



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

**“DISEÑO DE UN SISTEMA PARA EL CONTROL DE RIEGO MEDIANTE TÉCNICAS
DE APRENDIZAJE AUTOMÁTICO APLICADA A LA AGRICULTURA DE
PRECISIÓN EN LA GRANJA LA PRADERA DE LA UNIVERSIDAD TÉCNICA DEL
NORTE”**

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

AUTOR: HÉCTOR DARÍO CADENA LEMA
DIRECTOR: MSC. EDGAR ALBERTO MAYA OLALLA
ASESOR: MSC. MAURICIO DOMÍNGUEZ
ASESOR: MSC. PAÚL ROSERO

Ibarra-Ecuador

2020



UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA

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

1. IDENTIFICACIÓN DE LA OBRA.

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

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	100456219-3		
APELLIDOS Y NOMBRES:	Cadena Lema Héctor Darío		
DIRECCIÓN:	Otavalo, Rocafuerte y Guayaquil		
EMAIL:	hdcadena@utn.edu.ec		
TELÉFONO FIJO:	062925220	TELÉFONO MÓVIL:	0981561829

DATOS DE LA OBRA	
TÍTULO:	Diseño de un sistema para el control de riego mediante técnicas de aprendizaje automático aplicada a la agricultura de precisión en la granja La Pradera de la Universidad Técnica del Norte
AUTOR (ES):	Cadena Lema Héctor Darío
FECHA: DD/MM/AAAA	28/Febrero/2020
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	Ingeniero en Electrónica y Redes de Comunicación
ASESOR /DIRECTOR:	MSc. Edgar Alberto Maya Olalla

2. CONSTANCIAS.

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

Ibarra, a los 28 días del mes de Febrero de 2020

EL AUTOR:



.....
Cadena Lema Héctor Darío
100456219-3



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN.

MAGISTER EDGAR MAYA, DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN
CERTIFICA:

Que, el presente trabajo de Titulación “DISEÑO DE UN SISTEMA PARA EL CONTROL DE RIEGO MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO APLICADA A LA AGRICULTURA DE PRECISIÓN EN LA GRANJA LA PRADERA DE LA UNIVERSIDAD TECNICA DEL NORTE.” Ha sido desarrollado por el señor Cadena Lema Héctor Darío bajo mi supervisión.

Es todo en cuanto puedo certificar en honor de la verdad.

MSc. Edgar Maya

DIRECTOR

Dedicatoria

A mi madre Maricela Lema, por ser una madre abnegada y ejemplar; sin su cariño y paciencia, nunca podría haber llegado tan lejos.

A mi padre Manuel Cadena, por ser el mejor ejemplo de vida que he tenido gracias por enseñarme a ser una persona constante y a no darme por vencido tan fácilmente.

A mi hermana Mayra, quien a pesar de la distancia siempre he sentido su apoyo y cariño incondicional.

A mi hermano Juan Pablo, que a pesar de su corta edad me ha brindado su ayuda, cariño y apoyo en todo momento.

Finalmente, este trabajo de investigación va dedicado a mi persona por ser siempre constante y por no haberme dejado vencer fácilmente por todas las adversidades que se presentaron durante mi estadía en esta prestigiosa institución.

Agradecimientos

Este trabajo de tesis ha sido una gran bendición en todo sentido y te lo agradezco padre Celestial, y no cesan mis ganas de decir que es gracias a ti que esta meta está cumplida, espero seguir contando con tu bendición y protección para poder alcanzar muchos más logros en mi vida.

A mis padres quienes han sido un gran soporte y aporte a lo largo de mi vida ya que, gracias a su eterno amor y apoyo, he logrado culminar esta anhelada meta de vida, además de que me han formado como persona, son y seguirán siendo un gran ejemplo para mí y mis hermanos.

A mis hermanos Mayra y Juan Pablo, por su tolerancia e impulso, los cuales fueron fundamentales para el cumplimiento y culminación de esta larga travesía.

A mis tres grandes amigos Luis, Israel y César, por brindarme su ayuda y motivación hasta en los momentos más difíciles.

A ti querida amiga, muchas gracias por todo, fue un gusto haber compartido todo este tiempo contigo, gracias también por tu apoyo y por tu amistad, pienso que hicimos un buen equipo de trabajo durante este tiempo de estudios.

A toda mi familia por su apoyo incondicional, en especial a mis tías Lourdes y Enma que en momentos en los que he requerido de su ayuda y apoyo siempre me han tendido una mano.

A todos y cada uno de mis profesores, quienes fueron los responsables de mi formación académica.

ÍNDICE

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE	I
CERTIFICACIÓN.	III
DEDICATORIA	IV
AGRADECIMIENTOS	V
ÍNDICE	VI
ÍNDICE DE FIGURAS	XIII
ÍNDICE DE ECUACIONES.....	XVI
ÍNDICE DE TABLAS	XVII
RESUMEN.....	XIX
ABSTRACT	XX
1. CAPITULO I. ANTECEDENTES.....	1
1.1. TEMA	1
1.2. PLANTEAMIENTO DEL PROBLEMA	1
1.3. OBJETIVOS	3
1.3.1. Objetivo General	3
1.3.2. Objetivos Específicos.....	3
1.4. ALCANCE.....	3
1.5. JUSTIFICACIÓN	5
2. CAPÍTULO II. FUNDAMENTACIÓN TEÓRICA	7
2.1. Agricultura de precisión.....	7
2.2. Factores o Variables a medir.....	9
2.2.1. Humedad del suelo	9
2.2.2. Humedad Relativa	10

2.2.3.	Temperatura Ambiental.....	10
2.2.4.	Luminosidad.....	11
2.3.	Sistemas de Riego.....	11
2.3.1.	Riego por aspersión.....	12
2.3.2.	Riego por micro-compuertas.....	13
2.3.3.	Riego por goteo.....	13
2.4.	Aprendizaje Automático.....	18
2.4.1.	Aprendizaje no supervisado.....	18
2.4.2.	Aprendizaje Supervisado.....	19
2.5.	Redes Neuronales Artificiales.....	20
2.5.1.	Reseña Histórica.....	20
2.5.2.	Fundamentos Biológicos de las Redes Neuronales Artificiales.....	21
2.5.3.	Neurona Biológica.....	22
2.5.3.1.	Dendritas.....	23
2.5.3.2.	Sinapsis.....	24
2.5.3.3.	Soma.....	24
2.5.3.4.	Axón.....	25
2.5.4.	Neurona Artificial.....	25
2.5.4.1.	Entradas.....	27
2.5.4.2.	Pesos.....	28
2.5.4.3.	Regla de Propagación.....	28
2.5.4.4.	Función de Activación.....	29
2.5.4.5.	Función de salida.....	31
2.5.5.	Niveles o Capas de Neuronas.....	32
2.5.6.	Tipos de neuronas artificiales.....	32

2.5.7.	Fase de Entrenamiento	33
2.5.8.	Problemas comunes en el entrenamiento de la red neuronal	34
2.5.9.	Fase de Validación o Test	35
2.6.	Low Power Wide Área Network (LPWAN).....	36
2.6.1.	LoRa	39
2.6.2.	LoRaWAN	39
2.6.3.	Arquitectura.....	39
2.6.4.	Modulación y espectro.....	40
2.6.5.	Parámetros de interés en LoRaWAN.....	41
2.6.6.	Estructura del mensaje LoRa.....	43
2.6.7.	Clases de nodos.....	44
2.6.8.	Seguridad y Modos de Activación	44
2.6.9.	Conexiones	45
2.7.	Sistemas Embebidos	46
2.7.1.	Componentes de un sistema embebido	46
3.	CAPÍTULO III. DESARROLLO EXPERIMENTAL	48
3.1.	Metodología	48
3.2.	Modelo en V	48
3.3.	Etapa de Análisis.....	50
3.3.1.	Situación actual	50
3.4.	Descripción General del Sistema de Riego Propuesto.....	52
3.5.	Determinación de Stakeholders	53
3.6.	Requerimientos de diseño	54
3.6.1.	Nomenclatura de requerimientos	54
3.6.2.	Requerimientos de Stakeholders	55

3.6.3.	Requerimientos funcionales del sistema	55
3.6.4.	Requerimientos de arquitectura.....	57
3.7.	Elección de Software y Hardware para la implementación del sistema	58
3.7.1.	Elección del Hardware	58
3.7.1.1.	Elección de hardware para la Zona 1	58
3.7.1.2.	Elección de hardware para la Zona 2	61
3.7.2.	Elección del Software.....	62
3.7.2.1.	Elección del software para la zona 1	62
3.7.2.2.	Elección del Software para la zona 2	63
3.8.	Hardware destinado para el sistema.....	65
3.8.1.	Sistema embebido Raspberry Pi 3.....	65
3.8.2.	Microcontrolador Arduino UNO.....	66
3.8.3.	Módulo LoRaWAN RN-2483.....	68
3.8.4.	Válvula Solenoide	68
3.8.5.	Diagramas de puertos y pines del hardware seleccionado	69
3.8.5.1.	Diagrama de pines y puertos – Raspberry Pi 3	70
3.8.5.2.	Diagrama de pines y puertos – Arduino UNO R3	70
3.8.5.3.	Diagrama de pines y puertos – Módulo LoRaWAN (RN2483).....	71
3.9.	Diseño del sistema	72
3.9.1.	Diagrama de bloques general del sistema	72
3.9.1.1.	Diagrama de bloques de la primera etapa del sistema.	73
3.9.1.2.	Diagrama de bloques de la segunda etapa.....	74
3.9.2.	Topología de la red.....	75
3.9.3.	Diagrama Circuital del Sistema.....	76
3.9.3.1.	Zona 1.....	77

3.9.3.2.Zona 2.....	78
3.9.4. Alimentación del Sistema.....	79
3.9.4.1.Cálculos de alimentación eléctrica.....	79
3.10. Diseño del software del sistema.....	81
3.10.1. Recolección de datos para el entrenamiento de la red neuronal.....	81
3.10.2. Diseño de la red neuronal artificial para automatizar el sistema de riego	
82	
3.10.2.1.Número de capas de la RNA.....	83
3.10.2.2.Número de neuronas en la capa intermedia o capa oculta	84
3.10.2.3.Modelo de la red neuronal artificial del sistema	85
3.10.2.4.Función de activación.....	86
3.10.2.5.Extracción de datos de entrenamiento.....	86
3.10.2.6.Preprocesamiento de datos de entrenamiento	87
3.10.2.7.Algoritmo de entrenamiento.....	87
3.10.2.8.Pesos actualizados de la RNA.....	88
3.10.2.9.Rendimiento de la red neuronal	88
3.10.2.10.Guardar red neuronal entrenada	91
3.10.3. Diagramas de Flujo	91
3.10.3.1.Recolección de datos para entrenamiento de la red neuronal.	91
3.10.3.2.Implementación del Sistema de Riego Inteligente.....	93
3.10.3.3.Diagrama de flujo del nodo actuador	94
3.11. Cálculos del enlace inalámbrico entre la zona 1 y la zona 2	95
4. CAPÍTULO IV. PRUEBAS DE FUNCIONAMIENTO	98
4.1. Pruebas de funcionalidad del sistema de riego	98
4.1.1. Test eléctrico del sistema	98

4.1.2.	Test del subsistema pasivo	100
4.1.3.	Test del subsistema activo.....	101
4.1.4.	Test de hardware	103
4.2.	Preparación del cultivo e instalación de nodos	104
4.3.	Análisis del cultivo	106
4.3.1.	Altura de la planta	107
4.3.2.	Coloración de la hoja.....	108
4.3.3.	Frondosidad	109
4.3.4.	Tamaño de la Raíz.....	110
4.3.5.	Eficiencia en el uso del agua	111
4.4.	Precisión del sistema de riego automático.	114
4.5.	Costo del sistema	115
4.5.1.	Costos de hardware	115
4.5.2.	Costo de infraestructura	116
4.5.3.	Costo de ingeniería.....	117
4.5.4.	Costo de software	117
4.5.5.	Costo de implementación del sistema general	118
4.6.	Conclusiones	120
4.7.	Recomendaciones	122
BIBLIOGRAFÍA.....		123
ANEXO 1: HOJA DE DATOS MÓDULO LORAWAN RN2483		129
ANEXO 2: CÓDIGO DE RECOLECCIÓN DE DATOS PARA ENTRENAMIENTO DE LA RNA. (ZONA 1)		136
ANEXO 3: CÓDIGO DE RECOLECCIÓN DE DATOS PARA ENTRENAMIENTO DE LA RNA. (ZONA 2)		138

ANEXO 4: CREACIÓN DE TABLAS PARA EL ALMACENAMIENTO DE DATOS DE ENTRENAMIENTO DE LA RNA.	141
ANEXO 5: CREACIÓN, ENTRENAMIENTO, ANÁLISIS Y GUARDADO DE LA RED NEURONAL ARTIFICIAL.....	142
ANEXO 6: INTEGRACIÓN DE LA RED NEURONAL ARTIFICIAL A LA WSN PARA AUTOMATIZAR EL RIEGO.....	144
ANEXO 7: CÓDIGO DEL NODO ACTUADOR.....	147
ANEXO 8: CREACIÓN DE TABLAS PARA EL ALMACENAMIENTO DE DATOS DEL SISTEMA INTEGRADO.....	149
ANEXO 9: CREACIÓN DE LA PÁGINA WEB PARA VISUALIZAR LAS VECES DE RIEGO.....	149
ANEXO 10: GUÍA DE COMANDOS DE CONFIGURACIÓN DEL MÓDULO RN2483	153
ANEXO 11: GUÍA DE USUARIO DE LA LIBRERÍA SKLEARN	153
ANEXO 12: DATASET DE ENTRENAMIENTO DE LA RNA.....	153
ANEXO 13: CIRCUITO IMPRESO SHIELD MÓDULO LORA PARA RASPBERRY PI	153
ANEXO 14: CIRCUITO IMPRESO SHIELD MÓDULO LORA PARA ARDUINO....	153
ANEXO 15: CIRCUITO IMPRESO SHIELD NODO ACTUADOR	153

ÍNDICE DE FIGURAS

Figura 1. Esquema Agricultura de Precisión.....	8
Figura 2. Riego por Aspersión.....	12
Figura 3. Riego por Micro-compuertas.....	13
Figura 4. Reservorio de Agua.....	14
Figura 5. Equipo de bombeo de agua.....	15
Figura 6. Sistema de filtrado para riego por goteo.....	16
Figura 7. Mangueras PVC de conducción.....	16
Figura 8. Gotero de botón regulable.....	17
Figura 9. Electroválvula solenoide.....	17
Figura 10. Partes de una neurona biológica.....	23
Figura 11. Modelo de una Neurona Artificial de una con n entradas y una salida.....	26
Figura 12. Gráfica Función de activación lineal.....	30
Figura 13. Función de activación sigmoidea.....	31
Figura 14. Función tangente hiperbólica.....	31
Figura 15. Diagrama radial comparativo entre redes LPWAN, redes celulares y Zigbee.....	37
Figura 16. Esquema de comunicación tecnología LoRa.....	40
Figura 17. Estructura del Paquete LoRa.....	43
Figura 18. Componentes de un sistema embebido (Nivel Físico).....	47
Figura 19. Modelo en V para el desarrollo de proyectos.....	49
Figura 20. Vista aérea del área de cultivo y lugar del nodo central.....	51
Figura 21. Distribución Zonal.....	53
Figura 22. Placa Arduino UNO.....	67
Figura 23. Módulo LoRaWAN RN2483.....	68
Figura 24. Electroválvula Solenoide.....	69

Figura 25. Pines y puertos del Sistema Embebido Raspberry Pi 3.....	70
Figura 26. Pines y puertos de la placa reducida Arduino UNO R3	71
Figura 27. Pines y puertos del Módulo LoRaWAN (RN2483)	71
Figura 28. Diagrama de bloques general del sistema	73
Figura 29. Diagrama de bloques - Primera etapa del sistema.....	74
Figura 30. Diagrama de bloques - Segunda etapa del sistema.....	75
Figura 31. Topología del sistema de riego.....	76
Figura 32. Diagrama de conexión de la Zona 1 (Activación/Desactivación de Riego).....	77
Figura 33. Diagrama de Conexión de la Zona 1 (Recolección de datos)	78
Figura 34. Diagrama de conexión de la Zona 2.....	79
Figura 35. Elementos de la Red Neuronal Artificial	82
Figura 36. Proceso de implementación de la red neuronal	83
Figura 37. Representación del número de capas de la RNA	84
Figura 38. Modelo de la RNA	85
Figura 39. Base de datos para el entrenamiento de la RNA	86
Figura 40. Etapa de preprocesamiento de datos	87
Figura 41. Pesos actualizados de la RNA	88
Figura 42. Archivo “.pkl” generado.....	91
Figura 43. Diagrama de Flujo Zona 1 (Recolección de datos para entrenamiento de la RNA)	92
Figura 44. Diagrama de Flujo Zona 2 (Recolección de datos para entrenamiento de la RNA)	93
Figura 45. Diagrama de flujo implementación del Sistema de Riego Inteligente	94
Figura 46. Diagrama de flujo del nodo actuador	95
Figura 47. Enlace en Radio Mobile	97

Figura 48. Visualización del enlace inalámbrico en Google Earth.....	97
Figura 49. Reconocimiento de las placas de procesamiento a) nodo de sensores b) nodo actuador.....	102
Figura 50. Carga exitosa de sketches en los nodos y uso de memoria	102
Figura 51. Conexión exitosa de nodos con el gateway.....	103
Figura 52. Lectura de los sensores.....	104
Figura 53. Instalación del nodo de sensores (Recolección de datos).....	105
Figura 54. Preparación de la parcela e instalación de mangueras	106
Figura 55. Comparación en el tamaño de la planta usando a) riego automatizado y b) riego manual.....	107
Figura 56. Resultados de hacer el riego durante el día	108
Figura 57. Resultados de hacer el riego durante la noche	109
Figura 58. Frondosidad de la planta: a) con riego manual y b) con riego automático	110
Figura 59. Tamaño de la raíz: a) con riego manual y b) con riego automático	111

ÍNDICE DE ECUACIONES

Ecuación 1. Sumatoria de las entradas pesadas	27
Ecuación 2. Productoria de las entradas pesadas	27
Ecuación 3. Máximo de las entradas pesadas	27
Ecuación 4. Suma empoderada	29
Ecuación 5. Función de activación lineal.....	30
Ecuación 6. Función de activación sigmoidea	30
Ecuación 7. Función tangente hiperbólica	31
Ecuación 8. Fórmula de tiempo de vida de una batería	80
Ecuación 9. Regla de la pirámide incremental.....	84
Ecuación 10. Fórmula para calcular la precisión de la RNA	89
Ecuación 11. Fórmula para calcular la sensibilidad.....	90
Ecuación 12. Fórmula para calcular la especificidad.....	90
Ecuación 13. Ecuación del modelo Hokumura - Hata	96
Ecuación 14. Cálculo de Precisión.....	114

ÍNDICE DE TABLAS

Tabla 1. Lista de stakeholders del sistema.....	54
Tabla 2. Definición de Acrónimos y Abreviaturas	54
Tabla 3. Requerimientos de Stakeholders.....	55
Tabla 4. Requerimientos Funcionales del Sistema	56
Tabla 5. Requerimientos de Arquitectura	57
Tabla 6. Elección del microcontrolador en la Zona 1	59
Tabla 7. Elección de la tecnología inalámbrica	60
Tabla 8. Elección de la electroválvula solenoide	61
Tabla 9. Elección del Sistema Embebido de la Zona 2.....	62
Tabla 10. Elección del Software de la Zona 1	63
Tabla 11. Elección del Software de la Zona 2 (Gateway y RNA)	64
Tabla 12. Elección del Software de la Zona 2 (Almacenamiento de datos)	64
Tabla 13. Especificaciones Técnicas del sistema embebido Raspberry Pi 3	66
Tabla 14. Especificaciones técnicas del Arduino UNO.....	67
Tabla 15. Especificaciones técnicas módulo LoRaWAN RN-2483	68
Tabla 16. Especificaciones técnicas de la electroválvula solenoide	69
Tabla 17. Consumo eléctrico de dispositivos.....	80
Tabla 18. Matriz de confusión	89
Tabla 19. Test del subsistema eléctrico	99
Tabla 20. Pruebas del subsistema pasivo	100
Tabla 21. Pruebas de verificación del subsistema activo.....	101
Tabla 22. Test de hardware	103
Tabla 23. Consumo de agua en el primer día de pruebas	112
Tabla 24. Consumo de agua luego de 7 días de pruebas.....	113

Tabla 25. Costos de hardware	115
Tabla 26. Costos de infraestructura.....	116
Tabla 27. Costos de ingeniería.....	117
Tabla 28. Costos de software	118
Tabla 29. Costo total de implementación.....	118

RESUMEN

Este proyecto describe el desarrollo de un sistema electrónico para automatizar el proceso de riego en cultivos basado en redes neuronales artificiales, para este caso el cultivo de muestra es de plantas de alfalfa. Un sistema de riego obedece a las condiciones ambientales tales como: temperatura ambiente, humedad relativa, humedad del suelo y luminosidad.

El sistema hace uso de una red de sensores implementada en (Domínguez, 2020) donde se recolectan datos de variables ambientales mencionadas anteriormente; estos datos van a ser utilizados primeramente en el proceso de entrenamiento y aprendizaje de la red neuronal artificial y posteriormente, servirán para poner en funcionamiento el algoritmo de aprendizaje automático en tiempo real.

En la Granja La Pradera la técnica de riego que ha perdurado en la zona de forrajes (alfalfa) es riego por inundación que no es muy eficiente debido a que no se tiene una distribución uniforme de riego en todas las áreas del cultivo y utiliza mayor cantidad de agua, además de esto el sistema de riego se pone en funcionamiento de forma manual, por lo cual en el presente proyecto se desarrolla un moderno sistema de riego inteligente basado en redes neuronales artificiales para cultivos de alfalfa, el control está formado por datos de entradas y salidas de la RNA, donde las entradas corresponden a las variables ambientales y la salida corresponde a si es o no necesario activar el proceso de riego.

Finalmente, para comprobar la eficiencia de este sistema inteligente de riego se realizan pruebas de campo en el lugar del cultivo y se realiza la comparación de resultados en la producción de alfalfa entre el sistema de riego manual y el sistema de riego inteligente.

ABSTRACT

This project describes the development of an electronic system to automate the irrigation process in crops based on artificial neural networks, for this case the sample crop is alfalfa plants. An irrigation system responds to environmental conditions, such as ambient temperature, relative humidity, soil moisture and luminosity.

The system makes use of a sensor network implemented in (Dominguez, 2020) where data of environmental variables mentioned above are collected; these data are going to be used first in the training and learning process of the artificial neural network and later, they will serve to put into operation the automatic learning algorithm in real time.

At La Granja La Pradera, the irrigation technique that has lasted in the forage area (alfalfa) is flood irrigation which is not very efficient due to the fact that there is not a uniform distribution of irrigation in all areas of the crop and it uses a greater amount of water. In addition to this the irrigation system is put into operation manually. Therefore, in the present project an intelligent irrigation system based on artificial neuronal networks for alfalfa crops is developed. The control is formed by data from inputs and outputs of the ANN, where the inputs correspond to the environmental variables and the output corresponds to whether or not it is necessary to activate the irrigation process.

Finally, in order to check the efficiency of this intelligent irrigation system, field tests are carried out on the site of the crop and the results of alfalfa production are compared between the manual irrigation system and the intelligent irrigation system.



1. Capítulo I. Antecedentes.

En este capítulo se encuentran detalladas las bases para el desarrollo del presente trabajo de titulación, siendo éstos: el tema, la problemática, los objetivos, el alcance y la justificación, con la finalidad de expresar la importancia de la realización e implementación de este proyecto.

1.1. Tema

DISEÑO DE UN SISTEMA PARA EL CONTROL DE RIEGO MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO APLICADA A LA AGRICULTURA DE PRECISIÓN EN LA GRANJA LA PRADERA DE LA UNIVERSIDAD TECNICA DEL NORTE

1.2. Planteamiento del problema

En la granja “La Pradera” de la Universidad Técnica del Norte en el sector de Chaltura, se tienen amplias áreas de sembríos con distintos tipos de cultivo. Los mencionados sembríos han reactivado su función gracias a varios profesores y alumnos de la Facultad de Ingeniería en Ciencias Agropecuarias y Ambientales específicamente de la Carrera de Ingeniería Agropecuaria.

Para el riego en estas áreas de sembríos se hace uso de la técnica de riego por goteo, la cual funciona a base de mangueras que son agujeradas que esta ubicadas por toda el área del sembrío, y de esta forma, cada área del cultivo recibe la suficiente cantidad de agua, pero aun así esta sigue siendo una técnica artesanal, manual y que está expuesta a distintos fallos por la falta de monitoreo que existe en el lugar.

En el análisis de la conducta de los sembríos se debe tomar en cuenta varios parámetros que permitan al agrario evaluar el desarrollo de los cultivos y la manera de actuar en caso de que se presente algún tipo de eventualidad en los sembríos. Concretamente, en estas áreas de cultivos no se tienen implementado un sistema que permita a los involucrados con los cultivos tener un control de los distintos parámetros que están inmersos en el desarrollo y crecimiento de las plantas, entre dichos parámetros están considerados la temperatura del ambiente, la humedad del suelo, la iluminación, e incluso los niveles de CO₂ del área donde se encuentra el cultivo.

Se propone la implementación de un Sistema de Riego basado en redes neuronales artificiales la cual es un algoritmo de Aprendizaje Automático, el cual determinará cuando el cultivo necesita del riego, haciendo que las plantas reciban una dosificación adecuada de agua y así lograr una mejora en la producción agrícola.

Por tanto, para tener un control del desarrollo y crecimiento de las plantas de los cultivos se debe hacer una revisión de campo que consiste en observar personalmente todas las áreas de sembríos y además no se tiene una forma de almacenar la información de una forma ordenada de cada uno de los parámetros que influyen en el buen desarrollo de los cultivos para que posteriormente sea analizada. Con el pasar de los años se ha visto un gran avance en el área tecnológica y también han ido ingresando nuevas áreas como la Inteligencia Artificial que puede también ser aplicada al área de agricultura de precisión, lo cual nos va a permitir tener un mejor monitoreo del estado de los cultivos y así obtener productos de mejor calidad.

1.3. Objetivos

1.3.1. Objetivo General

Diseñar un sistema que permita controlar el riego de forma automática de los sembríos en la Granja “La Pradera”, aplicando técnicas de Aprendizaje Automático para ayudar a reducir el consumo innecesario de agua y lograr obtener productos de mejor calidad.

1.3.2. Objetivos Específicos

Buscar y analizar repositorios, bases de datos y bibliotecas virtuales, en las cuales se pueda realizar el estudio de los parámetros como: humedad del suelo, humedad relativa, temperatura ambiental, niveles de iluminación.

Investigar sobre el algoritmo de redes neuronales artificiales, específicamente como este algoritmo hace el tratamiento de la información recogida de una red de sensores, para realizar las predicciones de riego en base a los parámetros que inciden en la agricultura de precisión.

Automatizar el sistema de riego por goteo implementando el algoritmo de redes neuronales artificiales.

Realizar las pruebas de funcionamiento del Sistema de Control de riego en los cultivos.

1.4. Alcance

En los cultivos, con el análisis de cada uno de los parámetros que influyen en el desarrollo de las plantas tales como: temperatura ambiental, humedad del piso, humedad relativa, niveles de

iluminación y niveles de CO₂, se pretende analizar el comportamiento de estas y poder determinar cómo inciden en el desarrollo de los sembríos.

Haciendo uso del análisis de los parámetros anteriormente mencionados, específicamente los valores de humedad del piso, se realizará un sistema de control automático del riego en los sembríos, para lo cual se realizará una investigación sobre el algoritmo de redes neuronales artificiales el cual es usado en el área Aprendizaje Automático, para poder realizar las predicciones de cuando el cultivo requiere de riego.

Para realizar la implementación del algoritmo se va a hacer uso del software de programación Python, mediante el cual se vamos a receptar los datos recolectados de una red de sensores para que posteriormente sean analizados por el sistema y este pueda determinar si el cultivo requiere o no de riego de agua y de esa forma asegurar que la planta reciba una dosis adecuada para su buen desarrollo.

Una vez que los datos hayan sido receptados y analizados por el software y si este determina que se necesita de riego en el sembrío, se enviará de forma inalámbrica la orden para que la electroválvula permita el paso de flujo de agua hacia el sembrío, hasta que el sistema considere que la dosis de agua suministrada es suficiente y así evitar el riego innecesario de agua en los cultivos ayudando de esa forma a que la planta pueda tener un desarrollo óptimo, también se tendrá una considerable reducción del consumo de agua y además el trabajo de riego que se usa en la actualidad dejará de ser manual y artesanal pasando a ser un sistema automatizado.

Para la comprobación de que el sistema propuesto está suministrando la cantidad adecuada de agua al cultivo se procederá a realizar varias pruebas de funcionamiento y de ser necesario

realizar los ajustes pertinentes para garantizar que el sistema de riego tiene un óptimo funcionamiento.

1.5. Justificación

El uso de Inteligencia Artificial para el análisis de datos de procesos agrícolas es un tema que está en reciente desarrollo en la actualidad. Hoy por hoy, muchos de los datos en el área agrícola son tomados de forma manual, y no se tiene una forma de almacenamiento ordenado de esta información recolectada, ni se le da el respectivo proceso de análisis, para obtener mejoras en el desarrollo de los sembríos. Una alternativa muy buena para superar estos inconvenientes, es la ejecución de la agricultura de precisión, la cual, con el uso de una base científica, documental y tecnológica, permita aprovechar al máximo los recursos que se tienen disponibles.

El Plan Nacional del Buen Vivir del año 2017 al 2021 plantea la Revolución del Conocimiento, la cual propone la innovación, ciencia y tecnología, como base para el cambio de la matriz de producción. Mediante el presente proyecto se pretende hacer uso de Inteligencia Artificial en los cultivos de la granja “La Pradera”, lugar del cual se van a obtener los datos para posteriormente realizar un análisis y tratamiento de esta información, haciendo uso de los parámetros, más importantes que intervienen en el desarrollo de los cultivos, con el único objetivo de implantar la agricultura de precisión en dichas áreas de cultivos. Además, con el desarrollo de este proyecto se ayuda al cumplimiento de la misión de la Universidad Técnica del Norte, la cual trata de generar procesos de investigación, de promover el conocimiento científico, tecnológico y de innovación. Con la penetración tecnológica que se ve hoy en día en la mayoría de las áreas, el sector de la agricultura genera una gran cantidad de datos antes y después de la producción, siendo muy necesario el uso de nuevas tecnologías y técnicas que permitan analizar estos datos y de esta

manera tener un mejor control sobre los sembríos, para así obtener productos de calidad y aprovechar al máximo todos los recursos que se tienen disponibles.

2. Capítulo II. Fundamentación Teórica

En este capítulo se presenta la recopilación bibliográfica de la investigación realizada para el desarrollo del presente proyecto, se describe información referente a las formas de cultivos, parámetros que influyen en el desarrollo y producción de la siembra los cuales pueden ser medidos, tipos de riegos que se pueden utilizar. También se habla acerca de temas de comunicación inalámbrica en este caso tecnología LoRaWAN. Finalmente se trata sobre conceptos de aprendizaje automático específicamente sobre Redes Neuronales Artificiales que tienen el objetivo de determinar el momento exacto de inicio y parada del sistema de riego.

2.1. Agricultura de precisión

La agricultura de precisión es un conjunto de técnicas que están encaminadas a la optimización del uso de insumos agrícolas y a la mejora de resultados en la producción de alimentos de calidad bajo el concepto de gestión de cultivos, fundamentado en la presencia de variabilidad en campo. (Bongiovanni, Mantovani, Best, & Rogel, 2006)

Mediante el uso de tecnologías actuales como GPS, sensores, satélites e imágenes aéreas se puede recopilar información la cual nos permite analizar muchos de los factores que inciden directamente con los resultados de las cosechas. Las decisiones tomadas a partir de la información recolectada permiten optimizar los resultados tomando en cuenta diferentes puntos de vista, tales como: (Rojas, Lezcano, & Medina, 2016)

- La visión agronómica la cual permite realizar arreglos acordes a la necesidad de las plantas, lo que afecta de una manera positiva sobre los resultados en la producción.

- Desde el punto de vista del medio ambiente se puede reducir los impactos dañinos que podrían tener las técnicas aplicadas en el cultivo.
- Por último, con relación al aspecto económico se puede lograr una mejora en el uso de recursos, de tal manera que se obtenga una considerable reducción en gastos lo que permitirá al agricultor muchas más ganancias y ser más competitivos en el mercado con productos de mejor calidad.

Entre las ventajas que presenta el uso de este tipo de técnicas tenemos: Reducción de costos en el uso de insumos, mayores rendimientos en los cultivos con la misma cantidad de materiales y mayor calidad de los productos en las cosechas. (García & Flego, 2015)

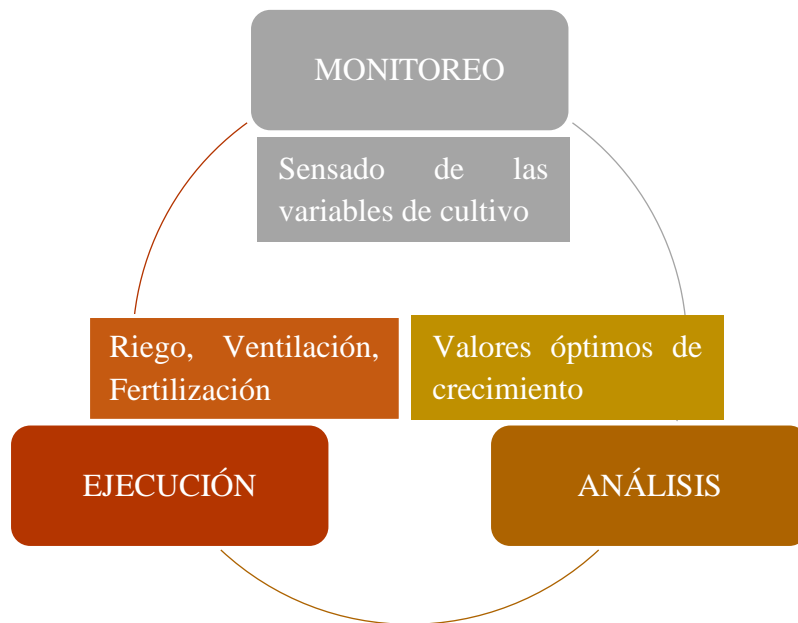


Figura 1. Esquema Agricultura de Precisión.
Fuente. Autoría

2.2. Factores o Variables a medir

Según (InfoAgro, 2010), las variables que influyen en el buen desarrollo del cultivo de alfalfa son: Humedad del suelo, Humedad Relativa, Temperatura Ambiental y Luminosidad, las cuales tienen mucha importancia para determinar cuándo es posible o no activar el proceso de riego.

2.2.1. Humedad del suelo

Esta variable hace referencia a la cantidad de agua que existe por volumen de tierra en el área de cultivo. El correcto manejo de esta variable, ayuda a la mejora de la producción de las plantas. Si las plantas quieren absorber agua del suelo deben ejercer una fuerza de absorción mayor que la tensión con la que el agua está retenida, esta tensión con la que el suelo retiene el agua, oscila entre 0,1 y 0,3 atmósferas. Además, se debe mencionar que los valores de humedad experimentan continuas variaciones a lo largo del tiempo ya que el suelo recibe agua no solamente del riego sino también de las lluvias además de que también se producen pérdidas por escorrentía superficial, por drenaje a las capas profundas y también por evo transpiración (Ibañez, 2006).

La humedad del suelo en los cultivos varía conforme a los requerimientos de la planta en sí. La ventaja de tener cultivos bajo monitoreo constante es que podemos tener un control de este factor y lograr un buen desarrollo de la planta, ya que el exceso o falta de humedad puede ocasionar que la planta no llegue a completar su desarrollo. El aporte de agua en caso de riego por inundación es de 1000 m³/ha y en el caso de riego por aspersión será de 880 m³/ha. (Leon, 2002)

2.2.2. Humedad Relativa

La humedad relativa es uno de los factores medioambientales que influyen en un cultivo, la cual representa la cantidad de vapor de agua contenida en el aire. Las plantas tienen que transpirar agua para poder transportar nutrientes, para refrigerarse y para regular su crecimiento (Huertas, 2008).

Si la humedad relativa es demasiado baja, el crecimiento de la planta se verá afectado ya que los cultivos tardarán más tiempo en alcanzar el tamaño adecuado para su posterior consumo o venta, y si por el contrario la humedad relativa es demasiada alta, las plantas llegarán a contraer enfermedades tanto en las hojas como en sus raíces y sus hojas, por lo tanto, estas van a necesitar de pesticidas para poder controlar estas enfermedades a lo que hará que la planta tenga un crecimiento débil. (Peery, 2017)

2.2.3. Temperatura Ambiental

En el desarrollo de los cultivos, la temperatura es uno de los factores más importantes, es por esto que la posibilidad de crear condiciones climáticas se ha convertido en una de las ventajas primordiales de los invernaderos. Controlando este factor, se puede prevenir daños en los cultivos debido a las heladas o a las altas temperaturas. Para sus procesos de crecimiento y correcto desarrollo, las plantas necesitan de una temperatura adecuada, de no ser así, estos procesos se detienen. Cuando este factor desciende a cero grados o menos, las plantas pueden sufrir daños severos en sus tejidos, así como suele suceder cuando se encuentran al en el aire libre durante las heladas nocturnas. En general, el efecto favorable que produce el invernadero sobre el desarrollo

de las raíces y del cultivo es mantener la adecuada temperatura tanto del aire como del suelo (Barrios, 2004).

La semilla del cultivo de muestra en este caso alfalfa germina a temperaturas de 2°C a 3°C, siempre y cuando las demás condiciones ambientales lo permitan. A medida que se incrementa la temperatura la germinación es más rápida hasta llegar a un valor óptimo a los 28-30°C. Si las temperaturas llegasen a ser superiores a los 38°C sería letal para las plántulas. La temperatura media para para la producción forrajera es alrededor de 15°C, siendo el rango óptimo de temperaturas, según el tipo, de 18°C a 28°C; aunque existen variedades de esta planta que toleran temperaturas muy bajas de hasta -10°C (InfoAgro, 2010).

2.2.4. Luminosidad

Las condiciones de iluminación, como elemento fundamental, para un invernadero, así como la calidad de y cantidad de luz son las que determinan las posibilidades biológicas y agronómicas del mismo (Pacheco, 2010).

Esencialmente toda la luz visible es capaz de promover la fotosíntesis, pero las regiones de 400 a 500 y de 600 a 700 nm son las más eficaces. (Iglesias, 2009).

2.3. Sistemas de Riego

Desde tiempos antiguos, el riego se ha utilizado para cubrir las necesidades de agua en los sembríos cuando no existe la suficiente precipitación. Entre los métodos de riego tradicionales o que eran más usados se tienen los surcos en la tierra para canalizar el agua (huerta tradicional) y

los que inundaban los terrenos (caso de cultivos de arroz). Con el pasar de los años los métodos de riego han ido mejorando hacia sistemas mucho más eficientes (Martínez, 2017). A continuación, se hace mención a algunos de ellos: Riego por aspersión, Riego por micro-compuertas, Riego por inundación y Riego por goteo.

2.3.1. Riego por aspersión

Este sistema de irrigación se lo hace uso principalmente en siembra directa para lograr una buena germinación y para proteger a los cultivos recientemente sembrados contra las heladas. Este sistema hace que el agua llegue a las plantas en forma de lluvia tal como se muestra en la Figura 2, así se logra dosificar con una buena precisión. Sin embargo, en regiones donde exista altas temperaturas no se recomienda usarlo para ciertos cultivos como tomate, cebolla, sandía, maíz dulce, entre otros; debido a que favorece el ataque de hongos hacia las plantas de la siembra (Seminis, 2016).

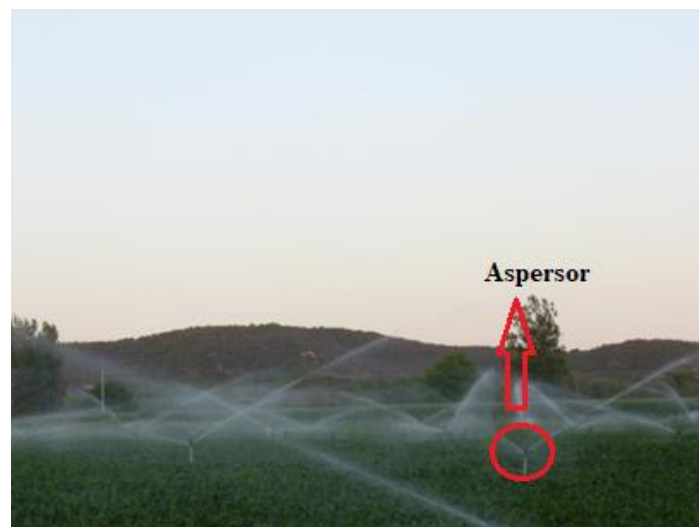


Figura 2. Riego por Aspersión
Fuente. (Pevifer, 2009)

2.3.2. Riego por micro-compuertas

El riego por micro-compuertas, es la versión actual y mucho más técnica del riego por gravedad; en él, se hace uso de tuberías para conducir y distribuir el agua directamente en cada uno de los surcos del sembrío, logrando mitigar las pérdidas producidas en los canales y regaderas lo que eleva la eficiencia de este sistema de riego (SIRHC Hidráulica y Construcciones, 2016).

Este tipo de técnica de riego consta de tubos, válvulas y conexiones de PVC, con protección UV y empalme de unión flexible, de fácil ensamblaje y desarmado (Ministerio de Agricultura y Riego, 2016).



Figura 3. Riego por Micro-compuertas
Fuente. (SIRHC Hidráulica y Construcciones, 2016)

2.3.3. Riego por goteo

El riego por goteo radica en un conjunto de tuberías interconectadas entre sí, las cuales tienen pequeños orificios o goteros, como se muestra en la figura 4, que se sitúan al pie de las plantas y por donde el agua va cayendo según la velocidad programada, abriendo o cerrando las válvulas que permiten el paso del agua (Martínez, 2017).

El riego por goteo destaca por su considerable ahorro de agua, esta técnica de irrigación reduce el consumo en un 50% en comparación con el riego por aspersión, además reduce los riesgos de enfermedades pues no humedece ni el follaje ni los frutos, con esto se logra una disminución en el crecimiento de maleza y, si se consigue un buen manejo de esta técnica de riego, se puede evitar la erosión del suelo (Seminis, 2016).

Entre los componentes del riego por goteo se puede mencionar los siguientes:

Las fuentes de abastecimiento de agua para el equipo pueden provenir de procedimientos de entrega de agua de la red de riego en zonas donde es necesario realizar la extracción de agua subterránea a través de perforaciones. La extracción de agua se puede acumular en reservorios tal como se muestra en la figura 5, cuyas superficies dependen del área a irrigar, su función es la de abastecer de agua en forma permanente al sistema de riego (Liotta, Carrión, Ciancaglini, & Olgún, 2015) o también puede ser tomada directamente de tuberías de agua potable en algunos casos.



Figura 4. Reservorio de Agua
Fuente: (Ministerio de Agricultura y Ganadería, 2017)

Este equipo está formado por una o varias bombas (Figura 5) cuyo tamaño y potencia depende del área que se vaya a regar. Para lo cual el dimensionamiento de la bomba debe ser tal que la presión que se requiere sea suficiente para vencer las diferencias de cota y las pérdidas de carga de todo el sistema. Las bombas más comúnmente usadas son las del tipo centrífuga las cuales son abastecidas por energía eléctrica (Liotta, Carrión, Ciancaglini, & Olgún, 2015).



*Figura 5. Equipo de bombeo de agua
Fuente. (Quindiaguas, s.f.)*

El sistema de filtrado (Figura 6) es una parte importante del sistema ya que ayuda a evitar uno de los problemas más graves que suele presentarse en los equipos de riego como es la obstrucción del gotero, lo cual se puede producir por las siguientes causas: Partículas minerales en suspensión (arcilla, limo y arena), Materia orgánica y Precipitados (principalmente carbonatos) (Liotta, Carrión, Ciancaglini, & Olgún, 2015).



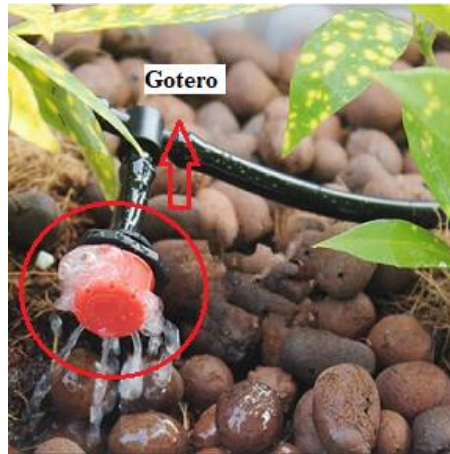
*Figura 6. Sistema de filtrado para riego por goteo
Fuente. (ITSA - Sistemas Agrícolas, 2016)*

Las tuberías y mangueras (Figura 7) más usadas para este tipo de sistemas de riego son las de cloruro de polivinilo (PVC) y de polietileno. El PVC se usa en diámetros superiores a 50mm para las líneas de repartición primaria, secundaria y terciaria. Los diámetros más comunes son de 50mm, 63mm, 75mm, 90mm, 110mm y 160mm. Las tuberías principales tienen diámetros superiores a los 200mm en grandes instalaciones. Estas pueden ser clasificadas por clases en relación a la presión que son capaces de soportar, por ejemplo 4, 6, 10, etc. que representa la presión máxima de trabajo expresada en kg/cm^2 (Liotta, Carrión, Ciancaglini, & Olgúin, 2015).



*Figura 7. Mangueras PVC de conducción.
Fuente. Autoría*

Existen una gran variedad de goteros de diferentes formas y configuración para disipar la presión. Los más utilizados operan con caudales entre 1 y 4 litros/hora. Los principales goteros que se usan actualmente son: de laberinto y tipo Vortex o de botón.



*Figura 8. Gotero de botón regulable
Fuente. Autoría*

La electroválvula, mostrada en la Figura 9, es una válvula que se instala en el campo para permitir el suministro de agua al campo de riego. En este caso es una válvula electrónica que nos permite realizar el riego de forma automatizada.



*Figura 9. Electroválvula solenoide
Fuente. (Alvarez, 2013)*

2.4. Aprendizaje Automático

El aprendizaje automático (o Machine Learning, por su nombre en inglés) es la rama de la Inteligencia Artificial que tiene como objeto desarrollar métodos que permitan a las computadoras aprender. De una forma más clara, se trata de desarrollar algoritmos capaces de generalizar comportamientos y reconocer patrones a partir de una información entregada en forma de ejemplos. A un nivel muy simple, se puede decir que una de las tareas del Aprendizaje Automático (AA) es intentar extraer conocimiento sobre algunas propiedades no observadas de un objeto basándose en las propiedades que si han sido observadas o de las que si se tiene registro, es decir, consiste en predecir un comportamiento futuro a partir de información pasada (Caparrini, 2017).

Una de las particularidades primordiales de las Redes Neuronales Artificiales (RNA) es su adaptabilidad y su susceptibilidad de aprendizaje, a través de la modificación de los pesos de las interconexiones entre varias neuronas (Lara, 2014).

Hay dos formas de aprendizaje: aprendizaje supervisado y aprendizaje no supervisado.

2.4.1. Aprendizaje no supervisado

El *aprendizaje no supervisado* se da cuando no se dispone de datos “etiquetados” para el entrenamiento de la Red Neuronal. Solo se conocen los datos de entrada, pero no existen datos de salida que pertenezcan a un determinado input. Por lo tanto, sólo se puede representar la estructura de los datos, para intentar encontrar algún tipo de organización que logre facilitar el proceso de análisis. Por ello, tienen un carácter de prueba (Recuero de los Santos, 2017).

En el aprendizaje no supervisado se hace uso de la función de agrupación, por lo que el algoritmo debe clasificar los datos por similitud y de esa manera crear los grupos, sin tener la capacidad de precisar cómo es cada individualidad de cada uno de los datos que conforman los grupos (Zambrano, 2018).

Los tipos de algoritmos mayormente utilizados en aprendizaje no supervisado son los siguientes: Algoritmos de clustering, Aprendizaje Hebbiano y Aprendizaje competitivo y comparativo (Recuero de los Santos, 2017).

2.4.2. Aprendizaje Supervisado

El aprendizaje supervisado requiere de la presencia de un tutor externo, es decir, que el proceso de aprendizaje se realiza por medio de un entrenamiento que está bajo control de un agente externo el cual establece la respuesta que debería generar la red a partir de una entrada determinada. El supervisor o tutor se encarga de controlar la salida de la red y en caso de que esta no concuerde con la esperada se procede a modificar los pesos de las conexiones, con el objetivo de lograr que la salida alcanzada se aproxime a la esperada para esto se hace uso de un conjunto de datos previos denominados datos de entrenamiento los cuales incluyen datos de entrada y valores resultantes, de todo ese conjunto de datos se utiliza el 70% como datos de entrenamiento y el 30% restante como datos de prueba para validar el correcto funcionamiento del algoritmo (Llanos & Romero, 2013).

Según (MathWorks, 2017) dentro del aprendizaje automático supervisado están dos categorías de algoritmos, las cuales son: Clasificación y Regresión.

Algoritmos de *Clasificación* usados para valores de respuesta categóricos, en donde los datos se pueden dividir en clases definidas. (MathWorks, 2017). El propósito de este tipo de algoritmo es inducir un modelo que pueda predecir un comportamiento dados los valores de los atributos (Sucar, 2018). Entre los algoritmos de clasificación más comunes se tiene los siguientes: Máquinas de vectores de soporte, Redes Neuronales, Clasificador Naïve Bayes, Árboles de decisión, Análisis discriminante y Vecinos más cercanos (kNN).

Algoritmos de *Regresión* usados para valores de respuesta continua. El propósito es inducir un modelo que pueda predecir el valor de una clase dados los valores de los atributos. Entre los algoritmos de regresión que se usan están, por ejemplo: Árboles de regresión, regresión lineal y no lineal, Redes Neuronales, Árboles de decisión y Modelos Lineales generalizados (Sucar, 2018).

2.5. Redes Neuronales Artificiales

En los últimos años, las Redes Neuronales Artificiales (RNA) han recibido un interés especial como tecnología para minería de datos, ya que tiene los medios para poder modelar de manera eficaz problemas que sean grandes y complejos. Estos modelos de RNA son encaminados a partir de los datos, es decir, están en la capacidad de localizar relaciones (patrones) de manera inductiva por medio de los algoritmos de aprendizaje basados en los datos existentes (Salas, 2005).

2.5.1. Reseña Histórica

El primer modelo matemático de una RNA viene de la mano de Warren McCulloch neurofísico y Walter Pitts un matemático, en el año 1943. Este concepto se dio a conocer mediante la publicación de un libro llamado: “A logical calculus of the ideas immanent in nervous activity”

en el cual se mostraba un análisis matemático del funcionamiento del cerebro y además mostraron el primer modelado de una neurona simple, que fue construida con circuitos electrónicos de esa época, a la cual llamaron: “Perceptrón” (Caparrini, 2017).

En 1982 se daría el renacimiento de las redes neuronales artificiales de la mano de John Hopfield del Instituto Tecnológico de Massachussets al publicar el libro “Neural Computation of Decision in Optimization Problems” que mostraba la idea de un conjunto de neuronas, conjunto de perceptrones que al ser combinados en una estructura podrían solucionar problemas no-lineales, y lo demostraron con el mismo problema que años atrás había destruido al perceptrón (López, 2007).

Desde entonces las RNA han ido evolucionando constantemente, teniendo ahora un gran número de alternativas que van desde un multi-layer perceptrón con Back Propagation, hasta lo más conocido como: Redes Neuronales Difusas (López, 2007).

2.5.2. Fundamentos Biológicos de las Redes Neuronales Artificiales

Las Redes Neuronales Artificiales se fundamentan en el comportamiento del sistema neuronal del cuerpo humano, en el cual encontramos tres elementos principales: Los órganos receptores los cuales son encargados de recoger datos o información del exterior, el sistema nervioso que transmite, analiza y guarda la información elaborada, y finalmente los órganos efectores que cogen la información de parte del sistema nervioso y la convierte en cualquier acción que se requiera ejecutar (Solomatine, 2002).

Las redes Neuronales Artificiales se asimilan al cerebro por presentar una serie de particularidades propias del mismo, como el aprender de hechos pasados. Por ejemplo, las RNA, aprenden de la experiencia, sistematizando de ejemplo previos a ejemplos nuevos, es decir desglosan las características principales de una serie de datos. Las RNA pueden cambiar su conducta en función del ambiente. Se les muestra un conjunto de entradas y ellas mismas se ajustan para producir unas salidas consistentes, es decir estas pueden aprender. Además, pueden generalizar de forma automática datos, debido a su propia estructura y naturaleza. Estas redes pueden brindar, dentro de un margen, respuestas correctas a entradas que muestran pequeñas variaciones ocasionadas por los efectos de ruido o distorsión (Castellano, 2009).

2.5.3. Neurona Biológica

Las redes neuronales artificiales se basan en el comportamiento de las redes neuronales biológicas, por este motivo es necesario empezar comprendiendo el funcionamiento de una neurona biológica, la cual sin entrar en mayor detalle se componen de 4 partes las cuales se pueden ver en la Figura 10, y son: dendritas, sinapsis, soma y axón (López D. , 2007).

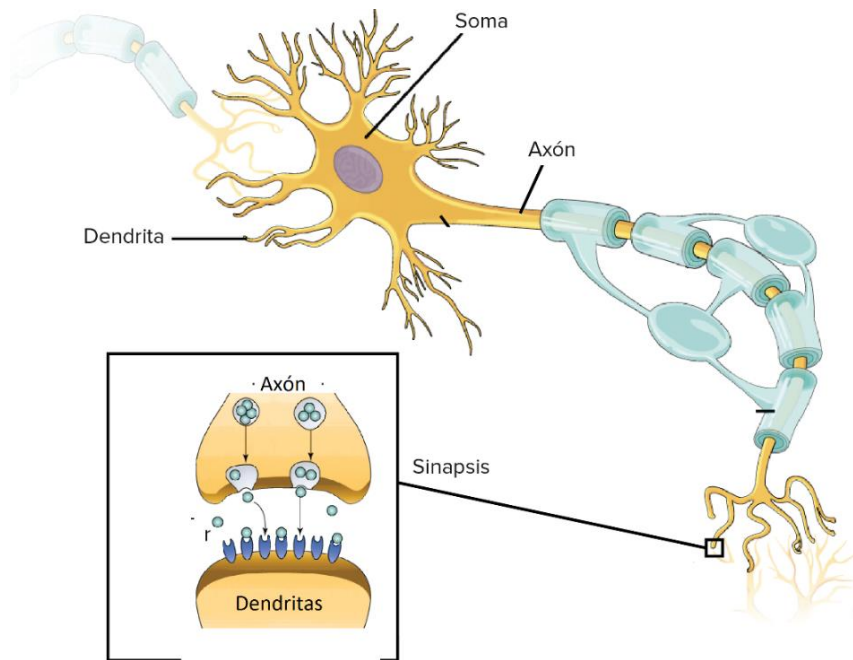


Figura 10. Partes de una neurona biológica
Fuente. (Khan Academy, 2016)

2.5.3.1. Dendritas

Las dendritas son capilares terminales que posee cada neurona cuya función es la recepción de estímulos y la alimentación celular. Estas funcionan como receptores de impulsos nerviosos que provienen desde un axón que pertenece a otra neurona. Su principal función es la de receptor los impulsos de otras neuronas y trasladarlas hasta el soma de la neurona (Lexikon, 2005).

Los impulsos neuronales son cargas electro-químicas que se producen por el paso de iones eléctricos de varios componentes químicos como el potasio y sodio. A lo largo de las dendritas existen las espinas dendríticas que son pequeñas prolongaciones que son sitios de sinapsis (López, 2007).

2.5.3.2. Sinapsis

Es una membrana permeable que se localiza en el extremo de cada dendrita. Esta sinapsis altera el impulso eléctrico que entra por la dendrita, atenuándolo o amplificándolo, de tal forma que es tomada como la “*ganancia*” de una dendrita. La ganancia que tiene la dendrita es la causante directa del resultado que la neurona genere, de esta forma es considerada a este nivel como la “*memoria*” de la neurona, puesto que depende de la sinapsis para que una neurona se comporte de una forma u otra (López, 2007).

2.5.3.3. Soma

El soma o cuerpo celular es el centro metabólico de la neurona, es el sitio donde se producen las moléculas y se realizan las acciones esenciales para conservar la vida y las funciones de la célula nerviosa o neurona (Saura, 2012).

En esta parte de la neurona se realiza la síntesis de la mayor parte de proteínas que se encuentran en la neurona, y entre ellas las que van a dar comienzo a la síntesis de la mayor parte de neurotransmisores. Es a partir de aquí que se derivan otras partes de la neurona como el axón y las dendritas (Castillero, 2016).

Es aquí donde se procesan todas las señales que llegan a través de las dendritas para formar una señal resultante. Básicamente este proceso se divide en dos etapas: Suma de señales, simplemente se hace una sumatoria de todas las señales que se obtienen de las dendritas. Función de Transferencia, una vez que se obtiene la señal resultante, esta pasa por una función de transmisión propia de cada tipo de neurona, y se genera una señal resultado (López, 2007).

2.5.3.4. Axón

El Axón es la salida de la neurona, es decir, es la señal que transporta el resultado que ofrece el SOMA luego de procesar las señales provenientes de las dendritas (López, 2007). Al final de los axones hay una parte de fibra nerviosa que se contrae al momento en que la señal eléctrica llega a su destino, a esta parte se la llama espacio sináptico que es el punto en el cual las neuronas se comunican entre sí por medio de señales químicas, es decir, en la terminación de cada axón el impulso eléctrico se transforma en un patrón de liberación de partículas químicas que llegan a la siguiente neurona por medio del espacio sináptico (Triglia, 2016).

2.5.4. Neurona Artificial

Cualquier modelo de red neuronal consta de módulos fundamentales de procesamiento, para este caso las neuronas. A partir de estas, se pueden formar representaciones específicas, de tal manera que un estado conjunto de estas pueda representar una letra, un número o cualquier otro objeto. La tecnología basada en RNA es de carácter bio-inspirado pues su funcionalidad trata de imitar la conducta de elementos biológicos que en este caso componen las neuronas del cerebro. (López J. , 2000)

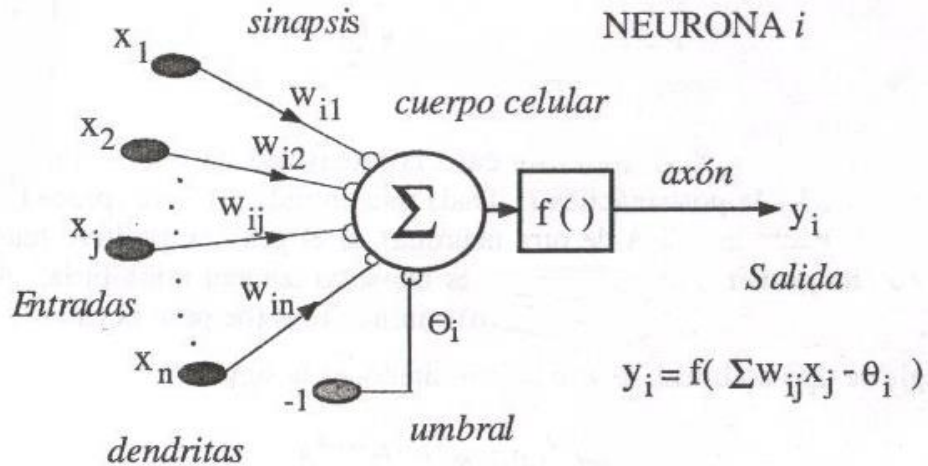


Figura 11. Modelo de una Neurona Artificial de una con n entradas y una salida
Fuente. (Palacios, 2003)

Una neurona artificial es un procesador básico, en el sentido de que procesa un vector X ($X_1, X_2, X_3, \dots, X_N$) de entradas y provoca una o varias respuestas o salidas, dependiendo de la aplicación para la cual vaya a ser usada. Los elementos clave de una neurona artificial se pueden observar en la Figura 11, los cuales son: función de entrada, pesos, regla de propagación, función de activación y función de salida.

La nomenclatura utilizada en la Figura 11 es la siguiente: x_{ij} = valores de entrada de la neurona, w_{ij} = pesos correspondientes a cada una de las entradas e y_i = Salida de la neurona N_i . El conjunto de las x entradas $x_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{ij})$ es generalmente conocido como “vector entrada” (Matich, 2002).

2.5.4.1. Entradas

Según se menciona en (Palacios, 2003), las entradas son las encargadas de recibir la información. En comparación con una neurona biológica cumple la función de las dendritas.

En (Matich, 2002), se menciona que las funciones de entrada más conocidas y mayormente utilizadas son: Sumatoria de las entradas pesadas, Productoria de las entradas pesadas y Máximo de las entradas pesadas.

- ✓ *Sumatoria de las entradas pesadas*: es la suma de los valores de entrada de la neurona, cada uno multiplicados por sus pesos correspondientes. La Ecuación 1 mostrada a continuación representa a esta función de entrada:

$$\sum_j (x_{ij}, w_{ij}), \quad \text{con } j = 1, 2, 3, \dots, n$$

Ecuación 1. Sumatoria de las entradas pesadas

- ✓ *Productoria de las entradas pesadas*: En la Ecuación 2 se muestra la representación matemática de esta función de entrada la cual es el producto de los valores de entrada de la neurona, multiplicados por los pesos correspondientes a cada una.

$$\prod_j (x_{ij}, w_{ij}), \quad \text{con } j = 1, 2, 3, \dots, n$$

Ecuación 2. Productoria de las entradas pesadas

- ✓ *Máximo de las entradas pesadas*: Solo se toma en cuenta el valor de la entrada más fuerte, previo haber sido multiplicada por su peso correspondiente. Esta función de entrada es representada por la Ecuación 3:

$$\max_j (x_{ij}, w_{ij}), \quad \text{con } j = 1, 2, 3, \dots, n$$

Ecuación 3. Máximo de las entradas pesadas

2.5.4.2. Pesos

Las conexiones que unen a la neurona en una RNA tienen asociado un peso (W_{ij}), que es lo que hace que la red adquiera conocimiento. Una neurona normalmente recibe múltiples entradas o pesos de forma simultánea. Como ya se mencionó cada entrada tiene su propio peso relativo el cual provee la importancia de la entrada dentro de la función de agregación de la neurona. Estos pesos tienen la misma función que realizan las fuerzas sinápticas de las neuronas biológicas. En ambos casos, algunas entradas tienen mayor importancia que otras, de modo que tienen mayor efecto sobre el procesamiento de la neurona al combinarse para causar la respuesta neuronal. Los pesos son factores que pueden adaptarse dentro de la red que establecen la intensidad de la señal de entrada registrada por la neurona artificial. Ellos son la medida de la fuerza de un enlace de entrada. Estas fuerzas pueden ser reformadas en réplica de los ejemplos de entrenamiento de acuerdo a la topología específica o debido a las reglas de entrenamiento (ATGC Grupo de Tecnologías Avanzadas en Computación , 2007).

2.5.4.3. Regla de Propagación

Con esas entradas y los pesos sinápticos, se suele hacer algún tipo de operación para obtener el valor del potencial postsináptico (valor de una función de las entradas y los pesos y que es el que se utiliza en último término para realizar el procesamiento). Una de las operaciones más comunes es sumar las entradas, pero teniendo en cuenta la importancia de cada una. Es lo que se llama suma empoderada, aunque otras operaciones también son posibles (Palacios, 2003).

La Ecuación 4, representa la suma empoderada:

$$y_i(t) = \sum_j x_j w_{ij}$$

Ecuación 4. Suma empoderada

La otra regla de propagación según (Palacios, 2003) más frecuente es la distancia euclídea. Este es el tipo de regla que tienen redes como el SOM o las RBF.

2.5.4.4. Función de Activación

El valor de obtenido con la regla de propagación, se filtra a través de una función conocida como función de activación y es la que nos da salida de la neurona. Entre las neuronas que constituyen una RNA existe un grupo de enlaces que las unen. Cada módulo transmite señales a aquellas que están enlazadas a su salida. Asociada con cada unidad U_j hay una función de activación (Fact), que convierte la entrada neta que la unidad muestra como consecuencia de estar conectada con otras unidades que le proveen información, en el valor de salida Y_j (López J. , 2000).

(Matich, 2002) explica que la función de activación calcula el estado de actividad que tiene una neuronal, convirtiendo la entrada global en un valor o estado de activación, cuyo rango va de (0 a 1) o de (-1 a 1), debido a que una neurona puede estar totalmente inactiva (-1 o 0) o activa (1). La función de activación, es una función de entrada global (gx_i) menos el umbral (Θ_i). Las funciones de activación más comúnmente utilizadas son las siguientes: función lineal, función Sigmoidea y función tangente hiperbólica.

- ✓ *Función Lineal:* Los valores conseguidos por intermedio de esta función de activación serán: $a*(gx_i - \Theta_i)$, cuando el argumento ($gx_i - \Theta_i$) este comprendido dentro del rango

$(-1/a, 1/a)$. Cuando $a = 1$, la salida es similar a la entrada, tal como se muestra en la Figura 12.

$$f(x) = \begin{cases} -1 & x \leq -\frac{1}{a} \\ a * x & -\frac{1}{a} < x < \frac{1}{a} \\ 1 & x \geq \frac{1}{a} \end{cases}$$

Ecuación 5. Función de activación lineal

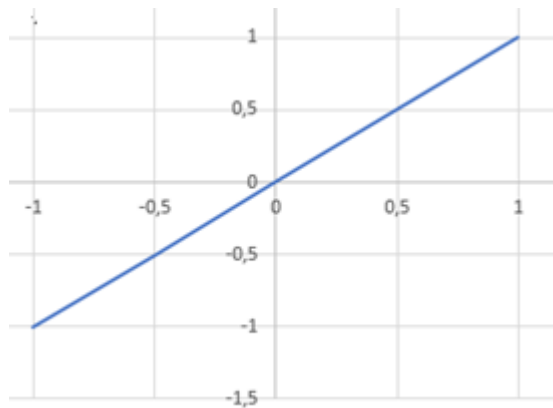


Figura 12. Gráfica Función de activación lineal
Fuente. Autoría

- ✓ **Función sigmoidea:** Los valores resultantes que proporciona esta función están comprendidos en un rango de 0 a 1. Al cambiar el valor de g , se ve afectada la pendiente de la función de activación.

$$f(x) = \frac{1}{1 + e^{-gx}}, \text{ con } x = gxi - \theta i$$

Ecuación 6. Función de activación sigmoidea

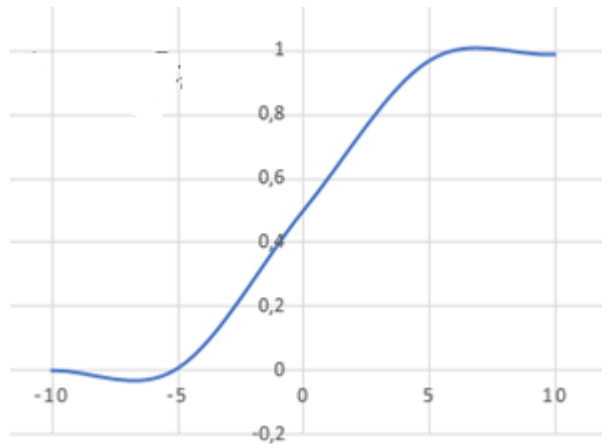


Figura 13. Función de activación sigmoidea
Fuente. Autoría

- ✓ **Función tangente hiperbólica:** Los valores de salida de esta función están incluidos en un rango de -1 a 1. Al cambiar el valor de g , se ve alterada la pendiente de la función de activación.

$$f(x) = \frac{e^{gx} - e^{-gx}}{e^{gx} + e^{-gx}}, \text{ con } x = gxi - \theta_i$$

Ecuación 7. Función tangente hiperbólica

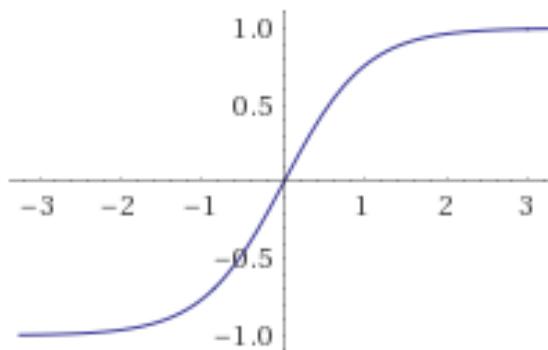


Figura 14. Función tangente hiperbólica
Fuente. Autoría

2.5.4.5. Función de salida

Cada elemento de procesamiento tiene permitido una única salida $Y_i(t)$ que puede estar asociada con un valor elevado de otras neuronas. Normalmente, la salida es claramente semejante

al valor resultante de la función de activación (ATGC Grupo de Tecnologías Avanzadas en Computación , 2007).

Normalmente, no cualquier valor es permitido como entrada para una neurona, por lo tanto, los valores a la salida están dentro del rango $[0, 1]$ o $[-1, 1]$. También pueden ser binarios $\{0, 1\}$ o $\{-1, 1\}$. Según (Matich, 2002) las funciones de salida más comunes son: función identidad (valor de salida igual al valor de entrada) y binaria (0 o 1).

2.5.5. Niveles o Capas de Neuronas

La repartición de neuronas dentro de una red se realiza formando capas de un número de neuronas fijo, donde a una capa o nivel se le conoce como un grupo de neuronas cuyas entradas proceden del mismo origen o fuente y cuyas salidas se dirigen al mismo destino (López J. , 2000).

En las neuronas se pueden distinguir tres tipos de niveles o capas: *Nivel o capa de entrada* es quien recibe la información que proviene de fuentes exteriores de la red directamente, *Nivel o capa oculta* este nivel es interno de la red y no tiene ningún contacto directo con el exterior y el *nivel o capa de salida* envían la información de la red hacia el exterior de la misma (López J. , 2000).

2.5.6. Tipos de neuronas artificiales

Las neuronas artificiales se las puede clasificar conforme a los valores que pueden tomar, entre las más comunes tenemos las *neuronas binarias*, estas únicamente toman valores que se encuentran dentro del intervalo $\{0,1\}$ o $\{-1,1\}$ y las *neuronas reales* toman valores dentro del rango $[0,1]$ o $[-1,1]$. Los pesos no están restringidos normalmente a cierto intervalo, aunque para ciertas aplicaciones esta restricción puede ser necesaria (Matich, 2011).

2.5.7. Fase de Entrenamiento

Se puede definir al aprendizaje como la modificación de la conducta inducida por la interacción con el entorno y como resultado de experiencias conducentes al establecimiento de nuevos modelos de contestaciones ante estímulos de fuentes externas. En una red neuronal biológica se suele admitir que la información retenida en el cerebro tiene relación con los valores sinápticos de la interconexión entre neuronas que con ellas mismas; es decir, que el conocimiento se logra adquirir por la sinapsis. En cambio, para el caso de las RNA, se considera que el conocimiento está representado en los pesos de las conexiones entre las neuronas. Todo proceso de aprendizaje involucra cierta cantidad de cambios en estas conexiones. En verdad, se puede decir que se logra el aprendizaje por la modificación de los pesos de la red (López J. , 2000).

El proceso de entrenamiento de la red neuronal puede llevarse a cabo a dos niveles: Por medio de modelado de la sinapsis y por destrucción y creación de neuronas.

Por medio del modelado de las sinapsis que consiste en alterar los pesos sinápticos siguiendo cierta regla de aprendizaje, construido normalmente desde la optimización de una función de coste o error, que mide la eficacia existente con relación a la operación de la red. Si denominamos $W_{ij}(t)$ al peso que conecta la neurona presináptica j con la postsináptica i en la iteración t , el algoritmo de aprendizaje, en función de las señales que en el instante t llegan procedentes del entorno facilitará en valor $\Delta W_{ij}(t)$ que da la modificación que se debe incorporar en dicho peso, el cual quedará de acuerdo a la siguiente expresión matemática (Conceptos Básicos sobre Redes Neuronales, 2011): $w_{ij}(t + 1) = w_{ij}(t) + \Delta w_{ij}(t)$.

El segundo nivel es a través de la destrucción o creación de neuronas, la cual trata de realizar modificaciones en la arquitectura de la red.

2.5.8. Problemas comunes en el entrenamiento de la red neuronal

Uno de los principales problemas más comunes de entrenar una RNA es que la red no logre ser entrenada con la suficiente precisión; es decir, que posea un porcentaje de fallos muy elevado que no se logre reducir por más veces que se le pase el juego de ensayo o que la red tome demasiado tiempo en entrenarse. Si esto sucede se debe realizar un análisis en el diseño de la red y en el juego de ensayo. Una de las opciones que se suele usar para mitigar estos problemas es el de modificar el número de niveles o capas ocultas lo que hace que el tiempo de aprendizaje aumente o disminuya, pero no hará ningún efecto a la precisión o proporción en las respuestas correctas del sistema. Si los resultados obtenidos son pobres al presentar una colección de patrones de entrada, será importante hacer una comprobación del juego de ensayo, ya que la información puede ser engañosa o contradictoria. Para eso debemos corregir o modificar el juego de ensayo añadiendo nuevos hechos o corrigiendo los ya existentes, pero al realizar este proceso debemos tener en cuenta que la red debe pasar nuevamente por el proceso de entrenamiento con la totalidad de los hechos y no solo con los nuevos o modificados. Si la red sigue presentando un comportamiento poco satisfactorio después de haber hecho todo este proceso y de verificar que la red ha recibido los suficientes hechos, es decir que el juego de ensayo es completo y cubre todos los problemas a resolver; se deberá tomar en consideración realizar un cambio en el modo de presentar la información en la entrada a la red. Otro de los problemas que se puede presentar, el cual de hecho es muy común es el del sobre entrenamiento. Este problema puede ser observado al realizar un test en una red que acabe de ser entrenada obteniendo como efectos un alto porcentaje de aciertos en

tal entrenamiento, esto puede significar que la red ha aprendido las pautas existentes en el juego de ensayo pero que en realidad no ha sido capaz de abstraer y generalizar las relaciones entre los mismos por lo cual la red no es aplicable en el mundo real. Este problema puede ser ocasionado por dos motivos, el primero puede ser que la topología de la red sea demasiado compleja, es decir que el número de capas sea muy elevado y la segunda causa puede que la red haya sido expuesta a demasiados ciclos de entrenamiento (Gestal, 2013).

2.5.9. Fase de Validación o Test

Después de la fase de aprendizaje o entrenamiento sigue la fase de ejecución, durante la cual se pide a la red que responda a estímulos que son diferentes a los que fueron presentados durante el proceso de entrenamiento. Gracias a los ejemplos aprendidos del juego de ensayo, la red debe estar en capacidad de generalizar y dar resultados correctos ante patrones de estímulos nuevos, en otras palabras, que una vez que se haya culminado el proceso de aprendizaje la red debe estar en capacidad de generalizar; es decir, se debe obtener salidas correctas ante entradas similares a las del juego de ensayo usado en el proceso de aprendizaje. Para lograr conseguir el rendimiento más óptimo de generalización, los datos que vayan a ser usados para el entrenamiento, deben lograr cubrir un rango muy amplio de hechos, este es debido a que cuando se aumenta el tamaño y variedad del juego de ensayo, se logra disminuir la necesidad de que los datos de entrada durante el proceso de trabajo normal se asemejen mucho a los patrones del juego de ensayo; es decir, que se logrará que la red pueda generalizar mejor. En vista de la gran cantidad de datos que se le proporcionará a la red, se debe buscar un criterio de selección para crear el juego de ensayo, ya que en este debe haber suficientes hechos, es decir, debe haber parejas “patrones de estímulo –

respuesta correcta”. Lo mejor es preparar una amplia cantidad de hechos de entrenamiento que envuelvan todos los problemas a los que se pueda tener que afrontar la RNA (Gestal, 2013).

2.6. Low Power Wide Área Network (LPWAN)

Las redes de área amplia de baja potencia (LPWAN, por sus siglas en inglés) son tecnologías de comunicación inalámbrica empleadas dentro del área de IoT (Internet of Things). Las tecnologías LPWAN son ideales para conectar dispositivos que requieren enviar pequeñas cantidades de datos a largas distancias y maximizando el tiempo de vida de una batería. Las principales características de las LPWAN son las siguientes: comunicación de largo alcance (hasta 15km con línea de vista), baja velocidad de transmisión (se envía menos de 5000 bits de datos por segundo) y bajo consumo de energía (una batería puede llegar a durar hasta 10 años)(Aguilar, 2019).

En la Figura 15 se puede observar un diagrama radial de las características principales de las redes LPWAN en comparación con tecnologías celulares y la tecnología Zigbee (802.15.4). El área marcada de color azul corresponde a las redes LPWAN las cuales presentan mayor cobertura geográfica, mayor rango de distancia mayor latencia de transmisión, un menor número de estaciones base, no tienen costo de uso del radioespectro, menor ancho de banda y menor consumo de energía en comparación a las redes celulares (área roja) y redes ZigBee (IEEE 802.15.4, área verde) lo que la hacen ser una tecnología ideal para la implementación de IoT.

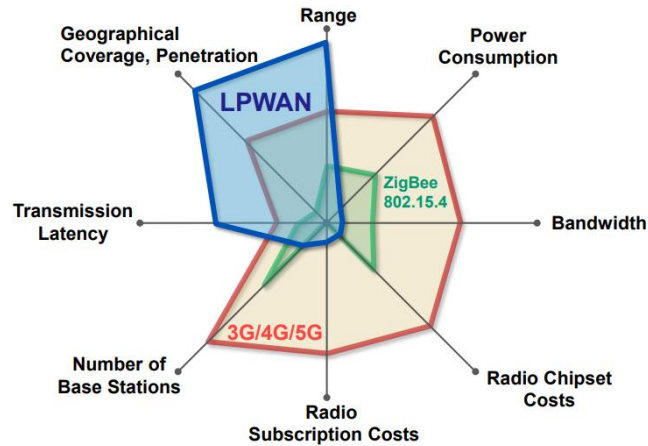


Figura 15. Diagrama radial comparativo entre redes LPWAN, redes celulares y Zigbee
Fuente. (Egli, 2016)

En la actualidad existen varios tipos de redes LPWAN, entre las más usadas y destacadas tenemos las siguientes: SigFox, Weightless, Narrow Band (NB-IoT), LTE-M y **LoRa/LoRaWAN**.

SigFox es un operador de red LPWAN francés al igual que cualquier operador de telefonía móvil. Es una de las redes con mayor despliegue en todo el mundo, la cual opera sobre la banda de frecuencia de 868MHz y 902 MHz. Utiliza la tecnología de modulación de radio Banda Ultra Estrecha o UNB (siglas en inglés de Ultra Narrow Band). Las características principales que tiene son las siguientes: Se pueden enviar hasta 140 mensajes al día con un tamaño máximo de 12 bytes cada mensaje. La comunicación puede ser bidireccional. SigFox ofrece 4 mensajes de bajada de 8 bytes cada uno. Consigue una velocidad de transmisión de hasta 100 bps (del Valle Hernández, 2017).

Weightless es un estándar abierto que opera bajo el espectro sin licencia, lo que permite que operador de telefonía y el proveedor del hardware sean totalmente independientes, pero menos comerciales. Existen tres tipos de Weightless: *Weightless-W* el cual usa las frecuencias sin licencia entre las bandas de estaciones de TV, *Weightless-N* el cual hace uso de un protocolo de banda

estrecha sin licencia y *Weightless-P* emplea el rango de banda estrecha de 12,5 KHz y ofrece comunicaciones bidireccionales (McClelland, 2016).

Narrow Band Internet of Things (NB-IoT) que quiere decir “banda estrecha para el internet de las cosas”. NB-IoT ofrece una velocidad máxima de descarga de 250 Kbps y una velocidad de subida de 250 Kbps con multitono o 20 Kbps con monoton. Gracias a que maneja una nueva capa física con sus propias señales, esta tecnología puede coexistir con equipos que operen en redes 2G, 3G y 4G. La batería de uno de estos dispositivos puede llegar a tener una duración de 10 años.

LTE-M es una tecnología de redes de telefonía basadas en 4G. Normalmente, los dispositivos de telefonía no son de baja potencia, y eso se sabe debido a que a un móvil hay que cargarlo de manera frecuente. A fin de compensar el criterio de LPWAN se han diseñado chips inalámbricos 4G con un modo de ahorro de energía especial, el cual permanece apagado la mayor parte del tiempo y solo se enciende en tiempos predeterminados. Además, son semidúplex, de tal forma que usan menos energía cuando están encendidos. Estos chips son mucho más lentos que una conexión 4G tradicional ya que para las redes LPWAN la velocidad máxima de datos es de 100 Kbps, pero está limitada a áreas que tienen cobertura LTE (Paessler, 2018).

LoRa/LoRaWAN es un estándar patentado de tecnología inalámbrica de largo alcance y baja potencia para aplicaciones Máquina a Máquina (M2M) e IoT, que hace uso del espectro de radiofrecuencias sin licenciamiento ISM, específicamente trabaja con frecuencias de 433 MHz, 868 MHz y 915 MHz (Gracia, 2017).

2.6.1. LoRa

Según (Mayobre, 2018) LoRa es un protocolo de capa de enlace y es perfecto para usarse en comunicaciones P2P. Es un protocolo de red que está diseñado para optimizar dispositivos alimentados por baterías, que pueden estar posicionados en lugares fijos o pueden ser móviles. Permite variar dinámicamente el *bit rate* de los equipos que son parte de la red, desde valores de 0,3 Kbps hasta 50 Kbps, dependiendo del chip utilizado y parámetros de modulación.

2.6.2. LoRaWAN

Según (Latre, Weyn, & Lannoo, 2014), LoRaWAN también incluye la capa de red, por lo que es posible enviar la información a cualquier estación base ya conectada a una plataforma en la nube. Los módulos LoRaWAN pueden funcionar en diferentes frecuencias simplemente conectando la antena correcta. Las frecuencias en las que trabaja este tipo de tecnología de comunicación inalámbrica son: 433MHz, 868MHz y 915Mhz.

Por otra parte, el protocolo y arquitectura LoRaWAN, determinan la vida útil de la batería de un nodo, la capacidad de red, la calidad de servicio y la diversidad de aplicaciones de la red; además admiten comunicaciones bidireccionales de bajo costo, móviles y seguras para Internet de las cosas (IoT), diseñado para garantizar la fiabilidad y seguridad (Pérez, 2017).

2.6.3. Arquitectura

Según (Mayobre, 2018) la arquitectura básica de una solución que hace uso de una red LoRaWAN se implementa en una topología estrella, en la cual se hace uso de 3 tipos de equipos: nodos, gateways y servidor de aplicación. Es posible implementar redes tipo *mesh o mall*, en la

cual los nodos retransmiten mensajes de otros nodos, pero para que esto sea posible los nodos siempre deberían estar recibiendo y transmitiendo mensajes, lo que significaría tener un consumo alto de energía en cada nodo, por lo cual, no es recomendable hacer uso de este tipo de topología. En la figura 16 se puede observar la estructura de la arquitectura de una red basada en tecnología Lora, en la cual se pueden observar dispositivos finales (End devices) los cuales son los nodos encargados de recolectar y enviar información hacia los LoRa gateway, estos tienen la función de recibir la información enviada por los nodos para posteriormente enviarla a almacenar en una base de datos sea esta de un local host o en la nube mediante conexión IP.

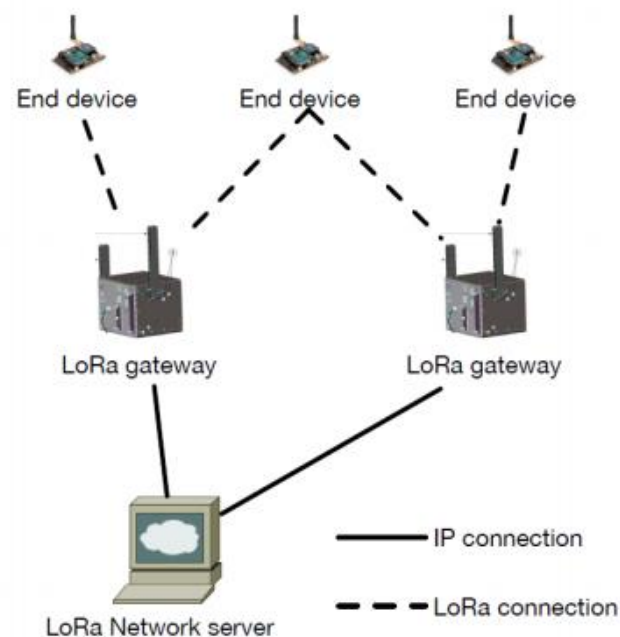


Figura 16. Esquema de comunicación tecnología LoRa
Fuente. (Mayobre, 2018)

2.6.4. Modulación y espectro

LoRa utiliza modulación basada en *Chirp Spread Spectrum* (CSS) debido a su confiabilidad en la transmisión ya que es muy robusta contra disturbios, el desvanecimiento de la

señal por trayectos múltiples, también utiliza el ancho de banda dado de forma más efectiva además de que tiene un bajo consumo de energía al momento de transmitir información ya que reduce significativamente la potencia de transmisión requerida y nos permite enviar información a distancias más largas. Utiliza las bandas ISM de 433MHz, 868MHz y 915MHz según la versión del chip utilizado. Es posible enviar entre 2 y 255 octetos en cada transmisión y su velocidad puede llegar hasta los 50Kbps (Mayobre, 2018).

En las bandas de 433 MHz y 868Mhz se puede hacer uso de 8 canales con un ancho de banda de 0.3MHz por canal. Para la banda de 915MHz se puede hacer uso de 13 canales con un ancho de banda de 2,16MHz por canal (Pérez García, 2017).

2.6.5. Parámetros de interés en LoRaWAN

La modulación LoRa permite cambiar tres parámetros, que son fundamentales en la implementación de una red LoRa, los cuales definen el desempeño del enlace de comunicación, estos son: ancho de banda (BW, bandwidth), factor de dispersión (SF, Spreading factor) y Coding Rate (CR).

El *Ancho de banda* es la longitud de la extensión de frecuencias en la que se encuentra la mayor potencia de señal. Si se requiere una transmisión rápida, la opción de 500MHz es la mejor, pero si se necesita transmitir la información a distancias más largas el módulo debe ser configurado con el valor de 125kHz. Mientras menor sea el ancho de banda más tiempo en el aire permanece la transmisión, pero a su vez la sensibilidad aumenta, por lo que la comunicación tiene un mayor alcance (Gonzáles, 2018).

El *Coding Rate (CR)* puede tomar unos de estos cuatro valores: 4/5, 4/6, 4/7 y 4/8. El primer bit indica cuantos bits son de información (payload) y el segundo los bits enviados. Cuanto menor CR, mayor es el tiempo en el aire de la transmisión, esto hace que la tarea de recepción sea más rápida ya que el receptor puede demodular los paquetes con una menor potencia de recepción, pero también se tarda más tiempo en terminar la transmisión del paquete, esto quiere decir que el consumo de energía aumenta.

El *Factor de dispersión (SF)* es el número de *chirps (pulso de frecuencia modulada)* por símbolo utilizados en el tratamiento de datos antes de la transmisión de la señal. Cada bit de información (payload) es representado por múltiples *chirps* de información. Su valor es un entero entre 7 y 12. Este factor incide en el *bit rate* efectivo y la inmunidad a la interferencia por ruido, es decir, cuanto mayor sea su valor (12), mayor capacidad tiene el receptor para separar el ruido de la señal recibida (González, 2018).

Estos valores también afectan a la *sensibilidad* del receptor, se debe recordar que la sensibilidad se define como la potencia mínima que una señal transmitida debe tener para poder ser captada por el receptor para que pueda ser decodificada (Mayobre, 2018). La modificación de los parámetros anteriormente mencionados influye en la sensibilidad del decodificador, además del tiempo en el aire de la transmisión. Al aumentar el *Ancho de Banda* se disminuye la sensibilidad del receptor, al aumentar el valor del *factor de dispersión* se logra un aumento en la sensibilidad y al disminuir el valor del *Coding Rate* se reduce la ratio de paquetes con error en presencia de pequeñas ráfagas de interferencia, pero el bit rate efectivo disminuye.

2.6.6. Estructura del mensaje LoRa

En la Figura 17 se indica la estructura de un mensaje LoRa:

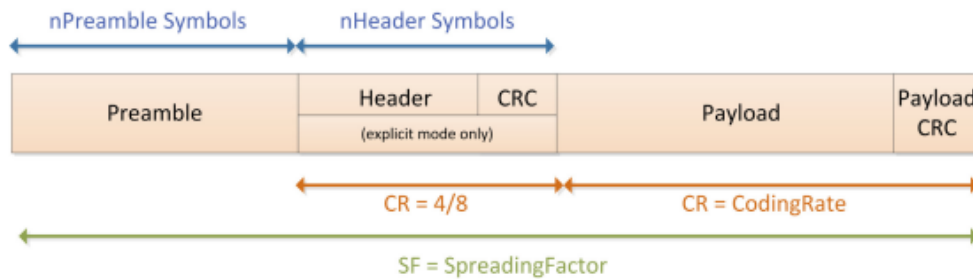


Figura 17. Estructura del Paquete LoRa
Fuente. (Semtech, 2019)

El **preamble o preámbulo** es una serie de bits que se utiliza para sincronizar el receptor con el flujo de datos entrantes. Por defecto, el paquete está configurado con una secuencia larga de 12 símbolos. El último byte informa sobre la red a la cual está conectado el dispositivo permitiendo a los gateways diferenciar su propia red y otras que estén emitiendo en la misma frecuencia (Sáenz, 2018).

El **header o encabezado** tiene la función de añadir más información sobre la trama y el control de errores. Existen dos tipos de encabezados posibles, según el modo que se utilice: *Modo Implícito* (no hay encabezado) el receptor debe conocer de antemano el tamaño del *payload* y del CR (Code Rate) utilizado y *Modo Explícito* este es el modo por defecto, en el campo *header* del paquete se envía información sobre el tamaño de la carga, CR (Code Rate) y CRC (Cyclic Redundancy Check – Verificación de redundancia cíclica) (Mayobre, 2018).

El **payload o carga útil** es el elemento donde se encuentra los datos proporcionados por los sensores que se pretenden enviar hacia el servidor. De acuerdo con el estándar LoRaWAN 1.1 la

longitud máxima de esta parte de la trama depende del tipo de configuración SF que se utilice que puede ser de 2 a 255 bytes.

El *CRC Payload* es un campo de *Forward Error Correction Code* que utiliza la corrección de errores CRC de 16 bits para el payload.

2.6.7. Clases de nodos

Hay tres clases de dispositivos de nodo final, los cuales permiten la comunicación bidireccional y puede iniciar una subida de información a los servidores a través de una puerta de enlace (Pickering, 2017), estos son: Dispositivos Clase A, Clase B y Clase C.

Los dispositivos LoRa *Clase A* una vez que realizan una transmisión en el canal de subida abren dos ventanas para la escucha del canal de bajada por un corto período de tiempo, los dispositivos LoRa *Clase B* se añaden funcionalidades en el momento en el que el dispositivo envía información por el canal de subida, como la sincronización con los concentradores, lo que permite a los servidores de red saber cuándo el dispositivo espera una respuesta por parte de ellos y los dispositivos LoRa *Clase C* el cual a diferencia de las dos clases anteriores tiene la ventana para la recepción de información desde el canal de bajada siempre abierto, que se cierra únicamente cuando el dispositivo tiene información para mandar al concentrador.

2.6.8. Seguridad y Modos de Activación

Los dispositivos LoRaWAN manejan dos posibles maneras de registrarse en la red. Por lo cual, cada uno de los nodos tiene un identificador único dentro de la red denominado *DevAddr* y además cada aplicación puede tener una clave de seguridad extremo a extremo llamada *AppSkey*,

lo cual permite que cada nodo que pertenezca a la red envíe datos a un determinado sistema. Adicionalmente, para hacer a la red aún más segura existe también una segunda clave a nivel de red la cual nos permite el acceso del dispositivo a la red denominada como *NwkSkey* (Mayobre, 2018).

Ahora, para que un módulo pueda ser parte de la red LoRa y sea capaz de enviar mensajes hasta el servidor. Existen dos modos que normalizan el acceso: Activación por personalización (en inglés, Activation by Personalization, abreviado, ABP) y Activación por aire (Over The Air Activation, en inglés, OTAA, abreviado). En primer lugar, está el modo ABP, el cual permite la configuración manual del *DevAddr* y las claves de seguridad del nodo. Cuando se hace la configuración del nodo se ingresan estos datos de forma manual. En segundo lugar, está el modo OTAA, es considerada la forma más segura para que un nodo se una a la red. En este método el *DevAddr* y las claves de seguridad son negociadas con el dispositivo final de manera similar a la negociación que se lleva a cabo mediante DHCP en una red IP. Esta negociación de claves se lleva a cabo entre el nodo y el servidor de red (Mayobre, 2018).

2.6.9. Conexiones

Los módulos LoRa soportan transmisiones Unicast y Broadcast. Las comunicaciones Unicast se hacen uso cuando se quiere enviar un paquete a un nodo específico, y siempre se ejecutan mecanismos de control de tiempo (time-out) para evitar una espera perpetua de un paquete. Las comunicaciones broadcast se utilizan para enviar un paquete a todos los nodos de la red. Cualquier módulo que se encuentre dentro del rango de cobertura aceptará el paquete que contiene la dirección de broadcast, que será la 0. En este modo no hay posibilidad de confirmaciones de ACK ni reintentos (González, 2018).

Para la creación de una red LoRa, es necesario únicamente fijar dos parámetros en LoRa: el canal y modo. En caso de que la red ya esté previamente creada, el nuevo dispositivo final se debe configurar con el mismo canal y modo de la red a la que quiere pertenecer.

2.7. Sistemas Embebidos

Se definen como dispositivos electrónicos o computacionales que procesan información digital o analógica con propósitos de realizar una o varias funciones específicas en tiempo real. Estos sistemas son muy utilizados debido a que funcionan con un bajo consumo de energía además de que tienen un tamaño reducido, tienen también un bajo costo y resultan ser de alta disponibilidad y confiabilidad.

2.7.1. Componentes de un sistema embebido

Por lo general un sistema embebido está conformado por algunas partes tales como: Memoria Principal, Procesador, Memoria Flash, puertos de Comunicación Serial, Controladores USB, Generador de reloj, Puerto para alimentación de corriente, Puerto Ethernet, WiFi, pines de Input/Output; donde algunas de ellas se pueden visualizar en la figura 18.

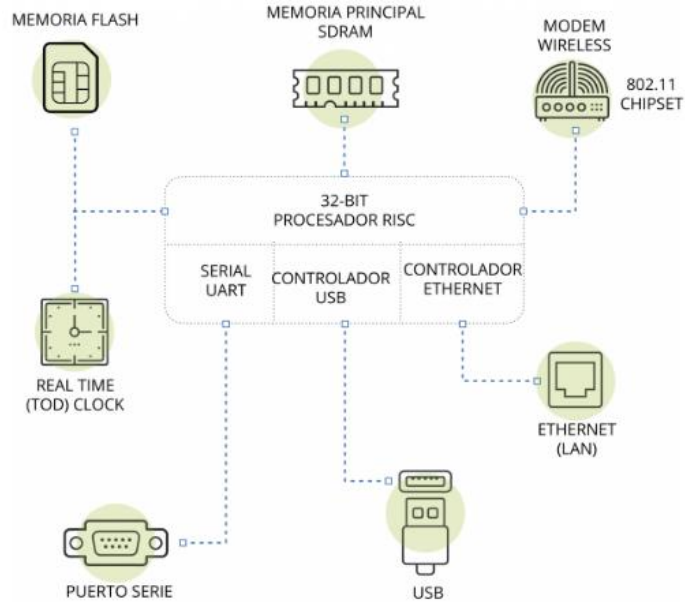


Figura 18. Componentes de un sistema embebido (Nivel Físico)
Fuente. (INCIBE, 2018)

El principal componente de un sistema embebido es su microcontrolador, el cual su función principal es la de procesar la información generada por sus varios pines los cuales pueden tener conectados sensores que miden variables del entorno; la velocidad de procesamiento varía dependiendo el fabricante de este. Gracias a estos componentes es posible crear aplicaciones que nos permitan controlar diferentes procesos de manera eficiente, todos estos componentes están ubicados sobre una tarjeta PCB que puede cumplir con funciones parecidas a las de un minicomputador.

3. Capítulo III. Desarrollo Experimental

Este capítulo abarca temas sobre lo que implica el diseño y construcción del sistema de riego, la integración de las redes neuronales artificiales al control, la evaluación de requerimientos que ayuden a la elección de los componentes de Hardware y Software para la puesta en marcha de un prototipo correctamente funcional.

3.1. Metodología

Para el desarrollo de este proyecto es necesario el uso de metodologías de diseño de sistemas que incluyan el prototipado, por esta razón se escoge el modelo en V, el cual guía la integración del sistema permitiendo la retroalimentación durante el proceso de desarrollo. Adicionalmente, se utiliza el estándar IEEE-29148 como complemento al modelo en V para hacer la selección de componentes tanto de hardware como de software involucrados en el presente trabajo de titulación.

3.2. Modelo en V

El Modelo en V, permite llevar a cabo un proceso de control adecuado para el desarrollo de proyectos, debido a que este modelo relaciona las pruebas de validación con las etapas de análisis, diseño e implementación del proyecto. De esta manera, se puede identificar en qué fase se debe iniciar si falla alguna prueba, lo que garantiza altas posibilidades de éxito y ayuda a minimizar los riesgos de fallas en la fase final del proyecto (Perez, Berreteaga, Ruiz de Olano, Urkidi, & Perez, 2006).

En la figura 19 se muestran las dos áreas y las etapas en la que se encuentra dividido el modelo en V; el área de la izquierda contiene el análisis, los requerimientos y el diseño del sistema hasta llegar a la implementación; mientras que en el área derecha se encuentran las validaciones del proyecto diseñadas específicamente para cada fase.



Figura 19. Modelo en V para el desarrollo de proyectos
Fuente. Adaptado de (Perez, Berreteaga, Ruiz de Olano, Urkidi, & Perez, 2006)

En el primer nivel de la figura 19 se deben concretar y documentar los requisitos y especificaciones que debe tener el presente proyecto, recolectando datos e información que permitan conocer los factores que influyan en la implementación de este.

Así mismo, en el segundo nivel se determina cuáles son los requerimientos del sistema; este nivel comprende una de las fases más importantes para que el proyecto tenga éxito, debido a que definen la funcionalidad del sistema propuesto. En la fase de integración del sistema se realizan pruebas para poder verificar la funcionalidad de cada módulo que forma parte del proyecto.

El tercer nivel define todos los estados que se desarrollará en cada parte del sistema con la finalidad de obtener un diseño y visión general del proyecto. En la fase de prueba unitaria se ejecuta la verificación independiente de cada parte del proyecto, comprobando la correcta implementación, coherencia y cumplimiento de las especificaciones respectivas.

Finalmente, en el último nivel se encuentra la fase de la implementación con la cual se materializa el diseño del proyecto y se procede a la programación para cada parte del sistema del proyecto que se va a realizar.

3.3. Etapa de Análisis

Para determinar los factores que intervienen en el desarrollo del sistema de riego propuesto, se necesita realizar un análisis del entorno en el cual se va a implementar el presente proyecto. En este caso, tiene que ver con el aspecto espacial en cuanto a las dimensiones de implementación en el terreno y de los estudios que se hacen sobre el cultivo, además se obtiene la información necesaria para buscar lugares estratégicos y determinar la ubicación y modelo del hardware a emplearse, para todo esto es necesario una visita al sitio de implementación del sistema de riego.

3.3.1. Situación actual

La presente investigación se realiza en las instalaciones de la Granja “La Pradera” de la Universidad Técnica del Norte, en la parroquia San José de Chaltura perteneciente al cantón Antonio Ante; específicamente se lleva a cabo en la zona destinada para cultivos de alfalfa, de la cual se toma un área de 45m² como muestra para el desarrollo de esta investigación.

Mediante una visita in situ al cultivo de alfalfa donde se va a implementar el sistema de riego propuesto, se evidencia que el mecanismo de riego utilizado actualmente es manual, utilizándose el riego por inundación el cual no es muy eficiente, ya que se desperdicia demasiado recurso hídrico.

Adicionalmente se recolecta datos relevantes de la zona del sembrío en la cual se va a realizar la implementación del sistema de riego, como son: mediciones del área del terreno de la zona de estudio, medición de la distancia entre el terreno y el sitio donde se encuentra la fuente de agua que servirá para proveer de agua de riego.

La Figura 20 muestra los sitios a interconectarse, en donde el área de cultivo se denomina zona 1 y al sitio donde se ubica el centro de mando, zona 2: los cuales está separados una distancia de 400 metros.

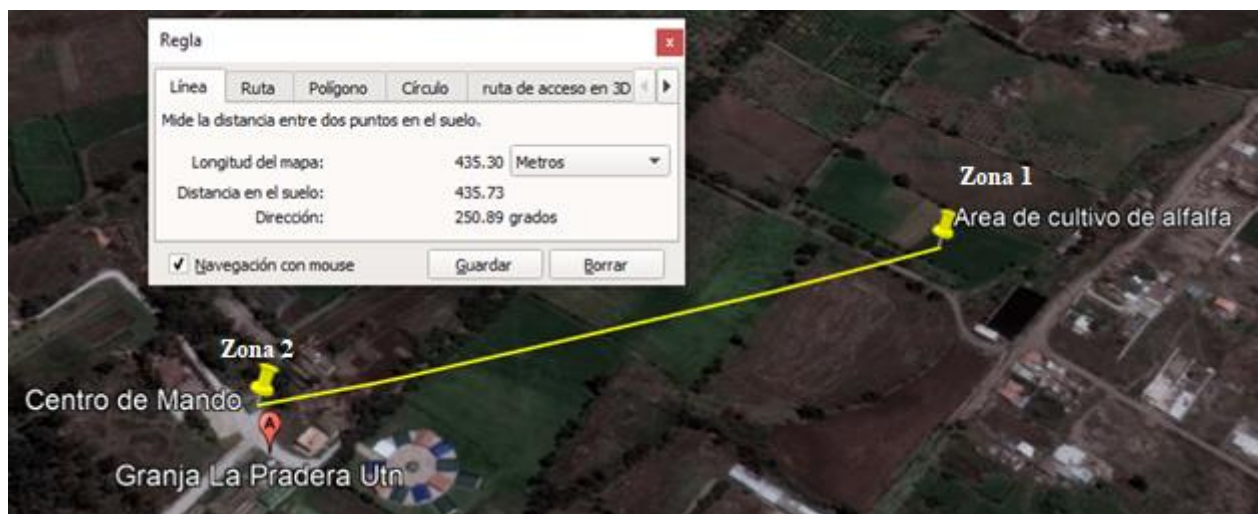


Figura 20. Vista aérea del área de cultivo y lugar del nodo central
Fuente. Autoría

3.4. Descripción General del Sistema de Riego Propuesto

El proyecto a desarrollarse propone dos métodos de riego. El primer método consiste en que el operador determine cuando es necesario realizar el riego en base a las variables meteorológicas presentes en el área de cultivo tales como: temperatura ambiental , humedad del ambiente, luminosidad y humedad del suelo; estos serán recolectados por los sensores correspondientes y servirán para posteriormente implementar el sistema de riego aplicando el algoritmo de inteligencia artificial propuesto para este proyecto, es decir, los datos recogidos servirán para el entrenamiento de este algoritmo. El segundo método es aplicando el algoritmo de *Redes Neuronales Artificiales* el cual, en base a los datos recolectados en el primer método se determinará cuando es necesario iniciar el proceso de riego y cuando apagar este sistema.

Por otra parte, se debe considerar el paso y bloqueo del flujo de agua, para lo cual se propone una electroválvula solenoide, ya que este tipo de dispositivos electrónicos permite controlarlos al aplicarles señales eléctricas. Los datos de ingreso para el sistema de riego basado en la red neuronal propuesto, son tomados de la red de sensores inalámbricos (WSN) desarrollado en (Domínguez, 2020). El sistema está comprendido por dos zonas, la primera zona comprende los sensores que recogen los datos necesarios de las variables meteorológicas y en esta misma zona se hará la instalación de la electroválvula que será controlada por la estación de control del sistema de riego.

En cuanto a la técnica de riego a utilizarse en base al análisis hecho por el técnico agricultor de la granja, el Ing. Agrónomo Gabriel Chimbo, se opta por un riego por goteo que, según su opinión, es una de las técnicas de riego óptimas debido a que es menos costosa y es más frecuentemente usada.

Por otro lado, el área de trabajo se la divide en dos zonas con el propósito de mejorar la gestión de desarrollo de este proyecto, por tanto, la división de zonas queda de la siguiente manera: la zona 1 pertenece al área de cultivos de alfalfa y la zona 2 comprende el área de gestión del riego. En la Figura 21 se muestra una visualización general del sistema de riego inteligente con redes neuronales artificiales distribuido en zonas. Es trascendental que la comunicación entre la zona de cultivo y la zona donde se procesa el control del riego se confiable, ya que es posible que se genere pérdidas de transmisión entre estas dos zonas, lo que reduce la operatividad de esta técnica de riego que se quiere implementar.

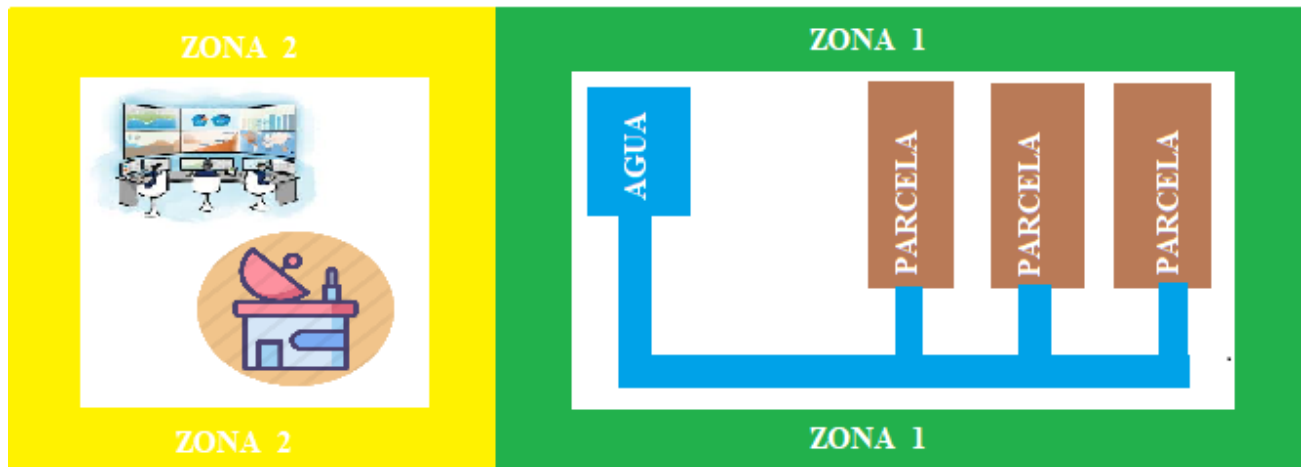


Figura 21. Distribución Zonal
Fuente: Autoría

3.5. Determinación de Stakeholders

Un stakeholder es todo un grupo o individuo que tiene un interés directo en el resultado obtenido por el desarrollo del proyecto. La definición de los requerimientos del stakeholder (StSR) tiene como finalidad identificar los requisitos de los interesados para el sistema. En la Tabla 1 se listan los implicados o stakeholders que se toman en cuenta para el desarrollo de este proyecto.

Tabla 1. Lista de stakeholders del sistema.

Lista de stakeholders	
1	Granja “La Pradera”
2	Docentes expertos en el área agrícola y estudiantes
3	Ing. Edgar Maya - Director del Presente Trabajo de Titulación
4	Darío Cadena – Autor de la investigación

Fuente. Autoría

3.6. Requerimientos de diseño

Se definen los requerimientos de stakeholders, requerimientos del sistema y requerimientos de arquitectura, para al final tener un prototipo funcional con la finalidad de satisfacer las necesidades del usuario, el alcance del sistema y de esa manera lograr que la presente investigación beneficie a los involucrados en el cultivo de alfalfa.

3.6.1. Nomenclatura de requerimientos

En la Tabla 2 se muestran los acrónimos que se emplean para referirse de forma abreviada a cada uno de los requerimientos analizados.

Tabla 2. Definición de Acrónimos y Abreviaturas

<i>Acrónimo</i>	<i>Descripción</i>
StSR	Requerimientos de Stakeholders
SySR	Requerimientos Funcionales del Sistema
RSRH	Requerimientos de Arquitectura

Fuente: (Barreto, 2014)

3.6.2. Requerimientos de Stakeholders

Con el propósito de definir los requisitos de los interesados en el desarrollo del sistema, se analiza un conjunto común de los requerimientos que expresan la interacción que el sistema tendrá con su entorno y con los usuarios involucrados en el presente proyecto. La tabla 3 contiene los requisitos operacionales y de usuario considerados en los Requerimientos de Stakeholders (StSR).

Tabla 3. Requerimientos de Stakeholders

StSR					
REQUERIMIENTOS DE STAKEHOLDERS					
#	REQUERIMIENTOS DE USO	Prioridad			Relación
		Alta	Media	Baja	
REQUERIMIENTOS OPERACIONALES					
StSR1	Conectividad a Inalámbrica		X		
StSR2	El sistema de presentar bajo consumo de energía	X			
StSR3	Adquisición de datos	X			
StSR4	Visualización de datos		X		
StSR5	Automatización del Sistema	X			
StSR6	Operación permanente	X			
StSR7	Almacenamiento en base de datos		X		
REQUERIMIENTOS DE USUARIOS					
StSR8	El nodo de control de automatización del sistema de riego debe estar totalmente impermeabilizado debido a que va a estar ubicado en el área de cultivo	X			
StSR9	Conocimientos en sistemas de riego		X		
StSR10	Facilidad de uso para un usuario común	X			
StSR11	Dispositivo tolerante a agentes externos como agua, calor y humedad.	X			

Fuente. Autoría

3.6.3. Requerimientos funcionales del sistema

Hace referencia a una lista de requerimientos con respecto a cada una de las funciones, comportamiento y forma de uso del sistema. En la Tabla 4 se detalla los requerimientos funcionales que permiten diseñar el prototipo.

Tabla 4. Requerimientos Funcionales del Sistema

SySR					
REQUERIMIENTOS INICIALES DEL SISTEMA					
#	REQUERIMIENTOS	Prioridad			Relación
		Alta	Media	Baja	
REQUERIMIENTOS DE INTERFAZ					
SySR1	Pines de entrada y salida de información del sistema embebido	X			
SySR2	Comunicación a través del puerto serial del sistema embebido con una placa para disminuir el procesamiento	X			
SySR3	El sistema electrónico (Sistema embebido, válvula solenoide, placas de comunicación inalámbrica y la alimentación eléctrica) deberán interactuar entre sí.	X			
REQUERIMIENTOS DE PERFORMANCE					
SySR4	Tener acceso constante a una BD que contengan datos actualizados de los parámetros medidos del cultivo (Humedad relativa, Humedad del suelo, Temperatura Ambiental)	X			
SySR5	Menor consumo de energía	X			
SySR6	Menor retardos de tiempos en la transmisión de datos entre zonas	X			
SySR7	Menor error en los datos adquiridos por los sensores	X			
SySR8	Establecer el estado del suelo (humedad suficiente o humedad insuficiente) con el algoritmo de aprendizaje automático	X			
SySR9	Iniciar con el proceso de riego/Detener el proceso de riego	X			
REQUERIMIENTOS DE MODOS/ESTADOS					
SySR10	El riego deber permanecer encendido hasta que se llegue a un valor adecuado de humedad del suelo esto va a ser determinado por el algoritmo de aprendizaje automático.	X			
SySR11	Estado de funcionamiento y carga con batería independiente				X
REQUERIMIENTOS FÍSICOS					
SySR12	Resistente al calor, agua y humedad	X			
SySR13	Tamaño reducido				X

Fuente: Autoría

3.6.4. Requerimientos de arquitectura

Se refiere a los componentes que conforman el sistema, tanto software como hardware, el análisis correcto de estos requerimientos permite hacer una correcta elección de materiales necesarios, los cuales se listan en la Tabla 5.

Tabla 5. Requerimientos de Arquitectura

SRSH					
REQUERIMIENTOS DE ARQUITECTURA					
#	REQUERIMIENTOS	Prioridad			Relación
		Alta	Media	Baja	
REQUERIMIENTOS HARDWARE					
SRSH1	Estabilidad de comunicación inalámbrica	X			
SRSH2	El sistema embebido requiere de un procesador que soporte el tratamiento de información en tiempo real.	X			
SRSH3	Sistema embebido con entradas y salidas análogas y/o digitales para el manejo de la válvula solenoide.	X			
SRSH4	Se requiere un bajo consumo de energía	X			
SRSH5	Sistema embebido con puertos de transmisión (Tx) y recepción (Rx) para realizar el envío de información entre módulos inalámbricos.	X			
SRSH6	Cobertura inalámbrica mayor a 520m	X			
REQUERIMIENTOS DE SOFTWARE					
SRSH7	Lenguaje de programación de código abierto	X			
SRSH8	Compatibilidad del software con la librería para el algoritmo de aprendizaje automático (RNA)	X			
SRSH9	Se requiere un software que permita la ejecución del código de Redes Neuronales Artificiales desde un sistema embebido en tiempo real.		X		
SRSH10	Compatibilidad de librerías con el microcontrolador	X			
SRSH11	Compatibilidad con plataformas de programación (Python)	X			
SRSH12	Control de motores eléctricos	X			
SRSH13	Evite sobrecargar el procesador	X			
SRSH14	Compatibilidad con módulos para crear aplicaciones funcionales				
SRSH15	Facilidad de conexión				
SRSH16	Menor carga computacional				
REQUERIMIENTOS ELECTRICOS					
SRSH17	Fuente de alimentación eléctrica	X			

SRS18	Fuente de alimentación externa para recargar baterías.	X
-------	--	---

Fuente. Autoría

3.7. Elección de Software y Hardware para la implementación del sistema

La selección tanto de hardware como de software se la realiza a través de la Tabla 5 la cual contiene los requerimientos de hardware y software para el correcto desarrollo del proyecto, para ello se realiza una tabla comparativa entre los sistemas embebidos que pueden cubrir todos o la mayoría de requerimientos y se procede a asignar una valoración (*1: cumple y 0: no cumple*) a cada uno de los sistemas embebidos dependiendo de las características de cada una de estas placas embebidas y se elegirá a la que obtenga la mayor ponderación.

3.7.1. Elección del Hardware

Esta etapa es la determinación de los componentes generales del sistema; para lograr reducir la complejidad en el diseño del sistema y con el fin de optimizar la distribución de los recursos de hardware, la mejor forma de lograr esto es establecer una división en zonas sobre el área donde se desarrolla la investigación, la cual fue descrita en la sección 3.4. (Descripción General del Sistema Propuesto)

3.7.1.1. Elección de hardware para la Zona 1

El primer componente que se va a seleccionar es el microcontrolador. La Tabla 6 contiene una serie de microcontroladores existentes en el mercado con los lineamientos que cumplen cada uno en relación con los requerimientos establecidos anteriormente.

Tabla 6. Elección del microcontrolador en la Zona 1

HARDWARE	REQUERIMIENTOS					VALORACION
	SRSH3	SRSH5	SRSH7	SySR5	SRSH13	TOTAL
Arduino UNO	1	1	1	1	1	5
Arduino Mega	1	1	1	0	0	3
Raspberry Pi3	1	1	1	0	1	4

1 - Cumple
0 - No cumple

De la Tabla 6 se deriva el siguiente resultado: el componente que cumple con el mayor número de requerimientos que necesita el sistema es un microcontrolador Arduino UNO, ya que es el dispositivo que más se ajusta a las necesidades del proyecto por su tamaño reducido (68.5mm x 5.8mm) y bajo consumo energético (46mA), además de que su capacidad de procesamiento de información es suficiente para esta zona, en comparación con el Arduino Mega que su tamaño es mayor (101.6mm x 53.34mm) y además tiene un mayor consumo de energía (93mA) al igual que la raspberry Pi3 con un tamaño de 56mm x 85mm y un consumo energético de 350mA. Las características del microcontrolador seleccionado se las puede observar en el apartado 3.8.2.

Entre las dos zonas se realiza el envío y recepción de datos de manera constante, a esto se suma su ubicación geográfica, por lo que es indispensable un medio inalámbrico para permitir la comunicación entre zonas. La tabla 7 presenta algunas de las tecnologías inalámbricas que se relacionan con los requerimientos del sistema.

Tabla 7. Elección de la tecnología inalámbrica

HARDWARE	REQUERIMIENTOS				VALORACION
	SRSH1	SRSH4	SRSH6	SRSH11	TOTAL
Bluetooth	0	1	0	1	2
WIFI 802.11	1	1	0	1	2
ZigBee - IEEE 802.15.4	1	1	0	1	3
LoRaWAN	1	1	1	1	4

1 - Cumple
0 - No cumple

Fuente. Autoría

De la tabla 7 se deriva el siguiente resultado: La tecnología que cumple con el mayor número de requerimientos que necesita el sistema de control de riego basado en redes neuronales artificiales es LoRaWAN, por lo tanto, mediante el uso de esta tecnología se implementa el sistema de comunicación inalámbrica entre la zona 1 y la zona 2, debido a que permite una mayor cobertura en comparación con las otras tecnologías listadas en la Tabla 7. Uno de los dispositivos que trabaja bajo esta tecnología de comunicación inalámbrica es el chip RN2483, el cual además tiene un tamaño reducido, tiene un bajo consumo de energía lo que ayuda a extender la vida la batería y está disponible en el mercado nacional. Las características de este chip se pueden observar en el apartado 3.8.3.

Por otra parte, para permitir o bloquear el paso de flujo de agua en el sistema de riego es indispensable el uso de un dispositivo que cumpla con esta función para lo cual se hace uso de electroválvulas solenoides. En la tabla 10 se presentan algunas de estas válvulas que se encuentran disponibles en el mercado.

Tabla 8. Elección de la electroválvula solenoide

HARDWARE	REQUERIMIENTOS		VALORACIÓN TOTAL
	SRS4	SySR13	
ZE-4F180	0	0	0
Solenoide 12V	1	1	2
ROUND-START	0	0	0

1 - Cumple
0 - No cumple

Fuente. Autoría

De la tabla 10 se deriva el siguiente resultado: El elemento que cumple con el mayor número de requerimientos para la implementación de este sistema es la válvula solenoide 12V, por lo tanto, se hace la selección de este dispositivo, debido a que tiene menor consumo energético (350mA), es de menor tamaño (83.8mm x 42.9mm x 56.8mm), se ajusta a las necesidades del sistema y también tiene un costo relativamente bajo en comparación con las otras dos electroválvulas. Las características de este dispositivo se listan en el apartado 3.8.4.

3.7.1.2. Elección de hardware para la Zona 2

La zona 2 está conformada por la estación de trabajo, la cual es considerada como la zona de control de riego. Esta zona está compuesta por dos componentes: un módulo inalámbrico LoRaWAN RN-2483 (Receptor) que ya fue seleccionado previamente en la descripción de la zona 1 y el sistema embebido principal que acompaña a este módulo de comunicación el cual es el núcleo del sistema, donde estará alojada la Red Neuronal Artificial encargada de la automatización del riego. La tabla 9 contiene los diferentes modelos de sistemas embebidos disponibles en el mercado que pueden ser usados en este proyecto.

Tabla 9. Elección del Sistema Embebido de la Zona 2

HARDWARE	REQUERIMIENTOS					VALORACION TOTAL
	SRSH2	SRSH5	SRSH9	SRSH11	SRSH16	
Raspberry Pi 3	1	1	1	1	1	5
Intel Galileo	1	1	1	1	1	5
Beagle Bone	1	1	1	1	1	5

1 - Cumple
0 - No cumple

Fuente. Autoría

De la tabla 9 se deriva el siguiente resultado: Todos los dispositivos listados cumplen con los requerimientos del sistema, tales como procesamiento de información en tiempo real, puertos de transmisión y recepción, compatibilidad con la plataforma de Python; pero se selecciona el sistema embebido Raspberry Pi3 por su bajo costo y disponibilidad en el mercado nacional. Las características del hardware seleccionado se detallan en el apartado 3.8.1.

3.7.2. Elección del Software

El software a escoger debe ofrecer una relación con los aspectos definidos en la etapa de requerimientos, de esta forma se garantiza un desempeño sólido. Al igual que el hardware, la selección del software se la realiza para las dos zonas, un software que cumpla las características para la zona 1 y otro para la zona 2.

3.7.2.1. Elección del software para la zona 1

Los parámetros que influyen en la selección del software destinado en esta zona deben cumplir con los requerimientos que se indican en la tabla 5. A continuación, en la Tabla 10 se enlistan distintos softwares de programación.

Tabla 10. Elección del Software de la Zona 1

SOFTWARE	REQUERIMIENTOS				VALORACION
	SRSH7	SRSH10	SRSH12	SRSH13	TOTAL
IDE Arduino	1	1	1	1	4
Python	1	1	0	1	3
Java	1	1	0	1	3

1 - Cumple
0 - No cumple

Fuente. Autoría

De la tabla 10 se deriva el siguiente resultado: El software que cumple con la mayoría de los requerimientos para el desarrollo del sistema es el IDE de Arduino, por tal razón se decide la selección de este, tomando en cuenta también que la placa de procesamiento seleccionada en el apartado 3.7.1.1 es un Arduino UNO el cual trabaja bajo esta plataforma.

3.7.2.2. Elección del Software para la zona 2

Esta sección la conforma un sistema embebido principal, el cual está encaminado en brindar una capacidad de procesamiento más elevada que en la zona 1, debido a la cantidad de datos que debe procesar entorno al objetivo de control, por otro lado, es quien se encarga de ejecutar las ordenes que conlleva el sistema. Además, va a ser el software el cual va a ser utilizado para realizar la implementación de la red neuronal artificial que será utilizada para la automatización del sistema de riego. En la Tabla 11 se hace la elección del software más idóneo para obtener un sistema funcional y totalmente operativo.

Tabla 11. Elección del Software de la Zona 2 (Gateway y RNA)

SOFTWARE	REQUERIMIENTOS				VALORACION
	SRSH7	SRSH12	SRSH13	SRSH14	TOTAL
Python	1	1	1	1	4
Matlab	1	0	0	0	1
Visual Basic	1	0	0	0	1

1 - Cumple
0 - No cumple

Fuente. Autoría

De la tabla 11 se deriva el siguiente resultado: Python es el software que cumple con los requerimientos expuestos en la tabla anterior por lo cual es el software elegido para el desarrollo del prototipo, además de que Python es el software que más se utiliza para el desarrollo de proyectos basados en inteligencia artificial debido a que tienen muchas librerías que ayudan en el desarrollo de este tipo de proyectos, al ser ejecutado tiene una menor carga computacional en el sistema embebido y también posee licencia libre.

Tabla 12. Elección del Software de la Zona 2 (Almacenamiento de datos)

SOFTWARE	REQUERIMIENTOS			VALORACION
	StSR10	SRSH7	SRSH13	TOTAL
Microsoft Access	1	0	1	2
MySQL	1	1	1	3

1 - Cumple
0 - No cumple

Fuente. Autoría

Mientras que, para el almacenamiento de los datos respecto al criterio de decisión de riego, Tabla 12 presenta la selección entre las bases de datos Microsoft Access y MySQL, de donde se deduce lo siguiente: MySQL cumple con la mayor cantidad de requerimientos expuestos, se selecciona este software para el almacenamiento de los datos, ya que uno de los motivos es su

licencia libre, así como el fácil manejo para el usuario, en comparación con Microsoft Access que requiere compra de licencias.

3.8. Hardware destinado para el sistema

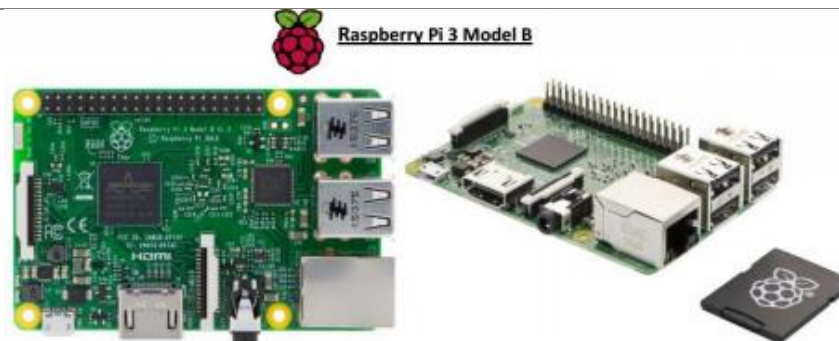
La estructura física de este sistema está conformada por una red de sensores inalámbrica tomada de (Domínguez, 2020) y el nodo de control de la electroválvula que será controlado por la red neuronal implementada en este trabajo de investigación, ubicados en la zona 1 referente a la ubicación del cultivo de alfalfa y la zona 2 o zona de mando lugar en el cual se implementa la RNA para la parte del control del riego. La zona 1 está compuesta por los siguientes elementos: Fuente de alimentación eléctrica, sensores necesarios para la medición de cada una de las variables meteorológicas, módulo relé, electroválvula solenoide, el microcontrolador Arduino UNO y un módulo de comunicación con tecnología LoRaWAN (RN2483). La zona 2 está compuesta por el sistema embebido Raspberry Pi 3, un módulo de comunicación LoRaWAN y su respectiva fuente de eléctrica. A continuación, se hace una breve descripción de cada uno de los componentes mencionados, los cuales fueron seleccionados en el ítem 3.7.1.

3.8.1. Sistema embebido Raspberry Pi 3

Es una placa de tamaño reducido que tiene la capacidad de trabajar como un mini computador, la cual está compuesta por varios pines llamados GPIO con una ranura para insertar una tarjeta MicroSD en la cual se instala previamente un sistema operativo (Raspberry Pi Foundation , 2018).

A continuación, en la Tabla 13 se muestran las características técnicas principales de la placa Raspberry Pi 3, la información a detalle de este sistema embebido se puede acceder en la página oficial de Raspberry:

Tabla 13. Especificaciones Técnicas del sistema embebido Raspberry Pi 3



Chip	Broadcom BCM 2835
CPU	1,2 GHz de 64 bits con 4 núcleos ARMv8
GPU	VideoCore IV
Memoria del sistema	1Gb LPDDR2
Almacenamiento	Tarjeta microSD, ranura para un máximo de 32GB
Salida de video	HDMI, jack 3.5mm, Conector LVDS de a bordo
Audio I/O	HDMI, 3.5mm jack de salida estéreo
Red	Ethernet 10/100Mbps RJ45, Wi-Fi, 802.11n Wireless LAN
USB	4 puertos USB 2.0
Expansión	UART, PWM, GPIO, I2C, SPI, conector CSI cámara, conector monitor LVDS
Consumo	2,5A / 12,5w / 5V
Dimensiones	85 x 56 x 17 mm
Peso	45 g
Sistemas Operativos	CPU Raspbian, Pidora, Arch Linux, Media center Openlec, Raspbmc

Fuente: (Raspberry Pi Foundation , 2018)

3.8.2. Microcontrolador Arduino UNO

Es una placa de desarrollo basada en el IDE de Arduino, contiene pines para comunicación con otros equipos (Rx y Tx), además posee 14 pines para lectura de datos de sensores tanto

analógicos como digitales, para control de motores y otros. Está orientado el desarrollo de proyectos que permitan tener el control en cualquier área que se desee aplicar. La figura 22 muestra un aspecto visual de este dispositivo y en la Tabla 14 sus especificaciones técnicas (Guerrero, 2014).



Figura 22. Placa Arduino UNO
Fuente. (Diaz, 2016)

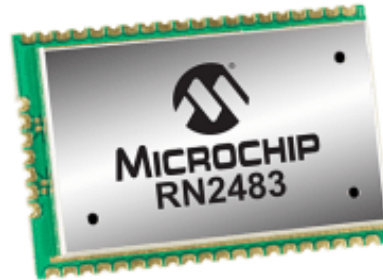
Tabla 14. Especificaciones técnicas del Arduino UNO

Característica	Valor
Microcontrolador	ATmega328P-PU
Voltaje de operación	5V
Voltaje de entrada	7 – 12 V (Recomendado) 6 – 20 V (límite)
Digital I/O pins	14
PWM Digital I/O Pins	6
Entradas analógicas	6
Corriente máxima por pin I/O	40 mA
Corriente máxima por pin 3.3V	50 mA
Memoria Flash	32 KB
SRAM	2 KB
EEPROM	1 KB
Frecuencia del clock	16 MHz
USB	Tipo B
Dimensiones y peso	68.6 x 53.4 mm 25 g

Fuente. (Diaz, 2016)

3.8.3. Módulo LoRaWAN RN-2483

El módulo LoRaWAN RN-2483 es un módulo certificado totalmente, basado en la tecnología inalámbrica LoRaWAN. Este módulo utiliza una modulación de espectro ensanchado única dentro de la banda Sub-GHz para lograr transmitir a largas distancias y a baja potencia con menor consumo de energía. La Figura 23 muestra es aspecto visual de este módulo y la Tabla 15 contiene sus especificaciones técnicas (Microchip Technology Inc, 2016).



*Figura 23. Módulo LoRaWAN RN2483
Fuente. (Microchip Technology Inc, 2016)*

Tabla 15. Especificaciones técnicas módulo LoRaWAN RN-2483

Característica	Valor
Voltaje de alimentación operativo	2.1 a 3.6 V
Consumo de Transmisión	40 mA
Consumo de corriente	14.2 mA
Banda de frecuencia	433 MHz y 868 MHz
Temperatura de trabajo máxima	85 °C
Temperatura de trabajo mínima	-40 °C
Tipo de interfaz de host	UART
Alcance	20 km con línea de vista

Fuente. (Microchip Technology Inc, 2016)

3.8.4. Válvula Solenoide

Este dispositivo está conformado por varios componentes internos, su principal elemento es la bobina eléctrica que está sujeta a su estructura, normalmente la válvula está cerrada, esto

ayuda a que no haya fugas si no está en funcionamiento, solamente cuando se le aplica un voltaje en los terminales se permite la circulación de líquido. La Figura 24 muestra el aspecto visual de este dispositivo y la Tabla 16 contiene sus especificaciones técnicas (Oliveira, 2019).



Figura 24. *Electroválvula Solenoide*
Fuente. (PC-FACTORY, 2015)

Tabla 16. *Especificaciones técnicas de la electroválvula solenoide*

Característica	Valor
Voltaje de operación	6 – 12V (12V nominal)
Diámetro del orificio	1/2" NPS
Tiempo de respuesta	≤ 0,15 s (al abrir) ≤ 0,3 s (al cerrar)
Presión de trabajo	0,2 mpa – 1Mpa
Consumo de corriente	350mA
Temperatura de trabajo	1 °C a 75 °C
Estado inicial	Cerrada
Dimensiones y peso	83.8mm x 42.9mm x 56.8mm 120g

Fuente. (PC-FACTORY, 2015)

3.8.5. Diagramas de puertos y pines del hardware seleccionado

Para el desarrollo del proyecto se emplean diversos dispositivos, por lo que es necesario realizar un diagrama de conexión que muestre los pines que se van a emplear en el sistema; los dispositivos de los cuales se realizarán los diagramas de conexión son: el sistema embebido Raspberry Pi 3 Model B, la placa reducida Arduino UNO y el módulo de comunicación LoRaWAN (RN2483).

3.8.5.1. Diagrama de pines y puertos – Raspberry Pi 3

Del sistema embebido Raspberry Pi 3 se emplean los pines GPIO para conectar el módulo de comunicación LoRaWAN, también se hace uso de los puertos USB para la navegación en la interfaz del sistema embebido como son: mouse y teclado, también se hace uso del puerto ethernet para poder acceder a la base de datos de los sensores que va a estar integrada en otro Sistema Embebido Raspberry Pi 3.

En la Figura 25 se muestra la distribución de pines y puerto del sistema embebido Raspberry Pi 3.

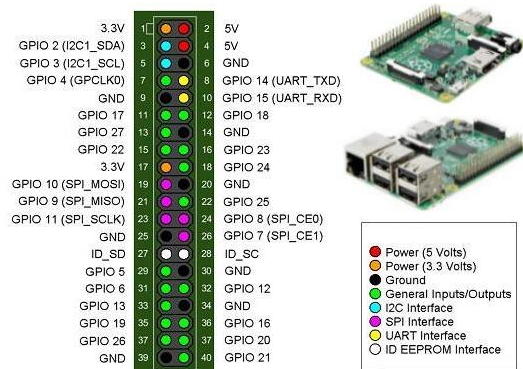
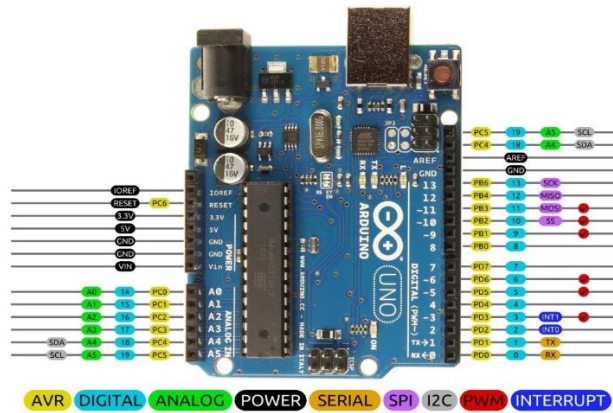


Figura 25. Pines y puertos del Sistema Embebido Raspberry Pi 3
Fuente. (Mónaco, 2014)

3.8.5.2. Diagrama de pines y puertos – Arduino UNO R3

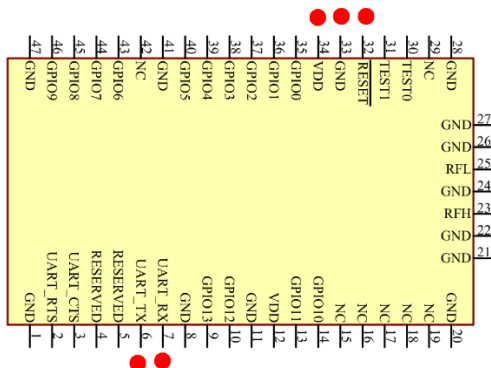
De la placa reducida Arduino UNO R3 se emplean las entradas lógicas para conectar la electroválvula y la placa de comunicación LoRaWAN además de los puertos VCC y GND para la alimentación de esta. A continuación, en la Figura 26 se muestra la distribución de pines de la placa Arduino UNO R3.



*Figura 26. Pines y puertos de la placa reducida Arduino UNO R3
Fuente. (IX23 Explore, 2018)*

3.8.5.3. Diagrama de pines y puertos – Módulo LoRaWAN (RN2483)

De la placa de comunicación LoRaWAN se hace uso de los pines VCC (pin 34) y GND (pin 33) para su alimentación, además se hace uso de los pines de transmisión para el envío de datos (pin 6) y de recepción para recibir datos (pin 7) y el de Reset (pin 32). A continuación, en la Figura 27 se muestra la distribución de los pines del Módulo LoRaWAN (RN2483) descritos anteriormente.



*Figura 27. Pines y puertos del Módulo LoRaWAN (RN2483)
Fuente. (Microchip Technology Inc, 2016)*

3.9. Diseño del sistema

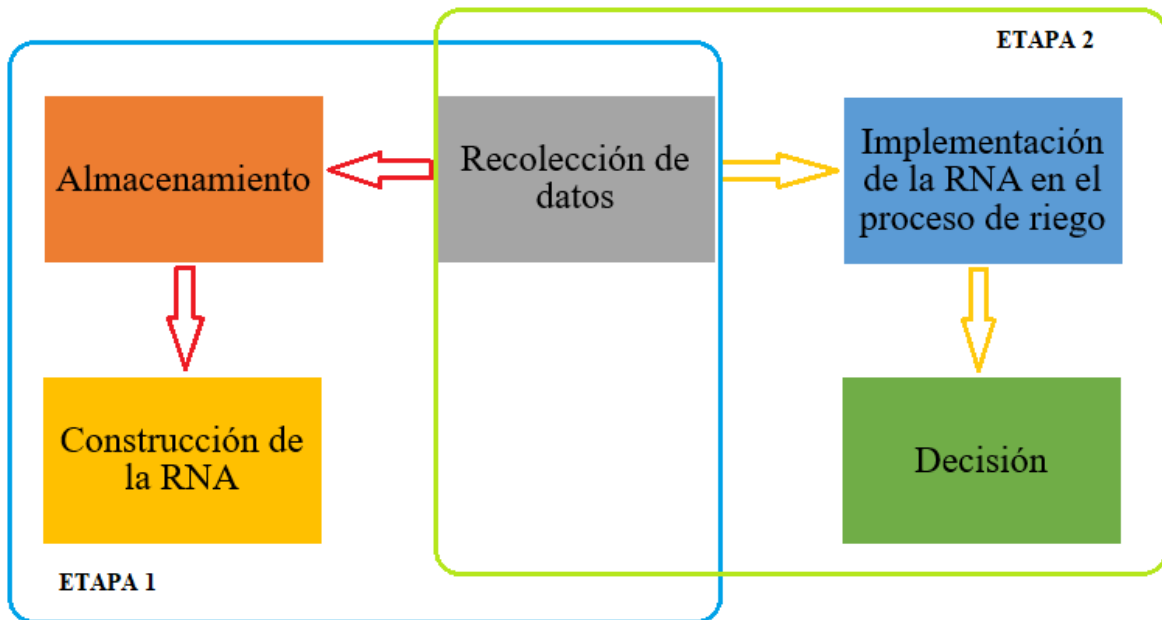
Una vez realizado el análisis y la recolección de información para la selección de los componentes que integrarán el sistema, se procede con el diseño con la finalidad de presentar una visión general del funcionamiento del sistema de una forma estructurada; para poder definir las directrices del diseño del sistema se deben considerar los criterios y limitaciones tomados en cuenta durante el análisis y los requerimientos del sistema que permitirán el desarrollo e implementación del presente proyecto.

Como parte del diseño del sistema se muestra a continuación el diagrama de bloques, diagrama circuital y diagrama de flujo que guiará el funcionamiento físico y lógico del sistema, así como los procesos para desarrollar adecuadamente los scripts y el algoritmo de aprendizaje automático, para subsiguientemente realizar la fase de pruebas unitarias para verificar independientemente cada módulo que forma parte del sistema de riego basado en Inteligencia Artificial, con la finalidad de garantizar el correcto funcionamiento del mismo.

3.9.1. Diagrama de bloques general del sistema

El proceso funcional del sistema se presenta mediante un diagrama de bloques, que permite tener una perspectiva clara de la funcionalidad del sistema propuesto. A continuación, la Figura 28 muestra el diagrama de bloques general, el cual posteriormente se divide en dos etapas; la primera etapa está formada por la recolección de datos ambientales, almacenamiento de dicha información y la construcción del algoritmo de aprendizaje automático (RNA); y, la segunda etapa presenta una retroalimentación del bloque de adquisición de datos ambientales debido a que estos son necesarios para el funcionamiento de la red neuronal en tiempo real, ya que son datos de

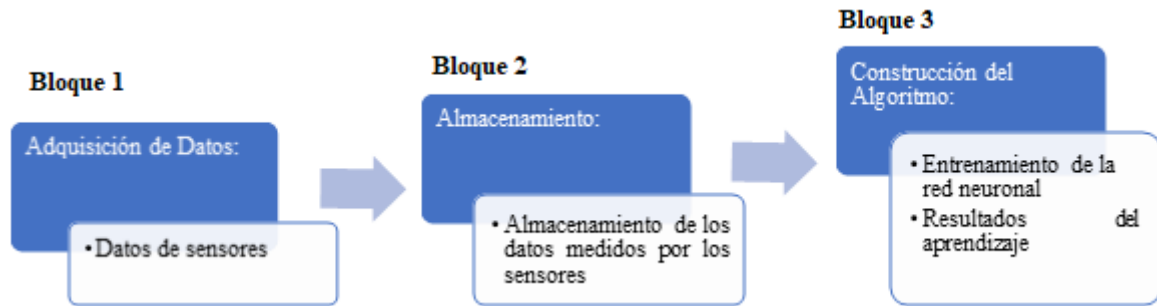
entrada de la red neuronal los cuales van a ser procesados por el algoritmo y de esa forma determinar si es o no necesario iniciar el proceso de riego en el bloque de decisión.



*Figura 28. Diagrama de bloques general del sistema
Fuente. Autoría*

3.9.1.1. Diagrama de bloques de la primera etapa del sistema.

El diagrama de bloques que se muestra a continuación es el que representa a la primera etapa, desarrollada para la implementación del algoritmo de aprendizaje automático para la automatización del riego, la primera etapa está conformada por 3 bloques los cuales a su vez contienen subprocesos; se han trazado los bloques dependiendo de las funciones que cada uno debe cumplir. En la Figura 29 se presenta el diagrama de bloques correspondiente a esta etapa.



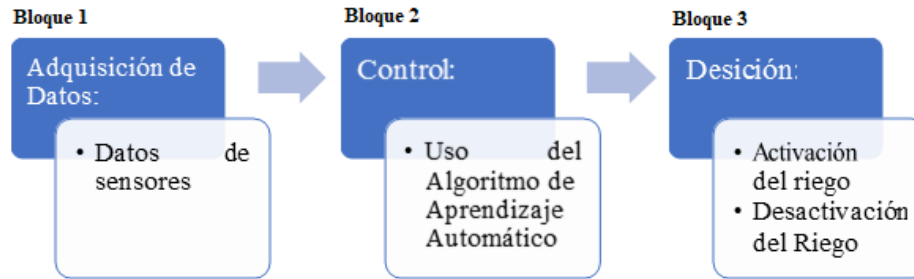
*Figura 29. Diagrama de bloques - Primera etapa del sistema
Fuente. Autoría*

La primera fase del proyecto inicia con la adquisición de datos en el bloque 1, en esta etapa se inicia con la recolección de datos de cada uno de los sensores requeridos de la WSN, los cuales en el bloque 2 van a ser almacenados en una base de datos, esta información será utilizada posteriormente como datos de entrenamiento de la RNA, al decir datos de entrenamiento se hace referencia a la información que va a utilizar la red neuronal para su proceso de aprendizaje.

Finalmente, en el tercer bloque se procede a la construcción del algoritmo de aprendizaje automático, para lo cual se ingresan datos de entrenamiento mencionados en el párrafo anterior, a continuación, se procede a entrenar la red neuronal y mostrar los resultados del aprendizaje automático llevado a cabo.

3.9.1.2. Diagrama de bloques de la segunda etapa

El diagrama de bloques de la segunda etapa del sistema que se muestra en la Figura 30, se presenta con el propósito de dar a conocer el funcionamiento del sistema de automatización de riego, está conformado por tres bloques con una pequeña descripción de las funciones que cada uno realiza.



*Figura 30. Diagrama de bloques - Segunda etapa del sistema
Fuente. Autoría*

En el primer bloque (*Adquisición de datos*) de la segunda etapa se presenta la retroalimentación de la adquisición de datos como ya se mencionó anteriormente este proceso se hace necesario ya que son datos que sirven para que la red neuronal realice las predicciones de riego en tiempo real.

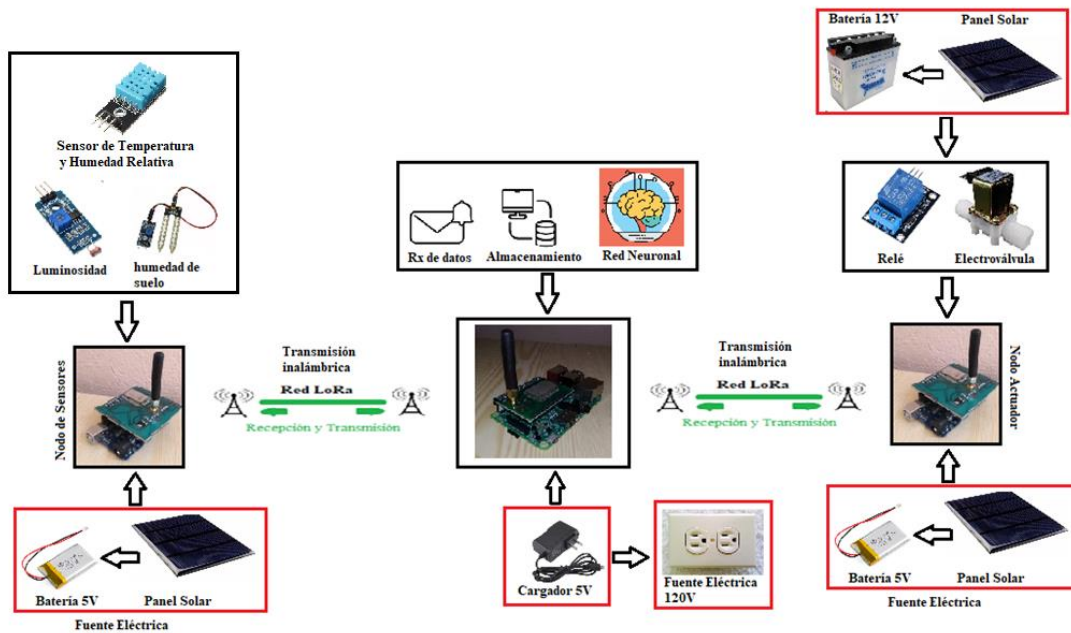
El segundo bloque (*Control*) comprende la etapa de control en la cual entra ya en funcionamiento el algoritmo de aprendizaje automático, construido en la primera etapa del proyecto, la función que se realiza en este bloque es determinar si se requiere o no riego en el cultivo.

Finalmente, en el último bloque (*decisión*) se realiza la función de activar o no el sistema de riego enviando la orden por medio del módulo LoRaWAN al nodo que controla la válvula solenoide.

3.9.2. Topología de la red

El sistema de riego inteligente basado en redes neuronales artificiales está conformado por una red WSN tomada de (Domínguez, 2020), la topología que permite implementar este sistema es la topología en estrella conformada por dos zonas, las cuales son: la zona 1 que está conformada

por la WSN de la cual se obtienen los datos de temperatura ambiental, humedad relativa, luminosidad y humedad de suelo; y nodo actuador el cual está conformado por la electroválvula; y la zona 2 como estación terminal en donde se hace la recepción y almacenamiento de datos enviados desde la zona 1, además es en donde se va a implementar también la red neuronal para el control del riego automático el cual es el propósito de esta investigación, cabe recalcar que para la comunicación entre ambas zonas se implemente tecnología de comunicación inalámbrica LoRaWAN, tal como se muestra en la Figura 31.



*Figura 31. Topología del sistema de riego
Fuente. Autoría*

3.9.3. Diagrama Circuitual del Sistema

En este apartado se muestra los diagramas de conexión tanto de la zona 1 como de la zona 2. Para ello se hace uso del simulador de circuitos online fritzing.

3.9.3.1. Zona 1

Esta sección está comprendida por un nodo que se encarga de la toma de las magnitudes meteorológicas que son los datos proporcionados por los sensores, un módulo de comunicación LoRaWAN para transmitir y recibir los datos hacia la estación de control, la fuente de alimentación necesaria para satisfacer el consumo de energía de cada uno de los elementos de esta zona, el Arduino UNO que se encargara de procesar los datos de los sensores y por un segundo nodo que se encarga de permitir o no el flujo de agua mediante el uso de una electroválvula que es controlada por un relé, además también se hace uso de un Arduino UNO y una fuente de alimentación. En la Figura 32 se muestra el diagrama de conexión del nodo de activación/desactivación del riego y en la Figura 33 se muestra el diagrama de conexión del nodo de toma de datos ambos nodos pertenecientes a la zona 1.

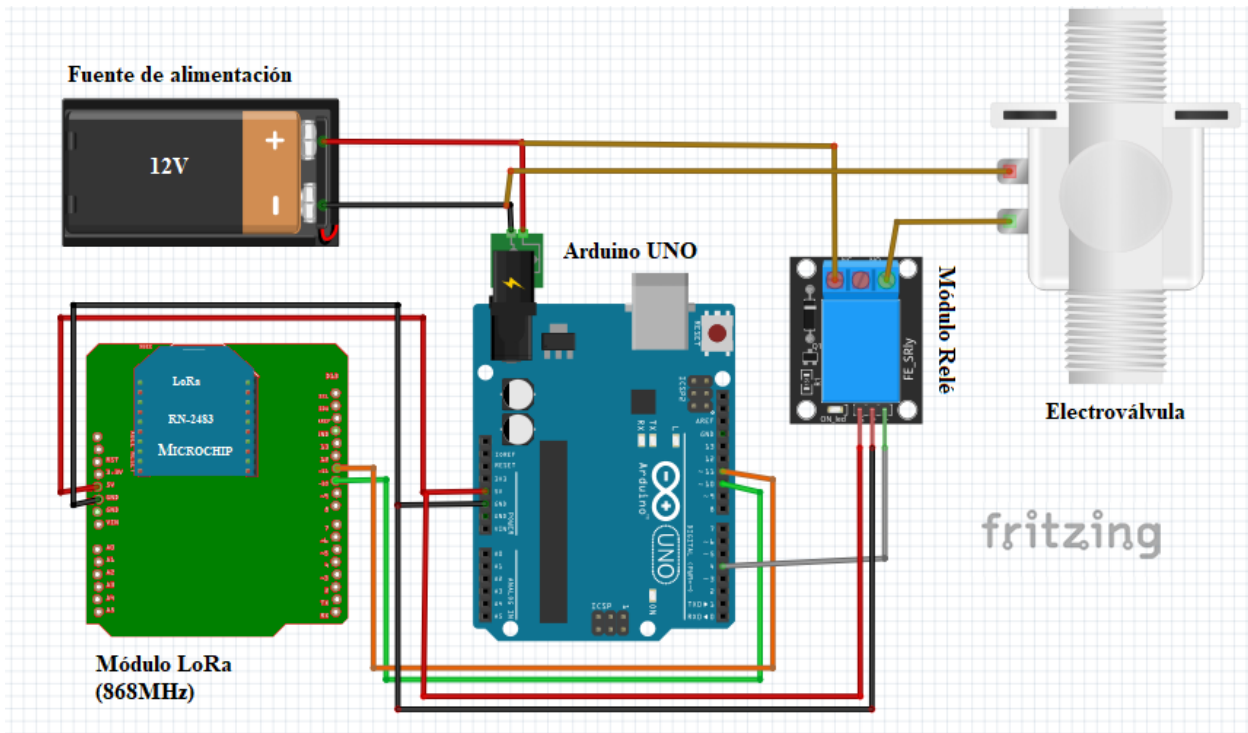


Figura 32. Diagrama de conexión de la Zona 1 (Activación/Desactivación de Riego)

Fuente. Autoría

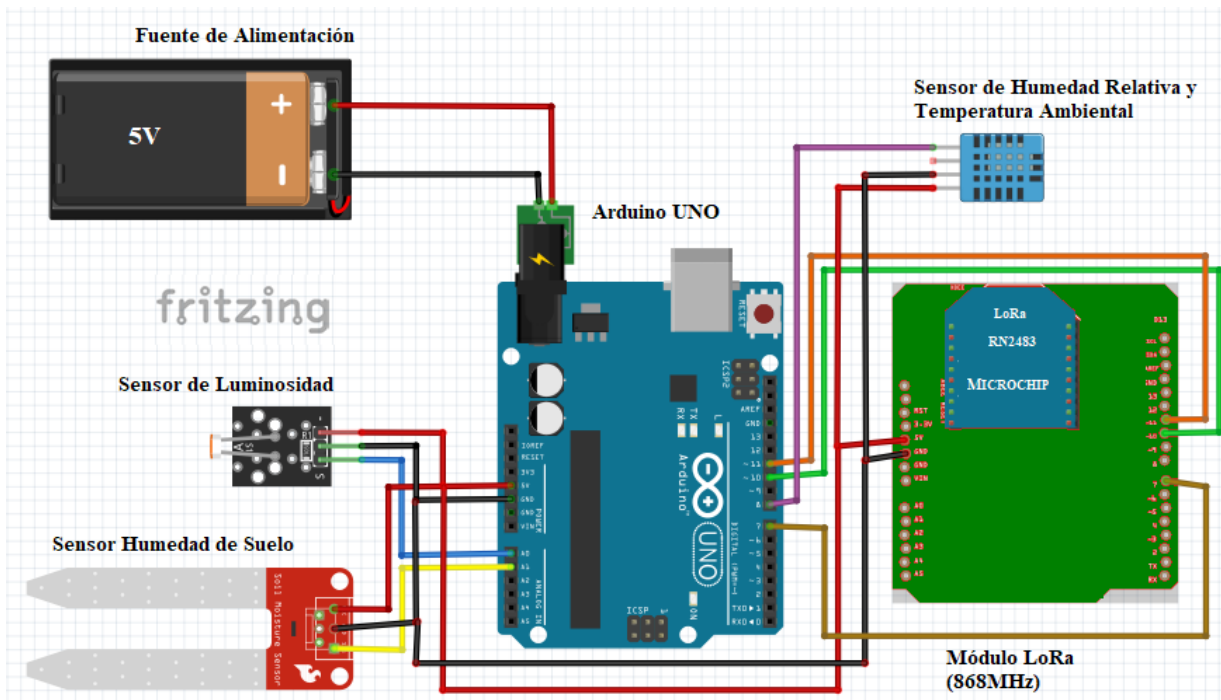


Figura 33. Diagrama de Conexión de la Zona 1 (Recolección de datos)

Fuente. Autoría

3.9.3.2. Zona 2

Es considerada como el área central la cual está compuesta principalmente por dos componentes, tales como: el módulo de comunicación LoRaWAN RN-2483 y el sistema embebido Raspberry Pi 3, que están conectadas con la red de sensores WSN, la cual cumple la función de recibir los datos enviados desde la zona 1 para posteriormente procesar esta información con el algoritmo de redes neuronales artificiales encargado de la automatización del riego, además es la encargada de enviar la orden de encendido o apagado del sistema de riego. En la Figura 34 se muestra el diagrama de conexión del área central o nodo de control correspondiente a la zona 2.

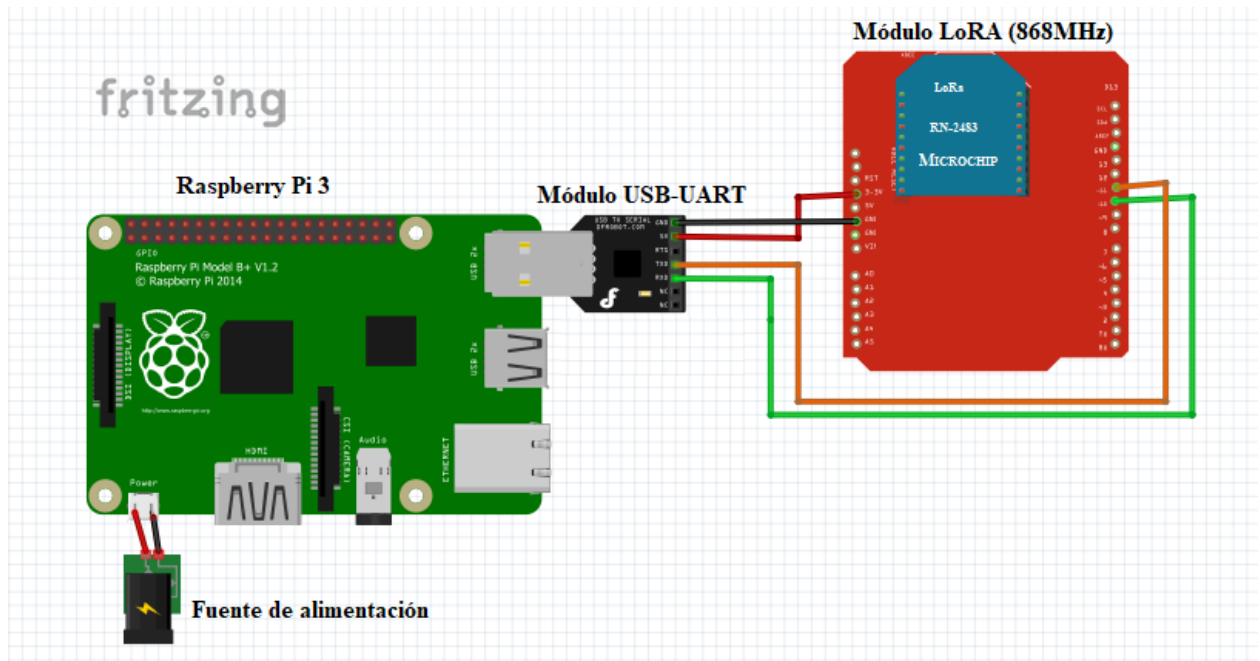


Figura 34. Diagrama de conexión de la Zona 2
Fuente. Autoría

3.9.4. Alimentación del Sistema

Para que existe un correcto funcionamiento del nodo actuador del sistema de riego, se necesita de una fuente de alimentación eléctrica. En este caso se toma en cuenta el hecho de que es un nodo que se va a comunicar de forma inalámbrica con el nodo central y además que en el lugar de su ubicación de este nodo no existe una fuente eléctrica, se va a utilizar se utiliza baterías recargables para proporcionar energía a este nodo.

3.9.4.1. Cálculos de alimentación eléctrica

Para la alimentación eléctrica del nodo actuador se escoge baterías de 5V a 3000mA, tomando en consideración las características de la batería, los cálculos de la vida útil son los siguientes:

Tabla 17. Consumo eléctrico de dispositivos

Elemento	Modo Normal (mA)	Modo Dormido (mA)
Módulo LoRaWAN	40mA	14,2mA
Arduino UNO	46mA	46mA
Electroválvula	350mA	-
TOTAL	436mA	60,2mA

Fuente. Autoría

La Ecuación 8 es utilizada para calcular el consumo de la vida de la batería, como se indica a continuación:

$$Consumo = \frac{(Tcn * Icn) + (Tcd * Icd)}{Tcn + Tcd}$$

Ecuación 8. Fórmula de tiempo de vida de una batería

De donde:

Tcn = Tiempo de Consumo Normal

Tcd = Tiempo de Consumo Dormido

Icn = Intensidad de Corriente Consumo Normal

Icd = Intensidad de Corriente Consumo Dormido

Para la implementación de este nodo se toma en cuenta los siguientes valores:

Tcd = 5 minutos, Tcn = 55 minutos, Icn = 436mA, Icd = 60,2mA

Con los datos detallados anteriormente se reemplaza en la Ecuación 8 y se tiene como resultado el consumo eléctrico del nodo actuador:

$$Consumo = \frac{(300 * 436) + (3300 * 60,2)}{300 + 3300} = 91,52mA$$

Se tiene disponible una batería de 5V con 3000mA, el cálculo de la duración de la batería se lleva a cabo haciendo la división de la capacidad de la batería sobre el consumo de corriente del nodo como se indica a continuación:

$$\textit{Duración de la Batería} = \frac{3000}{91,52} = 32,77 \textit{ horas}$$

3.10. Diseño del software del sistema

El software se encuentra dividido en bloques que son: toma de datos para entrenamiento de la red neuronal (Tx/Rx y almacenamiento) e implementación de la red neuronal.

3.10.1. Recolección de datos para el entrenamiento de la red neuronal

Para la recolección de datos para entrenar a la red neuronal se debe tener en cuenta las características de la planta, en este caso de la alfalfa, para eso se pidió asesoría al Ingeniero Agrónomo Gabriel Chimbo, el cual supo manifestar que la alfalfa es una planta que no consume demasiada agua, además recomendó realizar el proceso de riego durante la noche debido a que al realizar este proceso durante el día el agua tiende a evaporarse mucho más rápido lo que le quita tiempo de hidratación a la planta, razón por la cual no se desarrolla de la mejor manera y se tienen plantas mucho más pequeñas, en cambio al realizar el riego durante la noche se tiene más tiempo de hidratación y la alfalfa se desarrolla de mejor manera alcanzando mayor tamaño. En base a esta información se realiza la toma de muestras, este ciclo tuvo una duración de 20 días con un período de muestreo de 5 minutos. Para las etiquetas de salida de la red neuronal se toma el valor de 1 como activación y el valor de 0 como no activación del sistema de riego. La toma de datos y la asignación de las etiquetas se la realiza tomando en consideración que el riego se lo va a realizar

durante la noche debido a que la planta tiene más tiempo para hidratarse de forma adecuada, ya que durante un día soleado la evaporación del agua es mucho más rápida lo que quiere decir que el recurso hídrico se desperdicia y le resta tiempo de hidratación a la planta. En el literal 3.10.3.1 se puede observar el diagrama de flujo correspondiente a este punto de recolección de datos.

3.10.2. Diseño de la red neuronal artificial para automatizar el sistema de riego

El diseño de la red neuronal se desarrolla principalmente a partir de tres elementos: el primero de estos son los datos de entrada que corresponden a los datos recolectados y enviados por el nodo de sensores del cultivo, sensores de humedad de suelo, humedad relativa, temperatura ambiente y luminosidad, de todos estos sensores se toma en consideración solamente los valores de luminosidad y humedad del suelo, ya que se desea realizar el riego durante la noche; el segundo elemento corresponde al algoritmo de la red neuronal que será la encargada de procesar la información de sus entradas y el tercer elemento corresponde a los datos de salida en este caso un valor que represente la activación (1) o no activación (0) del sistema de riego, tal como se muestra en la Figura 35.

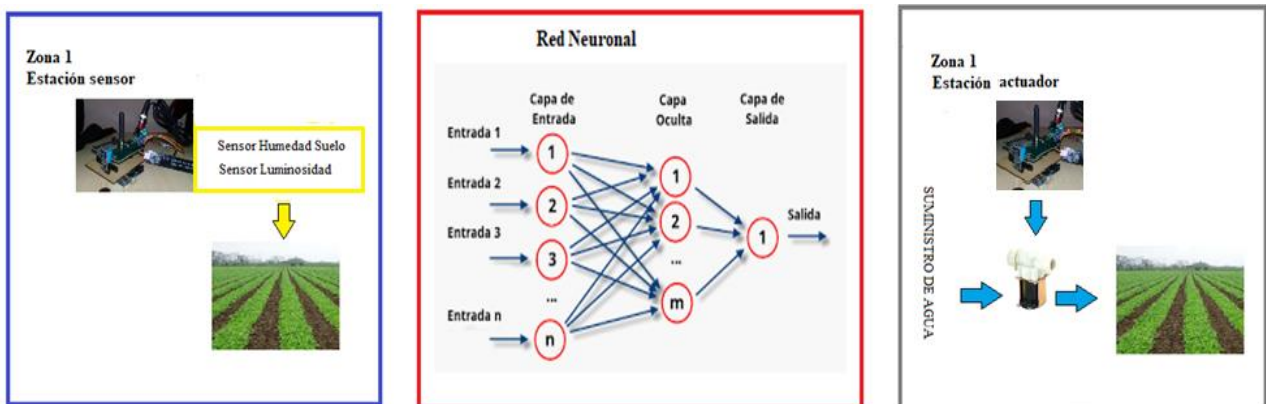
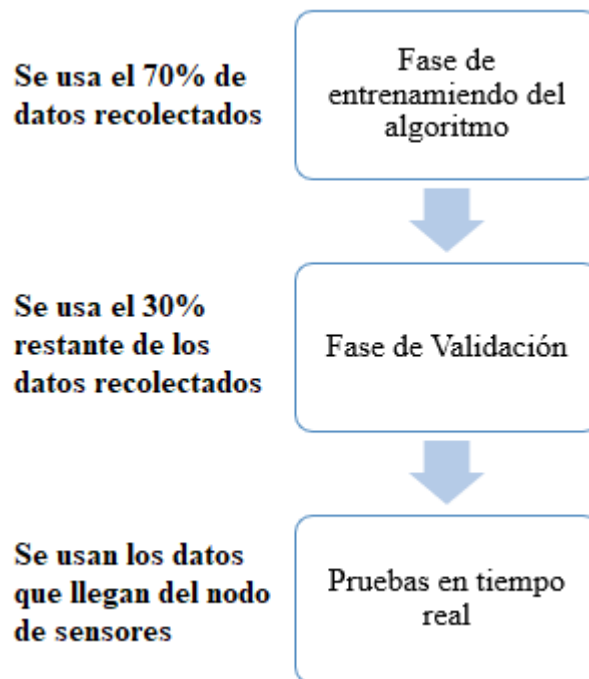


Figura 35. Elementos de la Red Neuronal Artificial
Fuente. Autoría

En la figura 36 se muestra un diagrama de bloques del proceso de creación de la RNA, la cual inicia por el proceso de entrenamiento del algoritmo de aprendizaje automático, una vez finalizado el proceso de entrenamiento de la red neuronal se procede a la fase de validación del mismo donde se determina que tan precisa es la Red Neuronal Artificial y finalmente se pone a prueba el modelo entrenado con datos del cultivo en tiempo real.



*Figura 36. Proceso de implementación de la red neuronal
Fuente. Autoría*

3.10.2.1. Número de capas de la RNA

La primera característica es definir el número de capas que va a tener la red neuronal. La red neuronal a usarse en esta investigación consta de 3 capas: capa de entrada, una capa intermedia y una capa de salida, tal como se muestra en la Figura 37.



Figura 37. Representación del número de capas de la RNA
Fuente. Autoría

3.10.2.2. Número de neuronas en la capa intermedia o capa oculta

La elección de un número adecuado de neuronas en la capa oculta es muy importante, ya que si se hace uso de muy pocas neuronas se tendría pocos recursos para resolver el problema para la cual fue creada la red neuronal, en cambio si se hace uso de demasiadas neuronas aumenta el tiempo de entrenamiento además puede causar problemas de sobre ajuste, razón por la cual para obtener un número aproximado de neuronas en la capa oculta se usa la regla de la pirámide incremental (Masters, 1993).

$$h = \sqrt{m * p}$$

Ecuación 9. Regla de la pirámide incremental

Donde p es el número de neuronas a la entrada y m es el número de neuronas en la salida. Como se mencionó anteriormente los parámetros ambientales que se van a tomar en cuenta son cuatro, valores de temperatura ambiental, humedad relativa, luminosidad y humedad del suelo, entonces el número de neuronas en la capa de entrada es cuatro y para la capa de salida el número de neuronas es de uno ya que se tiene solamente la variable de riego como resultado. Por lo tanto, reemplazamos estos valores en la Ecuación 9 y tenemos:

$$h = \sqrt{1 * 4} = \sqrt{4} = 2$$

En base al resultado anterior se determina que la red neuronal debe tener dos neuronas en la capa oculta.

3.10.2.3. Modelo de la red neuronal artificial del sistema

Una vez que se conoce el número de capas de la red neuronal y el número de neuronas en cada una de las capas se procede a modelar la red neuronal la cual se puede observar en la Figura 38, en la cual se tienen cuatro neuronas en la capa de entrada las cuales corresponden a cada una de las variables ambientales medidas (temperatura ambiental, humedad relativa, humedad de suelo, luminosidad), en la capa oculta tenemos dos neuronas y en la capa de salida una sola neurona que corresponde a la variable de riego (encendido o apagado).

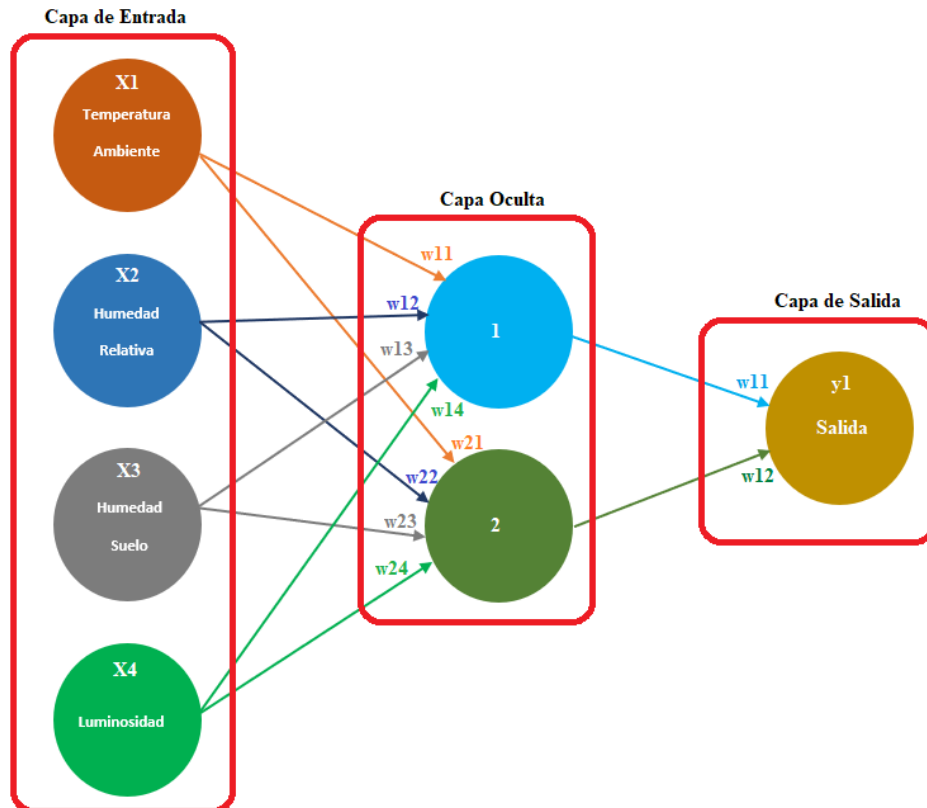


Figura 38. Modelo de la RNA
Fuente. Autoría

3.10.2.4. Función de activación

La función de activación de utilizada en el desarrollo de esta investigación es la ReLU, esta es una función simple que tiene una gran importancia práctica debido a que el cálculo es bastante rápido a diferencia de otras técnicas como las funciones: sigmoide, tangente hiperbólica (tanh), lineal, entre otras.

3.10.2.5. Extracción de datos de entrenamiento

Se cuenta con una base de datos que contiene variables de humedad relativa, temperatura ambiente, humedad del suelo, luminosidad la cual consta de 4660 datos por cada una de estas variables, donde el 70% de esta información se utiliza para la fase de aprendizaje y el 30% restante para pruebas de validación. Algunos de los datos adquiridos se presentan en la Figura 39.

temp	humr	hums	lum	riego
14	83	58.5	34.2	0
14	83	58.5	33.5	0
14	83	58.5	32.8	0
14	83	58.5	32.2	0
14	82	58.5	31.4	0
14	84	58.6	30.6	0
14	85	58.5	29.9	0
14	84	58.5	29.1	0
14	84	58.5	28.5	0
14	83	58.5	27.9	0
14	83	58.5	27.3	0
14	85	58.5	26.8	0

Figura 39. Base de datos para el entrenamiento de la RNA
Fuente. Autoría

De la Figura 39 se puede deducir que la red neuronal tendrá una sola salida la cual tiene dos estados. El primer estado es iniciar el proceso de riego que se la representa con el valor de 1 y

el segundo estado es no iniciar el riego representada con el valor de 0, es decir, se implementa una red neuronal con una salida binaria (1 o 0).

3.10.2.6.Preprocesamiento de datos de entrenamiento

Cuando se trabaja con un modelo de aprendizaje automático, es muy importante escalar los datos que va a ser utilizados para el entrenamiento del algoritmo para este caso de la red neuronal artificial, debido a que se logra un mejor funcionamiento o una convergencia más rápida cuando las características o variables de entrada están en una escala centrada alrededor de cero. La librería SciKit Learn proporciona una clase para escalar valores de manera rápida y fácil, para ello existe la clase StandarScaler de la librería mencionada. En la Figura 40 se puede observar un conjunto de datos escalados.

```
[[-0.69365181  0.59582074 -0.26431529 -0.63613449]
 [-0.33099635  0.64646601 -2.35467185 -0.8694532 ]
 [ 0.93829775 -1.32869967 -0.14709904  1.42965368]
 ...
 [-0.51232408  0.64646601 -0.11779497 -0.65186384]
 [-0.69365181  0.59582074  0.04826139 -0.40543733]
 [ 1.48228094 -1.83515241  0.34130203  1.47946329]]
```

Figura 40. Etapa de preprocesamiento de datos
Fuente. Autoría

3.10.2.7.Algoritmo de entrenamiento

El entrenamiento de la red neuronal se basa en el reajuste de los pesos en cada una de las entradas de las neuronas, los pesos son valores numéricos que precisan la fuerza de enlace sináptico entre dos neuronas. Para esta investigación se utiliza el algoritmo de backpropagation, el cual consta de un procedimiento de propagación del error de atrás hacia adelante, es decir, se empieza

calculando el error en la capa de salida para posteriormente realizar un reajuste de los pesos unidos a esta última capa para lograr una reducción de este error calculado. Seguido de esto se calcula el error en la capa intermedia y de la misma manera que en la capa de salida se reajustan los pesos de esta capa con la finalidad de reducir el error, por ende, tener una mejor precisión en los resultados.

3.10.2.8. Pesos actualizados de la RNA

Una vez terminado el proceso de entrenamiento de la red neuronal artificial se puede verificar los pesos actualizados de los enlaces entre la capa de entrada y la capa intermedia; y entre la capa intermedia y capa de salida, como se observa en la Figura 41.

```
Pesos entre: capa de entrada - capa oculta
[[-0.69366773 -0.03024093]
 [ 0.46314999 -0.03321379]
 [-0.61825518 -0.03216995]
 [-0.55936157 -0.03039094]]
Pesos entre: capa oculta - capa de salida
[[-2.08155722]
 [-0.0136937 ]]
```

Figura 41. Pesos actualizados de la RNA
Fuente. Autoría

3.10.2.9. Rendimiento de la red neuronal

Luego de realizar el entrenamiento de la red neuronal se procede a verificar en software el porcentaje de efectividad de la RNA una de las formas en las que se realiza esto es mediante el uso de una matriz de confusión, en la cual se tienen la cantidad de falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos, con esta información podemos verificar el grado de acierto que va a tener el algoritmo, la Tabla 18 muestra la forma de la matriz de confusión.

Tabla 18. Matriz de confusión

Matiz de Confusión		Predicción	
		Negativo	Positivo
Valor Real	Negativo	a	b
	Positivo	c	d

Fuente. (Amores, 2017)

Para ello se va a utilizar la Ecuación 10, donde, P es la precisión del algoritmo, a representa la cantidad de verdaderos negativos, b la de falsos positivos, c la de falsos negativos y d los verdaderos positivos.

$$P = \frac{a + d}{a + b + c + d}$$

Ecuación 10. Fórmula para calcular la precisión de la RNA

Fuente. (Amores, 2017)

A continuación, se presenta la matriz de confusión resultante de la RNA implementada en el desarrollo de este proyecto.

Matiz de Confusión		Predicción	
		Negativo	Positivo
Valor Real	Negativo	570	31
	Positivo	28	485

Ahora con los datos que se visualizan en la matriz de confusión procedemos a reemplazar los valores en la Ecuación 10, para calcular el grado de efectividad de la red neuronal artificial.

$$P = \frac{570 + 485}{570 + 31 + 28 + 485}$$

$$P = \frac{1055}{1114} = 0,947$$

Mediante el cálculo podemos ver que la red neuronal posee un 94.7% de precisión.

En base a los datos arrojados en la matriz de confusión también podemos calcular la sensibilidad representada por la Ecuación 11 y la especificidad por la Ecuación 12.

$$\text{Sensibilidad} = \frac{a}{a + c}$$

*Ecuación 11. Fórmula para calcular la sensibilidad
Fuente.*

$$\text{Especificidad} = \frac{d}{b + d}$$

*Ecuación 12. Fórmula para calcular la especificidad
Fuente.*

Reemplazamos los valores mostrados en la Tabla 18 correspondiente a la matriz de confusión, en la Ecuación 11 y Ecuación 12 para obtener el porcentaje de sensibilidad y especificidad.

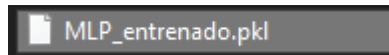
$$\text{Sensibilidad} = \frac{570}{570 + 28} = 0.953$$

$$\text{Especificidad} = \frac{485}{31 + 485} = 0.939$$

Luego de reemplazar los valores en cada una de las ecuaciones correspondientes tenemos un porcentaje del 95.3% para la sensibilidad y 93.9% para la especificidad de la red neuronal artificial.

3.10.2.10. Guardar red neuronal entrenada

Después de entrenar un modelo de RNA, suele ser necesario tener una forma de guardar una red neuronal y así evitar el proceso de entrenamiento cada vez que ejecutamos el script que contenga la RNA programada. Para ello la librería sklearn, ofrece un módulo para realizar dicha acción el cual nos guarda un archivo “.pkl” que se visualiza en la Figura 42, el cual contiene todos los parámetros de la red neuronal ya entrenada y así poder cargarla nuevamente en otro proyecto de ser necesario. En el Anexo 4 se puede visualizar la forma como se logra guardar el modelo entrenado.



*Figura 42. Archivo “.pkl” generado
Fuente. Autoría*

3.10.3. Diagramas de Flujo

Es necesario detallar el algoritmo de funcionamiento que se desarrolla en estas dos zonas, con el objetivo de determinar el inicio y fin del papel que se cumple en cada una de estas.

3.10.3.1. Recolección de datos para entrenamiento de la red neuronal.

La recolección de datos consiste en almacenar la información enviada desde los nodos para posteriormente ser utilizada en el entrenamiento de la red neuronal. El proceso que se muestra en la Figura 43 corresponde al proceso que se lleva a cabo en el nodo de la zona 1 el cual se encarga de recolectar y enviar los datos de los sensores al nodo central, mientras en la Figura 44 muestra el proceso que se hace en el nodo de la zona 2 en cual se encarga de recibir y almacenar los datos.

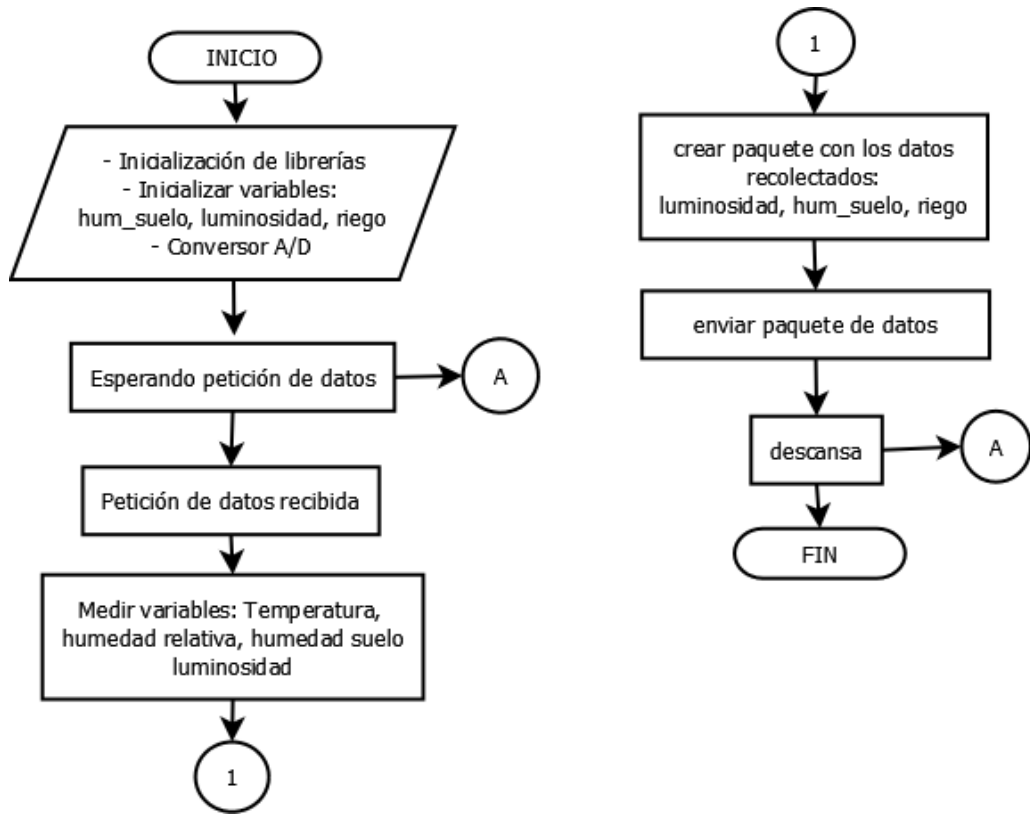
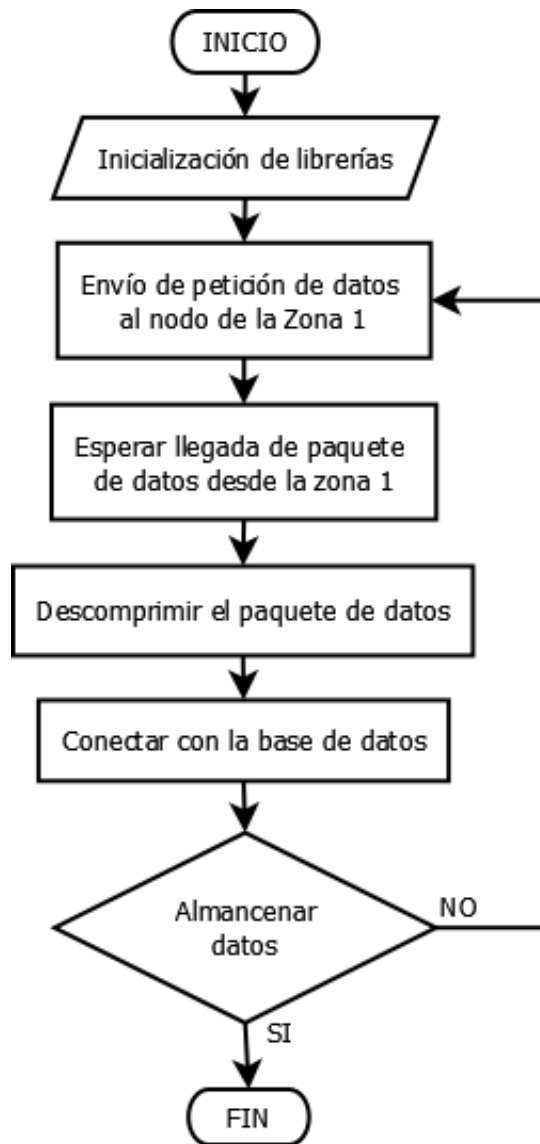


Figura 43. Diagrama de Flujo Zona 1 (Recolección de datos para entrenamiento de la RNA)
Fuente. Autoría



*Figura 44. Diagrama de Flujo Zona 2 (Recolección de datos para entrenamiento de la RNA)
Fuente. Autoría*

3.10.3.2. Implementación del Sistema de Riego Inteligente

En la Figura 45 se muestra el proceso de funcionamiento del sistema de riego inteligente, la cual va a ser la encargada de determinar si es o no necesario realizar el riego en el cultivo de alfalfa.

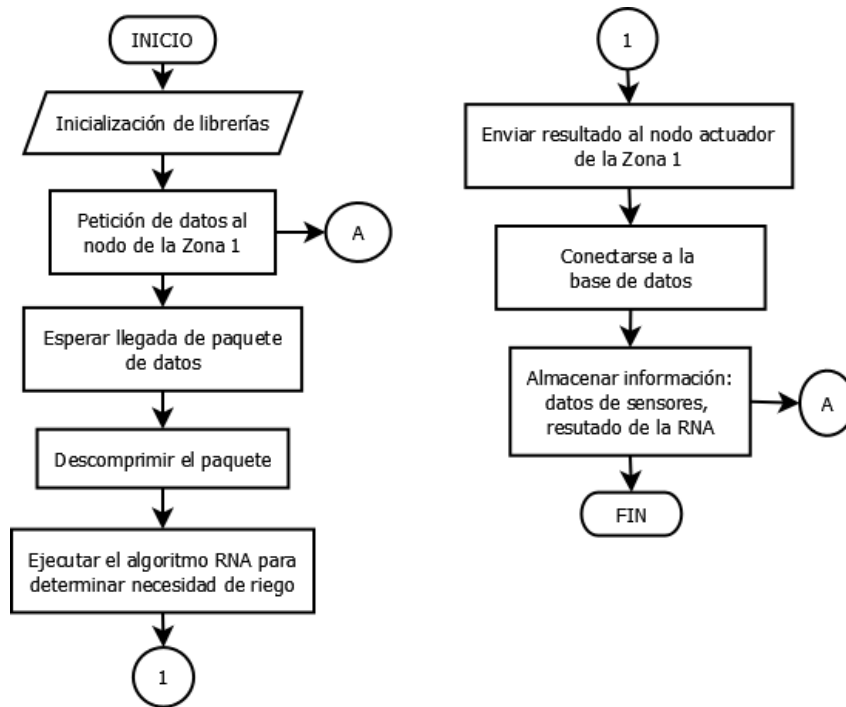


Figura 45. Diagrama de flujo implementación del Sistema de Riego Inteligente
Fuente. Autoría

3.10.3.3. Diagrama de flujo del nodo actuador

En la Figura 46 se puede observar el proceso que realiza el nodo actuador, es decir, este nodo es el encargado de dar o no paso al flujo de agua hacia el cultivo de alfalfa en base al resultado que se obtiene en la salida de la red neuronal.

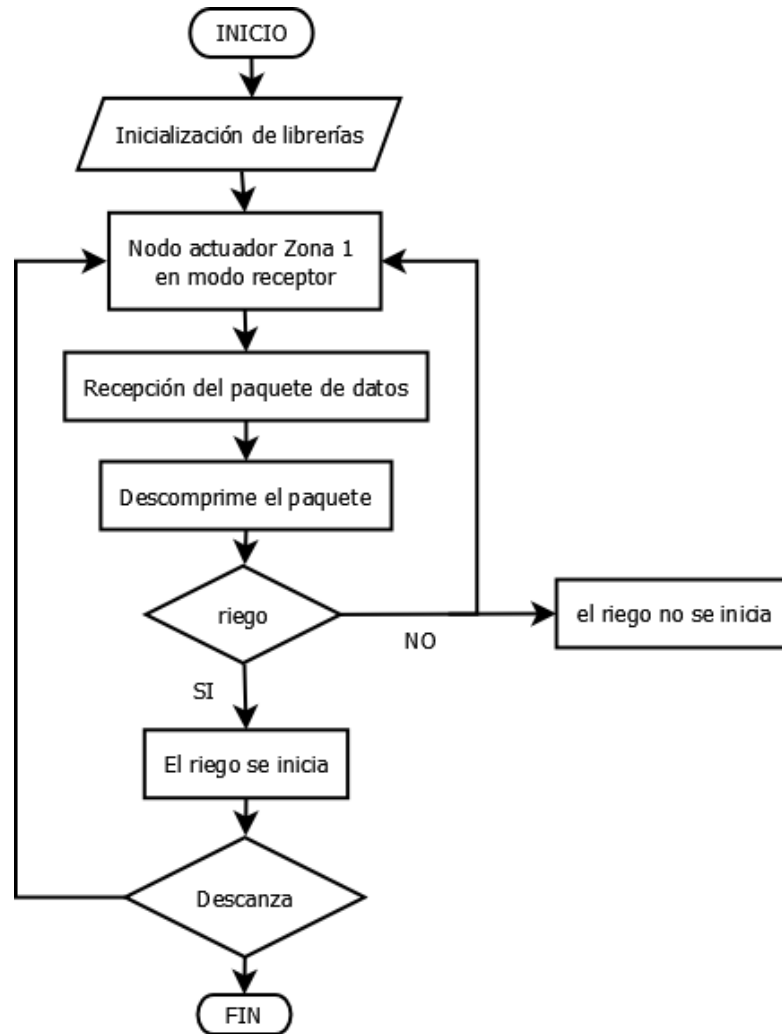


Figura 46. Diagrama de flujo del nodo actuador

Fuente. Autoría

3.11. Cálculos del enlace inalámbrico entre la zona 1 y la zona 2

A continuación, se presentan los cálculos de enlace inalámbrico entre la zona 1, lugar de instalación del nodo actuador y la zona 2, lugar de la implementación de la red neuronal. El cálculo está basado en el modelo de Okumura-Hata correspondiente a la Ecuación 12, el cual representa a un modelo de propagación diseñado para realizar estimaciones del alcance de enlaces inalámbricos, el cual trabaja dentro de un rango de frecuencias que va desde los 150 MHz a 1500

MHz, además permite obtener los valores de pérdidas durante la propagación de la señal. A continuación, se muestra el valor de pérdidas de un enlace entre la zona 1 y la zona 2.

$$L(\text{Urbano})(dB) = 69.55 + 26.16 \log fc - 13.82 \log h_{te} - a(h_{re}) + (44.9 - 6.55 \log d)$$

Ecuación 13. Ecuación del modelo Hokumura - Hata

De donde: f_c corresponde a la frecuencia de la portadora valor que debe estar en unidad de MHz, h_{te} representa a la altura de la antena transmisora, h_{re} es la altura de la antena receptora, $a(h_{re})$ es el factor de corrección por la altura efectiva del móvil que es función del tipo de área de servicio y d que representa la distancia entre el transmisor y receptor en kilómetros.

Reemplazando valores propios del sistema, en la *Ecuación 4*: tenemos a la frecuencia con un valor de 868Mhz, la altura de la antena transmisora de 3.5 m, la altura de la antena receptora de 4.5 m, al factor de corrección igual a 5.08 y a la distancia entre los dos nodos de 0.425 Km.

$$L(dB) = 69.55 + 26.16 \log(868) - 13.82 \log(3.5) - 5.08 + 44.9 - 6.55 \log(0.425)$$

$$L = 87,5dB$$

Con respecto a la Ecuación 13, se tiene un valor de pérdida de 87.5 dB, este valor se lo puede comprobar al realizar la simulación de este enlace inalámbrico en Radio Mobile tal como se muestra en la Figura 47, comprobando que el nivel de pérdida por medio de la ecuación del modelo de Okumura-Hata y la simulación en Radio Mobile es correcta, donde además se puede observar que entre ambos puntos existe línea de vista y la primera zona de Fresnel esta despejada garantizándose así que existe una buena comunicación entre las dos zonas.

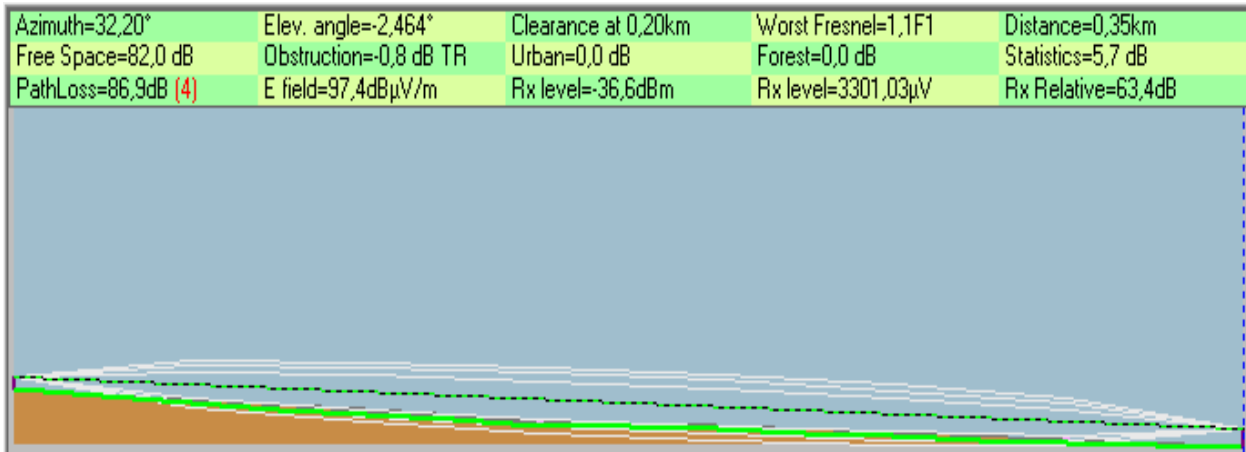


Figura 47. Enlace en Radio Mobile
Fuente. Autoría

Con la ayuda del software Google Earth se puede visualizar de mejor manera en la Figura 48 que la primera Zona de Fresnel del enlace inalámbrico entre la zona 1 y la zona 2 representada por la línea color verde, como se observa la línea no presenta ningún corte en todo su trayecto lo que quiere decir que la zona esta despejada y se tiene un enlace inalámbrico adecuado.

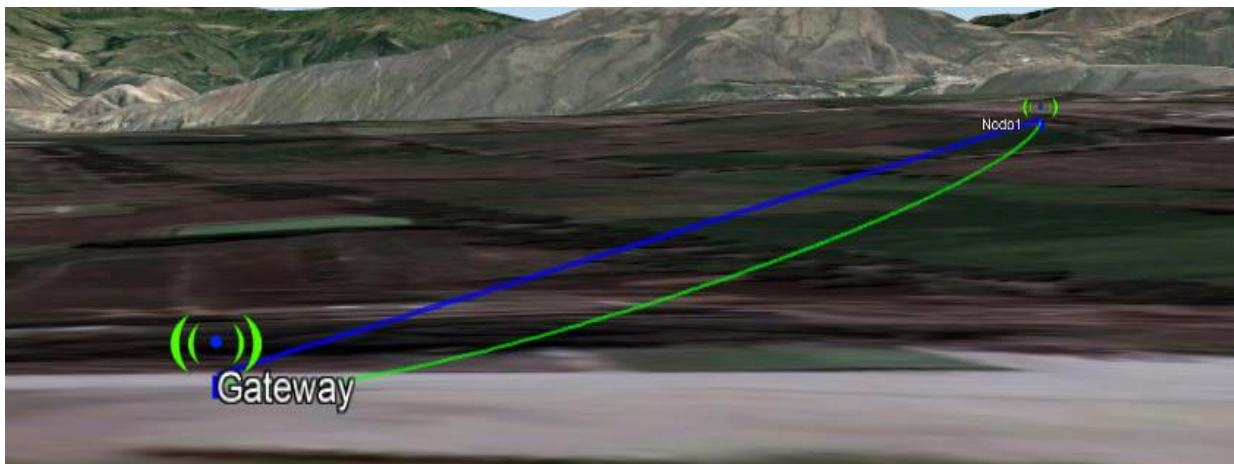


Figura 48. Visualización del enlace inalámbrico en Google Earth
Fuente. Autoría

4. Capítulo IV. Pruebas de Funcionamiento

En este capítulo se hace mención a las pruebas realizadas en el cultivo de alfalfa; es decir son las pruebas de laboratorio en donde principalmente se realiza la verificación del funcionamiento de sistema de riego, para ello se analizan todos los subsistemas del prototipo final verificando la funcionalidad del sistema completo, posterior a eso, se analizan el cultivo de muestra verificando el desarrollo de la planta.

4.1. Pruebas de funcionalidad del sistema de riego

A continuación, se detalla los casos de pruebas que determinan si el sistema cumple o no con cada uno de los requerimientos planteados en el capítulo anterior en el apartado 3.6. Se realiza un test de pruebas basados en el modelo en V, se presentan algunos casos de estudio tales como: test eléctrico, test del subsistema pasivo, test de subsistema activo, test de hardware, test de software, test de aplicación y finalmente un test de funcionalidad.

4.1.1. Test eléctrico del sistema

Este apartado consiste en comprobar que los requerimientos de alimentación eléctrica se cumplan, con la finalidad de lograr un correcto funcionamiento de los dispositivos electrónicos, en la Tabla 19 se presenta el test eléctrico y sus requerimientos.

Tabla 19. Test del subsistema eléctrico

Test Eléctrico	Subsistema de prueba:	Eléctrico
	Prueba de despliegue	Si / No
Descripción: Pruebas del diseño del prototipo basado en subsistema eléctrico		
Prerrequisitos:		
<ol style="list-style-type: none"> 1. Adquisición de baterías de 5V y 12V. 2. Conexiones desde las placas programables hacia la placa de comunicación, los sensores y electroválvula 3. Verificar posiciones de los dispositivos. 		
Pasos:		
<ul style="list-style-type: none"> ✓ Verificación del encendido de las placas programables ✓ Verificación de encendido de la placa de comunicación, los sensores y electroválvula ✓ Comprobación de voltaje de baterías ✓ Comprobación de suministro de energía continua hacia los dispositivos electrónicos ✓ Verificación de duración de tiempo de funcionamiento de la batería en funcionamiento continuo. 		
Resultado esperado:		
<p>Contar con una fuente eléctrica necesaria para alimentar tanto a las placas programables y a los diferentes dispositivos electrónicos que esta conectados a ellas, para que el dispositivo cumpla con las funciones para las cuales fue diseñado.</p>		

Fuente: Autoría

El resultado obtenido es satisfactorio comprobándose el encendido de los dispositivos mediante el indicador led de las placas programables y de las placas de comunicación, al no tener un led de encendido en los sensores y en la electroválvula se comprueba su funcionamiento con la medición de voltaje en sus terminales.

El voltaje es el adecuado, por lo cual se procede al uso de una batería de 5V a 3000mA para el nodo de sensores y de una batería de 12V a 3000mA para el nodo actuador esto debido a que la electroválvula se alimenta con un voltaje de 12V.

4.1.2. Test del subsistema pasivo

Consiste en comprobar si se cumplen los requerimientos de conectividad del dispositivo con el fin de interconectar todos los elementos del sistema tal cual se evidencia en el esquema de diseño del circuito mostrado en el apartado del diseño del hardware. En la Tabla 20 se muestra de manera detallada el test del subsistema pasivo y sus requerimientos.

Tabla 20. Pruebas del subsistema pasivo

Test del subsistema pasivo	Subsistema de prueba:	Pasivo
	Prueba de despliegue	Si / No
Descripción: Pruebas del diseño del prototipo basado en el subsistema pasivo		
Prerrequisitos:		
<ol style="list-style-type: none">1. Adquisición de cables USB tipo B.2. Adquisición de cables para conexiones de los sensores y electroválvula.3. Circuitos impresos del nodo de sensores y del nodo actuador.		
Pasos:		
<ul style="list-style-type: none">✓ Verificación del estado de los cables.✓ Verificación de las conexiones de los dispositivos electrónicos hacia la placa programable.✓ Reconocimiento de la placa programable en el computador al momento de conectar con el cable USB.✓ Verificación de continuidad en las pistas de los circuitos impresos.		
Resultado esperado:		
Las conexiones de los dispositivos electrónicos hacia las placas programables deben ser estables y no deben desconectarse fácilmente, el funcionamiento de los dispositivos no debe verse afectado por la desconexión de los cables y debe existir continuidad en el circuito para garantizar un adecuado funcionamiento.		

Fuente: Autoría

Como resultado de este apartado se verifica con éxito las conexiones de los diferentes dispositivos, en donde se comprueba que existe conectividad entre las pistas de cobre de la baquelita, los sensores, las placas de comunicación y la electroválvula son alimentadas correctamente y están listos para cumplir con sus funciones.

4.1.3. Test del subsistema activo

Consiste en comprobar si el dispositivo encargado del procesamiento trabaja correctamente, en esta prueba se valida el reconocimiento de las placas programables en la PC, la conexión a través del puerto COM y la carga del sketch en las mismas, además se verifica el funcionamiento del módulo LoRa.WAN En la Tabla 21 se presenta en detalle el test del subsistema activo y sus requerimientos.

Tabla 21. Pruebas de verificación del subsistema activo

Test del subsistema activo	Subsistema de prueba:	Activo
	Prueba de despliegue	Si / No
Descripción: Pruebas del diseño del prototipo basado en el subsistema activo		
Prerrequisitos:		
<ol style="list-style-type: none">1. Adquisición de placas de procesamiento y placas de comunicación2. Revisión de puertos COM en la PC		
Pasos:		
<ul style="list-style-type: none">✓ Reconocimiento de los arduinos en la PC.✓ Cargar los sketches de forma exitosa.✓ Mantener conectividad en la Red LoRaWAN.		
Resultado esperado:		
Conexión y reconocimiento del puerto COM de las placas de procesamiento en el PC, luego proceder a cargar los sketches, verificar la conexión de los nodos tanto de sensores como del nodo actuador al Gateway, mediante la petición de datos.		

Fuente: Autoría

Se tiene como resultado que las placas de procesamiento son reconocidas por la PC a través de los puertos COM5 y COM6. En la Figura 49 se muestra el reconocimiento de los arduinos en la PC por medio del IDE de Arduino.

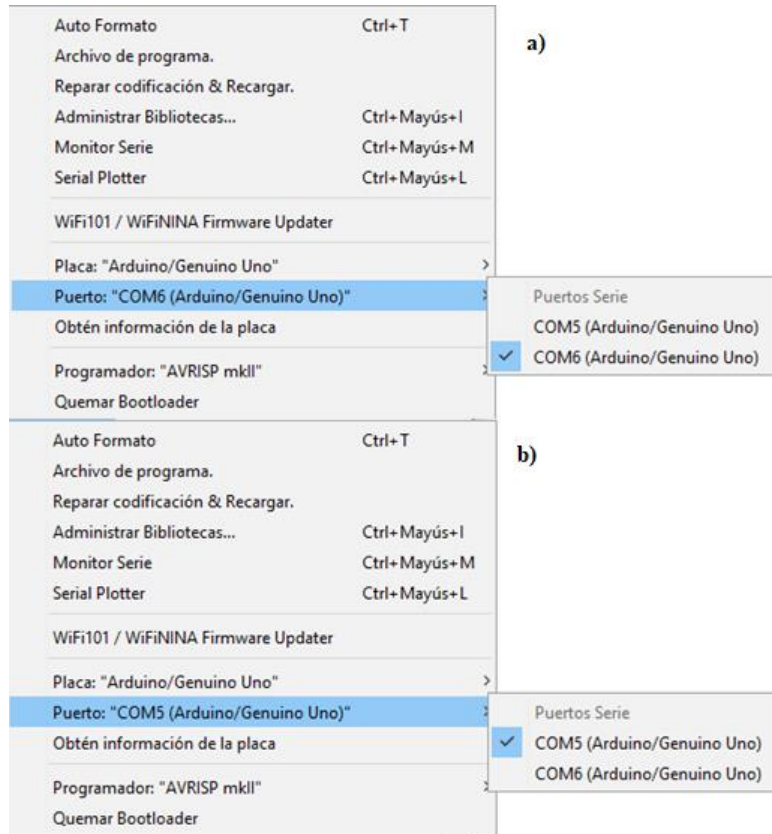


Figura 49. Reconocimiento de las placas de procesamiento a) nodo de sensores b) nodo actuador
Fuente. Autoría

Se verifica la carga de los sketches en cada una de las placas de procesamiento, así como se muestra en la Figura 50.

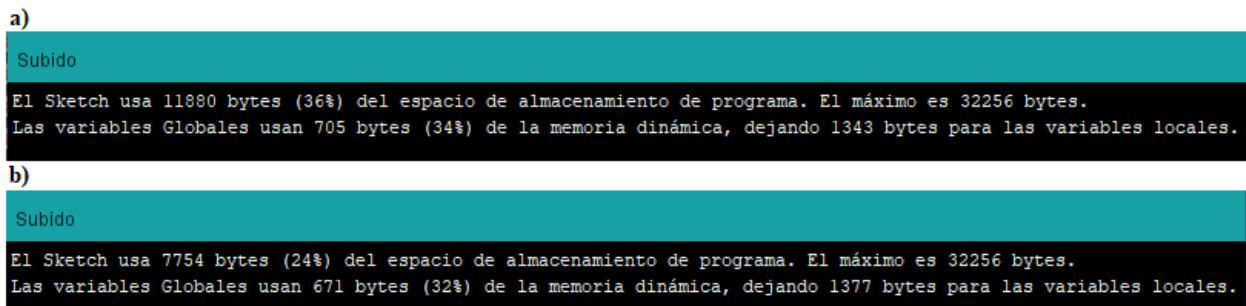


Figura 50. Carga exitosa de sketches en los nodos y uso de memoria
Fuente. Autoría

También se comprueba que existe conexión entre los nodos y el gateway tal como se aprecia en la Figura 51.



Figura 51. Conexión exitosa de nodos con el gateway
Fuente. Autoría

4.1.4. Test de hardware

Consiste en la evaluación del funcionamiento de los dispositivos que conforman el sistema, en este caso son los sensores encargados de la medición de variables tanto de luminosidad como de humedad de suelo, además de la electroválvula. En la Tabla 22 que se presenta a continuación se detalla el test de hardware y sus requerimientos.

Tabla 22. Test de hardware

Test de hardware	Subsistema de prueba:	Hardware
	Prueba de despliegue	Si / No
Descripción: Pruebas de los dispositivos que conforman el sistema		
Prerrequisitos:		
<ol style="list-style-type: none"> 1. Adquisición de los sensores de luminosidad y humedad de suelo, además de las placas de procesamiento, placas de comunicación y la electroválvula. 2. Conexión de dispositivos. 3. Inclusión de las librerías correspondientes 		
Pasos:		
<ul style="list-style-type: none"> ✓ Verificar el encendido de cada elemento. ✓ Análisis de funcionalidad de cada dispositivo 		

-
- ✓ Funcionamiento de la placa de procesamiento y de comunicación.
 - ✓ Verificación de conectores.
-

Resultado esperado:

Los sensores miden correctamente valores de humedad de suelo y luminosidad, la electroválvula se abre y cierra de forma adecuada.

Fuente: Autoría

Como resultado de este test se obtiene que todos los sensores funcionan correctamente, se evidencia su encendido por el led indicador del sensor de humedad del suelo, el sensor de luminosidad no posee un led indicador por lo cual se evidencia su encendido con la medición de voltajes en sus pines.

En la Figura 52 se muestra la lectura exitosa de los sensores de humedad de suelo y de luminosidad verificándose así el correcto funcionamiento de cada sensor.

```
Recolectando información....  
Luminosidad: 779  
humedad de suelo: 531
```

*Figura 52. Lectura de los sensores
Fuente. Autoría*

4.2. Preparación del cultivo e instalación de nodos

Inicialmente se procede a la instalación del nodo de sensores para realizar la recolección de datos que posteriormente va a ser utilizada en el juego de entrenamiento de la red neuronal. Este nodo de recolección de información fue instalado en un lugar fuera del área de cultivo destinado para el sistema de riego. En la Figura 53 se puede observar el nodo de sensores instalado.



Figura 53. *Instalación del nodo de sensores (Recolección de datos)*
Fuente. Autoría

Luego de culminar con la etapa de recolección de datos se procede a arreglar el lugar donde se va a hacer la instalación del sistema de riego, se definió el área de evaluación en un espacio del área de cultivo de alfalfa, donde es necesario existan ciertas condiciones: proporcionar seguridad física a los equipos para evitar daños por aspectos climáticos y tener acceso a la fuente de agua que es el principal recurso para este sistema de riego inteligente.

Con ayuda del técnico agricultor se procede a colocar la alfalfa en una parcela con dimensiones de 4.5 x 10 m (45m²); cuya distribución se conforma en 9 hileras. Además, se procede a instalar las mangueras para el riego por goteo. En la figura 54 se puede observar la parcela preparada con las mangueras instaladas.



Figura 54. Preparación de la parcela e instalación de mangueras
Fuente. Autoría

El nodo actuador se encuentra conectado a la manguera principal de distribución de recurso hídrico destinado para el riego en la zona del cultivo de alfalfa.

4.3. Análisis del cultivo

El principal punto de interés de esta prueba es evaluar que tan eficiente es el sistema de riego. Para eso se hace una comparación entre una porción de cultivo con riego tradicional y otra porción de cultivo con el sistema de riego planteado en esta investigación, para ello se analizan tres variables en el desarrollo de las plantas como son la altura de la planta, coloración de la hoja y frondosidad.

Para la realización de esta prueba se ha tomado dos porciones de terreno de 4.5m x 5 cada una, en la una porción se implementa un sistema de riego manual y en otra porción el sistema de riego controlado con redes neuronales artificiales.

4.3.1. Altura de la planta

En la Figura 55 sección a) se muestra la altura máxima que alcanzó la planta con el sistema de riego manual y en la sección b) se indica la altura máxima de la planta con el sistema de riego inteligente, observándose así la diferencia que existe en el desarrollo de una planta con la otra. La altura de la planta se mide desde el suelo hacia la punta de la misma.



Figura 55. Comparación en el tamaño de la planta usando a) riego automatizado y b) riego manual
Fuente. Autoría

Como se puede observar en la Figura 55, la planta con mayor tamaño pertenece a la del sistema de riego automatizado, lo que indica que con el sistema de riego propuesto en esta investigación se obtienen mejores resultados.

4.3.2. Coloración de la hoja

Para validar el motivo de porque se realiza el riego del cultivo durante la noche con el sistema de la presente investigación, se procedió a hacer el riego manual de las plantas durante el día, en la Figura 56, se puede observar que con el paso de los días las hojas de las plantas empezaron a tornarse de un color amarillento y en algunos casos la planta empezó a secarse, lo que resulta algo infructuoso para la comercialización de la alfalfa debido a que precisamente la parte consumible de esta planta es la hoja.



*Figura 56. Resultados de hacer el riego durante el día
Fuente. Autoría*

En cambio, con el sistema de riego automatizado se lleva a cabo el riego durante en la noche, en la Figura 57 se puede observar que la hoja tiene un color verde lo que hace que la planta sea más atractiva al momento de comercializarla.



*Figura 57. Resultados de hacer el riego durante la noche
Fuente. Autoría*

4.3.3. Frondosidad

Durante el periodo de pruebas, se puede observar que las plantas con el sistema de riego manual desarrollan menor cantidad de ramificaciones y con menor cantidad de hojas, en cambio las plantas con el sistema de riego automático desarrollan mayor cantidad de ramificaciones y poseen mayor cantidad de hojas en cada una de estas. En la Figura 58 se puede observar la comparación de la frondosidad que se logra usando el riego manual y el riego automático.



Figura 58. Frondosidad de la planta: a) con riego manual y b) con riego automático
Fuente. Autoría

4.3.4. Tamaño de la Raíz

Durante el periodo de pruebas, se pudo observar que las plantas con el sistema de riego manual tienen un menor tamaño en su raíz lo cual es motivo para que la planta adquiera un tamaño pequeño y con menor ramificaciones, en cambio con el sistema de riego automático se pudo verificar que la planta desarrolló una raíz más grande por lo cual la planta adquirió un mayor tamaño y con mayor cantidad de hojas, lo que también la hace más resistente a cambios bruscos en el ambiente, esto lo podemos observar en la Figura 59.



Figura 59. Tamaño de la raíz: a) con riego manual y b) con riego automático
Fuente. Autoría

4.3.5. Eficiencia en el uso del agua

En la Tabla 23 se muestran los aspectos que se observan en el primer día de pruebas desde la instalación de los dos sistemas de riego, por aspectos se tiene en referencia los cambios positivos y negativos que se pueden ver en este día, además se incluye el consumo de agua durante el primer día de pruebas.

Tabla 23. Consumo de agua en el primer día de pruebas

Tipo de riego	Tiempo de pruebas (día)	Cambios positivos	Cambios negativos	Consumo de agua (litros)
<i>Sistema inteligente</i>	1	- Hojas de alfalfa verdes -Suelo en condiciones aptas para el desarrollo de la planta -Plantas de alfalfa de tamaño mayor	-Parte lógica del sistema con pequeños fallos causados por nuevos valores ambientales. -Realizó 3 riegos en 24 horas	20
<i>Sistema manual</i>		-Se realizó un solo riego en 24 horas	- Plantas de alfalfas de menor tamaño -La humedad de suelo es menor que el área del sistema de riego automático -Se empieza a ver plantas con decoloración en la hoja -El suelo pierde humedad mucho más rápido	23

Fuente. Autoría

Como se puede observar en la Tabla 23 en el primer día de pruebas en cuanto a los aspectos positivos y negativos se logra ver que existe mayores ventajas de usar el riego automatizado frente al riego manual, aunque se presentan pequeñas dificultades en la RNA debido a datos ambientales nuevos, en cuanto al consumo de recursos hídricos se puede ver que hay un consumo mayor de agua utilizando el riego automatizado.

En la Tabla 24 se listan nuevamente aspectos positivos y negativos después de 7 días de pruebas a partir de la instalación de los sistemas de riego (manual y automático).

Tabla 24. Consumo de agua luego de 7 días de pruebas

Tipo de Riego	Tiempo de pruebas (día)	Cambios positivos	Cambios negativos	Consumo de agua (litros)
<i>Sistema Inteligente</i>	7	- Hojas de la alfalfa verdes -Suelo en condiciones aptas para el desarrollo de la planta -Plantas de alfalfa de tamaño mayor -Índice de mortalidad del 0%	-Realizó 3 riegos en 7 días	60
<i>Sistema Manual</i>		-Se realizó 4 riegos en 7 días	-Plantas de alfalfas de menor tamaño -La humedad de suelo es menor que el área del sistema de riego automático -Se empieza a ver plantas con decoloración en la hoja -El suelo pierde humedad mucho más rápido -Se empieza a ver un proceso en secado en algunas hojas de las plantas	92

Fuente. Autoría

Como se puede observar en la Tabla 24 a partir del 7 día de pruebas se empieza a ver un ahorro considerable de agua del sistema de riego automatizado con respecto al sistema de riego manual, además de los cambios positivos y negativos que se listan entre un sistema y el otro. El

ahorro que se empieza a notar en el uso de agua, se debe a que el sistema de riego se activa cuando verdaderamente se requiere de riego en el cultivo.

Mediante investigación bibliográfica, para la asignación del tiempo de riego del sistema se toma en cuenta los valores mencionados en (Salazar, 2019), en el cual para el porcentaje mínimo de humedad de suelo asignado al 40%, valor tomado como mínimo para iniciar el proceso de riego en esta investigación, se le asigna un tiempo de 1800 segundos que es equivalente a 30 minutos de riego.

4.4. Precisión del sistema de riego automático.

En este apartado se hace el cálculo de la precisión del sistema de riego, para ello, se toma en consideración la cantidad de aciertos y el total de pruebas realizadas, en donde el valor de precisión es igual a la división del número de aciertos sobre el número total de pruebas, como se indica en la Ecuación 11.

$$Precisión = \frac{\# \text{ de aciertos}}{\# \text{ total de pruebas}}$$

Ecuación 14. Cálculo de Precisión

Se toma como referencia un total de 30 pruebas de las cuales se obtuvo 26 aciertos y 4 desaciertos, a continuación, se reemplazan estos valores en la Ecuación 11.

$$Precisión = \frac{26}{30}$$
$$Precisión = 0.866$$

Como resultado se tiene un equivalente a 86.6% de fiabilidad del sistema, lo que equivale a una tasa de error del 14.4% de error, por lo cual el proyecto resulta de mucho beneficio en el riego del cultivo de alfalfa.

4.5. Costo del sistema

En este apartado se detalla el costo de todos los componentes que conforman este sistema de riego. Dentro de los componentes que son parte de este proyecto se incluyen la parte de hardware, software, infraestructura, y trabajo de ingeniería.

4.5.1. Costos de hardware

En la Tabla 25 se listan los elementos de hardware utilizados en el sistema de riego, con la cantidad y su respectivo costo.

Tabla 25. Costos de hardware

Equipos	Cantidad	Precio Unitario (USD)	Subtotal (USD)
- Raspberry Pi 3	1	60.00	60.00
- Arduino Uno	1	12.00	12.00
- Módulo LoRaWAN RN2483	2	90.00	180.00
- Antenas de 868MHz - 3dB	2	10.00	20.00
- Electroválvula solenoide de ½ pulgada	1	15.00	15.00
- MicroSD de 32GB	1	16.00	16.00
- Módulo Relé	1	3.50	3.50

- Elementos electrónicos varios	1	25.00	25.00
- Banco de baterías con panel solar	1	12.00	12.00
		TOTAL (USD):	343.50

4.5.2. Costo de infraestructura

En el presente apartado se presentan los costos de infraestructura entre los cuales se toman en cuenta la instalación de mangueras, goteros, y carcasas de los nodos de cada zona, como se detalla a continuación en la Tabla 26.

Tabla 26. Costos de infraestructura.

Infraestructura	Cantidad	Precio Unitario (USD)	Subtotal (USD)
- Rollo de manguera de ½ pulgada	1	31.85	31.85
- Uniones PVC en T de ½ pulgada	8	0.85	6.80
- Codo PVC de media pulgada	1	0.85	0.85
- Tapones de ½ pulgada para manguera	9	1.50	13.50
- Carcasas de los nodos para la zona 1 y zona 2	2	12.00	24.00
- Goteros	300	0.10	30.00
- Cables y extras	1	25.00	25.00
		TOTAL (USD):	132.00

4.5.3. Costo de ingeniería

En esta sección se toma en cuenta los honorarios de la persona encargada tanto del diseño como de la implementación del proyecto, el cual incluye un estudio de campo y documentación, en la Tabla 27 se detalla estos costos.

Tabla 27. Costos de ingeniería

Actividades	Cantidad	Precio Unitario (USD)	Subtotal (USD)
- Estudio de campo, diseño y revisión de la infraestructura del sistema	1	600.00	600.00
- Documentación	1	200.00	200.00
- Mantenimiento del sistema de riego	1	150.00	150.00
		TOTAL (USD):	950.00

4.5.4. Costo de software

En la Tabla 28 se detalla los costos de software, el cual posee un valor de 0.00 USD, esto se debe a que las plataformas de programación utilizadas en el desarrollo de esta investigación son OpenSource.

Tabla 28. Costos de software

Software	Cantidad	Precio Unitario (USD)	Subtotal (USD)
- IDE Arduino	1	0.00	0.00
- Python	1	0.00	0.00
- Sistema Operativo Ubuntu Mate	1	0.00	0.00
TOTAL (USD):			0.00

4.5.5. Costo de implementación del sistema general

Para calcular el costo total de la implementación del sistema de riego, se hace la sumatoria de los costos de hardware, software, infraestructura e ingeniería, los costos mencionados se detallan a continuación en la Tabla 29.

Tabla 29. Costo total de implementación

Descripción	Subtotal (USD)
- Costos de hardware	343.50
- Costos de software	0.00
- Costos de infraestructura	132.00
- Costos de ingeniería	950.00
TOTAL (USD):	1425.00

Como se puede observar en la Tabla 29 el costo de implementación del sistema de riego automatizado es de 14250.00 USD, el cual es un valor considerablemente grande si se desea expandir a áreas de terreno mucho más grandes, pero que sería una inversión válida debido a los beneficios que se logran obtener en la producción del cultivo. Cabe recalcar que en esta investigación no se hace la adquisición de sensores para crear la red WSN debido a que este proyecto complementa al trabajo de investigación de (Domínguez, 2020) que trata de una red de sensores inalámbricos basado en una red LPWAN con módulos de comunicación LoRaWAN.

4.6. Conclusiones

- Mediante la investigación de bibliografía se establecieron que parámetros influyen en el desarrollo de la alfalfa, y cuales tienen incidencia directamente con el proceso de riego del cultivo. Con la información recolectada de la revisión bibliográfica se establecieron los parámetros para la recolección de datos ambientales para el dataset de entrenamiento de la red neuronal.
- Con la investigación realizada sobre el funcionamiento de las redes neuronales artificiales se logró establecer el número de capas de la red neuronal, la cantidad de neuronas en cada una de ellas, es decir, se pudo establecer un modelo ideal para el desarrollo del sistema de riego.
- Durante el periodo de pruebas del sistema de riego basado en redes neuronales artificiales se vio necesario implementar dos tipos de escenarios de pruebas, con la finalidad de hacer una comparación al finalizar las pruebas y de esa manera visualizar los cambios presentados en el cultivo de alfalfa utilizando técnicas de riego tradicionales en comparación con el sistema de riego propuesto en esta investigación.
- Se logró demostrar que utilizar redes neuronales artificiales con aprendizaje supervisado en aplicaciones de riego inteligente es una opción muy válida y viable como sistema de control binario, aun cuando se tiene una baja densidad de datos para el entrenamiento de la red neuronal.
- Con la implementación del sistema de riego basado en redes neuronales artificiales se logró evidenciar grandes cambios positivos en cuanto al desarrollo del cultivo, ya que se pudo observar que las plantas alcanzaron un mayor tamaño, la cantidad

de hojas era más abundante y de un color verde mucho más atractivo que las plantas con el sistema de riego manual.

- Se logró cumplir con cada uno de los objetivos establecidos al inicio de esta investigación, se cuenta con un sistema de riego automatizado que clasifica de manera adecuada los patrones de datos que provienen de una red de sensores inalámbrica los cuales indican si es o no necesario realizar el riego en el cultivo.

4.7. Recomendaciones

- Es necesario tener un monitoreo constante sobre el abastecimiento del recurso hídrico al sistema para evitar que el funcionamiento del sistema se vea afectado por falta de agua, precisamente al momento de realizar el riego, ya que la falta del líquido ocasiona inoperatividad del sistema de riego con RNA.
- Se debe tener un constante monitoreo sobre el estado de las baterías ya que la falta de energía en cualquiera de los nodos también ocasionaría fallas en el funcionamiento del sistema de riego.
- Se debe realizar un estudio sobre las características de las plantas de muestra y el ambiente donde se va a implementar el sistema de riego y en base a al estudio hacer la recolección de datos para el dataset de entrenamiento, incluso si el cultivo se lleva a cabo en un ambiente controlado se puede incrementar las acciones a realizar por parte del sistema.
- Si se desea realizar un sistema similar se recomienda realizar una comparación entre algoritmos de aprendizaje automático de clasificación ya que pueden existir otros que realicen el proceso de clasificación de mejor manera.

Bibliografía

- Aguilar, S. (2019). *DISEÑO DE UN SISTEMA BASADO EN IoT PARA LA SUPERVISIÓN Y CONTROL DE ESTACIONES REMOTAS DE LA DIRECCIÓN DE HIDROGRAFÍA Y NAVEGACIÓN DE LA MARINA DE GUERRA DEL PERÚ*.
- Alvarez, L. (1 de Mayo de 2013). *Ingeniería Mecatrónica*. Obtenido de <http://elagmecatronica.blogspot.com/2013/05/electrovalvula-controlada-con-arduino.html>
- Amores, A. (2017). *RECONOCIMIENTO DE IMAGENES EN FRAMES DE VIDEO UTILIZANDO REDES NEURONALES* (ESPE). Retrieved from <https://repositorio.espe.edu.ec/bitstream/21000/13458/1/T-ESPE-057368.pdf>
- ATGC Grupo de Tecnologías Avanzadas en Computación . (31 de Agosto de 2007). Obtenido de <https://advancedtech.wordpress.com/2007/08/31/elementos-basicos-de-una-red-neuronal-artificial/>
- Barreto, F. (13 de Mayo de 2014). *IEEE 29148*:. Obtenido de <https://prezi.com/8mki1yjihkdp/ieee-29148/>
- Barrios, O. (2004). *Construcción de un Invernadero*.
- Bongiovanni, R., Mantovani, E., Best, S., & Rogel, Á. (2006). *Agricultura de Precisión: Integrando conocimientos para una agricultura moderna y sustentable*. Montevideo: Procisur/IICA.
- Caparrini, S. (23 de Septiembre de 2017). Obtenido de <http://www.cs.us.es/~fsancho/?e=75>
- Castellano, M. (Diciembre de 2009). Obtenido de http://dm.udc.es/matematicas/es/system/files/Tesis_Maria_Castellano.pdf
- Castillero, O. (2016). *Psicología y Mente*. Obtenido de <https://psicologiaymente.net/neurociencias/soma-neuronal>
- Conceptos Básicos sobre Redes Neuronales*. (2011). Obtenido de <http://grupo.us.es/gtocom/pid/pid10/RedesNeuronales.htm#estructuraANS>
- del Valle Hernández, L. (2017). *ProgramarFacil*. Obtenido de Introducción Arduino MKRFOX1200 SigFox y redes LPWAN: https://programarfacil.com/blog/arduino-blog/arduino-mkrfox1200-sigfox-lpwan/#Que_es_SigFox

- Díaz, J. (21 de enero de 2016). *PLACA ARDUINO UNO*. Obtenido de <http://www.iescamp.es/miarduino/2016/01/21/placa-arduino-uno/>
- Egli, P. R. (2016). *Overview of Emerging Technologies For Low Power Wide Area Networks In Internet of Things AND M2M Scenarios*. Retrieved from <https://www.google.com/patents/US20160345265>
- García, E., & Flego, F. (2015). *Agricultura de Precisión*. Buenos Aires.
- Gestal, M. (Agosto de 2013). *Dpto. Computación: Área de Tecnologías de la Información y las Comunicaciones*. Obtenido de <http://sabia.tic.udc.es/mgestal/cv/RNATutorial/TutorialRNA.pdf>
- Gracia, L. (24 de mayo de 2017). *BLOG DE SOFIA2 IOT PLATFORM*. Obtenido de SIGFOX VS LORA: COMPARANDO REDES LPWAN: <https://about.sofia2.com/2017/05/24/sigfox-vs-lora/>
- González, M. (2018). *DISEÑO E IMPLEMENTACIÓN DE UN PROTOCOLO DE COMUNICACIONES ENTRE SENSORES IOT Y UNA PLATAFORMA CLOUD*.
- Huertas, L. (Marzo de 2008). *Horticom*. Obtenido de <http://www.horticom.com/pd/imagenes/69/757/69757.pdf>
- Ibañez, J. (26 de Junio de 2006). *Madrid Blogs*. Obtenido de <http://www.madrimasd.org/blogs/universo/2006/06/26/33002>
- Iglesias, N. (2009). *INTA*. Obtenido de <http://inta.gob.ar/sites/default/files/script-tmp>
- INCIBE. (8 de febrero de 2018). *S.M.E. INSTITUTO NACIONAL DE CIBERSEGURIDAD DE ESPAÑA M.P. SA*. Obtenido de Introducción a los sistemas embebidos: <https://www.incibe-cert.es/blog/introduccion-los-sistemas-embebidos>
- InfoAgro. (2010). *El cultivo de la Alfalfa*. Obtenido de <https://www.infoagro.com/herbaceos/forrajes/alfalfa.htm>
- ISO/IEC/IEEE. (1 de Diciembre de 2011). *29148-2011 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering*. Obtenido de <https://standards.ieee.org/standard/29148-2011.html>
- ITSA - Sistemas Agrícolas*. (2016). Obtenido de <http://itsamexico.com/Servicios/RiegoAgricola>
- IX23 Explore*. (6 de Febrero de 2018). Obtenido de <http://ix23.com/arduino-uno-r3-pinouts-chart/>

- Khan Academy. (2016). Obtenido de <https://es.khanacademy.org/science/biology/human-biology/neuron-nervous-system/a/overview-of-neuron-structure-and-function>
- Lara, F. (2014). *Fundamentos de Redes Neuronales Artificiales*. Obtenido de http://conceptos.sociales.unam.mx/conceptos_final/598trabajo.pdf
- Latré, S., Weyn, M., & Lannoo, B. (2014). Sub-GHz LPWAN network coexistence, management and virtualization: an overview and open research challenges. *University of Antwerp*, 29.
- Leon, R. (2002). Producción y manejo de pastos y forrajes. Editorial Universitaria.
- Lexikon, R. (2005). *Medizin*. Urban & Fisher.
- Liotta, M., Carrión, R., Ciancaglini, N., & Olguín, A. (2015). *Manual de Capacitación - Riego por goteo*. Rivadavia: Ediciones para UCAR. .
- Llanos, D., & Romero, M. (4 de Junio de 2013). *Aprendizaje Supervisado y no Supervisado*. Obtenido de <http://redesneuronal.blogspot.com/>
- López, D. (2007). *DISEÑO Y CONTRUCCIÓN DE UNA RED NEURONAL ARTIFICIAL DE PROPÓSITO GENERAL*. Quito.
- López, J. (03 de Agosto de 2000). Obtenido de http://members.tripod.com/jesus_alfonso_lopez/RnaIntro.html
- Martínez, A. (3 de Agosto de 2017). *twenergy*. Obtenido de <https://twenergy.com/a/sistemas-de-riegos-para-momentos-de-necesidad-de-ahorro-de-agua-2709>
- Masters, T. (1993). *Practical Neural Network Recipes in C++*. San Francisco, California: Morgan Kaufmann.
- MathWorks. (2017). © 1994-2019 The MathWorks, Inc. Obtenido de <https://es.mathworks.com/discovery/aprendizaje-supervisado.html>
- Matich, D. J. (2002). Redes Neuronales: Conceptos Básicos y Aplicaciones. *Cátedra: Informática Aplicada a La Ingeniería de Procesos-Orientación I*, 6. Retrieved from https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograis/matich-redesneuronales.pdf
- Mayobre, R. (2018). *Telemetría de contenedores de residuos*.
- McClelland, C. (29 de diciembre de 2016). *IoT for all*. Obtenido de Comparación de LPWAN Technologies - ¿Cuál es mejor para mí?: <https://www.iotforall.com/comparison-of-lpwan-technologies/>

- MICROCHIP. (2016). *RN2483*. Obtenido de <https://www.microchip.com/wwwproducts/en/RN2483>
- Microchip Technology Inc. (11 de Julio de 2016). Obtenido de <http://ww1.microchip.com/downloads/en/devicedoc/50002346c.pdf>
- Ministerio de Agricultura y Ganadería. (2017). *Noticias: Ministerio de Agricultura y Ganadería*. Obtenido de <https://www.agricultura.gob.ec/zapotillo-94-familias-beneficiadas-con-micro-reservorios/>
- Ministerio de Agricultura y Riego. (marzo de 2016). *Proyecto Subsectorial de Irrigación*. Obtenido de http://www.psi.gob.pe/wp-content/uploads/2016/03/biblioteca_hojas_riego_multicompuertas.pdf
- Mónaco, A. (17 de Agosto de 2014). *Peatonet*. Obtenido de <http://www.peatonet.com/raspberry-pi-y-los-pines-gpio-que-son-y-que-usos-practicos-les-podemos-dar/>
- Pacheco, A. (7 de Junio de 2010). *ACEA: Invernaderos para el mundo*. Obtenido de <http://acea.com.mx/articulos-tecnicos/alex-j-pacheco/60-los-factores-ambientales-y-su-influencia-en-invernaderos-313-la-luz-dentro-del-invernadero>
- Paessler. (Septiembre de 2018). *IT Explained: LPWA*. Obtenido de <https://www.es.paessler.com/it-explained/lpwa>
- Palacios, F. (Abril de 2003). *Herramientas en GNU/LINUX para estudiantes universitarios*. Obtenido de https://www.ibiblio.org/pub/linux/docs/LuCaS/Presentaciones/200304curso-glisa/redes_neuronales/curso-glisa-redes_neuronales-html/x38.html
- PC-FACTORY. (2015). Obtenido de PCF OEM® Electro valvula 12V 1/2" para Estanque: <https://www.pcfactory.cl/producto/27390-pcf-oem-electro-valvula-12v-1-2-para-estanque#parentHorizontalTab1>
- Peery, J. (12 de Septiembre de 2017). *PROMIX*. Obtenido de <https://www.pthorticulture.com/es/centro-de-formacion/como-influye-la-humedad-en-la-calidad-de-los-cultivos/>
- Pérez García, R. (2017). *Evaluación de LoRa/LoRaWAN para escenarios de Smart City*. Retrieved from <https://upcommons.upc.edu/handle/2117/100922>

- Pérez García, A. (2013). *GAMIFICACIÓN: ENCUESTAS DE SATISFACCIÓN DE LOS SERVICIOS DE ALIMENTACIÓN JAVERIANOS*:. Obtenido de <https://repository.javeriana.edu.co/handle/10554/12643>
- Perez, A., Berreteaga, O., Ruiz de Olano, A., Urkidi, A., & Perez, J. (2006). *UNA METODOLOGÍA PARA EL DESARROLLO DE HARDWARE Y SOFTWARE EMBEBIDOS EN SISTEMAS CRITICOS DE SEGURIDAD*:. Obtenido de [http://www.iiisci.org/journal/CV\\$/risici/pdfs/C863GM.pdf](http://www.iiisci.org/journal/CV$/risici/pdfs/C863GM.pdf)
- Pérez, R. (2017). *Evaluación de LoRa/LoRaWAN para escenarios de Smart City*. Catalunya. *Pevifer*. (Julio de 2009). Obtenido de <https://www.verpueblos.com/castilla+y+leon/leon/santa+olaja+de+porma/foto/450520/>
- Pickering, P. (29 de Junio de 2017). *Digi-Key*. Obtenido de <https://www.digikkey.es/es/articles/techzone/2017/jun/develop-lora-for-low-rate-long-range-iot-applications>
- Quindiaguas. (s.f.). *Quindiaguas*. Obtenido de <http://quindiaguas.com/contenido-index-id-22.htm>
- Recuero de los Santos, P. (16 de noviembre de 2017). *Los 2 tipos de aprendizaje en Machine Learning: supervisado y no supervisado*. Obtenido de <https://empresas.blogthinkbig.com/que-algoritmo-elegir-en-ml-aprendizaje/>
- Rojas, F., Lezcano, M., & Medina, F. (2016). *Agricultura de precisión con sensores inalámbricos*. Santa Cruz de la Sierra.
- Sáenz, E. (2018). *Estudio de los parámetros principales de la tecnología LoRa para el despliegue de redes e implementación de servicios IoT en la CAV-EAE*.
- Salas, R. (2005). *Redes Neuronales Artificiales-Rodrigo Salas*. 7. Retrieved from https://s3.amazonaws.com/academia.edu.documents/37429671/Redes_Neuronales_Artificiales.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1556258957&Signature=xSYptS80OJgE%2B4aLrnXZNtCa%2F1U%3D&response-content-disposition=inline%3B filename%3DRedes_Neuronal
- Saldarriaga, V. (12 de febrero de 2012). *JARDINERIA ON*. Obtenido de <https://www.jardineriaon.com/temperatura-en-los-invernaderos.html>
- Saura, O. (20 de Febrero de 2012). *Rincondemente*. Obtenido de <https://rincondemente.wordpress.com/2012/02/20/cuerpo-celular-o-soma/>

- Seminis*. (18 de Julio de 2016). Obtenido de <http://www.seminis.mx/blog-los-principales-sistemas-de-riego/>
- Semtech. (2019). *SEMTECH SX1272/73 - 860 MHz to 1020 MHz Low Power Long Range Transceiver*. (January).
- SIRHC Hidráulica y Construcciones*. (2016). Obtenido de Sistema de Riego por Micro-compuertas: <http://www.hidraulicayconstrucciones.com.mx/compuerta.php>
- Solomatine, D. (2002). *Data-driven modelling: machine learning and data mining in water related*.
- Sucar, E. (2018). *Instituto Nacional de Astrofísica Óptica y Electrónica*. Obtenido de INAOE: <https://ccc.inaoep.mx/~esucar/Clases-MetIA/MetIA-16.pdf>
- Triglia, A. (2016). *Psicología y Mente*. Obtenido de <https://psicologiaymente.net/neurociencias/axones>
- Zambrano, J. (30 de marzo de 2018). *Medium Corporation* . Obtenido de ¿Aprendizaje supervisado o no supervisado? Conoce sus diferencias dentro del machine learning y la automatización inteligente: <https://medium.com/@juanzambrano/aprendizaje-supervisado-o-no-supervisado-39ccf1fd6e7b>

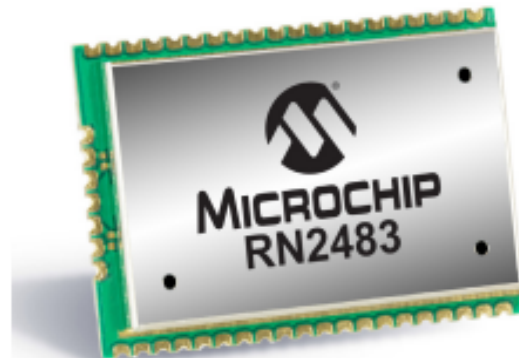


RN2483

Low-Power Long Range LoRa® Technology Transceiver Module

General Features

- On-Board LoRaWAN™ Protocol Stack
- ASCII Command Interface over UART
- Compact Form Factor: 17.8 x 26.7 x 3.34 mm
- Castellated SMT Pads for Easy and Reliable PCB Mounting
- Environmentally Friendly, RoHS Compliant
- European R&TTE Directive Assessed Radio Module
- Device Firmware Upgrade (DFU) over UART, see "RN2483 LoRa® Technology Module Command Reference User's Guide" (DS40001784)



Operational

- Single Operating Voltage: 2.1V to 3.6V (3.3V typical)
- Temperature Range: -40°C to +85°C
- Low-Power Consumption
- Programmable RF Communication Bit Rate up to 300 kbps with FSK Modulation, 10937 bps with LoRa® Technology Modulation
- Integrated MCU, Crystal, EU1-64 Node Identity Serial EEPROM, Radio Transceiver with Analog Front End, Matching Circuitry
- 14 GPIOs for Control and Status, Shared with 13 Analog Inputs

RF/Analog Features

- Low-Power Long Range Transceiver Operating in the 433 MHz and 868 MHz Frequency Bands
- High Receiver Sensitivity: Down to -146 dBm
- TX Power: Adjustable up to +14 dBm high Efficiency PA
- FSK, GFSK, and LoRa Technology Modulation
- IIP3 = -11 dBm
- Up to 15 km Coverage at Suburban and up to 5 km Coverage at Urban Area

Description

Microchip's RN2483 Low-Power Long Range LoRa Technology Transceiver module provides an easy to use, low-power solution for long range wireless data transmission. The advanced command interface offers rapid time to market.

The RN2483 module complies with the LoRaWAN Class A protocol specifications. It integrates RF, a baseband controller, command Application Programming Interface (API) processor, making it a complete long range solution.

The RN2483 module is suitable for simple long range sensor applications with external host MCU.

Applications

- Automated Meter Reading
- Home and Building Automation
- Wireless Alarm and Security Systems
- Industrial Monitoring and Control
- Machine to Machine (M2M)
- Internet of Things (IoT)

1.0 DEVICE OVERVIEW

The RN2483 transceiver module features LoRa Technology RF modulation, which provides long range spread spectrum communication with high interference immunity.

Using LoRa Technology modulation technique, RN2483 can achieve a receiver sensitivity of -146 dBm. The high sensitivity combined with the integrated +14 dBm power amplifier yields industry leading link budget, which makes it optimal for applications requiring extended range and robustness.

LoRa Technology modulation also provides significant advantages in both blocking and selectivity compared to the conventional modulation techniques, solving the traditional design compromise between extended range, interference immunity, and low-power consumption.

The RN2483 module delivers exceptional phase noise, selectivity, receiver linearity, and IIP3 for significantly lower power consumption. Figure 1-1, Figure 1-2, and Figure 1-3 show the top view, the pinout, and the block diagram of the module.

FIGURE 1-1: RN2483 TOP VIEW

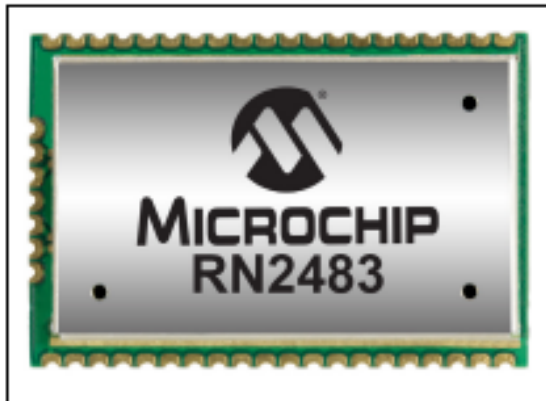


FIGURE 1-2: RN2483 PIN DIAGRAM

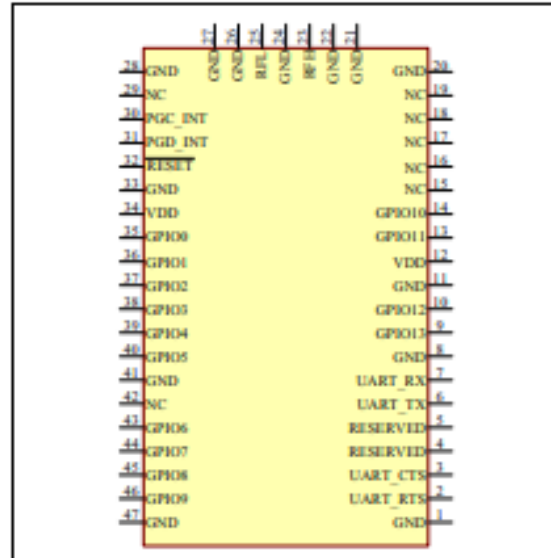


FIGURE 1-3: RN2483 BLOCK DIAGRAM

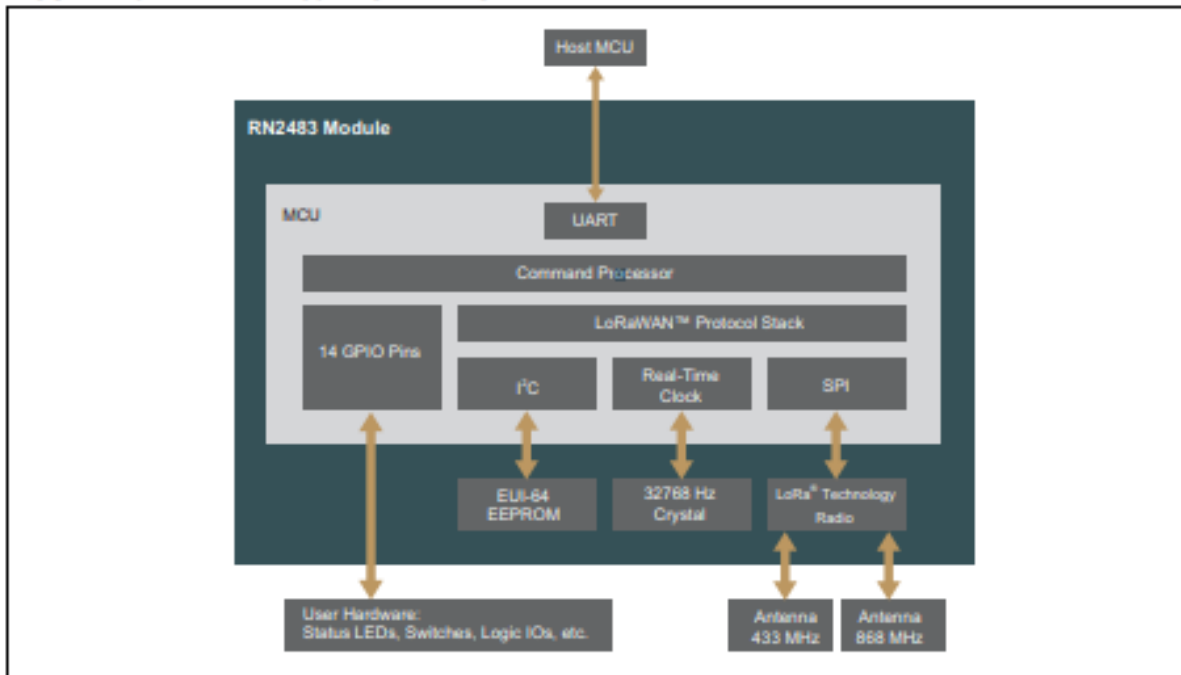


Table 1-1 describes the RN2483 pins.

TABLE 1-1: PIN DESCRIPTION

Pin	Name	Type	Description
1	GND	Power	Ground supply terminal
2	UART_RTS	Output	Communication UART RTS signal ⁽¹⁾ , or GPIO
3	UART_CTS	Input	Communication UART CTS signal ⁽¹⁾ , or GPIO
4	RESERVED	—	Do not connect
5	RESERVED	—	Do not connect
6	UART_TX	Output	Communication UART Transmit (TX)
7	UART_RX	Input	Communication UART Receive (RX)
8	GND	Power	Ground supply terminal
9	GPIO13	Input/Output	General purpose I/O pin or analog input
10	GPIO12	Input/Output	General purpose I/O pin or analog input
11	GND	Power	Ground supply terminal
12	VDD	Power	Positive supply terminal
13	GPIO11	Input/Output	General purpose I/O pin or analog input
14	GPIO10	Input/Output	General purpose I/O pin or analog input
15	NC	—	Not connected
16	NC	—	Not connected
17	NC	—	Not connected
18	NC	—	Not connected
19	NC	—	Not connected
20	GND	Power	Ground supply terminal
21	GND	Power	Ground supply terminal
22	GND	Power	Ground supply terminal
23	RFH	RF analog	RF signal pin for high band
24	GND	Power	Ground supply terminal
25	RFL	RF analog	RF signal pin for low band
26	GND	Power	Ground supply terminal
27	GND	Power	Ground supply terminal
28	GND	Power	Ground supply terminal
29	NC	—	Not connected
30	PGC_INT	Input/Output	Internal MCU ICSP program clock or general purpose I/O pin ⁽²⁾
31	PGD_INT	Input/Output	Internal MCU ICSP program data or general purpose I/O pin ⁽²⁾
32	RESET	Input	Active-low device Reset input
33	GND	Power	Ground supply terminal
34	VDD	Power	Positive supply terminal
35	GPIO0	Input/Output	General purpose I/O pin or analog input
36	GPIO1	Input/Output	General purpose I/O pin or analog input
37	GPIO2	Input/Output	General purpose I/O pin or analog input
38	GPIO3	Input/Output	General purpose I/O pin or analog input
39	GPIO4	Input/Output	General purpose I/O pin
40	GPIO5	Input/Output	General purpose I/O pin or analog input
41	GND	Power	Ground supply terminal
42	NC	—	Not connected

2.0 GENERAL SPECIFICATIONS

Table 2-1 provides the general specifications for the module. Table 2-2, Table 2-3, and Table 2-4 provide the electrical characteristics, current consumption, and

dimensions of the module, respectively. Table 2-5 shows the RF output power calibration data. Table 2-6 shows the RF output power at different supply voltages and temperatures.

TABLE 2-1: GENERAL SPECIFICATIONS

Specification	Description
Frequency Band	863.000 MHz to 870.000 MHz; 433.050 MHz to 434.790 MHz
Modulation Method	FSK, GFSK, and LoRa [®] Technology modulation
Maximum Over-the-Air Data Rate	300 kbps with FSK modulation; 10937 bps with LoRa Technology modulation
RF Connection	Board edge connection
Interface	UART
Operation Range	Up to 15 km coverage at suburban; up to 5 km coverage at urban area
Sensitivity at 1% PER	-146 dBm ⁽¹⁾
RF TX Power	Adjustable up to max. 10 dBm on 433 MHz band (limited to meet regulations); max. 14 dBm on the 868 MHz band ⁽²⁾
Temperature (operating)	-40°C to +85°C
Temperature (storage)	-40°C to +115°C
Humidity	10% ~ 90% non-condensing

Note 1: Dependent on modulation settings, Receiver Bandwidth (RBW), and Spreading Factor (SF).

2: TX power is adjustable. For more information, refer to the "RN2483 LoRa[®] Technology Module Command Reference User's Guide" (DS40001784).

TABLE 2-2: ELECTRICAL CHARACTERISTICS

Parameter	Min.	Typ.	Max.	Units
Supply Voltage	2.1	—	3.6	V
Voltage on any pin with respect to VSS (except VDD) and RESET	-0.3	—	VDD + 0.3	V
Voltage on VDD with respect to VSS	-0.3	—	3.9	V
Voltage on RESET with respect to VSS	0	—	+11	V
Input Clamp Current (I _{IK}) (V _I < 0 or V _I > VDD)	—	—	+/-20	mA
Output Clamp Current (I _{OK}) (V _O < 0 or V _O > VDD)	—	—	+/-20	mA
GPIO sink/source current each	—	—	25/25	mA
Total GPIO sink/source current	—	—	200/185	mA
RAM Data Retention Voltage (in Sleep mode or Reset state)	1.5	—	—	V
VDD Start Voltage to ensure internal Power-on Reset signal	—	—	0.7	V
VDD Rise Rate to ensure internal Power-on Reset signal	0.05	—	—	V/ms
Brown-out Reset Voltage	1.75	1.9	2.05	V
Logic Input Low Voltage	—	—	0.15 x VDD	V
Logic Input High Voltage	0.8 x VDD	—	—	V
Input Leakage at <25°C (VSS < V _{PIN} < VDD, Pin at high-impedance)	—	0.1	50	nA
Input Leakage at +60°C (VSS < V _{PIN} < VDD, Pin at high-impedance)	—	0.7	100	nA
Input Leakage at +85°C (VSS < V _{PIN} < VDD, Pin at high-impedance)	—	4	200	nA
RF Input Level	—	—	+10	dBm

TABLE 2-3: CURRENT CONSUMPTION

Mode	Temperature (°C)	Typical Current (mA)		
		VDD = 2.1V	VDD = 3.3V	VDD = 3.6V
Idle	-40 to +85	1.7	2.8	3.1
Transmit	25	28.6	38.9	44.5
Sleep	-40	0.0011	0.0013	0.0014
	25	0.0015	0.0016	0.0016
	85	0.002	0.0026	0.0026

TABLE 2-4: MODULE DIMENSIONS

Parameter	Value
Dimensions	17.8 x 26.7 x 3.34 mm
Weight	2.05g

TABLE 2-5: OUTPUT POWER OF TX POWER SETTING

Band	TX Power Setting	Output Power (dBm)	Typical Supply Current at 3.3V (mA)
868 MHz	-3	-4.0	17.3
	-2	-2.9	18.0
	-1	-1.9	18.7
	0	-1.7	20.2
	1	-0.6	21.2
	2	0.4	22.3
	3	1.4	23.5
	4	2.5	24.7
	5	3.6	26.1
	6	4.7	27.5
	7	5.8	28.8
	8	6.9	30.0
	9	8.1	31.2
	10	9.3	32.4
	11	10.4	33.7
12	11.6	35.1	
13	12.5	36.5	
14	13.5	38.0	
15	14.1	38.9	

TABLE 2-5: OUTPUT POWER OF TX POWER SETTING (CONTINUED)

Band	TX Power Setting	Output Power (dBm)	Typical Supply Current at 3.3V (mA)
433 MHz	-3	-3.5	14.7
	-2	-2.3	15.1
	-1	-1.3	15.6
	0	-2.3	15.8
	1	-1.2	16.4
	2	-0.1	17.0
	3	1.0	17.7
	4	2.1	18.5
	5	3.2	19.4
	6	4.3	20.3
	7	5.4	21.4
	8	6.5	22.3
	9	7.6	23.3
	10	8.8	24.5
	11	9.9	25.8
	12	10.9	27.3
13	11.9	28.8	
14	12.9	30.7	
15	13.6	32.9	

TABLE 2-6: OUTPUT POWER OF SUPPLY VOLTAGE AND TEMPERATURE

Temperature (°C)	Typical Output Power at 868 MHz (dBm)		
	VDD = 2.1V	VDD = 3.3V	VDD = 3.6V
-40	10.5	13.8	13.7
25	10.0	14.1	14.6
85	9.1	13.4	13.7

TABLE 2-7: OUTPUT POWER OF SUPPLY VOLTAGE AND TEMPERATURE

Temperature (°C)	Typical Output Power at 434 MHz (dBm)		
	VDD = 2.1V	VDD = 3.3V	VDD = 3.6V
-40	10.1	13.2	13.2
25	9.7	13.6	14.2
85	9.3	13.0	13.4

4.0 PHYSICAL DIMENSIONS

Figure 4-1 and Figure 4-2 illustrate the physical dimensions and the recommended PCB layout for the RN2483 module.

FIGURE 4-1: RN2483 PHYSICAL DIMENSIONS

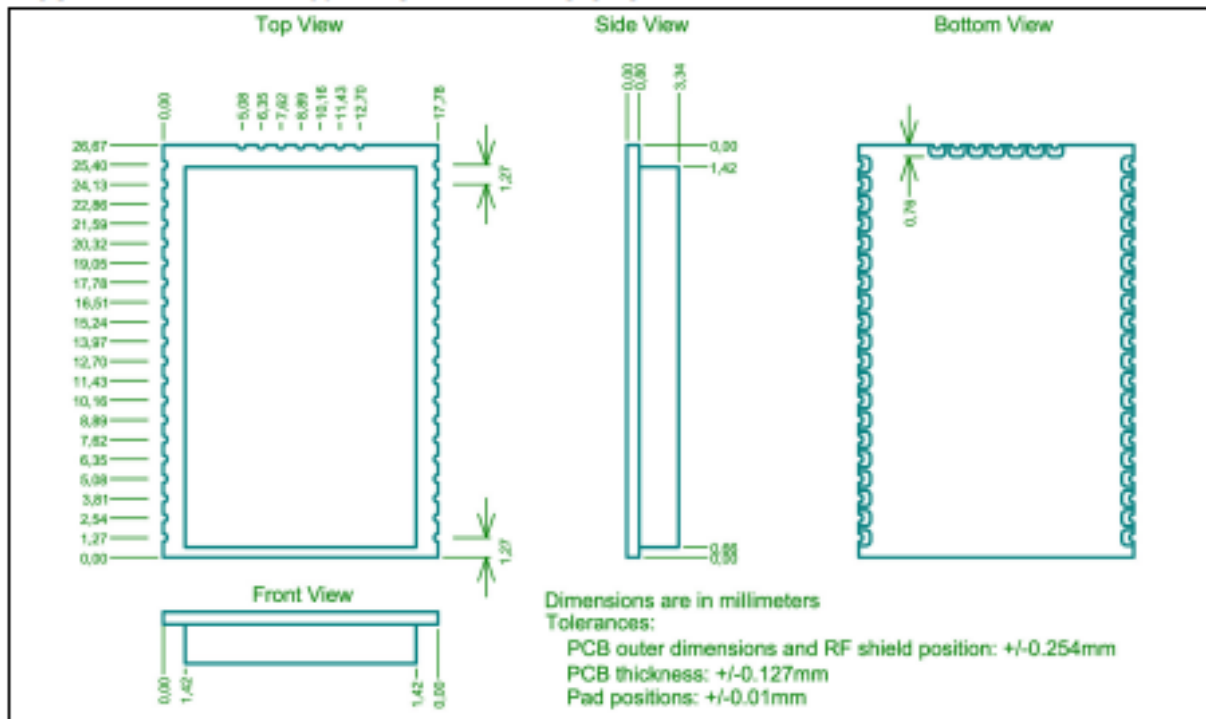
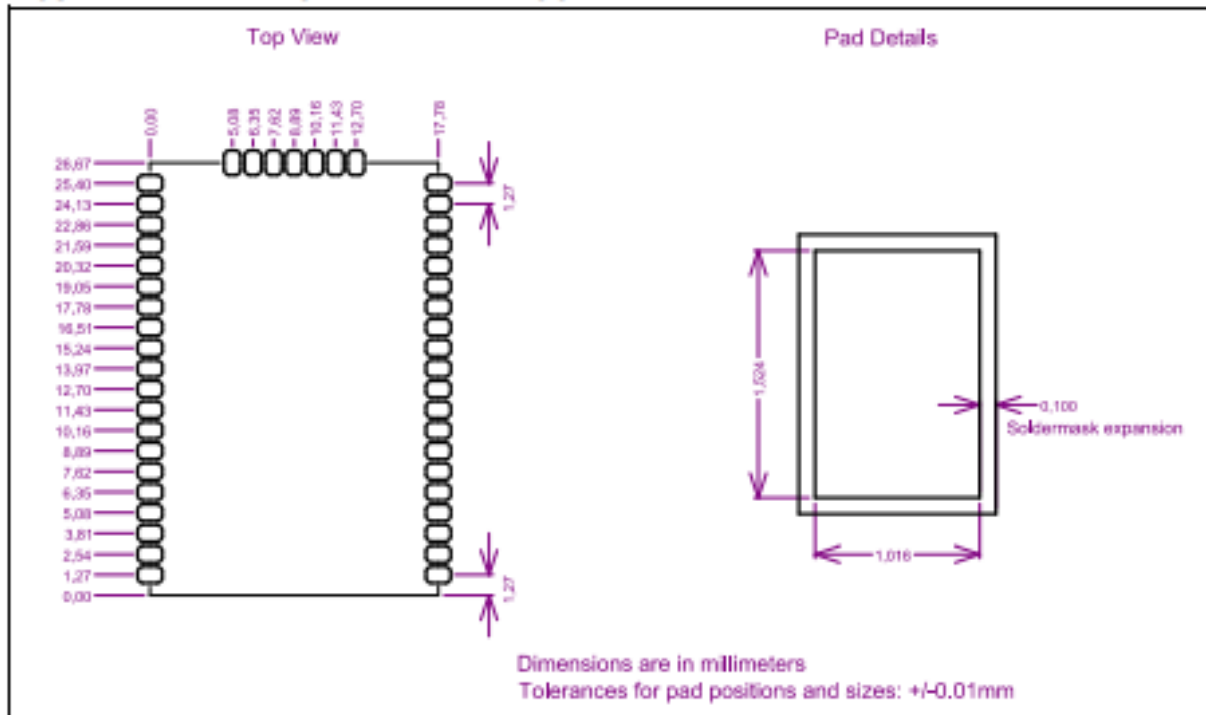


FIGURE 4-2: RECOMMENDED PCB FOOTPRINT



ANEXO 2: Código de recolección de datos para entrenamiento de la RNA. (Zona 1)

```
#include <SoftwareSerial.h> //inicialización de comunicación serial
#include <Wire.h>           //inicialización de comunicación I2C
#include "DHT.h"           //librería sensor DHT
#define DHTPIN 8           //pin de lectura de sensor DHT
#define DHTTYPE DHT11     //tipo de sensor DHT puede ser 11, 21 o 22
DHT dht(DHTPIN, DHTTYPE);

SoftwareSerial loranode(10, 11); //Pines para comunicación serial con el módulo
RN2483

String str; //Variable para comunicación con módulo LoRa
int lux = 0; //LUMINOSIDAD
int temperatura = 0; //TEMPERATURA AMBIENTE
int humedad = 0; //HUMEDAD RELATIVA
int capacitive = 0; // HUMEDAD DE SUELO

void setup() {
  Serial.begin(9600); //Velocidad de Tx de datos por puerto serial
  Arduino
  loranode.begin(57600); //Velocidad de Tx de datos por puerto serial (Arduino -
  LoRaWAN)

  Wire.begin();
  dht.begin(); //Inicializa sensor DHT11

  while (!loranode); //Inicializa al módulo RN2483
  pinMode(3,OUTPUT); // Puerto para led indicador
  digitalWrite(3,HIGH); // Enciende el led indicador

  //Resetea el módulo RN2483
  pinMode(7,OUTPUT); //Puerto para el pin Reset del módulo RN2483
  digitalWrite(7,HIGH); //Puerto encendido
  delay(500);
  digitalWrite(7,LOW); //Puerto apagado
  delay(2000);

  //Configuración de parámetros del módulo RN2483
  loranode.setTimeout(100); //establece 100 milisegundos para esperar datos por
  puerto serial
  str = loranode.readStringUntil('\n');
  Serial.println(str);
  loranode.println("radio set wdt 0"); //Tiempo de espera de rx en ms
  str = loranode.readStringUntil('\n');
  Serial.println(str);
  loranode.println("radio set pwr 14"); //Asignación de potencia de Tx del módulo
  RN2483
  str = loranode.readStringUntil('\n');
  Serial.println(str);
  loranode.println("radio set sf sf12"); //Asignación del factor de propagación usado
  durante la Tx
  str = loranode.readStringUntil('\n');
  Serial.println(str);
```

```

loranode.println("mac pause");          //Permite cambiar al módulo de Rx a Tx y
viceversa
str = loranode.readStringUntil('\n');
Serial.println(str);
digitalWrite(3,LOW);    //Apaga el Led indicador
}
void loop() {

loranode.println("radio rx 0");    //pone al módulo en modo receptor
str = loranode.readStringUntil('\n');    //lee la respuesta del módulo por puerto
serial
Serial.println(str);    //imprime la respuesta del módulo
if ( str.indexOf("ok") == 0 ) {    //evalúa si el módulo cambió a modo de recepción
int ok=0;    //a la respuesta del módulo se le asigna un valor de 0
while ( ok == 0 ) {
str = loranode.readStringUntil('\n');
if ( str.length() > 1 ) {    // evalúa si se recibe algún dato (petición)
if ( str.indexOf("radio_rx") >= 0 ) {    // evalúa si sigue en modo de
recepción
if ( str.indexOf("111111111111") >= 0 ) {    // evalúa si la etiqueta de
petición le corresponde a este nodo
Serial.println("Conectandose a la Red LoRa");
Serial.println("Petición de datos Recibida");
Serial.println("Recolectando información...");
lux = analogRead(4);    // Mide nivel de luminosidad (valor analógico)
capacitive = analogRead(0);    // Mide el nivel de humedad de suelo
(valor analógico)
humedad = dht.readHumidity(); // Se lee la humedad relativa
temperatura= dht.readTemperature(); // Se lee la temperatura ambiente
ph = analogRead(2);    //Se lee nivel de PH en el suelo
delay(1000);    //Tiempo de espera de 1 segundo

String cadena1 = String(lux, DEC);    // a la variable de luminosidad
se le convierte a tipo string
String cadena2 = String(capacitive, DEC);    // a la variable de la
humedad del suelo se le convierte a tipo string
String cadena3 = String(humedad, DEC);    // a la variable de la
humedad relativa se le convierte a tipo string
String cadena4 = String(temperatura, DEC);    // a la variable de
temperatura se le convierte a tipo string

//se iguala el tamaño de las variables a 4 caracteres para su posterior envío para
ello se completa con ceros
if(lux <= 0) cadena1 = "0000";
if(lux >=1 && lux <= 99){cadena1 = "00" + cadena1;}
if(lux >=100 && lux <= 999){cadena1 = "0" + cadena1;}

if(capacitive <= 0) cadena2 = "0000";
if(capacitive >=1 && capacitive <= 99){cadena2 = "00" + cadena2;}
if(capacitive >=100 && capacitive <= 999){cadena2 = "0" + cadena2;}

if(humedad <= 0) cadena3 = "0000";
if(humedad >=1 && humedad <= 99){cadena3 = "00" + cadena3;}
if(humedad >=100 && humedad <= 999){cadena3 = "0" + cadena3;}

```


1

```
lora.write(b'radio tx 111111111111\r\n') #Envía la petición de datos al Nodo

print("SOLICITUD AL RN1")
print(str(lora.readline())) #Imprime la respuesta del módulo RN2483
print(str(lora.readline()))

time.sleep(1)

print("CAMBIO A Rx")
lora.write(b'radio rx 0\r\n') #Cambio a modo de Rx
if(str(lora.readline()) != "busy"): #Evalúa que el módulo este libre
    a= str(lora.readline()) #Variable del paquete recibido
    #Separación bit a bit de los datos recibidos
    A1=a[10]+a[11]+a[12]+a[13] #Variable para el sensor de luminosidad
    A2=a[14]+a[15]+a[16]+a[17] #Variable para el sensor de hum. suelo
    A3=a[18]+a[19]+a[20]+a[21] #Variable para el sensor de hum. Rel.
    A4=a[22]+a[23]+a[24]+a[25] #Variable para el sensor de temperatura

    #Conversión de valores analógicos a digitales de los sensores
    sensor1=(A1) # Sensor de luminosidad
    s1=float(sensor1) #Convierte de string a tipo float
    SenLx=((s1-1023)*(100-1)/(1-1023)+1)
    SenLx("%.1f" % SenLx) #Mostrar un solo decimal

    sensor2=(A2) # Sensor de humedad de suelo
    s2=float(sensor2) #Convierte de string a tipo float
    SenHum=((s2-1023)*(100-1)/(1-1023)+1)
    SenHum("%.1f" % SenHum) #Mostrar un solo decimal

    sensor3=(A3) # Sensor de Humedad Relativa
    s3=float(sensor3) #Convierte de string a tipo float
    SenHR=(s3)
    SenHR("%.1f" % SenHR) #Mostrar un solo decimal

    sensor4=(A4) # Sensor de Temperatura
    s4=float(sensor4) #Convierte de string a tipo float
    SenTem=(s4)
    SenTem("%.1f" % SenTem) #Mostrar un solo decimal

    #Imprime los valores digitales calculados
    print("LECTURAS VARIABLE SENSORES")
    print(SenLx)
    print(SenHum)
    print(SenHR)
    print(SenTem)

    $sentencia para asignar las etiquetas de salida para la RNA
    if (SenLx <= 30.0 && SenHum <= 40.0 && SenTem <= 14.0)
        riego = 1
    else:
        riego = 0

    #ALMACENAMIENTO EN LA BASE DE DATOS DEL SERVIDOR LOCAL
    #Se crea una conexión con las credenciales de acceso
```



```

connection
mysql.connector.connect(user='tesis',password='123tesis',database='Tesis')
    cursor = connection.cursor() #Se crea un cursor para la conexión

    sql = """INSERT INTO Dataset(tempA,humA,humS,lum,riego) VALUES
(%s,%s,%s,%s)""" #Accede a la tabla donde se va a almacenar los datos
    datos = (SenTem,SenHR,SenHum,SenLx,riego) #Datos a guardar
    cursor.execute(sql, datos) #se ejecuta el cursor
    cursor.close() #Se cierra el cursor
    connection.close() #se cierra la conexión a MySQL
    time.sleep(300)

if __name__ == '__main__':
    set_modulo()
    while (True): #Ciclo While
        nodo_a() #Se ejecuta la subrutina nodo_a encargada de la petición de datos
        time.sleep(1)

```

ANEXO 4: Creación de tablas para el almacenamiento de datos de entrenamiento de la RNA.

```
CREATE TABLE IF NOT EXISTS `Dataset` (  
  `Fecha` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
  `tempA` float NOT NULL,  
  `humA` float NOT NULL,  
  `humS` float NOT NULL,  
  `lum` float NOT NULL,  
  `riego` int NOT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

ANEXO 5: Creación, entrenamiento, análisis y guardado de la red neuronal artificial.

#IMPORTACIÓN DE LIBRERÍAS

```
import pandas as pd      #librería Pandas para el manejo de dataset de entrenamiento
import numpy as np      #librería numpy para el manejo de vectores
from sklearn.metrics import confusion_matrix  #Librería para visualización de la
matriz de confusión de la red neuronal
from sklearn.metrics import accuracy_score  #librería para la calificación de
precisión de la RNA
import matplotlib.pyplot as plt  #Librería para crear gráficas
import seaborn as sb  #Librería para crear gráficas basada en matplotlib
```

#Carga del dataset de entrenamiento

```
riego = pd.read_csv(r"entrenamiento.csv", sep=";") #variable del dataset de
entrenamiento
riego.head() #comprueba que el dataset se carga correctamente
X = riegos[["temp", "humr", "hums", "lum"]] #Asignación de cada una de las columnas de
datos del dataset como datos de entrada de la RNA
y = riegos["riego"] #Asignación de la columna del dataset como salida de la RNA
```

#CREACIÓN DE LA ESTRUCTURA DE LA RNA

```
from sklearn.neural_network import MLPClassifier #librería de la RNA
#se asigna la variable 'mlp' como nombre de la red neuronal
mlp = MLPClassifier(hidden_layer_sizes = (2,), activation = 'relu', max_iter = 5000,
alpha=0.0001, solver='adam', learning_rate_init = 0.001, random_state = 21,
tol=0.000000001)
# Número de neuronas en la capa oculta: 2, función de activación: relu,
# número máximo de iteraciones: 5000, término de regularización: 0.0001
# solucionador para optimización de peso: adam, tasa de aprendizaje inicial = 0.001,
# tolerancia para la optimización: 1e-4
```

#PROCESO DE ENTRENAMIENTO DE LA RED NEURONAL ARTIFICIAL

```
from sklearn.model_selection import train_test_split #Librería para el
entrenamiento de la RNA
validation_size = 0.3 #Se asigna el 30% del dataset para validación del
entrenamiento de la RNA
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=validation_size,
random_state=None) # variables para el proceso de entrenamiento
# X_train: variable de entrenamiento para datos de entrada de la RNA,
# X_test: variable de entrada para validación de la RNA, y_train: variable # de
entrenamiento para datos de salida de la RNA, y_train: variable objetivo # para
validación de la RNA

mlp.fit(X_train, y_train) # Se entrena la RNA con las muestras de entrenando:
valores de entradas y valores de salida u objetivo

predictions=mlp.predict(X_test)
print(mlp.score(X,y)) # imprime el porcentaje de precisión en el entrenamiento de la
RNA
print(confusion_matrix(y_test, predictions)) # imprime la matriz de confusión
```

```

print(accuracy_score(y_test, predictions))    # imprime el porcentaje de precisión de
la RNA

riego.drop(['riego'],1).hist()    # grafica los datos de entrenamiento excepto la
variable objetivo riego
plt.show()
print ("Pesos entre la capa de entrada - capa oculta:")
print(mlp.coefs_[0])    # muestra los pesos actualizados entre la capa de entrada y la
capa oculta

print ("Pesos entre la capa oculta - capa de salida:")
print(mlp.coefs_[1])    #muestra los pesos actualizados entre la capa oculta y la
capa de salida

sb.pairplot(riego.dropna(),
hue="riego",height=4,vars=["temp","humr","hums","lum"],kind='reg')
# grafica la relación entre las variables de entrada de la RNA

from sklearn.externals import joblib    //librería para guardar la RNA
joblib.dump(mlp, 'RNA_entrenada.pkl')    //generación del archivo de la RNA entrenada

```

ANEXO 6: Integración de la red neuronal artificial a la WSN para automatizar el riego.

```
#!/usr/bin/env python3
#-*- coding: utf-8 -*- Codificación
import time          #librería de tiempo
import serial        #librería puerto serial
import mysql.connector #librería para conexiones a MySQL
lora = serial.Serial('/dev/ttyUSB0', 57600) #configuración del puerto serial

def set_modulo():
    # Configuración de parámetros del módulo LoRaWAN
    print("VERSION")
    lora.write(b'sys get ver\r\n') #Versión del módulo
    print(str(lora.readline()))

    print("SETEO wdt")
    lora.write(b'radio set wdt 105000 \r\n') #Tiempo de espera de Rx en ms
    print(str(lora.readline()))

    print("SETEO pwr")
    lora.write(b'radio set pwr 14\r\n') #Asignación de potencia de Tx
    print(str(lora.readline()))

    print("SETEO sf")
    lora.write(b'radio set sf 12\r\n') #Asignación del factor de propagación usado
    # durante la Tx
    print(str(lora.readline()))

    print("MAC PAUSE")
    lora.write(b'mac pause\r\n') #Permite cambiar al módulo de Rx a Tx y viceversa
    print(str(lora.readline()))

def nodo_a():
    while true:
        lora.write(b'radio tx 11111111111\r\n') #Envía la petición de datos al Nodo
        # 1
        print("SOLICITUD AL RN1")
        print(str(lora.readline())) #Imprime la respuesta del módulo RN2483
        print(str(lora.readline()))

        time.sleep(1)

        print("CAMBIO A Rx")
        lora.write(b'radio rx 0\r\n') #Cambio a modo de Rx
        if(str(lora.readline()) != "busy"): #Evalúa que el módulo este libre
            a= str(lora.readline()) #Variable del paquete recibido
            #Separación bit a bit de los datos recibidos
            A1=a[10]+a[11]+a[12]+a[13] #Variable para el sensor de luminosidad
            A2=a[14]+a[15]+a[16]+a[17] #Variable para el sensor de hum. suelo
            A3=a[18]+a[19]+a[20]+a[21] #Variable para el sensor de hum. Rel.
            A4=a[22]+a[23]+a[24]+a[25] #Variable para el sensor de temperatura

            #Conversión de valores analógicos a digitales de los sensores
            sensor1=(A1) # Sensor de luminosidad
            s1=float(sensor1) #Convierte de string a tipo float
```

```

SenLx=((s1-1023)*(100-1)/(1-1023)+1)
SenLx("%.1f" % SenLx) #Mostrar un solo decimal

sensor2=(A2) # Sensor de humedad de suelo
s2=float(sensor2) #Convierte de string a tipo float
SenHum=((s2-1023)*(100-1)/(1-1023)+1)
SenHum("%.1f" % SenHum) #Mostrar un solo decimal

sensor3=(A3) # Sensor de Humedad Relativa
s3=float(sensor3) #Convierte de string a tipo float
SenHR=(s3)
SenHR("%.1f" % SenHR) #Mostrar un solo decimal

sensor4=(A4) # Sensor de Temperatura
s4=float(sensor4) #Convierte de string a tipo float
SenTem=(s4)
SenTem("%.1f" % SenTem) #Mostrar un solo decimal

#Imprime los valores digitales calculados
print("LECTURAS VARIABLE SENSORES")
print(SenLx)
print(SenHum)
print(SenHR)
print(SenTem)

#Red neuronal artificial
from sklearn.externals import joblib
import pandas as pd
import numpy as np

mlp = joblib.load('mlp_entrenado.pkl') #carga el archivo de la RNA entrenada
test_val = [[SenTem, SenHR, SenHum, SenLx]] #vector de entrada a la red
neuronal
riego = (mlp.predict(test_val)) resultado de la red neuronal

if riego == 1
    print("Enviando orden de activación de riego")
    lora.write(b'radio tx 333333333333\r\n')
else:
    print("No se necesita riego")

#ALMACENAMIENTO EN LA BASE DE DATOS DEL SERVIDOR LOCAL
#Se crea una conexión con las credenciales de acceso
connection =
mysql.connector.connect(user='tesis',password='123tesis',database='DatosTesis')
cursor = connection.cursor() #Se crea un cursor para la conexión

sql = """INSERT INTO tesis(tempA,humA,humS,lum,riego) VALUES (%s,%s,%s,%s)"""
#Accede a la tabla donde se va a almacenar los datos
datos = (SenTem,SenHR,SenHum,SenLx,riego) #Datos a guardar
cursor.execute(sql, datos) #se ejecuta el cursor
cursor.close() #Se cierra el cursor
connection.close() #se cierra la conexión a MySQL
time.sleep(3600) #se inicia el ciclo while cada hora

```

```
if __name__ == '__main__':  
    set_modulo() #Se ejecuta una vez la subrutina de configuración del módulo  
    while (True): #Ciclo While  
        nodo_a() #Se ejecuta la subrutina de petición de datos y RNA  
        time.sleep(1)
```

ANEXO 7: Código del Nodo Actuador

```
#include <SoftwareSerial.h> //Librería comunicación serial
#include <Wire.h> //Librería comunicación I2C
SoftwareSerial loranode(10, 11); //Puertos de comunicación serial módulo RN2483

#define electrovalvula 5 //Puerto para la electroválvula
String str; //Variable para comunicación con módulo LoRaWAN
int riego = 1800000; //Variable del tiempo de riego

void setup() {
  Serial.begin(9600); //Velocidad de Tx de datos por puerto serial del Arduino
  loranode.begin(57600); //Velocidad de Tx de datos por puerto serial (Arduino -
  LoRaWAN)
  Wire.begin();
  pinMode(electrovalvula, OUTPUT); //Pin de electroválvula como salida

  while (!loranode); //Inicializa al módulo RN2483
  pinMode(3,OUTPUT); // Puerto para led indicador
  digitalWrite(3,HIGH); // Enciende el led indicador

  //Resetea el módulo RN2483
  pinMode(7,OUTPUT); //Puerto para el pin Reset del módulo RN2483
  digitalWrite(7,HIGH); //Puerto encendido
  delay(500);
  digitalWrite(7,LOW); //Puerto apagado
  delay(2000);

  //Configuración de parámetros del módulo RN2483
  loranode.setTimeout(100); //establece 100 milisegundos para esperar datos por
  puerto serial
  str = loranode.readStringUntil('\n');
  Serial.println(str);
  loranode.println("radio set wdt 0"); //Tiempo de espera de rx en ms
  str = loranode.readStringUntil('\n');
  Serial.println(str);
  loranode.println("radio set pwr 14"); //Asignación de potencia de Tx del módulo
  RN2483
  str = loranode.readStringUntil('\n');
  Serial.println(str);
  loranode.println("radio set sf sf12"); //Asignación del factor de propagación usado
  durante la Tx
  str = loranode.readStringUntil('\n');
  Serial.println(str);
  loranode.println("mac pause"); //Permite cambiar al módulo de Rx a Tx y
  viceversa
  str = loranode.readStringUntil('\n');
  Serial.println(str);
  digitalWrite(3,LOW); //Apaga el Led indicador
}
void loop() {

  loranode.println("radio rx 0"); //pone al módulo en modo receptor
```


ANEXO 8: Creación de tablas para el almacenamiento de datos del sistema integrado

```
CREATE TABLE IF NOT EXISTS `tesis` (  
  `Fecha` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
  CURRENT_TIMESTAMP,  
  `tempA` float NOT NULL,  
  `humA` float NOT NULL,  
  `humS` float NOT NULL,  
  `lum` float NOT NULL,  
  `riego` int NOT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

ANEXO 9: Creación de la página web para visualizar las veces de riego

```
<?php  
function conectarBD(){  
    $server = "localhost";  
    $usuario = "tesis";  
    $pass = "123tesis";  
    $BD = "DatosTesis";  
    //variable que guarda la conexión de la base de datos  
    $conexion = mysqli_connect($server, $usuario, $pass, $BD);  
    //Comprobamos si la conexión ha tenido éxito  
    if(!$conexion){  
        echo 'Ha sucedido un error inesperado en la conexión de la base de datos<br>';  
    }  
    //devolvemos el objeto de conexión para usarlo en las consultas  
    return $conexion;  
}  
/*Desconectar la conexión a la base de datos*/  
function desconectarBD($conexion){  
    //Cierra la conexión y guarda el estado de la operación en una variable  
    $close = mysqli_close($conexion);  
    //Comprobamos si se ha cerrado la conexión correctamente  
    if(!$close){  
        echo 'Ha sucedido un error inesperado en la desconexión de la base de datos<br>';  
    }  
    //devuelve el estado del cierre de conexión
```

```

        return $close;
    }

    //Devuelve un array multidimensional con el resultado de la consulta
    function getArraySQL($sql){
        //Creamos la conexión
        $conexion = conectarBD();
        //generamos la consulta
        if(!$result = mysqli_query($conexion, $sql)) die();

        $rawdata = array();
        //guardamos en un array multidimensional todos los datos de la consulta
        $i=0;
        while($row = mysqli_fetch_array($result))
        {
            //guardamos en rawdata todos los vectores/filas que nos devuelve la consulta
            $rawdata[$i] = $row;
            $i++;
        }
        //Cerramos la base de datos
        desconectarBD($conexion);
        //devolvemos rawdata
        return $rawdata;
    }

    //Sentencia SQL
    $sql = "SELECT riego from tesis;";
    //Array Multidimensional
    $rawdata = getArraySQL($sql);

    //Adaptar el tiempo
    for($i=0;$i<count($rawdata);$i++){
        $time = $rawdata[$i]["time"];
        $date = new DateTime($time);
        $rawdata[$i]["time"]=$date->getTimestamp()*1000;
    }

    ?>

```

//Proceso de graficar variables de tablas

<HTML>

<BODY>

<meta charset="utf-8">

<!-- Importo el archivo jquery directamente desde su servidor -->

<script src="https://code.jquery.com/jquery.js"></script>

<!-- Importo el archivo Javascript de Highcharts directamente desde su servidor -->

<script src="http://code.highcharts.com/stock/highstock.js"></script>

<script src="http://code.highcharts.com/modules/exporting.js"></script>

<div id="container">

</div>

<script type='text/javascript'>

\$(function () {

\$(document).ready(function() {

Highcharts.setOptions({

global: {

useUTC: false

}

});

var chart;

\$('#container').highcharts({

chart: {

type: 'spline',

animation: Highcharts.svg,

marginRight: 10,

events: {

load: function() {

}

}

},

title: {

text: 'Riego' //Título de la página web

},

xAxis: {

type: 'datetime',

tickPixelInterval: 150

},

yAxis: {

title: {

```

        text: 'Value'
    },
    plotLines: [{
        value: 0,
        width: 1,
        color: '#808080'
    }]
},
tooltip: {
    formatter: function() {
        return '<b>'+ this.series.name +'</b><br/>'+
            Highcharts.dateFormat('%Y-%m-%d %H:%M:%S', this.x) +'<br/>'+
            Highcharts.numberFormat(this.y, 2);
    }
},
legend: {
    enabled: true
},
exporting: {
    enabled: true
},
series: [{
    name: 'Riego',
    data: (function() {
        var data = [];
        <?php
            for($i = 0 ;$i<count($rawdata);$i++){
                ?>
                data.push([<?php echo $rawdata[$i]["time"];?>,<?php echo
$rawdata[$i]["riego"];?>]);
                <?php } ?>
            return data;
        })()
    },{
    }]
});
});
</script>
</html>

```

ANEXO 10: Guía de comandos de configuración del módulo RN2483

Seguir el enlace: <https://1drv.ms/b/s!AiwyysTVRWDlhiThJExFcVhOQMfe>

ANEXO 11: Guía de usuario de la librería sklearn

Seguir en enlace: https://1drv.ms/b/s!AiwyysTVRWDlhiM017hV1rBj8_d5

ANEXO 12: Dataset de entrenamiento de la RNA

Seguir el enlace: <https://1drv.ms/u/s!AiwyysTVRWDlhiUSikwBFXCQuF51?e=bFDDLt>

ANEXO 13: Circuito Impreso Shield Módulo LoRa para Raspberry Pi

Seguir el enlace: <https://1drv.ms/u/s!AiwyysTVRWDlhidvm8IZXfoRiCsc?e=z7uoML>

ANEXO 14: Circuito Impreso Shield Módulo LoRa para Arduino

Seguir Enlace: https://1drv.ms/u/s!AiwyysTVRWDlhih9Nb0vm1KTRzi_?e=hxkefM

ANEXO 15: Circuito Impreso Shield Nodo Actuador

Seguir Enlace: <https://1drv.ms/u/s!AiwyysTVRWDlhibCeeIfdO0PRzuH?e=j85RIL>