

Desarrollo de un sistema de información para la gestión remota de una red inalámbrica de sensores aplicado en cultivos de café

Autor:

Cristian Santiago Rincón Caicedo

Universidad de Cundinamarca

Facultad de Ingeniería

Ingeniería Electrónica

Fusagasugá, Colombia

2023

Desarrollo de un sistema de información para la gestión remota de una red inalámbrica de sensores aplicado en cultivos de café

Trabajo de grado presentado como requisito parcial para optar por el título de ingeniero electrónico

Autor:
Cristian Santiago Rincón Caicedo

Director:
Edgar Hernando Criollo Velásquez

Co-director:
Andres Felipe Guerrero Guerrero

Línea de investigación:
Software, sistemas emergentes y nuevas tecnologías

Universidad de Cundinamarca
Facultad de Ingeniería
Ingeniería Electrónica
Fusagasugá, Colombia
2023

Dedicatoria

Dedicado este trabajo con todo mi amor y gratitud a mis padres y mi hermana, quienes han sido un pilar y un apoyo incondicional en cada paso de mi vida. Su constante amor, paciencia y sacrificio han hecho posible la culminación de mi carrera y cumpliendo uno de mis mayores sueños. A mi familia, por su cariño y aliento en cada etapa de mi vida y de mi carrera. Este logro no solo es mío, sino de todos ustedes, quienes han sido una gran inspiración. Les agradezco infinitamente por ser mi fuente de fortaleza y por nunca dejarme desfallecer. Les amo con todo mi corazón.

Agradecimientos

Quiero expresar mi más profundo agradecimiento a todas las personas que hicieron posible la realización de mi proyecto de tesis de pregrado.

En primer lugar, agradezco a mis padres, Herardo Arturo Rincón y Elizabeth Caido Rodríguez, por su amor incondicional, su apoyo y su constante aliento en mi educación. Gracias por su sacrificio y por brindarme la oportunidad de obtener una educación de calidad. Sin su apoyo emocional y financiero, no habría podido alcanzar este logro.

También quiero agradecer a mis directores, Andres Felipe Guerrero y Edgar Hernando Criollo, por su orientación, paciencia y sabiduría en la enseñanza. Sus consejos y sugerencias me permitieron mejorar mi trabajo y alcanzar mis objetivos. Agradezco su dedicación y compromiso en mi formación académica.

Agradezco a mis padrinos, Flor Alba Rincón y Eliecer Ospina, por su valioso apoyo a lo largo de mi formación académica. Sus comentarios, consejos y sugerencias fueron de gran ayuda para culminar este proceso de formación como profesional. Gracias por su tiempo y disposición para brindarme su orientación y apoyo.

Agradezco a toda mi familia, los cuales de una u otra forma me han brindado un apoyo emocional y un aliento constante. Gracias por estar ahí en todo momento, por sus palabras de aliento y por creer en mí.

Para finalizar, agradezco a todas las personas que de alguna manera contribuyeron a mi éxito académico. A todos ustedes, gracias por su tiempo, dedicación y apoyo en mi formación como estudiante, como persona y como profesional. Su contribución ha sido fundamental en mi carrera universitaria.

Resumen

Los agricultores a menudo no tienen en cuenta factores ambientales como el suelo, el agua, el clima o el ecosistema, sin saber que la interacción de estos puede llegar a afectar negativamente el desarrollo del grano, lo que resulta en una disminución del tamaño y la calidad. Por lo tanto, es crucial comprender la evolución de las diferentes variables climáticas en un cultivo para implementar estrategias de manejo efectivas que minimicen el impacto ambiental y tengan una baja repercusión en el rendimiento y la calidad del cultivo.

El presente trabajo de grado tiene como objetivo desarrollar un sistema de adquisición y gestión remota de datos para una red inalámbrica de sensores que recopila periódicamente datos sobre las principales variables climáticas que influyen en la respuesta fisiológica de las plantas de café. El desarrollo del proyecto busca facilitar el análisis en tiempo real para la toma de decisiones informadas por parte de los expertos, y así, contribuir a la mejora de la producción y calidad final del grano de café. Lo anterior, se logra mediante el diseño e implementación de una interfaz de gráfica desplegada en la nube conectada al nodo central de la red inalámbrica de sensores a través de la plataforma de Google Firebase. La implementación de este proyecto permite recopilar información detallada acerca de las diferentes variables climáticas y en función de estas, brindar una herramienta de ayuda para la toma de decisiones por parte de los expertos. Además, proporciona una alternativa para la gestión y monitoreo remoto de las redes inalámbricas de sensores.

Con base a los resultados obtenidos en las pruebas de funcionamiento se concluye que el sistema web permite a los usuarios acceder a la información sobre el estado de un cultivo en tiempo real desde cualquier dispositivo conectado a Internet. Dicho sistema cuenta con una interfaz de usuario intuitiva y fácil de usar, mostrando información sobre variables climáticas a través de tablas de datos y gráficas interactivas que se actualizan en tiempo real. El sistema administra roles de usuario para proteger la privacidad de los datos y garantizar la seguridad de la red de sensores. Por último, permite la configuración remota de parámetros de la red inalámbrica de sensores para optimizar el proceso de análisis de datos.

Palabras clave: Cultivos de café, red inalámbrica de sensores, sistemas de información, internet de las cosas, tecnologías 4.0.

Abstract

Farmers often do not take into account environmental factors such as soil, water, climate or ecosystem, not knowing that the interaction of these can negatively affect grain development, resulting in a decrease in size and quality. Therefore, it is crucial to understand the evolution of different climatic variables in a crop in order to implement effective management strategies that minimize environmental impact and have a low impact on crop yield and quality.

The objective of this degree project is to develop a remote data acquisition and management system for a wireless sensor network that periodically collects data on the main climatic variables that influence the physiological response of coffee plants. The development of the project seeks to facilitate real-time analysis for informed decision making by experts, and thus contribute to the improvement of the production and final quality of the coffee bean. This is achieved through the design and implementation of a graphical interface deployed in the cloud connected to the central node of the wireless sensor network through the Google Firebase platform. The implementation of this project allows the collection of detailed information about the different climatic variables and based on these, provide a decision support tool for decision making by experts. In addition, it provides an alternative for the remote management and monitoring of wireless sensor networks.

Based on the results obtained in the operational tests, it is concluded that the web system allows users to access information on the status of a crop in real time from any device connected to the Internet. The system has an intuitive and easy-to-use user interface, displaying information on climatic variables through data tables and interactive graphs that are updated in real time. The system manages user roles to protect data privacy and ensure the security of the sensor network. Finally, it allows remote configuration of wireless sensor network parameters to optimize the data analysis process.

Keywords: Coffee crops, wireless sensor network, information systems, internet of things, 4.0 technologies.

Contenido

	pág.
Resumen	5
Abstract	6
INTRODUCCIÓN	11
PLANTEAMIENTO DEL PROBLEMA	12
OBJETIVOS DEL ESTUDIO	13
Objetivo general	13
Objetivos específicos	13
JUSTIFICACIÓN	13
ALCANCES Y LIMITACIONES	14
Alcances	14
Limitaciones	15
1. MARCO REFERENCIAL	16
1.1. ESTADO DEL ARTE	16
1.2. FUNDAMENTOS TEÓRICOS	22
1.2.1. Internet de las cosas	22
1.2.2. Redes inalámbricas de sensores	22
1.2.3. Nodo central o Gateway	22
1.2.4. ECMAScript 2020	22
1.2.5. Responsive Design o diseño responsivo	23
1.2.6. Framework	23
1.2.7. API (Application Programming Interface)	23
1.2.8. Experiencia de usuario (UX)	23
1.2.9. Interfaz de usuario (UI)	24
2. DISEÑO METODOLÓGICO	25
3. DESARROLLO DEL SISTEMA DE GESTIÓN Y ADQUISICIÓN DE DATOS	27
3.1. ANÁLISIS	27
3.1.1. Necesidades del sistema	27
3.1.2. Compatibilidad con diferentes dispositivos	28
3.1.3. Autenticación y almacenamiento de datos en la nube	32
3.1.4. Fácil interacción y visualización de los datos	34
3.1.5. Discusión	35
3.2. DISEÑO	37
3.2.1. Requerimientos basados en el estándar IEEE 830	37
3.2.2. Modelamiento de la base de datos	43
3.2.3. Estructura general del sistema	46

3.2.4. Diseño de la interfaz y experiencia de usuario (UI/UX)	47
3.3. IMPLEMENTACIÓN	51
3.3.1. Herramientas y tecnologías utilizadas	52
3.3.2. Módulos del sistema	55
3.4. PRUEBAS DE FUNCIONAMIENTO	63
3.4.1. Pruebas del diseño responsivo	63
3.4.2. Pruebas del módulo Home Page	64
3.4.3. Pruebas del módulo de registro	65
3.4.4. Pruebas del módulo de autenticación	67
3.4.5. Pruebas del módulo de monitoreo	70
3.4.6. Pruebas del módulo de configuraciones	73
3.4.7. Pruebas con el nodo central	75
4. ANÁLISIS DE RESULTADOS Y DISCUSIÓN	78
5. CONCLUSIONES Y TRABAJOS FUTUROS	80
5.1. CONCLUSIONES	80
5.2. PERSPECTIVAS DE TRABAJO FUTURO	82
BIBLIOGRAFÍA	84

Índice de figuras

Figura 1.	Pasos para el desarrollo del proyecto marco.	12
Figura 2.	Visualización de datos a través de la plataforma Ubidots [20]. .	17
Figura 3.	Visualización de datos a través de la plataforma Sentilio [26]. .	18
Figura 4.	Framework propuesto en [31].	20
Figura 5.	Fases para el desarrollo del proyecto.	25
Figura 6.	Funcionalidad del sistema de adquisición y gestión de datos. .	38
Figura 7.	Estructura para la colección de medidas de las variables. . . .	43
Figura 8.	Estructura para la colección de los nodos.	44
Figura 9.	Estructura para la colección de configuración de parámetros. .	45
Figura 10.	Estructura de funcionamiento del sistema.	46
Figura 11.	Imágenes para generar la paleta de colores.	48
Figura 12.	Paleta de colores usada en la interfaz gráfica.	48
Figura 13.	Tipografías seleccionadas para la página web.	49
Figura 14.	Alerta emergente informativa.	50
Figura 15.	Layouts de la página web.	51
Figura 16.	Estructura de la aplicación creada por medio del CLI de angular.	52
Figura 17.	Diagrama general del sistema de gestión remota.	55
Figura 18.	Módulo “Home Page”.	56
Figura 19.	Secciones habilitadas para usuarios sin sesión activa.	56
Figura 20.	Módulo de registro.	57
Figura 21.	Ejemplo de notificación de error en el módulo de registro. . . .	58
Figura 22.	Mensaje de verificación de correo electrónico.	58
Figura 23.	Módulo de autenticación.	59
Figura 24.	Sección para recuperación de contraseña.	60
Figura 25.	Módulo de configuraciones.	61
Figura 26.	Módulo de monitoreo.	62
Figura 27.	Vista del módulo de monitoreo desde un dispositivo móvil . . .	64
Figura 28.	Estructura del correo para la verificación de cuenta.	67
Figura 29.	Notificación tras haber ingresado un correo no registrado. . . .	69
Figura 30.	Tarjetas que muestran la última medida registrada.	71
Figura 31.	Tabla de datos que muestran los registros de la base de datos.	71
Figura 32.	Formulario para el filtrado por fecha del módulo de monitoreo. .	72
Figura 33.	Envío de datos hacia la base de datos desde la Raspberry Pi.	75
Figura 34.	Ejecucion remota de scripts en la Raspberry Pi.	76

Índice de tablas

Tabla 1.	Ventajas y desventajas de los frameworks Angular y React. . . .	29
Tabla 2.	Comparación entre diferentes servicios de hosting.	32
Tabla 3.	Comparación de los servicios de base de datos y autenticación.	33
Tabla 4.	Herramientas, servicios y parámetros de diseño para la aplicación.	36
Tabla 5.	Resultados de las pruebas en diferentes resoluciones de pantalla.	63
Tabla 6.	Pruebas de navegación del módulo “Home Page”.	65
Tabla 7.	Pruebas de validación de formularios del módulo de registro. . .	66
Tabla 8.	Prueba de envío del link de verificación del correo electrónico. .	67
Tabla 9.	Pruebas de validación de formularios del módulo de autenticación.	68
Tabla 10.	Pruebas del formulario para la recuperación de contraseña. . .	69
Tabla 11.	Prueba de envío del link de recuperación de contraseña.	69
Tabla 12.	Pruebas de validación del formulario para el filtrado por fecha. .	72
Tabla 13.	Pruebas a la gráfica interactiva al enviar diferentes tipos de datos.	73
Tabla 14.	Pruebas de formularios del módulo de configuraciones.	74

INTRODUCCIÓN

Este trabajo de grado se desarrolla en el contexto del proyecto marco de investigación denominado “Sistema de monitoreo de variables ambientales a través de tecnologías 4.0 en cultivos de café en la provincia del Sumapaz (Cundinamarca-Colombia)”, el cual pretende utilizar las tecnologías de nueva generación en el desarrollo de un prototipo para monitorización de variables ambientales como la radiación solar, dióxido de carbono (CO_2), temperatura y humedad relativa del suelo y el ambiente en cultivos de café. Su propósito es el fortalecimiento de la capacidad productiva y calidad del café para tener la posibilidad de ingresar al mercado nacional y/o internacional con una marca institucional. Así mismo, este proyecto es una alternativa para que la Universidad de Cundinamarca se proyecte como centro de desarrollo del cultivo de café, con enfoque en las necesidades de los productores del departamento.

Por otra parte, debido al desconocimiento de la interacción de las plantas con el medio ambiente, los factores como el suelo, el agua, el clima o el mismo ecosistema muchas veces son ignorados. Estos factores forman un conjunto del cual depende la producción y calidad final del café. La interacción de estos factores sin supervisión afecta el desarrollo del grano, disminuyendo su tamaño y aspectos organolépticos, con lo cual se disminuye la posibilidad de comercialización [1]. Sumado a ello, la ausencia de control de estos factores incrementa la probabilidad de saturación o falta de componentes vitales, provocando enfermedades que se traducen en disminución de producción y una desventaja a la hora de comercializar el producto. Por esta razón, es importante hacer uso de herramientas de base tecnológica, que permitan monitorear el estado de las variables que impactan considerablemente en la comercialización nacional e internacional [2]. En este sentido, con el desarrollo de este proyecto se espera impactar de manera positiva en los cultivos de café para así lograr un incremento en la producción y calidad de los granos. Adicionalmente, se pretende incentivar el uso de tecnologías 4.0 en zonas rurales en diferentes cultivos de la región. Teniendo en cuenta lo anterior, el proyecto marco de investigación se desarrolla en tres fases, como se muestra en la Figura 1.

Conocer el proceso de evolución de las diferentes variables climáticas presentes en un cultivo es indispensable para llevar a cabo estrategias de manejo efectivas, que permitan causar el menor daño posible al ambiente y que repercutan en lo mínimo al rendimiento y calidad del cultivo [3]. Actualmente existen diferentes dispositivos que permiten medir estas variables. La información recolectada es interpretada por un software y posteriormente comprendida por el funcionario del cultivo. La evolución de herramientas para el monitoreo de cultivos se ha centrado principalmente en el uso de pantallas sensibles al tacto con la capacidad de mostrar una gran cantidad de información en tiempo real acerca del cultivo, lo anterior con el fin de mejorar la simplicidad en el manejo por parte del operario [4]. Uno de los principales inconvenientes de implementar estos sistemas es que pueden representar un alto costo

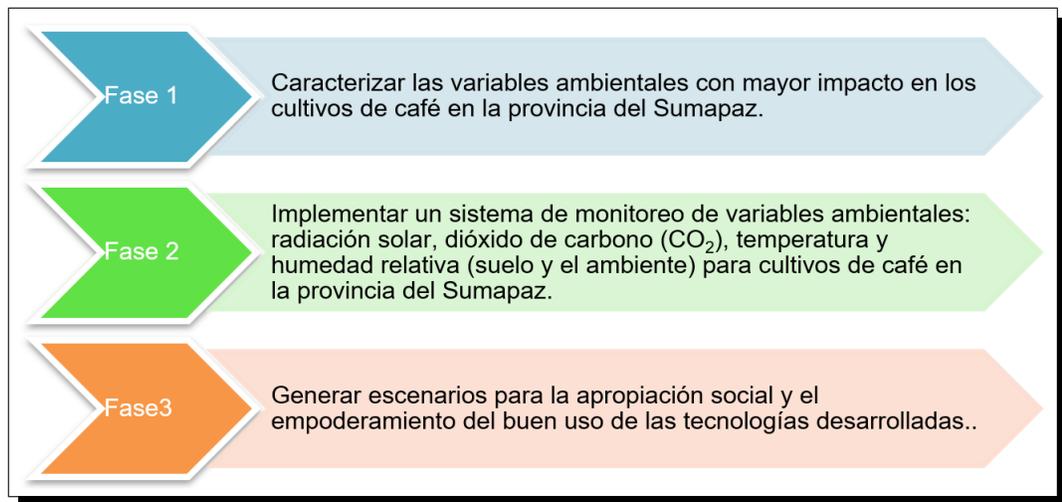


Figura 1. Pasos para el desarrollo del proyecto marco.

monetario, y a su vez, dichos sistemas no permiten un monitoreo remoto, por ende, es necesaria la presencia del operario para conocer el estado actual del cultivo.

PLANTEAMIENTO DEL PROBLEMA

La producción de café en Colombia se ve afectada por variables climáticas que tienen impacto en el desarrollo de las plantas, esto se debe en gran parte a que los cultivos de café en áreas rurales cuentan con un factor tecnológico casi nulo. Por esta razón es primordial incentivar la implementación de elementos electrónicos que faciliten las tareas y permitan monitorear remotamente estas variables, de la misma forma, prevenir problemas que disminuyan la calidad de los granos como: enfermedades en las plantas, saturación o falta de componentes vitales para estas, y otras que signifiquen una desventaja a la hora de comercializar el producto [5].

El desconocimiento de la interacción que tienen las plantas con el medio ambiente es una de las principales causas de pérdidas para los agroempresarios, estos muchas veces ignoran que el suelo, el agua, el clima y el ecosistema forman un conjunto del cual depende la producción y su calidad. Existen sensores que se enfocan en tareas específicas, de esta manera el nivel freático de la tierra, la concentración de vapor en el aire y otras variables pueden ser analizadas con precisión, permitiendo analizar de una forma la respuesta sintomatológica y fisiológica de las plantas de café frente al devenir de dichas variables [6].

A nivel nacional solo las grandes empresas productoras de café cuentan con la posibilidad de mejorar la infraestructura de sus cultivos [7], implementando dispositivos electrónicos que muchas veces pueden realizar el trabajo de una persona. Siendo el factor humano el que generalmente se encarga de revisar periódicamente los cultivos, aparte de realizar otras labores como riego, abonado y recolección, se genera

así, un entorno en el que sea probable pasar por alto detalles que estos dispositivos estarían dispuestos a medir cuando les sea ordenado [8].

Teniendo en cuenta lo anterior, surge la necesidad de integrar un sistema de gestión que permita a los caficultores interactuar con estos dispositivos de una manera sencilla, configurando los parámetros de medición según las necesidades particulares, y además, ver datos en tiempo real o un historial de las diferentes mediciones tomadas, para así, en función de estas se tomen las decisiones pertinentes para un correcto desarrollo del cultivo.

Con base en lo anteriormente mencionado surge el siguiente interrogante: “¿Cómo generar a partir de la información centralizada de una red inalámbrica de sensores una herramienta interactiva de apoyo en la supervisión de variables edafoclimáticas de un cultivo de café?”

OBJETIVOS DEL ESTUDIO

Objetivo general

- Desarrollar un sistema de adquisición y gestión de datos en una red inalámbrica de sensores para la supervisión de los factores que influyen en la respuesta fisiológica de las plantas de café.

Objetivos específicos

- Diseñar una interfaz gráfica para la visualización de datos y configuración de parámetros de los nodos en una red inalámbrica de sensores a partir de requerimientos del usuario final y las características de las variables a supervisar.
- Implementar un sistema de supervisión y gestión remota a partir de la integración de herramientas compatibles que permitan la configuración de parámetros de los nodos en una red inalámbrica de sensores.
- Verificar la funcionalidad del sistema de supervisión e interacción mediante escenarios de prueba y casos de aplicación.

JUSTIFICACIÓN

De acuerdo con la opinión de expertos, las variables que pueden afectar los rendimientos y calidad de la producción de café son:

La radiación solar como respuesta al proceso fotosintético el cual se expresa en el crecimiento y desarrollo metabólico de la planta [9], [10]. La humedad del suelo como relación planta nutrientes en solución acuosa que pueden suplir las necesidades

nutricionales del cultivo [11], [12]. La humedad relativa del ambiente como aquella variable que responde al estrés causado por cambios de temperatura en el ambiente que pueden afectar el metabolismo de la planta afectando la productividad debido a la probabilidad de aumento de presencia de agentes patógenos [9].

Por otra parte, se requiere evaluar la capacidad de conversión de CO₂ y la temperatura ambiente para correlacionar con aspectos de cambio climático [13], [14]. Debido a lo anterior, realizar seguimiento y control a las variables mencionadas de un sistema productivo, permite proponer procesos de seguimiento y control preventivo en búsqueda de la reducción de costos por las acciones correctivas que se hacen en los cultivos, contrarrestando la presencia de plagas, enfermedades o afectaciones localizadas, generadas por diferentes factores, acciones que se pueden evitar haciendo uso de los análisis de datos y la tecnología denominada Agricultura 4.0 [15]. En la actualidad la Unidad Agroambiental La Esperanza de la Universidad de Cundinamarca cuenta con un área agrícola de 2.3 hectáreas dedicada al cultivo de café, de las cuales 1.3 hectáreas, se encuentran en su tercer año de producción. En este momento, no se cuenta con un sistema de monitoreo adecuado que garantice la producción y los rendimientos del producto final obtenido de la cosecha de café (C.P.S, por sus siglas), lo que en consecuencia genera afecciones en el grano como hongos, baja uniformidad en el secado, entre otras. Los factores mencionados anteriormente afectan la calidad del producto, limitando las posibilidades de mejoramiento del sistema productivo, reduciendo la ganancia y afectando la comercialización en mercados locales, restringiendo las posibilidades de ser reconocidos en el gremio del sector cafetero. Adicionalmente, la implementación de prácticas de monitoreo contribuye a la generación de registros en los procesos de producción como evidencias de seguimiento y control para la obtención y renovación de la certificación de Buenas Prácticas Agrícolas (BPA, por sus siglas), de acuerdo con las resoluciones 030021 de 2017 [16] y su actualización 082394 de 2020 [17].

Con el desarrollo e implementación de un sistema para la gestión y adquisición de datos se busca permitir a los caficultores monitorear de forma remota las variables ambientales que afectan el crecimiento y calidad de los cultivos, permitiendo la toma de decisiones informadas en tiempo real. También se busca mejorar la eficiencia y la rentabilidad de la producción de cultivos por medio del monitoreo y análisis de los datos recopilados por los sensores. Dicha información permite a los agricultores optimizar la cantidad y frecuencia de los recursos utilizados, como el agua y los fertilizantes, lo que puede reducir costos y aumentar la producción.

ALCANCES Y LIMITACIONES

Alcances

El sistema de gestión y adquisición de datos tratado en este documento se define como una interfaz de software que recopila, procesa, almacena y analiza diferentes datos de variables climáticas en tiempo real provenientes desde una red inalámbrica.

ca de sensores. Este sistema permite a los usuarios obtener una visualización del comportamiento de las variables en el cultivo, lo que a su vez, les permite tomar decisiones informadas en cuanto a la gestión de los procesos y la optimización del rendimiento. El sistema permite la gestión remota de los nodos, los umbrales de alerta o el periodo de muestreo de cada nodo.

El sistema denominado “Coffee GUI” permite monitorear factores ambientales relevantes para el cultivo del café como la humedad del suelo, la temperatura, la humedad relativa y la concentración de CO₂. Con el monitoreo en tiempo real y la recopilación de información en una base de datos centralizada en la nube, el sistema proporciona una herramienta valiosa para los caficultores a la hora de tomar decisiones para la administración del cultivo. Además, el sistema facilita la recopilación de datos históricos, lo que permite a los productores analizarlos y mediante estudios estadísticos tomar decisiones informadas sobre la producción futura. Finalmente, el sistema permite la gestión remota de la red inalámbrica de sensores instalada en el cultivo, lo cual significa que no es necesaria la presencia del caficultor en el predio para modificar periodos de muestreo, umbrales de alerta o el estado (encendido o apagado) de alguno de los nodos.

Limitaciones

El sistema para la gestión de información remota denominado “*Coffee GUI*” cuenta con algunas limitaciones de las que se establecen a continuación. El sistema debe ser utilizado con una conexión estable a internet para asegurar una buena experiencia de usuario en los tiempos de carga de los diferentes módulos. Se debe hacer uso de un navegador web que soporte el estándar de JavaScript ECMA Script 2020 (ES11). Se debe acceder por medio del dominio público de internet “<https://app-coffee-gui.web.app>” o “<https://app-coffee-gui.firebaseio.com>”. El funcionamiento del sistema presenta una dependencia directa de la red de sensores instalada en el cultivo de café, por ende, la visualización y configuración de parámetros estarán sujetos al correcto funcionamiento de la misma.

1. MARCO REFERENCIAL

En esta sección se observarán algunos proyectos realizados recientemente con el fin de obtener un vistazo general de las soluciones propuestas, además, se define un conjunto de conceptos que facilitarán la lectura y comprensión del presente documento.

1.1. ESTADO DEL ARTE

En este apartado se brinda un panorama general con respecto a los avances realizados en la temática de monitoreo, control remoto y visualización de datos en diferentes áreas de la agricultura, los siguientes proyectos se usarán como guía para dar cumplimiento a los objetivos planteados anteriormente.

La agricultura moderna debe utilizar las tecnologías y herramientas derivadas de las investigaciones y adelantos científicos para mitigar los grandes desafíos relacionados con la construcción de un futuro sostenible, dificultado con el aumento poblacional, la urbanización, la contaminación ambiental, la migración y el cambio climático. Hay una amplia gama de aplicaciones posibles: gestión de maquinarias, vigilancia del ganado, piscicultura, cuidado forestal, cultivos urbanos, entre otras. Todas las tecnologías implicadas giran en torno al concepto del *Internet de las Cosas* (IoT por sus siglas en inglés Internet of Things) y ayudan a los agricultores a través de sistemas de apoyo a la toma de decisiones [18]. Uno de los elementos fundamentales para una aplicación de IoT es la visualización de los datos, debido a que permite la interacción del usuario con el entorno. Para que un usuario se beneficie por completo de la revolución de IoT, se necesita crear una visualización atractiva y fácil de entender. Esto también permitirá a los formuladores de políticas convertir los datos en conocimiento, lo cual es fundamental para la toma rápida de decisiones [19].

En los trabajos presentados en [20] y [21], se diseñó e implementó una red de sensores para el monitoreo de cultivos de alfalfa y monitoreo continuo de variables ambientales respectivamente. El proceso de visualización de los datos se realizó a través de la plataforma IoT Ubidots de manera remota y en tiempo real. Una de las ventajas que ofrece esta plataforma es el soporte de varios protocolos de comunicación como HTTP, MQTT o TCP/UDP [22], la plataforma accedía a través del protocolo HTTP [20] y MQTT [21] a un servidor creado de manera local, obtenía los datos y los mostraba por pantalla. Para el almacenamiento de la información proveniente de los diferentes nodos sensores, en ambos casos se usó un sistema de almacenamiento basado en MySQL de manera local por medio de un ordenador

de placa simple (Raspberry). En la Figura 2 se presenta la interfaz gráfica de usuario para el sistema propuesto en [20], donde se observa el comportamiento de las variables de temperatura, humedad del aire y humedad del suelo visualizadas en gráficas en función del tiempo. Del mismo modo, existen widgets que muestran el promedio de las variables, los parámetros de configuración, un reloj en tiempo real y una imagen del cultivo.

Por otra parte, en los artículos [23] y [24], se realizó una propuesta similar de monitoreo para casas de cultivo (invernaderos) y una piscicultura de truchas en el año 2020. En este caso, para la visualización de los datos se adoptó el uso de la plataforma ThingsBoard la cual proporciona de manera nativa una base de datos en PostgreSQL que se empleó para el almacenamiento de los datos recolectados por los sensores desplegados en campo. La transmisión de información se realizó a través del protocolo MQTT debido a que dicho protocolo para redes IoT presenta menos carga en comparación con HTTP [25].

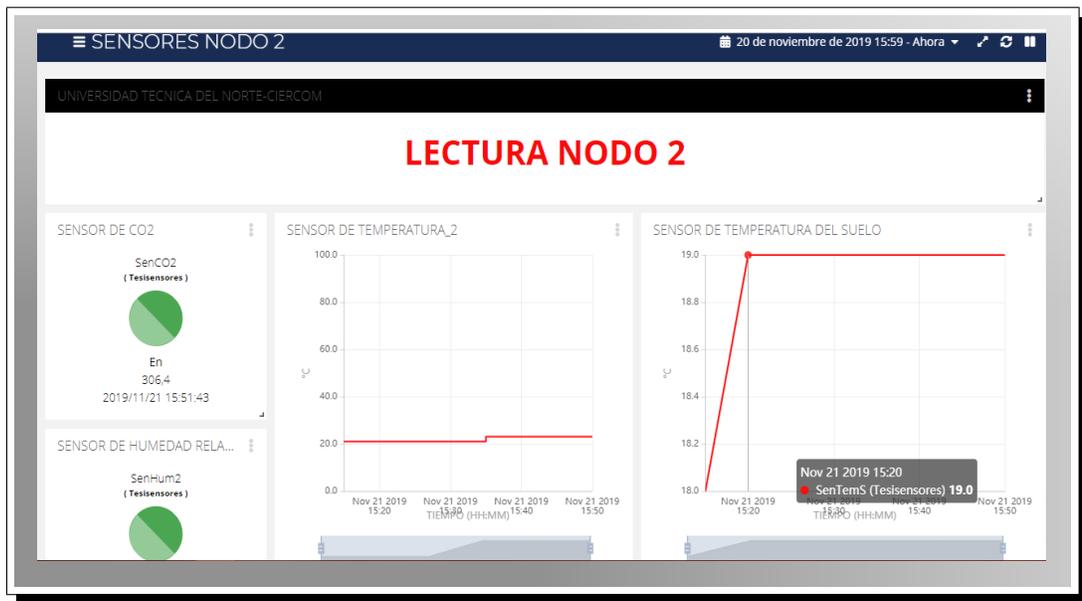


Figura 2. Visualización de datos a través de la plataforma Ubidots [20].

Por limitaciones de conectividad a internet, en el artículo presentado en [23] se implementó el entorno de monitoreo de manera local, es decir, no era posible acceder a la información fuera de la red de área local (LAN, por sus siglas en inglés Local Network Area). Por otra parte, en el trabajo presentado en [24] se visualizaron los datos directamente en campo mediante una pantalla de cristal líquido (LCD, por sus siglas en inglés Liquid Crystal Display) conectada al microcontrolador mediante el protocolo UART y, en cuanto a la visualización remota, se empleó la plataforma ThingsBoard.

En los cuatro proyectos descritos anteriormente se observa el uso común de las plataformas Ubidots y ThingsBoard, aunque en el mercado actual existen muchas más alternativas Open Source como Sentilio, Thinger.io, Grafana o alternativas de pago como ThingSpeak cada una con diversas características, ventajas, desventajas y capacidades de adaptarse a los requerimientos del usuario. A continuación, se describen algunos proyectos que usaron dichas alternativas.

Las propuestas para el monitoreo de variables ambientales en cultivos verticales y de tomate cherry realizadas en [26] y [27], ocuparon las plataformas Sentilio y Thinger.io respectivamente. Ambas alternativas optaron por el uso de bases de datos en un entorno remoto, es decir, todos los datos recolectados y procesados eran enviados directamente hacia un servidor donde posteriormente la plataforma IoT correspondiente debía acceder a ellos por medio del protocolo HTTP. El despliegue de la plataforma Sentilio presentado en [26] se realizó mediante una máquina virtual (proporcionada por la página web oficial de Sentilio) instalada en un servidor Linux perteneciente al grupo de investigación SISTEMIC, ubicado en la sede central de la Universidad de Antioquia, Colombia. Con dicho servidor desplegado fue posible publicar las medidas registradas por los sensores con el método POST del protocolo HTTP, para realizar dicha publicación el Gateway debía enviar una petición hacia la URL “<http://sistemic.udea.edu.co:4000/data/Cultivo1>” junto con la información de los sensores en el “body” del mensaje. Además, al usar el protocolo HTTP fue posible utilizar los diferentes métodos que este ofrece como PUT, DELETE o GET para actualizar, eliminar u obtener datos respectivamente. En función a los datos enviados por el Gateway, la plataforma Sentilio permitió mostrar la información por pantalla como se observa en la Figura 3, o si, por el contrario, desde la plataforma Sentilio se generaba una petición para realizar un cambio en el estado de los relés o periodos de muestreo, desde el Gateway se realizaba el envío de la información a los nodos sensores correspondientes.

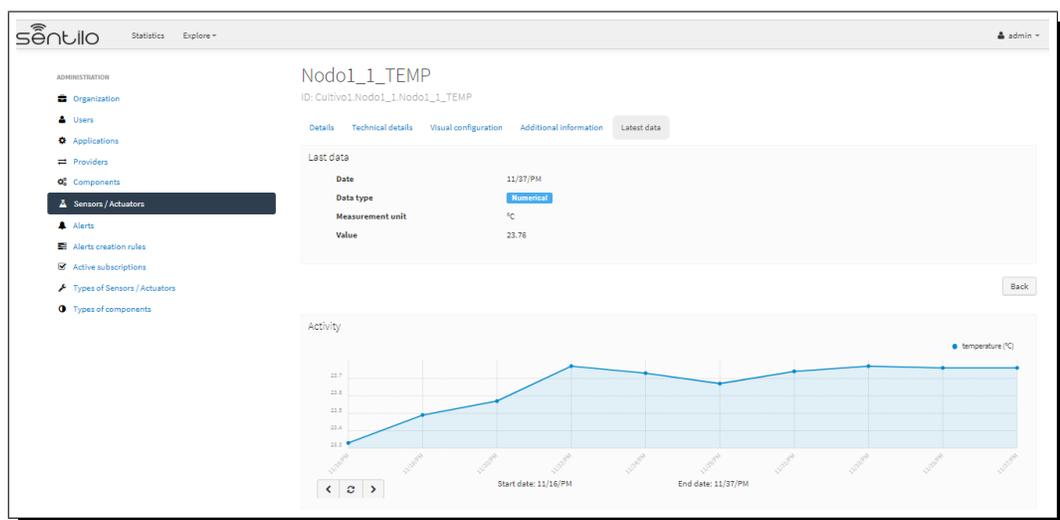


Figura 3. Visualización de datos a través de la plataforma Sentilio [26].

El sistema de monitoreo en un laboratorio de experimentación de cannabis usando tecnologías de IoT presentado en [28] hace empleo de la plataforma Grafana, y a diferencia de las propuestas anteriormente expuestas, hace uso de dos bases de datos, una en un entorno local y otra en la nube a través de Google Cloud. Para almacenar la información en las bases de datos se creó una tabla en MySQL y se publicaron los datos desde el nodo central o Gateway (ESP8266) por medio del protocolo MQTT. Es decir, el Gateway guardaba los datos del sensor, se conectaba a un servidor en la nube y enviaba a través de MQTT un mensaje con la información recolectada. Además, dicho mensaje se almacenaba paralelamente al servidor local. Lo anterior permitió contar con un respaldo de los datos en caso de presentar algún incidente en alguna de las dos bases. Por último, se creó un conjunto de alertas que informaban al usuario sobre un comportamiento atípico en el cultivo (medidas fuera de los rangos establecidos para las diferentes variables), el usuario podía acceder a un dashboard en tiempo real para la visualización y control de las variables ambientales a medir por medio de Grafana.

Finalmente, la plataforma ThingSpeak fue utilizada en el prototipo de monitoreo en cultivos de arroz presentado en [29]. El diseño del prototipo usó como base la plataforma IoT ThingSpeak para la visualización de datos enviados por los sensores, esta plataforma permite recopilar, observar y analizar flujos de información en tiempo real en la nube, además, proporciona visualizaciones instantáneas de los datos enviados [30]. Dicha plataforma cuenta con una interfaz de programación de aplicaciones (API, por sus siglas en inglés application programming interface) para almacenar y recuperar datos vía HTTP, sobre Internet o sobre una LAN. Además, una característica diferencial con respecto a las otras plataformas es la integración con software de programación y cálculo numérico Matlab, que permite crear diferentes algoritmos para aumentar el rendimiento en los cultivos o crear algunos modelos de predicción en función a las lecturas registradas por los sensores. Con dicha plataforma se logró visualizar en tiempo real a través del dashboard los datos enviados por los sensores cada 5 minutos de una manera gráfica y amigable para el usuario. Adicionalmente, se implementaron alertas mediante el uso de correos electrónicos y tweets; también se logró el control remoto para el encendido y apagado de una bomba de agua mediante el protocolo HTTP.

Hasta el momento todas las soluciones vistas presentan un común denominador y es la visualización por medio de la misma plataforma IoT. Lo anterior permite concluir algunas desventajas significativas cuando se presenta alguno de los siguientes casos:

- La plataforma no permite manipular los elementos gráficos de la interfaz, es decir, entrega un panel por defecto y no es posible de editarlo.
- La plataforma restringe algunas funciones de personalización de la interfaz gráfica para el usuario final en versiones gratuitas.

- La plataforma no es lo suficientemente intuitiva para que una persona, la cual no esté familiarizada con la manipulación de dispositivos electrónicos, pueda navegar libremente en ella.

Partiendo de la anterior premisa, tener la libertad de diseñar desde cero todo el apartado gráfico con el que va a interactuar el usuario día a día para abastecerse de información sobre su cultivo y tomar decisiones en función de dicha información representa un valor agregado. A continuación, se presentan tres soluciones para la representación de información en cultivos que no hacen uso de ninguna plataforma IoT o la usan como intermediaria para posteriormente mostrar los datos en una aplicación diseñada desde cero y ajustada al usuario final, para que este tenga una rápida curva de aprendizaje.

En el artículo presentado en [31], se realizó el diseño e implementación de un framework para el monitoreo de temperatura en tiempo real completo, configurable, escalable y de bajo costo aplicando la tecnología IoT. La arquitectura de red propuesta fue la mostrada en la Figura 4, la cual cuenta con tres capas.

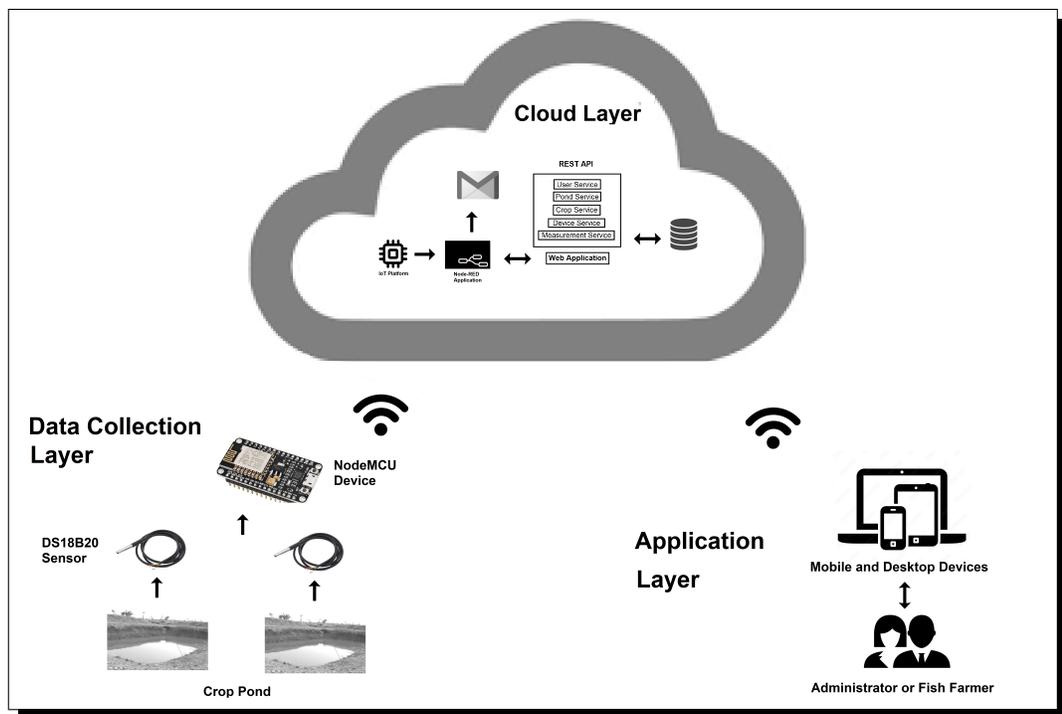


Figura 4. Framework propuesto en [31].

- La “*capa recolectora de datos*” que permite recolectar los datos del cultivo y enviarlos al microcontrolador nodeMCU, mediante el cual se recibe la data y se envía a la capa de nube a través de su módulo WIFI ESP8266 hacia el servicio IoT Platform de IBM Cloud.

- La “*capa Nube*” que hace uso del servicio de IoT Platform de IBM Cloud, el uso de Heroku Cloud para el alojamiento de la base de datos en MySQL y del RESTful Api, uso del servicio Firebase de Google Cloud para el alojamiento de la aplicación web desarrollada con el framework de JavaScript Angular. Esta capa recibe la data enviada por la capa anterior y mediante el uso de un aplicativo NodeRed alojado en IBM Cloud realiza el almacenamiento de los datos leídos en la base de datos en la nube a través del uso del RESTful Api.
- La “*capa de aplicación*” incluye el uso de un navegador web por medio de un dispositivo móvil o un equipo de escritorio donde los usuarios y el administrador del cultivo pueden usar el aplicativo y acceder a módulos de la solución de acuerdo al perfil que tenga.

Como resultado se obtuvo un aplicativo completamente responsivo, que permitió al usuario y/o al administrador acceder y consultar su contenido desde diferentes dispositivos, según sea necesario.

Por último, los trabajos presentados en [32] y [33], implementaron un sistema de monitoreo remoto sin hacer uso de ninguna plataforma IoT. El proyecto se desarrolló a través de la integración de una red de sensores ubicada en el sitio de estudio, una base de datos en la nube (Firebase en [33]) y el diseño de una aplicación web y móvil respectivamente para la realización de labores de administración o supervisión. La red de sensores envía los datos procesados hacia un Gateway, el cual accede y escribe dichos datos por medio del protocolo HTTP en la base de datos alojada en la nube. Una vez dispuestos los datos en la nube, las aplicaciones diseñadas acceden a ellos igualmente por HTTP y muestran los datos de una manera adecuada para el usuario. Cabe resaltar que en el trabajo presentado en [33] se logró manipular el estado de actuadores de manera remota.

En general, los proyectos anteriormente descritos como resultado final obtuvieron un sistema automático de monitoreo y supervisión de parámetros ambientales básicos que logra una reducción significativa del tiempo empleado para monitorear y analizar el estado de los cultivos en comparación con la forma manual. Los sistemas permitieron una correcta toma de decisiones para la mejora en la calidad de la producción y una visualización de los datos a través de una interfaz gráfica amigable y de fácil manejo para evaluar el desempeño de las variables con el paso del tiempo.

Una vez observadas las diferentes soluciones actuales en el mercado junto con sus respectivas limitantes, es posible concluir que para lograr el cumplimiento de los objetivos planteados inicialmente se debe hacer uso de una plataforma que actúe como intermediaria entre la comunicación del nodo central de la red de sensores y una base de datos alojada en la nube. Lo anterior con el objetivo de acceder a esta por medio de una aplicación web 100% responsive, esto permite eliminar la limitante del sistema operativo o dispositivo que emplee el usuario final.

1.2. FUNDAMENTOS TEÓRICOS

En este apartado se presentan algunas definiciones y conceptos que facilitan la lectura e interpretación del presente documento.

1.2.1. Internet de las cosas

El internet de las cosas (IoT) permite la comunicación multidireccional entre máquinas, personas y productos, facilitando la toma de decisiones basadas en tecnologías de la información recogidas del entorno. Utiliza nuevos sensores y actuadores combinados con análisis de big data y computación en la nube para habilitar máquinas autónomas y sistemas inteligentes. El Internet de las cosas es una tecnología clave para que la industria manufacturera cree productos inteligentes (incorpore servicios en los productos), cree relaciones más estrechas con los consumidores finales y capture información sobre el rendimiento y el uso de sus productos, incluso si son propietarios del cliente [34].

1.2.2. Redes inalámbricas de sensores

Una red de sensores es una infraestructura que consta de elementos de procesamiento, medición y comunicación de datos que permiten a los administradores implementar, observar y responder a eventos y fenómenos en un entorno determinado. Las redes inalámbricas de sensores son una clase de sistemas distribuidos que son una parte integral del espacio físico en el que residen. A diferencia de la mayoría de las computadoras, que trabajan principalmente con datos generados por humanos, las redes de sensores juzgan el estado del mundo que las encarna [35].

1.2.3. Nodo central o Gateway

El nodo central o Gateway es un dispositivo que utiliza una conexión IP estándar para transmitir información a un servidor web central ubicado en la nube. El nodo central es un puente transparente entre los dispositivos finales y los servidores de una red inalámbrica de sensores, además, permite la interconexión de objetos inteligentes sin una instalación compleja, brindando flexibilidad a los usuarios finales, desarrolladores y empresas que necesitan implementar redes IoT [36].

1.2.4. ECMAScript 2020

ECMAScript es un lenguaje de secuencias de comandos estandarizado por Ecma International en las especificaciones ECMA-262 e ISO/IEC 16262. ECMAScript se basa en varias tecnologías fuente diferentes, las más populares de las cuales son JavaScript (Netscape) y JScript (Microsoft). ECMAScript 2020 es la undécima edición de la especificación del lenguaje ECMAScript [37], [38].

1.2.5. Responsive Design o diseño responsivo

La palabra *responsive* en el diseño web hace referencia a la técnica de crear sitios web que se ajustan automáticamente al tamaño de pantalla de un dispositivo, ya sea un ordenador, una tableta o un teléfono móvil. Esto permite que un sitio web sea fácilmente legible y navegable en cualquier dispositivo sin necesidad de crear una versión separada para cada uno. El objetivo es ofrecer una experiencia de usuario consistente y satisfactoria en cualquier dispositivo que se utilice para acceder a un sitio web [39].

1.2.6. Framework

En el desarrollo de software, un framework es una estructura de soporte definida sobre la cual se pueden organizar y desarrollar otros proyectos de software. En programación, es un conjunto de funciones o código común que realiza tareas habituales y frecuentes (crear objetos, conectarse a bases de datos, etc.) en todo tipo de aplicaciones. Proporciona una base confiable para el desarrollo de aplicaciones especiales, le permite ignorar la parte de desarrollo más trivial y general del desarrollo, así como los tipos abiertos de diseñadores y programadores y pasar más tiempo en las necesidades de software [40].

1.2.7. API (Application Programming Interface)

Una API, o interfaz de programación de aplicaciones, es una solución que ofrece una representación simplificada de un problema y establece las pautas para que los clientes interactúen con los componentes de software que lo resuelven. Estos componentes, en general, se presentan como una biblioteca de software que puede utilizarse en diversas aplicaciones. En términos generales, las API definen componentes reutilizables que permiten integrar módulos de funcionalidad en diferentes aplicaciones para el usuario final [41].

1.2.8. Experiencia de usuario (UX)

El proceso de diseño de la experiencia del usuario (UX) es un conjunto progresivo de actividades que tienen como objetivo ayudar tanto al equipo de desarrollo como a las partes interesadas en el proyecto a enfocarse en brindar una experiencia satisfactoria a los usuarios del producto de software. A diferencia del diseño de la interfaz de usuario o la ingeniería de usabilidad, el diseño de la UX es un proceso más amplio que debe iniciarse en las primeras etapas del ciclo de vida del proyecto para ser eficaz. Si los desarrolladores esperan hasta el final del proyecto para agregar la funcionalidad de la interfaz de usuario, es poco probable que brinden una experiencia agradable a los usuarios [42].

1.2.9. Interfaz de usuario (UI)

La interfaz de usuario (UI) es el medio a través del cual los usuarios interactúan con un software o una aplicación. Se trata de un conjunto de elementos de diseño visual, como botones, iconos, menús, campos de entrada y salidas de información, que permiten a los usuarios realizar acciones en el software y recibir información de salida. La UI es una parte esencial del diseño de la experiencia del usuario (UX) y debe ser fácil de usar, accesible y atractiva visualmente para brindar una experiencia satisfactoria al usuario. A diferencia del diseño de la UX, que abarca todo el proceso de diseño centrado en el usuario, el diseño de la UI se enfoca principalmente en la apariencia y la funcionalidad visual del software [42].

2. DISEÑO METODOLÓGICO

En esta sección se describen las herramientas, estrategias, técnicas y procedimientos requeridos para satisfacer el objetivo principal del proyecto y llegar a una solución de la problemática. El desarrollo de esta se asume tratando individualmente cada objetivo específico detallando las actividades necesarias para su total cumplimiento. Las fases para el desarrollo del proyecto se observan en la Figura 5.



Figura 5. Fases para el desarrollo del proyecto.

En la fase inicial se definió la plataforma donde se llevaría a cabo el diseño de la interfaz gráfica, considerando opciones tales como escritorio, web, Android, iOS u otras. A continuación, se revisaron diferentes lenguajes de programación que podrían ser utilizados, de acuerdo con el entorno establecido. Luego, se profundizó en los diferentes conceptos y herramientas disponibles del lenguaje elegido para desarrollar la primera versión. Por otra parte, se establecieron los requisitos y funcionalidades específicas que se incluyen en la interfaz gráfica, lo que permitió satisfacer las necesidades de los usuarios y asegurarse de que se cumpla el cometido principal, la adquisición y gestión de datos en una red inalámbrica de sensores. Para concluir esta etapa se realizó el modelamiento de la base de datos, es decir, se definió la información a guardar y las relaciones que existan entre ellas.

Una vez diseñada la parte gráfica (estética de la aplicación) en la segunda fase se realizó la vinculación del nodo central de la red y el sistema con un servidor común en la nube para la lectura de datos y configuración de los diferentes parámetros de la red inalámbrica de sensores a través de internet.

Finalmente, en la última fase se plantearon diversas pruebas de lectura y manipulación de los diferentes nodos de la red en un entorno local, con el fin de identificar los fallos (bugs) presentados, esto con el objetivo de realizar la oportuna corrección de hardware o software (según sea el caso) para así, obtener un correcto funcionamiento de la red. Con las pruebas locales superadas se procedió a realizar una

prueba final de toda la red de sensores de manera remota para verificar su correcto funcionamiento.

Una vez terminadas cada una de las fases anteriormente mencionadas se desarrolló un manual de usuario y una cartilla de implementación para todo el sistema completo con el fin de capacitar a la comunidad externa y académica interesada.

3. DESARROLLO DEL SISTEMA DE GESTIÓN Y ADQUISICIÓN DE DATOS

Este capítulo presenta el proceso detallado del análisis, diseño, implementación y pruebas de funcionamiento del software. Se incluyen detalles sobre la estructura y el desarrollo de los diferentes componentes del sistema y su integración para formar una solución de gestión remota completa.

3.1. ANÁLISIS

En este apartado se presenta una descripción detallada de las condiciones generales que debe satisfacer el sistema de adquisición y gestión de datos para cumplir con cada uno de los objetivos previamente establecidos. Se detallan los procesos de análisis y selección de los recursos necesarios para la implementación del sistema, así como las consideraciones y decisiones tomadas para garantizar la fiabilidad de los datos adquiridos y gestionados.

3.1.1. Necesidades del sistema

Según el Centro Nacional de Investigaciones de Café (Cenicafé) la productividad del cultivo de café depende de la interacción de diversos factores de la producción, entre los cuales se resaltan factores genéticos, de manejo y ambientales [43]. Con respecto a este último, las variables climatológicas como la radiación solar, temperatura, humedad y precipitación tienen un impacto directo y heterogéneo sobre la productividad y rentabilidad de los sistemas de producción, y por ende sobre el bienestar de los agricultores [44]. Desde el punto de vista biofísico, los sistemas de producción de café en Colombia son vulnerables al déficit y el exceso hídrico, a la reducción o aumento de la radiación solar, a la reducción o aumento de la temperatura del aire, al viento y el granizo, a las pérdidas de suelo por erosión y movimientos en masa y de nutrientes por percolación y escorrentía [45]. Teniendo en cuenta lo anterior, y que desde la creación de la Federación Nacional de Cafeteros en 1927, se ha generado un interés en conocer el comportamiento climático en los cultivos de café [46], el monitoreo meteorológico permite conocer la disponibilidad y los cambios en los elementos climáticos de mayor interés, y a partir de ellos, obtener información para la toma de decisiones [47].

A partir de las necesidades identificadas, el sistema de gestión y adquisición de datos para la supervisión de variables en un cultivo de café, cumple los siguientes requisitos:

- **Monitoreo de variables ambientales:** el sistema debe recopilar y monitorear datos ambientales críticos como la temperatura, humedad, radiación solar y concentración de CO₂, considerando que estas variables influyen en el crecimiento y la salud de las plantas de café.
- **Interfaz de usuario:** el sistema debe contar con una interfaz gráfica de usuario (GUI) multiplataforma intuitiva y de fácil uso que permita a los caficultores o usuarios interesados supervisar constantemente el estado de las variables climáticas.
- **Almacenamiento de datos:** el sistema debe almacenar de manera confiable y segura los diferentes registros de las medidas, de tal manera que puedan ser consultados en cualquier momento.
- **Acceso remoto:** el sistema debe permitir acceso a los datos desde cualquier parte del mundo con una conexión estable a internet para una mayor comodidad del usuario. Además, debe permitir configurar los parámetros de la red inalámbrica de sensores de manera remota con el fin de eliminar la necesidad de desplazarse físicamente a cada ubicación de monitoreo, ahorrando tiempo y recursos.
- **Escalabilidad:** el sistema debe ser escalable para permitir la expansión a medida que las necesidades del usuario o del cultivo cambien, como por ejemplo, la adición de una nueva variable climática o una nueva funcionalidad en el sistema.

A continuación, se analiza las diferentes alternativas para el desarrollo de una aplicación como la descrita con anterioridad.

3.1.2. Compatibilidad con diferentes dispositivos

En la actualidad existen diferentes dispositivos de uso personal que permiten la visualización de información, como por ejemplo teléfonos móviles, tabletas, ordenadores de escritorio o portátiles. Cada uno de estos cuenta con un sistema operativo diferente como Android, iOS, Windows o Linux. Lo anterior supone un reto debido a que si se desea desarrollar una aplicación compatible con los diferentes sistemas sería necesaria la creación de diferentes proyectos de la misma aplicación en diferentes lenguajes de programación como Java, C, Swift o Python.

Desarrollo de una aplicación multiplataforma o web: con el objetivo de evitar el desarrollo de la misma aplicación para diferentes plataformas, hoy día existen diferentes frameworks para la construcción de aplicaciones multiplataforma como NativeScript, React Native, Flutter o Xamarin. El trabajo presentado en [48] muestra algunas de las ventajas y desventajas que presenta cada framework y concluye que estos son solo una herramienta cuyo objetivo principal es reducir tiempo, recursos

e incluso código, sin embargo, no reemplazan a las tecnologías que permiten un desarrollo completamente nativo. Teniendo en cuenta lo anterior, optar por el desarrollo de una aplicación 100 % web podría ser una excelente alternativa. En el libro de programación de [49] se muestran algunas de las principales ventajas que tiene el uso de una aplicación web, estas son:

- **Accesibilidad:** una aplicación web se puede acceder desde cualquier dispositivo con un navegador web, mientras que una aplicación multiplataforma requiere una descarga e instalación específicas para cada plataforma.
- **Facilidad de actualización:** las actualizaciones de una aplicación web se realizan automáticamente en el servidor donde estas son alojadas, mientras que las actualizaciones de una aplicación multiplataforma deben ser descargadas e instaladas por cada usuario.
- **Menor tamaño de instalación:** una aplicación web no requiere descarga ni instalación, por lo que su tamaño será menor que el de una aplicación multiplataforma.
- **Evita la fragmentación del mercado:** en el mercado Android, en particular, tiene una gran variedad de dispositivos y versiones de sistema operativo, lo que hace que sea difícil asegurar la compatibilidad de una aplicación multiplataforma con todos los dispositivos.

Hoy por hoy, al igual que existen diferentes frameworks para el desarrollo de aplicaciones multiplataforma como los mencionados anteriormente, existen frameworks 100 % dedicados al desarrollo web, siendo React y Angular los más populares y usados actualmente en el mercado [39]. En la Tabla 1 se muestran las principales ventajas y desventajas de cada framework, dicha información se determinó en función de las características presentadas en los artículos [39] y [50].

Tabla 1. Ventajas y desventajas de los frameworks Angular y React.

Framework	Angular	React
Ventajas	<ul style="list-style-type: none"> • Enfoque basado en componentes para el desarrollo de aplicaciones web. • Gran cantidad de funciones y características integradas para facilitar el desarrollo de aplicaciones complejas. 	<ul style="list-style-type: none"> • Permite crear componentes de interfaz de usuario reutilizables. • Ofrece un gran rendimiento y velocidad.
Desventajas	<ul style="list-style-type: none"> • El código generado puede ser menos optimizado en comparación con otras opciones. 	<ul style="list-style-type: none"> • No proporciona una gran cantidad de funciones y características integradas como Angular.

Cabe resaltar que ambas alternativas pueden ser más difíciles de aprender para nuevos desarrolladores en comparación con otros frameworks. En contraparte, tanto Angular como React tienen una gran comunidad y una gran cantidad de recursos de aprendizaje disponibles, además, de ofrecer una excelente integración con otras herramientas y tecnologías.

Con base en la experiencia de desarrollo en ambos frameworks y la información del libro presentado en [51] permiten determinar dos características diferenciales de Angular con respecto a React que ayudan en la etapa de desarrollo de una aplicación.

Angular maneja una estructura de directorios más jerarquizada y organizada, donde se separan los componentes, servicios, módulos y recursos de la aplicación en diferentes directorios. Además, permite el manejo independiente de diferentes ficheros HTML, CSS y TS, lo cual permite no mezclar lenguajes en un mismo fichero. Lo anterior, hace que sea más fácil organizar, entender y mantener la aplicación a medida que el proyecto crece en tamaño y complejidad.

Angular también trabaja de manera nativa con TypeScript (TypeScript es un superconjunto de JavaScript que agrega características de programación orientada a objetos y de tipado estático al lenguaje), lo cual permite aprovechar las características avanzadas del mismo, como decoradores y tipado fuerte de variables. Lo anterior, ayuda a los desarrolladores a estructurar el código de manera más robusta y confiable. Algunos de los beneficios que se obtienen al utilizar TypeScript son:

- **Mejora la productividad del desarrollador:** proporciona un mejor soporte para el tipado de variables y una mayor capacidad de detección de errores en tiempo de desarrollo, lo que ayuda a los desarrolladores a escribir código más robusto y confiable.
- **Mayor escalabilidad:** es más adecuado para proyectos grandes y complejos, proporciona una mayor organización y estructura para el código.
- **Mejora la documentación:** al tener un soporte de tipado fuerte, es más fácil entender el código y el propósito de cada variable o función.
- **Mayor compatibilidad con otras herramientas y librerías:** es compatible con una amplia variedad de herramientas y librerías de JavaScript, lo que facilita la integración con otras tecnologías.
- **Mayor seguridad:** al tener un soporte de tipado fuerte, se reduce el riesgo de errores que pueden causar comportamientos inesperados en el sistema, ya que se detectan en el desarrollo.

Diseño responsivo o responsive design: otro factor importante a tener en cuenta a la hora de pretender desarrollar una aplicación web es que esta pueda ser renderizada de una manera correcta desde cualquier dispositivo, lo que en el diseño web se conoce como *“Responsive design”* o diseño responsivo traducido al español.

La manera más sencilla de conseguir un diseño responsivo, es hacer uso de algún framework de HTML, CSS y JavaScript, siendo Bootstrap el más popular actualmente para el desarrollo de sitios web responsivos y amigables para dispositivos móviles [52]. Bootstrap es un framework de código abierto para desarrollo de sitios web y aplicaciones móviles que fue desarrollado por Twitter y es ampliamente utilizado para crear diseños responsive. Este framework proporciona una gran cantidad de componentes pre-construidos como menús desplegados, botones, formularios, tablas, grillas, alertas y otros elementos comunes de la interfaz de usuario [53]. También incluye un sistema de rejilla (grid system) flexible que permite crear diseños fácilmente adaptativos, algunas de las principales razones por las cuales usar Bootstrap se presentan en el artículo [52].

- **Compatibilidad:** es compatible con los navegadores web más populares.
- **Fácil de comenzar:** con solo el conocimiento de HTML y CSS, cualquiera puede comenzar con Bootstrap. Además, el sitio oficial de Bootstrap tiene una buena documentación.
- **Código abierto:** al ser un framework de código abierto permite el aporte de la comunidad para su mejora.

Servicio de hosting: finalmente, una vez abarcados los temas de desarrollo web, surge el último aspecto a tener en cuenta para un completo despliegue de una aplicación web. Como se mencionó anteriormente, las aplicaciones web se actualizan automáticamente en el servidor donde se alojan, es decir, es necesario contar con un servicio de hosting, el cual permite alojar los archivos HTML, CSS y Javascript que serán renderizados por los navegadores de los usuarios cuando accedan a ellos a través de Internet por medio de un dominio público, ya sea otorgado por la plataforma de hosting o uno propio en caso de contar con este.

El proveedor de alojamiento web y dominios web Hostinger define el servicio de hosting de la siguiente manera: “Un hosting es un servicio online que hace que se pueda acceder a tu sitio web en Internet. Cuando obtienes un hosting, básicamente alquilas un espacio en un servidor que almacena todos los archivos y datos de tu sitio web para que funcione correctamente [54]”.

Algunos de los servicios de hosting más populares y gratuitos (hasta cierto punto) se muestran en la Tabla 2, allí se aprecian diferentes características en sus planes gratuitos, donde el símbolo (✓) indica que el servicio SI cuenta con esa característica, y las celdas vacías indican que NO cuenta con dicha característica. La información presentada se basa de los sitios webs oficiales de cada servicio [55], [56], [57], [58] y [59].

Tabla 2. Comparación entre diferentes servicios de hosting.

	GitHub	Heroku	AWS	Firebase	Netlify
Fácil de usar y configurar	✓	✓		✓	✓
Integración nativa con GitHub	✓	✓	✓		✓
Límite de tráfico		✓	✓	✓	✓
Límite de almacenamiento		✓	✓	✓	✓
Soporte de webs dinámicas		✓	✓	✓	✓
Escalabilidad automática		✓	✓	✓	✓
Soporte para diferentes lenguajes de programación		✓	✓	✓	✓
Cargos por uso excesivo		✓	✓	✓	✓
Integración nativa con herramientas de Google				✓	
Límite de peticiones o descargas	∞	18M/mes	20k/mes	10GB/mes	100k/día
Servicio de base de datos y autenticación nativo		✓	✓	✓	

3.1.3. Autenticación y almacenamiento de datos en la nube

Para lograr visualizar y gestionar los diferentes datos generados por la red inalámbrica de sensores es necesario el despliegue de una base de datos que permita almacenar las diferentes lecturas tomadas a lo largo del día por los nodos sensores y llevar un histórico de las mismas. La base de datos podría alojarse en un servidor local en el lugar donde se implemente la red de sensores, con algunas limitaciones de acceso. Lo anterior puede solventarse a partir de una base de datos desplegada en la nube, obteniendo las siguientes ventajas [60]:

- **Escalabilidad:** permite escalar fácilmente el almacenamiento y el procesamiento de datos a medida que el número de nodos sensores o la cantidad de datos recolectados aumenta.

- **Accesibilidad:** permite acceder a los datos desde cualquier lugar con una conexión a internet, lo que facilita la monitorización y el análisis de los datos en tiempo real.
- **Fiabilidad:** ofrece alta disponibilidad y redundancia, lo que garantiza que los datos estarán disponibles incluso si uno de los nodos sensores falla.
- **Seguridad:** proporciona medidas de seguridad avanzadas, como la encriptación de datos y la autenticación de usuarios, para proteger algunos datos sensibles.
- **Integración:** permite integrar fácilmente con otras herramientas de análisis y visualización de datos, lo que permite obtener información valiosa a partir de los datos recolectados por los nodos sensores.

Por otro lado, la autenticación de usuarios es un proceso esencial para acceder a cualquier página web. Este proceso otorga una serie de privilegios y opciones que, de lo contrario, no estarían disponibles para el usuario [61]. La integración de un servicio de autenticación ayuda a que solo los usuarios administradores del cultivo sean capaces de modificar el comportamiento de los diferentes nodos, o de eliminar los registros guardados en la base de datos. Teniendo en cuenta los aspectos anteriormente vistos, se puede llegar a la conclusión de que la integración de estos dos servicios dentro del desarrollo de la aplicación es de vital importancia, y con el objetivo de mantener todos los servicios (hosting, base de datos y autenticación) unificados en un solo ecosistema, en la Tabla 3 se analizan las ventajas y desventajas de las soluciones ofrecidas por Heroku, Firebase y AWS. La información presentada se basa de los sitios webs oficiales de cada servicio [57], [58] y [59].

Tabla 3. Comparación de los servicios de base de datos y autenticación de usuarios de diferentes plataformas.

	Firestore	Heroku	AWS
Ventajas	<ul style="list-style-type: none"> ● Interfaz de usuario amigable y fácil de usar. ● Autenticación de dos factores y la verificación de contraseñas débiles. 	<ul style="list-style-type: none"> ● Compatible con una variedad de lenguajes de programación. 	<ul style="list-style-type: none"> ● Personalizable e integrable con otras herramientas de Amazon. ● Autenticación de dos factores y multifactor (MFA).
Desventajas	<ul style="list-style-type: none"> ● Menos personalizable que otras opciones de bases de datos. 	<ul style="list-style-type: none"> ● Menos control sobre el sistema operativo. ● No cuenta con seguridad avanzada. 	<ul style="list-style-type: none"> ● Más difícil de configurar y utilizar que otras opciones.

Adicionalmente, a las características presentadas, un punto importante a resaltar es la gran escalabilidad y rendimiento que ofrecen los tres servicios. Aspectos como la facilidad de configuración, planes gratuitos para pequeños proyectos, autenticación por medio de correo electrónico y contraseña, teléfono, cuentas de Google, Facebook, Twitter, Github y más son compartidos por Firebase y Heroku. En contraparte, el plan gratuito en cuanto a almacenamiento y número de solicitudes se ve bastante limitado por los tres servicios si se desea escalar o trabajar en un proyecto más grande. Un aspecto crucial en cuanto a Firebase es que suele ser más sencillo de integrar con Angular o React en comparación con las otras propuestas presentadas con anterioridad, esto debido a la gran cantidad de recursos y documentación disponibles, así como la facilidad de configuración. En Angular, se encuentra disponible *AngularFire*, una biblioteca oficial de Firebase para Angular que proporciona un conjunto de módulos, los cuales integran una API para interactuar con Firebase y así conseguir una integración fácil y rápida [62]. Por su parte, en React se puede utilizar la biblioteca oficial de Firebase para React denominada *ReactFire*, que proporciona igualmente una API para interactuar con Firebase [63].

3.1.4. Fácil interacción y visualización de los datos

Finalmente, uno de los aspectos más importantes para garantizar una buena experiencia de usuario, es la correcta y fácil visualización de la información, los temas tratados hasta el momento resultan totalmente indiferentes para el usuario final. El artículo presentado en [64] concluye que un buen diseño de página web debe satisfacer parámetros básicos como el color, contraste, organización del texto, selección de fuentes, tamaño adecuado de los elementos, entre otros aspectos que son claves para crear un producto atractivo para el usuario. A continuación, se profundiza un poco más en los diferentes aspectos a tener en cuenta, estos se basan en [64] y [65].

Usabilidad intuitiva: en relación a la facilidad de uso, la navegación, los menús y las funciones de la página web deben ser claras y comprensibles, utilizando etiquetas y botones fácilmente identificables y descriptivos.

Diseño responsivo: permite que una página se ajuste automáticamente a la pantalla del dispositivo en uso, lo que garantiza una experiencia de usuario satisfactoria.

Jerarquía clara de la información: la jerarquía es importante para asegurar que la información se presente de manera organizada y fácil de entender. Los encabezados y los subtítulos deben ser claros y diferenciarse visualmente del resto del contenido, las imágenes y los gráficos deben estar claramente etiquetados y relacionados con el texto circundante.

Carga rápida: la velocidad de carga es esencial para la experiencia de usuario. Las páginas web deben cargar rápidamente para evitar que los usuarios se aburran o se

frustren con el uso de estas. La optimización de la página para la velocidad de carga incluye la compresión de imágenes, la optimización del código, entre otras cosas. El desarrollo de un sitio web diseñado como *aplicación de una sola página* (SPA por sus siglas en inglés Single Page Application) puede mejorar la velocidad de carga en ciertas circunstancias. Las SPA's cargan los datos y componentes en una sola página, y mantienen la información en la memoria del navegador, lo que significa que no es necesario volver a cargar la página completa cada vez que el usuario realiza una acción. Esto puede resultar en una experiencia de usuario más fluida y rápida.

Fuentes legibles: las fuentes deben ser claras y legibles para mejorar la experiencia de lectura de los usuarios, evitando fuentes muy pequeñas o con caligrafía complicada.

Colores contrastantes: el uso de colores contrastantes ayuda a mejorar la legibilidad de la página y a destacar la información importante. Algunos aspectos importantes a tener en cuenta son:

- **Contraste:** es importante elegir una combinación de colores que brinde un contraste adecuado entre el fondo y el texto, para garantizar una buena legibilidad. El contraste también debe ser suficiente entre los diferentes elementos de la página para que la información sea clara y fácil de identificar.
- **Colores:** una paleta de colores limitada ayuda a mantener una sensación coherente y unificada en la página. Se recomienda elegir entre 2 y 3 colores principales y uno o dos colores secundarios para los detalles. Es importante resaltar que no se debe escoger una paleta de colores muy clara debido a que causa fatiga visual, o una paleta de colores muy oscura, esto debido a que dificultará la visualización de la información para algunos usuarios.

Interacción y retroalimentación clara: la interacción con una página web debe ser clara y fácil de entender. Los usuarios deben recibir retroalimentación clara sobre el estado actual de la página y estrategias de interacción, por ejemplo, a través de pantallas de carga, animaciones o mensajes emergentes de confirmación, alerta o error.

3.1.5. Discusión

Tomando en cuenta cada uno de los requisitos generales anteriormente vistos, este proyecto opta por el uso de las herramientas y servicios listados en la Tabla 4.

Tabla 4. Herramientas, servicios y parámetros de diseño escogidos para el desarrollo de la aplicación.

Requerimiento	Elección
Tipo de aplicación	100 % Web
Framework de desarrollo JavaScript	Angular
Servicio de hosting	Firebase
Servicio de base de datos	
Servicio de autenticación	
Framework de desarrollo CSS	Bootstrap
Parámetros de diseño	• Usabilidad intuitiva
	• Diseño responsive
	• Jerarquía clara de la información
	• Fuentes legibles
	• Carga rápida
	• Colores contrastantes
	• Interacción y retroalimentación clara

Finalmente, se especifica cada uno de los parámetros de diseño que cumplirá la aplicación final.

- **Home page o página de inicio:** este apartado brindará información y un panorama general del alcance que tendrán los usuarios con el uso de la aplicación.
- **Formularios de registro e inicio de sesión:** estos formularios permiten que los usuarios realicen el registro e inicio de sesión de su cuenta personal, este deberá incluir un correo electrónico activo (el cual se usará para enviar un link de verificación en el momento del registro) y una contraseña con una longitud mayor a 6 caracteres. Además, se incluirá la opción de inicio de sesión con una cuenta de Google para una mayor comodidad del usuario. Al integrar el servicio de autenticación de Firebase, las opciones de verificación en dos pasos o recuperación de contraseña estarán contempladas de igual manera.
- **Panel de información:** sección de la aplicación que brindará información sobre el cultivo con datos como su extensión, tipo de cultivo, archivos multimedia (noticias, videos, fotos, etc.), ubicación, entre otros.
- **Panel de monitoreo y configuración:** este apartado permitirá visualizar todas las medidas registradas por medio de tablas y gráficas interactivas, filtrar los datos por una fecha de interés, exportar los datos en formato .xlsx (Excel) para realizar análisis en caso de ser requerido por el usuario, eliminar los registros guardados y acceder a las configuraciones de los nodos sensores ubicados en

el cultivo para su configuración remota. Las anteriores acciones estarán habilitadas para los usuarios autenticados (usuarios registrados), además, según el rol asignado (invitado o administrador) podrá realizar o no ciertas acciones.

3.2. DISEÑO

Tomando en cuenta el análisis realizado, y la conclusión a la cual se llegó en la sección 3.1.5 a continuación, se describirán los requerimientos del sistema basados en el estándar IEEE 830 tomando como eje central el usuario final y el desarrollo del sistema por módulos, se realiza el modelamiento de la base de datos y se muestra la estructura general con la que cuenta el sistema.

3.2.1. Requerimientos basados en el estándar IEEE 830

En este apartado se describe detalladamente la Especificación de Requisitos de Software (ERS) para el desarrollo de un sistema de adquisición y gestión de datos en una red inalámbrica de sensores para la supervisión de variables climáticas. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998 [66].

Propósito: la presente sección tiene como propósito definir las especificaciones funcionales y no funcionales para el desarrollo de un sistema de gestión y adquisición de datos.

Definiciones, acrónimos y abreviaturas: lista de acrónimos y definiciones para la correcta interpretación de la presente sección.

- **RF:** requisito funcional
- **RFN:** requisito no funcional
- **Add-on:** subcategoría de los plugins, es decir, pequeños programas que amplían la funcionalidad del navegador.

Perspectiva del producto: el sistema de adquisición y gestión de datos está diseñado para trabajar en entornos 100 % web, lo que permite aprovechar las ventajas mencionadas en la sección 3.1.2. Adicionalmente, el sistema cuenta con un instructivo de uso descargable para usuarios nuevos, una interfaz de usuario amigable, un diseño totalmente responsive, servicios de autenticación y diferentes paneles de información, monitoreo y configuración de parámetros para los diferentes nodos de la red inalámbrica de sensores que permite una gestión eficiente del sistema y una toma de decisiones basadas en datos precisos y actualizados en tiempo real. Todos estos elementos en conjunto hacen del sistema de adquisición y gestión de datos

una herramienta completa, accesible y fácil de usar para agricultores, docentes y estudiantes interesados.

Funcionalidad del producto: la Figura 6 presenta una visión general de la funcionalidad del producto y los diferentes actores que interactúan con el sistema, así como los permisos que tienen asignados en función de su rol. Cada una de las acciones está resaltada con un color que relaciona la manera de como llega cada uno de los usuarios a realizar dichas las acciones como consecuencia de la modificación de algunos parámetros en los diferentes módulos del sistema y colecciones de la base de datos.

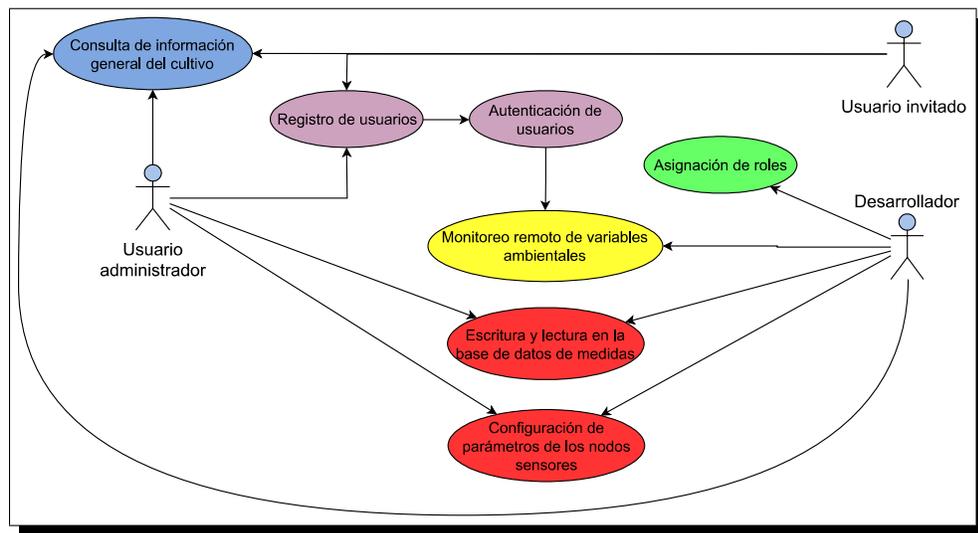


Figura 6. Funcionalidad del sistema de adquisición y gestión de datos.

Características de los usuarios: el sistema presenta tres tipos de usuarios, los cuales se describen a continuación.

- **Desarrollador (Superusuario)**

- **Formación:** ingeniero electrónico, ingeniero de sistemas o técnico en desarrollo de software.
- **Actividades:** proporcionar soporte y mantenimiento al sistema, administrar roles de los usuarios, manejo general del sistema.

- **Administrador**

- **Formación:** ingeniero electrónico o agrónomo.
- **Actividades:** monitoreo periódico de las variables, modificación del modo de operación de los sensores en caso de ser necesario, gestión de los datos recolectados.

- **Invitado**

- **Formación:** no aplica.
- **Actividades:** lectura y/o análisis de la información recolectada.

Restricciones: el siguiente listado muestra las restricciones del sistema.

- El aplicativo debe ser usado con una conexión estable a internet.
- Uso de dominio.
- El navegador debe soportar al menos el estándar ECMAScript 2020 (ES11).
- El navegador debe tener desactivado cualquier Add-on (extensión) de modo oscuro para asegurar una correcta interpretación de colores.
- Lenguajes, servicios y frameworks en uso: HTML, CSS, Typescript, JavaScript, Firebase, Angular, Bootstrap.
- Los servidores deben ser capaces de atender consultas concurrentemente.
- El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.

Suposiciones y dependencias: el siguiente listado muestra las suposiciones y dependencias para el funcionamiento del sistema.

- Se asume que los requisitos aquí descritos son estables.
- Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos antes indicados para garantizar una ejecución correcta del mismo.

Requisitos específicos: este apartado define los requerimientos que cumple el sistema de adquisición y gestión de datos, especificando los requerimientos funcionales y no funcionales.

- **RF01 Registro de usuarios**

- **Características:** los usuarios deben registrarse para acceder a los módulos de monitoreo y/o configuración de parámetros de los nodos sensores.
- **Descripción:** el sistema permite registrarse con un correo electrónico activo y una contraseña con longitud igual o superior a seis caracteres. Adicionalmente, el sistema valida los datos ingresados por el usuario para evitar una operación incorrecta.
- **Requerimientos no funcionales:** RNF01, RNF02, RNF04
- **Prioridad:** alta.

- **RF02 Autenticación de usuarios**

- **Características:** los usuarios deben identificarse (iniciar sesión) para acceder a los módulos de monitoreo y/o configuración de parámetros de los nodos sensores en caso de contar con un rol de administrador.
- **Descripción:** el sistema permite el inicio de sesión a través de un correo electrónico y una contraseña registrados previamente o a través de una cuenta de Google. Por otro lado, el sistema permite la recuperación de contraseña por medio de un link enviado al correo electrónico registrado en caso de ser perdida u olvidada por el usuario. Finalmente, el sistema valida los datos ingresados por el usuario para evitar una operación incorrecta.
- **Requerimientos no funcionales:** RNF01, RNF02, RNF04
- **Prioridad:** alta.

- **RF03 Consulta de información general del cultivo**

- **Características:** el sistema muestra información relacionada con el cultivo haciendo uso de algunos recursos multimedia como imágenes, videos o mapas interactivos.
- **Descripción:** los usuarios pueden acceder (sin estar autenticados) a diversa información del cultivo como su extensión, tipo de cultivo, extensión dedicada a la siembra de café, archivos multimedia (noticias, videos, fotos, etc.), ubicación, entre otros datos que pueden resultar de interés particular.
- **Requerimientos no funcionales:** RNF01, RNF04
- **Prioridad:** media.

- **RF04 Asignación de roles**

- **Características:** el sistema permite asignar roles de invitado o administrador a los usuarios registrados.
- **Descripción:** el sistema permite al desarrollador consultar los identificadores (UID) de los nuevos usuarios para asignar los roles de administrador o invitado. Por defecto a todos los usuarios se les asigna el rol de invitado.
- **Requerimientos no funcionales:** RNF03, RNF04
- **Prioridad:** alta.

- **RF05 Escritura y lectura en la base de datos de medidas**

- **Características:** el sistema permite la lectura y escritura en la base de datos de medidas por parte de los usuarios autenticados.

- **Descripción:** el sistema permite al usuario consultar el número total de registros por variable y eliminarlos en caso de requerirlo. Este módulo está habilitado únicamente para usuarios autenticados con rol de administrador.
 - **Requerimientos no funcionales:** RNF01, RFN02, RFN03
 - **Prioridad:** alta.
- **RF06 Monitoreo remoto de variables ambientales del cultivo**
 - **Características:** el sistema permite visualizar las medidas registradas por los diferentes nodos sensores de manera remota y en tiempo real.
 - **Descripción:** el sistema muestra por medio de gráficas interactivas y tablas de datos la magnitud, fecha y hora del total de medidas registradas para (las diferentes variables temperatura, humedad, CO2, etc) del cultivo, además, señala algunos datos estadísticos como la máxima y mínima medida registrada o el promedio de las mismas. Por otro lado, se integra la función de filtrado por fecha y la exportación de información en formato Excel (.xlsx). Este módulo está habilitado para usuarios autenticados, ya sea con rol de invitado o administrador.
 - **Requerimientos no funcionales:** RNF01, RFN02, RFN03
 - **Prioridad:** alta.
 - **RF07 Configuración de parámetros de los nodos sensores**
 - **Características:** el sistema permite manipular el modo de operación (definir modos de trabajo), umbrales e intervalos de medida de los nodos sensores.
 - **Descripción:** el sistema cuenta con formularios y listas desplegadas que permiten configurar el modo de operación, umbrales e intervalos de medida en los nodos sensores de manera remota. El sistema valida los datos ingresados por el usuario para evitar una operación incorrecta.
 - **Requerimientos no funcionales:** RNF01, RFN02, RFN03, RFN04.
 - **Prioridad:** alta.
 - **RFN01 Interfaz del sistema**
 - **Características:** el sistema cuenta con una interfaz sencilla e intuitiva.
 - **Descripción:** el sistema presenta una interfaz de usuario sencilla, intuitiva y amigable visualmente que permite una fácil navegación a través de ella para los nuevos usuarios.
 - **Prioridad:** alta.

- **RFN02 Recursos de ayuda**

- **Características:** el sistema posee secciones de ayuda que permiten a los nuevos usuarios familiarizarse con el formato de navegación del sistema.
- **Descripción:** en la primera carga del sistema, se despliega un banner que ofrece la descarga de un instructivo en formato PDF. Este documento detalla todos los módulos del sistema y las instrucciones para acceder y utilizar cada uno de ellos correctamente. Además, en la página principal (home page) se incluye una sección de preguntas frecuentes que está disponible para todos los usuarios.
- **Prioridad:** alta.

- **RFN03 Desempeño**

- **Características:** el sistema garantiza tiempos de carga reducidos siempre y cuando la conexión a internet sea estable.
- **Descripción:** se garantiza el desempeño del sistema a los diferentes usuarios. En este sentido, la información o registros modificados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta.
- **Prioridad:** alta.

- **RFN04 Confiabilidad continua del sistema**

- **Características:** el sistema funciona 24 horas del día los 7 días de la semana.
- **Descripción:** la disponibilidad del sistema es continua con un nivel de servicio para los usuarios de 7 días por 24 horas, siempre y cuando la red de nodos sensores esté en condiciones normales de funcionamiento.
- **Prioridad:** alta.

Requisitos comunes de las interfaces: relacionados con las interfaces de hardware, software y usuario descritos a continuación.

- **Interfaces de usuario:** la interfaz de usuario se compone de diferentes módulos (rutas) en las que se encuentran un conjunto de listas, formularios, botones y recursos multimedia que permiten al usuario revisar información, interactuar o realizar determinadas acciones en función de la ruta donde se encuentre y de su privilegio asignado. Dicha interfaz es visualizada desde cualquier tipo de dispositivo el cual soporte y cuente con un navegador web.
- **Interfaces de hardware:** es necesario contar con un equipo de cómputo, el cual integre (o pueda ser integrado externamente) una tarjeta de red, monitor, mouse y teclado. Los anteriores requisitos aplican para ordenadores, en el caso de dispositivos como teléfonos móviles o tabletas solo es necesario la tarjeta de red.

- **Interfaces de software**

- **Sistema operativo:** Windows, Linux (cualquier distribución con interfaz gráfica), Mac OS, Android o iOS.
- **Navegador web:** Google Chrome, Mozilla Firefox, Microsoft Edge, Brave, Opera, LibreWolf o Safari. Cualquier navegador con soporte para el estándar ES2020 (ES11).

3.2.2. Modelamiento de la base de datos

Anteriormente en la sección 3.1.5 se seleccionó la base de datos proporcionada por la plataforma Firebase denominada *Firestore*, esta es una base de datos NoSQL en tiempo real que forma parte de las herramientas ofrecidas por la plataforma [59]. A pesar de ser una base de datos no relacional, Firestore permite almacenar datos en documentos y colecciones, que son estructuras similares a las tablas y filas en una base de datos relacional (SQL). Cada documento es una entidad con un ID único que contiene uno o más campos, donde cada campo tiene un nombre y un valor (con un tipo de dato establecido). La base de datos cuenta con tres colecciones para almacenar información de interés, a continuación se describe cada una de ellas.

Colección para las medidas de las variables

En esta colección se almacena la información relacionada con las variables monitoreadas. En la Figura 7 se muestra la estructura con la cual deben ser registrados los datos.

```
{
  "dateAndTime": Timestamp ,
  "measure": number ,
  "node": number
}
```

Figura 7. Estructura para la colección de medidas de las variables.

La colección cuenta con tres campos principales. El primer campo es llamado “*dateAndTime*”, este recibe un tipo de dato de fecha y hora llamado “Timestamp” (se define como la cantidad de segundos transcurridos desde la medianoche UTC del 1 de enero de 1970, sin contar segundos intercalares) que representa la fecha y la hora en que el sensor tomó la medida. El segundo campo es llamado “*measure*”, este es un número que representa la magnitud medida de la variable en ese momento. El tercer campo llamado “*node*”, es un número que representa el nodo desde el cual

proviene la medición. La estructura de la colección fue diseñada para almacenar y ordenar las mediciones por fecha y hora, para distinguir las mediciones de los diferentes nodos de la red y para facilitar los cálculos de medias, máximos o mínimos de las medidas que se registren.

Colección para los nodos

En esta colección se almacena la información relacionada con los diferentes nodos de la red de sensores. En la Figura 8 se muestra la estructura con la cual deben ser registrados los datos. La colección cuenta con tres campos principales.

```
{
  "mac": string ,
  "nodeId": number ,
  "nodeStatus": boolean ,
  "latitude": number ,
  "longitudo": number
}
```

Figura 8. Estructura para la colección de los nodos.

El primer campo es “*mac*”, que es una cadena de texto que representa la dirección MAC única de identificación del nodo dentro de la red. El segundo campo es “*nodeId*”, que es un número que representa el identificador único del nodo dentro de la red, cabe resaltar que este ID, corresponde al campo “*node*” mostrado en la Figura 7. El tercer campo es “*nodeStatus*”, que es un valor booleano que indica si el nodo está activo (true) o inactivo (false). El cuarto y quinto campo son “*latitude*” y “*longitudo*”, que son datos numéricos que representan la ubicación geográfica del nodo. La estructura de la colección permite almacenar información importante como la dirección MAC del nodo y su localización geográfica para mostrar cada nodo por medio de un mapa.

Colección para los parámetros

En esta colección se almacena la información relacionada con la configuración de umbrales e intervalos de medición. En la Figura 9 se muestra la estructura con la cual deben ser registrados los datos. La colección cuenta con dos subcolecciones, la primera es “*thresholds*”, que contiene valores de umbral para distintas variables como la temperatura, el CO₂, la humedad ambiente y del suelo, y la radiación solar.

```
{
  "thresholds": {
    "Temperatura": number ,
    "CO2": number ,
    "HumedadA": number ,
    "HumedadS": number ,
    "Rad": number
  },
  "intervals": {
    "minutes": number
  }
}
```

Figura 9. Estructura para la colección de configuración de parámetros.

Cada variable tiene un valor numérico asignado que representa el umbral de esa variable en la red de sensores. Es de vital importancia que los campos de esta subcolección sean llamados como se muestra en la Figura 9. La segunda subcolección es “*intervals*”, que contiene un único valor numérico para el intervalo de tiempo en el que se desea tomar las medidas en la red de sensores. En este caso, el valor numérico representa el número de minutos entre medidas consecutivas. La estructura de colección permite configurar y ajustar los parámetros de la red de sensores. Los valores de umbral en la subcolección “*thresholds*” se utilizan para establecer los límites de cada variable medida por los sensores, de manera que si alguno de los valores supera el umbral, el caficultor tome las decisiones pertinentes. La subcolección “*intervals*” se utiliza para establecer el tiempo entre cada medida, lo que permite ajustar la frecuencia de toma de datos en función de las necesidades específicas de la aplicación.

Finalmente, es de vital importancia mencionar que el nodo central de la red inalámbrica de sensores debe estar en escucha constante de cambios en las colecciones de nodos y configuración de parámetros. El objetivo es permitir la gestión remota de la red, para lograrlo, cuando un usuario administrador modifique los umbrales o intervalos de medida, el nodo central ajustará las alertas y notificará a todos los nodos sobre el nuevo periodo de muestreo. Además, si se agrega o se apaga un nodo, se realizará la gestión necesaria para enrutamiento o suspensión del nodo. De esta manera, se logra una gestión eficiente y efectiva de la red inalámbrica de sensores. Lo anterior no supone gran dificultad debido a que *Firestore* se puede integrar con una amplia variedad de lenguajes de programación entre los cuales se encuentran:

- Java
- Kotlin
- Swift
- JavaScript
- Node.js
- Python
- Go
- C++
- Unity (C#)

Firestore ofrece bibliotecas oficiales para estos lenguajes, lo que facilita su integración con aplicaciones de diferentes plataformas. Además, también se puede acceder a Firestore a través de la API REST de Firebase, lo que permite su integración con cualquier lenguaje de programación que pueda realizar solicitudes HTTP [59].

3.2.3. Estructura general del sistema

La Figura 10 muestra el funcionamiento y el flujo de comunicaciones entre los módulos del software y la red inalámbrica de sensores, utilizando como intermediario la plataforma Firebase.

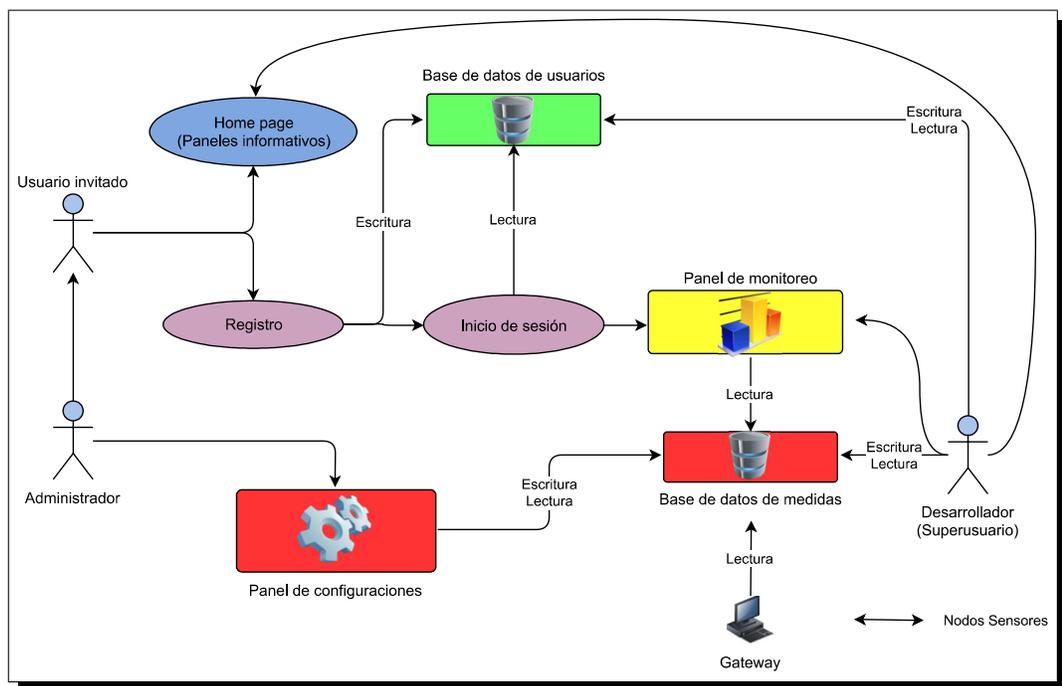


Figura 10. Estructura de funcionamiento del sistema.

Como se ha mencionado en la sección 3.2.1, el sistema cuenta con tres diferentes usuarios. Los usuarios denominados “administrador ” e “invitado” poseen permisos similares, ambos deben pasar por una etapa de registro, esta tiene permisos totales de escritura sobre la base de datos de usuarios, allí se asigna un “UID” único a cada usuario registrado, el “UID” permitirá posteriormente al sistema diferenciar los roles de los usuarios. Seguido a la etapa de registro se procede a la autenticación, allí hay

un permiso total de lectura sobre la base de datos de usuarios, en función de las credenciales proporcionadas por el usuario y la información registrada en la base de datos, el sistema permitirá o denegará el ingreso hacia el panel de monitoreo. Desde el panel de monitoreo se otorga al usuario un permiso de lectura sobre la base de datos de medidas, esta base de datos contiene la información de los nodos, las medidas registradas por los nodos y los parámetros de configuración de la red, dicha base de datos se detalló en profundidad en la sección 3.2.2. Lo descrito hasta el momento es común para ambos tipos de usuario (administrador e invitado), el privilegio añadido que posee el usuario administrador radica en la capacidad de lectura y escritura sobre la base de datos de medidas, es decir, este usuario puede modificar los registros en los nodos o en los parámetros de configuración, lo que en consecuencia ocasiona una manipulación directa en la red inalámbrica de sensores. La explicación del proceso se detalla al final de la sección 3.2.2.

3.2.4. Diseño de la interfaz y experiencia de usuario (UI/UX)

El diseño de interfaz de usuario (UI por sus siglas en inglés User Interface) y la experiencia de usuario (UX por sus siglas en inglés User Experience) son aspectos fundamentales en el desarrollo de productos y servicios digitales. Una UI y UX bien diseñada no solo mejora la satisfacción del usuario, sino que también aumenta la retención del usuario. Además, una UI y UX bien diseñada puede aumentar la eficiencia y la productividad de los usuarios [67]. A continuación se detallan los diferentes aspectos que se tuvieron en cuenta para realizar el diseño de la interfaz gráfica del sistema.

Paleta de colores: en el diseño de la interfaz gráfica en cuestión, se tomaron como base las fotografías presentadas en las Figuras 11 (a) y Figura 11 (b), y se utilizó la herramienta en línea proporcionada por Adobe [68] para generar la paleta de colores presentada en la Figura 12. Cada color fue utilizado para ciertos apartados de la interfaz.

- **Color A:** fondo o background de la página web.
- **Color B:** títulos de los diferentes módulos.
- **Color C:** fondo o background de los botones de envío de información.
- **Color D:** barra de navegación, pie de página o footer y fondo o background de los botones de consulta de información.
- **Color E:** textos descriptivos diferentes a los títulos.



(a)

(b)

Figura 11. Imágenes base para la extracción de la paleta de colores.

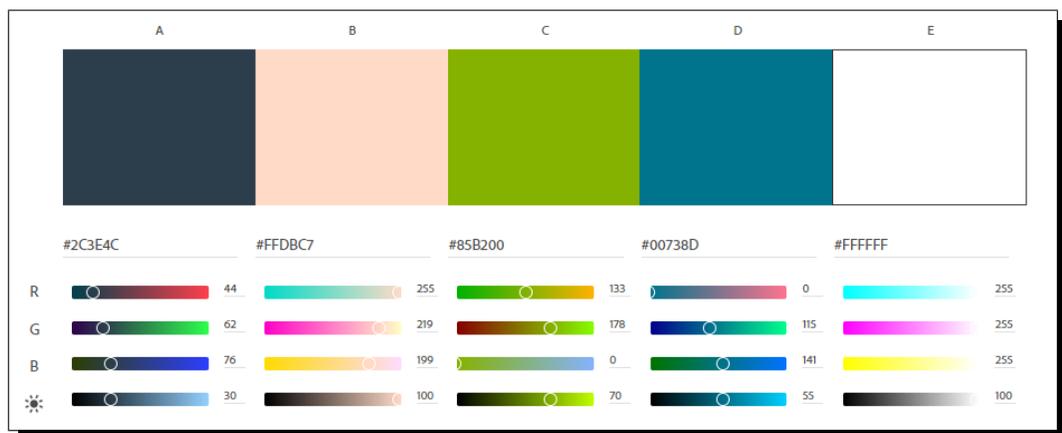


Figura 12. Paleta de colores usada en la interfaz gráfica.

Tipografía: la elección de la tipografía adecuada en el diseño de una interfaz web puede ser complicada debido a las limitaciones y características que presenta cada sistema operativo [67]. Una de las limitaciones más destacadas es que las fuentes disponibles pueden variar en cada sistema operativo, lo que puede dificultar la

presentación coherente de la interfaz. Para abordar este problema, se eligieron tres fuentes diferentes en el diseño de la interfaz gráfica con el fin de proporcionar dos alternativas (Helvetica y Arial, presentes en la mayoría de sistemas operativos) al usuario en caso de que la fuente principal (Roboto) no esté disponible en su dispositivo o presente problemas de carga. Las fuentes seleccionadas se muestran en la Figura 13, y se destacan por su legibilidad, adaptabilidad y neutralidad, lo que permite una fácil lectura y una apariencia limpia y profesional en la interfaz. Además, estas fuentes tienen una amplia gama de variantes, lo que las hace versátiles para su uso en diferentes partes de la interfaz gráfica, como títulos, subtítulos, cuerpos de texto, botones y otros elementos.

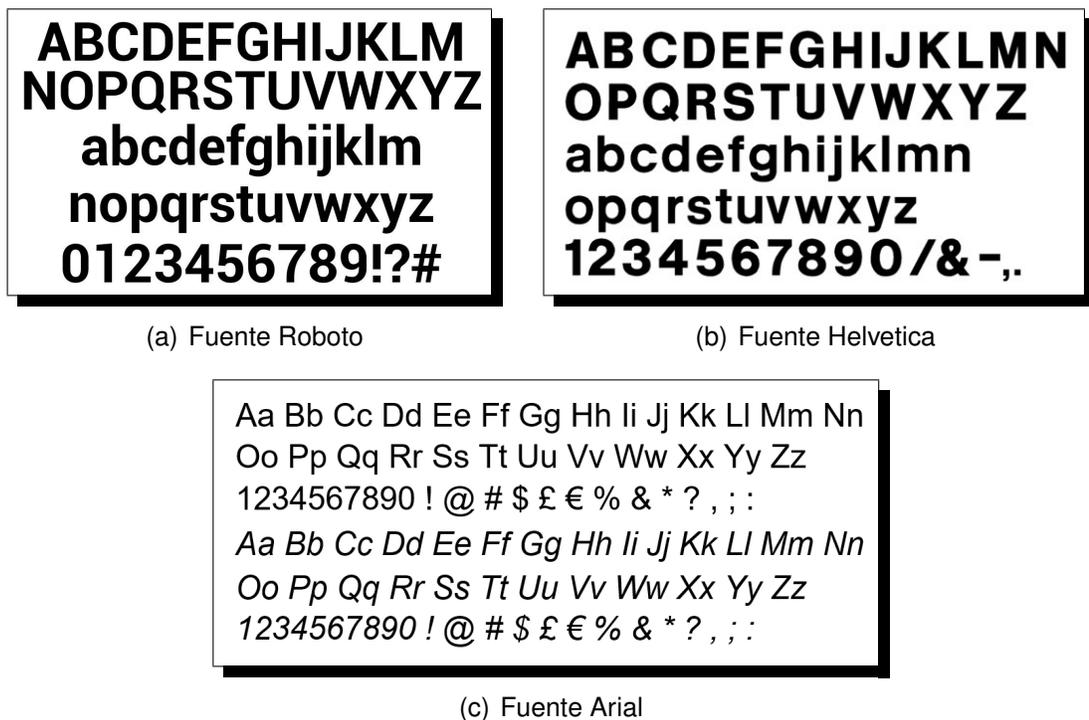


Figura 13. Tipografías seleccionadas para la página web.

Alertas emergentes: en la página web diseñada, se utilizaron diferentes alertas emergentes para brindar al usuario una retroalimentación constante sobre lo que está sucediendo. Además, las alertas permiten que el usuario interactúe con ellas, lo que mejora la experiencia de navegación. En un ejemplo específico, una alerta emergente muestra información detallada sobre la sesión actual del usuario, incluyendo su nombre, correo electrónico y rol otorgado dentro del sistema. Desde allí, el usuario tiene la opción de cerrar la alerta o finalizar la sesión si es necesario, como se puede ver en la Figura 14. Al utilizar alertas emergentes de manera efectiva, se puede mejorar la experiencia del usuario y aumentar la interacción con la página web.

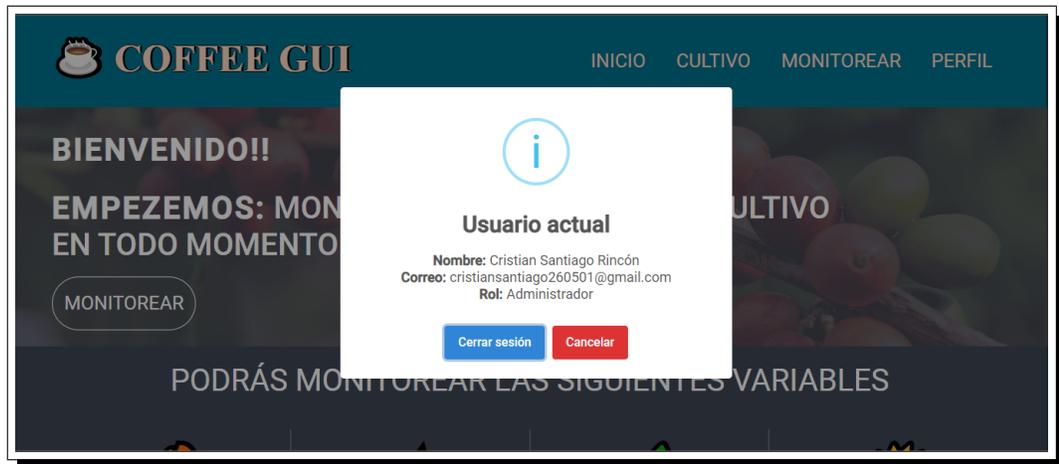
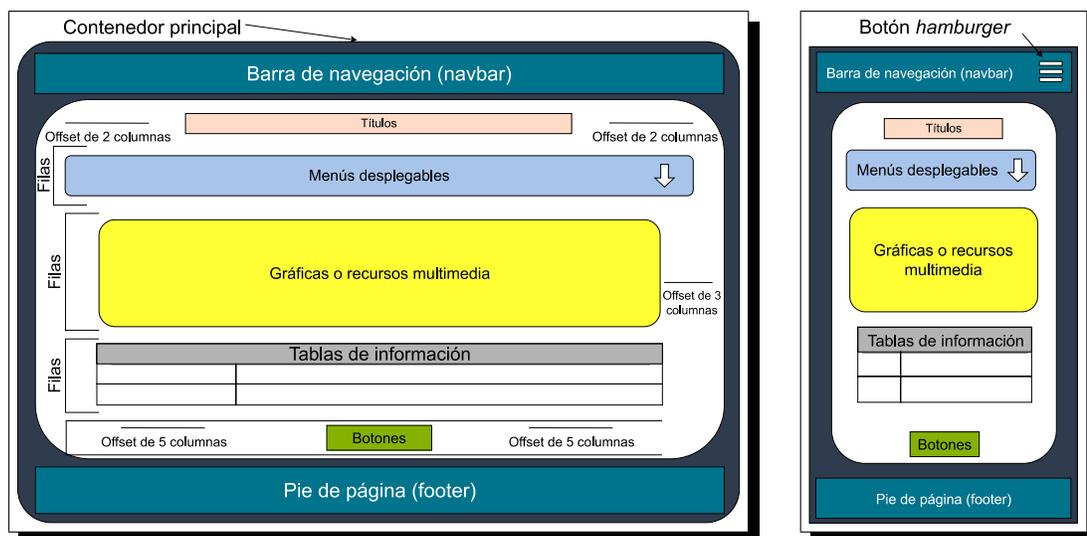


Figura 14. Alerta emergente informativa.

Animaciones: para el diseño de la página web se implementaron diferentes animaciones con el fin de mejorar la experiencia del usuario. Las animaciones son utilizadas para guiar al usuario a través de diferentes partes de la aplicación o para mostrar cambios importantes en la información o en el estado de la aplicación, como por ejemplo la actualización de medidas de alguna variable. Por otro lado, las animaciones también mejoran la usabilidad de la aplicación, al hacer que las transiciones entre diferentes pantallas o secciones sean más fluidas y naturales.

Velocidad de carga: una de las medidas más importantes es la compresión de los diferentes recursos que componen la página, como imágenes, scripts, iconos, tipografías externas y hojas de estilos. La compresión de estos recursos permite que sean descargados más rápidamente, lo que a su vez reduce el tiempo de carga de la página. Otra técnica implementada en este proyecto fue el uso de una SPA, que se refiere a un tipo de aplicación web que carga solo una página HTML y luego actualiza dinámicamente el contenido de la página en respuesta a las acciones del usuario, sin la necesidad de recargar la página completa, debido a que la información se guarda en la memoria del navegador.

Distribución de los componentes: con el fin de evitar fallos de diseño relacionados con la representación de la página web [69], se establecieron las plantillas mostradas en la Figura 15 (a) y Figura 15 (b) con el sistema de grillas que maneja el framework Bootstrap. Por otra parte, se siguieron líneas de diseño que se encuentran en la mayoría de páginas web, por ejemplo, una barra de navegación en la parte superior ocupando el 100 % del ancho de la pantalla (en versiones de escritorio) o un menú desplegable desde un lateral de la pantalla (en versiones móviles), el inicio de sesión en la parte superior derecha, la información que se destaca ubicada en la parte central y un pie de página o también denominado footer en la parte inferior con información sobre el desarrollador y página en sí.



(a) Layout para escritorio.

(b) Layout para móvil.

Figura 15. Layouts de la página web.

Bootstrap permite el manejo de un sistema de grillas que divide el total del ancho de la página web en 12 columnas, permitiendo una distribución flexible de los elementos y componentes en la pantalla. Cada fila (row) está compuesta por un contenedor (container) y una o más columnas (column), a su vez, el contenedor se utiliza para envolver las columnas y limitar su ancho. Las columnas, por otro lado, son elementos que tienen el contenido real de la página, como texto, imágenes, formularios, etc. Todos los elementos de la página web se encuentran contenidos en un contenedor principal (div) y a partir de este se empieza a generar diferentes filas para distribuir los elementos dentro de la página. Además, se establece un número determinado de columnas en cada fila, dependiendo de la cantidad de elementos que se van a contener en la misma. Por otra parte, tanto en el contenedor principal como en cada una de las filas generadas se deja un *offset* a cada lado para evitar que los elementos se peguen demasiado al borde de pantalla. Finalmente, en cuanto a la plantilla de la versión móvil solo presenta una diferencia en su menú de navegación, debido a que este se despliega por medio de un botón (parte superior derecha) tipo *hamburger*, a diferencia de la plantilla de versión de escritorio, el cual cuenta con los botones de menú en la barra de navegación directamente.

3.3. IMPLEMENTACIÓN

Teniendo en cuenta el diseño presentado en la sección 3.2 a continuación, se muestra la implementación del sistema desarrollado con Angular y Firebase.

3.3.1. Herramientas y tecnologías utilizadas

En la Tabla 4 se mencionaron de manera general las herramientas y tecnologías seleccionadas para el desarrollo de la aplicación web; en esta sección se describirá con más detenimiento cada una de ellas.

Sistema de gestión de paquetes: en la etapa de desarrollo de una aplicación web es esencial la instalación y gestión de algunas librerías y paquetes que facilitan ciertas tareas. Una de las herramientas más populares dentro del desarrollo web para el manejo de dichos paquetes y librerías es *npm* (Node Package Manager), este gestor de paquetes es el registro de software más grande del mundo. Los desarrolladores de código abierto usan *npm* para compartir y tomar prestados paquetes [70]. Para el desarrollo de la aplicación web se empleó la versión 9.4.1 de *npm*, esta permitió la instalación de las diferentes dependencias y librerías necesarias para el correcto avance del proyecto. Algunas de estas son:

- @angular/cli
- @angular/animations
- @angular/fire
- @angular/forms
- @swimlane/ngx-charts
- animate.css
- bootstrap
- firebase
- sweetalert2

Angular CLI: debido a que el framework de desarrollo de JavaScript escogido en la sección 3.1 fue Angular, el empleo de la interfaz de línea de comandos (CLI por sus siglas en inglés Command Line Interface) de angular fue primordial.

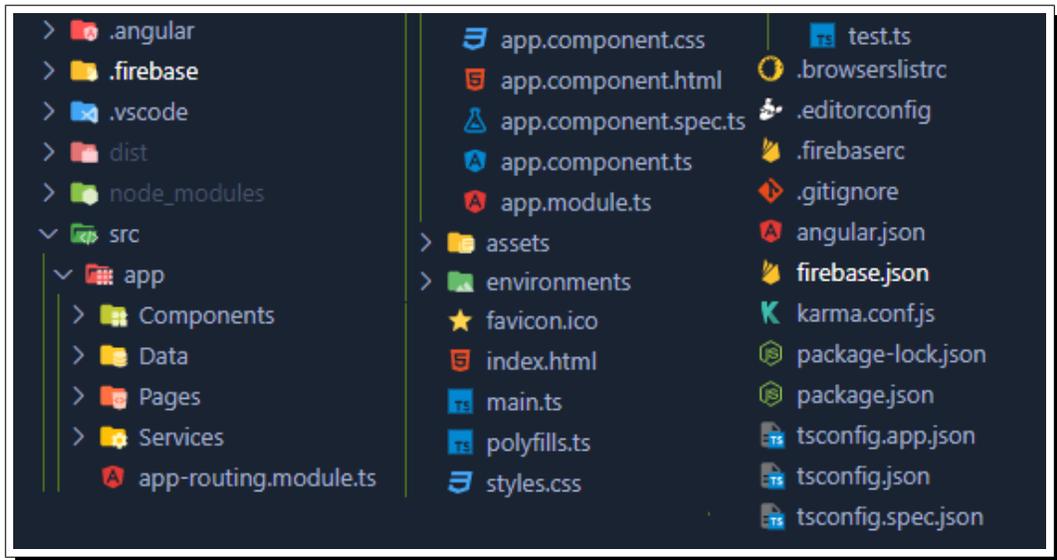


Figura 16. Estructura de carpetas y ficheros representada de forma lineal de la aplicación creada por medio del CLI de angular.

El CLI de Angular es la forma más rápida, sencilla y recomendada de desarrollar aplicaciones de Angular, este simplifica algunas tareas como por ejemplo la compilación de la aplicación, la generación de archivos basados en un esquema establecido (componentes, servicios, etc.), ejecución de pruebas unitarias, entre otras tareas [71]. Para el desarrollo de la aplicación web se empleó la versión 14.2.3 del CLI de angular, este permitió la creación de los diferentes componentes, servicios y archivos que conforman la aplicación final, en la Figura 16 se observa la estructura de las diferentes carpetas y ficheros creados.

A continuación se detallan los principales ficheros y carpetas de la aplicación:

- **Carpeta *dist*:** la carpeta *dist* contiene los archivos finales que se generan después de que el código fuente de la aplicación Angular se ha compilado y optimizado para la producción. Incluye el código JavaScript y CSS que se sube al servidor de hosting.
- **Carpeta *node_modules*:** la carpeta contiene los paquetes de software que el proyecto de Angular necesita para funcionar correctamente. Incluye tanto las dependencias directas del proyecto (es decir, las bibliotecas que el proyecto ha solicitado explícitamente a través de su archivo *package.json*) como las dependencias indirectas (es decir, las bibliotecas que las bibliotecas directas del proyecto necesitan para funcionar).
- **Carpeta *app*:** dentro de la carpeta *app*, se encuentran los componentes, servicios, directivas, tuberías (pipes) y otros elementos que componen la lógica y la presentación de la aplicación. Estos archivos se organizan en subcarpetas, como *components*, *services*, *pages*, o *data* según el tipo de elemento que representan.
- **Carpeta *src*:** la carpeta *src* en una aplicación de Angular es donde se guarda la mayor parte del código fuente. Contiene las siguientes carpetas y archivos clave: *app*, *assets*, *environments*, *index.html* y *styles.css*.
- **Fichero *index.html*:** este archivo es la página principal de la aplicación y se carga cuando se accede a la aplicación en el navegador. Angular automáticamente inyecta el contenido de la aplicación dentro del elemento *app-root*, donde se carga el componente raíz de la aplicación.
- **Fichero *styles.ccs*:** este archivo contiene los estilos globales que se aplican en toda la aplicación.
- **Fichero *app-routing.module.ts*:** este fichero es un módulo de enrutamiento en Angular que define las rutas de la aplicación y la navegación a través de ellas. Permite que la aplicación tenga una única página principal y que los componentes se carguen dinámicamente a medida que se navega por la aplicación.

- **Fichero *app.module.ts*:** este fichero es el componente raíz de la aplicación. Este componente se encarga de renderizar el esqueleto de la aplicación, incluyendo el menú de navegación y el enrutamiento básico.
- **Carpeta *assets*:** la carpeta *assets* es el lugar donde se almacenan los archivos estáticos que se utilizan en la aplicación. Estos archivos pueden incluir imágenes, archivos de audio, hojas de estilo CSS, fuentes, etc.
- **Fichero *firebase.json*:** este fichero es un archivo de configuración utilizado para definir las opciones de implementación y hosting en Firebase. Contiene información sobre la configuración de la base de datos, las reglas de seguridad, las opciones de almacenamiento y los parámetros de alojamiento web.
- **Fichero *package.json*:** este fichero es un archivo de configuración de Node.js que se utiliza para definir las dependencias y los scripts del proyecto. Contiene información sobre las dependencias del proyecto, incluyendo los paquetes de Node.js que se utilizan, las versiones de los paquetes y las dependencias de terceros. También contiene información sobre los scripts de construcción y prueba del proyecto. El archivo se utiliza junto con el gestor de paquetes “npm” para instalar y gestionar las dependencias del proyecto.

Bootstrap: para obtener un diseño responsivo de la aplicación, en la sección 3.1 se seleccionó el framework de CSS Bootstrap. Para el desarrollo de la aplicación web se empleó la versión 5.2.3, este permitió hacer uso de algunos componentes prediseñados como formularios, botones, barras de navegación, entre otros componentes, además, permitió modificar el estilo de dichos componentes y adecuarlos a los requerimientos de la aplicación. Así mismo, Bootstrap permitió el uso de un diseño basado en grillas, a través de este se logró organizar y distribuir el contenido de la página de forma más eficiente y estética. Finalmente, el framework proporcionó una mayor velocidad en el desarrollo del diseño de la aplicación, debido a que no fue necesario crear los componentes y estilos desde cero, por el contrario se hizo uso de los recursos que el framework ofrece.

Firestore Tools: como se concluyó en la sección 3.1, Firestore fue la plataforma escogida para prestar los servicios de base de datos, autenticación y hosting. Este último servicio hace necesaria la implementación del CLI de Firestore denominado “*Firestore-tools*”, esta herramienta permite trabajar con servicios de Firestore. Se utiliza para implementar y administrar aplicaciones de Firestore, incluyendo la implementación de aplicaciones web, la administración de bases de datos en tiempo real, la autenticación de usuarios, la administración de notificaciones push y la configuración de reglas de seguridad [59]. Para el desarrollo de la aplicación web se empleó la versión 8.15.0 de las *Firestore-tools*, esta permitió el despliegue de la aplicación web de Angular compilada en los servidores de firestore, obteniendo los dominios de acceso público <https://app-coffee-gui.web.app> y <https://app-coffee-gui.firebaseio.com>.

3.3.2. Módulos del sistema

El sistema denominado “Coffe GUI” cuenta con cinco diferentes módulos, en la Figura 17 se muestra de manera general cada uno de ellos y las acciones permitidas desde los mismos.

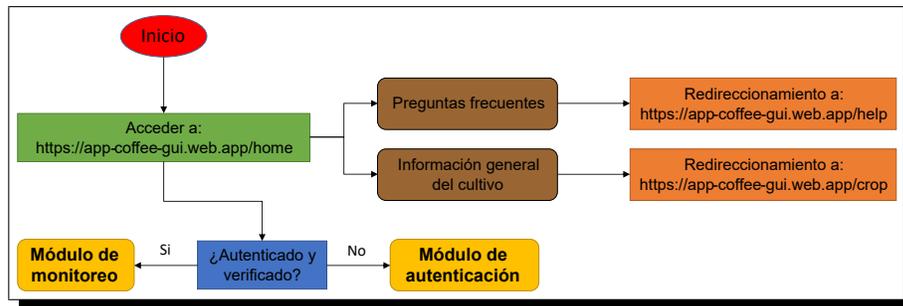


Figura 17. Diagrama general del sistema de gestión remota.

A continuación, se describen los diferentes módulos especificando la forma de navegación a través de ellos y el alcance de permisos para cada uno de los usuarios correspondientes.

Home Page: el diagrama de navegación y la interfaz de usuario del módulo denominado “Home Page” se presentan en la Figura 18 (a) y Figura 18 (b) respectivamente, este es el primer módulo en presentarse una vez se ingresa a la página web a través del dominio <https://app-coffee-gui.web.app>. Desde este módulo el usuario tiene acceso a la información general del cultivo y a la sección de preguntas frecuentes sin necesidad de contar con una sesión activa, en la Figura 19 (a) y Figura 19 (b) se observan los apartados anteriormente mencionados. Este módulo verifica el estado de autenticación del usuario, es decir, si hay o no una sesión activa, en función de

dicha verificación se permite el acceso al módulo de monitoreo o de autenticación según corresponda.



(a) Diagrama de navegación del módulo "Home Page".



(b) Interfaz de usuario del módulo "Home Page".

Figura 18. Módulo "Home Page".



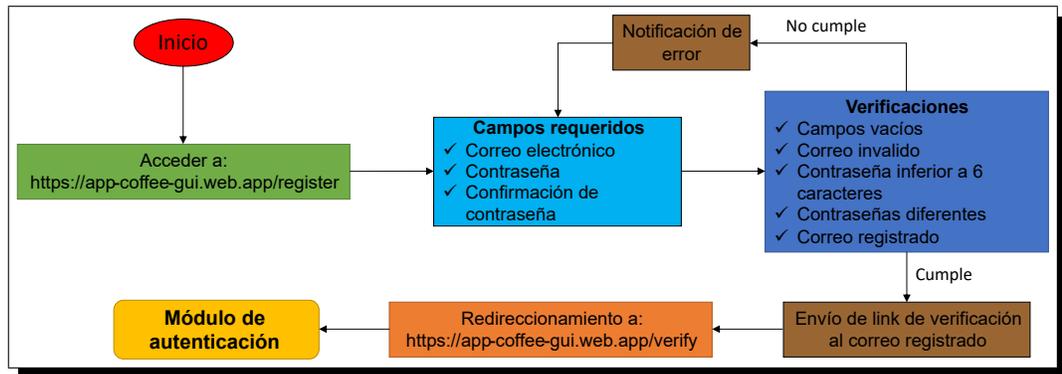
(a) Interfaz de la información general del cultivo.



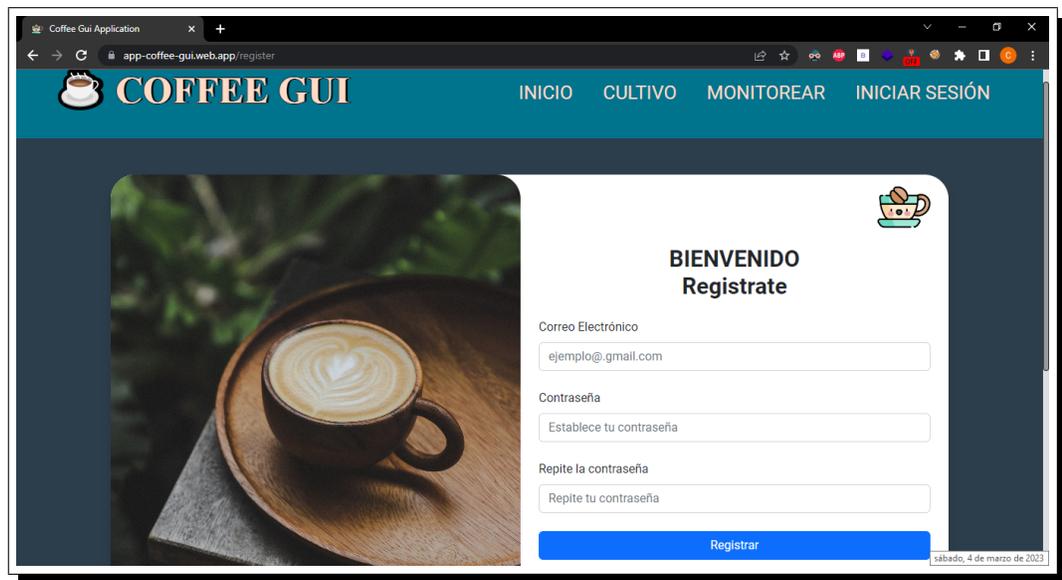
(b) Interfaz de la sección de preguntas frecuentes.

Figura 19. Secciones habilitadas para usuarios sin sesión activa.

Módulo de registro: el diagrama de navegación y la interfaz de usuario del módulo de registro se presenta en la Figura 20 (a) y Figura 20 (b) respectivamente, desde allí se permite al usuario crear una cuenta proporcionando a través de un formulario, un correo electrónico activo y una contraseña de longitud superior a seis caracteres.



(a) Diagrama de navegación del módulo de registro.



(b) Interfaz de usuario del módulo de registro.

Figura 20. Módulo de registro.

Algunas validaciones que se implementan sobre los datos ingresados por el usuario son correos inválidos o registrados, campos vacíos y contraseñas diferentes o débiles. Los errores son notificados como se muestra en la Figura 21 en caso de no cumplirse alguna de las validaciones anteriormente mencionadas. Una vez todos los campos sean diligenciados correctamente se procede al envío de un link, que permite verificar el correo ingresado, por esta razón es de vital importancia el registro de un correo activo. En caso de no verificar el correo por medio de dicho

link, el sistema no permitirá el inicio de sesión y redirigirá al usuario hacia la sección mostrada en la Figura 22.

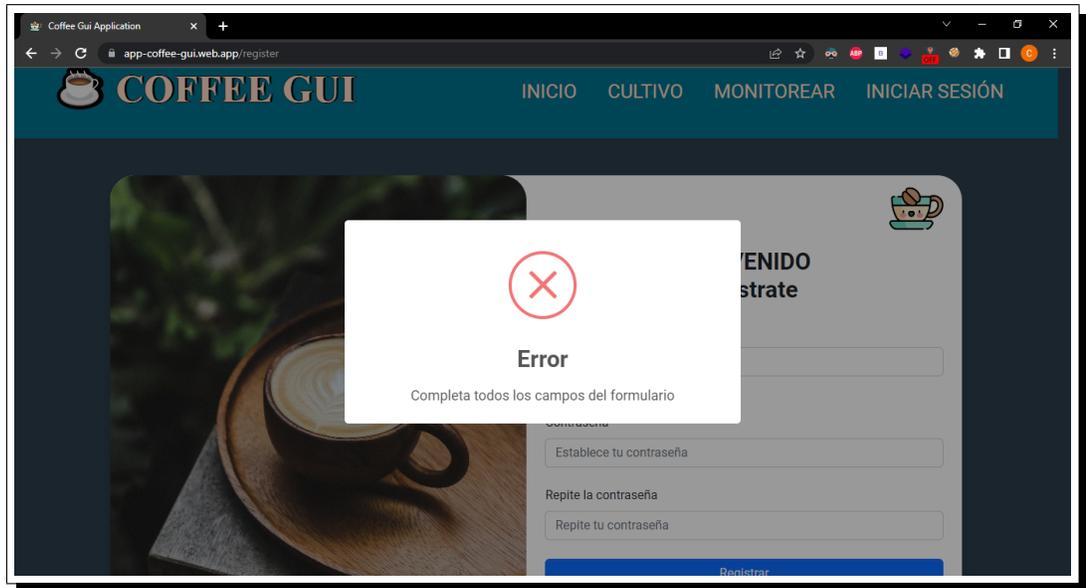


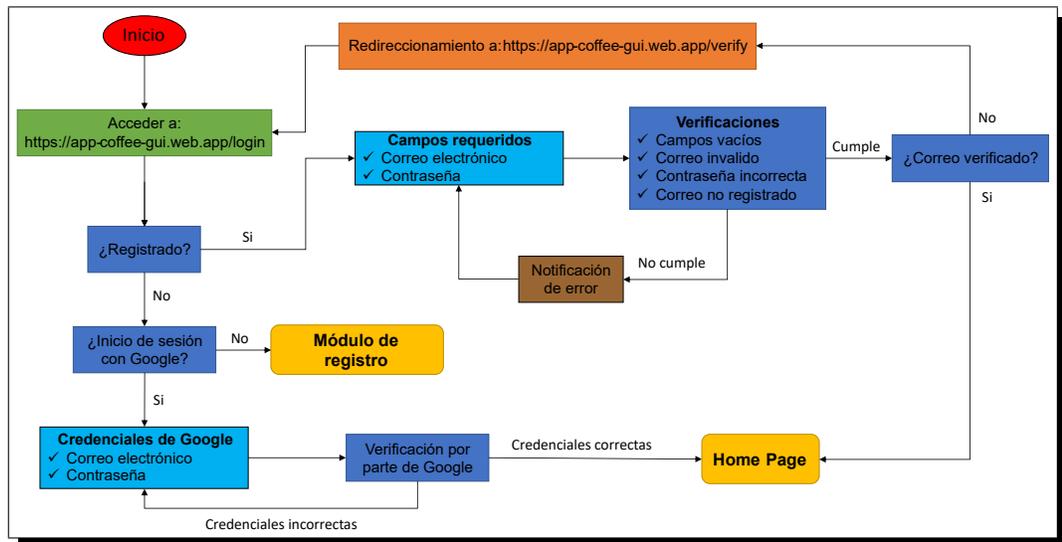
Figura 21. Ejemplo de notificación de error en el módulo de registro.



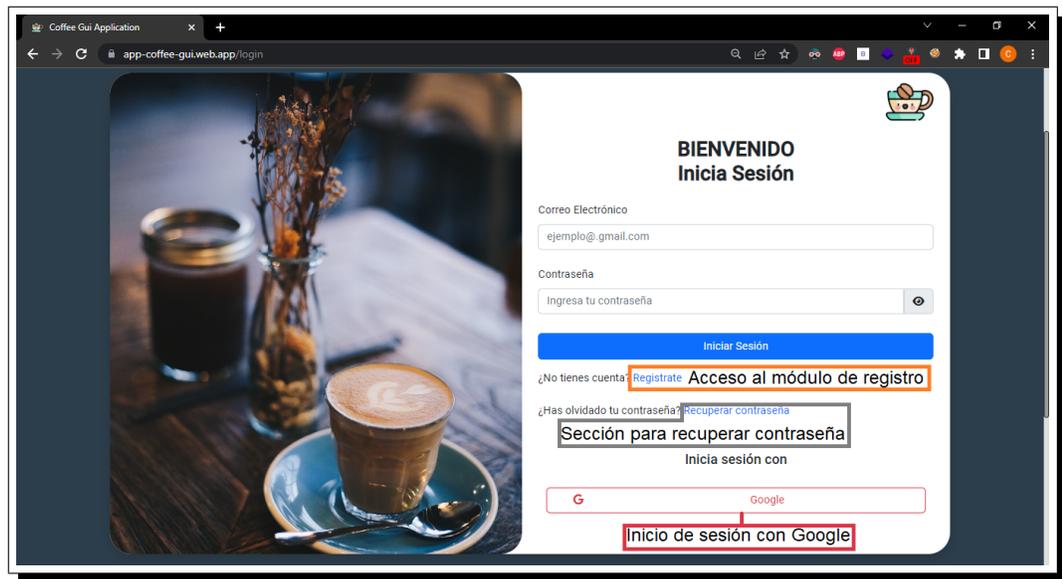
Figura 22. Mensaje mostrado al intentar iniciar sesión sin previa verificación del correo electrónico.

Módulo de autenticación: el diagrama de navegación y la interfaz del módulo de autenticación se presenta en la Figura 23 (a) y Figura 23 (b) respectivamente, des-

de este módulo se permite al usuario identificarse por medio de un correo y una contraseña previamente registrados.



(a) Diagrama de navegación del módulo de autenticación.



(b) Interfaz de usuario del módulo de autenticación.

Figura 23. Módulo de autenticación.

Los formularios desplegados para el ingreso de credenciales (correo electrónico y contraseña) realizan diferentes verificaciones, con el fin de tener un control total sobre el “input” del usuario, en caso de presentarse algún error o fallo, este es notificado mediante alertas emergentes similares a la mostrada en la Figura 21. Cabe resaltar que el sistema permite al usuario la recuperación de contraseña, en la Fi-

gura 24 se observa el formulario que debe ser completado para llevar a cabo esta acción.

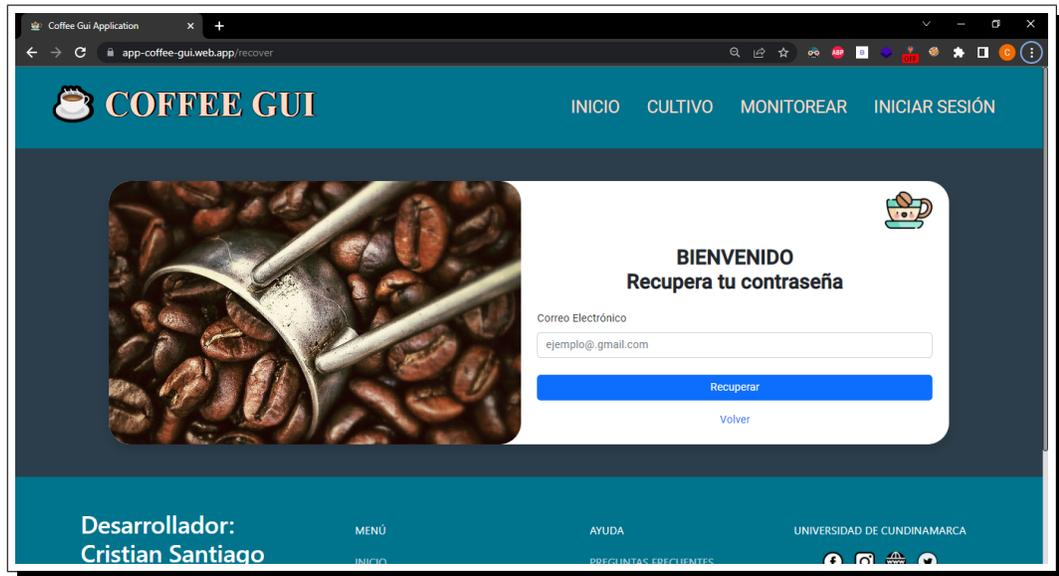
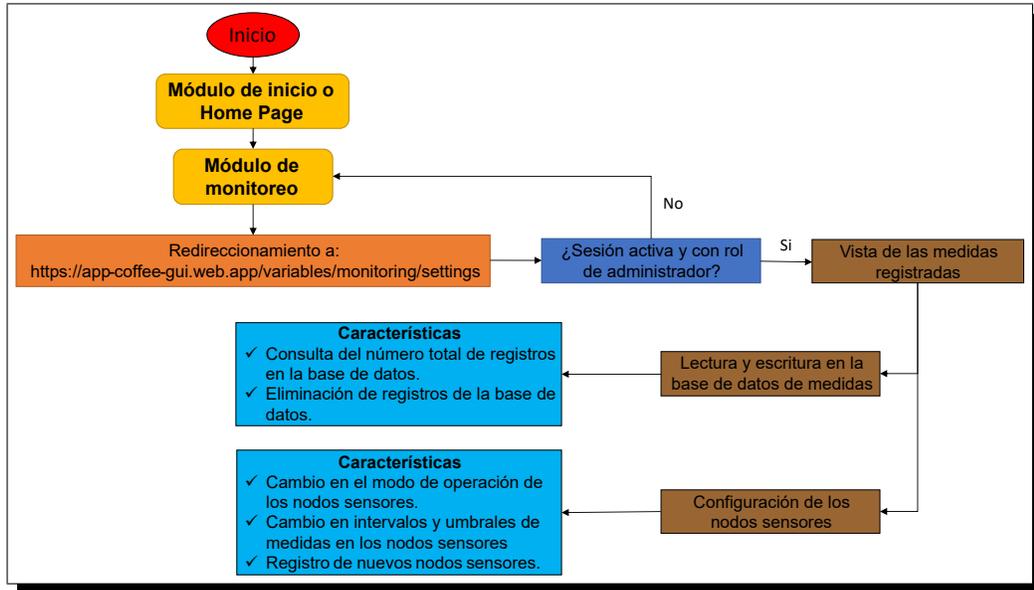


Figura 24. Sección para recuperación de contraseña.

El módulo comprueba si el usuario verificó el correo electrónico a través del link enviado previamente por el módulo de registro, de no ser así, el acceso es denegado y redirige al usuario hacia la sección mostrada en la Figura 22. En caso de que el usuario no esté registrado, el sistema le permite iniciar sesión a través de una cuenta de Google. Una vez que se proporcionan las credenciales correspondientes, se concede acceso al Home Page con una cuenta activa, siempre y cuando las credenciales sean correctas. Finalmente, si ninguna de las anteriores alternativas permiten al usuario la autenticación, es posible acceder al módulo de registro, el cual se detalló anteriormente.

Módulo de configuraciones: el diagrama de navegación y la interfaz del módulo de configuración se presenta en la Figura 25 (a) y Figura 25 (b) respectivamente, desde este módulo se permite a los usuarios con rol de administrador realizar configuraciones en la base de datos de medidas (eliminación de registros) y en la red inalámbrica de nodos de manera remota, lo que significa que pueden realizar cambios y ajustes importantes desde cualquier lugar con conexión a Internet. Algunos de estos cambios pueden ser modificación en los umbrales e intervalos de medida por variable, manipulación en el estado de los nodos (encender o apagar) o registro de nuevos nodos. Al igual que los anteriores módulos, en cada uno de los formularios desplegados en el módulo incluye diferentes verificaciones para garantizar la integridad de los datos ingresados. Esto significa que se realizan diferentes tipos de verificaciones como validar el tipo de datos ingresados, la estructura de las cadenas

de texto ingresadas, campos vacíos, y muchas otras. Estas verificaciones garantizan que los datos ingresados sean correctos y coherentes con las configuraciones que se están realizando.



(a) Diagrama de navegación del módulo de configuraciones.

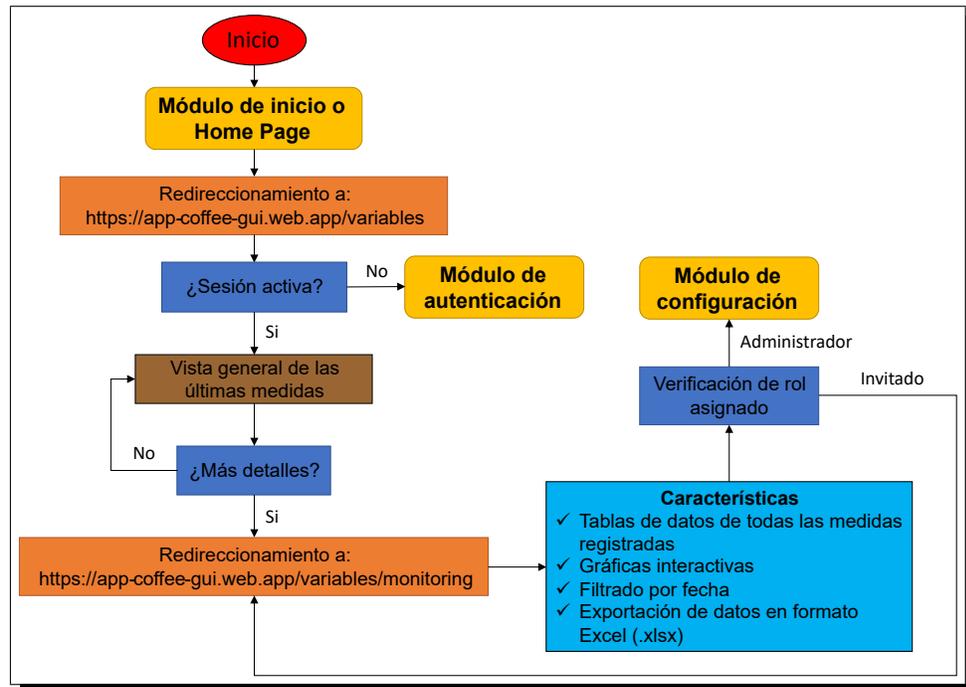


(b) Interfaz de usuario del módulo de configuraciones.

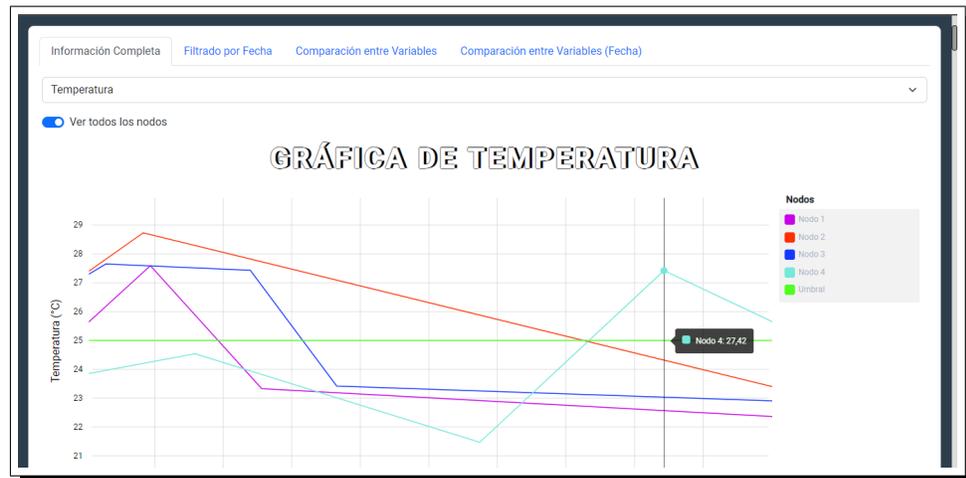
Figura 25. Módulo de configuraciones.

Módulo de monitoreo: el diagrama de navegación y la interfaz del módulo de monitoreo se presenta en la Figura 26 (a) y Figura 26 (b) respectivamente, desde este módulo se permite a los usuarios autenticados visualizar el registro total o parcial

(por fecha) de medidas de los diferentes nodos sensores, filtrar por variable o por nodo, exportar la información en formato Excel (.xlsx) y acceder al módulo de configuraciones en caso de contar con un rol de administrador. La información se presenta a través de gráficas de magnitudes vs tiempo y tablas de datos.



(a) Diagrama de navegación del módulo de monitoreo.



(b) Interfaz de usuario del módulo de monitoreo.

Figura 26. Módulo de monitoreo.

3.4. PRUEBAS DE FUNCIONAMIENTO

En esta sección se presentan las pruebas que se llevaron a cabo en los diferentes módulos del software desarrollado, se diseñaron y ejecutaron pruebas específicas para cada uno de los módulos del software, con el fin de detectar y corregir cualquier error o anomalía presente en el sistema. A continuación, se describen detalladamente las pruebas realizadas y los resultados obtenidos en cada una de ellas.

3.4.1. Pruebas del diseño responsivo

Como se mencionó en la sección 3.1.2, el diseño responsivo permite mejorar la experiencia del usuario y aumenta la accesibilidad. Por lo tanto, las pruebas para comprobar la eficacia de este diseño son cruciales para garantizar que la aplicación se vea y funcione de manera óptima en diferentes dispositivos y tamaños de pantalla.

Pruebas de interacción con la interfaz de usuario: esta prueba consistió en probar cada uno de los diferentes elementos con los que interactúa el usuario a la hora de navegar por la página web, como botones, menús, hipervínculos, mapas, ventanas emergentes, gráficas, tablas o fotos. Cada uno de estos elementos se probó en diferentes dispositivos con distintas resoluciones de pantalla, con el objetivo de observar el funcionamiento de los mismos. En la Tabla 5 se observan los resultados obtenidos.

Tabla 5. Resultados de las pruebas en diferentes dispositivos con distintas resoluciones de pantalla.

Dispositivo	Resolución de pantalla	Respuesta del sistema
Ordenador	1366 x 768	✓
	1920 x 1080	✓
	2560 x 1440	✓
	1280 x 1024	✓
Teléfono móvil	750 x 1334	✓
	1080 x 1920	✓
	1440 x 2560	✓
Tabletas	768 x 1024	✓
	1536 x 2048	✓
	1668 x 2224	✓

Los resultados de las pruebas realizadas indican que el diseño responsivo de la aplicación web es exitoso en todos los casos, lo que significa que la aplicación web es

accesible y funcional en diferentes resoluciones de pantalla, incluyendo ordenadores, dispositivos móviles y tablets. Sin embargo, se recomienda que en la sección o módulo de monitoreo, la navegación se realice en formato horizontal en dispositivos móviles y tablets, esto con el fin de mejorar la experiencia del usuario (ver Figura 27 (a)). Un formato vertical puede dificultar la navegación y visualización en dispositivos móviles y tablets (ver Figura 27(b)).

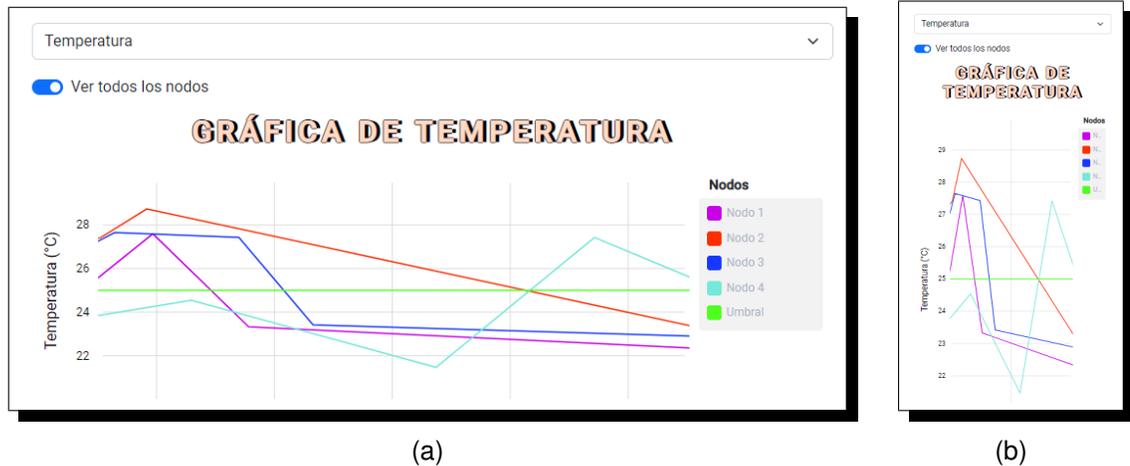


Figura 27. Vista del módulo de monitoreo desde un dispositivo móvil en formato horizontal (a) y vertical (b).

3.4.2. Pruebas del módulo Home Page

El módulo de “Home Page” es la primera vista que los usuarios tienen al cargar la página web, además, permite la navegación hacia los demás módulos del sistema. Por lo tanto, es fundamental que este módulo este diseñado de manera efectiva y presente la información de manera clara y accesible para los usuarios. Con el objetivo de verificar la funcionalidad de este módulo, se realizó la siguiente prueba de funcionamiento.

Pruebas de navegación entre los diferentes módulos del sistema: esta prueba consistió en navegar hacia los diferentes módulos del sistema que están disponibles desde el “Home Page” en dos diferentes escenarios, uno teniendo una sesión activa y otro sin haber iniciado sesión. Los resultados de la prueba se presentan en la Tabla 6, y permiten concluir que el módulo presenta un correcto funcionamiento, debido a que autoriza o restringe el acceso de manera lógica y correcta según los requerimientos establecidos en la sección 3.2.1.

3.4.3. Pruebas del módulo de registro

El módulo de registro permite la creación de una cuenta de usuario para autenticarse posteriormente ante el sistema. Este módulo permite que los usuarios ingresen información, lo que a su vez da cabida a posibilidad del ingreso de datos incorrectos que provoquen un fallo o mal funcionamiento en el sistema. Con el objetivo de evitar algún fallo, se diseñaron pruebas específicas para verificar que este módulo funcione de manera efectiva y cumpla con las necesidades de los usuarios y del sistema.

Tabla 6. Resultados de las pruebas de navegación del módulo “Home Page”.

Prueba	Respuesta con sesión activa	Respuesta sin sesión activa
Navegación a la información general del cultivo	Acceso permitido	Acceso permitido
Navegación a la sección de preguntas frecuentes	Acceso permitido	Acceso permitido
Navegación al módulo de registro	Acceso denegado, solo se permite cerrar o ver información sobre la sesión activa	Acceso permitido a través del módulo de autenticación
Navegación al módulo de autenticación	Acceso denegado, solo se permite cerrar o ver información sobre la sesión activa	Acceso permitido
Navegación al módulo de monitoreo	Acceso permitido	Acceso denegado, redireccionamiento al módulo de autenticación
Navegación al módulo de configuraciones	Acceso permitido a través del módulo de monitoreo	Acceso denegado

Pruebas de validaciones de los formularios: esta prueba consistió en establecer diferentes escenarios de entradas de usuarios por los cuales puede pasar el formulario del módulo de registro (ver Figura 20 (b)), se contemplaron campos vacíos, contraseñas diferentes, caracteres especiales, etc. Los resultados de cada escenario se muestran en la Tabla 7, el símbolo (✓) significa que hubo un registro exitoso, por el contrario, el símbolo (×) significa que se produce una notificación de error como la mostrada en la Figura 21.

Con los resultados obtenidos con los diferentes escenarios, se concluye que los formularios del módulo de registro presenta un correcto funcionamiento, debido a que no se permite el envío de información con un formato al no esperado por el sistema. Sin embargo, cabe resaltar que al ingresar caracteres especiales en el campo de correo se permite el envío de información, esto debido a que caracteres de signos de puntuación, como la coma (,), el punto y coma (;), los dos puntos (:), entre otros, no se pueden utilizar en la parte local del correo electrónico, pero sí en el dominio. Entonces, es importante tener en cuenta que algunos proveedores de correo electrónico pueden tener o no restricciones específicas sobre los caracteres especiales permitidos.

Tabla 7. Resultados de las pruebas de validación de formularios del módulo de registro.

Prueba	Respuesta del sistema
Ingreso de un correo inválido	×
Ingreso de contraseñas diferentes	×
Ingreso de contraseña inferior a 6 caracteres	×
Campo de correo vacío	×
Campo de contraseña vacío	×
Campo de repetir contraseña vacío	×
Ingreso de caracteres especiales en los campos de contraseña	✓
Ingreso de caracteres especiales en el campo de correo electrónico	✓
Ingreso de un correo registrado	×

Prueba de envío del link de verificación del correo electrónico: el proceso de verificación de correo electrónico es un paso importante en la seguridad y la autenticación de los usuarios en un sistema. Es esencial que el sistema pueda enviar correctamente el link de verificación al correo electrónico registrado para garantizar que los usuarios puedan acceder al sistema de manera segura. La Figura 28 muestra un ejemplo del correo electrónico que es enviado para la verificación de cuenta, este correo muestra información clara y fácil de entender para que los usuarios puedan completar con éxito el proceso de verificación.

Esta prueba consistió en registrar 15 correos diferentes, con el objetivo de verificar el envío del link, el tiempo que tarda en llegar y el lugar donde se recibe (bandeja de entrada o spam). Los resultados de la prueba se presentan en la Tabla 8.

Los resultados obtenidos con el registro de los diferentes correos permiten concluir que el link de verificación tiene un acierto del 100%, además, el promedio de tiempo de llegada del link es menor a un minuto. Por último, es importante resaltar que la mayoría de las veces, el link llegará a la bandeja de correo no deseado (spam),

por lo cual, se advierte al usuario que revise dicha bandeja una vez se registra (ver Figura 22).

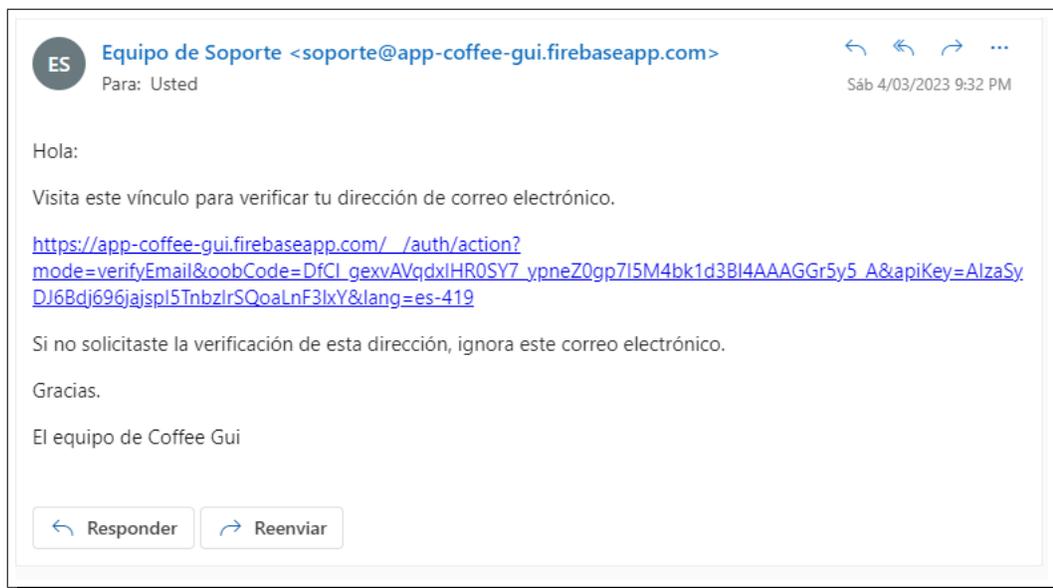


Figura 28. Estructura del correo enviado por firebase para la verificación de cuenta.

Tabla 8. Resultados de la prueba de envío del link de verificación del correo electrónico para el módulo de registro.

	Correos registrados
Envío del link a la bandeja de entrada	3 correos
Envío del link a la bandeja de spam	12 correos
Tiempo de envío menor a un minuto	9 correos
Tiempo de envío entre un minuto y tres minutos	5 correos
Tiempo de envío superior a tres minutos	1 correo

3.4.4. Pruebas del módulo de autenticación

El módulo de autenticación permite que los usuarios ingresen al sistema al proporcionar sus credenciales de inicio de sesión. Dado que la autenticación es una funcionalidad crítica, se llevaron a cabo pruebas específicas para garantizar que este módulo esté funcionando correctamente y que no pueda ser vulnerado fácilmente.

Pruebas de validaciones de los formularios: esta prueba consistió en establecer diferentes escenarios de entradas de usuarios por los cuales puede pasar el formulario mostrado en el módulo de autenticación (ver Figura 23 (b)), se recrearon 8

escenarios diferentes. Los resultados de cada escenario se muestran en la Tabla 9, el símbolo ✓ significa que hubo un inicio de sesión satisfactorio, por el contrario, el símbolo × significa que se produce una notificación de error similar a la mostrada en la Figura 21.

Tabla 9. Resultados de las pruebas de validación de formularios del módulo de autenticación.

Prueba	Respuesta del sistema
Ingreso de un correo con formato inválido	×
Ingreso de una contraseña incorrecta	×
Campos de correo y/o contraseña vacíos	×
Ingreso de un correo no registrado	!
Ingreso de caracteres especiales en el campo de contraseña	✓
Ingreso de caracteres especiales en el campo de correo	✓
Credenciales inválidas en la ventana emergente de autenticación con Google	×
Cerrar la ventana emergente de autenticación de Google	×

De acuerdo con los resultados obtenidos con los diferentes escenarios, se concluye que los formularios del módulo de autenticación presentan un correcto funcionamiento, debido a que no se permite la autenticación del usuario si proporciona información en un formato o credenciales incorrectas. Cabe resaltar que al ingresar caracteres especiales en el campo de correo o contraseña se permite el inicio de sesión, siempre y cuando dichos caracteres hayan sido registrados previamente en el módulo de registro, de no ser así, se generará una notificación de error de correo o contraseña incorrecta. Por último, se observa que en la prueba donde se ingresa un correo no registrado aparece el símbolo (!), este quiere decir que no se genera una notificación de error, como en los demás casos, por el contrario, se genera una notificación de información, como la que se muestra en la Figura 29.

Pruebas de envío del link de recuperación de contraseña: el módulo de autenticación también cuenta con un formulario para la recuperación de contraseña, es importante garantizar el correcto envío de la información, y además, garantizar que el 100 % de las veces el link de recuperación sea enviado correctamente. Esta prueba se dividió en dos, la primera parte consistió en establecer diferentes escenarios de entradas de usuarios por los cuales puede pasar el formulario y la segunda prueba consistió en recuperar las contraseñas de los 15 correos registrados con an-

terioridad en las pruebas del módulo de registro. Los resultados de las pruebas se presentan en la Tabla 10 y la Tabla 11 respectivamente.

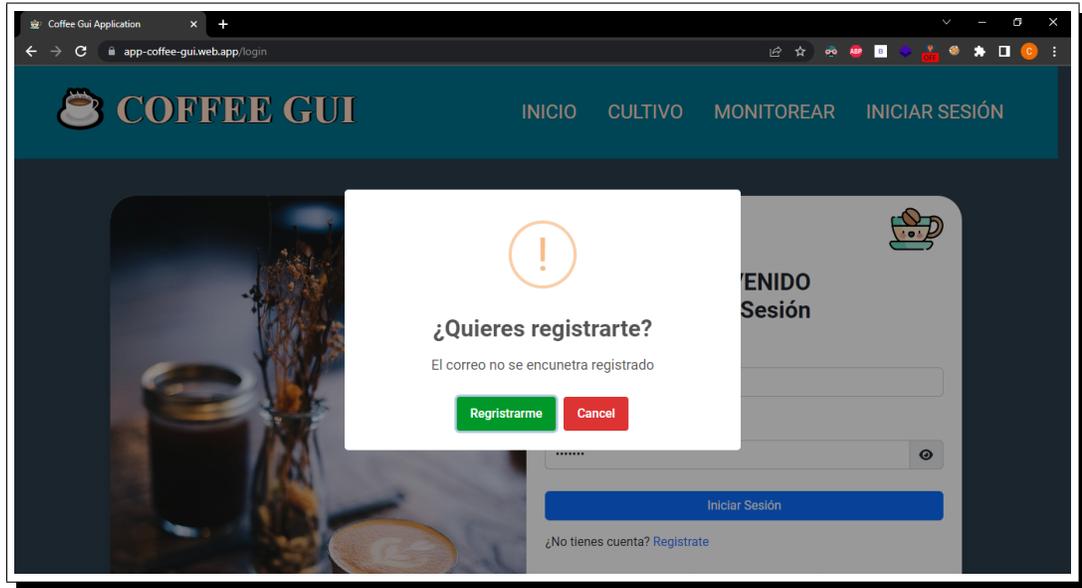


Figura 29. Notificación de información tras haber ingresado un correo no registrado al módulo de autenticación.

Tabla 10. Resultados de las pruebas de validación del formulario para la recuperación de contraseña del módulo de autenticación.

Prueba	Respuesta del sistema
Ingreso de un correo con formato inválido	×
Campos de correo vacío	×
Ingreso de un correo no registrado	!
Ingreso de caracteres especiales en el campo de correo	✓

Tabla 11. Resultados de la prueba de envío del link de recuperación de contraseña para el módulo de autenticación.

	Correos registrados
Envío del link a la bandeja de entrada	3 correos
Envío del link a la bandeja de spam	12 correos
Tiempo de envío menor a un minuto	12 correos
Tiempo de envío entre un minuto y tres minutos	3 correos

Con los resultados obtenidos en los diferentes escenarios, se concluye que el formulario de recuperación de contraseña del módulo de autenticación y el envío del link de recuperación de contraseña presentan un correcto funcionamiento, debido a que no se permite el envío de información en un formato incorrecto, y además el 100 % de las veces el link es enviado correctamente. Cabe resaltar que el comportamiento en las pruebas de la Tabla 10 que tienen los símbolos ✓ y !, es el mismo que el descrito en la anterior prueba “validaciones de los formularios”.

Pruebas de acceso a correos no verificados: esta prueba consistió en intentar autenticarse con 10 correos diferentes, los cuales aún no se encontraban verificados, es decir, el usuario aún no había ingresado al link enviado previamente por el módulo de registro. Como resultado se obtuvo un redireccionamiento hacia la pantalla mostrada en la Figura 22 en los 10 casos probados. Con lo anterior se concluye que el módulo de autenticación presenta un correcto funcionamiento debido a que no permite el ingreso a usuarios no verificados.

3.4.5. Pruebas del módulo de monitoreo

El módulo de monitoreo permite a los usuarios monitorear y obtener información en tiempo real sobre el estado de las variables en el cultivo, lo que les permite tomar decisiones basadas en datos y mantener la estabilidad del cultivo. Con el objetivo de garantizar la calidad del módulo de monitoreo, se diseñaron pruebas específicas para verificar que este módulo funcione de manera efectiva y brinde información precisa y oportuna.

Pruebas de lectura de la base de datos: esta prueba consistió en registrar en la base de datos de medidas descrita en la sección 3.2.2, diferentes valores, en las diferentes variables, y observar el comportamiento de la aplicación web, sin necesidad de actualizar la página (recargar el navegador). Las tarjetas mostradas en la Figura 30 permiten visualizar la última medida registrada junto con su hora y fecha en las diferentes variables. Se realizó el registro de 10 datos desde la consola de Firebase, y como resultado, las tarjetas mostraron correctamente el último dato registrado por cada variable.

De igual manera, en el apartado del módulo de monitoreo, donde se detallan cada una de las variables, las gráficas y las tablas se actualizan en tiempo real a medida que se registran datos. Adicionalmente, los cálculos de promedio, máxima y mínima medida se actualizan en tiempo real, teniendo un acierto total en el 100 % de los registros. Por último, las tablas de datos resaltan correctamente en color rojo aquellas medidas que superan el umbral establecido como se muestra en la Figura 31, lo que indica una correcta lectura de la base de datos de parámetros. Con lo anterior, se concluye que en el módulo de monitoreo se realiza una correcta lectura de la base de datos y se efectúan correctamente los cálculos de medias registradas.



Figura 30. Tarjetas que muestran la última medida registrada en la base de datos de medidas.

El umbral establecido de humedad ambiente es: 72 (%)
 El promedio total de humedad ambiente es: 50.06 (%)
 El valor máximo de humedad ambiente registrado fue de: 99.63 (%)
 El valor mínimo de humedad ambiente registrado fue de: 1.14 (%)

Fecha	Hora	Medida (%)	Nodo (ID)
28/02/2023	06:17 PM	12.75	1
28/02/2023	11:12 AM	10.96	1
27/02/2023	08:27 PM	19.8	1
27/02/2023	11:03 AM	47.73	1
26/02/2023	09:07 PM	6.09	1
24/02/2023	04:48 AM	86.57	1
24/02/2023	02:09 AM	37.22	1
23/02/2023	01:00 PM	60.25	1
22/02/2023	11:51 AM	91.52	1
22/02/2023	09:00 AM	6.42	1

Figura 31. Tabla de datos que muestran los registros de la base de datos de medidas.

Pruebas de validación de los formularios: el módulo de monitoreo permite realizar filtrado por fecha de los registros guardados, lo cual exige la integración de un formulario para que el usuario ingrese el rango de fechas que desea buscar. Esta prueba consistió en establecer diferentes escenarios de entradas de usuarios por los cuales puede pasar el formulario mostrado en el módulo de monitoreo (ver Figura 32), se contempló campos vacíos, fechas incorrectas, orden incorrecto del ingreso

de fechas, entre otros. Los resultados de cada escenario se muestran en la Tabla 12, el símbolo (✓) significa que hubo un correcto filtrado y por ende, se muestran satisfactoriamente los datos, por el contrario, el símbolo (×) significa que se produce una notificación de error similar a la mostrada en la Figura 21. Los resultados obtenidos permiten concluir que el formulario del módulo de monitoreo para filtrado por fecha presenta un correcto funcionamiento. Cabe resaltar que al ingresar una fecha inicial mayor a la fecha final, el sistema organiza dichas fechas para darle un orden lógico. Por último, se observa que en la prueba donde se ingresa un rango de fechas en el que no hay registro de medidas aparece el símbolo (!), este quiere decir que no se genera una notificación de error como en los demás casos, por el contrario, se genera una pantalla de información.

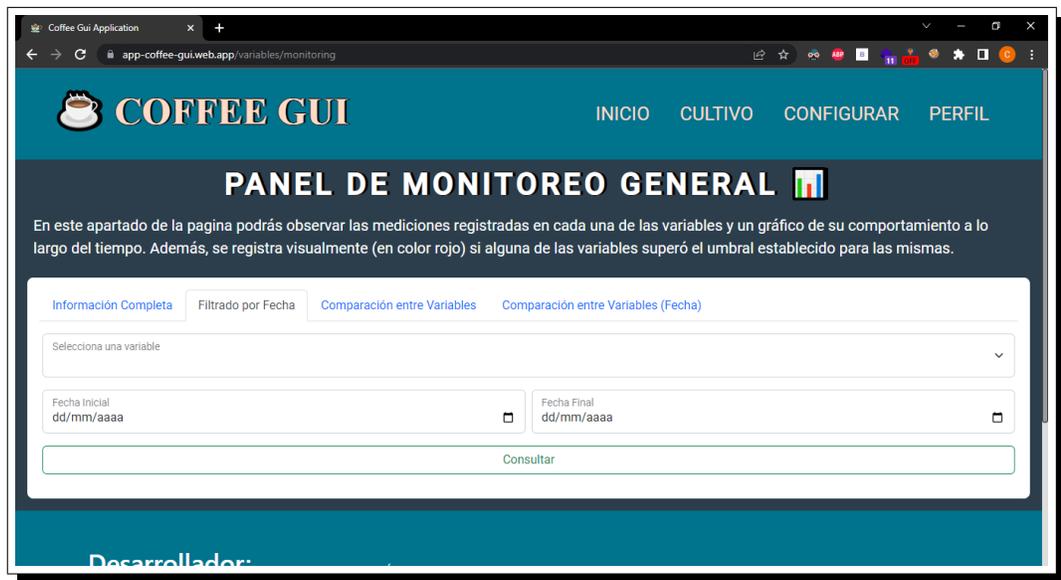


Figura 32. Formulario para el filtrado por fecha del módulo de monitoreo.

Tabla 12. Resultados de las pruebas de validación del formulario para el filtrado por fecha del módulo de monitoreo.

Prueba	Respuesta del sistema
Campo de consulta de fecha inicial o final vacío	×
Campo de variable vacío	×
Ingreso de una fecha incorrecta (día 32 o mes 13)	×
Ingreso de fechas iguales	×
Ingreso de fecha inicial mayor a la fecha final	✓
Ingreso de rango de fechas donde no hay registro de medidas	!

Pruebas de funcionalidad de las gráficas y tablas de datos: el módulo de monitoreo ofrece al usuario gráficas interactivas para visualizar los datos de una manera más dinámica, estas gráficas representan en el eje horizontal el tiempo y en el eje vertical la magnitud de la variable en cuestión. La prueba consistió en registrar información en la base de datos de medidas con diferentes tipos de datos, para observar el comportamiento del sistema. Los resultados de las pruebas se presentan en la Tabla 13. La simbología para interpretar los resultados se describe a continuación: El símbolo ✓ en el eje horizontal significa una correcta visualización de los datos de tiempo, es decir, se ordenan de manera correcta y se permite navegar a través de ellos por medio de una línea de tiempo. En cuanto al eje vertical, significa que hay una correcta visualización de los datos, y además permite el cálculo de promedio, máximas y mínimas medidas. El símbolo ○ en el eje horizontal significa una correcta visualización de los datos de tiempo en cuanto a orden, es decir, se ordenan de manera correcta, pero no se permite navegar a través de ellos por medio de una línea de tiempo. En cuanto al eje vertical significa que hay una correcta visualización de los datos, pero el cálculo de promedio, máximas y mínimas medidas no se hace de manera correcta. El símbolo × en el eje horizontal significa una incorrecta visualización de los datos, es decir, no se ordenan de manera correcta y por ende, no se permite navegar a través de ellos por medio de una línea de tiempo. En cuanto al eje vertical significa que hay una incorrecta interpretación de los datos.

Tabla 13. Resultados de las pruebas a la gráfica interactiva al enviar diferentes tipos de datos.

Tipo de dato	Respuesta del eje horizontal	Respuesta del eje vertical
Entero	○	✓
Flotante	×	✓
Timestamp	✓	×
Cadena de texto	×	○

De acuerdo con los resultados obtenidos, se concluye que para brindar al usuario una correcta experiencia de visualización y navegación es necesario hacer el envío de datos en los formatos timestamp y flotante para los ejes horizontal y vertical respectivamente (tiempo y magnitud de la variable).

3.4.6. Pruebas del módulo de configuraciones

El módulo de configuraciones permite a los usuarios configurar diversos aspectos en la red inalámbrica de sensores, como los umbrales de alerta, la gestión de los nodos, los periodos de muestreo, entre otros. Con el objetivo de garantizar la calidad del módulo de configuraciones, se diseñaron pruebas específicas para verificar

que este módulo funcione de manera efectiva y permita a los usuarios autorizados modificar los parámetros según sus necesidades o preferencias.

Pruebas de acceso al módulo: esta prueba consistió en intentar ingresar al módulo con 10 usuarios diferentes, la mitad de ellos con un rol de invitado, la otra mitad con un rol de administrador. Como resultado, el sistema no permitió el acceso a los cinco usuarios con rol de invitado, por el contrario, permitió en el primer intento el ingreso a cada uno de los cinco usuarios con rol de administrador. Lo anterior, verifica el correcto funcionamiento del módulo, debido a que solo permite a los usuarios administradores (autorizados previamente por el desarrollador) realizar configuraciones en los parámetros de la red inalámbrica de sensores.

Pruebas de validación de formularios: el módulo de configuraciones cuenta con diferentes formularios que permiten modificar parámetros en la red de sensores como umbrales de alerta, gestión de nuevos nodos, entre otros. Esta prueba consistió en ingresar diferentes entradas a los campos mostrados en el módulo de configuraciones (ver Figura 25 (b)). Los resultados de las pruebas se observan en la Tabla 14.

Tabla 14. Resultados de las pruebas de validación de formularios del módulo de configuraciones.

Prueba	Respuesta del sistema
Campos vacíos	×
Dirección MAC en formato inválido	×
Tipo de datos incorrectos	×

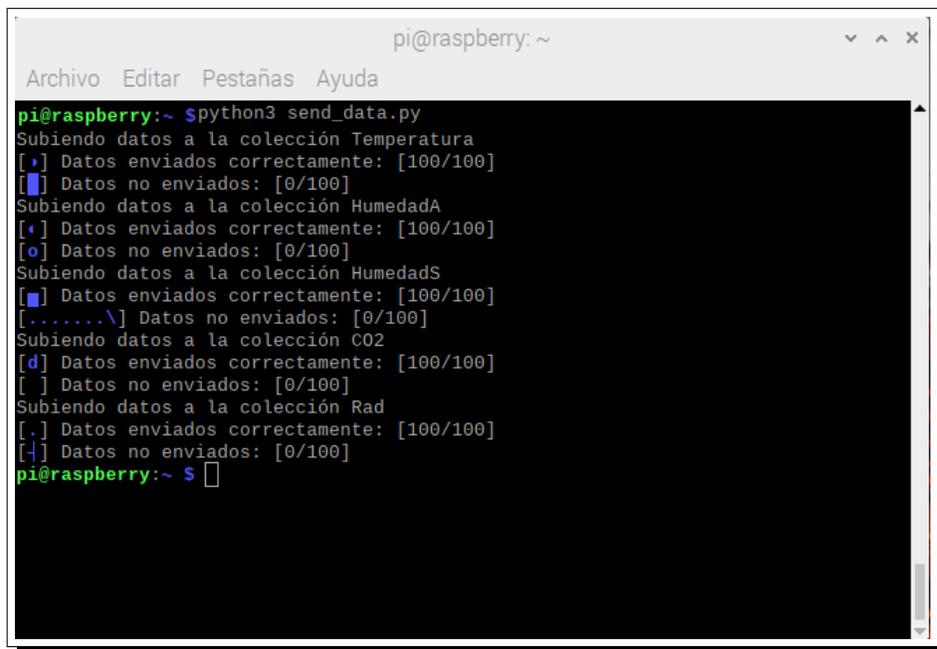
De acuerdo con los resultados obtenidos en los diferentes escenarios, se concluye que los formularios del módulo de configuración presentan un correcto funcionamiento, debido a que en todos los casos se genera una notificación de error, por ende, no se permite el envío de información en un formato incorrecto.

Pruebas de escritura en la colección de configuración de parámetros: esta prueba consistió en realizar 10 modificaciones y observar el comportamiento en la base de datos en cada uno de los parámetros permitidos desde el módulo de configuraciones, es decir, en la modificación de umbrales de medida, eliminación de registros actuales, agregar un nuevo nodo, configurar el estado de un nodo y modificación en el periodo de muestreo. Como resultado se obtuvo un 100 % de eficacia en la escritura de la base de datos, es decir, cada vez que se modificaba un parámetro desde la aplicación web se actualizaba en tiempo real en la base de datos, lo cual en consecuencia hace que el nodo central ejecute las acciones pertinentes en la red de sensores para modificar el parámetro en cuestión.

3.4.7. Pruebas con el nodo central

En este proyecto se optó por utilizar como nodo central una Raspberry Pi debido a las diferentes ventajas que otorga, como por ejemplo: bajo costo, conectividad inalámbrica nativa, amplia variedad de software y alto potencial de procesamiento. Las pruebas realizadas con el nodo central son esenciales para garantizar que el software y la red de sensores funcionen correctamente, promoviendo la gestión efectiva. Estas pruebas permiten concluir si el envío de datos y la configuración remota de parámetros es exitosa.

Prueba de escritura en la base de datos: hasta el momento, todas las pruebas de escritura en la base de datos se habían efectuado desde el CLI de Firebase, a continuación, se describe la prueba de envío de datos realizada desde el nodo central. Esta prueba consistió en realizar el envío de 100 datos aleatorios uno tras otro desde la Raspberry Pi a través de un script en Python. El script importaba las bibliotecas necesarias como pytz, firebase_admin, credentials, firestore y datetime, para posteriormente conectarse a la base de datos de Firebase usando las credenciales proporcionadas en un archivo “credentials.json” obtenido desde el sitio web de Firebase. Finalmente, se creaba un diccionario llamado “data” con la estructura mostrada en la Figura 7 y se enviaba la información a una colección de la base de datos de Firestore obtenida con anterioridad. Los resultados de envíos exitosos o fallidos se reportaron por consola como se muestra en la Figura 33.



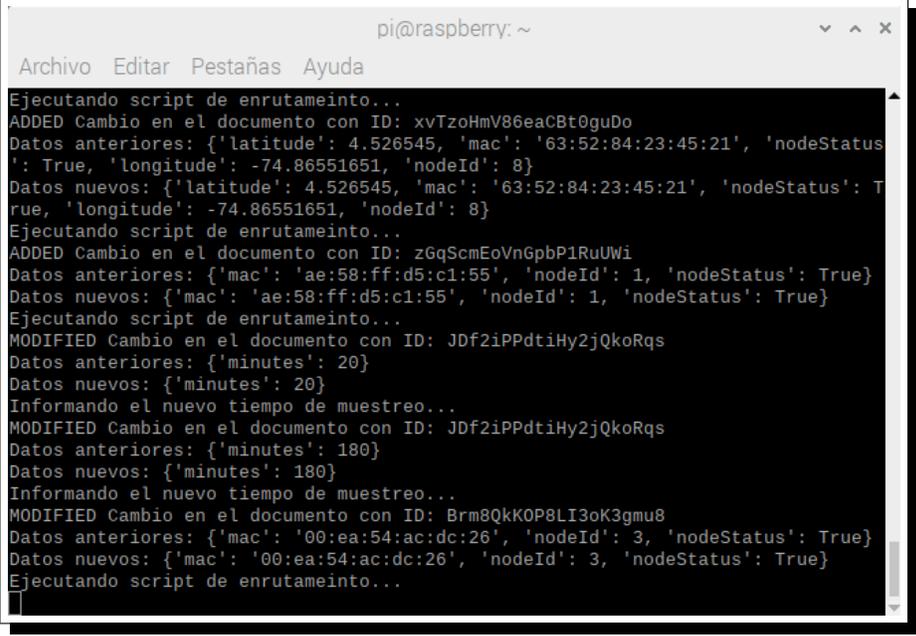
```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~$ python3 send_data.py  
Subiendo datos a la colección Temperatura  
[.] Datos enviados correctamente: [100/100]  
[.] Datos no enviados: [0/100]  
Subiendo datos a la colección HumedadA  
[.] Datos enviados correctamente: [100/100]  
[.] Datos no enviados: [0/100]  
Subiendo datos a la colección HumedadS  
[.] Datos enviados correctamente: [100/100]  
[.....\] Datos no enviados: [0/100]  
Subiendo datos a la colección C02  
[d] Datos enviados correctamente: [100/100]  
[ ] Datos no enviados: [0/100]  
Subiendo datos a la colección Rad  
[.] Datos enviados correctamente: [100/100]  
[+] Datos no enviados: [0/100]  
pi@raspberrypi:~$
```

Figura 33. Envío de datos hacia la base de datos Firestore desde la Raspberry Pi.

Esta prueba se repitió para cada una de las colecciones que almacena los datos

de las variables, es decir, hubo un total de 500 datos enviados. Como resultado se obtuvo un 100% de eficacia en cada una de las colecciones. La integridad de los datos se comprobó por medio de la aplicación web y el dashboard de Firebase. Con lo anterior, se demuestra que el envío remoto de datos desde el nodo central hacia la base de datos funciona de manera correcta.

Pruebas de gestión remota de parámetros: esta prueba consistió en crear dos “listeners” en el nodo central a través de un script de Python para las colecciones de nodos y configuración de parámetros descritas en la sección 3.2.2. El script importaba las bibliotecas necesarias, incluyendo signal, sys y firebase_admin, esta última utilizada para autenticar y conectarse a la base de datos de Firestore por medio del archivo “credentials.json” descrito en la anterior prueba. Posteriormente, se definieron dos funciones de devolución de llamada (callback) que se ejecutaban cuando se registraba algún cambio en alguna de las colecciones. La ejecución de una u otra función dependía de la colección la cual se registraba el cambio. Si el cambio se registraba en la colección de nodos, se ejecutaba un script, que gestionaba el enrutamiento en la red de sensores para agregar el nuevo nodo. Por el contrario, si el cambio se registraba en la colección de configuración de parámetros, se ejecuta un script que informa a los nodos el nuevo periodo de muestreo. La ejecución de scripts se reporta por consola como se muestra en la Figura 34.



```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
Ejecutando script de enrutameinto...
ADDED Cambio en el documento con ID: xVTzoHmV86eaCBt0guDo
Datos anteriores: {'latitude': 4.526545, 'mac': '63:52:84:23:45:21', 'nodeStatus': True, 'longitude': -74.86551651, 'nodeId': 8}
Datos nuevos: {'latitude': 4.526545, 'mac': '63:52:84:23:45:21', 'nodeStatus': True, 'longitude': -74.86551651, 'nodeId': 8}
Ejecutando script de enrutameinto...
ADDED Cambio en el documento con ID: zGqScmEoVnGpbP1RuUWi
Datos anteriores: {'mac': 'ae:58:ff:d5:c1:55', 'nodeId': 1, 'nodeStatus': True}
Datos nuevos: {'mac': 'ae:58:ff:d5:c1:55', 'nodeId': 1, 'nodeStatus': True}
Ejecutando script de enrutameinto...
MODIFIED Cambio en el documento con ID: JDF2iPPdtiHy2jQkoRqs
Datos anteriores: {'minutes': 20}
Datos nuevos: {'minutes': 20}
Informando el nuevo tiempo de muestreo...
MODIFIED Cambio en el documento con ID: JDF2iPPdtiHy2jQkoRqs
Datos anteriores: {'minutes': 180}
Datos nuevos: {'minutes': 180}
Informando el nuevo tiempo de muestreo...
MODIFIED Cambio en el documento con ID: Brm8QkKOP8LI3oK3gmu8
Datos anteriores: {'mac': '00:ea:54:ac:dc:26', 'nodeId': 3, 'nodeStatus': True}
Datos nuevos: {'mac': '00:ea:54:ac:dc:26', 'nodeId': 3, 'nodeStatus': True}
Ejecutando script de enrutameinto...
```

Figura 34. Ejecucion de scripts en la Raspberry Pi al detectar un cambio en alguna de las colecciones de nodos o configuracion de parámetros.

Esta prueba se repitió cinco veces para cada una de las colecciones, es decir, se realizaron operaciones de escritura en la base de datos desde la aplicación web en

cada una de las colecciones. Como resultado se obtuvo un 100% de eficacia para cada una de las colecciones, en cada escritura se ejecutaba el script asignado al listener de la colección correspondiente. Lo anterior, demuestra que la modificación de parámetros de manera remota desde la aplicación hacia el nodo central se puede efectuar de manera correcta.

4. ANÁLISIS DE RESULTADOS Y DISCUSIÓN

En esta sección se abordan detalladamente los logros y los desafíos encontrados en la implementación de la interfaz gráfica para la red inalámbrica de sensores en los cultivos de café.

El análisis presentado en la sección 3.1 permitió identificar los posibles beneficios que podría traer la implementación de una aplicación web para el monitoreo de variables climáticas. Estos beneficios incluyen: la fácil accesibilidad a la aplicación desde cualquier dispositivo con un navegador web. Esto significa que el usuario final puede acceder a la información de monitoreo desde cualquier lugar y en cualquier momento, lo que podría aumentar la eficiencia y la velocidad en la toma de decisiones. El monitoreo en tiempo real de las variables climáticas que permite al usuario final mayor velocidad en la recopilación y análisis de datos recolectados. El acceso remoto desde cualquier parte del mundo con una conexión a internet, esto permite al usuario final monitorear y analizar los datos de la red de sensores desde cualquier lugar del mundo, lo que aumenta la eficiencia y la capacidad de respuesta del sistema. Por otra parte, en función de las condiciones generales que debería satisfacer el sistema de adquisición y gestión de datos, se determinaron las herramientas, tecnologías y servicios necesarios para un correcto despliegue de la aplicación. Esto incluyó diferentes paquetes, bibliotecas, frameworks y librerías open source, así como la estructuración del sistema en diferentes módulos para una mayor facilidad en la navegación.

Posteriormente, en la sección de diseño 3.2 se establecieron de manera precisa los requerimientos de la aplicación, dichos requerimientos se dispusieron en función al previo análisis realizado y tomando como referencia el documento del estándar IEEE 830. Este define parámetros como: los requisitos de software, los objetivos y necesidades del software, la descripción de los usuarios y sus tareas, los requisitos funcionales y no funcionales, entre otros. Una vez definidos los requerimientos del sistema se procedió a realizar el modelamiento de la base de datos, este proceso resulta fundamental debido a que permite la representación de la estructura y la organización de la información que será utilizada por la aplicación. Este proceso incluyó la definición de colecciones y campos usados para almacenar los datos, las relaciones entre colecciones y la definición de las restricciones de integridad que garantizarán la consistencia de los datos. Por último, planteó la estructura general del sistema, allí se determinó como los usuarios, la red de sensores y el sistema interactuaban entre sí.

Una vez que se completó el análisis y diseño del sistema, se procedió a su implementación. Para ello, se estructuró una página web que se divide en diferentes módulos, cada uno de los cuales cumple una función específica e incluyen información, autenticación, monitoreo y configuración. Para garantizar una implementación correcta y sin problemas, se utilizaron diferentes paquetes, bibliotecas y librerías disponibles en internet. Estos recursos permitieron agregar algunas funciones adicionales al sistema y resolver problemas que surgieron durante el proceso de implementación. Además, se optimizó el sistema para ser visualizado desde cualquier tipo de dispositivo, incluyendo computadoras, tabletas y teléfonos móviles. Se aseguró que el tiempo de respuesta fuera lo más bajo posible para garantizar una experiencia de usuario fluida y eficiente. La estructuración del sistema en diferentes módulos permite que los usuarios puedan acceder fácilmente a la información que necesitan, en función de sus necesidades y permisos de acceso. El módulo de autenticación garantiza que solo los usuarios autorizados puedan acceder al sistema y realizar cambios en la configuración de la red de sensores. El módulo de monitoreo permite el seguimiento en tiempo real de las variables climáticas, lo que permite tomar decisiones rápidas y eficaces. El módulo de configuración remota, como se mencionó anteriormente, permite ajustar los parámetros de la red de sensores desde cualquier ubicación.

Finalmente, se procedió a realizar pruebas específicas para cada uno de los módulos y para el sistema en conjunto. Estas pruebas tuvieron como objetivo comprobar el correcto funcionamiento del sistema en diferentes escenarios y situaciones. Entre las pruebas realizadas se encontraron la comprobación de formularios, la visualización de la aplicación en diferentes dispositivos, el envío, lectura y escritura en la base de datos, la configuración remota de parámetros, entre otras. Cada una de estas pruebas fue diseñada para detectar posibles errores y comportamientos inesperados del sistema, y así garantizar su correcto funcionamiento en diferentes situaciones. Los resultados obtenidos fueron satisfactorios, lo que permitió determinar el correcto funcionamiento del sistema. Cualquier error o comportamiento inesperado encontrado durante las pruebas fue corregido y se volvieron a realizar las pruebas hasta obtener los resultados deseados.

5. CONCLUSIONES Y TRABAJOS FUTUROS

En esta sección se exponen las conclusiones obtenidas a través de la ejecución de este proyecto. Se detallan los resultados y se destacan las contribuciones más relevantes en cuanto al desarrollo del sistema de adquisición y gestión de datos para una red inalámbrica de sensores. Además, se presenta una sección dedicada a posibles trabajos futuros que podrían ser realizados a partir de los resultados obtenidos en este proyecto.

5.1. CONCLUSIONES

El desarrollo de este trabajo de grado permitió satisfacer las necesidades finales del usuario, lo anterior, mediante el análisis, diseño e implementación del sistema de adquisición y gestión de datos. A continuación, se resaltan los logros obtenidos:

El sistema consiguió centralizar la información proveniente de la red inalámbrica de sensores, de tal manera que permite conocer en tiempo real factores climatológicos críticos (temperatura, humedad, CO₂ y radiación solar) que puedan afectar a los cultivos de café, adicionalmente, al centralizar los datos, se simplifica la tarea de acceder a la información de manera rápida y eficiente. Lo anterior, puede resultar especialmente útil e importante cuando se desee trabajar con grandes cantidades de datos para su análisis.

El sistema fue desarrollado en un entorno 100% web, lo que facilitó a los usuarios acceder a la información desde cualquier dispositivo o sistema operativo con conexión estable a internet y un navegador web compatible con la aplicación. Del mismo modo, se realizó un diseño de interfaz gráfica intuitivo y de fácil uso para cualquier tipo de usuario que esté interesado en conocer el estado del cultivo. La interfaz muestra la información de cada una de las variables climáticas por medio de tablas de datos (Figura 31), que pueden ser exportadas en formato Excel, además, se muestra la información a través de gráficas interactivas (Figura 26 (b)), que se actualizan en tiempo real a medida que el nodo central de la red envía información a la base de datos centralizada en la nube. Como resultado, los usuarios pueden acceder e interpretar la información desde ubicaciones remotas y desde cualquier dispositivo.

El sistema administró de manera correcta la asignación de roles como se observa en la Tabla 6, garantizando que cada usuario tenga acceso solo a la información y

las funciones necesarias para desempeñar su trabajo, por tal razón, se protegió la privacidad de los datos y garantizó la seguridad de la red de sensores permitiendo la configuración de parámetros únicamente a los usuarios administradores del cultivo.

El sistema permitió la modificación remota de parámetros en la red inalámbrica de sensores, otorgando al usuario una herramienta de apoyo para la gestión de la red inalámbrica de sensores, permitiendo cambiar umbrales de alerta para cada una de las variables, añadir nuevos nodos a la red de sensores, modificar periodos de muestreo, eliminar los registros de cada variable y alterar el estado (encendido o apagado) de un nodo en particular. La configuración remota de parámetros puede ayudar a los usuarios más especializados a ajustar la red de sensores en tiempo real para optimizar el proceso de análisis de datos, por ejemplo, modificando el periodo de muestreo de los nodos para obtener una mayor cantidad de información en un tiempo dado o apagando ciertos nodos que no esten siendo analizados en un momento particular.

Finalmente, al lograr un 100 % de eficacia en las pruebas preliminares realizadas en la sección 3.4.7 de envío de datos desde el nodo central de la red, se puede concluir que el sistema funciona correctamente y está diseñado para proporcionar información confiable, detallada y oportuna acerca de las diferentes variables climáticas. Con esta información, los expertos pueden tomar decisiones basadas en datos precisos y actualizados en tiempo real, que les permite implementar acciones correctivas de manera más rápida y eficiente en respuesta a situaciones de emergencia o para prevenir situaciones adversas en el futuro.

Por otra parte, el uso de los frameworks Angular y Bootstrap facilitaron el desarrollo e implementación de la aplicación web. Algunos aspectos que se pueden resaltar con el análisis realizado en la sección 3.1 son: la eficiencia, debido a que estos frameworks proporcionan una estructura y un conjunto de herramientas predefinidas para el desarrollo de aplicaciones, lo que permite ahorrar tiempo y esfuerzo. La estética, ya que Bootstrap está enfocado a la creación de interfaces de usuario atractivas y responsivas. La facilidad de uso de ambos frameworks por la accesibilidad para los desarrolladores de diferentes niveles de experiencia y a su curva de aprendizaje relativamente corta. La escalabilidad, puesto que estos frameworks están diseñados para escalar con la aplicación, lo que significa que pueden manejar fácilmente aplicaciones web de mayor tamaño y complejidad.

Del mismo modo, el uso de la plataforma de Google Firebase, fue un pilar fundamental para la construcción del sistema. Esta plataforma de desarrollo fue el puente entre la red inalámbrica de sensores y la aplicación web. El servicio de autenticación fácil y seguro que se proporciona permite integrar funciones de inicio de sesión con una gran variedad de proveedores, incluidos Google, Facebook, Twitter y más, o crear un sistema de registro clásico. El servicio de hosting permitió desplegar la

aplicación en un servidor público, y así, hacerla accesible desde cualquier parte del mundo con una conexión a internet. La base de datos en tiempo real y 100 % desplegada en la nube permitió la comunicación y almacenamiento de datos del nodo central, el cual recibía las medidas registradas por los nodos sensores. Asimismo, a través de la integración que tiene Firebase con diferentes lenguajes de programación fue posible la creación de diferentes *listeners* mediante el uso de Python en el nodo central, que permitieron configurar parámetros como la activación o desactivación de un nodo, el periodo de muestreo o los umbrales de alerta de la red de sensores de manera remota.

En definitiva, la elaboración de este proyecto de grado proporciona una primera versión del sistema para futuros desarrollos que buscan optimizar y automatizar los procesos agrícolas, sin importar el tipo de cultivo mediante redes inalámbricas de sensores. De igual manera, este proyecto sirve como punto de partida para futuros investigadores y desarrolladores que deseen diseñar aplicaciones similares basadas en redes de sensores.

5.2. PERSPECTIVAS DE TRABAJO FUTURO

En esta sección se plantean diferentes posibilidades de trabajos futuros a partir del desarrollo actual del proyecto. Estas propuestas buscan mejorar el sistema y utilizar este como base para crear sistemas más complejos que puedan ser aplicados en diferentes entornos.

A medida que la tecnología de redes inalámbricas de sensores continúa avanzando, es crucial que los sistemas de adquisición y gestión de datos adscritos a estas también evolucionen para aprovechar al máximo el potencial de las redes. Una de las áreas en las que se podría trabajar para mejorar el sistema de monitoreo es aumentar el nivel de precisión en las medidas. Esto podría lograrse mediante el empleo de sensores más precisos y la implementación de algoritmos avanzados de procesamiento de señales. De esta manera, se podría reducir aún más el margen de error en las mediciones y obtener información más precisa y confiable sobre las variables monitoreadas. La mejora en la precisión de las medidas tendría un impacto positivo en la agricultura, obteniendo por ejemplo: una mejor comprensión de las necesidades de agua y nutrientes de las plantas o detección temprana de situaciones de estrés en las plantas, como enfermedades o condiciones climáticas adversas.

Por otra parte, la integración de tecnologías de seguridad avanzada en el monitoreo de cultivos traería consigo un impacto positivo en la protección de la red de posibles ataques externos y garantizaría la integridad de los datos recopilados. Lo anterior resulta de vital importancia, ya que los datos recopilados en el monitoreo de los cultivos son valiosos y críticos para la toma de decisiones por parte de los agricultores.

La incorporación de estas tecnologías permitiría reducir el riesgo de que terceros no autorizados accedan a la información y puedan manipularla o alterarla, lo que podría tener consecuencias graves en la producción de cultivos.

Finalmente, otra posible mejora en la interfaz gráfica sería la implementación de sugerencias en tiempo real para el cultivo, basadas en los datos recopilados por los sensores. Por ejemplo, la interfaz podría proporcionar al usuario recomendaciones sobre el riego de las plantas, la cantidad apropiada de fertilizante a aplicar, el momento oportuno para podar las plantas y la manera de podarlas adecuadamente para maximizar la producción de café y la salud de las plantas, todo basado en los datos que se estén recopilando en tiempo real.

BIBLIOGRAFÍA

- [1] W. Aunsa-Ard y T. Kerdcharoen, "Electronic Nose for Analysis of Coffee Beans Obtained from Different Altitudes and Origin," en *2022 14th International Conference on Knowledge and Smart Technology (KST)*, 2022, págs. 147-151. DOI: 10.1109/KST53302.2022.9729071.
- [2] R. K. Kodali, S. Soratkal y L. Boppana, "WSN in coffee cultivation," en *2016 International Conference on Computing, Communication and Automation (ICCA)*, 2016, págs. 661-666. DOI: 10.1109/ICCA.2016.7813804.
- [3] J. Pérez, "El monitoreo: herramienta básica en los programas MIP/MIE," *Revista de Protección Vegetal*, vol. 36, n.º 2, págs. 35-42, 2021. [En línea]. Disponible en: <https://www.intagri.com/articulos/fitosanidad/el-monitoreo-herramienta-basica-en-los-programas-mip-mie#:~:text=El%20monitoreo%20consiste%20en%20revisar, en%20la%20producci%C3%B3n%20del%20cultivo..>
- [4] F. Scaramuzza, A. Méndez, D. Villarroel, M. Bragachini y J. Vélez, "Monitoreo de rendimiento," [En línea]. Disponible en: https://inta.gob.ar/sites/default/files/script-tmp-inta_monitoreo_de_rendimiento_tomar_decisiones_con_i.pdf.
- [5] V. S. Castillo, Y. A. Pizo, A. G. Astudillo y C. Gómez, "Cambio climático y café (Coffea arábica) en Acevedo, Huila: una lectura desde sus cultivadores," *I+ D Revista de investigaciones*, vol. 12, n.º 2, págs. 59-69, 2018. [En línea]. Disponible en: <http://www.udi.edu.co/revistainvestigaciones/index.php/ID/article/view/187/184>.
- [6] M. Cañón Hernández, "Determinación de las necesidades hídricas en un cultivo de café, en etapa productiva, con base en el conocimiento del balance hídrico," B.S. thesis, Universidad de Manizales, 2019, pág. 76. [En línea]. Disponible en: <https://ridum.umanizales.edu.co/xmlui/handle/20.500.12746/3627>.
- [7] D. Murillo Carvajal y M. C. Salazar Carrillo, "Oportunidades de los productores de cafés especiales en Risaralda en relación con las certificaciones internacionales para exportar," 2019. [En línea]. Disponible en: <https://hdl.handle.net/10901/17810>.
- [8] C. A. Valencia Nieto, O. E. Ávila Bernal et al., "Diseño de un sistema de automatización SCADA de una planta trilladora de café pergamino, caso de estudio cooperativa de caficultores del Cauca," B.S. thesis, Universidad Piloto de Colombia, 2017. [En línea]. Disponible en: <http://repository.unipiloto.edu.co/handle/20.500.12277/1073>.

- [9] M. A. D. Manrique y C. A. C. Vargas, "Estudio de la estructura del agroecosistema cafetero mediante el diagrama de ciclos causales. Estudio de caso (Cundinamarca, Colombia)," *Ager: Revista de estudios sobre despoblación y desarrollo rural Journal of depopulation and rural development studies*, n.º 28, págs. 135-160, 2020. DOI: 10.4422/ager.2019.08.
- [10] L. A. Rodríguez Larramendi, F. Guevara Hernández, H. Gómez Castro, M. Fonseca Flores, J. C. Gómez Castañeda y R. Pinto Ruiz, "Anatomía foliar relacionada con la ruta fotosintética en árboles de café (*Coffea arabica* L., var. Caturra Rojo) expuestos a diferentes niveles de radiación solar en la Sierra Maestra, Granma, Cuba," *Acta Agronómica*, vol. 65, n.º 3, págs. 248-254, 2016. DOI: 10.15446/acag.v65n3.46731.
- [11] H. d. P. Carvalho, B. d. Melo, R. T. Atarassi, R. d. Camargo, C. R. da Silva, M. Moraes et al., "Development weighing lysimeters for coffee culture.," *Bioscience Journal*, vol. 29, n.º 6, págs. 1750-1760, 2013.
- [12] G. A. Sánchez-Hernández, H. E. Júnez-Ferreira, W. Alvarez-Robles, J. González-Trinidad, C. Bautista-Capetillo y G. d. S. Herrera-Zamarrón, "Optimización del monitoreo del nivel del agua subterránea para una frecuencia fija," *Tecnología y ciencias del agua*, vol. 8, n.º 5, págs. 19-38, 2017. DOI: 10.24850/j-tyca-2017-05-02.
- [13] H. J. Andrade y P. C. Zapata Arango, "Desempeño ecofisiológico de café (*coffea arabica* l.) cv. castillo a la sombra en San Juan de Rioseco, Colombia," *Revista de Investigación Agraria y Ambiental*, vol. 11, n.º 1, págs. 15-27, dic. de 2019. DOI: 10.22490/21456453.2915. [En línea]. Disponible en: <https://hemeroteca.unad.edu.co/index.php/riaa/article/view/2915>.
- [14] V. A. Saldarriaga Lucas, J. E. Chavarría Párraga, Á. M. Guzman Cedeño y N. L. Tarazona Meza, "Efecto de las variables climáticas sobre la fluctuación del nivel freático en suelos fluvisoles del valle del río carrizal," *Ciencia y Tecnología*, vol. 13, n.º 2, págs. 55-62, dic. de 2020. DOI: 10.18779/cyt.v13i2.393. [En línea]. Disponible en: <https://revistas.uteq.edu.ec/index.php/cyt/article/view/393>.
- [15] J. E. Chaparro Mesa, N. Barrera Lombana y F. A. León Socha, "Módulo Terminal Remoto, para la adquisición de datos, monitoreo y control de procesos Agroindustriales-AgricultIC," *Ingeniare. Revista chilena de ingeniería*, vol. 29, n.º 2, págs. 245-264, 2021. DOI: 10.4067/S0718-33052021000200245.
- [16] Instituto Colombiano Agropecuario (ICA), *Resolución 030021 de 2017*, [consultado: Mar. 04,2023], 2017. [En línea]. Disponible en: <https://www.ica.gov.co/getattachment/9d8fe0fa-66d2-4feb-9513-cbba30dc4844/2017r30021.aspx>.
- [17] Instituto Colombiano Agropecuario (ICA), *Resolución 082394 de 2020*, [consultado: Mar. 04,2023], 2017. [En línea]. Disponible en: <https://www.ica.gov.co/getattachment/446ac25a-0fd7-4fd8-ae9f-2e50f0047c8b/2020R82394.aspx>.

- [18] S. Cirani, G. Ferrari, M. Picone y L. Veltri, *Internet of Things: Architectures, Protocols and Standards*. Wiley, 2018, ISBN: 9781119359685. [En línea]. Disponible en: <https://books.google.com.co/books?id=cg1rDwAAQBAJ>.
- [19] J. Gubbi, R. Buyya, S. Marusic y M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, n.º 7, págs. 1645-1660, 2013, ISSN: 0167-739X. DOI: <https://doi.org/10.1016/j.future.2013.01.010>.
- [20] A. K. Domínguez Pasquel, "Diseño de una red de sensores inalámbricos LP-WAN para el monitoreo de cultivos y materia orgánica en la granja experimental La Pradera de la Universidad Técnica del Norte," B.S. thesis, Universidad Técnica del Norte, 2020. [En línea]. Disponible en: <http://repositorio.utn.edu.ec/handle/123456789/10297>.
- [21] C. D. Cadena Chavarro, L. C. Pulido Feo et al., *WSN para el monitoreo continuo de variables ambientales*, 2021. [En línea]. Disponible en: <http://hdl.handle.net/10554/62006>.
- [22] Ubidots, Inc., *Ubidots*, <https://ubidots.com/>, (consultado Dic. 08,2022), 2021.
- [23] I. Santana Ching, A. J. Cárdenas Rivero, R. Sosa López y J. A. Portal Díaz, "Monitoreo de parámetros ambientales en casas de cultivo a través de aplicación IoT," *Revista Cubana de Transformación Digital*, vol. 1, n.º 1, págs. 53-62, abr. de 2020. [En línea]. Disponible en: <https://rctd.uic.cu/rctd/article/view/46>.
- [24] T. M. Piamba Mamian, L. E. Zambrano, L. A. Montaña Rúales y F. A. Rojas Gonzales, "Implementación de un sistema de monitoreo IoT aplicado a una piscicultura de trucha," *Informador Técnico*, vol. 85, n.º 1, págs. 3-19, oct. de 2020. DOI: 10.23850/22565035.2937.
- [25] T. Yokotani e Y. Sasaki, "Comparison with HTTP and MQTT on required network resources for IoT," en *2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, 2016, págs. 1-6. DOI: 10.1109/ICCEREC.2016.7814989.
- [26] M. K. Plazas Olaya, "Adquisición de datos para el monitoreo remoto de variables de un cultivo vertical a través de una plataforma IoT," Tesis de maestría., Universidad de Antioquia, 2020. [En línea]. Disponible en: <https://hdl.handle.net/10495/17203>.
- [27] H. D. Barrionuevo Apaza, "Implementación de un sistema de monitoreo remoto y de control para la mayor eficiencia en un cultivo de tomates cherry por hidroponía en sustrato preparado, en el cono sur de Arequipa," Tesis de maestría., Universidad Nacional de San Agustín de Arequipa, 2020. [En línea]. Disponible en: <http://hdl.handle.net/20.500.12773/12292>.

- [28] M. C. Castrillón Gómez y C. C. Osorio Rodríguez, “Monitoreo de temperatura y humedad relativa en un laboratorio de experimentación de cannabis usando tecnologías de IoT,” 2021. [En línea]. Disponible en: <https://repositorio.uco.edu.co/jspui/handle/20.500.13064/1408>.
- [29] L. M. Mosquera Meléndrez y C. D. Cevallos Rojas, “Diseño e implementación de un Prototipo IoT para el monitoreo de parámetros ambientales aplicados al cultivo de arroz utilizando ESP32 y Thingspeak,” B.S. thesis, Universidad Politécnica Salesiana, 2022. [En línea]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/22884>.
- [30] ThingSpeak, Inc., *ThingSpeak*, <https://thingspeak.com/>, (consultado Dic. 08,2022), 2021.
- [31] A. D. Arteaga Quico y L. R. Wong Portllo, “Framework for Monitoring the Temperature of Aquaculture Crops based on IoT,” *DYNA*, vol. 88, n.º 218, págs. 239-246, jul. de 2021. DOI: 10.15446/dyna.v88n218.90626.
- [32] A. Baloria Reyes y D. Biosca Rojas, “Sistema IoT para monitoreo y control remoto del proceso de curado de tabaco de alto rendimiento,” *Tono, Revista Técnica de la Empresa de Telecomunicaciones de Cuba S.A*, vol. 18, n.º 1, págs. 9-20, ago. de 2022. [En línea]. Disponible en: <http://www.revistatonoetecsa.cu/index.php/tono/article/view/418>.
- [33] S. Berrios Gómez, “Diseño de un sistema IoT para el monitoreo y control del cultivo de lechugas en un invernadero,” 2022. [En línea]. Disponible en: <https://repositorio.upt.edu.pe/handle/20.500.12969/2362>.
- [34] A. Basco, G. Beliz, D. Coatz y P. Garnero, *Industria 4.0: Fabricando el Futuro* (Monografía del BID). Inter-American Development Bank, 2018. [En línea]. Disponible en: <https://books.google.com.co/books?id=geiGDwAAQBAJ>.
- [35] C. Raghavendra, K. Sivalingam y T. Znati, *Wireless Sensor Networks* (Ercoftac Series). Springer US, 2006, ISBN: 9781402078842. [En línea]. Disponible en: <https://books.google.com.co/books?id=4zyDBwAAQBAJ>.
- [36] J. I. Vega-Luna, M. A. Lagos-Acosta y G. Salgado-Guzmán, “Monitoreo de concentración de monóxido de carbono usando tecnología Long-Range,” *Ingenius. Revista de Ciencia y Tecnología*, n.º 18, págs. 73-83, 2017. DOI: 10.17163/ings.n18.2017.09.
- [37] *ECMA-262: ECMAScript Language Specification*, [consultado: Feb. 25,2023], 2020. [En línea]. Disponible en: <https://262.ecma-international.org/11.0/>.
- [38] N. Prusty, *Learning ECMAScript 6* (Community experience distilled). Packt Publishing, 2015, ISBN: 9781785884443. [En línea]. Disponible en: <https://books.google.com.co/books?id=velsawEACAAJ>.
- [39] E. Saks, “JavaScript Frameworks: Angular vs React vs Vue.,” 2019. [En línea]. Disponible en: <https://www.theseus.fi/bitstream/handle/10024/261970/Thesis-Elar-Saks.pdf>.

- [40] J. R. M. Ríos, N. M. L. Mora, M. P. Z. Ordóñez y E. L. L. Sojos, “Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python,” *Revista latinoamericana de Ingeniería de Software*, vol. 4, n.º 4, págs. 201-207, 2016.
- [41] M. Reddy, *API Design for C++*. Elsevier Science, 2011, ISBN: 9780123850041. [En línea]. Disponible en: <https://books.google.com.co/books?id=IY29LylT85wC>.
- [42] B. R. Pressman Roger S.; Maxim, *Ingeniería de software. Un enfoque práctico*. McGraw-Hill Interamericana, 2021, ISBN: 9781456284756. [En línea]. Disponible en: <https://www-ebooks7-24-com.ucundinamarca.basesdedatosezproxy.com/?il=16414>.
- [43] V. Ramírez, A. Jaramillo y J. Arcila, “Factores climáticos que intervienen en la producción del café en Colombia,” en vol. 1, Cenicafé, 2013, págs. 205-237. [En línea]. Disponible en: <https://biblioteca.cenicafe.org/handle/10778/4322>.
- [44] F. FARFAN, “Aumentando la resiliencia a eventos climáticos extremos en el sector cafetero colombiano: Percepción de los caficultores de los municipios de Salamina (Caldas) Santuario y Balboa (Risaralda) frente a la variabilidad climática,” *inf. téc.*, 2017. [En línea]. Disponible en: <https://biblioteca.cenicafe.org/handle/10778/961>.
- [45] V. Ramírez, A. Jaramillo y A. Peña, “Gestión del riesgo agroclimático: vulnerabilidad y capacidad de adaptación del sistema de producción de café,” en vol. 1, Cenicafé, 2013, págs. 91-114. [En línea]. Disponible en: <https://biblioteca.cenicafe.org/handle/10778/4319>.
- [46] J. Hoyos, J. P. Carvajal-Bohórquez, S. Salazar, N. Sarmiento-Herrera y J. C. García-López, “Herramientas agroclimáticas,” en Cenicafé, 2019, págs. 32-51. DOI: https://doi.org/10.38141/10791/0005_2.
- [47] K. A. Hincapié-Velásquez, D. Orozco-Jaramillo, L. A. Á. Monsalve, P. B. Machado, S. M. M. López y J. C. García-López, “Plataforma Agroclimática Cafetera agroclima. cenicafe.org,” *Avances Técnicos Cenicafé*, vol. 502, págs. 1-8, 2019. DOI: <https://doi.org/10.38141/10779/0502>.
- [48] K. S. Zurita Núñez, “Frameworks para el desarrollo de aplicaciones móviles multiplataforma compiladas de forma nativa—estudio comparativo y ejemplos prácticos,” B.S. thesis, PUCE-Quito, 2021. [En línea]. Disponible en: <http://repositorio.puce.edu.ec/handle/22000/19736>.
- [49] M. Sanz et al., “Programación web en el entorno cliente,” en Editorial RA-MA, 2016, págs. 22-23. [En línea]. Disponible en: <https://books.google.com.co/books?id=9I2fDwAAQBAJ>.
- [50] M. de Sousa y A. Gonçalves, “humanportal - A React.js case study,” en *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, 2020, págs. 1-6. DOI: [10.23919/CISTI49556.2020.9141070](https://doi.org/10.23919/CISTI49556.2020.9141070).

- [51] A. Moiseev e Y. Fain, *Angular Development with TypeScript*. Manning, 2018, ISBN: 9781638355250. [En línea]. Disponible en: <https://books.google.com.co/books?id=1TgzEAAAQBAJ>.
- [52] S. S. Gaikwad y P. Adkar, "A review paper on bootstrap framework," *IRE Journals*, vol. 2, n.º 10, págs. 349-351, 2019. [En línea]. Disponible en: <http://www.irejournals.com/formatedpaper/1701173.pdf>.
- [53] Twitter Inc., *Bootstrap*, <https://getbootstrap.com/>, (consultado Feb. 11,2023), 2011.
- [54] Hostinger International Ltd., *Hostinger*, <https://www.hostinger.co/>, (consultado Feb. 11,2023), 2021.
- [55] GitHub Inc., *GitHub Pages*, <https://pages.github.com/>, (consultado Feb. 11,2023), 2021.
- [56] Netlify Inc., *Netlify*, <https://www.netlify.com/>, (consultado Feb. 11,2023), 2021.
- [57] Heroku Inc., *Heroku*, <https://www.heroku.com/>, (consultado Feb. 11,2023), 2021.
- [58] Amazon Web Services Inc., *Amazon Web Services (AWS)*, <https://aws.amazon.com/>, (consultado Feb. 11,2023), 2021.
- [59] Google Inc., *Firebase*, <https://firebase.google.com/>, (consultado Feb. 11,2023), 2021.
- [60] Á. Gutiérrez, M. Agustín, C. Vargas y J. Daniel, *Almacenamiento en la nube*, 2018. [En línea]. Disponible en: <https://www.acta.es/medios/informes/2018004.pdf>.
- [61] M. Escarcena y M. Pérez, *PROGRAMACION PAGINAS WEB JAVASCRIPT Y PHP*. RA-MA, 2020, ISBN: 9788418551130. [En línea]. Disponible en: <https://books.google.com.co/books?id=pcEwzgEACAAJ>.
- [62] Angular Team, *AngularFire*, <https://github.com/angular/angularfire>, (consultado Feb. 11,2023), 2021.
- [63] FirebaseExtended Team, *ReactFire*, <https://github.com/FirebaseExtended/reactfire>, (consultado Feb. 11,2023), 2021.
- [64] R. Bala, S. Lata y H. Kaur, "Web Designing: A Review on Websites Design," *International Journal of Research Publication and Reviews*, vol. 3, n.º 8, págs. 2191-2193, ago. de 2022. [En línea]. Disponible en: <https://ijrpr.com/uploads/V3ISSUE8/IJRPR6701.pdf>.
- [65] P. Casado, *Diseño y construcción de paginas web*. RA-MA, 2020, ISBN: 9788499649962. [En línea]. Disponible en: https://www.ra-ma.es/libro/diseño-y-construcción-de-paginas-web_115725/.
- [66] "IEEE Recommended Practice for Software Requirements Specifications," *IEEE Std 830-1998*, págs. 1-40, 1998. DOI: 10.1109/IEEESTD.1998.88286.

- [67] J. Tendo, *Diseño de interfaces web*. 2014, ISBN: 9788499643632. [En línea]. Disponible en: <https://www-ebooks7-24-com.ucundinamarca.basesdedatosezproxy.com/?il=11731>.
- [68] *Adobe Color*, [consultado: Mar. 04,2023], 2020. [En línea]. Disponible en: <https://color.adobe.com/es/create/image>.
- [69] S. Hallé, N. Bergeron, F. Guérin, G. Le Breton y O. Beroual, “Declarative layout constraints for testing web applications,” *Journal of Logical and Algebraic Methods in Programming*, vol. 85, n.º 5, págs. 737-758, 2016, ISSN: 2352-2208. DOI: <https://doi.org/10.1016/j.jlamp.2016.04.001>. [En línea]. Disponible en: <https://www.sciencedirect.com/science/article/pii/S2352220816300293>.
- [70] *About npm*, [consultado: Mar. 04,2023], 2021. [En línea]. Disponible en: <https://docs.npmjs.com/about-npm>.
- [71] *Angular*, [consultado: Mar. 04,2023], 2021. [En línea]. Disponible en: <https://angular.io/guide/what-is-angular>.