



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN
ELECTRÓNICA Y REDES DE COMUNICACIÓN**

TEMA:

**“MEJORAMIENTO DE LA EFICIENCIA DE UN SISTEMA DE RIEGO POR
ASPERSIÓN BASADO EN REDES DE SENSORES Y REDES NEURONALES
ARTIFICIALES PARA EL CULTIVO DE PASTO EN LA PARROQUIA DE
TOCACHI PERTENECIENTE AL CANTÓN PEDRO MONCAYO”**

AUTOR: EVELYN PATRICIA BONILLA FONTE

DIRECTOR: MSC. EDGAR ALBERTO MAYA OLALLA

IBARRA-ECUADOR

2023



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

IDENTIFICACIÓN DE LA OBRA

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

DATOS DEL CONTACTO	
Cedula de identidad	1004476683
Apellidos y nombres	Bonilla Fonte Evelyn Patricia
Dirección	Imbabura – Otavalo - González Suarez
E-mail	eponillaf@utn.edu.ec
Teléfono móvil	0968242994
DATOS DE LA OBRA	
Título	“Mejoramiento de la eficiencia de un sistema de riego por aspersión basado en redes de sensores y redes neuronales artificiales para el cultivo de pasto en la parroquia de Tocachi perteneciente al cantón Pedro Moncayo”
Autor	Bonilla Fonte Evelyn Patricia
Fecha	15 de febrero del 2023
Programa	Pregrado
Título por el que opta	Ingeniería en Electrónica y Redes de Comunicación
Director	Msc. Edgar Alberto Maya Olalla



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CONSTANCIAS

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

Ibarra, a 22 días del mes de marzo del 2023

EL AUTCR:

Evelyn Patricia Bonilla Fonte

C.I :1004476683



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN

Msc. Edgar Alberto Maya Olalla director del presente trabajo de titulación certifica :

Que, el presente trabajo de titulación denominado “MEJORAMIENTO DE LA EFICIENCIA DE UN SISTEMA DE RIEGO POR ASPERSIÓN BASADO EN REDES DE SENSORES Y REDES NEURONALES ARTIFICIALES PARA CULTIVO DE PASTO EN LA PARROQUIA DE TOCACHI PERTENECIENTE AL CANTÓN PEDRO MONCAYO” . Ha sido desarrollado por la Srta. Evelyn Patricia Bonilla Fonte bajo mi supervisión.

Esto todo en cuanto puedo certificar en honor a la verdad .

Msc. Edgar Alberto Maya Olalla

Director



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

AGRADECIMIENTO

En primer lugar, agradezco a Dios por concederme la vida y permitirme llegar hasta esta etapa de mi vida como profesional.

Agradezco a mis padres, Antonio Bonilla y Margarita Fonte por brindarme todo su amor, confianza y apoyo incondicional en cada una de las decisiones que he tomado y enseñarme con su ejemplo que con humildad y perseverancia puedo llegar lejos.

Agradezco a mis hermanas, Mishell y Jhoana, quienes me han acompañado en momentos buenos y malos, con su cariño y un simple abrazo me ayudan a sobre llevar las cosas.

A ustedes, Alexander y Paulette; mi pequeña y nueva familia, son la fuerza necesaria que he necesitado para no rendirme.

Agradezco a toda mi familia, quienes, con sus consejos, palabras de aliento y el carisma que los caracteriza logran generar en mí el impulso que necesito para conseguir mis ideales.

Agradezco, a mis amigos quienes hicieron de la vida universitaria una experiencia inolvidable.

Agradezco, de manera especial a mi director MSc. Edgar Alberto Maya Olalla y mi asesor MSc. Luis Edilberto Suárez Zambrano quienes me brindaron su tiempo, compartieron sus conocimientos y asesoramiento, en este trabajo de titulación.

Evelyn Patricia Bonilla Fonte



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS DEDICATORIA

Quiero dedicar este trabajo a todos los que confiaron en mí, a quienes de una u otra forma aportaron granitos de arena para convertirme en la persona y profesional que soy actualmente.

A Dios por guiarme y darme la fortaleza necesaria para superar cada una de las pruebas que ha puesto en mi vida.

A mis padres, por ser los principales pilares de mi vida, quienes con su humildad, esfuerzo y trabajo me han enseñado que nada es imposible. Por ese amor incondicional, por luchar incansablemente por verme cumplir cada una de mis metas, y por enseñarme el verdadero valor de las cosas.

A mi hija, Paulette por ser el motor de mi vida, quien me motiva día tras día a ser mejor persona, madre y profesional. Quién me ha mantenido firme durante todo este proceso para alcanzar mi meta.

Con cariño les dedico mi trabajo.

Evelyn Patricia Bonilla Fonte

RESUMEN

Este proyecto describe el desarrollo de un sistema que permite mejorar la eficiencia de un sistema de riego por aspersión basado en redes de sensores y redes neuronales artificiales.

El proyecto se divide en dos iteraciones, en la primera iteración se diseña la red de sensores inalámbricos WSN con tecnología LoRa para el monitoreo de variables ambientales que influye en el desarrollo del cultivo de pasto.

La red WSN está compuesta por un nodo sensor, un nodo central y un nodo actuador.

El nodo sensor consta de un sensor de humedad del suelo, un sensor de radiación UV, un sensor de temperatura y un sensor de lluvia. Los datos recolectados por el nodo sensor son enviados al nodo central o Gateway a través de comunicación LoRa donde los datos son procesados y luego enviados a una plataforma en la nube llamada ThingSpeak utilizando el protocolo MQTT donde los datos son visualizados en forma de gráficos.

En la segunda iteración se realiza el entrenamiento de la red neuronal con dos variables recolectadas por el nodo sensor. La red neuronal realiza predicciones del estado del riego, es decir, predice cuando el cultivo necesita de riego. Posteriormente, el nodo actuador ejecuta la orden enviada desde el nodo central; activa o desactiva la electroválvula permitiendo o denegando el flujo de agua hacia los aspersores.

Finalmente se realizan pruebas de funcionamiento. Una comparación entre el sistema de riego manual y el sistema automatizado con la red neuronal, donde se verifica que el sistema automatizado reduce considerablemente el consumo de agua y aporta en el desarrollo del cultivo de pasto.

ABSTRACT

This project describes the development of a system that allows to improve the efficiency of a sprinkler irrigation system based on sensor networks and artificial neural networks.

The project is divided in two iterations, on the first iteration the wireless sensor network with LoRa technology is designed for environmental variables monitoring that influence in the grass crop development.

The WSN network is composed of a sensor node, a central node and an actuator node.

The sensor node consists of a soil moisture, UV radiation, temperature, and a rain sensor. The data collected by the sensor node is sent to the central node or Gateway through LoRa communication where data is processed, and it is sent to a cloud platform called ThingSpeak using MQTT protocol where data is displayed in the form of graphs.

In the second iteration, the neural network is trained with two collected variables by the sensor node. The neural network makes predictions of irrigation status, that is to say, it predicts when the crop needs irrigation. Subsequently, actuator node executes order sent from central node; activates or deactivates the solenoid valve allowing or denying the water flow to the sprinklers.

Finally, functional tests are performed. A comparison between the manual irrigation system and the automated system with the neural network, where is verified that automated system considerably reduces water consumption and contributes to the development grass crop.

ÍNDICE DE CONTENIDOS

CAPÍTULO I. Antecedentes.....	16
1.1. Introducción.....	16
1.2. Tema.....	16
1.3. Problema.....	16
1.4. Objetivos:	19
1.4.1. Objetivo General.....	19
1.4.2. Objetivo Específico.....	19
1.5. Alcance	20
1.6. Justificación	22
CAPÍTULO II. Revisión Bibliográfica.....	24
2.1. Introducción.....	24
2.2. Agricultura y Ganadería de Precisión	24
2.3. Tecnologías de la Información y Comunicación TIC’S en la Agricultura y Ganadería de Precisión	25
2.4. Red de Sensores Inalámbricos	26
2.4.1. Arquitectura de la red de sensores WSN	26
2.4.2. Topologías de las Redes de Sensores Inalámbricos	28
2.5. Internet de las Cosas (IoT)	29
2.5.1. Arquitectura IoT	30
2.5.2. Protocolos de Comunicación de IoT.....	31
2.6. Low Power Wide Area Network (LPWAN)	33
2.6.1. LoRa.....	34
2.6.2. LoRa WAN	41
2.7. Redes Neuronales Artificiales	43
2.7.1. Modelo de una Red Neuronal Artificial.....	45
2.7.2. Estructura de una Red Neuronal Artificial.....	47
2.7.3. Mecanismos de Aprendizaje de Redes Neuronales.....	48
2.7.4. Algoritmos de Optimización	50
2.7.5. Validación de una Red Neuronal.....	52
2.7.6. Aplicaciones de las Redes Neuronales	52
2.8. Importancia de los Pastos y Forrajes.....	52
2.8.1. Factores que Influyen en la Producción y Calidad de Pastos y Forrajes	54
2.8.2. Importancia del Riego en Cultivos de Pasto.	55
CAPÍTULO III. Desarrollo de la Propuesta	58
3.1. Modelo Iterativo: Primera Iteración.....	59
3.1.1. Requerimientos del Sistema	59
3.1.2. Análisis de la Situación Actual	73
3.1.3. Diseño del Sistema Propuesto: Primera Iteración	75
3.1.4. Diseño de la Comunicación Inalámbrica	88
3.1.5. Codificación del Software del Sistema	105

3.1.6.	Pruebas del sistema: Primera Iteración	116
3.1.7.	Conclusiones: Primera Iteración.....	139
3.2.	Modelo Iterativo: Segunda Iteración.....	141
3.2.1.	Requerimientos del Sistema	141
3.2.2.	Análisis de situación actual.....	144
3.2.3.	Diseño del Sistema Propuesto	145
3.2.4.	Codificación del Sistema	148
3.2.5.	Pruebas del sistema: Segunda Iteración.....	169
3.2.6.	Conclusiones: Segunda iteración.....	174
CAPÍTULO IV. Implementación del Sistema		176
4.1.	Análisis del Consumo de Agua al Implementar el Sistema Automatizado.....	180
4.2.	Análisis del Cultivo de Pasto al Implementar el Sistema Automatizado	184
4.3.	Costo-Beneficio	187
CAPÍTULO V. Conclusiones, Recomendaciones y Trabajo a Futuro		189
5.1.	Conclusiones.....	189
5.2.	Recomendaciones	191
5.3.	Trabajo a Futuro.....	192
REFERENCIAS.....		193
ANEXOS		197
Anexo 1.	Dragino LoRa Shield	197
Anexo 2.	Hoja de especificaciones del Transceptor LoRa SX1276.....	198
Anexo 3.	Simulaciones de Radio Enlace	200
Anexo 4.	Cálculos del tiempo en el Aire de la transmisión.....	204
Anexo 5.	Análisis del Espectro en Keyling Analyzer	207
Anexo 6.	Instalación de Anaconda Navigator.....	210
Anexo 7.	Dataset para el entrenamiento de la red Neuronal.....	213
Anexo 8.	Base de datos de ThingSpeak.....	214
Anexo 9.	Manual de Usuario del Sistema.....	215
Anexo 10.	Código de la red WSN LoRa.....	220
Anexo 11.	Código de la Red Neuronal en Jupyter Notebook	227

ÍNDICE DE FIGURAS

Figura 1. Arquitectura de un nodo sensor	27
Figura 2. Topología de red WSN.....	29
Figura 3. Arquitectura de IoT.....	30
Figura 4. Arquitectura del protocolo MQTT.....	32
Figura 5. Paquetes del protocolo MQTT.....	33
Figura 6. Modelo de referencia OSI siete capas.....	35
Figura 7. Trama de la capa física LoRa.....	36
Figura 8. Ejemplo de modulación LoRa.....	38
Figura 9. Tasa de codificación con SF 8.....	40
Figura 10. Modelo de referencia LoRaWAN.....	41
Figura 11. Arquitectura de red LoRaWAN.....	42
Figura 12. Estructura de una neurona biológica.....	43
Figura 13. Sinapsis de una neurona biológica.....	44
Figura 14. Modelo de una red neuronal artificial.....	45
Figura 15. Tipos de funciones de activación.....	47
Figura 16. Estructura en capas de la red neuronal.....	48
Figura 17. Mecanismo de aprendizaje de la red neuronal.....	49
Figura 18. Ejemplo del descenso del gradiente.....	51
Figura 19. Partes de una gramínea.....	53
Figura 20. Esquema de riego por aspersión.....	57
Figura 21. Modelo de desarrollo iterativo.....	58
Figura 22. Extensión del cultivo para implementar el sistema.....	73
Figura 23. Vista aérea de la parcela y centro de control.....	74
Figura 24. Modelo de referencia IoT aplicado al sistema de riego.....	76
Figura 25. Mapas de pines de Arduino Uno.....	78
Figura 26. IDE de Arduino.....	78
Figura 27. Sensor de humedad del suelo capacitivo.....	79
Figura 28. Sensor de temperatura LM35.....	80
Figura 29. Sensor de radiación UV GUVVA S12SD.....	81
Figura 30. Esquema de montaje del sensor de lluvia YL-83.....	81
Figura 31. Mapa de pines de Dragino LoRa Shield.....	82
Figura 32. Diagrama esquemático del nodo sensor.....	83
Figura 33. Componentes Raspberry Pi3 Modelo B.....	84
Figura 34. Diagrama esquemático del nodo central.....	85
Figura 35. Arquitectura general del sistema.....	86
Figura 36. Placa de alimentación nodo sensor.....	88
Figura 37. Diagrama de red del sistema.....	89
Figura 38. Zona de Fresnel.....	95
Figura 39. Creación de las unidades con el nombre identificativo de cada nodo.....	98
Figura 40. Configuración de los prametros principales de la red.....	99
Figura 41. Configuración de los parametros principales del sistema de enlace.....	100
Figura 42. Línea de vista entre el nodo sensor y nodo central.....	101

Figura 43. Zona de fresnel del enlace entre el nodo sensor y nodo central en archivo RMPath.....	102
Figura 44. Zona de fresnel radio enlace entre el nodo central y nodo actuador.....	103
Figura 45. Tabla de índice UV	106
Figura 46. Librerías SPI.h y RH-RF95.h para arduino Uno	107
Figura 47. Configuraciones básicas para el funcionamiento del módulo LoRa-Nodo Sensor.....	109
Figura 48. Diagrama de flujo del nodo sensor	110
Figura 49. Configuraciones básicas modulo LoRa nodo central.....	111
Figura 50. Diagrama de flujo del nodo central.....	112
Figura 51. Arquitectura MQTT Publicación/Suscripción del sistema	113
Figura 52. Creación de un canal en ThingSpeak	114
Figura 53. IDE del canal y claves API de lectura y escritura.....	115
Figura 54. Configuración básicas para utilizar el protocolo MQTT.....	115
Figura 55. Visualización de datos publicados por el cliente MQTT en ThingSpeak ...	116
Figura 56. Nodo sensor alimentado con una batería de litio.....	117
Figura 57. Nodo sensor alimentado con una batería de litio y panel solar	118
Figura 58. Datos recolectados por el nodo sensor en horas de la mañana	121
Figura 59. Datos recolectados por el nodo sensor en horas de la tarde	122
Figura 60. Envío y recepción de datos nodo sensor-puerta de enlace	123
Figura 61. Presentación de datos en la plataforma ThingSpeak.....	124
Figura 62. Distancia entre la puerta de enlace y el nodo sensor, punto mas lejano	125
Figura 63. Datos recolectados a una distancia de 205.17 metros.....	126
Figura 64. Calculadora LoRa para el cálculo de tiempo en el aire.....	129
Figura 65. Zona espectral con frecuencia central de 915 Mhz.....	131
Figura 66. Espectro de la señal al variar el factor de propagación de SF7 a SF12.....	132
Figura 67. Paquetes transmitidos del nodo sensor al nodo central con una frecuencia de 915Mhz	133
Figura 68. Señal LoRa en una frecuencia de 915 Mhz	134
Figura 69. Tiempo en el aire de un paquete LoRa utilizando la herramienta Matlab ..	135
Figura 70. Mensajes del protocolo MQTT-Connect Command.....	136
Figura 71. Mensaje del protocolo MQTT-Connect ACK	137
Figura 72. Mensaje del protocolo MQTT-Publish	138
Figura 73. Mensaje del protocolo MQTT-Disconnect.....	139
Figura 74. Diagrama esquemático del nodo actuador.....	144
Figura 75. Diagrama de flujo de la aplicación de la red neuronal al sistema.....	146
Figura 76. Diagrama esquemático -planta de alimentación del nodo actuador.....	147
Figura 77. Etapas para el entrenamiento de la red neuronal	148
Figura 78. Tratamiento de datos de humedad del suelo y radiación UV	149
Figura 79. Conjunto de datos para el entrenamiento de la red neuronal.....	151
Figura 80. Modelo de la red neuronal con una capa oculta.....	152
Figura 81. Modelo de la red neuronal con dos capas ocultas.....	153
Figura 82. Diagrama de flujo para determinar el número de neuronas y capas ocultas	154
Figura 83. Anaconda Navigator3- interfaz gráfica.	156

Figura 84. Librerías necesarias para el entrenamiento de la red neuronal	157
Figura 85. Preprocesamiento de datos	157
Figura 86. Representación gráfica de datos de entrenamiento	158
Figura 87. Modelo de la red neuronal con la forma de entradas y salidas	161
Figura 88. Función de activación de la red neuronal	162
Figura 89. Hiperparámetros de la fase de entrenamiento de la red neuronal, fase de compilación	163
Figura 90. Hiperparámetros de la fase de entrenamiento, fase de ajuste	164
Figura 91. Historial de entrenamiento del modelo	165
Figura 92. Representación gráfica de la pérdida de magnitud del conjunto de datos de entrenamiento	166
Figura 93. Representación gráfica de la predicción de la red neuronal	167
Figura 94. Diagrama de flujo del nodo actuador	168
Figura 95. Configuraciones básicas del nodo actuador.....	169
Figura 96. Nodo actuador alimentado con una batería de litio.....	170
Figura 97. Predicción de la red neuronal-Regar	172
Figura 98. Predicción de la red neuronal-No regar.....	173
Figura 99. Lectura del nodo actuador en estado “Regando”	173
Figura 100. Lectura del nodo actuador en estado “Riego Detenido”	174
Figura 101. Nodos que conforman la red de sensores inalámbricos WSN	177
Figura 102. Implementación de los nodos sensor, central y actuador	177
Figura 103. Visualización de datos recolectados por el nodo sensor y estado de riego	178
Figura 104. Ubicación de los nodos en la parcela	179
Figura 105. Base de datos del estado de riego recuperado de ThingSpeak	181
Figura 106. Horario de riego más y menos frecuente del sistema automatizado	183
Figura 107. Cultivo de pasto en etapa de defoliación.....	184
Figura 108. Cultivo de pasto con el sistema manual Vs sistema de riego automatizado con la red neuronal.....	185
Figura 109. Tamaño de la raíz, sistema de riego manual Vs automatizado con la red neuronal.....	186

ÍNDICE DE TABLAS

Tabla 1. Comparación de LoRa, SigFox, NB-IoT y LTE-M	34
Tabla 2. Relación entre SF y longitud de chirridos	39
Tabla 3. Nomenclatura de los requerimientos del diseño.	60
Tabla 4. Lista de Stakeholders.....	60
Tabla 5. Requerimientos de Stakeholders	61
Tabla 6. Requerimientos funcionales del sistema.....	62
Tabla 7. Requerimientos de arquitectura	63
Tabla 8. Selección de hardware del nodo sensor	65
Tabla 9. Elección de módulo LoRa para el nodo sensor.....	66
Tabla 10. Elección de sensor de humedad del suelo para el nodo sensor	67
Tabla 11. Elección de sensor de temperatura para el nodo sensor	68
Tabla 12. Elección de sensor de radiación UV para el nodo sensor.....	69
Tabla 13. Comparativa de sensores de lluvia para el nodo sensor	70
Tabla 14. Selección de placas de procesamiento para el diseño del Gateway o puerta de enlace	71
Tabla 15. Comparación de plataformas en la nube para el nodo central	72
Tabla 16. Datos del caudal y consumo de agua del sistema de riego manual actual.....	75
Tabla 17. Consumo energético de nodo sensor	87
Tabla 18. Coordenadas de cada nodo que conforma la red.....	90
Tabla 19. SNR en función del factor de propagación.....	93
Tabla 20. Escenarios simulados radio enlace nodo sensor -nodo central	102
Tabla 21. Escenarios simulados radio enlace, nodo central – nodo actuador	104
Tabla 22. Comparación de valores calculados y simulados del enlace de radio	105
Tabla 23. Test de alimentación nodo sensor	119
Tabla 24. Rangos establecidos para las variables ambientales.....	120
Tabla 25. Tiempo en el aire y sensibilidad del receptor al variar el SF.....	130
Tabla 26. Selección del software de programación para la red neuronal artificial.....	141
Tabla 27. Elección de electroválvula selenoide para el nodo actuador	142
Tabla 28. Elección de módulos relé para el nodo actuador.....	143
Tabla 29. Consumo energético de los dispositivos que conforman el nodos actuador.	147
Tabla 30. Decisión de riego de acuerdo con las variables de humedad del suelo y radiación UV	150
Tabla 31. Modelos para el entrenamiento de la red neuronal	159
Tabla 32. Test de alimentación del nodo actuador	170
Tabla 33. Consumo de agua en el sistema de riego manual Vs sistema automatizado con la red neuronal	182
Tabla 34. Costo de hardware del sistema	187
Tabla 35. Costo de software del sistema	188

ÍNDICE DE ECUACIONES

Ecuación 1. Tiempo en el aire.....	40
Ecuación 2. Función de entrada de una red neuronal	46
Ecuación 3. Funcion de salida de una red neuronal.....	46
Ecuación 4. Descenso del gradiente.....	51
Ecuación 5. Voltaje de salida del sensor GUVA 12SD	80
Ecuación 6. Pérdidas en el espacio libre	91
Ecuación 7. Potencia del receptor.....	91
Ecuación 8. Sensibilidad del receptor	93
Ecuación 9. Margen de enlace	94
Ecuación 10. Presupuesto de enlace	95
Ecuación 11. Fórmula para calcular el radio de la zona de Fresnel	96
Ecuación 12. Cálculo del ángulo de azimut	96
Ecuación 13. Diferencia entre la variación de longitud y latitud	96
Ecuación 14. Ángulo de elevación	97
Ecuación 15. Diferencia entre la variación de longitud y latitud	97
Ecuación 16. Fórmula para calcular la temperatura del sensor LM35	107
Ecuación 17. Fórmula para calcular el tiempo del preámbulo	127
Ecuación 18. Fórmula para calcular el tiempo del símbolo	127
Ecuación 19. Fórmula para calcular el número de símbolos en la carga útil	127
Ecuación 20. Tiempo de payload.....	128
Ecuación 21. Regla de la pirámide geométrica	152
Ecuación 22. Número de neuronas en la primera capa oculta.....	153
Ecuación 23. Número de neuronas en la segunda capa oculta	153
Ecuación 24. Precisión del modelo de la red neuronal	164

CAPÍTULO I. Antecedentes

1.1. Introducción

En el presente capítulo se detallan las bases y justificación para el desarrollo del trabajo de titulación, por consiguiente, se establece el tema, se describe el problema, se presentan los objetivos que justifican el proceso a seguir, así como el alcance y las limitaciones a las que se sujeta la investigación.

1.2. Tema

MEJORAMIENTO DE LA EFICIENCIA DE UN SISTEMA DE RIEGO POR ASPERSIÓN BASADO EN REDES DE SENSORES Y REDES NEURONALES ARTIFICIALES PARA EL CULTIVO DE PASTO EN LA PARROQUIA DE TOCACHI PERTENECIENTE AL CANTÓN PEDRO MONCAYO.

1.3. Problema

El cantón Pedro Moncayo, es conocido como Capital mundial de la Rosa, debido a que el 25% de la producción de rosas que exporta Ecuador son generadas en las 140 fincas de este cantón, sin embargo, la actividad ganadera en esta parte de la región también es importante, tanto como la de las flores. En los últimos años la producción agrícola ha disminuido considerablemente debido a que sus habitantes se han dedicado intensivamente a la ganadería, es habitual encontrar planicies de ganado vacuno al arribar a este cantón, los ganaderos son parte fundamental del crecimiento económico cantonal tanto con la producción de leche como carne. (Castro & Espin, 2016). Por lo que se puede asegurar que el pasto es el cultivo predominante en el cantón Pedro Moncayo.

Ya que toda la producción animal se basa en la alimentación por medio de pastos y forrajes, es necesario obtener pastos con gran valor nutritivo y en grandes cantidades, con el fin de proporcionar proteínas y carbohidratos al ganado para que

produzcan más leche y carne. La duración y el rendimiento de los pastos dependen de varios factores como la planificación del cultivo, fertilización, corte adecuado y un factor muy importante que es el riego (Chacón Carlos, 2015), puesto que las plantas en sus etapas iniciales son muy susceptibles a la falta o al exceso de humedad, se debe contar con humedad constante, ni exceso ni deficiencia para que los cultivos puedan alcanzar su desarrollo normal.

Tocachi es una parroquia rural perteneciente al cantón Pedro Moncayo, ubicado al norte de la provincia de Pichincha, esta parroquia posee una asociación lechera formado por pequeños ganaderos de la zona, que cuentan con cabezas de ganado lechero que demandan para su alimentación cantidades de forraje verde, siendo el cultivo de pasto el forraje preferido, además de ser económico los nutrientes de este tipo de cultivo genera gran producción de leche.

Por esta razón, los ganaderos de la zona se han dedicado al cultivo de pasto y han implementado sistemas de riego en sus extensiones de terreno. La técnica de riego que utilizan es por aspersión, irrigan sus cultivos en base al tradicionalismo, los aspersores están ubicados en diferentes partes del cultivo brindando una gran cantidad de agua a todo el pastizal, pero sin una distribución adecuada para evitar el desperdicio de agua ya que no pueden determinar que porción del terreno realmente necesita ser regada y la cantidad apropiada para este tipo de cultivos. Además, los factores climáticos tales como la temperatura, lluvia y radiación solar afectan de forma importante a la producción y calidad del pasto (RED IAPD, 2011), por tanto, la persona encargada del cultivo debe tomar muy en cuenta estos parámetros para un manejo adecuado de pasturas.

A pesar de que el clima es el principal factor que controla el desarrollo y el crecimiento de los pastos y forrajes, el hombre tiene poco control sobre este factor, más

aún con el cambio climático. En el cantón Pedro Moncayo la precipitación media anual va de 400 a 1300 milímetros, con un periodo seco entre los meses de julio y octubre, que representa un déficit hídrico de 0 a 330 milímetros (Hidalgo et al., 2018). En la mayoría de los ecosistemas del país, la tasa de crecimiento diaria de pasto en verano es 5-20m kg/Ms/ha/día, en tanto que en el invierno la tasa sube hasta niveles de 60-80 kg/Ms/ha/día. (Leon et al., 2018). Por ello, la aplicación de agua de riego es muy importante para aumentar la producción de pasto y al mejorar la eficiencia del sistema de riego por aspersión permitirá reducir el consumo innecesario de agua.

Se propone el diseño de un sistema de riego inteligente, para el cultivo de pasto que utiliza el método de irrigación por aspersión, este sistema recolectará datos de factores climáticos antes mencionados, a través de una red de sensores, estos datos serán enviados a una plataforma IoT, llamada ThingSpeak donde serán almacenados. Además, los datos almacenados servirán para realizar el entrenamiento de la red neuronal y obtener predicciones de cuando el cultivo necesita ser irrigado. Una vez que el sistema determine si el cultivo necesita de riego, se enviará de forma inalámbrica la orden para que una electroválvula active de forma automática el flujo de agua hasta cuando considere que la dosis de agua suministrada en el cultivo sea suficiente, evitando de esta manera el consumo innecesario de agua y que el pasto tenga un desarrollo adecuado.

1.4. Objetivos:

1.4.1. Objetivo General

Diseñar un sistema para controlar el riego por aspersión de forma automática en cultivos de pasto, basado en redes de sensores y redes neuronales artificiales el cual, mediante predicciones realizadas por el sistema de cuando activar el riego permita reducir el consumo innecesario de agua y lograr un buen desarrollo de la planta.

1.4.2. Objetivo Especifico

Realizar una investigación bibliográfica sobre las bases teóricas necesarias para la realización del sistema.

Determinar los requerimientos de Software y Hardware para el correcto funcionamiento del sistema.

Diseñar un sistema de monitorización basado en redes de sensores utilizando módulos LORA.

Diseñar un sistema de riego inteligente basado en el algoritmo de redes neuronales artificiales.

Realizar las pruebas de funcionamiento del sistema de riego inteligente para la presentación de resultados.

1.5. Alcance

El presente proyecto tiene como finalidad diseñar un sistema de riego inteligente para cultivo de pasto basado en redes de sensores y redes neuronales con el propósito de reducir el consumo innecesario de agua y lograr un buen desarrollo del cultivo, partiendo de la investigación bibliográfica sobre WSN, tecnologías LPWAN, IoT y Aprendizaje automático.

Se utilizará el modelo iterativo como metodología, para el correcto desarrollo del software y hardware, donde, para cada una de las iteraciones, se construye una parte pequeña del sistema. Para esta parte del sistema, se realiza todo el proceso; análisis, diseño, programación y pruebas. Se terminan la iteración con un ejecutable con todas las partes del sistema construidas hasta ese momento.(Palasi Llallana, 2004)

En la primera parte del proyecto se diseñará un sistema de monitorización basada en redes de sensores con tecnología LPWAN, esta tecnología está ganando cada vez más popularidad en los comunicados industriales de investigación debido a sus características de comunicación de bajo consumo, largo alcance y bajo costo.(Kais Mekki, Eddy Bajic, Frederic Chaxel, 2018)

Este sistema constará de una placa programable con módulo LoRa para la comunicación inalámbrica entre nodos y puerta de enlace. El nodo sensor tendrá sensores de humedad del suelo, temperatura, lluvia y radiación UV, estará ubicado en el cultivo de pasto con una expansión de $1000m^2$. Una vez recopilada la información por la WSN, se enviará los datos al Gateway LoRa para luego almacenarlos en una plataforma IoT, llamada ThingSpeak que permite recopilar, visualizar y analizar datos en tiempo real utilizando el protocolo MQTT como punto central de comunicación (Nettikadan & Raj, 2018), este es un protocolo cliente-servidor sobre el que se publican/suscriben los mensajes entre los dispositivos interconectados. Está orientado

principalmente a la comunicación de sensores, debido a su bajo consumo de banda ancha y mínima sobrecarga puede ser utilizado en dispositivos con pocos recursos. Los mensajes se transmiten por TCP-IP (Ballesteros de Andrés et al., 2017).

La segunda parte del proyecto es el diseño de un sistema de riego inteligente, donde los datos recolectados de la WSN servirán como variables de entrada de la red neuronal artificial la cual obtiene experiencia analizando automáticamente y sistemáticamente los datos para determinar reglas de comportamiento, con base a ellas, realizar predicciones y determinar si el cultivo requiere riego o no. La RNA tiene sus bases y funcionan de forma similar a las redes neuronales biológicas del cerebro de una persona (Requena et al., 2017). Este proceso se realizará en un Raspberry Pi 3 el cual recibirá los datos de la WSN para procesar la información e implementar el algoritmo de redes neuronales mediante el software de programación Python. Respectivamente, la orden de la decisión tomada por el algoritmo de redes neuronales será enviada de forma inalámbrica mediante un módulo LoRa a un nodo actuador el cual se encargará de permitir o denegar el flujo de agua, esto mediante el uso de una electroválvula el cual será controlada por un relé. El caudal de agua a controlar es de 18 l/min.

Para comprobar si la cantidad de agua suministrada en el cultivo de pasto es adecuada, se procederá a realizar varias pruebas de funcionamiento. Además, para evidenciar si se obtuvo un buen desarrollo de la planta con la implementación del sistema de riego inteligente se realizarán pruebas comparativas con otro cultivo de pasto que utiliza un sistema de riego por aspersión de forma manual. En este caso se realiza un análisis del cultivo para evidenciar parámetros como la coloración y tamaño de las hojas, altura de la planta y tamaño de la raíz.

1.6. Justificación

Según el Ministerio de Agricultura y ganadería, uno de los objetivos del Gobierno Nacional, es manejar integralmente los recursos hídricos con enfoque de cuenca hídrica, a través del desarrollo de políticas públicas integrales de manejo del recurso agua con criterios de racionalidad social y equidad entre poblaciones y regiones, y entre campo ciudad; siendo acciones relevantes: garantizar agua segura y apta para el consumo de todos los hogares de país, ampliar la superficie regada e incrementar la producción agropecuaria en estas zonas, generar capacidades para alcanzar una visión prospectiva de las demandas del recurso hídrico y su impacto sobre las fuentes y los ecosistemas.(MAGAP, 2014)

Uno de los mejores usos que se le puede dar al recurso hídrico es dedicarlo a la producción agropecuaria, donde en términos medios, la irrigación eleva en cuatro o cinco veces la productividad de la tierra, y a su vez, con la ayuda de la ciencia y tecnología amplían las áreas cultivables a través de diversos sistemas de mejoramiento de suelos y regadíos.

A demás, la Constitución señala que uno de los objetivos de la política económica es la de asegurar la soberanía alimentaria, lo que implica incentivar, en iguales condiciones, la producción convencional y la producción agroecológica de la Agricultura Familiar Campesina, a través de la redistribución de factores de producción, esto acompañado por cambios tecnológicos que consideren investigación, desarrollo e innovación. Donde la investigación y desarrollo se apoyará en el contingente de las universidades y centros de investigación, los cuales deben responder con pertinencia y oportunidad a través de la creación de conocimiento a las necesidades de los habitantes rurales.(SENPLADES, 2017)

Por lo antes expuesto, una de las maneras de contribuir con estas políticas de Estado, es mediante la optimización del riego, para esto se pretende hacer uso de la inteligencia artificial y diseñar un sistema que permita controlar el riego de forma automática, reduciendo así el uso innecesario del agua y conseguir un buen desarrollo del cultivo de pasto en la parroquia rural Tocachi del cantón Pedro Moncayo, ya que la alimentación es el pilar para el desarrollo de la ganadería lechera. El logro para incrementar la producción y productividad de leche necesariamente pasa por aplicar innovaciones tecnológicas orientadas a mejorar el manejo adecuado de pastos y forrajes. (FAO, 2011)

CAPÍTULO II. Revisión Bibliográfica

2.1. Introducción

En este capítulo, se presentan las bases teóricas para el desarrollo del trabajo de investigación, se abordan contenidos referentes a la agricultura y ganadería de precisión, la importancia de cultivos de pastos y forrajes, los factores que influyen en el desarrollo de este y la importancia de un sistema de riego; por otro lado, se describen las redes WSN, el Internet de las Cosas (IoT) y las tecnologías inalámbricas de largo alcance y bajo consumo energético (LPWAN). Además, se trata conceptos de aprendizaje automático, específicamente sobre Redes Neuronales Artificiales que tienen el objetivo de determinar cuándo debe aplicarse el riego. Finalmente se hace un análisis de las metodologías utilizadas para el diseño de proyectos tecnológicos.

2.2. Agricultura y Ganadería de Precisión

La agricultura de precisión se refiere a las herramientas o estrategias que permiten a los agricultores optimizar y aumentar la productividad del suelo. Se denomina “precisión” porque gracias al uso de tecnologías avanzadas permite la intervención correcta, en el lugar correcto y en el momento correcto respondiendo a las demandas específicas del cultivo y áreas individuales del terreno con altos niveles de precisión (Santillán & Rentería, 2018). De esta manera se puede satisfacer una de las exigencias de la agricultura moderna que es, el manejo óptimo de grandes extensiones.(Montecé et al., 2020).

Las tecnologías de la agricultura de precisión se utilizan en primer lugar para la recolección de datos del terreno, cultivo, clima, etc. De estos datos se extrae información útil para ser analizados y tratados para su interpretación, necesario para la toma de decisiones. Y en segundo lugar para implementar las acciones correctivas, se

debe aplicar los recursos necesarios o realizar las operaciones requeridas en el terreno o cultivo. (Santillán & Rentería, 2018)

Por otro lado, la ganadería de precisión (GdP) es un sistema sostenible que permite producir alimento de origen animal (leche, carne) con la ayuda de tecnologías de la información y comunicación (TIC'S). Mediante la GdP se recolectan y analizan datos en tiempo real para tomar decisiones que permiten mejorar la producción pecuaria. A nivel mundial, el sector pecuario que han adoptado en mayor medida la GdP es el de los productores de leche bovina, también los productores de carne, aunque en menor proporción. (Santillán, 2018)

La agricultura y ganadería de precisión son dos opciones modernas y sostenibles que ayudan a producir alimentos de calidad y en cantidades suficientes.

2.3. Tecnologías de la Información y Comunicación TIC'S en la Agricultura y Ganadería de Precisión

Las tecnologías de la información y comunicación TIC'S contemplan al conjunto de herramientas relacionadas con la transmisión, procesamiento y almacenamiento digitalizado de la información, como al conjunto de procesos y productos derivados de las nuevas herramientas hardware y software.

El uso adecuado de las tecnologías de la información en la ganadería permite la gestión diaria del ganado contribuyendo a mejorar la vida de los animales, alertando de todo eventual problema al ganadero y él pueda actuar de forma inmediata tomando las mejores decisiones. Dentro de las tecnologías más utilizadas se encuentran RFID (Identificación por radio frecuencia), el uso de drones y aplicaciones móviles, etc. (Laguna, 2019)

En cuanto a la agricultura, permite lograr el uso eficiente de los insumos agrícolas para ello, los sistemas integrales de información son muy importantes ya que

permiten recopilar datos de los cultivos. Entre las tecnologías más destacadas se puede mencionar los sistemas de información geográfica (SIG), el uso de drones y las Redes de sensores inalámbricos (WSN). Las WSN en la agricultura de precisión permite determinar con gran exactitud las necesidades de riego y fertilizante, los puntos óptimos de siembra y recolección y demás actividades agrícolas primordiales.(López et al., 2016)

2.4. Red de Sensores Inalámbricos

Las redes de sensores inalámbricos (WSN), es un sistema diseñado para monitorear y controlar un fenómeno o evento específico de forma remota.

Consiste en un conjunto de nodos sensores denominados también motes que se encuentran distribuidos espacialmente y se comunican entre sí a través de comunicación inalámbrica (Santamaría & Archila, 2013) con el fin de detectar el entorno e impartir los datos acumulados del campo monitoreado.

Estas redes se caracterizan por su facilidad de despliegue, por ser auto configurables y su gestión eficiente de energía, que les permite obtener una alta tasa de autonomía que las hace plenamente operativas. (Fernández et al., 2009)

2.4.1. Arquitectura de la red de sensores WSN

(Fernández et al., 2009)menciona que una red de sensores inalámbricos WSN está compuesta por nodos inalámbricos, puertas de enlace y estaciones base.

El modelo seguido típicamente por las aplicaciones es la siguiente:

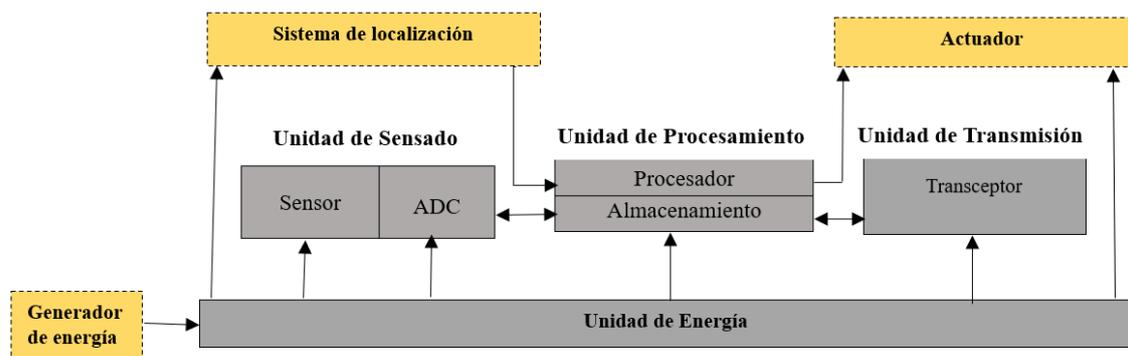
Realiza una serie de mediciones sobre el medio, transforma la información en digital en el propio nodo y la transmite fuera de la red de sensores vía un elemento Gateway (puerta de enlace) a una estación base donde será almacenada, y posteriormente será enviada a un servidor con mayor capacidad.

2.4.1.1. Nodo Inalámbrico.

Son dispositivos electrónicos de bajo costo, consumo y reducido tamaño, encargados de medir o detectar propiedades físicas o ambientales y convertirlos en señales para fines de monitorización y control. Un nodo sensor se compone básicamente de cuatro unidades básicas como se puede ver en la Figura 1, una unidad de transmisión, una unidad de sensado, una unidad de procesamiento y una unidad de energía (Hamami & Nassereddine, 2020).

Figura 1

Arquitectura de un nodo sensor



Adaptado de *Application of wireless sensor networks in the field of irrigation: A review* (p.3), por L. Hamami, B. Nassereddine, 2020, *Computer and Electronic in Agriculture*.

Unidad de Sensado: Es el elemento principal del nodo sensor, consta por lo general de dos subunidades que son: un sensor y un convertidor de analógico a digital (ADC) para medir y convertir datos.

Unidad de Energía: Es un elemento vital, suministra corriente a todos los componentes del nodo para su funcionamiento. Este puede ser abastecida por unidades captadoras de energía como los paneles solares.

Unidad de Procesamiento: Consiste en una unidad de computadora (procesador) y una unidad de almacenamiento (memoria).

Unidad de Transmisión: Maneja la transmisión y recepción de datos en la red a través de un medio inalámbrico.

Aunque puede tener componentes adicionales dependiendo de su aplicación como un sistema de localización, un generador de energía y actuadores.

2.4.1.2. Puerta de Enlace.

Es un dispositivo que tiene la funcionalidad de interconectar dos redes de diferente naturaleza, en este caso conectar las redes de sensores y una red de datos (TCP/IP) ya que en aplicaciones donde se usan redes de sensores, estas no pueden operar completamente aisladas y deben contar con alguna forma de monitoreo y acceso a la información adquirida por los nodos (Fernández et al., 2009).

2.4.1.3. Estación Base

Es un recolector de datos, basado en un ordenador común. Todos los datos van a pasar a un equipo servidor dentro de una base de datos, desde donde los usuarios podrán acceder remotamente a la información. (Fernández et al., 2009).

2.4.2. Topologías de las Redes de Sensores Inalámbricos

Las WSN se agrupan de diferentes formas con el fin de encontrar la mejor ubicación entre los nodos para el envío y recepción de datos. Básicamente los nodos WSN se organizan en tres tipos de topologías de red que son estrella, malla o una híbrida. (Nayibe et al., 2011)

En la topología en estrella todos los nodos están en comunicación directa con el Gateway. Su ventaja es la capacidad de mantener al mínimo el consumo de energía del nodo remoto, además de la baja latencia entre el nodo y la estación base.

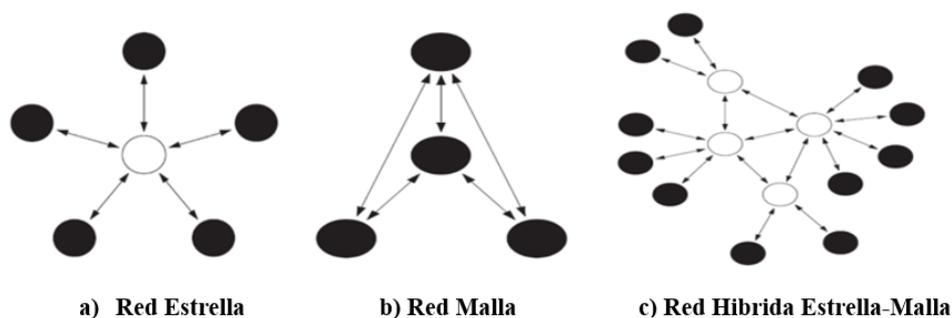
La topología en malla es un sistema multisalto, cada nodo está conectado a los demás nodos de la red. Cada nodo puede enviar y recibir información de otro nodo y del

Gateway. Esta topología tiene la ventaja de redundancia y escalabilidad.(Garg et al., 2011)

Una topología híbrida cubre las características de la topología en malla y estrella, es decir la simplicidad y el bajo consumo de energía. Además de cubrir una gran extensión y reorganizarse ante un fallo en algún nodo. (Garcia & Mora, 2010)

Figura 2

Topología de red WSN



Adaptado de *Review of Wireless Sensor Networks-Architecture and Applications* (p.36), por P. Garg, S. Kuntal, L. Ruchika, 2011, *International Journal of Computer Science and Management Studies*.

Las redes de sensores inalámbricos WSN se ha convertido en un gran método de recopilación de información para construir sistemas de información y comunicación que mejora la confiabilidad y eficiencia de los sistemas de infraestructura. En comparación con soluciones cableadas las WSN presentan una implementación mucho más sencilla y una mayor flexibilidad de los dispositivos. Con el rápido desarrollo tecnológico de los sensores, las redes de sensores inalámbricos se están convirtiendo en una tecnología clave para la implementación de internet de las cosas (IoT).(International Electrotechnical Commission, 2016)

2.5. Internet de las Cosas (IoT)

A pesar de que IoT es un tema generalizado, no existe una definición única y universalmente aceptada para el término, pero los conceptos se tornan similares.

Se puede definir al Internet de las cosas como una red de objetos físico, cosas, sensores y dispositivos conectados que se comunican entre sí a través de protocolos de comunicación, estos objetos recopilan datos que son analizados para proporcionar servicios de valor agregado a través del procesamiento inteligente de datos. Su objetivo principal es permitir que las cosas estén conectadas en cualquier momento, lugar, con cualquier cosa y persona.(Patel & Patel, 2016)

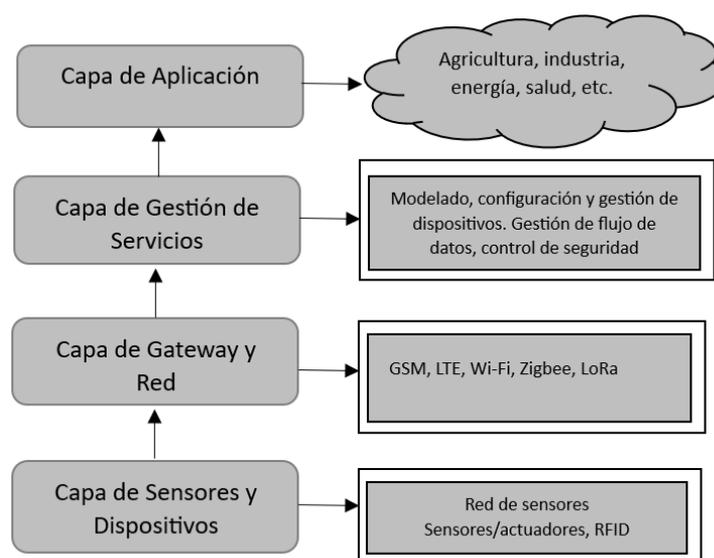
2.5.1. Arquitectura IoT

(Benítez David & Anías, 2016) la arquitectura de IoT debe cumplir un conjunto de principios, dentro de esos se encuentran: permitir escalabilidad, ampliación de capacidades y soporte de nuevos estándares, servir de modelo para la creación de arquitecturas más específica, y permitir la integración de diferentes soluciones IoT.

La arquitectura IoT consta de cuatro capas principales que se presenta en la Figura 3, estas son: Capa de sensores y dispositivos, Capa de Gateway y red, Capa de gestión de servicios y Capa de aplicación.

Figura 3

Arquitectura de IoT



Adaptado de *The Internet of Things: Vision, Architecture and Applications* (p.3), por P. Deshpande, A. Damkonde, V. Chavan, 2017, *International Journal of Computer Application*, 178(2)

La capa de dispositivos se encuentra en la base de la arquitectura, representa los sensores físicos de una aplicación IoT, tiene como objetivo recopilar información y sus principales funciones es sensar y actuar sobre el medio físico. (Benítez David & Anías, 2016).

En cuanto a la capa de Gateway y red, es la responsable de soportar la comunicación, debe admitir los grandes volúmenes de datos de IoT producidos por sensores inalámbricos y dispositivos inteligentes y transferirlos de forma segura al sistema de procesamiento de información.

La capa de gestión de servicios es la encargada del correcto funcionamiento del sistema IoT, procesa los datos recibidos, toma decisiones en función de los resultados obtenidos y brinda servicios requeridos al solicitante.

Finalmente, la capa de aplicación tiene como finalidad utilizar la información brindada por el sistema IoT y gestionar métodos de interacción con los usuarios para satisfacer sus necesidades. (Deshpande et al., 2017).

2.5.2. Protocolos de Comunicación de IoT

Los protocolos de comunicación de IoT son modos de comunicación que proporciona seguridad a los datos que se comparten entre dos o más dispositivos conectados. En IoT estos protocolos se puede clasificar en dos categorías.

Por un lado, se tiene a los protocolos de red de IoT, utilizados para conectar dispositivos a través de una red, normalmente a través de Internet. Permitiendo la comunicación de datos de extremo a extremo, entre ellos se puede mencionar a HTTP, Bluetooth, ZigBee y LoRaWAN.

Por otro lado, se encuentran los protocolos de datos de IoT, que se utilizan para conectar dispositivos IoT de baja potencia, estos protocolos proporcionan comunicación

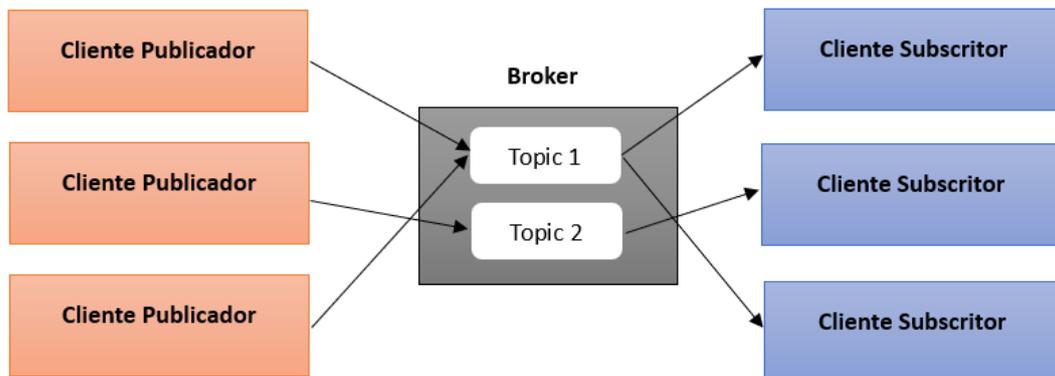
punto a punto con el hardware en el lado del usuario. Algunos de los protocolos de datos son MQTT, CoAP, M2M, etc. (Uppalapati, 2019)

2.5.2.1. Protocolo MQTT

El protocolo MQTT (Message Queuing Telemetry Transport) es utilizado para la comunicación donde el ancho de banda y recursos (CPU, RAM) son limitados. La comunicación puede cifrarse mediante protocolos TLS/SSL ya que funciona a través de TCP, garantizando la comunicación segura y protegida. En cuanto a su arquitectura sigue una topología de estrella donde un nodo central hace de servidor (broker) y es el encargado de gestionar la red y transmitir los mensajes para mantener el canal activo. Las operaciones básicas de este protocolo son: conectarse al broker, publicar un mensaje, suscribirse a un "topic"; creado por el cliente que publica el mensaje, y cancelar la suscripción de un determinado topic.

Figura 4

Arquitectura del Protocolo MQTT



Adaptado de *MQTT V5.0*, por OASIS Standar, 2019.

El protocolo MQTT opera intercambiando una serie de paquetes de control MQTT y consta de tres partes, como se muestra en la Figura 5.

Figura 5

Paquetes de control MQTT

Cabecera fija		Cabecera variable	Carga útil
Presente en todos los paquetes de control MQTT		Presente en algunos paquetes de control MQTT	Presentes en algunos paquetes de control MQTT
Cabecera de control	Longitud del paquete	El tamaño depende del tipo de mensaje	Contiene los datos reales del mensaje
1 byte	1-4 bytes	0 - Y bytes	0 - 256 bytes

Adaptado de *MQTT V5.0*, por OASIS Standar, 2019.

El creciente dominio de las tecnologías IoT se ve favorecida gracias a la vinculación de objetos físicos a internet mediante varios tipos de tecnología inalámbrica de corto alcance como ZigBee, RFID, redes de sensores y tecnologías basadas en la ubicación. Las tecnologías de corto alcance son las más apropiadas en IoT, pero estas tecnologías no son adecuadas para todas las aplicaciones. Algunas aplicaciones como la medición inteligente necesitan niveles de consumo y costos bajos, pero requieren comunicación IoT de largo alcance, por lo que se considera a LPWAN como una solución definitiva (Poursafar et al., 2017).

2.6. Low Power Wide Area Network (LPWAN)

La red de área amplia de baja potencia LPWAN es una de las mejores tecnologías de comunicación inalámbrica que permite una gama muy amplia de aplicaciones IoT, radica en su capacidad de transmitir una tasa de bits relativamente bajos en un intervalo de tiempo específico y ofrece conectividad de bajo consumo en un área de cobertura de larga distancia.

Operan en la región sub-GHz sin licencia del espectro, en comparación con otras bandas libres de licencia, la atenuación y el desvanecimiento por trayectos múltiples son

menores. (Sisinni & Mahmood, 2021). En la Tabla 1 se realiza un análisis de tecnologías líderes para redes LPWAN.

Tabla 1

Comparación de LoRa, SigFox, NB-IoT y LTE-M

CARACTERISTICAS	LoRa	SigFox	NB-IoT	LTE-M
Frecuencia	433-868-915 MHz	433-868-915 MHz	180 kHz	14 MHz
Ancho de banda del canal	125-500 kHz	100 Hz	200 kHz	1,4 MHz
Tiempo de vida de la batería	8-10 años	7-8 años	7-8 años	10 años
Eficiencia energética	Muy alto	Muy alto	Muy alto	Muy alto
Presupuesto de enlace	161-175 dB	146 dB	164 dB	155 dB
Modulación	CSS	BPSK	QPSK	QPSK/16QAM
Velocidad de datos	20-50 Kbps	100 kbps 128 bytes	100 kbps 128 bytes	100-150 Kbps 128 bytes
Seguridad	Alta	Baja	Alta	Alta
Rango de cobertura	2-5 Km urbano 15 Km sub -urbano 50 Km rural	3-10 Km urbano 30-50 Km rural	1-5 Km urbano 20-40 Km rural	10 km urbano 50Km rural
Interferencia	Muy alto	Bajo	Bajo	Bajo
Espectro	No licenciado	No licenciado	Con licencia celular	Con licencia celular
Estandarización	LoRa Alliance	ETSI	3GPP Rel. 13+	3GPP Rel. 13+
Latencia	1 seg	10 seg	1.5-10 seg	50-100 seg

Adaptado de *A survey of LPWAN Technology in Agricultural Field* (p.315), por M. Liya, D. Arjun, 2020.

2.6.1. LoRa

LoRa (Long Range) es una tecnología inalámbrica de capa física (PHY), según lo define el modelo de siete capas OSI que se muestra en la Figura 6. Fue introducida y patentada originalmente por Semtech, diseñada para conexiones de baja potencia a

grandes distancias, donde la cobertura de comunicaciones es escasa o incluso inexistentes.

Figura 6

Modelo de referencia OSI siete capas



Adaptado de *LoRa and LoRaWAN: A Technical Overview*, por Semtech Corporation, 2019.

Utiliza el medio para transportar ondas de radio LoRa desde un transmisor de radio frecuencia RF en un dispositivo IoT a un receptor de radio frecuencia RF en una puerta de enlace y viceversa. Las bandas de radio ISM sin licencia varía según la región del mundo, en Europa 863-870 MHz (en 8 canales de 0.3 MHz), Estados Unidos y Canadá 902-928 MHz (13 canales de 2.16 MHz).

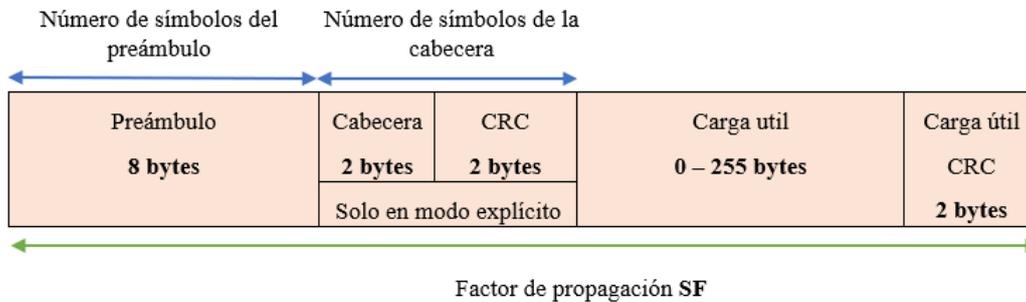
2.6.1.1. Trama de la Capa Física LoRa

La trama de la capa física LoRa consta de cuatro elementos que se puede identificar en la Figura 7, estas son: preámbulo, cabecera (header), CRC (verificación de

redundancia cíclica) y la carga útil (payload); aunque se puede incluir un último elemento, carga útil CRC para la integridad de los datos.

Figura 7

Trama de la capa física LoRa



Adaptado de *Semtech Wireless and Sensing Products*, (p.29), por SEMTECH, 2020

El preámbulo se utiliza para sincronizar el transmisor con el receptor, por defecto está configurada con una secuencia larga de 8 símbolos y puede extenderse hasta una longitud total de 65,532 símbolos; sin embargo, la longitud mínima es suficiente para todas las comunicaciones. El receptor lleva a cabo un proceso de detección de preámbulo que se reinicia periódicamente, por tal razón, la longitud del preámbulo debe configurarse idéntica a la longitud del preámbulo del transmisor.

En cuanto a la cabecera, dependiendo del modo de operación, hay dos tipos de encabezado disponible: encabezado explícito y encabezado implícito. El modo de encabezado explícito es el modo de funcionamiento predeterminado, aquí el encabezado proporciona información sobre la carga útil como: longitud de carga útil en bytes, la tasa de código de corrección de errores hacia delante, presencia de un CRC (comprobación de redundancia cíclica) de 16 bits opcional para carga útil.

El modo de encabezado implícito es utilizado en ciertos escenarios donde la carga útil, la tasa de codificación y la presencia de CRC son fijas o conocidas de antemano, puede ser una ventaja para reducir el tiempo de la transmisión. Estos parámetros se deben

configurar manualmente en ambos lados del enlace. Además, en este modo el encabezado se elimina del paquete.

La carga útil del paquete es un campo de longitud variable que contiene los datos reales codificados con la tasa de error especificada en el encabezado en modo explícito o en la configuración del registro en modo implícito. Se puede agregar un CRC opcional sobre la carga útil.

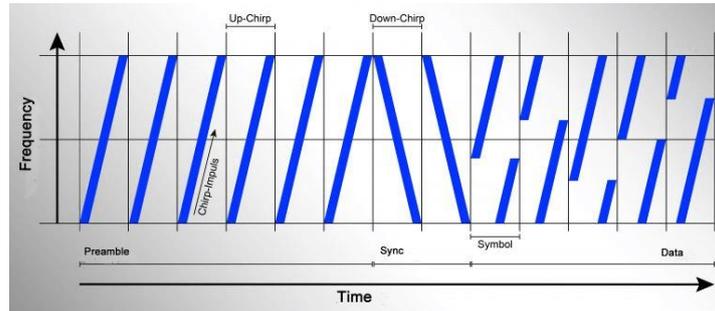
2.6.1.2. Modulación LoRa

Se basa en la modulación CSS (Chirp Spread Spectrum) una técnica de espectro ensanchado que utiliza pulsos de chirp modulados en frecuencia lineal de banda ancha para codificar la información. Los llamados pulsos de chirrido se envían como símbolos que aumenta o disminuye la frecuencia de LoRa de forma continua a lo largo del tiempo.

Como se puede observar en la Figura 8, cuando se procesa un mensaje LoRa, se transmite el preámbulo largo de “chirrido constante” para lograr la detección de la señal LoRa. En este ejemplo se envían 6 chirridos ascendentes (up-chirp) como símbolos de preámbulo, el final del preámbulo es seguido por el “chirrido inverso”, en este caso 2 chirridos descendentes (down-chirp) para la sincronización del tiempo. Finalmente comienza la transmisión de datos, que tienen una serie de señales moduladas que se parece a un “salto-chirrido”, saltar en la frecuencia representa los datos (carga útil).

Figura 8

Ejemplo de modulación LoRa



Adaptado de *Conceptos de actualidad LoRa y LoRaWAN*, por 2CI Group, 2015, (<https://www.2cigroup.com/es/conceptos-de-actualidad-lora-y-lorawan/>)

Cada tiempo de símbolo la frecuencia da un salto a un valor fijo y comienza a crecer desde ahí, el valor al que salta representa el símbolo del mensaje enviado. La cantidad de bits que se puede codificar por símbolo vienen dado por un parámetro ajustable llamado factor de propagación SF (spread factor), por lo cual, si este tiene un valor de N, el símbolo representa N bits y puede tener 2^N posibles valores de frecuencia a los que puede dar el salto. (Edgar Solera, 2018)

2.6.1.3. Factor de Propagación (SF)

El factor de propagación controla la tasa de chirp y, por lo tanto, controla la velocidad de transmisión de datos. El SF varía entre 7 y 12; los factores de propagación más bajos (SF7) significan que se envían más chirridos por segundo, por lo tanto, puede codificar más datos por segundo, esto implica una tasa de transmisión de datos más alta. A diferencia, un factor de propagación más alto (SF12) implica menos chirridos por segundo; por lo tanto, hay menos datos para codificar por segundo.

Los factores de propagación altos significan una mayor ganancia de procesamiento, es decir, una señal modulada con un SF grande se puede recibir con menos errores y recorrer una distancia más larga, en comparación a una señal con un factor de propagación más bajo.

Por otro lado, en comparación con un SF más bajo, enviar la misma cantidad de cantidad de datos (carga útil), con un ancho de banda fijo y con un factor de propagación más alto necesita más tiempo de transmisión; conocido como tiempo en el aire.

En la Tabla 2 se da a conocer como los factores de propagación afectan la sensibilidad del receptor con un ancho de banda fijo de 125 kHz.

Tabla 2

Relación entre el SF y longitud de chirridos

Factor de Propagación (SF)	Sensibilidad del receptor Ancho de banda 125 kHz
7	-123
8	-126
9	-129
10	-132
11	-134
12	-137

Fuente: Autoría

Finalmente, la duración de la batería también depende del factor de propagación, los SF más bajos da como resultado un periodo más corto de activación de los transceptores y una vida útil más larga de la batería.(The Thing Network, 2020)

2.6.1.4. Tiempo en el aire (ToA)

LoRa es un método de modulación que puede transmitir datos a largas distancias, pero con un tiempo de símbolo muy largo, esto tiene como resultado un tiempo muy largo en el aire. El tiempo en el aire es la duración de un paquete en llegar desde un transmisor al receptor, se puede calcular a través la Ec. 1 (Semtech, 2020).

$$T_{oA} = T_{preambulo} + T_{payload} \quad \text{Ec. 1}$$

Es decir, simplemente la suma entre el tiempo del preámbulo ($T_{preambulo}$) y la duración de la carga útil ($T_{payload}$).

2.6.1.5. Tasa de corrección de errores CR (Coding Rate)

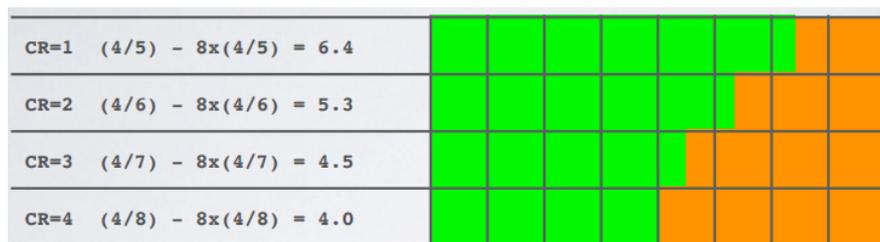
Como LoRa es una tecnología orientada a conexión, es necesario un algoritmo de corrección de errores, en este caso se utiliza la corrección de errores hacia delante FEC, que se puede configurar a través de la tasa de codificación (CR). FEC es proceso en el que se agregan bits de corrección de errores a los datos transmitidos.

La tasa de codificación se refiere a la proporción de los bits transmitidos que realmente transportan información. La tasa de codificación permitidos por LoRa es 4/5, 4/6, 4/7 y 4/8.

En el siguiente ejemplo se tiene un factor de propagación de SF=8, es decir ocho bits transmitidos. Por tanto, si la tasa de codificación es 4/5 entonces 6,4 bits transportan información real y el resto es para corrección de errores.

Figura 9

Tasas de codificación con SF 8



Nota. El color verde representa la información real transmitida y el color naranja representa los bits para la corrección de errores. Adaptado de *Forward Error Correction y Coding Rate*, por Mobilefish, 2018. (https://www.mobilefish.com/download/lora/lora_part14.pdf)

2.6.1.6. Ancho de banda BW (Bandwidth)

El ancho de banda es otro factor muy importante, ya que este parámetro permite aumentar o disminuir la tasa de datos efectiva. LoRa proporciona tres ajustes de BW escalables de 125, 250 y 500 kHz de acuerdo con la banda de frecuencia utilizada. Un

incremento del BW permite el uso de una mayor velocidad efectiva de datos, reduciendo así el tiempo de transmisión, pero ocasiona una reducción de la sensibilidad.

2.6.2. LoRa WAN

(Ertürk et al., 2019) menciona que LoRa hace referencia a Long Range, es una modulación de radio frecuencia que corresponde la capa física en el modelo OSI, por otro lado, LoRaWAN es un protocolo que habilita la capa MAC y su objetivo es resolver problemas de gestión del medio y congestión en la red. En la Figura 10 se muestra el modelo de referencia de LoRaWAN.

Figura 10

Modelo de Referencia LoRaWAN



Adaptado de *LoRa and LoRaWAN: A Technical Overview* por Semtech Corporation, 2018.

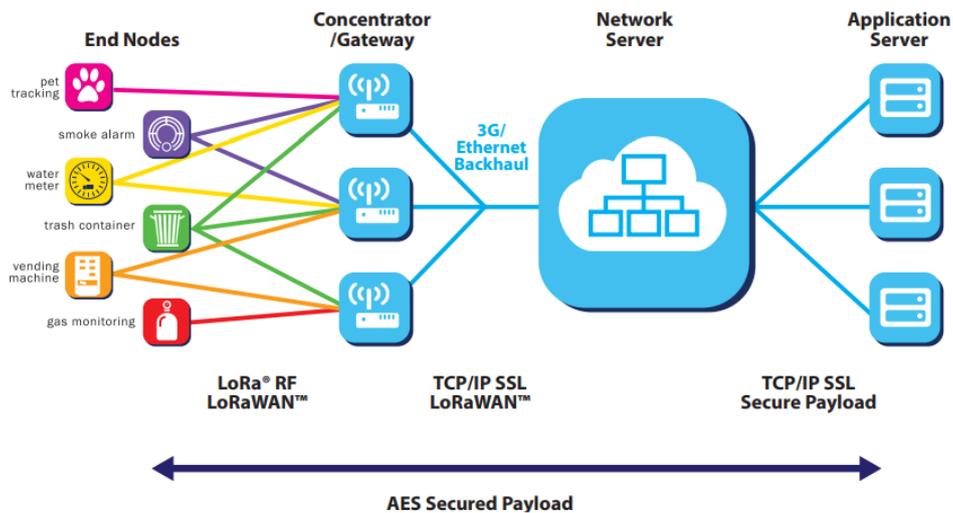
2.6.2.1. Arquitectura de Red LoRaWAN

La arquitectura de red LoRaWAN se implementa en una topología de estrella de estrellas en la que las puertas de enlace (Gateway) transmiten mensajes entre los dispositivos finales y un servidor de red. Las puertas de enlace están conectadas al

servidor de red a través de conexión IP y actúan como un puente, convirtiendo paquetes de RF en paquetes IP y viceversa. La comunicación inalámbrica aprovecha las características de largo alcance de la capa física LoRa, que permite un enlace de un solo salto entre dispositivos finales y puertas de enlace. (LoRa Alliance, 2022)

Figura 11

Arquitectura de red LoRaWAN



Adaptado de *A technical overview of LoRa® and LoRaWAN™*, (p.8), por LoRa Alliance, 2015

2.6.2.2. Clases LoRaWAN

En las redes LoRa puede coexistir diferentes tipos de dispositivos finales, las cuales se puede usar de diferentes formas para sus diferentes propósitos y requerimientos; por esto el protocolo de capa MAC LoRaWAN ofrece estrategias de acceso al canal agrupadas en tres clases. La Clase A es obligatoria y la Clase B y C son opcionales.

Los dispositivos de Clase A permiten comunicaciones bidireccionales, tienen el menor consumo de energía, envían pequeñas cantidades de datos al Gateway y la mayor cantidad de tiempo permanecen inactivos. La comunicación de enlace ascendente (Up Link) al Gateway va seguida de dos ventanas de recepción de enlace descendente (Down Link) con dos retardos distintos (Miles et al., 2020).

Los dispositivos de Clase B controlan las ventanas de recepción con intervalos de tiempo predeterminados. Los dispositivos finales reciben un Beacon sincronizado con el tiempo del Gateway y abren las ventanas de enlace descendente solo en los horarios programados.

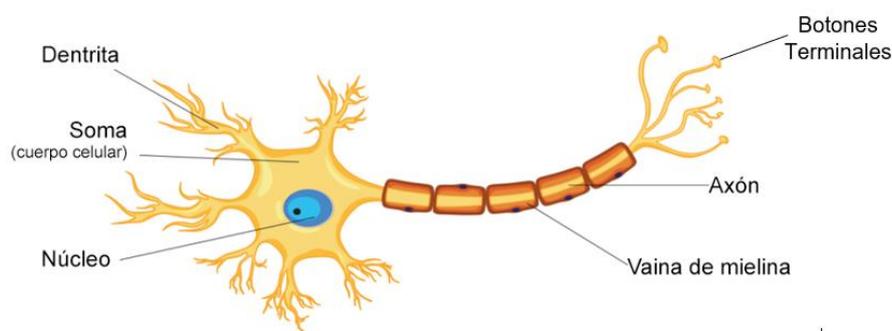
A diferencia de los dispositivos de Clase A y B, la Clase C consumen más energía debido a que tiene una ventana de recepción que siempre esta activa, excepto durante la transmisión de datos.

2.7. Redes Neuronales Artificiales

Una red neuronal artificial (RNA) es un modelo computacional basado en la estructura y los aspectos funcionales de las redes neuronales biológicas del cerebro humano. En la mayoría de los casos es un sistema adaptativo que cambia su estructura en base a información externa o interna que fluye a través de la red durante la fase de aprendizaje (Hormozi et al., 2012). Para comprender de mejor manera como funciona una red neuronal en la Figura 12 se puede observar la neurona biológica; donde se realiza todas las conexiones y que da lugar a la toma de decisiones.

Figura 12

Estructura de una neurona biológica



Adaptado de *Diagrama de la Anatomía de una Neurona* por freepik, 2020, (https://www.freepik.es/vector-premium/diagrama-anatomia-neurona_2480497.htm)

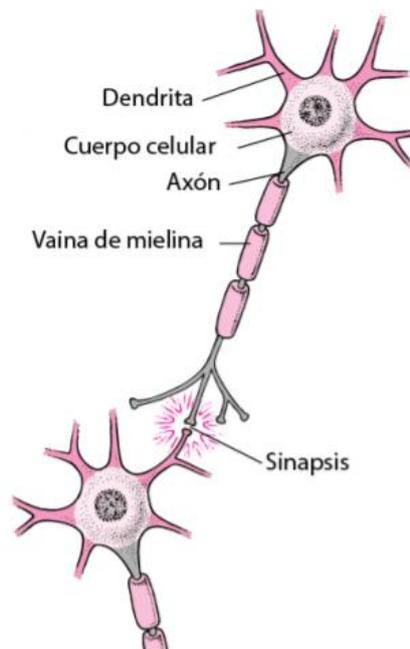
Las dendritas son ramificaciones donde la neurona se conectan con algunas otras neuronas o va a poder recibir información para poder procesarla. Posteriormente se tiene el cuerpo y el núcleo donde la neurona acumula toda la información y de acuerdo con el proceso de toda esa información decide si se activa o no, es decir manda una señal o genera un potencial hacia otras neuronas como salida.

El axón le da soporte a la neurona y es por donde va a comenzar a viajar toda la información que proporciona la neurona. La mielina y los nodos de Ranvier son los encargados de que la información llegue a su destino. Finalmente, para entregar toda esa información están los botones terminales, que son la salida de la neurona.

Por medio de las conexiones de varias neuronas se logra el procesamiento de la información. La Figura 13 representa el proceso de la sinapsis de una neurona biológica.

Figura 13

Sinapsis de una neurona biológica



Adaptado de *Estructura típica de una neurona*, por Manual MSD, 2020,
(https://www.msmanuals.com/es-cl/hogar/multimedia/figure/neu_nerve_cell_es)

Las neuronas se comunican entre sí en las uniones llamadas sinapsis que es el espacio entre dos neuronas vecinas, las entradas de una neurona corresponden a la salida

de las señales de las otras neuronas. De este proceso se toma referencia para el modelo de la neurona artificial

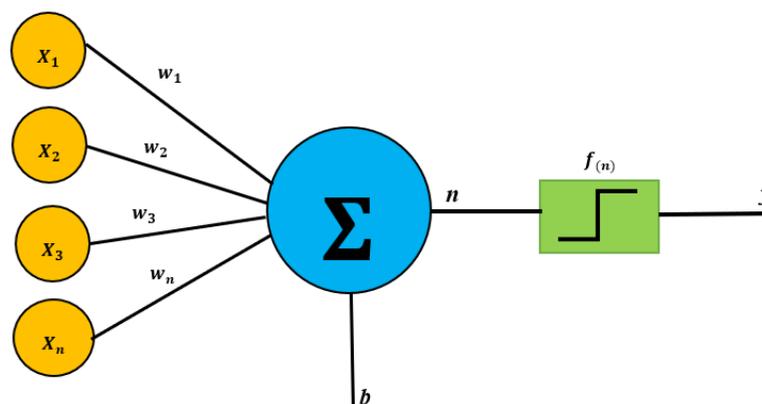
2.7.1. Modelo de una Red Neuronal Artificial

Las redes neuronales artificiales generalmente están inspiradas en las redes neuronales de tipo biológico, aunque poseen otras funcionalidades y estructuras de conexión. El modelo de una red neuronal artificial se da a conocer en la Figura 14, una neurona con n entradas consiste en:

- Un conjunto de entradas $x_1 \dots \dots x_n$
- Pesos sinápticos $w_1 \dots \dots w_n$, correspondientes a cada entrada
- Una función de agregación Σ
- Una función de activación $f_{(n)}$
- Una salida y

Figura 14

Modelo de una Red Neuronal Artificial



Adaptado de *Modelo Neuronal* por D. Torres, 2020, Codideep (<https://codideep.com/blogpost/redes-neuronales-artificiales-y-sus-aplicaciones>)

En este modelo, los estímulos externos que llegan a la neurona corresponden a diferentes valores de variables de entrada x_n , estos valores son multiplicados por los pesos sinápticos w_n que denotan la fuerza de conexión sináptica entre las entradas y las dendritas de las neuronas. Seguido (Berzal, 2018), las señales ponderadas llegan al

cuerpo celular del modelo donde son acumulados a través de una suma, más una entrada especial conocida como bias o sesgo (b) que controla que tan propenso esta la neurona a disparar un 1 o un 0, independiente de los pesos. En la Ec. 2 se da a conocer la función de entrada de una red neuronal.

$$\mathbf{n} = x_1w_1 + x_2w_2 + x_3w_3 + \dots + x_nw_n + b \quad \text{Ec. 2}$$

Una vez obtenido este valor llega a una función de activación que define el estado inhibido o excitado de la neurona. A partir de ahí se da la transmisión del resultado o salida a través del axón de la neurona (y) hacia otras. (Berzal, 2018)

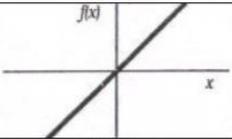
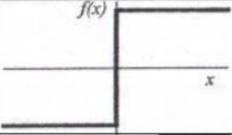
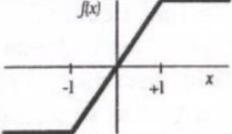
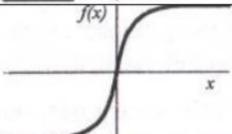
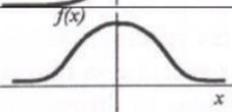
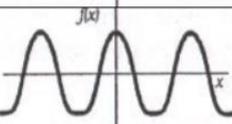
$$y = f(n)$$

$$y = f(x_1w_1 + x_2w_2 + x_3w_3 + \dots + x_nw_n + b) \quad \text{Ec. 3}$$

Una función de activación determina el estado de activación actual de la neurona en función del estado anterior, sin embargo, la mayoría de los modelos ignoran el estado anterior de la neurona, definiendo el estado de activación como el potencial resultante cuyo rango va de (0 a 1) o de (-1 a 1) ya que una neurona puede estar totalmente activa (1) o inactiva (0 o -1). Las funciones de activación más utilizadas según (Acevedo et al., 2017) son las que se muestra en la Figura 15.

Figura 15

Tipos de funciones de activación

	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, \infty]$	
Escalón	$y = \begin{cases} 1, & \text{si } x \geq 0 \\ 0, & \text{si } x < 0 \end{cases}$ $y = \begin{cases} 1, & \text{si } x \geq 0 \\ -1, & \text{si } x < 0 \end{cases}$	$[0, 1]$ $[-1, 1]$	
Lineal a tramos	$y = \begin{cases} 1, & \text{si } x > 1 \\ x, & \text{si } -1 \leq x \leq 1 \\ -1, & \text{si } x < -1 \end{cases}$	$[-1, 1]$	
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$ $y = \tanh(x)$	$[0, 1]$ $[-1, 1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, 1]$	
Sinusoidal	$y = A \sin(wx + \varphi)$	$[-1, 1]$	

Adaptado de *Principios y características de las redes neuronales artificiales* (p.177), por E. Serna, A, Serna, E, Acevedo, 2017

2.7.2. Estructura de una Red Neuronal Artificial

La distribución de las neuronas dentro de la red se realiza formando niveles o capas. Una capa es un conjunto de neuronas, de manera que las neuronas de cada capa esta conectadas con las neuronas de la siguiente capa, a partir de la situación dentro de la red, se puede distinguir tres tipos de capas: entrada, oculta y salida.

La capa de entrada recibe datos o señales del exterior, los datos que la red neuronal artificial procesará.

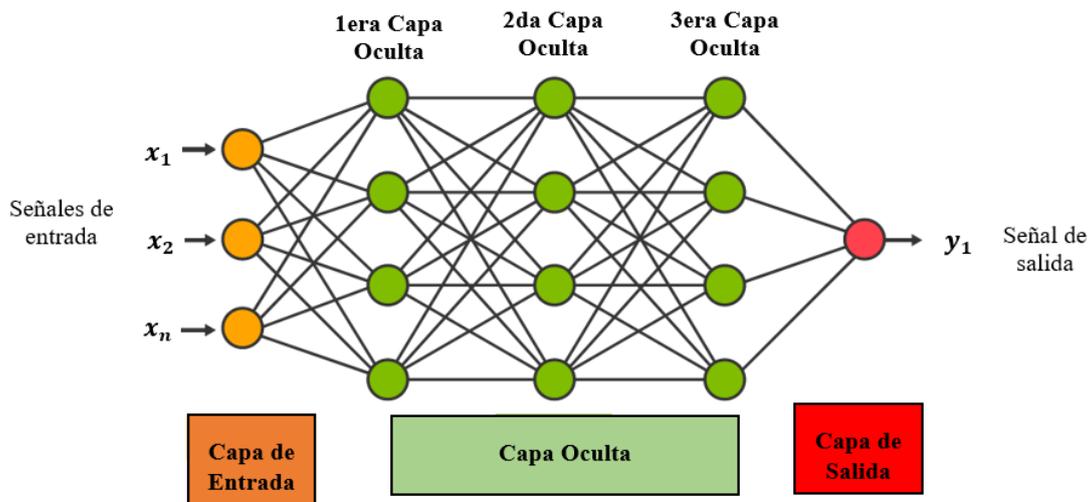
La capa oculta no tiene contacto con el exterior, es un nivel interno de la red y se encarga de realizar el procesamiento de la información de la red.

La capa de salida recibe la información procesada y la transfiere hacia el exterior, según su aprendizaje puede ser una respuesta correcta o incorrecta.

Por lo tanto, una red neuronal siempre está compuesta por una capa de entrada, una capa de salida y cero o más capas ocultas.

Figura 16

Estructura en capas de la Red Neuronal



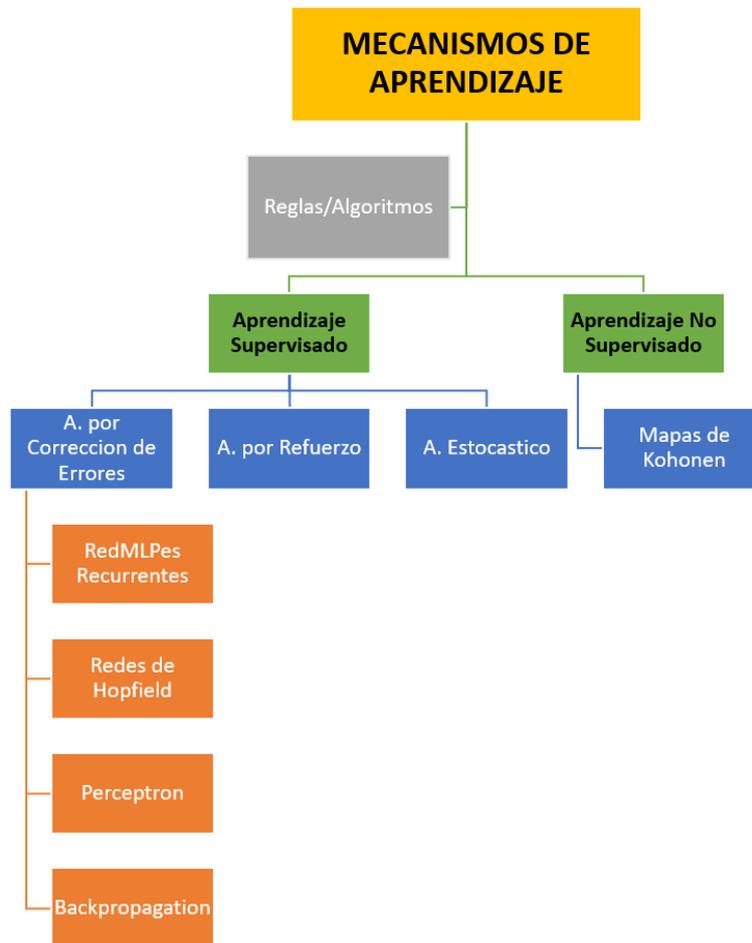
Adaptado de *Estructura en capas de las redes neuronales*, por TIBCO, 2022
(<https://www.tibco.com/es/reference-center/what-is-a-neural-network>)

2.7.3. Mecanismos de Aprendizaje de Redes Neuronales

El aprendizaje se puede definir como el proceso mediante el cual una red neuronal ajusta los pesos sinápticos en respuesta a la información de entrada. Existen varias reglas o algoritmo para aprendizaje supervisado y para aprendizaje no supervisado. En la Figura 17 se resume a través de un mapa conceptual los mecanismos de aprendizaje.

Figura 17

Mecanismos de Aprendizaje de la Red Neuronal



Fuente: *Autoría*

Aprendizaje supervisado consiste en entrenar un algoritmo otorgándole atributos (entradas), y etiquetas (salidas), con el fin de que el algoritmo las combine y pueda realizar predicciones. Es decir, dada alguna entrada, tiene como objetivo predecir o devolver una función que mejor se aproxime algunas de sus características. (Bonetto, 2021)

Entre los algoritmos de aprendizaje supervisado se tiene: aprendizaje por corrección de errores, aprendizaje por refuerzo, aprendizaje estocástico.

El aprendizaje por corrección de errores consiste en exponer a la red un conjunto de pares de datos que representa la entrada y la salida deseada para dicha entrada. Los

pesos de las conexiones de la red se ajustan en función de la diferencia entre los valores deseados y los valores obtenidos en la salida de la red. Es decir, en función de errores cometidos en la salida.

Entre las reglas del aprendizaje por corrección de errores esta Backpropagation o propagación hacia atrás, aplicada a modelos de redes que tienen más de dos capas. Resulta de la forma en el que el error es propagado hacia atrás desde la capa de salida de la red neuronal.

Se distinguen dos fases, en la primera fase se aplica un patrón de entrada como estímulo para la primera capa, este a su vez se propagan por las distintas capas superiores que componen la red hasta producir la salida. El resultado obtenido es comparado con las salidas que se desea tener y se calcula un valor de error para cada neurona de salida. Estos errores se transmiten hacia atrás, partiendo de la capa de salida, hacia todas las neuronas intermedias. Cada neurona recibe un error que es proporcional a su contribución sobre el error total de la red. Basándose en el error recibido se ajustan los errores de los pesos sinápticos de cada neurona. (Monte, 2019)

Como se había mencionado anteriormente, los errores se corrigen durante la retro propagación de la red, pero el valor de los pesos se actualiza mediante el descenso del gradiente; un algoritmo de optimización de una red neuronal artificial

2.7.4. Algoritmos de Optimización

El descenso del gradiente es iterativo, es decir que se repite el mismo procedimiento varias veces teniendo en cuenta dos parámetros:

El número de iteraciones: cuantas veces se repetirá el procedimiento

Tasa de aprendizaje: define la cantidad de iteraciones requerida para que el algoritmo encuentre el mínimo de la función (α)

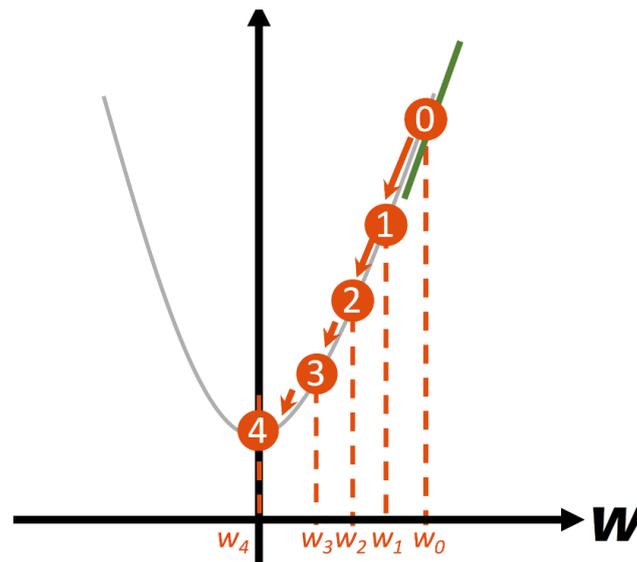
Lo que hace este algoritmo es seleccionar un valor inicial aleatorio, se deriva la función para encontrar la pendiente. Una vez encontrada la pendiente el algoritmo sabe hacia dónde dirigirse para intentar llegar a un mínimo global de la función de error. En la Ec. 4 se puede observar la fórmula del descenso del gradiente.

$$\omega = w - \alpha * \text{gradiente} \quad \text{Ec. 4}$$

En cada iteración se actualiza el valor de ω usando la Ec. 4, se repite este paso hasta completar el número de iteraciones requeridas por el usuario. En la Figura 18 se puede observar un ejemplo del descenso del gradiente

Figura 18

Ejemplo del descenso del gradiente



Adaptado de *Gradiente Descendiente*, por M. Sotaquira, 2018 (<https://www.codificandobits.com/blog/el-gradiente-descendiente/>)

Los algoritmos de optimización basados en gradiente comúnmente utilizados son SGD, Adam y RMSProp. El algoritmo de optimización Adam (Estimación adaptativa de momentos), es una extensión del descenso de gradiente estocástico que actualiza los pesos de la red de forma iterativa en función de los datos de entrenamiento. Además, combina funciones RMSProp y Descenso del gradiente estocástico con Momentum, calculando una combinación lineal entre el incremento actual y el incremento anterior, y

considera los gradientes recientemente aparecidos en las actualizaciones para mantener diferentes tasas de aprendizaje por variable.

2.7.5. Validación de una Red Neuronal

Una vez que se concluye con el entrenamiento de la red neuronal los pesos sinápticos de las conexiones en la red quedan fijos, luego de esto se debe comprobar si la red neuronal puede resolver nuevos problemas de forma general para el cual ha sido entrenada, para esto se requiere de otro conjunto de datos denominados *conjunto de validación o testeo* usados con el propósito de validar la red neuronal y prevenir el sobre entrenamiento.

2.7.6. Aplicaciones de las Redes Neuronales

Entre las áreas de aplicación de las redes neuronales se encuentran entre varias, las redes neuronales para la toma de decisiones en el sector agrícola y los trabajos que se llevan a cabo en esta área. Las redes neuronales han demostrado ser una herramienta eficaz para caracterizar, modelar y predecir una gran cantidad de procesos no lineales con resultados adecuados en la toma de decisiones requeridas en problemas agrícolas como priorizar y clasificar productos, el reconocimiento de patrones, la predicción de cosechas y riego, etc.

En esta investigación se aplica las redes neuronales en el sector agrícola para realizar predicciones de riego en un cultivo de pasto.

2.8. Importancia de los Pastos y Forrajes

La importancia de los pastos y forrajes es reconocida desde el momento en el que el hombre domesticó a los animales; el pasto es la base principal de alimentación de los animales herbívoros con fines de explotación, ya sea de leche o carne. La alimentación de los herbívoros debe basarse en pastos y forrajes porque son la fuente de alimento más

barata y completa (fuente de energía y proteína) principalmente al asociar gramíneas con leguminosas proveen un alimento completo y balanceado al ganado. (Leon et al., 2018). Una de las principales características de los pastos cultivados es su gran capacidad de rebrote y soportar el pisoteo de los animales.

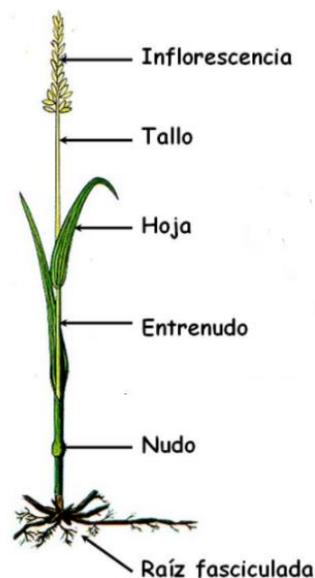
En una pastura se encuentra tres tipos de plantas: gramíneas (poáceas), leguminosas y adventicias.

La mayoría de las especies forrajeras son gramíneas, también conocidas como pastos o gramas, son ricas en hidratos de carbono, soporta mejor el pastoreo, controla la erosión y viven más que las leguminosas. Crecen en todo tipo de ambientes, con diferentes exigencias en cuanto a agua, fertilidad de suelo y temperaturas.

Las gramíneas presentan características morfológicas como raíces poco profundas, tallos cilíndricos con nudos, sus hojas envuelven el tallo y terminan en punta y sus flores en la mayoría de los casos son espiguillas. (ROCALBA, 2015). En la Figura 19 se puede observar una gramínea con sus respectivas partes.

Figura 19

Partes de una Gramínea



Adaptado de *Familia Gramíneas (Gramineae) o Poáceas*, por Universidad Politécnica de Valencia, 2018, (<http://www.euita.upv.es/varios/biologia/temas%20angiospermas/comenilidas/gramineas.htm>)

Entre las especies más importantes que se cultivan en la zona andina se encuentran el Ryegrass y el trébol. El Ryegrass es una gramínea forrajera que se establece rápidamente y tiene buena producción inicial, pero las plantas se agotan rápidamente, desapareciendo en un periodo corto de tiempo. Además, son muy exigentes en humedad. El trébol es una planta perenne y resistente, crecen bien en gran diversidad de suelo cuando la humedad es adecuada. El trébol aumenta el valor alimenticio de una pradera de Ryegrass porque produce altos niveles de proteína digestible y contenido mineral.

2.8.1. Factores que Influyen en la Producción y Calidad de Pastos y Forrajes

Según (Leon et al., 2018), los componentes principales del clima que influyen en la producción de pastos y forraje son: temperatura, lluvia y radiación solar, estos elementos contribuyen en la vida y sus procesos en el suelo y sobre el suelo, afectando indirectamente la productividad final de la planta.

2.8.1.1. Temperatura

La temperatura controla las reacciones bioquímicas de la planta, crecimiento y metabolismo, la reacción más importante que es influenciada por la temperatura es la fotosíntesis. Los pastos de clima templado tienen óptimo crecimiento a temperaturas entre 10° y 20 °C, cuando supera el valor óptimo de temperatura; los pastos utilizan mecanismos estructurales para reducir los efectos de estrés por altas temperaturas, influyendo así en un aumento del contenido de la pared celular específicamente de la lignina, la cual reduce la digestibilidad y la calidad de los pastos. (Izquierdo & Mantilla, 2015).

2.8.1.2. Humedad del suelo

La humedad del suelo es la cantidad de agua contenida por unidad de volumen de tierra que hay en un terreno. El exceso de humedad en el suelo en épocas de lluvia produce efectos negativos en el crecimiento del pasto, causa anoxia (falta de oxígeno) en las raíces, afectando la respiración aeróbica de la planta además de la absorción de minerales y agua, causando cambios morfológicos, tales como la reducción en el crecimiento de los tallos y el aumento de la proporción de las hojas, lo que es característico en el retraso de la madurez de la planta. Sin embargo, la escasez de agua también es perjudicial para los cultivos, el déficit disminuye la concentración de la pared celular en las hojas y tallos de los forrajes. Por lo tanto, es muy importante monitorear esta variable, controlar regularmente el nivel de humedad del suelo nos permitirá determinar el momento apropiado para administrar el riego. (Izquierdo & Mantilla, 2015)

2.8.1.3. Radiación solar

La radiación solar es el flujo de energía que se recibe del sol en forma de ondas electromagnéticas de diferentes frecuencias (luz visible, infrarroja y ultravioleta). Las plantas realizan mayor fotosíntesis cuando están expuestas a todo el espectro visible que a una parte de este. Las plantas requieren de energía solar para utilizar el carbono atmosférico mediante la fotosíntesis. (Leon et al., 2018)

2.8.2. Importancia del Riego en Cultivos de Pasto.

El riego es uno de los aspectos más importantes que las personas vinculadas a la ganadería deben aplicar de forma adecuada para lograr altos rendimientos del pasto. El crecimiento y producción de forrajes dependerá de la cantidad de agua de lluvia, de las posibilidades de riego y de los nutrientes del suelo.

Durante el periodo de germinación de la semilla y el desarrollo del pasto, el suelo debe permanecer húmedo en superficie, si las condiciones de humedad y temperatura son favorables, la semilla absorbe el agua requerida para su germinación. En su etapa de crecimiento, por su fase de desarrollo, el pasto no tolera el estrés hídrico, por lo que debe haber humedad suficiente. Si el suelo se seca, se detienen el desarrollo de la planta y puede ocurrir mortandad.

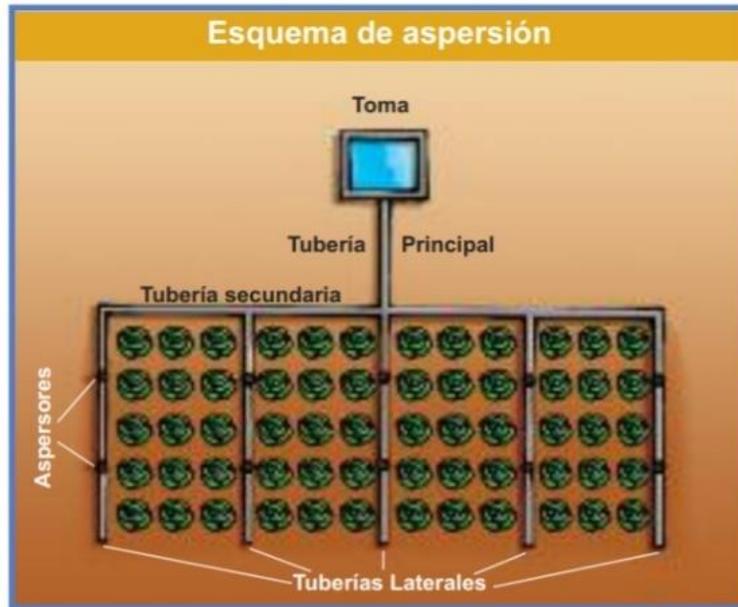
Los riegos de pre o post siembran de pasturas, siempre deben ser por aspersión, nunca gravitacional para evitar que la semilla salga de la profundidad de la tierra, se encostre el suelo y la distribución de humedad sea desigual (Leon et al., 2018).

Los principales métodos de riego de pasturas son: inundación y aspersión. En el riego por aspersión se aplica agua en forma de lluvia, cuando las gotas llegan al suelo, infiltran en ese punto y no se escurren sobre el terreno. Es distribuida a través de aspersores, los cuales producen gotas de agua de diferentes tamaños, imitando una precipitación natural.

Su inversión inicial en materiales y equipos es alta y tiene una eficiencia del 80-85 % (FAO, 2016), ya que aprovecha con mayor eficacia los pequeños caudales de agua, debido a que un sistema de aspersores bien diseñados distribuye mejor el agua que otros métodos. En la Figura 20 se muestra un esquema de riego por aspersión.

Figura 20

Esquema de riego por aspersión



Adaptado de *Manual de Riego Parcelario* (p.6), por MAGAP, 2014

La presión en este método de riego se obtiene por medio de energía, normalmente a partir de bombas hidráulicas que aspiran el agua desde canales, ríos, pozos o embalses. Sin embargo, también se puede operar sin bombas cuando la fuente de agua se encuentra en posiciones más elevadas que el cultivo (Herrera et al., 2013)

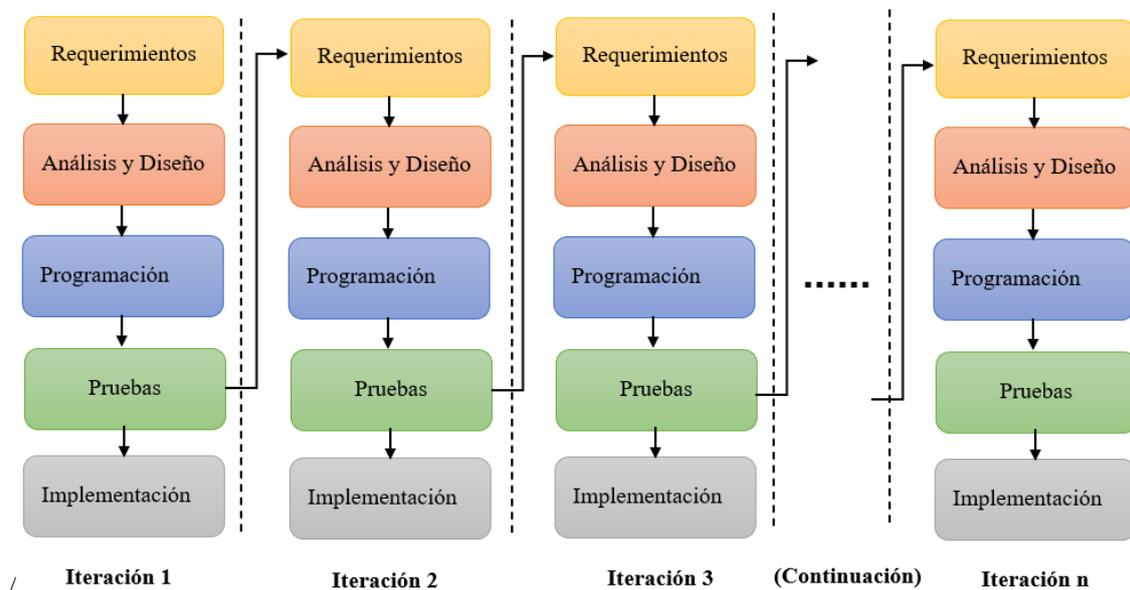
CAPÍTULO III. Desarrollo de la Propuesta

En este capítulo se aplica la metodología del modelo iterativo para el diseño y desarrollo del proyecto. Para seleccionar la metodología adecuada se debe tomar en cuenta los requerimientos y propósitos del mismo, este debe permitir garantizar los resultados finales y su proceso debe ser sistemático para detectar y disminuir riesgos.

El modelo iterativo se basa en dividir el proyecto de desarrollo en varias etapas denominadas iteraciones, la idea central es que, en cada iteración se construya una pequeña parte del sistema. Para esa parte, se realiza todo el proceso de requerimientos, análisis y diseño, programación y pruebas. Se culmina la iteración con un ejecutable que incluye todas las partes del sistema construidas hasta el momento (Palasi Llallana, 2004). En la Figura 21 se puede apreciar las etapas del modelo Iterativo.

Figura 21

Modelo de desarrollo Iterativo



Adaptado de *Modelos de desarrollo Iterativos*, por V, Palasi, 2004

Este modelo no necesita la lista completa de requerimientos antes de iniciar el proyecto, tiene flexibilidad a los cambios en los requisitos. Además, cuanto más corta es la iteración, más fáciles son las etapas de prueba y depuración (Ossetskyi, 2017).

Este apartado consta de dos iteraciones. En la primera iteración se realiza el diseño de la red de sensores inalámbricos WSN LoRa, y en la segunda iteración el entrenamiento de la red neuronal para la predicción del riego.

3.1. Modelo Iterativo: Primera Iteración

Se inicia con la primera iteración para el desarrollo del sistema, en esta iteración se realiza el diseño de la red WSN con módulos LoRa. En la primera etapa del modelo iterativo están los requerimientos, estos no quedan totalmente fijados hasta el final del proyecto, por lo que se puede realizar cambios de forma flexible en cada iteración.

3.1.1. *Requerimientos del Sistema*

Para que el sistema lleve a cabo una función específica y cumpla con las expectativas del usuario final, es necesario establecer requerimientos; evaluando diferentes condiciones del entorno para que el sistema se desarrolle de forma adecuada y cumpla su objetivo. Para ello se considera los siguientes requerimientos:

- Requerimientos de Stakeholders
- Requerimientos del sistema
- Requerimientos de arquitectura

Para una mejor comprensión y manejo de los datos se ha establecido abreviaturas a cada uno de los requerimientos mencionados anteriormente, tal como se puede apreciar en la Tabla 3.

Tabla 3.*Nomenclatura de los requerimientos del Diseño*

Abreviatura	Descripción
RSHS	Requerimientos de Stakeholders
RFS	Requerimientos del Sistema
RAS	Requerimientos de Arquitectura

Fuente: *Autoría***3.1.1.1.Requerimientos de Stakeholders**

Los Stakeholders desempeñan un papel importante en las especificación y validación de los requerimientos. El desarrollo de un sistema requiere de una serie de actividades y entre las más importantes se encuentra comprender las necesidades y cubrir las expectativas de las personas que intervienen de forma directa o indirecta en el desarrollo del sistema. En la Tabla 4 se presenta la lista de Stakeholders que intervienen en el proyecto.

Tabla 4.*Lista de Stakeholders*

Lista de Stakeholders	
Usuario directo	Propietario del terreno
Desarrollador del Proyecto	Sra. Bonilla Evelyn
Director y fiscalizadores del proyecto	Msc. Edgar Maya Msc. Luis Suarez

Fuente: *Autoría*

La Tabla 5 contiene los requerimientos operacionales y de usuario, este expresa la iteración que tendrá el sistema con el entorno y los involucrados en el proyecto. Los involucrados deberán evaluar el sistema después de su implementación.

Tabla 5*Requerimientos de Stakeholders*

RSHS				
#	Requerimientos de Stakeholders	Prioridad		
		Alta	Media	Baja
REQUERIMIENTOS OPERACIONALES				
RSHS1	El sistema debe implementarse en el cultivo de pasto	X		
RSHS2	El sistema debe tener bajo consumo de energía	X		
RSHS3	Conectividad a Internet		X	
RSHS4	Visualización de datos		X	
RSHS5	Automatización del Sistema	X		
RSHS6	Operación permanente	X		
RSHS7	Almacenamiento de datos		X	
REQUERIMIENTO DEL USUARIO				
RSHS8	Conocimientos en sistemas de riego		X	
RSHS9	Facilidad de uso para el usuario	X		
RSHS10	Los nodos deben ser tolerantes a agentes externos como agua, calor y humedad.	X		
RSHS11	Presentación de resultados claros y precisos	X		

Fuente: *Autoría***3.1.1.2. Requerimientos Funcionales del Sistema**

Los requerimientos funcionales son actividades o servicios que proveerá el sistema, es decir, las funciones, comportamiento y forma de uso del sistema. Esto es procedente de la observación y necesidades del usuario final. En la Tabla 6 se detallan los requerimientos funcionales del sistema.

Tabla 6*Requerimientos funcionales del sistema*

RFS				
#	Requerimientos del Sistema	Prioridad		
		Alta	Media	Baja
RFS - Uso				
RFS1	El sistema debe estar energizado	X		
RFS2	Los datos recolectados por los nodos deben ser tipo entero	X		
RFS3	Simplicidad de manejo del sistema	X		
RFS - Desempeño				
RFS4	Conexión a una red inalámbrica	X		
RFS5	Envío de estados característicos del ambiente	X		
RFS6	Bajo consumo de energía	X		
RFS7	Bajos retardos de tiempo en la transmisión de datos	X		
RFS8	Baja tasa de error en los datos adquiridos por los sensores	X		
RFS9	Iniciar/Detener proceso de riego	X		
RFS10	Predicción de riego con el algoritmo de aprendizaje automático	X		
RFS - Interfaces				
RFS11	Pines de entrada y salida del sistema embebido	X		
RFS12	El sistema embebido, las placas de comunicación inalámbrica, la válvula solenoide y la fuente de alimentación eléctrica deben interactuar entre sí.	X		
RFS – Modos/Estados				
RFS13	El sistema debe estar en funcionamiento permanente	X		
RFS14	Estado de funcionamiento y carga con batería independiente		X	
RFS15	El riego se activa de acuerdo con la predicción determinado por el algoritmo de aprendizaje automático.	X		
RFS - Físicos				
RFS16	Los nodos de recolección deben tener tapas herméticas	X		
RFS17	Tamaño reducido			
RFS18	Nodos ubicados a una altura considerable del suelo para evitar interferencias	X		
RFS19	Sensor de humedad ubicado a una profundidad considerable con respecto a la raíz del cultivo de pasto.	X		

Fuente: *Autoría*

3.1.1.3. Requerimientos de Arquitectura

Los requerimientos de arquitectura describen los componentes que forman parte del sistema tanto software, hardware y el subsistema eléctrico.

Tabla 7

Requerimientos de arquitectura

RAS				
#	Requerimientos de Arquitectura	Prioridad		
		Alta	Media	Baja
RAS- Lógicos				
RAS1	Entradas y salidas Analógico/Digital	X		
RAS2	Comunicación entre los nodos y la puerta de enlace	X		
RAS3	Calibración de cada uno de los sensores	X		
RAS4	Los nodos deben enviar los datos recolectados a una distancia de 160m	X		
RAS – Diseño				
RAS5	Circuito protegido por una capa hermética	X		
RAS6	Hardware y software de distribución libre	X		
RAS7	Ubicación de sensores en lugares estratégicos cubriendo toda el área del cultivo.	X		
RAS8	La puerta de enlace debe estar dentro de la zona de cobertura de los nodos.	X		
RAS9	Los dispositivos de transmisión y recepción deben ser compatibles con medios de transmisión inalámbrica LoRa.	X		
RAS10	Implementación de un sistema de bajo costo	X		

RAS - Software		
RAS11	Lenguaje de programación de código abierto	X
RAS12	Interfaz amigable con el usuario	X
RAS13	Compatibilidad del software con librerías para el algoritmo de aprendizaje automático.	X
RAS14	Compatibilidad entre el sistema embebido y las librerías de los sensores.	X
RAS15	Compatibilidad con plataformas de programación (Python)	X
RAS16	Control de motores eléctricos	X
RAS - Hardware		
RAS17	Comunicación inalámbrica estable	X
RAS18	Entradas y salidas analógico/digital suficientes para la conexión de sensores y válvula solenoide	X
RAS19	La fuente de energía debe alimentar a la placa programable, sensores y módulo LoRa.	X
RAS20	Sistema embebido con puertos de transmisión y recepción de datos para enviar información entre módulos inalámbricos.	X
RAS21	Sistema embebido con un procesador que soporte tratamiento de información en tiempo real.	X
RAS - Eléctrico		
RAS22	Batería de Litio para la alimentación del sistema	X
RAS23	Panel Solar	X

Fuente: *Autoría*

3.1.1.4. Selección del Hardware y Software del Sistema

La elección del hardware y software se realiza en base a los requerimientos del desarrollador y la arquitectura del sistema.

Se inicia por la elección de la placa para el nodo sensor, encargado del sensado de las variables ambientales. Para escoger la más adecuada se realiza una comparación de diferentes placas Arduino, tal como se muestra en la Tabla 8.

Tabla 8

Selección de hardware del nodo sensor

CARACTERISTICAS	MODELOS		
	Arduino UNO	Arduino MEGA	Arduino NANO
			
Microcontrolador	ATMega328	AT	AT
Voltaje de operación	5V	5V	5V
Voltaje recomendado	7-12 V	7-12 V	7-12 V
Velocidad de reloj	16MHz	16MHz	16MHz
Entrada/Salida Digital	14(6 son PWM)	54(15 son PWM)	14(6 son PWM)
Entrada/Salida Analógico	6 (solo entradas)	16 (solo entradas)	8 (solo entradas)
Memoria Flash	32 kB	256 kB	32kB
Memoria EPROM	1kB	4 kB	1 kB
Compatibilidad con Shield	Si	Si	No
Dimensión	69x54 mm	101x54	18x45
Precio aproximado	\$10	\$17	\$8

Fuente: Autoría

Se escoge el Arduino UNO, una placa de microcontrolador basado en el microchip ATmega328. A diferencia de las demás placas, este es de tamaño reducido, su precio es económico y tiene compatibilidad con placas Shield. Además, tiene la cantidad de pines analógico/digital necesarios.

Como es indispensable establecer la comunicación inalámbrica entre el nodo sensor y el nodo central, se realiza una comparación de transceptores LoRa, tomando en cuenta las características que tienen para emitir comunicación entre nodos y disponibilidad en el mercado, tal como se muestra en la Tabla 9.

Tabla 9

Elección de módulo LoRa para el nodo sensor

CARACTERISTICAS	MODELOS	
	Dragino LoRa Shield	Cytron LoRa Shield
		
Compatible con Arduino	SI	SI
Distancia TX/RX	>2 Km	2 km
Consumo de energía	Bajo	Bajo
Banda de frecuencia	433/868/915 MHz	915 MHz
Antena incorporada	SI	NO
Velocidad de bits programable	300 kbps	300kbps
Precio aproximado	\$ 45	\$55
Sensibilidad de recepción	-148dBm	-146dBm
Tamaño	62x43x23mm	55x47x24 mm

Fuente: Autoría

De acuerdo con la comparativa de los transeceptores LoRa, se puede evidenciar que comparten características similares, pero se selecciona el Dragino LoRa Shield basado en el chip Sx1276, cubre una distancia mayor a 2Km y su precio y tamaño son reducidos. Además, existe disponibilidad en el mercado nacional a diferencia del Cytron LoRa Shield.

En la Tabla 10 se realiza la comparación de sensores de humedad del suelo que pueden llegar a ser parte del nodo sensor. Para su elección se considera la practicidad de uso, costo, la sensibilidad y disponibilidad en el mercado nacional.

Tabla 10

Elección de sensor de humedad del suelo para el nodo sensor

CARACTERISTICAS	MODELOS		
	FC-28	YL69	Sensor de humedad del suelo capacitivo
			
Compatible con Arduino	SI	SI	SI
Voltaje de Operación	3.3V a 5V	3.3V a 5V	3.3 V a 5 V
Corriente de operación	35 mA	32 mA	5mA
Fiabilidad	Alta	Alta	Alta
Vida Útil	3 a 6 meses	3 a 6 meses	3 años min
Precio aproximado	\$3	\$4	\$8
Tamaño	3x1.6 cm	6x3 cm	9.8x2.3

Fuente: Autoría

De acuerdo con la Tabla 10, se selecciona el sensor de humedad del suelo capacitivo ya que cuenta con bajo consumo energético y una alta resistencia a la corrosión (no se oxida fácilmente). Mide la humedad del suelo mediante detección capacitiva, en lugar de detección resistiva como lo hacen los otros sensores, además tienen una vida útil mayor a tres años.

Por otro lado, la temperatura es otro parámetro importante que considerar en el nodo sensor, por lo cual en la Tabla 11 se muestran varias opciones de sensores con sus características técnicas.

Tabla 11

Elección de sensor de temperatura para el nodo sensor

CARACTERISTICAS	MODELOS		
	 DTH-11	 DTH22	 LM35
Compatible con Arduino	SI	SI	SI
Voltaje de operación	3 V-5 V	3 V-6V	4V -30V
Fiabilidad	Alta	Alta	Alta
Rango de Medición	Humedad de 20 a 80 % Temperatura de 0 a 50°C	Humedad de 0 a 100 % Temperatura de -40 a 80°C	-55°C hasta +150°C
Precisión de medición	Temperatura ± 2.0 °C Humedad $\pm 5\%$	Temperatura ± 0.5 °C Humedad $\pm 2\%$ a 5%	-10°C hasta 85°C ± 0.5 °C
Velocidad de muestreo	1 Hz	0.5 Hz	-
Precio aproximado	\$3.50	\$10	\$3.00
Tamaño	1.6x1.2 cm	2.0x1.5 cm	0.46x4.6 mm

Fuente: Autoría

Se selecciona el sensor de temperatura LM35 debido a su reducido tamaño, el rango de medición es mayor en comparación con los otros sensores, teniendo una ventaja con respecto a los sensores calibrados en kelvin, y es que no requiere un voltaje alto a la salida para obtener un escalado en grados centígrados. Adicionalmente, no necesita ser calibrado para proporcionar precisión y su precio es bajo.

En la Tabla 12 se realiza una comparación de sensores de radiación UV, detallando sus características técnicas para escoger la que mejor se adapte a los requerimientos del sistema.

Tabla 12

Elección de sensor de radiación UV para el nodo sensor

CARACTERISTICAS	MODELOS		
	GUVA-S12SD 	GY-ML8511 	UVM-30 
Compatible con Arduino	SI	SI	SI
Voltaje de operación	3.3 V-5 V	5 V	3 V-6V
Región de sensibilidad	Rayos UV-A UV-B	Rayos UV-A UV-B	Rayos UV-A UV-B
Longitud de onda de detección	200-370 nm	280nm a 390nm	200-370nm
Precisión	+-1	+-1	+-1
Precio aproximado	\$6	\$8	\$16
Tamaño	1.9 x1.5 cm	1.4x1.2cm	2.8X1.2 cm

Fuente: Autoría

Como se aprecia en la Tabla 12, las características de los diferentes modelos son bastantes similares, por lo que, se selecciona el sensor GUVA-S12SD por sus

características en cuanto precio, tamaño reducido y disponibilidad en el mercado nacional.

Finalmente, en la Tabla 13 se pueden observar las características de sensores de agua o lluvia, el que cuente con las mejores características será elegido para formar parte del nodo sensor.

Tabla 13

Comparativa de sensores de lluvia para el nodo sensor

CARACTERISTICAS	MODELOS	
	MH-RD YL-83	T1592
		
Compatible con Arduino	SI	SI
Voltaje de operación	3.3 V-5 V	3.3V -5 V
Corriente de trabajo	15mA	<20mA
Zona de detección	32mm x 14mm	40mm x 16mm
Área de sensado	7.2cm ²	6.4cm ²
Temperatura de funcionamiento	-15 °C – 55 °C	10 °C – 30 °C
Precio aproximado	\$2.50	\$3
Tamaño	1.9 x1.5 cm	62x20x8mm

Fuente: *Autoría*

Los sensores presentan características similares, pero se elige el sensor MH-RD YL-83 debido a que tiene un área mayor de sensado, tiene un costo reducido y existe disponibilidad en el mercado nacional.

Por otra parte, para el nodo central, el Gateway o puerta de enlace; puente entre el nodo sensor y el servidor. Tiene que ser lo suficientemente potente como para recopilar el tráfico de todos los nodos que conforman la red inalámbrica. En la Tabla 14 se realiza una comparación de diferentes placas de procesamiento para el diseño del Gateway.

Tabla 14

Selección de placas de procesamiento para el diseño del Gateway o puerta de enlace

CARACTERISITICAS	MODELOS	
	Raspberry Pi 3	Beagle Bone
		
Procesador	Broadcom BCM2837	AM335X
CPU	1,2 GHz de 64 bits con 4 núcleos ARMv8	Cortex-A8 + 2Xpru (200Mhz)
Memoria del sistema	1Gb LPDDR2	256 MB LPDDR
Almacenamiento	Tarjeta microSD	Tarjeta microSD
Conectividad de red	Ethernet 10/100Mbps, RJ45, Wi-Fi, 802.11n Wireless LAN	Ethernet 10/100 Mbps
Consumo	2,5 A/12,5w/5V	2 A/5V
Sistema Operativo	Linux	Linux, Windows, Android
Dimensiones	8,5X5,6X1,7cm	12,38X7,20 cm
Precio	\$60	\$85

Fuente: *Autoría*

Debido a sus características de costo, reducido tamaño y disponibilidad en el mercado nacional se elige la placa Raspberry Pi 3; el nodo central está encaminado a

brindar una mayor capacidad de procesamiento que el nodo sensor y el nodo actuador, debido a la cantidad de datos que debe procesar.

Para la comunicación inalámbrica con el nodo sensor se utiliza el Dragino LoRa Shield seleccionado previamente en la Tabla 9 y a su vez, la placa Arduino Uno seleccionado en la Tabla 8, ya que el shield LoRa es diseñado específicamente para esta placa.

Por otra parte, para registrar los datos enviados por cada uno de los sensores y que el usuario los visualice de una forma más comprensible mediante graficas desde cualquier lugar a través de internet se realiza una comparación de dos plataformas IoT.

Tabla 15

Comparación de plataformas en la nube para el nodo central

	
CARACTERISITCAS	
Es una plataforma de código abierto	Es una plataforma de código abierto
Permite analizar y visualizar datos con herramientas de MATLAB.	Permite analizar y visualizar datos
Integración con diversos dispositivos Hardware y software	API y protocolos que puede conectar desde cualquier hardware a Ubidots por medio de HTTP, MQTT, TCP, UDP.
Realiza análisis de datos en tiempo real	Permite el análisis de datos en tiempo real
Comparte datos con canales públicos, API , MQTT, MATLAB analytics.	Tiene gestión de usuario
Recopila datos en canales públicos y privados	Posee almacenamiento y Backend
Tiene una interfaz amigable	Realiza Informes programados

Fuente: *Autoría*

La plataforma que se elige es ThingSpeak, ya que es una plataforma de código abierto, fácil de configurar y tiene una API bien documentada. Además, tiene una

interfaz amigable con el usuario. En cuanto se refiere a cargar los datos, esta plataforma permite la carga simultanea de datos, por lo que no hay un retraso que sea significativo.

3.1.2. *Análisis de la Situación Actual*

El presente proyecto se desarrolla en la parroquia de Tocachi perteneciente al cantón Pedro Moncayo donde gran parte de sus habitantes se dedican a la ganadería, gracias a ello han formado una asociación lechera con el fin de generar ingresos económicos y aportar al bienestar de sus familias. Para ello han destinado parte de sus extensiones de terreno al cultivo de pasto para la alimentación del ganado.

Para esta investigación un miembro de la asociación lechera ha permitido el acceso a su cultivo de pasto que tiene una extensión de $3400m^2$, del cual se toma una un área de $1000m^2$ considerada una parcela como muestra para el desarrollo de esta investigación.

Figura 22

Extensión del cultivo para implementar el sistema



Nota. El rectángulo de color verde representa la extensión total del cultivo y las líneas de color rojo representa una parcela. **Fuente:** Autoría

Mediante la investigación in situ al cultivo de pasto se pudo evidenciar que el mecanismo de riego utilizado es manual y se aplica el método de riego por aspersión. El usuario manifiesta que distribuye los aspersores en toda la extensión a una distancia prudente uno del otro en base a su experiencia. Los aspersores están en funcionamiento de tres a cinco horas, estos son reubicados para abastecer de agua toda la extensión de terreno. Este proceso lo realiza todos los días dependiendo de las condiciones climáticas presentes.

Adicionalmente, se recolecta datos de la zona del cultivo como la distancia que existe entre la parcela y la fuente de agua, y la distancia entre el centro de control (nodo central) y la parcela. Estos datos se observan en la Figura 23.

Figura 23

Vista aérea de la parcela y el centro de control



Nota. El rectángulo de color azul representa la fuente de agua y el cuadrado de color naranja representa el centro de control. **Fuente:** Autoría

Dado que el modelo iterativo plantea un entorno de etapas con retroalimentación, para la primera iteración se va a iniciar con el análisis del caudal y el consumo de agua, antes del diseño y pruebas del sistema propuesto. Para ello se recolecta datos utilizando el sistema de riego manual empleado actualmente, asumiendo que el riego lo realiza 5 horas diarias, estos datos se observan en la Tabla 16.

Tabla 16

Datos del caudal y consumo de agua del sistema de riego manual actual

Caudal	Consumo de Agua diarios	Consumo de agua mensual
18 l/min	5.4m ³	162m ³

Fuente: Autoría

3.1.3. Diseño del Sistema Propuesto: Primera Iteración

La red de sensores inalámbricos se construye con un nodo sensor que se encargan de la recolección de datos a través de sensores, y la emisión de señales de control se realiza a través de un nodo actuador encargado de ejecutar la orden que envía el nodo central; el nodo actuador se pondrá en ejecución en la segunda iteración.

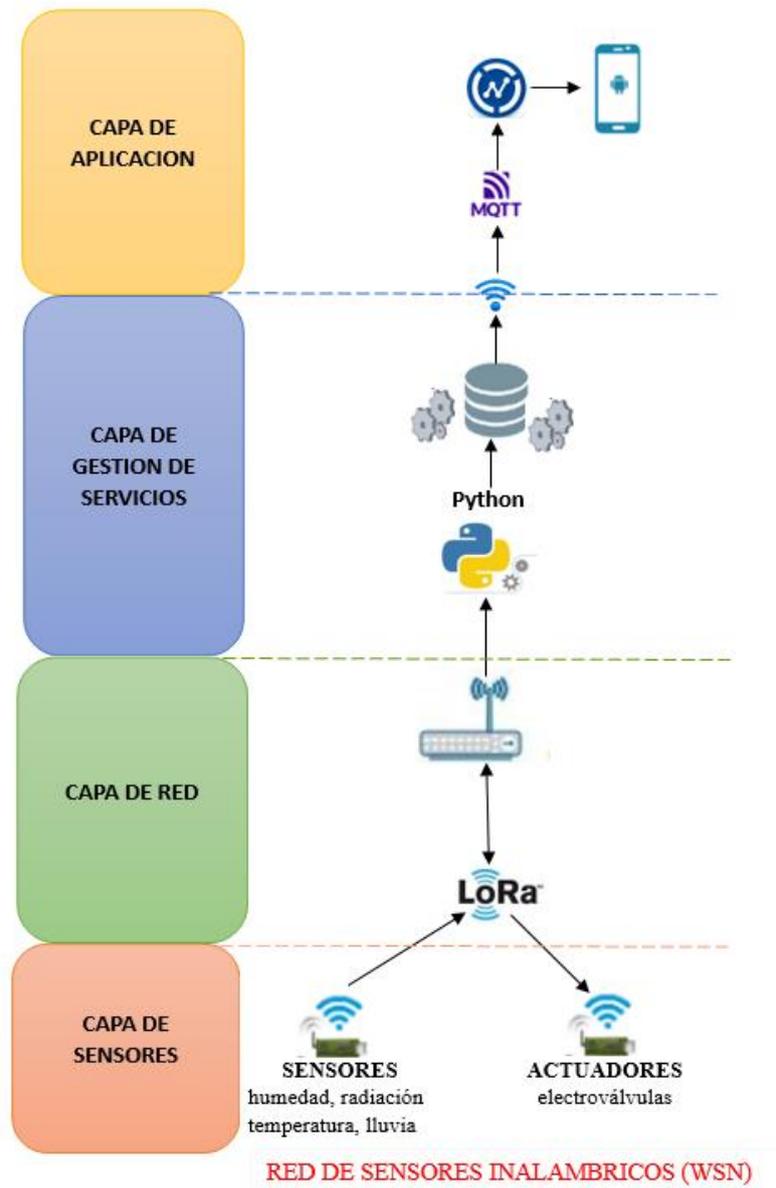
La comunicación entre los nodos y el Gateway, ubicado en el nodo central, se realiza a través de tecnología LoRa, cada nodo se comunica directamente con el Gateway formando una topología tipo estrella. La tecnología LoRa utiliza la topología tipo estrella ya que, al ser una tecnología de bajo consumo de energía, cada nodo se comunica con el nodo principal, evitando el consumo de recursos al intercomunicarse con otros nodos de la red.

El nodo central receptara los datos presentándolos en una plataforma IoT mediante gráficos de acuerdo con el tiempo y los valores obtenidos por los sensores a través del protocolo MQTT.

El modelo de referencia IoT aplicado al sistema se muestra en la Figura 24. Además, se describe su estructura y modo de operación.

Figura 24

Modelo de referencia IoT aplicado al sistema de riego



Fuente: Autoría

Capa de sensores: En esta capa se ubican el nodo sensor y el nodo actuador. El nodo sensor encargado del sensado de variables ambientales y el nodo actuador cumple la orden que envía el nodo central, es decir activar o desactivar el riego.

Capa de red: En esta capa trabaja LoRa, tecnología inalámbrica que permite la comunicación por radiofrecuencia entre el nodo sensor, nodo central y nodo actuador, respectivamente.

Capa de gestión de servicios: El Gateway o puerta de enlace recibe los datos enviados por el nodo sensor y usando del lenguaje de programación Python procesa y envía los datos hacia el servidor IoT a través del protocolo MQTT.

Capa de aplicación: Los datos recibidos en el servidor IoT también son visualizados en la aplicación móvil Thing View Free, estos datos son representados a través de gráficos fácil de comprender y el usuario puede acceder a ellos desde cualquier lugar con conexión a internet.

A continuación, se da a conocer los componentes que forman parte de la red de sensores inalámbricos WSN.

3.1.3.1. Nodo Sensor

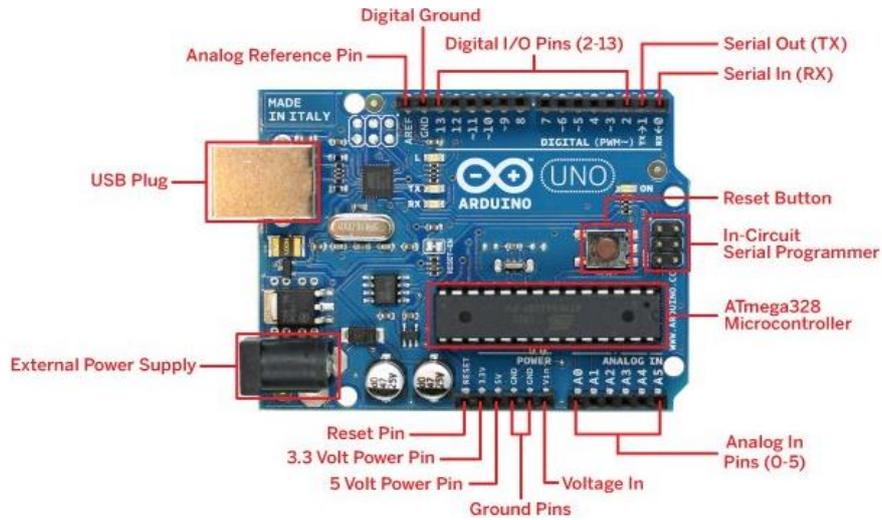
El nodo sensor es el encargado de recopilar y transmitir datos de los parámetros climáticos que influye en el desarrollo del cultivo de pasto. Cada nodo sensor está formado por tres partes importantes que son: microcontrolador, sensor y transceptor.

El microprocesador Arduino UNO, poseen puertos de entrada, salida y de comunicación. Los pines digitales pueden configurarse como entrada para leer sensores y como salida para actuadores. Los pines analógicos de entrada usan un conversor analógico/digital y sirven para leer sensores analógicos.

En la Figura 25 se detalla más a profundidad las partes del Arduino UNO.

Figura 25

Mapa de pines de Arduino UNO

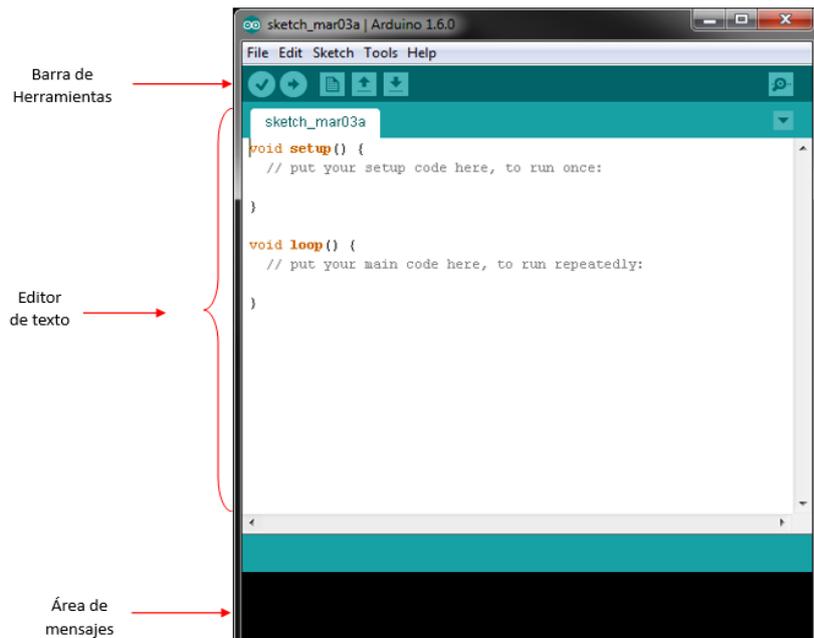


Adaptado de *Mapa de pines Arduino UNO* por J. Crespo, 2018, *Aprendiendo Arduino* (<https://aprendiendoarduino.wordpress.com>)

En cuanto al software, Arduino utiliza Arduino IDE, un software de código abierto compuesto por un editor de texto, un área de mensajes y una barra de herramientas.

Figura 26

IDE de Arduino



Adaptado de *Uso del Software Arduino (IDE)* por L. Aljundi, 2022, *Guía de Arduino* (<https://docs.arduino.cc/learn/starting-guide/the-arduino-software-ide>)

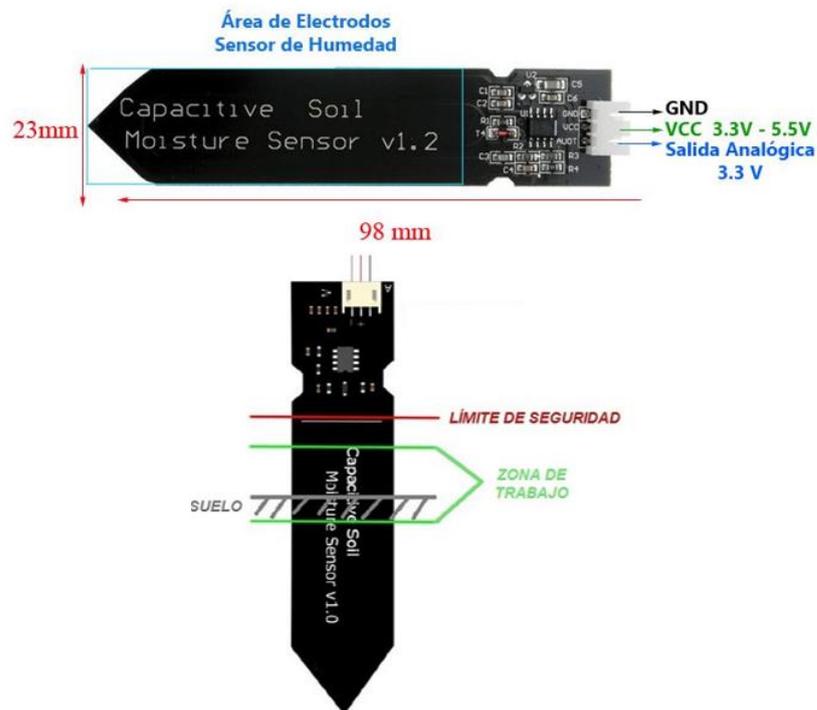
Ahora bien, los sensores que sensan las variables ambientales son: un sensor de humedad del suelo capacitivo, un sensor de temperatura LM35, un sensor de radiación UV GUVA S12SD y un sensor de pluviosidad MH-RD LY-83.

El sensor de humedad del suelo capacitivo mide la humedad del suelo utilizando el principio de capacitancia entre electrodos. La capacitancia dependerá de la humedad del suelo, así que para un suelo muy húmedo tendrá una capacitancia muy baja y para un suelo muy seco tendrá una capacitancia muy alta. La salida analógica entrega un voltaje analógico desde 0V para el suelo muy húmedo hasta 5V para un suelo muy seco.

Como este sensor va a estar en contacto permanente con el suelo se debe respetar un límite de profundidad al instalar el sensor en el terreno, en la Figura 27 se puede observar los pines del sensor capacitivo, la zona de trabajo y el límite de seguridad que se debe mantener para usar este tipo de sensor.

Figura 27

Sensor de humedad del suelo capacitivo



Adaptado de *Sensor de humedad capacitivo* por M. Acosta, 2018, Mi Arduino Uno (<https://uelectronics.com>)

El sensor de temperatura LM35 es un dispositivo de circuito integrado de precisión con un voltaje de salida proporcional a la temperatura centígrada. No requiere ninguna calibración para proporcionar precisión garantizada en el rango de -55 a 150 °C. Este sensor entrega una lectura en una salida analógica directamente en grados centígrados, por cada grado centígrado medido, entrega en su salida un valor de voltaje de 10mV. Asimismo, tiene un calentamiento muy bajo, inferior a 0,1 °C ya que solo extrae 60 µA. En la Figura 28 se da a conocer los pines del sensor LM35.

Figura 28

Sensor de temperatura LM35



Adaptado de *Esquema eléctrico LM35* por L. Llamas, 2015, Ingeniería, Informática y Diseño (<https://www.luisllamas.es/medir-temperatura-con-arduino-y-sensor-lm35/>)

El sensor de radiación UV GUVA S12SD está basada en un fotodiodo UV que puede detectar un rango de 240-370nm, cubriendo el espectro UV-B y la mayor parte del UV-A con un ángulo de detección de 130 grados. Permite medir la radiación ultravioleta a partir de su salida analógica a través de la Ec. 5 (Kakao Corp, 2021)

$$V_o = 4.3 \times \text{consumo del fotodiodo} (\mu A) \quad \text{Ec. 5}$$

En la Figura 29 se puede apreciar el sensor de radiación UV con los pines correspondientes.

Figura 29

Sensor de radiación UV GUVVA S12SD



Adaptado de *Sensor GUVAS12SD* por Kakao Corp, 2020, Arduino Sensor for Everyone (<https://arduino.steamedu123.com/entry/GUVA-S12SD-Ultraviolet-Light-Sensor>)

El sensor de lluvia YL-83 funciona a través de la detección de gotas de lluvia en lugar de un umbral de nivel de señal. Lo que hace este módulo es crear un corto circuito cada vez que las pistas se mojan. El agua hace que se cree un camino de baja resistencia entre las pistas con polaridad positiva y las pistas conectadas al GND.

Actúa como una resistencia variable que cambia de 100Kilo ohm cuando esta mojado a 2 mega ohm cuando está seco.

Figura 30

Esquema de montaje del sensor de lluvia YL-83



Adaptado de *Esquema sensor de lluvia* por L. Llamas, 2015, Ingeniería, Informática y Diseño (<https://www.luisllamas.es/arduino-lluvia>)

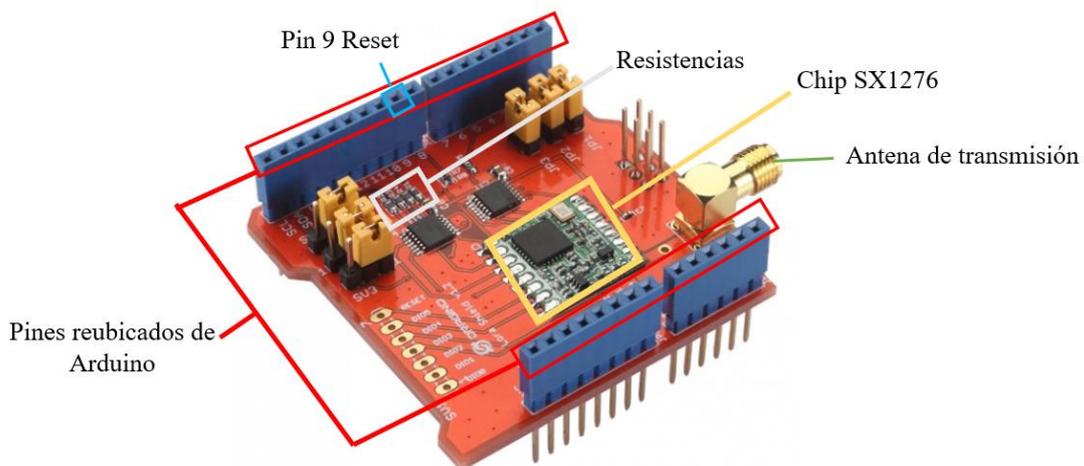
También, cuenta con un circuito de retardo especial que permite un intervalo de aproximadamente dos minutos entre gotas de lluvia antes de asumir el estado de apagado, es decir sin lluvia. Esto es importante ya que permite que el sensor distinga con precisión entre el cese de la lluvia y la lluvia ligera.

La señal digital que envían dichos sensores ingresa al microcontrolador y mediante comunicación UART los datos obtenidos pasan al transceptor LoRa para seguidamente transmitirlos de forma inalámbrica hacia el nodo central.

El Dragino LoRa Shield, es un shield para Arduino basado en el chip Semtech Sx1276, trabaja con las bandas 915Mhz/868Mhz/433 MHz, alcanza una sensibilidad de -148dBm, tiene una capacidad de enviar paquetes a largas distancias con baja velocidad de datos mediante la técnica de modulación LoRa de espectro ensanchado. Además, se posee bibliotecas de código abierto compatible con Arduino UNO, utilizado en este proyecto.

Figura 31

Mapa de pines de Dragino LoRa Shield



Adaptado de *Dragino LoRa Shield* por Dragino, 2018, LoRa Shield
(https://wiki.dragino.com/index.php?title=Lora_Shield)

Entre sus especificaciones según (Dragino Technology Co. LTD., 2019) se encuentran las siguientes:

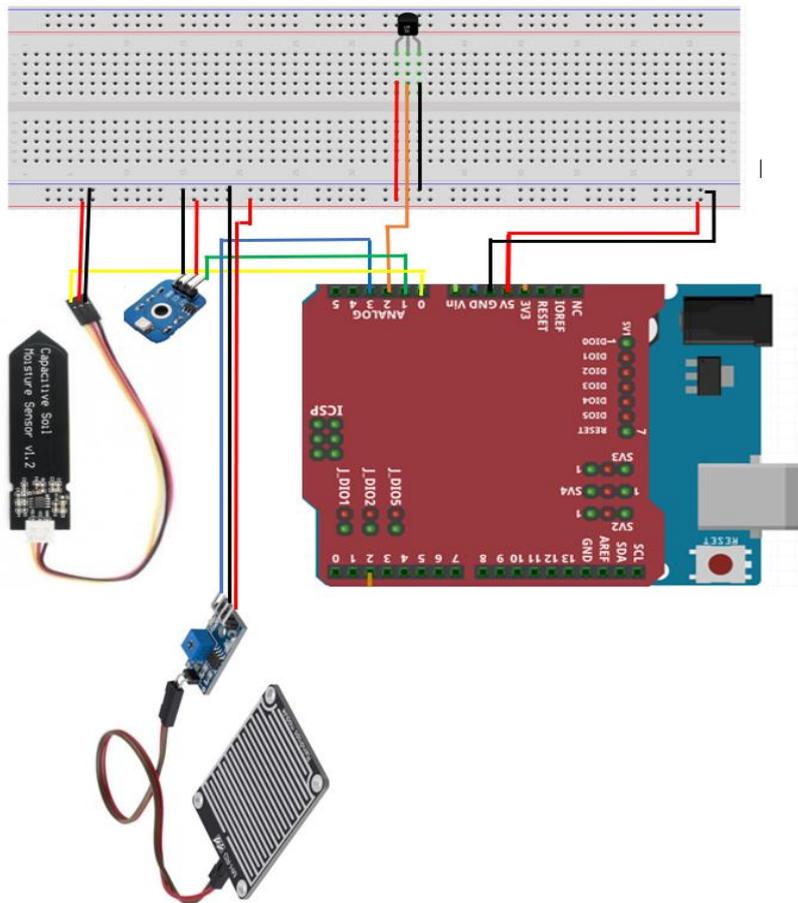
- Presupuesto de enlace máximo de 168 dB
- Potencia de salida de +20dBm -100mW
- Tasa de bits programables hasta 300 kbps
- Alta sensibilidad hasta -148dBm
- Excelente inmunidad a interferencias

- RRSI dinámico de 127dB
- Detección de preámbulos
- Modulación FSK, GFSK, MSK, GMSK, LoRaTM y OOK
- Paquetes de hasta 256 bytes con CRC

En el diagrama que se muestra en la Figura 32 se observa el diseño esquemático del nodo sensor. Dragino LoRa Shield se encuentra montado sobre Arduino Uno y los sensores: GUVVA-S12SD, LM35, MH-RD YL-83 y sensor de humedad del suelo capacitivo se encuentran conectados a los pines analógicos del Arduino UNO respectivamente.

Figura 32

Diagrama esquemático del Nodo Sensor



Fuente: Autoría

3.1.3.2. Nodo Central

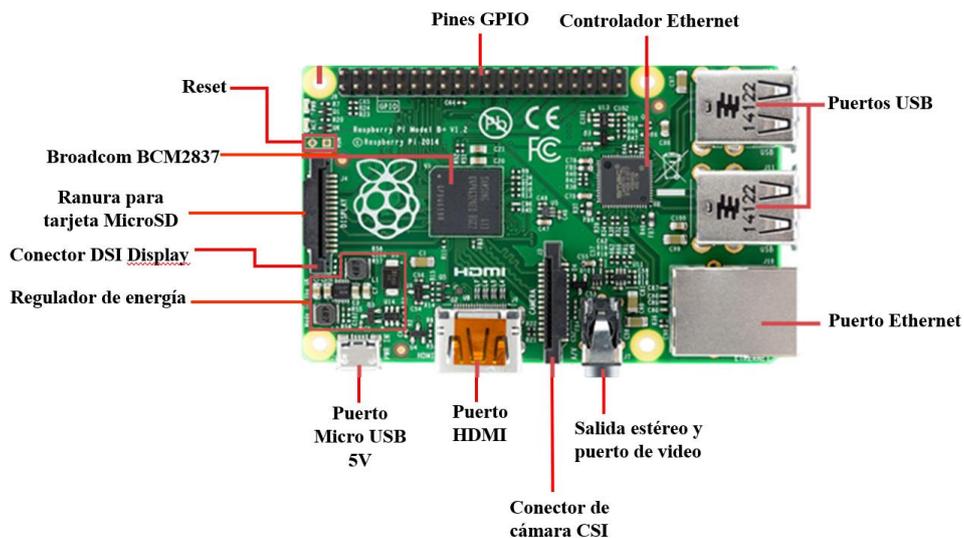
El nodo central cumple la función de Gateway (puerta de enlace), recibe los datos enviados por el nodo sensor mediante comunicación inalámbrica LoRa, procesa estos datos y reenvían los paquetes de datos a un servidor de red ThingSpeak a través del protocolo MQTT que se ejecuta sobre TCP/IP.

Por otro parte, en el nodo central también se realiza el entrenamiento de la red neuronal artificial con los datos recolectados por el nodo sensor; esto se menciona a profundidad en la segunda iteración.

Al igual que el nodo sensor, el nodo central consta de un Arduino UNO y un Shield LoRa para la comunicación y recepción de datos. En cuanto al procesamiento de la información se utiliza una Raspberry Pi 3 es una placa de desarrollo que se puede considerar como una computadora de placa única que funciona con el sistema operativo LINUX y tiene una gran velocidad de procesamiento. El modelo B de Raspberry Pi 3, incluye conectividad inalámbrica, importante para él envío de datos al servidor en la nube. Los componentes de este modelo se muestran en la Figura 33.

Figura 33

Componentes Raspberry Pi 3 Modelo B

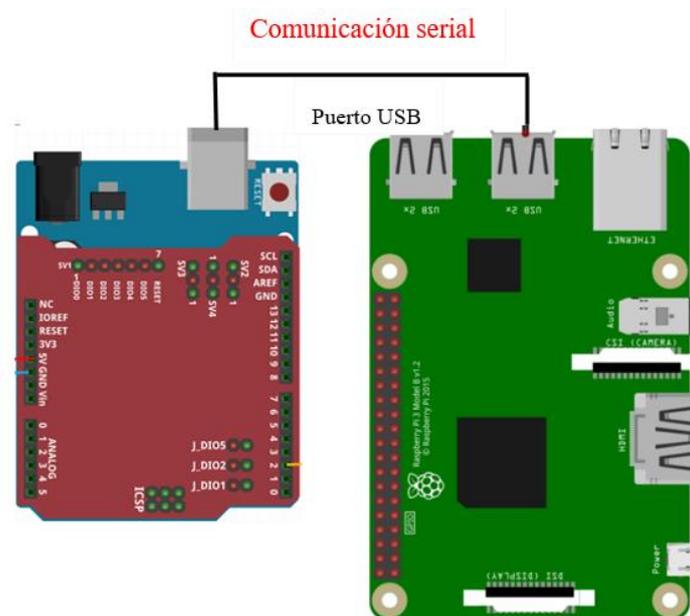


Adaptado de *Raspberry Pi*, 2018, Raspberry Pi 3 Modelo B (<https://www.raspberrypi.com/products/raspberry-pi-3-model-b/>)

Y en la Figura 34 se puede observar el diagrama esquemático del nodo central, Arduino recibe los datos de los sensores por medio de la comunicación LoRa y los envía a la Raspberry Pi 3 mediante comunicación serial donde se leen y realiza el procesamiento de datos, entre si conforman un solo cuerpo creando una puerta de enlace o Gateway hacia la plataforma en la nube.

Figura 34

Diagrama esquemático del nodo central

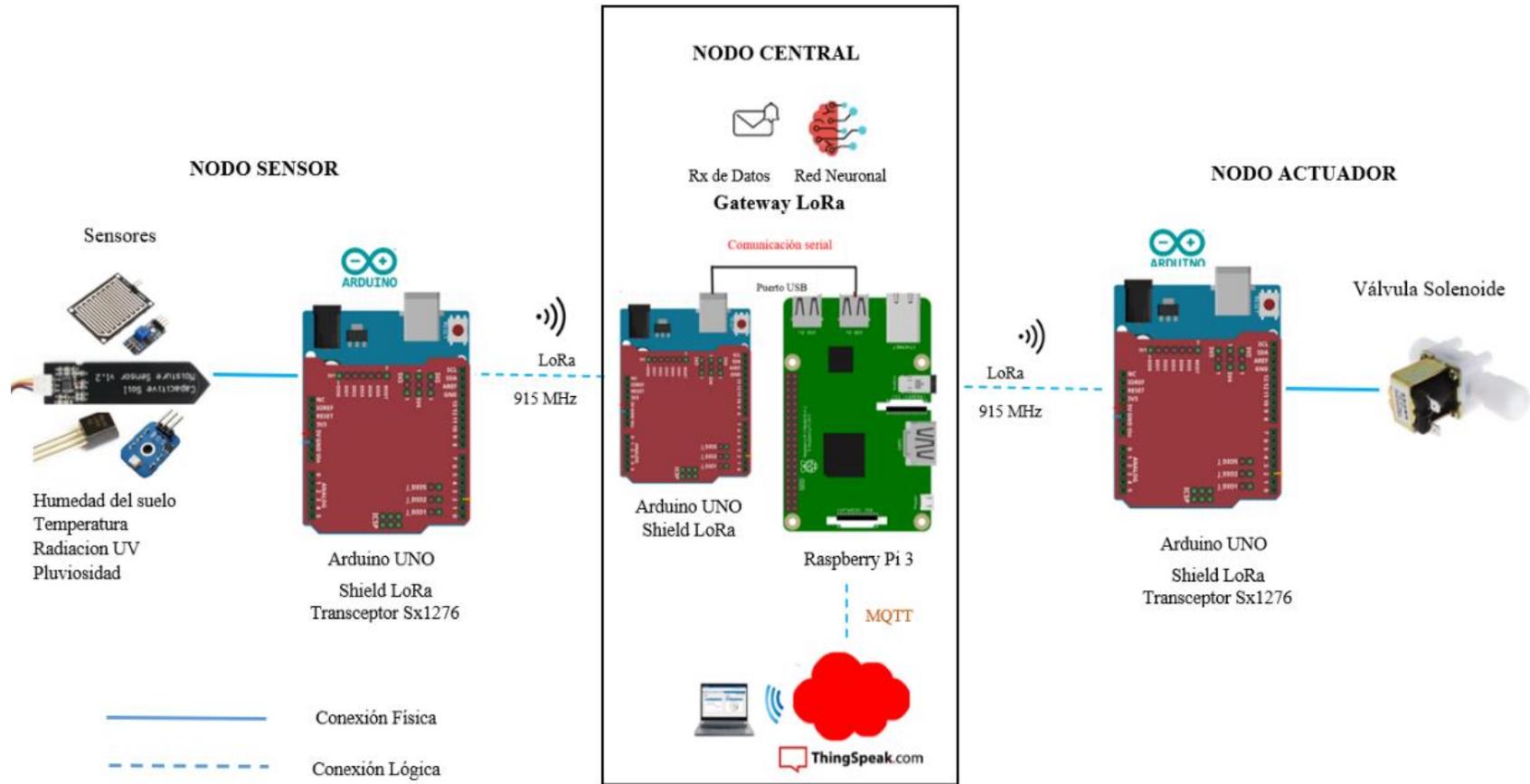


Fuente: Autoría

Finalmente, en la Figura 35 se observa la arquitectura general del sistema y a detalle los dispositivos que forman parte de cada uno de los nodos que conforman la red WSN con tecnología LoRa.

Figura 35

Arquitectura General del Sistema



Fuente: Aatoria

3.1.3.3. Alimentación Eléctrica

Para el correcto funcionamiento de los nodos de la red WSN, se necesita una fuente de alimentación eléctrica. Para que sean nodos netamente inalámbricos no debe tener una fuente de alimentación cableada, por ello se utiliza baterías de litio para proporcionar energía al nodo sensor. Mientras que, el Gateway se alimentara mediante una conexión a la red eléctrica a través de un adaptador de alimentación 110 VAC a 5VDC.

Una vez seleccionado el hardware para el nodo sensor es posible conocer el consumo de corriente de cada dispositivo a través del datasheet. En la Tabla 17 se detalla el consumo energético teórico de cada uno de los dispositivos que forman parte del nodo sensor.

Tabla 17

Consumo energético del nodo sensor

Dispositivo	Voltaje	Corriente	
		Modo Normal	Modo Dormido
Arduino Uno	3.3-5 V	48 mA	46 mA
Modulo LoRa	3.3-5V	10.3 mA	0.0003mA
Sensor capacitivo de humedad del suelo	3.3-5V	5 mA	5mA
Sensor LM35	3-5V	60 mA	60mA
Sensor GUV A S12SD	4-30V	0.06mA	0.06mA
Sensor MH-RD YL-83	3.3-5V	50mA	50mA
TOTAL	3-5V	173.36 mA	161.060mA

Fuente: *Autoría*

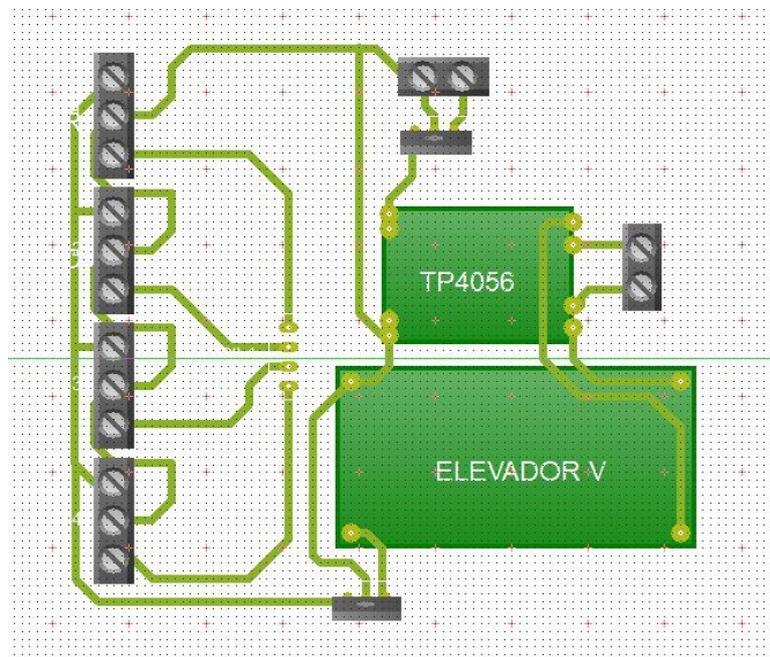
Se toma el consumo de corriente en modo normal ya que el sistema va a estar operativo 24 horas al día. Para ello se tiene disponible una batería de litio de 3.7V a

6000mAh, la batería de litio y su conexión con un panel solar será el encargado de suministrar energía al nodo sensor.

El diseño de la placa de alimentación se observa en la Figura 36, está compuesta por un módulo TP4056; un cargador de batería de litio y un elevador de voltaje DC-DC que convierte el voltaje de entrada de 3,7 a 5V para alimentar el Arduino UNO, el Shield LoRa y los respectivos sensores.

Figura 36

Placa de alimentación nodo sensor



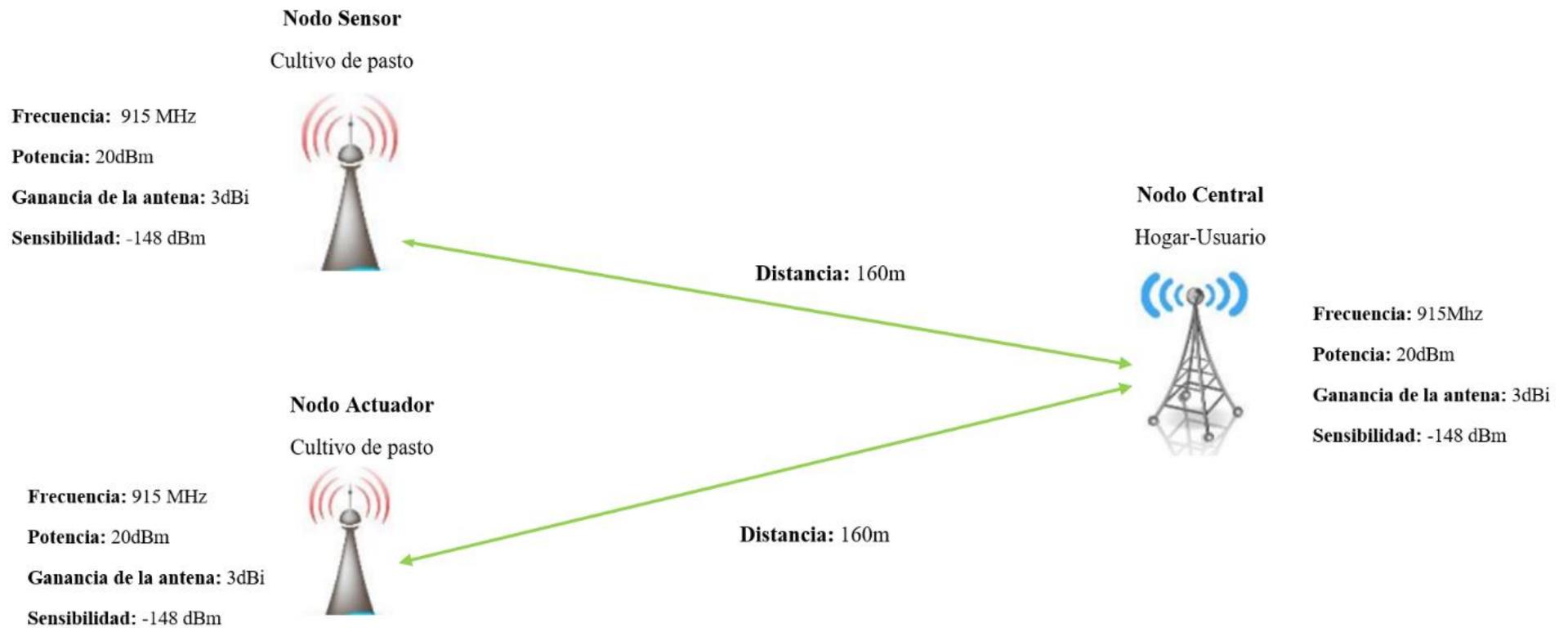
Fuente: Autoría

3.1.4. Diseño de la Comunicación Inalámbrica

En base a los requerimientos definidos se realiza el diseño de la comunicación inalámbrica haciendo uso del software de simulación Radio Mobile el cual permite la simulación del radio enlace. En la Figura 37 se muestra el diagrama de red donde se especifica los parámetros de los nodos y la distancia entre los nodos y el Gateway.

Figura 37

*bytes*Diagrama de red del sistema



Fuente: Autoría

Para determinar la ubicación exacta de los nodos se utiliza la herramienta Google Earth, para establecer la latitud y longitud de cada uno, tal como se especifica en la Tabla 18.

Tabla 18

Coordenadas de cada nodo que conforma la red

COORDENADAS DE LOS NODOS					
Nodos	Coordenadas Geográficas		Coordenadas Angulares		Distancia hacia el Gateway
	Latitud	Longitud	Latitud	Longitud	
Gateway	0° 2'35.21" N	78°17'35.03" O	0.0431	-78.2930	-
Nodo Sensor	0° 2'31.65" N	78°17'38.81" O	0.0421	-78.2941	160.13m
Nodo Actuador	0° 2'31.58" N	78°17'38.77" O	0.0421	-78.2941	160.13m

Fuente: Autoría

La ubicación de los nodos se realizará en una extensión de $1000m^2$. El nodo central o Gateway está ubicado a una distancia de 160 metros del cultivo de pasto.

La distancia de comunicación que permite un enlace inalámbrico es limitada, debido a que puede existir pérdidas por espacio libre, línea de vista, potencia del transmisor y sensibilidad del receptor.

Como parte del diseño del radioenlace entre los diferentes nodos se cuantifican las pérdidas que sufre la señal desde el radio transmisor hasta el receptor, de esta manera verificar la eficiencia que tiene el diseño y a la vez evaluar si es posible su implementación.

Se realizan los cálculos pertinentes del enlace inalámbrico para posteriormente compararlos con la simulación del radio enlace en el software Radio Mobile.

Para que un nodo pueda comunicarse con otro deben realizarlo en la misma frecuencia. Además, los módulos LoRa comparten los mismos parámetros de comunicación, por lo ende, tanto nodo sensor como actuador comparten los mismos resultados.

3.1.4.1. Pérdida del Trayecto en el Espacio Libre

Se realiza los cálculos de las pérdidas en el espacio libre ya que, a pesar de que el medio por el cual transmite un sistema inalámbrico es el aire, también genera pérdidas. Las pérdidas ocurren cuando la onda electromagnética se propaga en el vacío sin tomar en cuenta la absorción ni reflexión de los objetos cercanos, debido a esto, la energía se dispersa desde el transmisor hasta llegar al receptor disminuyendo la potencia de la señal. Para su cálculo se utiliza la Ec. 6 (Abdollah Ghasemi, Ali Abedi, n.d.)

$$L_p = 32,4 + 20 \log(d[km]) + 20 \log(f[MHz]) \quad \text{Ec. 6}$$

Donde:

L_p : Pérdidas en el espacio libre en dB

D : Distancia entre el nodo final y la puerta de enlace en kilómetros

f : Frecuencia de operación en MHz

Reemplazando los valores en la Ec. 6 se obtiene el siguiente resultado.

$$L_p(dB) = 32,4 + 20 \log(0.16[km]) + 20 \log(915[MHz])$$

$$L_p = 75.75 \text{ dB}$$

3.1.4.2. Potencia del Receptor

La potencia del receptor es el nivel de la señal entrante que recibe desde el dispositivo final (nodo sensor) a una distancia lejana. La potencia del receptor es igual a la potencia del transmisor más todas las ganancias de la antena transmisora y receptora menos todas las pérdidas del conector de las antenas.

$$P_{Rx} (dBm) = P_{Tx} (dBm) + G_{Tx}(dBi) - L_{cTx}(dB) + G_{Rx} (dBi) - L_{cRx}(dB) \quad \text{Ec. 7}$$

Donde:

P_{Tx} : potencia del transmisor (dBm)

L_{cTx} : pérdida en el conector de la antena del transmisor (dBm)

G_{Tx} : ganancia de la antena transmisora (dBi)

G_{Rx} : ganancia de la antena receptora (dBi)

L_{cRx} : pérdida en el conector de la antena del receptor (dB)

Reemplazando los valores de la Ec. 7 se consigue los siguientes resultados:

$$P_{Rx} (dBm) = 20(dBm) + 3(dBi) - 0.5(dB) + 3(dBi) - 0.5(dB)$$

$$P_{Rx} (dBm) = 25$$

Al obtener el valor de la potencia del receptor se realiza una comparación con el valor de la sensibilidad del receptor. El objetivo es, que la potencia del receptor sea más alta que la sensibilidad del receptor. Si está por debajo de la sensibilidad, los datos se pierden en el aire. En este caso, el Shield LoRa tiene una alta sensibilidad hasta -148dBm, debido al uso de Chirp Spread Spectrum.

3.1.4.3. Indicación de la Señal Recibida RSSI

La RSSI es la potencia de la señal recibida, una escala de referencia para medir que tan bien un receptor puede escuchar la señal de un transmisor. Este es un valor negativo y se mide en dBm, cuanto más cerca de 0, mejor es la señal recibida. Los valores típicos de LoRa RSSI son:

Si RSSI=-30dBm la señal es fuerte

Si RSSI=-120dBm la señal es débil.

3.1.4.4. Relación Señal a Ruido SNR

Es la relación mínima entre la potencia de la señal deseada y el ruido que se puede demodular, es decir, una relación entre la potencia de la señal recibida y el nivel de potencia del piso del ruido. El piso del ruido son todas las fuentes de señales no deseadas que pueden corromper la señal transmitida; que producirán retransmisiones.

Si el SNR es mayor que 0, la señal recibida opera por encima del piso del ruido y si el SNR es menor que 0, la señal recibida opera por debajo del piso del ruido.

Normalmente, el piso del ruido es el límite físico de sensibilidad. Sin embargo, LoRa trabaja por debajo del nivel del ruido.

Los valores típicos de LoRa están entre -20dB y +10dB. Un valor más cercano a +10dB significa que la señal recibida esta menos dañada. LoRa puede demodular señales que están entre -7.5dB y -20dB por debajo del piso del ruido.

Para cada factor de propagación, existe el límite de SNR, si se alcanza este límite, el receptor no podrá demodular la señal. En la Tabla 19 se da a conocer el límite de SNR que se puede encontrar en función del factor de propagación.

Tabla 19

SNR en función del factor de propagación

Factor de Propagación SF	Relación señal a ruido limite SNR (dB)
7	-7.5
8	-10
9	-12.5
10	-15
11	-17.5
12	-20

Adaptado de *Modulación LoRa: Long Range Modulation*, por E. Solera, 2018
(<https://medium.com/pruebas-de-laboratorio-de-la-modulacion-lora/modulacion-lora-4ad74cabd59e>)

3.1.4.5. Sensibilidad del Receptor

La sensibilidad del receptor indica el nivel de potencia más bajo que el receptor puede recibir o demodular la señal, para ello se requiere el valor mínimo de SNR para que la información pueda decodificar correctamente. Se puede calcular con la Ec. 8 (2CIGroup, 2021)

$$S_{RX}(dBm) = -174 + 10 \log(BW) + SNR_{limite}(dB) + NF(dB) \quad \text{Ec. 8}$$

Donde:

174: es el ruido térmico en 1Hz de ancho de banda a temperatura ambiente

BW: es el ancho de banda del canal en Hz

NF: es el factor de ruido

El factor del ruido es una degradación del SNR causado por la implementación del hardware del receptor en dB. Para chips transceptores de nodo final LoRa SX1272 y SX1276, el factor del ruido es 6dB.

Remplazando los valores en la Ec. 8 se consigue una sensibilidad de -125dBm.

$$S_{RX}(dBm) = -174 + 10 \log(125000) + (-7.5)(dB) + 6(dB)$$

$$S_{RX}(dBm) = -125$$

3.1.4.6. Margen de Enlace

El margen de enlace es la diferencia entre la potencia del receptor y la sensibilidad del receptor (la potencia del receptor debe ser mayor que la sensibilidad del receptor). El margen del enlace debe ser positivo y debe ser de al menos unos pocos dB para que el receptor demodule correctamente la señal.

$$M_{enlace} = P_{RX}(dBm) - S_{RX}(dBm) \quad \text{Ec. 9}$$

Remplazando los valores en la Ec. 9 se tiene un margen de enlace de 150 dBm.

$$M_{enlace} = 25(dBm) - (-125)(dBm)$$

$$M_{enlace} = 150dBm$$

3.1.4.7. Presupuesto de enlace

El presupuesto de enlace es el cálculo de ganancias y pérdidas que sufre la señal del transmisor a través de cables, conectores y espacio libre hasta llegar al receptor. Los componentes que forman parte del presupuesto de enlace son tres: la potencia del transmisor, las pérdidas en la propagación y la sensibilidad del receptor.

El cálculo del presupuesto de enlace se lo realiza para cuantificar el rendimiento del enlace o indicar la calidad de un canal de transmisión. El presupuesto de enlace de LoRaWAN se calcula simplemente con la Ec. 10.

$$P_{\text{enlace}} (\text{dBm}) = P_{Tx} (\text{dBm}) - S_{Rx} (\text{dBm}) \quad \text{Ec. 10}$$

Remplazando la Ec. 10 con los valores correspondientes se tiene un balance máximo del enlace de 145 dBm.

$$P_{\text{enlace}} (\text{dBm}) = 20 (\text{dBm}) + 125 (\text{dBm})$$

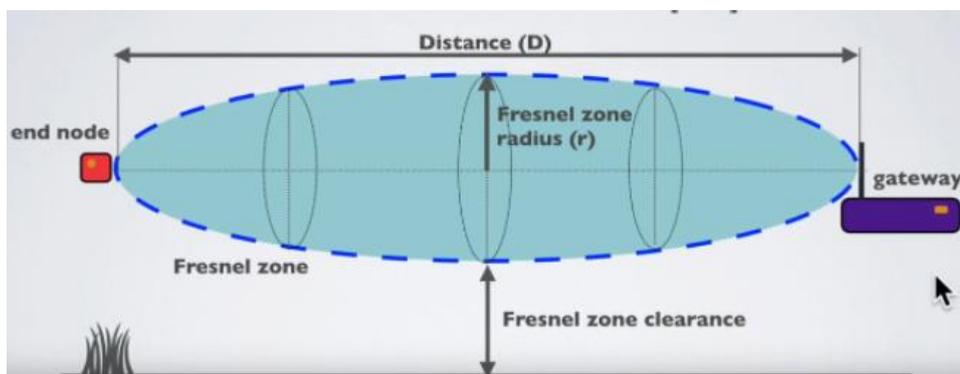
$$P_{\text{enlace}} (\text{dBm}) = 145$$

3.1.4.8. Zona de Fresnel

La zona de Fresnel es una zona de forma elíptica generada por la energía que propaga la antena que transmite, a la antena que recibe, ensanchándose dicha zona el centro del enlace entre las dos antenas. Cualquier objeto dentro de esta zona como por ejemplo árboles, edificios, colinas, puede degradar la señal emitida causando reducción de la potencia, por lo tanto, es importante que al menos el 60% de la primera zona de Fresnel no tenga obstáculos, es decir, este despejada para tener un alcance satisfactorio.

Figura 38

Zona de Fresnel



Adaptada de *Zona de Fresnel*, por LoRa, 2018 (<https://lora.readthedocs.io/en/latest/#free-space-losses>)

La fórmula establecida para el cálculo de la zona de Fresnel es la que se muestra en la Ec. 11.

$$r = 8.657x \sqrt{\frac{D}{f}} \quad \text{Ec. 11}$$

Donde (Ahmadi, 2016) detalla que:

r: Radio de la zona de Fresnel en metros

D: Distancia total entre el nodo final y la puerta de enlace en kilómetros

f: Frecuencia en GHz

Se procede a calcular la zona de Fresnel para el enlace entre el nodo final (nodo sensor y nodo actuador) y la puerta de enlace (nodo central) para determinar si hay un alto grado de obstrucción en el trayecto que deteriore la transmisión.

$$\begin{aligned} r &= 8.657x \sqrt{\frac{D}{f}} \\ r &= 8.657x \sqrt{\frac{0.16}{0.915}} \\ r &= 3.62 \text{ metros} \end{aligned}$$

Se tiene como resultado que el radio óptimo de la primera zona de Fresnel es 3.62 metros, 100 % libre. Normalmente es suficiente despejar el 60% del radio de la primera zona de Fresnel para tener un enlace satisfactorio, es decir, la primera zona de Fresnel debe tener un radio de al menos 2.18 metros para una buena transmisión.

3.1.4.9. Angulo de Azimut

Es el ángulo que indica la orientación en el que se debe girar la antena con respecto al plano horizontal. (Abdollah Ghasemi, Ali Abedi, n.d.)

$$\text{azimut} = \text{tang}^{-1}(\theta) \quad \text{Ec. 12}$$

Donde, θ es la diferencia entre la variación de longitud y latitud

$$\theta = \frac{\Delta \text{longitud}}{\Delta \text{latitud}} \quad \text{Ec. 13}$$

Donde:

LoNs: Longitud del nodo sensor

LoNc: Longitud del nodo central

LaNs: Latitud del nodo sensor

LaNc: Latitud del nodo central

Reemplazando los valores en la Ec. 13 se tiene los siguientes resultados:

$$\theta = \frac{LoNs - LoNc}{LaNs - LaNc}$$
$$\theta = \frac{-78.2941 + 78.2930}{0.0421 - 0.0431}$$
$$\theta = 1.1$$
$$azimut = \text{tang}^{-1}(1.1)$$
$$azimut = 47.72^{\circ}$$

3.1.4.10. Ángulo de elevación

El ángulo de elevación indica la orientación en el que se debe girar la antena con respecto al plano vertical y se calcula con la Ecuación 14 (Abdollah Ghasemi, Ali Abedi, n.d.)

$$\text{Angulo de elevacion} = \text{tang}^{-1}(\alpha) \quad \text{Ec. 14}$$

Donde, α es la diferencia entre las alturas y la distancia

$$\alpha = \frac{\text{AlturaNs} - \text{AlturaNc}}{\text{distancia}} \quad \text{Ec. 15}$$

Reemplazando los valores se consigue el siguiente resultado

$$\alpha = \frac{1.5 - 6(m)}{160(m)}$$
$$\alpha = -0.0281$$

$$\text{Angulo de elevacion} = \text{tang}^{-1}(-0.0281)$$

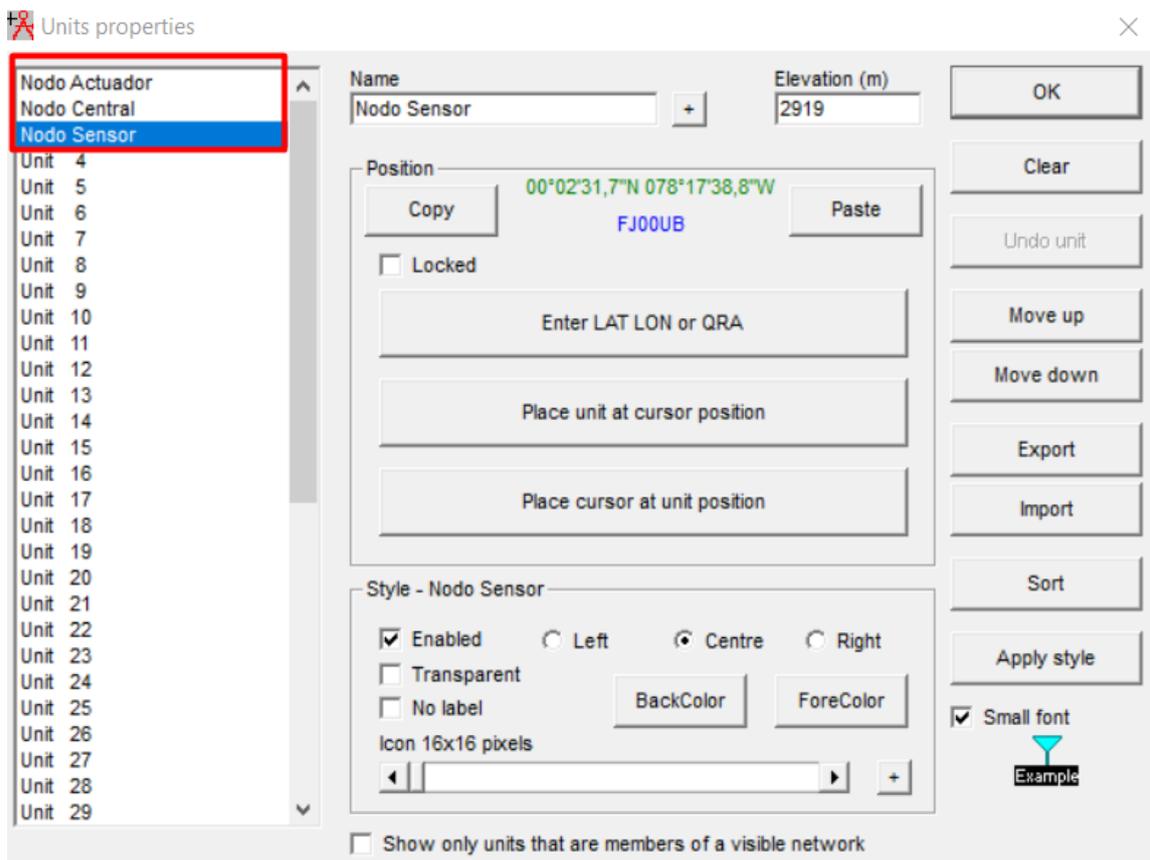
$$\text{Angulo de elevacion} = 1.611^{\circ}$$

3.1.4.11. Simulación del Radio Enlace

Posteriormente, se realiza la simulación del enlace de radio para verificar que existe línea de vista entre los nodos y el Gateway. Con las coordenadas geográficas obtenidas se crean las unidades que conforman el radio enlace, tal como se muestra en la Figura 39, se aprecia la ventana del software Radio Mobile y las coordenadas de los diferentes nodos.

Figura 39

Creación de las unidades con el nombre identificativo de cada nodo

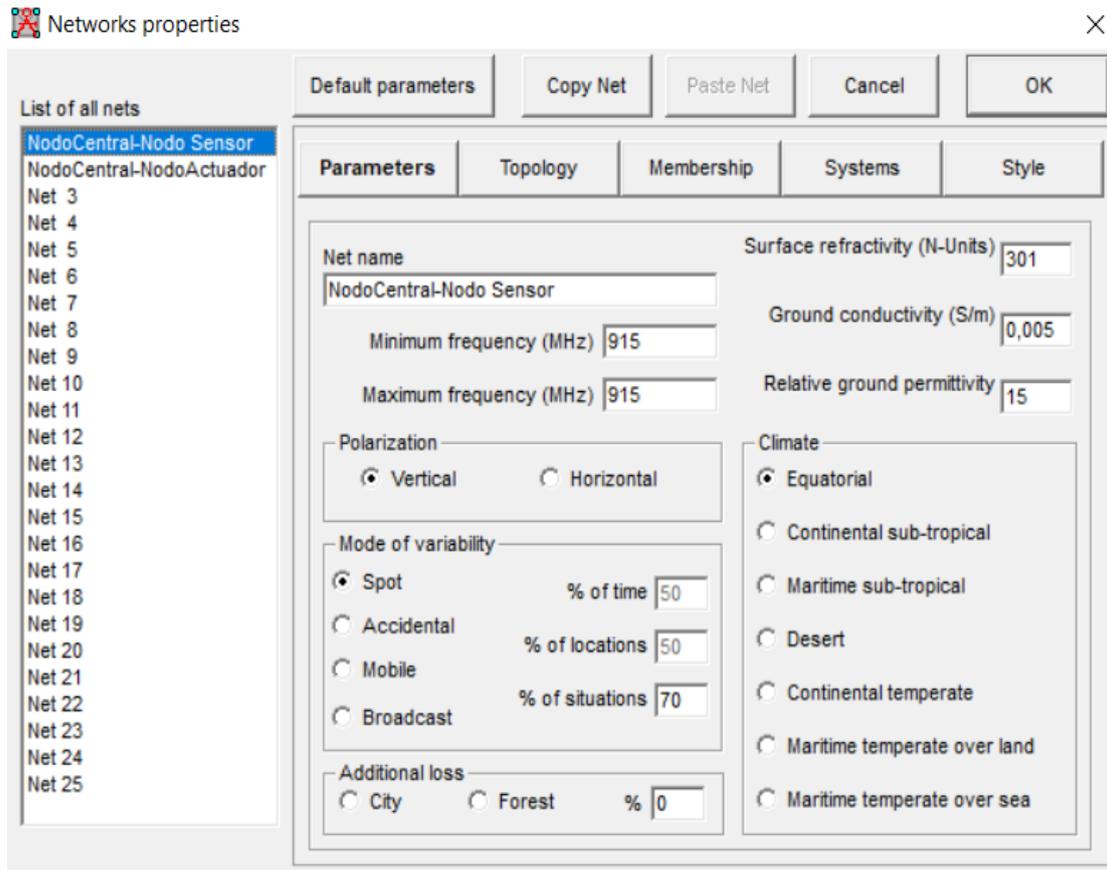


Fuente: Autoría

El siguiente paso es configurar las propiedades de la red, donde los parámetros importantes a tomar en cuenta son los que se muestran en la Figura 40. La frecuencia en la que trabajan los dispositivos de comunicación es de 915Mhz, mientras que la polarización de la antena es vertical.

Figura 40

Configuración de los parámetros principales de la red



Fuente: *Autoría*

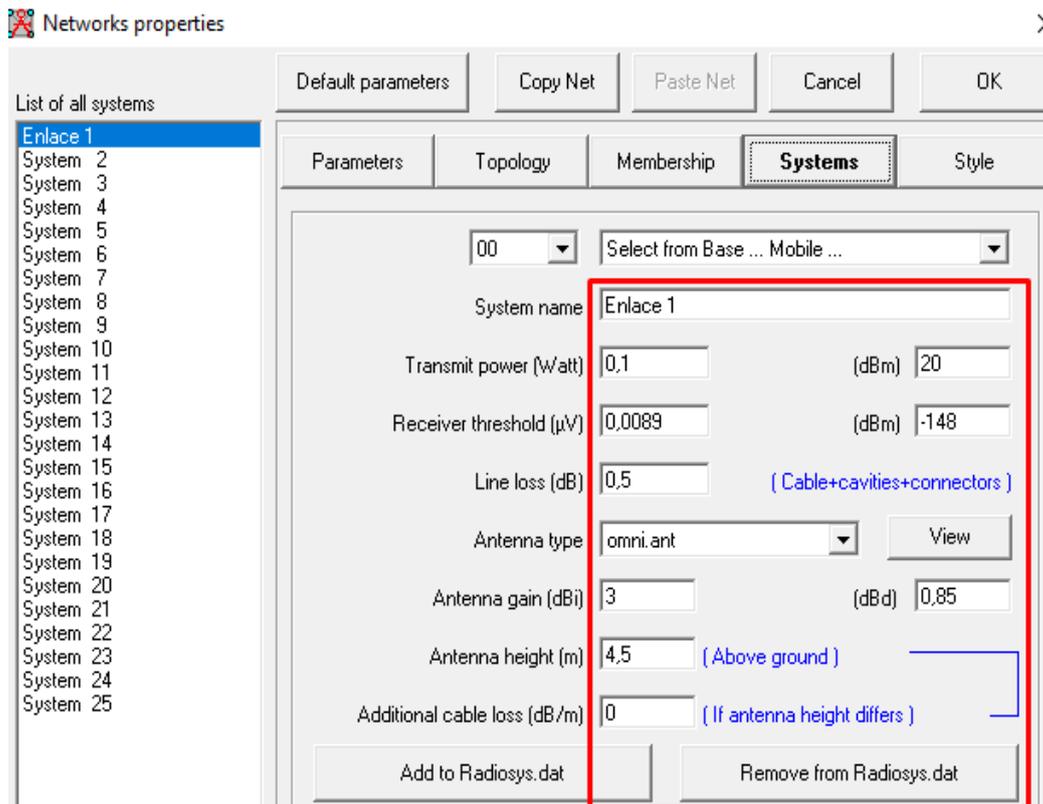
Del mismo modo se debe configurar los parámetros del sistema de comunicación de cada enlace, en este caso, se tienen dos enlaces. Un enlace entre el nodo sensor y el nodo central y otro enlace entre el nodo central y el nodo actuador. En la Figura 41 se observa la configuración del enlace entre el nodo sensor y el nodo central, el enlace entre el nodo central y el nodo actuador comparte las mismas características. Entre los parámetros principales se encuentran los siguientes:

- Potencia de Transmisión (dBm): 20dBm
- Umbral de Recepción (dBm): -148dBm
- Pérdida de línea (dB): 0.5dB
- Tipo de antena: Omnidireccional
- Ganancia de la Antena (dBi): 3dBi
- Altura de la Antena (m): varía de acuerdo con los resultados del enlace

Estas características se establecen conforme la hoja de datos del fabricante del transceptor LoRa y de la antena a implementarse. En cuanto a la altura de las antenas se varia hasta obtener una zona de Fresnel totalmente despejada.

Figura 41

Configuración de los parámetros principales del sistema de enlace

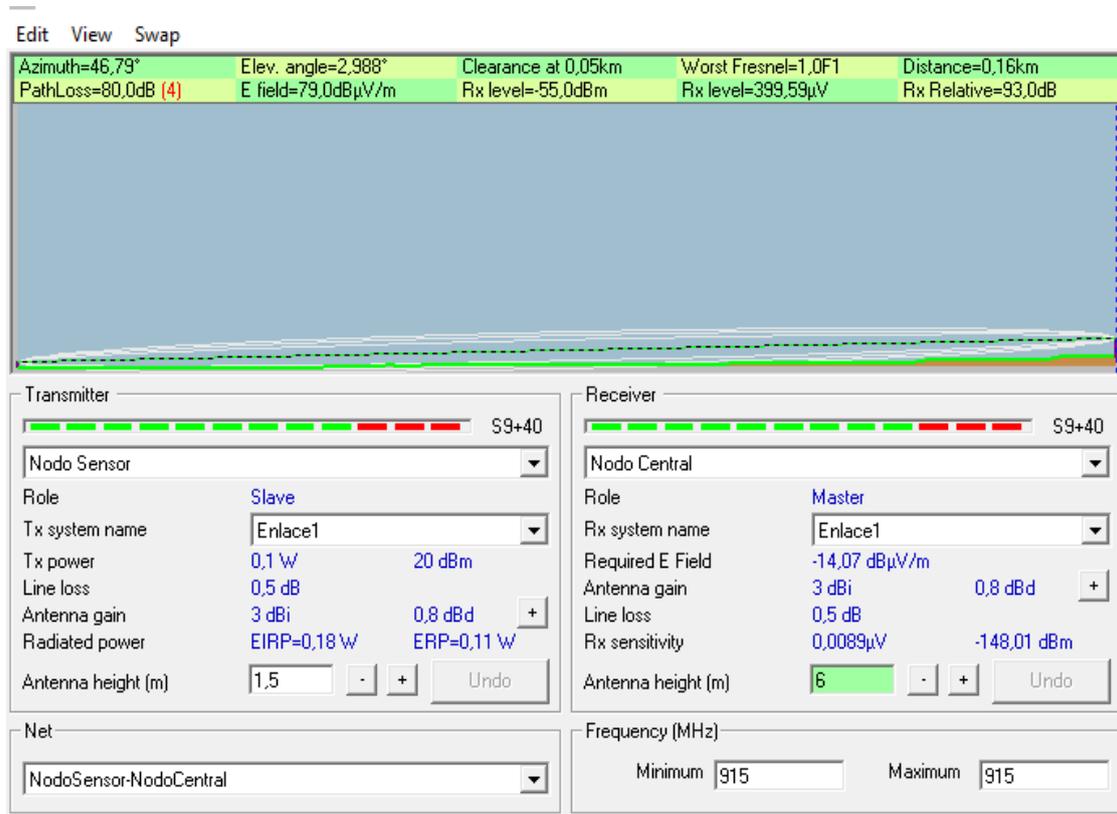


Fuente: Autoría.

En la Figura 42 se puede apreciar los resultados del enlace, es decir la línea de vista entre el nodo sensor y el nodo central.

Figura 42

Línea de vista entre el nodo sensor y nodo central

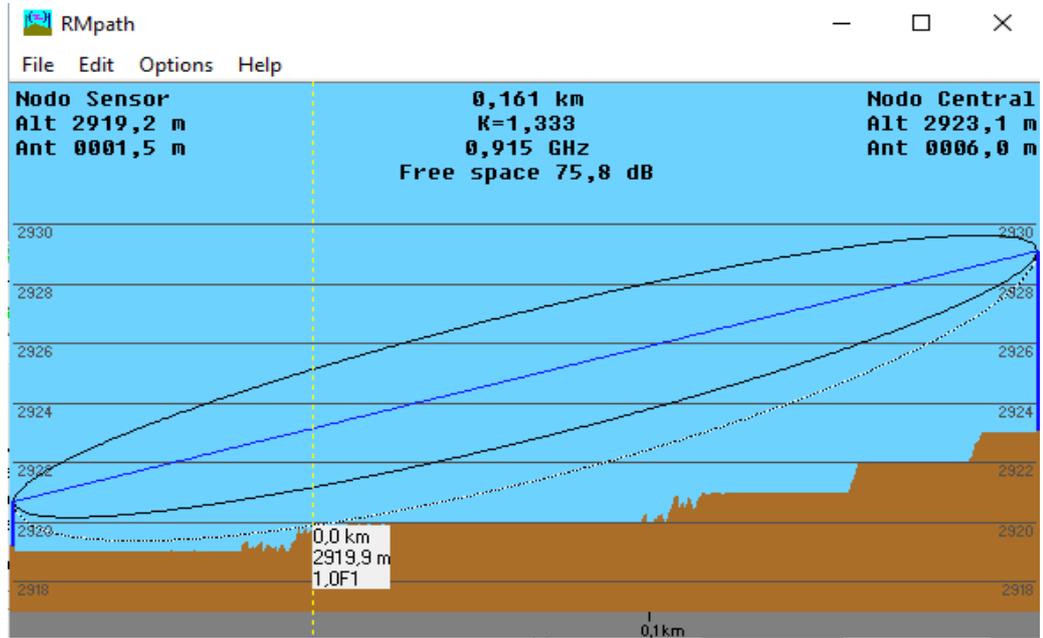


Fuente: Autoría

Para apreciar de mejor manera el perfil y los datos detallados de la zona de Fresnel se exporta el radio enlace a un archivo RMpath. Como se puede observar en la Figura 43, la altura del nodo sensor es de 1.5 metros y el nodo central está ubicado a una altura de 6 metros con una distancia entre ellos de 160 metros. Con respecto a la zona de Fresnel se puede evidenciar que el peor Fresnel es 1.0F ningún pico debe ser menor a 0.6, si es inferior a este valor habría pérdida de señal, por lo que se debe modificar la altura de las antenas.

Figura 43

Zona de Fresnel del enlace entre el Nodo Sensor y Nodo Central en archivo RMPATH



Fuente: Autoría.

Se varia la altura de las antenas para conseguir una zona de Fresnel totalmente despejada, los diferentes escenarios simulados se detallan en la Tabla 20.

Tabla 20

Escenarios simulados radio enlace Nodo Sensor – Nodo Central

ESCENARIOS SIMULADOS NODO SENSOR-NODO CENTRAL									
N	Altura antena Nodo Sensor	Altura antena Nodo Central	Frecuencia (MHz)	Peor Fresnel	Perdida en el espacio libre (dB)	Obstrucción (dB)	Perdida de propagación total (dB)	Ganancia del Sistema (dB)	Peor recepción (dB)
1	0.5	6	915	0.6F1	75.8	0.5	81.3	173	91.7
2	1	6	915	0.8 F1	75.8	-0.5	80.3	173	92.7
3	1.5	6	915	1.0 F1	75.7	-0.8	80	173	93
4	3	6	915	1.2F1	75.7	0.0	80.8	173	92.2
5	4.5	6	915	1.6F1	75.7	0.2	81.0	173	92.0

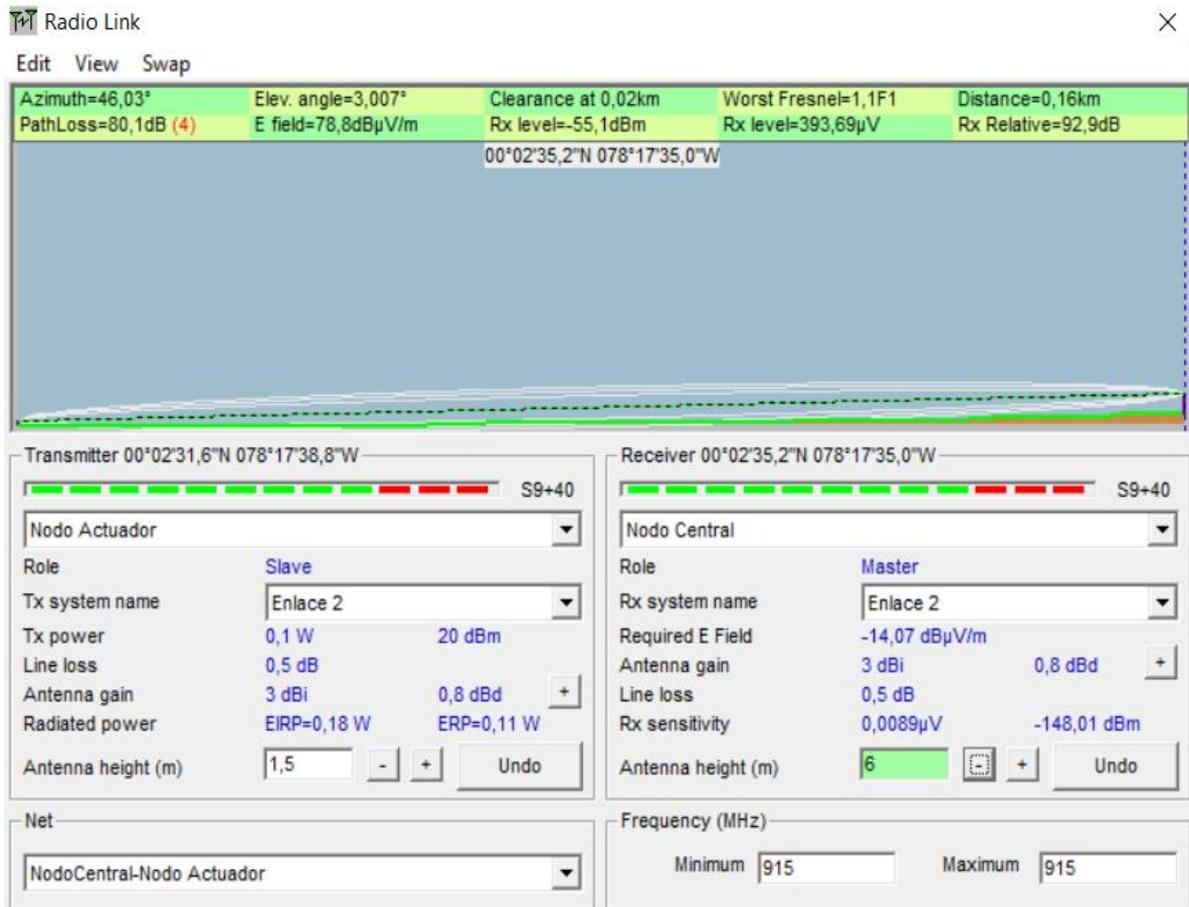
Fuente: Autoría

En el perfil del enlace se puede identificar que las pérdidas de espacio libre no varían mucho con respecto a las diferentes alturas, mientras que las pérdidas de propagación total se reducen como varía la altura de las antenas. Se tiene un Fresnel mínimo de 0.6F1, eso quiere decir que no habrá ningún problema con la señal emitida, ya que siempre debe mantenerse mayor o igual a 0.6, es decir el 60%. La peor recepción varía de 91.7 a 93 dB, este valor permite identificar el valor de margen respecto a la sensibilidad del receptor, cuanto mayor sea este valor más garantía existe que la señal llegue al receptor.

Por otro lado, en la Figura 44 se puede apreciar los resultados del enlace entre el nodo sensor y el nodo central.

Figura 44

Zona de Fresnel radio enlace entre el nodo central y el nodo actuador



Fuente: Autoría

La línea de vista entre el nodo central y el nodo actuador con los diferentes escenarios simulados se observan en la Tabla 21.

Tabla 21

Escenarios simulados radio enlace Nodo Central-Nodo Actuador

ESCENARIOS SIMULADOS NODO CENTRAL-NODO ACTUADOR									
N	Altura antena Nodo Central	Altura antena Nodo Actuador	Frecuencia (MHz)	Peor Fresnel	Pérdida en el espacio libre (dB)	Obstrucción (dB)	Pérdida de propagación total (dB)	Ganancia del Sistema (dB)	Peor recepción (dB)
1	6	0.5	915	0.6F1	75.8	0.3	81.2	173	91.9
2	6	1	915	0.9 F1	75.8	-0.6	80.2	173	92.8
3	6	1.5	915	1.1 F1	75.8	-0.8	80.1	173	92.9
4	6	3	915	1.3F1	75.8	0.7	81.6	173	91.4
5	6	4.5	915	1.6F1	75.8	-0.2	80.7	173	92.4

Fuente: *Autoría*

El peor Fresnel es de 0.6 F1, las pérdidas en el espacio libre son aproximados a 76dB, se tiene una obstrucción que varía de -0.8dB a 0.7dB. Las pérdidas de propagación total son de 81.6dB y la peor recepción se aproxima a 93 dB.

De acuerdo con los datos obtenidos de los diferentes escenarios se puede concluir que todos los enlaces de radio son aceptables, ya que la primera zona de Fresnel supera al 60%. Sin embargo, el escenario 3 es el que se implementa en este proyecto, es decir la altura de la antena del nodo sensor y actuador es de 1, 5 metros y la altura del nodo central a 6 metros.

Finamente se realiza la comparación de los datos calculados con los valores simulación, tal como se puede apreciar en la Tabla 22.

Tabla 22*Comparación de valores calculados y simulados del enlace de radio*

PARÁMETROS	VALORES CALCULADOS	VALORES SIMULADOS	
		Nodo Central Nodo Actuador	Nodo Sensor Nodo Central
Perdidas en el espacio libre	75.75 (dB)	75.8 (dB)	75.8 (dB)
Peor Fresnel	1.0 F1	1.2 F1	1.0 F1
Angulo de azimut	47.72°	46.03°	46.03°
Angulo de elevación	1.611°	3.007°	2.988°

Fuente: *Autoría*

3.1.5. Codificación del Software del Sistema

En esta sección se procede a la codificación de los nodos; la función del nodo sensor es la lectura de las variables ambientales, en el nodo central se configura la puerta de enlace para enviar datos a la nube a través de internet mediante el protocolo MQTT.

3.1.5.1. Nodo Sensor

El nodo sensor es el encargado de recolectar los datos de las variables de humedad, radiación UV, temperatura y pluviosidad.

La lectura del sensor de humedad del suelo capacitivo es muy sencilla, simplemente se realiza la lectura empleando una entrada analógica. Se puede calibrar el sensor obteniendo los valores máximos y mínimos de la salida del sensor, con estos valores se estima el porcentaje de humedad del suelo, en este caso al calibrar el sensor de humedad se tiene valores de 596 en el aire y 252 en el agua.

En cuanto al sensor de radiación UV se obtienen lecturas usando la entrada analógica y entrega a la salida valores entre 0 y 1200mV. El tiempo de lectura es bastante rápido y las respuestas son racionalmente estables, por ello para eliminar desviaciones de posibles interferencias se realiza varias mediciones en cierto periodo de

tiempo y se obtiene un promedio, este valor se compara con los de la tabla del índice UV.

En la Figura 45 se muestran los valores analógicos, el voltaje de salida y once índices UV que se consideran para la lectura de datos del sensor GIVA-S12SD.

Por ejemplo, aplicando la Ec. 5, si el consumo del fotodiodo es 1uA, entonces la tensión de salida será 4.3V.

También puede convertir el voltaje en índice UV dividiendo la tensión de salida en 0.1V. Así que si la tensión de salida es 0.5V, el índice UV es aproximadamente 5.

Figura 45

Tabla de índice UV

	BAJO			MODERADO		
Índice UV	0	1	2	3	4	5
Vout (mV)	<50	227	318	408	503	606
Valor analógico	<10	46	65	83	103	124

	ALTO	MUY ALTO			EXTREMO	
Índice UV	6	7	8	9	10	11 ⁺
Vout (mV)	696	795	881	976	1079	1170+
Valor analógico	142	162	180	200	221	240

Adaptado de *Sensor GUVAS12SD* por Kakao Corp, 2020, Arduino Sensor for Everyone (<https://arduino.steamedu123.com/entry/GUVA-S12SD-Ultraviolet-Light-Sensor>)

Por otra parte, para la codificación del sensor de temperatura hay que tomar en cuenta un aspecto muy importante, si el valor de entrada es 0V la salida será 0 y si el valor de entrada es 5V la salida será 1023. Con esta información se procede a calcular la temperatura en función del voltaje a partir de la Ec. 16.

$$Temp = \frac{Valor \times 5 \times 100}{1024} \quad \text{Ec. 16}$$

Esta fórmula determina el voltaje de entrada analógica del sensor dividiendo 5V para 1023 teniendo como resultado 0.0048V de precisión, a continuación, se debe transformar los voltios en grados, para ello se divide el valor del voltaje obtenido antes entre 0.01V ya que en la ficha técnica especifica que 1°C equivale a 10mV (0.01V).

Posteriormente, para la lectura del sensor de lluvia hay que considerar que este sensor ofrece valores de 0 a 1023. Si el valor de entrada es 5V (sequedad en el tablero de lluvia) , se lee como 1023 y el valor de entrada es 0V (tablero de lluvia completamente mojado) se lee como un valor de 0.

Por otro lado, los dispositivos LoRa cuentan con los parámetros importantes que hay que establecer estos son: Factor de propagación (SF), Ancho de banda y Codificación de errores, la combinación de estos parámetros permite determinar el alcance, la velocidad de datos y sobrecarga por corrección de errores en los datos. Estos parámetros deben ser los mismos en todos los dispositivos de radio LoRa de la red inalámbrica.

Para la configuración del Dragino LoRa Shield se debe descargar la librería correspondiente al microcontrolador, en este caso la librería SPI y RH-RF95.h.

Figura 46

Librerías SPI.h y RH-RF95.h para arduino UNO

```
#include <SPI.h>
#include <RH_RF95.h>
```

Fuente: *Autoría*

La interfaz periférica en Serie (SPI), es un protocolo de transferencia de datos seriales sincronizado que utilizan los microcontroladores para comunicaciones rápidas entre uno más dispositivos.

La librería RadioHead, trabaja con microcontroladores M95 del grupo Sx1276/77/78 y 79, mediante esta librería se tiene compatibilidad entre el módulo LoRa y Arduino UNO, cuenta con una biblioteca completa para enviar y recibir paquetes de datos. Las configuraciones que se deben realizar en esta librería son las que se muestran a continuación.

Frecuencia (Frequency): La frecuencia habilitada para el continente y con la que trabaja el Dragino LoRa Shield es 915 MHz. Se compara la frecuencia del nodo sensor con el nodo central, si la frecuencia no coincide no se establece la conexión.

Potencia de Transmisión (TxPower): Se puede establecer potencias de 2 a 20 dBm utilizando el pin del transmisor PA_BOOST ya que se utiliza el módulo RFM95.

Factor de Propagación (SpreadingFactor): Se encuentra establecida en una escala de 1 al 12. Entre mayor sea este valor, se tendrá un mejor rendimiento en la transmisión de datos.

Ancho de Banda (SignalBandwidth): Entre más bajo sea este valor, el tiempo durante la transmisión será mayor.

Tasa de codificación de errores (CodingRate4): Entre mayor sea este valor, mayor será la fiabilidad de los datos, pero con una sobrecarga en el tiempo de transmisión. Es configurable en un rango de 4/5 a 4/8.

En la Figura 47 se puede observar las configuraciones de la librería Radio Head RF95, donde se detallan los parámetros configurados para el funcionamiento del módulo LoRa.

Figura 47

Configuraciones básicas para el funcionamiento del módulo LoRa- Nodo Sensor

```
Obtencion_datos_Sensores_H_UV_T_LM35_lluvia.ino
--
19 unsigned int lectura_sensor_UV;
20 int valor_indice_uv[CANTIDAD_INDICES_UV] = {46, 65, 83,
21 byte indice;
22 boolean buscando_indice_uv;
23
24 // VARIABLES SENSOR HUMEDAD
25 const int v_aire = 596;
26 const int v_agua = 252;
27 int inter = (v_aire - v_agua) / 3;
28 unsigned int lectura_sensor_H;
29
30 //variables envio
31 int cont_datos = 0;
32 const int num_datos = 10; //TIEMPO DE ENVIO(MODIFICAR)
33
34 void setup() {
35   Serial.begin(9600);
36   if (!rf95.init())
37     Serial.println("init failed");
38   rf95.setFrequency(915.0);
39   rf95.setTxPower(20);
40   rf95.setSpreadingFactor(7);
41   rf95.setSignalBandwidth(125000);
42   rf95.setCodingRate4(5);

```

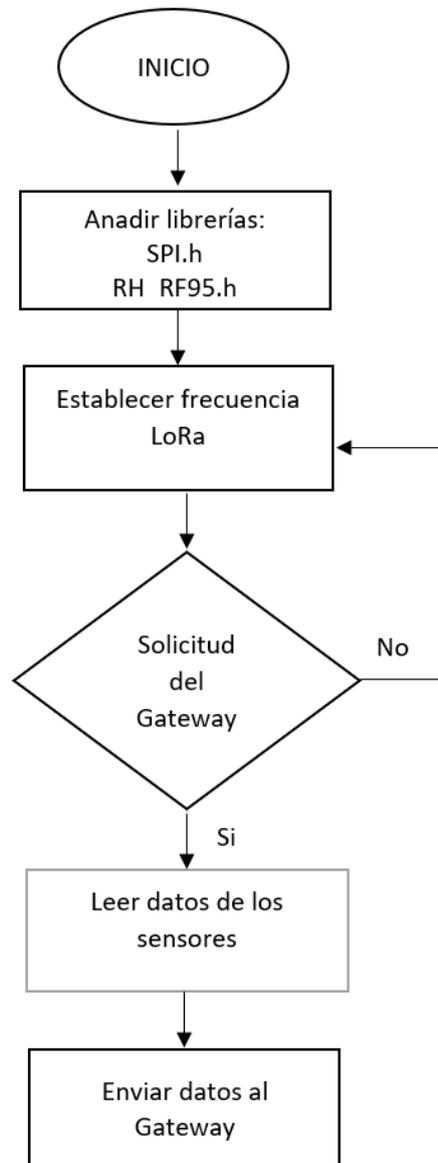
Fuente: Autoría

En la Figura 48 se pretende ilustrar de manera general el funcionamiento del nodo sensor, el software esta desarrollado en el IDE de Arduino, es aquí donde se establece la frecuencia con la cual se va a enviar y recibir paquetes del protocolo de comunicación LoRa.

El nodo sensor espera para recibir una solicitud desde el Gateway (puerta de enlace) para posteriormente enviar los datos obtenidos por los sensores.

Figura 48

Diagrama de flujo del nodo sensor



Fuente: *Autoría*

3.1.5.2. Nodo Central

Como se mencionó anteriormente las configuraciones de los parámetros básicos de todos los dispositivos LoRa deben ser los mismos. En este caso las configuraciones realizadas en el nodo sensor no varían en el nodo central.

Figura 49

Configuraciones básicas modulo LoRa- Nodo Central

```
Nodo_server_arduino_recibir_enviar.ino
1  #include <SPI.h>
2  #include <RH_RF95.h>
3  RH_RF95 rf95;
4  int led = 13;
5  //memoria
6  uint8_t buf_ant;
7  uint8_t len_ant;
8
9  void setup()
10 {
11   pinMode(led, OUTPUT);
12   Serial.begin(9600);
13   while (!Serial) ; // Wait for serial port to be available
14   if (!rf95.init())
15     Serial.println("init failed");
16   // Frecuencia
17   rf95.setFrequency(915.0);
18   // Potencia de Transmision
19   rf95.setTxPower(20);
20   // Ancho de banda
21   rf95.setSignalBandwidth(125000);
22   // Factor de Propagacion
23   rf95.setSpreadingFactor(7);
24   //Tasa de codificacion de errores 5(4/5), 6(4/6), 7(4/7), 8(4/8)
25   rf95.setCodingRate4(5);

```

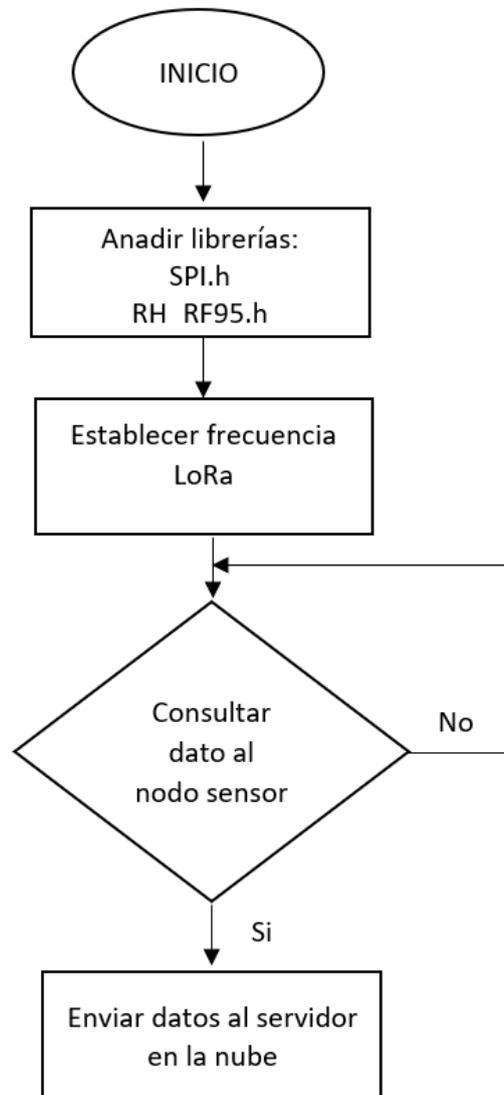
Output

Fuente: *Autoría*

A través del IDE de Arduino se establece la frecuencia para la conexión inalámbrica, y se leen los datos enviados por el nodo sensor. Por otra parte, en el IDE de Python se obtiene los datos de los sensores y se envía al servidor en la nube ThingSpeak a través del protocolo MQTT. El diagrama de flujo se aprecia en la Figura 50.

Figura 50

Diagrama de flujo del nodo central

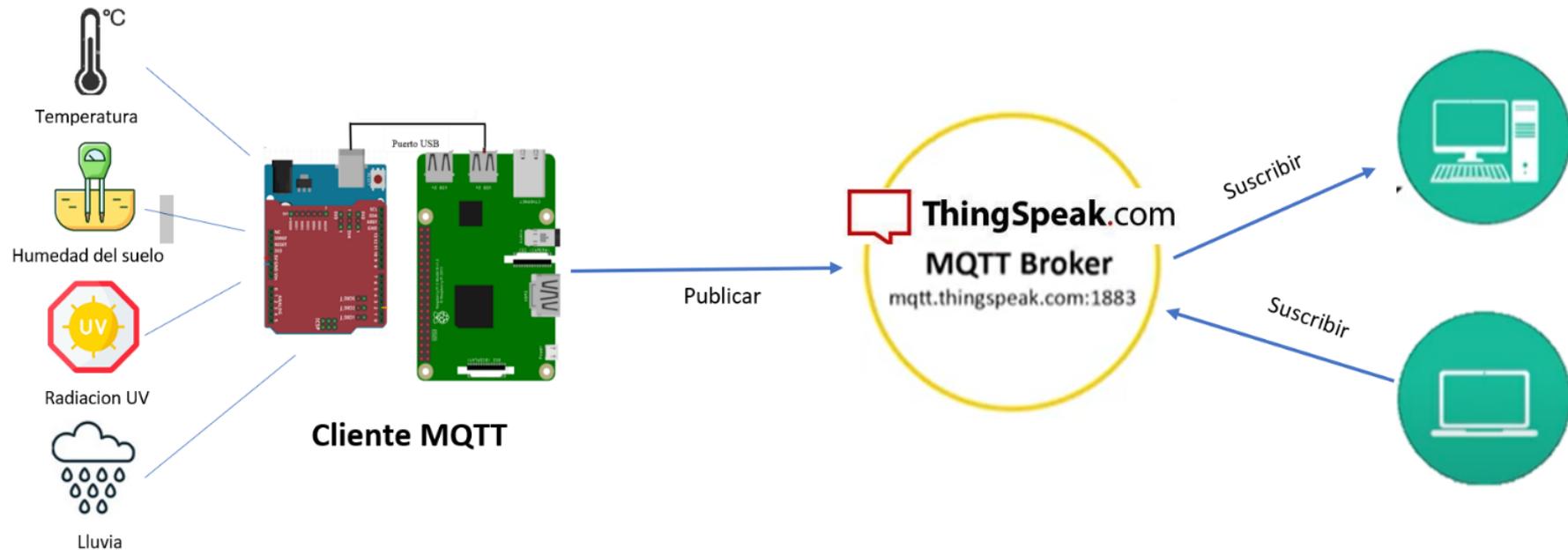


Fuente: *Autoría*

La estructura básica del sistema que utiliza el protocolo MQTT es la siguiente; utiliza una topología en estrella, es decir, todos los clientes se conectan directamente al punto central que hace de servidor, en MQTT este servidor se llama Broker. En este caso la arquitectura del sistema es la que se muestra en la Figura 51.

Figura 51

Arquitectura Publicación/Suscripción de MQTT

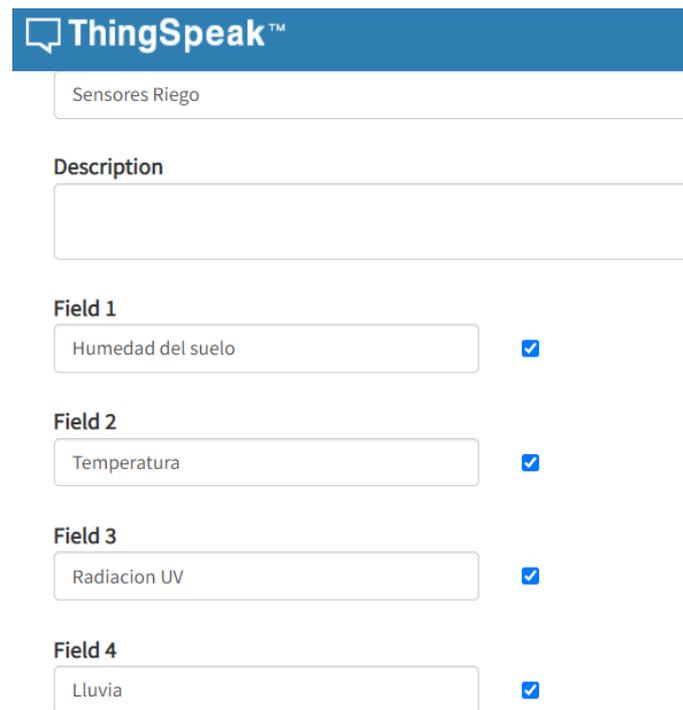


Fuente: Autoría

Se utiliza la API MQTT para publicar datos de temperatura, radiación UV, humedad del suelo y lluvia a un canal ThingSpeak. Para lo cual se crea un canal ThingSpeak, donde se detalla el nombre, una descripción y los campos a crear. Como se observa en la Figura 52, se crean cuatro campos correspondientes a cada sensor.

Figura 52

Creación de un canal en ThingSpeak



The image shows the 'Create Channel' page in ThingSpeak. At the top is the ThingSpeak logo. Below it is a text input field containing 'Sensores Riego'. Underneath is a 'Description' label followed by an empty text area. Below that are four 'Field' sections, each with a text input field and a checked checkbox to its right:

- Field 1**: Humedad del suelo
- Field 2**: Temperatura
- Field 3**: Radiacion UV
- Field 4**: Lluvia

Fuente: Autoría

Una vez guardado la configuración se obtiene el IDE del canal y la clave API de escritura y lectura para el canal privado. Esto se utiliza para escribir y leer datos hacia o desde el canal respectivamente.

Figura 53

IDE del canal y claves API de lectura y escritura



Fuente: Autoría

Para configurar la raspberry Pi y publicar datos en canal creado se utiliza la librería paho.MQTT. La configuración básica para utilizar el protocolo MQTT se detalla en la Figura 54.

Figura 54

Configuraciones básicas para utilizar el protocolo MQTT

```
import time
import serial
import requests
import paho.mqtt.publish as publish
import tensorflow as tf
import numpy as np

model=tf.keras.models.load_model("riego")
#model.summary()

# Datos Chanel
channelID="1632371"
ApiKey="REPNCZFDFA2FZCO"
topic="channels/" + str(channelID) + "/publish/" + str(ApiKey)
mqttHost="mqtt.thingspeak.com"
```

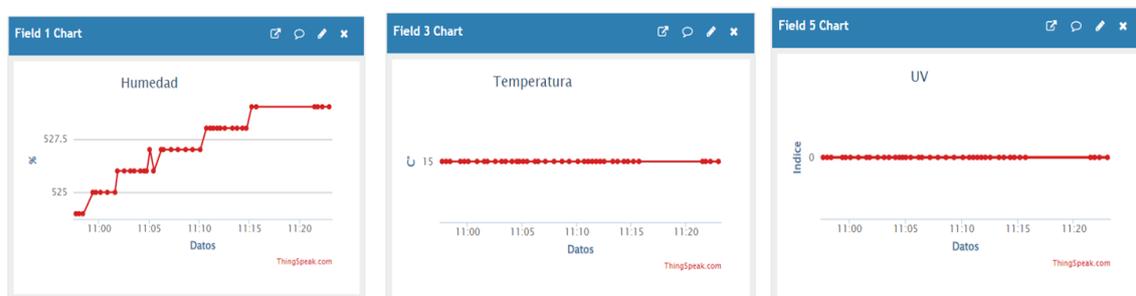
Fuente: Autoría

Es indispensable ingresar la ID del canal y la clave API de escritura del canal. También, se almacena la dirección del host MQTT como una cadena en una variable y se crea el topic para publicar datos.

Finalmente se asigna los valores de temperatura, humedad, radiación UV y lluvia a los campos creados en la plataforma y se podrán observar los valores de cada campo que se actualizan en el canal.

Figura 55

Visualización de datos publicados por el cliente MQTT en ThingSpeak



Fuente: Autoría

3.1.6. Pruebas del sistema: Primera Iteración

En esta etapa de proceder a realizar las pruebas de funcionamiento de la red WSN con tecnología LoRa, donde se verificarán los siguientes aspectos:

Test eléctrico del nodo sensor: Verificar si cumple con los requerimientos de corriente del nodo con el fin de obtener un funcionamiento correcto de los sensores, modulo LoRa y Arduino UNO.

Recolección de información del nodo sensor: Cada sensor debe leer valores de las variables ambientales (humedad del suelo, radiación UV, temperatura y lluvia) de forma correcta, se verifican en el monitor serial de Arduino UNO.

Recepción de datos en el nodo central: En el IDE Tonny de la Raspberry Pi 3 se corrobora la recepción de datos enviados por el nodo sensor mediante comunicación inalámbrica LoRa.

Presentación de datos en la plataforma ThingSpeak: Los datos enviados por el nodo sensor deben ser representados en forma de gráfica en la plataforma ThingSpeak.

Pruebas de comunicación Inalámbrica: Se debe verificar el alcance del módulo LoRa, se calcula el tiempo en el aire del paquete y se realiza el análisis de paquetes MQTT.

3.1.6.1. Test eléctrico del nodo sensor

Las pruebas del Test eléctrico se inician verificando el encendido de todos los dispositivos mediante el indicador led, los sensores que no cuentan con este indicador se realizó la mediación de voltaje en sus terminales. Se obtuvo un resultado satisfactorio.

Figura 56

Nodo sensor alimentado con una batería Lito



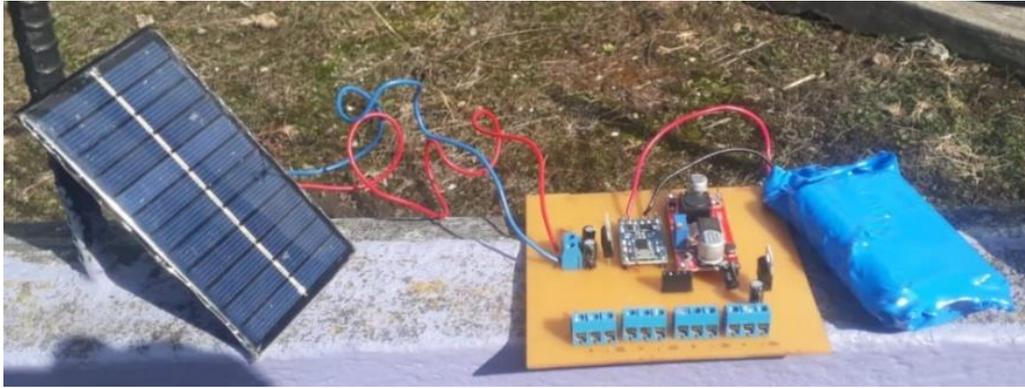
Fuente: Autoría

El voltaje suministrado es adecuado, por lo cual, se procede al uso de la batería de litio de 3.7 voltios a 6000 mA para la alimentación del nodo sensor. También se

verifico el funcionamiento del panel solar para recargar la batería de litio con el indicador led que posee el módulo de carga TP4056.

Figura 57

Nodo sensor alimentado con una batería de litio y panel solar



Fuente: *Autoría*

La duración de la batería depende del tiempo que el nodo esté en funcionamiento; mientras más tiempo este activo, más energía consume y la descarga será mucho más rápido. El panel solar será el encargado de recargar la batería con energía limpia.

Las pruebas de funcionamiento de la placa de alimentación se realizaron de manera experimental, la primera prueba se realiza cuando el nodo sensor está en funcionamiento y el panel solar no está conectada y la segunda cuando el nodo sensor está en funcionamiento y el panel solar está conectado a la placa de alimentación; los resultados se dan a conocer en la Tabla 23 donde se calcula la duración de la batería para ambos casos.

Tabla 23*Test de alimentación del nodo sensor*

Nº Prueba	Hora de Inicio	Hora Fin	Tiempo de duración
1	16:00	21:00	29 horas
2	08:00	18:00	34 horas

Fuente: *Autoría*

Los resultados obtenidos en la primera prueba es una duración de 29 horas de funcionamiento, demostrando que durante ese tiempo el sistema se encuentra operativo y que la batería es capaz de compensar ese tiempo; en la prueba 2 se evidencia que el sistema funciona durante 34 horas lo que indica que el panel solar alarga el tiempo de funcionamiento del nodo sensor.

3.1.6.2. Recolección de Información del Nodo Sensor

El nodo sensor es el encargado de medir las variables ambientales: humedad del suelo, radiación UV, temperatura y lluvia en el cultivo de pasto. Posterior a eso, los datos se envían a través de la red inalámbrica LoRa a la puerta de enlace ubicado en el nodo central.

Se han establecido una escala para cada una de las variables ambientales para identificar a que corresponde los valores que lee cada uno de los sensores, tal como se muestra en la Tabla 24 la variable de radiación UV y temperatura consta de 5 niveles, la variable de humedad del suelo tiene 4 niveles y finalmente la variable de lluvia consta de dos niveles.

Tabla 24*Rangos establecidos para las variables ambientales*

RADIACIÓN UV	Bajo	<2
	Moderado	De 3 a 5
	Alto	De 6 a 7
	Muy alto	De 8 a 10
	Extremadamente alto	11
TEMPERATURA	Muy Frio	De 8 a 10
	Frio	De 11 a 15
	Templado	De 16 a 25
	Cálido	De 26 a 36
	Muy cálido	De 37 a 42
HUMEDAD DEL SUELO	Muy húmedo	De 200 a 298
	Húmedo	De 299 a 349
	Seco	De 350 a 449
	Muy Seco	De 450 a 550
LLUVIA	Llueve	De 0 a 500
	No llueve	De 501 a 1023

Fuente: *Autoría*

Con la escala establecida se realizaron varias pruebas en diferentes días y horarios, en la Figura 58 se puede identificar los datos tomados por el nodo sensor en horas de la mañana; en el transcurso de una hora se tiene valores de humedad del suelo en un rango de 283 a 285 que corresponde a MUY HUMEDO de acuerdo a la escala establecida, la radiación UV se mantiene en 0 es decir, una radiación baja, en cuanto a la temperatura mantuvo valores de 16 y 17 que según la escala corresponde a TEMPLADO, finalmente, como no hay presencia de lluvia el sensor lee valores de 1006 y 1007.

Figura 58

Datos recolectados por el nodo sensor en horas de la mañana

```
Serial Monitor x
Message (Ctrl+Enter to send message)
1 UV=0 H=284 T=17 ll=1007
1 UV=0 H=284 T=16 ll=1007
1 UV=0 H=285 T=16 ll=1007
1 UV=0 H=284 T=16 ll=1007
1 UV=0 H=284 T=16 ll=1006
1 UV=0 H=284 T=16 ll=1007
1 UV=0 H=284 T=16 ll=1006
1 UV=0 H=284 T=16 ll=1006
1 UV=0 H=283 T=16 ll=1006
1 UV=0 H=284 T=16 ll=1006
1 UV=0 H=283 T=16 ll=1006
1 UV=0 H=283 T=16 ll=1006
1 UV=0 H=283 T=16 ll=1007
1 UV=0 H=284 T=26 ll=1006
1 UV=0 H=283 T=17 ll=1006
1 UV=0 H=284 T=16 ll=1006
1 UV=0 H=284 T=16 ll=1006
1 UV=0 H=283 T=16 ll=1006
1 UV=0 H=283 T=16 ll=1006
1 UV=0 H=283 T=16 ll=1006
1 UV=0 H=284 T=17 ll=1006
1 UV=0 H=283 T=17 ll=1006
```

Nota: UV hace referencia a radiación UV, H= humedad del suelo, T= temperatura y ll = lluvia.

Fuente: Autoría

En horas de la tarde se pueden observar como la escala de los diferentes sensores varían, el valor más alto de la radiación es 5 correspondiente a MODERADO, la humedad del suelo toma valores de 312 a 316 equivalente a HUMEDO, en cuanto a la temperatura varia de 13 (frio) a 17 (templado) grados y los valores del sensor de lluvia permanecen en un rango de 104 a 106, es decir que no hay presencia de lluvia.

Figura 59

Datos recolectados por el nodo sensor en horas de la tarde

```
Serial Monitor x
Message (Ctrl+Enter to send message to
. UV=1 H=312 T=13 ll=1005
. UV=1 H=312 T=13 ll=1004
. UV=2 H=312 T=30 ll=1005
. UV=2 H=312 T=27 ll=1005
. UV=2 H=312 T=31 ll=1005
. UV=0 H=312 T=17 ll=1005
. UV=2 H=313 T=32 ll=1005
. UV=3 H=312 T=36 ll=1005
. UV=5 H=313 T=20 ll=1005
. UV=4 H=314 T=27 ll=1005
. UV=3 H=314 T=37 ll=1005
. UV=4 H=314 T=28 ll=1006
. UV=4 H=314 T=27 ll=1005
. UV=4 H=313 T=27 ll=1005
. UV=4 H=313 T=27 ll=1005
. UV=4 H=314 T=27 ll=1005
. UV=5 H=313 T=20 ll=1006
. UV=5 H=313 T=20 ll=1005
. UV=4 H=314 T=28 ll=1006
. UV=3 H=315 T=37 ll=1005
. UV=3 H=315 T=36 ll=1006
. UV=3 H=316 T=36 ll=1006
. UV=3 H=316 T=37 ll=1006
. UV=3 H=316 T=36 ll=1006
. UV=3 H=316 T=36 ll=1006
```

Nota: UV hace referencia a radiación UV, H= humedad del suelo, T= temperatura y ll = lluvia.

Fuente: Autoría

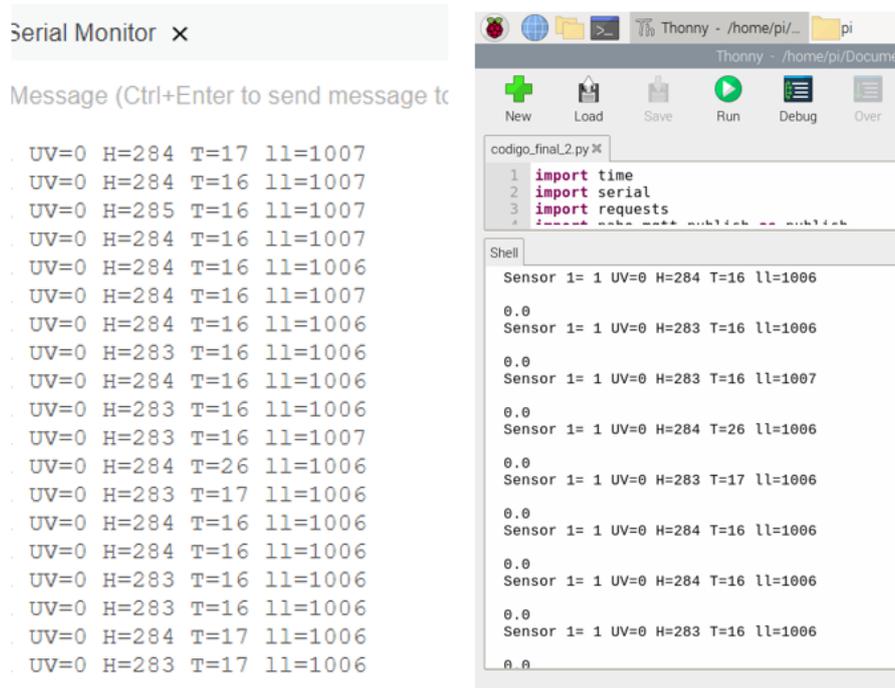
De esta manera se puede comprobar que el nodo sensor está funcionando correctamente, por una parte, cada sensor realiza su mediación y por otra, la mediación realizada por los sensores es correcta, los datos recolectados por el nodo sensor cumple la escala establecida y se comprueba visiblemente en campo.

3.1.6.3. Recepción de datos en el nodo central

Una vez recolectado los datos es necesario que el nodo sensor envíe los datos al nodo central, donde se encuentra la puerta de enlace; este proceso se lo realiza por medio de comunicación inalámbrica utilizando el módulo LoRa que trabaja con una frecuencia de 915 MHz.

Figura 60

Envío y recepción de datos Nodo Sensor-Puerta de enlace



a) Monitor serial Arduino Uno

b) Shell IDE Tonny Raspberry Pi 3

Nodo Sensor

Puerta de enlace

Fuente: *Autoría*

En la Figura 60 se puede observar el literal **a)** el puerto de comunicación serial de Arduino UNO que corresponde al nodo sensor y el literal **b)** la línea de comandos Shell de Thonny correspondiente a la puerta de enlace. El proceso de envío y recepción de datos inicia con la puerta de enlace comprobando la frecuencia de trabajo con el nodo sensor, en este caso 915MHz, el nodo sensor inicia el proceso de transmisión y una vez establecida la conexión se puede evidenciar los datos enviados por el nodo sensor a la puerta de enlace, logrando una comunicación exitosa.

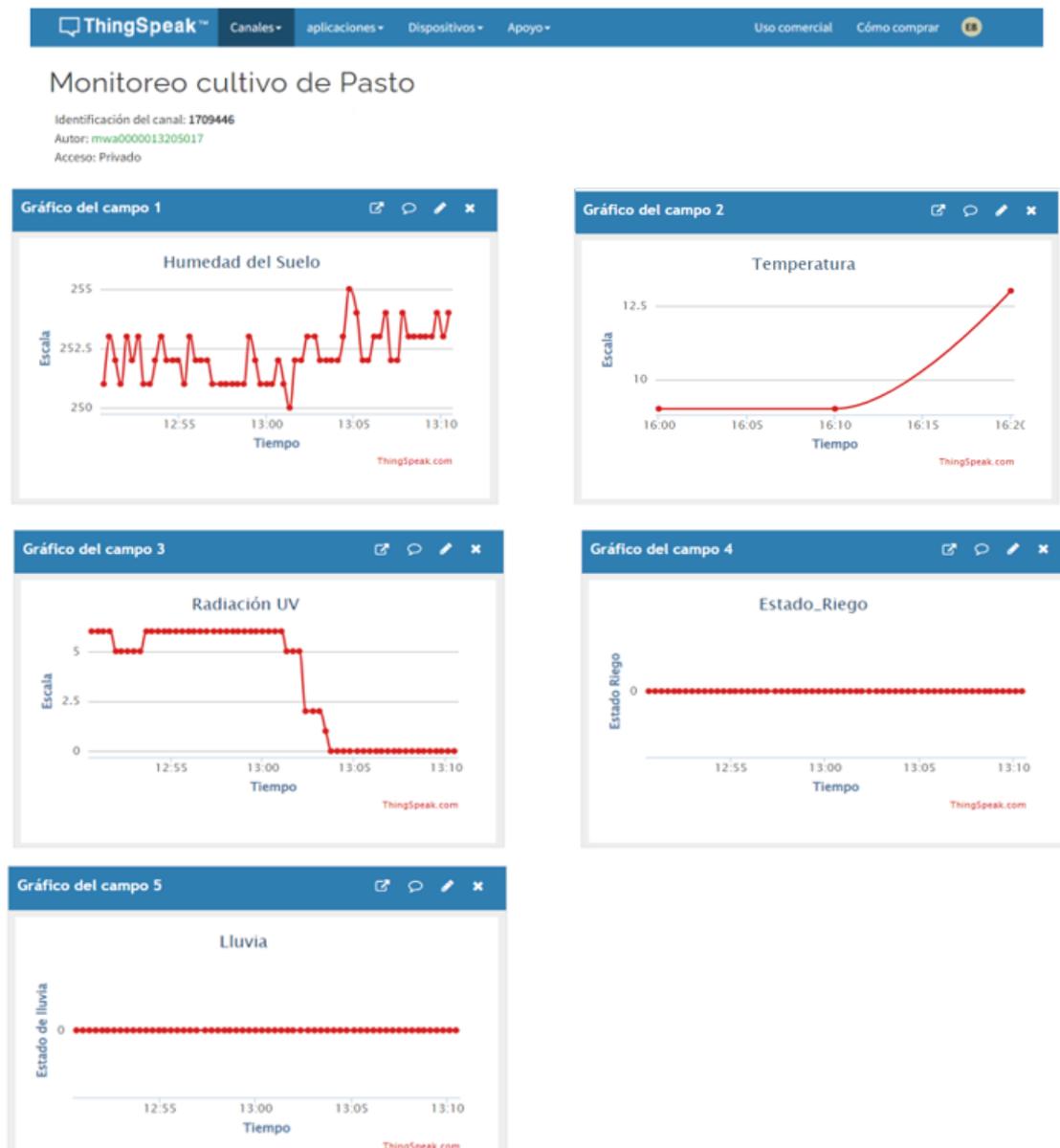
3.1.6.4. Presentación de datos en la plataforma ThingSpeak

Los datos recibidos por la puerta de enlace se publican en la plataforma ThingSpeak a través de internet mediante el protocolo MQTT. En la Figura 61 se puede apreciar el resultado de la comunicación entre el nodo sensor y la puerta de enlace

presentando gráficamente los datos de las variables ambientales. Se puede identificar en cada gráfica que el eje de las X corresponde a la escala predefinida y el eje de las Y corresponde al tiempo.

Figura 61

Presentación de datos en la plataforma ThingSpeak



Fuente: Autoría

3.1.6.5. Pruebas de comunicación Inalámbrica

La tecnología LoRa permite lograr comunicaciones de largo alcance, para comprobar el alcance de la red WSN se calcula la distancia en Google Earth entre la

puerta de enlace y el punto más lejano que se puede ubicar el nodo sensor, teniendo como resultado una distancia estimada de 205,17 metros.

Figura 62

Distancia entre la puerta de enlace y el nodo sensor, punto más lejano.

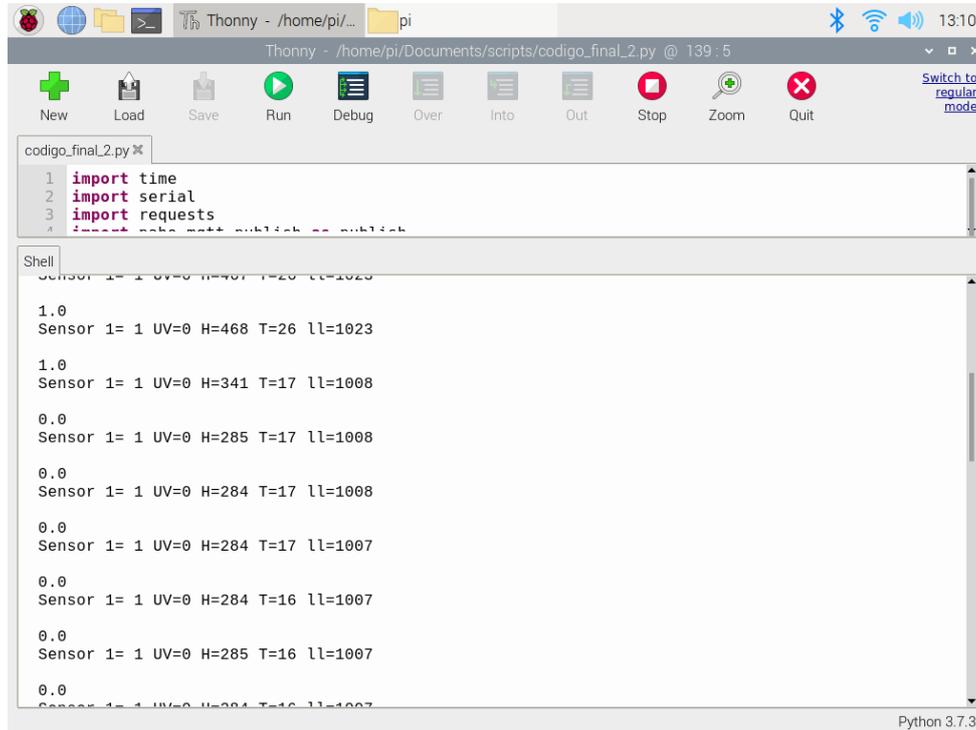


Fuente: Autoría

Se coloca el nodo sensor manualmente en la extensión de terreno en el lugar más lejano que se podría ubicar y se procede a la recolección de información, verificando que los datos lleguen a la puerta de enlace. Estos datos se pueden apreciar en la Figura 63.

Figura 63

Datos recolectados a una distancia de 205.17 metros



Fuente: Autoría

Se puede comprobar que la recepción de datos en la puerta de enlace a una distancia de 205,17 metros es exitosa.

3.1.6.6. Tiempo en el Aire de la transmisión

Cuando un nodo sensor envía una señal, necesita cierto tiempo antes de que el nodo receptor (puerta de enlace) reciba esta señal. Por ello se realiza el cálculo del tiempo en el aire, es decir, la duración de la transmisión de un paquete.

Para una combinación determinada de factor de propagación (SF), tasa de codificación (CR), ancho de banda de la señal (BW), el tiempo en el aire de un paquete LoRa se puede calcular a partir de la Ec. 1 (Semtech, 2020)

$$ToA = T_{preambulo} + T_{payload}$$

Donde el tiempo de preámbulo se calcula con la Ec. 17, $n_{preambulo}$ es la longitud del preámbulo, según la hoja de datos de Semtech tiene un registro programado de 8 para transceptores SX1276/77/78/79.

$$T_{preambulo} = (n_{preambulo} + 4.25) \times T_{simbolo} \quad \text{Ec. 17}$$

El tiempo del símbolo se calcula con la Ec. 18, donde SF es el factor de propagación y BW el ancho de banda.

$$T_{simbolo} = \frac{2^{SF}}{BW} \quad \text{Ec. 18}$$

Para calcular el tiempo de símbolo se reemplaza los valores establecidos, en este caso el factor de propagación es SF7 y el ancho de banda es 125 KHz.

$$T_{simbolo} = \frac{2^7}{125}$$

$$T_{simbolo} = 1.024$$

Reemplazando este valor en la Ec. 18 se obtiene que el tiempo del preámbulo es de 12.54 ms.

$$T_{preambulo} = (8 + 4.25) \times 1.024$$

$$T_{preambulo} = 12.544$$

Para calcular el tiempo del payload (carga útil) es necesario conocer la longitud de símbolos en la carga útil para ello se utiliza la Ec. 19.

$$N_{payload} = 8 + \left(\left[\frac{(8PL - 4SF + 28 + 16CRC - 20IH)}{4(SF - 2DE)} \right] (CR + 4) \right) \quad \text{Ec. 19}$$

Donde:

IH: cabecera (Header), cuando está habilitada el valor IH es igual a 0 caso contrario es 1.

DE: Optimización de tasa de datos baja (Low Data Rate Optimize), valor de DE es 1 cuando está habilitada y el valor DE es igual a 0 para el otro caso.

PL indica el tamaño de la carga útil en bytes

SF indica el factor de propagación

CRC indica la comprobación de redundancia cíclica utilizada para la detección de errores, cuando este habilitado toma el valor de 1, caso contrario 0, por defecto esta deshabilitado.

CR indica la tasa de codificación, puede estar en el rango de 1 a 4 (1 correspondiente a 4/5, 4 a 4/8).

Reemplazando este valor en la Ec. 19, se tiene un valor de 58 símbolos en la carga útil.

$$N_{payload} = 8 + \max\left(\text{ceil}\left[\frac{(8(35) - 4(7) + 28 + 16(0) - 20(0))}{4(7 - 2(0))}\right], (1 + 4), 0\right)$$

$$N_{payload} = 58 \text{ simbolos}$$

Una vez obtenido este valor se puede calcular la duración de la carga útil con la Ec. 20.

$$T_{payload} = N_{payload} \times T_{simbolo} \quad \text{Ec. 20}$$

Reemplazando los valores se consigue un tiempo de payload de 59.392 ms

$$T_{payload} = 58 \times 1.024$$

$$T_{payload} = 59.392 \text{ ms}$$

Finalmente, se obtiene el valor del tiempo en el aire del paquete reemplazando la Ec. 1.

$$T_{oA} = T_{preambulo} + T_{payload}$$

$$T_{oA} = 12.54 + 59.392$$

$$T_{oA} = 71.932 \text{ ms}$$

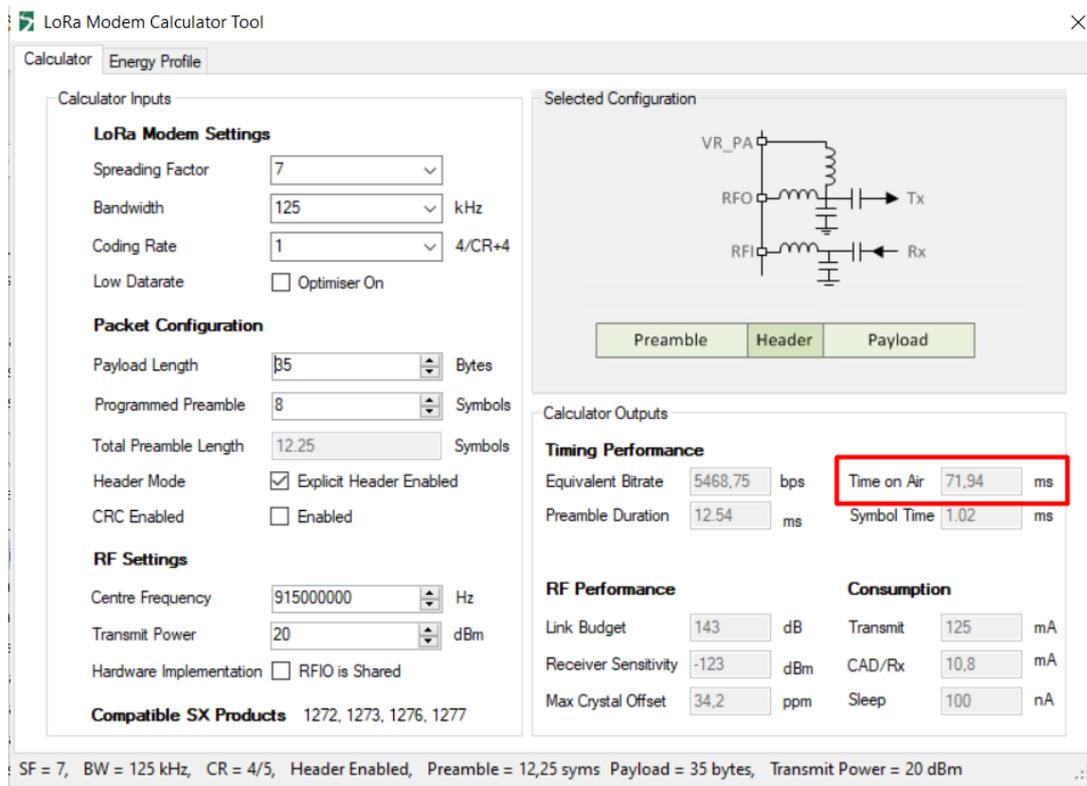
Existe una calculadora online para el cálculo del tiempo en el aire del paquete LoRa para módems Semtech compatibles con productos SX 1272, 1273, 1276 y 1277 en el cual se ingresan los parámetros establecidos para la transmisión y se obtiene a su

salida el rendimiento de duración como tiempo de símbolo, tiempo en el aire, duración del preámbulo y la tasa de bits equivalente.

Como se observa en la Figura 64 el tiempo en el aire es de 71.94 ms, una diferencia de 0.002 ms con los cálculos realizados.

Figura 64

Calculadora LoRa para el cálculo de tiempo en el aire



Fuente: Autoría

Además, en la Tabla 25 se puede observar como el tiempo en el aire del paquete incrementa significativamente mientras la sensibilidad del receptor disminuye al aumentar el factor de propagación.

Tabla 25*Tiempo en el aire y sensibilidad del receptor al variar el SF*

Factor de Propagación SF	Tiempo en el aire	Sensibilidad del receptor
SF 7	71.94 ms	-123 dBm
SF 8	133,63 ms	-126 dBm
SF 9	246,78 ms	-129 dBm
SF 10	452,61 ms	-132 dBm
SF 11	823,3 ms	-134,5 dBm
SF12	1646,59 ms	-137 dBm

Fuente: *Autoría*

3.1.6.7. Análisis Espectral de la transmisión LoRa

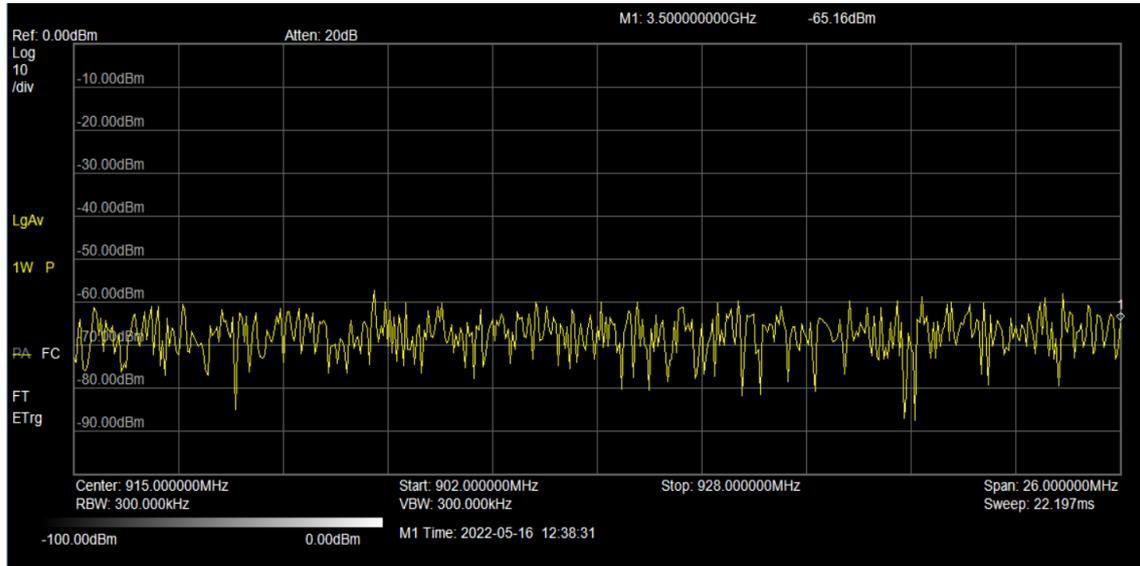
El análisis espectral se refiere a la descomposición de señales en componentes sinusoidales, cada componente oscilando a una frecuencia y amplitud determinada. La representación obtenida mediante el análisis espectral se denomina espectrograma.

El analizador de espectro que se utiliza es el Keysinght N9322C, este analizador trabaja con un rango de frecuencias de 9Khz a 7Ghz. Entre sus funciones incluye el monitoreo de espectro el cual permite grabar y reproducir los rastros de la señal capturados.

En la Figura 65 se establece la zona del espectro que se va a analizar, se ingresa la frecuencia en la que trabaja los nodos LoRa; en este caso, una frecuencia central de 915 MHz, una frecuencia inicial de 902MHz y una frecuencia final o de parada de 928 MHz.

Figura 65

Zona espectral con frecuencia central de 915 MHz

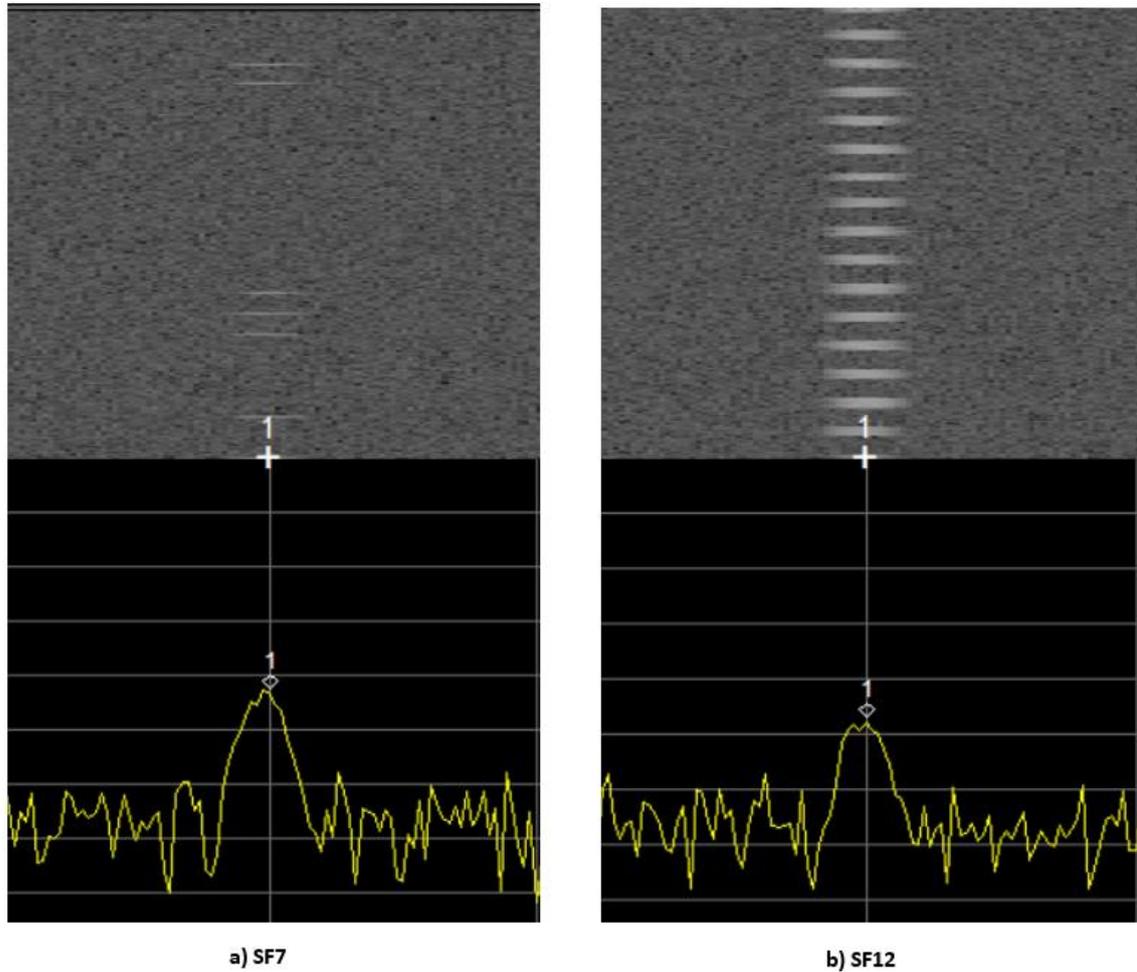


Fuente: *Autoría*

Para poder apreciar el espectro de la señal en una transmisión LoRa con diferente factor de propagación se hace uso del espectrograma y se varía el factor de propagación de S7 a S12 en los nodos LoRa. Los resultados obtenidos en el analizador de espectro se pueden apreciar en la Figura 66.

Figura 66

Espectro de la señal al variar el factor de propagación de SF7 a SF



Nota. En la mitad superior de cada imagen se tiene los espectrogramas y en la mitad inferior la presentación convencional del espectro de la señal. **Fuente:** Autoría

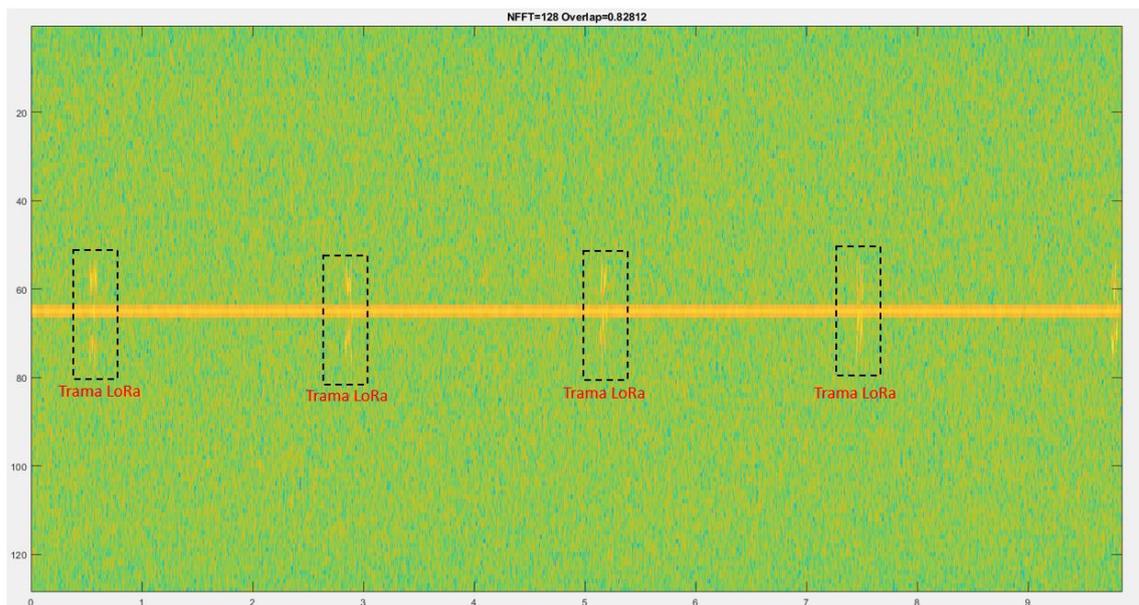
Cada espectro consiste en una línea horizontal coloreada según las distintas amplitudes de las señales. Cada nuevo espectro que se analiza y cada nuevo espectro que es adquirido empuja hacia arriba a todos los anteriores, de modo que los más antiguos se pierden por la parte superior del diagrama. De este modo se puede entender que el eje horizontal representa la frecuencia de la zona del espectro que se está analizando y el eje vertical representa el tiempo, verificando así la evolución del espectro a lo largo del tiempo.

En la Figura 66 sección **a)** se puede observar el espectro de una señal que utiliza un factor de propagación SF7, como se puede evidenciar en la Tabla 24, el tiempo en el aire del paquete es 71.94 ms, como este tiempo es muy corto se puede distinguir una señal muy leve, a diferencia del espectro de la señal que utiliza un factor de propagación SF12, como el tiempo en el aire de esta señal es 1646,59 ms un tiempo mayor, se puede distinguir una señal mucha más prolongada.

Este mismo análisis se realiza a través de la herramienta Matlab utilizando la transformada de Fourier, en este caso, el eje horizontal corresponde al tiempo y el eje vertical a la frecuencia, de esta forma se puede obtener el tiempo en el aire del paquete de los nodos LoRa, verificando que los cálculos matemáticos realizados son correctos. En la Figura 67 se puede observar los paquetes transmitidos desde el nodo sensor al nodo central en una frecuencia de 915 MHz.

Figura 67

Paquetes transmitidos del nodo sensor al nodo central con una frecuencia de 915 MHz



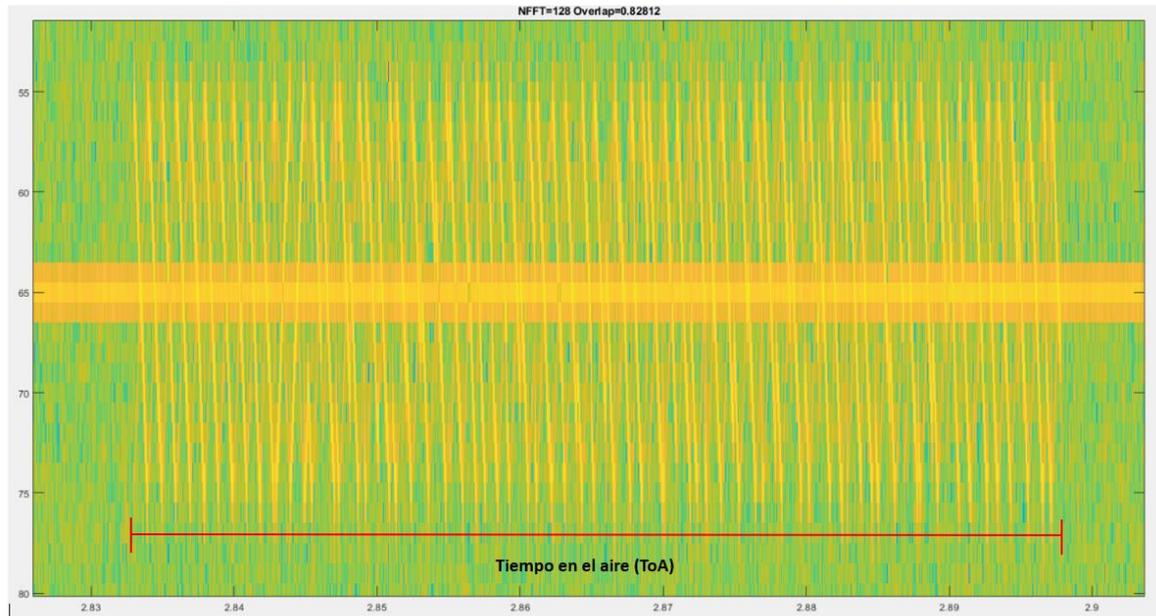
Fuente: Autoría

Una vista mucho más amplia de la trama LoRa se puede observar en la Figura 68, en esta señal se puede calcular la duración del paquete LoRa con un factor de

propagación de SF7, un ancho de banda de 125 kHz, una tasa de codificación de errores de 4/5 y una carga útil (payload) de 35 bytes.

Figura 68

Señal LoRa en una frecuencia de 915 MHz

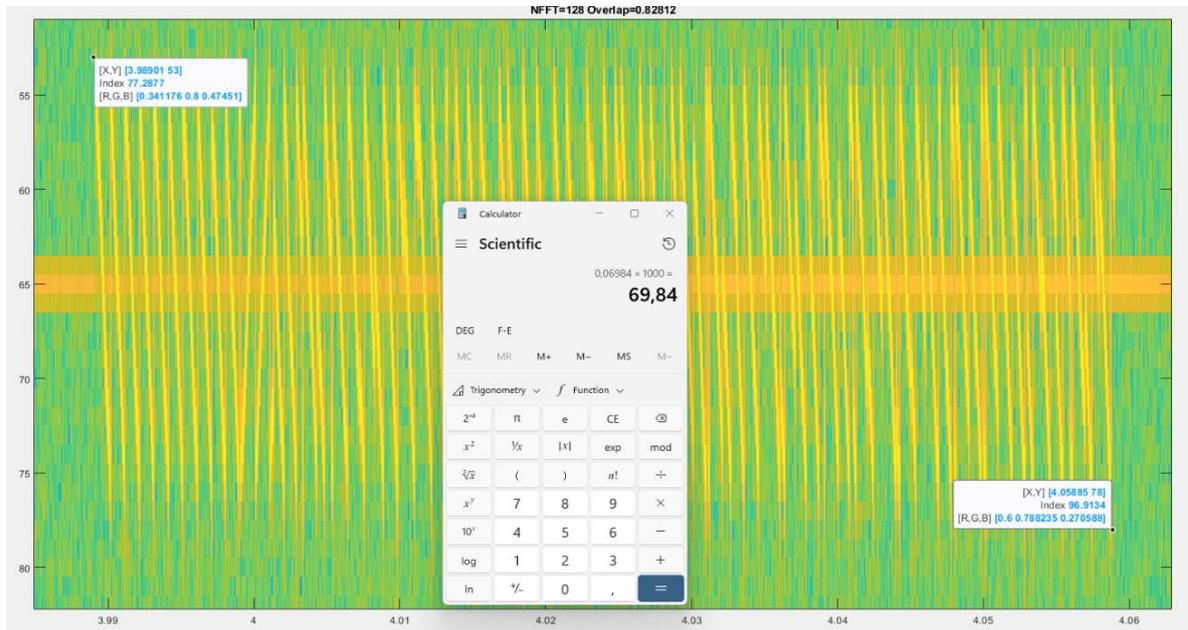


Fuente: *Autoría*

Al tomar los valores de inicio y fin de la señal se puede calcular que la duración del mismo es de 69.84 ms, es decir que el tiempo en el aire de la transmisión LoRa es aproximadamente 70 ms, una diferencia de 2.1 ms con los cálculos realizados.

Figura 69

Tiempo en el aire de un paquete LoRa utilizando la herramienta Matlab



Fuente: Autoría

Con el análisis del factor de propagación se puede comprender el porqué de la elección del mismo. El SF determina cuantos chirridos se envían por segundo, al utilizar un SF bajo (SF7) significa que se envían más chirridos por segundo; por lo tanto, se puede codificar más datos por segundos. A diferencia, al utilizar un SF alto (SF12) implica menos chirridos por segundo; por lo tanto, hay menos datos para codificar por segundo.

En cuanto a la duración de la transmisión, al enviar la misma cantidad de datos un SF alto necesita más tiempo de transmisión (tiempo en el aire), esto quiere decir que el módulo LoRa está activo, funcionando por más tiempo y consumiendo más energía.

Una ventaja de un SF alto es que un tiempo en el aire más prolongado brinda al receptor una mejor sensibilidad, esto quiere decir que puede recibir la señal mucho más lejos, obteniendo una mejor cobertura.

3.1.6.8. Análisis del protocolo MQTT

Una vez que el cliente completa el protocolo de enlace de tres vías de TCP, se inicia con los mensajes que se intercambian en el protocolo MQTT.

Antes de que el cliente pueda publicar y suscribirse a mensajes, primero debe conectarse al broker. Esta conexión no es solo para establecer una conexión TCP, sino también para establecer una conexión en la capa de aplicación. El proceso de establecimiento de conexión MQTT es el siguiente:

El cliente envía un paquete CONNECT (conexión) al broker. Como se observa en la Figura 70 en este campo se puede identificar la versión del protocolo MQTT y *User name Flag*, el cual permite conocer si el paquete cuenta con identificación por nombre.

Figura 70

Mensajes del protocolo MQTT- Connect Command

The screenshot shows a Wireshark capture of a MQTT Connect Command. The packet list pane highlights the MQTT packet (No. 64) with a red box around the protocol and '80 Connect Command'. The packet details pane shows the following information:

- MQ Telemetry Transport Protocol, Connect Command
- Header Flags: 0x10, Message Type: Connect Command
- 0001 = Message Type: Connect Command (1)
- 0000 = Reserved: 0
- Msg Len: 12
- Protocol Name Length: 4
- Protocol Name: MQTT
- Version: MQTT v3.1.1 (4)
- Connect Flags: 0x02, QoS Level: At most once delivery (Fire and Forget), Clean Session Flag
- 0... = User Name Flag: Not set
- .0. = Password Flag: Not set
- ..0. = Will Retain: Not set
- ...0. = QoS Level: At most once delivery (Fire and Forget) (0)
-0. = Will Flag: Not set
-1. = Clean Session Flag: Set
-0 = (Reserved): Not set
- Keep Alive: 60
- Client ID Length: 0
- Client ID:

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```
0000 60 32 b1 13 b3 6c b8 27 eb 11 ec 3c 08 00 45 00 2. .1. . . . .  
0010 00 42 2e b7 40 00 40 06 5b 4f c0 a8 00 6a 2c c2 .B. @ @ [0. .j.  
0020 c2 db 97 5f 07 5b 3a 75 47 7e 5a a8 d0 51 80 18 . . . . [ : u G-Z . Q .  
0030 01 f6 b2 02 00 00 01 01 08 0a 37 2f ba 09 5b c0 . . . . . / / . [ .
```

Fuente: Autoría

Una vez que el broker recibe el paquete CONNECT, si se le permite el acceso responderá con un paquete CONNECT ACK (acuse de recibo). Este contiene códigos de retorno que especifica el estado de éxito (retorno de 0) o error (retorno distinto de 0) de la transmisión. En la Figura 71 se puede observar que el código de retorno es 0, es decir estado de la transmisión es exitosa.

Figura 71

Mensaje del protocolo MQTT- Connect ACK

The screenshot displays a network traffic capture in Wireshark. The top pane shows a list of captured packets. Packet 66 is highlighted, showing it is an MQTT message of type '70 Connect Ack' with a length of 70 bytes. The middle pane provides a detailed view of this packet, showing it is a 'MQ Telemetry Transport Protocol, Connect Ack' with a 'Return Code: Connection Accepted (0)'. The bottom pane shows the raw hexadecimal and ASCII data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
60	18.001395794	192.168.0.106	44.194.194.219	TCP	74	38751 → 1883 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=925874593 TSecr=0
62	18.105011540	44.194.194.219	192.168.0.106	TCP	74	1883 → 38751 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0 MSS=1460 SACK_PERM=1 TSval=153935935
63	18.105132448	192.168.0.106	44.194.194.219	TCP	66	38751 → 1883 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=925874697 TSecr=1539359593
64	18.105422295	192.168.0.106	44.194.194.219	MQTT	80	Connect Command
65	18.209534958	44.194.194.219	192.168.0.106	TCP	66	1883 → 38751 [ACK] Seq=1 Ack=15 Win=26880 Len=0 TSval=1539359697 TSecr=925874697
66	18.214516270	44.194.194.219	192.168.0.106	MQTT	70	Connect Ack
67	18.214575968	192.168.0.106	44.194.194.219	TCP	66	38751 → 1883 [ACK] Seq=15 Ack=5 Win=64256 Len=0 TSval=925874807 TSecr=1539359705
68	18.215240939	192.168.0.106	44.194.194.219	MQTT	140	Publish Message [channels/1632371/publish/REPNCZFDFA2FZCO]
69	18.215466450	192.168.0.106	44.194.194.219	MQTT	68	Disconnect Req
71	18.321443947	44.194.194.219	192.168.0.106	TCP	66	1883 → 38751 [ACK] Seq=5 Ack=92 Win=26880 Len=0 TSval=1539359809 TSecr=925874807
72	18.321639401	44.194.194.219	192.168.0.106	TCP	66	1883 → 38751 [FIN, ACK] Seq=5 Ack=92 Win=26880 Len=0 TSval=1539359809 TSecr=925874807

Packet Details for Frame 66:

- > Ethernet II, Src: Tp-LinkT_13:b3:6c (60:32:b1:13:b3:6c), Dst: Raspberr_11:ec:3c (b8:27:eb:11:ec:3c)
- > Internet Protocol Version 4, Src: 44.194.194.219, Dst: 192.168.0.106
- > Transmission Control Protocol, Src Port: 1883, Dst Port: 38751, Seq: 1, Ack: 15, Len: 4
- MQ Telemetry Transport Protocol, Connect Ack
 - Header Flags: 0x20, Message Type: Connect Ack
 - 0010 ... = Message Type: Connect Ack (2)
 - ... 0000 = Reserved: 0
 - Msg Len: 2
 - Acknowledge Flags: 0x00
 - 0000 000. = Reserved: Not set
 -0 = Session Present: Not set
 - Return Code: Connection Accepted (0)

Packet Bytes:

```

0000 b8 27 eb 11 ec 3c 60 32 b1 13 b3 6c 08 00 45 00  .....<2 ...1:
0010 00 38 25 95 40 00 d8 06 cc 7a 2c c2 d8 c0 a8  -8%:@...z,....
0020 00 6a 07 5b 97 5f 5a a8 d0 51 3a 75 47 8c 00 18  -j[_Z-Q:uG...
0030 00 69 49 0d 00 01 01 08 0a 5b c0 c3 d9 37 2f  -iI.....[...7/
0040 ba 09 20 02 00 00  .....
```

Fuente: Autoría

Una vez que la conexión es aceptada se puede publicar (PUBLISH) el mensaje, este puede ser desde el cliente hacia el broker o viceversa. En este campo se determina un parámetro importante que es el *Topic* donde se identifica el ID del canal de ThingSpeak.

Figura 72

Mensaje del protocolo MQTT- Publish

The screenshot shows a Wireshark capture of an MQTT Publish Message. The packet list pane displays the following information for packet 68:

No.	Time	Source	Destination	Protocol	Length	Info
68	18.215240939	192.168.0.106	44.194.194.219	MQTT	140	Publish Message [channels/1632371/publish/REPNCZFDFA2FZCO]

The packet details pane shows the following information for the MQTT Publish Message:

- MQ Telemetry Transport Protocol, Publish Message
- Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
- 0011 = Message Type: Publish Message (3)
- 0... = DUP Flag: Not set
-00. = QoS Level: At most once delivery (Fire and Forget) (0)
-0 = Retain: Not set
- Msg Len: 72
- Topic Length: 41
- Topic: channels/1632371/publish/REPNCZFDFA2FZCO
- Message: 6669656c64313d393430266669656c64333d3434266669656c64353d30

The packet bytes pane shows the raw data of the message:

```
0000 60 32 b1 13 b3 6c b8 27 eb 11 ec 3c 08 00 45 00 2...l'...<...E
0010 00 7e 2e b9 40 00 40 06 5b 11 c0 a8 00 6a 2c c2 ....@.@. [...],
0020 c2 db 97 5f 07 5b 3a 75 47 8c 5a a8 d0 55 80 18 ...[:u G-Z-U..
0030 01 f6 57 86 00 00 01 01 08 0a 37 2f ba 77 5b c0 ..W.....-7/-w[
0040 c3 d9 30 48 00 29 63 68 61 6e 6e 65 6c 73 2f 31 ..0H.)ch annels/1
0050 36 33 32 33 37 31 2f 70 75 62 6c 69 73 68 2f 52 632371/p ublish/R
0060 45 50 4e 43 41 5a 46 44 46 41 32 46 5a 43 4f 66 EPNCZFD FA2FZCOF
0070 69 65 6c 64 31 3d 39 34 30 26 66 69 65 6c 64 33 ield1=94 0&field3
0080 3d 34 34 26 66 69 65 6c 64 35 3d 30 =44&fiel d5=0
```

Fuente: Autoría

El último mensaje que se captura es DISCONNECT (desconexión) este mensaje es enviado desde el cliente al broker o viceversa informando el cierre de conexión.

Figura 73

Mensaje del protocolo MQTT- Disconnect

No.	Time	Source	Destination	Protocol	Length	Info
60	18.001395794	192.168.0.106	44.194.194.219	TCP	74	38751 → 1883 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=925874593 TSecr=0
62	18.105011540	44.194.194.219	192.168.0.106	TCP	74	1883 → 38751 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0 MSS=1460 SACK_PERM=1 TSval=1539359
63	18.105132448	192.168.0.106	44.194.194.219	TCP	66	38751 → 1883 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=925874697 TSecr=1539359593
64	18.105422295	192.168.0.106	44.194.194.219	MQTT	80	Connect Command
65	18.209534958	44.194.194.219	192.168.0.106	TCP	66	1883 → 38751 [ACK] Seq=1 Ack=15 Win=26880 Len=0 TSval=1539359697 TSecr=925874697
66	18.214516270	44.194.194.219	192.168.0.106	MQTT	70	Connect Ack
67	18.214575968	192.168.0.106	44.194.194.219	TCP	66	38751 → 1883 [ACK] Seq=15 Ack=5 Win=64256 Len=0 TSval=925874807 TSecr=1539359705
68	18.215240939	192.168.0.106	44.194.194.219	MQTT	140	Publish Message [channels/1632371/publish/REPINCAZFDFA2FZCO]
69	18.215466450	192.168.0.106	44.194.194.219	MQTT	68	Disconnect Req
71	18.321443947	44.194.194.219	192.168.0.106	TCP	66	1883 → 38751 [ACK] Seq=5 Ack=92 Win=26880 Len=0 TSval=1539359809 TSecr=925874807
72	18.321639401	44.194.194.219	192.168.0.106	TCP	66	1883 → 38751 [FIN, ACK] Seq=5 Ack=92 Win=26880 Len=0 TSval=1539359809 TSecr=925874807

> Frame 69: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface wlan0, id 0
> Ethernet II, Src: Raspberr_11:ec:3c (b8:27:eb:11:ec:3c), Dst: Tp-LinkT_13:b3:6c (60:32:b1:13:b3:6c)
> Internet Protocol Version 4, Src: 192.168.0.106, Dst: 44.194.194.219
> Transmission Control Protocol, Src Port: 38751, Dst Port: 1883, Seq: 89, Ack: 5, Len: 2
▼ MQ Telemetry Transport Protocol, Disconnect Req
 ▼ Header Flags: 0xe0, Message Type: Disconnect Req
 1110 = Message Type: Disconnect Req (14)
 0000 = Reserved: 0
 Msg Len: 0

```
0000 60 32 b1 13 b3 6c b8 27 eb 11 ec 3c 08 00 45 00  `2...].'.<...E.  
0010 00 36 2e ba 40 00 40 06 5b 58 c0 a8 00 6a 2c c2  .6.@.@ [X...],.  
0020 c2 db 97 5f 07 5b 3a 75 47 d6 5a a8 d0 55 80 19  ...[:u G-Z..U..  
0030 01 f6 86 c6 00 00 01 01 08 0a 37 2f ba 77 5b c0  .....-7/-w[.  
0040 c3 d9 e0 00                                     ....
```

Fuente: Autoría

Existen varios mensajes de protocolo MQTT que se pueden capturar y analizar como *subscribe*, *sub ack*, *unsubscribe*, entre otros. Pero, en este caso, el cliente simplemente publica datos de los sensores, por lo tanto, solo se pueden capturar los mensajes MQTT mencionados anteriormente.

3.1.7. Conclusiones: Primera Iteración

Los dispositivos seleccionados para el nodo sensor y el nodo central cumplen con los requerimientos establecidos por el sistema, por lo tanto, no hay cambios que realizar en cuanto a hardware y software en la primera iteración.

Mediante el test eléctrico se verifica que la placa de alimentación abastece de energía al nodo sensor aproximadamente 34 horas, en caso de ser necesario el módulo

de carga TP5046 tiene incorporado un puerto micro USB para recargar la batería con 5 voltios.

El envío de datos de las variables ambientales desde el nodo sensor al nodo central se realizó con éxito, se cumplen con los parámetros establecidos en la simulación de radio enlace para obtener línea de vista en la transmisión. En cuanto a los parámetros para la comunicación con tecnología LoRa se determinó un factor de propagación SF7, un ancho de banda de 125 KHz, una tasa de codificación de errores de 4/5 y una potencia de transmisión de 20dBm; con estos parámetros se tienen un tiempo en el aire de aproximadamente 70 ms.

Los datos de las variables ambientales enviados por el nodo sensor se publicaron a través del protocolo MQTT en la plataforma ThingSpeak donde se representan a través de gráficas. Debido a que se utiliza una cuenta de usuario gratuita las actualizaciones son cada 15 segundos.

3.2. Modelo Iterativo: Segunda Iteración

La segunda iteración del sistema es el entrenamiento de la red neuronal para la predicción del riego en el cultivo de pasto. El nodo central será el encargado de enviar la orden al nodo actuador para activar o desactivar la electroválvula.

3.2.1. Requerimientos del Sistema

En la primera etapa del modelo iterativo están los requerimientos, estos se establecieron de forma general en la primera interacción y no se sugiere ningún cambio en la segunda iteración. En cuanto, a la elección de hardware y software se describe a continuación:

La red neuronal es ejecutada en la raspberry Pi seleccionado previamente en la primera interacción. Ahora bien, en la Tabla 26 se realiza una comparativa del software de programación para el diseño de la red neuronal artificial.

Tabla 26

Selección del software de programación para la red neuronal artificial

 Python	 Matlab
CARACTERÍSTICAS	
Plataforma de código abierto y gratuito.	No es gratuito ni de código abierto
Grandes bibliotecas para aprendizaje automático e inteligencia artificial.	Posee gran cantidad de bibliotecas
Tiene una gran comunidad de usuarios en GitHub, Stackoverflow, etc.	No tienen paquetes disponibles para inteligencia artificial.
Usa lenguaje orientado a objetos	Lenguaje no orientado a objetos
Sintaxis fácil de aprender y usar	Usa herramientas Simulink
Requiere de experiencia	Es fácil de aprender

Fuente: Autoría

El software de programación que se elige para la implementación de la red neuronal artificial es Python. Este software es el más utilizado para el desarrollo de proyectos basados en inteligencia artificial, tiene gran cantidad de librerías que ayudan en el desarrollo de proyectos, posee licencia libre y al ser ejecutado en la Raspberry Pi tienen menor carga computacional.

Por otra parte, el nodo actuador está compuesto por un Arduino UNO, un transceptor LoRa seleccionados previamente en la Tabla 8 y Tabla 9, un módulo relé y una electroválvula. El tipo de electroválvula a utilizar se da a conocer en la Tabla 27.

Tabla 27

Elección de electroválvula solenoide para el nodo actuador

CARACTERÍSTICAS	MODELOS	
	AQT15SP-J	ZE-4F180
		
Diámetro entrada/salida	½	½
Flujo	5 l/min	3 l/m
Presión de trabajo	0.04-0.8Mpa	0.02-0.8 Mpa
Tiempo de respuesta (open)	0.15	0.15 seg
Tiempo de respuesta (close)	0.3	0.3 seg
Voltaje de operación	12V	12V
Vida Útil	>1millon de ciclos	50millones de ciclos
Temperatura de trabajo	75 °C	75 °C
Dimensiones	90x50x60mm	85x56x50 mm
Precio	\$12	\$15

Fuente: Autoría

Como se observa en la Tabla 26, las características que posee cada modelo son bastantes similares, por ende, se selecciona el modelo AQT15SP-J, por el precio, tamaño y disponibilidad en el mercado nacional.

Para finalizar, es necesario un relé que permita la apertura o cierre de la electroválvula mediante la placa Arduino, por ese motivo se realiza una comparación de módulos relés, tal como se muestra en la Tabla 28.

Tabla 28

Elección de módulos relé para el nodo actuador

CARACTERISITICAS	MODELOS	
	Modulo Relé 4 canales	Shield Relé de 4 canales
		
Numero de canales	4	4
Alimentación	5V	5V
Compatible con Arduino	SI	SI
Indicador LED	SI	SI
Pines de control	IN1, IN2, IN3, IN4	GPIO D4, D5, D6, D7
Corriente nominal	10A	3A
Dimensiones	6.8x4.9x1.6mm	15.7x11.4x10.4 mm
Precio	\$4	\$6

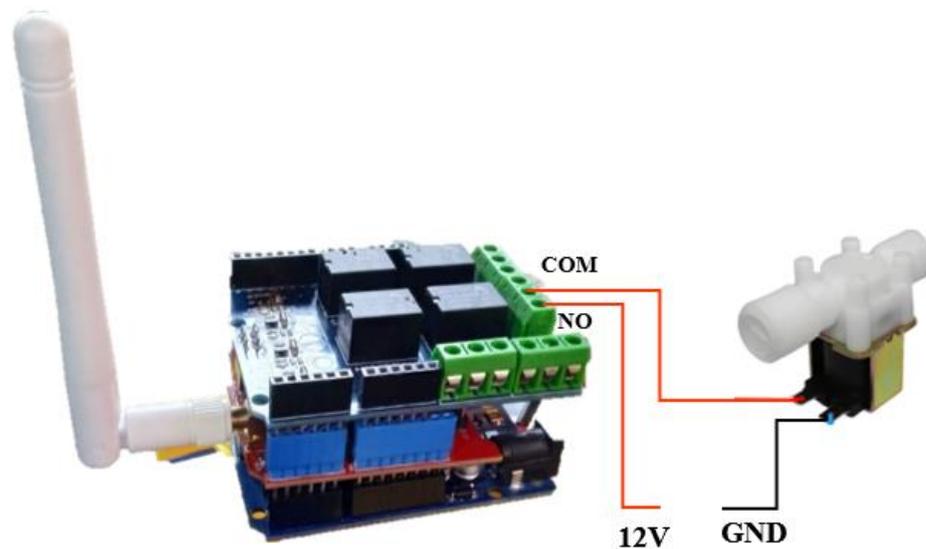
Fuente: *Autoría*

Se puede observar que los relés comparten características similares, pero se escoge el Shield Relé ya que monta directamente sobre la placa Arduino Uno mediante el acoplamiento de sus pines sin necesidad de alguna otra conexión externa.

En la Figura 74 se puede observar el diagrama esquemático del nodo actuador. Se tiene la placa Arduino Uno, montado sobre este el módulo de comunicación LoRa y sobre el mismo un Shield Relé de 4 canales. El relé tiene una línea activa que va al punto común (COM), como se quiere que inicialmente el circuito este abierto, la electroválvula está en su posición de reposo. Se utiliza el canal abierto (NO) para conectar el cable activo de la alimentación a la electroválvula.

Figura 74

Diagrama Esquemático del Nodo Actuador



Fuente: *Autoría*

3.2.2. Análisis de situación actual

Se cuenta con una red de sensores inalámbricos WSN con tecnología LoRa el cual permite el monitoreo de las variables ambientales que influyen el en desarrollo del cultivo de pasto, pudiendo ser visualizadas los datos en forma de gráficos en una plataforma IoT a través de internet y en una aplicación móvil llamada ThingView.

3.2.3. Diseño del Sistema Propuesto

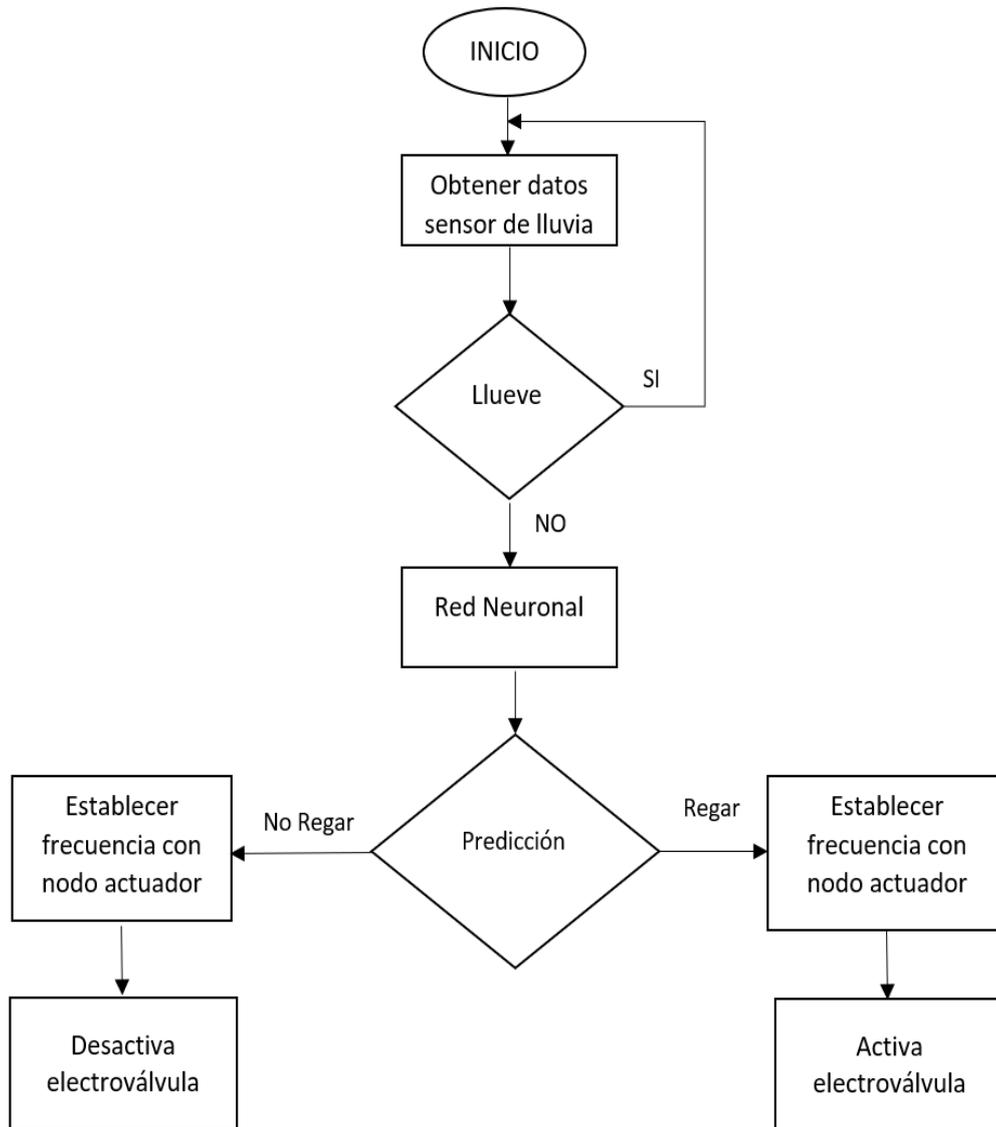
Los datos recolectados a través del nodo sensor en la primera iteración sirven como datos de entrenamiento para la red neuronal artificial el cual realizará predicciones de cuando el cultivo necesita se irrigado. El nodo central enviará la orden de cuando aplicar el riego hacia un nodo actuador, donde una electroválvula iniciará o detendrá el flujo de agua de riego hacia el cultivo.

El nodo sensor transmite datos de cuatro sensores al nodo central, estas son: humedad del suelo, temperatura, radiación UV y lluvia. Se toma los datos del sensor de lluvia como referencia para aplicar el algoritmo de la red neuronal, ya que, si el valor detectado por el sensor corresponde a lluvia el sistema no se activa, caso contrario, el sistema activa la red neuronal artificial y realiza las predicciones de riego.

Para comprender mejor cómo funciona la red neuronal en el sistema se realiza un diagrama de flujo que se muestra en la Figura 75.

Figura 75

Diagrama de flujo de la aplicación de la red neuronal al sistema.



Fuente: Autoría

3.2.3.1. Alimentación Eléctrica

Para el nodo actuador se escoge una batería de 3.7V a 6000mA, tomando en consideración las características de consumo de cada dispositivo que forma parte del nodo, tal como se muestra en la Tabla 29.

Tabla 29

Consumo energético de los dispositivos que conforman el nodo actuador

Dispositivo	Voltaje	Corriente	
		Modo Normal	Modo Dormido
Arduino Uno	3.3-5 V	48 mA	46 mA
Modulo LoRa	3.3-5V	10.3 mA	0.0002mA
Electroválvula	12V	250mA	-
Modulo Relé	5V	15 mA	-
TOTAL	3.3-12V	323.3 mA	46.0002mA

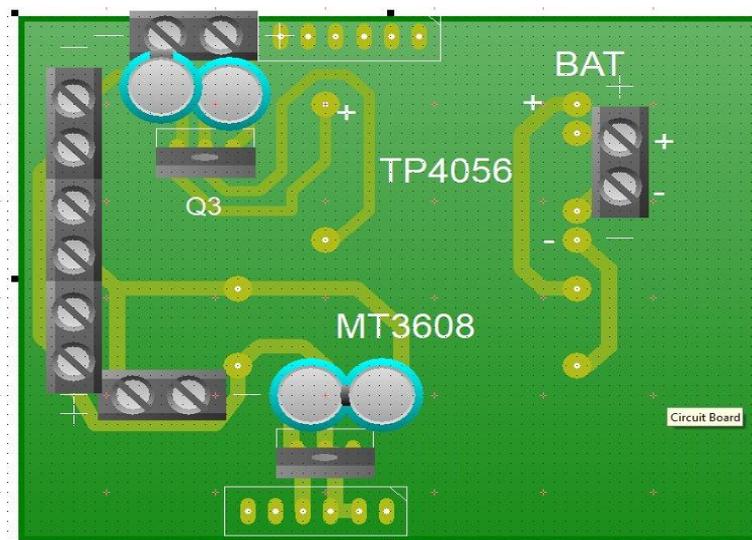
Fuente: Autoría

Al igual que el nodo sensor, el nodo actuador se alimentará con una batería de litio y conexión a un panel solar.

En la Figura 76, se observa el diagrama esquemático de la placa de alimentación, está compuesta por un módulo TP4056, cargador de batería de litio y un elevador DC-DC MT3608 que convierte el voltaje de entrada de 3,7V a 12 voltios para la alimentación de la electroválvula.

Figura 76

Diagrama esquemático- placa de alimentación del nodo actuador



Fuente: Autoría

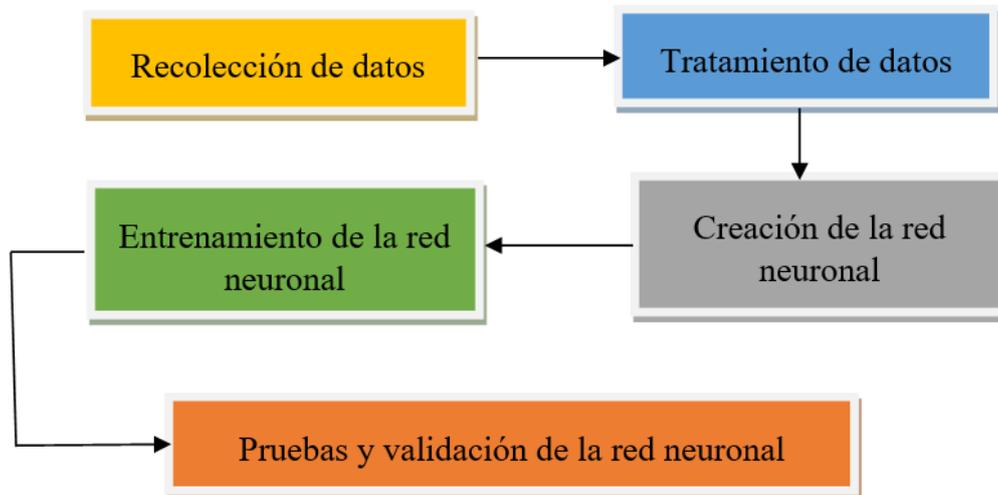
3.2.4. Codificación del Sistema

En esta sección se procede con la codificación de la red Neuronal y del nodo actuador.

Para el entrenamiento de la red neuronal se deben seguir cinco etapas indispensables que son: recolección de datos, tratamiento de datos, creación de la red neuronal, entrenamiento de la red y pruebas y validación.

Figura 77

Etapas para el entrenamiento de la red neuronal



Fuente: Autoría

Por otra parte, el nodo actuador debe contar con los mismos parámetros de comunicación LoRa que el nodo central, ya que se comunican entre sí.

3.2.4.1. Red Neuronal Artificial para el Sistema de Riego por Aspersión

Se detalla a profundidad cada una de las etapas de la red neuronal artificial para el sistema de riego por aspersión aplicado al cultivo de pasto.

Recolección de datos: La primera etapa se inicia con la recolección de datos, estos datos fueron adquiridos de los sensores de la red WSN y servirán para el proceso de aprendizaje de la red neuronal.

Tratamiento de datos: Esta etapa es primordial, el conjunto de datos debe ser tratados correctamente ya que de estos dependerá la decisión que toma la red neuronal.

Se obtuvieron 8822 datos tanto de humedad del suelo como de radiación UV, a los cuales se asignaron rangos de valores para determinar una decisión. Como se puede observar en la Figura 78, se establecen cuatro rangos para el estado del suelo y cinco para la radiación UV. De acuerdo con el estado del suelo y la radiación UV se toma la decisión de regar (1) o no regar (0) el cultivo de pasto.

Figura 78

Tratamiento de datos de humedad del suelo y radiación UV

DATASET									
HUMEDAD DEL SUELO				RADIACION UV				Decisión	
Muy Humedo	200-298	1		Bajo	(0-1-2)	1	Regar	1	
Humedo	299-349	2		Moderado	(3-4-5)	2	No Regar	0	
Seco	350-449	3		Alto	(6-7)	3			
Muy Seco	450-550	4		Muy alto	(8-9-10)	4			
				Extremadamente alto	11	5			

Humedad				Radiacion				Decisión	
Muy seco	491	4		Moderado	3	2	Regar	1	
Muy seco	489	4		Moderado	3	2	Regar	1	
Muy seco	489	4		Moderado	3	2	Regar	1	
Muy seco	489	4		Moderado	3	2	Regar	1	
Muy seco	483	4		Moderado	3	2	Regar	1	
Muy seco	484	4		Moderado	3	2	Regar	1	
Muy seco	482	4		Moderado	3	2	Regar	1	
Muy seco	481	4		Bajo	2	1	Regar	1	
Muy seco	482	4		Bajo	2	1	Regar	1	
Muy seco	481	4		Bajo	2	1	Regar	1	
Muy seco	480	4		Bajo	2	1	Regar	1	
Muy seco	481	4		Bajo	1	1	Regar	1	

Fuente: Autoría

La decisión de regar o no regar depende de las dos variables, de acuerdo con los rangos establecidos la decisión del riego se dispone tal como se observa en la Tabla 30. Se decide no regar cuando el suelo esta húmedo o muy húmedo sin importar la condición de la radiación UV. Mientras que, se decide regar cuando el suelo está seco y

muy seco siempre y cuando la radiación UV sea bajo, moderado y alto, caso contrario la decisión será no regar.

Tabla 30

Decisión de riego de acuerdo con las variables de humedad del suelo y radiación UV

Estado del suelo	Radiación UV	Decisión
Húmedo Muy Húmedo	Bajo	NO REGAR
	Moderado	
	Alto	
	Muy Alto	
	Extremadamente Alto	
Seco Muy Seco	Bajo	REGAR
	Moderado	
	Alto	
	Muy Alto	NO REGAR
	Extremadamente Alto	

Fuente: *Autoría*

Se toma la decisión de no regar cuando la radiación UV sea muy alto y extremadamente alto, puesto que cuando hay mayor exposición solar el agua se evapora más rápidamente, lo cual provoca que la planta no reciba los nutrientes necesarios. Asimismo, las hojas de las plantas pueden quemarse o dañarse gravemente a causa del cambio térmico que produciría el agua fría en contacto con la planta caliente. Finalmente se obtiene un conjunto de datos con los valores de humedad del suelo, radiación UV y la decisión.

Figura 79

Conjunto de datos para el entrenamiento de la red neuronal

1	DATASET - RED NEURONAL		
3	Humedad	radiacion	Decisión
4	200	0	0
5	201	0	0
6	202	0	0
7	203	0	0
8	204	0	0
9	205	0	0
10	206	0	0
11	207	0	0
12	208	0	0
13	209	0	0
14	210	0	0
15	211	0	0
16	212	0	0
17	213	0	0
18	214	0	0
19	215	0	0
20	216	0	0
21	217	0	0
22	218	0	0
23	219	0	0
24	220	0	0
25	221	0	0
26	222	0	0

Fuente: *Autoría*

Creación de la red neuronal: Se define la arquitectura o estructura de la red, es decir el número de capas y el número de neuronas que va a tener la misma. Existen tres capas en una red neuronal: entrada, salida y ocultas.

La capa de entrada recibe datos de variable procedentes del entorno, la capa de salida proporciona la respuesta de la red neuronal y la capa oculta proporciona grados de libertad a la red neuronal, es decir permite que la red sea capaz de representar más fidedignamente determinadas características del entorno que trata de modelar.

En este caso, el número de neuronas en la capa de entrada son dos, que corresponden a las variables de humedad del suelo y radiación UV, el número de neuronas en la capa de salida es una, que corresponde a la decisión del riego (regar o no regar), la capa oculta no tiene un número de neuronas determinada, pero se puede calcular mediante la regla de la pirámide geométrica, una regla aproximada para

seleccionar el número de neuronas en las capas ocultas. Para una red de tres capas, con una sola capa oculta se utiliza la Ec. 21 (Zepeda Hernandez et al., 2015)

$$h = \sqrt{m \times n} \quad \text{Ec. 21}$$

Donde:

h: número inicial de neuronas en la capa oculta

m: número de neuronas en la capa de salida

n: número de neuronas en la capa de entrada

Aplicando esta regla se obtiene que el número de neuronas en la capa oculta es 1.41 neuronas, siendo inferior al número de neuronas de entrada, pero superior al número de neuronas de salida.

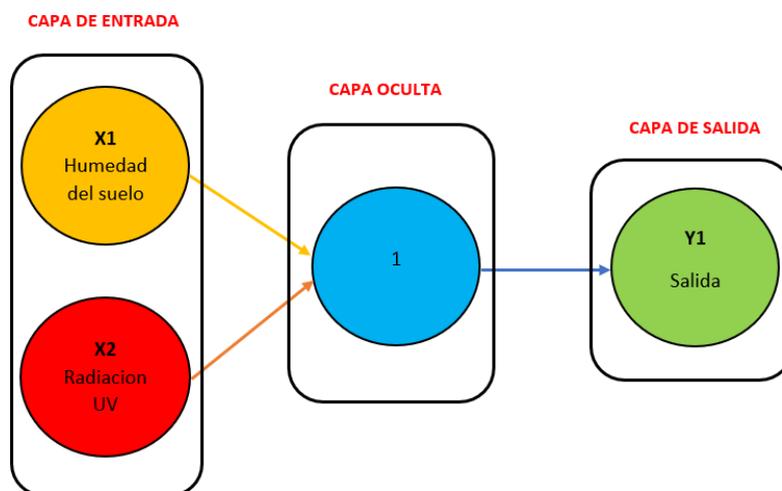
$$h = \sqrt{1 \times 2}$$

$$h = 1.41$$

Es muy complejo determinar el número de capas y el número de neuronas a incluir en cada capa oculta, ya que incide directamente en la capacidad de generalización del modelo. De acuerdo con esta regla, el modelo de la red sería la que se muestra en la Figura 80.

Figura 80

Modelo de la red neuronal con una capa oculta



Fuente: Autoría

Para determinar el número de neuronas de una red con dos capas ocultas se utiliza la Ec. 22 para calcular el número de neuronas en la primera capa oculta y la Ec. 23 para determinar el número de neuronas de la segunda capa oculta (Pano Azucena, 2015).

$$h_1 = m \times r^2 \quad \text{Ec. 22}$$

$$h_2 = m \times r \quad \text{Ec. 23}$$

Donde:

$$r = 3 \sqrt{\frac{n}{m}}$$

h_1 : número de neuronas en la primera capa oculta

h_2 : número de neuronas en la segunda capa oculta

Reemplazando los valores se tiene los siguientes resultados: El número de neuronas en la primera capa oculta es 18 y el número de neuronas en la segunda capa oculta es 4,24. El modelo de la red con dos capas ocultas se observa en la Figura 81.

$$r = 3\sqrt{2} = 4.24$$

$$h_1 = 1 \times (3\sqrt{2})^2$$

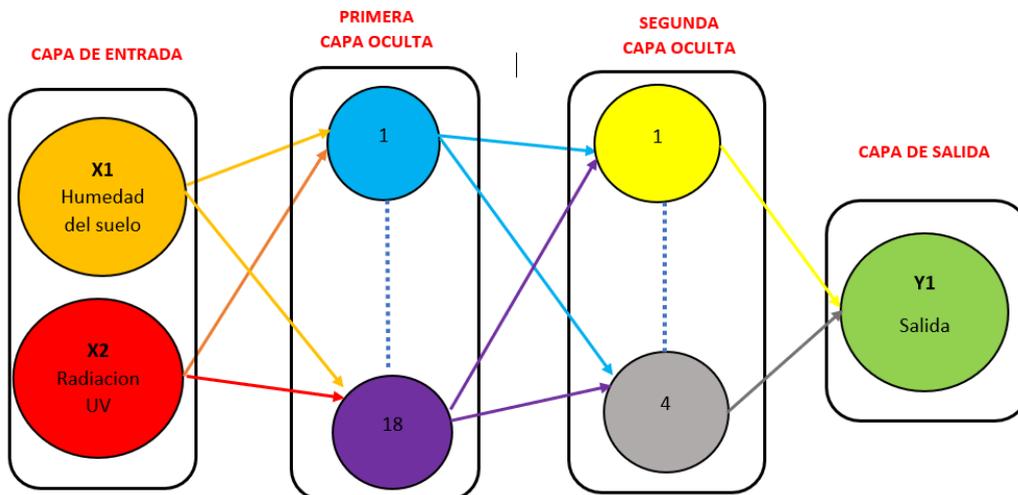
$$h_1 = 18$$

$$h_2 = 1 \times 3\sqrt{2}$$

$$h_2 = 4.24$$

Figura 81

Modelo de la red neuronal con dos capas ocultas

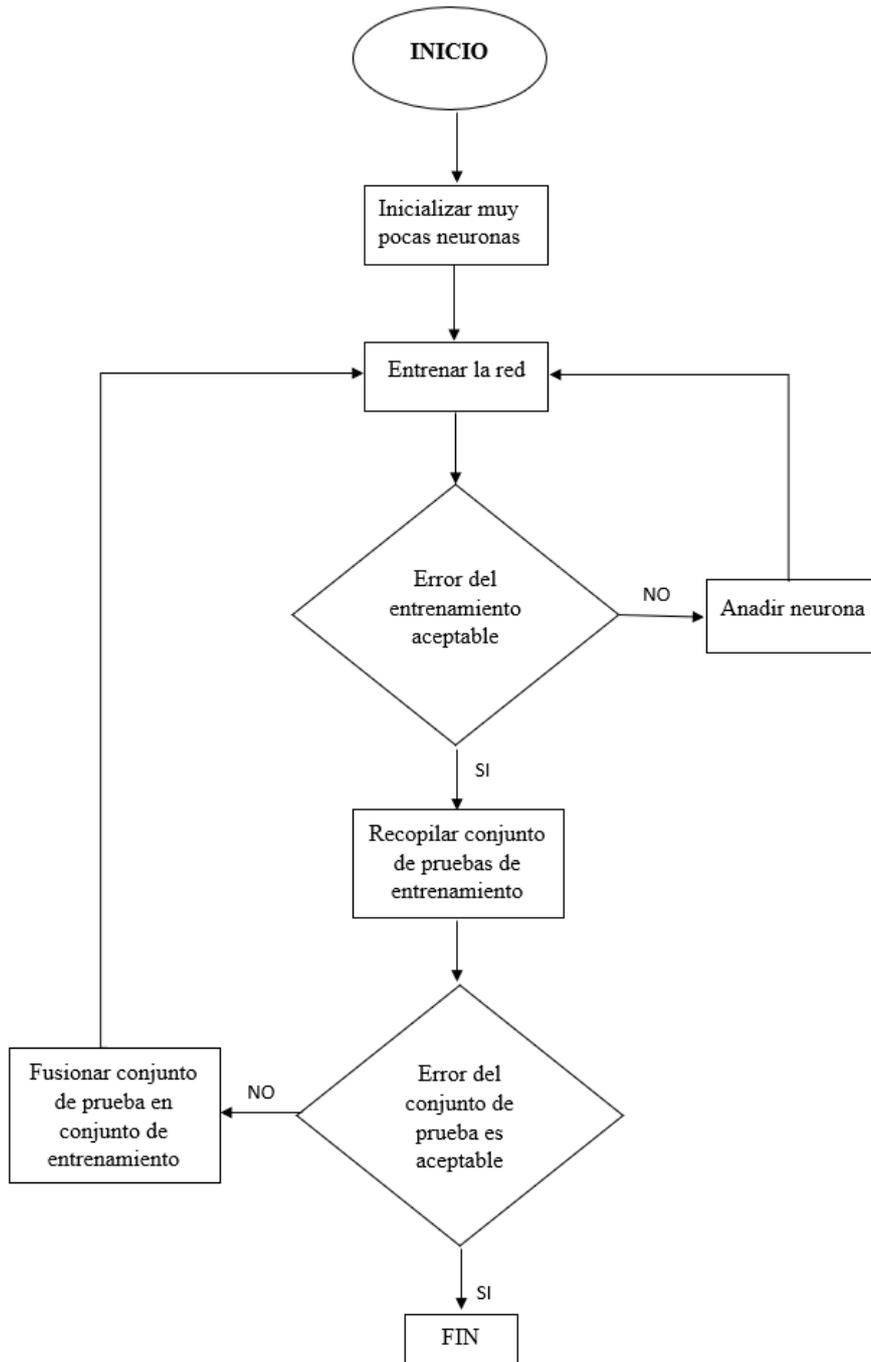


Fuente: Autoría

Las fórmulas anteriores son aproximaciones del tamaño ideal de las capas ocultas, sin embargo, existe un gran número de reglas, métodos o procedimientos a utilizar según sea el caso.

Figura 82

Diagrama de flujo para determinar el número de neuronas y capas ocultas.



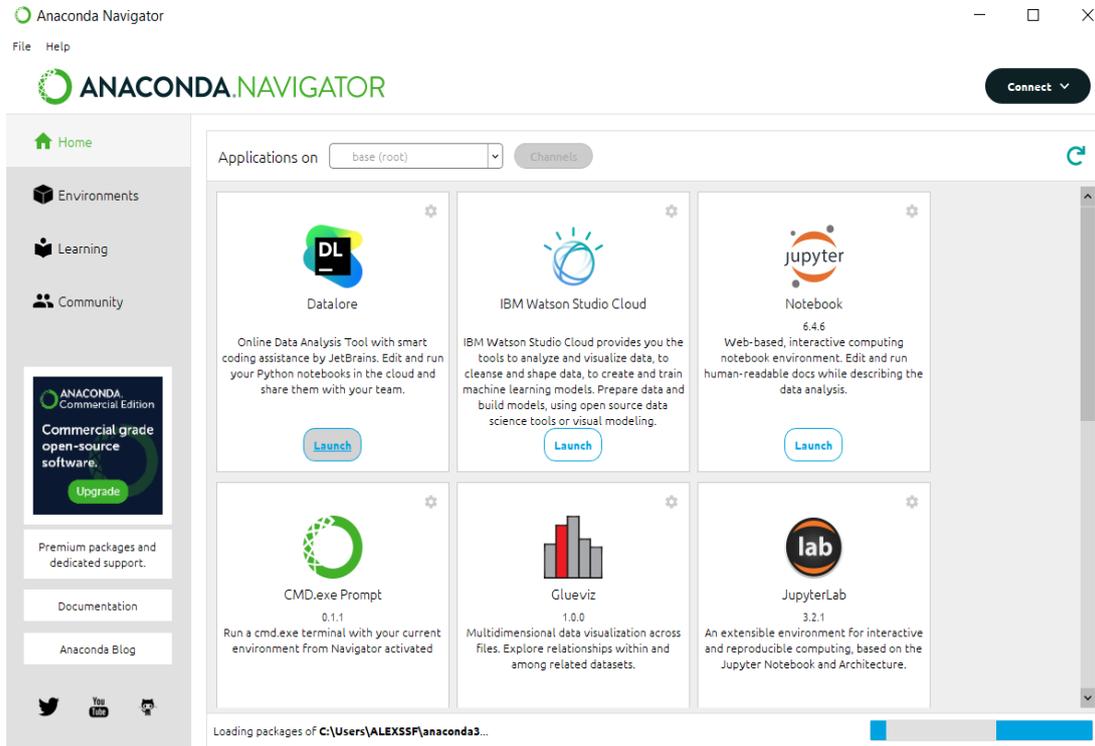
Adaptado de *Diseño de redes neuronales multicapa y entrenamiento*, por ESCOM, 2019. (<https://es.slideshare.net/mentelibre/diseo-de-redes-neuronales-multicapa-y-entrenamiento>)

Así que, una de las formas para obtener el número óptimo de neuronas es iniciar con un número pequeño de neuronas en la capa oculta y seleccionar un criterio apropiado para evaluar, entrenar y probar la red. En otras palabras, entrenar y probar la red con diferentes números de neuronas en la capa oculta hasta asegurar un desempeño adecuado con el conjunto de entrenamiento. En la Figura 82 se muestra el proceso que se sigue para determinar el número de neuronas y capas ocultas.

Entrenamiento de la Red Neuronal: La cuarta etapa es el entrenamiento de la red neuronal, para ello se procede a configurar el entorno de desarrollo utilizando la distribución Anaconda, una plataforma que facilita la ciencia de datos y el aprendizaje automático con Python. Una vez instalado Anaconda, se tiene fácil acceso a varias de sus herramientas, una de ellas es Anaconda Navigator una interfaz gráfica de usuario (GUI) de escritorio donde se creará el entorno de trabajo y accediendo a ella se puede utilizar Jupyter Notebook una aplicación web de código abierto donde se ejecutará el código para el entrenamiento de la red neuronal.

Figura 83.

Anaconda Navigator3



Fuente: *Autoría*

Las librerías importantes que se importan para el entrenamiento de la red neuronal son las siguiente:

Tensor Flow: una biblioteca para aplicaciones de aprendizaje automático y redes neuronales profundas que ejecuta de forma eficiente gráficos de flujos.

Keras: una biblioteca de redes neuronales escrita en Python con una sintaxis homogénea ampliable para la creación de redes neuronales

Pandas: Proporciona estructuras de datos y herramientas para el análisis de datos.

Numpy: Brinda soporte para crear vectores y matrices.

Matplotlib: Generación de gráficos a partir de datos obtenidos de listas.

Figura 84

Librerías necesarias para el entrenamiento de la red neuronal

```
In [1]: ▶ import pandas as pd
import tensorflow as tf
import numpy as np
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import matplotlib.pyplot as plt
print(tf.__version__)

2.1.0
```

Fuente: *Autoría*

Como siguiente paso se lleva a cabo el preprocesamiento del conjunto de datos recopilados en la segunda etapa del archivo DatasetV2. Como se observa en la Figura 85 se tiene una cantidad de 8222 filas y tres columnas correspondientes a la humedad, radiación y decisión.

Figura 85

Preprocesamiento de datos

```
▶ #preprosesamiento de datos
df = pd.read_excel("DatasetV2.xlsx", sheet_name="data")
print(df)
print(df.columns)
```

	Humedad	radiacion	Decisión
0	200	0	0
1	201	0	0
2	202	0	0
3	203	0	0
4	204	0	0
...
8817	596	10	0
8818	597	10	0
8819	598	10	0
8820	599	10	0
8821	600	10	0

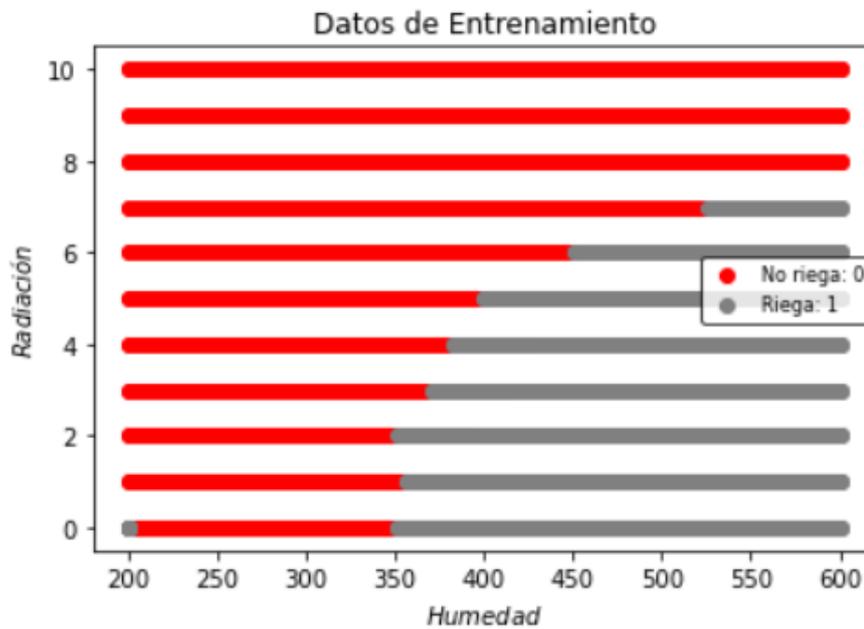
[8822 rows x 3 columns]
Index(['Humedad', 'radiacion', 'Decisión'], dtype='object')

Fuente: *Autoría*

En la Figura 86 se representa de forma gráfica el conjunto de datos de entrenamiento, el eje de las X representa la humedad del suelo y el eje de las Y la radiación UV donde, la sección de color gris corresponde a la decisión de cuando el sistema debe regar y la de color rojo corresponde a cuando el sistema no debe regar.

Figura 86

Representación gráfica de datos de entrenamiento



Nota. El eje de las X representa la humedad del suelo y el eje Y la radiación UV. Los valores de color rojo representan la predicción de No Regar y los valores de color gris la predicción de Regar.

Fuente: Autoría

Posterior a esto, ya definido el modelo, es el momento de entrenarlo para ver la actuación de este sobre los datos disponibles. Como se mencionó en la tercera etapa, es difícil identificar el número suficiente de capas y neuronas, hasta que la red no es probada, no se puede deducir que la red neuronal responda correctamente a la predicción que se desea obtener. Por lo tanto, que se realiza pruebas con diferentes números de capas y neuronas en la capa oculta.

Tal como se puede observar en la Tabla 30, se inicia con el primer modelo, este modelo consta de una capa de entrada, dos capas ocultas y una capa de salida, con 18

neuronas en la primera capa oculta y 4 neuronas en la segunda. Al entrenar este modelo se no se obtuvo la salida deseada, es decir la predicción no va acorde a los datos de entrenamiento, por lo tanto, se aumenta el número de neuronas de la capa oculta.

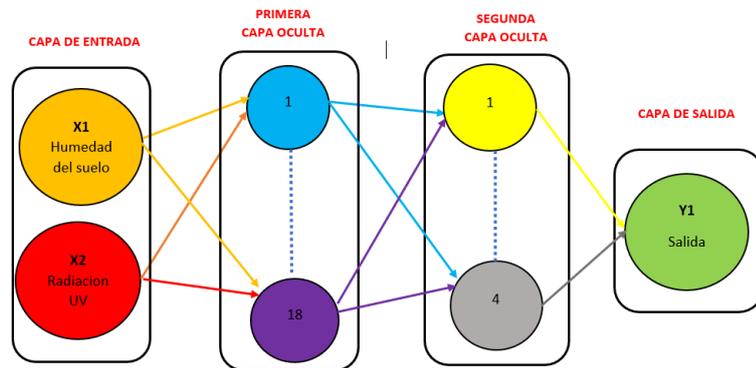
Se prueba un segundo modelo de red neuronal que consta de una capa de entrada, 3 capas ocultas y una capa de salida; con 20 neuronas en cada capa oculta. Al realizar el entrenamiento de la red con este modelo se obtuvo un resultado positivo, pero no cumple la predicción con algunos valores de acuerdo con los datos de entrenamiento. Por consiguiente, se aumenta el número de neuronas en cada capa oculta.

Tabla 31

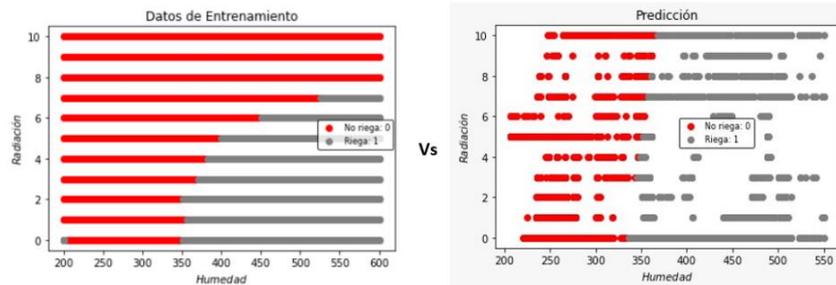
Modelos para el entrenamiento de la red neuronal

MODELOS DE PRUEBA PARA EL ENTRENAMIENTO DE LA RED NEURONAL

Primer Modelo
1 capa de entrada
2 capas Ocultas
 18 neuronas en la primera capa
 4 neuronas en la segunda capa
1 capa de salida

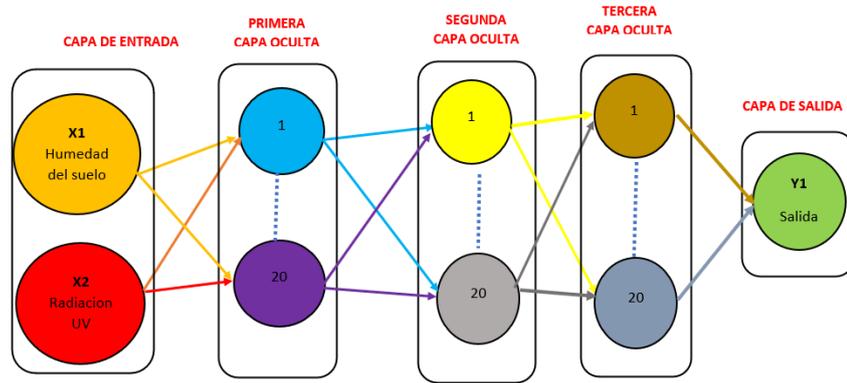


Resultado del entrenamiento

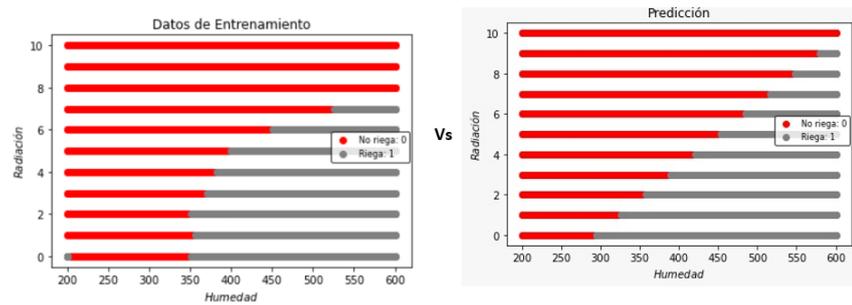


Segundo Modelo

1 capa de entrada
3 capas ocultas
20 neuronas en la primera capa
20 neuronas en la segunda capa
20 neuronas en la tercera capa
1 capa de salida

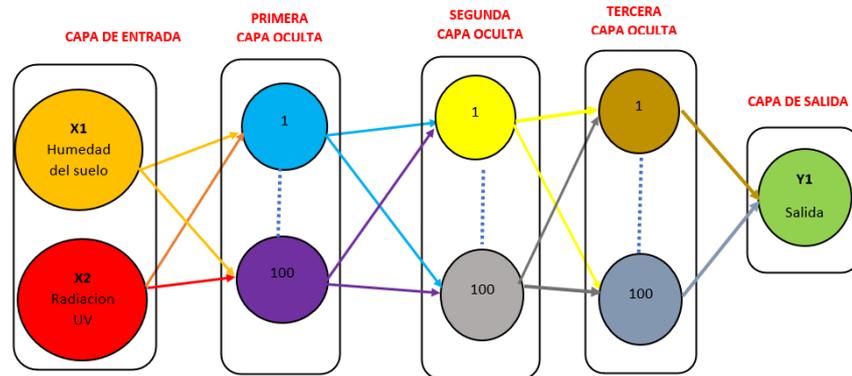


Resultado del entrenamiento

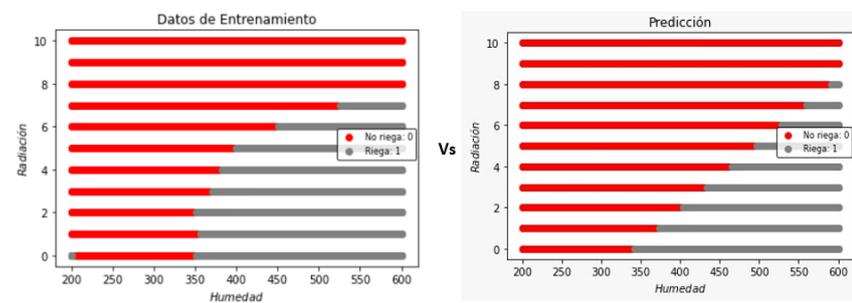


Tercer Modelo

1 capa de entrada
3 capas ocultas
100 neuronas en la primera capa
100 neuronas en la segunda capa
100 neuronas en la tercera capa
1 capa de salida



Resultado del entrenamiento



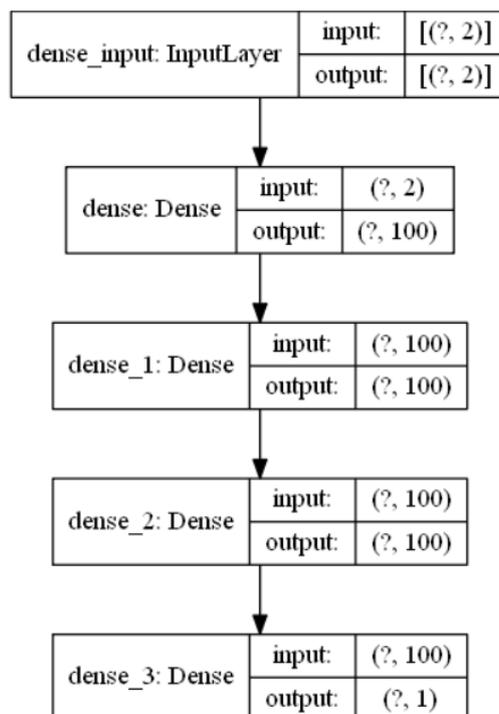
Fuente: Autoría

El tercer modelo de la red neuronal consta de una capa de entrada, tres capas ocultas y una capa de salida; con 100 neuronas en cada capa oculta. Con este modelo se obtuvo predicciones más cercanas al conjunto de datos de entrenamiento por lo que se elige este modelo y se detalla más a profundidad el proceso del entrenamiento del mismo.

En la Figura 87 se puede observar el modelo de la red con la forma de entrada y salida de cada una de las capas. Se inicia con la capa de entrada, a esta capa ingresan dos neuronas y salen las mismas dos neuronas, pasa a la siguiente capa, que es la primera capa oculta, a esta capa ingresan las dos neuronas y salen cien, prosigue a la segunda capa oculta con cien neuronas a la entrada y cien neuronas a la salida, continua con la tercera capa oculta donde ingresa cien neuronas y se tiene cien neuronas a la salida y finalmente termina con la capa de salida en el cual ingresan cien neuronas y se tiene una neurona a la salida.

Figura 87

Modelo de la red neuronal con la forma de entradas y salidas



Fuente: Autoría

Las *funciones de activación* que se utiliza en este modelo son: ReLU para las capas ocultas y sigmoide para la capa de salida. La función de activación permite definir la relación entre los nodos en el paso entre capa y capa.

Figura 88

Función de activación de la red neuronal

```
# define model
model = Sequential()
model.add(Dense(100, input_dim=2, activation='relu'))
model.add(Dense(100, activation='relu'))
model.add(Dense(100, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

Fuente: Autoría

La función ReLU transforma los valores introducidos anulando los valores negativos y manteniendo los valores positivos y la función sigmoide permite proporcionar una salida comprendida entre 0-1 para la clasificación binaria.

Por otra parte, el *algoritmo de aprendizaje* que se utiliza es Backpropagation (propagación del error hacia atrás), donde se presenta ante la red entrenada, un patrón de entrada que se transmite por las diferentes capas hasta obtener una salida y, después, una etapa de entrenamiento donde se modifica los pesos de la red de manera que coincida la salida deseada con la salida obtenida por la red.

Además, los *hiperparámetros* que permiten controlar el proceso de entrenamiento de este modelo se da a conocer en la Figura 89, donde se distingue dos fases; la compilación y el ajuste.

- 1. Compilación:** Antes de entrenar el modelo se realiza el proceso posterior al aprendizaje donde se define el optimizador, la función de pérdida y la lista de métricas.

El *optimizador* se utiliza al momento de entrenar la red neuronal para optimizar la función de coste. Este valor es la media de la función de pérdida entre el valor predicho

y el valor actual. El valor predicho se obtiene durante el proceso de Backpropagation (propagación del error hacia atrás) y hace uso de los pesos y bias o sesgo de la red. Con la ayuda del algoritmo de optimización se minimiza el valor de la función de coste, actualizando las variables de entrenamiento.

La función utilizada en este modelo es denominada *Adam* (Estimación adaptativa de momentos) que permite tasas de aprendizaje adaptativo por cada parámetro. Por lo tanto, cuanto mayor sea este valor, se requerirá un mayor número de *epochs* (número de veces o iteraciones) de entrenamiento proporcionando pequeños cambios en los pesos en cada actualización.

Figura 89

Hiperparámetros de la fase de entrenamiento de la red neuronal, fase de compilación

```
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Fuente: Autoría

La *función de pérdida* es la función que se requiere minimizar con la función de optimización Adam, es decir la función de coste. Donde *loss* representa la diferencia entre el valor predicho y el valor real.

La función de pérdida usada en este modelo es **Binary Cross-Entropy**, el principal objetivo de esta función es comparar que tan bueno es el modelo en las predicciones, es decir, como de bien es la salida de la función *sigmoid* (función de activación sigmoide) con las dos posibles categorías 1 (regar) 0 (no regar).

La *lista de métricas* se define los aspectos del modelo que se deben evaluar durante el proceso de entrenamiento y pruebas; estos son precisión (Accuracy) y pérdida (Loss). La precisión representa la fracción de predicciones que el modelo realizó correctamente y se calcula con la Ec. 24.

$$\text{Precisión} = \frac{N^{\circ} \text{ correctas}}{N^{\circ} \text{ totales}}$$

Ec. 24

Por otro lado, la pérdida indica que tan incorrecto ha sido la predicción del modelo; si la predicción es perfecta, la pérdida es cero, de lo contrario el valor es mayor. El objetivo es que las pérdidas sean bajas en la mayoría de los ejemplos.

- 2. Ajuste:** Los parámetros de ajuste en el modelo de la red neuronal son los siguientes: entradas y salidas, tamaño de batch (lote), epochs (número de veces o iteraciones) y separación de validación (validation split)

Figura 90

Hiperparámetros de la fase de entrenamiento, fase de ajuste

```
num_epochs = 5000
batch_size = X.shape[0]

history = model.fit(X, Y,
                    epochs=num_epochs,
                    batch_size=batch_size,
                    validation_split = 0.3,
                    verbose=1)
```

Fuente: Autoría

La *entrada y salida* (x, y) representan los datos de entrada (x), en este caso, las variables recolectadas por el nodo sensor (humedad del suelo y radiación UV). Y la salida (y) el resultado de riego, es decir la predicción.

Epochs especifica el número de veces o iteraciones que el algoritmo de aprendizaje va a trabajar con todo el dataset.

El tamaño de Bach o lote define el número de datos que se van a utilizar cada vez que el modelo actualiza sus parámetros internos.

La separación de validación permite determinar un porcentaje de los datos de entrada que serán usados únicamente para probar el modelo durante el entrenamiento.

Pruebas y validación de la red neuronal: La quinta etapa es pruebas y validación; el número de épocas, o número de veces (iteraciones) que se entrenó la red

es de 5000 veces sobre la totalidad de los datos proporcionados, por otro lado, se estableció un parámetro de validación de 0.3, es decir, la cantidad de datos de entrenamiento que se usa para la validación es el 30%; los datos de validación no se utilizan para el entrenamiento, sino para evaluar la pérdida y la precisión. En la Figura 91 se visualiza un registro de métricas de entrenamiento para cada época, esto incluye la pérdida y la precisión del modelo de entrenamiento, así como la pérdida y la precisión del conjunto de datos de validación.

Figura 91

Historial de entrenamiento del modelo

```
Epoch 4989/5000
6175/6175 [=====] - 0s 1us/step - loss: 0.1271 - accuracy: 0.9456 - val_loss: 0.1856 - val_accuracy: 0.9112
Epoch 4990/5000
6175/6175 [=====] - 0s 1us/step - loss: 0.1276 - accuracy: 0.9409 - val_loss: 0.1993 - val_accuracy: 0.9120
Epoch 4991/5000
6175/6175 [=====] - 0s 1us/step - loss: 0.1282 - accuracy: 0.9461 - val_loss: 0.1857 - val_accuracy: 0.9105
Epoch 4992/5000
6175/6175 [=====] - 0s 2us/step - loss: 0.1292 - accuracy: 0.9393 - val_loss: 0.2050 - val_accuracy: 0.9101
Epoch 4993/5000
6175/6175 [=====] - 0s 1us/step - loss: 0.1305 - accuracy: 0.9459 - val_loss: 0.1871 - val_accuracy: 0.9086
Epoch 4994/5000
6175/6175 [=====] - 0s 1us/step - loss: 0.1326 - accuracy: 0.9372 - val_loss: 0.2151 - val_accuracy: 0.9074
Epoch 4995/5000
6175/6175 [=====] - 0s 1us/step - loss: 0.1352 - accuracy: 0.9438 - val_loss: 0.1917 - val_accuracy: 0.9056
Epoch 4996/5000
6175/6175 [=====] - 0s 1us/step - loss: 0.1393 - accuracy: 0.9326 - val_loss: 0.2320 - val_accuracy: 0.9029
Epoch 4997/5000
6175/6175 [=====] - 0s 1us/step - loss: 0.1439 - accuracy: 0.9399 - val_loss: 0.2029 - val_accuracy: 0.9014
Epoch 4998/5000
6175/6175 [=====] - 0s 1us/step - loss: 0.1516 - accuracy: 0.9266 - val_loss: 0.2580 - val_accuracy: 0.8946
Epoch 4999/5000
6175/6175 [=====] - 0s 2us/step - loss: 0.1586 - accuracy: 0.9328 - val_loss: 0.2219 - val_accuracy: 0.8976
Epoch 5000/5000
6175/6175 [=====] - 0s 1us/step - loss: 0.1706 - accuracy: 0.9202 - val_loss: 0.2890 - val_accuracy: 0.8870
```

Nota. La opción **accuary** representa la precisión del conjunto de datos de entrenamiento y **val_accuary** representa la precisión del conjunto de datos de validación, **loss** corresponde la pérdida del conjunto de datos de entrenamiento y **val_loos** la pérdida del conjunto de datos de validación. **Fuente:** *Autoría*

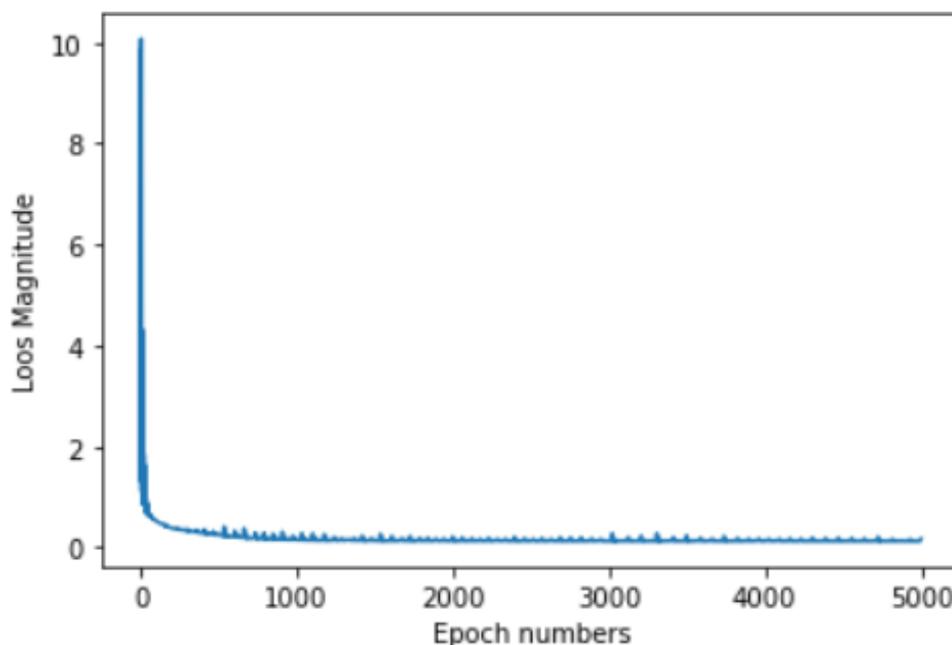
Al comparar la precisión en los conjuntos de datos de entrenamiento y validación durante las épocas de entrenamiento se puede ver que el modelo está completamente entrenado ya que la tendencia de la precisión en ambos conjuntos no

aumenta en las últimas épocas, si los valores siguieran en aumento probablemente podría entrenarse un poco más.

En cuanto a la pérdida en los conjuntos de datos de entrenamiento y validación durante las épocas de entrenamiento se puede ver que el modelo tiene un rendimiento comparable tanto en conjunto de datos de entrenamiento como de validación. En la Figura 92 se puede apreciar de mejor manera mediante una gráfica la pérdida de magnitud, inicia con una pérdida de 10 en la época uno, y va cayendo hasta aproximarse a 0 en la época cinco mil.

Figura 92

Representación gráfica de la pérdida de magnitud del conjunto de datos de entrenamiento.



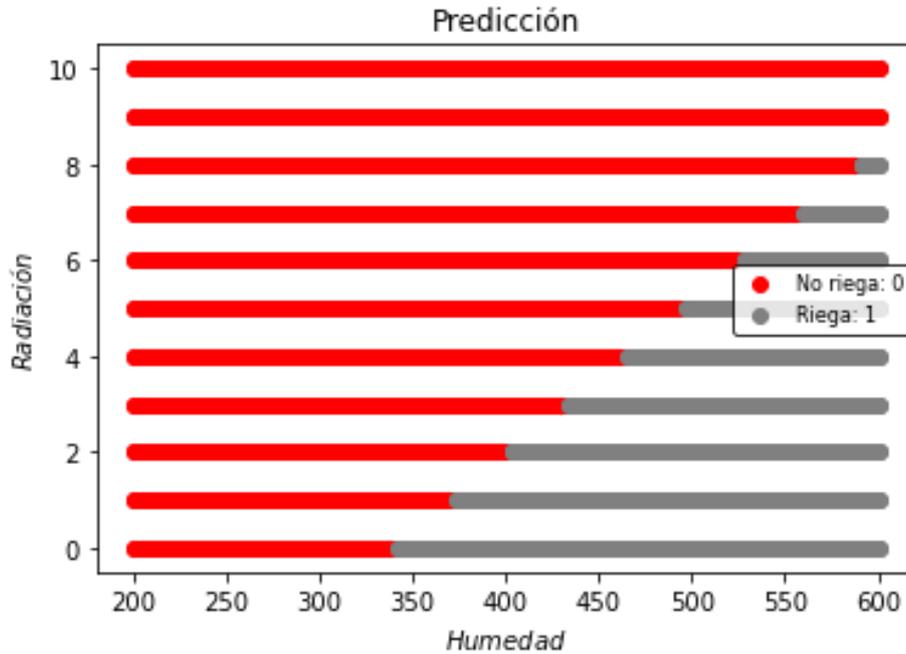
Nota. El eje de las **x** representa el número de épocas y el eje de las **y** la pérdida de magnitud.

Fuente: Autoría

Finalmente se puede observar en la Figura 93 la predicción del sistema de riego, con las variables de humedad del suelo en el eje de las **X**, con un rango de 200 (muy húmedo) a 600 (muy seco) y la radiación UV en el eje de las **Y** con un rango de 0 (bajo) a 10 (extremadamente alto).

Figura 93

Representación gráfica de la predicción de la red neuronal



Nota. El eje de las X representa la humedad del suelo y el eje Y la radiación UV. Los valores de color rojo representan la predicción de No Regar y los valores de color gris la predicción de Regar.

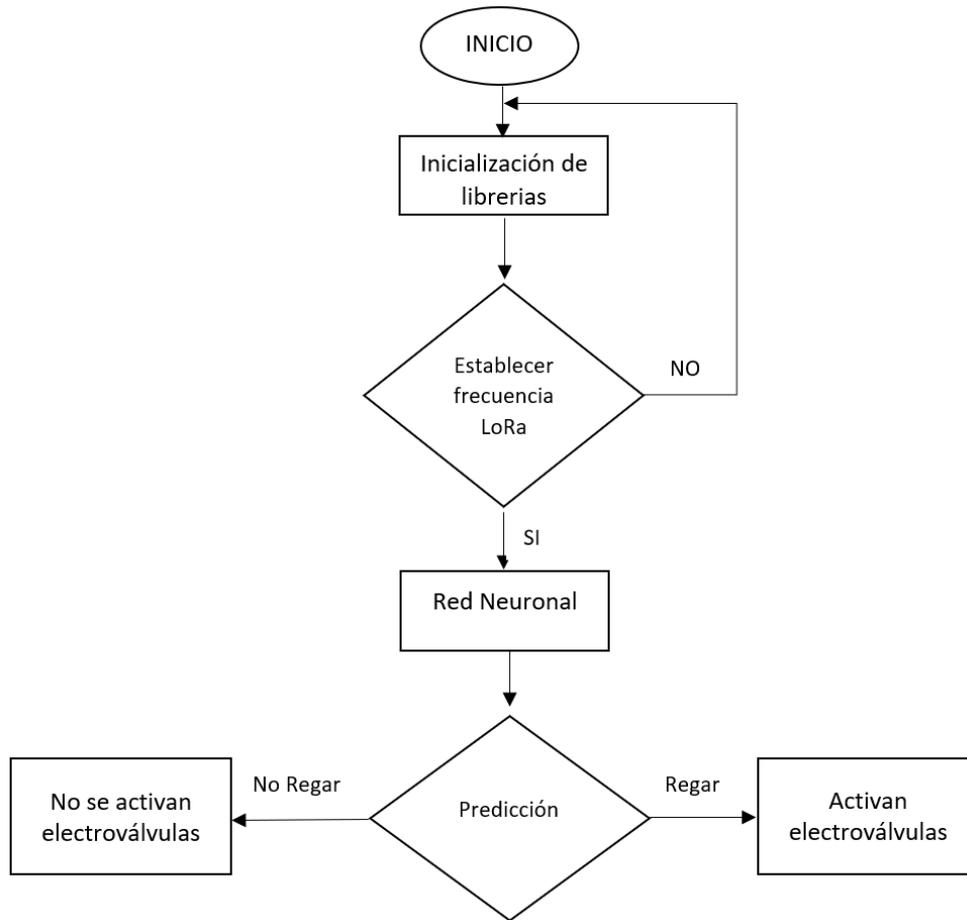
Fuente: Autoría

3.2.4.2. Nodo Actuador

El nodo actuador ejecuta las predicciones realizadas por la red neuronal a través de la orden enviada desde el nodo central; activara o desactivara la electroválvula que permite el paso del flujo de agua hacia los aspersores en el cultivo de pasto. El diagrama de flujo del nodo actuador se puede visualizar en la Figura 94.

Figura 94

Diagrama de flujo del nodo actuador



Fuente: Autoría

La codificación del módulo LoRa para la comunicación inalámbrica deberá cumplir con las mismas configuraciones que nodo central. Los parámetros de la comunicación LoRa quedan establecidos con una potencia de transmisión de 20 dBm, un factor de propagación SF7, un ancho de banda de 125 kHz y una tasa de codificación de errores de 4/5 tal como se puede observar en la Figura 95.

Figura 95

Configuraciones básicas del nodo actuador

```
Nodo_actuador.ino
 8 void setup()
 9 {
10   pinMode(led, OUTPUT);
11   pinMode(electrovalvula, OUTPUT);
12   Serial.begin(9600);
13   while (!Serial) ; // Wait for serial port to be available
14   if (!rf95.init())
15     Serial.println("init failed");
16   rf95.setFrequency(915.0);
17   rf95.setTxPower(20);
18   rf95.setSpreadingFactor(7);
19   rf95.setSignalBandwidth(125000);
20   rf95.setCodingRate4(5);
21 }
22
```

Fuente: *Autoría*

3.2.5. Pruebas del sistema: Segunda Iteración

En esta etapa se realizan las pruebas de funcionamiento del sistema de riego en base a las predicciones realizadas por la red neuronal, se verifica el correcto funcionamiento del sistema dividido en tres etapas.

Test eléctrico del nodo actuador: Verificar si cumple con los requerimientos de corriente del nodo con el fin de obtener un funcionamiento correcto del módulo LoRa, Arduino UNO y electroválvulas.

Predicción de la red neuronal: Se verifica la predicción de la red neuronal ante los datos de las variables ambientales recolectados por el nodo sensor en el cultivo de pasto.

Comportamiento del nodo actuador: El nodo actuador debe ejecutar la orden enviada desde el nodo central, es decir, la predicción realiza por la red neuronal, mediante la orden la electroválvula debe activarse o desactivarse respectivamente.

3.2.5.1. Test Eléctrico del Nodo Actuador

Se verifica el encendido de cada uno de los dispositivos que conforma el nodo actuador mediante el indicador led, los dispositivos que no cuentan con este indicador se realizó la medición de voltaje en sus terminales. Se obtuvo un resultado satisfactorio.

Figura 96

Nodo actuador alimentado con una batería de litio



Fuente: Autoría

Las pruebas de funcionamiento de la placa de alimentación se realizaron a través de dos pruebas; la primera, se realiza cuando el nodo sensor está en funcionamiento y el panel solar no está conectada y la segunda cuando el nodo sensor está en funcionamiento y el panel solar está conectado a la placa de alimentación; los resultados se dan a conocer en la Tabla 32 donde se calcula la duración de la batería para ambos casos.

Tabla 32

Test de alimentación del nodo actuador

Nº Prueba	Hora de Inicio	Hora Fin	Tiempo de duración
1	16:00	10:00	18 horas
2	08:00	14:00	22 horas

Fuente: Autoría

El voltaje suministrado es adecuado, por lo cual, se procede al uso de la batería de litio de 3.7 voltios a 6000 mA para la alimentación del nodo actuador. También se verificó el funcionamiento del panel solar para recargar la batería de litio con el indicador led que posee el módulo de carga TP4056.

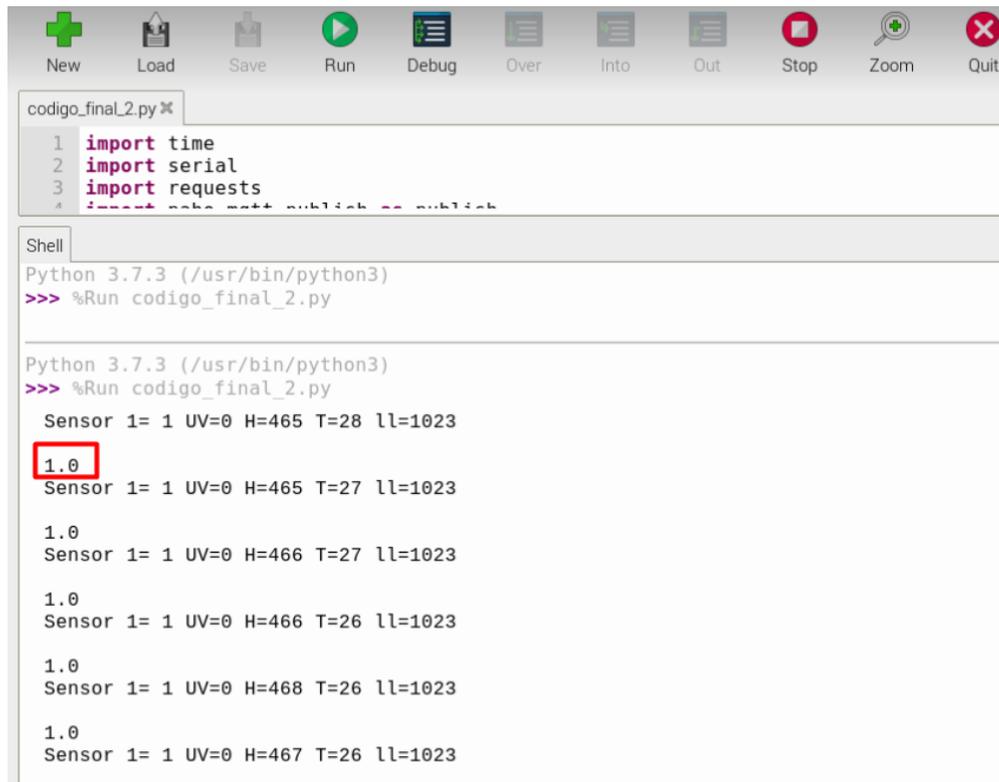
3.2.5.2.Predicción de la Red Neuronal

Los datos de las variables ambientales enviadas por los sensores y la predicción de la red neuronal se pueden visualizar en el nodo central a través del Shell del entorno de desarrollo integrado Tonny de Python. Para que se ejecute el modelo de entrenamiento de la red neuronal el valor del sensor de lluvia debe sobrepasar el valor de 500, es decir, que no existe presencia de lluvia.

Una vez que se ejecuta el modelo de entrenamiento, la red realiza la predicción; regar (1) y no regar (0) en base a los datos del sensor de humedad del suelo y radiación UV. Como se puede observar en la Figura 97, el valor del sensor de lluvia es 1023, es decir que el modelo se ejecuta, el valor del sensor UV es 0 (radiación UV Bajo) y el sensor de humedad del suelo sobrepasa el valor de 450 (Muy Seco), por lo tanto, la predicción de la red neuronal es Regar, generando el valor de 1.0

Figura 97

Predicción de la Red Neuronal- Regar



The image shows a screenshot of a Python IDE interface. At the top, there is a toolbar with icons for New, Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, and Quit. Below the toolbar, a code editor window titled 'codigo_final_2.py' contains the following Python code:

```
1 import time
2 import serial
3 import requests
4 import sys
```

Below the code editor is a Shell window. It shows the execution of the Python script. The output of the script is as follows:

```
Python 3.7.3 (/usr/bin/python3)
>>> %Run codigo_final_2.py

Python 3.7.3 (/usr/bin/python3)
>>> %Run codigo_final_2.py

Sensor 1= 1 UV=0 H=465 T=28 ll=1023
1.0
Sensor 1= 1 UV=0 H=465 T=27 ll=1023

1.0
Sensor 1= 1 UV=0 H=466 T=27 ll=1023

1.0
Sensor 1= 1 UV=0 H=466 T=26 ll=1023

1.0
Sensor 1= 1 UV=0 H=468 T=26 ll=1023

1.0
Sensor 1= 1 UV=0 H=467 T=26 ll=1023
```

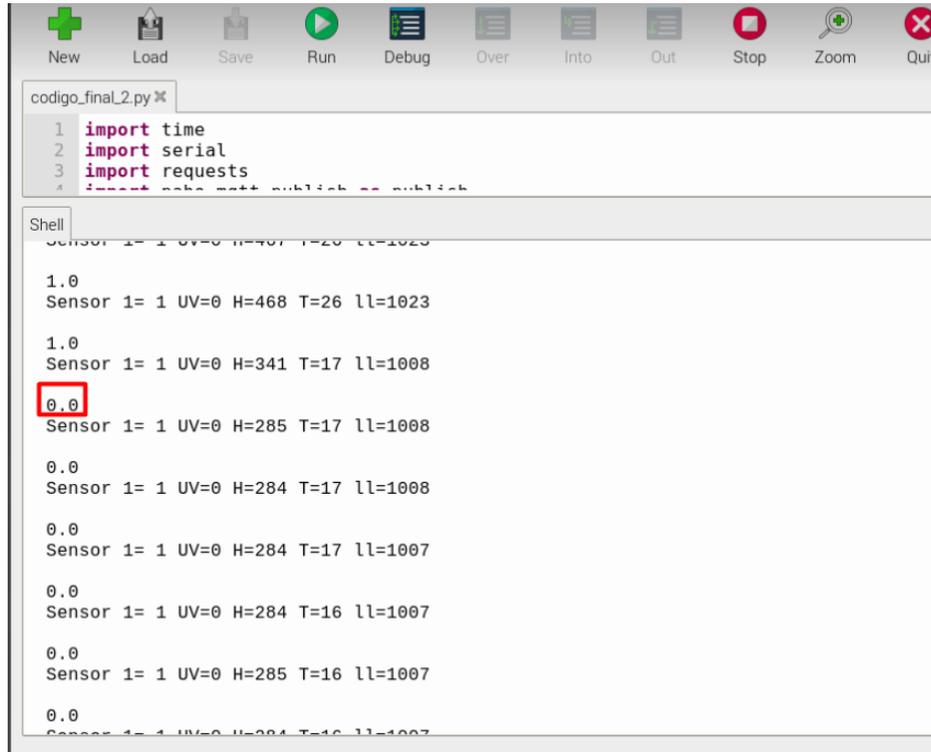
The value '1.0' is highlighted with a red box in the first instance of the output.

Fuente: Autoría

Mientras tanto, en la Figura 98 se puede observar que el valor del sensor de humedad del suelo está por debajo de 350 esta escala corresponde a un suelo Húmedo (299-349) y Muy Húmedo (200-298). La predicción de la red neuronal es No regar, generando un valor de 0.0.

Figura 98

Predicción de la Red Neuronal- No Regar



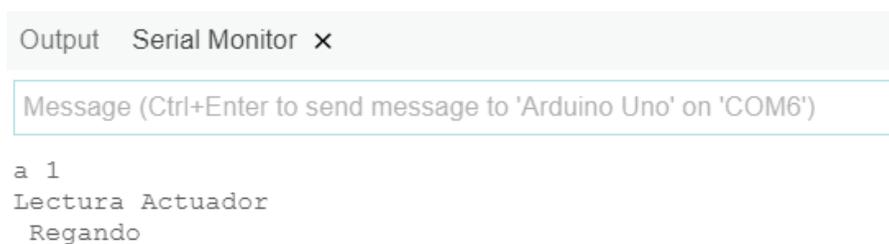
Fuente: Autoría

3.2.5.3. Comportamiento del Nodo Actuador

La predicción realizada por la red neuronal en el nodo central es enviada a través de comunicación inalámbrica LoRa al nodo actuador que se encuentra a una distancia de 160 metros. Como se puede observar en la Figura 99, la orden enviada desde el nodo central (predicción de la red neuronal) es Regar (1), en el monitor serial de Arduino se observa la lectura del nodo actuador “Regando”.

Figura 99

Lectura del nodo actuador en estado “Regando”

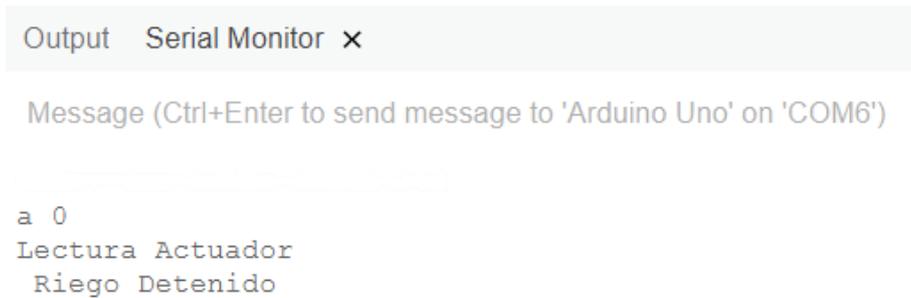


Fuente: Autoría

Por otro lado, cuando la predicción de la red neuronal es No Regar (0) y el nodo central envía la orden al nodo actuador, en el monitor serial de Arduino se puede observar que la lectura del nodo actuado es “Riego Detenido”

Figura 100

Lectura del nodo actuador en estado "Riego Detenido"



```
Output Serial Monitor x
Message (Ctrl+Enter to send message to 'Arduino Uno' on 'COM6')
a 0
Lectura Actuador
Riego Detenido
```

Fuente: Autoría

3.2.6. Conclusiones: Segunda iteración

Los dispositivos seleccionados para el nodo actuador cumplen con los requerimientos establecidos por el sistema, por lo tanto, no hay cambios que realizar en cuanto a hardware y software en la segunda iteración, culminando así con el número de iteraciones.

Mediante el test eléctrico se verificó que la placa de alimentación abastece de energía al nodo actuador aproximadamente 22 horas, en caso de ser necesario el módulo de carga TP5046 tiene incorporado un puerto micro USB para recargar la batería con 5 voltios.

La respuesta de la red neuronal ante el comportamiento de las variables ambientales es adecuada, es decir, si el dato enviado por el sensor de lluvia corresponde a ausencia de lluvia la red neuronal se activa, caso contrario el sistema no se activa. Además, se pudo evidenciar que la predicción realizada por la red neuronal es correcta.

La comunicación entre el nodo central y el nodo actuador es exitosa, ya que se encuentra ubicado a la misma distancia que el nodo sensor, comparten los mismos

parámetros de comunicación LoRa garantizando la transmisión y recepción de información.

Por otro lado, el comportamiento del nodo actuador es correcto, si la predicción de la red neuronal es “Regar” las electroválvulas se activan, caso contrario si la predicción de la red neuronal es “No Regar” las electroválvulas se desactivan.

CAPÍTULO IV. Implementación del Sistema

En este apartado implementa el sistema de riego por aspersión automatizado en la parcela de cultivo de pasto. El sistema se implementa en la etapa de rebrote; proceso por el cual el pasto crece luego de una defoliación. La velocidad de rebrote depende de varios factores, uno de ellos es el ambiente; particularmente la humedad del suelo y la temperatura. La exigencia de agua en una pastura que acaba de ser defoliada es mucho mayor, ya que inicia nuevamente su ciclo de vida.

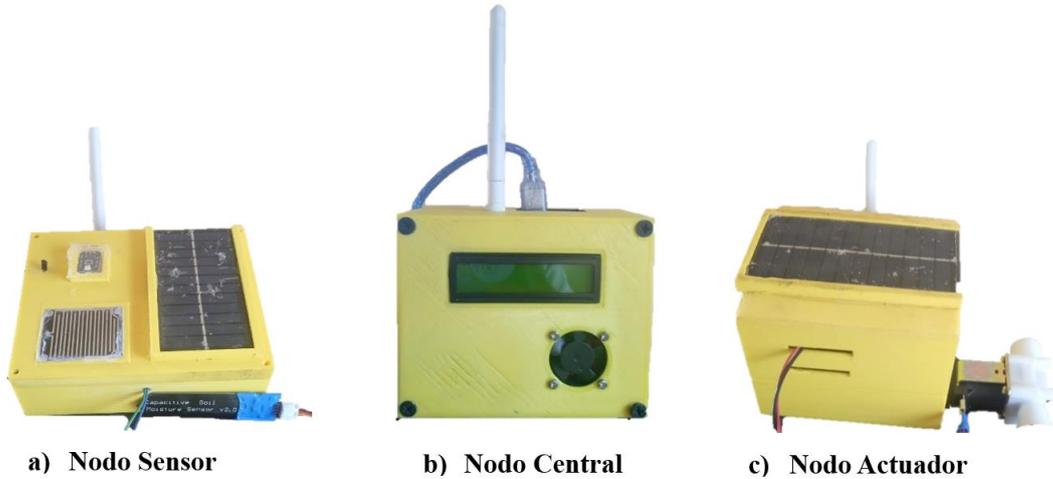
Posterior a eso, se analiza el consumo de agua haciendo una comparación con el sistema de riego antiguo (manual) y finalmente se realiza un análisis de la planta considerando parámetros como coloración y tamaño de las hojas, altura de la planta y tamaño de la raíz.

Tanto el nodo sensor, nodo central y nodo actuador están protegidos por una carcasa hermética protegiendo a los dispositivos de agentes externos como agua, calor y humedad, tal como se puede apreciar en la Figura 101.

El nodo sensor en su exterior cuenta con sensores de humedad del suelo, radiación UV, temperatura para el monitoreo del cultivo y un panel solar para recargar la batería. En el nodo central se puede observar un ventilador para evitar que la Raspberry Pi 3 se sobrecaliente y una LCD en el cual se puede verificar los datos recibidos; humedad del suelo, radiación UV, temperatura y estado del riego. Finalmente, el nodo actuador en su exterior cuenta con un panel solar y una electroválvula.

Figura 101

Nodos que conforman la red de sensores inalámbricos WSN



a) Nodo Sensor

b) Nodo Central

c) Nodo Actuator

Fuente: *Autoría*

El nodo central está ubicado a exteriores del hogar del usuario aproximadamente a 6 metros de altura. Mientras que el nodo sensor y el nodo actuador están ubicados en el cultivo de pasto a una altura de 1.5 metros cada uno y con una distancia de separación de 160 metros del nodo central.

Figura 102

Implementación de los nodos sensor, central y actuador



**a) Nodo Sensor
ubicado a 1.5 metros de altura**



**b) Nodo Central
ubicado a 6 metros de altura**



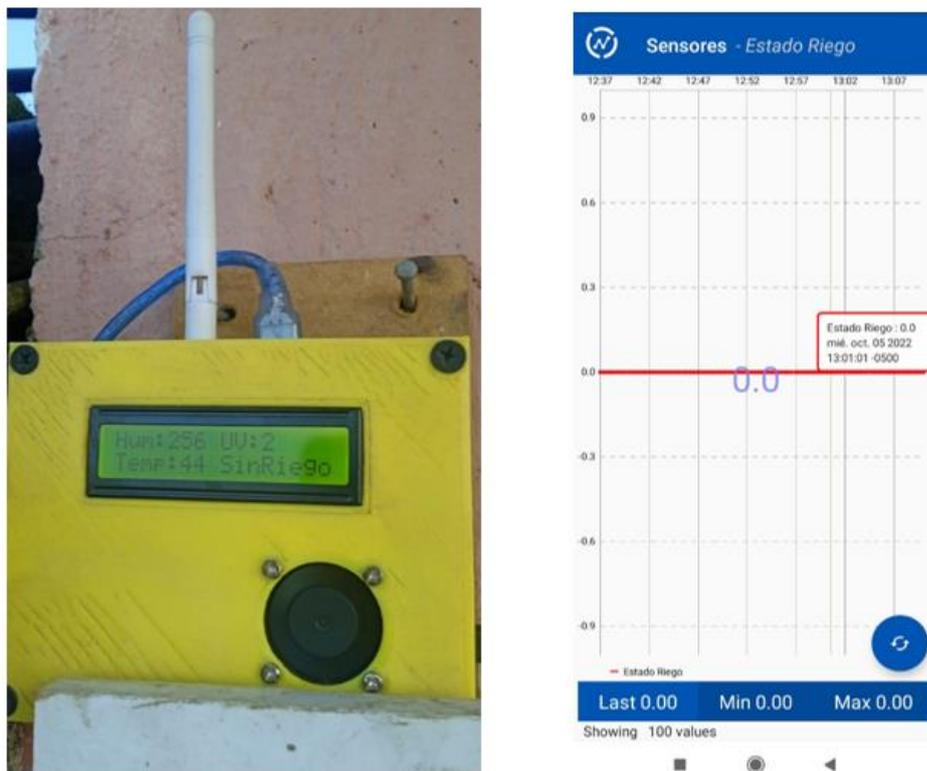
**c)Nodo Actuator
ubicado a 1.5 metros de altura**

Fuente: *Autoría*

En la LCD del nodo central se puede observar los parámetros enviados por el nodo sensor; humedad, radiación UV, temperatura y la predicción de la red neuronal o el estado del riego (riego/sin riego). Además, en la Figura 103 literal **b)** se muestra los datos en forma de gráficos en la App Móvil de ThingSpeak denominado, ThingView.

Figura 103

Visualización de datos recolectados por el nodo sensor y estado de riego



a) Visualización de datos en el nodo central b) Visualización de datos en Thing View

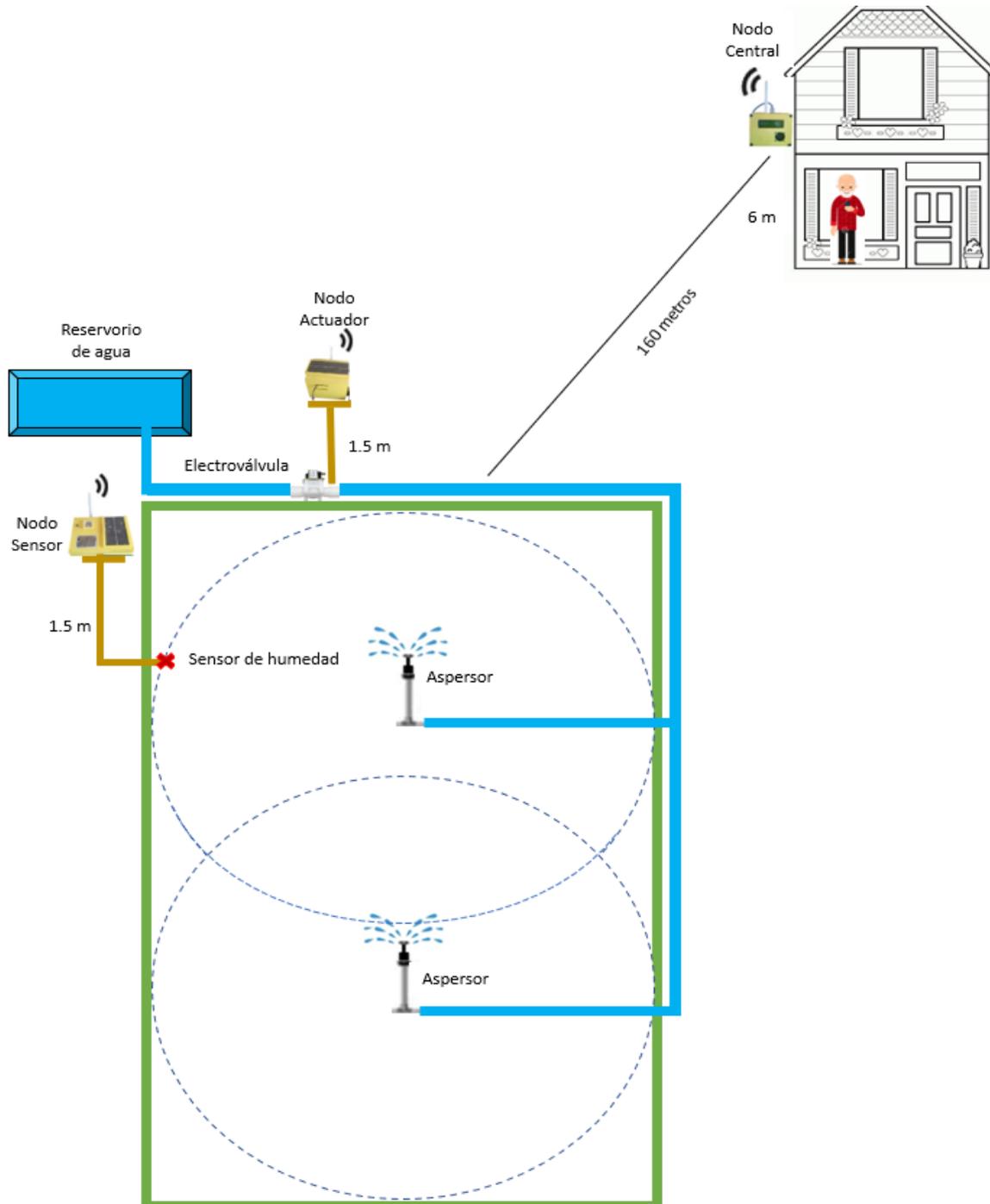
Nota: La gráfica del estado de riego muestra una línea recta de color rojo en el valor 0.0 que representa el estado de riego, en este caso, sin riego. **Fuente:** Autoría

En la Figura 104 se puede apreciar cómo se realiza la ubicación de cada uno de los nodos. Se inicia visualizando el número de aspersores que se encuentran en la parcela, cada aspersor cubre un radio de 9m. El nodo sensor se ubica fuera del radio de riego para que el sensor de lluvia no detecte las gotas del aspersor como lluvia. Sin embargo, el sensor de humedad del suelo debe estar en el límite del radio, para el sensado de la variable de humedad.

El nodo central está ubicado en el hogar del usuario, protegido del cualquier agente externo. Además, para que el usuario pueda visualizar los datos de las variables ambientales y el estado de riego directamente en la pantalla LCD.

Figura 104

Ubicación de los nodos en la parcela



Fuente: Autoría

Finalmente, el nodo actuador está ubicado en la tubería secundaria de la parcela, la electroválvula activa y desactiva el flujo de agua de acuerdo con la predicción de riego enviada por nodo central.

4.1. Análisis del Consumo de Agua al Implementar el Sistema Automatizado

El sistema se implementa en una parcela de aproximadamente 1000 m^2 de extensión, en esta parcela se encuentran ubicados dos aspersores rotativos, con el sistema de riego manual y el caudal disponible. El usuario manifiesta que los aspersores trabajan tres horas sin interrupciones, asumiendo que en ese tiempo el suelo queda completamente húmedo; si el suelo está completamente seco el usuario mantiene los aspersores en funcionamiento hasta 5 horas.

Para verificar que el sistema implementado mejora la eficiencia del sistema de riego por aspersión, reduciendo el consumo de agua al irrigar el cultivo de pasto se hace un seguimiento de los dos sistemas. En el sistema de riego manual se calcula el consumo de agua en el transcurso de tres y cinco horas respectivamente, asumiendo la experiencia del usuario.

En el sistema de riego automatizado por la red neuronal se calcula el tiempo y el consumo de agua desde que la red predice irrigar el cultivo hasta que el sistema detenga el riego (cambio de un estado al otro), es decir hasta que el suelo este húmedo.

El tiempo de riego del sistema automatizado se obtuvo a través de la base de datos recuperada de ThingSpeak Anexo 7, donde se puede apreciar la hora exacta del cambio de estado, el decir se puede identificar la hora en la que inicia el riego y el momento en el que se detiene; calculando así el tiempo de riego.

Figura 105

Base de datos del estado de riego recuperado de ThingSpeak

BASE DE DATOS-ESTADO DE RIEGO					
Estado de riego	1	Regando			
	0	Sin riego			
	Field 1	Field 2	Field 3	Field 4	Field 5
Fecha	Humedad del suelo	Temperatura	Radiacion UV	Estado de Riego	Lluvia
2022-10-05T18:08:56	415	21	0	1	0
2022-10-05T18:10:23	416	21	0	1	0
2022-10-05T18:10:48	415	22	0	1	0
2022-10-05T18:11:07	415	23	0	1	0
2022-10-05T18:11:27	414	22	0	1	0
2022-10-05T18:11:48	415	21	0	1	0
2022-10-05T18:12:14	440	22	0	1	0
2022-10-05T18:12:34	453	21	0	1	0
2022-10-05T18:13:08	480	23	0	1	0
2022-10-05T18:15:12	482	22	0	1	0
2022-10-05T18:16:33	483	22	0	1	0
2022-10-05T18:17:59	484	56	0	1	0
2022-10-05T18:19:25	484	21	0	1	0
2022-10-05T18:20:45	480	21	0	1	0
2022-10-05T18:22:12	479	21	0	1	0
2022-10-05T18:25:38	477	21	0	1	0
2022-10-05T18:26:57	451	22	0	1	0
2022-10-05T18:27:17	472	22	0	1	0
2022-10-05T18:29:37	463	21	0	1	0
2022-10-05T18:32:58	439	21	0	1	0
2022-10-05T18:33:19	430	21	0	1	0
2022-10-05T18:33:43	450	22	0	1	0
2022-10-05T18:34:12	467	15	0	1	0
2022-10-05T18:34:32	462	15	0	1	0
2022-10-05T18:35:00	451	15	0	1	0
2022-10-05T18:35:24	440	15	0	1	0
2022-10-05T18:35:51	410	15	0	1	0
2022-10-05T18:36:14	405	15	0	1	0
2022-10-05T18:37:37	400	15	0	1	0
2022-10-05T18:38:04	398	16	0	1	0
2022-10-05T18:39:28	398	21	0	1	0
2022-10-05T18:40:48	395	22	0	1	0
2022-10-05T18:41:23	391	21	0	1	0
2022-10-05T18:43:27	391	18	0	1	0
2022-10-05T18:44:14	385	16	0	1	0
2022-10-05T18:45:34	364	15	0	1	0
2022-10-05T18:46:28	341	16	0	0	0

Fuente: Autoría

Como se puede observar en la Figura 105, el sistema automatizado con la red neuronal activa el riego (1) a las 18:08 y detiene el riego (0) a las 18:46, es decir, el

riego dura menos de una hora cuando el suelo está seco. Cuando el suelo está muy seco el riego dura más de 1 hora, Anexo 7.

En la Tabla 33 se dan a conocer los resultados obtenidos del sistema de riego manual Vs riego automatizado con la red neuronal.

Tabla 33

Consumo de agua en el sistema de riego manual Vs sistema automatizado con la red neuronal

	SISTEMA DE RIEGO MANUAL		SISTEMA DE RIEGO AUTOMATIZADO	
	Tiempo de Riego (horas)	Consumo de agua (m³)	Tiempo de Riego (horas)	Consumo de agua (m³)
Día 1	5:00	5.4	1:30	1.62
Día 2	3:00	3.24	1:36	1.72
Día 3	0:00	0	0:55	0.99
Día 4	5:00	5.4	1:35	1.7
Día 5	3:00	3.24	1:36	1.72
Día 6	3:00	3.24	1:35	1.7
Día 7	0:00	0	2:11	2.35
Día 8	3:00	3.24	0:00	0
Día 9	5:00	5.4	1:50	1.98
Día 10	0:00	0	2:28	2.66
Día 11	5:00	5.4	0:00	0
Día 12	3:00	3.24	2:04	2.23
Día 13	3:00	3.24	2:17	2.46
Día 14	0:00	0	0:00	0
Día 15	5:00	5.4	1:42	1.83
TOTAL		46,44		22,03

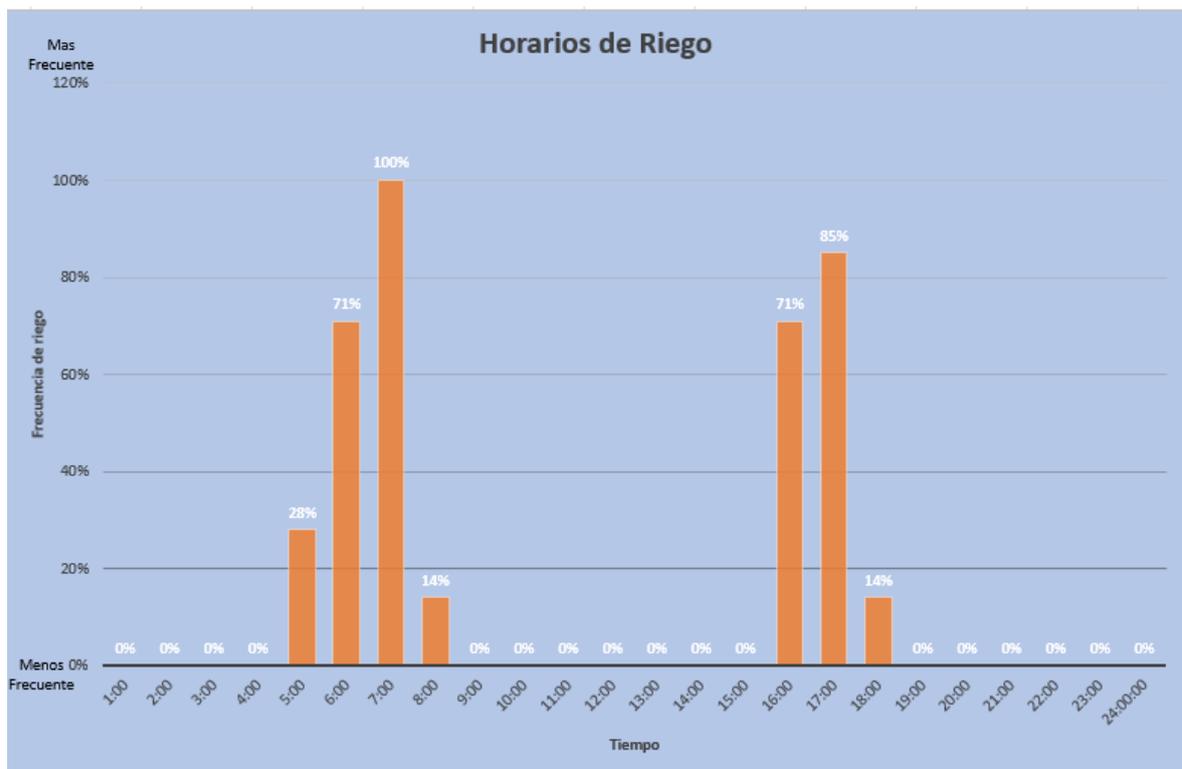
Fuente: Autoría

Con los resultados obtenidos se puede evidenciar que el consumo de agua con el sistema de riego manual en el transcurso de 15 días es de 46.44 m^3 . Por el contrario, en el mismo periodo de tiempo el sistema de riego automatizado con la red neuronal consume un 22.89 m^3 , por debajo de la mitad del sistema manual.

También se puede evidenciar que los horarios de riego son con frecuencia en horas de la mañana y al atardecer. El riego es más frecuente a las 7 de la mañana y a las 5 de la tarde, seguido a las 6 de la mañana y 4 de la tarde y menor frecuencia a las 8 de la mañana y 6 de la tarde. Claro, esto varía teniendo en cuenta la variable de radiación UV.

Figura 106

Horario de riego más y menos frecuente del sistema



Fuente: Autoría

4.2. Análisis del Cultivo de Pasto al Implementar el Sistema Automatizado

El objetivo de esta prueba es evaluar el desarrollo del cultivo de pasto al implementar el sistema de riego automatizado con la red neuronal después de la etapa de defoliación.

Figura 107

Cultivo de pasto en etapa de defoliación.



Fuente: Autoría

Se realiza un análisis del desarrollo de la planta después de la implementación del sistema. Se observa el tamaño de la planta y coloración de hojas en el transcurso de 15 días, tiempo en que la planta recupera su área foliar. En la Figura 108 se puede observar dos plantas, la sección **a)** corresponde a una planta del cultivo que utiliza el sistema de riego automatizado y la sección **b)** cultivo de pasto que utiliza el riego manual.

Figura 108

Cultivo de pasto con el sistema de riego manual Vs sistema de riego automatizado con la red neuronal



Fuente: Autoría

Los resultados son evidentes, la dosificación adecuada de riego que brinda el sistema automatizado ha logrado la aparición de hojas nuevas alargadas, con un color verdoso y logrando un rebrote uniforme.

En cuanto al tamaño de la raíz, se puede observar en la Figura 109 sección **a)** que las raíces son delgadas, alargadas y abundantes, a diferencia de la sección **b)** que tiene una raíz pequeña y más gruesa que no ha profundizado lo suficiente. Esto debido a que los riegos muy largos producen encharcamientos y las raíces no pueden sujetarse al suelo.

Figura 109

Tamaño de la raíz sistema de riego manual Vs automatizado con red neuronal



Fuente: Autoría

Conseguir un buen rebrote es la finalidad de un cultivo de pasto, ya que esto conlleva a una buena producción, un buen aprovechamiento del ganado y a una alta persistencia del pasto en el transcurso tiempo.

4.3.Costo-Beneficio

En este apartado se detalla el costo y el beneficio del sistema de riego automatizado con la red de sensores y la red neuronal. Donde se detallan los componentes necesarios para el óptimo funcionamiento del mismo.

Los componentes de hardware que forman parte del sistema se dan a conocer en la Tabla 34.

Tabla 34

Costo de hardware del sistema

COMPONENTE	PRECIO UNITARIO (USD)	SUBTOTAL (USD)
Arduino Uno	11	33
Raspberry pi 3	60	60
Dragino LoRa Shield	45	135
Sensor de humedad del suelo capacitivo	8.25	8.25
Sensor de radiación UV S12SD	9	9
Sensor de temperatura LM35	3	3
Sensor de lluvia MH-RD YL-83	2.50	2.50
Micro SD de 32GB	16	16
Electroválvula solenoide de 1/2 pulgada	12.25	12.25
Modulo Relé	1.10	1.10
Panel Solar de 12V	6.50	13
Batería de litio 3.7 V	10	20
Modulo cargador de batería de litio TP4056	1.10	1.10
LCD 16x2	8.50	8.50
Modulo elevador de voltaje	3	3
Elementos varios	70	70
Carcasa	40	120
TOTAL		515.70

Fuente: Autoría

Y los componentes de software se da a conocer en la Tabla 35, del cual se obtiene un valor de \$0.00, esto debido a que las plataformas utilizadas son de código abierto (Open Source).

Tabla 35

Costo de hardware del sistema

COMPONENTE	PRECIO UNITARIO (USD)	SUBTOTAL (USD)
IDE Arduino	0	0
Python	0	0
Anaconda	0	0
Tonny	0	0
ThingSpeak	0	0
ThingView	0	0
TOTAL		0

Fuente: Autoría

Por lo tanto, el costo total para automatizar el riego aplicando redes de sensores y redes neuronales es de \$515.70, un valor bastante aceptable considerando la escalabilidad del sistema a grandes hectáreas. El módulo LoRa tiene un alcance mayor a 2 kilómetros, por lo tanto, la distancia para la comunicación entre nodos no será un inconveniente, simplemente se deberá aumentar el número de sensores de humedad del suelo y electroválvulas de acuerdo con las necesidades del usuario y del cultivo.

Sin olvidar el principal beneficio que es, reducir el consumo innecesario del recurso hídrico, garantizando la humedad necesaria al cultivo y lograr una buena producción.

CAPÍTULO V. Conclusiones, Recomendaciones y Trabajo a Futuro

En este apartado final se detallan las conclusiones y recomendaciones encontradas en el desarrollo de los cuatro capítulos del proyecto, es un capítulo corto, pero se dan a conocer los resultados de la investigación. A partir de estas conclusiones se pueden tomar criterios para investigaciones futuras y continuar con la implementación de dispositivos tecnológicos e inteligencia artificial en el campo de la agricultura y ganadería.

5.1. Conclusiones

- Por medio de la revisión bibliográfica se estableció los parámetros que influyen en el desarrollo del cultivo de pasto en el cantón Pedro Moncayo, con esta información se determinó los sensores necesarios para el monitoreo de cada tipo de variable.
- El sistema de riego automatizado es una aplicación electrónica en base a hardware y software libre, un sistema viable para la implementación en cultivos de pasto. Actualmente los criterios de riego por aspersión únicamente se basan en la percepción humana; en base a las experiencias obtenidas por los usuarios. Es por esto que, la medición de condiciones ambientales en los cultivos de pasto es un buen inicio de la agricultura y ganadería de precisión en el cantón Pedro Moncayo.
- Se diseñó una red de sensores inalámbricos WSN LoRa que permite el monitoreo de variables ambientales que influyen en el desarrollo del cultivo de pasto desde cualquier lugar a través de un dispositivo electrónico con acceso a internet. A través de las mediciones realizadas por los nodos sensores se generan históricos y gráficas que servirán como antecedentes para futuros estudios si el usuario así lo requiere.

- Debido a que el usuario no está muy familiarizado con la cultura digital, y uno de los requerimientos es la presentación de resultados claros y precisos, se hace uso del software ThingView, una aplicación móvil con conexión a internet que permite acceder a los canales de ThingSpeak a través de una interfaz amigable con el usuario y los resultados se presentan a través de valores y gráficas fáciles de comprender. Sin embargo, si el usuario no tiene acceso a internet los datos pueden ser visualizados a través de la pantalla LCD que dispone el nodo central en el domicilio del usuario.
- La implementación de una red de sensores inalámbricos con tecnología LoRa tiene una gran versatilidad ya que actualmente existen diversos fabricantes con soluciones que se ajustan a un presupuesto, objetivos específicos y alcance. A pesar de que el sistema se implementó en una parcela donde la distancia entre los nodos sensores y la puerta de enlace es menor a un kilómetro, la tecnología LoRa ofrece una cobertura mayor a 2 kilómetros, por lo que la implantación en grandes hectáreas no sería un inconveniente.
- Se automatiza un sistema de riego que utiliza el método de irrigación por aspersión aplicando un modelo de redes neuronales artificiales para la predicción del riego en cultivos de pasto.
- Para un modelo de red neuronal con una capa de entrada con dos neuronas, tres capas ocultas con cien neuronas cada una y una capa de salida, el número de épocas o iteraciones suficientes para el entrenamiento es de 5000. Con este número de épocas la red neuronal aprende correctamente y se evita el sobreentrenamiento.
- De acuerdo con las predicciones realizadas por las redes neuronales, el sistema de riego por aspersión irriga el cultivo la mayor parte del tiempo al amanecer, al

atardecer. Esto permite que el agua se filtre mejor en la tierra, ya que, en presencia directa del sol provoca evaporación y el riego no es tan efectivo.

- Se puede evidenciar que el sistema de riego por aspersión manual consume alrededor $46,44 \text{ m}^3$. Por el contrario, en el mismo periodo de tiempo el sistema de riego automatizado consume alrededor de $22,03 \text{ m}^3$, una diferencia de $15,89$ milímetros cúbicos. Está claro que, el sistema de riego automatizado con una red neuronal artificial mejora en gran medida la eficiencia de los sistemas de riego que utilizan el método por aspersión.
- A pesar de que el crecimiento y desarrollo de un cultivo de pasto no depende simplemente del agua, sino, además, del tipo y nutrientes que contiene el suelo, se puede afirmar que una adecuada distribución de agua en el cultivo puede contribuir en desarrollo de la planta.

5.2. Recomendaciones

- Se debe tomar en cuenta que la puerta de enlace debe contar con una buena conexión a internet para cumplir un eficiente enrutamiento de paquetes y los datos se puedan visualizar en la nube.
- Para la predicción del riego se puede incorporar más variables como el viento y neblina ya que son factores a considerar en un sistema de riego que utiliza el método de aspersión.
- Debido a que los nodos sensor y actuador utilizan una batería de litio para su alimentación se recomienda realizar una carga inicial a través del puerto micro USB.
- Hay que considerar que exista línea de vista entre el nodo final y la puerta de enlace ya que la influencia de algún obstáculo en la red puede debilitar la señal

transmitida; esto puede llevar a un rediseño de la red durante el proceso de desarrollo.

- Los datos de entrenamiento de una red neuronal deben ser cuidadosamente seleccionados ya que las redes neuronales aprenden de los datos a los que están expuestas; de estos datos dependerá el correcto aprendizaje de la red para realizar predicciones.
- Hay que tomar en cuenta que al crear una red neuronal es difícil definir el número de capas que conformara la capa oculta y el número de neuronas en cada capa. Esto se convertirá en un proceso de ensayo y error hasta determinar el número correcto.
- Al igual que determinar el número de neuronas y capas ocultas es un poco complicado, conocer el número de épocas o iteraciones también es un proceso de prueba y error. Se debe generar un número adecuado de iteraciones evitando que no sea muy bajo ni muy alto para evitar el sobre entrenamiento.

5.3. Trabajo a Futuro

En la plataforma ThingSpeak se puede visualizar los datos de las variables ambientales recolectadas por el nodo sensor y el estado del riego a través de gráficas. Esta plataforma genera una base de datos que puede ser utilizada para diseñar un software que dé a conocer a través de una interfaz gráfica un histórico del riego. Es decir, las veces que el sistema irriga el cultivo de pasto, el consumo de agua diario, el horario de riego, etc.

Estos datos pueden llegar a ser muy importantes para las personas interesadas en hacer un estudio antes de la implementación de sistemas de riego a gran escala o para personas que quieran llevar un control diario del riego de su cultivo.

REFERENCIAS

- Abdollah Ghasemi, Ali Abedi, F. G. (n.d.). Propagation Engineering in Wireless Communications. <https://www.ptonline.com/articles/how-to-get-better-mfi-results>
- Acevedo, E., Serna, A., & Serna, E. (2017). Principios y características de las redes neuronales artificiales. *Desarrollo e Innovación En Ingeniería*, March 2019, 173–183.
https://www.researchgate.net/profile/Edgar_Serna_M/publication/331385353_Desarrollo_e_innovacion_en_ingenieria_ed_2/links/5c76e4ce92851c69504663b5/Desarrollo-e-innovacion-en-ingenieria-ed-2.pdf#page=379
- Ahmadi, J. (2016). The effects of Fresnel Zone in communication theory based on radio waves. In *Bulletin de la Société Royale des Sciences de Liège* (Vol. 85).
- Ballesteros de Andrés, C., Sypko, D., Ballesteros de Andrés, C., & Sypko, D. (2017). Sistema IoT para monitorización del estado de un centro de proceso de datos de grandes dimensiones.
- Benítez David, & Anías, C. (2016). Propuesta de Arquitectura Para Internet de las Cosas. 18 Convención Científica de Ingeniería y Arquitectura, November 2016, 1–12. <https://www.researchgate.net/publication/320353907>
- Berzal, F. (2018). Redes Neuronales. 753. <https://deep-learning.ikor.org>
- Bonetto, R. (2021). Machine learning. In *Computing in Communication Networks*. Elsevier Inc. <https://doi.org/10.1016/B978-0-12-820488-7.00021-9>
- Castro, M., & Espin, F. (2016). Análisis de los factores que determinan la sostenibilidad y sustentabilidad de la economía social y solidaria para la crianza y comercialización de vacuno en pie y faenado en los cantones Quito, Cayambe y Pedro Moncayo. Tesis, 1, 17.
<http://dspace.ups.edu.ec/bitstream/123456789/5081/1/UPS-CYT00109.pdf>
- Chacón Carlos, P. (2015). Cultivo de pastos. SN Power Proyecto Cheves, 1, 40. https://www.swisscontact.org/fileadmin/user_upload/COUNTRIES/Peru/Documents/Publications/MANUAL_PASTOS_CULTIVADOS.pdf
- Deshpande, P., Damkonde, A., & Chavan, V. (2017). The Internet of Things: Vision, Architecture and Applications. *International Journal of Computer Applications*, 178(2), 1–14. <https://doi.org/10.5120/ijca2017915722>
- Dragino Technology Co. LTD. (2019). Arduino Shield featuring LoRa® technology. Dragino Technology Development Co.LTD. <https://www.dragino.com/products/lora/item/102-lora-shield.html>
- Edgar Solera. (2018). Modulación LoRa: Long Range Modulation. Medium, 1–10. <https://medium.com/pruebas-de-laboratorio-de-la-modulación-lora/modulación-lora-4ad74cabd59e>

- Ertürk, M. A., Aydın, M. A., Büyükakkaşlar, M. T., & Evirgen, H. (2019). A Survey on LoRaWAN Architecture, Protocol and Technologies. *Future Internet*, 11(10), 216. <https://doi.org/10.3390/fi11100216>
- FAO. (2011). Establecimiento y manejo de pasturas para ganado tipo lechero. <http://www.fao.org/3/a-bc982s.pdf>
- FAO. (2016). Factores que se deben cosiderar para seleccionar el sistema de riego mas adecuado. 4(1), 64–75.
- Fernández, R., Ordieres, J., Martinez de Pisón, F. J., González, A., Alba, F., Lostado, R., & Pernia, V. (2009). Redes inalámbricas de sensores: teoria y aplicacion practica.
- Garcia, M., & Mora, J. (2010). Guía para la Implementación de Redes de Sensores Inalámbricos.
- Garg, P., Saroha, K., & Lochab, R. (2011). Review of Wireless Sensor Networks-Architecture and Applications. *International Journal of Computer Science and Management Studies*, 11(01), 34–38.
- Hamami, L., & Nassereddine, B. (2020). Application of wireless sensor networks in the field of irrigation: A review. *Computers and Electronics in Agriculture*, 179(April), 105782. <https://doi.org/10.1016/j.compag.2020.105782>
- Herrera, J., Hernández, P., & Castillo, J. (2013). Manual Para Capacitación De Operadores. www.imta.gob.m
- Hidalgo, K., Chontasi, L., Jaramillo, J., Guerreo, R., Gallardo, J., Robalino, L., & Quiroz, H. (2018). Plan de Desarrollo y Ordenamiento Territorial-Pedro Moncayo 2018-2025. In Documento ejecutivo para autoridades provinciales.
- Hormozi, H., Hormozi, E., & Nohooji, H. R. (2012). The Classification of the Applicable Machine Learning Methods in Robot Manipulators. *International Journal of Machine Learning and Computing*, 2(5), 560–563. <https://doi.org/10.7763/ijmlc.2012.v2.189>
- International Electrotechnical Commission. (2016). Internet of Things: Wireless Sensor Networks. IEC, 78. <http://www.iec.ch/whitepaper/pdf/iecWP-internetofthings-LR-en.pdf>
- Izquierdo, D., & Mantilla, N. (2015). Efecto de la intensidad lumínica y precipitación sobre el crecimiento del pasto brachiaria brizantha cv toledo en piedemonte llanero. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699.
- Kais Mekki, Eddy Bajic, Frederic Chaxel, F. M. (2018). A comparative study of LPWAN technologies for large-scale IoT deployment. In ScienceDirect.
- Kakao Corp. (2021). Arduino Sensors for Everyone GUVA-S12SD Ultraviolet Light Sensor.

- Laguna, D. (2019). El reto de la ganadería: Implementación de las TICS. Universidad de Los Andes. <https://agronegocios.uniandes.edu.co/2019/10/04/el-reto-de-la-ganaderia-implementacion-de-las-tics/>
- Leon, R., Bonifaz, N., & Guitierrez, F. (2018). Pastos y Forrajes del Ecuador.
- López, J., Soto, F., Suardíaz, J., & Iborra, A. (2016). Red de Sensores Inalámbrica para Agricultura de Precisión. *Telecoforum*, red 3, 3–4. <http://repositorio.bib.upct.es/dspace/handle/10317/876%5Cnhttp://hdl.handle.net/10317/876%5Cnhttp://repositorio.bib.upct.es/dspace/handle/10317/322>
- LoRa Alliance. (2022). Especificación LoRaWAN. 1–3.
- MAGAP. (2014). Manual de riego parcelario. Organización De Las Naciones Unidas Para La Agricultura Y La Alimentación, cap 1, 1–25.
- Miles, B., Bourennane, E. B., Boucherkha, S., & Chikhi, S. (2020). A study of LoRaWAN protocol performance for IoT applications in smart agriculture. *Computer Communications*, 164, 148–157. <https://doi.org/10.1016/j.comcom.2020.10.009>
- Monte, C. (2019). Aplicación de redes neuronales en el cálculo de sobretensiones y tasa de contorneamientos. 141–232.
- Montecé, F., Cadena, D., Alcívar, A., Caicedo, O., & Ruíz, I. (2020). Actualidad de las tecnologías de la Información y Comunicación TIC'S en la producción agropecuaria. *Journal of Science and Research*, 5, 134–144.
- Nayibe, C. C., Tibaduiza, D., Aparicio, L., & Caro, L. (2011). Redes de sensores inalámbricas. *Journal of Pediatric and Adolescent Gynecology*, 24(6), 404–409. <http://www.ncbi.nlm.nih.gov/pubmed/22187499>
- Nettikadan, D., & Raj, S. (2018). Smart Community Monitoring System using Thingspeak IoT Platform. Article in *International Journal of Applied Engineering Research*, 13(October), 13402–13408.
- Osetskyi. (2017). Explicación de los modelos SDLC:agil, en cascada, en forma de V, espiral. <https://existek.com/blog/sdlc-models/>
- Palasi Lllallana, V. R. (2004). Modelos de desarrollo iterativos. In Universidad Francisco Gavidia (pp. 1–8). [http://www.redicces.org.sv/jspui/bitstream/10972/3320/1/Modelos de desarrollo iterativos.pdf](http://www.redicces.org.sv/jspui/bitstream/10972/3320/1/Modelos%20de%20desarrollo%20iterativos.pdf)
- Pano Azucena, A. D. (2015). Realización FPGA de una Red Neuronal Artificial para Reproducir Comportamiento Caótico. 97.
- Patel, K. K., & Patel, S. M. (2016). Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges. *International Journal of Engineering Science and Computing*, 6(5), 1–10.

<https://doi.org/10.4010/2016.1482>

Poursafar, N., Alahi, M. E., & Mukhopadhyay, S. (2017). Long-range wireless technologies for IoT applications: A review. *Proceedings of the International Conference on Sensing Technology, ICST, 2017-Decem*, 1–6. <https://doi.org/10.1109/ICSensT.2017.8304507>

RED IAPD. (2011). *Libro de actas del I Congreso en Investigación en Agricultura para el Desarrollo*.

Requena, K. E., Rozo, D. M., & Arévalo, J. E. (2017). Principios y características de las redes neuronales artificiales. *Desarrollo e Innovación En Ingeniería*, March 2019, 173–183.

https://www.researchgate.net/profile/Edgar_Serna_M/publication/331385353_Desarrollo_e_innovacion_en_ingenieria_ed_2/links/5c76e4ce92851c69504663b5/Desarrollo-e-innovacion-en-ingenieria-ed-2.pdf#page=379

Rocalba. (2015). Semillas forrajeras y pratenses. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699.

Santamaría, F., & Archila, D. (2013). Estado Del Arte De Las Redes De Sensores Inalámbricos. *Tecnología Investigación y Academia*, 1(2).

Santillán, O. (2018). Ganadería de precisión. *INCyTU*, 52(55).

Santillán, O., & Rentería, M. (2018). Agricultura de Presición. 52(55), 6. www.foroconsultivo.org.mx

Semtech. (2020). Semtech SX1276. In *Wireless and Sensing Products (Issue May)*. <https://www.semtech.com/products/wireless-rf/lora-core/sx1276>

SENPLADES. (2017). Plan Nacional del Buen vivir 2017-2021. *Educational Research*, 1, 150. <http://www.planificacion.gob.ec/wp-content/uploads/downloads/2017/07/Plan-Nacional-para-el-Buen-Vivir-2017-2021.pdf>

Sisinni, E., & Mahmood, A. (2021). *Wireless Communications for Industrial Internet of Things : The LPWAN Solutions*. 79–101.

The Thing Network. (2020). *Factores de propagación- La Red de las cosas*. 6–10.

Uppalapati, K. (2019). *Untangling the Potentials of IoT Protocols and Standards*. <https://www.kelltontech.com/kellton-tech-blog/internet-of-things-protocols-standards>

Zepeda Hernandez, J. Á., Aguilar Castillejos, A. E., Hernandez Sol, Á., & Salgado Gutiérrez, M. C. (2015). Metodología para determinar la topología, patrones de parámetros eléctricos y entrenamiento de una Red Neuronal Artificial para un control inteligente. *Revista Tecnología Digital*, 5(1), 103–113. www.revistatecnologiadigital.com

ANEXOS

Anexo 1. Dragino LoRa Shield

Se da a conocer los componentes y pines que conforma el shield LoRa. Además, las especificaciones de consumo de energía.

Pin Mapping and Unused Pins

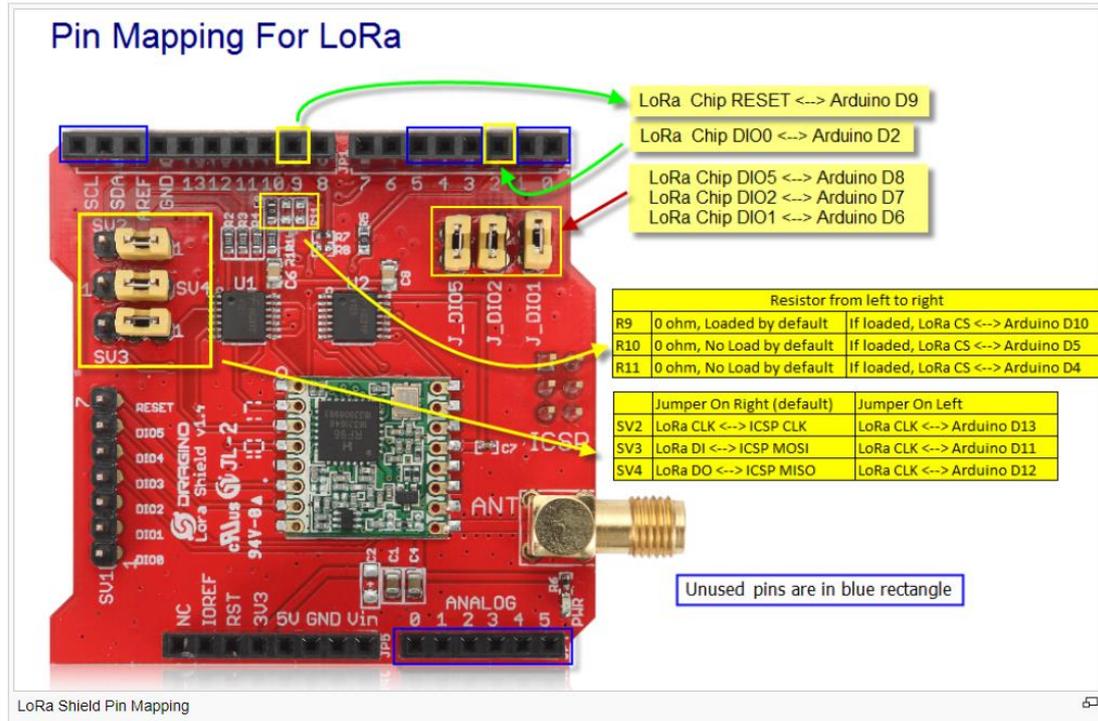


Table 6 Power Consumption Specification

Symbol	Description	Conditions	Min	Typ	Max	Unit
IDDSL	Supply current in Sleep mode		-	0.2	1	uA
IDDIDLE	Supply current in Idle mode	RC oscillator enabled	-	1.5	-	uA
IDDST	Supply current in Standby mode	Crystal oscillator enabled	-	1.6	1.8	mA
IDDFS	Supply current in Synthesizer mode	FSRx	-	5.8	-	mA
IDDR	Supply current in Receive mode	<i>LnaBoost</i> Off, band 1 <i>LnaBoost</i> On, band 1 Bands 2&3	- - -	10.8 11.5 12.0	- - -	mA
IDDT	Supply current in Transmit mode with impedance matching	RFOP = +20 dBm, on PA_BOOST RFOP = +17 dBm, on PA_BOOST RFOP = +13 dBm, on RFO_LF/HF pin RFOP = +7 dBm, on RFO_LF/HF pin	- - - -	120 87 29 20	- - - -	mA mA mA mA

Anexo 2. Hoja de especificaciones del Transceptor LoRa SX1276

El archivo completo se encuentra ingresando al siguiente enlace:

<https://www.mouser.com/datasheet/2/761/sx1276-1278113.pdf>



SX1276/77/78/79

WIRELESS & SENSING PRODUCTS

DATASHEET

1. General Description

The SX1276/77/78/79 incorporates the LoRa spread spectrum modem which is capable of achieving significantly longer range than existing systems based on FSK or OOK modulation. At maximum data rates of LoRa the sensitivity is 8dB better than FSK, but using a low cost bill of materials with a 20ppm XTAL LoRa can improve receiver sensitivity by more than 20dB compared to FSK. LoRa also provides significant advances in selectivity and blocking performance, further improving communication reliability. For maximum flexibility the user may decide on the spread spectrum modulation bandwidth (BW), spreading factor (SF) and error correction rate (CR). Another benefit of the spread modulation is that each spreading factor is orthogonal - thus multiple transmitted signals can occupy the same channel without interfering. This also permits simple coexistence with existing FSK based systems. Standard GFSK, FSK, OOK, and GMSK modulation is also provided to allow compatibility with existing systems or standards such as wireless MBUS and IEEE 802.15.4g.

The SX1276 and SX1279 offer bandwidth options ranging from 7.8 kHz to 500 kHz with spreading factors ranging from 6 to 12, and covering all available frequency bands. The SX1277 offers the same bandwidth and frequency band options with spreading factors from 6 to 9. The SX1278 offers bandwidths and spreading factor options, but only covers the lower UHF bands.

1.1. Simplified Block Diagram

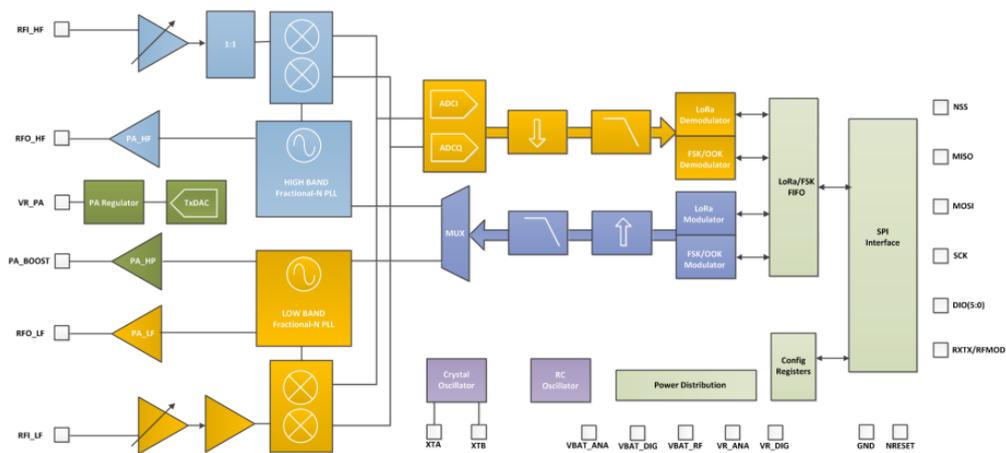


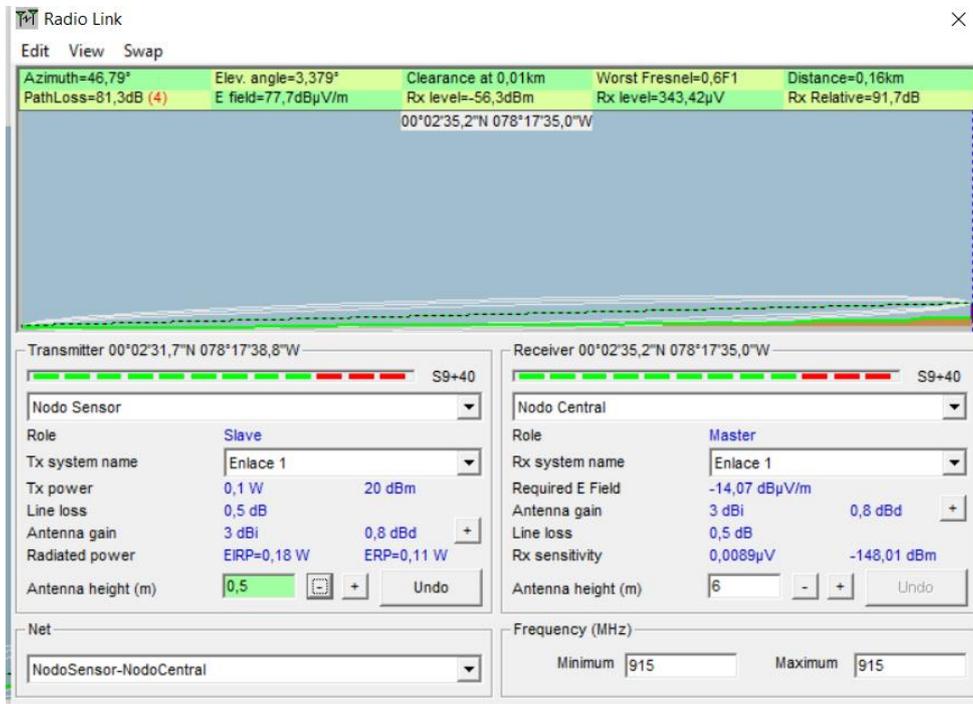
Figure 1. Block Diagram

1.4. Pin Description
Table 2 Pin Description

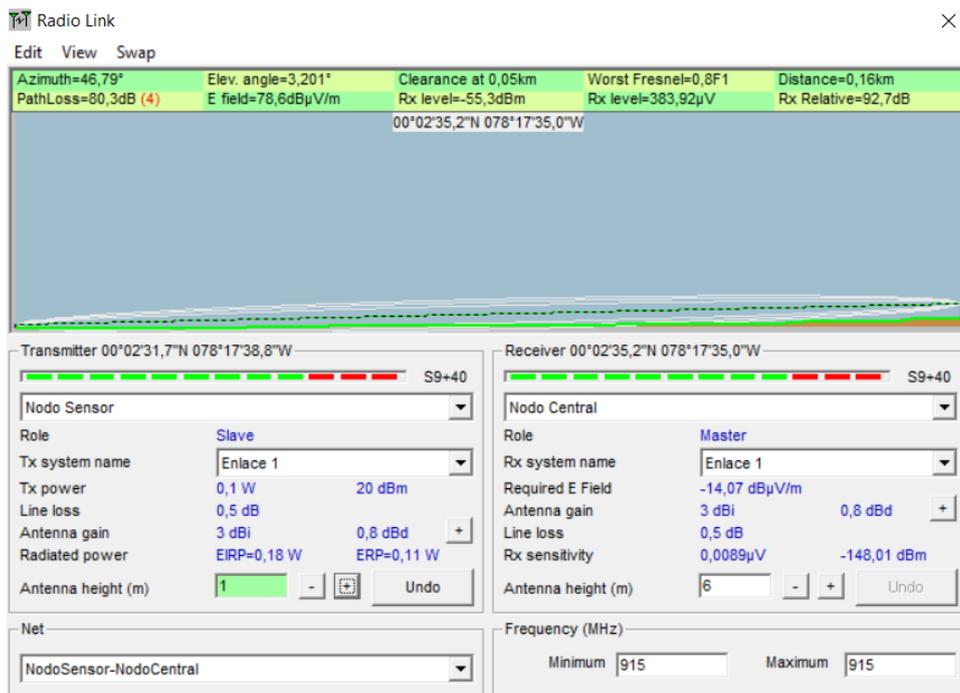
Number	Name	Type	Description
	SX1276/77/79/(78)	SX1276/77/79/(78)	SX1276/77/79/(78)
0	GROUND	-	Exposed ground pad
1	RFI_LF	I	RF input for bands 2&3
2	VR_ANA	-	Regulated supply voltage for analogue circuitry
3	VBAT_ANA	-	Supply voltage for analogue circuitry
4	VR_DIG	-	Regulated supply voltage for digital blocks
5	XTA	I/O	XTAL connection or TCXO input
6	XTB	I/O	XTAL connection
7	NRESET	I/O	Reset trigger input
8	DIO0	I/O	Digital I/O, software configured
9	DIO1/DCLK	I/O	Digital I/O, software configured
10	DIO2/DATA	I/O	Digital I/O, software configured
11	DIO3	I/O	Digital I/O, software configured
12	DIO4	I/O	Digital I/O, software configured
13	DIO5	I/O	Digital I/O, software configured
14	VBAT_DIG	-	Supply voltage for digital blocks
15	GND	-	Ground
16	SCK	I	SPI Clock input
17	MISO	O	SPI Data output
18	MOSI	I	SPI Data input
19	NSS	I	SPI Chip select input
20	RXTX/RF_MOD	O	Rx/Tx switch control: high in Tx
21	RFI_HF (GND)	I (-)	RF input for band 1 (Ground)
22	RFO_HF (GND)	O (-)	RF output for band 1 (Ground)
23	GND	-	Ground
24	VBAT_RF	-	Supply voltage for RF blocks
25	VR_PA	-	Regulated supply for the PA
26	GND	-	Ground
27	PA_BOOST	O	Optional high-power PA output, all frequency bands
28	RFO_LF	O	RF output for bands 2&3

Anexo 3. Simulaciones de Radio Enlace

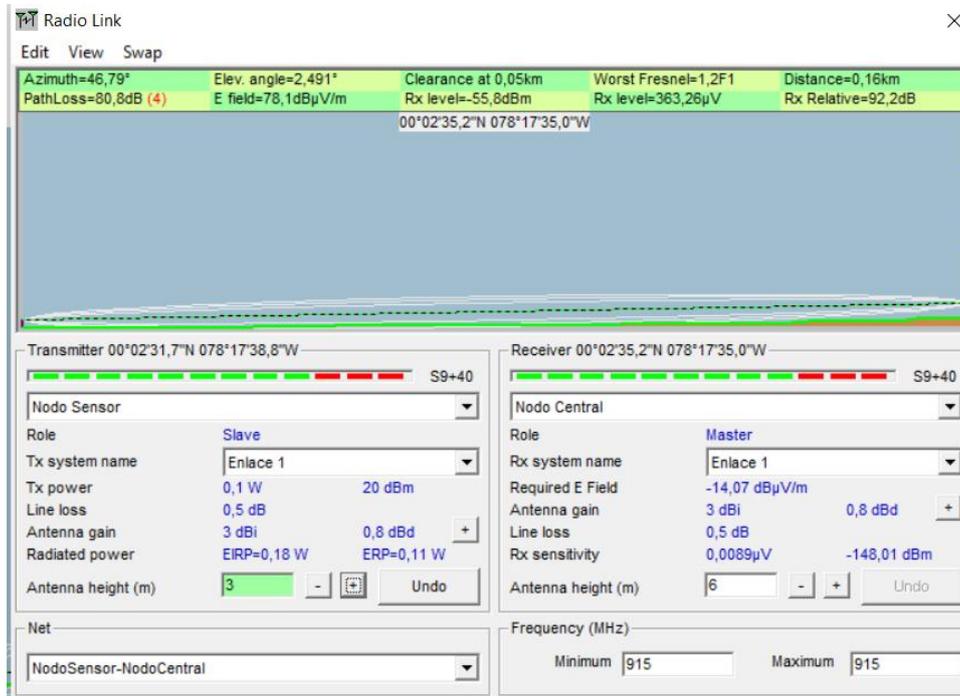
Radio Enlace entre el Nodo Sensor (altura de 0.5 metros) y el Nodo Central (altura de 6 metros)



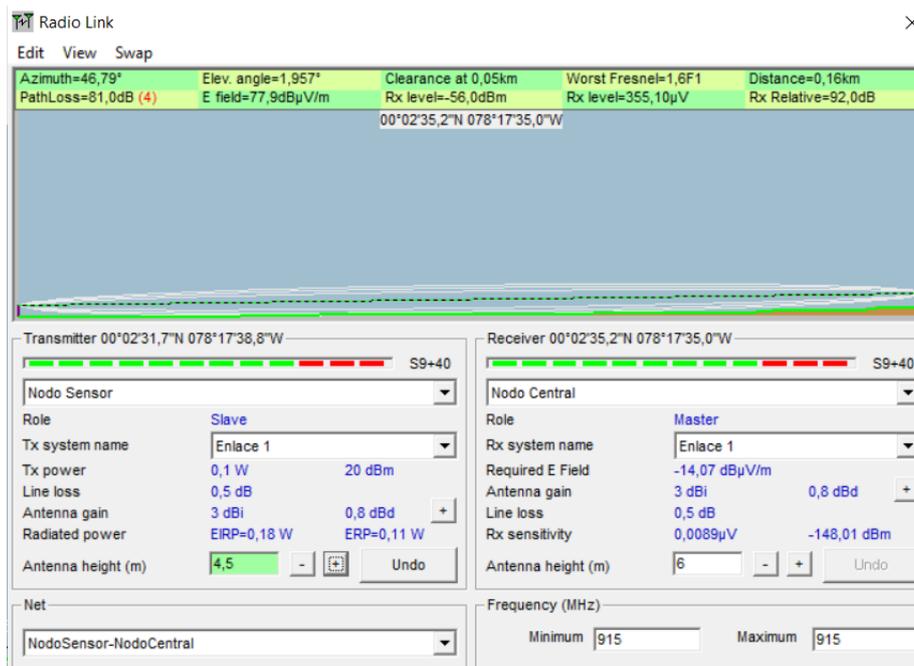
Radio Enlace entre el Nodo Sensor (altura de 1 metro) y el Nodo Central (altura de 6 metros)



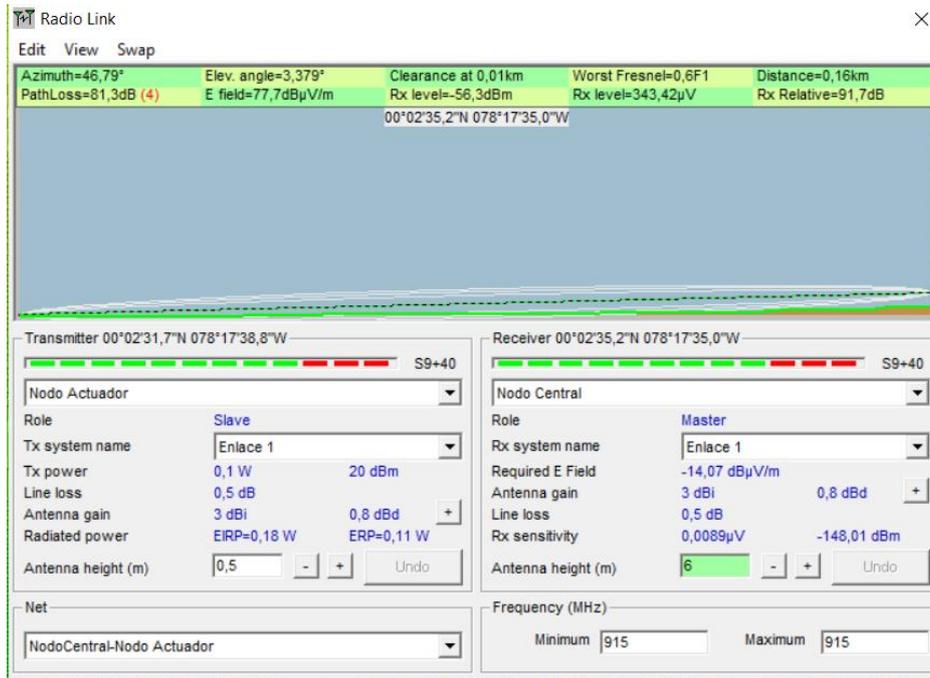
Radio Enlace entre el Nodo Sensor (altura de 3 metro) y el Nodo Central (altura de 6 metros)



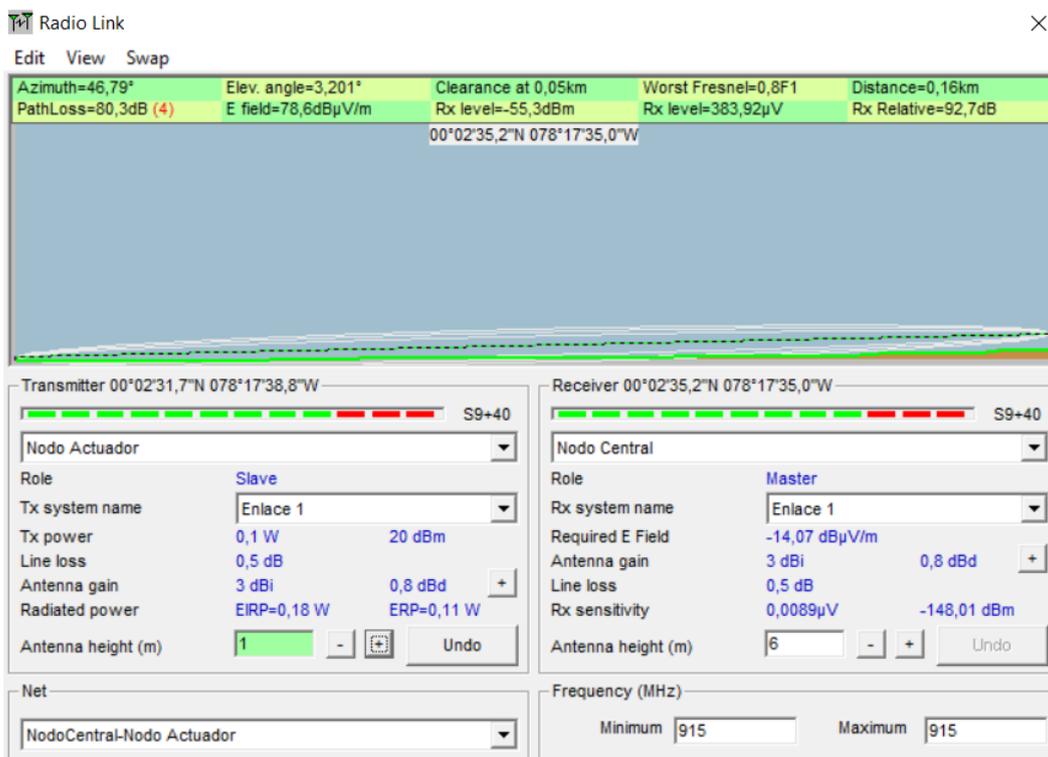
Radio Enlace entre el Nodo Sensor (altura de 4.5 metro) y el Nodo Central (altura de 6 metros)



Radio Enlace entre el Nodo Central (altura de 6 metros) y el Nodo Actuador (altura de 0.5 metros)



Radio Enlace entre el Nodo Central (altura de 6 metros) y el Nodo Actuador (altura de 1 metro)



Radio Enlace entre el Nodo Central (altura de 6 metros) y el Nodo Actuador (altura de 3 metros)

Radio Link [Close]

Edit View Swap

Azimuth=46,79°	Elev. angle=2,491°	Clearance at 0,05km	Worst Fresnel=1,2F1	Distance=0,16km
PathLoss=80,8dB (4)	E field=78,1dBμV/m	Rx level=-55,8dBm	Rx level=363,26μV	Rx Relative=92,2dB

00°02'35,2"N 078°17'35,0"W

Transmitter 00°02'31,7"N 078°17'38,8"W

Role: **Slave**

Tx system name: Enlace 1

Tx power: 0,1 W (20 dBm)

Line loss: 0,5 dB

Antenna gain: 3 dBi (0,8 dBd)

Radiated power: EIRP=0,18 W (ERP=0,11 W)

Antenna height (m): 3

Receiver 00°02'35,2"N 078°17'35,0"W

Role: **Master**

Rx system name: Enlace 1

Required E Field: -14,07 dBμV/m

Antenna gain: 3 dBi (0,8 dBd)

Line loss: 0,5 dB

Rx sensitivity: 0,0089μV (-148,01 dBm)

Antenna height (m): 6

Net: NodoCentral-Nodo Actuador

Frequency (MHz): Minimum 915, Maximum 915

Radio Enlace entre el Nodo Central (altura de 6 metros) y el Nodo Actuador (altura de 4.5 metros)

Radio Link [Close]

Edit View Swap

Azimuth=46,79°	Elev. angle=1,957°	Clearance at 0,05km	Worst Fresnel=1,6F1	Distance=0,16km
PathLoss=81,0dB (4)	E field=77,9dBμV/m	Rx level=-56,0dBm	Rx level=355,10μV	Rx Relative=92,0dB

00°02'35,2"N 078°17'35,0"W

Transmitter 00°02'31,7"N 078°17'38,8"W

Role: **Slave**

Tx system name: Enlace 1

Tx power: 0,1 W (20 dBm)

Line loss: 0,5 dB

Antenna gain: 3 dBi (0,8 dBd)

Radiated power: EIRP=0,18 W (ERP=0,11 W)

Antenna height (m): 4,5

Receiver 00°02'35,2"N 078°17'35,0"W

Role: **Master**

Rx system name: Enlace 1

Required E Field: -14,07 dBμV/m

Antenna gain: 3 dBi (0,8 dBd)

Line loss: 0,5 dB

Rx sensitivity: 0,0089μV (-148,01 dBm)

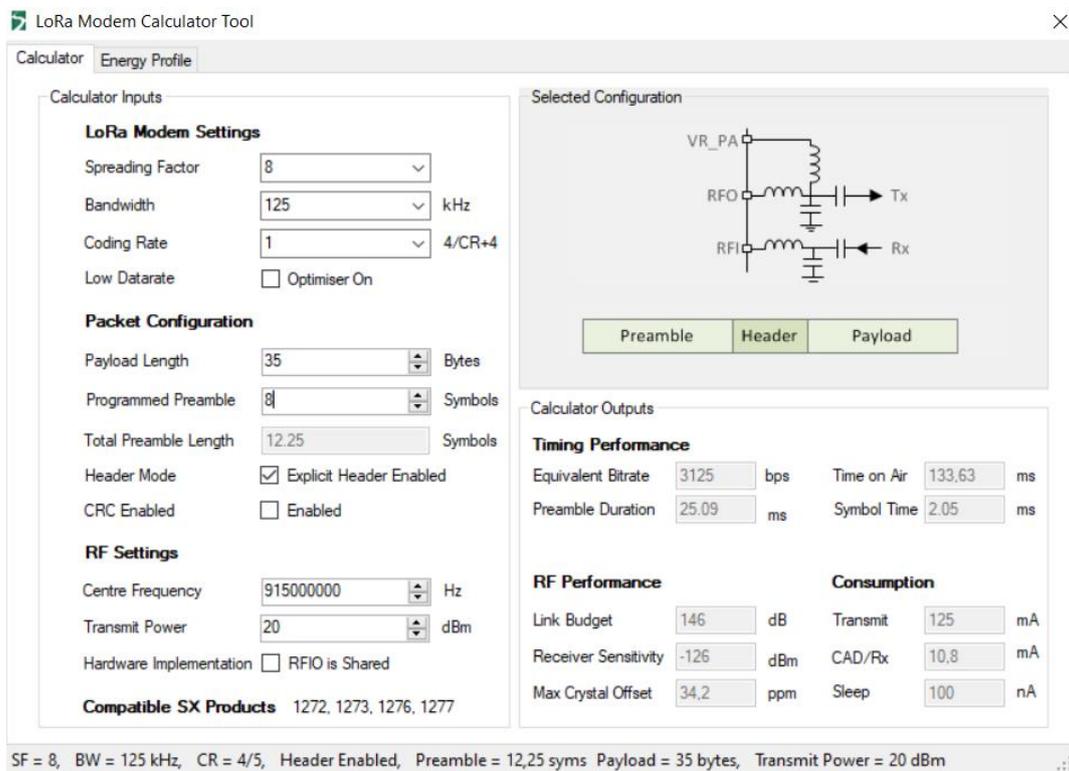
Antenna height (m): 6

Net: NodoCentral-Nodo Actuador

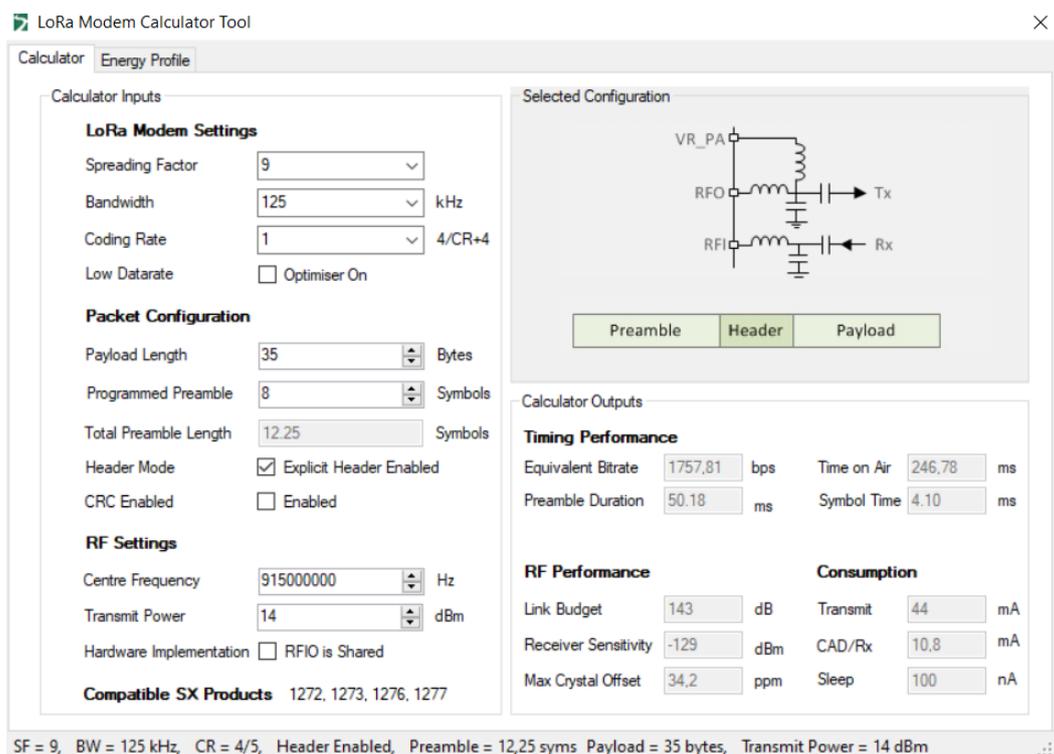
Frequency (MHz): Minimum 915, Maximum 915

Anexo 4. Cálculos del tiempo en el Aire de la transmisión

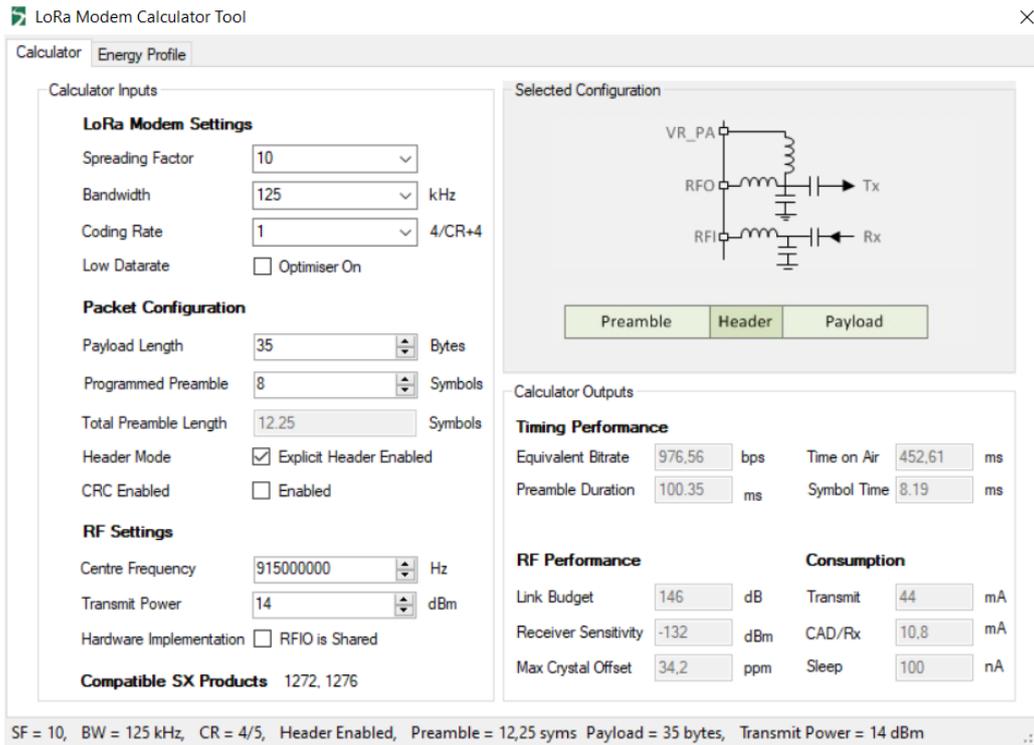
Cálculo del tiempo en el aire de la transmisión con un factor de propagación SF8



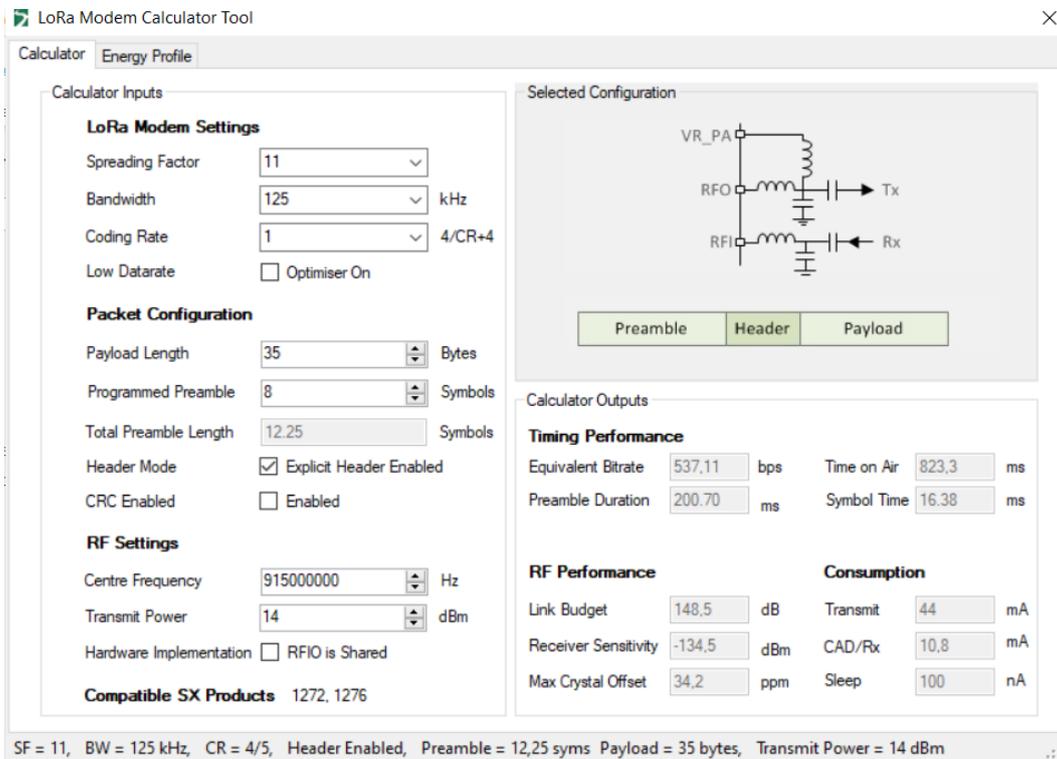
Cálculo del tiempo en el aire de la transmisión con un factor de propagación SF9



Cálculo del tiempo en el aire de la transmisión con un factor de propagación SF10



Cálculo del tiempo en el aire de la transmisión con un factor de propagación SF11



Cálculo del tiempo en el aire de la transmisión con un factor de propagación SF12

LoRa Modem Calculator Tool

Calculator
Energy Profile
×

Calculator Inputs

LoRa Modem Settings

Spreading Factor: 12

Bandwidth: 125 kHz

Coding Rate: 1 (4/CR+4)

Low Datarate: Optimiser On

Packet Configuration

Payload Length: 35 Bytes

Programmed Preamble: 8 Symbols

Total Preamble Length: 12.25 Symbols

Header Mode: Explicit Header Enabled

CRC Enabled: Enabled

RF Settings

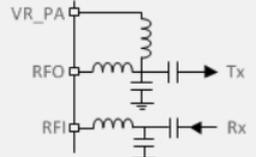
Centre Frequency: 915000000 Hz

Transmit Power: 14 dBm

Hardware Implementation: RFIO is Shared

Compatible SX Products 1272, 1276

Selected Configuration



Preamble Header Payload

Calculator Outputs

Timing Performance

Equivalent Bitrate	292.97 bps	Time on Air	1646.59 ms
Preamble Duration	401.41 ms	Symbol Time	32.77 ms

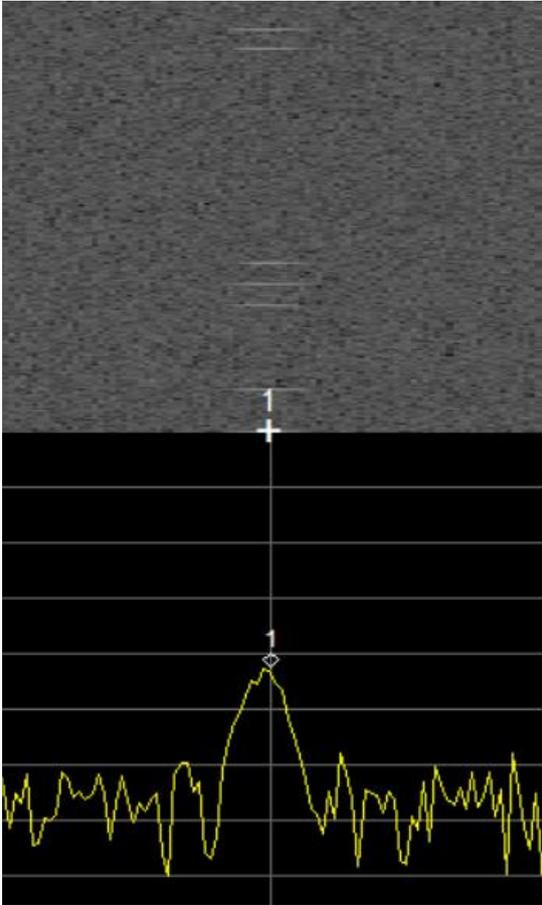
RF Performance

Link Budget	151 dB	Transmit	44 mA
Receiver Sensitivity	-137 dBm	CAD/Rx	10.8 mA
Max Crystal Offset	34.2 ppm	Sleep	100 nA

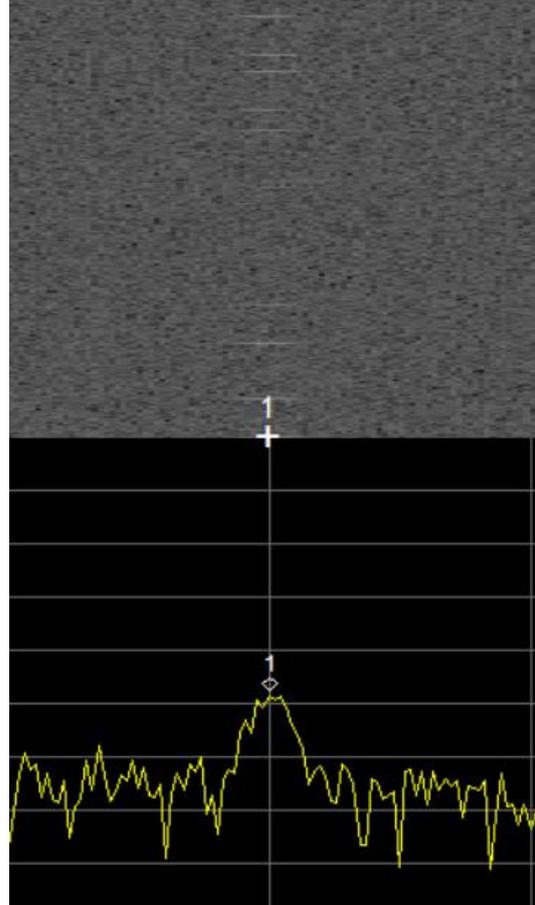
SF = 12, BW = 125 kHz, CR = 4/5, Header Enabled, Preamble = 12,25 syms Payload = 35 bytes, Transmit Power = 14 dBm

Anexo 5. Análisis del Espectro en Keyling Analyzer

Se establece a una frecuencia de 915 MHz , un ancho de banda 125 kHz y un factor de Propagación SF7 y SF8 respectivamente.

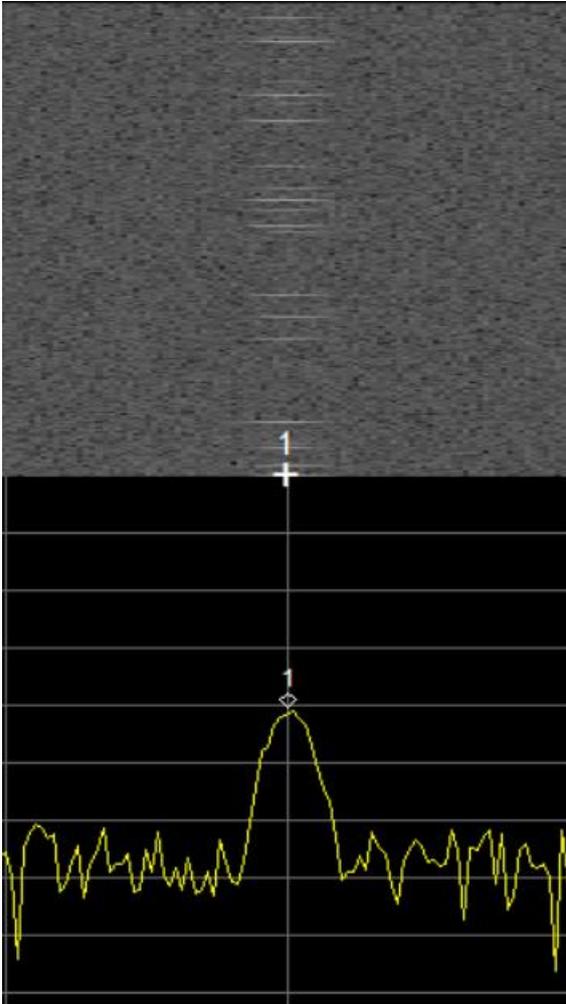


SF7

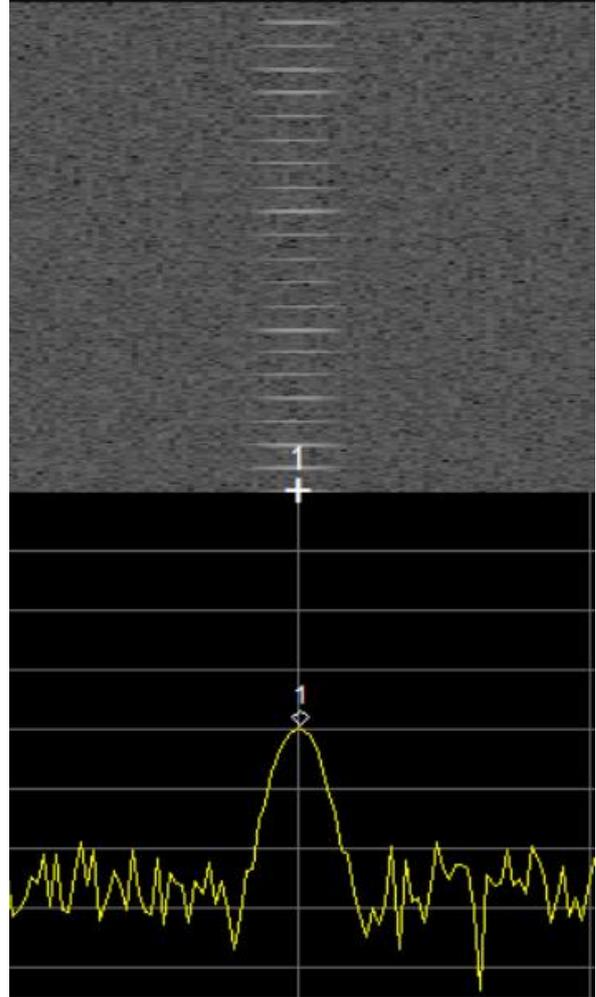


SF8

Se establece a una frecuencia de 915 MHz , un ancho de banda 125 kHz y un factor de Propagación SF9 y SF10.

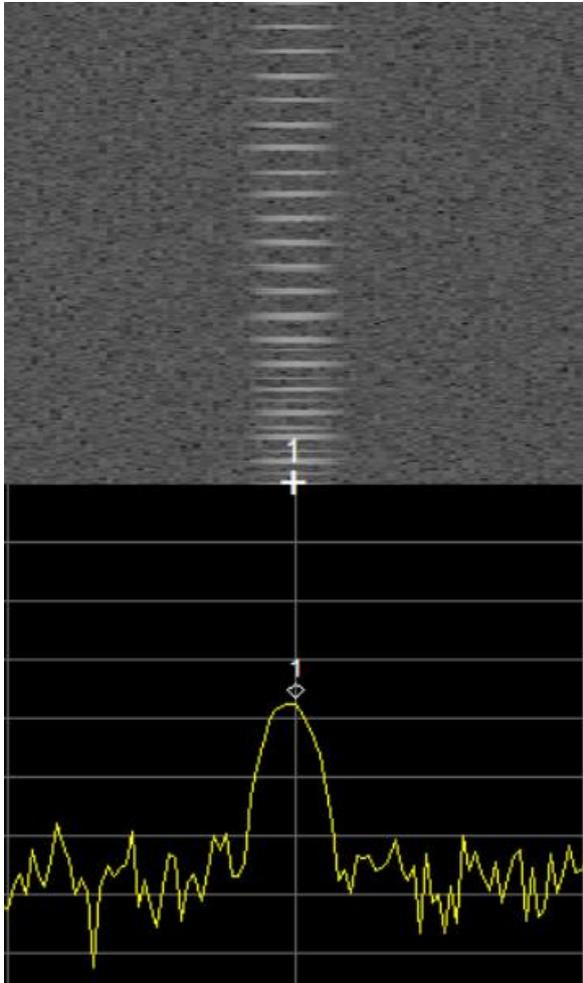


SF9

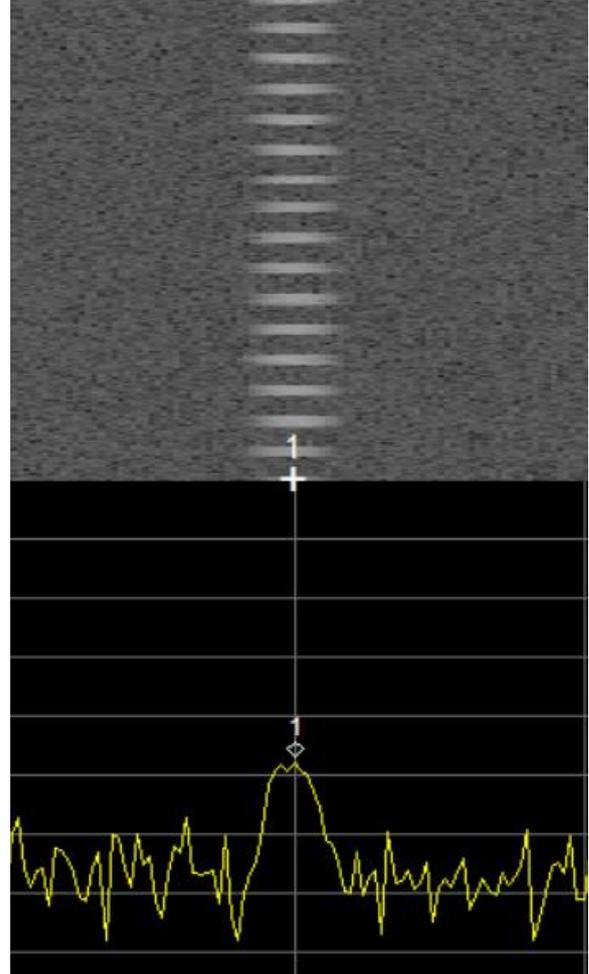


SF10

Se establece a una frecuencia de 915 MHz , un ancho de banda 125 kHz y un factor de Propagación SF11 y SF12.



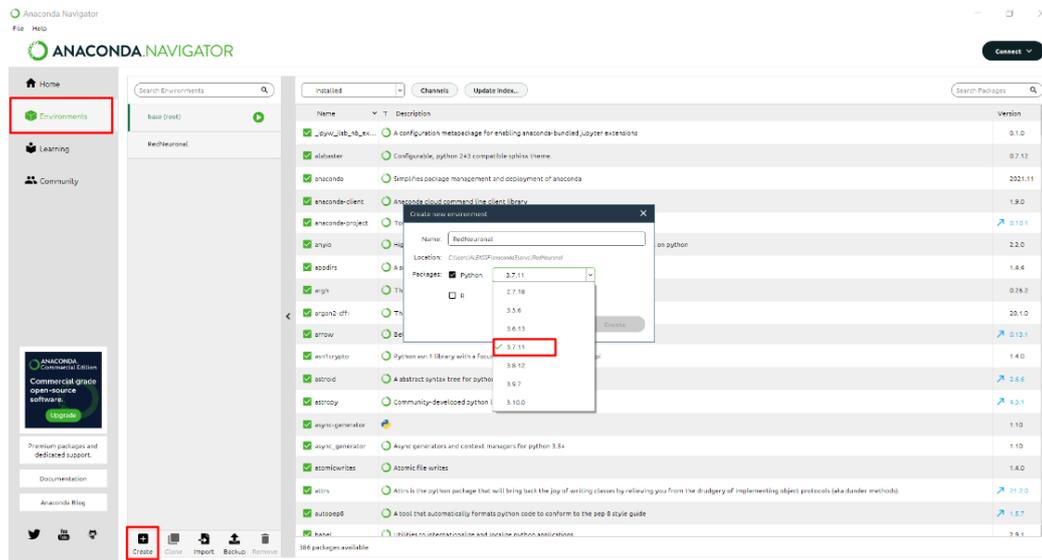
SF11



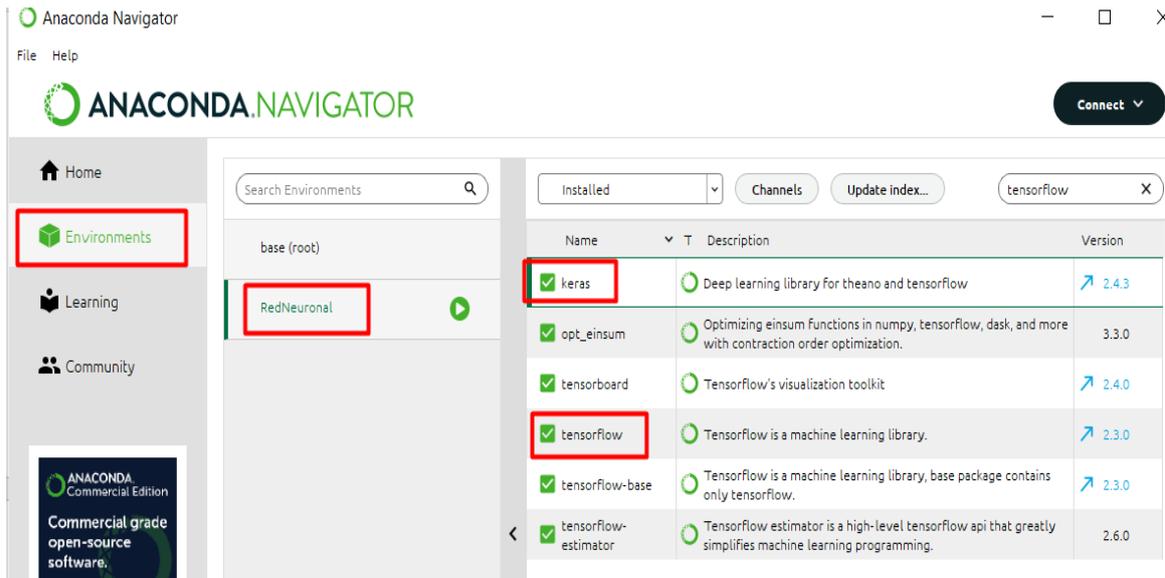
SF12

Anexo 6. Instalación de Anaconda Navigator

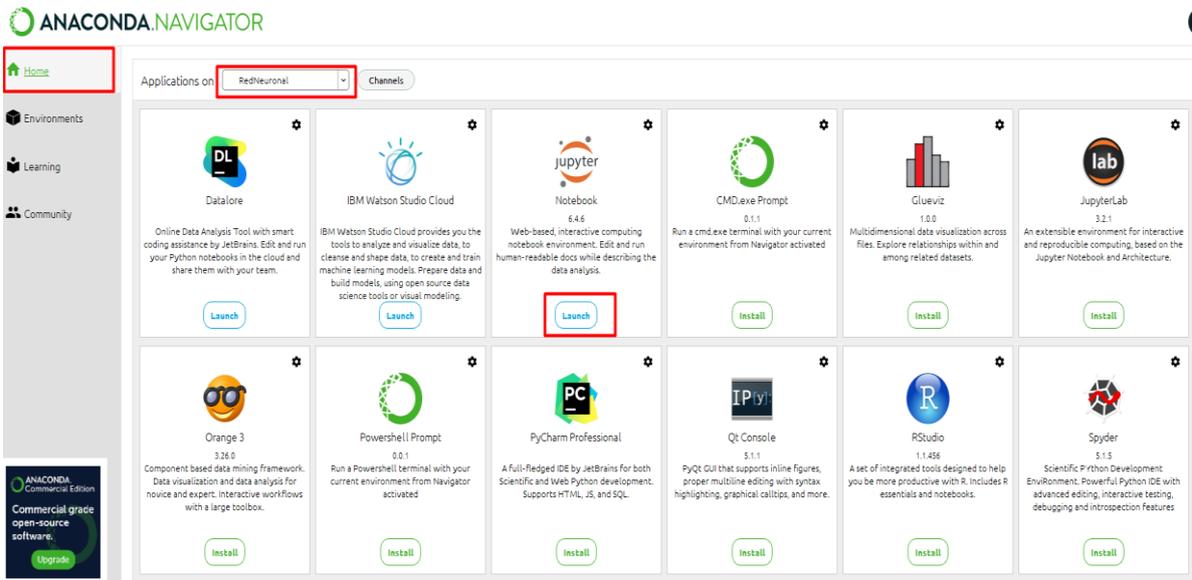
Para crear el entorno de trabajo se escoge la pestaña Environment y seleccionar la opción *create*. Agregar el nombre para el entorno y la versión de Python a utilizar, en esta ocasión se trabaja con Python 3.7.11



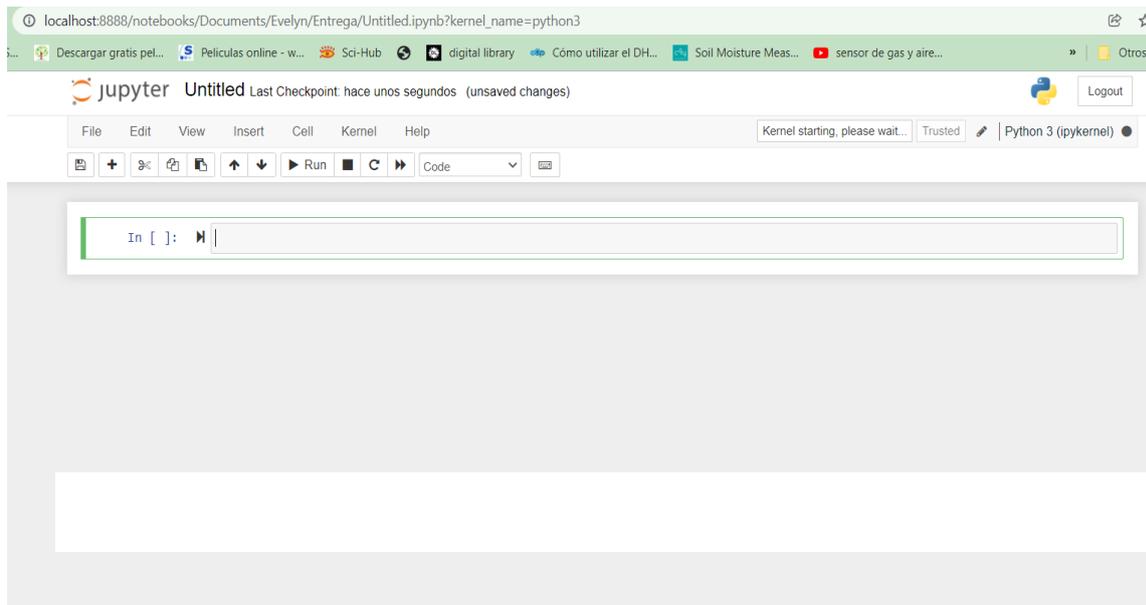
Además, es necesario instalar Tensor Flow y Keras, dos librerías de código abierto. Tensorflow es una biblioteca desarrollada por Google Brain para aplicaciones de aprendizaje automático y redes neuronales profundas que ejecuta de forma eficiente gráficos de flujos. Por su parte Keras es una biblioteca de redes neuronales escrita en Python con una sintaxis homogénea ampliable para la creación de redes neuronales. La combinación de ambas es perfecta ya que incrementando la sencillez de uso y ejecución de las redes neuronales. Para descargar keras y Tensorflow hay que realizarlo desde el entorno de trabajo, solo hay que seleccionar las herramientas y se procede con la descarga.



Una vez ubicados en el entorno de trabajo creado, se tiene acceso a Jupyter Notebook, una aplicación web de código abierto que permite crear y compartir documentos que contienen código en vivo, modelo estadístico, visualización de datos, aprendizaje automático y mucho más.



Con un simple clic en el icono de Jupyter Notebook abrirá en la computadora un servidor jupyter (apuntando a <http://localhost:8888>, el puerto puede variar), en un navegador web predeterminado (Google Chrome) aquí se creará un nuevo proyecto para ejecutar el código de la red neuronal.



Anexo 7. Dataset para el entrenamiento de la red Neuronal

El archivo completo se encuentra ingresando al siguiente enlace: https://utneduec-my.sharepoint.com/:x/g/personal/epbonillaf@utn.edu.ec/EWswCaXz8VRLtSCenklzJsMBfVqMkf-t11lf99b_tdKXgA?e=UwZYsI

DATASET

HUMEDAD DEL SUELO		
Muy húmedo	200-298	1
húmedo	299-349	2
Seco	350-449	3
Muy Seco	450-550	4

RADIACION UV		
Bajo	(1-2)	1
Moderado	(3-4-5)	2
Alto	(6-7)	3
Muy alto	(8-9-10)	4
Extremadamente alto	11	5

Decisión	
Regar	1
No Regar	0

Humedad			Radiacion			Decisión	
Muy seco	491	4	Moderado	3	2	Regar	1
Muy seco	489	4	Moderado	3	2	Regar	1
Muy seco	489	4	Moderado	3	2	Regar	1
Muy seco	489	4	Moderado	3	2	Regar	1
Muy seco	483	4	Moderado	3	2	Regar	1
Muy seco	484	4	Moderado	3	2	Regar	1
Muy seco	482	4	Moderado	3	2	Regar	1
Muy seco	481	4	Bajo	2	1	Regar	1
Muy seco	482	4	Bajo	2	1	Regar	1
Muy seco	481	4	Bajo	2	1	Regar	1
Muy seco	480	4	Bajo	2	1	Regar	1
Muy seco	481	4	Bajo	1	1	Regar	1
Muy seco	486	4	Bajo	1	1	Regar	1
Muy seco	485	4	Bajo	1	1	Regar	1
Muy seco	487	4	Bajo	1	1	Regar	1
Muy seco	486	4	Bajo	1	1	Regar	1
Muy seco	489	4	Bajo	1	1	Regar	1
Muy seco	489	4	Bajo	1	1	Regar	1
Muy seco	480	4	Bajo	1	1	Regar	1
Muy seco	482	4	Bajo	1	1	Regar	1
Muy seco	483	4	Bajo	1	1	Regar	1
Muy seco	483	4	Bajo	1	1	Regar	1
Muy seco	491	4	Bajo	1	1	Regar	1
Muy seco	492	4	Bajo	1	1	Regar	1
Muy seco	492	4	Moderado	4	2	Regar	1
Muy seco	489	4	Bajo	1	1	Regar	1
Muy seco	493	4	Bajo	1	1	Regar	1
Muy seco	491	4	Bajo	1	1	Regar	1
Muy seco	490	4	Bajo	1	1	Regar	1

Anexo 8. Base de datos de ThingSpeak

Con la base de datos de ThingSpeak se puede identificar el estado de riego y a través de este, calcular el tiempo de riego del sistema. El archivo completo se encuentra ingresando al siguiente enlace:

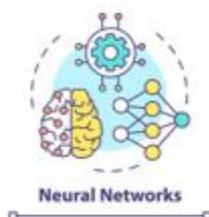
<https://utneduecmy.sharepoint.com/:x:/g/personal/epbonillaf@utn.edu.ec/EXziMMdzn1dGj6d0M5mo5kgBw4lfOmgfrKpcaRNHxz9R1w?e=S8aqJW>

BASE DE DATOS-ESTADO DE RIEGO

Filed 4- Estado de riego	1	Regando
	0	Sin riego

Fecha	ID	Field 1 Humedad del suelo	Filed 2 Temperatura	Field 3 Radiacion UV	Field 4 Estado de Riego	Field 5 Lluvia
2022-05-10T10:30:11-05:00	541	930	24	0	1	0
2022-05-10T10:32:16-05:00	542	927	30	0	1	0
2022-05-10T10:34:09-05:00	543	926	30	0	1	0
2022-05-10T10:35:06-05:00	544	927	30	0	1	0
2022-05-10T10:36:05-05:00	545	927	31	0	1	0
2022-05-10T10:36:59-05:00	546	667	32	0	1	0
2022-05-10T10:37:51-05:00	547	436	32	0	1	0
2022-05-10T10:38:47-05:00	548	493	33	0	1	0
2022-05-10T10:39:45-05:00	549	508	33	0	1	0
2022-05-10T10:40:39-05:00	550	515	33	0	1	0
2022-05-10T10:41:35-05:00	551	520	32	0	1	0
2022-05-10T10:42:30-05:00	552	523	32	0	1	0
2022-05-10T10:45:01-05:00	553	529	33	0	0	1
2022-05-10T10:45:56-05:00	554	530	33	0	0	1
2022-05-10T10:47:11-05:00	555	884	41	8	0	1
2022-05-10T10:48:14-05:00	556	874	69	7	1	0
2022-05-10T10:49:04-05:00	557	895	58	8	1	0
2022-05-10T10:49:59-05:00	558	896	42	9	1	0
2022-05-10T10:50:54-05:00	559	898	43	9	1	0
2022-05-10T10:51:50-05:00	560	898	43	9	1	0
2022-05-10T10:52:46-05:00	561	897	43	9	1	0
2022-05-10T10:53:42-05:00	562	895	42	9	1	0
2022-05-10T10:54:38-05:00	563	895	42	9	1	0
2022-05-10T10:55:37-05:00	564	895	39	10	1	0
2022-05-10T10:56:32-05:00	565	896	43	9	1	0
2022-05-10T10:57:26-05:00	566	895	41	10	1	0
2022-05-10T10:58:22-05:00	567	904	46	3	1	0
2022-05-10T10:59:18-05:00	568	892	43	9	1	0
2022-05-10T11:00:14-05:00	569	901	63	2	1	0
2022-06-13T15:45:48-05:00	570	654	23	0	1	0
2022-06-13T15:46:08-05:00	571	654	22	0	1	0
2022-06-13T15:46:18-05:00	572	654	22	0	1	0
2022-06-13T15:46:35-05:00	573	655	23	0	1	0
2022-06-13T15:46:49-05:00	574	655	23	0	1	0
2022-06-13T15:47:03-05:00	575	654	22	0	1	0
2022-06-13T15:47:17-05:00	576	654	21	0	1	0
2022-06-13T15:47:32-05:00	577	655	21	0	1	0
2022-06-13T15:47:46-05:00	578	653	22	0	1	0
2022-06-13T15:48:03-05:00	579	654	21	0	1	0
2022-06-13T15:48:14-05:00	580	653	21	0	1	0
2022-06-13T15:48:30-05:00	581	654	22	0	1	0

MANUAL DE USUARIO



**“MEJORAMIENTO DE LA EFICIENCIA DE UN SISTEMA DE RIEGO POR
ASPERSIÓN BASADO EN REDES DE SENSORES Y REDES NEURONALES
ARTIFICIALES PARA EL CULTIVO DE PASTO EN LA PARROQUIA DE
TOCACHI PERTENECIENTE AL CANTÓN PEDRO MONCAYO”**

AUTOR: BONILLA FONTE EVELYN PATRICIA

IBARRA-ECUADOR

2022

A. Ubicación de los nodos

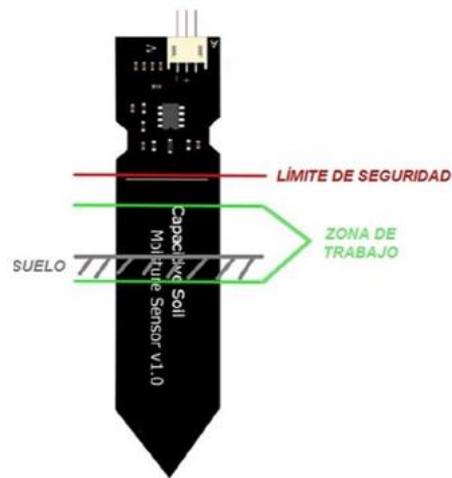
El nodo central debe ubicarse en el lugar establecido por el usuario; fuera de su domicilio, planta alta y a una distancia de 6 metros del suelo.



El nodo sensor debe estar ubicado en la parcela, a una altura de 1.5 metros del suelo, fuera del límite de riego del aspersor para evitar que se active el sensor de lluvia.



Sin embargo, el sensor de humedad del suelo debe colocarse dentro del límite de riego, tomar en cuenta que el sensor de humedad no se debe introducir por completo en el suelo, hay que considerar el límite de seguridad.

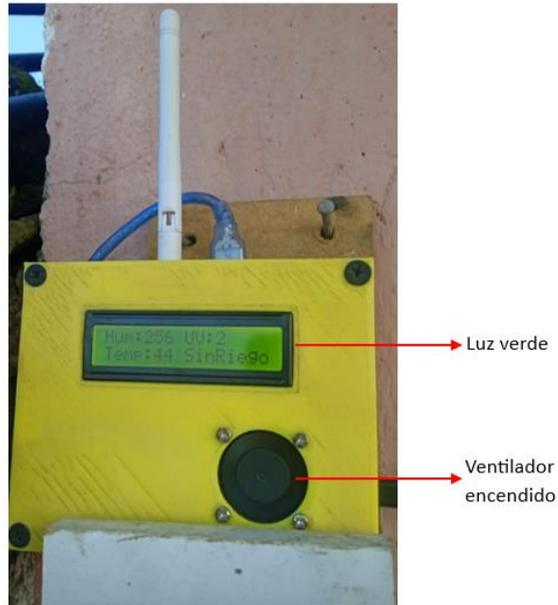


El nodo actuador, al igual que el nodo sensor, debe estar ubicado a una altura de 1.5 metros del suelo y fuera del límite de riego. La electroválvula se coloca en la tubería secundaria, considerando la dirección del flujo de agua.



B. Encendido del Sistema

El nodo central debe conectarse al tomacorriente del domicilio del usuario, se verifica el encendido cuando el ventilador empieza a girar y la luz de la LCD se enciende con un fondo verde.



Para encender el nodo sensor hay que activar el interruptor que se encuentra en la parte derecha del equipo, se verifica el encendido cuando el led que esta junto al interruptor se ponga de color verde. De igual forma el nodo actuador tiene un interruptor para el encendido.



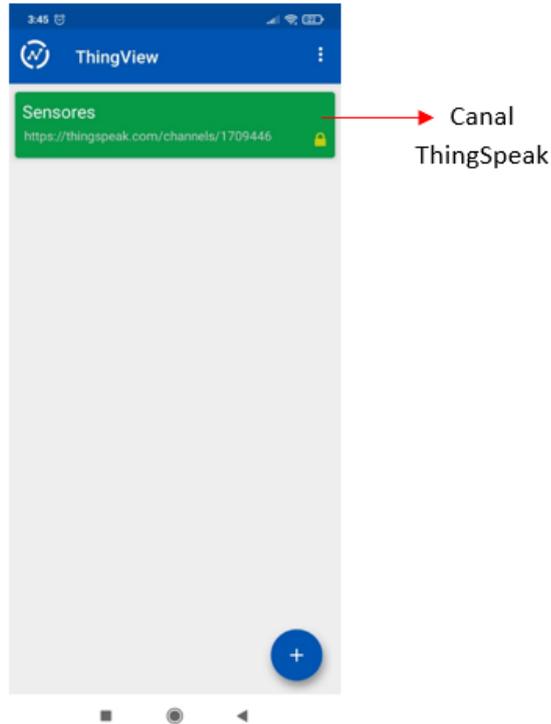
Interruptor y led de encendido



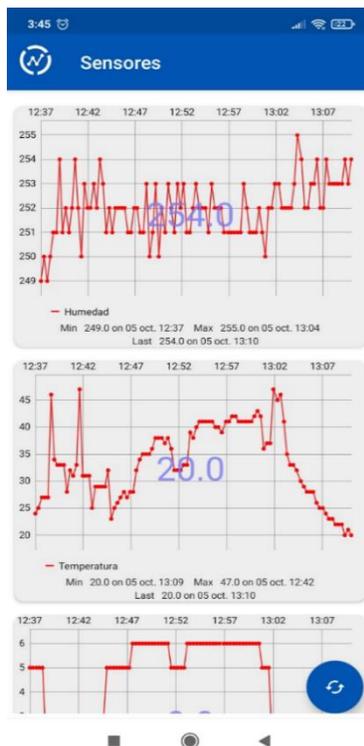
Interruptor de encendido

C. Visualización de resultados

El usuario abre la aplicación ThingView desde su celular y selecciona el canal de ThingSpeak.



Una vez seleccionado el canal aparecerán los datos enviados por los sensores en forma de gráficos.



Anexo 10. Código de la red WSN LoRa

Nodo Sensor

```
#include <SPI.h>
#include <RH_RF95.h>
RH_RF95 rf95;
//DECLARACION CONSTANTES NECESARIAS
#define PIN_H A0 // Pin al que se conecta el sensor de humedad
#define PIN_UV A1 // Pin al que se conecta el módulo U
#define LM35 A2
#define LLUVIA A3
#define CANTIDAD_INDICES_UV 11

int tempC; // Variable para almacenar el valor obtenido del sensor (0 a 1
023)
unsigned int lectura_sensor_lluvia;

// VARIABLES MODULO UV
unsigned int lectura_sensor_UV;
int valor_indice_uv[CANTIDAD_INDICES_UV] = {46, 65, 83, 103, 124, 142, 16
3, 180, 200, 221, 239}; // De 1 a 11
byte indice;
boolean buscando_indice_uv;

// VARIABLES SENSOR HUMEDAD
const int v_aire = 596;
const int v_agua = 252;
int inter = (v_aire - v_agua) / 3;
unsigned int lectura_sensor_H;

//variables envio
int cont_datos = 0;
const int num_datos = 10; //TIEMPO DE ENVIO DE DATOS

void setup() {
  Serial.begin(9600);
  if (!rf95.init())
    Serial.println("init failed");

  //Parámetros de configuración LoRa
  rf95.setFrequency(915.0);
  rf95.setTxPower(20);
  rf95.setSpreadingFactor(7);
  rf95.setSignalBandwidth(125000);
  rf95.setCodingRate4(5);

  lectura_sensor_H = analogRead(PIN_H);
```

```

    lectura_sensor_UV = analogRead(PIN_UV);
}
void loop() {
    obtener_valores();
    delay(600);
    cont_datos += 1;
    Serial.println(cont_datos);
}
void obtener_valores() {
    lectura_sensor_H = analogRead(PIN_H);
    lectura_sensor_UV = analogRead(PIN_UV);
    lectura_sensor_lluvia = analogRead(LLUVIA);

    buscando_indice_uv = true;
    indice = CANTIDAD_INDICES_UV;
    while (buscando_indice_uv && indice > 0)
    {
        indice--;
        if (lectura_sensor_UV > valor_indice_uv[indice])
        {
            buscando_indice_uv = false;
        }
    }
    int T = lectura_T();
    String data = "1 UV=" + String(indice) + " H=" + String(lectura_sensor_
H) + " T=" + String(T) + " ll=" + String(lectura_sensor_lluvia);

    if (cont_datos == num_datos) {
        envio_datos(data);
        cont_datos=0;
    }
}
void envio_datos(String data) {
    int dataLength = data.length(); dataLength++;
    uint8_t total[dataLength];
    data.toCharArray(total, dataLength);
    Serial.println(data);
    rf95.send(total, dataLength);
    rf95.waitPacketSent();
    delay(500);
}
int lectura_T() {
    // Con analogRead leemos el sensor, recordar que es un valor de 0 a 1023
    tempC = analogRead(LM35);
    // Calculamos la temperatura con la fórmula
    tempC = (5.0 * tempC * 100.0) / 1024.0;
    return tempC;
}

```

Nodo Central-Puerta de Enlace

Arduino

```
#include <SPI.h>
#include <RH_RF95.h>
RH_RF95 rf95;
int led = 13;
//memoria
uint8_t buf_ant;
uint8_t len_ant;

void setup()
{
  pinMode(led, OUTPUT);
  Serial.begin(9600);
  while (!Serial) ; // Wait for serial port to be available
  if (!rf95.init())
    Serial.println("init failed");
  // Frecuencia
  rf95.setFrequency(915.0);
  // Potencia de Transmisión
  rf95.setTxPower(20);
  // Ancho de Banda
  rf95.setSignalBandwidth(125000);
  // Factor de Propagación
  rf95.setSpreadingFactor(7);
  //Tasa de codificación de errores 5(4/5), 6(4/6), 7(4/7), 8(4/8)
  rf95.setCodingRate4(5);
}

void loop()
{
  leer_lora();
  leer_Puerto_Serie();
  delay(500);
}

void leer_lora() {
  if (rf95.available())
  {
    uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
    uint8_t len = sizeof(buf);
    if (rf95.recv(buf, &len))
    {
      buf_ant = buf;
      len_ant = len;
      digitalWrite(led, HIGH);
      Serial.println((char*)buf);
    }
  }
}
```

```

        digitalWrite(led, LOW);
        delay(500);
    }
}
else {

    if (rf95.recv(buf_ant, &len_ant))
    {
        digitalWrite(led, HIGH);
        Serial.println((char*)buf_ant);
        digitalWrite(led, LOW);
        delay(500);
    }
    delay(100);
}
}
void leer_Puerto_Serie() {
    if (Serial.available() > 0) {
        //leemos la opción enviada
        String datos = Serial.readString();
        enviar_lora(datos);

    }
}
void enviar_lora(String data ) {
    int dataLength = data.length(); dataLength++;
    uint8_t total[dataLength];
    data.toCharArray(total, dataLength);
    //Serial.println(data);
    rf95.send(total, dataLength);
    rf95.waitPacketSent();
    delay(500);
}
}

```

Nodo Central-Puerta de enlace

Raspberry Pi

```
import time
import serial
import requests
ser= serial.Serial("/dev/ttyACM0",baudrate=9600)
try:
    while True:
        lectura = ser.readline()
        lectura = str(lectura.decode("utf-8"))
        datos=lectura.split(" ")
        if(len(datos)<3):
            continue
# obtención datos UV
        UV=datos [0]
        UV=UV[3:]
        #print(UV)
#Obtencion datos Humedad
        H=datos[1]
        H=H[2:]
        #print(H)
#obtencion datos Temperatura
        T=datos[2]
        T=T[2:]
        #print(T)
        enviar=requests.get("https://api.thingspeak.com/update?api_key=UJ8WNTDSS5
        BSXUE1&field1="+str(H)+"&field2="+str(T)+"&field3="+str(UV))
        print(lectura)
        time.sleep(0.1)
except KeyboardInterrupt:
    print("Interrupcion por teclado")
except ValueError as ve:
    print(ve)
    print("Otra interrupcion")
finally:
    ser.closeg()
```

Nodo Actuador

```
#include <SPI.h>
#include <RH_RF95.h>
RH_RF95 rf95;
int led = 13;
int electrovalvula=7;
String memoria = "0";

void setup()
{
  pinMode(led, OUTPUT);
  pinMode(electrovalvula, OUTPUT);
  Serial.begin(9600);
  while (!Serial) ; // Wait for serial port to be available
  if (!rf95.init())
    Serial.println("init failed");

  //Parámetros modulo LoRa
  rf95.setFrequency(915.0);
  rf95.setTxPower(20);
  rf95.setSpreadingFactor(7);
  rf95.setSignalBandwidth(125000);
  rf95.setCodingRate4(5);
}

void loop()
{
  if (rf95.available())
  {
    uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
    uint8_t len = sizeof(buf);

    if (rf95.recv(buf, &len))
    {
      digitalWrite(led, HIGH);
      Serial.println((char*)buf);
      String dato = String(buf[0]);
      if (dato == "97") {
        Serial.println("Lectura Actuador");
        String desi = String(buf[2]);
        if (desi != memoria) {
          if (desi == "48") {
            // No riega
            memoria = desi;
            digitalWrite(electrovalvula,LOW);
            Serial.println(" Riego Detenido");
          }
        }
      }
      else if (desi == "49") {
```

```
        // Riega
        memoria = desi;
        digitalWrite(electrovalvula,HIGH);
        Serial.println(" Regando");
    }
}
Serial.println(desi);
}
digitalWrite(led, LOW);
}
}
}
```

Anexo 11. Código de la Red Neuronal en Jupyter Notebook

jupyter Entrenamiento_Red Last Checkpoint: 26/01/2022 (autosaved) Python 3 (ipykernel)

```
In [1]: import pandas as pd
import tensorflow as tf
import numpy as np
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
import matplotlib.pyplot as plt
print(tf.__version__)

2.1.0
```

Preprocesamiento de datos

```
In [2]: #preprocesamiento de datos
df = pd.read_excel("DatasetV2.xlsx", sheet_name="data")
print(df)
print(df.columns)

Humedad radiacion Decisión
0      200      0      0
1      201      0      0
2      202      0      0
3      203      0      0
4      204      0      0
...     ...     ...     ...
8817   596     10      0
8818   597     10      0
8819   598     10      0
8820   599     10      0
8821   600     10      0

[8822 rows x 3 columns]
Index(['Humedad', 'radiacion', 'Decisión'], dtype='object')
```

```
In [3]: dataset=df.values
```

```
In [4]: print(dataset)

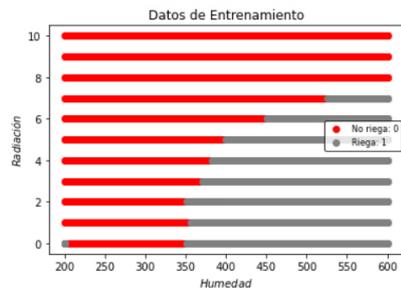
[[200  0  0]
 [201  0  0]
 [202  0  0]
 ...
 [598 10  0]
 [599 10  0]
 [600 10  0]]
```

```
In [5]: X=dataset[:, :2]
print(X)
Y=dataset[:, 2:]
print(Y)

[[200  0]
 [201  0]
 [202  0]
 ...
 [598 10]
 [599 10]
 [600 10]]
[[0]
 [0]
 [0]
 ...
 [0]
 [0]
 [0]]
```

```
In [6]: idx0 = np.where(Y==0)
idx1 = np.where(Y==1)
```

```
In [7]: plt.scatter(X[idx0,0],X[idx0,1],color='red',label='No riega: 0')
plt.scatter(X[idx1,0],X[idx1,1],color='gray',label='Riega: 1')
plt.xlabel('$Humedad$')
plt.ylabel('$Radiación$')
plt.legend(bbox_to_anchor=(0.765,0.6),fontsize=8,edgecolor='black')
plt.title("Datos de Entrenamiento")
plt.show()
```



```
In [8]: np.random.seed(1)
```

Contruimos el modelo

```
model= keras.Sequential([
    #keras.layers.Dense(4,input_shape=[2]),
    #keras.layers.Dense(8),
    #keras.layers.Dense(8),
    #keras.layers.Dense(8),
    keras.layers.Dense(1, activation="sigmoid",input_shape=[2])
])
```

Copilamos el modelo

```
optimizer = tf.keras.optimizers.RMSprop(0.001)
model.compile(loss='mse', optimizer=optimizer, metrics=['mae', 'mse'])
```

```
In [9]: # define model
model = Sequential()
model.add(Dense(100, input_dim=2, activation='relu'))
model.add(Dense(100, activation='relu'))
model.add(Dense(100, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
In [10]: model.compile(loss='binary_crossentropy', optimizer=tf.optimizers.Adam(learning_rate=0.001), metrics=['accuracy'])
#model.compile(loss='mse', optimizer=tf.optimizers.Adam(learning_rate=0.001), metrics=['accuracy'])
```

Mostramos las características del modelo

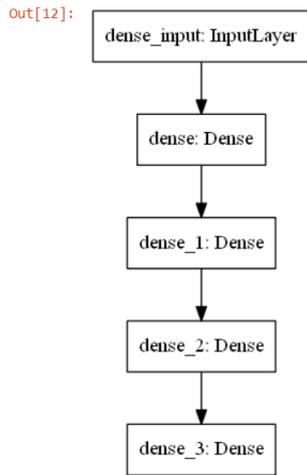
```
In [11]: model.summary()
```

```
Model: "sequential"
-----
```

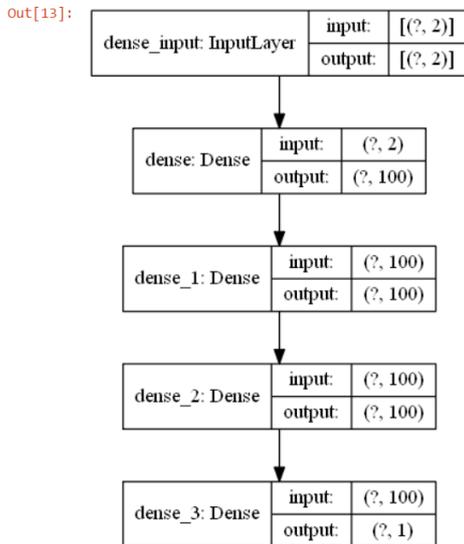
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 100)	300
dense_1 (Dense)	(None, 100)	10100
dense_2 (Dense)	(None, 100)	10100
dense_3 (Dense)	(None, 1)	101

```
-----
Total params: 20,601
Trainable params: 20,601
Non-trainable params: 0
```

```
In [12]: # graficamos el modelo de la red neuronal
tf.keras.utils.plot_model(model, "my_first_model.png")
```



```
In [13]: #Grafico con la forma de entrada y salida de cada capa
tf.keras.utils.plot_model(model, "my_first_model_with_shape_info.png", show_shapes=True)
```



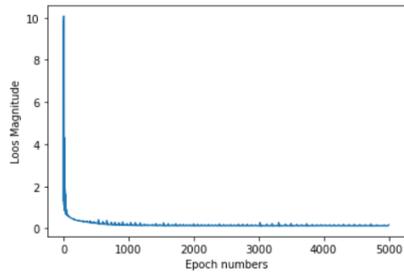
```
In [85]: num_epochs = 5000
batch_size = X.shape[0]
```

```
In [86]: history = model.fit(X, Y,
epochs=num_epochs,
batch_size=batch_size,
validation_split = 0.3,
verbose=1)
```

```
Train on 6175 samples, validate on 2647 samples
Epoch 1/5000
6175/6175 [=====] - 0s 24us/step - loss: 9.8356 - accuracy: 0.6002 - val_loss: 0.8567 - val_accu
racy: 0.7571
Epoch 2/5000
6175/6175 [=====] - 0s 6us/step - loss: 1.3189 - accuracy: 0.6002 - val_loss: 11.7906 - val_accu
racy: 0.2429
Epoch 3/5000
6175/6175 [=====] - 0s 11us/step - loss: 8.2874 - accuracy: 0.3998 - val_loss: 14.3218 - val_acc
uracy: 0.2429
Epoch 4/5000
6175/6175 [=====] - 0s 5us/step - loss: 10.0733 - accuracy: 0.3998 - val_loss: 11.8317 - val_acc
uracy: 0.2429
Epoch 5/5000
6175/6175 [=====] - 0s 4us/step - loss: 8.3106 - accuracy: 0.3998 - val_loss: 6.3565 - val_accu
racy: 0.2429
Epoch 6/5000
6175/6175 [=====] - 0s 6us/step - loss: 4.4380 - accuracy: 0.3998 - val_loss: 0.7529 - val_accu
racy: 0.7571
```

```
In [87]: plt.xlabel("Epoch numbers")
plt.ylabel("Loos Magnitude")
plt.plot(history.history["loss"])
```

Out[87]: <matplotlib.lines.Line2D at 0x29ff61ed908>



```
In [105]: #Riega
result = model.predict(X)
print(result)
result.shape
```

```
[[0.00061861]
 [0.00064281]
 [0.00066794]
 ...
 [0.04902571]
 [0.05155007]
 [0.05419726]]
```

Out[105]: (8822, 1)

```
In [106]: res = result[:,0]
res
```

Out[106]: array([0.00061861, 0.00064281, 0.00066794, ..., 0.04902571, 0.05155007, 0.05419726], dtype=float32)

```
In [107]: r=[]
for a in res:
    if a>0.5:
        a=1
        r.append(a)
    elif a<0.5:
        a=0
        r.append(a)
```

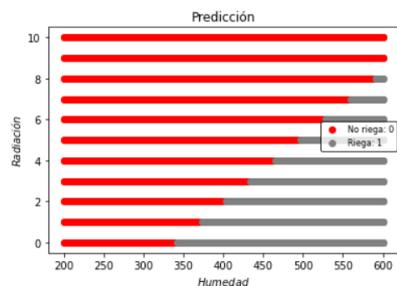
```
In [108]: r=np.array(r)
r
```

Out[108]: array([0, 0, 0, ..., 0, 0, 0])

```
In [109]: x0 = np.where(r==0)
x1 = np.where(r==1)
x1
```

Out[109]: (array([142, 143, 144, ..., 8017, 8018, 8019], dtype=int64),)

```
In [110]: plt.scatter(X[x0,0],X[x0,1],color='red',label='No riego: 0')
plt.scatter(X[x1,0],X[x1,1],color='gray',label='Riego: 1')
plt.xlabel('$Humedad$')
plt.ylabel('$Radiación$')
plt.legend(bbox_to_anchor=(0.765,0.6),fontsize=8,edgecolor='black')
plt.title("Predicción")
plt.show()
```



```
In [111]: model.save("riego")
```

```
In [112]: model = tf.keras.models.load_model("riego")
```

```
In [113]: print(model.predict([[550,2]]))
```