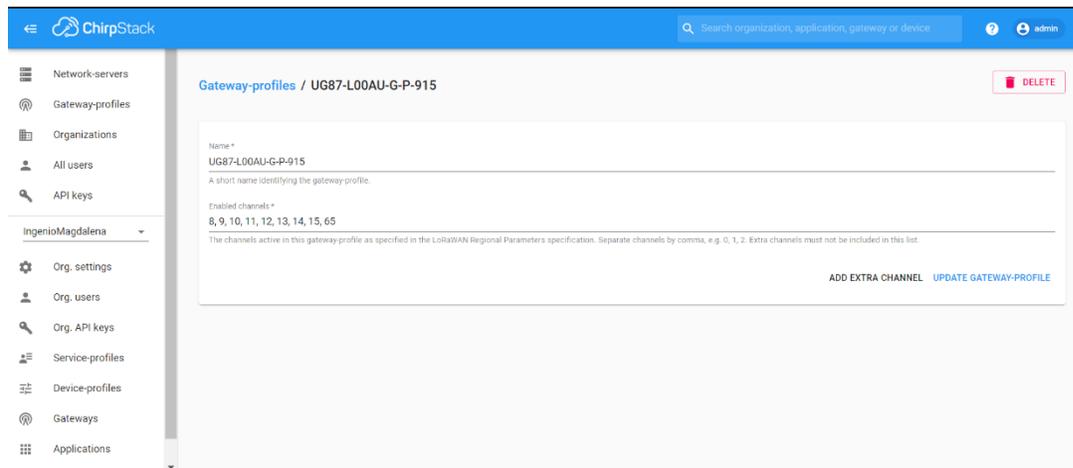


Fuente: elaboración propia.

Para la conexión del gateway en el servidor Chirpstack inicialmente debe crear un perfil de *gateway*, ya que es posible que se cuente con equipos de distintas marcas. Los pasos para realizar este perfil se realizan en la interfaz web siguiendo los siguientes pasos. Login en Chirpstack > Gateway-profiles > CREATE > Se ingresa un nombre > Se habilitan los canales que acepta.

En este paso se hace una pausa para agregar los mismos canales que están agregados en la configuración de este servidor, los cuales son 8,9,10,11,12,13,14,15,65. Posterior a esto se selecciona el network server y se crea en CREATE GATEWAY-PROFILE.

Figura 58. Creación de perfil de Gateway UG87



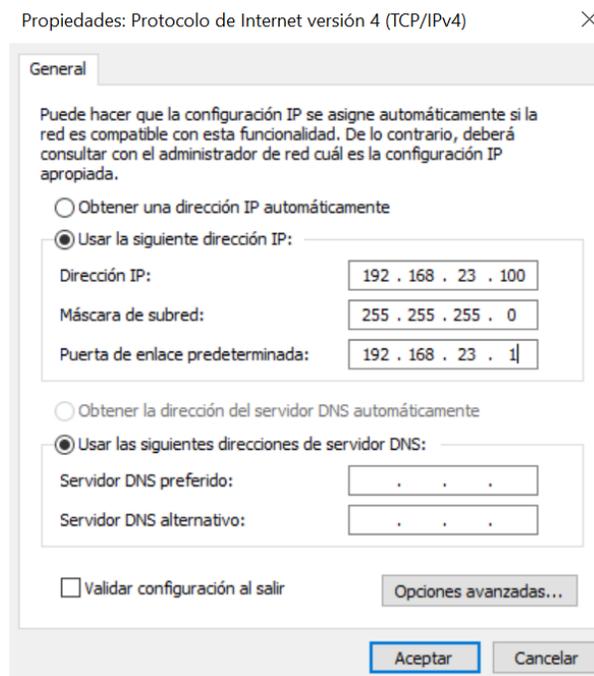
Fuente: elaboración propia.

Se procede a agregar el *gateway* a la red y para esto se necesita el GatewayID el cual es un identificador único, para obtenerlo basta con buscar en la etiqueta. Los pasos para agregarlo a la red son los siguientes, siempre estando en la interfaz web del servidor. Gateways > CREATE > Se ingresa un nombre > Se ingresa descripción > Se inserta el GatewayID > Se selecciona servidor y perfil > Se habilita la opción “Gateway Discovery Enabled” > Se crea en CREATE GATEWAY.

Después de estos sencillos pasos se configura el gateway URSALINK para apuntar hacia el servidor de Chirpstack y configurar los canales de frecuencia. Para esto el gateway también incluye una opción de configuración por interfaz web. Los valores por defecto son IP192.168.23.150, usuario: “admin” y contraseña: “password”.

Así que se conecta el equipo a una computadora con un cable Ethernet, previamente colocando la configuración de interfaz a estática con una IP en la misma red que el gateway.

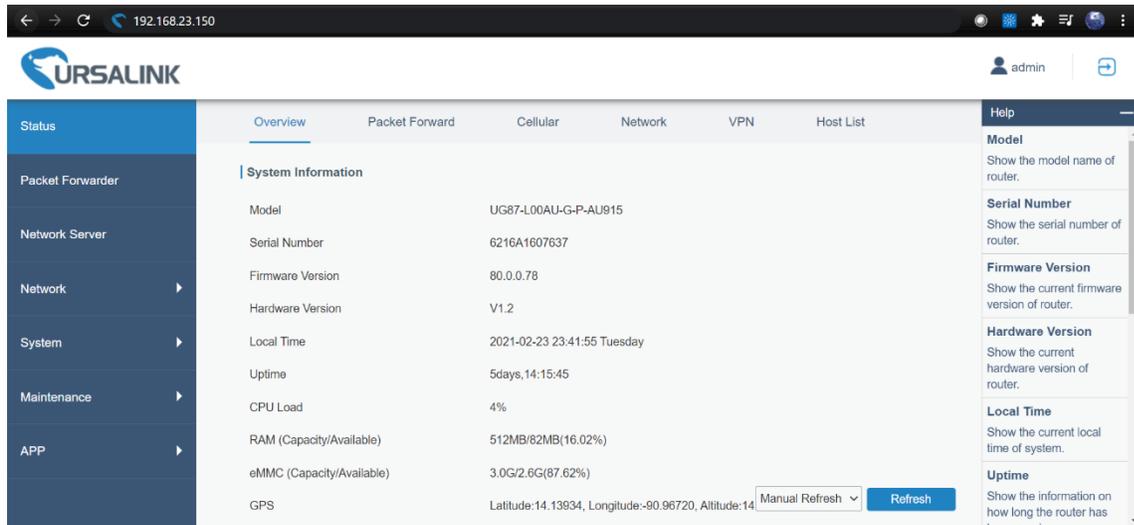
Figura 59. **IP para configuración de Gateway**



Fuente: elaboración propia.

Se ingresa al explorador de la computadora a la IP 192.168.23.150 del gateway y se ingresa con las credenciales anteriormente mencionadas. Estando en la interfaz de configuración se realiza el cambio de contraseña que nos solicita y se procede a configurar el servicio Packet Forwarder.

Figura 60. Interfaz de configuración de Gateway UG87



Fuente: elaboración propia.

Se selecciona Packet Forwarder > Se elige tipo “Chirpstack-Generic” > Se escribe la dirección IP del servidor con el puerto de MQTT > Se selecciona “Enable” > Save.

Figura 61. Configuración de Packet Forwarder Gateway UG87

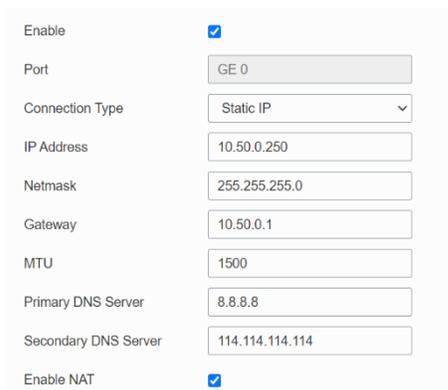
The screenshot shows the configuration form for the Packet Forwarder. The 'Enable' checkbox is checked. The 'Type' dropdown is set to 'ChirpStack-Generic'. The 'Server Address' dropdown is set to '10.13.15.14'. The 'MQTT Port' text input field contains '1883'. The 'User Credentials' and 'TLS Authentication' checkboxes are unchecked. A blue 'Save' button is located at the bottom of the form.

Fuente: elaboración propia.

En el mismo apartado de Packet Forwarder se ingresa a la sección de “Radios” y se verifica que el plan de frecuencias sea US915 y los radios correspondan a los canales que se han establecido para la red. Se guarda y realiza otras configuraciones como la zona horaria y algunas preferencias.

Es importante mencionar que se debe hacer que tanto el *gateway* como servidor se encuentren en la misma red o que tengan conexión. Para esto se cambiará la IP del equipo entrando al apartado de Network > Interface > Se agrega la IP de la red y se guarda.

Figura 62. **Configuración IP de Gateway UG87**



Enable	<input checked="" type="checkbox"/>
Port	GE 0
Connection Type	Static IP
IP Address	10.50.0.250
Netmask	255.255.255.0
Gateway	10.50.0.1
MTU	1500
Primary DNS Server	8.8.8.8
Secondary DNS Server	114.114.114.114
Enable NAT	<input checked="" type="checkbox"/>

Fuente: elaboración propia.

Al momento de aplicar todos los cambios el *gateway* podrá publicar en el servicio MQTT y será reconocido por la red y estará listo para recibir datos LoRa y trasladarlos al servidor.

Para agregar más *gateways* a la red basta con seguir los mismos pasos, esta es una solución altamente escalable debido a que se pueden agregar más *gateways* y también más dispositivos LoRaWAN, manejarlo todo en un solo lugar.

Figura 63. **Habilitación de gateway UG87**

Gateways / G_011_TESIS DELETE

GATEWAY DETAILS | GATEWAY CONFIGURATION | CERTIFICATE | GATEWAY DISCOVERY | LIVE LORAWAN FRAMES

Gateway details	
Gateway ID	24e124ffef0b39a
Altitude	92 meters
GPS coordinates	14.13936, -90.96724
Last seen at	Feb 23, 2021 11:56 PM

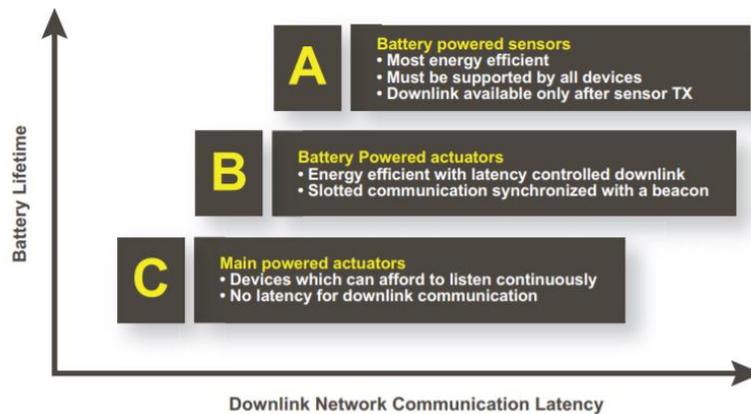
Map showing the location of the gateway (blue pin) near a river and a road labeled RD ESC-2. Leaflet | © OpenStreetMap contributors

Fuente: elaboración propia.

3.5. **Nodos LoRaWAN**

Las aplicaciones de internet de las cosas son diversas y los dispositivos deben de tener diferentes características en estos casos. Para esto LoRaWAN categoriza los dispositivos en tres clases en función al tipo de comunicación de *downlink* que establecen y por consiguiente en lo eficiente que son energéticamente.

Figura 64. Clase de nodos LoRaWAN



Fuente: LoRa Alliance. *LoRaWAN™ What is it?* <https://loro-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf>. Consulta: 2 de febrero de 2021.

Las categorías son A, B y C, cada una establece comunicaciones tipo downlink de forma distinta. La clase A solo puede recibir un paquete desde el servidor cuando apertura dos ventanas de recepción inmediatamente después de cada transmisión tipo *uplink*, es decir que este tipo de nodo solo puede recibir datos cuando él mismo inicia la comunicación y no el servidor.

Por esta razón, como se muestra en la figura 60, este tipo de nodo es utilizado en aplicaciones de sensoramiento donde las comunicaciones más importantes son de tipo *uplink* y son las más eficientes energéticamente.

Los dispositivos de clase B, además de funcionar como los de clase A, también tienen la capacidad de establecer horarios para una transmisión tipo *downlink*, por lo que tienen que estar sincronizados al servidor. Estos son utilizados en aplicaciones con actuadores que tienen acción en tiempos establecidos y energéticamente son menos eficientes que los de tipo A.

Los nodos de clase C están continuamente escuchando transmisiones tipo *downlink*, por lo que no existe latencia entre mensajes enviados desde el servidor hacia el nodo. Esto hace que sean los menos eficientes energéticamente, pero de gran capacidad para responder a eventos en tiempo real.

El pluviómetro de balancín es solamente de sensoramiento y no tiene actuadores, por lo que será un nodo de clase A.

3.5.1. Pluviómetro de balancín en red LoRaWAN

En este apartado se configura la tarjeta de pluviómetro de balancín para enviar datos a través de la red LoRaWAN hacia un servicio de visualización en tiempo real.

Se ha mencionado anteriormente que la red LoRaWAN puede funcionar para múltiples aplicaciones en simultáneo, teniendo dispositivos de distintos tipos comunicándose con el servidor. Es por lo que se creará una aplicación en nuestra red llamada "Medición_lluvia" que englobará todos los dispositivos utilizados para medir la lluvia.

Una vez más en la interfaz web para manejo de la red se ingresa a Applications > CREATE > Se ingresa nombre de la aplicación y descripción > Se selecciona el servicio > CREATE APPLICATION.

Después de crear la aplicación se debe establecer un perfil de dispositivo para el pluviómetro, esta parte es muy importante para que se puedan entender los paquetes provenientes del nodo hacia el servidor. Para crear el perfil de nuestra tarjeta electrónica se selecciona Device-profiles > CREATE > Se ingresa el nombre del perfil > Se selecciona el servidor > La versión de LoRaWAN, la

librería LMIC interpreta la versión 1.0.2 > Se selecciona nodo de tipo A > EIRP máx 14dBm > Se selecciona la pestaña de CODEC > Se selecciona Custom JavaScript codec funtions.

Al llegar a este paso se desplegarán dos apartados para escribir dos funciones de JavaScript, una para decodificar el *payload* enviado que se estableció en la programación de la tarjeta y otro para codificar los paquetes que serán enviados del servidor hacia el dispositivo.

En la programación de la tarjeta de pluviómetro (tabla 8 y 9) se establecieron dos *payloads* enviados por diferentes puertos, uno que contiene los volteos acumulados en el puerto 1 y otro que contiene la ubicación en el puerto 5. Estos datos están codificados por bytes y se deben decodificar decodificarlos para asignarlos a una variable que será enviada al servicio de visualización en tiempo real u otra integración que se desee agregar.

Para el caso de una transmisión tipo *downlink*, no se tiene establecido en la tarjeta cómo decodificar los mensajes recibidos por el servidor por lo que no se llenará este apartado.

La función Decode tiene la forma Decode(fPort, bytes, variables) donde los parámetros son fPort que indica el puerto en que fue enviado el mensaje, bytes el cual es un arreglo que contiene los datos de los sensores y variables que contiene metadata. Esta función devolverá un objeto con formato json con los datos enviados por la tarjeta de pluviómetro y es la siguiente.

```
function Decode(fPort, bytes, variables) {  
  devolver = {}  
  if(fPort==1){
```

```

        var clicks = (bytes[0]<<24) | bytes[1]<<16 | bytes[2]<<8 | bytes[3];
        devolver.clicks = clicks;
    }else if (fPort = 5){
        var lat = (bytes[0]<<24) | bytes[1]<<16 | bytes[2]<<8 | bytes[3];
        var lon = (bytes[4]<<24) | bytes[5]<<16 | bytes[6]<<8 | bytes[7];
        devolver.latitud = lat/1000000;
        devolver.longitud= lon/1000000;
    }
    return devolver;
}

```

Después de ingresar este código se crea el perfil y ya se está listo para vincular la tarjeta de pluviómetro a la red, para esto se ingresa a la aplicación que se creó previamente en Applications > medición_lluvia > CREATE > Se ingresa nombre y descripción > Se genera un Device EUI > Se selecciona el perfil que se creó para el pluviómetro > Se deshabilita la validación de conteo de paquetes > CREATE DEVICE.

Figura 65. Creación de dispositivo en Chirpstack

Applications / medicion_lluvia / Devices / Create

GENERAL VARIABLES TAGS

Device name *
pluviometro_de_balancin_01
The name may only contain words, numbers and dashes.

Device description *
Tarjeta de pluviómetro de balancin

Device EUI *
d3 1c 14 f7 b3 55 5a 2d MSB

Device-profile *
pluviometroEcoloch

Disable frame-counter validation
Note that disabling the frame-counter validation will compromise security as it enables people to perform replay-attacks.

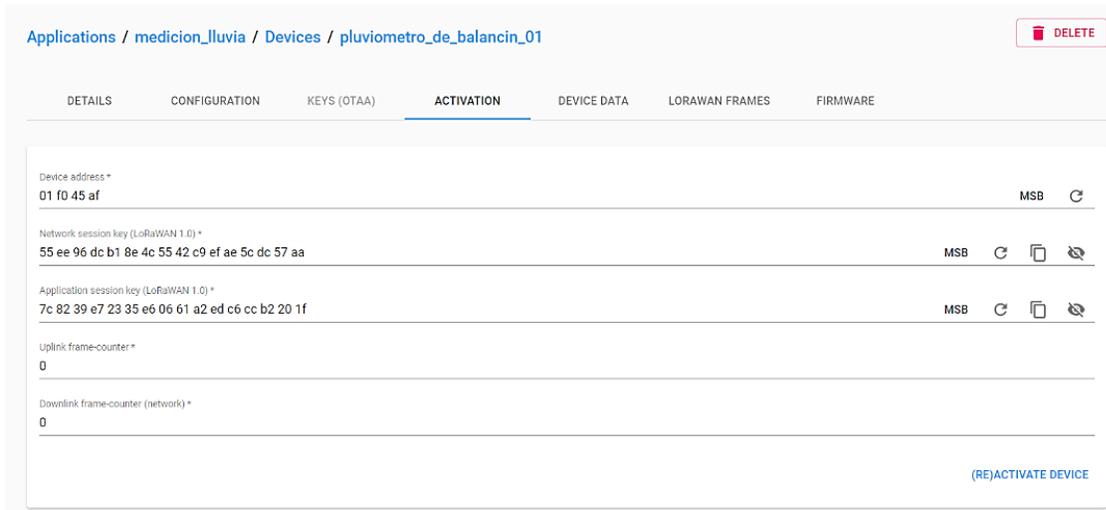
Device is disabled
ChirpStack Network Server will ignore received uplink frames and Join-requests from disabled devices.

CREATE DEVICE

Fuente: elaboración propia, empleando Chirpstack.

Después de crear el dispositivo se abrirá un menú de ACTIVATION que servirá para generar las llaves para que el dispositivo se conecte a la red. Debido a que el dispositivo es de autenticación ABP, se deben generar las llaves y trasladarlas manualmente a la tarjeta. Se genera Device address > Se genera Network session key > Se genera application session key > (RE) ACTIVATE DEVICE.

Figura 66. Autenticación de nodo en Chirpstack



Fuente: elaboración propia, empleando Chirpstack.

Estas claves se deben trasladarse al programa de la tarjeta en el formato aceptado por Arduino.

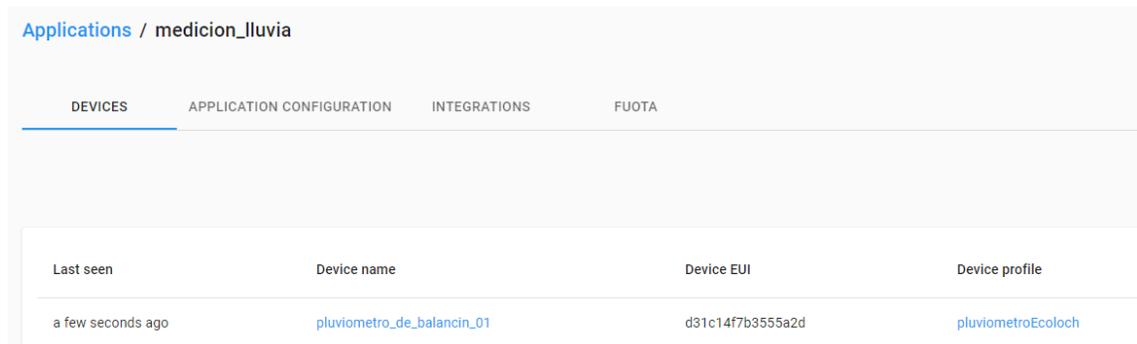
Figura 67. Llaves en programa de tarjeta de pluviómetro

```
// LoRaWAN Network Session Key, llave para autenticación en servidor LoRaWAN
static const PROGMEM ul_t NWKSKEY[16] = { 0x55, 0xee, 0x96, 0xdc, 0xb1, 0x8e, 0x4c, 0x55, 0x42, 0xc9, 0xef, 0xae, 0x5c, 0xdc, 0x57, 0xaa };
// LoRaWAN Application Session Key, llave para autenticación en aplicación LoRaWAN
static const ul_t PROGMEM APPSKEY[16] = { 0x7c, 0x82, 0x39, 0xe7, 0x23, 0x35, 0xe6, 0x06, 0x61, 0xa2, 0xed, 0xc6, 0xcc, 0xb2, 0x20, 0x1f };
// LoRaWAN Device Address, dirección lógica en red LoRaWAN.
static const u4_t DEVADDR = 0x01fd45af ;
```

Fuente: elaboración propia, empleando Chirpstack.

Se compila y traslada el programa con las llaves a la tarjeta de pluviómetro y este iniciará a transmitir datos a la red.

Figura 68. **Dispositivo activo en red LoRaWAN**



The screenshot shows the 'Applications / medicion_lluvia' page in Chirpstack. The 'DEVICES' tab is selected. A table displays the details of an active device:

Last seen	Device name	Device EUI	Device profile
a few seconds ago	pluviometro_de_balancin_01	d31c14f7b3555a2d	pluviometroEcoloch

Fuente: elaboración propia, empleando Chirpstack.

Para poder observar el paquete enviado por el dispositivo, basta con ingresar a la aplicación y al dispositivo y posicionarse en la pestaña DEVICE DATA, acá se mostrarán los paquetes transmitidos.

4. VISUALIZACIÓN DE DATOS

Hasta ahora se ha construido desde el sensor que medirá la variable de lluvia, la tarjeta electrónica que registrará la información y la red por la cual trasladará los datos desde el campo hasta un servidor de red. Queda implementar la capa más alta de nuestra aplicación que es la visualización de datos y la presentación al usuario final.

Para una utilidad de Internet de las Cosas no basta con almacenar los datos en una base de datos y consultarlos cada vez que se necesiten. Para sacarle un mejor provecho se deben presentar los datos de forma visual para que el usuario tenga una herramienta útil para tomar decisiones basadas en datos y en consecuencia optimizar recursos y reducir costos en su actividad.

En este apartado se explicará el funcionamiento de un sistema integrado de servicios para visualización de datos en tiempo real llamado Thingsboard y su implementación para este proyecto.

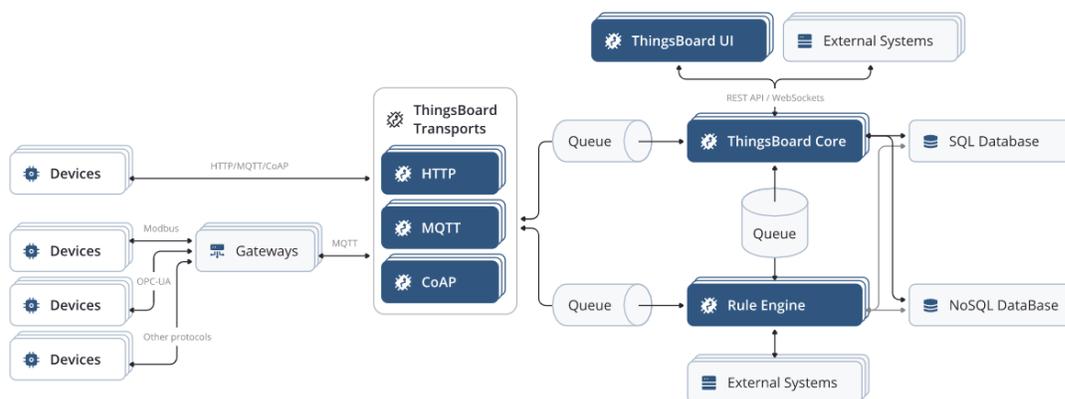
4.1. Thingsboard.io

Thingsboard se define en su página como una plataforma 100% *open source* para manejo de dispositivos, almacenamiento, procesamiento y visualización de datos de aplicaciones de Internet de las Cosas. Es una empresa estadounidense fundada en 2016 por medio de un centro de innovación en Ucrania.

Esta solución para Internet de las Cosas ofrece un conjunto de servicios integrados en una plataforma de rápida implementación para nuevos productos y empresas en el entorno de iot.

Los productos que ofrece son licencias para la utilización del *software* de manera local o en la nube con distintos planes de precios, dependiendo de la cantidad de dispositivos a conectar. También cuentan con una versión de comunidad gratis para instalación en una instancia propia.

Figura 69. **Arquitectura Thingsboard**



Fuente: Community Edition. Thingsboard *architecture*. <https://Thingsboard.io/docs/reference/>.

Consulta: 12 de febrero de 2021.

La arquitectura de servicios de Thingsboard inicia con una capa de transporte que acepta 3 protocolos para ingresar datos HTTP, MQTT y CoAP, estos clientes conectan por medio de servicios de colas hacia las reglas de manejo de datos, visualización y la base de datos.

Para el caso del proyecto de pluviómetro de balancín se utiliza la integración de Chirpstack hacia Thingsboard, la cual utiliza HTTP para el traslado de información.

4.2. Instalación y configuración Thingsboard

Para el proyecto de pluviómetros digitales se utiliza la versión de comunidad instalándolo en una instancia propia con sistema operativo Ubuntu 18.04 LTS. La memoria RAM será de 4 GB y el almacenamiento de 250 GB suficiente para almacenar muchos registros de lluvia.

Después de tener configurada la instancia con la imagen del sistema operativo Ubuntu 18.04LTS se procede a instalar los prerequisites.

```
sudo apt update  
sudo apt install openjdk-8-jdk
```

Se descarga el repositorio de Thingsboard

```
wget  
https://github.com/Thingsboard/Thingsboard/releases/download/v3.2.1/Thingsboard-3.2.1.deb
```

Se instala Thingsboard como un servicio.

```
sudo dpkg -i Thingsboard-3.2.1.deb
```

Se instala e inicializa el gestor de base de datos PostgreSQL.

```
sudo apt install postgresql  
sudo service postgresql start
```

Se utiliza el usuario postgresql creado para el sistema operativo y se ingresa a la consola del gestor de base de datos.

```
sudo su - postgres  
psql
```

Estando en el gestor, se configura el acceso a la base de datos creando una contraseña que deberá ser ingresada después del comando `\password`.

```
\password  
\q
```

Estando en el gestor, se configura el acceso a la base de datos creando una contraseña que deberá ser ingresada después del comando `\password`.

```
\password
```

Se crea una base de datos PostgreSQL llamada Thingsboard, en donde se insertarán todos los datos de dispositivos, usuarios, reglas y todos los datos de los servicios y se vuelve al sistema operativo.

```
CREATE DATABASE Thingsboard;  
\q
```

Después de haber realizado los pasos anteriores, se realiza la configuración de la plataforma y esto se realiza por medio de un archivo de configuración, se abre con la siguiente dirección.

```
sudo nano /etc/Thingsboard/conf/Thingsboard.conf
```

En este archivo se debe agregar las siguientes líneas de configuración de la base de datos creada. No olvidar cambiar el *password* que creado para el gestor y cambiar la ip en caso de que no se haya utilizado la instancia local.

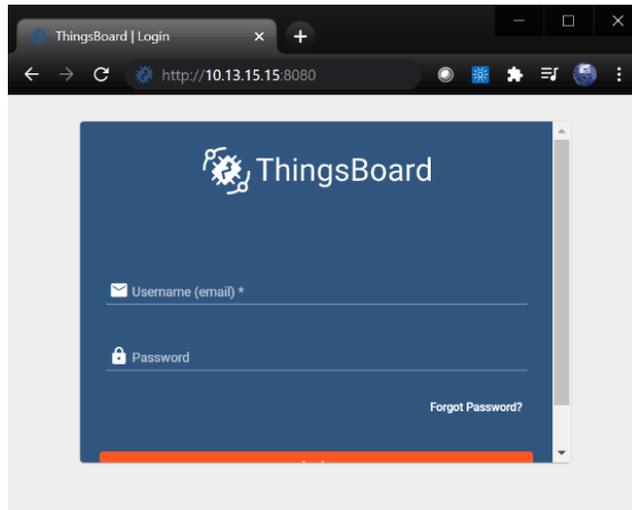
```
# DB Configuration
export DATABASE_ENTITIES_TYPE=sql
export DATABASE_TS_TYPE=sql
export
SPRING_JPA_DATABASE_PLATFORM=org.hibernate.dialect.PostgreSQLDialect
export SPRING_DRIVER_CLASS_NAME=org.postgresql.Driver
export
SPRING_DATASOURCE_URL=jdbc:postgresql://localhost:5432/Thingsboard
export SPRING_DATASOURCE_USERNAME=postgres
export SPRING_DATASOURCE_PASSWORD=PASSWORD
export SPRING_DATASOURCE_MAXIMUM_POOL_SIZE=5
# Specify partitioning size for timestamp key-value storage. Allowed values:
DAYS, MONTHS, YEARS, INDEFINITE.
export SQL_POSTGRES_TS_KV_PARTITIONING=MONTHS
```

Posteriormente se guarda y procede a levantar el servicio de Thingsboard por medio del comando:

```
sudo service Thingsboard start
```

Al tener el servicio de Thingsboard activo se inicializará un servidor web donde se podrá interactuar con la plataforma, configurar dispositivos, reglas y crear dashboards de visualización de datos. La dirección es la IP de la instancia en la que fue instalada en el puerto 8080, en este proyecto <http://10.13.15.15:8080>.

Figura 70. **Thingsboard WEB**



Fuente: elaboración propia.

Solicitará las credenciales para acceder, las cuales por defecto son:

Administrador de plataforma: `sysadmin@Thingsboard.org` / `sysadmin`

Administrador de organización: `tenant@thingsboard.org` / `tenant`

Usuario de cliente: `customer@Thingsboard.org` / `customer`

Thingsboard define estas tres entidades para el manejo de la plataforma, la primera se encarga de la configuración global y creación de organizaciones y sus administradores, pueden existir múltiples empresas teniendo cada una la administración independiente de sus dispositivos, dashboards, reglas, entre otros.

El segundo rol se encarga de la administración de todas las características que se pueden modificar de los dispositivos de la organización del cual es

administrador, así como los clientes finales que cada una tiene para sacarle provecho a los datos.

El usuario de cliente es la persona que tendrá acceso a los *dashboards*, vistas y alertas que les servirán para tomar decisiones basadas en los datos recolectados por los dispositivos iot.

Como configuración general, se cambia el correo electrónico de administrador y contraseña, posteriormente se crea una organización y un administrador de la organización creada.

Para la creación de la organización se ingresa a Tenants > Add Tenant (+) > Se llenan los datos de la organización > ADD.

Figura 71. **Crear organización Thingsboard WEB**

The screenshot shows a modal window titled "Add Tenant" with a close button (X) and a help icon (?). The form contains the following fields and values:

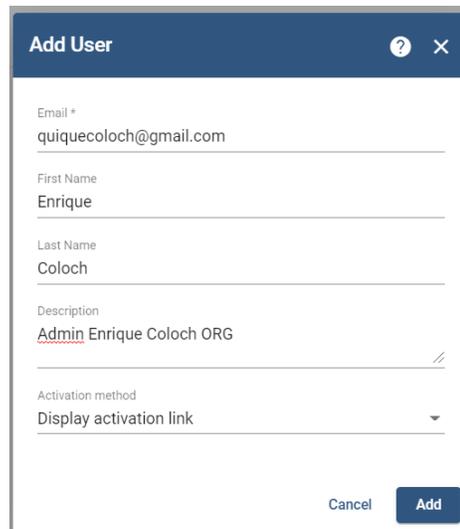
- Title *: Enrique Coloch ORG
- Tenant profile *: Default (with edit and delete icons)
- Description: Enrique Coloch ORG
- Country: Guatemala (dropdown menu)
- City: Guatemala
- State / Province: Guatemala
- Zip / Postal Code: 01212
- Address: Guatemala
- Address 2: (empty)
- Phone: | (empty)

At the bottom right, there are "Cancel" and "Add" buttons.

Fuente: elaboración propia, empleando Thingsboard WEB.

Posteriormente se agrega al administrador de la organización en Tenants > Se ubica la organización > Manage tenant admins > ADD User.

Figura 72. **Crear usuario administrador**



The screenshot shows a modal window titled "Add User". It contains the following fields and values:

- Email *: quiquecoloch@gmail.com
- First Name: Enrique
- Last Name: Coloch
- Description: Admin Enrique Coloch ORG
- Activation method: Display activation link

Buttons: Cancel, Add

Fuente: elaboración propia.

Al agregar al usuario despliega un *link* el cual sirve para crear una contraseña para el nuevo usuario. Se crea la contraseña y ya se puede ingresar al sitio de administración de dispositivos y entorno de la organización.

4.3. Integración de pluviómetros de red LoRaWAN

Habiendo creado la organización y el usuario para administrarla, se procede a integrar los dispositivos de la red LoRaWAN al servicio de almacenamiento y visualización de datos de Thingsboard.

Como primer paso se debe ingresar con el usuario administrador de la Organización y crear un perfil de dispositivo siguiendo el siguiente orden. Device Profiles > Add device profile > Se llenan los campos por defecto > Add. En el campo de transporte se deja la configuración por defecto, la cual es http.

Figura 73. Perfil de dispositivo pluviómetro

The screenshot shows a web form for adding a device profile. The title is "Add device profile". The form is divided into four steps: "Device profile details" (current step), "Transport configuration" (optional), "Alarm rules (0)" (optional), and "Device provisioning" (optional). The "Name" field contains "pluviometro". The "Rule chain" field contains "Root Rule Chain". The "Queue Name" field contains "Main". The "Description" field contains "pluviómetro de balancín tesis". At the bottom right, there is a "Next: Transport configuration" button, a "Cancel" button, and an "Add" button.

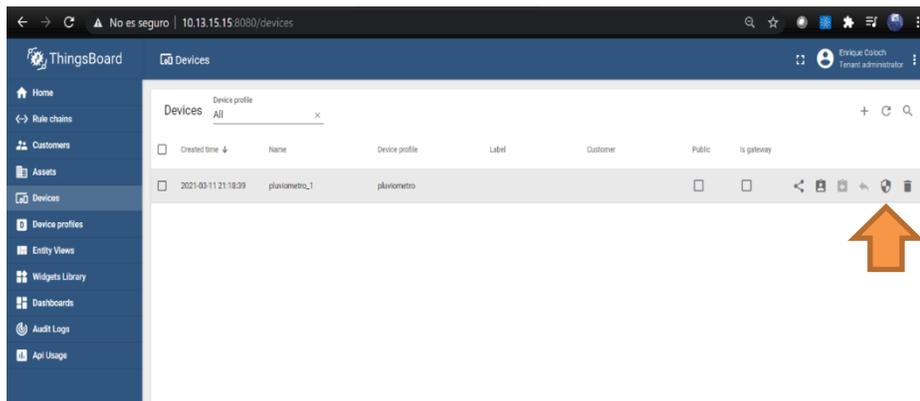
Fuente: elaboración propia, empleando Thingsboard WEB.

Creado el perfil del dispositivo, ya se puede crear la entidad del pluviómetro de balancín para que reciba y almacene datos. Para agregar el nuevo dispositivo se dirige a Devices > Add Device > Add new device > Se ingresa nombre > Se selecciona el perfil anteriormente creado > Las otras opciones por defecto > Add.

En este punto ya estará listo para recibir datos, y ahora se direccionan los datos provenientes de la red LoRaWAN en Chirpstack hacia Thingsboard mediante el protocolo http y un token creado automáticamente para el dispositivo.

Para obtener el token se traslada al dispositivo creado en Thingsboard y se selecciona Manage Credentials y se despliega una ventana con el token el cual se copia, para ser agregado en Chirpstack.

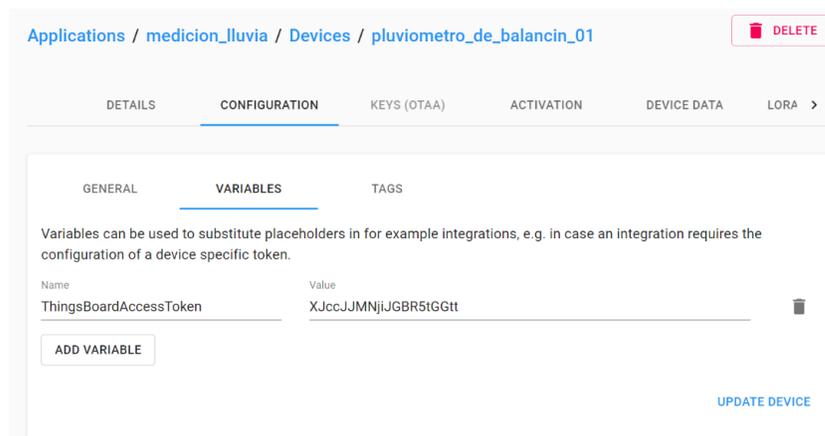
Figura 74. **Credenciales de dispositivo Thingsboard**



Fuente: elaboración propia, empleando Thingsboard WEB.

Se abre la interfaz web del servidor LoRaWAN Chirpstack, luego se dirige al pluviómetro de balancín y en la pestaña de configuración y en las variables se agrega una llamada ThingsboardAccessToken y el valor será el token generado en Thingsboard.

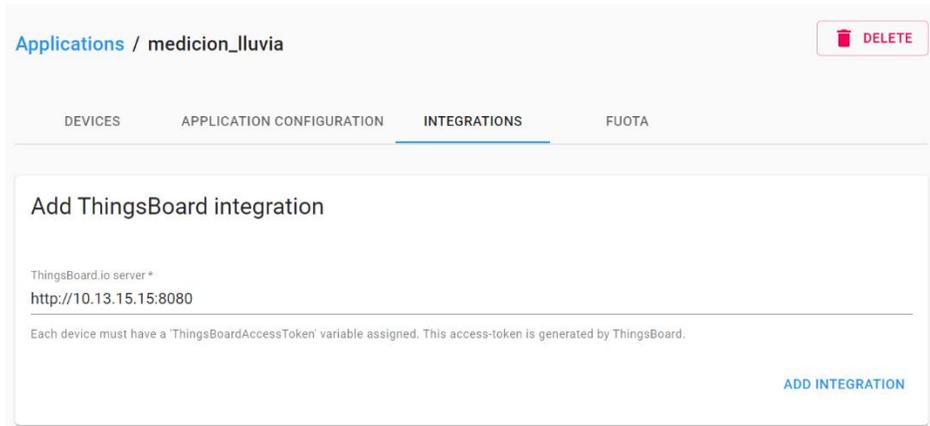
Figura 75. Credenciales en Chirpstack



Fuente: elaboración propia, empleando Chirpstack.

Para trasladar los datos del pluviómetro de balancín desde Chirpstack hacia Thingsboard solo hace falta un paso, el cual es redireccionar el tráfico de la aplicación “medición_lluvia” al servidor Thingsboard. Esto se realiza ingresando a la aplicación y entrando a las opciones de integraciones, de estas opciones se selecciona la de Thingsboard cuya configuración es solamente colocar el servidor con el puerto.

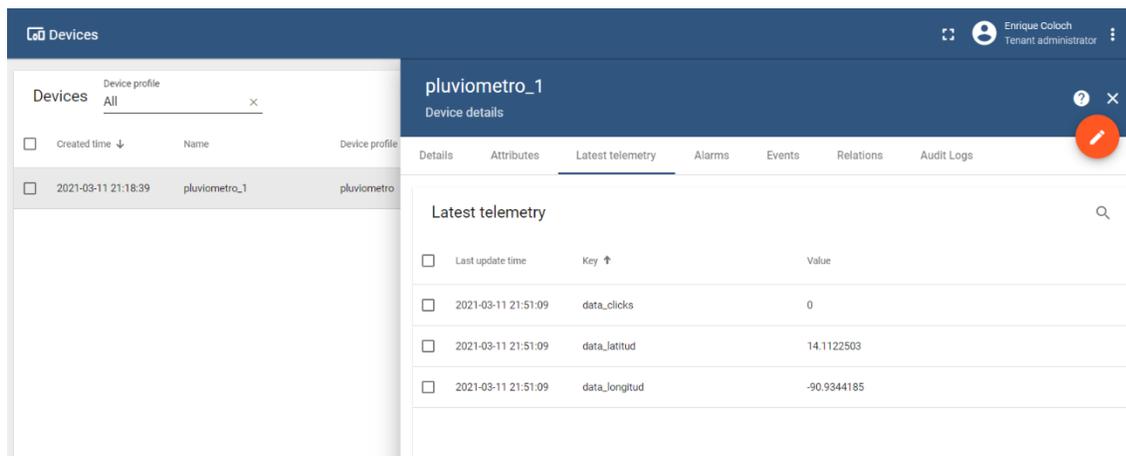
Figura 76. Integración Thingsboard



Fuente: elaboración propia, empleando Chirpstack.

Con estas configuraciones los pluviómetros de la aplicación “medicion_lluvia” trasladarán los datos de los dispositivos hacia Thingsboard y pueden verse los datos en la opción de *Latest telemetry* en tiempo real.

Figura 77. Última telemetría Thingsboard



Fuente: elaboración propia, empleando Dashboard.

4.4. Construcción de dashboard

Una representación gráfica de los datos siempre es de mayor provecho que ver una tabla de valores, ya que por naturaleza el cerebro humano trata de encontrar patrones y tendencias para tomar alguna decisión.

En este apartado se explicará cuál fue el procedimiento tomado para la creación de una pantalla de visualización de datos del pluviómetro de balancín en Thingsboard.

Estando en la interfaz de web de Thingsboard, se dirige al apartado de Dashboards y se crea uno nuevo e ingresa al modo edición.

Figura 78. Edición de Dashboard



Fuente: elaboración propia, empleando Dashboard.

El primer paso para ordenar los datos en el *dashboard* es crear los alias de los cuales se mostrará información. Para crear estas entidades se ingresa a la opción marcada en la flecha verde de la figura 78. Se crea el alias del pluviómetro de balancín, este alias servirá de referencia para mostrar datos en los *widgets* que se agreguen al *dashboard*.

Figura 79. Creación de alias

The image shows a modal window titled "Add alias" with a close button (X) in the top right corner. The form inside has the following fields:

- Alias name ***: A text input field containing "pluviometro". To its right is a toggle switch labeled "Resolve as multiple entities", which is currently turned off.
- Filter type ***: A dropdown menu with "Single entity" selected.
- Type ***: A dropdown menu with "Device" selected.
- Device ***: A text input field containing "pluviometro_1" with a clear button (X) to its right.

At the bottom right of the modal, there are two buttons: "Cancel" and "Add".

Fuente: elaboración propia, empleando Dashboard.

Luego de creado el alias se puede agregar *widgets* para mostrar la información del pluviómetro. En este caso el *dashboard* tendrá una gráfica tipo *Chart* para mostrar datos de lluvia, un mapa para visualizar la ubicación y un log de alertas.

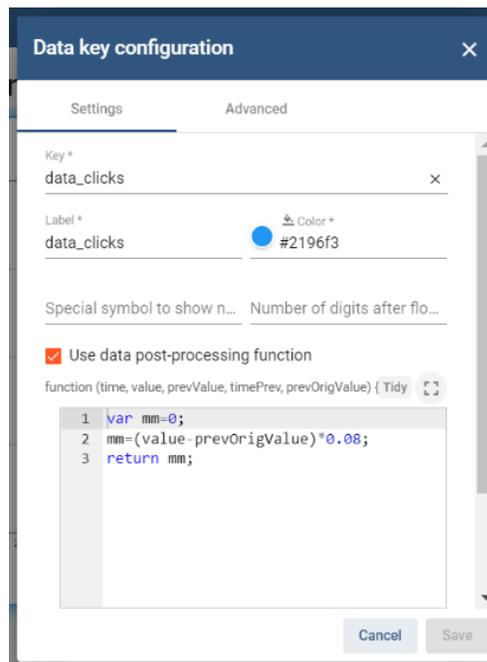
Para configurar la gráfica *Chart*, la se agrega y selecciona el alias creado para el pluviómetro y el dato a graficar será "*clicks*". Se debe recordar que esta variable representa el número de volteos acumulados capturados por el pluviómetro.

Para tener la información de lluvia en milímetros en un período de tiempo, se debe realizar la siguiente operación.

$$\text{milímetros} = (\text{clicksActual} - \text{clicksAnterior}) * 0,08 \text{ [mm]}$$

Donde `clicksActual` representa el número de volteos enviados por el pluviómetro en la transmisión más reciente y `clicksAnterior` representa el número de volteos enviados por el pluviómetro la última vez que transmitió datos. Y el valor 0.08 es el factor de conversión a milímetros por volteo calculado en la ecuación 4. En la función en JavaScript para el gráfico quedaría de la siguiente forma.

Figura 80. **Función de cálculo de milímetros**

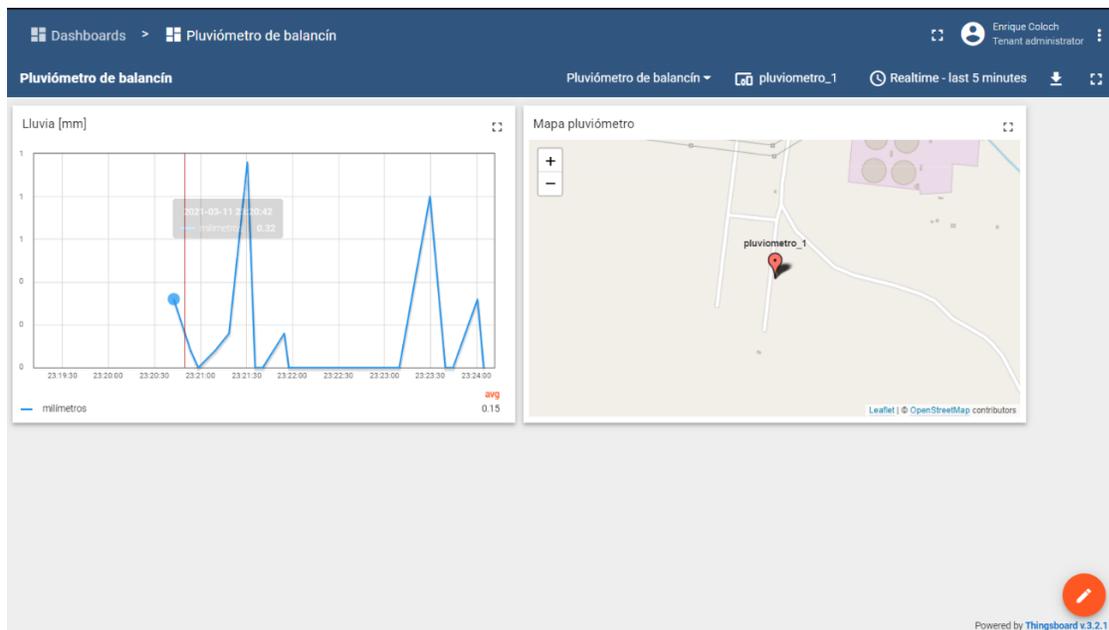


Fuente: elaboración propia, empleando Dashboard.

Se agrega el *widget* de mapa con los datos de latitud y longitud enviados por el pluviómetro y se realizan algunos ajustes de diseño. Finalmente se tiene un *dashboard* en tiempo real de medición de lluvia.

Este gráfico está corriendo en tiempo real y también es posible consultar datos históricos y realizar algunas funciones básicas como mostrar el máximo, mínimo, promedio y sumatoria en el período de tiempo que se elija, permitiendo al usuario conocer información histórica de la lluvia para planificar sus actividades y también tomar acciones con los datos en tiempo real.

Figura 81. Visualización en tiempo real



Fuente: elaboración propia, empleando Dashboard.

4.5. Alertas

Por lo regular un usuario de las aplicaciones de Internet de las cosas no se encuentra monitoreando las variables en todo momento del día. Si no tiene programadas alertas que le indicarán cuando una de las variables presenta un valor atípico o sale de los parámetros que se establecen para tomar acciones correctivas.

Para la aplicación de pluviómetros digitales con LoRaWAN se considera importante que el usuario tenga visibilidad de la detección de un evento de lluvia y en Thingsboard la programación de este tipo de alertas es posible.

Se utilizará SMS como medio de envío hacia el usuario, el cual recibirá una notificación a su celular cuando se detecte el evento de lluvia.

Thingsboard tiene integración directa con los servicios AWS SNS y Twilio, esos son servicios que existen en la nube y son de paga por cada mensaje enviado. En este proyecto se utilizará una cuenta de prueba de Amazon Web Services y el envío de mensajes será por medio del servicio AWS SNS.

No se mostrará la creación de una cuenta de Amazon Web Services, pero se pueden seguir los pasos en el siguiente enlace: <https://aws.amazon.com/premiumsupport/knowledge-center/create-and-activate-aws-account/>.

4.5.1. AWS SNS

Servicio sencillo de notificación SNS por sus siglas en inglés, es un servicio de Amazon Web Services completamente administrado para la comunicación entre aplicaciones y aplicaciones y personas.

Este servicio proporciona un sistema de mensajería de alto rendimiento, gracias a su núcleo basado en publicación-suscripción. Esto permite que las aplicaciones puedan distribuir mensajería a muchos clientes aplicativos y personas al mismo tiempo. La distribución a personas puede realizarse por medio de correos electrónicos, *push* móvil y mensajes de texto.

El funcionamiento de este servicio es muy parecido al protocolo MQTT que se explicó en el capítulo de la red LoRaWAN, en donde existen tópicos, publicadores, suscriptores y mensajes. Los publicadores envían un mensaje al bróker identificando el tópico y automáticamente todos los suscriptores a ese tópico recibirán el mensaje.

En este caso se debe configurar el tópico en el servicio SNS y suscribir al servidor de Thingsboard al servicio.

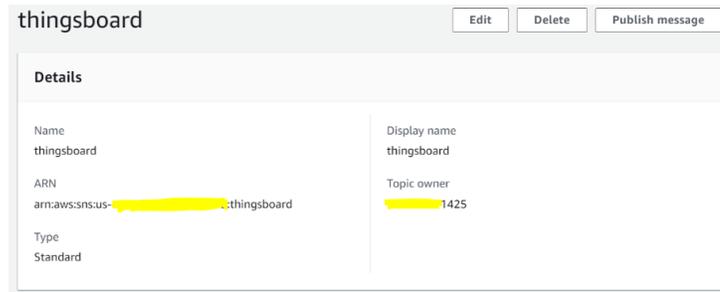
4.5.1.1. Configuración AWS SNS

Para la configuración del servicio se ingresa primero a la consola de AWS y se busca el servicio Amazon SNS.

En esta página se tiene la opción de crear un nuevo tópico y escribiendo el nombre y dirigirá a una pestaña de configuración en donde se puede realizar una configuración avanzada como alguna política o cifrado. Se deja por defecto la configuración y se crea el tópico.

Este tópico creado tendrá parámetros que servirán para configurarlo en la plataforma web de Thingsboard.

Figura 82. Configuración tópico Amazon SNS

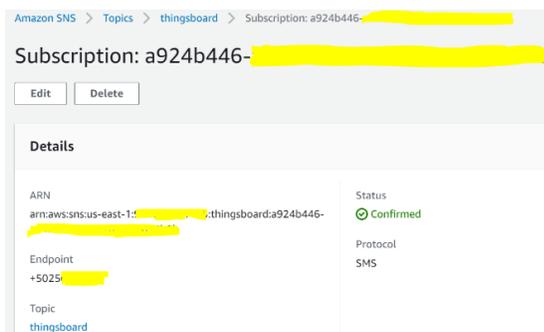


Fuente: elaboración propia, empleando Amazon SNS.

El siguiente paso es agregar un suscriptor para el tópico, este suscriptor será el número de celular o celulares que recibirán las alertas de eventos de lluvia.

Para agregar un suscriptor se va al tópico creado y se ve la opción de suscripciones y posteriormente la opción de crear suscripción. Esto dirigirá a una pestaña de configuración, se selecciona el protocolo SMS, el *endpoint*, el número de teléfono y aceptar.

Figura 83. Configuración tópico Amazon SNS



Fuente: elaboración propia, empleando Amazon SNS.

En este punto cualquier mensaje publicado en el servicio bajo el tópico “Thingsboard” llegará como mensaje de texto al teléfono suscrito.

4.5.1.2. Creación de usuario AWS para Thingsboard

Para que el servidor de Thingsboard pueda publicar mensajes en el servicio de SNS se le debe dar autorización en la plataforma de Amazon, esto es por la seguridad del servicio.

Para gestionar permisos y políticas de acceso, Amazon tiene una consola llamada “IAM” en donde se pueden crear políticas que se le asignan a los roles. Estos roles son asignados a usuarios de acceso a la plataforma y a estos usuarios se les permite ingresar y realizar las actividades que les son permitidas en las políticas.

También estos usuarios son pertenecientes a grupos para identificar qué actividades realizan, esto es parte del manejo de la infraestructura montada en la nube.

Para el proyecto de pluviómetros se realizará una configuración rápida para tener acceso desde Thingsboard a SNS.

Primero se direcciona al dashboard de “IAM” dentro de Amazon y se crea un grupo asignándole la política llamada “AmazonSNSFullAccess”. Posterior a esto se crea un usuario de tipo “Programmatic access” y se asigna al grupo que creado anteriormente.

Al crear al usuario de esta manera se obtienen las credenciales Access key ID y Secret Access Key, las cuales se utilizarán en Thingsboard para acceder a Amazon Web Services con los permisos de utilización del servicio de SNS.

Figura 84. Creación de usuario AWS para Thingsboard

Add user 1 2 3 4 5

✔ **Success**
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: [https://\[redacted\].signin.aws.amazon.com/console](https://[redacted].signin.aws.amazon.com/console)

Download .csv

	User	Access key ID	Secret access key
▶	✔ tboardUser	AKIA6J[redacted]	***** Show

Fuente: elaboración propia, empleando Amazon SNS.

4.5.1.3. Configuración SNS Thingsboard

Anteriormente se visualizaron las características de Thingsboard y la gran capacidad para el manejo de dispositivos y la integración de muchos microservicios que hacen más fácil el desarrollo de la plataforma de Internet de las Cosas.

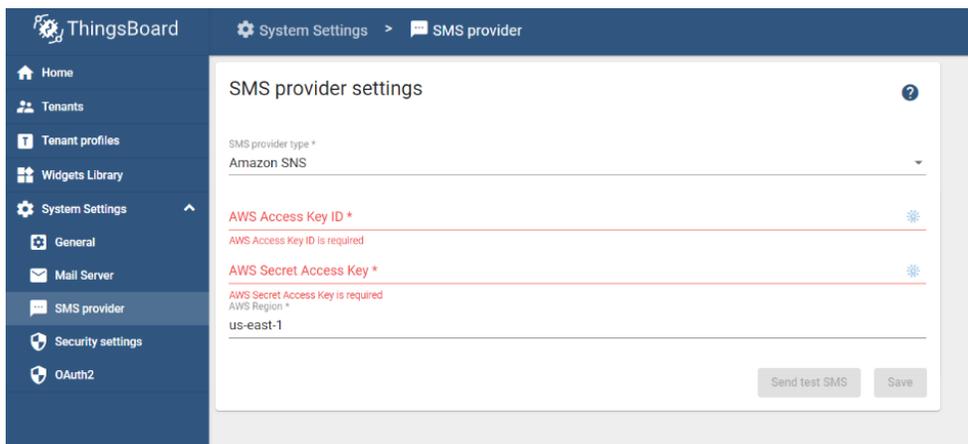
Dentro de estos microservicios integrados se encuentran la conexión directa con Amazon SNS y la programación de alertas mediante las reglas de flujo. A continuación, se abordará la configuración de estos en la plataforma de Thingsboard para el envío de alertas en caso de que se detecte lluvia.

Esta configuración es muy sencilla y se trata solamente de ingresar las credenciales que se obtuvieron al crear un usuario en la consola de IAM de Amazon Web Services. (figura 84).

Primero se ingresa al portal de Thingsboard con el rol de administrador global, en este portal se encuentra una opción de Configuración de sistema y posterior proveedor SMS.

En estos campos se ingresan las credenciales obtenidas, se puede hacer una prueba de funcionamiento y se guarda.

Figura 85. **Configuración SMS Thingsboard**



Fuente: elaboración propia, empleando Thingsboard.

4.5.2. **Alertas de lluvia en Thingsboard**

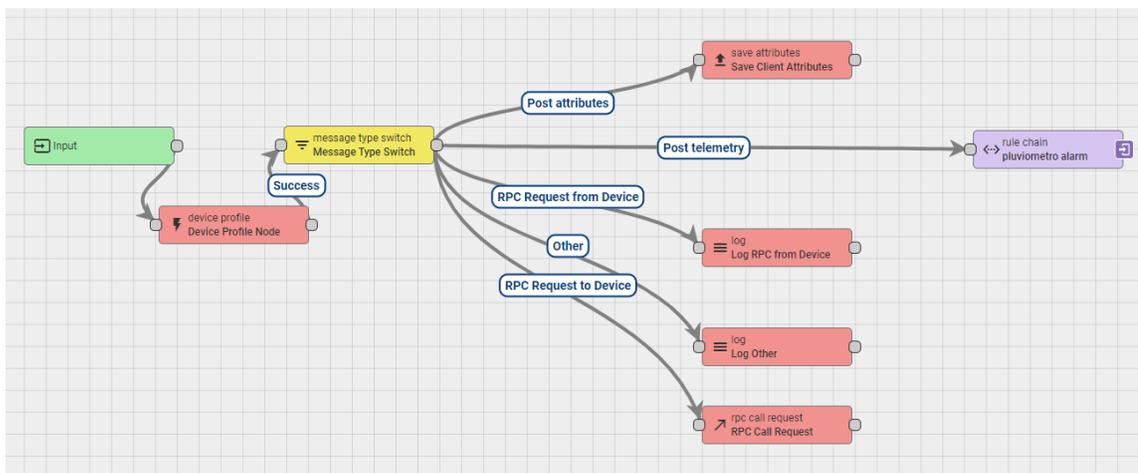
Thingsboard es una plataforma completa de microservicios diseñado específicamente para resolver necesidades de Internet de las Cosas. Una de estas necesidades son las alertas de eventos que llegan desde los dispositivos.

El desarrollo de estas alertas en Thingsboard no es muy complicado, se pueden definir las acciones gráficamente mediante reglas.

En primera instancia, se debe definir qué originará la alerta que se desea enviar, en este caso se notifica al usuario el inicio de un evento de lluvia.

Thingsboard por defecto contiene una regla principal, la cual define qué se realiza cuando existe una entrada en el servidor, para la configuración de la alerta se debe añadir una regla adicional a la regla principal llamada “pluviómetro alarm”, en donde se definirá cuándo se debe enviar la alarma y por qué medio.

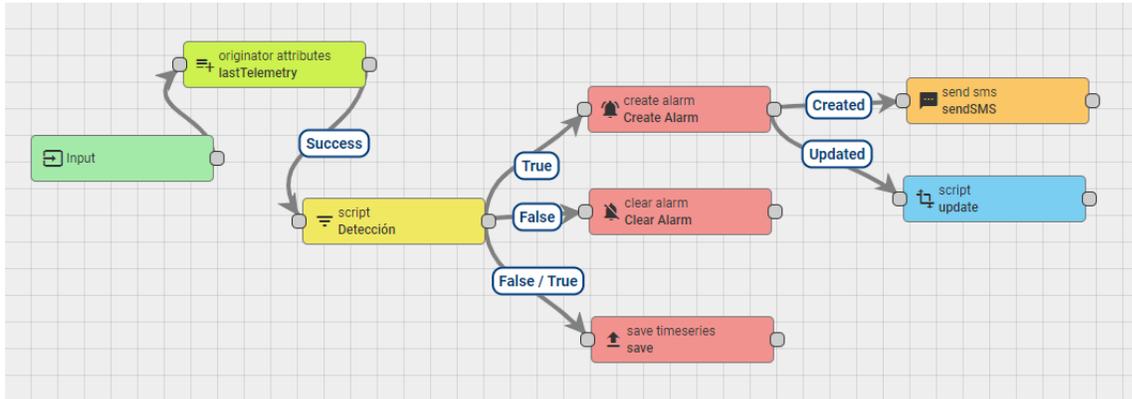
Figura 86. **Regla principal Thingsboard**



Fuente: elaboración propia, empleando Thingsboard.

En la regla principal se da una entrada y ordenamiento de los datos en cada uno de los campos que Thingsboard tiene definido para llevar el manejo de los dispositivos en la plataforma.

Figura 87. Regla de alarmas SMS Thingsboard



Fuente: elaboración propia, empleando Thingsboard.

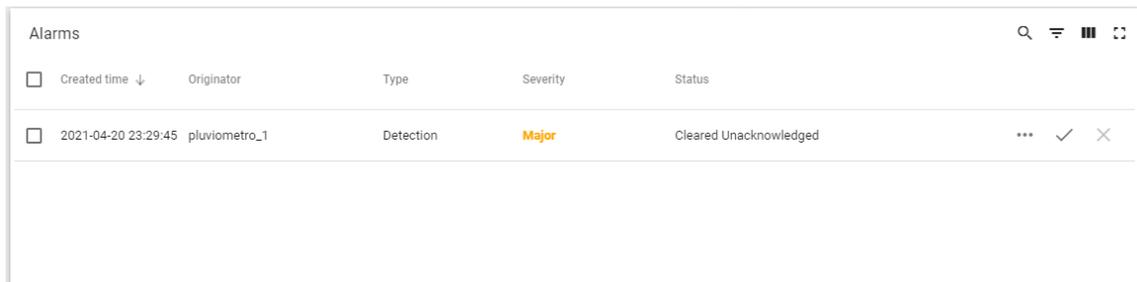
La regla de alarmas tiene como entrada la regla principal, existe un bloque llamado “lastTelemetry” el cual trae de la base de datos el último valor guardado de número de volteos, esto porque servirá para comparar si el valor sigue siendo el mismo o ha cambiado.

En bloque con nombre “Detección” se realiza la comparación del último valor guardado en la base de datos con el valor actual recibido, si este cambia significa que ha iniciado un evento de lluvia porque se detectó un volteo del balancín y se dirige al bloque “Create Alarm”. Si no ha cambiado limpia las variables yendo al bloque de “Clear Alarm”. Al mismo tiempo almacena el nuevo valor ingresado para tenerlo listo en un próximo envío de datos del dispositivo.

El bloque de creación de la alarma tiene dos salidas, una con evento “created” que enviará el mensaje de texto al usuario por medio del servicio de Amazon SNS previamente configurado. Y la otra salida con evento “update”, porque en caso de que siga lloviendo solo se actualice la alarma y no envíe mensajes en el transcurso de lluvia y solamente al inicio.

Se agrega un *widget* al *dashboard* para visualizar las alarmas activas y realizando las pruebas, las alarmas están listas para ser recibidas por los usuarios.

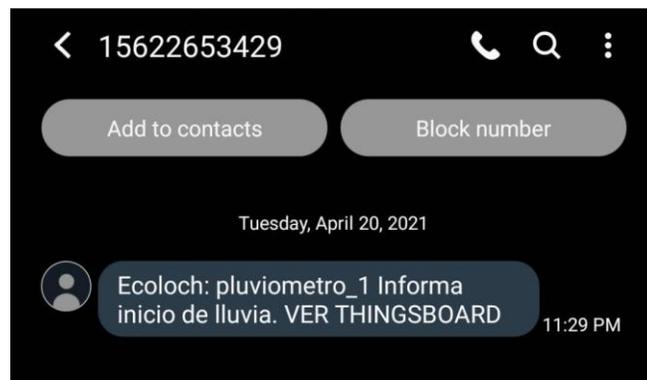
Figura 88. **Alarmas dashboard**



Created time ↓	Originator	Type	Severity	Status
2021-04-20 23:29:45	pluviometro_1	Detection	Major	Cleared Unacknowledged

Fuente: elaboración propia, empleando Thingsboard.

Figura 89. **Alarma SMS**



Fuente: elaboración propia.

4.6. Escalabilidad

Finalizado el alcance del proyecto desde la medición de lluvia, red de datos y visualización en tiempo real con alertas. Existe la necesidad de evaluar la escalabilidad del proyecto, entendiéndose escalabilidad como: poder replicar el modelo una o más veces. También evaluar hasta qué punto ya no es posible copiarlo.

Iniciando con la tarjeta electrónica, es completamente posible fabricar más tarjetas idénticas, ya que se tiene un diseño y un *software* que puede ser replicado infinidad de veces. Por lo tanto, se es posible decir que puede ser un producto comercial completamente utilizable por cualquier red LoRaWAN.

También se puede considerar que puede ser utilizado para más usos, no solamente realizar las mediciones de lluvia. Se sabe que el microcontrolador tiene entradas y salidas analógicas que pueden utilizarse para medir alguna otra variable. Sin ir muy lejos se podría construir una estación meteorológica agregando además de la lluvia, humedad, dirección del viento, velocidad del viento o radiación.

También se podría medir cualquier variable relacionada o no con meteorología. Es probable que con la interrupción se puedan medir las revoluciones de un motor o también es completamente posible utilizar el GPS de la tarjeta electrónica para monitorear la ubicación de algún vehículo o activo de suma importancia.

Las posibilidades del *hardware* son amplias y con la versatilidad de modificar el *software*, la tarjeta electrónica se convierte en una solución poderosa como dispositivo de Internet de las Cosas.

Hablando ahora de la red LoRaWAN, está de más decir que las aplicaciones que pueden hacer uso de esta infraestructura son incontables. Desde un monitoreo como el proyecto tratado en este documento, hasta un control con actuadores colocados en campo. Esto aunado a un posible análisis de los datos para crear sistemas automatizados que actúen en base a predicciones realizadas en una capa superior a lo tratado en este proyecto.

LoRaWAN busca resolver los problemas de conectividad de dispositivos de Internet de las Cosas en un ambiente fuera de lo convencional, es por esta razón que muchos desarrollos están ocurriendo, empleando esta tecnología. Muchas empresas están apostando por este protocolo a pesar de la entrada al mercado de otros modelos emergentes como el 5G. Incluso se puede mencionar el caso de Helium, una red LoRaWAN global basada en *blockchain* cuyo modelo de minería es para cubrir una zona con señal LoRa, un concepto nuevo llamado proof of coverage.

No solo este caso se puede traer a mención, hoy en día existen y se están formando redes LoRaWAN privadas o públicas que convergen entre sí respetando los distintos protocolos y buenas prácticas para una correcta utilización del espectro radioeléctrico.

Como bien se explicó en el capítulo de LoRaWAN, el Network Server ofrece múltiples integraciones hacia otros servicios que se pueden aprovechar para exprimir todo el potencial de los datos capturados por los dispositivos. En este proyecto se optó por utilizar Thingsboard por las excelentes características que se mencionaron en su momento y por la gran facilidad de uso e implementación.

Esta plataforma es muy escalable, no solamente porque puede integrarse a nuestra red LoRaWAN, sino tiene la posibilidad de integrarse a otras redes de Internet de las Cosas.

Se deben pensar en la unión de una red Wifi, Zigbee, Bluetooth, GPRS y LoRaWAN, todas comandadas por esta plataforma, donde se puedan programar tareas, alertas, controles, dependiendo de los datos recabados por múltiples dispositivos conectados a las distintas redes. Esto da mayor visibilidad al universo de aplicaciones que pueden existir en el mundo del Internet de las Cosas.

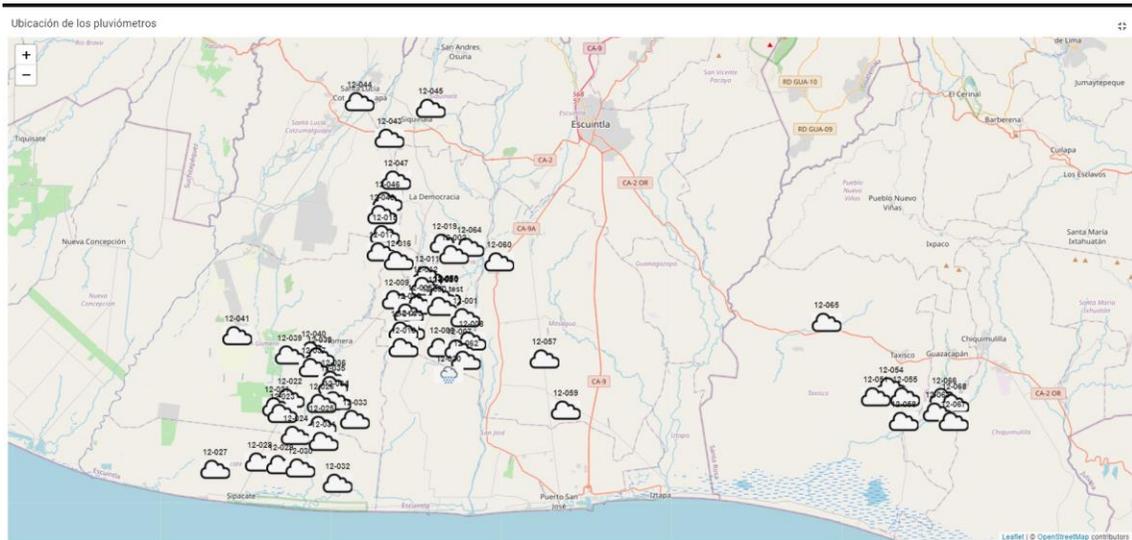
Finalmente, cabe mencionar la escalabilidad que se puede dar específicamente al proyecto presentado en este documento, proyecto que tiene un potencial enorme para aumentar la productividad en la industria de la agricultura, donde uno de los rubros más importante es el riego y conocer el dato preciso de lluvia en determinado lugar es de gran impacto para este rubro.

Al tener más muestras de lluvia por metro cuadrado se puede ser más específico al tomar decisiones, un concepto llamado agricultura de precisión. Basta con fabricar las tarjetas electrónicas, realizar exactamente el mismo procedimiento mostrado en este documento para tener más puntos de medición.

La escalabilidad es tanta, que se puede tener una red global de pluviómetros electrónicos funcionando sobre la misma red, con el modelo de tarjeta electrónica y empleando la misma plataforma de visualización de datos y alertas, aprovechando que Thingsboard tiene la característica de manejo de usuarios individuales.

Finalmente, como muestra de la aplicación real de lo desarrollado en este proyecto se puede mostrar el proyecto de red de 175 pluviómetros digitales conectados en una red LoRaWAN con 8 gateways desplegada en la costa sur de Guatemala, proyecto en la empresa Ingenio Magdalena S.A. ubicada en La Democracia Escuintla, empleando la tarjeta electrónica y sistema desarrollado en este documento.

Figura 90. Proyecto de 175 pluviómetros con LoRaWAN



Fuente: elaboración propia, empleando Thingsboard.

CONCLUSIONES

1. El funcionamiento del pluviómetro de balancín se basa en un sistema de pivoteo por gravedad para contabilizar el volumen de agua que ingresa a un área determinada y la metodología de medición de lluvia consiste en un método indirecto, dado por una ecuación que relaciona el volumen de agua obtenido y el área de captación. Principalmente del pluviómetro se obtienen pulsos que representan una cantidad en milímetros.
2. El desempeño de la tarjeta electrónica para pluviómetro de balancín es favorable para realizar la medición de lluvia mediante un circuito con interrupción. Se verifica el alcance de las transmisiones con señal LoRa mayores a 6km empleando solamente una potencia de 255 mW.
3. Una red LoRaWAN es un sistema que permite la interconexión de una gran cantidad de dispositivos de internet de las cosas mediante *gateways* y un servidor central. Esto permite la conectividad de la tarjeta de pluviómetro hacia la aplicación en internet Thingsboard para la visualización de datos y programación de alertas.
4. Thingsboard es una plataforma integrada de microservicios que permite el aprovechamiento de datos provenientes de dispositivos iot como el pluviómetro de balancín. La versatilidad para crear aplicaciones de utilidad es muy buena ya que tomó solamente unas horas en conectar la red LoRaWAN y construir una gráfica en tiempo real con alertas programadas.

RECOMENDACIONES

1. Medir la lluvia de manera automatizada, ya que, es una herramienta muy poderosa para la toma de decisiones, no resulta ser muy costosa y ofrece muy buenos beneficios. La utilización de una red LoRaWAN es ideal para entornos rurales, por el largo alcance y la eficiencia energética.
2. Realizar una actualización de *firmware* para la eliminación del panel solar y aumentar al máximo la eficiencia energética de la tarjeta electrónica. Esto se puede realizar con el concepto de ultra baja energía, en donde el microcontrolador entra a un modo reposo y solamente despertará cuando detecte un cambio en la interrupción del pluviómetro o alcance un tiempo establecido para reportarse como conectado.
3. Mantener un radio de cinco metros a la redonda de la base del pluviómetro libre de malezas, para evitar que el viento pueda levantar hojas al área de captación y que insectos se vean atraídos a habitar dentro del balancín, ocasionando obstrucciones en el flujo de agua o que ya no pueda pivotear.
4. Reducir el diámetro inferior del embudo en zonas donde existan lluvias extremas, con una intensidad muy alta. Es conveniente para mantener un caudal menor hacia el balancín y que no se sature.

5. Proteger La red LoRaWAN ante ataques de piratas informáticos, por lo que cada servicio de Chirpstack debería tener autenticación y cifrado, iniciando con el bróker MQTT y añadiendo capas de seguridad TLS y certificados SSL para el acceso al portal.

BIBLIOGRAFÍA

1. Agrotecnología tropical. *Importancia a medir las precipitaciones*. [en línea]. <https://www.agro-tecnologia-tropical.com/Importancia_precipitacion.php#:~:text=En%20el%20campo%20de%20la,forma%20m%C3%A1s%20acertada%20diversas%20planificaciones>. [Consulta: 2 de enero de 2019].
2. Aiteco Consultores. Desarrollo y gestion. *Lo que no se mide, no se puede mejorar*. [en línea]. <<https://www.aiteco.com/lo-que-no-se-mide/#:~:text=Esta%20frase%2C%20atribuida%20frecuentemente%20a,mide%2C%20no%20se%20puede%20mejorar>>. [Consulta: 2 de abril de 2020].
3. All Spares. *¿Qué es una batería 18650?* [en línea]. <<https://all-spares.com/es/technical-articles/what-does-18650-battery-mean.html#:~:text=Pila%2018650%20es%20una%20bater%C3%ADa,de%201600%20a%203600%20mAh.&text=Las%20ventajas%20principales%20de%20estas,ausencia%20de%20efecto%20de%20memoria>>. [Consulta: 2 de abril de 2020].
4. Arduino cc. *attachInterrupt* (). [en línea]. <<https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>>. [Consulta: 2 de abril de 2020].

5. Arduino.cc. *Arduino IDE 1.8.13.* [en línea].
<<https://www.arduino.cc/en/Reference/SoftwareSerial>>. [Consulta: 2 de abril de 2020].
6. _____. *Software Biblioteca de serie.* [en línea].
<<https://www.arduino.cc/en/Reference/SoftwareSerial>>. [Consulta: 2 de abril de 2020].
7. AREIZA, Y.; VARGAS, G.; PÉREZ, C. E.; ARISTIZABAL, J. C.: SALAZAR, A. *Diseño, construcción y prueba de un pluviómetro de cazoletas con comunicación inalámbrica.* [en línea].
<<http://revcolfis.org/ojs/index.php/rcf/article/download/440109/364>> . [Consulta: 2 de enero de 2019].
8. Atmel. *8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash.* [en línea].
<https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf>.
[Consulta: 1 de abril de 2020].
9. B.E.N. *What is an Arduino?* [en línea].
<<https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>>.
[Consulta: 3 de mayo de 2020].
10. BOEMO, Eduardo. *Aspectos eléctricos del diseño digital.* Madrid, España: Escuela Politécnica Superior. 2018. 13 p.

11. CORONA RAMÍREZ, Leonel; ABARCA JIMÉNEZ, Griselda; MARES CARREÑO, Jesús. *Sensores y actuadores*. México: Grupo Editorial Patria. 2014. 305 p.
12. Electrouni's blog. *Convertidores DAC-ADC*. [en línea]. <<https://electrouni.wordpress.com/unidad-iii/>>. [Consulta: 1 de abril de 2020].
13. FERNÁNDEZ, Pablo. *¿Como se forma la lluvia?* [en línea]. <<https://www.vix.com/es/btg/curiosidades/2011/02/02/%C2%BFcomo-se-forma-la-lluvia>>. [Consulta: 2 de enero de 2019].
14. GÜNTHER GRIDLING, Bettina Weiss. *Introduction to microcontrollers*. [en línea]. <<https://www.skylineuniversity.ac.ae/pdf/software-engineering/Microcontroller.pdf>>. [Consulta: 1 de abril de 2020].
15. HÉRNANDEZ CABALLERO, Silvia. *Estudio en detalle LoRaWAN*. [en línea]. <<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/106369/7/shernandezc6TFM0120memoria.pdf>>. [Consulta: 2 de febrero de 2021].
16. HIVEMQ. *Key Features of HiveMQ MQTT Broker*. [en línea]. <<https://www.hivemq.com/hivemq/mqtt-broker/>>. [Consulta: 2 de febrero de 2021].
17. ISHIKAWA, Masato; MARUTA, Ichiro. *Rapid Prototyping for Control Education using Arduino and Open-Source Technologies*. [en línea]. <<https://www.sciencedirect.com/science/article/pii/S1474667015316396>>. [Consulta: 3 de mayo de 2020].

18. KOOIJMAN, Mathijs; TELKAMP, Thomas. *Arduino LoraWan MAC in C (LMIC)*. [en línea]. <<https://redmine.laas.fr/attachments/download/1505/LMIC-v2.3.pdf>>. [Consulta: 2 de mayo de 2020].

19. KRISHNAN, Adarsh. *LoRaWAN and Multi-RAN Architecture connecting the Next Billion IoT devices*. New York: ABI research. 2020. 14 p.

20. LÓPEZ-REY, David. *Pluviómetros y pluviógrafos: Un afán histórico por medir la lluvia*. [en línea]. <<https://www.tutiempo.net/meteorologia/articulos/pluviometro-y-pluviografo.html>>. [Consulta: 1 de marzo de 2020].

21. LoRa Alliance. RP002-1.0.2 LoRaWAN Regional Parameters. [en línea]. <https://lora-alliance.org/wp-content/uploads/2020/11/RP_2-1.0.2.pdf>. [Consulta: 2 de junio de 2020].

22. LLAMAS, Luis. *El bus SPI en Arduino*. [en línea]. <<https://www.luisllamas.es/arduino-spi/>>. [Consulta: 1 de abril de 2020].

23. MANDADO PÉREZ, Enrique; MENÉNDEZ FUERTES, Luis Manuel; FERNÁNDEZ FERREIRA, Luis; LÓPEZ MATOS, Emilio. *Microcontroladores PIC. Sistema integrado para el autoaprendizaje*. Barcelona, España: MARCOMBO Ediciones Técnicas. 2007. 347 p.

24. MeteoLobios. *Lluvia*. [en línea]. <<http://www.meteolobios.es/lluvia.htm#:~:text=La%20precipitaci%C3%B3n%20se%20mide>>

%20en,L%20de%20agua%20por%20m%C2%B2>. [Consulta: 2 de enero de 2019].

25. Mi Sol. *Weather Station*. [en línea]. <<http://www.misolie.net/misol-spare-part-for-weather-station-to-measure-the-rain-volume-for-rain-meter-for-rain-gauge-p-513.html>>. [Consulta: 2 de marzo de 2020].
26. MOLANO PULIDO, José Francisco. *USART básico*. [en línea]. <[http://www.coffeebrain.org/wiki/index.php?title=USART_B%C3%A1sico#:~:text=USART%20\(Universal%20Synchronous%2FAynchronous%20Receiver,de%20recibir%20y%20transmitir%20simult%C3%A1neamente\)](http://www.coffeebrain.org/wiki/index.php?title=USART_B%C3%A1sico#:~:text=USART%20(Universal%20Synchronous%2FAynchronous%20Receiver,de%20recibir%20y%20transmitir%20simult%C3%A1neamente)>. [Consulta: 8 de abril de 2020].
27. NewieVentures. *LoraWAN. OTAA O ABP*. [en línea]. <<https://www.newieventures.com.au/blogtext/2018/2/26/lorawan-otaa-or-abp>>. [Consulta: 2 de junio de 2020].
28. PÉREZ COLÓN, Rafael; NAVAJAS, Sergio; TERRY, *Elizabeth*. *IOT EN ALC: Tomando el pulso al Internet de las Cosas en América Latina y el Caribe*. [en línea]. <https://publications.iadb.org/publications/spanish/document/loT_en_ALC_2019_Tomando_el_pulso_al_Internet_de_las_Cosas_en_Am%C3%A9rica_Latina_y_el_Caribe_es.pdf>. [Consulta: 2 de abril de 2020].
29. REYES CORTÉS, Fernando; CID MONJARAZ, Jaime. *Arduino. Aplicaciones en robótica, mecatrónica e ingenierías*. Barcelona, España: MARCOMBO. 2015. 468 p.

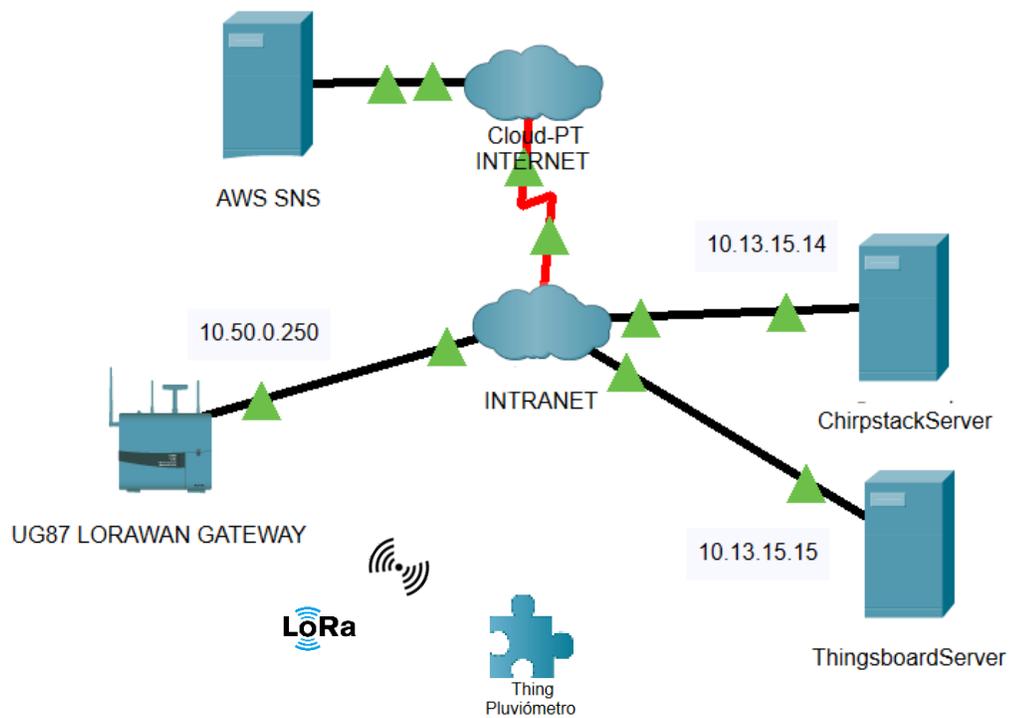
30. RODRÍGUEZ SANCHEZ, Cristina; VAQUERO LÓPEZ, Joaquín. *Domótica y sistemas de embebidos para comunicaciones inalámbricas*. [en línea]. <<https://pdfslide.tips/documents/tema-3-sistemas-embebidos-pic.html>>. [Consulta: 2 de febrero de 2021].
31. ROJAS BARRET, Pedro. *Diseño Estaciones Meteorológicas en las Cuencas Hidrográficas de CONELECTRICAS R.L.* [en línea]. <<https://docplayer.es/68639575-Instituto-tecnologico-de-costa-rica-escuela-de-ingenieria-electronica.html>>. [Consulta: 2 de enero de 2019].
32. SetUp. *La tecnología CMOS*. [en línea]. <<http://www.set-up.es/la-tecnologia-cmos/>>. [Consulta: 3 de mayo de 2020].
33. SORIA, Emilio. *Introducción al procesado digital de señales. Conversión A/D, D/A*. [en línea]. <https://www.uv.es/soriae/tema_1_pds.pdf>. [Consulta: 2 de abril de 2020].
34. Superintendencia de Telecomunicaciones. *Disposiciones técnicas asociadas al despliegue de redes inalámbricas de área extensa usando baja o muy baja potencia*. [en línea]. <https://sit.gob.gt/wp-content/uploads/2019/09/SIT-DSI_349-2019.pdf>. [Consulta: 2 de febrero de 2021].
35. _____. *Registro de comercializadores de equipos terminales móviles y tarjetas SIM en línea*. [en línea]. <<https://sit.gob.gt/>>. [Consulta: Consulta: 2 de febrero de 2021].

36. _____. *Resolución SIT-DSI-349-2019*. [en línea]. <https://sit.gob.gt/2019/09/11/resolucion-sit-dsi_349-2019/>. Consulta: 2 de febrero de 2021.
37. _____. *Tabla nacional de atribución de frecuencias radioeléctricas (Cuadro)*. [en línea]. <<https://bit.ly/34OHliy>>. [Consulta: 2 de febrero de 2021].
38. _____. *Tabla nacional de atribución de frecuencias radioeléctricas*. [en línea]. <[https://sit.gob.gt/download/872/tabla-nacional-de-atribucion-de-frecuencias-frecuancias/1zKgbuwSQuK3ytp3hprvQcljXxX2WyCbd/4.%20Tabla%20Nacional%20Atribuci%C3%B3n%20Frecuencias%20\(Pies%20de%20P%C3%A1gina\).pdf](https://sit.gob.gt/download/872/tabla-nacional-de-atribucion-de-frecuencias-frecuancias/1zKgbuwSQuK3ytp3hprvQcljXxX2WyCbd/4.%20Tabla%20Nacional%20Atribuci%C3%B3n%20Frecuencias%20(Pies%20de%20P%C3%A1gina).pdf)>. [Consulta: 2 de febrero de 2021].
39. The things Network. *Ground breaking world record! LoRaWAN packet received at 702 km (436 miles) distance*. [en línea]. <<https://www.thethingsnetwork.org/article/ground-breaking-world-record-lorawan-packet-received-at-702-km-436-miles-distance>>. [Consulta: 2 de mayo de 2020].
40. Thingsboard. *Plataforma IoT de código Abierto*. [en línea]. <<https://Thingsboard.io/>>. [Consulta: 2 de febrero de 2021].
41. Universidad Estatal de Arizona. *WMO Rapporteur of Weather & Climate Extremes*. [en línea]. <<https://wmo.asu.edu/content/world-meteorological-organization-global-weather-climate-extremes-archive>>. [Consulta: 2 de abril de 2020].

42. VALLEJO, Henry. *Calibración de pluviómetro*. [en línea].
<<http://www.henryvallejo.com/wp-content/uploads/2016/12/calibracion-pluviometro.pdf>>. [Consulta: 2 de marzo de 2020].

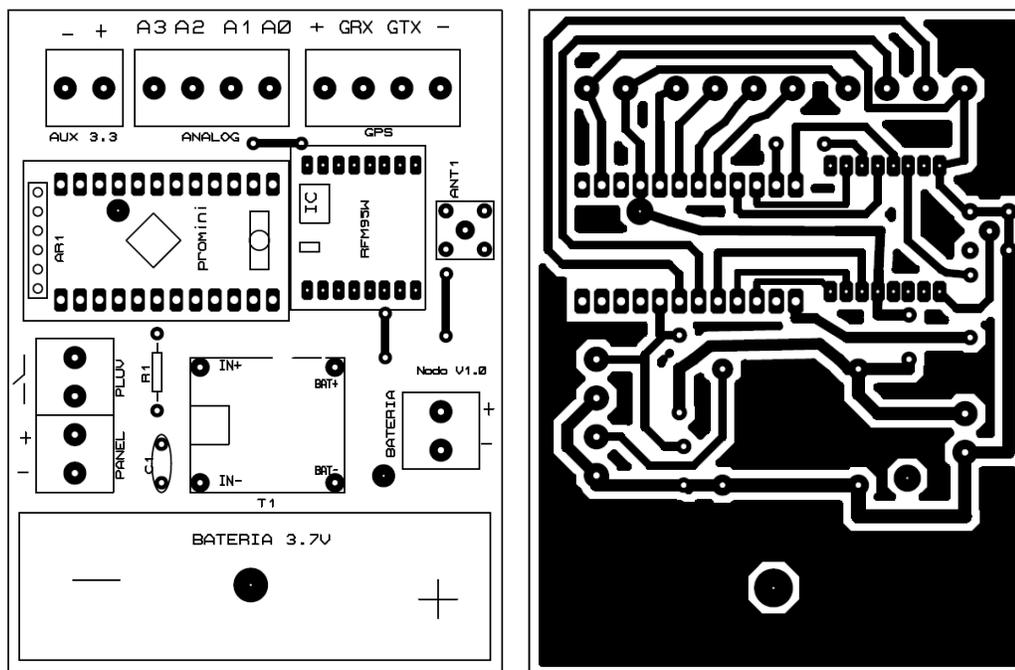
APÉNDICES

Apéndice 1. Arquitectura de red



Fuente: elaboración propia, empleando Packet Tracer.

Apéndice 3. Circuito PCB



Fuente: elaboración propia, empleando Proteus

Apéndice 4. Listado de materiales y presupuesto

	DESCRIPCIÓN	CANTIDAD	PRECIO	SUBTOTAL
1	TARJETA PRO MINI 3,3V 8 MHZ	1	Q 55,00	Q 55,00
2	MÓDULO RFM95W LORA 915 MHZ	1	Q 133,00	Q 133,00
3	MÓDULO DE CARGA TP4056	1	Q 14,00	Q 14,00
4	CONECTOR COAXIAL P/ANTENA SMA PCB 0,53"	1	Q 50,00	Q 50,00
5	CAPACITOR CERÁMICO 100 nF 104	1	Q 1,00	Q 1,00
6	RESISTENCIA 231K OHM 1/4 WATT	1	Q 1,00	Q 1,00
7	BATERIA RECARGABLE 18650 5000 MAH 3,7 V	1	Q 40,00	Q 40,00
8	SOPORTE P/BATERIA 18650	1	Q 11,00	Q 11,00

Continuación del apéndice 4.

9	BLOCK D/TERMINAL 2 PINES	8	Q 1,75	Q 14,00
10	CELULA SOLAR 5,5 V 160 MA 90X80MM	1	Q 50,00	Q 50,00
11	CABLE D/EXTENSION COAXIAL SMA 5 M	1	Q 70,00	Q 70,00
12	ANTENA P/NODO LORAWAN SMA MALE	1	Q 40,00	Q 40,00
13	TARJETA PCB P/PLUVIÓMETRO V 1.0	1	Q 20,00	Q 20,00
14	DADO RJ11 CATEGORIA 3	1	Q 10,00	Q 10,00
15	CAJA D/DERIVACION UNIVERSAL 3,9X3,9X2,8"	1	Q 40,00	Q 40,00
16	BALANCÍN MISOL WH-SP-RG	1	Q 120,00	Q 120,00
17	EXTENSOR Y BASE PLUVIÓMETRO	1	Q 160,00	Q 160,00
18	GATEWAY URSALINK UG87 LORAWAN	1	Q9 000,00	Q9 000,00
TOTAL				Q9 829,00

Fuente: elaboración propia, empleando Microsoft Excel 365.

Apéndice 5. **Fotografía de pluviómetro instalado**



Fuente: elaboración propia.

Apéndice 6. Código fuente de tarjeta de pluviómetro

```
/*
*****
* Copyright (c) 2020 Enrique Coloch basado en ejemplo ttn-abp de librería LMIC
* Firmware para tarjeta de pluviómetro de balancín
* Conectividad LoRaWAN/Pluviometro con sensor tipo reed switch por interrupcion
* Este programa envía la cantidad de volteos acumulados del balancín cada "TXINTERVAL" segundos.
* La activacion usada es ABP (Activation-by-personalisation), donde DevAddr
* NSKEY y APPKEY estan preconfiguradas y obtenidas del servidor Chirpstack.io LoRaWAN
* Se tiene configurado la subbanda 1 en frecuencia de US915MHZ con SF7 y CR 4/5
* La definición de la frecuencia se realiza en arduino-lmic/project_config/lmic_project_config.h
*****
#include <EEPROM.h> //Librería para control de EEPROM
#include <lmic.h> //Librería para control de radio LoRa y framework LoRaWAN
#include <hal/hal.h> //Librería para capa de control de hardware microcontrolador
#include <SPI.h> //Librería para interfaz SPI
#include <TinyGPS.h> //Librería para decodificar GPS
#include <SoftwareSerial.h> //Librería de interfaz serial por software
#define spluv 2 //Se define el pin para entrada de sensor de lluvia

TinyGPS gps; //Se instancia un objeto tipo gps
SoftwareSerial serialgps(3,4); //Se configuran los pines 3 y 4 como TX y RX del serial
float latitude, longitude; //Variables para almacenar latitud y longitud decodificada
long latitude_long=0; //Variable para almacenar la latitud para transmitir
long longitude_long=0; //Variable para almacenar la longitud para transmitir
uint32_t volteosAcumulados; //Variable para almacenar volteosAcumulados
int contadorTX=0; //Variable para contar número de transmisiones.

const unsigned TX_INTERVAL = 30; //Variable para definir el período de transmisión

const String info = "Name: Tarjeta pluviometro de balancin\nVersion: 1.0\nGuatemala 2021\n(c)Enrique Coloch\nBy:quiquecoloch@gmail.com";

// LoRaWAN Network Session Key, llave para autenticación en servidor LoRaWAN
static const PROGMEM u1_t NWKSKEY[16] = { 0x55, 0xee, 0x96, 0xdc, 0xb1, 0x8e, 0x4c, 0x55, 0x42, 0xc9,
0xef, 0xae, 0x5c, 0xdc, 0x57, 0xaa };
// LoRaWAN Application Session Key, llave para autenticación en aplicación LoRaWAN
```

Continuación del apéndice 6.

```
static const u1_t PROGMEM APPSKEY[16] = { 0x7c, 0x82, 0x39, 0xe7, 0x23, 0x35, 0xe6, 0x06, 0x61, 0xa2,
0xed, 0xc6, 0xcc, 0xb2, 0x20, 0x1f };
// LoRaWAN Device Address, dirección lógica en red LoRaWAN.
static const u4_t DEVADDR = 0x01f045af ;

// Almacena credenciales asignadas por la red LoRaWAN
void os_getArtEui (u1_t* buf) { }
void os_getDevEui (u1_t* buf) { }
void os_getDevKey (u1_t* buf) { }

static osjob_t sendjob; //Realiza la instancia del procedimiento de transmisión

//Objeto para conexión del módulo LoRa RFM95W
const lmic_pinmap lmic_pins = {
    .nss = 10, // Chip Select de SPI
    .rxtx = LMIC_UNUSED_PIN, // Selecciona la dirección de la antena
    .rst = 9, // Pin para reset del módulo LoRa
    .dio = {5,6,7}, // Pines de estado de chip LoRa
};

//Método de eventos de protocolo LoRaWAN implementado con LMIC

void onEvent (ev_t ev) {
    Serial.print(os_getTime());
    Serial.print(": ");
    switch(ev) {
        case EV_SCAN_TIMEOUT:
            Serial.println(F("EV_SCAN_TIMEOUT"));
            break;
        case EV_BEACON_FOUND:
            Serial.println(F("EV_BEACON_FOUND"));
            break;
        case EV_BEACON_MISSED:
            Serial.println(F("EV_BEACON_MISSED"));
            break;
        case EV_BEACON_TRACKED:
```

Continuación del apéndice 6.

```
    Serial.println(F("EV_BEACON_TRACKED"));
    break;
case EV_JOINING:
    Serial.println(F("EV_JOINING"));
    break;
case EV_JOINED:
    Serial.println(F("EV_JOINED"));
    break;
case EV_JOIN_FAILED:
    Serial.println(F("EV_JOIN_FAILED"));
    break;
case EV_REJOIN_FAILED:
    Serial.println(F("EV_REJOIN_FAILED"));
    break;
case EV_TXCOMPLETE:                                     //Transmisión completa
    Serial.println(F("EV_TXCOMPLETE, Se abren ventanas de recepcion"));

    if (LMIC.txrxFlags & TXRX_ACK)                       //Se verifica que hayan paquetes recibidos.
        Serial.println(F("Received ack"));
    if (LMIC.dataLen) {
        Serial.println(F("Received "));
        Serial.println(LMIC.dataLen);
        Serial.println(F(" bytes of payload"));
    }

    uint32_t volteosAcumuladosEeprom;                    //Variable temporal para obtener volteos
acumulados de EEPROM
    EEPROM.get(0,volteosAcumuladosEeprom);              //Obtiene volteos acumulados de
EEPROM
    //Serial.println(volteosAcumuladosEeprom);
    if(volteosAcumuladosEeprom != volteosAcumulados){   //Compara si el nuevo valor
transmitido ha cambiado
        EEPROM.put(0, volteosAcumulados);              //Actualiza el valor de volteos acumulados
        Serial.println("Eeprom se actualizo");
    }else{
```

Continuación del apéndice 6.

```
        Serial.println("Eeprom no se actualizo");           //No actualiza el valor de volteos acumulados
    }

    contadorTX=contadorTX+1;                               //Suma una transmisión completa

    os_setTimedCallback(&sendjob, os_getTime()+sec2osticks(TX_INTERVAL), do_send);
//Programa un nuevo envío en un periodo TX_INTERVAL.
    break;
case EV_LOST_TSYNC:
    Serial.println(F("EV_LOST_TSYNC"));
    break;
case EV_RESET:
    Serial.println(F("EV_RESET"));
    break;
case EV_RXCOMPLETE:
    // data received in ping slot
    Serial.println(F("EV_RXCOMPLETE"));
    break;
case EV_LINK_DEAD:
    Serial.println(F("EV_LINK_DEAD"));
    break;
case EV_LINK_ALIVE:
    Serial.println(F("EV_LINK_ALIVE"));
    break;

    default:
        Serial.print(F("Unknown event: "));
        Serial.println((unsigned) ev);
        break;
}
}

//Metodo de transmision de datos

void do_send(osjob_t* j){
```

Continuación del apéndice 6.

```
if (LMIC.opmode & OP_TXRXPEND) { //Verifica que el módulo no esté ocupado
  Serial.println(F("OP_TXRXPEND, not sending"));
} else {
  //Preparación de los paquetes
  //Si no son las dos primeras transmisiones envía datos de lluvia
  if(contadorTX >= 2){
    uint8_t payload[5];

//Separa por bytes la variable numérica por medio de corrimientos de bits

    payload[0] = (byte) ((volteosAcumulados & 0xFF000000) >> 24 );
    payload[1] = (byte) ((volteosAcumulados & 0x00FF0000) >> 16 );
    payload[2] = (byte) ((volteosAcumulados & 0x0000FF00) >> 8 );
    payload[3] = (byte) (volteosAcumulados & 0x000000FF);
    payload[4] = 0;

    //Envía datos de lluvia en el puerto 1 sin solicitar confirmación

    LMIC_setTxData2(1, payload, sizeof(payload)-1, 0);
    Serial.println(F("Packet queued"));

    //Si son las dos primeras transmisiones después de encendido

  }else{

    //Obtiene la lectura del GPS actualizando las variables latitude_long, longitude_long

    leerGPS();
    uint8_t payload[9];
    payload[0] = (byte) ((latitude_long & 0xFF000000) >> 24 );
    payload[1] = (byte) ((latitude_long & 0x00FF0000) >> 16 );
    payload[2] = (byte) ((latitude_long & 0x0000FF00) >> 8 );
    payload[3] = (byte) ((latitude_long & 0X000000FF));

    payload[4] = (byte) ((longitude_long & 0xFF000000) >> 24 );
    payload[5] = (byte) ((longitude_long & 0x00FF0000) >> 16 );
```

Continuación del apéndice 6.

```
payload[6] = (byte) ((longitude_long & 0x0000FF00) >> 8);
payload[7] = (byte) ((longitude_long & 0X000000FF));

    //Envia datos de GPS en el puerto 5 sin solicitar confirmación
    LMIC_setTxData2(5, payload, sizeof(payload)-1, 0);
    Serial.println(F("Packet queued"));

}
}
}

//Obtiene GPS

void leerGPS(){
    serialgps.begin(9600);                //Inicializa modulo

//Se mantiene en el bucle hasta que encuentre una coordenada

    while((latitude_long == 0 ) && longitude_long == 0){

        while(serialgps.available()){           //Consulta si está disponible el serial
            int c = serialgps.read();           //Lee códigos NMEA del modulo GPS
            if(gps.encode(c)){                 //Consulta si tiene la codificación
                gps.f_get_position(&latitude, &longitude); //Decodifica NMEA a valores de latitud y longitud
            }
        }
    }

    Serial.print("Latitud/Longitud: ");
    Serial.print(latitude,8);
    Serial.print(", ");
    Serial.println(longitude,8);

    latitude_long = latitude * 1000000;        //Convierte a un formato de envío
    longitude_long = longitude * 1000000;     //Convierte a un formato de envío
}
```

Continuación del apéndice 6.

```
    delay(1000);
  }

  serialgps.end();          //Finaliza la utilización del serial
}

//Metodo de interrupcion
void MedicionLluvia(){
//Si se detecta un volteo del balancin, suma uno al acumulado
  volteosAcumulados = volteosAcumulados + 1;
  Serial.println(volteosAcumulados);
}

//Configuracion del programa

void setup() {

  EEPROM.get( 0, volteosAcumulados );          //Obtiene los volteos acumulados en la
EEPROM al encender
  pinMode(spluv,INPUT);
  attachInterrupt(digitalPinToInterrupt(spluv),MedicionLluvia,FALLING);
//Configura la interrupcion con Signal Edge FALLING

  while (!Serial);
  Serial.begin(115200);          //Inicializa el monitor serial
  delay(100);
  Serial.println(info);
  Serial.println(F("Iniciando"));

  os_init();          // Iniciando configuracion de LMIC
  LMIC_reset();          // Reinicia valores
  LMIC_setClockError(MAX_CLOCK_ERROR * 1 / 100);
// Divide la frecuencia de reloj para reducir errores

#ifdef PROGMEM
uint8_t appskey[sizeof(APPSKEY)];          //Variable de tamaño de vector de llave de aplicacion
```

Continuación del apéndice 6.

```
uint8_t nwkskey[sizeof(NWKSKEY)];          //Variable de tamaño de vector de llave de network session
memcpy_P(appskey, APPSKEY, sizeof(APPSKEY)); //Copia las llaves a la memoria SRAM
memcpy_P(nwkskey, NWKSKEY, sizeof(NWKSKEY)); //Copia las llaves a la memoria SRAM

LMIC_setSession (0x13, DEVADDR, nwkskey, appskey); //Coloca paquetes para inicio de sesion
#else
LMIC_setSession (0x13, DEVADDR, NWKSKEY, APPSKEY);
//Coloca paquetes de inicio de sesion si no se copiaron a la SRAM
#endif

//Configuracion de canales frecuencia de EU868Mhz si se elige esa frecuencia

#if defined(CFG_eu868)
LMIC_setupChannel(0, 868100000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI);
LMIC_setupChannel(1, 868300000, DR_RANGE_MAP(DR_SF12, DR_SF7B), BAND_CENTI);
LMIC_setupChannel(2, 868500000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI);
LMIC_setupChannel(3, 867100000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI);
LMIC_setupChannel(4, 867300000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI);
LMIC_setupChannel(5, 867500000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI);
LMIC_setupChannel(6, 867700000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI);
LMIC_setupChannel(7, 867900000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI);
LMIC_setupChannel(8, 868800000, DR_RANGE_MAP(DR_FSK, DR_FSK), BAND_MILLI);

//Seleccion de canales de frecuencia US915MHz se selecciona en el archivo de configuracion de la libreria

#elif defined(CFG_us915)

LMIC_selectSubBand(1); //Se selecciona la sub-banda 1 de US915MHz

#elif defined(CFG_au921) //Si se elige esa frecuencia

#endif

//Se desahabilita la sub-banda 0.
for (int i = 0; i <= 7; i++) {
```

Continuación del apéndice 6.

```
    LMIC_disableChannel(i);
}

//Se deshabilita las sub-bandas 2 a la 8.

for (int i = 16; i <= 64; i++) {
    LMIC_disableChannel(i);
}

//Se deshabilitan la sub-banda 9
for (int i = 66; i <= 72; i++) {
    LMIC_disableChannel(i);
}

LMIC_setAdrMode(0);    // Se deshabilita el ADR (Adaptative Data Rate)

LMIC_setLinkCheckMode(0); //Se deshabilita direccion de validacion

LMIC.dn2Dr = DR_SF9;    //Se configura el Data Rate y Spreading Factor para recepcion.

LMIC_setDrTxpow(DR_SF7,14); //Se configura un Spreading Factor de 7 y potencia de 14dBm para
transmision

do_send(&sendjob);    //Inicia trabajo de transmision
}

void loop() {
    unsigned long now;
    now = millis();
    if ((now & 512) != 0) {
        digitalWrite(13, HIGH);
    }
    else {
        digitalWrite(13, LOW);
    }
}
```

Continuación del apéndice 6.

```
os_runloop_once();    //Inicia el ciclo de transmisiones  
  
}
```

Fuente: elaboración propia.

Apéndice 7. **Código fuente *decoder* tarjeta de pluviómetro**

```
function Decode(fPort, bytes, variables) {  
  devolver = {}          //Crea objeto para añadir al Json  
  
  if(fPort==1){         //Si el paquete viene en el puerto 1 es lluvia  
  
    //Decodifica uniendo los bytes.  
    var clicks = (bytes[0]<<24) | bytes[1]<<16 | bytes[2]<<8 | bytes[3];  
    devolver.clicks = clicks;  
  
  }else if (fPort = 5){ //Si el paquete viene en el puerto 5 es GPS  
  
    var lat = (bytes[0]<<24) | bytes[1]<<16 | bytes[2]<<8 | bytes[3];  
    var lon = (bytes[4]<<24) | bytes[5]<<16 | bytes[6]<<8 | bytes[7];  
    devolver.latitud = lat/1000000;  
    devolver.longitud= lon/1000000;  
  }  
  
  return devolver;     //Devuelve el objeto  
}
```

Fuente: elaboración propia, empleando Bloc de Notas.

Apéndice 8. Código fuente paquete JSON generado en Chirsptack de 3 mensajes

```
[
  {
    "type": "up",
    "payload": {
      "applicationID": "1",
      "applicationName": "medicion_lluvia",
      "deviceName": "pluviometro_de_balancin_01",
      "devEUI": "0xwU97NVWi0=",
      "rxInfo": [
        {
          "gatewayID": "JOEk//7xPsl=",
          "time": "2021-05-13T16:42:33.986237Z",
          "timeSinceGPSEpoch": "1304959344.058s",
          "rssi": -116,
          "loRaSNR": -5.2,
          "channel": 2,
          "rfChain": 0,
          "board": 0,
          "antenna": 0,
          "location": {
            "latitude": 14.12585,
            "longitude": -90.93121,
            "altitude": 120,
            "source": "UNKNOWN",
            "accuracy": 0
          },
          "fineTimestampType": "NONE",
          "context": "tA9OWw==",
          "uplinkID": "GzsqjTWhtpeBF/WH11Syvg==",
          "crcStatus": "NO_CRC"
        }
      ],
      "txInfo": {
        "frequency": 904300000,
        "modulation": "LORA",
```

Continuación del apéndice 8.

```
"loRaModulationInfo": {
  "bandwidth": 125,
  "spreadingFactor": 7,
  "codeRate": "4/5",
  "polarizationInversion": false
}
},
"adr": false,
"dr": 3,
"fCnt": 2,
"fPort": 1,
"data": "AAAAGQ==",
"objectJSON": "{\"clicks\":25}",
"tags": {},
"confirmedUplink": false,
"devAddr": "AfBFrw=="
}
},
{
  "type": "up",
  "payload": {
    "applicationID": "1",
    "applicationName": "medicion_lluvia",
    "deviceName": "pluviometro_de_balancin_01",
    "devEUI": "0xwU97NVWi0=",
    "rxInfo": [
      {
        "gatewayID": "JOEk//7xPsl=",
        "time": "2021-05-13T16:42:02.750347Z",
        "timeSinceGPSEpoch": "1304959312.818s",
        "rssi": -119,
        "loRaSNR": -6.8,
        "channel": 1,
        "rfChain": 0,
        "board": 0,
        "antenna": 0,

```

Continuación del apéndice 8.

```
"location": {
  "latitude": 14.12585,
  "longitude": -90.93121,
  "altitude": 121,
  "source": "UNKNOWN",
  "accuracy": 0
},
"fineTimestampType": "NONE",
"context": "sjKvOw==",
"uplinkID": "VUh/Jc2QQ+aAlziPUvMfbQ==",
"crcStatus": "NO_CRC"
}
],
"txInfo": {
  "frequency": 904100000,
  "modulation": "LORA",
  "loRaModulationInfo": {
    "bandwidth": 125,
    "spreadingFactor": 7,
    "codeRate": "4/5",
    "polarizationInversion": false
  }
},
"adr": false,
"dr": 3,
"fCnt": 1,
"fPort": 5,
"data": "ANd2VfqUbYA=",
"objectJSON": "{\"latitud\":14.120533,\"longitud\":-90.935936}",
"tags": {},
"confirmedUplink": false,
"devAddr": "AfBFrw=="
}
},
{
  "type": "up",
  "payload": {
```

Continuación del apéndice 8.

```
"applicationID": "1",
"applicationName": "medicion_lluvia",
"deviceName": "pluviometro_de_balancin_01",
"devEUI": "0xwU97NVWi0=",
"rxInfo": [
  {
    "gatewayID": "JOEk//7xPsl=",
    "time": "2021-05-13T16:41:30.471425Z",
    "timeSinceGPSEPOCH": "1304959280.541s",
    "rssi": -119,
    "loRaSNR": -8.8,
    "channel": 0,
    "rfChain": 0,
    "board": 0,
    "antenna": 0,
    "location": {
      "latitude": 14.12585,
      "longitude": -90.93122,
      "altitude": 120,
      "source": "UNKNOWN",
      "accuracy": 0
    },
    "fineTimestampType": "NONE",
    "context": "sEYlww==",
    "uplinkID": "d+ZcKaOIT4eRJSPhe7RXvA==",
    "crcStatus": "NO_CRC"
  },
],
"txInfo": {
  "frequency": 903900000,
  "modulation": "LORA",
  "loRaModulationInfo": {
    "bandwidth": 125,
    "spreadingFactor": 7,
    "codeRate": "4/5",
    "polarizationInversion": false
  }
}
```

Continuación del apéndice 8.

```
    }  
  },  
  "adr": false,  
  "dr": 3,  
  "fCnt": 0,  
  "fPort": 5,  
  "data": "ANd2VfqUbYA=",  
  "objectJSON": "{\"latitud\":14.120533,\"longitud\":-90.935936}",  
  "tags": {},  
  "confirmedUplink": false,  
  "devAddr": "AfBFrw=="  
}  
}  
]
```

Fuente: elaboración propia, utilizando Bloc de Notas.