



UNIVERSIDAD TÉCNICA DEL NORTE

Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Ingeniería en Mecatrónica

**APLICACIÓN WEB DE ADMINISTRACIÓN DE BASE DE DATOS Y DE
APLICACIÓN DE APRENDIZAJE DE MÁQUINA EN LÍNEA PARA UN SISTEMA
IoT DE RIEGO INTELIGENTE**

Trabajo de grado previo a la obtención del título de Ingeniero en Mecatrónica

Autor:

Edison Javier Cangás Román

Director:

Carlos Xavier Rosero Chandi

Ibarra - Ecuador

2023



Universidad Técnica del Norte
Biblioteca Universitaria
Autorización de uso y publicación a favor de la
Universidad Técnica del Norte
Identificación de la Obra

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

Datos del Autor		
Cédula de identidad	0402046791	
Apellidos y Nombres	Cangás Román Edison Javier	
Dirección	Maldonado y Teodoro, Ibarra	
Email	ejcangasr@utn.edu.ec	
Teléfono fijo	Teléfono móvil	0980591851
Datos de la Obra		
Tema	“APLICACIÓN WEB DE ADMINISTRACIÓN DE BASE DE DATOS Y DE APLICACIÓN DE APRENDIZAJE DE MÁQUINA EN LÍNEA PARA UN SISTEMA IOT DE RIEGO INTELIGENTE”	
Apellidos y Nombres	Cangás Román Edison Javier	
Fecha	30 de noviembre de 2023	
Programa	Pregrado	
Título por el que opta	Ingeniero en Mecatrónica	
Director	Carlos Xavier Rosero Chandi	



Universidad Técnica del Norte
Facultad de Ingeniería en Ciencias Aplicadas
Constancia

El autor manifiesta que la obra objeto de la presente autorización es original y se desarrolló sin violar derechos de autor de terceros; por lo tanto, la obra es original y que es 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 30 días del mes de noviembre de 2023

Edison Javier Cargás Román

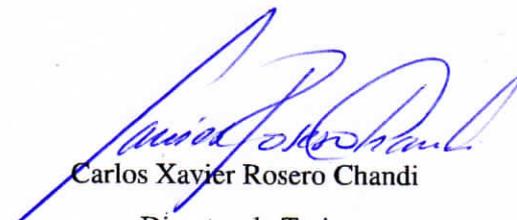
C.I: 0402046791



Universidad Técnica del Norte
Facultad de Ingeniería en Ciencias Aplicadas
Certificación del director del trabajo de grado

En mi calidad de director del trabajo de grado “APLICACIÓN WEB DE ADMINISTRACIÓN DE BASE DE DATOS Y DE APLICACIÓN DE APRENDIZAJE DE MÁQUINA EN LÍNEA PARA UN SISTEMA IoT DE RIEGO INTELIGENTE”, presentado por el egresado Edison Javier Cangás Román, que opta por el título de Ingeniero en Mecatrónica, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Ibarra, 30 de noviembre de 2023


Carlos Xavier Rosero Chandi
Director de Tesis

Dedicatorias

Dedico este trabajo de tesis a mis padres, German Cangás y Olga Román, por su inquebrantable apoyo y paciencia. Este trabajo no solo representa el fruto de mi esfuerzo y dedicación, sino también el reflejo del amor y el apoyo que he recibido de ustedes. Han sido mi fuente de inspiración desde mi infancia y lo seguirán siendo en cada paso que dé. A mis queridos hermanos, Ronald, Magaly y Johanna, con quienes comparto y vivo los mejores momentos de mi vida, por su constante respaldo. Han sido mi apoyo incondicional, mis confidentes y mi guía a lo largo de esta travesía académica.

Agradecimientos

Agradezco de todo corazón a mi familia por su apoyo constante a lo largo de mi trayecto académico. El constante aliento y respaldo que me han ofrecido han sido fundamentales en mi logro. Estoy profundamente agradecido por tenerlos en mi vida.

Expresar mi gratitud hacia mis amigos, Anderson y Alexander, con los que hemos compartido triunfos y fracasos, pero siempre hemos seguido adelante. Agradezco la buena amistad que hemos cultivado durante todos estos años.

A mi tutor de tesis, el PhD. Xavier Rosero, Su dedicación y su vasto conocimiento han sido fundamentales en la realización de este trabajo. Estoy sinceramente agradecido por su valiosa contribución a este proyecto.

Índice general

Cesión de derechos de autor a favor de la Universidad Técnica del Norte	II
Declaración	III
Certificación del director del trabajo de grado	IV
Dedicatorias	V
Agradecimientos	VI
Índice general	VII
Índice de figuras	X
Índice de tablas	XII
Resumen	XIII
Abstract	XIV
I. Introducción	1
1.1. Planteamiento del problema	1
1.2. Objetivos	2
1.2.1. Objetivo general	2
1.2.2. Objetivos específicos	2
1.3. Justificación	2

1.4.	Alcance	3
II.	Revisión literaria	5
2.1.	Antecedentes	5
2.2.	Fundamentos y tecnologías utilizadas	7
2.2.1.	Internet de las cosas (IoT)	7
2.2.2.	Riego inteligente	9
2.2.3.	Modelos de datos para sistemas de riego inteligente	10
2.2.4.	Variables ambientales	11
2.2.5.	Importancia de la gestión de bases de datos en sistemas IoT	12
2.2.6.	Firestore	13
2.2.7.	Recolección de datos en tiempo real	14
2.2.8.	Aplicación de aprendizaje de máquina en sistemas IoT	15
2.2.9.	Dispositivos utilizados	18
III.	Metodología	23
3.1.	Arquitectura de sistema IoT	23
3.2.	Sistema local	23
3.2.1.	Sensores de las variables ambientales	24
3.2.2.	Bomba	26
3.2.3.	Microcontrolador	29
3.2.4.	Microcomputador	30
3.2.5.	Diagrama de conexión	33
3.3.	Integración en la nube	34
3.3.1.	Base de datos de tiempo real	35
3.4.	Integración de dispositivos remotos	36
3.4.1.	Servidor	36
3.4.2.	Aplicación web	42

IV. Resultados y análisis	45
4.1. Implementación del enlace entre bases de datos	45
4.1.1. Enlace entre la base de datos de tiempo real y la base de datos SQL	45
4.1.2. Gestión de datos temporales y etiquetado preciso	46
4.1.3. Almacenamiento efectivo y referencia histórica	47
4.1.4. Estructura y tipos de datos en SQLite	48
4.2. Arquitectura de hardware	49
4.2.1. Hardware implementado	49
4.2.2. Validación del hardware	51
4.2.3. Análisis de datos	54
4.3. Aprendizaje de máquina	56
4.3.1. Selección de técnicas de aprendizaje de máquina	56
4.3.2. Conjunto de datos	57
4.3.3. Implementación de KNN	60
4.3.4. Precisión del modelo KNN	62
V. Conclusiones, recomendaciones y trabajo a futuro	65
5.1. Conclusiones	65
5.2. Recomendaciones	66
5.3. Trabajo a futuro	67
Anexos	72
A. Programación Raspberry Pi 3	73
B. Programación Servidor	79
C. Implementación	81

Índice de figuras

2.1. Funciones principales de Firebase	14
2.2. Placa Raspberry Pi 3	19
2.3. Placa Arduino Nano	20
2.4. Sensor de humedad de suelo capacitivo V2.0	21
2.5. Módulo DHT11	22
3.1. Arquitectura de sistema IoT utilizada.	23
3.2. Conexión del sensor DHT11	25
3.3. Conexión del sensor humedad de suelo	26
3.4. Conexión del módulo relé.	28
3.5. Diagrama de flujo de microcontrolador.	29
3.6. Diagrama de flujo microcomputador.	31
3.7. Diagrama general de conexiones.	34
3.8. Almacenamiento de variables y datos temporales.	35
3.9. Diagrama de flujo para generar base de datos local.	39
3.10. Diagrama de la forma de actuar el servidor.	41
3.11. Trama de la lógica de la aplicación.	43
3.12. Interfaz web.	44
4.1. Estructura de base de datos SQLite	49
4.2. Implementación hardware	50
4.3. Funcionamiento del sensor de humedad del suelo.	51
4.4. Funcionalidad del sistema.	54

4.5. Precisión para obtención del valor K	61
4.6. Matriz de confusión	63
C.1. Cultivo en macetas	81
C.2. Gotero ajustable	82
C.3. Ubicación de los goteros ajustables	82
C.4. Sistema completo	83

Índice de tablas

2.1. Características de la Raspberry Pi 3	19
2.2. Características de Arduino Nano	20
2.3. Características de sensor de humedad de suelo Capacitivo V2.0	21
2.4. Características de módulo DHT11	22
3.1. Descripción de componentes y procesos del programa.	31
4.1. Tabla de variación de medidas en intervalos de tiempo.	52
4.2. Tabla de resultados de los parámetros de los sensores.	53
4.3. Datos de humedad del suelo, humedad del aire y temperatura.	55
4.4. Tabla de precisión y exactitud	57

Resumen

La agricultura de precisión utiliza tecnologías para mejorar la productividad, la rentabilidad y la sostenibilidad de los cultivos. Los sistemas de riego inteligente son una parte importante en esta tendencia, ya que pueden ayudar a los usuarios a ahorrar agua y energía. Esta tesis se enfoca en desarrollar un sistema de riego inteligente basado en IoT con aprendizaje automático para la toma de decisiones. El sistema propuesto consta de una aplicación web que permite a los usuarios monitorear variables ambientales y controlar el riego automática o manualmente según su preferencia. Además, incluye de una arquitectura de hardware que recopila datos de sensores y los envía a una base de datos en tiempo real. Posteriormente, se almacenan en una base de datos local para garantizar la disponibilidad de datos actualizados. Estos datos se utilizan para el análisis de tendencias climáticas, ayudando a los usuarios a identificar patrones en las precipitaciones, temperatura y otros factores climáticos. Esta información puede utilizarse para optimizar las rutinas de riego, asegurando que los cultivos reciban la cantidad adecuada de agua en el momento oportuno. La investigación aborda la necesidad de mejorar la eficiencia del riego en la agricultura mediante la implementación de sistemas de riego inteligentes. Los resultados de la investigación muestran que el sistema propuesto es capaz de tomar decisiones de riego precisas y eficientes, mejorando la gestión del riego agrícola. La aplicación web simplifica su uso y administración, respondiendo a la necesidad de aplicar tecnología para optimizar el uso de recursos y aumentar la producción en un contexto donde los sistemas de riego basados en IoT ganan interés.

Palabras clave: Agricultura de precisión, Riego inteligente, IoT (Internet de las cosas) Aprendizaje de máquina, Eficiencia en la gestión del riego agrícola.

Abstract

Precision agriculture uses technologies to improve crop productivity, profitability, and sustainability. Smart irrigation systems are an important part of this trend, as they can help users save water and energy. This thesis focuses on developing an IoT-based smart irrigation system with machine learning for decision-making. The proposed system consists of a web application that allows users to monitor environmental variables and control irrigation automatically or manually according to their preference. It also includes a hardware architecture that collects sensor data and sends it to a real-time database. This data is then stored in a local database to ensure the availability of up-to-date data. These data are used for climate trend analysis, helping users identify patterns in rainfall, temperature, and other weather factors. This information can be used to optimize irrigation routines, ensuring that crops receive the right amount of water at the right time. The research addresses the need to improve irrigation efficiency in agriculture by implementing smart irrigation systems. The research results show that the proposed system is able to make accurate and efficient irrigation decisions, improving agricultural irrigation management. The web application simplifies its use and administration, responding to the need to apply technology to optimize resource use and increase production in a context where IoT-based irrigation systems are gaining interest.

Keywords: Precision agriculture, Smart irrigation, IoT (Internet of Things), Machine learning, Efficiency in agricultural irrigation management.

Capítulo I

Introducción

1.1. Planteamiento del problema

La agricultura de precisión consiste en la aplicación de herramientas basadas en tecnologías informáticas actuales en conjunto con sensores y herramientas de internet de las cosas (IoT, Internet of Things)[1]. Su aporte radica en mejorar el rendimiento, la rentabilidad y la calidad de los cultivos, con una menor afectación al medio ambiente. Dentro de la agricultura de precisión, IoT y aprendizaje de máquina son facilitadores para la implementación de sistemas de riego inteligentes que incluyen monitoreo y control continuos de varios parámetros.

Nuevas tendencias de ingeniería apoyadas sobre computación distribuida se enfocan en sistemas de riego semiautomáticos, automáticos e inteligentes, tanto para aplicaciones urbanas como para rurales. Todas ellas tienen como factor común el uso de sensores de variables ambientales y actuadores de apertura/cierre, conectados a un sistema microprocesado que a la vez se enlaza con un servidor en la nube. En el caso de sistemas inteligentes de riego, estos toman ventaja de su experiencia con datos de variables ambientales y con decisiones manuales y automáticas de riego, para ajustar su dinámica y mejorar su rendimiento [2].

Bajo la arquitectura descrita, el servidor en la nube es parte de los servicios proporcionados por plataformas IoT gratuitas y pagadas. Dentro de este contexto, si bien se puede adquirir información a través de bases de datos de tiempo real, por ejemplo, Firebase [3], aparece la dificultad de guardar esta información en una base de datos para su posterior análisis.

Entre las técnicas usadas para realizar el análisis de la información se encuentra el aprendizaje de máquina en línea que permite resolver problemas mientras el sistema se encuentra trabajando, usando información obtenida desde la base de datos. Esta información ha sido obtenida en campo y corresponde a variables ambientales y de riegos manuales (hora y duración)[4].

Se propone realizar una aplicación web dinámica que permita enlazar la base de datos de tiempo real de Firebase con una base de datos SQL, para administrar la información de variables ambientales y rutinas de riego, además de aplicar en línea algoritmos de aprendizaje de máquina para mejorar el rendimiento de sistemas de riego inteligente.

1.2. Objetivos

1.2.1. Objetivo general

- Desarrollar una aplicación web de administración de base de datos y de aplicación de aprendizaje de máquina en línea para un sistema IoT de riego inteligente.

1.2.2. Objetivos específicos

- Diseñar un modelo general de enlace entre la base de datos de tiempo real y la base de datos tipo SQL que integre la recolección y preparación de variables ambientales con el aprendizaje de modelos de riego.
- Proponer una arquitectura de hardware para la prueba de concepto del modelo.
- Examinar el rendimiento de varias técnicas de aprendizaje de máquina sobre el sistema desarrollado.

1.3. Justificación

La agricultura basada en datos es una forma de gestión agrícola que ayuda a los agricultores a manejar sus operaciones. Desde el rendimiento del suelo hasta la calidad de los cultivos. Por lo

tanto, permite a los agricultores tomar decisiones precisas, rápidas e inteligentes, las cuales están basadas en hechos y cifras reales, en lugar de confiar únicamente en sus instintos. Mediante un servidor el agricultor tendrá acceso a los patrones de cambios climáticos y necesidades de riego [5].

Un riego inteligente usa tecnologías para gestionar de un modo óptimo esta labor. El propósito es hacer un uso más eficiente de los recursos de producción como agua, energía y fertilizantes. Además de ahorrar en estos recursos, también aumentará la producción, y de esta manera se logrará generar más con menos [6].

Un sistema IoT de riego inteligente se compone de una serie de sensores que miden desde la humedad hasta la temperatura. En ese sentido, el sistema puede indicar el momento más adecuado para empezar a regar un cultivo. De esta forma, solo se gastan recursos cuando sea necesario, incluso en áreas de cultivos muy específicas [7].

El tema del presente trabajo de grado permite aplicar habilidades y aptitudes adquiridas a lo largo de la carrera por el candidato a ingeniero en mecatrónica. Específicamente se trabajará en tres de las cuatro áreas de pertinencia de la ingeniería en mecatrónica tales como la computación, los sistemas electrónicos y los sistemas de control.

1.4. Alcance

La propuesta consta de las fases de análisis, diseño, implementación y pruebas. La fase de análisis comprende la revisión literaria y el estudio de las tecnologías concernientes, además de las conclusiones relacionadas con éstos. Dentro de la fase de diseño se encuentra la propuesta del modelo de enlace de la base de datos de Firebase con una base de datos tipo SQL que además permita la obtención y el tratamiento en línea de la información para su aplicación a modelos de aprendizaje de máquina. Para la implementación, se desarrolla el hardware a manera de prueba de concepto. Empieza por la definición de requisitos en relación con variables adquiridas, almacenamiento, acceso a la nube y aprendizaje automático. Finaliza con el ensayo de diferentes alternativas de aprendizaje de máquina para la obtención de modelos inteligentes de riego. La propuesta comprende dos modos de funcionamiento: riegos manuales en los que se

toma información sobre las variables ambientales con sus respectivas marcas de tiempo, y riegos automáticos en los que el sistema toma decisiones de riego en base a modelos adquiridos a través de aprendizaje de máquina. Para llevar a cabo cualquiera de los modos de funcionamiento, la obtención de datos y la preparación de estos, son pasos esenciales. Los datos recolectados corresponden a los campos de humedad del suelo, humedad del ambiente y temperatura, con sus marcas de tiempo asociadas. En lo referente a la preparación de los datos, se contemplan la eliminación de anomalías en los valores, la normalización de las medidas y la identificación y etiquetado de los puntos de riego en el conjunto de datos. En cuanto a la aplicación del modelo, en el modo manual, luego de cierto tiempo de adquisición de la información, se entrena con los datos y se obtiene un modelo para la toma de decisión de riego. En el modo automático, el riego es realizado aplicando el modelo aprendido en el modo manual. El hardware de soporte del sistema consiste en sensores de variables ambientales y en una bomba de agua. Estos dispositivos finales están conectados a un sistema microprocesado que envía la información a una base de datos de tiempo real en la nube. Un computador conectado a la base de datos de tiempo real guarda la información en una base de datos interna para su posterior análisis a través de varias técnicas de aprendizaje de máquina.

Capítulo II

Revisión literaria

2.1. Antecedentes

Un estudio realizado por J. Laverde y C. Laverde Mena [8] propone la implementación de Internet de las Cosa (IoT) en sistemas de riego utilizados en la agricultura del Ecuador. Para ello se utilizó una placa NodeMCU programada para transmitir datos a la plataforma IoT ThinkSpeak mediante sensores. Además, se empleó un bot en Telegram para informar al usuario sobre el estado del suelo y permitirle activar el riego. La adopción de IoT en la agricultura mejoró la calidad y productividad de los cultivos. Se habían realizado investigaciones previas en viveros de café y jardines, utilizando IoT para ahorrar agua, controlar variables ambientales y optimizar la calidad de las plantas. La implementación de sistemas de riego automatizados en jardines había demostrado ser factible y beneficiaba la salud pública. La utilización de IoT en el sector agrícola permitió recolectar datos para optimizar el funcionamiento de las plantas. Un caso de estudio mostró mejoras en la calidad y producción de champiñones mediante soluciones de software y hardware de código abierto. Sin embargo, la falta de conocimiento limitaba la implementación de nuevas tecnologías en la agricultura ecuatoriana. A pesar de esto, se propuso utilizar IoT para el monitoreo autónomo del suelo y la irrigación de cultivos. El sistema propuesto utilizó sensores de humedad del suelo y envió información a ThingSpeak y Telegram. Si las condiciones eran óptimas pero el nivel de agua era insuficiente, el sistema podía activar el riego automáticamente, mejorando la eficiencia en el uso del agua. El sistema redujo la mano de obra y permitió el ahorro de agua, mientras que el propietario pudo monitorear las variaciones del suelo.

"Gestión de sistema de riego inteligente para el cuidado del parque Palomino Flores de la Ciudad de Baños de Agua Santa"[9]. El proyecto de investigación se centró en la concepción y

desarrollo de un prototipo de sistema de riego inteligente que se aplicó en el parque "Palomino Flores", ubicado en la ciudad de Baños de Agua Santa. El objetivo era automatizar el proceso de riego a través de la monitorización de las variables climáticas y del suelo. Se utilizó un módulo ESP32 que adquirió los valores de humedad y temperatura, los cuales fueron almacenados en una base de datos para su posterior análisis y procesamiento. El sistema también incluyó una interfaz web para visualizar la información recolectada, permitiendo la supervisión, el control y el seguimiento del riego. Además de la automatización del riego, el proyecto contempló el desarrollo de documentación con recomendaciones y cuidados para el crecimiento de las plantas. Asimismo, se utilizó una bomba de agua que se activó cuando las condiciones ambientales y del suelo fueron óptimas o cuando el usuario lo solicitó, optimizando el uso de los recursos hídricos y reduciendo los costos asociados. El proyecto se realizó con la finalidad de innovar y mantener la estética del parque, ahorrando recursos económicos y naturales, especialmente el agua. Con base en los antecedentes investigativos, se determinó la necesidad de mejorar la calidad, la interconexión y el rendimiento de los servicios urbanos mediante la incorporación de tecnologías de la información y la comunicación (TIC) en el ámbito de las ciudades inteligentes. Por ello, se consideró que el proyecto era una oportunidad para aplicar estos conocimientos en un caso práctico, lo que permitió desarrollar nuevas soluciones para mejorar la gestión de los recursos naturales y el bienestar de las personas en el entorno urbano.

“Aplicación web para la gestión de los datos asociados a los cultivos de la Empresa La Quinta”[10]. La investigación se ha enfocado en el desarrollo de una aplicación web para la gestión de datos relacionados con los cultivos de la empresa La Quinta. En este contexto, se ha subrayado la importancia de las tecnologías de la información y comunicación (TIC) en la agricultura, especialmente en la Agricultura Inteligente o Smart Farming, que permite mejorar la producción y calidad de los productos agrícolas y facilitar la ejecución de los procesos. Se ha introducido también el concepto de Agricultura de Precisión (AP), que se utiliza para monitorear las condiciones físicas, químicas y climáticas del cultivo con el fin de optimizar el uso de recursos y mejorar la calidad final de los productos. Se ha destacado que, aunque las Granjas Inteligentes son una forma de producir alimentos de alta calidad y utilizar los recursos de manera eficiente, el hecho de no contar con un registro y control sistematizado de las variables ambientales puede provocar pérdidas en la producción agrícola, incluso cuando se dispone de invernaderos. Por esta razón, se ha propuesto el desarrollo de una aplicación web que permita el registro y control de las variables ambientales y facilite la toma de decisiones en el interior de los invernaderos, con el objetivo de aumentar la producción y mejorar la eficiencia en el uso de los recursos. El desarrollo de la aplicación web propuesta tiene como objetivo contribuir a mejorar la producción y la sostenibilidad a largo plazo en el cultivo de la empresa La Quinta.

En definitiva, esta investigación ha sido fundamental para la implementación de nuevas tecnologías en la agricultura y para la mejora de los procesos productivos en este ámbito.

“Aprendizaje Automático para toma Decisiones en Aplicaciones de Riego Inteligente”[11]. Este proyecto se centró en el desarrollo de un sistema de soporte de decisiones para la irrigación inteligente en un huerto experimental utilizando el dispositivo Agro-sensor. El objetivo fue generar una base de datos de reglas de irrigación propuestas y utilizar la librería Scikit-learn en Python para implementar cuatro clasificadores supervisados: SGDClassifier, MLPClassifier, Adaboost y Gradientboosting. Los clasificadores fueron entrenados y validados con los datos de la base de datos, demostrando que Adaboost fue el clasificador con mejor desempeño, con una puntuación de F1-score del 84,35 %. El proyecto se basó en la necesidad de interpretar el impacto de los cambios climáticos sobre los cultivos y mejorar las técnicas en la agricultura, como el uso eficiente de los fertilizantes y del agua. La agricultura era la actividad que más consumía agua, y se estimaba que la producción de alimentos debía aumentar alrededor del 60 % para 2050 debido al crecimiento de la población y las limitaciones en el área de cultivo. El cambio climático era una de las principales preocupaciones de la sociedad y se esperaba que la actividad agrícola disminuyera el efecto invernadero. Las soluciones de riego inteligente se habían implementado en los recientes años para el uso eficiente del agua, pero algunas presentaban varias aristas que necesitaban mejorar. La inteligencia artificial se había utilizado para recopilar información de las estaciones agroclimáticas y crear soluciones de riego más eficientes, pero estas soluciones tendían a desplazar al agricultor de las decisiones de riego y desaprovechaban su experiencia y conocimiento al respecto.

2.2. Fundamentos y tecnologías utilizadas

En este apartado se abordará de manera técnica los fundamentos teóricos y conceptuales. Asimismo, se examinarán las tecnologías y herramientas empleadas en el proyecto, resaltando aquellas que posibilitan la visualización, captura, procesamiento y gestión eficiente de la información. Además, se describirán los principios y enfoques teóricos proporcionando un marco teórico sólido para comprender el proceso de desglose de información y su relación con la generación del proyecto.

2.2.1. Internet de las cosas (IoT)

Internet de las cosas o, IoT, se refiere a la interconexión y comunicación de dispositivos y objetos a través de redes, ya sea privadas o Internet. Esta conexión permite que los dispositivos

sean visibles y capaces de interactuar entre sí. Los objetos y dispositivos involucrados pueden ser de diversos tipos, que van desde sensores y dispositivos mecánicos hasta objetos de uso diario, como electrodomésticos, calzado o prendas de vestir. La idea central es que cualquier cosa imaginable puede estar conectada a Internet y ser capaz de interactuar sin intervención humana. En este contexto, se busca una interacción de máquina a máquina, también conocida como M2M (machine to machine) o interacción de dispositivos M2M [12].

2.2.1.1. Aplicaciones y beneficios del IoT en el ámbito agrícola y de riego

El Internet de las cosas (IoT) está experimentando una creciente adopción en el ámbito agrícola y de riego, siendo utilizado para diversas aplicaciones. Estas incluyen el monitoreo meteorológico, el seguimiento de la composición y propiedades del suelo, así como la automatización del proceso de riego. La implementación del IoT en este sector brinda a los agricultores y terratenientes la capacidad de acceder a datos en tiempo real sobre las condiciones de sus cultivos, lo que les permite tomar decisiones fundamentadas en cuanto al riego, fertilización y control de plagas y enfermedades. Una ventaja destacada es la automatización del sistema de riego, lo cual conlleva a un uso más eficiente del agua, siendo especialmente valioso en regiones donde su escasez es un problema. Adicionalmente, el IoT contribuye a prevenir el desperdicio de agua y reducir los costos de producción en general. En resumen, la implementación del IoT en el ámbito agrícola y de riego puede generar mejoras sustanciales en la eficiencia y rentabilidad, al tiempo que se reduce el impacto ambiental asociado [12].

2.2.1.2. Desafíos y consideraciones de seguridad en los sistemas IoT

La implementación de sistemas IoT plantea diversos retos y aspectos a considerar en términos de seguridad, debido a la amplia gama de dispositivos interconectados y la transmisión de datos a través de redes. Uno de los principales desafíos reside en la ausencia de estándares de seguridad homogéneos, lo que puede dar lugar a vulnerabilidades. Además, los dispositivos IoT suelen contar con recursos limitados, lo que dificulta la aplicación de medidas de seguridad sólidas. Otras consideraciones abarcan la necesidad de autenticación segura y salvaguarda de datos personales.

Para abordar estos desafíos, es necesario contar con una estrategia de seguridad sólida que incluya la identificación y mitigación de riesgos, la implementación de medidas de seguridad apropiadas y la capacitación del usuario final en las mejores prácticas de seguridad [13].

2.2.2. Riego inteligente

Un sistema de riego inteligente es una solución tecnológica que permite controlar de manera precisa y eficiente la cantidad y frecuencia del riego en áreas verdes, como jardines y terrenos. Los pilares fundamentales de este sistema comprenden la medición precisa de la humedad del suelo, la programación adaptada al clima y a la evapotranspiración de las plantas, así como la utilización de sensores para evitar el riego excesivo o inadecuado. La implementación de sistemas de riego inteligentes tiene como objetivo mejorar la eficiencia en el uso del agua y reducir su consumo, lo cual supone ventajas tanto para los propietarios de viviendas como para el medio ambiente. La base de un sistema de riego inteligente radica en la monitorización constante de la humedad del suelo mediante sensores de alta precisión. Estos dispositivos recopilan datos en tiempo real y proporcionan información detallada sobre las necesidades hídricas de las plantas. A su vez, estos datos se combinan con la información climática, incluyendo la temperatura, la radiación solar y la velocidad del viento, para ajustar automáticamente el programa de riego. Otra característica clave de los sistemas de riego inteligentes es la utilización de algoritmos sofisticados que tienen en cuenta la evapotranspiración de las plantas. Esto implica calcular la cantidad de agua que se pierde por la evaporación del suelo y la transpiración de las plantas, con el fin de determinar la cantidad de agua necesaria para mantener un nivel óptimo de hidratación de las plantas [14].

2.2.2.1. Métodos de riego inteligente

- Sistemas de riego automatizados que emplean sensores y datos meteorológicos para adecuar los horarios de riego según las exigencias de las plantas y las condiciones ambientales. Estos sistemas se valen de dispositivos sensoriales que captan información sobre la humedad del suelo, la temperatura y la precipitación, entre otros parámetros relevantes. Estos datos se utilizan para determinar cuándo y cuánta agua debe suministrarse a las plantas, optimizando así la eficiencia hídrica y evitando el desperdicio [15].
- Sistemas de riego por goteo, los cuales proporcionan agua de forma precisa y directa a la base de las plantas. Este método se caracteriza por la aplicación controlada de pequeñas cantidades de agua a través de emisores, como goteros o micro aspersores, distribuidos estratégicamente cerca de las raíces de las plantas. De este modo, se minimiza la evaporación y el escurrimiento, reduciendo la pérdida de agua y aumentando la eficacia del riego [16].
- Sensores de humedad del suelo, dispositivos que miden la cantidad de agua presente

en el suelo de manera precisa y en tiempo real. Estos sensores, basados en tecnología electromagnética o capacitiva, detectan la humedad del suelo en diferentes niveles de profundidad y transmiten esta información a los sistemas de riego. De esta forma, se logra una irrigación precisa y ajustada a las necesidades hídricas de las plantas, evitando tanto el riego excesivo como el riego insuficiente [17].

- Aplicaciones móviles y sistemas de control remoto que permiten a los usuarios supervisar y ajustar los horarios de riego desde cualquier ubicación. Estas herramientas tecnológicas posibilitan la gestión remota del sistema de riego, brindando a los usuarios la capacidad de monitorear los parámetros relacionados con el riego, como la humedad del suelo o los datos climáticos, y realizar ajustes en los horarios de riego en función de las necesidades específicas de las plantas. Esta funcionalidad ofrece comodidad y flexibilidad al permitir la toma de decisiones informadas sobre el riego sin la necesidad de estar físicamente presentes en el lugar del cultivo [18].

2.2.3. Modelos de datos para sistemas de riego inteligente

En los sistemas de riego inteligente, la selección del modelo de datos y la estructura de la base de datos son elementos cruciales para asegurar la eficiencia y efectividad del riego. A continuación, se detallan algunos de los modelos de datos más utilizados en sistemas de riego inteligente:

- **Datos climáticos:** Los datos climáticos son elementos fundamentales en la implementación de estrategias de gestión inteligente del riego. Estos datos comprenden diversas variables atmosféricas, incluyendo la temperatura, humedad relativa, velocidad y dirección del viento, y la precipitación. La adquisición de estos datos se realiza mediante la utilización de sensores meteorológicos ubicados estratégicamente, así como la recopilación y análisis de datos históricos. La integración de esta información es crucial para la construcción de un modelo climático preciso y confiable, que permita realizar predicciones y tomar decisiones informadas en relación al riego y otras actividades agrícolas [19].
- **Datos del suelo:** Los datos del suelo desempeñan un papel de vital importancia en la gestión precisa y eficiente del riego agrícola. Estos datos incluyen información detallada sobre las propiedades físicas y químicas del suelo, como la textura, estructura, contenido de materia orgánica, capacidad de retención de agua, niveles de nutrientes. La recopilación y análisis de estos datos son esenciales para determinar la cantidad óptima de agua que una planta requiere en un momento dado, evitando tanto el estrés hídrico como el

desperdicio de recursos. Los sensores de humedad del suelo se utilizan comúnmente para medir el contenido de agua en el suelo en tiempo real y obtener información sobre la disponibilidad de agua para las plantas [20].

- **Datos de las plantas:** Los datos de las plantas desempeñan un papel esencial en la gestión inteligente del riego, ya que permiten determinar las necesidades hídricas específicas de cada especie vegetal. Estos datos comprenden una variedad de información, como el tipo de especie, la etapa de crecimiento y la tasa de transpiración. La comprensión detallada de las características de cada especie de planta es crucial para establecer estrategias de riego eficientes y optimizar el uso del agua. Mediante investigaciones y estudios especializados, es posible obtener datos precisos sobre los requerimientos de agua de diferentes especies de plantas. Estos estudios analizan factores como la capacidad de absorción de agua de las raíces, la tasa de evaporación de las hojas y la transpiración, así como las necesidades específicas de agua en cada etapa del ciclo de crecimiento de la planta [21].

2.2.4. Variables ambientales

Las variables ambientales suelen ser factores determinantes en el éxito del riego inteligente. Entre las variables más importantes se encuentran la temperatura del aire, la humedad relativa y la humedad del suelo. Al tener en cuenta estas variables, es posible ajustar el riego de manera adecuada y lograr un uso más eficiente del agua.

2.2.4.1. Temperatura del aire

La temperatura del aire es una variable ambiental extremadamente relevante en el contexto de un sistema de riego inteligente. La temperatura afecta directamente la tasa de evaporación del agua y la transpiración de las plantas, lo que a su vez influye en las necesidades de riego. Cuando la temperatura del aire es alta, la tasa de evaporación del agua aumenta. Esto significa que el agua aplicada a las plantas se evaporará más rápidamente, lo que puede resultar en una mayor demanda de agua por parte de las plantas. En estas condiciones, es posible que sea necesario aumentar la frecuencia y la duración del riego para satisfacer las necesidades hídricas de las plantas. Por otro lado, en temperaturas más bajas, la tasa de evaporación es menor. En estas condiciones, es posible que las plantas requieran menos agua, por lo que se puede reducir la frecuencia y la cantidad de riego. La temperatura del aire también puede influir en la transpiración de las plantas, que es el proceso por el cual liberan vapor de agua a través de sus hojas. Una mayor temperatura del aire puede aumentar la transpiración y, por lo tanto, las necesidades de agua de las plantas [22].

2.2.4.2. Humedad relativa

La humedad relativa del aire se refiere a la cantidad de vapor de agua presente en el aire en relación con la cantidad máxima que el aire puede contener a una determinada temperatura. La humedad relativa puede influir en el proceso de transpiración de las plantas y, por lo tanto, en sus necesidades de agua. En condiciones de alta humedad relativa, el aire ya está saturado de vapor de agua, lo que dificulta la evaporación del agua de las hojas de las plantas. Como resultado, las plantas pueden requerir menos agua, ya que la transpiración se reduce. En estas condiciones, el sistema de riego inteligente puede ajustar la frecuencia y la duración del riego para evitar un exceso de humedad en el suelo. Por otro lado, en condiciones de baja humedad relativa, el aire puede contener más vapor de agua, lo que facilita la evaporación del agua de las plantas. En estas condiciones, las plantas pueden experimentar una mayor pérdida de agua a través de la transpiración, lo que aumenta sus necesidades hídricas. Un sistema de riego inteligente puede detectar una baja humedad relativa y aumentar la frecuencia o la cantidad de riego para compensar la pérdida de agua [23].

2.2.4.3. Humedad del suelo

La humedad del suelo es la cantidad de agua presente en el suelo y su conocimiento es fundamental para determinar la cantidad y el momento adecuado de aplicación del riego a las plantas. El monitoreo de la humedad del suelo permite evaluar la disponibilidad de agua para las raíces de las plantas, manteniendo un equilibrio adecuado de humedad en el suelo para garantizar un crecimiento saludable y prevenir problemas como el estrés hídrico o el encharcamiento. Además, conocer la humedad del suelo permite establecer criterios precisos de riego, evitando usar agua de manera excesiva y contribuyendo a la conservación del agua y eficiencia del sistema de riego. Al utilizar sensores de humedad del suelo en diferentes áreas del sistema de riego, se pueden obtener datos para realizar ajustes precisos en la aplicación de agua en las áreas que lo necesiten y evitar desperdiciar agua en áreas ya suficientemente húmedas [24].

2.2.5. Importancia de la gestión de bases de datos en sistemas IoT

La gestión de bases de datos desempeña un papel crucial en los sistemas de IoT debido a la enorme cantidad de datos generados, que requieren ser almacenados y procesados de manera eficiente. Estos sistemas involucran una multiplicidad de dispositivos interconectados que capturan y generan datos en tiempo real, potencialmente en diversos formatos. La gestión eficaz de esta ingente cantidad de información resulta fundamental para lograr que los sistemas de IoT sean efectivos y proporcionen información de valor. En este sentido, es necesario contar con

la capacidad de procesar grandes volúmenes de datos, asegurar su almacenamiento de forma segura y estructurada, así como disponer de capacidades analíticas en tiempo real para tomar decisiones precisas. En consecuencia, una óptima gestión de bases de datos se vuelve esencial para asegurar el correcto funcionamiento y la relevancia de los sistemas de IoT [25].

2.2.5.1. Tipos de bases de datos utilizadas en sistemas IoT

En el contexto de los sistemas de IoT, se pueden emplear una variedad de tipos de bases de datos según las necesidades particulares del sistema. Entre los tipos de bases de datos más utilizados en sistemas IoT, se destacan los siguientes:

- Bases de datos NoSQL: Estas bases de datos son especialmente adecuadas para almacenar grandes volúmenes de datos no estructurados generados en entornos de IoT. Algunos ejemplos notables de bases de datos NoSQL son MongoDB y Cassandra.
- Bases de datos de tiempo real: Estas bases de datos ofrecen la capacidad de procesar datos en tiempo real, lo que las convierte en una elección idónea para sistemas de IoT que generan datos de forma continua y en tiempo real. Ejemplos de bases de datos de tiempo real incluyen Apache Kafka y Firebase Realtime Database.
- Bases de datos relacionales: Estas bases de datos se caracterizan por su precisión, consistencia y eficiencia, lo cual las hace apropiadas para sistemas de IoT que procesan datos estructurados y requieren la capacidad de realizar consultas complejas. Entre las bases de datos relacionales más comunes se encuentran MySQL y PostgreSQL.

En términos generales, la selección del tipo de base de datos utilizada dependerá de las necesidades específicas del sistema de IoT en cuestión. Es fundamental evaluar cuidadosamente los requisitos de almacenamiento, procesamiento y consulta de datos para determinar la opción más adecuada. Cada tipo de base de datos presenta ventajas y consideraciones particulares, y es importante elegir la opción que mejor se ajuste a los objetivos y requerimientos del sistema de IoT en su conjunto [26].

2.2.6. Firebase

Firebase de Google es una plataforma tecnológica que proporciona una infraestructura eficiente y ágil para el desarrollo de aplicaciones web y móviles. Su enfoque principal se centra en la recopilación de datos en tiempo real, lo cual implica la capacidad de capturar, procesar y almacenar información de manera inmediata y continua. La recolección de datos en tiempo real

es un aspecto fundamental de Firebase, ya que permite a los desarrolladores obtener información actualizada y relevante sobre el comportamiento de los usuarios, las interacciones con la aplicación y otros eventos en tiempo real. Esta capacidad se logra a través de una variedad de servicios y herramientas, incluyendo bases de datos en tiempo real, notificaciones push y análisis de eventos. Firebase se encarga de recolectar y almacenar datos en tiempo real mediante su base de datos en tiempo real, que ofrece una sincronización en tiempo real entre los dispositivos y la nube. Esto permite a los desarrolladores obtener datos actualizados en tiempo real y proporcionar a los usuarios una experiencia fluida y en tiempo real [27].

Además de la recolección de datos en tiempo real, Firebase ofrece una amplia gama de características y funcionalidades para mejorar el rendimiento y la seguridad de las aplicaciones. Esto incluye servicios de autenticación, almacenamiento en la nube, hosting web, análisis de rendimiento y prueba de errores, entre otros. Como se muestra en la siguiente figura 2.1, estos servicios se integran de manera efectiva para proporcionar una solución completa para el desarrollo de aplicaciones.



Figura 2.1: Funciones principales de Firebase [28].

2.2.7. Recolección de datos en tiempo real

La recolección de datos en tiempo real es un proceso mediante el cual los datos se recopilan y analizan en tiempo real. Es utilizado para tomar decisiones informadas de manera más rápida, identificar tendencias y responder rápidamente a cambios en el mercado o la competencia. Firebase Realtime Database es una base de datos alojada en la nube que almacena datos en formato JSON y se sincroniza en tiempo real con cada cliente conectado. Esto permite experiencias colaborativas con capacidades sin conexión y puede ser accedido directamente desde dispositivos

móviles o navegadores web sin necesidad de un servidor de aplicaciones. Firebase simplifica el desarrollo de aplicaciones y proporciona herramientas esenciales para los desarrolladores, como bases de datos en tiempo real y autenticación de usuarios.

2.2.8. Aplicación de aprendizaje de máquina en sistemas IoT

El aprendizaje de máquina (machine learning) en el contexto del Internet de las cosas (IoT) ofrece un amplio rango de aplicaciones que contribuyen a mejorar la eficiencia y seguridad en diversos ámbitos. Una de estas aplicaciones se encuentra en la optimización de la eficiencia energética en edificios inteligentes. Mediante el uso de sensores IoT que recolectan datos sobre la temperatura, humedad, iluminación y otros factores ambientales, se pueden entrenar modelos de aprendizaje automático para controlar de manera más precisa los sistemas de ventilación, calefacción y aire acondicionado. Estos modelos pueden ajustar automáticamente los parámetros de los sistemas en función de las condiciones ambientales en tiempo real, lo que permite un uso más eficiente de la energía y una reducción en los costos de consumo [29].

Además, el aprendizaje de maquina en el IoT también desempeña un papel crucial en la detección de fraudes y anomalías. En un entorno de IoT, donde múltiples dispositivos y sistemas están interconectados, existe un riesgo potencial de actividades fraudulentas o comportamientos anómalos. Los sistemas de seguridad y monitoreo basados en el aprendizaje automático pueden analizar grandes volúmenes de datos en tiempo real para identificar patrones sospechosos o comportamientos inusuales. Estos modelos pueden aprender de forma continua, adaptándose a nuevas amenazas y desarrollando capacidades de detección más sofisticadas. La detección temprana de fraudes y anomalías en el IoT ayuda a proteger la integridad de los datos, prevenir pérdidas económicas y salvaguardar la privacidad de los usuarios [30].

Otra aplicación importante del aprendizaje automático en el IoT se encuentra en la gestión y optimización de recursos. Los sistemas de IoT generan una gran cantidad de datos en tiempo real, y el aprendizaje automático puede ser utilizado para analizar y extraer información valiosa de estos datos. Por ejemplo, en el sector industrial, los sensores IoT pueden monitorear el rendimiento de las máquinas y equipos, y los modelos de aprendizaje automático pueden analizar estos datos para predecir fallos o realizar un mantenimiento preventivo. Esto permite maximizar la eficiencia operativa, minimizar los tiempos de inactividad y optimizar el uso de los recursos disponibles [31].

2.2.8.1. Algoritmos y técnicas de aprendizaje de máquina

En los sistemas de riego inteligente, se pueden utilizar varias técnicas y algoritmos de aprendizaje de máquina para optimizar el riego y reducir el consumo de agua. Algunos de estos algoritmos y técnicas incluyen:

- **Árboles de decisión:** Los árboles de decisión son algoritmos utilizados en la modelización y predicción de la demanda de agua de las plantas en sistemas de riego inteligente. Estos algoritmos aprovechan factores ambientales como la temperatura, humedad, presión atmosférica, entre otros, para segmentar y categorizar la demanda de agua en distintos niveles de consumo. Esta segmentación permite identificar de manera precisa los momentos más apropiados para realizar el riego, optimizando así la cantidad de agua utilizada [32].
- **Redes neuronales:** Las redes neuronales son algoritmos empleados para modelar el comportamiento de las plantas en relación a diversas condiciones ambientales y las necesidades de agua en tiempo real. Estos algoritmos utilizan una estructura inspirada en el funcionamiento del cerebro humano, donde se simulan interconexiones entre "neuronas" artificiales. Esto permite capturar relaciones complejas y no lineales entre los factores ambientales y las necesidades de agua de las plantas, lo cual brinda un mayor nivel de precisión en el control del riego [33].
- **Algoritmos de agrupamiento:** Los algoritmos de agrupamiento se utilizan para clasificar diferentes tipos de plantas que presentan patrones de consumo de agua distintos. Al agrupar las plantas en clusters o grupos, es posible adaptar la cantidad de agua proporcionada durante el riego según las características particulares de cada grupo. De esta manera, se logra una distribución más eficiente del agua, atendiendo de manera adecuada a las necesidades específicas de cada tipo de planta. Esto conlleva a una mejora sustancial en la utilización del recurso hídrico, evitando el riego excesivo o insuficiente [34].

En conjunto, la aplicación de algoritmos como los árboles de decisión, redes neuronales y algoritmos de agrupamiento en los sistemas de riego inteligente permite una gestión más precisa y eficiente del agua en el cultivo de plantas. Estos algoritmos proporcionan información valiosa para determinar el momento y la cantidad de riego necesaria, considerando factores ambientales y las características específicas de cada planta. Como resultado, se logra una optimización del uso del agua, mejorando tanto la eficiencia del riego como el rendimiento de los cultivos.

2.2.8.2. Procesamiento y análisis de datos para la toma de decisiones en tiempo real

Existen diversas herramientas y técnicas para el procesamiento y análisis de datos que se utilizan para la toma de decisiones en tiempo real. Algunas de ellas son:

- **Sistemas de bases de datos en tiempo real:** Los sistemas de bases de datos en tiempo real son herramientas fundamentales para el almacenamiento y procesamiento de grandes volúmenes de datos en tiempo real. Estas herramientas permiten la captura, el almacenamiento y la gestión eficiente de los datos generados por los dispositivos interconectados del IoT. Además, facilitan el procesamiento y el análisis continuo de los datos en tiempo real, lo que permite obtener información actualizada y precisa para la toma de decisiones inmediatas. Estos sistemas son especialmente importantes en aplicaciones que requieren respuestas rápidas y en entornos donde la información en tiempo real es crítica [35].
- **Sistemas de monitoreo y visualización de datos en tiempo real:** Los sistemas de monitoreo y visualización de datos en tiempo real desempeñan un papel crucial en la gestión de los sistemas de IoT. Estas herramientas permiten la visualización de los datos generados por los dispositivos conectados en tiempo real, presentándolos de forma clara y comprensible. Esto facilita la supervisión continua de los datos, lo que a su vez facilita la toma de decisiones rápidas y precisas. Estos sistemas suelen incluir paneles de control y gráficos interactivos que permiten a los usuarios obtener una visión completa y actualizada del estado del sistema, detectar anomalías y tomar medidas correctivas de manera oportuna [36].
- **Análisis estadístico en tiempo real:** El análisis estadístico en tiempo real implica la aplicación de técnicas estadísticas y algoritmos de análisis de datos para procesar y extraer información útil de los datos generados en tiempo real. Estos análisis permiten identificar patrones, tendencias y correlaciones relevantes en los datos en tiempo real, lo que proporciona una base sólida para la toma de decisiones informadas y basadas en evidencias. Además, el análisis estadístico en tiempo real puede ayudar a identificar desviaciones significativas en los datos, lo que permite la detección temprana de anomalías o situaciones críticas [37].
- **Aprendizaje de máquina en tiempo real:** El uso del aprendizaje automático (machine learning) en tiempo real implica la aplicación de algoritmos y técnicas de aprendizaje automático para analizar, predecir y clasificar datos en tiempo real. Estos algoritmos se entrenan utilizando conjuntos de datos históricos y en tiempo real, lo que les permite aprender y adaptarse a medida que se generan nuevos datos. El aprendizaje de máquina

en tiempo real permite la toma de decisiones automatizadas y en tiempo real, ya que los modelos aprendidos pueden generar predicciones y clasificaciones instantáneas basadas en los datos entrantes. Esto es especialmente útil en aplicaciones que requieren respuestas rápidas y en entornos dinámicos donde los datos se actualizan constantemente [38].

- **Procesamiento de eventos complejos:** El procesamiento de eventos complejos se utiliza para analizar y procesar datos generados en tiempo real en entornos complejos. Esta técnica permite la detección de patrones, correlaciones y relaciones significativas entre múltiples eventos, lo que permite la identificación y comprensión de situaciones complejas en tiempo real. El procesamiento de eventos complejos se basa en reglas y patrones predefinidos, y utiliza algoritmos avanzados para analizar y evaluar eventos en tiempo real.

En definitiva, estos son solo algunos de los enfoques más comunes utilizados para el procesamiento y análisis de datos en tiempo real para la toma de decisiones. Existen muchos otros tipos de herramientas y técnicas disponibles, y su elección dependerá de las necesidades y los objetivos específicos de cada caso [38], [39].

2.2.9. Dispositivos utilizados

En esta sección se procede a analizar los componentes electrónicos que serán incorporados. Es importante destacar que cada elemento ha sido seleccionado en base a su usabilidad y capacidad para llevar a cabo las funciones requeridas en un sistema de riego, adaptándose adecuadamente a los requisitos del proyecto. Asimismo, cabe mencionar la existencia de una amplia comunidad de conocimiento en este campo, lo cual facilita el proceso de aprendizaje para la integración de estos dispositivos con otros y su programación correspondiente.

2.2.9.1. Raspberry Pi 3

Es una placa de desarrollo de bajo costo y tamaño reducido que combina un procesador de arquitectura ARM con capacidades de conectividad y ampliación. Se ha convertido en una herramienta popular para proyectos de electrónica y programación debido a su versatilidad y facilidad de uso [40].



Figura 2.2: Placa Raspberry Pi 3 [40].

Las características de esta placa se resume en la tabla 2.1 la cual detalla aspectos clave que definen su rendimiento y funcionalidades.

Tabla 2.1: Características de la Raspberry Pi 3 [40].

Característica	Descripción
Procesador	ARM Cortex-A53 de 64 bits a 1,2 GHz
Memoria	1 GB de RAM compartida entre el procesador y los gráficos
Conectividad	Wi-Fi 802.11n y Bluetooth 4.2 integrados
Puertos USB	4 puertos USB 2.0
Salida de video	Puerto HDMI para conectar a un monitor o televisor
Conectividad de red	Conector Ethernet para conexión a redes cableadas
Conector de audio	Conector de audio de 3,5 mm para salida de audio
Almacenamiento	Ranura para tarjetas microSD para el sistema operativo y datos
Sistema operativo	Compatible con diversos sistemas operativos
GPIO	Encabezado de E/S de propósito general (GPIO) de 40 pines

2.2.9.2. Arduino Nano

Un Arduino Nano es una placa microcontroladora diseñada para proyectos de electrónica con tamaño reducido y bajo costo. Es muy utilizado en proyectos de robótica, sistemas de sensores y proyectos de automatización, dada su facilidad de programación y su versatilidad, permitiendo soporte una amplia variedad de sensores y dispositivos.

Su tamaño reducido y su sencillez lo hacen una herramienta muy útil para proyectos de electrónica pequeños [41].



Figura 2.3: Placa Arduino Nano [41].

Las características de esta placa se resume en la tabla 2.2 la cual detalla aspectos clave que definen su rendimiento y funcionalidades.

Tabla 2.2: Características de Arduino Nano [41].

Característica	Descripción
Microcontrolador	ATmega328P
Voltaje de funcionamiento	5V
Voltaje de entrada	7-12V (recomendado), 6-20V (límite)
Pines digitales	14 (6 de ellos pueden ser usados como salidas PWM)
Pines analógicos	8
Corriente de salida DC	40 mA por pin
Memoria Flash	32 KB (2 KB utilizados por el bootloader)
SRAM	2 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz
Comunicación	UART, I2C, SPI
Conectividad	USB, ICSP (In-Circuit Serial Programming)
Dimensiones	18 x 45 mm

2.2.9.3. Sensor de humedad de suelo capacitivo V2.0

Es un dispositivo electrónico de medición de la humedad del suelo que se basa en la utilización de la capacitancia del suelo. Este sensor emplea la variación de la capacitancia entre dos electrodos, uno de ellos enterrado en el suelo y el otro expuesto al aire, para determinar la cantidad de agua presente en el suelo. A medida que las condiciones de humedad del suelo se modifican, la capacitancia entre los electrodos experimenta cambios, lo que permite al sensor establecer el nivel de humedad del suelo con gran precisión. Este sensor es ampliamente utilizado en aplicaciones agrícolas y de jardinería con el propósito de monitorizar y controlar el riego de forma precisa [42].

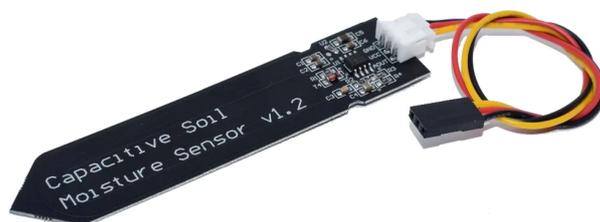


Figura 2.4: Sensor de humedad de suelo capacitivo V2.0 [42].

Las características de este sensor se resume en la tabla 2.3 la cual detalla aspectos clave que definen su rendimiento y funcionalidades.

Tabla 2.3: Características de sensor de humedad de suelo Capacitivo V2.0 [42].

Característica	Descripción
Voltaje de operación	3,3 V o 5 V
Corriente de operación	Menos de 20 mA
Rango de medición de humedad	0-100 % de humedad relativa
Precisión	Menos de 5 % de error
Salida de señal	Salida analógica
Tamaño	60 mm x 20 mm x 1,6 mm
Material del sensor	Placa PCB y chips de sensor de humedad

2.2.9.4. Módulo DHT11

El módulo DHT11 es un sensor de temperatura y humedad utilizado en proyectos de electrónica y domótica. Este sensor es capaz de medir la temperatura y la humedad relativa del aire circundante y enviar esta información a través de una señal digital.

El módulo DHT11 es de bajo costo y fácil de usar, lo que lo hace popular para proyectos de automatización y monitoreo ambiental [43].

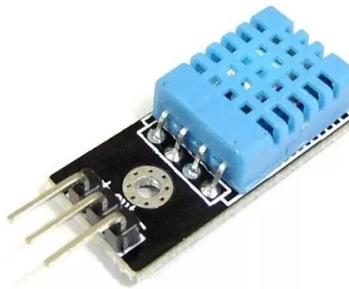


Figura 2.5: Módulo DHT11 [43].

Las características de este sensor se resume en la tabla 2.4 la cual detalla aspectos clave que definen su rendimiento y funcionalidades.

Tabla 2.4: Características de módulo DHT11 [43].

Característica	Descripción
Voltaje de operación	3,3 V o 5 V
Corriente de operación	Menos de 2,5 mA
Rango de medición de temperatura	0°C a 50°C
Precisión de temperatura	±2°C
Rango de medición de humedad	20 % a 90 % de humedad relativa
Precisión de humedad	±5 % de humedad relativa
Salida de señal	Señal digital

Capítulo III

Metodología

En este capítulo se describe detalladamente el proceso utilizado para el diseño y desarrollo del sistema.

3.1. Arquitectura de sistema IoT

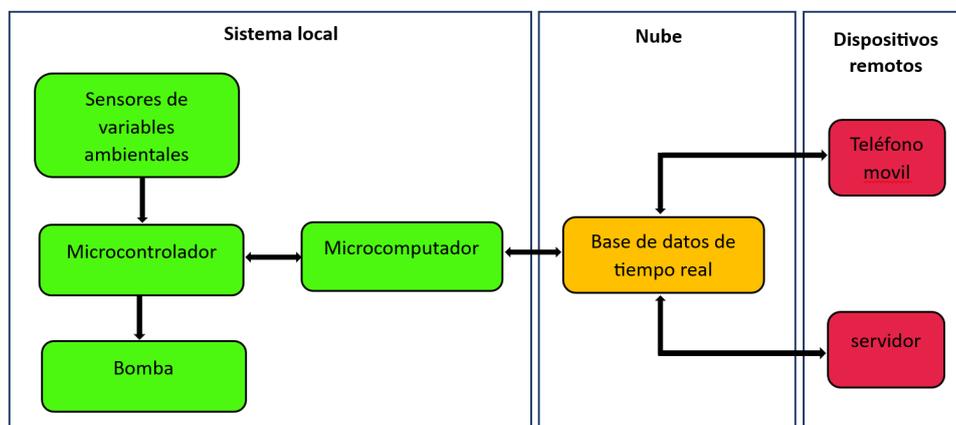


Figura 3.1: Arquitectura de sistema IoT utilizada.

3.2. Sistema local

En el apartado se aborda la implementación y configuración de un entorno local para el sistema de riego inteligente basado en IoT. En este sistema, los datos recopilados por los sensores serán enviados al microcontrolador Arduino Nano, el cual establecerá una comunicación con la Raspberry Pi. Esta última actuará como dispositivo central encargado de recibir y procesar los datos mediante un programa en Python.

3.2.1. Sensores de las variables ambientales

En el ámbito de un sistema de riego inteligente, la utilización de sensores para detectar y analizar las variables ambientales se vuelve esencial. Esta práctica permite una comprensión detallada de las necesidades específicas de las plantas y de las condiciones climáticas circundantes, posibilitando así una adaptación precisa en el suministro de agua. Al monitorear aspectos como la temperatura del aire, la humedad relativa y la humedad del suelo, se logra una optimización efectiva del riego al evitar tanto el exceso como la insuficiencia de agua. Este enfoque asegura un uso sostenible de este recurso vital. Además, el análisis continuo de estas variables ambientales proporciona datos valiosos en tiempo real, permitiendo ajustes inteligentes en la frecuencia y cantidad de agua aplicada. Esta capacidad de toma de decisiones instantánea contribuye a mejorar el rendimiento de los cultivos y a reducir los impactos negativos en el medio ambiente.

3.2.1.1. Sensor temperatura y humedad relativa

La temperatura del aire y la humedad relativa son factores clave en un sistema de riego inteligente, ya que permiten ajustar el riego de manera precisa y eficiente, evitando el desperdicio de agua y asegurando un suministro óptimo para el crecimiento saludable de las plantas.

El sensor utilizado para medir la temperatura del aire y la humedad relativa es el Sensor DHT11. Este dispositivo digital combina un sensor de temperatura y un sensor de humedad relativa. Utiliza una tecnología de detección de cambios capacitivos para medir la humedad relativa del aire y un termistor para medir la temperatura ambiente. A través de una interfaz de comunicación de un solo cable, este sensor puede proporcionar datos precisos y en tiempo real sobre la temperatura y la humedad del entorno.

En la figura 3.2 se observa, como el sensor se encuentra conectado al microcontrolador.

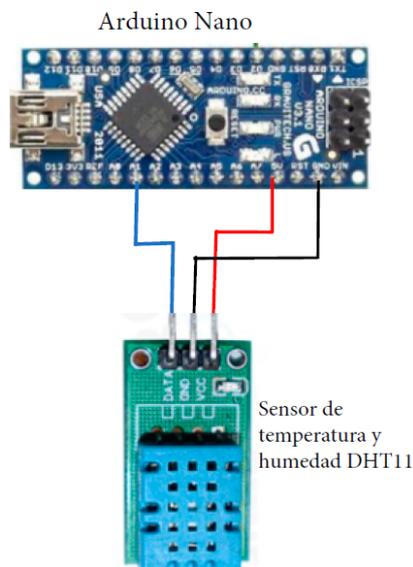


Figura 3.2: Conexión del sensor DHT11

Para el funcionamiento del sensor DHT11, se establece la conexión del pin A1 del Arduino Nano a la señal del sensor. Además, se conecta el pin 5V al pin VCC del sensor, y se establece la conexión del pin GND del microcontrolador al pin GND del sensor.

3.2.1.2. Sensor de humedad del suelo

La medición de la humedad del suelo es fundamental para el crecimiento saludable de las plantas y para minimizar el desperdicio de agua en el proceso de riego. Además, su inclusión contribuye al ahorro de agua y energía, lo que reduce los costos monetarios asociados al riego y optimiza el rendimiento del sistema en condiciones cambiantes.

Para medir la humedad del suelo se utiliza un sensor de humedad de suelo capacitivo V2.0. Este sensor utiliza una técnica basada en capacitancia para evaluar la cantidad de humedad presente en el suelo. Consiste en una sonda que se inserta en el suelo y un circuito de medición que detecta los cambios en la capacitancia provocados por la variación de humedad. De esta manera, proporciona información valiosa sobre el contenido de humedad del suelo, lo que resulta crucial para la agricultura, la jardinería y otras aplicaciones relacionadas con el control y monitoreo ambiental.

La figura 3.3 muestra, como el sensor se encuentra conectado al microcontrolador.

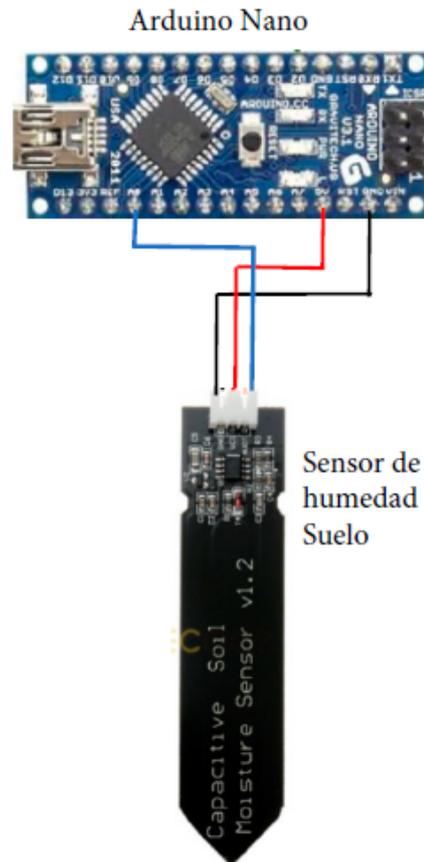


Figura 3.3: Conexión del sensor humedad de suelo

Para el funcionamiento del sensor de humedad del suelo capacitivo V2.0, se establece la conexión del pin VCC del sensor al pin 5V del Arduino, el pin GND al pin de tierra (GND) del Arduino y, por último, el pin AOUT al pin analógico A0 del Arduino. Este proceso garantiza una comunicación eficiente y precisa entre el sensor y la plataforma Arduino, posibilitando así la captura y procesamiento óptimos de datos analógicos.

3.2.2. Bomba

La bomba utilizada en este caso es una mini bomba de agua sumergible de 12 voltios. Este dispositivo, compacto y eficiente, presenta características técnicas que lo hacen idóneo para su aplicación específica. Con una capacidad de absorción máxima de 3 metros y un caudal de 240 litros por hora (4 litros por minuto), esta bomba ofrece un rendimiento óptimo para las

necesidades de bombeo de agua.

Las dimensiones de la bomba, siendo de tan solo 40x35x28 mm (longitud x ancho x altura), resaltan su diseño compacto, lo que facilita su integración en espacios reducidos. El cuerpo de la bomba está fabricado en plástico, proporcionando durabilidad y resistencia a los elementos ambientales.

Este componente es alimentado por un voltaje de 12VDC con una corriente de menos de 250 mA cuando está en pleno funcionamiento, garantizando eficiencia energética en su operación. La conexión de entrada y salida, ambas con un diámetro de 5 mm, contribuye a la versatilidad de la bomba en diferentes configuraciones.

Para gestionar el encendido y apagado de la bomba, se incorpora un relé en el sistema. El relé, actuando como un interruptor controlado por un circuito eléctrico, desempeña un papel crucial. Cuando el relé se activa, permite que la corriente fluya hacia la bomba, poniéndola en funcionamiento. En contraste, cuando el relé se desactiva, interrumpe el flujo de corriente, desactivando la bomba de manera eficaz.

Este mecanismo de control, proporcionado por el relé, confiere al sistema una capacidad de gestión eficiente y precisa. La coordinación entre la bomba y el relé asegura un uso efectivo de la energía, maximizando la vida útil de la bomba y optimizando su rendimiento en diversas aplicaciones. En resumen, la combinación de la mini bomba de agua sumergible y el relé ofrece una solución técnica integral para el manejo eficaz del flujo de agua en entornos específicos.

3.2.2.1. Conexiones necesarias

Para garantizar el funcionamiento adecuado y el control preciso de la bomba, es esencial comprender la interconexión entre sus componentes principales. En primer lugar, la bomba está conectada a un módulo relé que, a su vez, se encuentra conectado a un microcontrolador. Este último desempeña un papel crucial, ya que es el encargado de enviar señales al módulo relé para gestionar el encendido y apagado de la bomba de manera eficiente.

En términos técnicos, el módulo relé utilizado para esta aplicación específica es un módulo relé con capacidad de operar a 5 voltios. Este módulo presenta una notable versatilidad al soportar voltajes en los contactos que oscilan entre 110VAC y 220VAC, y puede gestionar una corriente máxima de hasta 10 amperios. Esta capacidad de corriente garantiza que la bomba pueda ser controlada de manera segura y eficaz mediante el módulo relé. La conexión física entre el módulo relé y la bomba se establece de manera que cuando el microcontrolador envía una señal al módulo relé, este responde activando la bomba. De manera correspondiente, cuando la señal de apagado es transmitida al módulo relé, este desactiva la bomba. Es crucial comprender esta dinámica de conexión para asegurar un control preciso y fiable del sistema.

Para lograr lo mencionada, se llevan a cabo las siguientes conexiones físicas. En primer lugar, el suministro de voltaje del relé, conocido como VCC, se conecta al pin de 5V del microcontrolador, proporcionando así la energía necesaria para su funcionamiento. Además, el terminal de tierra del relé, denominado GND, se conecta al pin GND del microcontrolador, estableciendo una referencia común para el flujo de corriente y asegurando la integridad del circuito. Por último, el pin de entrada digital IN1 del relé se conecta al pin digital D2 del microcontrolador, permitiendo así enviar señales de control al relé y activar o desactivar su función según sea necesario. En la figura 3.4 se pueden apreciar las conexiones entre el microcontrolador y el relé, así como la conexión de la bomba al relé.

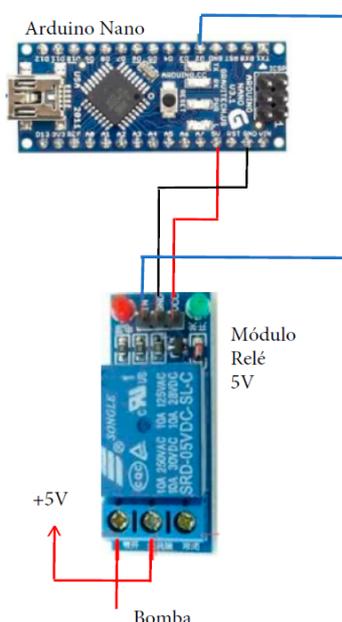


Figura 3.4: Conexión del módulo relé.

Para el funcionamiento de la bomba, es necesario establecer una conexión adecuada. En este caso, el positivo de la corriente se conecta al terminal COM (común) del módulo de relé, mientras que el positivo de la bomba se conecta al terminal NO del relé. De esta manera, cuando el relé esté activado, la bomba se encenderá. Además, para completar el circuito, es importante conectar el negativo de la bomba al negativo de la corriente. De acuerdo con la señal enviada al relé, la bomba se encenderá o apagará según sea necesario.

3.2.3. Microcontrolador

El microcontrolador utilizado en este proyecto es el Arduino Nano. Este microcontrolador es ampliamente utilizado en proyectos de electrónica y programación debido a su tamaño compacto y su capacidad para interactuar con diferentes sensores y actuadores. En este caso, el Arduino Nano se utiliza para capturar y procesar datos de sensores ambientales. El microcontrolador actúa como interfaz de comunicación entre los sensores y la Raspberry Pi. Su función es recopilar datos del entorno físico y transmitirlos al microcomputador para su procesamiento. A su vez, el microcontrolador intercambia información con el microcomputador mediante una conexión USB en el puerto serie, utilizando la clase SerialCx para facilitar la comunicación. Se desarrolla un programa diseñado para capturar estos datos, siguiendo una lógica eficiente y precisa, en la figura 3.5 se muestra la lógica que realiza el microcontrolador para la captura y envío de información.

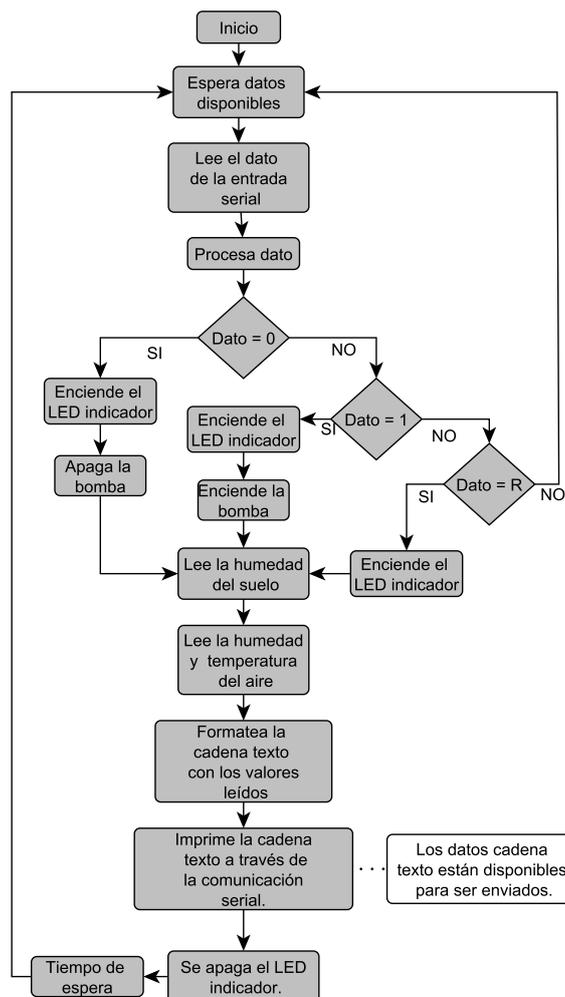


Figura 3.5: Diagrama de flujo de microcontrolador.

En cuanto a los componentes, se han integrado un sensor de temperatura y humedad (DHT11) junto con un sensor de humedad de suelo capacitivo (Sensor de Humedad de Suelo Capacitivo V2.0). La presencia de estos sensores ofrece una fuente confiable de datos esenciales para aplicaciones diversas, como agricultura y monitoreo climático. Además, se ha implementado un módulo relé, el cual se encarga del encendido y apagado de la bomba. La conexión de estos componentes al microcontrolador se llevó a cabo de manera coherente, estableciendo así una integración eficiente en el sistema. En el caso del sensor DHT11, se realizó una conexión adecuada como se observa en la figura 3.2, mientras que para el sensor de humedad de suelo capacitivo, la conexión se estableció según lo indicado en la figura 3.3. Asimismo, el módulo relé se conectó al microcontrolador según se muestra en la figura 3.4.

En lo referente a la programación, se realiza a través del entorno de desarrollo integrado (IDE) de Arduino. Para utilizar los sensores DHT11 y sensor de humedad del suelo Capacitivo V2.0, se implementan las bibliotecas necesarias. La Biblioteca DHT, específica de Arduino, facilita la interacción con los sensores de temperatura y humedad DHT, permitiendo la lectura de datos y cálculos para obtener temperatura y humedad relativa. La Biblioteca CapacitiveSensor, también de Arduino, se emplea para trabajar con sensores de humedad de suelo capacitivos, midiendo la humedad del suelo mediante la capacidad eléctrica y un circuito capacitivo. Antes de utilizar los sensores, es necesario definir los pines de entrada y salida utilizados por cada sensor en el microcontrolador, lo cual se realiza mediante el código del programa.

Se realiza la inicialización esto corresponde a la configuración inicial necesaria para que los sensores funcionen correctamente. Esto incluye la configuración de los parámetros de medición y la calibración de los sensores si es necesario.

Una vez que el sistema de adquisición de datos esté programado y funcionando correctamente, se puede integrar en el entorno o la aplicación específica según sea el escenario.

3.2.4. Microcomputador

El microcomputador, en este caso una placa Raspberry Pi, desempeña un papel esencial en la gestión del riego a través de un programa Python. Este proceso se lleva a cabo mediante la adquisición de datos provenientes de sensores y la interacción con un servidor Firebase. En esencia, el programa realiza un seguimiento del modo de riego actual (manual o automático) y supervisa constantemente el estado de la bomba.

Con base en esta información, activa o desactiva la bomba de riego según sea necesario para mantener un sistema eficiente. La figura 3.6 proporciona una representación visual del flujo operativo del programa.

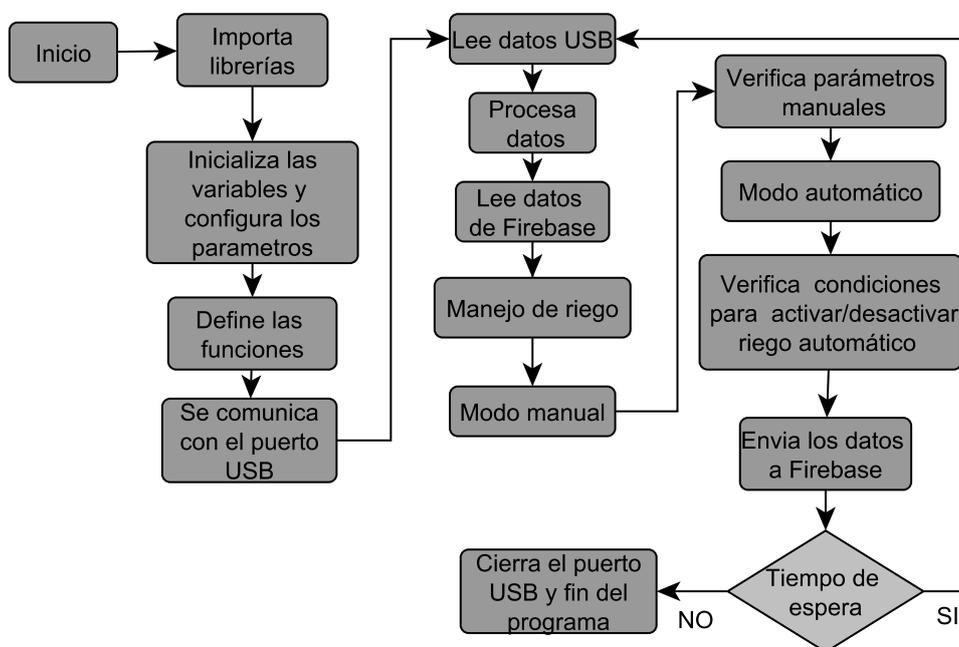


Figura 3.6: Diagrama de flujo microcomputador.

Para facilitar la comprensión del diagrama, se ha elaborado una tabla que detalla los diferentes procesos junto con breves descripciones del programa. Esta tabla proporciona una visión organizada de las funciones utilizadas para invocar cada proceso, ofreciendo así una guía clara sobre cómo se estructura el código. A continuación, se presenta la tabla 3.1 que ilustra de manera concisa los procedimientos y funciones empleados en el programa.

Tabla 3.1: Descripción de componentes y procesos del programa.

Componente/Proceso	Descripción
Inicio del programa	Inicia el programa principal.
Instancia de serialCx	Crea una instancia de la clase <code>serialCx</code> para la comunicación con el puerto USB.

Bucle principal	Ejecuta un bucle infinito que controla la lógica principal del programa.
Lectura de datos desde el microcontrolador	Utiliza <code>readData</code> para leer datos desde el microcontrolador a través del puerto serial.
Cálculos con datos del microcontrolador	Calcula la humedad del suelo, humedad del aire, temperatura del aire y estado de la bomba.
Lectura de datos desde Firebase	Utiliza <code>getData</code> para obtener información desde Firebase (modo, estado de riego, <code>autoF</code> , etc.).
Control de la bomba	Decide si encender o apagar la bomba basándose en el modo, estado de riego y variable <code>autoF</code> desde Firebase.
Envío de datos a Firebase	Utiliza <code>sendData</code> para enviar datos de sensores a Firebase.
Registro de datos en un archivo CSV	Registra datos actuales en un archivo CSV llamado 'logfile.csv'.
Espera y contadores	Espera 1 segundo antes de comenzar el próximo ciclo del bucle principal. Actualiza contadores.
Bucle infinito	Continúa ejecutando el programa en un bucle infinito.
Inicio del riego manual	Inicia el proceso de riego manual si el modo es manual y se activa el riego manual.
Encender la bomba	Utiliza <code>readData</code> para enviar la señal de encendido (<code>b'1'</code>) al microcontrolador.
Espera y contador de riego manual	Espera 1 segundo y actualiza el contador de riego manual.
Apagar la bomba al finalizar	Utiliza <code>readData</code> para enviar la señal de apagado (<code>b'0'</code>) al microcontrolador. Actualiza el estado de riego y envía los datos actualizados a Firebase.

Control de la bomba en modo automático	En modo automático, controla la bomba basándose en la variable <code>autof</code> desde Firebase.
Inicio del proceso de obtención de datos	Inicia el proceso de obtención de datos desde el microcontrolador.
Envío de comando al microcontrolador	Utiliza <code>readData</code> para enviar el comando de lectura (<code>b' R'</code>) al microcontrolador.
Espera y lectura de datos	Espera 0.05 segundos para dar tiempo al microcontrolador para procesar la solicitud.
Procesamiento de datos	Extrae y procesa la humedad del suelo, humedad del aire, temperatura del aire y estado de la bomba desde los datos recibidos.
Fin del proceso de obtención de datos	Los datos procesados están listos para ser utilizados en el bucle principal del programa.

El programa ejecutado en la Raspberry Pi opera en el lenguaje de programación Python y está diseñado para recibir datos provenientes del microcontrolador. Para desarrollar y ejecutar este programa, se utiliza un Entorno de Desarrollo Integrado (IDE) específico para Python. La comunicación con el microcontrolador Arduino se establece a través del puerto serie, permitiendo la transferencia de datos desde el dispositivo.

Una vez que la Raspberry Pi ha recibido los datos, el programa inicia un proceso de análisis. Este análisis implica el uso de algoritmos y lógica de programación para realizar cálculos y comparaciones. Además, se pueden establecer umbrales o condiciones específicas. Por ejemplo, el programa puede evaluar si la temperatura registrada ha alcanzado un nivel crítico. En caso afirmativo, se pueden tomar decisiones automáticas, como activar o desactivar la bomba de agua según la situación detectada. Este enfoque proporciona una capacidad de automatización basada en la interpretación de los datos recibidos, permitiendo respuestas inmediatas y específicas ante diferentes escenarios.

3.2.5. Diagrama de conexión

El diagrama global del sistema local, basado en las interconexiones establecidas, se compone de diversos componentes interrelacionados. En primer lugar, se encuentran los dispositivos

de detección. El Microcontrolador funciona como el punto de adquisición y control de datos, recibiendo señales de los sensores y transmitiéndolas a la Raspberry Pi mediante una conexión serial. En este contexto, también se encuentra conectado un relé al microcontrolador Arduino Nano. La Raspberry Pi es el dispositivo central del sistema, procesando los datos recibidos del Arduino Nano mediante un programa en Python. Este programa se encarga de leer, calcular y enviar los datos en una base de datos de tiempo real. La Raspberry Pi toma decisiones de control basadas en los datos adquiridos, como activar o desactivar la bomba de riego. En la figura 3.7 se muestra toda la conexión del sistema local.

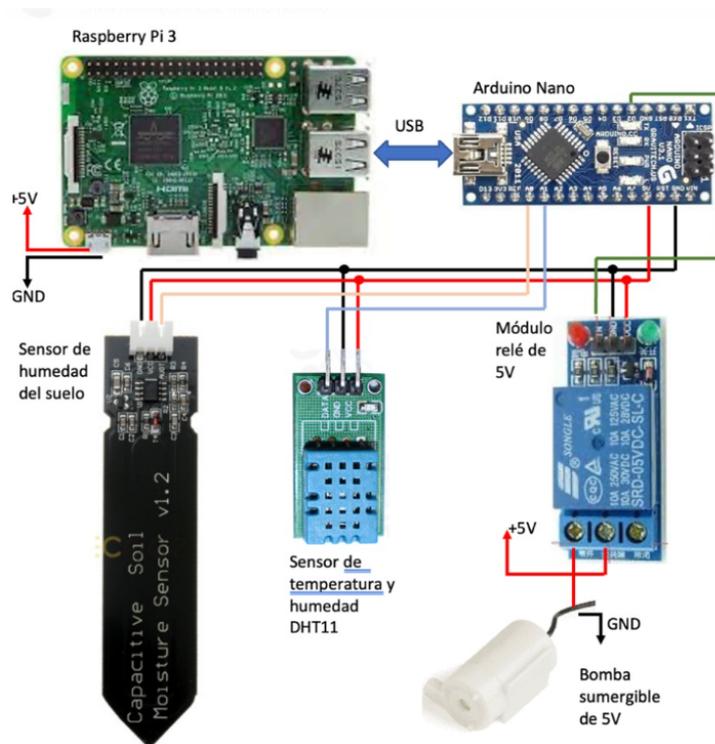


Figura 3.7: Diagrama general de conexiones.

3.3. Integración en la nube

En este apartado se aborda la integración de bases de datos para la adquisición y procesamiento de variables ambientales. Se explorará cómo se establece una conexión entre el sistema local y una infraestructura en la nube, permitiendo el almacenamiento y acceso a los datos de manera remota.

Se examina la importancia de utilizar la nube como plataforma centralizada para recolectar y gestionar los datos ambientales recopilados por los sensores. Además, se describirá el proceso

de preparación de los datos para su posterior análisis y toma de decisiones.

3.3.1. Base de datos de tiempo real

En este escenario, la base de datos de tiempo real empleada constituye un servicio provisto por Firebase, conocido como Realtime Database. Se trata de una base de datos NoSQL alojada en la nube que posibilita el almacenamiento y la sincronización de datos en tiempo real entre distintos usuarios. Su función principal radica en mantener la información actualizada y proporcionarla en tiempo real. El microcomputador desempeña un papel crucial en este proceso, ya que es el encargado en primera instancia de enviar los valores de las variables ambientales y de establecer la ruta.

Dentro del programa del microcomputador, existe un módulo encargado de transmitir información mediante un enlace URLDB, aquí es donde se especifica el enlace. Además, desde el código, se establece la ruta correspondiente. Asimismo, el programa utiliza la función necesaria para emplear paquetes de Firebase, facilitando así el envío y la recepción de datos. En este contexto, la información enviada incluye variables que representan datos ambientales, condiciones climáticas y el estado de la bomba.

Para proporcionar una visualización más clara de este proceso, la figura 3.8 muestra cómo se están almacenando temporalmente los datos en la base de datos de Firebase. Se observa las variables mencionadas anteriormente y cómo sus valores se actualizan en tiempo real.



Figura 3.8: Almacenamiento de variables y datos temporales.

La mecánica completa de cómo el microcomputador envía información a la base de datos de Firebase se encuentra en la configuración del programa correspondiente. La base de datos de Firebase opera como un buffer, reemplazando y actualizando la información cada vez que se envía un nuevo conjunto de datos. En consecuencia, cuando se requiere acceder a información específica, se solicita a esta base de datos, la cual proporciona datos siempre actualizados.

El servidor también juega un rol fundamental en este ecosistema. Posee un programa dedicado que envía información a la misma base de datos de Firebase, utilizando la URLDB y la ruta previamente definida. La aplicación sigue un proceso análogo, contribuyendo así a un sistema interconectado donde todos los componentes teóricamente se comunican entre sí de manera eficiente. Este entramado de comunicación garantiza la coherencia y actualización constante de la información en tiempo real, mejorando así la eficacia y confiabilidad del sistema en su conjunto.

En este contexto, es fundamental destacar que los datos actualizados que se almacenan en la base de datos de Firebase también se transmiten y registran en una base de datos local dentro del servidor. Este proceso se lleva a cabo mediante otro programa específico diseñado para recoger la información de Firebase y almacenarla en la base de datos interna del servidor en intervalos de tiempo determinados.

El servidor, además de su papel en la comunicación con la base de datos de Firebase, opera con un programa adicional encargado de capturar la información proveniente de Firebase y guardarla en su propia base de datos local. Esta práctica estratégica garantiza no solo la disponibilidad de datos en tiempo real desde Firebase, sino también su retención a lo largo del tiempo en la base de datos local del servidor.

3.4. Integración de dispositivos remotos

Este apartado se enfocará en la implementación de un sistema de integración en tiempo real entre dispositivos remotos.

3.4.1. Servidor

En el servidor, se aloja la base de datos SQLite, encargada de almacenar información crucial para el sistema de riego automático. En primer lugar, el servidor recopila la información para ello se establece una transferencia automática y continua de datos desde Firebase hacia la base de datos SQLite. Esto posibilita un almacenamiento eficiente y una gestión óptima de la información recopilada, asegurando un acceso actualizado para tomar decisiones efectivas en el

control del riego. Posteriormente, una vez que los datos de las variables ambientales están almacenados, se inicia el proceso de aprendizaje de máquina. Este procedimiento utiliza el modelo de aprendizaje más adecuado, consolidando así un sistema integral que no solo almacena datos eficientemente, sino que también emplea el aprendizaje automático para mejorar continuamente el rendimiento del sistema de riego.

3.4.1.1. Base de datos SQLite

En este apartado se abordará la planificación y configuración de una base de datos SQL en un servidor local para almacenar y administrar los datos relacionados con el sistema IoT de riego inteligente, se buscará garantizar un diseño eficiente y seguro que cumpla con los requisitos de almacenamiento y gestión de datos del sistema.

Para abordar eficientemente la obtención y almacenamiento de información, se inicia con la configuración de una base de datos SQLite, estableciendo así la base sobre la cual se recopilará y almacenará la información necesaria. La fuente de datos proviene de una base de datos en tiempo real de Firebase. Para comprender este proceso, es crucial explorar la sincronización entre la base de datos de Firebase y la de SQLite, asegurando que la información recopilada esté siempre actualizada y refleje con precisión los cambios en ambas bases de datos. Una vez comprendido el funcionamiento de la sincronización, el enfoque se desplaza hacia la forma en que la base de datos SQLite almacena la información. Es imperativo destacar que este proceso de obtención de datos desde Firebase y su almacenamiento en SQLite se ejecutan a través de un solo programa. Este proceso garantiza una eficiencia óptima y una gestión simplificada, actuando el programa como el vínculo cohesionado entre ambas bases de datos, facilitando la obtención de datos en tiempo real desde Firebase y su almacenamiento en SQLite de manera fluida y eficiente.

3.4.1.2. Sincronización entre Firebase y la base de datos SQL

En la implementación de la sincronización automática entre Firebase y la base de datos SQL, se busca el establecimiento de un proceso continuo de intercambio de datos entre los dos sistemas, con el propósito de asegurar la actualización y coherencia de la información almacenada. Esta etapa es esencial para garantizar que los datos recopilados por los sensores de humedad y temperatura del aire, así como la humedad del suelo, se reflejen simultáneamente en la base de datos local SQL y en Firebase en tiempo real.

1. **Configuración de las credenciales:** Se deben obtener las credenciales de Firebase, que permitirán establecer la conexión y autorización necesarias para la sincronización. Esto

implica generar y obtener el archivo JSON de configuración de Firebase, que contiene la información de autenticación y acceso a la base de datos

2. **Establecimiento de la conexión con Firebase:** Utilizando el archivo JSON de configuración y las bibliotecas o SDK adecuados, se debe establecer una conexión con Firebase desde el entorno de ejecución donde se encuentra la base de datos local. Esto puede implicar importar las bibliotecas necesarias y configurar la autenticación utilizando las credenciales proporcionadas.
3. **Monitoreo de cambios en Firebase:** Una vez establecida la conexión, se debe implementar un mecanismo para monitorear los cambios en la base de datos de Firebase en tiempo real. Esto se puede lograr utilizando funciones o eventos específicos proporcionados por las bibliotecas de Firebase, que permiten detectar adiciones, actualizaciones o eliminaciones de datos en la base de datos.
4. **Actualización de la base de datos local SQL:** Cada vez que se detecte un cambio en Firebase, ya sea la adición, actualización o eliminación de datos, se debe reflejar ese cambio en la base de datos local SQL. Esto implica capturar los datos modificados desde Firebase y realizar las operaciones correspondientes en la base de datos local utilizando sentencias SQL. Por ejemplo, para añadir un nuevo registro, se deberá realizar una inserción en la tabla "History" con los valores correspondientes.
5. **Manejo de errores y sincronización periódica:** Durante la sincronización automática, es importante considerar la posibilidad de errores de conexión o fallas en la comunicación entre los sistemas. Se debe implementar un manejo adecuado de errores y realizar intentos de sincronización periódicos para asegurarse de que los datos estén actualizados en ambos sistemas.

3.4.1.3. Requisitos de almacenamiento

En este caso, los requisitos implican el almacenamiento de datos de humedad del suelo, humedad del aire y temperatura del aire en una base de datos SQLite local, en la figura 3.9 se muestra el diagrama de flujo que sigue el programa para generar y almacenar los datos.

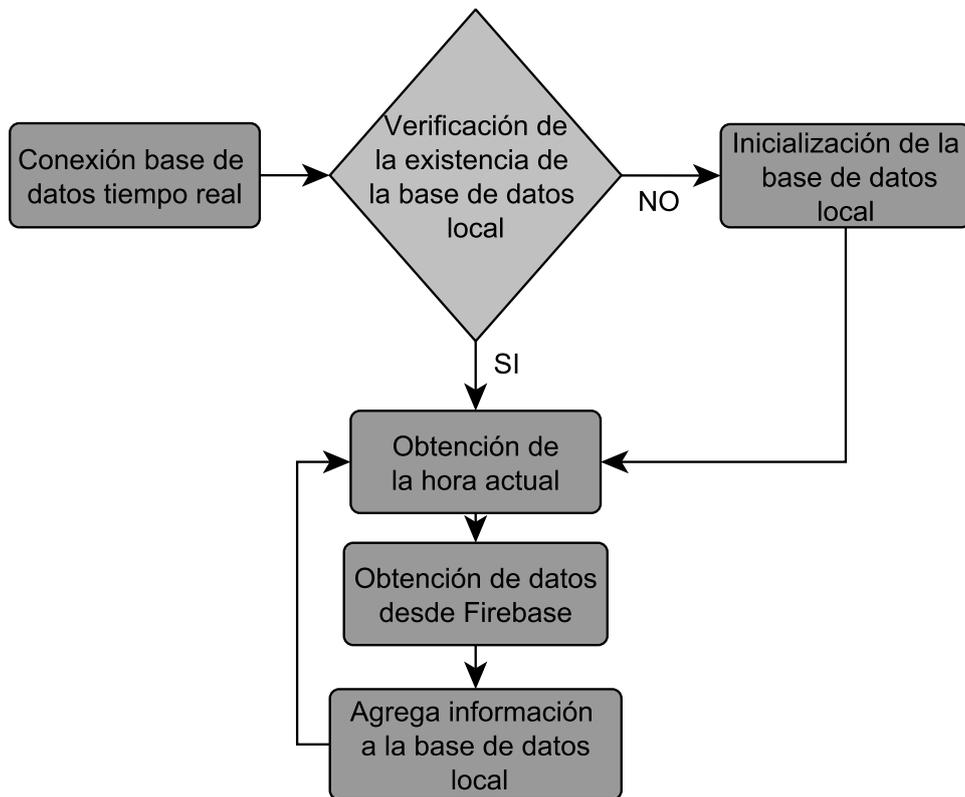


Figura 3.9: Diagrama de flujo para generar base de datos local.

Se proporcionan detalles sobre aspectos importantes relacionados con el almacenamiento de datos, específicamente abordando las características que se almacenarán en la base de datos SQLite.

1. **Tipo de datos:** Se define el tipo de datos necesarios para almacenar la humedad del suelo, humedad del aire y temperatura del aire. Se considerará utilizar tipo de datos numéricos, decimales y enteros para las mediciones de humedad y temperatura.
2. **Escala y precisión:** Determinar la escala y precisión necesaria para los datos de humedad y temperatura. Se especifica cuántos decimales se requieren para la humedad del suelo o si la temperatura debe almacenarse en grados Celsius o Fahrenheit.

3. **Volumen de datos:** Estimar el volumen de datos que se espera almacenar en la base de datos a lo largo del tiempo. Esto ayuda a definir la capacidad y escalabilidad requerida para el servidor local y la base de datos.
4. **Frecuencia de actualización:** Definir la frecuencia con la que se recibirán nuevos datos de humedad del suelo, humedad del aire y temperatura del aire. Esto influye en el diseño de la sincronización entre Firebase y la base de datos SQL.
5. **Acceso y consultas:** Determinar los requisitos de acceso y las consultas que se realizarán sobre los datos almacenados. Por ejemplo, si se necesita realizar consultas para obtener el promedio de la humedad del suelo en un período determinado.
6. **Integración con Firebase:** Debido a que se va a implementar la sincronización automática entre Firebase y la base de datos SQLite, es importante considerar la compatibilidad y los requisitos técnicos para establecer la conexión y garantizar la transferencia correcta de datos entre ambos sistemas.

3.4.1.4. Análisis de entidades y relaciones

1. Entidad: History

Atributos

- EstampaTiempo: Representa el momento en el que se registró la muestra de datos.
- HumedadSuelo: Valor de humedad del suelo en la muestra de datos.
- HumedadAire: Valor de humedad del aire en la muestra de datos.
- TemperaturaAire: Valor de temperatura del aire en la muestra de datos.

En este caso, la entidad "History" representa el historial de datos capturados de humedad del suelo, humedad del aire y temperatura del aire en diferentes momentos. Los atributos de la entidad representan las mediciones específicas en cada registro.

3.4.1.5. Aprendizaje automático

Para crear un modelo de aprendizaje automático para predecir el momento de riego autónomo de un cultivo, se recopilan datos de una base de datos, se eliminan los datos atípicos y los valores nulos, se etiquetan los datos de riego y se dividen los datos en un conjunto de entrenamiento y un conjunto de prueba. El modelo se entrena utilizando el conjunto de entrenamiento y

se evalúa utilizando el conjunto de prueba, en la figura 3.10 se muestra el diagrama de procesos que sigue el modelo.

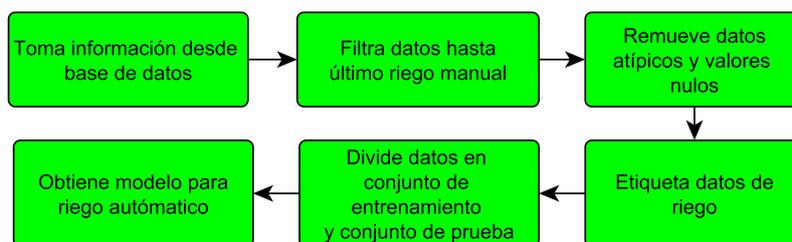


Figura 3.10: Diagrama de la forma de actuar el servidor.

3.4.1.6. Recopilación y almacenamiento de datos

La base de cualquier sistema de aprendizaje automático eficaz es la disponibilidad de datos relevantes y precisos. En nuestro caso, esto implica la recopilación constante de información sobre las condiciones del entorno y las plantas en nuestras macetas. Sensores de humedad del suelo, humedad relativa y temperatura del ambiente se encargan de capturar estos datos y enviarlos a una base de datos centralizada en un servidor. La riqueza de esta base de datos es esencial, ya que proporciona al sistema la información necesaria para aprender patrones y tomar decisiones.

3.4.1.7. Preprocesamiento de datos

Una vez que los datos se encuentran en la base de datos, el siguiente paso es el preprocesamiento. Esto implica limpiar los datos, identificar y corregir valores atípicos o faltantes, y prepararlos para su análisis. En nuestro caso, es crucial garantizar que los datos sean coherentes y de alta calidad, ya que cualquier error en esta etapa afectaría negativamente la precisión del sistema de aprendizaje automático.

3.4.1.8. Selección de algoritmos de aprendizaje automático

Con los datos preparados y enriquecidos, llegamos a la elección de los algoritmos de aprendizaje automático adecuados. Aquí es donde la versatilidad y el conocimiento profundo del aprendizaje automático entran en juego. En nuestro proyecto, hemos evaluado varias técnicas, incluyendo regresión lineal, redes neuronales y máquinas de soporte vectorial, para determinar cuáles son más apropiadas para predecir las necesidades de riego en función de las lecturas de

sensores. Cada algoritmo tiene sus propias fortalezas y debilidades, y es esencial entender cómo funcionan y cuándo aplicarlos.

3.4.1.9. Entrenamiento y validación de modelos

Una vez seleccionados los algoritmos, procedemos al entrenamiento de modelos. Esto implica el uso de datos históricos para enseñar al sistema cómo debe interpretar los datos de los sensores y tomar decisiones de riego. Pero entrenar no es suficiente; también debemos validar los modelos para asegurarnos de que sean precisos. La validación implica utilizar datos reales de riego y comparar las decisiones tomadas por el sistema con las decisiones óptimas. Solo cuando estamos seguros de que los modelos son precisos procedemos a la fase de implementación.

3.4.1.10. Implementación y adaptación continua

Finalmente, implementamos los modelos de aprendizaje automático en nuestro sistema IoT de riego inteligente. Es aquí donde el sistema comienza a operar en tiempo real y a tomar decisiones de riego basadas en el aprendizaje previo. Pero la implementación no marca el final de nuestro proceso, sino el comienzo de una fase de adaptación continua. A medida que el sistema recopila más datos y aprende, sus modelos pueden ajustarse y mejorar. Este ciclo de retroalimentación continua es lo que hace que nuestro sistema de riego sea verdaderamente inteligente.

3.4.2. Aplicación web

La aplicación se diseñó con un enfoque centrado en el usuario, donde la interacción aparente se entrelaza con un proceso interno meticulosamente estructurado. Al interactuar con la interfaz, el usuario desencadena eventos que son hábilmente capturados y procesados por un motor central. Este motor interpreta las acciones del usuario, traduciéndolas en comandos que desencadenan un proceso interno de procesamiento de datos. A través de consultas a bases de datos y ejecución de operaciones específicas, la aplicación genera respuestas coherentes y eficientes.

Para facilitar la comprensión, se presenta la figura 3.11 que consiste en un diagrama detallado que visualiza la arquitectura interna de la aplicación. Este diagrama destaca los procesos clave, ilustrando la interconexión que da vida al sistema. Esta representación visual refuerza la

idea de una aplicación robusta y eficaz, proporcionando una conexión tangible entre la experiencia del usuario y la complejidad interna que la respalda.

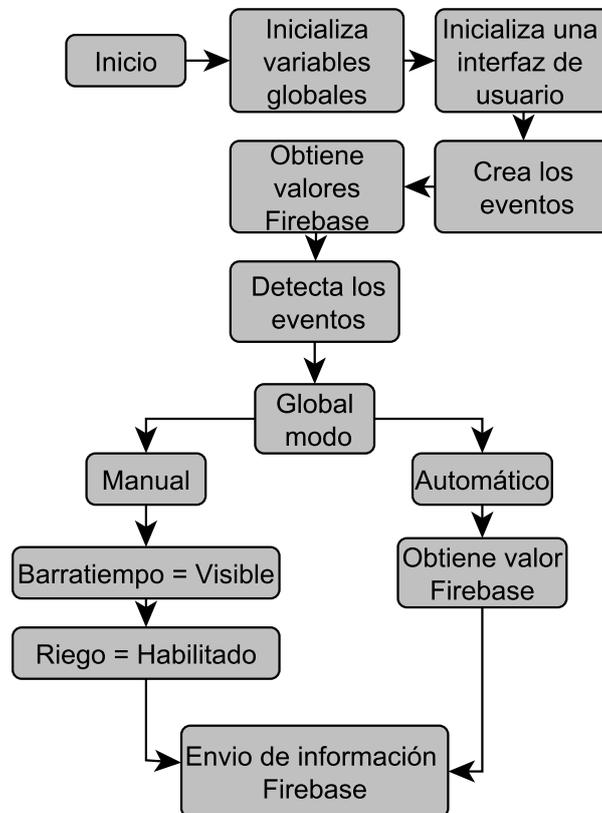


Figura 3.11: Trama de la lógica de la aplicación.

Tras comprender a fondo los procesos internos de la aplicación, es esencial explorar las características que definen su creación. Entre las particularidades destacadas se encuentra la elección cuidadosa de herramientas y tecnologías que sirven como cimiento para el desarrollo.

- **Acceso a la interfaz gráfica:** Utilizando el entorno de desarrollo de MIT App Inventor, se ha accedido a la sección de diseño de la aplicación, que permite la creación y configuración de la interfaz gráfica.
- **Selección y disposición de componentes visuales:** Se han seleccionado los componentes visuales necesarios para la interfaz, como botones, etiquetas, campos de entrada, listas, etc. Estos componentes se arrastran y se colocan en el área de diseño, permitiendo su disposición y alineación adecuada.
- **Configuración de propiedades de los componentes:** Cada componente visual tiene diversas propiedades que se pueden configurar, como el texto que se muestra, el tamaño,

el color, la fuente, entre otras. Estas propiedades se han ajustado según las necesidades y requisitos de la aplicación.

- **Definición de eventos y acciones:** Se han establecido los eventos que desencadenarán acciones en la aplicación. Por ejemplo, se ha configurado un botón para que al hacer clic en él, se realice una determinada acción, como enviar datos a Firebase o actualizar la visualización de datos.
- **Diseño y organización visual:** Se ha prestado atención a la estética y la organización visual de la interfaz. Se han utilizado técnicas de diseño para asegurar una disposición intuitiva y agradable de los componentes visuales, facilitando la interacción del usuario.
- **Adaptación a distintos dispositivos:** Se ha considerado la adaptabilidad de la interfaz a diferentes tamaños de pantalla y dispositivos. MIT App Inventor ofrece herramientas para facilitar el diseño responsive, asegurando que la aplicación sea accesible en diversos dispositivos móviles. La figura 3.12 muestra el aplicativo web con el cual el usuario va a interactuar.



Figura 3.12: Interfaz web.

Capítulo IV

Resultados y análisis

Este capítulo presenta el análisis y resultados del proyecto, comenzando con la integración de bases de datos, seguido por la propuesta de arquitectura de hardware. Finalmente, se aplican técnicas de aprendizaje de máquina sobre el sistema desarrollado.

4.1. Implementación del enlace entre bases de datos

En esta sección, se describe el diseño del modelo que permite la sincronización entre la base de datos de tiempo real y la base de datos SQL en el contexto del sistema de riego inteligente. Se establece el propósito y la relevancia de esta integración.

4.1.1. Enlace entre la base de datos de tiempo real y la base de datos SQL

Se presenta en detalle la implementación del modelo de enlace entre bases de datos, lo cual constituye un componente esencial para la gestión eficiente de los datos ambientales. Se describe cómo esta implementación permite la transferencia y el almacenamiento efectivo de los datos capturados en el programa.

La integración de Firebase y una base de datos local SQLite permite establecer una conexión sólida y bidireccional entre ambas bases de datos. Esta integración es útil para transferir y procesar datos entre la aplicación y Firebase.

- **Función de envío de información:** Esta función facilita la transferencia de datos desde la base de datos de tiempo real de Firebase hacia la base de datos local. Su objetivo principal es asegurar que los datos relevantes, como la humedad del aire, humedad del suelo y temperatura del aire, sean transmitidos correctamente para su posterior procesamiento y almacenamiento.

- **Función de obtención de información:** La función de obtención de información se utiliza para recuperar los datos almacenados en la base de datos de tiempo real de Firebase y transferirlos a la base de datos local. Esta función es esencial para garantizar que los datos actualizados y relevantes estén disponibles en la aplicación en todo momento.

La integración de Firebase y una base de datos local SQLite permite la sincronización de datos entre la aplicación y la nube. Esto facilita el almacenamiento, la recuperación y el procesamiento de información, proporcionando una manera eficiente y confiable de compartir información e interactuar con los datos almacenados.

4.1.2. Gestión de datos temporales y etiquetado preciso

El proceso de sincronización temporal es fundamental, ya que asegura una alineación precisa en el tiempo de cada conjunto de datos adquiridos. Esto se logra mediante el uso de una función de lectura de tiempo que marca cada conjunto de datos con una marca de tiempo precisa. Esta etapa de sincronización temporal es esencial tanto en Firebase como en la base de datos local, garantizando que los datos estén organizados de manera coherente.

Además, este proceso de sincronización temporal se relaciona con varios conceptos clave en la gestión de datos:

1. **Sincronización de datos:** Proceso global de asegurar que los datos en la aplicación y en la nube estén siempre actualizados y en consonancia entre sí. La sincronización temporal contribuye directamente a este proceso, ya que establece un marco temporal común para la organización y actualización de los datos.
2. **Almacenamiento de datos:** Una vez que los datos han sido marcados temporalmente, se almacenan en ubicaciones específicas, ya sea en la base de datos local o en la nube de Firebase. Esta acción de almacenamiento es vital para garantizar la persistencia y disponibilidad de los datos a lo largo del tiempo.
3. **Recuperación de datos:** La alineación temporal precisa permite una recuperación efectiva de los datos almacenados. Esto implica la capacidad de acceder y obtener los datos según las marcas de tiempo, facilitando su posterior uso o visualización.
4. **Procesamiento de información:** Los datos adquiridos pueden no ser útiles por sí solos. Aquí es donde entra en juego el procesamiento de información. Mediante operaciones y manipulaciones, los datos se transforman en información significativa y útil. La sincronización temporal previa garantiza que los datos utilizados en estas operaciones estén alineados correctamente.

5. **Compartir información:** La capacidad de intercambiar datos entre usuarios o sistemas es esencial para la colaboración y la transferencia de conocimientos. La sincronización temporal asegura que los datos compartidos estén en un contexto temporal coherente, evitando confusiones o desalineaciones.
6. **Interacción con los datos almacenados:** La interacción con los datos almacenados abarca operaciones, consultas y modificaciones. Ya sea mediante consultas SQL o la API de Firebase, la sincronización temporal garantiza que estas interacciones se realicen con datos consistentes en términos de tiempo.

4.1.3. Almacenamiento efectivo y referencia histórica

En este enfoque, los datos recopilados se almacenan de manera eficiente en una base de datos local. Se utiliza una función específica para insertar los valores de los datos recopilados en una tabla designada. Este método de almacenamiento permite un acceso rápido a los datos y proporciona una referencia histórica para análisis y comparaciones futuras.

4.1.3.1. Almacenamiento efectivo y referencia histórica mediante SQLite

1. **SQLite como base de datos ligera y autónoma:** SQLite base de datos ligera y autónoma que se destaca por su facilidad de implementación y su capacidad para manejar cantidades significativas de información de manera eficiente.
2. **Recopilación de datos y estrategia de almacenamiento efectivo:** La estrategia se basa en la premisa de que la recopilación de datos es esencial para tomar decisiones informadas y mejorar la eficacia de las operaciones. Los datos recopilados se insertan en una tabla específicamente diseñada en la base de datos SQLite, asegurando la integridad y la coherencia de los datos almacenados.
3. **Referencia histórica y seguimiento del progreso:** A medida que se acumulan datos con el tiempo, la base de datos SQLite se convierte en una invaluable fuente de referencia histórica. Esto permite a los usuarios rastrear el progreso, identificar tendencias y realizar comparaciones a lo largo del tiempo.
4. **Uso adecuado de la memoria:** SQLite tiene la capacidad de almacenar en caché parte de la base de datos en la memoria para acelerar las operaciones. Asegúrate de ajustar correctamente los parámetros de memoria para aprovechar al máximo esta funcionalidad.

4.1.4. Estructura y tipos de datos en SQLite

Los datos enviados a una base de datos SQLite han sido organizados mediante una estructura de tabla cuidadosamente diseñada. Esta estructura sirve como marco organizativo sólido que facilita tanto el almacenamiento como el acceso a los datos. Cada columna en la tabla ha sido diseñada para reflejar con precisión las distintas variables presentes en los datos obtenidos del sistema de riego inteligente.

1. **Asignación de tipos de datos:** Asignación de Tipos de Datos: Se han asignado tipos de datos apropiados a cada columna con el fin de garantizar la integridad de los datos almacenados. En el caso de las marcas de tiempo, se ha optado por el tipo REAL y se utiliza el sistema Unix. Esta elección se debe a que el tipo de dato real permite una representación más precisa de los valores numéricos asociados con las marcas de tiempo. Además, el sistema Unix es ampliamente utilizado y reconocido por su precisión y confiabilidad en la gestión de fechas y horas. Por lo tanto, al utilizar el sistema Unix en conjunto con el tipo de dato real, se asegura una mayor precisión y consistencia en las marcas de tiempo almacenadas. Esta combinación de tipo de dato y sistema operativo ofrece beneficios significativos en términos de precisión y facilidad de uso en el contexto de las marcas de tiempo.
2. **Humedad del suelo, humedad del aire y temperatura:** Se ha realizado una elección cuidadosa del tipo de dato para estas mediciones cruciales. Dependiendo de la magnitud de los valores reales, se han empleado opciones como INTEGER o REAL de manera adecuada. En situaciones donde los valores sean excepcionalmente grandes, se ha llevado a cabo una corrección de escala antes de almacenarlos para mantener la precisión.

Es importante mencionar que se ha utilizado el tipo de dato REAL para estas mediciones, lo cual permite representar valores decimales y mantener la precisión necesaria, en la figura 4.1 se muestra como están almacenados los datos.

	ID	EstampaTiempo	HumedadSuelo	HumedadAire	TemperaturaAire
	Filtro	Filtro	Filtro	Filtro	Filtro
1	1	1689715043.76243	42.0	66.0	23.0
2	2	1689716844.15294	42.0	66.0	23.0
3	3	1689718644.58168	42.0	66.0	23.0
4	4	1689720444.92537	40.0	66.0	23.0
5	5	1689722245.37553	40.0	66.0	22.0
6	6	1689724045.81081	38.0	65.0	22.0
7	7	1689725846.25057	33.0	65.0	22.0
8	8	1689727646.66818	33.0	65.0	22.0
9	9	1689729447.09469	33.0	65.0	22.0
10	10	1689731247.52727	33.0	66.0	22.0

Figura 4.1: Estructura de base de datos SQLite

4.2. Arquitectura de hardware

En esta sección, se presenta la arquitectura de hardware diseñada para validar el sistema. El objetivo principal de esta etapa es demostrar la viabilidad y eficacia del enlace entre las bases de datos y la aplicación de aprendizaje automático en tiempo real. Además, esta instancia servirá como fuente primordial para la adquisición de datos relativos a las variables ambientales.

4.2.1. Hardware implementado

Se proporciona una descripción detallada de la estructura de hardware utilizada para la prueba de concepto. Esto incluye la disposición de las macetas con las plantas, los sensores empleados para medir la humedad del suelo, humedad relativa y temperatura del aire, así como cualquier otra tecnología relevante integrada en el sistema, en la figura 4.2 se muestra el modelo de hardware implementado.

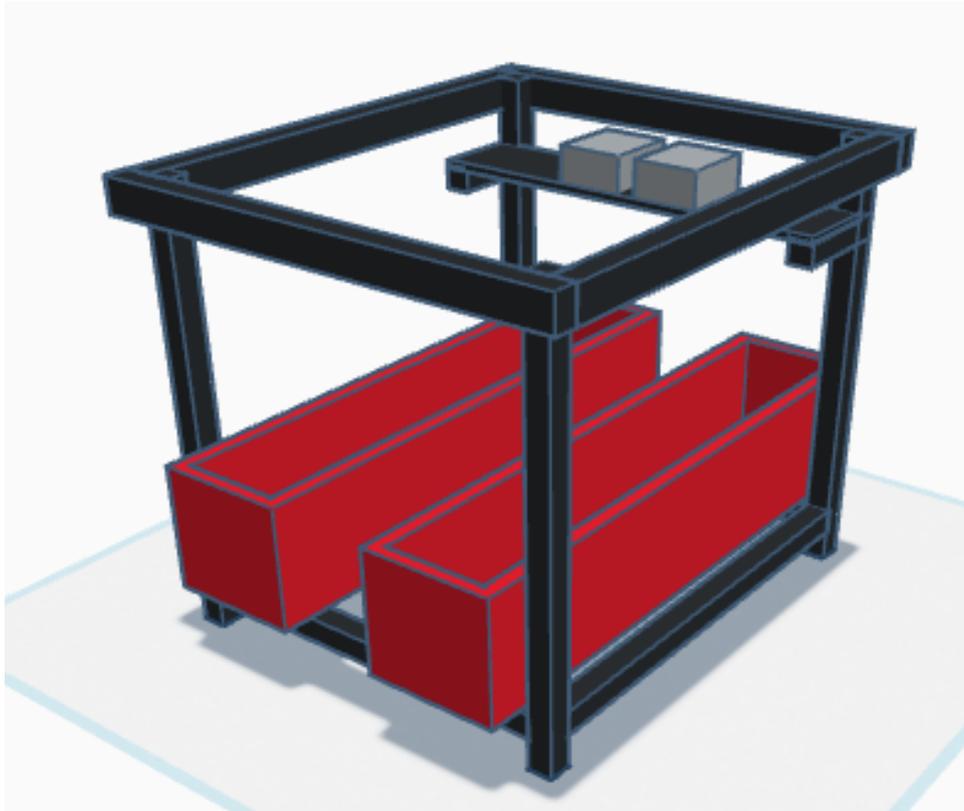


Figura 4.2: Implementación hardware

La estructura está compuesta por dos marcos, uno en la parte superior y otro en la parte inferior, ambos con forma rectangular. Entre estos marcos, se encuentra un espacio abierto. En la mitad de la parte inferior, hay un tubo horizontal que divide la anchura del marco en dos partes iguales. En el marco inferior, se ubican dos macetas con una separación de aproximadamente 5 cm entre ellas. Las macetas tienen dimensiones de 75 cm de largo por 55 cm de ancho. Esta disposición proporciona una estructura resistente y funcional para la presentación de las macetas.

Aproximadamente a 11 cm por debajo del marco superior, se encuentran dos tubos horizontales paralelos, que se extienden a lo largo de la estructura. Estos funcionan como una base adicional y ofrecen soporte para la colocación de una plataforma o superficie en la que se encuentran alojadas cajas IP55 especialmente diseñadas para proteger los componentes electrónicos de las condiciones ambientales adversas. Estas cajas IP55 albergan todos los elementos clave del sistema, que incluye los dispositivos de control para el riego.

Los sensores están conectados mediante cables a la unidad central de procesamiento y control. Estos cables son lo suficientemente largos para garantizar la ubicación estratégica de los sensores en las macetas sin comprometer la conectividad.

4.2.2. Validación del hardware

Se describe cómo el hardware implementado ha sido validado en la práctica. Proporcionando detalles sobre las pruebas que se realizó para asegurar que los sensores y dispositivos funcionen según lo esperado.

4.2.2.1. Pruebas de funcionamiento

Las pruebas de funcionamiento se llevaron a cabo en un entorno controlado que simula las condiciones reales del sistema de riego. Durante estas pruebas, se verificó que los sensores recopilaban datos de manera consistente y que estos datos se transmitían correctamente a la unidad central de procesamiento. Además, se evaluó la capacidad de los dispositivos de control para activar el riego de manera precisa y oportuna en respuesta a las lecturas de los sensores. Esto implicó simular diferentes niveles de humedad del suelo y condiciones de temperatura ambiente para evaluar la capacidad de respuesta del sistema, en la figura 4.3 se muestra la ubicación del sensor para las pruebas de funcionamiento.

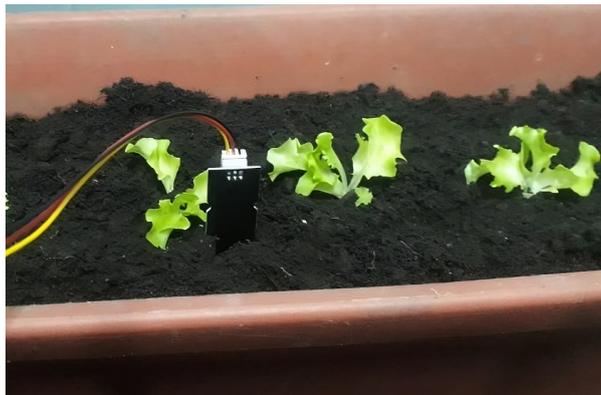


Figura 4.3: Funcionamiento del sensor de humedad del suelo.

4.2.2.2. Pruebas de precisión y estabilidad

La precisión y estabilidad de los sensores se sometieron a pruebas específicas. Se compararon las lecturas de los sensores con mediciones de referencia para determinar la precisión de las mediciones de humedad del suelo, humedad relativa y temperatura ambiental. Las pruebas de estabilidad se centraron en verificar que los sensores mantuvieran su funcionamiento óptimo a lo largo del tiempo y bajo diferentes condiciones ambientales. Se realizaron pruebas de resistencia a la humedad y a las variaciones de temperatura para asegurarse de que los sensores

podieran funcionar de manera confiable a lo largo de las estaciones y bajo diversas circunstancias climáticas.

Para demostrar la precisión y estabilidad de nuestro sistema, es fundamental analizar la variación de medidas a lo largo del tiempo. En este proceso, se emplean medidas de un instrumento patrón como referencia para garantizar la exactitud de las mediciones. En nuestro caso, nos enfocamos en la variación de medidas para diversas variables críticas, como temperatura, humedad del aire y humedad del suelo. Cada cierto intervalo de tiempo, hemos registrado medidas específicas, como se detalla en la tabla 4.1 adjunta, proporcionando una representación clara y sistemática de la evolución de cada variable.

Tabla 4.1: Tabla de variación de medidas en intervalos de tiempo.

Tiempo (Minutos)	Temperatura (°C)	Humedad aire (%)	Humedad suelo (%)
8	18.5	64	15.5
15	19.2	66	16.2
22	18.8	65.5	15.7
30	19.5	64.5	16.3
40	20	65	16
50	19.8	65.2	15.8
60	18.7	64.8	15.3
75	19.3	65.5	16.1
90	19.9	65	15.9

Una vez registrados los datos, llevamos a cabo un análisis coherente que se detalla de manera exhaustiva en la tabla 4.2. Esta tabla presenta los resultados derivados de los valores recopilados en la tabla 4.1, destacando la precisión y estabilidad de nuestro sistema.

Tabla 4.2: Tabla de resultados de los parámetros de los sensores.

Parámetro	Sensor DHT11 (Temperatura)	Sensor DHT11 (Humedad relativa)	Sensor de humedad de suelo
Precisión	± 2 °C	± 5 %	± 2 %
Rango de medición	18.5 °C a 20 °C	64 % a 66 %	15 % a 16.3 %
Error máximo	1.5 °C	2 %	1.3 %
Conclusión	Preciso, dentro del rango de precisión	Preciso, dentro del rango de precisión	Preciso, dentro del rango de precisión
Estabilidad	Variaciones ligeras dentro del rango aceptable	Variaciones ligeras dentro del rango aceptable	Variaciones ligeras dentro del rango aceptable

4.2.2.3. Evaluación del rendimiento

Esta evaluación exhaustiva del hardware ha respaldado la funcionalidad sólida y coherente de nuestro sistema de riego inteligente. Los resultados indican que el hardware es capaz de recopilar datos precisos, mantenerse estable en diversas condiciones y activar el riego de manera eficiente en respuesta a las necesidades de las plantas en las macetas. Estos hallazgos son un fundamento crucial para el éxito de nuestro sistema, destacando su capacidad para mejorar la gestión del riego agrícola y optimizar la utilización de recursos hídricos en el campo, en la figura 4.4 se muestra el sistema completo que se utilizó para evaluar el rendimiento.



Figura 4.4: Funcionalidad del sistema.

4.2.3. Análisis de datos

La humedad del suelo, la humedad del aire y la temperatura son tres variables ambientales que están estrechamente relacionadas entre sí. La humedad del suelo es la cantidad de agua que se encuentra en el suelo, la humedad del aire es la cantidad de vapor de agua que se encuentra en la atmósfera, y la temperatura es la medida de la energía cinética promedio de las partículas de un sistema.

Se analizaron los datos recopilados para identificar patrones, tendencias o relaciones entre las variables. Los datos se obtuvieron de sensores que se instalaron en un campo de cultivo. Los resultados muestran que los valores de humedad del suelo se encuentran en un rango de 10 a 20 %, los valores de humedad del aire se encuentran en un rango de 60 a 70 %, y los valores de temperatura se encuentran en un rango de 20 a 30 grados Celsius.

4.2.3.1. Patrones y tendencias

Se observó que la humedad del suelo y la humedad del aire tienden a ser inversamente proporcionales. Cuando la humedad del aire es alta, la humedad del suelo tiende a ser baja. Esto se debe a que el agua se evapora del suelo y entra a la atmósfera. Por otro lado, cuando la humedad del aire es baja, la humedad del suelo tiende a ser alta.

También se observó que la humedad del suelo y la temperatura tienden a ser directamente proporcionales. Cuando la temperatura es alta, la humedad del suelo tiende a ser baja. Esto se debe a que el agua se evapora del suelo más rápidamente a temperaturas más altas. Por otro lado, cuando la temperatura es baja, la humedad del suelo tiende a ser alta.

4.2.3.2. Implicaciones para el riego de plantas

Los datos recopilados tienen implicaciones importantes para el riego de plantas. Para que las plantas crezcan de manera saludable, es necesario que el suelo tenga un nivel adecuado de humedad. Sin embargo, si el suelo tiene demasiada humedad, las raíces de las plantas pueden pudrirse.

Basándose en los patrones y tendencias observados, se puede concluir que es mejor regar las plantas cuando la humedad del aire es baja y la temperatura es alta. En estas condiciones, el agua del suelo no se evaporará tan rápidamente, lo que ayudará a evitar la pudrición de las raíces.

4.2.3.3. Tabla comparativa

Tabla 4.3: Datos de humedad del suelo, humedad del aire y temperatura.

Día	Hora	Humedad del suelo (%)	Humedad del aire (%)	Temperatura (°C)
1	8:00 AM	15	70	20
1	12:00 PM	15	65	15
1	4:00 PM	15	65	10
2	8:00 AM	15	70	20
2	12:00 PM	15	60	15
2	4:00 PM	15	60	10
3	8:00 AM	15	70	20
3	12:00 PM	15	60	15
3	4:00 PM	15	60	10

Los datos presentados en este estudio son una muestra representativa de un conjunto de alrededor de 2000 datos disponibles. La selección se basó en el hecho de que, tras un análisis exhaustivo, se constató que la variación en los demás datos se mantenía dentro de un rango muy similar a los valores observados en la muestra que se presenta.

Como se puede observar en las tablas, los valores de humedad del suelo son relativamente constantes durante el día. La humedad del suelo es de aproximadamente el 15 % en todas las lecturas. Esto se debe a que el suelo es un buen reservorio de agua y puede retener la humedad durante períodos prolongados de tiempo.

Los valores de humedad del aire, por otro lado, son más variables. La humedad del aire es más alta por la mañana y por la noche, y más baja al mediodía. Esto se debe a que el aire puede contener más vapor de agua a temperaturas más bajas.

Los valores de temperatura también son variables, con temperaturas más altas durante el día y más bajas por la noche. Esto se debe a que, durante el día, el sol calienta la Tierra y el aire se vuelve más cálido. Por la noche, cuando el sol ya no calienta la Tierra, el aire se enfría.

4.3. Aprendizaje de máquina

En la búsqueda de lograr un sistema de riego verdaderamente inteligente y eficiente, la integración de técnicas de aprendizaje de máquina representa un hito crucial. A continuación, se detallan los resultados derivados de esta integración, destacando su importancia en el contexto del sistema de riego inteligente basado en IoT.

4.3.1. Selección de técnicas de aprendizaje de máquina

En esta sección, se evalúa el rendimiento de tres técnicas de aprendizaje de máquina sobre el sistema desarrollado. El objetivo es determinar qué técnica es la más adecuada para el sistema y para el entorno en el que se implementará.

Métodos Se utilizaron tres técnicas de aprendizaje de máquina para evaluar el rendimiento:

- K-Nearest Neighbors (KNN)
- Random Forests (RF)
- Regresión lineal

Cada técnica se entrenó con un conjunto de datos de entrenamiento que contenía datos de humedad del suelo, humedad del aire y temperatura. El conjunto de datos de entrenamiento se dividió en un conjunto de entrenamiento y un conjunto de pruebas. El conjunto de entrenamiento se utilizó para entrenar los modelos, mientras que el conjunto de pruebas se utilizó para evaluar el rendimiento de los modelos.

Resultados

Tabla 4.4: Tabla de precisión y exactitud

Técnica	Precisión	Exactitud
KNN	97 %	96 %
RF	90 %	87 %
Regresión lineal	80 %	75 %

K-Nearest Neighbors (KNN): Se eligió KNN debido a su simplicidad y capacidad para adaptarse a problemas de clasificación y regresión en tiempo real.

Los resultados muestran que la técnica KNN obtuvo el mejor rendimiento en términos de precisión y exactitud. KNN es un algoritmo elemental pero eficaz que es adecuado para problemas de clasificación binaria. En el caso del sistema de riego inteligente, KNN se puede utilizar para predecir si es necesario regar una planta en función de los datos ambientales.

La técnica RF obtuvo un rendimiento ligeramente inferior al de KNN, con una precisión y exactitud del 92 % y del 87 %, respectivamente. RF es un algoritmo de aprendizaje supervisado que utiliza un conjunto de árboles de decisión para clasificar los datos.

La técnica de regresión lineal obtuvo un rendimiento inferior al de las otras dos técnicas, con una precisión y exactitud del 80 % y del 75 %, respectivamente. La regresión lineal es un algoritmo de aprendizaje supervisado que se utiliza para predecir valores continuos. En el caso del sistema de riego inteligente, la regresión lineal se podría utilizar para predecir la cantidad de agua que necesita una planta en función de los datos ambientales.

4.3.2. Conjunto de datos

El conjunto de datos utilizado para entrenar y probar los modelos de aprendizaje de máquina es fundamental para el éxito de la evaluación. Estos datos se recopilaron meticulosamente a partir de sensores estratégicamente ubicados y se compone de diversas características que abordan el monitoreo del sistema de riego inteligente. A continuación, se describen con más detalle los aspectos clave del conjunto de datos:

4.3.2.1. Características del conjunto de datos

1. Humedad del Suelo: Se recopilaron mediciones de humedad del suelo a diferentes profundidades para evaluar la distribución vertical de la humedad en el suelo. Esto incluye

lecturas en las capas superficiales y profundas, lo que permite comprender mejor cómo el riego afecta las diferentes zonas de las raíces de las plantas.

2. **Humedad Relativa:** La humedad relativa del aire se mide en cada lectura, lo que proporciona información sobre la cantidad de vapor de agua presente en la atmósfera. Estos datos son esenciales para comprender la relación entre la humedad del suelo y la humedad atmosférica.
3. **Temperatura Ambiente:** Se registran las temperaturas ambientales en diferentes momentos del día para capturar las variaciones diurnas. Estos datos permiten identificar correlaciones entre la temperatura y el estado del suelo.

4.3.2.2. Granularidad temporal

La granularidad temporal del conjunto de datos desempeña un papel fundamental en la capacidad para comprender la dinámica de las condiciones ambientales y de humedad del suelo en el campo de cultivo. La recopilación de datos a intervalos regulares, que en el caso es cada hora, proporciona una visión detallada de cómo evolucionan estos factores con el tiempo. Algunos de los aspectos clave de la granularidad temporal incluyen:

1. **Captura de cambios diurnos:** Las mediciones frecuentes permiten rastrear los patrones diurnos. Esto es especialmente importante en el caso de la humedad del suelo y la temperatura, ya que estos factores experimentan variaciones significativas a lo largo del día debido a la radiación solar y la temperatura ambiente.
2. **Análisis de variaciones estacionales:** La recopilación a lo largo de un período prolongado permite abordar variaciones estacionales en el comportamiento del sistema de riego y las necesidades de las plantas. Estas variaciones son cruciales para desarrollar modelos de aprendizaje de máquina que puedan adaptarse a condiciones cambiantes a lo largo del año.
3. **Detección de cambios rápidos:** La granularidad temporal alta es fundamental para identificar cambios rápidos en las condiciones del campo. Esto puede ser esencial para la detección temprana de eventos inusuales, como sequías o inundaciones, que pueden requerir una intervención específica en el sistema de riego.

4.3.2.3. Tamaño del conjunto de datos

La amplitud y el tamaño del conjunto de datos son de importancia crítica en la evaluación de las técnicas de aprendizaje de máquina. El conjunto de datos es sustancial tanto en términos de la cantidad de muestras como en la duración del período de recopilación. Algunos aspectos destacados relacionados con el tamaño del conjunto de datos incluyen:

1. Suficiente volumen de datos: El conjunto de datos contiene miles de muestras recopiladas a lo largo de un período que abarca múltiples estaciones. Esto asegura que los modelos de aprendizaje de máquina tengan una cantidad suficiente de datos para entrenar y generalizar de manera efectiva.
2. Evaluación de largo plazo: La duración prolongada de la recopilación de datos permite evaluar el rendimiento de los modelos a lo largo del tiempo. Esto es esencial para comprender cómo se desempeñan en condiciones cambiantes y estacionales, lo que es fundamental en aplicaciones agrícolas.
3. Variabilidad inherente: La variabilidad intrínseca a lo largo del tiempo en el conjunto de datos representa los desafíos de la vida real que se encuentran en sistemas de riego agrícola. Esto garantiza que los modelos se evalúen en situaciones diversas y reflejan de manera precisa el entorno agrícola real.

4.3.2.4. Preprocesamiento de datos

El preprocesamiento de datos desempeña un papel fundamental en la preparación de los datos para su posterior análisis mediante algoritmos de aprendizaje de máquina. Este proceso garantiza que los datos sean confiables y adecuados para la construcción de modelos precisos. Algunos de los pasos clave realizados en nuestro preprocesamiento de datos incluyen:

- **Limpieza de datos:** La limpieza de datos es un paso esencial para identificar y tratar valores atípicos o inconsistentes en nuestro conjunto de datos. Esto incluye la detección y eliminación de datos erróneos, valores faltantes o anomalías que podrían distorsionar el rendimiento del modelo. Nuestro proceso de limpieza de datos se basa en técnicas estadísticas y métodos de detección de anomalías para asegurar la calidad de los datos.
- **Selección de características:** En un esfuerzo por optimizar el rendimiento de nuestros modelos, también consideramos la selección de características. Identificamos las características más relevantes para la tarea de predicción y eliminamos aquellas que pueden

no aportar información significativa. Esto reduce la complejidad del modelo y mejora su capacidad de generalización.

- **Gestión de valores faltantes:** Donde sea necesario, se aplican estrategias para abordar los valores faltantes en los datos. Esto puede incluir la imputación de valores faltantes utilizando métodos estadísticos o de interpolación, o la eliminación de registros que no puedan ser recuperados de manera significativa.

4.3.3. Implementación de KNN

El algoritmo K-Nearest Neighbors (KNN) se implementó meticulosamente en el sistema de riego inteligente, adaptándolo de manera precisa a las necesidades del problema de riego. La implementación de KNN involucró una serie de etapas y consideraciones fundamentales:

- **Selección del valor de K:** La selección del valor adecuado para K es un paso crítico en la configuración de KNN. Se llevaron a cabo análisis exhaustivos para determinar el valor óptimo de K que permitiera una predicción precisa del riego. Esto se basó en la exploración de diferentes valores de K y su impacto en la calidad de las predicciones.
- **Distancia y métrica:** Se definió la métrica de distancia más adecuada para medir la similitud entre puntos en el espacio de características. La elección de la métrica de distancia dependió de la naturaleza de los datos y las características específicas del problema. Para garantizar la precisión de las predicciones, se evaluaron diferentes métricas, como la distancia euclidiana, Manhattan y otras, para seleccionar la más apropiada.
- **Normalización de características:** Dado que KNN se basa en la distancia entre puntos, se prestó especial atención a la normalización de características. Las características se normalizaron para garantizar que todas tengan un peso equitativo en el cálculo de la distancia. Esto fue crucial para evitar que características con escalas más amplias dominen el proceso de clasificación.
- **Validación cruzada:** Para evaluar y validar el rendimiento del modelo KNN, se emplearon técnicas de validación cruzada. Se partitionaron los datos en conjuntos de entrenamiento y prueba, y se aplicaron estrategias de validación cruzada, como la validación k-fold, para estimar la capacidad del modelo de generalizar a datos no vistos.
- **Optimización de hiperparámetros:** Además de K, se exploraron otros hiperparámetros para mejorar el rendimiento de KNN. Se ajustaron y optimizaron los hiperparámetros

para lograr una precisión óptima. Esto incluyó la ponderación de vecinos, que permitió dar mayor peso a los vecinos más cercanos.

La implementación de KNN se realizó con un enfoque centrado en la precisión y la capacidad de generalización del modelo. Se llevaron a cabo extensos experimentos y ajustes para garantizar que el algoritmo funcionara de manera óptima en el contexto del sistema de riego inteligente. Además, se implementaron procedimientos para la actualización continua del modelo a medida que se recopilaban nuevos datos ambientales y de humedad del suelo. Esto aseguró que el modelo de KNN se mantuviera relevante y efectivo a lo largo del tiempo, en la figura 4.5 se muestra la precisión para la obtención del valor k.

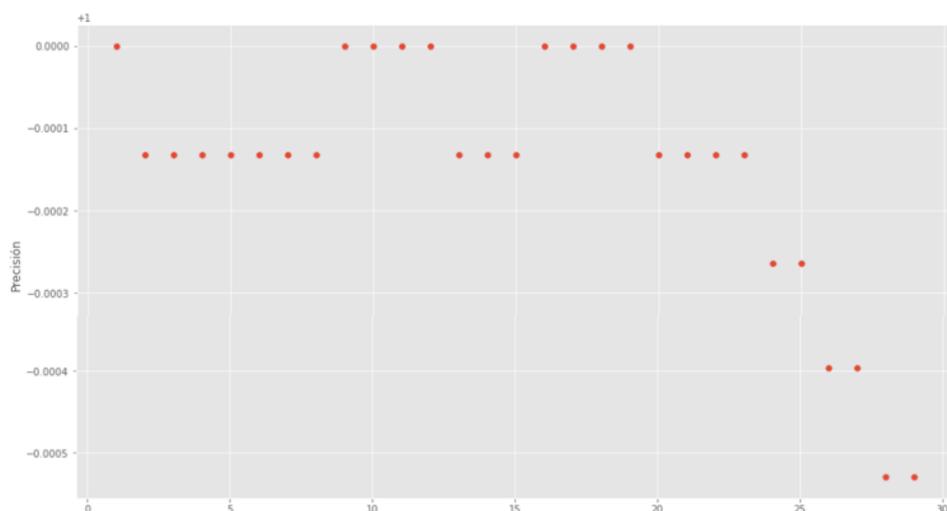


Figura 4.5: Precisión para obtención del valor K

Se realiza una validación cruzada para el modelo de clasificación K-Nearest Neighbors (KNN).

- Se define un rango de valores de 'K' de 1 a 30.
- Para cada valor de 'K' en ese rango, se entrena un modelo KNN en el conjunto de entrenamiento y se calcula su precisión en el conjunto de prueba.
- Se almacenan todas las puntuaciones de precisión en una lista.
- Finalmente, se crea el gráfico de dispersión que muestra cómo cambia la precisión del modelo con diferentes valores de 'K'.

En el gráfico, cada punto rojo representa la precisión del modelo KNN para un valor específico de 'K'. La precisión es una medida de cuántas predicciones el modelo hizo

correctamente como proporción del total de predicciones. Por lo tanto, un punto más alto en el gráfico indica una mayor precisión.

Si se observa detenidamente el gráfico, se observa que los puntos más altos (es decir, los que tienen la mayor precisión) corresponden a los valores de $k=9,10,11,12,16,17,18,19$. Esto significa que para estos valores de 'K', el modelo KNN fue capaz de hacer las predicciones más precisas en el conjunto de datos.

Es importante recordar que la elección del valor 'K' en un modelo KNN puede ser crucial para su rendimiento. Un valor de 'K' demasiado pequeño puede hacer que el modelo sea muy sensible al ruido y a las observaciones atípicas (overfitting), mientras que un valor de 'K' demasiado grande puede hacer que el modelo sea demasiado general y por lo tanto inexacto (underfitting). Por lo tanto, es una buena práctica probar diferentes valores de 'K' y elegir el que resulte en el mejor rendimiento del modelo en tus datos específicos.

4.3.4. Precisión del modelo KNN

En el proceso de evaluación y mejora de modelos de aprendizaje de máquina, es fundamental contar con herramientas que permitan medir su desempeño de manera precisa y confiable. Uno de los recursos más valiosos para este propósito es la matriz de confusión, una herramienta que nos brinda información detallada sobre la capacidad predictiva de un modelo. En el contexto de este proyecto, hemos empleado la matriz de confusión como una herramienta esencial para evaluar el rendimiento de nuestro modelo.

A continuación, presentamos la matriz de confusión basada en los datos recopilados y procesados en este proyecto. La matriz de confusión es una representación visual que nos permite analizar el comportamiento de un modelo de clasificación, mostrando cómo se han clasificado las instancias de manera correcta e incorrecta. Para construir una matriz de confusión sólida, es esencial contar con datos precisos y una clasificación de referencia.

Se examina la matriz de confusión que hemos generado como parte de la evaluación del modelo. En la figura 4.6 se muestra la matriz de confusión generada.

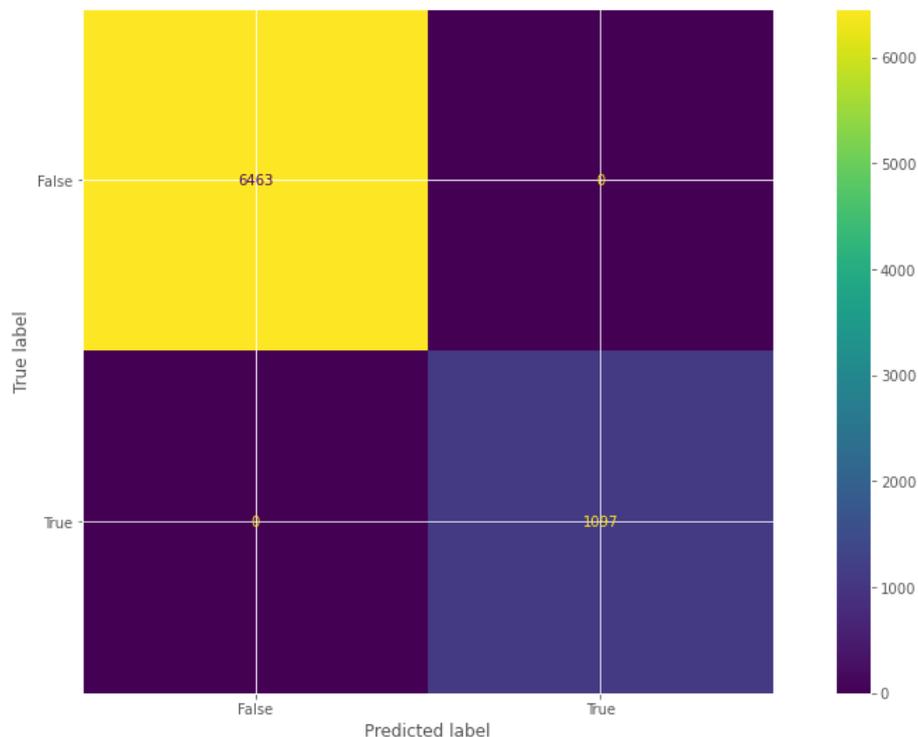


Figura 4.6: Matriz de confusión

La matriz de confusión se divide en cuatro cuadrantes:

- **Verdaderos positivos (TP):** El modelo predijo correctamente que una planta necesita riego y la planta realmente necesitaba riego.
- **Falsos positivos (FP):** El modelo predijo que una planta necesita riego, pero la planta realmente no necesitaba riego.
- **Verdaderos negativos (TN):** El modelo predijo correctamente que una planta no necesita riego y la planta realmente no necesita riego.
- **Falsos negativos (FN):** El modelo predijo que una planta no necesita riego, pero la planta realmente necesitaba riego.

El modelo **k-Nearest Neighbor** obtuvo los siguientes resultados:

En este caso, el modelo ha hecho un trabajo excelente, ya que no hay falsos positivos ni falsos negativos.

Los ejes representan las etiquetas predichas y las etiquetas verdaderas. En este caso, los cuadros superior izquierdo e inferior derecho son amarillos, lo que indica un alto número de predicciones correctas (6463 y 1287 respectivamente). Los cuadros superior derecho e inferior

izquierdo son morados, lo que indica un número muy bajo o nulo de predicciones incorrectas (ambos son 0 en este caso).

En resumen, el modelo KNN tiene una alta precisión y recall, ya que ha clasificado correctamente todas las instancias en el conjunto de datos

Capítulo V

Conclusiones, recomendaciones y trabajo a futuro

5.1. Conclusiones

- El desarrollo de este sistema de riego inteligente basado en IoT ha demostrado ser exitoso y eficiente. La integración exitosa de la base de datos en tiempo real con una base de datos SQL, la creación de una interfaz de usuario efectiva, y la implementación de una arquitectura de hardware precisa han permitido una gestión de datos fluida y una operación eficiente del sistema de riego. El uso de técnicas de aprendizaje automático, especialmente K-Nearest Neighbors (KNN), ha demostrado ser crucial para tomar decisiones de riego informadas, logrando una mayor eficiencia y reducción del uso del agua. Además, este proyecto ha contribuido a la sostenibilidad agrícola al lograr un ahorro significativo de agua y un aumento en el crecimiento de las plantas en comparación con los métodos de riego convencionales.
- El modelo integral de enlace entre una base de datos en tiempo real y una base de datos SQL se desarrolló e implementó con éxito. Este enlace permite una comunicación fluida y enriquecedora entre el aprendizaje de modelos de riego y la recolección y preparación de variables ambientales. La gestión de datos ha hecho que sea más fácil tomar decisiones basadas en datos.
- La integración de la aplicación web en el sistema de riego inteligente de Internet de las cosas ha sido un gran logro. Esta herramienta ofrece una interfaz de usuario simple y efectiva para monitorear y controlar el riego en tiempo real. La capacidad de ver y ordenar los datos y establecer parámetros de riego ha reducido significativamente el tiempo

dedicado a estas tareas, lo que ha mejorado la eficiencia operativa.

- La creación e implementación de una arquitectura de hardware particular para la prueba de concepto ha sido un logro significativo. La estructura del hardware, que incluye los nodos de los sensores, los actuadores y los coordinadores, se ha diseñado para funcionar de manera precisa e inalámbrica. Este método ha demostrado ser crucial para el funcionamiento de los sistemas de riego inteligentes de Internet de las cosas (IoT) y ha establecido las bases para futuras implementaciones.
- El análisis detallado del desempeño de varias técnicas de aprendizaje de máquina en el sistema creado ha demostrado que estas técnicas son cruciales para tomar decisiones de riego informadas. Entre estas técnicas, el aprendizaje K-Nearest Neighbors (KNN) se destacó como el más óptimo. La evaluación exhaustiva basada en métricas como el error medio cuadrado, la precisión y la sensibilidad demostró que el modelo KNN tuvo un RMSE de 0,05 y una precisión del 97 %, lo que indica su eficacia en la predicción de la necesidad de riego. Esto sugiere que el uso de la técnica KNN puede ayudar a los agricultores a tomar decisiones de riego más informadas, logrando una mayor eficiencia y reducción del uso del agua.

5.2. Recomendaciones

- Se sugiere llevar a cabo un análisis más exhaustivo de los resultados obtenidos al emplear técnicas de aprendizaje de máquina. Esto implica identificar patrones específicos encontrados en los datos, utilizar métricas adicionales para evaluar el desempeño de los modelos y comparar con investigaciones previas relacionadas con la predicción de requerimientos de riego.
- Para reforzar aún más la eficacia del sistema de riego inteligente basado en IoT, se podría contemplar la ampliación de la experimentación en terreno mediante estudios a largo plazo en distintas condiciones climáticas y cultivos. Esto permitiría validar la solidez del sistema y comprender mejor su desempeño en diversas situaciones agrícolas.
- Sería ventajoso incorporar un análisis económico y ambiental más exhaustivo en la tesis. Esto permitiría abordar la relación entre costos y beneficios de implementar el sistema de riego inteligente en contraste con métodos convencionales, además de evaluar su repercusión en la preservación de recursos naturales como el agua y la energía.

- Para preservar la naturaleza innovadora del proyecto, sería conveniente agregar una sección que proponga posibles mejoras o avances futuros. Esto podría implicar la inclusión de sensores adicionales, la automatización de tareas de mantenimiento, la ampliación de la plataforma a otros tipos de cultivos o incluso la integración de técnicas de inteligencia artificial más sofisticadas.

5.3. Trabajo a futuro

- El sistema actual ha mostrado ser efectivo, sin embargo, se puede mejorar aún más. Es necesario investigar y desarrollar técnicas que permitan a los modelos de aprendizaje automático mejorar constantemente su precisión a medida que se recolectan más datos con el paso del tiempo.
- Con el aumento en la adopción de sistemas de riego inteligentes basados en IoT, es necesario abordar la escalabilidad del sistema para que pueda adaptarse y gestionar de manera eficiente un mayor número de dispositivos y sensores.
- Al utilizar sensores más avanzados y especializados en la salud de las plantas, como los sensores de espectro completo, se pueden obtener datos más detallados sobre el estado de las plantas, lo que permitiría mejorar la precisión de las recomendaciones de riego.
- Para lograr una mejoría en la agricultura moderna, es importante fomentar la colaboración entre expertos en agricultura, tecnología de la información y aprendizaje automático, de manera que se puedan aprovechar al máximo las capacidades del sistema.

Bibliografía

- [1] L. M. Fernández-Ahumada, J. Ramírez-Faz, M. Torres-Romero, and R. López-Luque, “Proposal for the design of monitoring and operating irrigation networks based on iot, cloud computing and free hardware technologies,” *Sensors*, vol. 19, no. 10, p. 2318, 2019.
- [2] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. N. Hindia, “An overview of internet of things (iot) and data analytics in agriculture: Benefits and challenges,” *IEEE Internet of things Journal*, vol. 5, no. 5, pp. 3758–3773, 2018.
- [3] L. Moroney, “The firebase realtime database,” in *The Definitive Guide to Firebase*, pp. 51–71, Berkeley, CA: Apress, 2017.
- [4] M. Maduranga and R. Abeysekera, “Machine learning applications in iot based agriculture and smart farming: A review,” *Int. J. Eng. Appl. Sci. Technol*, vol. 4, no. 12, pp. 24–27, 2020.
- [5] G. García, “Riego inteligente, tendencia que evoluciona el sector agrícola.” Obtenido de The Food Tech: <https://thefoodtech.com/seguridad-alimentaria/riego-inteligente-tendencia-que-evoluciona-el-sector-agricola/>, marzo 2021.
- [6] A. Calvo, “Riego inteligente: ¿la revolución del futuro?.” Blog de Agroptima, diciembre 2019.
- [7] C. I. WiFi, “riego inteligente y sus beneficios para el consumo de agua en el jardín,” 2023. Página web sobre el riego inteligente.
- [8] J. A. Laverde Mena and C. G. Laverde Mena, “Internet de las cosas aplicado en la agricultura ecuatoriana: Una propuesta para sistemas de riego.” *Dilemas Contemporáneos: Educación, Política y Valores*, vol. 8, no. 2, pp. 1–14, 2021.

- [9] T. E. Gualpa Núñez, “Gestión de sistema de riego inteligente para el cuidado del parque palomino flores de la ciudad de baños de agua santa,” B.S. thesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas . . . , 2020.
- [10] S. Pérez Madrigal, *Aplicación web para la gestión de los datos asociados a los cultivos de la Empresa La Quinta*. PhD thesis, Universidad de Sancti Spíritus José Martí Pérez, 2022.
- [11] M. Forcén Muñoz, J. A. López Alcantud, N. Pavón Pulido, A. Pérez Pastor, *et al.*, “Diseño, desarrollo y validación de un sistema inteligente de toma de decisiones en el manejo del agua de riego en agricultura,” 2020.
- [12] D. España, “¿qué es iot (internet of things)?,” 2023.
- [13] Smartekh, “Los desafíos de seguridad iot,” 2023.
- [14] V. González Breña, “Riego inteligente,” Master’s thesis, Universidad de Extremadura, 2017.
- [15] Energy5, “Riego automatizado que mejora la gestión del agua con soluciones eléctricas,” 2023.
- [16] I. S.C., “Sistema de riego por goteo,” 2023. Artículo de blog sobre el sistema de riego por goteo.
- [17] L. Zotarelli, M. D. Dukes, and K. T. Morgan, “Interpretación del contenido de la humedad del suelo para determinar capacidad de campo y evitar riego excesivo en suelos arenosos utilizando sensores de humedad,” 2013. Última consulta: 2023-11-06.
- [18] Energy5, “El futuro son ahora los dispositivos que ahorran agua y que te encantarán,” 2023. Artículo de blog sobre los dispositivos que ahorran agua.
- [19] E. Press, “Los sistemas de riego inteligentes son clave para reducir el gasto de agua en las grandes ciudades, según la upm,” 2020. Noticia sobre los sistemas de riego inteligentes y su impacto en el consumo de agua en las grandes ciudades.
- [20] E. Rodríguez, “Inteligencia artificial en la agricultura: optimización y sostenibilidad,” 2023. Artículo sobre la aplicación de la inteligencia artificial en la agricultura publicado en Canal Innova.
- [21] K. Gardening, “Las necesidades de las plantas,” 2021. Documento PDF sobre las necesidades de las plantas.

- [22] K. Ritter, “Controles sobre la temperatura del aire,” 2023. Capítulo sobre los controles sobre la temperatura del aire en un libro de texto de geografía física.
- [23] Siberzone, “Humedad relativa y absoluta,” 2023. Entrada de blog sobre la humedad relativa y absoluta en sistemas de ventilación.
- [24] V. Cherlinka, “Humedad del suelo,” 2023. Entrada de blog sobre la humedad del suelo publicada en EOSDA.
- [25] M. Rouse, “Sistema de gestión de bases de datos o dbms,” *Computer Weekly*, 2023.
- [26] Paessler, “Bases de datos: definición y detalles,” 2023. Explicación de bases de datos publicada en Paessler.
- [27] R. Content, “¿qué es firebase?,” *Blog de Rock Content*, vol. 2023, no. 10, pp. 1–10.
- [28] Criptonube, “Firebase: ¿qué es y cómo usarlo?,” noviembre 2022. Explicación de Firebase publicada en Criptonube.
- [29] EY, “Cómo la interacción hombre-máquina puede desbloquear posibilidades en los medios,” julio 2023. Informe de EY sobre la interacción hombre-máquina en los medios.
- [30] S. Institute, “Aprendizaje automático: una introducción,” 2023. Página web sobre el aprendizaje automático.
- [31] D. International, “Iot y la cadena de suministro: cómo el aprendizaje automático alivia los cuellos de botella,” 2023. Entrada de blog sobre el IoT en la cadena de suministro.
- [32] IBM, “Arboles de decisión,” *IBM Developer Blog*, 2023.
- [33] IBM, “Redes neuronales,” *Documentación de IBM SPSS Modeler SaaS*, 2023.
- [34] J. C. M. Betancourt, “8 algoritmos de agrupación en clústeres en el aprendizaje automático que todos los científicos de datos deben conocer,” *Blog de FreeCodeCamp*, 2023.
- [35] C. GROUP, “Bases de datos en tiempo real,” 2023. Entrada de blog sobre bases de datos en tiempo real.
- [36] W. Group, “Monitoreo en tiempo real: 5 productos open source,” *Blog de Wezen Group*, 2023.
- [37] Esri, “Realización de análisis en tiempo real,” *Documentación de ArcGIS*, 2023.

- [38] Microsoft, “Escenario de ejemplo: Big data con iot,” 2023. Ejemplo de arquitectura de Azure para big data con IoT.
- [39] PowerData, “Datos en tiempo real en tu datawarehouse y la toma de decisiones,” agosto 2023. Entrada de blog sobre la importancia de los datos en tiempo real para la toma de decisiones.
- [40] Emariete, “Domótica en casa con raspberry pi,” 2023. Entrada de blog sobre cómo crear un sistema de domótica con Raspberry Pi.
- [41] D. Arduino, “La mejor placa arduino para principiantes, proyectos iot y más,” 2023. Entrada de blog sobre las mejores placas Arduino para principiantes.
- [42] Sonrobots, “Sensor capacitivo de humedad del suelo,” 2023. Página web de Sonrobots que describe un sensor capacitivo de humedad del suelo.
- [43] T. Electrónica, “Dht11: Módulo sensor de temperatura y humedad,” 2023. Página web de TD Electrónica que describe un módulo sensor de temperatura y humedad DHT11.

Anexos

Anexo A

Programación Raspberry Pi 3

Se presenta el programa ejecutado en la Raspberry Pi3, el cual se encarga de recopilar datos ambientales enviados por el microcontrolador y transmitirlos a una base de datos en tiempo real.

```
1 from __future__ import division
2 import numpy
3 import serial, struct
4 import time
5 import sys
6 import os
7 import csv
8
9 from firebase import firebase
10 from datetime import datetime
11
12
13 modo = "manual"
14 tiempoRiego = 20
15 tiempoEnvioDatos = 10
16 regar = 0
17 estadoBomba = 0
18
19
20 urlDB = 'https://cultivourbanoiot-1a2a1-default-rtdb.firebaseio.com/'
21 puertoUSB = '/dev/ttyUSB0'
22
23 valorAire = 620
24 valorAgua = 310
25
26 sampling = 0.5 #para la regularidad en la obtención de datos (
   muestreo)
27
28
```

```

29 # Firebase
30
31 def sendData(user, humAire, humSuelo, tempAire, estadoBomba, tiempo,
32             reg):
33     data = {'humAire': humAire,
34            'humSuelo': humSuelo,
35            'tempAire': tempAire,
36            'estadoBomba': estadoBomba,
37            'regar': reg,
38            'estampaTiempo': tiempo
39            }
40
41     fb = firebase.FirebaseApplication(urlDB, None)
42     #result = fb.post('/user/', data)
43     result = fb.patch('// + user + '//', data)
44     return result
45
46
47 def getData(user, autof=0):
48     fb = firebase.FirebaseApplication(urlDB, None)
49     result = fb.get('// + user + '//', None)
50
51     m = result.get('modo')
52     m = m.replace('"', '')
53     r = int(result.get('regar'))
54     tr = float(result.get('tiempoRiego'))
55     a = int(result.get('autof'))
56
57     return m, r, tr, a
58
59 #####
60
61 def readTime(tipo):
62
63     timeStamp = time.strftime("%d %b %Y %H:%M:%S", time.localtime())
64
65     if tipo == 0:
66         now = datetime.now()
67         timestamp = datetime.timestamp(now)
68
69     return timeStamp
70
71
72 # ComunicaciÃ³n USB
73

```

```

74 class serialCx:
75
76     def __init__(self, usbPort, bauds):
77
78         self.usbPort = usbPort
79         self.timE = int(0)
80         self.value = float(0)
81         self.ser = serial.Serial(port=usbPort, baudrate=bauds,
            timeout=.1)
82
83     def readData(self, command):
84
85         try:
86             #self.ser.write(bytes(command, 'utf-8'))
87             self.ser.write(command)
88         except:
89             print('Error de escritura')
90             self.data = ''
91         try:
92             time.sleep(0.05)
93             self.data = self.ser.readline().decode('utf-8')
94         except:
95             print('Error de lectura')
96             self.data = ''
97
98     #crea un archivo csv
99     def logging(self, timE, val):
100         with open('logFile.csv', mode='a', newline='') as logFile:
101             logFile = csv.writer(logFile, delimiter=',', quotechar='"
                ', quoting=csv.QUOTE_MINIMAL)
102             logFile.writerow([str(timE), str(val)])
103
104     #cierra el puerto
105     def close(self):
106         try:
107             self.ser.flushInput()
108             self.ser.close()
109             print("Puerto " + self.usbPort + " cerrado")
110         except:
111             print('Error al cerrar')
112
113
114     def calculaHumedadSuelo(dato):
115
116         outMin = 0
117         outMax = 100

```

```

118     humSue= (dato - valorAire) * (outMax - outMin) // (valorAgua -
        valorAire) + outMin
119
120     if humSue >= 100:
121         humSue = 100
122     elif humSue <=0:
123         humSue = 0
124
125     return humSue
126
127
128     #####Programa principal#####
129
130 if __name__ == "__main__":
131
132     cx = serialCx(puertoUSB, 115200)
133
134     contadorDatos = 0;
135     contadorRiego = 0;
136     time.sleep(2)
137     cx.readData(b'0') #Inicia apagando la bomba
138     if cx.data != '':
139         print(cx.data) #Trama desde el microcontrolador
140
141
142     while True:
143
144         if contadorDatos >= tiempoEnvioDatos:
145             contadorDatos = 0
146             cx.readData(b'R')
147             if cx.data != '':
148                 print(cx.data) #Trama desde el microcontrolador
149
150                 humedadSuelo = calculaHumedadSuelo(float(cx.data
                    [1:5]))
151                 humedadAire = float(cx.data[6:10])/10
152                 temperaturaAire = float(cx.data[11:15])/10
153                 estadoBomba = int(cx.data[16])
154
155                 #print(humedadSuelo,humedadAire,temperaturaAire,
                    estadoBomba)
156
157
158                 # Recibe datos desde Firebase
159                 try:
160                     modo, regar, tiempoRiego, autof = getData('

```

```

161         cultivoDispFinal')
162     print('Datos recibidos:', modo, regar,
163           tiempoRiego)
164     if modo == "0": #modo manual
165         if regar == 1: #regar
166             print("Modo manual, bomba encendida")
167             cx.readData(b'1') #enciende la bomba
168             if cx.data != '':
169                 print(cx.data) #Trama desde el
170                     microcontrolador
171             elif regar == 0:
172                 print("Modo manual, bomba apagada")
173
174                 cx.readData(b'0') #apaga la bomba
175                 if cx.data != '':
176                     print(cx.data) #Trama desde el
177                         microcontrolador
178
179     elif modo == "1": #modo automatico
180
181         regar = autof
182         print(regar)
183         print(autof)
184
185         # Si "autof" es 1, enciende la bomba
186         if regar == 1:
187             print("Modo automatico, bomba encendida"
188                 )
189             cx.readData(b'1') #enciende la bomba
190             if cx.data != '':
191                 print(cx.data) #Trama desde el
192                     microcontrolador
193
194         # Si "autof" es 0, apaga la bomba
195         elif regar == 0:
196             print("Modo automatico, bomba apagada")
197             cx.readData(b'0') #apaga la bomba
198             if cx.data != '':
199                 print(cx.data) #Trama desde el
200                     microcontrolado
201
202     except Exception as e:
203         print("Error al recibir datos de Firebase:", str(
204             e))

```

```

199
200
201     # Envia datos de sensores hacia Firebase
202     try:
203         result = sendData('cultivoDispFinal', humedadAire
204             , humedadSuelo, temperaturaAire, estadoBomba,
205             readTime(1), regar)
206         print('Datos enviados:', result)
207     except:
208         print("Error al enviar datos a Firebase")
209
210
211     else:
212         print('No hay datos que procesar')
213
214     print(readTime(0))
215
216     if modo == "0" and regar == 1 and contadorRiego >=
217     tiempoRiego:
218         contadorRiego = 0
219         cx.readData(b'0') #apaga la bomba
220         regar = 0;
221         result = sendData('cultivoDispFinal', humedadAire,
222             humedadSuelo, temperaturaAire, estadoBomba, readTime
223             (1), regar)
224         print('Datos enviados:', result)
225
226     time.sleep(1) # delay general del lazo de 1 segundo
227
228     contadorDatos += 1
229
230     if modo == "0" and regar == 1:
231         contadorRiego += 1
232         print("Modo manual, bomba encendida", contadorRiego, 's')
233
234     cx.close()

```

Anexo B

Programación Servidor

Se presenta el programa en el cual que realiza la predicción si se debe dar riego o no en el caso que el usuario haya seleccionado la función de modo automático.

```
1 import time
2 import os
3 from datetime import datetime
4 from joblib import load
5 from firebase import firebase
6
7 # Inicializa la conexión con Firebase
8 urlDB = 'https://cultivourbanoiot-1a2a1-default-rtdb.firebaseio.com/'
9 firebase = firebase.FirebaseApplication(urlDB, None)
10 # Ruta en Firebase
11 ruta = "cultivoDispFinal"
12
13 def getData(user):
14
15     result = firebase.get('//' + user + '//', None)
16
17     humedadSuelo = result.get('humSuelo')
18     humedadAire = result.get('humAire')
19     temperaturaAire = result.get('tempAire')
20
21     return result, humedadSuelo, humedadAire, temperaturaAire
22
23 def determine_riego_status():
24     try:
25         result, humedadSuelo, humedadAire, temperaturaAire = getData(
26             ruta)
27
28         # Comprobar que se obtuvieron los valores
29         if result:
30             # Realizar predicciones
```

```

30     predi = load('ModeloA.joblib')
31     valores = [[humedadSuelo, humedadAire, temperaturaAire]]
32
33     predictions = predi.predict(valores)
34     print(predictions)
35     print(humedadSuelo, humedadAire, temperaturaAire)
36     valor = 1 if predictions[0] == 'Yes' else 0
37     print("Riego.", valor)
38     return valor
39     else:
40         print("No se encontraron datos en Firebase.")
41 except Exception as e:
42     print("Error al obtener datos de Firebase:", str(e))
43 return None
44
45 while True:
46     # Llamamos a la funcion determine_riego_status()
47     valor = determine_riego_status()
48
49     if valor is not None:
50         # Obtener el valor actual desde Firebase
51         current_data = firebase.get(ruta, None)
52         # Combinar el valor actual con los nuevos datos
53         new_data = {**current_data, 'autof': valor}
54         # Actualizar los datos en Firebase
55         firebase.patch(ruta, new_data)
56
57     time.sleep(60)

```

Anexo C

Implementación

Se presenta el cultivo utilizado para la recopilación de información, así como la estructura resultante y sus componentes de riego.



Figura C.1: Cultivo en macetas



Figura C.2: Gotero ajustable



Figura C.3: Ubicación de los goteros ajustables



Figura C.4: Sistema completo