



Comparación de modelos predictivos usando sensores IoT y datos
meteorológicos del IDEAM en un cafetal de la finca los laureles, vereda San Inés
en San Vicente de Chucurí 2026

Modalidad: Proyecto de Investigación

Fabian Cárdenas Roa

CC 1095930641

Breimer Steven Almeida Pérez

CC 1006408492

UNIDADES TECNOLÓGICAS DE SANTANDER

Facultad de Ciencias Naturales e Ingenierías

Ingeniería electrónica

Bucaramanga 4 de octubre de 2026



Comparación de modelos predictivos usando sensores IoT y datos
meteorológicos del IDEAM en un cafetal de la finca los laureles, vereda San Inés
en San Vicente de Chucurí 2026

Modalidad: Proyecto de Investigación

Fabian Cárdenas Roa

CC 1095930641

Breimer Steven Almeida Pérez

CC 1006408492

Trabajo de Grado para optar al título de
Ingeniería electrónica

DIRECTOR

Cristhiam Jesid Gutiérrez Lozano

GICAV

UNIDADES TECNOLÓGICAS DE SANTANDER
Facultad de Ciencias Naturales e Ingenierías
Ingeniería electrónica
Bucaramanga 4 de octubre de 2026

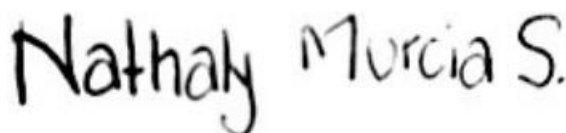
F-DC-125

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

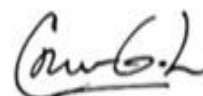
VERSIÓN: 2.0

Nota de Aceptación

Este informe final de trabajo de grado,
en modalidad Proyecto Investigación, fue APROBADO
en cumplimiento de uno de los requisitos exigidos por las
Universitaria Tecnológica de Santander para optar el
Título de Ingeniero Electrónico según acta No. 20 del 05
de octubre del 2026, del Comité de Trabajo de Grado



Firma del Evaluador: Nathaly Murcia Sepúlveda



Firma del Director: Cristhiam Jesid Gutiérrez Lozano

DEDICATORIA

A Universitaria Tecnológica de Santander, por el apoyo constante que hizo posible cada etapa de este camino. A la familia, que acompañó con paciencia las largas horas de estudio, y a quienes creyeron en este proyecto cuando apenas era una idea. Su respaldo silencioso está presente en cada página de este trabajo.

AGRADECIMIENTOS

Los autores expresan su reconocimiento a Cristhiam Jesid Gutierrez Lozano, por la orientación técnica y el acompañamiento brindados durante el desarrollo del trabajo de grado. Agradecen de manera especial al Instituto de Hidrología, Meteorología y Estudios Ambientales (IDEAM) por disponer de forma pública los registros de precipitación que sirvieron de insumo para el estudio, y a Alfonso Gómez y Omayra Rueda por permitir la instalación del sistema de monitoreo en su predio ubicado en la vereda Santa Inés. se extienden su gratitud a Universitaria Tecnológica de Santander y a los docentes del programa, cuyo apoyo académico hizo posible la culminación de este trabajo.

TABLA DE CONTENIDO

<u>RESUMEN EJECUTIVO</u>	<u>13</u>
<u>INTRODUCCIÓN</u>	<u>14</u>
<u>1. DESCRIPCIÓN DEL TRABAJO DE INVESTIGACIÓN</u>	<u>16</u>
1.1. PLANTEAMIENTO DEL PROBLEMA	16
1.2. JUSTIFICACIÓN	18
1.3. OBJETIVOS	20
1.3.1. OBJETIVO GENERAL	20
1.3.2. OBJETIVOS ESPECÍFICOS	20
1.4. ESTADO DEL ARTE	21
<u>2. MARCO REFERENCIAL</u>	<u>24</u>
2.1. MARCO TEÓRICO	24
2.1.1. HUMEDAD DEL SUELO Y BALANCE HÍDRICO EN SISTEMAS AGRÍCOLAS	24
2.1.2. BALANCE HÍDRICO EN SISTEMAS AGRÍCOLAS	25
2.1.3. FUNDAMENTOS DE LA AGRICULTURA DE PRECISIÓN	26
2.1.4. SISTEMAS DE MONITOREO DE HUMEDAD DEL SUELO EN AGRICULTURA DE PRECISIÓN	27
2.1.5. SISTEMAS IoT Y REDES DE SENSORES INALÁMBRICOS PARA EL MONITOREO DEL SUELO	28
2.2. MARCO CONCEPTUAL	29
2.3. MARCO LEGAL	31
2.4. MARCO AMBIENTAL	32

F-DC-125	INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO, MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO	VERSIÓN: 2.0
3.	<u>DISEÑO DE LA INVESTIGACIÓN</u>	<u>34</u>
4.	<u>DESARROLLO DEL TRABAJO DE GRADO</u>	<u>36</u>
4.1.	DISEÑO DEL SISTEMA DE MONITOREO IoT	36
4.2.	ARQUITECTURA GENERAL DEL SISTEMA	37
4.3.	SELECCIÓN DE SENSORES Y COMPONENTES ELECTRÓNICOS	38
4.4.	DISEÑO DEL CIRCUITO ELECTRÓNICO	40
4.5.	CONFIGURACIÓN DEL MICROCONTROLADOR ESP32	43
4.5.1.	CÓDIGO USADO	43
4.6.	DISEÑO DE LA CARCASA DE PROTECCIÓN	52
4.6.1.	REQUERIMIENTOS DE DISEÑO	53
4.6.2.	DIMENSIONAMIENTO Y GEOMETRÍA	54
4.7.	SELECCIÓN DEL MATERIAL Y PARÁMETROS DE IMPRESIÓN	56
4.7.1.	LIMITACIONES DEL ENCERRAMIENTO	58
4.8.	ADQUISICIÓN Y TRANSMISIÓN DE DATOS	58
4.9.	DISEÑO DE ALMACENAMIENTO DE DATOS	60
4.10.	IMPLEMENTACIÓN DEL SISTEMA DE MONITOREO AMBIENTAL	62
4.11.	INSTALACIÓN DE SENSORES EN EL CULTIVO	63
4.12.	CAPTURA DE VARIABLES AMBIENTALES	65
4.13.	FRECUENCIA DE MUESTREO Y RECOLECCIÓN DE DATOS	65
4.14.	INTEGRACIÓN DE DATOS METEOROLÓGICOS DEL IDEAM	67
4.15.	LIMPIEZA DE DATOS, NORMALIZACIÓN Y TRANSFORMACIÓN DE VARIABLES	69
4.15.1.	PROCESO ETL SOBRE LA BASE DE DATOS IDEAM	70
4.15.2.	PROCESO ETL SOBRE LA BASE DE DATOS OBTENIDA DE LA LECTURA DE LOS SENSORES	75
4.16.	ANÁLISIS EXPLORATORIO DE DATOS OBTENIDOS	78
4.16.1.	ANÁLISIS ESTADÍSTICO DATASET DEL IDEAM	78

F-DC-125 INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO VERSIÓN: 2.0

4.16.2.	ANÁLISIS ESTADÍSTICO DATASET DE LA LECTURA DE DATOS DE LOS SENSORES	82
4.17.	SELECCIÓN DE VARIABLES RELEVANTES	89
4.18.	DESARROLLO DEL MODELO PREDICTIVO	91
4.19.	SELECCIÓN DE ALGORITMOS DE APRENDIZAJE AUTOMÁTICO	94
4.20.	ENTRENAMIENTO DEL MODELO	96
4.21.	VALIDACIÓN DEL MODELO	100
4.21.1.	RESULTADOS DE LA VALIDACIÓN	103
4.22.	AJUSTE DE HIPERPARÁMETROS	105
4.23.	MÉTRICAS DE EVALUACIÓN	106
4.24.	COMPARACIÓN DE RESULTADOS.....	107
4.25.	ANÁLISIS DE PRECISIÓN DEL MODELO	112
4.26.	LIMITACIONES DEL MODELO DESARROLLADO	113
4.27.	PROPUESTA DE APOYO PARA LA TOMA DE DECISIONES DE RIEGO.....	114
4.28.	DEFINICIÓN DE CONDICIONES DE HUMEDAD DEL SUELO	115
4.29.	GENERACIÓN E INTERPRETACIÓN DE PREDICCIONES	116
4.30.	ESCENARIOS DE APOYO AL MANEJO HÍDRICO	118
5.	<u>RESULTADOS</u>	<u>120</u>
6.	<u>CONCLUSIONES</u>	<u>124</u>
7.	<u>RECOMENDACIONES</u>	<u>126</u>
8.	<u>REFERENCIAS BIBLIOGRÁFICAS</u>	<u>128</u>
9.	<u>ANEXOS</u>	<u>135</u>
9.1.	ANEXO A CÓDIGO DE PROGRAMACIÓN DEL MICROCONTROLADOR.....	135
9.2.	ANEXO B. CÓDIGO DE PROCESAMIENTO DE LOS DATOS	135
9.3.	ANEXO C CÓDIGO DE DESARROLLO DE LOS MODELOS PREDICTIVOS	135

LISTA DE FIGURAS

Figura 1. Arquitectura general del sistema de monitoreo	37
Figura 2. Esquema eléctrico del sistema de monitoreo	41
Figura 3. Diseño de la placa de circuito impreso	42
Figura 4. diagrama de bloques lógica suada en el algoritmo.....	52
Figura 5. Modelo tridimensional de la carcasa de protección	54
Figura 6. Creación de canal y espacios en Thingspeak	60
Figura 7. Data import/export en thingspeak.....	61
Figura 8. ejemplo de archivo .csv exportado desde Thingspeak	62
Figura 9. Dispositivo en el cultivo de café.....	64
Figura 10. Fragmento de base de datos del IDEAM.....	69
Figura 11. cargue del archivo en Google Colab y codificación	71
Figura 12. Revisión de la estructura general del archivo	72
Figura 13. Código usado para convertir la fecha y asegurar el formato de la columna valor	72
Figura 14. filtros y columnas a usar.....	73
Figura 15. cambio de nombres y orden	73
Figura 16. Revisión faltantes	74
Figura 17. Revisión final y guardado del archivo limpio.....	75
Figura 18. exploración inicial	76
Figura 19. conversión tipo de variable tiempo	77
Figura 20. Chequeo de rangos físicos.....	77
Figura 21. Estadísticas descriptivas de la precipitación.....	78
Figura 22. Visualización de la precipitación a lo largo del tiempo.....	79

Figura 23. Distribución de los valores de precipitación	80
Figura 24. Días con cero precipitación	81
Figura 25. Evolución de las variables ambientales durante el período de medición	83
Figura 26. Distribución de las cuatro variables ambientales	85
Figura 27. Matriz de correlación entre las variables ambientales	86
Figura 28. Perfil diario promedio de las variables ambientales.....	87
Figura 29. creación de variables derivadas	89
Figura 30. nuevas variables a considerar	90
Figura 31. Gestión de variables para modelo predictivo	91
Figura 32. Normalización de las variables.....	92
Figura 33. modelos entrenados.....	93
Figura 34. carga de los datasets	96
Figura 35. Preparación e integración del conjunto	97
Figura 36. Definición de variables de entrada y objetivo	98
Figura 37. Partición temporal	98
Figura 38. Normalización.....	99
Figura 39. Entrenamiento de los modelos	100
Figura 40. Métricas.....	101
Figura 41. detector sobreajuste	101
Figura 42. validación cruzada.....	102
Figura 43. comprobación visual.....	103
Figura 44. Tabla resumen metrico.....	108
Figura 45. comparación gráfica de modelos.....	109
Figura 46. Dispersion de los valores reales frente a los estimados	110
Figura 47. Distribución de los errores de estimación	111
Figura 48. Umbrales de referencia	116
Figura 49. generador de predicciones	117

F-DC-125

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

VERSIÓN: 2.0

Figura 50. Circuito impreso	121
Figura 51. Carcasa impresión 3D	122

LISTA DE TABLAS

Tabla 1.	Componentes electrónicos del sistema de monitoreo.....	39
Tabla 2.	Características geométricas de la carcasa de protección	55
Tabla 3.	Comparación de materiales de impresión para uso a la intemperie	57
Tabla 4.	Descripción de la base de datos del IDEAM	68
Tabla 5.	resumen de los resultados comparación modelos predictivos ...	123

RESUMEN EJECUTIVO

El presente trabajo desarrolla un modelo predictivo de humedad del suelo para un cultivo de café ubicado en la vereda Santa Inés, en el municipio de San Vicente de Chucurí, mediante la integración de datos de sensores IoT con información meteorológica del IDEAM. El objetivo general consiste en anticipar el estado de humedad del suelo para apoyar decisiones de riego más oportunas, y se apoya en cuatro objetivos específicos: diseñar el sistema de monitoreo, integrar las fuentes de datos, desarrollar modelos predictivos y evaluar su desempeño. La metodología es de tipo aplicado y enfoque cuantitativo. El sistema se construye sobre un microcontrolador ESP32 con sensores de humedad y temperatura del suelo y del aire, que transmite sus lecturas por Wi-Fi a la plataforma ThingSpeak cada cinco minutos. Esa información se une con la precipitación diaria de la estación Albania del IDEAM, tomando la fecha como elemento común, y se depura mediante un proceso de extracción, transformación y carga. Sobre el conjunto resultante se construyen variables derivadas de la lluvia y se comparan varias técnicas de análisis de datos, evaluadas con métricas estadísticas como el error cuadrático medio y el coeficiente de determinación. Los resultados muestran que la humedad del suelo responde sobre todo a la lluvia acumulada de los días previos y no a las condiciones instantáneas del aire, lo que confirma el valor de integrar los datos del IDEAM. El trabajo concluye que un sistema de bajo costo, apoyado en herramientas de código abierto, ofrece una alternativa viable para el monitoreo ambiental en fincas cafeteras y sienta las bases para futuros desarrollos en agricultura de precisión.

PALABRAS CLAVE: humedad del suelo, Internet de las Cosas, agricultura de precisión, modelo predictivo, IDEAM.

INTRODUCCIÓN

El café ocupa un lugar central en la economía agrícola de Colombia, y su producción depende en buena medida de las condiciones del clima y del agua disponible en el suelo. En los últimos años, la mayor irregularidad de las lluvias ha vuelto más difícil el manejo del cultivo, pues los periodos secos y los aguaceros intensos afectan tanto el rendimiento como la calidad del grano. En muchas fincas, el seguimiento de la humedad del suelo se sigue haciendo de forma empírica, lo que limita la toma de decisiones oportunas sobre el riego y el cuidado de las plantas.

Frente a ese panorama, la agricultura de precisión ofrece un camino distinto, apoyado en sensores y en el análisis de datos para decidir con información y no por intuición. La convergencia del Internet de las Cosas con las técnicas de análisis ha transformado la manera de monitorear los cultivos, al permitir la captura continua de variables ambientales y el aprovechamiento de esa información para anticipar situaciones. En lo que respecta al agua del suelo, varios estudios recientes muestran que los sistemas embebidos de bajo costo resultan adecuados para su seguimiento, incluso donde los recursos son limitados.

El interés, sin embargo, se ha desplazado de medir hacia predecir. El uso de técnicas de aprendizaje automático para estimar la humedad del suelo a partir de variables ambientales es un campo en crecimiento, en el que la calidad de los datos y la elección del modelo determinan el resultado. Buena parte de esos desarrollos se concentra, no obstante, en contextos ajenos a la caficultura colombiana y pocas veces combina las mediciones propias de una finca con la información meteorológica oficial disponible. Ahí reside el aporte de este trabajo, que integra los datos de un sistema IoT instalado en campo con los registros de precipitación del IDEAM para construir un modelo ajustado a las condiciones reales de un cafetal en San Vicente de Chucurí.

Para resolver el problema, el trabajo sigue un método de carácter aplicado y enfoque cuantitativo, que combina la medición experimental en campo con el análisis computacional de los datos. Un sistema basado en el microcontrolador ESP32 registra las variables ambientales del cultivo y las envía a una plataforma en la nube, desde donde se exportan para su procesamiento. A partir de esa información, depurada e integrada con la precipitación del IDEAM, se desarrollan y comparan varios modelos predictivos cuyo desempeño se mide con métricas estadísticas. El detalle de este procedimiento se presenta en la sección de diseño de la investigación y en el desarrollo del trabajo de grado.

1. DESCRIPCIÓN DEL TRABAJO DE INVESTIGACIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

El cultivo de café representa una de las actividades agrícolas más importantes en Colombia, tanto por su impacto económico como por su aporte al empleo rural. Este sector genera aproximadamente el 13,2 % del PIB agropecuario del país y cerca del 25 % del empleo rural, con alrededor de 960.000 empleos directos y más de 551.000 caficultores vinculados a esta actividad productiva (*Banco Agrario De Colombia, s.f.*).

A pesar de ello es importante destacar que la producción cafetera depende fuertemente de las condiciones climáticas y de la disponibilidad de agua en el suelo. De acuerdo con Salazar and H (2025) los estudios realizados en zonas cafeteras del país han demostrado que la humedad del suelo está directamente relacionada con la cantidad de lluvia acumulada, y que períodos prolongados con precipitaciones inferiores a ciertos niveles generan eventos de suelo seco que afectan el desarrollo del cultivo.

En los últimos años, la variabilidad climática ha incrementado la vulnerabilidad de los sistemas productivos del café. Por ejemplo, durante 2025 la producción nacional de café registró una disminución del 2 % debido a condiciones climáticas adversas asociadas a lluvias intensas, las cuales afectaron la floración y el desarrollo del fruto en varias regiones productoras (Digital, 2026).

Esta alta dependencia de factores climáticos representa una limitación para los caficultores, quienes en muchos casos toman decisiones de manejo del cultivo basadas únicamente en la experiencia o en observaciones empíricas del suelo, sin contar con herramientas tecnológicas que permitan monitorear variables ambientales en tiempo real o anticipar cambios en la humedad del suelo. Esta situación puede provocar una gestión ineficiente del recurso hídrico, afectando el rendimiento y la calidad del café.

los sistemas basados en Internet de las Cosas (IoT) representan una alternativa prometedora para recolectar datos ambientales en campo, mientras que la integración de información meteorológica proveniente del Instituto de Hidrología, Meteorología y Estudios Ambientales (IDEAM) podría mejorar la capacidad de predicción de variables agrícolas clave como la humedad del suelo.

1.2. JUSTIFICACIÓN

La producción agrícola, especialmente en cultivos de café, depende en gran medida de las condiciones climáticas y del estado de humedad del suelo. En muchas fincas cafeteras, el monitoreo de estas variables se realiza de manera empírica, lo que genera una limitación en la toma de decisiones oportunas sobre riego, protección del cultivo o manejo del suelo. Esta situación puede provocar pérdidas en la producción, afectando económicamente a los caficultores y repercutiendo en la calidad del producto que llega al consumidor.

Ante esta problemática, surge la necesidad de desarrollar herramientas tecnológicas que permitan monitorear y analizar las condiciones ambientales de los cultivos de manera continua. Es por que la implementación de un sistema basado en sensores IoT y datos meteorológicos del Instituto de Hidrología, Meteorología y Estudios Ambientales (IDEAM) permite recolectar información ambiental y generar modelos predictivos de humedad del suelo que contribuyan a mejorar la gestión del cultivo.

El desarrollo de esta propuesta permitirá anticipar cambios en las condiciones ambientales, facilitando que los caficultores tomen decisiones informadas, como ajustar el riego, proteger el cultivo ante lluvias intensas o prevenir condiciones que afecten su crecimiento. Por estas razones el proyecto se busca contribuir a una producción agrícola más eficiente y sostenible.

este proyecto resulta relevante para las Unidades Tecnológicas de Santander, ya que promueve el desarrollo de soluciones tecnológicas aplicadas al sector agropecuario y puede servir como base para futuras investigaciones en áreas como IoT, agricultura inteligente y análisis de datos. Además, la propuesta es

F-DC-125

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

VERSIÓN: 2.0

replicable en diferentes cultivos y regiones, lo que amplía su impacto académico,
tecnológico y social.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Desarrollar un modelo predictivo de humedad del suelo mediante la integración de datos recolectados por sensores IoT y variables meteorológicas del IDEAM, utilizando técnicas de análisis de datos para mejorar la toma de decisiones en el manejo de un cafetal en el municipio de San Vicente de Chucurí.

1.3.2. OBJETIVOS ESPECÍFICOS

1. Diseñar un sistema de monitoreo basado en sensores IoT para la medición de variables ambientales en un cafetal del municipio de San Vicente de Chucurí.
2. Integrar los datos recolectados por el sistema IoT con variables meteorológicas proporcionadas por el Instituto de Hidrología, Meteorología y Estudios Ambientales para la construcción de un conjunto de datos ambiental.
3. Desarrollar modelos predictivos de humedad del suelo a partir de las variables ambientales registradas por el sistema IoT y los datos meteorológicos externos.
4. Evaluar el desempeño de los modelos predictivos mediante métricas estadísticas para determinar la precisión en la estimación de la humedad del suelo.

1.4. ESTADO DEL ARTE

La humedad del suelo es una variable fundamental dentro del ciclo hidrológico terrestre, ya que influye directamente en la productividad agrícola, la gestión del agua y las interacciones entre suelo, vegetación y atmósfera. Debido a su alta variabilidad espacial y temporal, su medición y predicción representan un desafío importante para los sistemas agrícolas modernos y para los modelos hidrológicos tradicionales (Zhang et al., 2026).

En los últimos años, el uso de modelos basados en aprendizaje automático ha cobrado relevancia para la estimación y predicción de la humedad del suelo. Estos enfoques permiten analizar grandes volúmenes de datos ambientales y detectar relaciones complejas entre variables climáticas, propiedades del suelo y factores de vegetación. Según Zhang et al. (2026), los métodos de machine learning han demostrado mayor capacidad para capturar dinámicas no lineales en comparación con los modelos físicos convencionales, los cuales suelen requerir gran cantidad de parámetros y presentan limitaciones para representar heterogeneidad espacial en grandes áreas.

En esta línea, diversos estudios han explorado la integración de algoritmos avanzados para mejorar la precisión en la predicción de humedad del suelo. Arshad et al. (2026) desarrollaron un enfoque basado en reducción de dimensionalidad mediante t-Distributed Stochastic Neighbor Embedding (t-SNE) combinado con modelos de aprendizaje automático como Random Forest, Gradient Boosting Regression y Artificial Neural Networks. Los resultados demostraron que la incorporación de técnicas de reducción de dimensionalidad puede mejorar el rendimiento predictivo de los modelos, alcanzando coeficientes de determinación

superiores a 0.88 en regiones agrícolas irrigadas. Este tipo de enfoques resulta especialmente relevante en zonas con escasez de datos, donde la integración de múltiples fuentes de información ambiental permite mejorar las estimaciones de humedad del suelo.

Paralelamente, el desarrollo de tecnologías digitales ha impulsado el uso de sensores inteligentes, Internet de las Cosas (IoT) y sistemas de monitoreo en tiempo real para la gestión del agua en la agricultura. Sreeram et al. (2026) señalan que la combinación de sensores de humedad del suelo, plataformas IoT y análisis de datos permite optimizar el uso de recursos hídricos dentro de la agricultura de precisión. Estas tecnologías facilitan la recolección continua de datos ambientales, lo que permite aplicar modelos predictivos capaces de mejorar la planificación del riego y aumentar la eficiencia productiva.

En relación con los sistemas de predicción basados en sensores, Hendrickx et al. (2026) desarrollaron el sistema SWIM2, un modelo digital que integra datos provenientes de sensores de humedad del suelo con modelos de balance hídrico mediante técnicas de modelado inverso bayesiano. Este enfoque permite calibrar parámetros del suelo y del cultivo en tiempo real, generando predicciones confiables de humedad del suelo con horizontes de hasta siete días. Los resultados muestran que la combinación de datos de sensores y modelos probabilísticos puede mejorar significativamente la precisión de los sistemas de predicción y apoyar la toma de decisiones en el manejo del riego.

De manera complementaria, investigaciones orientadas a la agricultura de precisión han analizado el uso de sensores IoT para la toma de decisiones en sistemas de riego. Corrales Bonilla et al. (2025) propusieron un sistema de monitoreo ambiental para cultivos de café que integra sensores de humedad del

suelo, temperatura, radiación solar y variables climáticas. Los datos recolectados son procesados mediante algoritmos de aprendizaje automático, específicamente árboles de decisión, para determinar automáticamente cuándo realizar riego. Este tipo de sistemas permite mejorar la eficiencia en el uso del agua y aumentar la productividad agrícola al adaptar las decisiones de riego a las condiciones reales del cultivo.

2. MARCO REFERENCIAL

2.1. Marco teórico

2.1.1. *Humedad del suelo y balance hídrico en sistemas agrícolas*

La humedad del suelo es una de las variables más fundamentales dentro del ciclo hidrológico terrestre. Su importancia radica en que participa de manera directa en procesos tan diversos como la productividad agrícola, la gestión sostenible del agua, y las interacciones que ocurren entre el suelo, la vegetación y la atmósfera. A pesar de su relevancia, la humedad del suelo presenta una alta variabilidad tanto en el tiempo como en el espacio, lo que convierte su medición y predicción en uno de los mayores desafíos técnicos para los sistemas agrícolas modernos y para los modelos hidrológicos tradicionales (Zhang et al., 2024).

Desde una perspectiva física, la humedad del suelo hace referencia a la cantidad de agua almacenada en los poros del suelo dentro de una capa superficial relativamente delgada de la corteza terrestre. Esta variable es esencial en múltiples escalas espaciales, ya que interviene en procesos agrícolas, hidrológicos y de predicción meteorológica. En el contexto de la irrigación, la humedad del suelo resulta crítica para detectar el estrés hídrico de los cultivos y para tomar decisiones oportunas sobre cuándo y cuánto regar (Gamal et al., 2025).

En el caso específico del cultivo del café, la relación entre la humedad del suelo y el desarrollo de la planta es particularmente estrecha. Los estudios realizados en zonas cafeteras colombianas han demostrado que la humedad del suelo está directamente relacionada con la cantidad de lluvia acumulada, y que períodos prolongados con precipitaciones inferiores a ciertos umbrales generan condiciones de suelo seco que afectan negativamente el desarrollo del cultivo. Esta

dependencia climática convierte al monitoreo continuo de la humedad del suelo en una necesidad estratégica para los caficultores.

2.1.2. Balance hídrico en sistemas agrícolas

El balance hídrico es el marco conceptual que permite comprender cómo el agua entra, se almacena y sale de un sistema agrícola. En términos generales, el balance integra variables como la precipitación, la evapotranspiración, la escorrentía superficial, la percolación profunda y el contenido de humedad del suelo. La gestión eficiente del riego depende en gran medida de la capacidad para modelar y anticipar cada uno de estos componentes.

Gamal et al. (2023) señalan que más del 40 % del agua redirigida para riego se pierde en el nivel de la finca, ya sea por percolación profunda o por escorrentía superficial. Esta pérdida no solo representa un despilfarro del recurso hídrico, sino que también reduce la eficiencia productiva del sistema agrícola. Frente a esta realidad, los sistemas de monitoreo inteligente del suelo emergen como herramientas esenciales para optimizar el uso del agua, al proporcionar datos en tiempo real sobre el estado hídrico del suelo y permitir ajustes precisos en los calendarios de riego.

Los métodos tradicionales para estimar la humedad del suelo, como el método gravimétrico o termogravimétrico, ofrecen una alta precisión pero presentan limitaciones importantes para la toma de decisiones en tiempo real, ya que los resultados deben obtenerse al menos 48 horas después del muestreo, lo que impide su uso en la programación inmediata del riego (Gamal et al., 2025). Por esta razón, la comunidad científica ha venido desarrollando técnicas volumétricas indirectas que permiten obtener estimaciones instantáneas del contenido de agua en el suelo, como los sensores dieléctricos y las técnicas de moderación de neutrones.

2.1.3. Fundamentos de la agricultura de precisión

La agricultura de precisión surgió a mediados de la década de 1980 como una respuesta a la necesidad de optimizar el uso de los recursos naturales en la producción agrícola, reducir el impacto ambiental de las prácticas agronómicas y mejorar los rendimientos de los cultivos. Desde entonces, esta disciplina ha evolucionado de manera sostenida, integrando tecnologías de información con la experiencia práctica en campo para hacer frente a los altos costos laborales y a las limitaciones de recursos que enfrentan tanto los países desarrollados como las naciones en vías de desarrollo (Zhang et al., 2024).

Uno de los pilares centrales de la agricultura de precisión es la irrigación de precisión, que busca maximizar la productividad de los cultivos mientras minimiza el uso del agua. Zhang et al. (2024) destacan que en contextos con escasez hídrica, estas prácticas tienen el potencial de mejorar la eficiencia en el uso del agua hasta en un 95 %, en comparación con el 50 % que se logra mediante los sistemas de irrigación superficial tradicionales. Sin embargo, la adopción de estas tecnologías sigue siendo incipiente en muchos países en desarrollo, donde persisten barreras relacionadas con el costo de los equipos, la pequeña escala de los predios y la falta de infraestructura adecuada.

La agricultura de precisión es especialmente relevante en el contexto del cambio climático, ya que busca aumentar la productividad, mejorar la resiliencia frente a la variabilidad climática y reducir la huella ambiental de la actividad agrícola. Las tecnologías asociadas a esta disciplina, como los sensores de monitoreo del suelo, los vehículos aéreos no tripulados (UAV), la teledetección satelital y los sistemas basados en el Internet de las Cosas (IoT), ofrecen datos oportunos, rentables y detallados sobre la humedad del suelo que mejoran significativamente la precisión del riego (Zhang et al., 2024).

2.1.4. Sistemas de monitoreo de humedad del suelo en agricultura de precisión

El monitoreo de la humedad del suelo dentro de la agricultura de precisión se apoya en cuatro grandes categorías tecnológicas. La primera corresponde a los sensores en movimiento continuo (on-the-go sensors), que se instalan en maquinaria agrícola como tractores o sembradoras para recolectar datos del suelo mientras la maquinaria recorre el campo. Estos sensores, entre los que se destacan los eléctricos, electromagnéticos, ópticos y radiométricos, permiten obtener datos directos y de alta precisión al mantener contacto físico con el suelo, sin ser afectados por la cobertura vegetal o los residuos de cultivos (Zhang et al., 2024).

La segunda categoría incluye los vehículos aéreos no tripulados (UAV o drones), que se han popularizado en la última década gracias a su flexibilidad y capacidad para acceder a zonas de difícil acceso. Los UAV permiten obtener imágenes hiperespectrales, multiespectrales y térmicas que pueden correlacionarse con el contenido de humedad del suelo mediante algoritmos de aprendizaje automático. En estudios como el de Zhang et al. (2024), el modelo de regresión de bosques aleatorios (RFR) aplicado a imágenes capturadas por UAV logró las estimaciones más precisas de humedad del suelo en comparación con otros algoritmos.

La tercera categoría corresponde a la teledetección satelital (RS), que permite el monitoreo a gran escala y ha sido utilizada desde hace más de 50 años en estudios de ciencia del suelo. Satélites como el Sentinel-1A y el Sentinel-2 han demostrado su utilidad para estimar la distribución espacial de la humedad del suelo en grandes áreas agrícolas, aunque su resolución espacial y temporal sigue siendo una limitación para aplicaciones de alta precisión en parcelas individuales (Zhang et al., 2024).

La cuarta y más relevante para el presente proyecto es la categoría de sistemas basados en IoT y redes de sensores inalámbricos (WSN), que se analiza en detalle en la sección siguiente.

2.1.5. Sistemas IoT y redes de sensores inalámbricos para el monitoreo del suelo

Los sistemas basados en el Internet de las Cosas (IoT) y las redes de sensores inalámbricos (WSN) han experimentado un crecimiento notable en la última década como soluciones para el monitoreo continuo y en tiempo real de la humedad del suelo en contextos de agricultura de precisión. Estos sistemas se componen de cuatro elementos principales: sensores y dispositivos, conectividad, procesamiento de datos e interfaz de usuario. Gracias a esta arquitectura, permiten recolectar, almacenar, procesar y compartir datos sobre la humedad del suelo a través de Internet de manera eficiente (Zhang et al., 2024).

Una de las ventajas más significativas de los sistemas IoT frente a los métodos tradicionales es su capacidad para proporcionar datos en tiempo real desde cualquier lugar, a través de aplicaciones móviles o plataformas en línea. Esto reduce de manera considerable los costos laborales y los tiempos de respuesta, al tiempo que permite a los investigadores y agricultores tomar decisiones informadas sobre el manejo del riego con base en las condiciones reales del suelo (Zhang et al., 2024).

En la revisión de Gamal et al. (2023) sobre sistemas de irrigación inteligente, se destaca que la tecnología IoT ha propiciado el desarrollo de sistemas de riego que se actualizan en tiempo real con el objetivo de reducir el estrés hídrico de los cultivos, aumentar los rendimientos y mejorar la calidad de la producción. Estos sistemas integran sensores de humedad del suelo con modelos de inteligencia artificial, entre ellos redes neuronales artificiales (ANN) y algoritmos de aprendizaje

profundo como LSTM y CNN, para generar predicciones confiables del estado hídrico del suelo y automatizar las decisiones de riego.

Los sensores capacitivos de humedad del suelo son los más utilizados en los sistemas IoT y WSN debido a su bajo costo y facilidad de instalación. Entre los modelos más referenciados en la literatura se encuentran el SEN0193 de DFRobot y los sensores de la serie EC-5 del grupo METER, cuya precisión en suelos minerales es de aproximadamente $\pm 0.03 \text{ m}^3/\text{m}^3$ (Zhang et al., 2024). La información recolectada por estos sensores se transmite mediante protocolos inalámbricos como LoRa, Zigbee, WiFi o BLE hasta una estación central o a la nube, donde los datos son procesados y analizados para generar recomendaciones de riego.

La integración de sensores IoT con modelos predictivos basados en aprendizaje automático representa el estado del arte en la gestión inteligente del riego. Zhang et al. (2024) señalan que la combinación de redes de sensores instaladas en campo con algoritmos de machine learning y deep learning permite no solo monitorear las condiciones actuales del suelo, sino también anticipar su comportamiento futuro, lo que abre la puerta a una gestión verdaderamente proactiva del recurso hídrico en los sistemas agrícolas.

2.2. Marco conceptual

El trabajo se apoya en varios conceptos que conviene precisar. La agricultura de precisión se entiende como un enfoque de manejo que utiliza información detallada del cultivo para ajustar las labores a las necesidades reales de cada momento, en lugar de aplicar prácticas uniformes. Ese enfoque se sostiene en gran medida sobre el Internet de las Cosas, que designa la red de dispositivos físicos capaces de medir su entorno y comunicar esos datos a través de internet (Soussi et al., 2024). En el campo, esos dispositivos son los sensores, elementos que

transforman una magnitud física, como la temperatura o la humedad, en una señal que un sistema electrónico puede leer y registrar.

La humedad del suelo, variable central de este estudio, se refiere a la cantidad de agua presente en el terreno y disponible para las plantas. Su comportamiento no es inmediato, ya que depende de la lluvia recibida y de la velocidad con que el suelo se seca por la evaporación y el consumo del cultivo, de modo que conserva memoria de lo ocurrido en jornadas anteriores. Por eso su estimación suele apoyarse no solo en el dato del momento, sino en variables construidas a partir del historial de precipitación (Islam et al., 2025).

Un modelo predictivo es una representación matemática que, a partir de un conjunto de variables de entrada, estima el valor de una variable de interés. Cuando ese modelo se construye dejando que un algoritmo aprenda las relaciones directamente de los datos, se habla de aprendizaje automático, una rama de la inteligencia artificial muy empleada en la predicción de variables agrícolas (Afzal et al., 2025). La confiabilidad de estos modelos se juzga con métricas estadísticas que cuantifican la diferencia entre los valores estimados y los reales.

El manejo de la información suma dos conceptos más. La computación en la nube permite almacenar y procesar datos en servidores remotos accesibles por internet, lo que en este proyecto se concreta en una plataforma que recibe y guarda las lecturas de los sensores. Antes de analizar esa información se aplica un proceso conocido como ETL, sigla de extracción, transformación y carga, que reúne los datos desde sus fuentes, los depura y los organiza en un formato uniforme y listo para el análisis. En conjunto, estos conceptos articulan el recorrido que va desde la medición en el cultivo hasta la obtención de un modelo capaz de anticipar la humedad del suelo.

2.3. Marco legal

El proyecto se inscribe dentro del marco normativo ambiental colombiano, cuyo fundamento se encuentra en la Constitución Política de 1991. En su artículo 79 se reconoce el derecho de todas las personas a gozar de un ambiente sano y se asigna al Estado el deber de proteger la diversidad e integridad del entorno (Constitución Política de Colombia, 1991). A partir de ese mandato, la Ley 99 de 1993 organiza el Sistema Nacional Ambiental, crea el Ministerio del Medio Ambiente y da vida al Instituto de Hidrología, Meteorología y Estudios Ambientales, la entidad que produce y administra la información climática que este trabajo utiliza. De ahí que el uso de los registros de precipitación del IDEAM se apoye en una fuente oficial reconocida por la legislación.

Como el sistema busca respaldar un manejo más racional del riego en el cultivo, resulta especialmente pertinente la Ley 373 de 1997, que establece el programa para el uso eficiente y ahorro del agua y obliga a los distintos usuarios del recurso hídrico a adoptar acciones dirigidas a reducir su desperdicio. El propósito del proyecto, que consiste en anticipar el estado de humedad del suelo para tomar mejores decisiones de riego, se alinea de forma directa con el espíritu de esa norma.

En el terreno de las telecomunicaciones, el sistema transmite sus datos por Wi-Fi, tecnología que opera en las bandas de 2,4 y 5 GHz. Esas bandas pertenecen a la modalidad de uso libre del espectro radioeléctrico, de manera que su utilización no exige un permiso previo, según lo dispuesto en la Ley 1341 de 2009 y en la Resolución 105 de 2020 de la Agencia Nacional del Espectro, que actualiza el Cuadro Nacional de Atribución de Bandas de Frecuencias. Esto le permite al proyecto emplear la conectividad inalámbrica sin trámites adicionales ante la autoridad del espectro.

el sistema se construye con dispositivos electrónicos, como el microcontrolador, los sensores y la batería, que al terminar su vida útil se convierten en residuos de aparatos eléctricos y electrónicos. Su manejo lo regula la Ley 1672

de 2013, que fija los lineamientos para la gestión integral de estos residuos y adopta el principio de responsabilidad extendida del productor, el proyecto prevé que, una vez agotada la vida útil de sus componentes, estos se entreguen a través de los canales de gestión dispuestos para los RAEE y no se descarten como basura común.

2.4. Marco ambiental

Desde la perspectiva ambiental, el proyecto responde a un problema que la propia legislación colombiana reconoce como prioritario: la vulnerabilidad frente a la variabilidad y el cambio climático. La Ley 1931 de 2018 establece las directrices para la gestión del cambio climático y busca reducir la exposición de la población y de los ecosistemas a sus efectos. En la producción cafetera, esa vulnerabilidad se manifiesta en lluvias irregulares y periodos secos que golpean el rendimiento del cultivo, y es precisamente ahí donde un sistema de monitoreo entrega información para adaptarse mejor a esas condiciones cambiantes.

El principal aporte ambiental del sistema tiene que ver con el agua. Al ofrecer datos sobre la humedad del suelo, la herramienta ayuda a decidir cuándo el riego hace falta de verdad y cuándo no, con lo que se evita aplicar agua en exceso. Ese uso más consciente del recurso concuerda con el principio, recogido en la Ley 99 de 1993, de que el agua debe aprovecharse de manera racional, y con las metas de ahorro que impulsa la Ley 373 de 1997. En una zona agrícola, cada mejora en la eficiencia del riego contribuye a conservar las fuentes hídricas del territorio.

El trabajo también reconoce que su operación deja una huella ambiental, aunque modesta. Los componentes electrónicos contienen materiales que, mal dispuestos, pueden contaminar el suelo y el agua, y por eso su manejo responsable al final de la vida útil forma parte de las consideraciones ambientales del proyecto, en línea con la política nacional de residuos electrónicos. Alimentar el sistema con

una batería portátil recargable, en vez de pilas desechables, disminuye la cantidad de residuos que se generan durante la etapa de recolección.

3. DISEÑO DE LA INVESTIGACIÓN

La investigación es de tipo descriptivo con un componente correlacional, pues busca caracterizar el comportamiento de la humedad del suelo y, al mismo tiempo, establecer su relación con las demás variables ambientales para poder predecirla. El enfoque es cuantitativo, ya que todo el trabajo se sustenta en la medición de variables numéricas y en su tratamiento estadístico. El método es principalmente deductivo y de análisis, en la medida en que parte de relaciones ambientales conocidas y las contrasta con los datos recolectados en campo.

Las técnicas empleadas combinan la observación instrumental y el análisis de datos. La observación se realiza mediante un sistema de sensores que mide de forma automática las variables del cultivo, mientras que el análisis se apoya en herramientas de programación para depurar la información, construir variables y ajustar los modelos. A esa recolección propia se suma el uso de datos secundarios, correspondientes a los registros de precipitación que publica el IDEAM.

En cuanto al diseño experimental, el entorno de estudio es un cultivo de café situado en la vereda Santa Inés, en San Vicente de Chucurí. Los materiales principales son un microcontrolador ESP32, un sensor capacitivo de humedad del suelo, un sensor de temperatura del suelo y un sensor de temperatura y humedad del aire, alimentados por una batería portátil. Las condiciones de operación quedan definidas por una jornada de recolección de doce horas diarias y un intervalo de medición de cinco minutos, a lo largo de un periodo de tres meses. El muestreo es de carácter temporal, pues cada registro constituye una observación de las condiciones ambientales en un instante determinado.

El procedimiento se organiza en cuatro fases que corresponden a los objetivos específicos del trabajo. La primera consiste en el diseño y montaje del sistema de monitoreo. La segunda comprende la integración de los datos propios

con la información del IDEAM para conformar un único conjunto ambiental. La tercera aborda el desarrollo de los modelos predictivos a partir de ese conjunto. La cuarta corresponde a la evaluación del desempeño de los modelos mediante métricas estadísticas, con el fin de determinar cuál ofrece la mejor estimación de la humedad del suelo. Cada fase se detalla en la sección de desarrollo del trabajo de grado.

4. DESARROLLO DEL TRABAJO DE GRADO

4.1. Diseño del sistema de monitoreo IoT

El punto de partida del diseño fue una idea sencilla: contar con un dispositivo económico que registrara, sin necesidad de vigilancia constante, las variables ambientales que influyen en el estado hídrico del suelo del cafetal. Apoyarse en un enfoque de Internet de las Cosas tiene sentido porque estas arquitecturas recogen información por sí solas y durante períodos largos, algo que las mediciones manuales, tomadas de vez en cuando, no logran. La literatura respalda el camino elegido, pues varios autores han documentado que los sistemas embebidos de bajo costo dan buenos resultados en el seguimiento de la humedad del suelo, incluso donde los recursos son escasos (Abdelmoneim et al., 2023; Sudarmaji et al., 2024).

Antes de elegir el hardware hubo que decidir qué se iba a medir. La humedad del suelo quedó como la variable central, ya que es la que el modelo debe predecir, y a su alrededor se escogieron otras que la afectan de forma directa: la temperatura del suelo, la del aire y la humedad relativa del ambiente. Faltaba la lluvia, tan determinante para el suelo como imposible de captar con este circuito, de modo que esa información se tomó de los registros del Instituto de Hidrología, Meteorología y Estudios Ambientales.

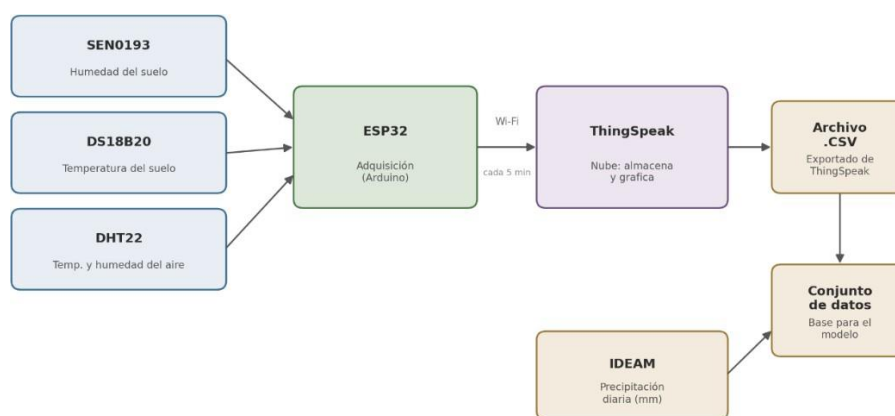
El hardware se organiza en torno al microcontrolador ESP32, que lee las señales de los sensores y las traduce a valores digitales. Su programación se hizo en el entorno de Arduino. Para no perder esas lecturas, un programa en Python recibe lo que el microcontrolador envía por medio de Wi Fi y lo guarda en el computador. La mezcla de hardware barato y software libre no es nueva; aparece una y otra vez en proyectos de agricultura de precisión con propósitos parecidos al de este trabajo (Afzal et al., 2025).

Varias consideraciones prácticas orientaron las decisiones. Interesaba que los componentes fueran baratos y fáciles de conseguir en el país, que se entendieran sin problema con el ESP32 y que aguantaran la intemperie de un cultivo a cielo abierto. Como también se pensó en que el montaje pudiera repetirse en otras fincas, se prefirieron piezas conocidas y bien documentadas, sencillas de reemplazar para quien quiera reproducirlo.

4.2. Arquitectura general del sistema

La arquitectura se puede leer como una cadena de cuatro etapas que acompaña a la información desde que se captura en el cultivo hasta que queda guardada en un formato listo para analizar. La Figura 1 muestra esa cadena y la manera en que se relacionan los sensores, el microcontrolador, el computador y la fuente externa de datos.

Figura 1. Arquitectura general del sistema de monitoreo



Fuente: Autores

Todo empieza con la adquisición. Los tres sensores instalados en el cafetal miden al mismo tiempo la humedad del suelo, su temperatura y las condiciones del aire, y cada uno le entrega su lectura al ESP32. El microcontrolador es el centro de la operación: recibe las señales, las convierte y las ordena en un registro. Aquí vale la pena señalar una diferencia técnica, y es que el sensor de humedad del suelo manda una señal analógica que el ESP32 digitaliza con su conversor interno, mientras que los de temperatura y humedad del aire ya se comunican en formato digital.

Con las lecturas listas, el ESP32 las envía por Wi Fi hasta un computador, donde un programa escrito en C con diferentes librerías las va leyendo y escribiendo en un archivo de valores separados por comas. A cada registro se le añade su marca de tiempo. Ese detalle, que parece menor, es lo que sostiene todo lo demás, porque sin una hora confiable no habría forma de emparejar más adelante las mediciones del sistema con la lluvia del IDEAM.

La última etapa arma el conjunto de datos. Los registros del archivo se cruzan con la serie de precipitación descargada del IDEAM tomando la fecha y la hora como llave, y de ahí sale una base ambiental que reúne, sobre una sola línea de tiempo, lo medido en campo y la lluvia de la estación de referencia. Para que el archivo no crezca sin control ni se llene de lecturas casi idénticas, se guardó un dato cada cierto intervalo, ajustado a la velocidad con la que las variables cambian de verdad.

4.3. Selección de sensores y componentes electrónicos

Escoger los sensores y los demás componentes fue cuestión de sopesar varios factores: el rango de medición, la precisión, el precio, la compatibilidad con el microcontrolador y la facilidad para conseguirlos. La regla de fondo fue que cada dispositivo cumpliera bien con la variable que le tocaba y que no fallara al quedar expuesto en el cultivo.

El ESP32 se quedó con el papel de unidad central por su capacidad de cómputo, por traer un conversor analógico digital incorporado y por entenderse con una buena cantidad de sensores y librerías abiertas. No es una apuesta arriesgada: este microcontrolador aparece con insistencia en proyectos de monitoreo agrícola, sobre todo por su bajo consumo y por lo que ofrece frente a lo poco que cuesta (Abdelmoneim et al., 2023).

Para la humedad del suelo se optó por el sensor capacitivo SEN0193. La razón es concreta: a diferencia de los resistivos, los sensores capacitivos no exponen sus electrodos al suelo, lo que los protege de la corrosión y alarga su vida útil (Sudarmaji et al., 2024). En un montaje pensado para pasar semanas enterrado, esa diferencia termina pesando.

La temperatura del suelo quedó a cargo del DS18B20, un sensor digital alojado en una sonda impermeable, cómoda de enterrar y de mantener en contacto directo con la tierra. Las condiciones del aire, temperatura y humedad relativa, las cubre el DHT22, que reúne ambas medidas en un solo integrado y las transmite por una única línea de datos. La Tabla 1 recoge la función y las características de cada uno.

Tabla 1.

Componentes electrónicos del sistema de monitoreo

Componente	Función y variable medida	Rango de medición	Precisión y resolución	Interfaz	Alimentación
ESP32	Procesamiento, adquisición y transmisión de datos	No aplica (unidad de control)	Conversor analógico digital de 12 bits (0 a 4095)	Wi-Fi 2,4 GHz y pines GPIO, programable en Arduino	3,3 V
SEN0193	Humedad del suelo	0 a 100 % (valor relativo, requiere calibración)	Salida analógica continua	Analógica, lectura por ADC	3,3 a 5,5 V
DS18B20	Temperatura del suelo	-55 a 125 °C	±0,5 °C (entre -10 y	Digital 1-Wire (OneWire)	3,0 a 5,5 V

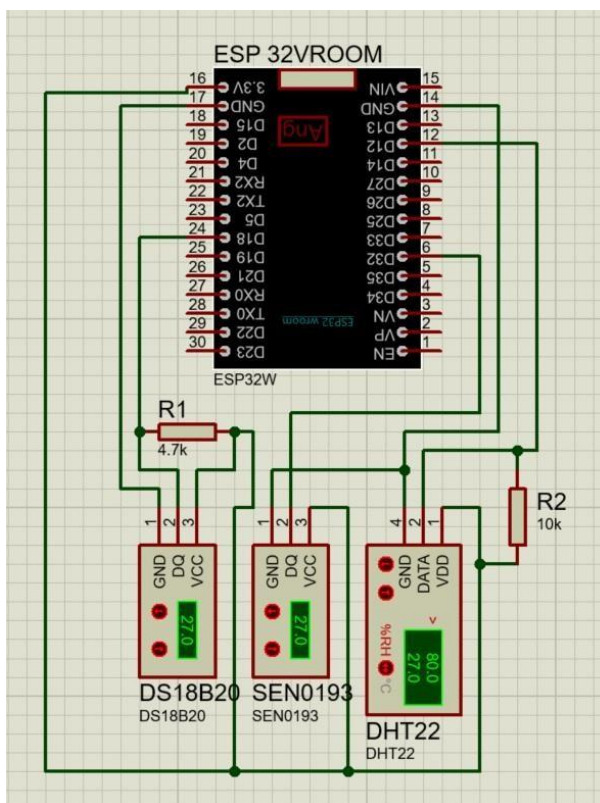
			85 °C), hasta 0,0625 °C		
DHT22	Temperatura y humedad del aire	Temperatura: - 40 a 80 °C; humedad: 0 a 100 %	±0,5 °C y ±2 % de humedad relativa; resolución 0,1	Digital de una sola línea	3,3 a 6 V

Nota. Elaboración propia a partir de las especificaciones técnicas de cada componente.

4.4. Diseño del circuito electrónico

El circuito se organizó alrededor del ESP32 como unidad central, al que se conectaron los tres sensores encargados de medir las variables ambientales. La Figura 2 presenta el esquema eléctrico completo. El sensor de humedad del suelo SEN0193 se enlazó a una entrada analógica del microcontrolador, mientras que el DS18B20 y el DHT22, que son digitales, se conectaron a sendos pines de datos. Todos los dispositivos comparten la misma referencia de tierra y se alimentan con los 3,3 V que entrega la placa, que es el voltaje de trabajo de los tres sensores.

Figura 2. Esquema eléctrico del sistema de monitoreo



Fuente: Autores

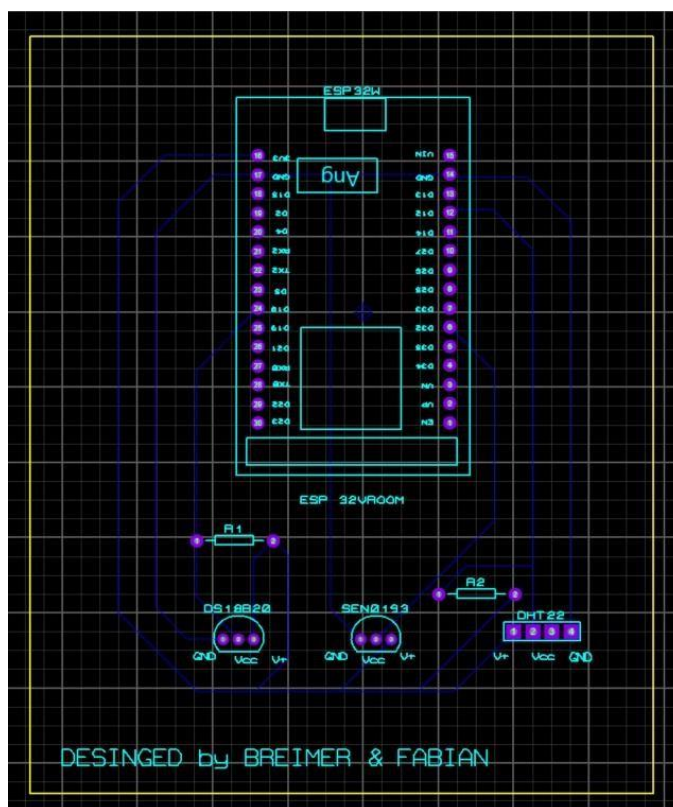
Dos resistencias cumplen un papel clave en la estabilidad de las lecturas. La R1, de 4,7 kΩ, funciona como resistencia pull-up en la línea de datos del DS18B20, tal como lo exige el protocolo OneWire con el que se comunica ese sensor; sin ella el bus quedaría en un estado indefinido y las lecturas serían erráticas. La R2, de 10 kΩ, hace lo mismo en la línea del DHT22, manteniendo el nivel alto en reposo para asegurar una comunicación fiable (Analog Devices, s.f.; DFRobot, s.f.).

La alimentación merece una mención aparte, porque en el cultivo no hay una toma de corriente cercana. Por eso el sistema se energiza con una batería portátil conectada al puerto USB de la placa, que internamente regula la tensión a los 3,3 V que necesitan el microcontrolador y los sensores. Una fuente autónoma como

esta resulta la opción más práctica para operar en campo sin depender de la red eléctrica.

Con el circuito validado en simulación, se diseñó una placa de circuito impreso para reemplazar el cableado de prototipo por una tarjeta compacta y más resistente frente a las vibraciones y la humedad del entorno. La Figura 3 muestra el trazado de la placa, donde se distribuyeron el ESP32, las resistencias y los conectores de los tres sensores. Pasar a una placa impresa reduce el riesgo de falsos contactos, que es uno de los problemas más frecuentes en montajes sobre protoboard expuestos a la intemperie.

Figura 3. Diseño de la placa de circuito impreso



Fuente: Autores

4.5. Configuración del microcontrolador ESP32

La programación del ESP32 se realiza en el entorno de desarrollo de Arduino, apoyándose en librerías de código abierto para cada sensor y para la conexión con el servicio en la nube. Cada pin se configuró según el tipo de señal que debía manejar. El pin asignado al SEN0193 se declaró como entrada analógica, aprovechando el conversor analógico digital interno del microcontrolador, mientras que los del DS18B20 y el DHT22 se configuraron para comunicación digital, cada uno con la librería propia de su protocolo.

Al asignar los pines se tuvo en cuenta una particularidad del ESP32: cuando su módulo Wi-Fi está activo, solo queda disponible el primer bloque de conversores analógicos. Por esa razón el sensor de humedad del suelo se conectó a un pin del bloque ADC1 y se evitaron los del bloque ADC2, que dejan de servir como entradas analógicas mientras hay conexión inalámbrica (Espressif Systems, s.f.). Pasar por alto este detalle habría dejado sin lectura justamente la variable objetivo del proyecto.

La conectividad se resuelve con la interfaz Wi-Fi que trae integrada la placa. Al encender, el microcontrolador se conecta a la red disponible en la finca y enseguida sincroniza su reloj con un servidor de tiempo en internet mediante el protocolo NTP, ajustado a la zona horaria de Colombia. Esa sincronización es muy importante porque gracias a ella cada lectura queda asociada a una hora real y confiable, condición indispensable para poder cruzar después los datos con los registros de precipitación del IDEAM.

4.5.1. Código usado

Para programar el microcontrolador ESP32 el código está compuesto por los siguientes elementos:

```
#include <WiFi.h>
```

```
#include <OneWire.h>
```

```
#include <DallasTemperature.h>
```

```
#include <DHT.h>
```

```
#include "ThingSpeak.h"
```

```
#include <time.h>
```

El programa comienza incluyendo las librerías externas que necesita. WiFi.h gestiona la conexión inalámbrica del ESP32. OneWire y DallasTemperature trabajan en conjunto: la primera implementa el protocolo OneWire, que es el bus por el que se comunica el DS18B20, y la segunda traduce esas señales en lecturas de temperatura ya procesadas. DHT.h controla el DHT22. ThingSpeak.h construye las peticiones hacia la nube sin que sea necesario escribir el HTTP manualmente. Por último, time.h aporta las funciones del sistema para solicitar y dar formato a la hora obtenida por internet.

```
const char* WIFI_SSID = "nombre del Wifi";
```

```
const char* WIFI_PASS = "contraseña del wifi";
```

```
unsigned long CANAL_ID = canal_ID;
```

```
const char* WRITE_API_KEY = "api Key de thingspeak";
```

Este bloque reúne los únicos parámetros que cambian de un montaje a otro, por eso se ubica al inicio. Contiene el nombre y la clave de la red, junto con los dos identificadores del canal de ThingSpeak: el CANAL_ID, un número que identifica el canal, y la WRITE_API_KEY, la llave que autoriza la escritura en él. El identificador se declara como unsigned long por tratarse de un número grande, mientras que los textos se declaran como const char*, es decir cadenas fijas que no se modifican durante la ejecución.

```
#define DS18B20_PIN 18
```

```
#define SEN0193_PIN 32
```

```
#define DHT_PIN 27
```

```
#define DHT_TYPE DHT22
```

La directiva `#define` asigna nombres a los números de pin, lo que evita tener que recordarlos a lo largo del código. El DS18B20 se conecta al GPIO 18, el DHT22 al 27 y el SEN0193 al 32. La elección del 32 responde a una restricción del ESP32: pertenece al bloque ADC1, y con el Wi-Fi activo los pines del bloque ADC2 dejan de funcionar como entradas analógicas. Como el SEN0193 entrega una señal analógica, debe conectarse necesariamente a un pin ADC1. Conviene aclarar que las resistencias pull-up R1 y R2 del esquema pertenecen al hardware; el DS18B20 y el DHT22 las requieren en sus líneas de datos, pero residen en la placa y no en el código.

```
const int SOIL_DRY = 3200;
const int SOIL_WET = 1500;
```

Estos dos valores permiten que el porcentaje de humedad tenga sentido físico. El ESP32 no mide humedad de forma directa, sino que lee un valor crudo entre 0 y 4095. Con el sensor en seco ese valor ronda 3200 y, sumergido, desciende a cerca de 1500. A partir de esos dos extremos el programa convierte después cualquier lectura intermedia en un porcentaje. Se trata de valores de referencia que deben ajustarse a los del sensor real, medidos en seco y en agua.

```
const unsigned long INTERVALO_MS = 300000UL; // 5 min

const long GMT_OFFSET_SEG = -5 * 3600;
const int DST_OFFSET_SEG = 0;
```

INTERVALO_MS define cada cuánto se realiza una medición, expresado en milisegundos: 300000 equivalen a cinco minutos. El sufijo UL obliga a tratar el número como unsigned long para evitar desbordamientos. Las dos líneas siguientes establecen la zona horaria del reloj sincronizado por internet: Colombia se encuentra a cinco horas menos que UTC, de ahí el valor $-5 * 3600$ en segundos, y no aplica horario de verano, razón por la cual el segundo ajuste es cero.

```
DHT dht(DHT_PIN, DHT_TYPE);
OneWire oneWire(DS18B20_PIN);
DallasTemperature ds18b20(&oneWire);
WiFiClient cliente;
```

```
unsigned long ultimaLectura = 0;
bool primeraVez = true;
```

En este punto se crean los objetos que representan a cada sensor y a la conexión de red. El objeto dht se comunica con el DHT22, ds18b20 con el sensor de suelo apoyándose en el bus oneWire, y cliente es la conexión que ThingSpeak utiliza internamente. Las dos últimas variables gobiernan el control del tiempo: ultimaLectura guarda el instante de la última medición y primeraVez asegura que la primera lectura ocurra apenas el sistema arranca, sin esperar el intervalo completo.

```
void conectarWiFi() {
    if (WiFi.status() == WL_CONNECTED) return;
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_SSID, WIFI_PASS);
    int intentos = 0;
    while (WiFi.status() != WL_CONNECTED && intentos < 30) {
```

```

        delay(500);
        intentos++;
    }
    ...
}
```

Esta función encapsula la conexión para poder reutilizarla. Primero comprueba si el enlace ya está activo y, en ese caso, termina de inmediato. Si no lo está, coloca el ESP32 en modo estación (WIFI_STA), con lo que se conecta a un router en lugar de crear su propia red, e inicia el intento de conexión. El ciclo while espera medio segundo por intento, hasta un máximo de treinta, es decir unos quince segundos, de modo que el programa no queda bloqueado indefinidamente si la red no aparece. Como la función se invoca tanto al arrancar como dentro del bucle principal, el sistema recupera la conexión por sí mismo si el Wi-Fi se interrumpe durante la campaña.

```

String horaActual() {
    struct tm t;
    if (!getLocalTime(&t, 5000)) return "sin-hora";
    char buf[20];
    strftime(buf, sizeof(buf), "%Y-%m-%d %H:%M:%S", &t);
    return String(buf);
}
```

Esta función devuelve la hora ya sincronizada por internet, con el formato AAAA-MM-DD HH:MM:SS. getLocalTime intenta obtenerla esperando hasta cinco segundos y, si aún no está disponible, devuelve el texto sin-hora en vez de bloquear

el programa. Esta marca temporal resulta esencial, ya que es la que permite después cruzar cada lectura con la precipitación del IDEAM.

```
void setup() {  
  Serial.begin(115200);  
  delay(3000);  
  dht.begin();  
  ds18b20.begin();  
  conectarWiFi();  
  configTime(GMT_OFFSET_SEG, DST_OFFSET_SEG, "pool.ntp.org",  
"time.nist.gov");  
  ...  
  ThingSpeak.begin(cliente);  
}
```

La función setup se ejecuta una sola vez al encender el equipo. Abre la comunicación serial para mostrar mensajes en el computador, espera tres segundos a que el DHT22 se estabilice, pues es un sensor lento al arrancar, inicializa los dos sensores digitales, establece la conexión Wi-Fi e inicia la sincronización de la hora con dos servidores NTP. Finalmente, ThingSpeak.begin(cliente) deja preparada la comunicación con la nube. Todo lo que contiene esta función corresponde a la preparación previa a la operación continua.

Esta función se repite de forma indefinida y concentra la operación del sistema. A continuación se describe por partes.

```
if (WiFi.status() != WL_CONNECTED) conectarWiFi();
```

```
if (primeraVez || (millis() - ultimaLectura >= INTERVALO_MS)) {
    primeraVez = false;
    ultimaLectura = millis();
```

Al inicio de cada vuelta el programa revisa el estado del Wi-Fi y reconecta si es necesario. Luego aplica el control del tiempo, donde se toma una decisión de diseño relevante: en lugar de un delay de cinco minutos, que congelaría el equipo, se compara millis(), los milisegundos transcurridos desde el encendido, con el instante de la última lectura. De esta manera el programa permanece activo entre mediciones y puede atender la reconexión del Wi-Fi. La condición primeraVez fuerza que la primera captura sea inmediata.

```
ds18b20.requestTemperatures();
float tempSuelo = ds18b20.getTempCByIndex(0);
bool okSuelo = (tempSuelo != DEVICE_DISCONNECTED_C &&
    tempSuelo > -40 && tempSuelo < 85);
```

El programa solicita la temperatura del suelo y la valida. DEVICE_DISCONNECTED_C es el valor que la librería devuelve cuando el sensor no responde, por lo que okSuelo solo resulta verdadero si el dato existe y se ubica en un rango físicamente razonable.

```
float humAire = dht.readHumidity();
float tempAire = dht.readTemperature();
bool okDHT = (!isnan(humAire) && !isnan(tempAire) &&
    tempAire > -40 && tempAire < 80 &&
    humAire >= 0 && humAire <= 100);
```

El DHT22 recibe el mismo tratamiento. La función `isnan` detecta cuando el sensor entrega un valor inválido, algo que ocurre de manera ocasional, y las comparaciones descartan lecturas imposibles. Solo si todas las condiciones se cumplen, `okDHT` queda en verdadero.

```
int soilRaw = analogRead(SEN0193_PIN);
int soilPct = map(soilRaw, SOIL_DRY, SOIL_WET, 0, 100);
soilPct = constrain(soilPct, 0, 100);
```

En este fragmento se emplea la calibración. `analogRead` toma el valor crudo, `map` lo convierte en porcentaje usando las dos referencias de seco y mojado, y `constrain` recorta el resultado para que nunca salga del rango de 0 a 100, incluso si una lectura supera los extremos.

```
if (WiFi.status() == WL_CONNECTED) {
  if (okSuelo) ThingSpeak.setField(1, tempSuelo);
  if (okDHT) ThingSpeak.setField(2, tempAire);
  if (okDHT) ThingSpeak.setField(3, humAire);
  ThingSpeak.setField(4, (float)soilPct);
  int codigo = ThingSpeak.writeFields(CANAL_ID, WRITE_API_KEY);
  ...
}
```

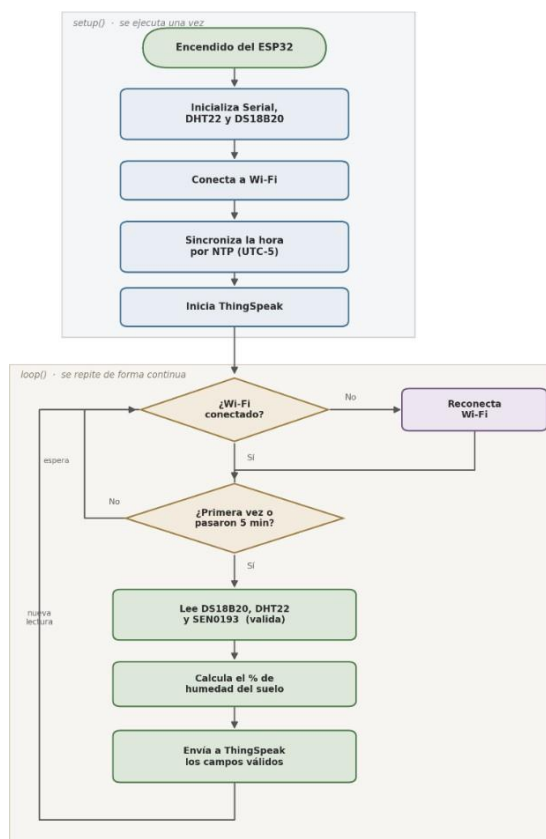
Este bloque realiza el envío. Cada variable se asigna a su campo mediante `setField`, pero con un matiz importante: los campos se cargan únicamente si el sensor entregó un dato válido. Si el DHT22 falla, sus campos 2 y 3 no se envían y quedan vacíos en el registro, que es precisamente el comportamiento buscado para

reflejar los fallos reales. writeFields ejecuta el envío y devuelve un código; el valor 200 indica que ThingSpeak lo recibió correctamente.

```
    delay(200);  
}
```

Al cierre, un delay mínimo de doscientos milisegundos ofrece un respiro al procesador. Este retardo no altera el intervalo de cinco minutos, que sigue determinado por la comparación con millis(). Toda la lógica empleada en este algoritmo se puede resumir en el siguiente diagrama de bloques:

Figura 4. diagrama de bloques lógica suada en el algoritmo



Fuente: Autores

Nota. El código empleado se encuentra disponible como anexo A

4.6. Diseño de la carcasa de protección

La tarjeta de circuito impreso resuelve la parte eléctrica del sistema, pero deja abierto un problema que el cafetal impone por sí solo. El equipo permanece semanas en un cultivo a cielo abierto, sin techo ni caseta, expuesto a la lluvia, al polvo, a la radiación solar y a los insectos. Ninguna de esas condiciones resulta compatible con una tarjeta descubierta, de modo que el sistema requiere un alojamiento propio. La solución adoptada consiste en una carcasa diseñada a la

medida de la placa y fabricada por manufactura aditiva, un recurso frecuente en los nodos de monitoreo agrícola de bajo costo porque permite ajustar la geometría a las dimensiones exactas del circuito y de los sensores que se utilizan (Dong y Hansen, 2024).

La manufactura aditiva ofrece además una ventaja que encaja con el propósito de este trabajo. Una caja comercial con grado de protección equivalente cuesta varias veces lo que cuesta el filamento necesario para imprimirla, y obliga a adaptar el montaje a un volumen que rara vez coincide con el de la tarjeta. Al imprimir la pieza, el diseño se ajusta a la placa y no al revés, y el archivo queda disponible para que cualquier persona interesada reproduzca el nodo en otra finca, que es justamente uno de los criterios que orientan el proyecto (Afzal et al., 2025).

4.6.1. Requerimientos de diseño

El diseño de la carcasa responde a cinco exigencias concretas. La primera consiste en proteger la electrónica del agua y del polvo, pues la humedad condensada dentro de la caja causa más fallas que la lluvia directa. La segunda obliga a dar salida a los cables de los tres sensores, ya que dos de ellos permanecen enterrados y el tercero debe quedar en contacto con el aire. La tercera exige mantener accesible el puerto USB por el que la batería portátil alimenta la placa, de modo que el reemplazo de la fuente no requiera desarmar el conjunto. La cuarta impone que la tarjeta quede sujeta y separada del piso de la caja, para que las vibraciones y los movimientos durante el transporte no generen falsos contactos. La quinta, finalmente, pide que la pieza se imprima en una impresora doméstica corriente, sin soportes complejos ni tiempos de fabricación excesivos.

A estas exigencias se suma una consideración de orden práctico. El mantenimiento en campo ocurre de pie, entre surcos, muchas veces con las manos sucias y sin herramientas a la mano, así que el cierre de la caja debe abrirse y volver a cerrarse con facilidad. Esa restricción determina varias de las decisiones geométricas que se describen a continuación.

4.6.2. Dimensionamiento y geometría

El punto de partida del dimensionamiento es la tarjeta de circuito impreso, que mide 90×107 mm. Alrededor de esa huella se define un interior útil de 92×109 mm, es decir, un milímetro de holgura por cada lado. Ese margen resulta suficiente para absorber la contracción del material al enfriarse y las tolerancias propias de la impresora, sin dejar un juego que permita el desplazamiento de la placa. Con un espesor de pared y de piso de 2,4 mm y una altura de 40 mm con la tapa puesta, el volumen exterior de la caja cerrada queda en $101,6 \times 118,6 \times 40$ mm. Siguiente figura presenta el modelo tridimensional con la PCB de referencia en su posición.

Figura 5. Modelo tridimensional de la carcasa de protección



Fuente: Autores

El espesor de 2,4 mm no es un número arbitrario. Equivale a seis perímetros impresos con una boquilla de 0,4 mm, que es la configuración estándar de la mayoría de las impresoras de escritorio. Un número alto de perímetros contiguos reduce los caminos por los que el agua podría filtrarse entre capas y, al mismo tiempo, entrega una pared con rigidez suficiente para que la caja no se deforme al

manipularla. La alternativa de aumentar el relleno interno resulta menos eficaz para este fin, porque en una pieza de pared delgada la estanqueidad depende del contorno y no del interior.

La pared frontal concentra tres aberturas de 20 × 15 mm, una por cada sensor. Agruparlas en un solo costado responde a la orientación prevista para el montaje, con esa cara mirando hacia el suelo, de manera que los cables del sensor capacitivo y de la sonda de temperatura descienden directamente hacia el punto de medición y el del DHT22 queda libre para ubicarse en el aire. Una ventana adicional de 14 × 9 mm da paso al cable USB de la batería. En el piso de la caja se levantan postes con perforación de 3,2 mm para tornillos M3, que sostienen la tarjeta a cierta distancia del fondo y dejan una cámara de aire que evita el contacto entre las pistas y cualquier condensación acumulada. La siguiente tabla resume las dimensiones y el criterio que sustenta cada una.

Tabla 2.

Características geométricas de la carcasa de protección

Elemento	Dimensión	Criterio de diseño
Caja exterior	101,6 × 118,6 × 40 mm	Volumen mínimo que aloja la tarjeta y el conjunto de conectores sin sobredimensionar la pieza ni el tiempo de impresión
Interior útil	92 × 109 mm	Corresponde a la PCB de 90 × 107 mm más 1 mm de holgura perimetral, suficiente para absorber la contracción del material y las tolerancias de la impresora
Espesor de pared y piso	2,4 mm	Equivale a seis perímetros con una boquilla de 0,4 mm, lo que aporta rigidez y limita la porosidad entre capas
Salidas de sensores	3 aberturas de 20 × 15 mm	Una por cada sensor, agrupadas en la pared frontal para que los cables salgan por el costado que mira hacia el suelo
Ventana de alimentación	14 × 9 mm	Permite conectar el cable USB de la batería portátil sin abrir la caja
Fijación de la tarjeta	Postes para tornillo M3, Ø 3,2 mm	Separan la PCB del piso y evitan que se desplace con las vibraciones

Cierre	Tapa a presión	Facilita el mantenimiento en campo sin herramientas y sin tornillos que se pierdan o se oxiden
--------	----------------	--

Nota: Elaboración propia a partir del modelo tridimensional del encerramiento.

La tapa cierra a presión sobre el cuerpo de la caja. Esta decisión privilegia el mantenimiento por encima del sellado absoluto, ya que un cierre atornillado ofrece mejor compresión pero exige herramientas y multiplica los puntos por donde entra el agua. En un prototipo que se revisa con frecuencia durante la campaña de medición, la facilidad de apertura pesa más que la hermeticidad teórica.

4.7. Selección del material y parámetros de impresión

La elección del filamento condiciona la vida útil de la pieza más que cualquier otra decisión del encerramiento. Los tres materiales habituales en impresión por deposición fundida se comportan de manera muy distinta frente al sol y al calor. El PLA se imprime sin dificultad y con buen acabado, pero su temperatura de transición vítrea ronda los 60 °C, un valor que la superficie de una caja expuesta al sol directo alcanza sin problema, y su resistencia a la radiación ultravioleta es baja, por lo que la pieza se vuelve quebradiza en pocos meses. El PETG tolera temperaturas cercanas a 85 °C, absorbe poca agua y presenta buena adherencia entre capas, características que lo hacen adecuado para piezas funcionales a la intemperie. El ASA incorpora estabilizadores frente a la radiación solar y conserva sus propiedades durante varios años, aunque su impresión exige una cámara cerrada para evitar el alabeo (Filamentive, 2025; Sovol3D, 2026). La siguiente tabla compara los tres materiales.

Tabla 3.

Comparación de materiales de impresión para uso a la intemperie

Material	Tg aproximada	Resistencia UV	Observaciones para el uso en campo
PLA	57 a 64 °C	Baja	Se imprime con facilidad y sin cámara cerrada, pero se reblandece con el calor acumulado bajo el sol directo y se vuelve quebradizo tras algunos meses de exposición
PETG	80 a 85 °C	Media	Buena tolerancia al calor y a la humedad, baja absorción de agua y adherencia entre capas favorable para piezas estancas
ASA	Cercana a 100 °C	Alta	Incorpora estabilizadores frente a la radiación solar y conserva sus propiedades durante varios años a la intemperie, aunque exige una impresora con cámara cerrada

Nota: Elaboración propia a partir de Filamentive (2025) y Sovol3D (2026).

El prototipo que se presenta en este documento se fabrica en [PLA / PETG: confirmar el filamento empleado], material disponible en el laboratorio al momento de la impresión. Para el despliegue prolongado en el cafetal se recomienda el PETG, que ofrece un compromiso razonable entre facilidad de impresión y desempeño frente al calor y la humedad, o bien el ASA cuando se dispone de una impresora con cámara cerrada. Una alternativa complementaria consiste en aplicar sobre la pieza terminada dos o tres capas de barniz de poliuretano con filtro ultravioleta, recurso que extiende de manera apreciable la vida de los encerramientos impresos que permanecen al sol (Zbotic, 2026).

Los parámetros de impresión siguen la misma lógica que el espesor de pared. Una altura de capa moderada, seis perímetros y una temperatura de boquilla en el extremo alto del rango recomendado por el fabricante favorecen la unión entre capas, que es el punto por donde suele entrar el agua en una pieza impresa. El relleno interno se mantiene bajo, ya que la resistencia mecánica requerida es modesta y no justifica el gasto adicional de material ni de tiempo.

4.7.1. Limitaciones del encerramiento

El diseño presenta tres limitaciones que conviene reconocer. La primera se refiere al grado de protección, pues la carcasa no se somete a un ensayo normalizado y por lo tanto no puede declararse un nivel IP determinado. Las aberturas de los sensores y de la alimentación constituyen los puntos críticos, y su sellado con silicona neutra o con prensaestopas mejora el comportamiento frente al agua, tal como se reporta en encerramientos impresos para nodos agrícolas (Dong y Hansen, 2024).

La segunda limitación tiene que ver con la temperatura interior. Una caja cerrada y expuesta al sol acumula calor, y ese calor afecta tanto a la electrónica como a la batería. El montaje bajo la sombra del cafetal atenúa el problema, aunque una versión posterior se beneficiaría de una cubierta superior que genere sombra y de aberturas de ventilación protegidas contra la lluvia.

La tercera se relaciona con el sensor de aire. El DHT22 debe permanecer fuera del encerramiento para medir las condiciones reales del ambiente, y en esa posición la radiación solar directa lo lleva a sobreestimar la temperatura. La solución habitual consiste en alojarlo en una pantalla de protección meteorológica, una pieza que también se fabrica por impresión y que se plantea como trabajo complementario para la etapa de instalación definitiva (Sudarmaji et al., 2024).

4.8. Adquisición y transmisión de datos

La captura se programa para ejecutarse de forma periódica cada cinco minutos. En cada ciclo el microcontrolador solicita una lectura a los tres sensores, comprueba que los valores caigan dentro de rangos físicamente posibles y descarta los que resulten inválidos, como ocurre de vez en cuando con el DHT22, que en algunas lecturas devuelve un dato nulo. Así solo se transmite información coherente.

La recolección no ocurre durante las veinticuatro horas, sino en una jornada de doce horas diarias. Esta decisión responde a la autonomía de la batería portátil que alimenta el sistema, cuya carga cubre con holgura ese periodo de trabajo. Como consecuencia, el conjunto de datos concentra sus registros en esa franja horaria, algo que se tiene presente al momento de analizar la información y que se declara como una condición del estudio.

Una vez validadas, las lecturas se envían por Wi-Fi hacia ThingSpeak, un servicio en la nube pensado para proyectos de Internet de las Cosas (MathWorks, s.f.). La transmisión se hace mediante una petición al servidor de la plataforma, en la que cada variable viaja asociada a un campo definido de antemano en el canal. Cuando alguna lectura no supera la validación, su campo se envía vacío, de modo que el registro conserva la marca de tiempo, aunque falte ese valor puntual.

Para poder realizar este proceso se requiere crear una cuenta en el sitio web oficial <https://thingspeak.mathworks.com/> una vez creada se crea un canal, para el caso de este proyecto fue titulado “Monitoreo cafetal”. Los canales en Thingspeak pueden tener hasta un máximo de 8 espacios en los cuales se puede procesar una variable, para este proyecto se usa 4 de esos espacios que corresponden a las variables donde se va procesar la temperatura y humedad del suelo y aire, en la siguiente imagen se puede observar la interfaz.

Figura 6. Creación de canal y espacios en Thingspeak

Thingspeak™ Channels ▾ Apps ▾ Devices ▾ Support ▾

Private View Public View Channel Settings Sharing API Keys Data Import

Channel Settings

Percentage Complete 30%

Channel ID 3482492

Name

Description

Field 1 ☒

Field 2 ☒

Field 3 ☒

Field 4 ☒

Fuente: Autores

4.9. Diseño de almacenamiento de datos

El almacenamiento se delegó en la propia plataforma Thingspeak, que guarda cada envío dentro de un canal configurado con cuatro campos, uno por cada variable medida: temperatura del suelo, temperatura del aire, humedad del aire y humedad del suelo. A cada registro la plataforma le agrega de manera automática una marca de tiempo, con lo que la serie queda ordenada cronológicamente sin ningún trabajo adicional.

Thingspeak permite exportar todo el histórico del canal en un archivo de valores separados por comas, que es el formato con el que se trabaja en la fase de análisis. Ese archivo trae una columna con la fecha y hora de cada registro y una columna por cada campo, además de algunos datos propios de la plataforma que

no se usan en el estudio y que se descartan al procesar la información. Para tal fin basta con ingresar a la pestaña Data Import/export y descargar el archivo de tipo csv que genera la recolección de información

Figura 7. Data import/export en thingspeak

The screenshot shows the Thingspeak interface for a channel named 'monitoreo cafetal'. The channel ID is 3482492, the author is mwa000042244507, and the access is Private. The 'Data Import / Export' tab is selected. The 'Import' section allows uploading a CSV file, with a 'File' dropdown showing 'Seleccionar archivo' and 'Sin archiv...lecionados', and a 'Time Zone' dropdown set to '(GMT+00:00) UTC'. An 'Upload' button is present. The 'Export' section allows downloading all channel feeds in CSV format, with a 'Time Zone' dropdown set to '(GMT+00:00) UTC' and a 'Download' button. A 'Help' section on the right explains the CSV import format, showing a sample of the data structure: 'created_at,field1,field3,field4,field8,elevation' followed by a timestamp and numerical values. It also mentions other import and export options like MATLAB, REST API, and MQTT API.

Fuente: Autores

Como los campos se exportan con nombres genéricos, el primer paso al procesar el archivo es renombrarlos con etiquetas claras que identifiquen cada variable, para no confundirlas más adelante. Este conjunto de datos, organizado por fecha y hora, es la base sobre la que se integrará después la precipitación del IDEAM, aprovechando que ambas fuentes comparten la misma referencia temporal. En la siguiente figura se puede observar un ejemplo de archivo exportado.

Figura 8. ejemplo de archivo .csv exportado desde Thingspeak

A	B	C	D	E	F
created_at,entry_id,field1,field2,field3,field4,latitude,longitude,elevation,status					
2026-09-05T15:28:22+00:00,1,,,,,100.00000,,,,,					
2026-09-05T15:33:22+00:00,2,,,,,100.00000,,,,,					
2026-09-05T15:38:22+00:00,3,,,,,100.00000,,,,,					
2026-09-05T15:43:22+00:00,4,,,,,100.00000,,,,,					
2026-09-05T15:48:21+00:00,5,,30.30000,55.60000,100.00000,,,,,					
2026-09-05T15:53:21+00:00,6,,30.60000,54.40000,0.00000,,,,,					
2026-09-05T15:58:21+00:00,7,,30.60000,55.50000,0.00000,,,,,					
2026-09-05T16:03:22+00:00,8,,30.60000,56.50000,0.00000,,,,,					
2026-09-05T16:08:22+00:00,9,,30.90000,57.60000,77.00000,,,,,					

Fuente: Autores

4.10. Implementación del sistema de monitoreo ambiental

La puesta en marcha se realizó directamente en el cultivo de café. Los sensores se ubicaron de manera que cada uno quedara en contacto con el medio que debía medir: el SEN0193 y el DS18B20 enterrados en el suelo, a la profundidad de interés para el cultivo, y el DHT22 al aire libre pero resguardado de la lluvia y de la radiación solar directa, para que sus lecturas reflejaran las condiciones reales del ambiente.

El microcontrolador se alimentó con una batería portátil, lo que permitió operar sin depender de una red eléctrica en la parcela. Con esa fuente el sistema funcionó en jornadas de doce horas diarias, durante las cuales registró y transmitió datos cada cinco minutos. Al terminar cada jornada la batería se recargaba para la sesión siguiente, de modo que la recolección se mantuvo de forma constante a lo largo del periodo previsto.

Antes de dar por iniciada la recolección definitiva se calibró el sensor de humedad del suelo. Para ello se tomaron sus lecturas en dos condiciones extremas, primero en seco y luego en un medio saturado de agua, y con esos dos valores de

referencia se ajustó la conversión a porcentaje, de manera que las mediciones representaran de forma fiel el contenido de humedad del terreno.

Durante la operación, la plataforma ThingSpeak resultó útil para verificar en tiempo real que los datos llegaran correctamente, lo que facilitó detectar de inmediato cualquier corte en la conexión o falla en algún sensor. Al cerrar el periodo de recolección, el histórico completo se descargó en formato CSV para su posterior análisis (Afzal et al., 2025; Sudarmaji et al., 2024).

4.11. Instalación de sensores en el cultivo

La instalación se planeó buscando que cada sensor quedara en contacto directo con el medio que debía medir y que su ubicación representara bien las condiciones del cultivo. El SEN0193, encargado de la humedad del suelo, y el DS18B20, que mide la temperatura del suelo, se enterraron en la zona de raíces del cafeto, a una profundidad acorde con la capa de terreno que interesa para el manejo del riego. Al enterrar el sensor capacitivo se cuidó que quedara en contacto firme con el suelo, sin bolsas de aire a su alrededor, porque un mal contacto altera la lectura y le resta fiabilidad a la medición (DFRobot, s.f.).

El DHT22, que registra la temperatura y la humedad del aire, se ubicó por encima del suelo, resguardado de la lluvia y de la radiación solar directa. Esta precaución es necesaria porque el sol dando de lleno sobre el sensor eleva de forma artificial la temperatura medida y distorsiona la humedad relativa, con lo cual los valores dejarían de reflejar el ambiente real del cultivo. Se buscó también una altura y una posición que se acercaran a las condiciones a las que están expuestas las plantas.

Sobre el punto de instalación dentro del lote, se escogió una zona intermedia del cultivo y se evitaron los bordes y los sitios con microcondiciones atípicas, como charcos o áreas de sombra permanente, que no representarían el comportamiento general del cafetal. El microcontrolador y la batería se protegieron dentro de una

carcasa que los aislara de la humedad y el polvo, por ser los componentes más sensibles a la intemperie. La placa de circuito impreso, más robusta que un montaje sobre protoboard, contribuye a esa estabilidad al disminuir el riesgo de falsos contactos (Sudarmaji et al., 2024).

Figura 9. Dispositivo en el cultivo de café



Fuente: Autores

Es importante mencionar que los sensores encargados de la medición de las variables del suelo se encuentran bajo tierra por lo cual no se observan a la vista,

sin embargo, si se aprecia la alimentación vía power bank y el sensor DHT encargado de tomar las lecturas del ambiente.

4.12. Captura de variables ambientales

El sistema registra cuatro variables ambientales, cada una con un papel definido en el estudio de la humedad del suelo. La variable central es la humedad del suelo, medida por el SEN0193, pues es la que el proyecto busca analizar y, más adelante, predecir. Las otras tres funcionan como factores que ayudan a explicar su comportamiento.

La temperatura y la humedad del aire, ambas tomadas por el DHT22, se relacionan de manera directa con la velocidad a la que el suelo pierde agua. Cuando el aire está caliente y seco, la evapotranspiración aumenta y el suelo se seca más rápido; cuando está fresco y húmedo, esa pérdida se vuelve más lenta. Por eso las dos aportan información valiosa sobre la dinámica de secado del terreno (Afzal et al., 2025).

La temperatura del suelo, registrada por el DS18B20, complementa esa lectura, ya que interviene en los procesos de evaporación y en la actividad biológica del suelo. A estas cuatro variables medidas en campo se suma la precipitación, que no proviene de los sensores sino de los registros del IDEAM, dado que el circuito no incluye un pluviómetro. La lluvia es determinante en el comportamiento de la humedad del suelo, porque es la principal fuente de agua que la incrementa, de modo que traerla desde una fuente externa resulta necesario para completar el panorama ambiental (Abdelmoneim et al., 2023).

4.13. Frecuencia de muestreo y recolección de datos

Los datos ambientales se capturan cada cinco minutos. Esta frecuencia responde a un equilibrio: un intervalo corto para seguir de cerca los cambios de las

variables a lo largo del día, pero sin generar un volumen inmanejable ni lecturas tan seguidas que apenas se diferencien entre sí.

La recolección se desarrolla en jornadas de doce horas diarias, condicionadas por la autonomía de la batería portátil que alimenta el sistema, y se extiende a lo largo de tres meses. Ese periodo no es arbitrario. Tres meses abarcan una porción representativa del régimen climático de la zona, con temporadas lluviosas, rachas secas y las transiciones entre unas y otras, de modo que el sistema observa el suelo en estados muy distintos: saturado tras las lluvias, secándose poco a poco y cercano a su punto más seco. Un modelo de humedad del suelo necesita esa diversidad de condiciones para aprender la relación entre el clima y el agua del terreno, y una ventana más corta correría el riesgo de capturar un solo tipo de situación y dejarlo sin material para generalizar (Islam et al., 2025). A esto se suma que la humedad del suelo conserva memoria de lo ocurrido en los días previos, ya que depende no solo de la lluvia del momento sino de la acumulada durante la última semana y del tiempo transcurrido desde el último evento. Tres meses ofrecen una serie lo bastante extensa para que esas variables derivadas tomen un rango amplio de valores y aporten información útil al modelo.

En cuanto al volumen de información, con una lectura cada cinco minutos se obtienen alrededor de ciento cuarenta y cuatro mediciones por jornada de doce horas. A lo largo de los tres meses de recolección esto representa cerca de trece mil registros por variable, una cantidad suficiente para entrenar los modelos y, sobre todo, para reservar periodos independientes con los cuales evaluar su desempeño sin mezclar los datos de ajuste con los de prueba. Conviene señalar que, al operar doce horas al día, la serie cubre solo una pequeña de la franja nocturna.

4.14. Integración de datos meteorológicos del IDEAM

Las variables meteorológicas del proyecto provienen del IDEAM, la entidad oficial encargada de la información hidrometeorológica en Colombia. La descarga se realiza desde el portal oficial de consulta y descarga de datos del instituto, del cual se obtiene el reporte de precipitación correspondiente a los meses de mayo, junio y julio, que coinciden con el periodo de recolección en campo (IDEAM, 2026). La variable de interés es la precipitación diaria, expresada en milímetros, ya que la lluvia es el factor que más incide en el comportamiento de la humedad del suelo y el sistema de sensores no cuenta con un pluviómetro propio.

La estación se selecciona con un criterio de cercanía geográfica. Entre las disponibles en la zona se elige la estación Albania, identificada con el código 24050110, por ser la más próxima a la vereda Santa Inés, lugar donde se encuentra la finca en la que se desarrolla el estudio. Esa cercanía resulta determinante, pues mientras más cerca está la estación del cultivo, mejor representa la lluvia que realmente cae sobre el terreno monitoreado. La estación entrega registros diarios de precipitación para los tres meses considerados, de modo que aporta una serie continua a lo largo del periodo de estudio. Conviene señalar que los datos se encuentran en nivel preliminar, es decir, sin una validación definitiva por parte del instituto, condición que se tiene presente al interpretarlos.

La integración de las dos fuentes se apoya en la fecha como elemento común. El sistema IoT registra sus lecturas cada cinco minutos, en tanto que el IDEAM reporta un solo valor de lluvia por día, así que ambas series parten de resoluciones temporales distintas. Para unir las, a cada registro de los sensores se le asocia la precipitación del día al que pertenece, con lo que el valor diario de lluvia acompaña a todas las mediciones de esa jornada. Antes de ese cruce, los datos del IDEAM pasan por una depuración, ya que el archivo descargado presenta

particularidades de formato, como la codificación del texto y el separador decimal, que se corrigen para poder procesarlos sin errores.

A partir de la precipitación ya integrada se construyen además variables derivadas que suelen ser más informativas que el dato crudo, como la lluvia acumulada en los días previos y el tiempo transcurrido desde el último evento de lluvia. Estas variables recogen la memoria del suelo, pues su humedad depende de lo ocurrido en las jornadas anteriores. El producto de todo este proceso es un único conjunto de datos ambiental que reúne, sobre una misma línea de tiempo, las variables medidas por los sensores y la precipitación aportada por el IDEAM, listo para la etapa de modelado.

La base de datos obtenida del IDEAM cuentan con la siguiente estructura:

Tabla 4.

Descripción de la base de datos del IDEAM

Columna	Descripción	Ejemplo
CodigoEstacion	Identificador numérico único de la estación meteorológica	24050110
NombreEstacion	Nombre de la estación acompañado de su código	ALBANIA [24050110]
Variable	Magnitud que registra la estación	PRECIPITACION
Parametro	Tipo de medición y forma de registro (lectura manual acumulada por día)	Día pluviométrico (convencional)
Fecha	Día de la observación, con la hora fija en las cero horas por tratarse del total de la jornada	2026-05-03 00:00
Unidad	Unidad en la que se expresa el valor	mm
Valor	Cantidad de lluvia caída durante el día	11.4
NivelAprobacion	Estado de validación del dato dentro del instituto	Preliminar

Nota: Tabla elaborada por el autor a partir del análisis del archivo .csv obtenido del IDEAM

En la siguiente imagen se observa un fragmento de la base de datos en formato separado por comas que se obtuvo del IDEAM y que representa el comportamiento de la lluvia en un periodo de 3 meses (mayo, junio y julio).

Figura 10. Fragmento de base de datos del IDEAM

A	B	C	D	E	F	G	H	I
CodigoEstacion,NombreEstacion,Variable,Parametro,Fecha,Unidad,Valor,NivelAprobacion								
24050110,ALBANIA [24050110],PRECIPITACION,DÃ-a pluviomÃ©trico (convencional),2026-05-01 00:00,mm,0,Preliminar								
24050110,ALBANIA [24050110],PRECIPITACION,DÃ-a pluviomÃ©trico (convencional),2026-05-02 00:00,mm,2.1,Preliminar								
24050110,ALBANIA [24050110],PRECIPITACION,DÃ-a pluviomÃ©trico (convencional),2026-05-03 00:00,mm,11.4,Preliminar								
24050110,ALBANIA [24050110],PRECIPITACION,DÃ-a pluviomÃ©trico (convencional),2026-05-04 00:00,mm,0,Preliminar								
24050110,ALBANIA [24050110],PRECIPITACION,DÃ-a pluviomÃ©trico (convencional),2026-05-05 00:00,mm,0,Preliminar								
24050110,ALBANIA [24050110],PRECIPITACION,DÃ-a pluviomÃ©trico (convencional),2026-05-06 00:00,mm,0,Preliminar								
24050110,ALBANIA [24050110],PRECIPITACION,DÃ-a pluviomÃ©trico (convencional),2026-05-07 00:00,mm,0,Preliminar								

Fuente: IDEAM

La base de datos completa usada y con los filtros correspondientes a departamento, municipio y estación usada para la elaboración de este proyecto se encuentra disponible en los anexos de este documento

4.15. Limpieza de datos, normalización y transformación de variables

El procesamiento de los datos se realiza en Google Colaboratory, un entorno de programación basado en la nube que permite escribir y ejecutar código en lenguaje Python directamente desde el navegador, sin necesidad de instalar programas en el computador (Google, s.f.). Esta característica resulta conveniente para el proyecto, pues el trabajo se puede desarrollar desde cualquier equipo con conexión a internet y los resultados quedan guardados y disponibles en línea.

La elección de esta herramienta responde a varias ventajas prácticas. El entorno trae ya instaladas las principales librerías de análisis de datos, de modo que se puede comenzar a trabajar de inmediato sin configuraciones complejas. Se integra además con el almacenamiento en la nube, lo que facilita cargar tanto el

archivo de mediciones de los sensores como el reporte del IDEAM, y permite compartir el cuaderno de trabajo con otras personas, algo útil en un contexto académico en el que el proyecto se revisa y se documenta.

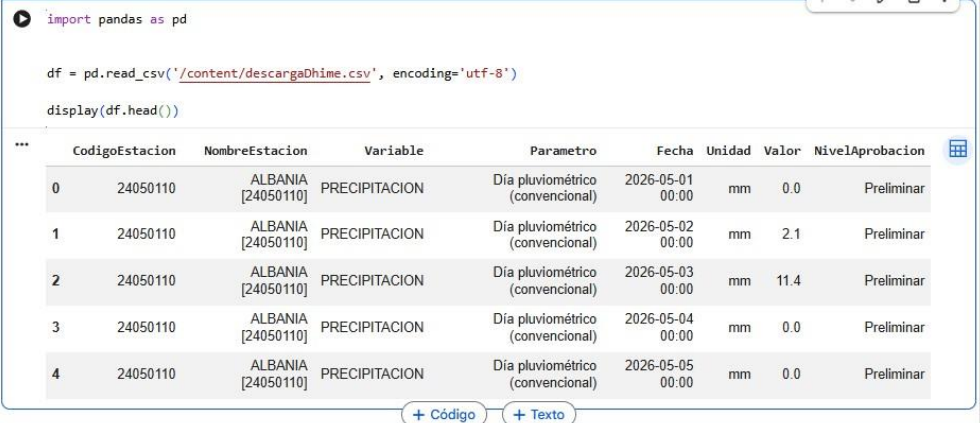
Dentro de este entorno, el análisis se apoya en Python y en un conjunto de librerías especializadas. La manipulación y limpieza de los datos se realiza con pandas, que ofrece estructuras adecuadas para trabajar con tablas y series de tiempo (McKinney, 2010), mientras que las tareas de normalización y transformación de variables se apoyan en scikit-learn, una librería ampliamente utilizada para el preprocesamiento de datos y el aprendizaje automático (Pedregosa et al., 2011). El apoyo visual, necesario para inspeccionar la información y comprobar cada paso, se obtiene con librerías de graficación como Matplotlib.

4.15.1. *Proceso ETL sobre la base de datos IDEAM*

Para empezar, es importante considerar que los datos descargados pueden estar en desorden o con tipo de datos incorrectos, por esta razón antes de comenzar a realizar cualquiera de los modelos existentes se requiere preparar la información que se puede describir como un proceso ETL, tres letras que resumen sus tres momentos: extraer, transformar y cargar. Primero se extraen los datos desde donde viven originalmente, luego se transforman, que es donde ocurre el verdadero trabajo de limpiar, corregir formatos y poner orden, y al final se cargan en un destino donde quedan listos para usarse (Vassiliadis, 2025).

A simple vista se observa que el archivo .csv que se obtuvo del IDEAM presenta inconsistencias, una de las notorias tiene que ver con los acentos, esto ocurre porque no se encuentra en codificación utf-8, así que la primera transformación es corregir ese texto dañado, en la siguiente figura se observa el fragmento de código que permite cargar el archivo y codificarlo correctamente y observar las 5 primeras filas del contenido.

Figura 11. cargue del archivo en Google Colab y codificación



```
import pandas as pd

df = pd.read_csv('/content/descargaDhime.csv', encoding='utf-8')

display(df.head())
```

	CodigoEstacion	NombreEstacion	Variable	Parametro	Fecha	Unidad	Valor	NivelAprobacion
0	24050110	ALBANIA [24050110]	PRECIPITACION	Día pluviométrico (convencional)	2026-05-01 00:00	mm	0.0	Preliminar
1	24050110	ALBANIA [24050110]	PRECIPITACION	Día pluviométrico (convencional)	2026-05-02 00:00	mm	2.1	Preliminar
2	24050110	ALBANIA [24050110]	PRECIPITACION	Día pluviométrico (convencional)	2026-05-03 00:00	mm	11.4	Preliminar
3	24050110	ALBANIA [24050110]	PRECIPITACION	Día pluviométrico (convencional)	2026-05-04 00:00	mm	0.0	Preliminar
4	24050110	ALBANIA [24050110]	PRECIPITACION	Día pluviométrico (convencional)	2026-05-05 00:00	mm	0.0	Preliminar

Fuente: Autores

El siguiente paso (como se muestra en la figura siguiente) es revisar el número de filas y columnas, el tipo de dato que pandas asignó a cada una y la cantidad de valores nulos que hay en el archivo. Esto sirve para detectar problemas de una vez, como una columna numérica que llegó como texto o registros vacíos que habrá que atender más adelante.

Figura 12. Revisión de la estructura general del archivo

```
# Revisar la estructura general del archivo
df.info()
print("\nDimensiones:", df.shape)
print("\nValores nulos por columna:")
print(df.isnull().sum())

*** <class 'pandas.core.frame.DataFrame'>
RangeIndex: 92 entries, 0 to 91
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  -
0   CodigoEstacion  92 non-null     int64
1   NombreEstacion  92 non-null     object
2   Variable        92 non-null     object
3   Parametro       92 non-null     object
4   Fecha           92 non-null     object
5   Unidad          92 non-null     object
6   Valor           92 non-null     float64
7   NivelAprobacion 92 non-null     object
dtypes: float64(1), int64(1), object(6)
memory usage: 5.9+ KB

Dimensiones: (92, 8)

Valores nulos por columna:
CodigoEstacion    0
NombreEstacion    0
Variable          0
Parametro         0
Fecha             0
Unidad            0
Valor             0
NivelAprobacion   0
dtype: int64
```

Fuente: Autores

Estos primeros resultados permiten observar que no hay nulos en las diferentes columnas, se destaca los tipos de datos, se observa que la fecha es un objeto y no es un tipo de datos tipo fecha motivo por el cual se debe corregir, aunque se observa que la columna “valor” es numérica se ejecuta el código de la figura para realizar la modificación y comprobación descrita.

Figura 13. Código usado para convertir la fecha y asegurar el formato de la columna valor

```
[3]
✓ 0 s # Convertir la fecha de texto a un formato de fecha real
df['Fecha'] = pd.to_datetime(df['Fecha'])

# Asegurar que el valor de precipitacion sea numerico
df['Valor'] = pd.to_numeric(df['Valor'], errors='coerce')
```

Fuente: Autores

El archivo trae varias columnas que repiten siempre la misma información, como el nombre de la estación, la variable o la unidad, y que no aportan nada al análisis. Aquí se filtra primero por el código de la estación Albania, para asegurar que solo queden sus registros, aunque la descarga incluya otras, y luego se conservan únicamente la fecha y el valor de precipitación, que son las dos columnas que de verdad importan. El uso de copy evita advertencias posteriores al seguir modificando la tabla.

Figura 14. filtros y columnas a usar

```
[4]
✓ 0 s
# Conservar solo la estacion Albania, por si el archivo trae otras
df = df[df['CodigoEstacion'].astype(str).str.strip() == '24050110']

# Quedarse con las dos columnas de interes: fecha y valor de lluvia
df = df[['Fecha', 'Valor']].copy()
```

Fuente: Autores

Para que la tabla sea fácil de entender más adelante, se cambian los nombres genéricos por otros que dicen con claridad qué contiene cada columna, en especial la de precipitación, a la que se le añade la unidad en milímetros. Después los registros se ordenan por fecha, del más antiguo al más reciente, y se reinicia el índice para que la numeración de las filas quede consecutiva. Un orden cronológico limpio es indispensable para los pasos que vienen. Para lograrlo se usa el código que se observa en la siguiente figura.

Figura 15. cambio de nombres y orden

```
[5]
✓ 0 s
# Renombrar las columnas con nombres claros
df = df.rename(columns={'Fecha': 'fecha', 'Valor': 'precipitacion_mm'})

# Ordenar los registros del mas antiguo al mas reciente
df = df.sort_values('fecha').reset_index(drop=True)
```

Fuente: Autores

Uno de los pasos más importantes es determinar si realmente están todos los datos, Para averiguarlo se construye el rango completo de fechas entre el primer y el último día registrado y se compara con las fechas que sí aparecen en el archivo. La diferencia entre ambos revela los días sin dato. La comprobación es importante porque, en una estación convencional, un día ausente significa que no se hizo la medición y conviene saberlo antes de decidir qué hacer con esos faltantes, el código de la siguiente figura realiza dicha exploración.

Figura 16. Revisión faltantes

```
# Construir el rango completo de días del periodo
rango = pd.date_range(df['fecha'].min(), df['fecha'].max(), freq='D')

# Comparar los días esperados con los que realmente estan
faltantes = rango.difference(df['fecha'])
print("Dias esperados :", len(rango))
print("Dias presentes :", df['fecha'].nunique())
print("Dias faltantes :", len(faltantes))

*** Dias esperados : 92
    Dias presentes : 92
    Dias faltantes : 0
```

Fuente: Autores

Como se observa, no hay valores faltantes por lo tanto el último fragmento de código cierra el proceso con una revisión y el guardado. Se imprime el rango de fechas y la lluvia total acumulada, cifras que permiten confirmar de un vistazo que el resultado tiene sentido, y se muestran las primeras filas de la tabla ya depurada. Comprobado esto, la serie diaria de precipitación se guarda en un nuevo archivo de valores separados por comas, que será el insumo del IDEAM en la etapa de integración con los datos de los sensores.

Figura 17. Revisión final y guardado del archivo limpio

```
7] 0s ▶ # Revision final del resultado
print("Rango:", df['fecha'].min().date(), "a", df['fecha'].max().date())
print("Lluvia total:", round(df['precipitacion_mm'].sum(), 1), "mm")
display(df.head())

# Guardar la serie diaria ya depurada
df.to_csv('ideam_limpio.csv', index=False)
```

✓ ... Rango: 2026-05-01 a 2026-07-31
Lluvia total: 415.5 mm

	fecha	precipitacion_mm
0	2026-05-01	0.0
1	2026-05-02	2.1
2	2026-05-03	11.4
3	2026-05-04	0.0
4	2026-05-05	0.0

Fuente: Autores

4.15.2. **Proceso ETL sobre la base de datos obtenida de la lectura de los sensores**

Para realizar este análisis se carga el dataset al entorno de Google Colab y se hace una inspección general aunque a diferencia del IDEAM, este no necesita corrección de codificación porque proviene de un formato propio y limpio. Con esto la extracción queda resuelta, en la siguiente figura se evidencia cuántas filas y columnas tiene, el tipo de cada variable y cuántos valores nulos hay.

Figura 18. exploración inicial

```
import pandas as pd

# Cargar el nuevo archivo CSV
df_sensores = pd.read_csv('/content/lecturas_sensores_may_jul_2026.csv', encoding='utf-8')

print("Primeras 5 filas del nuevo archivo:")
display(df_sensores.head())
```

	timestamp	temp_suelo_C	temp_aire_C	hum_aire_pct	hum_suelo_pct
0	2026-05-01 06:00:00	19.76	20.1	87.7	33.9
1	2026-05-01 06:05:00	19.56	19.8	87.0	33.9
2	2026-05-01 06:10:00	19.59	20.8	83.9	33.9
3	2026-05-01 06:15:00	19.52	20.9	82.9	33.9
4	2026-05-01 06:20:00	19.71	20.6	85.0	33.8

```
# Estructura general, tipos de dato y valores nulos
df_sensores.info()
print("\nDimensiones:", df_sensores.shape)
print("\nValores nulos por columna:")
print(df_sensores.isnull().sum())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13248 entries, 0 to 13247
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   timestamp       13248 non-null  object
1   temp_suelo_C    13248 non-null  float64
2   temp_aire_C     13289 non-null  float64
3   hum_aire_pct    13289 non-null  float64
4   hum_suelo_pct   13248 non-null  float64
dtypes: float64(4), object(1)
memory usage: 517.6+ KB

Dimensiones: (13248, 5)

Valores nulos por columna:
timestamp      0
temp_suelo_C   0
temp_aire_C    39
hum_aire_pct   39
hum_suelo_pct  0
dtype: int64
```

Fuente: Autores

Se observa que el tiempo es de tipo de dato objeto por lo que se debe modificar para que sea de tipo fecha, se usa el código que se observa en la siguiente figura, es de destacar que luego los registros se ordenan del más antiguo al más reciente, condición indispensable para una serie de tiempo y para la posterior unión con el IDEAM.

Figura 19. conversión tipo de variable tiempo

```
# Convertir la marca de tiempo de texto a formato de fecha y hora real
df_sensores['timestamp'] = pd.to_datetime(df_sensores['timestamp'])

# Ordenar cronologicamente y reiniciar el indice
df_sensores = df_sensores.sort_values('timestamp').reset_index(drop=True)
df_sensores.info()
```

```
>>> <class 'pandas.core.frame.DataFrame'>
RangeIndex: 13248 entries, 0 to 13247
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   timestamp      13248 non-null  datetime64[ns]
1   temp_suelo_C    13248 non-null  float64
2   temp_aire_C     13209 non-null  float64
3   hum_aire_pct    13209 non-null  float64
4   hum_suelo_pct   13248 non-null  float64
dtypes: datetime64[ns](1), float64(4)
memory usage: 517.6 KB
```

Fuente: Autores

Como medida de seguridad, cualquier lectura fuera de lo posible, como una humedad mayor a cien o una temperatura absurda, se marca como valor faltante para tratarla junto con los demás huecos. En este archivo no hay casos así, pero el paso protege el análisis frente a lecturas erráticas que podrían aparecer en una campaña real.

Figura 20. Chequeo de rangos físicos

```
import numpy as np

# Marcar como nulos los valores fuera de rango fisicamente posible
df_sensores.loc[~df_sensores['hum_aire_pct'].between(0, 100), 'hum_aire_pct'] = np.nan
df_sensores.loc[~df_sensores['hum_suelo_pct'].between(0, 100), 'hum_suelo_pct'] = np.nan
df_sensores.loc[~df_sensores['temp_aire_C'].between(-10, 60), 'temp_aire_C'] = np.nan
df_sensores.loc[~df_sensores['temp_suelo_C'].between(-10, 60), 'temp_suelo_C'] = np.nan
```

Fuente: Autores

Una vez corregida la información y con el dataset limpio se descarga porque es con ese dataset que se realizarán los pasos posteriores a este proyecto.

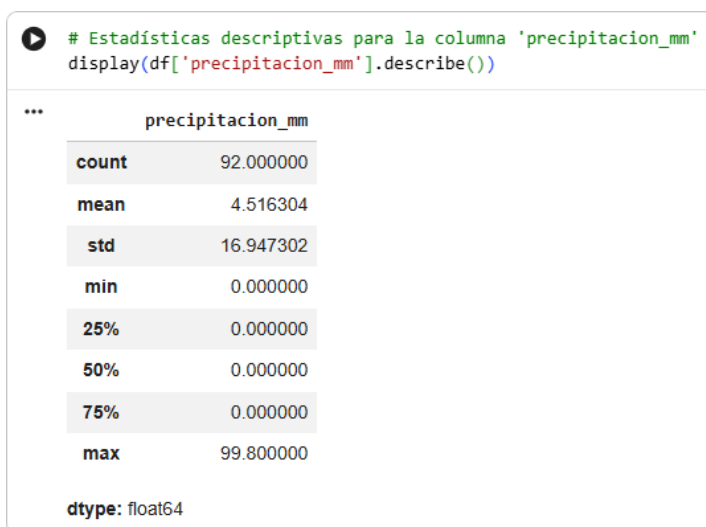
Nota. El código realizado en Google Colab se encuentran disponibles como anexo

4.16. Análisis exploratorio de datos obtenidos

4.16.1. Análisis estadístico dataset del IDEAM

Antes de modelar conviene mirar la variable con calma, y las estadísticas descriptivas ofrecen justamente ese primer resumen numérico. Este cálculo condensa en pocas cifras el comportamiento de la precipitación diaria a lo largo del periodo de estudio, un paso exploratorio que precede a cualquier análisis más elaborado.

Figura 21. Estadísticas descriptivas de la precipitación



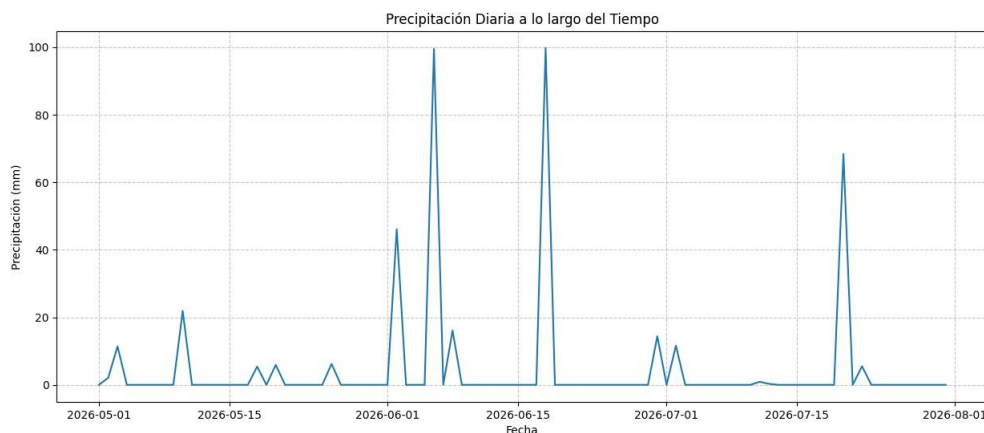
Fuente: Autores

El resumen abarca los 92 días registrados y revela una variable muy poco uniforme. La precipitación media es de apenas 4,5 milímetros diarios, pero la desviación estándar asciende a 16,9, casi cuatro veces la media, lo que anticipa

una dispersión enorme. El dato más revelador está en los cuartiles: tanto la mediana como el percentil 75 son cero, de modo que al menos tres de cada cuatro días no registran lluvia, mientras que el máximo llega a 99,8 milímetros. Esa distancia entre una media de 4,5 y una mediana de cero describe una distribución fuertemente sesgada hacia la derecha, en la que unos pocos días intensos elevan el promedio aunque la mayoría de las jornadas no aporte nada. Este perfil es habitual en los registros de precipitación (Wilks, 2019).

Para continuar con el análisis interesa ver cómo se distribuye la lluvia en el tiempo. La gráfica de línea presenta la precipitación diaria a lo largo de los tres meses y permite reconocer patrones que un resumen estadístico por sí solo no muestra.

Figura 22. Visualización de la precipitación a lo largo del tiempo



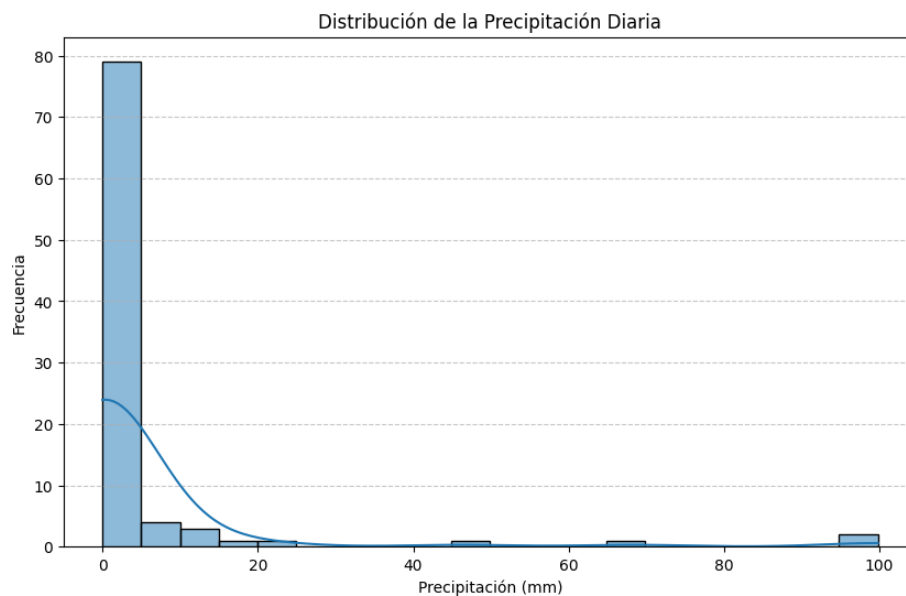
Fuente: Autores

La serie confirma de forma visual lo que sugieren las estadísticas. Predominan los tramos planos en cero, interrumpidos por picos aislados que concentran casi toda el agua del periodo. Los eventos más fuertes se ubican en junio, con dos días que rozan los 100 milímetros, y aparece otro repunte importante en julio cercano a los 68 milímetros, además de un evento menor a comienzos de

mayo. En conjunto, el gráfico muestra que la lluvia no cae de manera repartida, sino en episodios cortos e intensos separados por periodos secos, y que junio es el mes más lluvioso de la ventana estudiada. Esa concentración temporal es determinante para la humedad del suelo, pues cada pico es lo que recarga el terreno.

También es conveniente revisar histograma acompañado de la curva de densidad muestra con qué frecuencia ocurre cada magnitud de lluvia, es decir, cuántos días caen en cada rango de milímetros. Sirve para reconocer la forma de la distribución de la variable.

Figura 23. Distribución de los valores de precipitación



Fuente: Autores

La gran mayoría de los valores se agrupa cerca de cero, donde la primera barra reúne alrededor de 79 días, y la frecuencia cae abruptamente a medida que aumenta la cantidad de lluvia, dejando una cola larga y delgada que se extiende hasta valores próximos a los 100 milímetros. Esta forma asimétrica confirma de

manera gráfica lo que ya insinuaban las estadísticas descriptivas: los días secos dominan el conjunto y los días de lluvia intensa son escasos. Una asimetría tan marcada es característica de la precipitación y suele exigir un tratamiento particular durante el modelado, como el uso de transformaciones o de variables derivadas (Wilks, 2019).

Para acabar con este análisis se revisa el cálculo que cuantifica de forma directa cuántos días del periodo no registran lluvia, convirtiendo en un número concreto la impresión que dejan las gráficas anteriores. Esta información se puede evidenciar en la siguiente figura.

Figura 24. Días con cero precipitación

```
dias_sin_lluvia = df[df['precipitacion_mm'] == 0].shape[0]
total_dias = df.shape[0]
porcentaje_sin_lluvia = (dias_sin_lluvia / total_dias) * 100

print(f"Número de días con cero precipitación: {dias_sin_lluvia} de {total_dias} ({porcentaje_sin_lluvia:.2f}%)")
```

Número de días con cero precipitación: 76 de 92 (82.61%)

Fuente: Autores

De los 92 días analizados, 76 no presentan precipitación alguna, lo que equivale al 82,6 por ciento del periodo. Dicho de otro modo, más de cuatro de cada cinco días son secos y toda la lluvia del trimestre se concentra en menos de una quinta parte de las jornadas. Este hallazgo tiene una implicación directa para el modelo, ya que la señal útil proviene de un pequeño subconjunto de días. Por esa razón cobran valor las variables derivadas, como la lluvia acumulada de jornadas anteriores o el tiempo transcurrido desde el último evento, pues el valor diario por sí solo es cero la mayor parte del tiempo y aporta poca información.

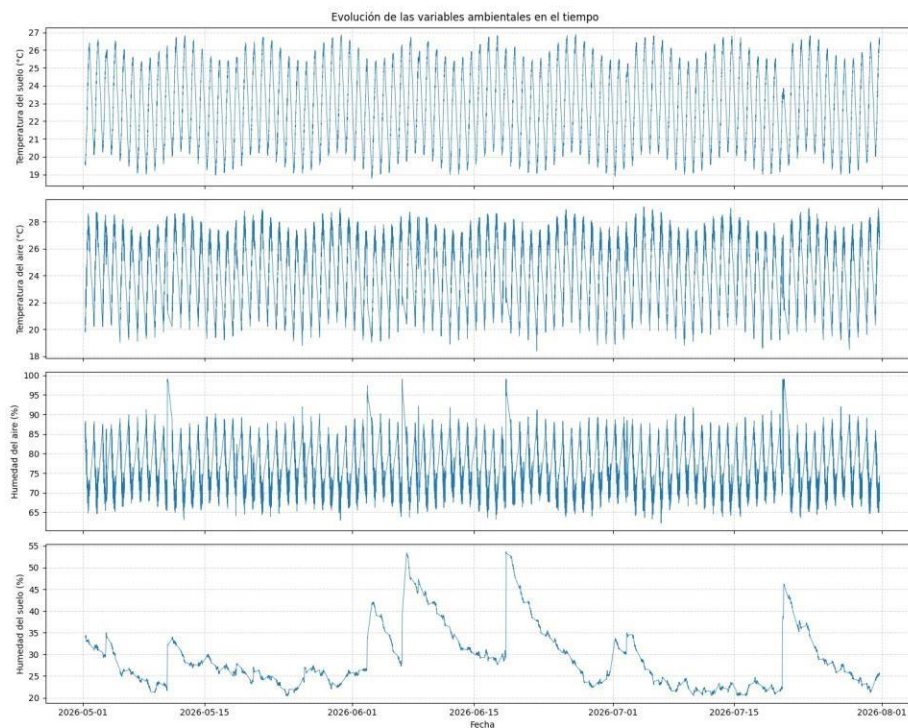
4.16.2. *Análisis estadístico dataset de la lectura de datos de los sensores*

Una vez el sistema de monitoreo opera de manera continua en el cafetal, el trabajo se desplaza del hardware hacia los datos. El conjunto reunido abarca poco más de tres meses de mediciones, entre el primero de mayo y el primero de agosto de 2026, y contiene las cuatro variables que registran los sensores. Antes de construir cualquier modelo predictivo conviene entender qué contiene ese archivo, cómo se comportan las variables y qué relaciones existen entre ellas, porque de ese diagnóstico dependen buena parte de las decisiones posteriores sobre la selección de predictores y la resolución temporal del modelo.

El análisis se desarrolla en Python, con pandas para la manipulación de la tabla y con matplotlib y seaborn para las representaciones gráficas. La secuencia recorre cinco pasos: el resumen numérico de las variables, la observación de su evolución en el tiempo, el estudio de sus distribuciones, la medición de las relaciones entre ellas y la reconstrucción del ciclo diario promedio.

Para empezar las cuatro variables se dibujan una debajo de otra y comparten el eje horizontal, de modo que cualquier evento se puede rastrear simultáneamente en los cuatro paneles.

Figura 25. Evolución de las variables ambientales durante el período de medición



Fuente: Autores

La figura revela dos comportamientos claramente distintos. Los tres primeros paneles muestran una oscilación regular que se repite día tras día sin interrupción, con temperaturas del suelo entre 19 y 27 grados, temperaturas del aire entre 18 y 29 grados y humedades relativas que se mueven en su mayoría entre 63 y 90 por ciento. Esa regularidad indica que el sistema de adquisición opera de manera estable durante los tres meses, sin interrupciones prolongadas ni derivas evidentes en los sensores.

El cuarto panel se comporta de otra manera. La humedad del suelo dibuja una serie de dientes de sierra en la que los ascensos son bruscos y los descensos, lentos y prolongados. Cada salto corresponde a un evento de lluvia que infiltra agua en el suelo, y la pendiente suave que sigue refleja la pérdida gradual de esa agua por evaporación, por drenaje y por absorción de las raíces. Durante el período se

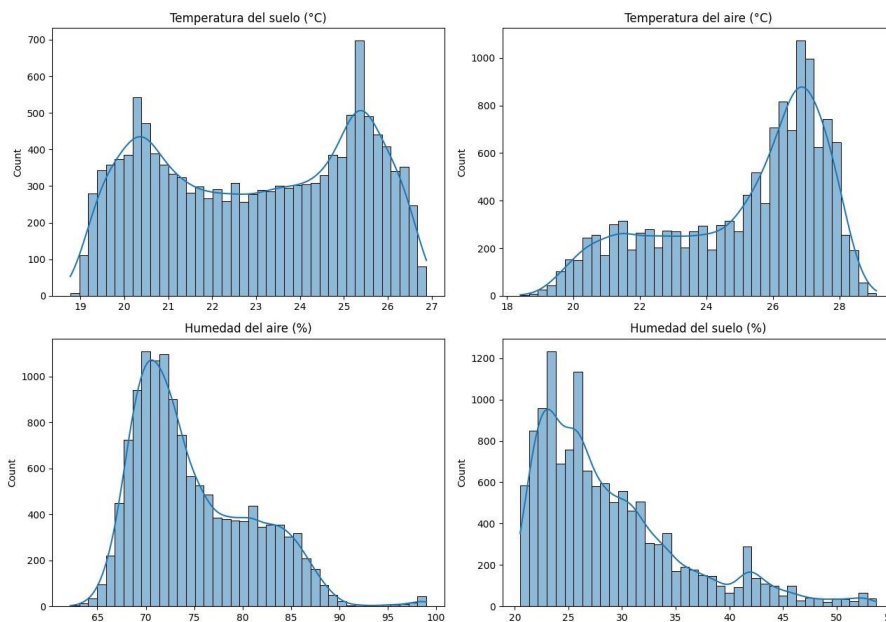
distinguen al menos seis episodios de este tipo, con los más marcados a comienzos de junio, a mediados del mismo mes y a mediados de julio, cuando la humedad pasa de valores cercanos a 22 por ciento hasta superar el 50 por ciento en cuestión de horas.

Esta diferencia de comportamiento es el hallazgo más importante de la exploración temporal. Mientras la temperatura y la humedad del aire responden al ciclo solar y por lo tanto se repiten con periodicidad diaria, la humedad del suelo responde a un forzamiento externo y esporádico, la lluvia, y conserva memoria de lo ocurrido en los días anteriores.

4.16.2.1 Distribución de cada variable

El siguiente paso examina cómo se reparten los valores de cada variable, con independencia del momento en que se registran. Cuatro histogramas con curva de densidad superpuesta permiten identificar la simetría, el sesgo y la presencia de más de una moda.

Figura 26. Distribución de las cuatro variables ambientales



Fuente: Autores

Las dos temperaturas presentan distribuciones de dos modas. En el suelo, un grupo de lecturas se concentra alrededor de 20,3 grados y otro alrededor de 25,4 grados, con una zona intermedia relativamente plana. Esa forma no indica anomalía alguna; corresponde a una variable que pasa buena parte de la jornada cerca de sus dos extremos y atraviesa con rapidez los valores intermedios. La temperatura del aire muestra el mismo patrón, aunque con la moda cálida mucho más pronunciada.

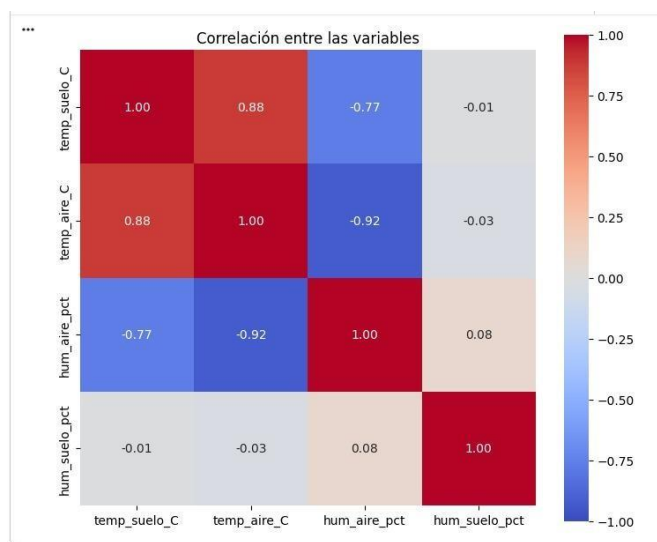
Las dos humedades, en cambio, se sesgan hacia la derecha. La del aire concentra sus lecturas cerca de 70 por ciento y presenta una cola que se extiende hasta valores próximos a la saturación, cola que corresponde a los episodios de lluvia y a las madrugadas húmedas. La del suelo muestra el sesgo más pronunciado de las cuatro, con la mayor concentración de datos alrededor de 23 por ciento y un segundo grupo, mucho menor, cerca de 42 por ciento. Ese reparto desigual confirma lo que ya sugería la serie temporal: el suelo permanece seco la mayor parte del

tiempo y solo alcanza valores altos durante intervalos breves después de las lluvias. Para el modelo predictivo, esta característica implica que los estados húmedos están poco representados y que conviene cuidar la partición de los datos para que esos casos no queden concentrados en un solo conjunto.

4.16.2.2 Correlación entre las variables

A continuación, se cuantifica las relaciones lineales entre las variables mediante la matriz de correlación de Pearson, representada como un mapa de calor.

Figura 27. Matriz de correlación entre las variables ambientales



Fuente: Autores

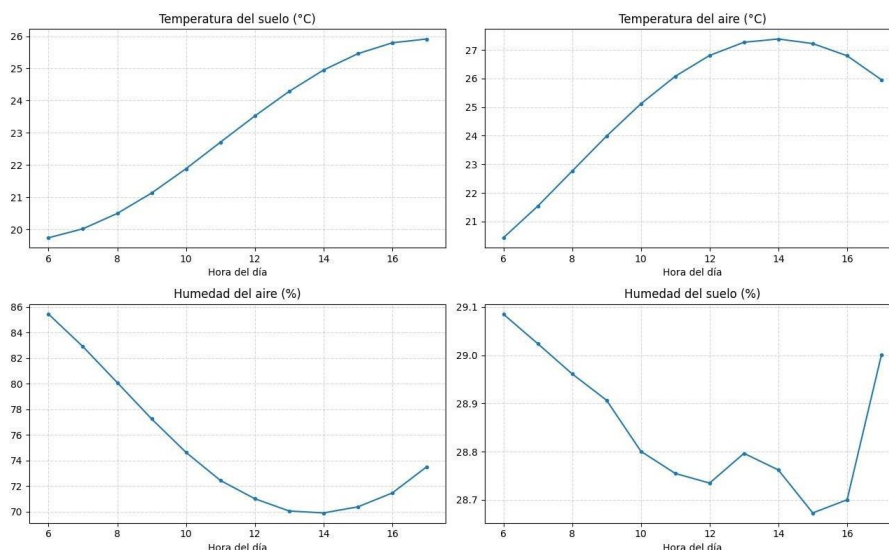
La matriz de correlación revela dos comportamientos muy distintos entre las variables. Por un lado, las que describen el estado del aire y el suelo se relacionan con fuerza entre sí: la temperatura del aire y la del suelo se mueven casi de la mano, con una correlación de 0,88, y ambas se oponen a la humedad del aire, que baja cuando la temperatura sube, como lo reflejan los valores de menos 0,92 y menos 0,77. Hasta ahí, nada sorprendente, pues responden al mismo ciclo diario de

calentamiento y enfriamiento. Lo llamativo aparece en la última fila y la última columna, donde la humedad del suelo muestra correlaciones prácticamente nulas con las demás variables, de apenas menos 0,01, menos 0,03 y 0,08. Lejos de ser un problema, este resultado confirma algo esperado y valioso: la humedad del suelo no depende de las condiciones instantáneas del aire, sino de la lluvia recibida y de su secado a lo largo de los días. Esa desconexión justifica de manera directa la decisión de incorporar la precipitación del IDEAM y de construir variables derivadas, como la lluvia acumulada, ya que sin ellas el modelo carecería de información suficiente para estimar el agua presente en el terreno.

4.16.2.3 Perfil diario promedio

El último paso reconstruye el ciclo diario. Cada registro se agrupa según la hora a la que fue tomado y se promedia, lo que descarta la variabilidad entre días y deja a la vista el patrón que se repite en la jornada.

Figura 28. Perfil diario promedio de las variables ambientales



Fuente: Autores

El perfil muestra un desfase que merece atención. La temperatura del aire alcanza su máximo entre las trece y las catorce horas, con un valor promedio cercano a 27,3 grados, y comienza a descender desde ese momento. La del suelo, en cambio, continúa subiendo durante toda la tarde y todavía no alcanza su máximo a las diecisiete horas, cuando registra 25,9 grados. Ese retraso es la expresión de la inercia térmica del suelo, que absorbe y libera calor con mayor lentitud que el aire. La consecuencia práctica es directa: si el modelo utiliza la temperatura del aire como predictor, conviene incluirla también con rezago, ya que el efecto del calentamiento atmosférico sobre el suelo se manifiesta horas después.

La humedad del aire se comporta como espejo de la temperatura y alcanza su mínimo, cercano a 70 por ciento, en la misma franja de la tarde en la que el aire está más caliente. El cuarto panel, sin embargo, es el que aporta la información más útil. La humedad del suelo varía a lo largo de la jornada entre 28,67 y 29,08 por ciento, es decir, apenas cuatro décimas de punto porcentual. Frente a un rango total de más de treinta puntos en la serie completa, esa variación intradiaria resulta despreciable.

De aquí se desprende una conclusión con efectos prácticos sobre el diseño del modelo. La humedad del suelo no tiene un ciclo diario relevante, sino una dinámica que se desarrolla en escala de días. Un intervalo de muestreo de cinco minutos entrega, por lo tanto, una resolución muy superior a la que el fenómeno exige, y agregar los datos a paso horario o diario reduce el volumen del conjunto y el ruido de medición sin sacrificar información útil. Esa agregación tiene además una ventaja adicional, pues facilita el emparejamiento con la serie de precipitación del IDEAM, que se publica con una resolución temporal más gruesa que la del sistema de monitoreo.

4.17. Selección de variables relevantes

La selección de variables parte de una observación clave que arroja el análisis exploratorio: la humedad del suelo apenas se relaciona con las condiciones instantáneas del aire y del suelo, de modo que esas mediciones, por sí solas, aportan poca capacidad de predicción. Esto ocurre porque el agua del terreno no responde al clima del momento, sino a la lluvia recibida y a su secado a lo largo de los días (Islam et al., 2025). Por esa razón, más que descartar variables, el trabajo se concentra en construir otras nuevas que capturen esa memoria del suelo.

A partir de la serie diaria de precipitación se generan tres variables derivadas: la lluvia acumulada de los últimos tres días, la de los últimos siete y el número de días transcurridos desde el último evento de lluvia. Estas variables resumen el historial reciente de agua y suelen ser más informativas que el dato puntual de un día. El siguiente fragmento las calcula y las une a las lecturas de los sensores mediante la fecha común.

Figura 29. creación de variables derivadas

```
import pandas as pd

# Cargar las dos fuentes ya depuradas
sensores = pd.read_csv('/content/sensores_limpio.csv', parse_dates=['timestamp'])
ideam = pd.read_csv('/content/ideam_limpio.csv', parse_dates=['fecha'])

# Variables derivadas a partir de la serie diaria de lluvia
ideam = ideam.sort_values('fecha').reset_index(drop=True)
ideam['lluvia_3d'] = ideam['precipitacion_mm'].rolling(3, min_periods=1).sum()
ideam['lluvia_7d'] = ideam['precipitacion_mm'].rolling(7, min_periods=1).sum()

# Días transcurridos desde el ultimo evento de lluvia
dias = 0
acumulado = []
for valor in ideam['precipitacion_mm']:
    dias = 0 if valor > 0 else dias + 1
    acumulado.append(dias)
ideam['dias_sin_lluvia'] = acumulado

# Unir la informacion diaria a cada lectura por la fecha comun
sensores['fecha'] = sensores['timestamp'].dt.normalize()
df = sensores.merge(ideam, on='fecha', how='left')
```

Fuente: Autores

El cálculo se hace sobre la serie diaria, que es donde la precipitación tiene sentido, y luego cada valor se asocia a todas las lecturas de cinco minutos de su día. Así el conjunto queda enriquecido con el contexto de lluvia que la humedad del suelo realmente necesita.

Con el conjunto ya enriquecido, se estima la influencia de cada variable mediante un modelo de Random Forest, que además de predecir permite medir cuánto aporta cada entrada al resultado. Esta técnica de importancia de variables es habitual en el análisis de datos ambientales, pues ofrece un criterio objetivo para quedarse con las más relevantes (Afzal et al., 2025). El siguiente fragmento de código ilustra el modelo auxiliar descrito.

Figura 30. nuevas variables a considerar

```
from sklearn.ensemble import RandomForestRegressor

# Variables candidatas y variable objetivo
entradas = ['temp_suelo_C', 'temp_aire_C', 'hum_aire_pct',
            'precipitacion_mm', 'lluvia_3d', 'lluvia_7d', 'dias_sin_lluvia']
objetivo = 'hum_suelo_pct'

# Modelo auxiliar para medir la influencia de cada variable
rf = RandomForestRegressor(n_estimators=200, random_state=42)
rf.fit(df[entradas], df[objetivo])

importancias = pd.Series(rf.feature_importances_, index=entradas)
print(importancias.sort_values(ascending=False))
```

lluvia_7d	0.597234
lluvia_3d	0.196466
precipitacion_mm	0.083792
dias_sin_lluvia	0.056948
hum_aire_pct	0.032695
temp_suelo_C	0.023751
temp_aire_C	0.009114
dtype:	float64

Fuente: Autores

El resultado confirma lo esperado, ya que las variables de lluvia acumulada encabezan la lista de importancia mientras que las mediciones instantáneas del aire quedan en los últimos lugares. Con base en ese orden se eligen las entradas que

alimentan el modelo, privilegiando las que describen el historial de lluvia y conservando alguna variable del suelo por su aporte complementario.

4.18. Desarrollo del modelo predictivo

El desarrollo del modelo comienza por separar las variables de entrada, ya seleccionadas, de la variable que se desea estimar, que es la humedad del suelo. Enseguida el conjunto se divide en una parte para entrenar y otra para evaluar, pero esta división no se hace al azar. Como los datos forman una serie de tiempo, dos lecturas seguidas se parecen mucho entre sí, y repartirlas aleatoriamente permitiría que el modelo viera durante el entrenamiento información casi idéntica a la de la prueba, lo que inflaría de manera engañosa su desempeño. Para evitar esa filtración, la partición es temporal: el modelo se entrena con el periodo inicial y se evalúa con el periodo final, que no ha visto.

Figura 31. Gestión de variables para modelo predictivo

```
# Variables mas influyentes segun el analisis anterior
entradas = ['lluvia_7d', 'lluvia_3d', 'dias_sin_lluvia', 'temp_suelo_c']
X = df[entradas]
y = df['hum_suelo_pct']

# Particion temporal: se entrena con el pasado y se evalua con el futuro
corte = int(len(df) * 0.8)
X_train, X_test = X.iloc[:corte], X.iloc[corte:]
y_train, y_test = y.iloc[:corte], y.iloc[corte:]
```

Fuente: Autores

Antes de entrenar se normalizan las variables, es decir, se llevan a una escala comparable para que ninguna pese más que otra por el solo hecho de manejar números más grandes. Un detalle importante es que el escalador se ajusta únicamente con los datos de entrenamiento y luego se aplica a los de prueba, de

modo que la información del periodo de evaluación no interviene en la preparación y la prueba conserva su validez.

Figura 32. Normalización de las variables

```
[4]
✓ 0 s  from sklearn.preprocessing import StandardScaler

# El escalador se ajusta solo con los datos de entrenamiento
escalador = StandardScaler()
X_train = escalador.fit_transform(X_train)
X_test = escalador.transform(X_test)
```

Fuente: Autores

Por último, se entrenan los modelos y se comparan sobre el mismo periodo de prueba. Cada uno aprende con los datos de entrenamiento, genera sus predicciones sobre el periodo de evaluación y se mide con tres métricas: el error cuadrático medio, el error absoluto medio y el coeficiente de determinación. Las dos primeras expresan, en las mismas unidades de la humedad, qué tan lejos quedan las estimaciones de los valores reales, mientras que la tercera indica qué parte del comportamiento de la humedad logra explicar el modelo.

Figura 33. modelos entrenados

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

modelos = {
    'Regresion lineal': LinearRegression(),
    'Random Forest': RandomForestRegressor(n_estimators=200, random_state=42),
}

for nombre, modelo in modelos.items():
    modelo.fit(X_train, y_train)
    pred = modelo.predict(X_test)
    rmse = np.sqrt(mean_squared_error(y_test, pred))
    mae = mean_absolute_error(y_test, pred)
    r2 = r2_score(y_test, pred)
    print(f"{nombre} -> RMSE: {rmse:.2f} | MAE: {mae:.2f} | R2: {r2:.3f}")

... Regresion lineal -> RMSE: 4.25 | MAE: 3.31 | R2: 0.538
    Random Forest -> RMSE: 6.04 | MAE: 4.77 | R2: 0.066
```

Fuente: Autores

La comparación de estas métricas es la que permite decidir, con criterio numérico y no por intuición, cuál de los modelos ofrece la mejor estimación de la humedad del suelo. Esa comparación corresponde a la evaluación de desempeño planteada en los objetivos del trabajo.

Los resultados dejan una lectura clara y, a primera vista, contraintuitiva: el modelo más sencillo se impone al más sofisticado. La regresión lineal alcanza un coeficiente de determinación de 0,538, lo que significa que explica cerca del 54 por ciento del comportamiento de la humedad del suelo, con un error cuadrático medio de 4,25 y un error absoluto medio de 3,31. El Random Forest, en cambio, se queda muy por detrás, con un coeficiente de apenas 0,066 y errores mayores en ambas métricas. En otras palabras, el modelo que se esperaba más potente termina prediciendo poco mejor que un valor promedio.

La razón de esta diferencia no está en un fallo del código, sino en la forma en que cada modelo enfrenta datos nuevos. Como la evaluación se hace con una partición temporal, el periodo de prueba corresponde a las últimas semanas, que pueden presentar condiciones de lluvia acumulada distintas a las vistas durante el entrenamiento. Ahí aparece una limitación conocida de los métodos basados en

árboles: el Random Forest no extrapola, es decir, no sabe estimar por fuera del rango de valores con el que aprendió, y ante una situación inédita tiende a repetir lo más parecido que conoce. La regresión lineal, por su naturaleza, sí proyecta la tendencia más allá de ese rango, y por eso se comporta mejor cuando el futuro se aparta de lo observado.

Este contraste deja una enseñanza más valiosa que el número en sí. Confirma que un modelo complejo no es necesariamente superior y que su desempeño depende del problema y de la manera de evaluarlo. También refuerza la decisión de haber usado una partición temporal en lugar de una aleatoria, pues una división al azar probablemente habría mostrado al Random Forest con un rendimiento alto pero engañoso, al permitirle ver durante el entrenamiento datos casi idénticos a los de la prueba. La partición temporal, más exigente, revela el comportamiento real de cada modelo frente a datos que de verdad no ha visto.

Con todo, ninguno de los dos resultados debe tomarse como definitivo. El coeficiente de la regresión lineal, cercano al 54%, indica que existe una relación aprovechable entre las variables de lluvia y la humedad del suelo, pero que aún queda margen de mejora. Ese margen abre el camino a los pasos siguientes, como incorporar nuevas variables derivadas, ajustar los parámetros de los modelos o probar algoritmos adicionales, con el fin de acercar la predicción a las necesidades reales del manejo del cultivo.

4.19. Selección de algoritmos de aprendizaje automático

La elección de los algoritmos responde a la naturaleza del problema y de los datos más que a una preferencia previa. La información se organiza en una tabla de variables numéricas, la relación entre la lluvia y la humedad del suelo no es lineal, y el volumen de datos es moderado. Ante ese panorama, el trabajo opta por comparar dos enfoques de complejidad creciente, en lugar de apostar desde el

principio por un único modelo, de modo que la decisión final se apoye en evidencia y no en suposiciones.

El primero es la regresión lineal, que cumple el papel de punto de referencia. Se trata de un modelo sencillo, rápido de entrenar y fácil de interpretar, cualidades que lo hacen útil como base de comparación. Aunque asume relaciones lineales entre las variables, su valor está en servir de umbral: un modelo más elaborado solo se justifica si logra superarlo con claridad. Esa lógica de contrastar siempre contra una referencia simple es una práctica recomendada en los estudios de predicción con aprendizaje automático (Islam et al., 2025).

El segundo es Random Forest, un método que combina múltiples árboles de decisión para producir una estimación conjunta. Se selecciona porque maneja bien las relaciones no lineales, resiste la presencia de valores atípicos, no exige un ajuste exhaustivo de parámetros para rendir de forma razonable y, además, entrega una medida de la importancia de cada variable, que resulta útil tanto para interpretar el modelo como para la etapa de selección de variables. Estas cualidades explican que sea uno de los algoritmos más empleados para la predicción de variables ambientales a partir de datos de sensores (Afzal et al., 2025; Soussi et al., 2024).

Los criterios que guían la elección pueden resumirse en cuatro. El primero es la compatibilidad con datos tabulares, que es la forma en que queda organizado el conjunto. El segundo es la capacidad de representar relaciones no lineales, necesaria porque la humedad del suelo no responde de manera proporcional a la lluvia. El tercero es la facilidad de interpretación, que permite entender qué variables pesan en la predicción y no solo obtener un resultado. Y el cuarto es un costo computacional razonable, acorde con el entorno de trabajo en la nube. Bajo estos criterios, ambos algoritmos se entrenan y evalúan en igualdad de condiciones, y la selección del modelo definitivo se realiza a partir de su desempeño medido con métricas estadísticas, de manera que quede respaldada por los datos.

4.20. Entrenamiento del modelo

La primera celda carga las dos fuentes ya depuradas y muestra sus dimensiones (el código completo corresponde al anexo C). El archivo de sensores tiene 13.248 filas y 5 columnas, que corresponden a la marca de tiempo y las cuatro variables medidas en campo durante los tres meses de recolección. El archivo del IDEAM, mucho más compacto, tiene 92 filas y 2 columnas, una por cada día del periodo con su respectivo valor de precipitación. La diferencia de tamaño no es un problema, sino el reflejo de que ambas fuentes miden a ritmos distintos, pues los sensores registran cada cinco minutos y el IDEAM entrega un solo dato por día.

Figura 34. carga de los datasets



```
[1]
✓ 0s
import pandas as pd
import numpy as np

# Cargar las dos fuentes ya depuradas
sensores = pd.read_csv('/content/sensores_limpio.csv', parse_dates=['timestamp'])
ideam = pd.read_csv('/content/ideam_limpio.csv', parse_dates=['fecha'])

print("Sensores:", sensores.shape)
print("IDEAM:", ideam.shape)

... Sensores: (13248, 5)
    IDEAM: (92, 2)
```

Fuente: Autores

Esta celda es la que transforma dos archivos separados en uno solo listo para modelar. A partir de la serie diaria de lluvia se construyen tres variables derivadas que capturan el historial reciente de agua: la lluvia acumulada de los últimos tres y siete días, y los días transcurridos desde el último evento. Luego ambas fuentes se unen por la fecha, de modo que a cada lectura de cinco minutos se le asocia la precipitación del día al que pertenece. El resultado es un conjunto integrado de 13.248 filas y 10 columnas, que conserva todas las mediciones

originales y suma las variables de lluvia necesarias para la predicción. Esta información se observa en la siguiente figura.

Figura 35. Preparación e integración del conjunto

```
# Variables derivadas a partir de la serie diaria de lluvia
ideam = ideam.sort_values('fecha').reset_index(drop=True)
ideam['lluvia_3d'] = ideam['precipitacion_mm'].rolling(3, min_periods=1).sum()
ideam['lluvia_7d'] = ideam['precipitacion_mm'].rolling(7, min_periods=1).sum()

# Dias transcurridos desde el ultimo evento de lluvia
dias = 0
acumulado = []
for valor in ideam['precipitacion_mm']:
    dias = 0 if valor > 0 else dias + 1
    acumulado.append(dias)
ideam['dias_sin_lluvia'] = acumulado

# Integrar ambas fuentes por la fecha comun
sensores['fecha'] = sensores['timestamp'].dt.normalize()
df = sensores.merge(ideam, on='fecha', how='left')

print("Conjunto integrado:", df.shape)
```

Conjunto integrado: (13248, 10)

Fuente: autores

El paso siguiente es donde se separa de forma explícita lo que el modelo recibe de lo que debe estimar. Las variables de entrada son la lluvia acumulada de siete y tres días, los días sin lluvia y la temperatura del suelo, es decir, las que mostraron mayor influencia en el análisis previo. La variable objetivo es la humedad del suelo. Esta distinción es la base de todo el aprendizaje, porque define qué información se usa para predecir y qué resultado se busca obtener. En la siguiente figura se observa el código empleado.

Figura 36. Definición de variables de entrada y objetivo

```
1) 0s # Variables predictoras y variable a estimar
entradas = ['lluvia_7d', 'lluvia_3d', 'dias_sin_lluvia', 'temp_suelo_c']
objetivo = 'hum_suelo_pct'

x = df[entradas]
y = df[objetivo]
```

Fuente: autores

En la siguiente celda de Google Colab se divide el conjunto en una parte para entrenar y otra para evaluar, respetando el orden del tiempo. El 80 por ciento inicial queda para el entrenamiento, con 10.598 registros, y el 20 por ciento final para la prueba, con 2.650 registros. La elección de una partición temporal en lugar de una aleatoria es deliberada, pues en una serie de tiempo las lecturas cercanas se parecen mucho entre sí, y mezclarlas al azar permitiría que el modelo viera durante el entrenamiento datos casi idénticos a los de la evaluación, lo que daría una impresión falsa de buen desempeño. Al separar por periodos, la prueba se hace con datos que el modelo no ha visto. En la siguiente figura se puede ver el fragmento de código que se usa para tal fin.

Figura 37. Partición temporal

```
15 # Particion temporal: el pasado entrena, el futuro evalua
corte = int(len(df) * 0.8)
X_train, X_test = X.iloc[:corte], X.iloc[corte:]
y_train, y_test = y.iloc[:corte], y.iloc[corte:]

print("Entrenamiento:", X_train.shape[0], "registros")
print("Prueba:", X_test.shape[0], "registros")

Entrenamiento: 10598 registros
Prueba: 2650 registros
```

Fuente: autores

A continuación, las variables a una escala comparable, de manera que ninguna influya más que otra por el simple hecho de manejar números de mayor magnitud. Un aspecto importante es que el escalador se ajusta únicamente con los datos de entrenamiento y después se aplica a los de prueba. Con ello se evita que la información del periodo de evaluación intervenga en la preparación, lo que conservaría la validez de la prueba. En la siguiente figura se puede ver el código.

Figura 38. Normalización

```
from sklearn.preprocessing import StandardScaler

# El escalador se ajusta solo con los datos de entrenamiento
escalador = StandardScaler()
X_train = escalador.fit_transform(X_train)
X_test = escalador.transform(X_test)
```

Fuente: autores

Después de la normalización se entrena los dos algoritmos seleccionados. Cada modelo aprende de los 10.598 registros de entrenamiento las relaciones entre las variables de lluvia y la humedad del suelo, y el mensaje de confirmación indica que tanto la regresión lineal como el Random Forest quedan ajustados y listos. En este punto los modelos ya están entrenados, aunque todavía no se conoce su desempeño, pues eso se determina en la etapa siguiente, cuando generan predicciones sobre el periodo de prueba y se comparan mediante métricas. La figura siguiente se puede observar el código usado en el entorno de desarrollo Google Colab.

Figura 39. Entrenamiento de los modelos

```
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor

# Modelos que se van a entrenar y comparar
modelos = {
    'Regresion lineal': LinearRegression(),
    'Random Forest': RandomForestRegressor(n_estimators=200, random_state=42),
}

# Entrenar cada modelo con los datos de entrenamiento
for nombre, modelo in modelos.items():
    modelo.fit(X_train, y_train)
    print(f"{nombre}: entrenado")

Regresion lineal: entrenado
Random Forest: entrenado
```

Fuente: autores

4.21. Validación del modelo

La validación busca comprobar que los modelos no solo se ajustan a los datos con los que aprenden, sino que conservan su capacidad de acierto ante información nueva. Ese contraste resulta esencial para descartar el sobreajuste, un problema que aparece cuando un modelo memoriza el detalle del entrenamiento y, por lo mismo, falla al enfrentarse a datos que no ha visto. En la predicción de variables ambientales, una validación cuidadosa es la que separa un buen desempeño real de uno aparente (Islam et al., 2025). Para ello se aplican tres comprobaciones complementarias y una revisión visual.

La primera comprobación mide el desempeño de cada modelo sobre el periodo de prueba, es decir, sobre los datos que quedaron apartados y que el modelo no utiliza para aprender. El siguiente fragmento calcula tres métricas por modelo: el error cuadrático medio, el error absoluto medio y el coeficiente de

determinación. En la siguiente figura se observan esos valores, que resumen qué tan cerca quedan las estimaciones de la humedad real.

Figura 40. Métricas

```
import numpy as np
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

# Evaluar cada modelo en el periodo de prueba
for nombre, modelo in modelos.items():
    pred = modelo.predict(X_test)
    rmse = np.sqrt(mean_squared_error(y_test, pred))
    mae = mean_absolute_error(y_test, pred)
    r2 = r2_score(y_test, pred)
    print(f"{nombre} -> RMSE: {rmse:.2f} | MAE: {mae:.2f} | R2: {r2:.3f}")

... Regresion lineal -> RMSE: 4.25 | MAE: 3.31 | R2: 0.538
    Random Forest -> RMSE: 6.04 | MAE: 4.77 | R2: 0.066
```

Fuente: autores

La segunda comprobación es la más directa para detectar el sobreajuste, y consiste en comparar el desempeño del modelo en los datos de entrenamiento con el que alcanza en los de prueba. La lógica es sencilla: si un modelo acierta casi a la perfección en el entrenamiento, pero cae de forma notable en la prueba, está memorizando en lugar de generalizar. El siguiente fragmento calcula el coeficiente de determinación en ambos conjuntos. En la siguiente figura se aprecia esa diferencia, que en los modelos más complejos suele ser amplia y delata el sobreajuste.

Figura 41. detector sobreajuste

```
# Comparar el desempeño en entrenamiento frente al de prueba
for nombre, modelo in modelos.items():
    r2_train = r2_score(y_train, modelo.predict(X_train))
    r2_test = r2_score(y_test, modelo.predict(X_test))
    print(f"{nombre} -> R2 entrenamiento: {r2_train:.3f} | R2 prueba: {r2_test:.3f}")

Regresion lineal -> R2 entrenamiento: 0.597 | R2 prueba: 0.538
Random Forest -> R2 entrenamiento: 0.988 | R2 prueba: 0.066
```

Fuente: autores

La tercera comprobación aplica una validación cruzada adaptada a las series de tiempo. En lugar de una sola división entre pasado y futuro, esta técnica repite la evaluación sobre varios cortes sucesivos, entrenando siempre con el tramo anterior y evaluando con el siguiente, de modo que respeta el orden temporal y nunca usa el futuro para predecir el pasado. Así se obtiene una visión más completa de la estabilidad del modelo a lo largo del periodo. El siguiente fragmento la ejecuta con cinco cortes. En la siguiente figura se muestran los resultados de cada tramo y su promedio, que revelan qué tan consistente es el desempeño ante condiciones cambiantes.

Figura 42. validación cruzada

```
from sklearn.model_selection import TimeSeriesSplit, cross_val_score
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor

# Validacion cruzada para series de tiempo (cinco pliegues)
tscv = TimeSeriesSplit(n_splits=5)
validation = {
    'Regresion lineal': LinearRegression(),
    'Random Forest': RandomForestRegressor(n_estimators=200, random_state=42),
}

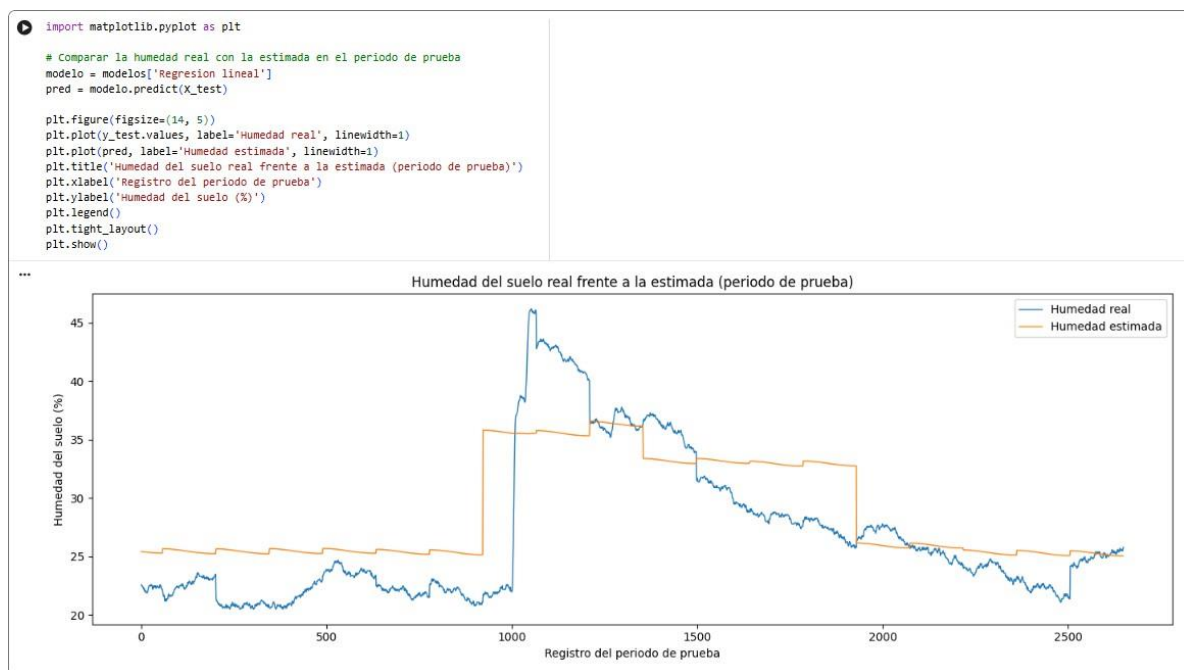
for nombre, modelo in validation.items():
    flujo = make_pipeline(StandardScaler(), modelo)
    scores = cross_val_score(flujo, X, y, cv=tscv, scoring='r2')
    print(f"{nombre} -> R2 por pliegue: {np.round(scores, 3)} | promedio: {scores.mean():.3f}")

*** Regresion lineal -> R2 por pliegue: [-4.8   -0.228  0.539 -0.03   0.544] | promedio: -0.795
    Random Forest -> R2 por pliegue: [-4.91  -0.955  0.786 -2.476 -0.09 ] | promedio: -1.529
```

Fuente: autores

Por último, una revisión visual complementa a las métricas. El siguiente fragmento grafica, sobre el periodo de prueba, la humedad del suelo real junto a la estimada por el modelo. En la siguiente figura se comparan ambas curvas, lo que permite ver no solo cuánto se equivoca el modelo, sino en qué momentos se acerca o se aparta del valor real, algo que un número por sí solo no refleja.

Figura 43. comprobación visual



Fuente: autores

4.21.1. Resultados de la validación

Los resultados de la validación cuentan una historia coherente y, a la vez, instructiva sobre los límites de cada modelo. Las métricas sobre el periodo de prueba muestran que el modelo más sencillo supera con claridad al más sofisticado. La regresión lineal alcanza un coeficiente de determinación de 0,538, con errores de 4,25 y 3,31 puntos de humedad, mientras que el Random Forest se queda en 0,066, con errores mayores en ambas medidas. Dicho de otro modo, el modelo que se esperaba más potente predice poco mejor que un valor promedio, lo que confirma el acierto de haber incluido una referencia simple contra la cual comparar.

La comparación entre entrenamiento y prueba explica ese desenlace y constituye el hallazgo central de la validación. El Random Forest obtiene un coeficiente de 0,988 en entrenamiento pero de apenas 0,066 en prueba, una brecha

enorme que es la señal inequívoca del sobreajuste: el modelo memoriza casi a la perfección los datos con los que aprende y fracasa ante los que no ha visto. La regresión lineal, en cambio, pasa de 0,597 a 0,538, una diferencia mínima que indica que generaliza bien y no depende de haber visto antes cada caso (Islam et al., 2025). Esta comparación es, justamente, la que permite descartar el sobreajuste con evidencia y no por intuición.

La validación cruzada temporal somete a los modelos a una prueba más exigente, y sus resultados invitan a la cautela. En ambos casos el promedio del coeficiente es negativo, de menos 0,795 para la regresión lineal y menos 1,529 para el Random Forest, con una fuerte variación entre tramos: algunos arrojan valores razonables, cercanos a 0,54, mientras que otros caen muy por debajo de cero. Un coeficiente negativo significa que, en esos tramos, el modelo predice peor que si simplemente repitiera el promedio. Este comportamiento no es un error, sino el reflejo de una dificultad real, pues cuando el modelo se entrena con un tramo corto del periodo y debe estimar condiciones que no aparecían en él, como el paso de un mes seco a uno muy lluvioso, su capacidad de acierto se desploma. La lección es clara, y es que generalizar a lo largo de regímenes de lluvia distintos exige más historia de la que cubren tres meses.

La gráfica obtenida complementa estas cifras con una lectura visual. En ella, la curva de humedad estimada sigue el nivel general de la humedad real y reproduce sus grandes movimientos, como el ascenso brusco asociado a las lluvias y el descenso sostenido de las semanas secas. Sin embargo, la estimación aparece escalonada, ya que se mantiene casi constante dentro de cada día, mientras que la humedad real fluctúa de forma más fina. La razón es que las variables más influyentes provienen de la lluvia, que solo cambia de un día a otro, de modo que el modelo capta la tendencia diaria pero no las pequeñas variaciones internas de la jornada, que dependen de factores como la evaporación a lo largo del día.

4.22. Ajuste de hiperparámetros

Los modelos de aprendizaje automático dependen de un conjunto de valores de configuración que no se aprenden de los datos, sino que se fijan antes del entrenamiento. Estos valores, conocidos como hiperparámetros, determinan aspectos como la complejidad del modelo o la manera en que aprende, y su elección influye de forma directa en el desempeño y en el riesgo de sobreajuste. El ajuste de hiperparámetros consiste, precisamente, en buscar la combinación que ofrece el mejor equilibrio entre capacidad de predicción y generalización (Islam et al., 2025).

En este trabajo el ajuste se concentra en el modelo de Random Forest, ya que es el que posee varios hiperparámetros relevantes y, además, el que muestra señales de sobreajuste durante la validación. La regresión lineal no requiere este proceso, pues carece de parámetros de configuración que optimizar. Los hiperparámetros sobre los que se trabaja en el Random Forest son, entre otros, el número de árboles del conjunto, la profundidad máxima de cada árbol y la cantidad mínima de muestras exigida para dividir un nodo o formar una hoja. Varios de ellos controlan directamente la complejidad del modelo, de modo que limitarlos ayuda a reducir el sobreajuste observado.

Para encontrar la mejor combinación se emplea la búsqueda en malla, una técnica que evalúa de manera sistemática todas las combinaciones posibles dentro de un conjunto de valores definido previamente para cada hiperparámetro. Cada combinación se valora mediante validación cruzada, de forma que su desempeño se mide sobre varios cortes de los datos y no sobre una sola división, lo que reduce la probabilidad de elegir una configuración afortunada por azar. Cuando el espacio de combinaciones es muy amplio, esta búsqueda exhaustiva se vuelve costosa, y en su lugar se recurre a la búsqueda aleatoria, que explora un número fijo de combinaciones elegidas al azar y suele alcanzar buenos resultados con mucho menos cómputo (Afzal et al., 2025).

Un cuidado particular en este proyecto es que la validación cruzada empleada durante la búsqueda respeta el orden temporal de los datos, entrenando siempre con el pasado y evaluando con el futuro, en coherencia con la estrategia aplicada en la validación del modelo. La combinación seleccionada es la que obtiene el mejor promedio en la métrica elegida, mientras que el conjunto de prueba se mantiene al margen de todo el proceso, de modo que la evaluación final conserve su independencia. el ajuste de hiperparámetros se integra con el resto de la metodología y busca, ante todo, contener el sobreajuste sin comprometer la validez de la prueba.

4.23. Métricas de evaluación

Para medir el desempeño de los modelos predictivos se emplean tres métricas estadísticas de uso extendido en los problemas de regresión, que valoran desde distintos ángulos qué tan cerca quedan las estimaciones de los valores reales. El uso conjunto de varias métricas es recomendable, pues cada una aporta una perspectiva particular y ninguna, por sí sola, describe todo el comportamiento del modelo (Islam et al., 2025).

La primera es el error cuadrático medio, conocido por sus siglas en inglés como RMSE. Esta métrica calcula la raíz del promedio de las diferencias al cuadrado entre los valores reales y los estimados, de modo que su resultado queda expresado en las mismas unidades de la variable, en este caso el porcentaje de humedad del suelo. Una particularidad importante es que, al elevar al cuadrado, penaliza con más fuerza los errores grandes, por lo que resulta útil cuando las equivocaciones severas son especialmente indeseables. Cuanto más bajo es su valor, mejor es el desempeño del modelo.

La segunda es el error absoluto medio, abreviado como MAE. Corresponde al promedio de las diferencias absolutas entre lo real y lo estimado, también en las

mismas unidades de la variable. A diferencia del anterior, trata todos los errores por igual sin importar su tamaño, lo que la convierte en una medida más estable frente a valores atípicos y de interpretación directa, ya que indica en cuántos puntos de humedad se equivoca el modelo en promedio. Al igual que el RMSE, un valor menor refleja mejores estimaciones. La comparación entre ambas métricas aporta información adicional, pues una diferencia amplia entre el RMSE y el MAE sugiere la presencia de algunos errores grandes que elevan el primero.

La tercera es el coeficiente de determinación, representado como R^2 . A diferencia de las dos anteriores, no se expresa en unidades de humedad, sino como una proporción que indica qué parte del comportamiento de la variable real logra explicar el modelo. Su valor ideal es uno, que corresponde a una predicción perfecta, mientras que un valor de cero equivale a no predecir mejor que el simple promedio de los datos. Puede incluso volverse negativo cuando el modelo estima peor que ese promedio, situación que delata un desempeño deficiente. Por esta razón, en este caso la interpretación se invierte respecto a los errores, ya que un valor más alto es el que indica un mejor ajuste.

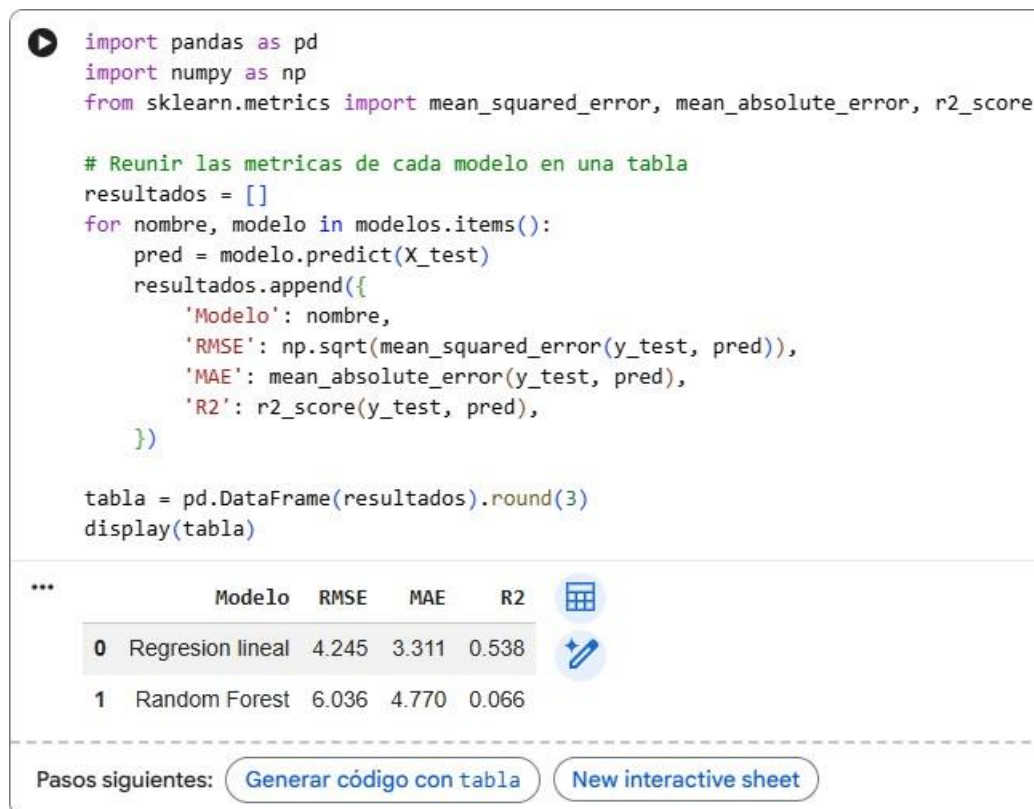
4.24. Comparación de resultados

La evaluación del desempeño reúne en un mismo lugar los resultados de los modelos y los compara con criterios objetivos, de modo que la elección del mejor no dependa de una impresión sino de la evidencia. Para ello se emplean tres métricas estadísticas de uso extendido en la predicción de variables ambientales, el error cuadrático medio, el error absoluto medio y el coeficiente de determinación (Afzal et al., 2025), acompañadas de representaciones visuales que ayudan a interpretar su significado.

El primer paso consolida en una sola tabla las métricas de cada modelo sobre el periodo de prueba, lo que permite compararlos de un vistazo. El siguiente

fragmento calcula las tres métricas y las organiza de forma ordenada. En la siguiente figura se observa ese resumen, en el que cada fila corresponde a un modelo y cada columna a una métrica.

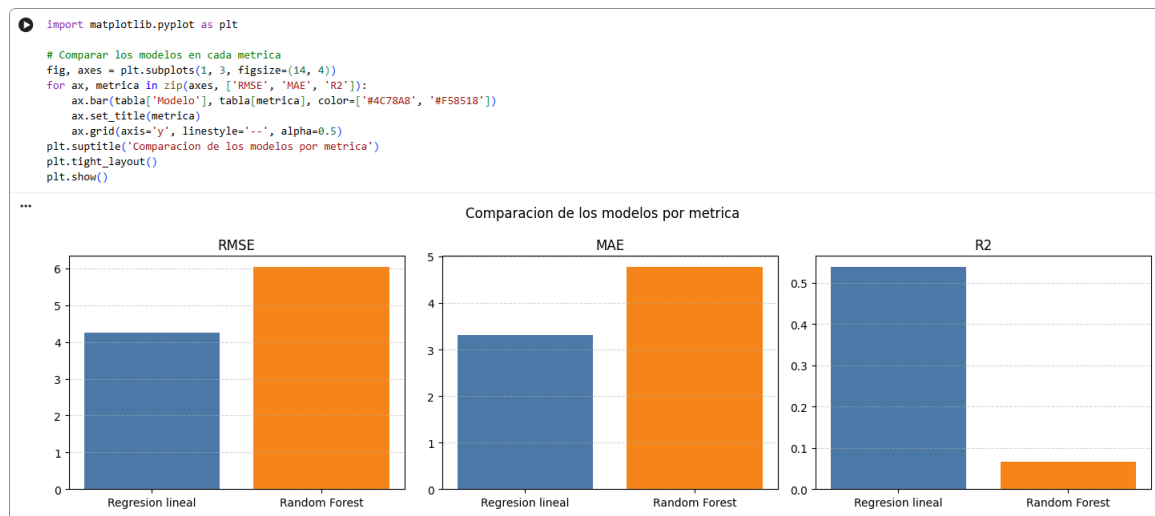
Figura 44. Tabla resumen metrico



Fuente: autores

Para hacer más evidente la comparación, las mismas métricas se llevan a un gráfico de barras que enfrenta a los dos modelos en cada indicador. El siguiente fragmento genera esa comparación. En la siguiente figura se aprecia que, en los errores, una barra más baja refleja un mejor desempeño, mientras que en el coeficiente de determinación ocurre lo contrario, ya que un valor más alto es mejor.

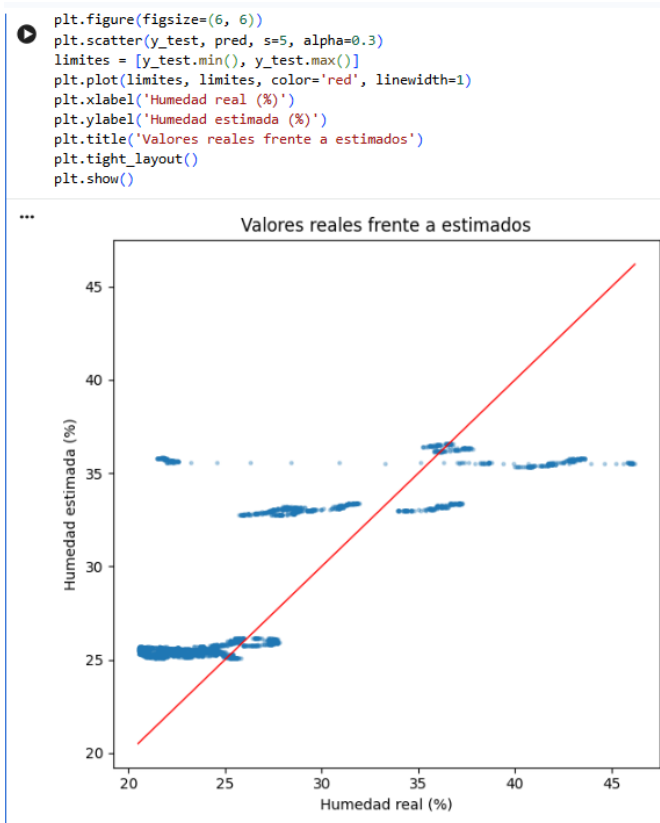
Figura 45. comparación gráfica de modelos



Fuente: autores

Más allá de los números, conviene observar cómo se relacionan los valores estimados con los reales. El siguiente fragmento construye un gráfico de dispersión en el que cada punto representa una estimación frente a su valor verdadero, junto a una línea diagonal que marca el acierto perfecto. En la siguiente figura, los puntos cercanos a esa línea corresponden a buenas estimaciones, mientras que los alejados señalan los mayores errores, y su disposición revela si el modelo tiende a sobrestimar o a subestimar en ciertos rangos.

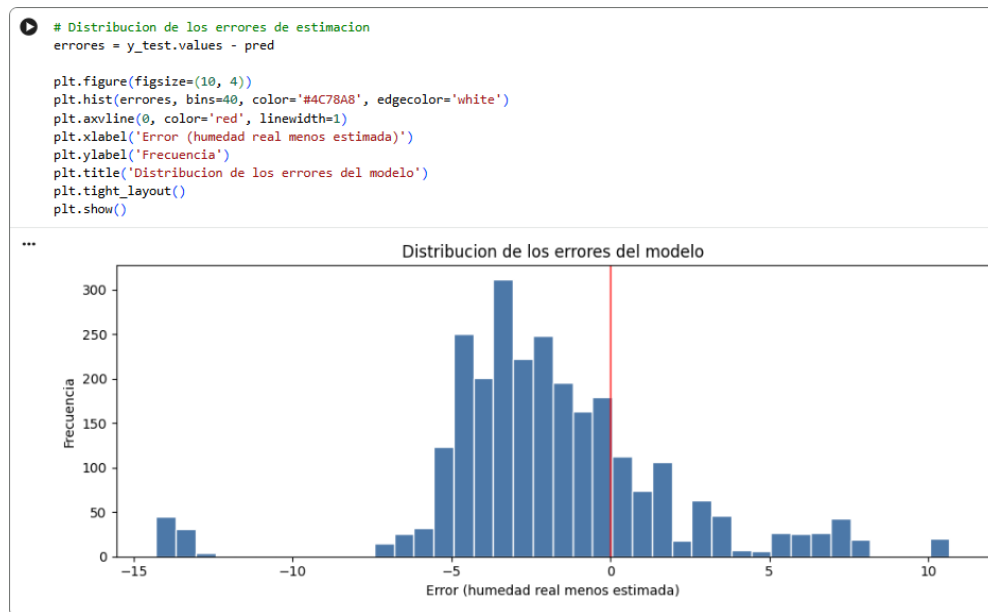
Figura 46. Dispersión de los valores reales frente a los estimados



Fuente: autores

se examina la distribución de los errores, es decir, la diferencia entre la humedad real y la estimada en cada registro. El siguiente fragmento la representa en un histograma centrado en el cero. En la siguiente figura, una distribución concentrada alrededor del cero indica un modelo sin sesgo sistemático, mientras que un desplazamiento hacia un lado revela que el modelo tiende a estimar de más o de menos.

Figura 47. Distribución de los errores de estimación



Fuente: autores

La evaluación del desempeño permite llegar a una conclusión clara: de los dos modelos probados, la regresión lineal es la que ofrece la mejor estimación de la humedad del suelo. Con un error cuadrático medio de 4,245, un error absoluto medio de 3,311 y un coeficiente de determinación de 0,538, supera en las tres métricas al Random Forest, que se queda en 6,036, 4,770 y 0,066. El gráfico de barras lo muestra sin ambigüedad, pues la regresión lineal presenta barras más bajas en los dos errores y una barra mucho más alta en el coeficiente de determinación, que es donde conviene estar arriba. Así, el modelo más sencillo resulta también el más confiable, lo que confirma que una mayor complejidad no garantiza mejores resultados.

El gráfico de dispersión matiza ese logro y señala con honestidad los límites del modelo. Los puntos siguen una tendencia ascendente, lo que indica que la

estimación acompaña el comportamiento general de la humedad real, pero se agrupan en bandas horizontales y se alejan de la línea de acierto perfecto en los valores altos, donde el modelo tiende a quedarse corto. Ese patrón escalonado se explica porque las variables más influyentes provienen de la lluvia, que cambia una vez al día, de modo que la predicción capta el nivel diario pero no las variaciones finas dentro de la jornada.

La distribución de los errores completa el diagnóstico. La mayor parte se concentra entre menos cinco y cinco puntos de humedad, lo que refleja estimaciones razonablemente cercanas a la realidad, aunque el centro de la distribución queda ligeramente desplazado hacia los valores negativos. Ese desplazamiento revela un pequeño sesgo sistemático, pues el modelo tiende a estimar la humedad un poco por encima del valor real. No es un error grave, pero sí una tendencia que conviene tener presente al interpretar sus resultados.

la evaluación cumple su propósito y deja un balance equilibrado. Identifica a la regresión lineal como el modelo más adecuado para este conjunto de datos, cuantifica su desempeño con métricas claras y, al mismo tiempo, reconoce que la predicción conserva un margen de mejora, tanto en la precisión de los valores extremos como en la captura del detalle dentro del día.

4.25. Análisis de precisión del modelo

El modelo seleccionado, la regresión lineal, estima la humedad del suelo con una precisión moderada y, sobre todo, desigual según las condiciones ambientales. En promedio se equivoca alrededor de tres puntos de humedad, lo que resulta aceptable para una primera aproximación, pero esa cifra global esconde un comportamiento que cambia de un escenario a otro.

El modelo acierta mejor en las condiciones intermedias, que son las más frecuentes en el periodo de estudio. Durante los tramos de secado gradual, cuando

la humedad desciende poco a poco tras varios días sin lluvia, la estimación sigue de cerca el valor real, pues las variables de lluvia acumulada describen bien esa tendencia. En cambio, la precisión se deteriora en los extremos. Tras los aguaceros fuertes, cuando la humedad del suelo sube de forma brusca, el modelo reacciona con lentitud y tiende a quedarse por debajo del valor real, de modo que subestima los picos más altos. Algo similar ocurre en los días más secos, donde las estimaciones se agrupan en unos pocos niveles y pierden sensibilidad ante las variaciones menores.

Este comportamiento se explica por la naturaleza de las variables que alimentan el modelo. Al provenir la información más influyente de la precipitación, que cambia una vez al día, la estimación capta bien el nivel general de cada jornada pero no las fluctuaciones finas que ocurren dentro de ella, asociadas a procesos como la evaporación a lo largo del día.

4.26. Limitaciones del modelo desarrollado

El desarrollo del modelo deja ver varias limitaciones que conviene reconocer con claridad, pues delimitan el alcance de los resultados y orientan el trabajo futuro. Se agrupan en tres planos: el técnico, el metodológico y el operativo.

En el plano técnico, la principal restricción proviene de la resolución de los datos de lluvia. La precipitación del IDEAM se registra una sola vez al día, mientras que los sensores miden cada cinco minutos, de modo que la variable que más influye en la humedad del suelo entra al modelo con un detalle temporal mucho menor que el resto. Esto limita la capacidad de estimar las variaciones que ocurren dentro de la jornada. A ello se suma que el sistema no incorpora un pluviómetro propio, por lo que depende de una estación externa, la de Albania, que si bien es la más cercana no se encuentra exactamente sobre el cultivo.

En el plano metodológico, la limitación más relevante es la extensión del periodo de datos. Tres meses de registro, por valiosos que sean, no alcanzan a cubrir la variabilidad climática de un año completo, y la validación cruzada lo confirma al mostrar un desempeño inestable cuando el modelo debe estimar condiciones distintas a las vistas durante el entrenamiento (Afzal et al., 2025).

En el plano operativo, el sistema funciona en jornadas de doce horas diarias, condicionadas por la autonomía de la batería portátil, lo que deja sin registro la franja nocturna y genera huecos periódicos en la serie. La dependencia de la conexión Wi-Fi es otra restricción, pues una interrupción del servicio implica la pérdida de las lecturas de ese lapso al no existir un almacenamiento local de respaldo. Estas limitaciones no invalidan los resultados, pero sí precisan su alcance y señalan mejoras concretas para una siguiente etapa, como ampliar el periodo de recolección, incorporar una fuente de lluvia más cercana o de mayor frecuencia y dotar al sistema de un respaldo de datos en campo.

4.27. Propuesta de apoyo para la toma de decisiones de riego

La propuesta se orienta a convertir las predicciones de humedad del suelo en una guía sencilla para la gestión del riego dentro del cafetal. La idea central es que el caficultor deje de depender únicamente de la observación empírica y cuente con una referencia objetiva sobre el estado hídrico del terreno, capaz de anticiparse a situaciones de déficit o exceso de agua.

La propuesta funciona como un sistema de apoyo y no de reemplazo. El modelo entrega una estimación de la humedad del suelo, esa estimación se compara con unos rangos de referencia definidos para el cultivo, y de esa comparación surge una recomendación orientativa, como regar, esperar o suspender el riego ante lluvias previstas. La decisión final sigue en manos del productor, que conoce su terreno, pero ahora apoyada en datos. Este enfoque, en

el que la analítica guía la gestión hídrica sin sustituir el criterio humano, es coherente con la tendencia actual de la agricultura de precisión hacia un uso más eficiente del agua (Soussi et al., 2024).

4.28. Definición de condiciones de humedad del suelo

Para que una predicción sea útil, primero debe poder interpretarse, y eso exige traducir el porcentaje de humedad en estados comprensibles para el manejo del cultivo. Con ese fin se definen rangos que agrupan los valores de humedad en condiciones cualitativas, de modo que una cifra se convierta en una situación concreta que oriente la acción.

Los umbrales se obtienen a partir de la distribución real de la humedad registrada durante el periodo de estudio, dividiendo los datos en tres tercios de frecuencia similar. Sobre esa base se establecen tres condiciones de referencia. La primera es la condición de suelo seco, que corresponde a los valores por debajo de 24,7 por ciento y señala escasez de agua y un posible estrés hídrico para la planta. La segunda es la condición adecuada, que abarca el rango entre 24,7 y 30,0 por ciento y representa el estado favorable para el desarrollo del cultivo. La tercera es la condición de suelo húmedo, que agrupa los valores por encima de 30,0 por ciento, asociados a los días posteriores a las lluvias, en los que el riego resulta innecesario e incluso podría ser perjudicial.

Llama la atención que el rango total de variación sea relativamente estrecho, pues la franja adecuada se mueve en apenas unos cinco puntos de humedad. Esto refleja que, en el periodo analizado, el suelo del cultivo se mantiene la mayor parte del tiempo en niveles moderados y que los cambios relevantes ocurren dentro de un margen acotado, lo que hace aún más importante disponer de una referencia precisa para distinguir una condición de otra.

Conviene precisar que estos rangos son orientativos y específicos de las condiciones del lote estudiado, no valores universales. Los umbrales exactos dependen del tipo de suelo, de la etapa del cultivo y de la calibración del sensor, por lo que se plantea revisarlos periódicamente y ajustarlos con la experiencia del caficultor y, de ser posible, con criterios agronómicos propios del cultivo del café. A medida que el sistema acumule más datos a lo largo del tiempo, estos límites podrán afinarse para representar mejor el comportamiento hídrico del terreno. En la siguiente figura se observa el código ejecutado para obtener los resultados descritos.

Figura 48. Umbrales de referencia

```
# Umbrales de referencia a partir de la distribucion real de la humedad
umbral_bajo = df['hum_suelo_pct'].quantile(0.33)
umbral_alto = df['hum_suelo_pct'].quantile(0.66)

print("Suelo seco: por debajo de", round(umbral_bajo, 1), "%")
print("Suelo adecuado: entre", round(umbral_bajo, 1), "y", round(umbral_alto, 1), "%")
print("Suelo humedo: por encima de", round(umbral_alto, 1), "%")

... Suelo seco: por debajo de 24.7 %
    Suelo adecuado: entre 24.7 y 30.0 %
    Suelo humedo: por encima de 30.0 %
```

Fuente: autores

4.29. Generación e interpretación de predicciones

Una vez entrenado y validado, el modelo cumple su función principal, que es generar predicciones de humedad del suelo a partir de las condiciones ambientales del cultivo. La predicción no aparece de forma espontánea, sino que surge en un momento preciso: cuando al modelo se le entregan los valores de las variables de entrada y este devuelve, mediante su método de estimación, el porcentaje de humedad correspondiente. Ese resultado es la salida que luego se interpreta para orientar el manejo del agua.

El procedimiento parte de reunir las condiciones actuales del cultivo en una sola observación, con las cuatro variables que el modelo utiliza: la lluvia acumulada de los últimos siete y tres días, los días transcurridos desde el último evento de lluvia y la temperatura del suelo. Estas variables se escalan con el mismo transformador empleado durante el entrenamiento, para que lleguen al modelo en las mismas condiciones con que aprendió, y enseguida el modelo entrega su estimación. El siguiente fragmento de código en la figura ilustra ese flujo completo.

Figura 49. generador de predicciones

```
import pandas as pd

# Condiciones actuales (una sola observacion con las variables de entrada)
condiciones = pd.DataFrame([{'lluvia_7d': 5.5,
                              'lluvia_3d': 0.0,
                              'dias_sin_lluvia': 4,
                              'temp_suelo_C': 23.1
                              }])

# El modelo entrega la prediccion
condiciones_esc = escalador.transform(condiciones)
prediccion = modelo.predict(condiciones_esc)[0]

print("Humedad estimada:", round(prediccion, 1), "%")
print(interpretar_humedad(prediccion, umbral_bajo, umbral_alto))

*** Humedad estimada: 21.9 %
    Suelo seco: considerar riego
```

Fuente: autores

El resultado de este fragmento es una humedad estimada de 21,9 por ciento, que el sistema clasifica en la condición de suelo seco y acompaña de la recomendación de considerar el riego. Con ese valor, la cadena completa queda a la vista: las condiciones de entrada producen una estimación numérica, esa estimación se ubica dentro de los rangos definidos para el cultivo, y de ahí nace una recomendación concreta de manejo. La cifra de 21,9 por ciento, al quedar por

debajo del umbral de 24,7 por ciento, cae en la franja baja del rango observado, lo que indica un terreno con escasez de agua y justifica la sugerencia de riego.

El valor de este proceso está en que transforma una salida técnica en información accionable. Para el caficultor, el número por sí solo diría poco, pero traducido a un estado del suelo y a una acción recomendada se convierte en una guía práctica para decidir cuándo intervenir. Esta forma de convertir las predicciones en recomendaciones de riego es, precisamente, el aporte que persiguen los sistemas de apoyo a la decisión en la agricultura de precisión (Afzal et al., 2025). Conviene recordar, eso sí, que la recomendación tiene un carácter orientativo y que su utilidad aumenta cuando se combina con la observación directa del cultivo y con el conocimiento del productor sobre su terreno.

4.30. Escenarios de apoyo al manejo hídrico

Para mostrar su utilidad más allá del análisis, la propuesta se concreta en varios escenarios de aplicación que ilustran cómo el modelo predictivo apoyaría el monitoreo y la gestión del riego en la práctica diaria del cafetal.

El primer escenario es el monitoreo continuo del estado hídrico. Como el sistema permanece conectado y actualiza sus lecturas de forma periódica, ofrece al caficultor un seguimiento permanente de la humedad del suelo sin necesidad de desplazarse al lote ni realizar mediciones manuales. Esta vigilancia constante facilita detectar a tiempo una tendencia hacia la sequedad y actuar antes de que el problema se agrave, algo que la observación empírica difícilmente logra con la misma oportunidad.

El segundo escenario es la planificación del riego en épocas secas. Durante los periodos prolongados sin lluvia, cuando la humedad desciende de manera sostenida, las predicciones permiten estimar el momento en que el suelo se acercará a la condición crítica. Con esa anticipación, el riego se programa de forma

preventiva, antes de que el cultivo entre en estrés hídrico, y no como una reacción tardía cuando el daño ya está en marcha.

El tercer escenario es la prevención del riego innecesario tras las lluvias. Después de un evento de precipitación, el modelo reconoce que el suelo se encuentra en condición húmeda y respalda la decisión de suspender el riego durante los días siguientes. Este uso tiene un efecto directo sobre el ahorro de agua y sobre los costos asociados, y se alinea con el objetivo de un uso más eficiente del recurso hídrico que persigue la agricultura de precisión (Soussi et al., 2024).

El cuarto escenario corresponde al registro histórico y al aprendizaje sobre el cultivo. Al acumular las predicciones y las mediciones a lo largo del tiempo, el sistema genera un registro del comportamiento hídrico del terreno que resulta valioso por sí mismo. Ese historial permite reconocer patrones estacionales, evaluar cómo responde el suelo a distintas condiciones de lluvia y afinar con el tiempo los rangos de referencia, de modo que las recomendaciones sean cada vez más ajustadas a la realidad de la finca.

5. RESULTADOS

El desarrollo del trabajo de grado produce cuatro resultados concretos que, en conjunto, dan solución a la problemática planteada. Dos de ellos son de naturaleza física, el circuito impreso y la carcasa fabricada en impresión 3D, que constituyen el soporte material del sistema de monitoreo. Los otros dos son de naturaleza informática, el conjunto de datos ambiental y el modelo predictivo, que representan el producto analítico del proyecto. A continuación, se documenta cada uno y se analiza en qué medida cumple con el propósito para el que fue creado.

El primer resultado es la placa de circuito impreso que integra el microcontrolador ESP32 y los tres sensores del sistema. El diseño reemplaza el montaje inicial sobre protoboard por una tarjeta compacta, en la que quedan distribuidos los componentes, las resistencias de polarización de los sensores y las conexiones necesarias para su funcionamiento.

El éxito de este resultado se valora por su funcionamiento en operación. La placa opera de manera estable durante todo el periodo de recolección, sin los falsos contactos propios de un montaje provisional, lo que confirma que el diseño cumple su objetivo de dar robustez al sistema en condiciones de campo. La llegada continua de datos a la plataforma durante los tres meses es, en la práctica, la evidencia de que el circuito funciona como se esperaba.

Figura 50. Circuito impreso



Fuente: autores

El segundo resultado es la carcasa diseñada y fabricada mediante impresión 3D, que aloja y protege la electrónica del sistema. Su función es resguardar el microcontrolador y la batería de la humedad, el polvo y la exposición directa a la intemperie, condiciones habituales en un cultivo a cielo abierto.

El análisis de este resultado se centra en la protección que brinda durante la operación. La carcasa permite instalar el equipo en campo y mantenerlo en funcionamiento a lo largo de la campaña sin afectaciones por el ambiente, lo que valida su diseño. Al tratarse de una pieza fabricada a la medida, se ajusta a las dimensiones de los componentes y facilita tanto el montaje como el mantenimiento, lo que representa una ventaja frente a una solución genérica.

Figura 51. Carcasa impresión 3D



Fuente: autores

El tercer resultado es el conjunto de datos construido durante el proyecto, que reúne las variables medidas por los sensores y la precipitación proveniente del IDEAM sobre una misma línea de tiempo. Este conjunto es, en sí mismo, un producto valioso, pues no existía antes y documenta el comportamiento ambiental del cultivo durante el periodo de estudio.

El conjunto final está conformado por los registros capturados cada cinco minutos en jornadas de doce horas, a lo largo de tres meses, a los que se suman la precipitación diaria y las variables derivadas de lluvia. La siguiente tabla resume sus características principales.

Tras el proceso de depuración, el conjunto queda completo, ordenado cronológicamente y libre de valores faltantes, lo que lo hace apto para el análisis. Su mayor valor es haber permitido construir y evaluar el modelo predictivo, función para la que fue concebido, además de servir como registro base para futuras investigaciones sobre el cultivo.

El cuarto resultado es el modelo predictivo de humedad del suelo, producto central del trabajo. Tras construir y comparar dos algoritmos, la regresión lineal y el Random Forest, se selecciona la regresión lineal por su mejor desempeño sobre los datos reales.

El modelo se evalúa con tres métricas estadísticas, cuyos valores se resumen en la siguiente tabla.

Tabla 5. resumen de los resultados comparación modelos predictivos

Modelo	RMSE	MAE	R ²
Regresión lineal	4,245	3,311	0,538
Random Forest	6,036	4,770	0,066

Nota. Elaboración propia a partir del resultado del código implementado

El análisis de estos resultados muestra un éxito moderado pero significativo. La regresión lineal supera al Random Forest en las tres métricas y logra explicar alrededor del 54 por ciento del comportamiento de la humedad del suelo, con un error medio cercano a tres puntos de humedad. Si bien queda margen de mejora, el resultado demuestra que existe una relación aprovechable entre las variables ambientales y la humedad del suelo, y que un modelo sencillo es capaz de capturarla. La comparación entre ambos algoritmos también aporta un hallazgo relevante, pues revela el sobreajuste del Random Forest y confirma, con evidencia, que una mayor complejidad no garantiza mejores resultados (Islam et al., 2025).

6. CONCLUSIONES

La integración de datos propios de sensores con información meteorológica oficial resulta determinante para estimar la humedad del suelo. El trabajo muestra que las mediciones instantáneas del cultivo, por sí solas, aportan poco a la predicción, y que el aporte real surge al incorporar el historial de lluvia proveniente del IDEAM. La razón de fondo es que la humedad del suelo no responde al clima del momento, sino a la lluvia recibida en los días previos y a su secado progresivo, de modo que son las variables que describen ese historial las que explican su comportamiento.

Un modelo sencillo logra representar mejor esa relación que uno complejo, y la razón está en la forma en que cada uno enfrenta datos nuevos. El modelo de mayor complejidad tiende a memorizar el detalle de los datos con los que aprende y pierde capacidad de respuesta ante situaciones que no ha visto, mientras que el modelo sencillo capta la tendencia general y la proyecta con mayor estabilidad. Este contraste deja ver que la sofisticación de un algoritmo no garantiza un mejor resultado, y que la elección debe guiarse por la naturaleza del problema y por una evaluación rigurosa.

La forma de validar los modelos condiciona la confianza que puede depositarse en ellos. Al respetar el orden temporal de los datos durante la evaluación, en lugar de mezclarlos al azar, se evita una apariencia engañosa de buen desempeño y se revela el comportamiento real del modelo frente a condiciones cambiantes. Esta decisión metodológica es la que permite afirmar que los resultados reflejan una capacidad predictiva genuina y no un artificio del procedimiento.

Durante el desarrollo se identifica como principal dificultad la diferencia de resolución temporal entre las dos fuentes de datos, pues los sensores registran cada pocos minutos mientras que la precipitación se reporta una vez al día. Esta disparidad amenaza con dejar sin aprovechar la variable más influyente. La solución adoptada consiste en unir ambas fuentes por la fecha y en construir variables derivadas que resumen el comportamiento reciente de la lluvia, con lo que la información diaria se incorpora de manera útil al análisis a pesar de su menor detalle.

Otra dificultad surge de la codificación y la estructura con que se descargan los datos meteorológicos, que impiden su uso directo. Esta situación se resuelve mediante un proceso ordenado de depuración que corrige el formato, filtra la estación de interés y completa la serie, lo que demuestra que la preparación cuidadosa de los datos es una etapa tan decisiva como el modelado mismo, ya que de su calidad depende la validez de todo el análisis posterior.

7. RECOMENDACIONES

Se recomienda extender el periodo de recolección de datos más allá de los tres meses iniciales, idealmente hasta cubrir un ciclo anual completo. Una serie más larga permitiría que el modelo aprenda de la variabilidad climática de las distintas épocas del año, en especial del contraste entre temporadas secas y lluviosas, lo que mejoraría su capacidad de generalizar ante condiciones que en esta etapa resultaron difíciles de estimar.

Conviene incorporar una fuente de lluvia de mayor resolución temporal o un pluviómetro propio integrado al sistema. Dado que la precipitación es la variable más influyente y que su registro diario limita la estimación de las variaciones dentro de la jornada, medir la lluvia directamente en el cultivo y con mayor frecuencia permitiría capturar esos cambios y afinar la predicción.

Se sugiere dotar al sistema de un almacenamiento local de respaldo, como una tarjeta de memoria, que conserve las lecturas cuando la conexión a internet falle. Esto reduciría la dependencia de la red disponible en la finca y evitaría la pérdida de datos durante las interrupciones del servicio, un riesgo presente en la configuración actual.

Para ampliar el alcance de la solución, se recomienda avanzar hacia la predicción de la humedad futura y no solo de la actual, incorporando pronósticos de lluvia como variable de entrada. Asimismo, resulta conveniente explorar algoritmos adicionales y técnicas de ajuste de parámetros, con el fin de superar el desempeño alcanzado y acercar la estimación a las exigencias del manejo del riego.

En cuanto a la validación del conocimiento generado, se recomienda contrastar los rangos de humedad definidos estadísticamente con criterios agronómicos propios del cultivo del café y del tipo de suelo de la zona, como la

capacidad de campo y el punto de marchitez. La participación de un profesional en agronomía permitiría respaldar esos umbrales con fundamento técnico y no solo con el comportamiento observado en los datos.

8. REFERENCIAS BIBLIOGRÁFICAS

Abdelmoneim, A. A., Khadra, R., Derardja, B., & Dragonetti, G. (2023). Internet of Things (IoT) for soil moisture tensiometer automation. *Micromachines*, 14(2), 263. <https://doi.org/10.3390/mi14020263>

Afzal, M., Saeed, I. A., Sohail, M. N., Saad, M. H. M., & Sarker, M. R. (2025). IoT-enabled adaptive watering system with ARIMA-based soil moisture prediction for smart agriculture. *IEEE Access*, 13, 27714-27728. <https://doi.org/10.1109/ACCESS.2025.3532447>

Sudarmaji, A., Gunawan, B., Nugraha, F., Arini, N., & Mulyani, S. (2024). Smart soil moisture control based on IoT ESP32 for horticulture cultivation in coastal area. *BIO Web of Conferences*, 96, 04001. <https://doi.org/10.1051/bioconf/20249604001>

Abdelmoneim, A. A., Khadra, R., Derardja, B., & Dragonetti, G. (2023). Internet of Things (IoT) for soil moisture tensiometer automation. *Micromachines*, 14(2), 263. <https://doi.org/10.3390/mi14020263>

Afzal, M., Saeed, I. A., Sohail, M. N., Saad, M. H. M., & Sarker, M. R. (2025). IoT-enabled adaptive watering system with ARIMA-based soil moisture prediction for smart agriculture. *IEEE Access*, 13, 27714-27728. <https://doi.org/10.1109/ACCESS.2025.3532447>

Analog Devices. (s.f.). DS18B20: Programmable resolution 1-Wire digital thermometer [Hoja de datos]. Recuperado el 5 de septiembre de 2026, de <https://www.analog.com>

DFRobot. (s.f.). Capacitive soil moisture sensor (SEN0193) [Documentación del producto]. Recuperado el 5 de septiembre de 2026, de <https://wiki.dfrobot.com>

Espressif Systems. (s.f.). ESP32 series datasheet. Recuperado el 5 de septiembre de 2026, de <https://www.espressif.com>

MathWorks. (s.f.). ThingSpeak documentation. Recuperado el 5 de septiembre de 2026, de <https://www.mathworks.com/help/thingspeak>

Sudarmaji, A., Gunawan, B., Nugraha, F., Arini, N., & Mulyani, S. (2024). Smart soil moisture control based on IoT ESP32 for horticulture cultivation in coastal area. *BIO Web of Conferences*, 96, 04001. <https://doi.org/10.1051/bioconf/20249604001>

Abdelmoneim, A. A., Khadra, R., Derardja, B., & Dragonetti, G. (2023). Internet of Things (IoT) for soil moisture tensiometer automation. *Micromachines*, 14(2), 263. <https://doi.org/10.3390/mi14020263>

Afzal, M., Saeed, I. A., Sohail, M. N., Saad, M. H. M., & Sarker, M. R. (2025). IoT-enabled adaptive watering system with ARIMA-based soil moisture prediction for smart agriculture. *IEEE Access*, 13, 27714-27728. <https://doi.org/10.1109/ACCESS.2025.3532447>

DFRobot. (s.f.). Capacitive soil moisture sensor (SEN0193) [Documentación del producto]. Recuperado el 5 de septiembre de 2026, de <https://wiki.dfrobot.com>

Sudarmaji, A., Gunawan, B., Nugraha, F., Arini, N., & Mulyani, S. (2024). Smart soil moisture control based on IoT ESP32 for horticulture cultivation in coastal area. *BIO Web of Conferences*, 96, 04001. <https://doi.org/10.1051/bioconf/20249604001>

Islam, M. B., Guerrieri, A., Gravina, R., Delaney, D. T., & Fortino, G. (2025). From traditional machine learning to fine-tuning large language models: A review for sensors-based soil moisture forecasting. *Sensors*, 25(22), 6903. <https://doi.org/10.3390/s25226903>

Agencia Nacional del Espectro. (2020). Resolución 105 de 2020. Por la cual se actualiza el Cuadro Nacional de Atribución de Bandas de Frecuencias. <https://www.ane.gov.co>

Congreso de la República de Colombia. (1993, 22 de diciembre). Ley 99 de 1993. Por la cual se crea el Ministerio del Medio Ambiente y se organiza el Sistema Nacional Ambiental (SINA). Diario Oficial No. 41.146.
<https://www.funcionpublica.gov.co/eva/gestornormativo>

Congreso de la República de Colombia. (1997, 6 de junio). Ley 373 de 1997. Por la cual se establece el programa para el uso eficiente y ahorro del agua. Diario Oficial No. 43.058.
<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=342>

Congreso de la República de Colombia. (2009, 30 de julio). Ley 1341 de 2009. Por la cual se definen principios y conceptos sobre la sociedad de la información y la organización de las Tecnologías de la Información y las Comunicaciones. Diario Oficial No. 47.426.
<https://www.funcionpublica.gov.co/eva/gestornormativo>

Congreso de la República de Colombia. (2013, 19 de julio). Ley 1672 de 2013. Por la cual se establecen los lineamientos para la adopción de una política pública de gestión integral de Residuos de Aparatos Eléctricos y Electrónicos (RAEE).
<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=53825>

Congreso de la República de Colombia. (2018, 27 de julio). Ley 1931 de 2018. Por la cual se establecen directrices para la gestión del cambio climático.
<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=87765>

Constitución Política de Colombia. (1991). Artículo 79 [Const.]. Gaceta Constitucional No. 116.

Instituto de Hidrología, Meteorología y Estudios Ambientales. (2026). *Precipitación diaria, estación Albania (24050110), mayo a julio de 2026* [Conjunto de datos]. Consulta y Descarga de Datos Hidrometeorológicos (DHIME).
<http://dhime.ideam.gov.co>

Google. (s.f.). *Google Colaboratory*. Recuperado el 8 de septiembre de 2026, de <https://colab.research.google.com>

McKinney, W. (2010). Data structures for statistical computing in Python. En *Proceedings of the 9th Python in Science Conference* (pp. 56-61). <https://doi.org/10.25080/Majora-92bf1922-00a>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

Vassiliadis, P. (2025). A survey of extract, transform, load technology. *International Journal of Data Warehousing and Mining*, 5(3), 1-27. <https://doi.org/10.4018/jdwm.2009070101>

Dong, Y., y Hansen, H. (2024). Design of an Internet of Things (IoT)-based photosynthetically active radiation (PAR) monitoring system. *AgriEngineering*, 6(1), 773-785. <https://doi.org/10.3390/agriengineering6010044>

Filamentive. (2025). *Outdoor 3D printing? 2025 UV resistant filament UK guide*. <https://www.filamentive.com/outdoor-3d-printing-2025-uv-resistant-filament-uk-guide/>

Sovol3D. (2026). *PLA vs PETG vs ASA for outdoor 3D printing: Which is best?* <https://www.sovol3d.com/blogs/news/pla-vs-petg-vs-asa-for-outdoor-3d-printing-which-is-best>

Zbotic. (2026). *3D printed IoT sensor enclosures: Weather-proof outdoor cases*. <https://zbotic.in/3d-printed-iot-sensor-enclosures-weather-proof-outdoor-cases/>

Houben, T., Pflug, B., Rakovec, O., y Attinger, S. (2025). Machine-learning based spatiotemporal prediction of soil moisture in a grassland hillslope. *Vadose Zone Journal*, 24(2), e70011. <https://doi.org/10.1002/vzj2.70011>

Wang, Y., Shi, L., Hu, Y., Hu, X., Song, W., y Wang, L. (2024). A comprehensive study of deep learning for soil moisture prediction. *Hydrology and Earth System Sciences*, 28(4), 917-943. <https://doi.org/10.5194/hess-28-917-2024>

Abdelmoneim, A. A., Khadra, R., Derardja, B., & Dragonetti, G. (2023). Internet of Things (IoT) for soil moisture tensiometer automation. *Micromachines*, 14(2), 263. <https://doi.org/10.3390/mi14020263>

Afzal, M., Saeed, I. A., Sohail, M. N., Saad, M. H. M., & Sarker, M. R. (2025). IoT-enabled adaptive watering system with ARIMA-based soil moisture prediction for smart agriculture. *IEEE Access*, 13, 27714-27728. <https://doi.org/10.1109/ACCESS.2025.3532447>

Islam, M. B., Guerrieri, A., Gravina, R., Delaney, D. T., & Fortino, G. (2025). From traditional machine learning to fine-tuning large language models: A review for sensors-based soil moisture forecasting. *Sensors*, 25(22), 6903. <https://doi.org/10.3390/s25226903>

Soussi, A., Zero, E., Sacile, R., Trinchero, D., & Fossa, M. (2024). Smart sensors and smart data for precision agriculture: A review. *Sensors*, 24(8), 2647. <https://doi.org/10.3390/s24082647>

Sudarmaji, A., Gunawan, B., Nugraha, F., Arini, N., & Mulyani, S. (2024). Smart soil moisture control based on IoT ESP32 for horticulture cultivation in coastal area. *BIO Web of Conferences*, 96, 04001. <https://doi.org/10.1051/bioconf/20249604001>

Afzal, M., Saeed, I. A., Sohail, M. N., Saad, M. H. M., & Sarker, M. R. (2025). IoT-enabled adaptive watering system with ARIMA-based soil moisture prediction for smart agriculture. *IEEE Access*, 13, 27714-27728. <https://doi.org/10.1109/ACCESS.2025.3532447>

Islam, M. B., Guerrieri, A., Gravina, R., Delaney, D. T., & Fortino, G. (2025). From traditional machine learning to fine-tuning large language models: A review for

sensors-based soil moisture forecasting. Sensors, 25(22), 6903.
<https://doi.org/10.3390/s25226903>

Soussi, A., Zero, E., Sacile, R., Trincherro, D., & Fossa, M. (2024). Smart sensors and smart data for precision agriculture: A review. Sensors, 24(8), 2647.
<https://doi.org/10.3390/s24082647>

Afzal, M., Saeed, I. A., Sohail, M. N., Saad, M. H. M., & Sarker, M. R. (2025). IoT-enabled adaptive watering system with ARIMA-based soil moisture prediction for smart agriculture. IEEE Access, 13, 27714-27728.
<https://doi.org/10.1109/ACCESS.2025.3532447>

Islam, M. B., Guerrieri, A., Gravina, R., Delaney, D. T., & Fortino, G. (2025). From traditional machine learning to fine-tuning large language models: A review for sensors-based soil moisture forecasting. Sensors, 25(22), 6903.
<https://doi.org/10.3390/s25226903>

Soussi, A., Zero, E., Sacile, R., Trincherro, D., & Fossa, M. (2024). Smart sensors and smart data for precision agriculture: A review. Sensors, 24(8), 2647.
<https://doi.org/10.3390/s24082647>

Islam, M. B., Guerrieri, A., Gravina, R., Delaney, D. T., & Fortino, G. (2025). From traditional machine learning to fine-tuning large language models: A review for sensors-based soil moisture forecasting. Sensors, 25(22), 6903.
<https://doi.org/10.3390/s25226903>

Islam, M. B., Guerrieri, A., Gravina, R., Delaney, D. T., & Fortino, G. (2025). From traditional machine learning to fine-tuning large language models: A review for sensors-based soil moisture forecasting. Sensors, 25(22), 6903.
<https://doi.org/10.3390/s25226903>

Afzal, M., Saeed, I. A., Sohail, M. N., Saad, M. H. M., & Sarker, M. R. (2025). IoT-enabled adaptive watering system with ARIMA-based soil moisture prediction for smart agriculture. IEEE Access, 13, 27714-27728. <https://doi.org/10.1109/ACCESS.2025.3532447>

Afzal, M., Saeed, I. A., Sohail, M. N., Saad, M. H. M., & Sarker, M. R. (2025). IoT-enabled adaptive watering system with ARIMA-based soil moisture prediction for smart agriculture. IEEE Access, 13, 27714-27728. <https://doi.org/10.1109/ACCESS.2025.3532447>

Islam, M. B., Guerrieri, A., Gravina, R., Delaney, D. T., & Fortino, G. (2025). From traditional machine learning to fine-tuning large language models: A review for sensors-based soil moisture forecasting. Sensors, 25(22), 6903. <https://doi.org/10.3390/s25226903>

Afzal, M., Saeed, I. A., Sohail, M. N., Saad, M. H. M., & Sarker, M. R. (2025). IoT-enabled adaptive watering system with ARIMA-based soil moisture prediction for smart agriculture. IEEE Access, 13, 27714-27728. <https://doi.org/10.1109/ACCESS.2025.3532447>

Soussi, A., Zero, E., Sacile, R., Trincherro, D., & Fossa, M. (2024). Smart sensors and smart data for precision agriculture: A review. Sensors, 24(8), 2647. <https://doi.org/10.3390/s24082647>

9. ANEXOS

9.1. Anexo A Código de programación del microcontrolador

Contiene el programa desarrollado para el ESP32, encargado de leer los sensores, sincronizar la hora y transmitir las lecturas por Wi-Fi hacia la plataforma en la nube. El repositorio se encuentra disponible en:

https://github.com/CardenasFabi/monitoreo_Cafetal

9.2. Anexo B. Código de procesamiento de los datos.

Reúne el cuaderno de Google Colab en el que se realiza el proceso de extracción, transformación y carga de las dos fuentes de información, los datos de los sensores y la precipitación del IDEAM, hasta conformar el conjunto de datos integrado. Disponible en:

<https://colab.research.google.com/drive/1t6JSB5PzmfCQtAc-xYkgV1L9qSLjdwzr?usp=sharing>

9.3. Anexo C Código de desarrollo de los modelos predictivos.

Corresponde al cuaderno de Google Colab en el que se construyen, entrenan, validan y comparan los modelos de estimación de la humedad del suelo. Disponible en:

https://colab.research.google.com/drive/1FDBdRz7n8SQ_EXzxLIs2FizuAU_YbALO5?usp=sharing

F-DC-125

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

VERSIÓN: 2.0

ELABORADO POR:
Docencia

REVISADO POR:
Sistema Integrado de Gestión

APROBADO POR: Líder del Sistema Integrado de Gestión
FECHA APROBACIÓN: Octubre de 2023