



Universidad
Nacional
de Loja

Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables

Carrera de Ingeniería en Sistemas

Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja

Irrigation system using IoT for crops at the National University of Loja.

Trabajo de Titulación previo a
la obtención del título de
Ingeniero en Sistemas.

AUTORES:

Angel Favian Minga Medina
Alex Santiago Nole Reyes

DIRECTOR:

Ing. Edwin René Guamán Quinche, Mg.Sc.

Loja-Ecuador

2023

Certificación

Loja, 11 de enero del 2023

Ing. Edwin René Guamán Quinche, Mg.Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

CERTIFICO:

Que he revisado y orientado todo el proceso de elaboración del Trabajo de Titulación denominado: **Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja**, previo a la obtención del título de **Ingeniero en Sistemas**, de la autoría de los estudiantes **Angel Favian Minga Medina** con cédula de identidad **Nro. 1104695190** y **Alex Santiago Nole Reyes** con cédula de identidad **Nro. 1105642076**. Una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación para la respectiva sustentación y defensa.

Ing. Edwin René Guamán Quinche, Mg.Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

Autoría

Nosotros, **Angel Favian Minga Medina y Alex Santiago Nole Reyes**, declaramos ser autores del presente Trabajo de Integración Curricular y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente aceptamos y autorizamos a la Universidad Nacional de Loja la publicación de nuestro Trabajo de Titulación, en el Repositorio Digital Institucional – Biblioteca Virtual.

Firma:

Cédula de identidad: 1105642076

Fecha: 26/06 /23.

Correo electrónico: alex.nole@unl.edu.ec

Teléfono: 0999724161

Firma:

Cédula de identidad: 1104695190

Fecha: 26/06 /23.

Correo electrónico: afmingam@unl.edu.ec

Teléfono: 0986293717

Carta de autorización por parte del autor/a, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo del Trabajo de Integración Curricular

Nosotros, **Angel Favian Minga Medina; Alex Santiago Nole Reyes**, declaramos ser autores del Trabajo de Titulación denominado: **Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja**, como requisito para optar por el título de **Ingeniero en Sistemas**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los veintiséis días del mes de junio de dos mil veintitrés.

Firma:

Autor: Alex Santiago Nole Reyes

Cédula: 1105642076

Dirección: Los Operadores

Correo electrónico: alex.nole@unl.edu.ec

Teléfono: 0999724161

Firma:

Autor: Angel Favian Minga Medina

Cédula: 1104695190

Dirección: IV Centenario

Correo electrónico: afmingam@unl.edu.ec

Teléfono: 0986293717

DATOS COMPLEMENTARIOS:

Director del Trabajo de Titulación: Ing. Edwin René Guamán Quinche, Mg.Sc.

Dedicatoria

El presente Trabajo de Titulación es dedicado a mis padres Angel Salvador Minga Sarango, Mariana de Jesus Medina Tene, por su apoyo incondicional, por nunca dudar de mí, porque a pesar de todo lo que hemos pasado siempre ha sido mi fortaleza y siempre lo será, gracias a usted estoy cumpliendo una de muchas metas que tengo, y sé que con su apoyo lo lograré.

Angel Favian Minga Medina

En primer lugar, agradezco a Dios por haberme dado la vida y permitirme llegar a esta etapa de mi formación profesional. A mis padres Wilson Nole y Miriam Reyes, por su apoyo incondicional en todos mis proyectos, y que gracias a ellos he podido lograr mis objetivos. A mi hija Valentina quien fue motor para esforzarme a conseguir este logro tan anhelado. De igual forma a mis hermanos quienes con todo el cariño me han impulsado a seguir adelante en todos mis proyectos.

Alex Santiago Nole Reyes

Agradecimiento

Agradezco primeramente a Dios por permitirme cumplir con mi objetivo y acompañarme siempre a todas partes que voy y en todas las acciones que realizo.

Alex Santiago Nole Reyes

Agradezco a mis padres, hermanos y a mi hija por darme todo el apoyo necesario para poder cumplir esta meta, al igual que brindarme toda la orientación indispensable para culminar exitosamente con esta actividad.

Alex Santiago Nole Reyes

A la Universidad Nacional de Loja, Facultad de Energía, las Industrias y los Recursos Naturales no Renovables; especialmente a la Carrera de Ingeniería en Sistemas y sus catedráticos, autores de nuestra formación académica.

Angel Favian Minga Medina

Agradezco a mis Padres Angel Salvador Minga Sarango, Mariana de Jesus Medina Tene por haber fomentado en mí el anhelo de superación, gracias por su apoyo, comprensión y sus consejos en los momentos difíciles.

Angel Favian Minga Medina

A mis maestros gracias por su entrega y paciencia al momento de compartir sus enseñanzas, por su confianza y apoyo, al Ing. René Guamán por su asesoramiento durante la elaboración de este Trabajo de Titulación, y finalmente a esta prestigiosa institución, por su labor de formar profesionales innovadores para la sociedad preparándolos para un futuro competitivo.

Alex Nole-Angel Minga

Finalmente; con gratitud, al Ing. Edwin René Guamán Quinche, por ser el guía durante el proceso de desarrollo de nuestro trabajo de titulación.

Alex Nole-Angel Minga

Índice de contenidos

Portada	i
Certificación	ii
Autoría	iii
Carta de autorización	iv
Dedicatoria	v
Agradecimiento	vi
Índice de contenidos	vii
Índice de tablas:.....	xv
Índice de figuras:	xvi
Índice de Anexos:.....	xviii
1. Título	1
2. Resumen	2
2.1. Abstract.....	3
3. Introducción	4
4. Marco teórico	6
4.1. Reseña histórica.....	6
4.2. Sistema de Irrigación	7
4.2.1. <i>Textura de suelo</i>	8
4.2.2. <i>Capacidad de Campo</i> :.....	8
4.2.3. <i>Punto de Marchitez Permanente</i> :.....	9
4.2.4. <i>Infiltración del suelo</i> :	9
4.3. Internet de las Cosas	10
4.3.1. <i>Importancia de Internet de las Cosas</i>	10
4.3.2. <i>Componentes de Internet de las Cosas</i>	10
4.3.3. <i>Beneficios Internet de las Cosas</i>	11
4.3.4. <i>Aplicaciones de Internet de las Cosas</i>	11
4.4. Sistema de Irrigación con IoT	11
4.5. Convertidor Analógico Digital.....	13
4.6. Arquitecturas de IoT para irrigación.....	16
4.6.1. <i>Arquitectura multicapa</i>	16
4.6.1.1. <i>Arquitectura tres capas</i> :.....	16
4.6.1.2. <i>Arquitectura cuatro capas</i> :	17
4.6.1.3. <i>Arquitectura cinco capas</i> :	18

4.6.1.4.	<i>La arquitectura Lambda</i>	19
4.6.1.5.	<i>Arquitectura Kappa</i>	20
4.7.	Protocolos.....	20
4.7.1.	<i>MQTT</i>	21
4.7.2.	<i>Hypertext Transfer Protocol (HTTP)</i>	21
4.7.3.	<i>CoAp</i>	22
4.7.4.	<i>Tabla comparativa de protocolos de IoT</i>	23
4.8.	Componentes de hardware.....	23
4.8.1.	<i>Selección de sensor de humedad</i>	23
4.8.2.	<i>Sensor de humedad FC-28</i>	24
4.8.3.	<i>Sensor de caudal</i>	24
4.8.4.	<i>Electroválvula</i>	25
4.8.5.	<i>Modulo Relay</i>	25
4.8.6.	<i>Microcomputador Raspberry pi</i>	26
4.8.7.	<i>Micro controlador Esp32</i>	26
4.9.	Tecnologías de desarrollo de software.....	27
4.9.1.	<i>JavaScript</i>	27
4.9.2.	<i>Node .js</i>	28
4.9.3.	<i>MongoDB</i>	28
4.9.4.	<i>React</i>	28
4.9.5.	<i>Flutter</i>	28
4.9.6.	<i>Lenguaje Dart</i>	29
4.9.7.	<i>Mosquitto</i>	29
4.10.	Programación eXtrema (XP).....	29
4.10.1.	<i>Características</i>	30
4.10.2.	<i>Base de la programación eXtrema</i>	30
4.10.3.	<i>Fases de la Programación eXtrema</i>	30
4.11.	Trabajos Relacionados.....	31
4.11.1.	<i>Primer Trabajo Relacionado</i>	31
4.11.2.	<i>Segundo Trabajo Relacionado</i>	32
4.11.3.	<i>Tercer Trabajo Relacionado</i>	33
4.11.4.	<i>Cuarto Trabajo Relacionado</i>	34
4.11.5.	<i>Quinto Trabajo Relacionado</i>	35
5.	Metodología	36
5.1.	Área de estudio.....	36

5.2.	Procedimiento	36
5.2.1.	<i>Investigar las técnicas de IoT para la construcción de aplicaciones de irrigación</i> 36	
5.2.1.1.	<i>Planificación de la revisión sistemática sobre las técnicas de IoT para la construcción de aplicaciones de irrigación.</i>	36
5.2.1.2.	<i>Desarrollar la revisión sistemática sobre técnicas de IoT para la construcción de aplicaciones de irrigación.</i>	37
5.2.1.3.	<i>Documentar los resultados obtenidos sobre las técnicas de IoT para la construcción de aplicaciones de irrigación.</i>	37
5.2.2.	<i>Diseñar la arquitectura de sistema de irrigación para recolectar los datos generados por los componentes IoT.</i>	37
5.2.2.1.	<i>Identificar los componentes y materiales necesarios para funcionamiento del sistema de irrigación con IoT.</i>	37
5.2.2.2.	<i>Diseñar la arquitectura del sistema de irrigación con IoT haciendo uso de los componentes y materiales anteriormente identificados.</i>	37
5.2.3.	<i>Desarrollar un módulo de control que permitirá administrar el sistema de irrigación con IoT mediante la metodología XP.</i>	38
5.2.3.1.	<i>Planificar las etapas para el desarrollo del módulo de control y administración.</i> 38	
5.2.3.2.	<i>Diseñar el módulo de control en base de la especificación de requisitos utilizando notación UML (Unified Modeling Language)</i>	38
5.2.3.3.	<i>Codificar el sistema de control y administración en base al diseño previamente establecido</i>	38
5.2.3.4.	<i>Realizar el prototipo de sistema de irrigación, implementando hardware (sensores, actuadores, tarjetas programables, microcomputadoras) y software (sistema de control y administración) haciendo uso de la arquitectura antes establecida</i>	39
5.3.	Recursos.....	39
5.2.4.	<i>Métodos</i>	39
5.2.4.1.	<i>Analítico</i>	39
5.2.4.2.	<i>Investigación-Acción</i>	39
5.2.4.3.	<i>Experimentación</i>	39
5.2.5.	<i>Técnicas</i>	40
5.2.5.1.	<i>Entrevistas</i>	40
5.2.5.2.	<i>Reuniones</i>	40
5.2.5.3.	<i>Revisión Bibliográfica</i>	40
5.2.5.4.	<i>Revisión Sistemática de Literatura</i>	40
5.2.6.	<i>Estándares</i>	40
5.2.6.1.	<i>IEEE 830</i>	40

5.2.6.2. <i>Modelo 4+1</i>	40
5.2.7. <i>Recursos Técnicos</i>	40
5.2.7.1. <i>Parsifal</i>	41
5.2.7.2. <i>Herramientas colaborativas</i>	41
5.2.8. <i>Recursos Hardware</i>	41
5.4. <i>Participantes:</i>	41
6. Resultados	43
6.1. OBJETIVO 1: Investigar las técnicas de IoT para la construcción de aplicaciones de irrigación.	43
6.1.1. <i>Planificación de la revisión sistemática sobre las técnicas de IoT para la construcción de aplicaciones de irrigación.</i>	43
6.1.2. <i>Desarrollar la revisión sistemática sobre técnicas de IoT para la construcción de aplicaciones de irrigación.</i>	45
6.1.2.1. <i>Arquitecturas analizadas/utilizadas en el desarrollo de sistemas de irrigación con IoT</i>	49
6.1.2.2. <i>Protocolos analizadas/utilizadas en el desarrollo de sistemas de irrigación con IoT</i>	50
6.1.2.3. <i>Herramientas analizadas/utilizadas en el desarrollo de sistemas de irrigación con IoT.</i>	51
6.1.2.4. <i>Problemas analizadas en el desarrollo de sistemas de irrigación con IoT</i>	52
6.1.3. <i>Documentar los resultados obtenidos sobre las técnicas de IoT para la construcción de aplicaciones de irrigación</i>	53
6.2. OBJETIVO 2: Diseñar la arquitectura de sistema de irrigación para recolectar los datos generados por los componentes IoT.	54
6.1.4. <i>Identificar los componentes y materiales necesarios para funcionamiento del sistema de irrigación con IoT.</i>	54
6.1.5. <i>Diseñar la arquitectura del sistema de irrigación con IoT haciendo uso de los componentes y materiales anteriormente identificados.</i>	55
6.1.5.1. <i>Diseño Arquitectónico IoT</i>	56
6.1.5.2. <i>Arquitectura Física</i>	57
6.1.5.3. <i>Estilo Arquitectónico de IoT (Broker)</i>	60
6.3. OBJETIVO 3: Desarrollar un módulo de control que permitirá administrar el sistema de irrigación con IoT mediante la metodología XP.	62
6.1.6. <i>Planificar las etapas para el desarrollo del módulo y administración.</i>	62
6.1.6.1. <i>Especificación de Requisitos:</i>	64
6.1.7. <i>Diseñar el módulo de control en base de la especificación de requisitos utilizando notación UML (Unified Modeling Language).</i>	67
6.1.7.1. <i>Vistas. - escenarios.</i>	67

6.1.7.2. Vista. – Procesos.....	68
6.1.7.3. Vista. - Lógica.....	70
6.1.7.4. Vista. - Desarrollo.....	70
6.1.7.5. Vista. – Física.....	72
6.1.8. Codificar el sistema de control y administración en base al diseño previamente establecido.	74
6.1.8.1. Diseño de la pantalla de Inicio de Sesión del aplicativo web.....	74
6.1.8.2. Diseño de la pantalla principal del aplicativo web	75
6.1.8.3. Diseño de pantalla para gestionar usuarios por el administrador del aplicativo web	76
6.1.8.4. Diseño de la pantalla de tipos de suelo del aplicativo web	77
6.1.8.5. Diseño de la pantalla de topic de sensores del aplicativo web.....	78
6.1.8.6. Diseño de pantalla inicio de sesión aplicativo móvil.....	79
6.1.8.7. Diseño de pantalla de control de riego (Encender/Apagar).....	80
6.1.8.8. Codificación del servicio web API-REST.....	81
6.1.8.9. Codificación del aplicativo web.....	84
6.1.8.10. Codificación del aplicativo móvil	86
6.1.9. Fase de pruebas	87
6.1.9.1. Pruebas de Aceptación de Usuario.....	87
6.1.9.2. Pruebas Unitarias del servicio web	87
6.1.9.3. Pruebas de carga y estrés	89
6.1.10. Realizar el prototipo de sistema de irrigación, implementando hardware (sensores, actuadores, microcontroladores) y software (sistema de control y administración) haciendo uso de la arquitectura antes establecida	90
6.1.10.1. Configuración de servidor broker.....	90
6.4. Experimentación.....	97
7. Discusión	101
7.1. Desarrollo de la propuesta alternativa.....	101
7.1.1. Objetivo Nro.1 Investigar las técnicas de IoT para la construcción de aplicaciones de irrigación.	101
7.1.2. Objetivo Nro.2 Diseñar la arquitectura de sistema de irrigación para recolectar los datos generados por los componentes IoT.....	102
7.1.3. Objetivo Nro.3 Desarrollar un módulo de control que permitirá administrar el sistema de irrigación con IoT mediante la metodología XP.....	103
7.2. Valoración Técnica Económica Ambiental	103
7.2.1. Valoración Técnica	103
7.2.2. Valoración Ambiental	103

7.2.3. Valoración Económica.....	104
8. Conclusiones	106
9. Recomendaciones	107
10. Bibliografía	108
11. Anexo.....	118
Anexo 1 Revisión Sistemática de literatura	118
Anexo 2 Manual RSL.....	129
Anexo 3 IEEE 830	140
1. Introducción.....	143
1.1. Propósito	143
1.2. Alcance.....	143
1.3. Personal involucrado	143
1.3.1. Responsables del proyecto a desarrollar	143
1.3.2. Responsables del control de Riego.....	143
1.3.3. Director de Trabajo de titulación	144
1.4. Definiciones, acrónimos y abreviaturas	144
1.5. Referencias	144
1.6. Resumen	144
2. Descripción general.....	145
2.1. Perspectiva del producto.....	145
2.2. Funcionalidad del producto.....	145
2.3. Características de los usuarios	146
2.4. Restricciones.....	146
2.5. Suposiciones y dependencias	147
2.6. Evolución previsible del sistema.....	147
3. Requisitos específicos	147
3.1. Requisitos Funcionales	147
3.2. Requisitos No Funcionales.....	152
3.3. Requisitos comunes de los interfaces	155
3.3.1. Interfaces de usuario.....	155
3.3.2. Interfaces de hardware.....	155
4. Anexos	156
Anexo 4 Arquitectura 4+1	157
1. Introducción.....	158

1.1.	Propósito	158
1.2.	Alcance.....	158
1.3.	Usuarios Interesados	158
2.	Referencias.....	158
3.	Definiciones, acrónimos y abreviaciones.....	159
4.	Framework Conceptual	160
4.1.	Descripción de la arquitectura en contexto	160
4.2.	Stakeholders y sus roles	160
5.	Descripciones prácticas de arquitectura	160
5.1.	Identificación de los Stakeholders y sus responsabilidades.....	160
5.2.	Selección de puntos de vista de la arquitectura	161
5.3.	Vistas de la arquitectura.....	161
Anexo 5 Manual de prototipo.....		202
1.	Introducción.....	203
1.1.	Propósito	203
2.	Definiciones, acrónimos y abreviaciones.....	203
3.	Herramientas Utilizadas.....	203
4.	Construcción del Prototipo de Sistema de Irrigación para Cultivos de La Universidad Nacional de Loja.....	204
4.1.	Configuración del servidor o Broker en Raspberry Pi	204
4.1.2.	<i>Habilitar acceso Remoto con Autenticación en broker mosquitto.....</i>	<i>205</i>
4.2.	Programación del suscriptor publicador o módulo Esp32 para sensores.....	207
4.2.1.	<i>Librerías utilizadas para la programación del Esp32.....</i>	<i>207</i>
4.3.	Conexión de Sensores y actuadores a modulo Esp32.....	209
4.4.	Programación del suscriptor publicador o módulo Esp32 para actuadores.....	212
4.5.	Conexión de broker con API-REST	214
4.6.	Conexión con aplicativo Front-end y aplicación Móvil.....	216
5.	Anexos	217
Anexo 6 Pruebas De software (Unitarias, Carga y estrés y De Usabilidad)		218
Anexo 7 Experimentación.....		243
Anexo 8 Certificados de Pruebas de Usabilidad.....		257
Anexo 9 Diagramas desarrollados en Proteus para impresión de placa del prototipo..		261
Anexo 10 Certificado de traducción del resumen del trabajo de titulación		262

Índice de tablas:

Tabla 1. Tabla comparativa de protocolos de IoT.....	23
Tabla 2. Tabla comparativa de sensor de humedad.....	24
Tabla 3. Herramientas hardware	41
Tabla 4. Fuente bibliográfica y cadena de búsqueda.....	44
Tabla 5. Criterios de inclusión y exclusión	45
Tabla 6. Artículos evaluados	46
Tabla 7. Tipos de arquitecturas	49
Tabla 8. Componentes electrónicos.....	52
Tabla 9. Componentes utilizados para el sistema de irrigación	54
Tabla 10. Preguntas de la entrevista.....	62
Tabla 11. Roles de usuarios.....	64
Tabla 12. Requisitos funcionales del sistema.....	64
Tabla 13. Requerimientos no funcionales	66
Tabla 14. Controlador, router, url.....	83
Tabla 15. Pruebas Unitarias Postman para Topic.....	87
Tabla 16. Prueba de carga y estrés	89
Tabla 17. Datos obtenidos en la primera repetición del experimento	98
Tabla 18. Valoración económica-servicios	104
Tabla 19. Recursos hardware, software y de riego.....	104
Tabla 20. Valores totales de trabajo de titulación	105

Índice de figuras:

Figura 1. Historia del riego	6
Figura 2. Proceso de un ADC	14
Figura 3. Proceso de Muestreo	15
Figura 4. Arquitectura tres capas	17
Figura 5. Arquitectura cuatro capas	18
Figura 6. Arquitectura cinco capas.....	19
Figura 7. Arquitectura Lambda	19
Figura 8. Arquitectura Kappat.....	20
Figura 9. Sensor de humedad FC-28.....	24
Figura 10. Sensor caudal	25
Figura 11. Electroválvula	25
Figura 12. Módulo Relay	26
Figura 13. Raspberry pi.....	26
Figura 14. Modulo ESP 32.....	27
Figura 15. Arquitectura 4 capas	31
Figura 16. Topología IoT	32
Figura 17. Arquitectura lógica (Diagrama de componentes)	33
Figura 18. Prototipo de las conexiones en el Esp32.....	33
Figura 19. Prototipo de sistema de irrigación y ventilación de un invernadero.....	34
Figura 20. Arquitectura de tres capas en sistema de monitoreo de cultivos	35
Figura 21. Diagrama de flujo paso de búsqueda	46
Figura 22. Protocolos para IoT.....	51
Figura 23. Problemas que se presentan en la Construcción	53
Figura 24. MQTT-Mosquitto	55
Figura 25. Diseño de la arquitectura en tres capas.....	57
Figura 26. Arquitectura Física.....	58
Figura 27. Diagrama esquemático del sistema de irrigación	59
Figura 28. Diseño arquitectónico de IoT Brok.....	61
Figura 29. Arquitectura de software para IoT	62
Figura 30. Caso de uso General	68
Figura 31. Diagrama de secuencia encender riego.....	69
Figura 32. Diagrama de secuencia apagar riego	69
Figura 33. Diagrama de secuencia capturar datos.....	69
Figura 34. Diagrama de clases	70
Figura 35. Diagrama de componentes.....	71
Figura 36. Diagrama de despliegue.....	73
Figura 37. Pantalla login	75
Figura 38. Interfaz principal.....	76
Figura 39. Pantalla lista usuarios	77
Figura 40. Pantalla tipo de suelo	78
Figura 41. Pantalla topic sensores	79
Figura 42. Pantalla inicio sesión	80
Figura 43. Pantalla control riego.....	81
Figura 44. Estructura del servicio web.....	82
Figura 45. Código ServerSuscriptorController	83
Figura 46. Buscar usuario	84
Figura 47. Estructura del aplicativo web.....	85
Figura 48. Estructura aplicativo móvil.....	86
Figura 49. Petición Post Login.....	88

Figura 50. Prueba carga y estrés de listar usuario	90
Figura 51. Servicio mosquitto	91
Figura 52. Configuración puertos broker	92
Figura 53. Topics configurados.....	92
Figura 54. Librerías de configuración	92
Figura 55. Puertos analógicos	93
Figura 56. Credenciales Wifi y Mqtt para la conexión	93
Figura 57. Código dentro de función setup, para conexión Wifi y broker.....	94
Figura 58. Código para envío de datos al broker	94
Figura 59. Diagrama de puertos del MCU Esp32	95
Figura 60. Conexión sensores a modulo esp-32.....	96
Figura 61. Presentación datos de humedad en vista.....	96
Figura 62. Estructura de prototipo final del sistema de irrigación	97
Figura 63. Diagrama de dispersión de repetición de suelo arenoso	99
Figura 64. Diagrama de dispersión de repetición en suelo Franco Limoso	100

Índice de Anexos:

Anexo 1. Revisión Sistemática de literatura	118
Anexo 2. Manual RSL	129
Anexo 3. IEEE 830	140
Anexo 4. Arquitectura 4+1.....	157
Anexo 5. Manual de prototipo	202
Anexo 6. Pruebas De software (Unitarias, Carga y estrés y De Usabilidad).....	218
Anexo 7. Experimentación.....	243
Anexo 8. Certificados de Pruebas de Usabilidad	257
Anexo 9. Diagramas desarrollados en Proteus para impresión de placa del prototipo	261
Anexo 10. Manual de Usuario	262
Anexo 11. Certificado de traducción del resumen del trabajo de titulación	279

1. Título

Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja

2. Resumen

En la actualidad Internet de las cosas (IoT) ha tenido un gran impacto en el desarrollo de soluciones tecnológicas permitiendo que varios componentes del mundo real se conecten a internet posibilitando interactuar con ellas. Existen grandes aplicaciones para la automatización de procesos en distintas áreas. Una de las aplicaciones en las que se puede implementar el IoT es la agricultura conocida como Smart Farming, que permite mejorar los procesos agrícolas, como el riego que es un proceso fundamental para los cultivos, ya que se puede monitorear las necesidades hídricas de las plantas en base a los datos enviados por los sensores y con esto llevar un control del agua utilizada. En este contexto el presente TT tiene como objetivo desarrollar un prototipo de sistema irrigación con IoT para cultivos de la Universidad Nacional de Loja. Este trabajo introduce el uso de sensores higrómetros que permiten capturar los cambios de humedad del suelo, para encender el riego cuando la humedad del suelo este en punto de marchitez permanente y se cierre el riego cuando la humedad este en Capacidad de campo. En tal sentido el marco metodológico utilizado consto de tres fases. I) Se realizó una RSL para obtener información relevante sobre la temática y responder las preguntas de investigación. II) Se desarrolló la solución informática haciendo uso de la metodología XP, aplicando estándares como el documento IEEE-830 para especificar los requisitos de software y el documento arquitectónico 4+1. III) Se realizó la experimentación para conocer el funcionamiento de los sensores y los datos que envían estos en base a la humedad del suelo en un tiempo específico. Por último, el desarrollo del prototipo se ejecutó en un ambiente académico, partiendo con el desarrollo del RSL en donde se obtuvieron 25 artículos de los cuales la mayoría de autores utilizaron la arquitectura de tres capas para la automatización de sistemas de irrigación, empleando el protocolo de comunicación MQTT mediante la herramienta mosquito que funciona como servidor o bróker por donde pasan los paquetes enviados desde los publicadores hasta los suscriptores. En el diseño de la arquitectura se utilizó la arquitectura de tres capas que se adaptó al prototipo para el control del riego y finalmente se realizó el desarrollo del módulo de control utilizando la metodología ágil de software XP y armado del prototipo con la información recabada en las fases anteriores. De esta manera desarrolló un prototipo de sistema de irrigación con IoT, que permitió capturar la humedad del suelo mediante sensores higrómetros, para determinar cuándo es necesario abrir y cerrar el riego.

Palabras Claves: Internet de las Cosas, software, prototipo, riego, automatización, Smart farming

2.1. Abstract

Currently, the Internet of Things (IoT) has had a significant impact on the development of technological solutions, enabling various real-world components to connect to the Internet and allowing interaction with them. There are extensive applications for process automation in various fields. One such application where IoT can be implemented is agriculture, known as Smart Farming, which improves agricultural processes, such as irrigation a vital process for crops. With IoT, it becomes possible to monitor the water needs of plants based on data sent by sensors, enabling better control of water usage. In this context, the objective of this research work is to develop a prototype IoT irrigation system for crops at the Universidad Nacional de Loja. This work introduces the use of hygrometer sensors to capture soil moisture changes, triggering irrigation when the soil moisture reaches the permanent wilting point and shutting off irrigation when it reaches field capacity. The methodological framework used consisted of three phases. I) A systematic literature review (SLR) was conducted to gather relevant information on the topic and address the research questions. II) The software solution was developed using the XP methodology, applying standards such as the IEEE-830 document to specify software requirements and the 4+1 architectural document. III) An experimentation phase was carried out to understand the functioning of the sensors and the data they send based on soil moisture at specific times. Finally, the prototype development was executed in an academic environment, starting with the SLR phase, which yielded 25 articles, most of which utilized a three-layer architecture for irrigation system automation, employing the MQTT communication protocol through the mosquitto tool, which functions as a server or broker for packets sent from publishers to subscribers. The three-layer architecture was adapted to the prototype design for irrigation control, and the control module was developed using the agile XP software methodology, followed by the assembly of the prototype using information gathered in the previous phases. In this way, a prototype IoT irrigation system was developed, capable of capturing soil moisture using hygrometer sensors to determine when it is necessary to open and close the irrigation system.

Keywords: Internet of Things, software, prototyping, irrigation, automation, Smart farming

3. Introducción

La irrigación es un conjunto de dispositivos capaces de proporcionar caudal de agua en las tierras más pobres de forma artificial y ordenada para una producción agrícola eficiente. Se trata de abastecer la tierra con la cantidad de agua necesaria, utilizando métodos artificiales para lograr una buena producción [1]. Es importante mencionar que mediante buenas condiciones ambientales y con el uso de recursos esenciales como el agua se puede obtener mejores resultados en los cultivos. Por tanto, agua y humedad en el suelo desempeña una función crucial en la agricultura. Sin embargo, el uso excesivo de agua (exceso de humedad), promueve la aparición de plagas y enfermedades que afectan negativamente la calidad de crecimiento de las plantas y estropean el producto, por otro lado, la falta de humedad puede hacer que los cultivos se marchiten. Mediante el uso de sistemas de riego eficientes, se logran ahorros significativos de recursos hídricos necesarios para los cultivos [2].

Según [3] expresa que la mayor parte del uso del agua en el Ecuador es para riego, que se estima en un 80% del uso total del agua. Sin embargo, la infraestructura de riego es extremadamente ineficiente, por lo cual existe un gran desperdicio de agua. En [4], varios estudios han determinado que la eficiencia de la infraestructura de riego pública es del 30%, mientras que la eficiencia del sector privado está entre el 16% y el 50%. Solo los usuarios que cultivan productos de exportación (banano, cacao) cuentan con tecnologías de riego (sistemas de riego automatizados con programadores de riego) que ahorran agua. Cabe destacar que el agua se desperdicia a menudo cuando se asignan pequeños terrenos que no cuentan con la tecnología suficiente para conservar el agua. De tal manera, en la provincia de Loja, la mayor parte de las concesiones otorgadas por la Secretaría de Estado del Agua (SENAGUA), corresponden al riego y consumo de agua en hogares, representando el 88,29% de las concesiones dadas, y su caudal total es el 97,55 % del total otorgado. En concreto, se destinan 37,91 m³/s para uso de riego y 0,92 m³/s para consumo doméstico. Por tanto, la agricultura bajo riego es la que más consume agua [5].

Se hace necesario destacar que los sistemas de irrigación tradicionales son un aspecto a mejorar. Dado que el riego tradicional puede ocasionar el uso excesivo de recursos hídricos o, por el contrario, que la cantidad de agua suministrada a los cultivos sea mínima o no sea lo suficiente [8]. Ahora bien, con los avances tecnológicos y el uso de internet de las cosas (IoT) se puede mejorar el proceso de riego significativamente. Por lo que los sistemas de irrigación con IoT son la tendencia actual para automatizar el proceso de riego proporcionando un control eficiente del mismo. Lo que permiten tener un control sobre la cantidad necesaria de agua para los

cultivos y el tiempo que riego que se implementará de acuerdo al tipo de suelo y a sus características específicas.

De este modo el presente trabajo de titulación tiene como objetivo desarrollar un prototipo de sistema de irrigación con IoT para cultivos de la Universidad Nacional de Loja. Que permita determinar la cantidad de agua requerida en base a la humedad de suelo obtenida por los sensores el tiempo específico.

En este contexto se desarrolló un prototipo de sistema de irrigación con IoT, que permitió obtener datos de humedad del suelo mediante sensores de humedad (higrómetros), y la cantidad de agua utilizada para el riego mediante sensores de caudal, lo que permitió automatizar el proceso de riego de acuerdo a las características de tipo de suelo y los datos percibidos por los sensores. La automatización del sistema de irrigación se desarrolló mediante la arquitectura de tres capas y con el uso del protocolo MQTT para la comunicación de sensores con el aplicativo web y móvil. Es importante mencionar que en el presente proyecto no se abarcó la implementación de modelos inteligentes para la automatización, por lo que estos están fuera del alcance del mismo.

Finalmente, el documento de la memoria del TT se estructuró siguiendo la normativa propuesta en la Universidad Nacional de Loja. En la sección de Marco Teórico se detallaron arquitecturas, protocolos y conceptualización sobre el área de estudio. En la sección de Metodología se detallan los recursos técnicos, científicos y técnicas para obtener información sobre el objeto de estudio como es el caso de un RSL y experimentación. En la sección de Resultados se describieron las evidencias obtenidas en el proceso de desarrollo del TT. En la sección de Discusión, se detalló de forma argumentativa la relación de los resultados con la teoría sobre el estado de la investigación. En la sección de Conclusión describe los resultados de la pregunta de investigación y como su palabra lo dice a que se concluyó con el desarrollo del sistema de irrigación. En la sección de Recomendaciones se describen las recomendaciones a tomar en consideración para el desarrollo del proyecto.

4. Marco teórico

En esta sección se presenta una revisión bibliográfica que ayuda al lector a entender el contexto del resultado final del Trabajo de Titulación (TT). Se parte con una reseña histórica sobre los sistemas de irrigación mediante una línea de tiempo, seguido de la contextualización de la irrigación, IoT, de igual forma la contextualización del internet de las cosas en la agricultura. Finalizando con los trabajos relacionados sobre el objeto de estudio.

4.1. Reseña histórica

El riego es aparentemente una técnica agrícola que riega artificialmente las plantas para apoyar su crecimiento. Además, es la base del desarrollo económico y de una cierta forma de vida, transformando el paisaje y permitiendo que las personas se asienten en áreas áridas y semiáridas que de otro modo no podrían sustentar una gran población [6]. Cabe destacar que el riego al ser un procedimiento que realizan los humanos ha venido sufriendo varias modificaciones a lo largo de la línea del tiempo como se puede apreciar en la figura 1.

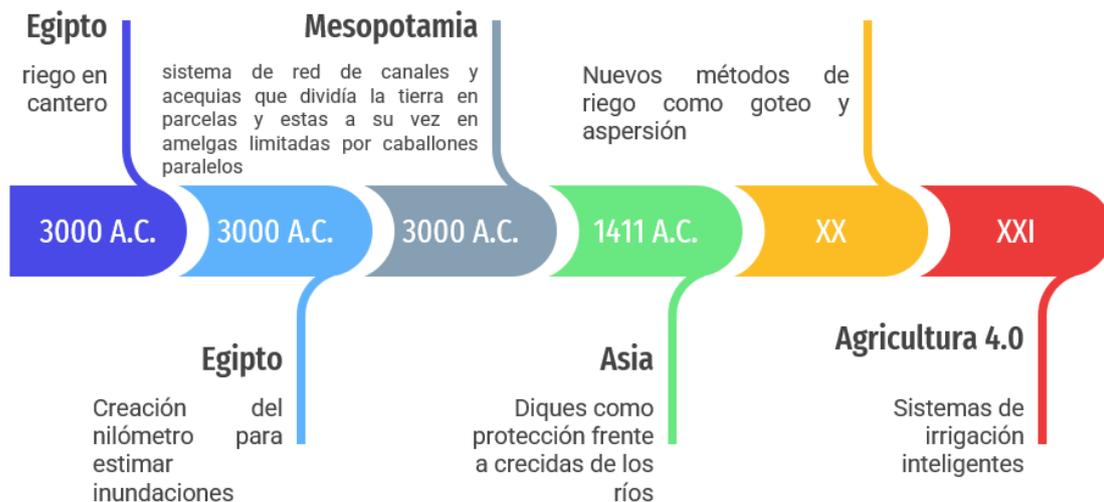


Figura 1. Historia del riego

Los sistemas de riego según [7], los sistemas de riego tuvieron sus inicios en el antiguo Egipto, con el riego canteros a gran escala creado por el rey Menes, quien fue el fundador de la primera dinastía en 3100 A.C. Consistía en la construcción de canales excavados perpendicularmente al río para llevar el agua a las zonas sembradas. Sus embocaduras permanecían cerradas hasta que el agua alcanzaba una suficiente altura en el río. La tierra inundada retenía tanto el agua como los sedimentos de materia orgánica y fosfatos. Cuando el agua de inundación saturaba un cantero, el agua sobrante se drenaba hacia un cantero inferior o hacia un canal.

Los antiguos egipcios desarrollaron un método preciso para medir el nivel del agua a través de escalas graduadas que se encontraban en los muros de los embarcaderos o templos ubicados en las orillas del río Nilo, llamados nilómetros. Esto permitía mantener un registro para comparar los datos con años anteriores y estimar el nivel de inundación esperado y, por lo tanto, la cosecha prevista. En Mesopotamia, se practicaba el riego continuo mediante inundación desde la siembra hasta la cosecha. El sistema estaba compuesto por una red de canales y acequias que dividían la tierra en parcelas, limitadas por caballones paralelos. Para distribuir el agua y regar, se requería un estricto control de los canales y la utilización de compuertas reguladoras.

Las civilizaciones posteriores que implementaron sistemas de riego se desarrollaron en Asia Central y en China, donde se llevaron a cabo legendarias obras hidráulicas realizadas por el emperador Yu, quien construyó diques para protegerse de las crecidas de los ríos hace 1411 años A.C, según lo mencionado en la fuente [8].

En las zonas cercanas al desierto, conocido como "desierto negro", en la frontera entre Siria y Jordania, así como en el sur de Yemen, donde antiguamente se le conocía como Arabia feliz, se utilizan técnicas muy sofisticadas para el manejo del agua. Estas técnicas aprovechan el agua de los cursos de agua efímeros, llamados wadis, que suelen estar secos pero que pueden transportar caudales importantes en forma de crecidas rápidas [9].

Durante el siglo XX, en las sociedades más desarrolladas, se observó un importante aumento en el uso de sistemas de riego. Por un lado, la aplicación de nuevas técnicas de riego como la aspersión y el goteo permitieron un mayor control en la distribución del agua, además de disminuir la cantidad de trabajo necesario en el campo. Por otro lado, se han desarrollado nuevas y mejores técnicas para la recolección de agua subterránea, lo que ha permitido un mayor uso de esta fuente de agua para riego, principalmente impulsado por la iniciativa privada. [8].

En la actualidad con la llamada agricultura 4.0 se han construido sistemas de irrigación combinados con tecnologías de comunicación, que facilitan la gestión del agua por medio de la tecnología.

4.2. Sistema de Irrigación

El uso de los sistemas de irrigación está orientado a lograr mayores rendimientos en la producción de cualquier cultivo. Asimismo, es una alternativa al actual problema de escasez de agua, que afecta a la baja producción agrícola [10].

La gestión del riego es un proceso complejo que involucra muchas decisiones tomadas por los agricultores para determinar cuándo y cuánta agua suministrar a los cultivos. En algunos casos, los agricultores están lejos de las tierras de cultivo y es posible que no estén al tanto de las condiciones actuales de los cultivos. Por lo tanto, existe la necesidad de una alternativa económica a la gestión eficaz del agua [11].

El propósito básico del riego es proporcionar la humedad necesaria y esto se puede hacer de varias maneras. Sin embargo, independientemente del método de riego utilizado, el propósito del riego es "reponer periódicamente las reservas de humedad del suelo" en la zona de raíces de la planta.

Para determinar la cantidad de recursos hídricos que los cultivos necesitan, se tiene que tomar en consideración las características del suelo, ya que el estado hídrico del suelo está determinado por el contenido de agua y la energía libre disponible. Estos factores tienen un impacto directo en el crecimiento de las plantas. Las propiedades físicas del suelo, como la densidad aparente, textura y porosidad, están estrechamente relacionadas con la productividad de los cultivos, ya que influyen en el almacenamiento y movimiento del agua en el suelo. [12].

Los parámetros físicos que se toman en consideración para los sistemas de irrigación son:

4.2.1. Textura de suelo

El suelo está compuesto por una mezcla de materiales orgánicos e inorgánicos que ocupan aproximadamente la mitad de su volumen total, mientras que la otra mitad está compuesta de poros que pueden contener agua y aire en diferentes proporciones. La textura del suelo se refiere al porcentaje en que se encuentran los diferentes elementos que lo componen, como la arena gruesa, la arena media, la arena fina, el limo y la arcilla. La condición hídrica del suelo se describe mediante el contenido y la energía libre del agua, que son factores que influyen directamente en el comportamiento de las plantas, y están estrechamente relacionados con las propiedades físicas del suelo, como la densidad aparente y la porosidad [13].

4.2.2. Capacidad de Campo:

La capacidad de campo se refiere a la cantidad de agua que un suelo puede retener después de estar saturado con agua [14].

Existe una fórmula llamada fórmula de Peele que se utiliza para determinar la capacidad de campo, la cual toma en cuenta el contenido de arcilla, limo y arena del suelo. La fórmula es:

$$Cc = 0.48 Ac + 0.162 L + 0.023 Ar + 2.62$$

Donde:

- Cc: Humedad a la capacidad de campo, expresada como humedad gravimétrica, en tanto por ciento
- Ac: contenido de arcilla, expresado en humedad gravimétrica
- L: contenido de limo, expresado en humedad gravimétrica
- Ar: contenido de arena, expresado en humedad gravimétrica

4.2.3. Punto de Marchitez Permanente:

El punto de marchitez permanente es la cantidad mínima de humedad en el suelo que las plantas pueden absorber antes de marchitarse [14].

La fórmula de Briggs se utiliza para calcular el punto de marchitez permanente.:

$$Pm = 0,302 Ac + 0,102 L + 0,0147 Ar$$

Donde:

- Pm: humedad en el punto de marchitez, expresado en humedad gravimétrica, en tanto por ciento
- Ac: contenido de arcilla, expresado en humedad gravimétrica
- L: contenido de limo, expresado en humedad gravimétrica
- Ar: contenido de arena, expresado en humedad gravimétrica

4.2.4. Infiltración del suelo:

La infiltración se refiere a la entrada de agua en el perfil del suelo en dirección vertical. La capacidad de infiltración del suelo es un factor clave que determina la velocidad a la cual el agua puede ser absorbida por la superficie del suelo sin que se produzca escurrimiento. [15].

Los parámetros físicos del suelo no son los únicos factores importantes a considerar, también es necesario tener en cuenta la disponibilidad de agua en el suelo, que se puede clasificar en tres categorías. La primera es el agua gravitacional, que se drena libremente debido a la fuerza de la gravedad. La segunda es el agua no disponible, que está firmemente unida a las partículas del suelo y no puede ser absorbida por las plantas. Finalmente, el agua disponible se encuentra entre el agua gravitacional y el agua no disponible, y está retenida por fuerzas capilares. [12].

Luego de tomar en consideración las características del suelo se puede utilizar el tipo de riego de acuerdo a las necesidades de los cultivos [16], entre los tipos de riego implementados en la agricultura se encuentran los descritos a continuación:

- **Irrigación de superficie:** conocida como riego por inundación, este tipo de irrigación consiste en inundar toda el área del cultivo.
- **Irrigación por Aspersión:** Este tipo de irrigación imita al riego por lluvia ya que son capaces de asperjar gotas muy finas de agua usando chorros rotatorios y otras configuraciones que demandan menos agua y pueden cumplir con su objetivo.
- **Irrigación por goteo:** Dentro de este tipo de irrigación existen varias categorías, todas basadas en la premisa de proveer agua en la planta o muy cerca de ella para alimentar específicamente el área de crecimiento de la planta.
- **Irrigación subterránea:** Este tipo de irrigación se lo conoce también como riego bajo la superficie ya que se encarga de llevar el agua por tuberías perforadas (porosas) la misma que humedece el suelo por difusión hacia la raíz del cultivo.

4.3. Internet de las Cosas

El internet de las cosas o Internet of Things (IoT) es un conjunto de dispositivos, instrumentos, ya sean vehículos, edificios, electrodomésticos, interconectados entre sí, los mismos que interactúan por medio de sensores, actuadores y software para recopilación e intercambio de información. IoT posibilita la detección y control remoto de objetos a través de infraestructuras de red ya existente, lo que permite una integración más directa del mundo físico en los sistemas informáticos. Esto conlleva a una mayor eficiencia y precisión en diversas operaciones [17].

4.3.1. Importancia de Internet de las Cosas

El IoT abarca una amplia gama de tecnologías, que van desde sensores que monitorean el entorno hasta etiquetas RFID que permiten a los usuarios interactuar con los objetos. En este mundo, cualquier objeto puede generar datos que pueden ser recopilados y analizados, lo que permite una transición de un sistema reactivo a uno proactivo. Básicamente IoT permite funciones de detección y respuesta automáticas [18].

4.3.2. Componentes de Internet de las Cosas

IoT abarca una amplia gama de dispositivos, algunos de los cuales tienen una interfaz de usuario (UI), pero en general se enfocan en proporcionar sensores, actuadores o ambos. La función principal del sistema es recopilar datos e información de múltiples dispositivos y almacenar,

analizar y actuar sobre esta información. En el ecosistema de IoT, los diferentes componentes se dividen en tres clases: nodos receptores, que incluyen sensores que monitorean el entorno físico y tienen restricciones de energía y comunicación; nodos de puerta de enlace, que actúan como agregadores de datos de sensores y proporcionan conectividad a otros servicios de IoT; y servicios de IoT, que recopilan datos de múltiples nodos de puerta de enlace y brindan servicios específicos para usuarios o eventos a través de interfaces de usuario, notificaciones o aplicaciones [19].

4.3.3. Beneficios Internet de las Cosas

La implementación de IoT en los negocios puede ofrecer una serie de beneficios, incluyendo la mejora de los activos, como equipos, maquinaria y herramientas. Los servicios integrados con sensores y cámaras permiten monitorear el rendimiento y las condiciones de trabajo, lo que ayuda a mantener el equipo seguro y en buen estado. Al utilizar los activos de manera productiva y eficiente, las empresas pueden reducir los costos de gastos y mantenimiento, lo que aumenta la rentabilidad y eficiencia general del negocio. Además, la tecnología IoT puede proporcionar oportunidades para desarrollar nuevos productos y servicios que satisfagan las necesidades y expectativas de los clientes, lo que a su vez puede mejorar la posición competitiva de una empresa en el mercado [20].

4.3.4. Aplicaciones de Internet de las Cosas

IoT brinda increíbles posibilidades todos los días a entornos como: agricultura, medicina, deportes, cría de animales, redes eléctricas, automatización industrial, seguimiento de activo e inventario [21].

La creciente tendencia del IoT ha permitido que muchos dispositivos electrónicos de uso cotidiano puedan comunicarse con plataformas virtuales en la infraestructura de Internet existente. Esto ha llevado a que los proyectos de casas inteligentes con tecnologías sostenibles sean solo algunas de las aplicaciones que el IoT puede manejar. Este desarrollo puede ser comparado con el cultivo de cultivos sostenibles, ya que permite obtener mejores resultados en la agricultura y en otros campos [22].

4.4. Sistema de Irrigación con IoT

La gestión adecuada del agua es importante en países con escasez de este recurso. Esta escasez afecta también a la agricultura, que requiere grandes cantidades de agua para su funcionamiento. Debido a las posibles consecuencias del calentamiento global, se está considerando la necesidad de implementar medidas de adaptación para garantizar la

disponibilidad de agua para la producción y consumo de alimentos. Los sensores comerciales típicos para sistemas de riego agrícola son muy caros, lo que hace imposible que los agricultores más pequeños implementen un sistema de irrigación con IoT. Sin embargo, los fabricantes actualmente ofrecen sensores de bajo costo que se pueden conectar a nodos para implementar sistemas asequibles para la gestión del riego y el monitoreo de la agricultura [23].

Las soluciones de riego automático, basadas en controladores configurados manualmente, compuestas por temporizadores y riego todos los días a la misma hora y duración, son la solución básica y la opción preferida. Este tipo de solución sigue siendo la opción principal en alrededor del 80 % de los proyectos de riego debido a los bajos precios y la simplicidad, pero este tipo de solución tiene poca eficiencia en lo que respecta a la gestión del agua; es común verlos iniciar el proceso de riego en períodos de lluvia, calor excesivo o vientos fuertes, condiciones que contribuyen al desperdicio de agua [24].

En este sistema, se utiliza una red inalámbrica para conectar un sensor de temperatura y sensores de higrómetros o de humedad del suelo colocados en la zona de las raíces de las plantas. Además, se emplea un sensor de nivel de agua para detectar el nivel del agua en el suelo [25].

Estos sensores se pueden usar para proporcionar la cantidad correcta de agua en el momento adecuado que se puede proporcionar a las plantas sin necesidad de mano de obra. La temperatura varía día a día, no se puede hacer constante fácilmente, pero mediante el uso de esta técnica podemos obtener la temperatura casi constante que requiere la planta. Los datos de temperatura, humedad y cantidad de agua se envían a IOT.

El riego convencional puede ser sustituido por un sistema automatizado utilizando las tecnologías actuales. En este método, un sensor de humedad del suelo se coloca cerca de las raíces de las plantas para detectar la humedad y transmitir esta información a un microcontrolador que controla el flujo de agua hacia las plantas.

El uso de un sistema de riego manual puede resultar en la pérdida de electricidad y agua si el agricultor olvida apagar el motor. Para resolver este problema, el sistema de riego inteligente puede encender y apagar automáticamente el motor según las condiciones del clima, del suelo y de la humedad. Los parámetros de riego se determinan utilizando diferentes sensores, como el sensor de humedad del suelo, el sensor de temperatura y humedad. Para asegurarse de que el nivel de humedad en el campo esté en el rango óptimo, las condiciones del suelo se monitorean periódicamente.

Para reducir la intervención de los agricultores y convertir este proceso en uno automatizado, se utilizan muchos sensores, como el sensor de humedad del suelo y sensores de temperatura, etc., y la salida de estos sensores se conecta a una tarjeta programable. La tecnología avanza día a día lo que a su vez reduce las complicaciones y las amenazas involucradas en el proceso.

Los sistemas integrados y de microcontroladores brindan soluciones para muchos problemas. Este sistema controla automáticamente el sistema de agua para riego mediante el uso de un sistema de microcontrolador basado en sensores. Esto se puede hacer colocando sensores para supervisar la temperatura y la humedad del suelo que pasa la información al Microcontrolador. Este sistema suministra el flujo de agua automáticamente mediante.

El objetivo de este sistema de riego inteligente es desarrollar un mecanismo de riego completamente automatizado que encienda y apague el motor automáticamente al detectar el contenido de humedad de la tierra agrícola usando el sensor de humedad del suelo sin la participación directa de mano de obra [26].

4.5. Convertidor Analógico Digital

Un ADC es un componente electrónico que convierte una señal continua en el tiempo y en amplitud, lo cual significa que puede convertir una amplia gama de valores en una señal discreta en el tiempo. Esta señal discreta tiene su amplitud cuantificada y codificada, generalmente utilizando un sistema binario [27]. El propósito principal de esta conversión es simplificar el procesamiento de la señal, haciéndola menos susceptible al ruido y otras interferencias que afectan más a las señales analógicas. Básicamente, un ADC tiene como objetivo convertir una señal eléctrica analógica en un número digital equivalente. Por otro lado, un DAC (Convertidor digital-analógico) transforma un número digital en una señal eléctrica analógica [28].

El esquema representado en la figura 2 muestra el proceso desde que la variable física entra al sistema hasta que se convierte en una señal digital mediante un código binario. Para que esta señal pueda ingresar al convertidor analógico-digital, debe ser muestreada, lo que implica tomar valores discretos en momentos específicos de la señal analógica, conocido como muestreo. Matemáticamente, esto equivale a multiplicar la señal analógica por una secuencia de impulsos con un intervalo de tiempo constante. Como resultado, se obtiene una serie de impulsos cuyas amplitudes están limitadas por la envolvente de la señal analógica [27][28].

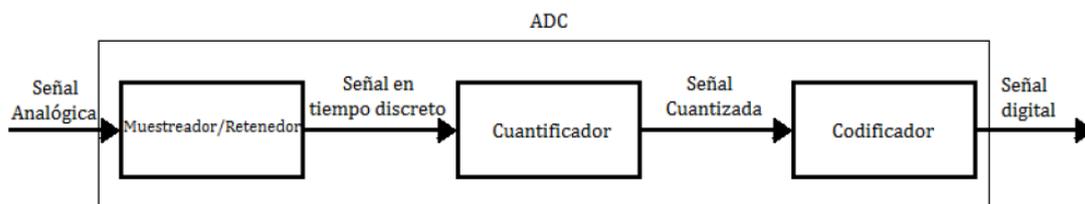
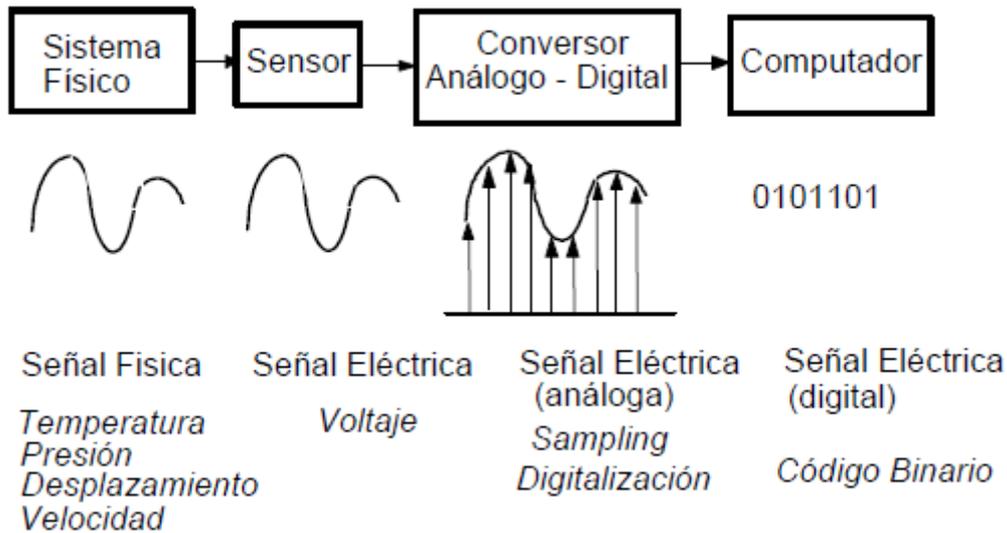


Figura 2. Proceso de un ADC

La digitalización o conversión analógica-digital implica realizar mediciones periódicas de la amplitud de la señal y luego traducirlas a un formato numérico. Los distintos procesos involucrados en la conversión analógica-digital se ilustran en la figura 2.

Conceptos Generales

Como se puede apreciar en la figura 2 para el proceso de conversión se divide en tres fases que son muestreo, cuantificador, y codificador.

- **Muestreo**

El muestreo de una señal continua en el tiempo implica obtener una serie de muestras o fragmentos de la señal original en momentos específicos. Por lo general, el muestreo se realiza de manera regular, donde los momentos de muestreo ocurren a intervalos de tiempo fijos. Este intervalo de tiempo se conoce como período de muestreo (T_s). El método de muestreo más comúnmente utilizado es el muestreo uniforme, lo que significa que las muestras se toman en intervalos de tiempo con la misma separación (T). Esto se puede expresar como[29]:

$$g(t) = f(kT); \leftarrow -\infty < k < +\infty,$$

Donde T es el periodo de muestreo, $f_s = 1/T$ es la frecuencia de muestreo y k el número de muestra.

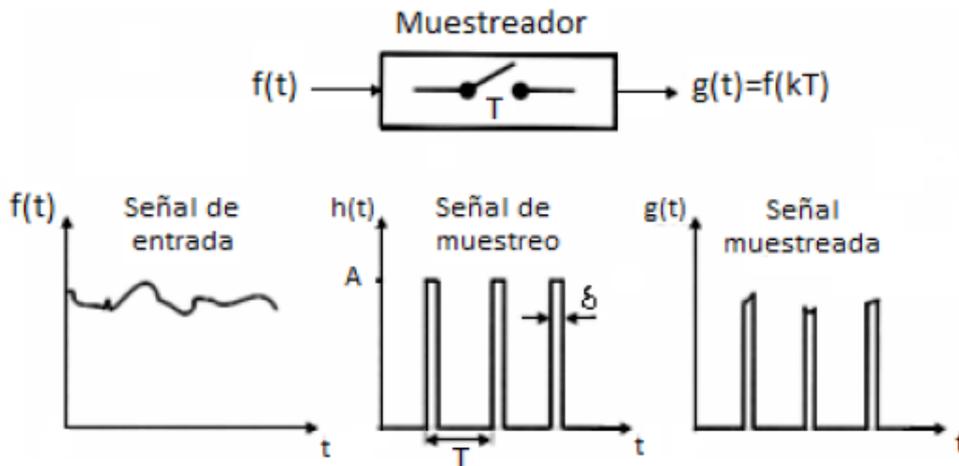


Figura 3. Proceso de Muestreo

- **Cuantificador**

La cuantificación es el procedimiento de convertir una señal en tiempo discreto, que tiene una amplitud definida en un intervalo continuo, en una señal en tiempo discreto que solo está definida para un conjunto de valores discretos de amplitud. La señal discreta se restringe a un número finito de posibles amplitudes, lo que resulta en una discrepancia entre la señal cuantificada y la señal original, conocida como error de cuantificación. En resumen, la cuantificación es un proceso en el cual se convierte una señal muestreada en un conjunto limitado de estados (N), que dependen del número de bits (n) utilizados. Normalmente, estos estados son potencias de 2, donde el exponente corresponde a la resolución en bits del convertidor [29][30].

- **Codificador**

La codificación es el procedimiento mediante el cual se representa con un código binario los distintos estados de un cuantificador, ya que este código es interpretado por los procesadores

digitales. En la codificación, se asigna un código a cada muestra de la señal, dependiendo del nivel de cuantificación al que pertenezca, es decir, en qué intervalo de cuantificación se encuentre. Los métodos de codificación más comúnmente empleados son el binario natural, el BCD (código decimal codificado en binario) y el código Gray [30].

ADC en placa programable MCU Esp 32

Al leer un valor analógico con el ESP32, se puede medir niveles de voltaje que varían entre 0 V y 3,3 V. A la tensión medida se le asigna un valor numérico que va de 0 a 4095, donde 0 representa 0 V y 4095 representa 3,3 V. Cualquier tensión dentro del rango de 0 V y 3,3 V se asignará al valor correspondiente en ese rango. Para leer valores de los puertos analógicos del ESP32, se puede utilizar la función "analogRead(GPIO)". Estos pines de entrada analógica tienen una resolución de 12 bits, lo que significa que su rango de lectura puede variar de 0 a 4095[31].

4.6. Arquitecturas de IoT para irrigación

Actualmente, la estandarización de una arquitectura para la IoT sigue siendo un proceso en desarrollo. La tendencia ha sido centrarse en la solución de dos problemas principales: encontrar una forma estándar de acceder a los dispositivos y al medio, y encontrar formas de integrar los dispositivos a internet [27]. Según [28] No existe una arquitectura única que cumpla con todas las necesidades de IoT por lo que varios investigadores han propuesto varias arquitecturas que aborden cuestiones específicas de arquitecturas genéricas. A continuación, se presentan varios ejemplos de arquitecturas propuestas para solventar las necesidades de la agricultura inteligente con IoT y para los sistemas de irrigación.

4.6.1. Arquitectura multicapa.

Dentro de la arquitectura cada capa muestra el nivel de abstracción mismos que le permite a cada capa aislarse de las responsabilidades de sus capas vecinas, lo que permite desacoplar y reutilizar los componentes del sistema. La arquitectura multicapa se divide en varias arquitecturas como lo son la arquitectura de tres capas, cuatro capas y cinco capas:

4.6.1.1. Arquitectura tres capas:

Según [29], consta con las capas de redes de sensores, de transmisión de res, centro de monitoreo y análisis de datos:

- **Capa de sensores.** - se ocupa del problema de que detectar y cómo detectar los datos por medio de los sensores.

- **Capa de transmisión de red.** - Es la capa encargada de transferir los datos de los sensores a la capa superior mediante un canal de transporte puede ser wifi, red satelital o GSM.
- **Centro de monitoreo y análisis de datos.** - Es la encargada del procesamiento de los datos obtenidos y transformarlos a un lenguaje estándar.

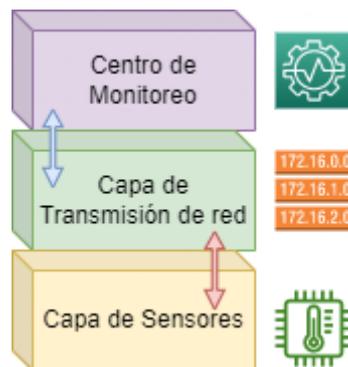


Figura 4. Arquitectura tres capas

4.6.1.2. Arquitectura cuatro capas:

Según [30] consta con las capas de sensores, puerta de enlace, servicio de administración y aplicación:

- **Capa de sensores.** - Es la capa más baja que consta de redes de sensores del sistema. Cada sensor presente en esta capa cuenta con identificador, almacenamiento y recopilación de información.
- **Capa de puerta de enlace de acceso y capa de red.** - Es la encargada de transferir la información recopilada por los sensores a las capas superiores. Es una de las capas más importantes por lo tanto debe admitir un protocolo escalable, flexible y estándar para la transferencia de datos desde varios dispositivos heterogéneos.
- **Capa de servicio de administración.** - Es la responsable de capturar gran cantidad de datos sin procesar y extraer información relevante tanto de datos almacenados o brindando seguridad de los mismos.
- **Capa de aplicación.** - Es la capa más cercana al usuario misma que presenta un interfaz amigable para acceder a varias aplicaciones de acuerdo a las necesidades de cada usuario.

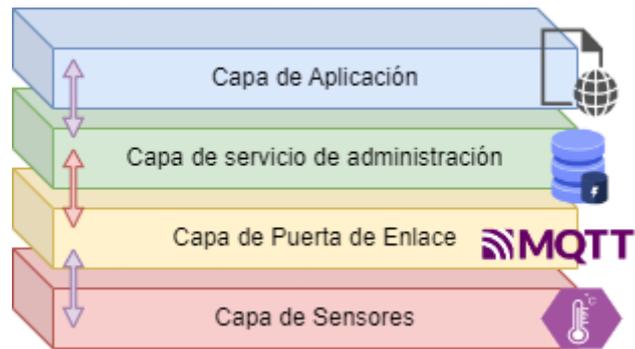


Figura 5. Arquitectura cuatro capas

4.6.1.3. Arquitectura cinco capas:

Según [31], consta de las capas de detección, de accesos, de red, middleware y las capas de aplicación:

- **Capa de detección.** – En esta capa se alojan las redes de sensores, tiene como característica principal la captura de información a gran escala así mismo compartir la información capturada a los dispositivos conectados a la red.
- **Capa de acceso.** - La capa de acceso tiene como característica principal compartir la información recolectada en la capa de sensores hacia la capa superior a través de redes inalámbricas, móviles, satelitales, entre otras.
- **Capa de red.** - La capa de red tiene como objetivo principal integrar los recursos de información de la red en una red inteligente basada en Internet, además de establecer una base de infraestructura confiable y eficiente para la gestión de servicios de aplicaciones industriales a gran escala.
- **Capa de middleware.** - Tiene como función administrar y controlar la información en tiempo real, de igual forma proporciona una interfaz de usuario para las capas superiores.
- **Capa de aplicación.** - La principal función de esta capa es integrar las funciones del sistema inferior para construir aplicaciones prácticas para varias industrias como puede ser los sistemas de irrigación.

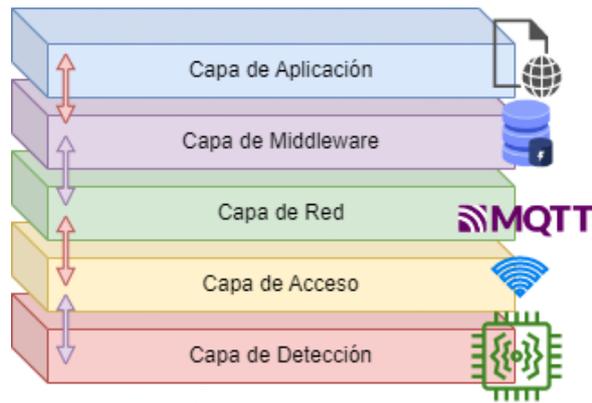


Figura 6. Arquitectura cinco capas

4.6.1.4. La arquitectura Lambda.

Se usa en sistemas que necesitan procesar y mostrar rápidamente grandes cantidades de datos de transmisión. Esta arquitectura en la nube está diseñada para manejar grandes cantidades de datos y resolver problemas complejos al combinar el procesamiento por lotes e integrar los últimos datos procesados en tiempo real. Esta arquitectura es genérica, extensible y tolerante a fallas contra fallas de hardware y errores humanos [28]. La arquitectura lambda [32] se compone de 4 componentes: (I) Data Capture que se encarga de recolectar los datos, (II) capa de procesamiento de velocidad que maneja solicitudes que requieren baja latencia utilizando algoritmos rápidos e incrementales que consideran solo datos recientes, (III) capa de almacenamiento por lotes que maneja la gestión de todos los datos adquiridos y crea un historial para ellos, y (IV) capa de consulta que presenta la información, ya sea en tiempo real o recuperando datos almacenados previamente.

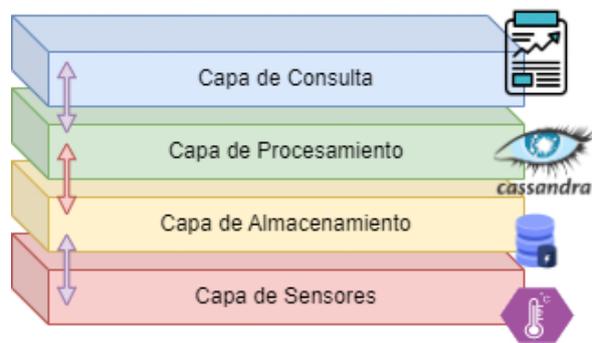


Figura 7. Arquitectura Lambda

4.6.1.5. Arquitectura Kappa.

simplicifica la arquitectura Lambda al combinar capas en tiempo real y por lotes. Esta arquitectura en la nube se diferencia de la arquitectura Lambda en el uso de un sistema de almacenamiento de datos no permanente en un archivo de registro inalterable, en consecuencia, permite solo el almacenamiento por un tiempo limitado para permitir una eventual reprocesamiento de estos datos. Las capas por lotes y de velocidad también se reemplazan por un motor de procesamiento de secuencias. Por lo tanto, la arquitectura Kappa se compone de dos capas: capas de transmisión y servicio, y se puede implementar con un mensaje de suscripción de publicación como Apache Kafka, que facilita la ingesta de datos. La principal ventaja de esta arquitectura es su sencillez. Evita tener que mantener dos bases de código separadas para las capas de velocidad y lote. Cuando el procesamiento de datos en tiempo real e históricos es el mismo, se debe utilizar una Arquitectura Kappa. La arquitectura de datos rápidos es una variante de la arquitectura Kappa en la que los datos ya no se leen de archivos sino de un mecanismo adicional como Kafka que captura múltiples flujos y los combina antes de ser procesados por la capa de velocidad [21].

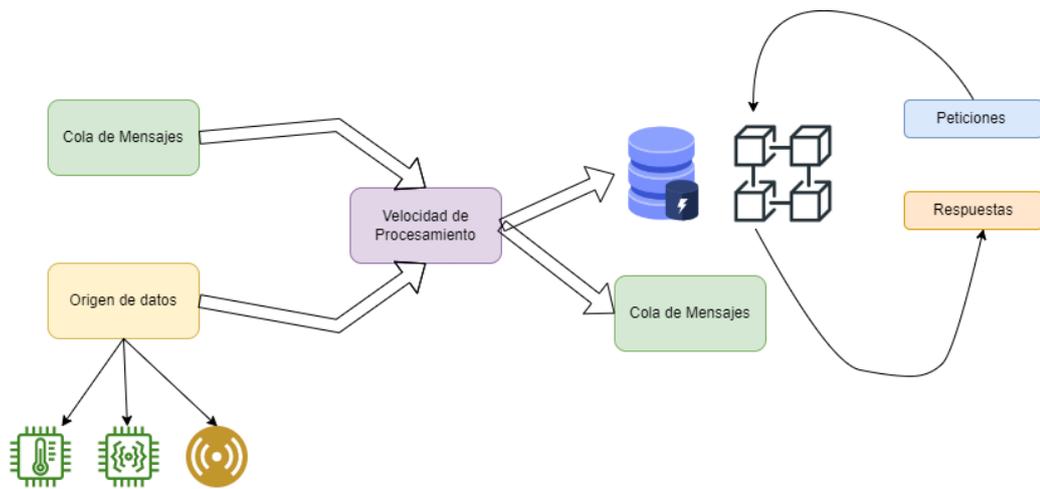


Figura 8. Arquitectura Kappat

4.7. Protocolos

Para el desarrollo de proyectos con IoT se encuentran diferentes protocolos utilizados para la comunicación de dispositivos electrónicos a la red y que son necesarios para la transmisión de información. A continuación, se realiza un análisis comparativo entre ellos:

4.7.1. MQTT

El Message Queue Telemetry Transport (MQTT) es un protocolo de comunicación de mensajes que utiliza el método publish/subscribe y se ejecuta sobre el protocolo TCP [33]. Se puede utilizar con diversos dispositivos embebidos mediante aplicaciones y mensajería asíncrona lo que lo hace adecuado para la IoT [34]. El MQTT es liviano y puede implementarse en redes con ancho de banda limitado y alta latencia. Además, su flexibilidad permite su uso en diferentes dispositivos y servicios de IoT [35].

El protocolo MQTT implementa calidad de servicio para tener confirmación de entrega de paquetes [33]. Los servicios de calidad brindados por el protocolo son:

- QoS 0, Lanzar y olvidar, no necesita comprobación, el mensaje es enviado una sola vez.
- QoS 1, Entregado al menos una sola vez, requiere un reconocimiento el mismo que puede ser enviado más de una vez.
- QoS 2, Entrega exactamente una sola vez, tiene un mecanismo de comprobación por la cual solo se entrega exactamente una vez el mensaje, sin repeticiones.

En resumen, el protocolo MQTT es especialmente adecuado para su uso en redes de baja velocidad, poco confiables, con alta latencia y dispositivos con recursos limitados. Esto se logra al minimizar los requisitos de recursos de los dispositivos conectados a la red, mientras se mantiene un alto nivel de seguridad entre ellos. [36].

4.7.2. Hypertext Transfer Protocol (HTTP)

El protocolo HTTP es utilizado para transmitir documentos hipermedia y funciona como una comunicación entre navegadores web y servidores web. Se sigue un modelo de cliente-servidor, donde un cliente abre una conexión para realizar una solicitud y espera una respuesta del servidor. A diferencia de otros protocolos, HTTP es sin estado, lo que significa que el servidor no mantiene ningún tipo de información o estado entre dos solicitudes. [30].

El protocolo HTTP se basa en un modelo de solicitud y respuesta en el que el cliente envía una solicitud al servidor para obtener información y el servidor responde con los datos solicitados. La solicitud está compuesta por un método, un recurso, algunas cabeceras y, opcionalmente, algún contenido. La respuesta, a su vez, consta de un código de estado de tres dígitos, algunas cabeceras y, también opcionalmente, algún contenido adicional [37].

4.7.3. CoAp

El protocolo de comunicación CoAP es utilizado para la comunicación de dispositivos electrónicos de baja potencia como sensores o controladores (raspberry pi, arduino) [38]. Este protocolo es una derivación del protocolo HTTP, con una diferencia que opera sobre UDP y tiene una simple capa de mensajería para la retransmisión de los paquetes pedidos en lugar de utilizar complejos controles de congestión que son propios de TCP. El primer beneficio que tiene el protocolo CoAP es que cuando se interconecta con HTTP se puede aprovechar de la estructura REST misma que involucra las comunicaciones con el cliente, gateway, servidores. Cabe destacar que existen carencias dentro de los servicios de calidad QoS en las redes de sensores inalámbricos, esto por la falta de fiabilidad de los enlaces inalámbricos de baja potencia [39].

El protocolo CoAP presenta una serie de características, tales como el uso de la conexión UDP con soporte opcional de fiabilidad, así como la capacidad de transmitir mensajes tanto de forma unicast como multicast de manera asíncrona. Además, también es fácil de mapear a HTTP y cuenta con una identificación de recursos mediante URI [30].

4.7.4. Tabla comparativa de protocolos de IoT

Tabla 1. Tabla comparativa de protocolos de IoT

Protocolos	Calidad de Servicio	Conexión	Seguridad	Arquitectura
<i>MQTT</i>	Garantiza la calidad de servicio para la entrega de paquetes	TCP	SSL/TLS	Publish/Suscriptor
<i>HTTP</i>	Comunicaciones de largo alcance, muy bajo consumo y bajo bitrate.	TCP	SSL/TLS	Cliente/Servidor
<i>CoAP</i>	Deficiente calidad de servicios por la falta de fiabilidad de enlaces inalámbricos de baja potencia	UDP	DTLS	Cliente/Servidor

4.8. Componentes de hardware

4.8.1. Selección de sensor de humedad

El método más comúnmente utilizado para determinar cuándo regar es mediante el monitoreo de la disminución del agua en el suelo alrededor de la zona de la raíz. A medida que la planta crece, utiliza el agua disponible en el suelo, lo que resulta en una disminución de la humedad del suelo. Si la humedad desciende por debajo del nivel necesario para el riego o si las plantas comienzan a sufrir estrés hídrico, es necesario aplicar agua para evitar que mueran de sed. Si no se aplica agua, la planta seguirá utilizando la poca humedad que le queda hasta que finalmente utilice toda el agua disponible en el suelo y muera por falta de agua. [12].

Para realizar el monitoreo de humedad de suelo se pueden utilizar sensores Higrómetro FC-28 o sensores capacitivos, mismos que detectan cambios en la humedad del suelo mediante variaciones de conductividad en la tabla 2, se muestran características de dos sensores higrómetros o denominados sensores de humedad del suelo.

Tabla 2. Tabla comparativa de sensor de humedad

Sensores	Voltaje	Señal de salida	Señal	Precio
<i>FC-28</i>	3.3v - 5v DC	Analógico/digital	0 a VCC/3.3V/5V TTL	3.50
<i>Sensor Capacitivo</i>	3.3V - 5V DC	Analógico	0 a 5V	4.00

Como se puede apreciar los sensores de humedad del suelo trabajan con voltajes de 3 a 5v de corriente continua, pero la diferencia de estos se presenta en el envío de señales, dado que el sensor fc-28 envía señales tanto digitales como analógicas lo que permite variar el resultado de acuerdo a las necesidades de los usuarios y en cambio el sensor capacitivo envía señales analógicas que van de 0 a 5V.

4.8.2. Sensor de humedad FC-28

En la figura 9 se muestra el sensor FC-28 que mide la humedad del suelo mediante la variación de su conductividad. Es empleado en sistemas de riego automatizados para determinar cuándo se deben activar los sistemas de bombeo de agua para los cultivos. [40].

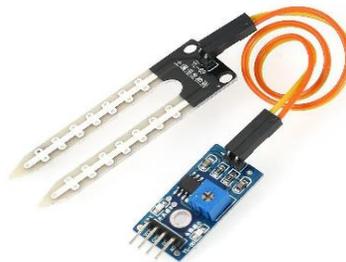


Figura 9. Sensor de humedad FC-28

4.8.3. Sensor de caudal

Un sensor de flujo o caudalímetro, se utiliza para medir el caudal volumétrico. El caudal es el volumen de líquido que circula por una tubería en un tiempo determinado, normalmente se expresa como: litros por minuto (l/m), litros por hora (l/h) o metros cúbicos por hora. Los sensores de flujo se utilizan para medir el flujo de agua. Existen varios tipos de sensores de flujo mismos que cuentan con diámetros de tubería de acuerdo a las necesidades. Los sensores

de caudal contienen un sensor de efecto hall que emite pulsos a través de uno de los cables del sensor [41]. En la Figura 10 se muestra el sensor de caudal YF-S201 que utilizado para medir el caudal de agua utilizados en sistemas de irrigación.



Figura 10. Sensor caudal

4.8.4. Electroválvula

Las electroválvulas son dispositivos electromecánicos diseñados para controlar el flujo de líquido que fluye por un conducto. Estas válvulas se mueven por la acción de una bobina solenoide, lo que permite tener las posiciones de abierto y cerrado, lo cual es ideal para el sistema de irrigación que permitirá controlar el flujo de agua que pasarán por los conductos necesarios para los cultivos [42]. En la figura 11 se presenta una electroválvula de 12v que cuenta con 2 pines para la conexión de corriente, lo cual le permitirá abrir la llave cuando reciba un voltaje de 12v y la cierre cuando reciba 0v.

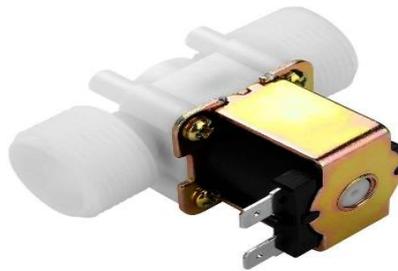


Figura 11. Electroválvula

4.8.5. Modulo Relay

El módulo relay es un dispositivo que actúa como un interruptor controlado por un circuito eléctrico y funciona mediante la activación de un electroimán para abrir o cerrar un circuito eléctrico. El módulo relay se utiliza para abrir y cerrar el paso de corriente a la electroválvula [43]. En la figura 12 se muestra un módulo relay que cuenta con 3 pines uno para voltaje de 5v, uno GND o tierra y el tercero para el control de voltaje.



Figura 12. Módulo Relay

4.8.6. Microcomputador Raspberry pi

Raspberry pi es un pequeño computador del tamaño de una tarjeta de crédito, que permite ser utilizado en varios proyectos electrónicos, ya que es capaz de ejecutar un sistema operativo Linux o ser utilizado como servidor, ya que brinda las funcionalidades de un computador normal, lo que facilita su configuración y manejo del mismo [44]. La raspberry pi es utilizada en sistemas de irrigación con IoT siendo utilizada como bróker para el envío de información mediando un servidor configurado en la misma. En la figura 13, se muestra la placa raspberry pi, que cuenta con 4 puertos USB, un puerto HDMI, puerto para alimentación de 5 voltios, una ranura para insertar micro SD en donde se almacenará el sistema operativo.

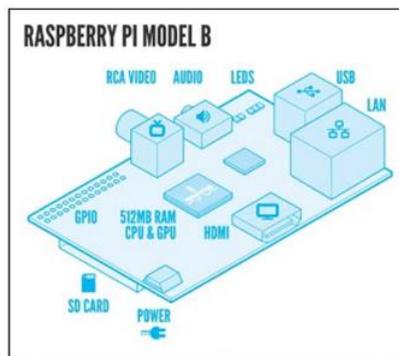


Figura 13: Raspberry pi

4.8.7. Micro controlador Esp32

Se requiere un dispositivo de comunicación para transmitir la información recopilada por los sensores, que establezca la conexión con una red y transfiera los datos a los componentes de recepción y almacenamiento del sistema. Para este propósito, se utiliza el módulo Esp32.

El ESP32 es una tarjeta programable con microcontrolador integrado (System On Chip) fabricado por TSMC. Es una placa programable que combina tecnologías WiFi y Bluetooth, y se utiliza como puente para conectarse a redes y soluciones de IoT [45]. Además, la serie ESP32

tiene la capacidad de ejecutar aplicaciones en tiempo real, lo que lo hace útil para desarrollar proyectos de IoT. La placa ESP32 de la figura 14 cuenta con pines analógicos, digitales, RTC y otros componentes.



Figura 14. Modulo ESP 32

4.9. Tecnologías de desarrollo de software

4.9.1. JavaScript

JavaScript es un lenguaje de programación que se emplea principalmente para crear páginas web dinámicas. Este lenguaje es interpretado, lo que significa que los programas no necesitan ser compilados antes de su ejecución. Los programas que se escriben con JavaScript pueden ser probados directamente en los navegadores sin requerir procesos intermedios.

JavaScript permite utilizar variables, de tipos numéricos, cadenas de caracteres o booleanos. Al declarar una variable no se le asigna un tipo, sino que es el propio intérprete le asigna según el valor [46].

Se puede declarar de dos maneras:

- Asignándole directamente un valor, por ejemplo, `x = 42`.
- Utilizando la palabra clave `var`, por ejemplo, `var x = 42`.

JavaScript es un lenguaje orientado a objetos que incluye un conjunto de propiedades que definen al objeto y una serie de métodos que permiten interactuar con el objeto, como funciones o procedimientos.[47].

Un ejemplo: Empleado de una empresa, tiene una serie de propiedades, como el nombre y apellidos, sexo, DNI, fecha de nacimiento, estado civil.

Por otro lado, existe una serie de acciones que se pueden realizar con un empleado, como contratar un nuevo empleado, despedir a un empleado, pagarle la nómina, subir de categoría, entre otras funciones.

4.9.2. Node .js

El entorno de ejecución Node.js es de código abierto y se utiliza para programar en JavaScript de manera multiplataforma. Su arquitectura se basa en eventos y en un esquema asíncrono de entrada y salida denominado EventLoop, lo que permite atender numerosas solicitudes sin la necesidad de hacer cambios de contexto [48].

Cuenta con un gran número de módulos y componentes orientados y optimizados para redes, que sirven como soporte de estándares y protocolos como es DNS, HTTPS, TCP, TLS/SSL Y UDP [49].

4.9.3. MongoDB

Es uno de los sistemas de base de datos NoSQL más conocidos. Sin embargo, MongoDB en sí brinda diferentes opciones de implementación, como la capacidad de identificar documentos anidados o referenciados, usar un operador de búsqueda para contrarrestar la necesidad de realizar una unión entre documentos, indexar aplicaciones para diferentes tipos de consultas. Estas diversas opciones pueden beneficiar a los desarrolladores al proporcionar comparaciones de rendimiento de consultas entre diferentes implementaciones [50].

MongoDB es una base de datos NoSQL que almacena los datos en forma de documentos BSON (Binary JSON), lo que permite la inclusión de tipos de datos como int, long y float en comparación con el popular formato JSON. Cada documento BSON tiene uno o más campos que contienen un valor específico de un tipo de datos, que pueden ser arreglos, datos binarios y sub-documentos.[51].

4.9.4. React

React es una biblioteca de código abierto de JavaScript que se utiliza para construir interfaces de usuario. Se destaca por ser una herramienta simple, declarativa, fácil de combinar y utiliza el patrón de diseño Modelo-Vista-Controlador (MVC) [52].

4.9.5. Flutter

Es un kit de desarrollo software de interfaz gráfica open source creado por Google. Permite crear compilados de forma nativa aplicaciones para móvil, web y escritorio desde un solo código base. El soporte incorporado actualmente es para plataformas móviles iOS y Android.

Además, proporciona su propio conjunto de objetos de interfaz, representación y un motor que implementa Flutter's animation, graphics, file and network I/O entre muchas otras bibliotecas principales.

Debido a que flutter sigue una arquitectura abierta y en capas, los desarrolladores pueden crear sus propios widgets que componen otros widgets en cualquier nivel de la arquitectura en capas. El equipo de Flutter hizo los widgets de alto nivel existentes y no hay barrera del marco para que los desarrolladores hagan lo mismo. Esta flexibilidad de personalización es inigualable. por kits de herramientas de interfaz de usuario de iOS o Android con implementaciones jerárquicas y limitado niveles de acceso [53].

4.9.6. Lenguaje Dart

Dart VM en modo de depuración es capaz de operar en modo de compilación JIT (Just-in-time) cargar y compilar dinámicamente el código fuente de Dart para facilitar la recarga en caliente, la depuración y otras características para mejorar el ciclo de desarrollo rápido. En el modo de lanzamiento, Dart VM sirve como una biblioteca de tiempo de ejecución en lugar de una máquina virtual en el sentido tradicional. El código fuente de Dart está compilado AOT (Ahead of Time) y Dart VM, como tiempo de ejecución, se usa para ejecutar código de máquina precompilado con basura rápida colección, búsqueda dinámica de métodos y más soportes de tiempo de ejecución [54].

4.9.7. Mosquitto

Eclipse Mosquitto es un software de intermediación de mensajes de código abierto con licencia EPL/EDL que admite varias versiones del protocolo MQTT. Es una solución liviana y adecuada para su uso en dispositivos de bajo consumo de energía, como computadoras de placa única, servidores completos y dispositivos móviles como teléfonos, computadoras integradas o microcontroladores [55]. Debido a estas características, se utiliza comúnmente como intermediario en sistemas de IoT para la comunicación entre sensores y los usuarios que acceden a los datos.

4.10. Programación eXtrema (XP)

La metodología XP es un enfoque ágil que se enfoca en las relaciones interpersonales como un factor clave para el éxito en el desarrollo de software. Promueve el trabajo en equipo y garantiza una alta calidad de trabajo. XP se basa en la retroalimentación constante entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, soluciones implementadas de manera simple y coraje para enfrentar los cambios. [56].

4.10.1. Características

Estas son algunas características de una metodología de desarrollo de software que se basa en el ensayo y error para obtener un software que funcione adecuadamente. Esta metodología se fundamenta en principios y está orientada hacia quienes producen y utilizan el software, y promueve una participación muy activa del cliente. Además, reduce el costo del cambio en todas las etapas del ciclo de vida del sistema. Esta metodología combina las mejores prácticas para desarrollar software y las lleva al extremo. El cliente está bien definido y los requisitos pueden cambiar. El equipo de desarrollo es pequeño (2-12 personas) y altamente integrado, y tiene una formación elevada y capacidad de aprender.

4.10.2. Base de la programación eXtrema

En [57] describe que se apoya en valores, esenciales para el trabajo colaborativo del equipo:

- **Comunicación:** Se enfatiza en la comunicación continua y efectiva con el cliente, quien es una parte integral del equipo de desarrollo. El cliente determina qué características del software tienen prioridad y debe estar disponible para responder preguntas y resolver cualquier duda.
- **Simplicidad:** Diseños simples para agilizar el desarrollo y mantenimiento
- **Retroalimentación:** Las entregas son en ciclos cortos, donde es fácil reconocer las partes que no cumplen con lo requerido y centrarse en lo importante.
- **Coraje:** La programación en parejas no es aceptada por algunos administradores ya que parece que la productividad se fuese a reducir a la mitad al por ello se necesita coraje para el desarrollo en equipo, y llegar a tener excelentes resultados.

4.10.3. Fases de la Programación eXtrema

En [58] describe 4 fases, las cuales son:

- **Planeación:** Se trata de un diálogo continuo entre las partes involucradas, cliente, programadores y coordinadores.
- **Diseño:** Diseños simples y claros.
- **Codificación:** Se establece el lenguaje de programación a utilizar, en la generación de métodos y funciones que tendrá cada módulo del sistema.
- **Pruebas:** Se enfatiza en la prueba realizadas al sistema, detección y corrección de errores, pruebas de aceptación, pruebas unitarias.

4.11. Trabajos Relacionados

Resulta de gran importancia conocer los diferentes estudios que se han desarrollado sobre la temática de automatización de sistemas de riego utilizando IoT los mismos que ayudan a obtener diferentes puntos de vista de la forma en la cual se lleva dicho proceso. Para lo cual se describen brevemente los puntos más relevantes de las investigaciones relacionadas a la temática.

4.11.1. Primer Trabajo Relacionado

En la tesis titulada “DESARROLLO DE UN SISTEMA AUTOMÁTICO DE IRRIGACIÓN PARA HUERTOS CASEROS POR MEDIO DE IOT CON SOFTWARE Y HARDWARE LIBRE” de la Universidad Politécnica Salesiana de Quito, se enfoca en el desarrollo de un sistema de riego para huertos caseros que utiliza software y hardware libre. Se emplea la plataforma de código abierto Openhab para automatizar el huerto casero, permitiendo la creación de una nube privada para que los usuarios puedan interactuar con el sistema de riego de manera remota. La arquitectura IoT se integra con tres módulos Esp32 que están programados con el protocolo MQTT y se comunican a través de una Raspberry Pi, que actúa como el intermediario (broker) en esta implementación. Automatizar el sistema de riego asegura que las plantas reciban la cantidad adecuada de agua y nutrientes para su reproducción, lo que resulta en un cultivo orgánico de alta calidad que se mantiene en óptimas condiciones hasta el día de la cosecha. El desarrollo en sí mismo requiere correr riesgos, pero el resultado final es muy beneficioso [34].

En este trabajo presentan una arquitectura de cuatro capas presentada en la figura 15. Además, de una topología estrella que el protocolo MQTT tiene incorporada como se puede ver en la figura 16:

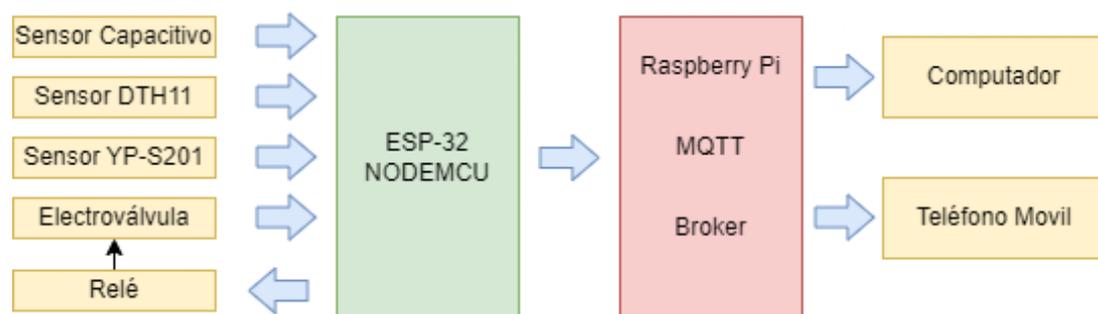


Figura 14. Arquitectura 4 capas

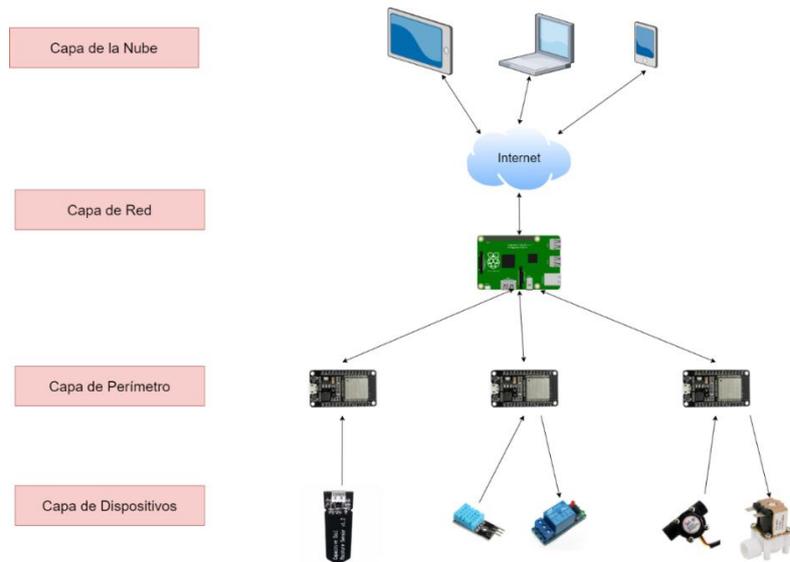


Figura 15. Topología IoT

4.11.2. Segundo Trabajo Relacionado

“ANÁLISIS, DISEÑO Y DESARROLLO DE UN SISTEMA INTELIGENTE Y AUTOMATIZADO DE MONITOREO Y CONTROL DE CULTIVOS CON IOT. CASO DE ESTUDIO: HUERTOS URBANOS” de la Universidad Central del Ecuador, se enfoca en desarrollar un sistema IoT inteligente y automatizado para jardines urbanos que monitorea regularmente las necesidades de las plantas mediante sensores, registrando información precisa en tiempo real en una base de datos en la nube. El sistema controla indicadores como la humedad del suelo, luminosidad, temperatura y humedad ambiental, y toma decisiones automáticas sobre la cantidad de agua para riego y la apertura o cierre de cortinas para mantener el jardín saludable [22].

En este trabajo presentan una arquitectura de tres capas que se la puede apreciar en la figura 17. La capa uno de elementos IoT (Things) en donde se encuentran los sensores y actuadores del sistema, la capa dos que corresponde a los servicios proporcionados en la nube y por último la capa tres corresponde a la aplicación móvil. En la figura 18 se ilustra la arquitectura física utilizada en el presente trabajo.

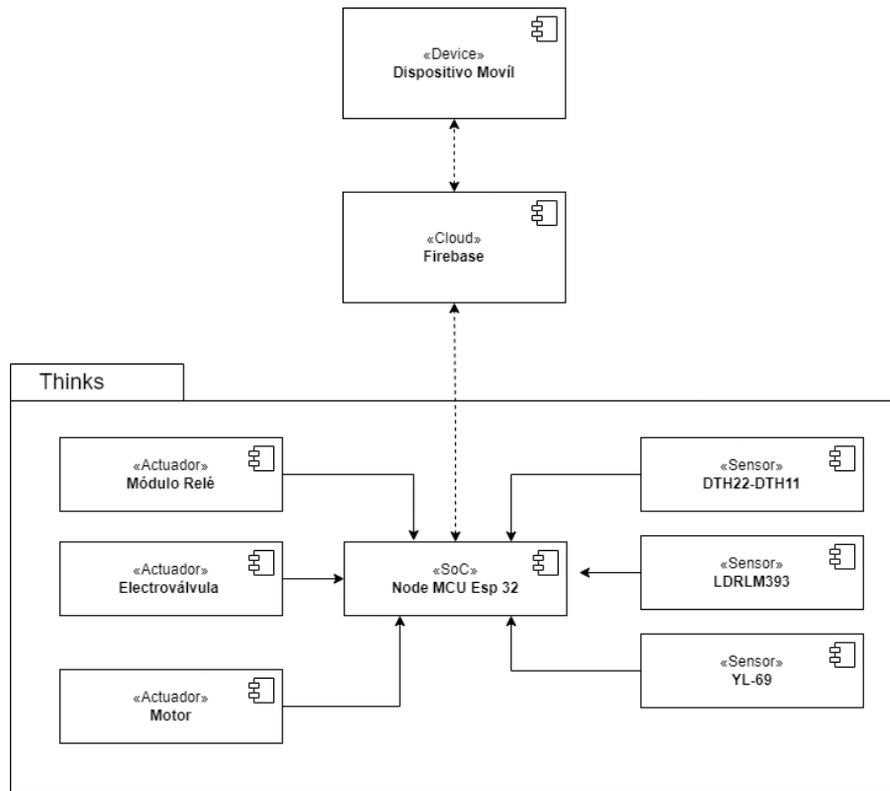


Figura 16. Arquitectura lógica (Diagrama de componentes)

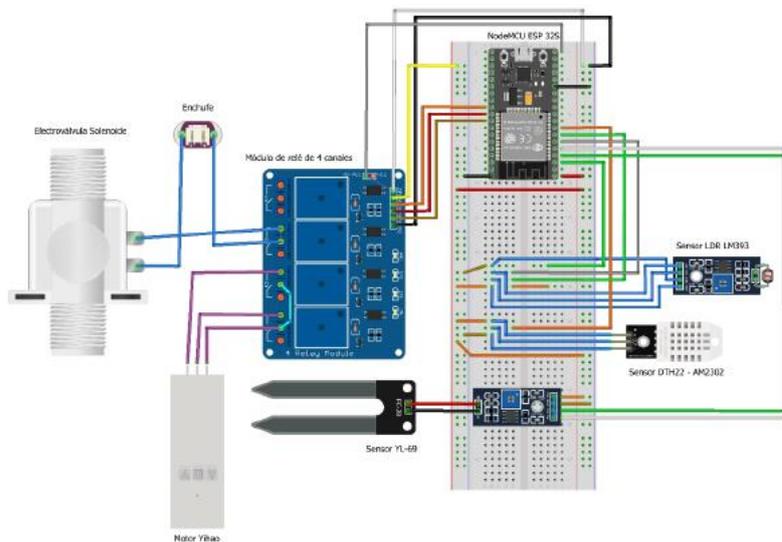


Figura 17. Prototipo de las conexiones en el Esp32

4.11.3. Tercer Trabajo Relacionado

En la investigación titulada “CONSTRUCCIÓN DE UN PROTOTIPO DE UN SISTEMA DE RIEGO Y VENTILACIÓN DE UN INVERNADERO BAJO EL CONCEPTO DE IOT” de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México. Este proyecto se enfoca en la creación de prototipos de sistemas de riego y ventilación para invernaderos utilizando el

Internet de las Cosas (IoT). El sistema permite monitorear regularmente la humedad y la temperatura en tiempo real a través de un sensor de humedad del suelo y un sensor de temperatura conectados a una tarjeta Huzzah32-ESP32. Los valores registrados se muestran en la plataforma Dashboard wia, que puede ser accesada por Internet a través de la interfaz de usuario. El objetivo de este sistema es mejorar el control de las variables ambientales para mejorar significativamente el crecimiento de los cultivos en el invernadero [60].

La arquitectura propuesta en el presente trabajo se compone de cuatro capas: La capa de sensores, capa central, capa de interfaz eléctrica, y la interfaz de usuario. Esta se la puede ver en la figura 19:

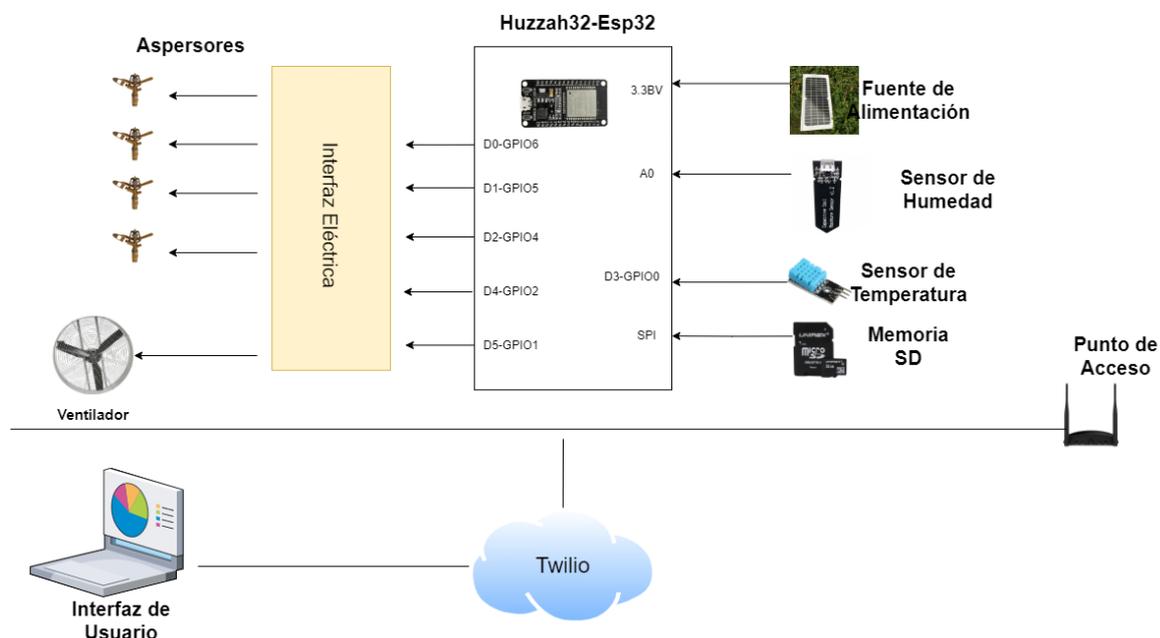


Figura 18. Prototipo de sistema de irrigación y ventilación de un invernadero

4.11.4. Cuarto Trabajo Relacionado

“ANÁLISIS DEL INTERNET DE LAS COSAS (IoT) Y SU APLICABILIDAD EN EL CONTROL REMOTO DEL SISTEMA DE RIEGO DE LA FINCA LA JOSEFA UBICADA EN MACONDO ZONA BANANERA DEL DEPARTAMENTO DEL MAGDALENA” de la Universidad Cooperativa de Colombia Centrado en el análisis de IoT y la idoneidad para el control remoto de los sistemas de riego agrícola para proporcionar y poseer procesos automatizados, problemas que surgen de áreas geográficas a las que es difícil acceder debido al entorno de los agricultores y al aislamiento de un mundo globalizado [61].

4.11.5. Quinto Trabajo Relacionado

En el artículo titulado “SISTEMA DE INTERNET DE LAS COSAS (IoT) PARA EL MONITOREO DE CULTIVOS PROTEGIDOS” Este proyecto propone la implementación de un sistema IoT para monitorear el crecimiento de cultivos protegidos. Para ello, se recopilan datos relacionados con el crecimiento de los cultivos y se envían a un servidor para su procesamiento mediante protocolos y procesos de IoT. La finalidad es obtener información en tiempo real para su análisis y permitir que el sistema y los agricultores tomen decisiones. Además, los usuarios pueden interactuar de forma remota con el sistema y recibir alertas y condiciones específicas [62].

En el presente artículo presenta una arquitectura cliente servidor dividida en tres capas: capa de captura de datos, capa de administración y procesamiento (servidor) y la capa del cliente como se muestra en la figura 20.

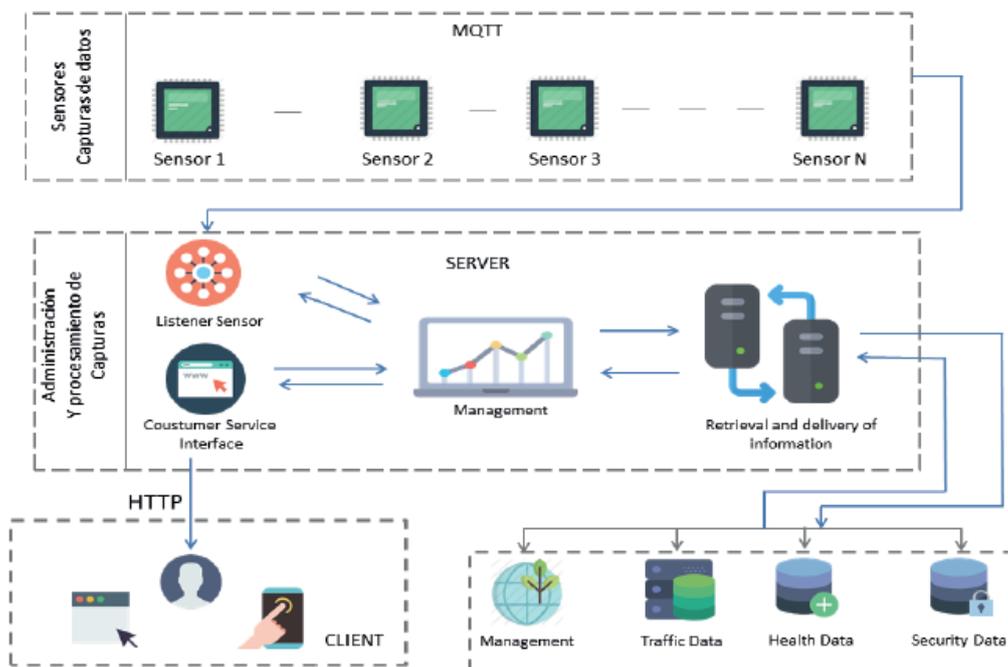


Figura 19. Arquitectura de tres capas en sistema de monitoreo de cultivos

5. Metodología

Para cumplir con los objetivos del presente Trabajo de Titulación, se aplicaron un conjunto de métodos y técnicas, que permitieron llevar a cabo de forma adecuada el desarrollo del proyecto. Con el uso de diferentes recursos tanto Hardware, Software, y recursos científicos (como la definición del contexto y procedimiento del TT), permitieron que el proyecto se desarrolle sin ningún inconveniente:

5.1. Área de estudio

El TT se realizó durante el periodo Abril-septiembre 2022 en la Carrera de Ingeniería Agrícola de la UNL, con la colaboración de los docentes: Ing. Diego Chamba, Ing. Romel Ramon y el tesista Wilson Girón, quienes brindaron información relevante para el desarrollo del proyecto.

5.2. Procedimiento

Para culminar el objetivo general del proyecto de titulación propuesto, se realizó el siguiente proceso para cada uno de los objetivos específicos.

5.2.1. Investigar las técnicas de IoT para la construcción de aplicaciones de irrigación

En el primer objetivo se desarrolló una Revisión Sistemática de Literatura (RSL). Una RSL es un medio para evaluar e interpretar toda la investigación disponible relevante para una o varias preguntas de investigación, sobre alguna área técnica o un tema de interés. Un RSL se compone de varias actividades discretas [63]. Por lo tanto, las etapas que se deben seguir para un RSL son:

- Planificación de la revisión (Planning): Se identifican las necesidades y se desarrolla un protocolo de revisión.
- Realización de la Revisión (Conducting): Se realiza una selección de los estudios primarios, evaluación, extracción, seguimientos y se realiza una síntesis de los datos.
- Informe de la revisión (Reporting): Se desarrolla un informe detallado de los resultados encontrados sobre las preguntas de investigación.

5.2.1.1. Planificación de la revisión sistemática sobre las técnicas de IoT para la construcción de aplicaciones de irrigación.

- En la primera etapa de la planificación de la revisión se identificó las necesidades de la investigación en este caso, buscar las arquitecturas, protocolos, herramientas y problemas, para la construcción de aplicaciones de irrigación, (Para más detalle de los resultados obtenidos, se presentan en la Sección 3.1)

- Se realizó la definición del protocolo, aquí se escogieron las preguntas de investigación, palabras claves, cadena de búsqueda y los criterios de inclusión y exclusión que serán necesarios para la selección de documentos en la primera revisión.

5.2.1.2. Desarrollar la revisión sistemática sobre técnicas de IoT para la construcción de aplicaciones de irrigación.

- En la segunda fase de la revisión se realizó la importación de documentos encontrados en bases de datos científicas.
- Se clasificó los documentos por duplicados, aceptados y rechazados, en base a los criterios de inclusión y exclusión.

5.2.1.3. Documentar los resultados obtenidos sobre las técnicas de IoT para la construcción de aplicaciones de irrigación.

En la última fase se desarrolló un documento de resultados basado en IEEE para documentar la información relevante obtenida en el desarrollo del RSL, en donde se contesta a las preguntas de investigación.

5.2.2. Diseñar la arquitectura de sistema de irrigación para recolectar los datos generados por los componentes IoT.

Para el segundo objetivo se desarrolló la arquitectura para el Sistema de Irrigación con IoT para cultivos de la Universidad Nacional de Loja.

5.2.2.1. Identificar los componentes y materiales necesarios para funcionamiento del sistema de irrigación con IoT.

Para cumplir con la fase para identificar los componentes y materiales necesarios para el funcionamiento del sistema de irrigación con IoT, se utilizó parte de los resultados del RSL (Ver Anexo 1) realizado en el primer objetivo, en donde una de las preguntas era identificar los componentes utilizados para el desarrollo de sistemas de irrigación con IoT. Con lo cual se pudo determinar los componentes que se acoplan al desarrollo del presente Trabajo de Titulación.

5.2.2.2. Diseñar la arquitectura del sistema de irrigación con IoT haciendo uso de los componentes y materiales anteriormente identificados.

Para cumplir con la etapa de diseñar la arquitectura del sistema de irrigación con IoT, se seleccionó la arquitectura basada en tres capas, que según el RSL (Ver Anexo 1) realizado en el primero objetivo es la más utilizada por los autores, esto porque se adapta a las necesidades del proyecto. Para lo cual se desarrolló el documento arquitectónico 4+1 mismo que se utilizó

para describir y realizar los diagramas UML tanto del hardware y de software (Ver Anexo 4 sección 5.3 vistas de la arquitectura).

5.2.3. Desarrollar un módulo de control que permitirá administrar el sistema de irrigación con IoT mediante la metodología XP.

Para el cumplimiento del tercer objetivo se realizó las siguientes actividades:

5.2.3.1. Planificar las etapas para el desarrollo del módulo de control y administración.

- En la etapa de planificación para el desarrollo del módulo de control y administración del sistema de irrigación con IoT, se realizó entrevistas a docentes expertos en riego de la carrera de Ingeniería Agrícola de la Universidad de Loja, así mismo se buscó trabajos relacionados y documentos sobre sistemas de riego, esto para recabar información importante para el desarrollo del módulo de control.
- Seguido, se desarrolló el documento de especificación de requisitos de software con IEE830 (Ver Anexo 3).

5.2.3.2. Diseñar el módulo de control en base de la especificación de requisitos utilizando notación UML (Unified Modeling Language)

Para la etapa de diseño se utilizó el documento arquitectónico 4+1 en donde desarrolló los modelos de cada vista mediante modelos UML tanto de software como de software (Ver Anexo 4 sección 5.3 vistas de la arquitectura).

5.2.3.3. Codificar el sistema de control y administración en base al diseño previamente establecido

- Se procedió al desarrollo del módulo de control, en donde se realizó la codificación del API-REST en el framework Nodejs, seguidamente se desarrolló la interfaz de usuario con la librería Reactjs y para el aplicativo móvil se lo desarrollo con el framework de Google Flutter.
- Luego, se realizó las pruebas unitarias del módulo de control con la herramienta postman, las pruebas de aceptación se la realizaron mediante una encuesta realizada por los ingenieros expertos en riego de la carrera de Ingeniería Agrícola de la Universidad Nacional de Loja y para las pruebas de carga y estrés de las realizo con la herramienta JMETER.

5.2.3.4. Realizar el prototipo de sistema de irrigación, implementando hardware (sensores, actuadores, tarjetas programables, microcomputadoras) y software (sistema de control y administración) haciendo uso de la arquitectura antes establecida

- Para el desarrollo del prototipo de sistema de irrigación con IoT se realizó la configuración del servidor bróker en el microcomputador raspberry pi. Seguidamente se programó los sensores de humedad de suelo, sensor de caudal y module relay, en los módulos de control Esp32.
- Llegados a este punto, se configuró la comunicación del escucha en el API-REST con la librería mqtt, para recibir los datos enviados por los sensores.
- Seguidamente se realizó la conexión de la electroválvula que va a permitir abrir el riego desde la ampliación web y móvil.

5.3. Recursos

Para culminar con los objetivos específicos, se utilizó los siguientes recursos.

5.2.4. Métodos

5.2.4.1. Analítico

El método analítico se utilizó para descomponer en varias partes el Trabajo de Titulación en varias fases para el desarrollo ya que este método de investigación consiste en la desmembración de un todo.

5.2.4.2. Investigación-Acción

El método de investigación acción permitió realizar varios ajustes (utilizar las características de suelo para determinar cuándo controlar el riego) mientras se desarrollaba el Trabajo de Titulación hasta alcanzar la solución del problema.

5.2.4.3. Experimentación

Este método permitió evaluar la solución informática y el funcionamiento de los sensores, en un ambiente controlado, utilizando varias muestras de suelo y validando que los sensores detectan el cambio de humedad en un tiempo determinado, con lo que se determina cuando abrir y cerrar el riego, esto tomando en consideración las características del suelo.

5.2.5. Técnicas

5.2.5.1. Entrevistas

La entrevista fue utilizada para obtener información sobre el proceso de riego y los factores que hay que tomar en consideración para el desarrollo del prototipo de Sistema de Irrigación con IoT. Cabe mencionar que se realizaron varias entrevistas a ingenieros expertos en riego de la carrera de Ingeniería Agrícola de la Universidad Nacional de Loja, para determinar los requisitos funcionales de la solución informática.

5.2.5.2. Reuniones

Esta técnica permitió reunirse con los expertos en 7 ocasiones, con la finalidad de presentar avances del prototipo a los expertos de riego, en donde se realizaron retroalimentaciones para culminar con éxito el desarrollo del TT.

5.2.5.3. Revisión Bibliográfica

Esta técnica ayudó a recolectar información científica y de literatura gris, en procesos de sistemas de irrigación con IoT, definiciones y trabajos relacionados para contextualizar el objeto de estudio.

5.2.5.4. Revisión Sistemática de Literatura

El RSL permitió dar contestación a varias preguntas de investigación que son importantes para el desarrollo de la solución informática. Para lo cual se buscó en bases de datos científicas artículos, revistas y conferencias, relevantes para el objeto de estudio.

5.2.6. Estándares

5.2.6.1. IEEE 830

El estándar IEEE 830 permitió establecer los requisitos funcionales y no funcionales del sistema. En base a las especificaciones brindadas por los ingenieros de la carrera Agrícola de la Universidad Nacional de Loja.

5.2.6.2. Modelo 4+1

El método 4+1 permitió tener un enfoque global de la solución informática mediante la arquitectura empleada. En el presente modelo se describió la arquitectura tanto de hardware y software a partir de múltiples vistas concurrentes y utilizando modelos UML, para prototipo de sistema de Irrigación con IoT para cultivos de la Universidad Nacional de Loja.

5.2.7. Recursos Técnicos

Las herramientas necesarias para el desarrollo del proyecto de trabajo de titulación son:

5.2.7.1. Parsifal

El aplicativo web se utilizó principalmente para llevar de forma ordenada RSL. Además de proporcionar una forma de documentar todo el proceso, la herramienta ayudó a recolectar datos importantes sobre el desarrollo de un sistema de irrigación con IoT [63].

5.2.7.2. Herramientas colaborativas

Las herramientas colaborativas son de gran importancia los cuales apoyaron en la comunicación como es Google Drive y Google Docs para editar y compartir archivos.

5.2.8. Recursos Hardware

En la tabla 3 se presentan las herramientas hardware utilizadas para el desarrollo del prototipo de sistema de Irrigación con IoT son:

Tabla 3. Herramientas hardware

<i>Herramienta</i>	Descripción
<i>Laptop</i>	Equipo fundamental para el desarrollo de la solución informática
<i>Raspberry Pi</i>	Minicomputador utilizado como servidor y encargado de la comunicación entre sensores y el usuario
<i>Modulo Esp32</i>	Tarjeta programable utilizado para la conexión de los sensores
<i>Sensor de humedad FC-28</i>	Sensor permite receptar la humedad del suelo
<i>Sensor de Caudal YF-S201</i>	Sensor que permite determinar la cantidad de agua utilizada, medida en litros o mili litros
<i>Electroválvula</i>	Electroválvula permite abrir y cerrar el caudal de agua mediante señales enviadas por el relay
<i>Module Relay</i>	Envía señales a la electroválvula para el control del riego

5.4. Participantes:

El TT fue realizado por los estudiantes Alex Santiago Nole Reyes y Angel Favian Minga Medina, estudiantes de la carrera de Ingeniería en Sistema, colectivamente con el asesoramiento del Ing. Edwin René Guamán Quinche, además con la colaboración de docentes expertos en

sistemas de riego de la carrera de Ingeniería Agrícola de la Universidad Nacional de Loja Ing. Romel Ramon, Ing. Diego Chamba y el tesista Wilson Giron, quienes validaron el funcionamiento del sistema de irrigación.

6. Resultados

En la presente sección se detalla la evidencia obtenida por cada objetivo específico, que está dividida en tres fases: En la sección 3.1 se expone los resultados del objetivo 1 en donde se detalla el proceso realizado en RSL y la contestación de las preguntas de investigación propuestas en el mismo. En la sección 3.2 se especifica los resultados del objetivo 2, en donde se presenta la arquitectura de IoT utilizada en el desarrollo del proyecto de titulación. Finalmente, en la sección 3.3 se detallan los resultados obtenidos en el objetivo 3, en donde se muestra el proceso de desarrollo de la solución informática utilizado la metodología XP.

6.1. OBJETIVO 1: Investigar las técnicas de IoT para la construcción de aplicaciones de irrigación.

Para el cumplimiento del objetivo 1 se llevó a cabo el desarrollo de un RSL y detalla un resumen de los resultados obtenidos en cada una de sus actividades. Es importante mencionar, para cumplir con el presente objetivo se categorizó en tres componentes (Arquitecturas, protocolos y materiales) que se utilizan para el desarrollo de prototipos con IoT, que son parte de las técnicas para la construcción de estos prototipos. El documento completo del procedimiento y resultados obtenidos en el RSL se encuentran en el Anexo 1 sección II resultados.

En el Anexo 2 se presenta una guía de los pasos seguidos para el cumplimiento del RSL. Para llevar a cabo el desarrollo del presente objetivo, se realizaron las siguientes actividades:

6.1.1. Planificación de la revisión sistemática sobre las técnicas de IoT para la construcción de aplicaciones de irrigación.

Para realizar la presente actividad se elaboró una revisión sistemática de literatura (RSL), basado en la metodología de Bárbara Kitchenham [64] Además se utilizó como apoyo la aplicación web Parsifal para seguir el proceso metodológico de forma ordenada de la RSL.

En el desarrollo de la RSL se tomó en consideración las siguientes preguntas de investigación:

1. ¿Qué componentes se necesitan para la construcción de aplicaciones de irrigación con IoT?,
2. ¿Qué arquitecturas de IoT son utilizadas para el desarrollo de sistemas de irrigación?,
3. ¿Qué protocolos de red son idóneos para la construcción de aplicaciones de irrigación?,
4. ¿Qué problemas se presentan en la construcción de aplicaciones de irrigación con IoT?

Además, se tomaron en consideración estudios realizados entre los años del 2016-2022, haciendo uso de una cadena de búsqueda que se generó en base a palabras claves. Estas palabras claves fueron la base fundamental para diseñar la cadena de búsqueda utilizada en varias bases de datos científicas. En la tabla 4 se puede apreciar la cadena de búsqueda y las fuentes bibliográficas utilizadas para la extracción de artículos necesarios para la RSL.

Tabla 4. Fuente bibliográfica y cadena de búsqueda

Fuentes Bibliográficas	Cadena de Búsqueda
<i>IEEE Library</i>	("IoT") AND ("Irrigation system" OR "Automated
<i>Scopus</i>	Irrigation" OR "Intelligent Irrigation System" OR
<i>ScienceDirect</i>	"Irrigation monitoring" OR "Precision irrigation"
<i>ACM Digital Library</i>	OR "Smart Farming" OR "Smart Irrigation" OR
	"Smart drip irrigation system") AND
	("Architecture" OR "Method" OR "Methodology"
	OR "Protocol" OR "Standard")

Luego se agregaron los criterios de inclusión y exclusión utilizados para descartar documentos que causen ruido o que no aporten con la información necesaria en la investigación. En la tabla 5 se presentan los criterios de inclusión y exclusión utilizados en la RSL.

Tabla 5. Criterios de inclusión y exclusión

Criterios de inclusión	Criterios de exclusión
Los Documentos deben estar escritos en ingles	Los Documentos hablan sobre agricultura, pero no especifican sobre irrigación con IoT
Los Documentos detallan las técnicas de IoT que se se pueden utilizar para la construcción de aplicaciones de irrigación	Los documentos no detallan las técnicas de IoT que se se pueden utilizar para la construcción de aplicaciones de irrigación
Los documentos deben contener información relacionada con sistemas de irrigación con IoT	Los documentos no detallan los componentes que se necesitan para la construcción de aplicaciones de irrigación con IoT
Los documentos detallan las arquitecturas de IoT para la construcción de sistemas de irrigación	Los documentos no detallan protocolos que se pueden utilizar para la construcción de aplicaciones de irrigación con IoT
Los documentos detallan los componentes que se necesitan para la construcción de aplicaciones de irrigación con IoT	Los documentos no detallan las arquitecturas de IoT para la construcción de sistemas de irrigación

6.1.2. Desarrollar la revisión sistemática sobre técnicas de IoT para la construcción de aplicaciones de irrigación.

El proceso empleado para el desarrollo de la RSL, partió desde la búsqueda de los documentos en las diferentes bases de datos científicas utilizando la cadena de búsqueda propuesta anteriormente, en donde se obtuvieron 819 documentos relacionados con el tema de investigación. Seguidamente utilizando la aplicación web Parsifal se filtraron los documentos repetidos que en un total fueron 173, quedando finalmente 646 documentos para la revisión.

Luego de descartar los documentos duplicados, se realizó una revisión de los estudios por su título, aplicando los criterios de inclusión y exclusión para descartar documentos que causaban ruido al objetivo de investigación, descartando así 604 documentos. Finalmente, con los 42 documentos restantes, se efectuó una revisión más detallada tomando en consideración el

resumen y los resultados obtenidos para descartar los artículos que no presentan información relevante. Quedando así para la revisión final del RSL 25 artículos que cuentan con información necesaria para contestar las preguntas de investigación, En la figura 21 se presenta el diagrama del proceso realizado para el RSL.

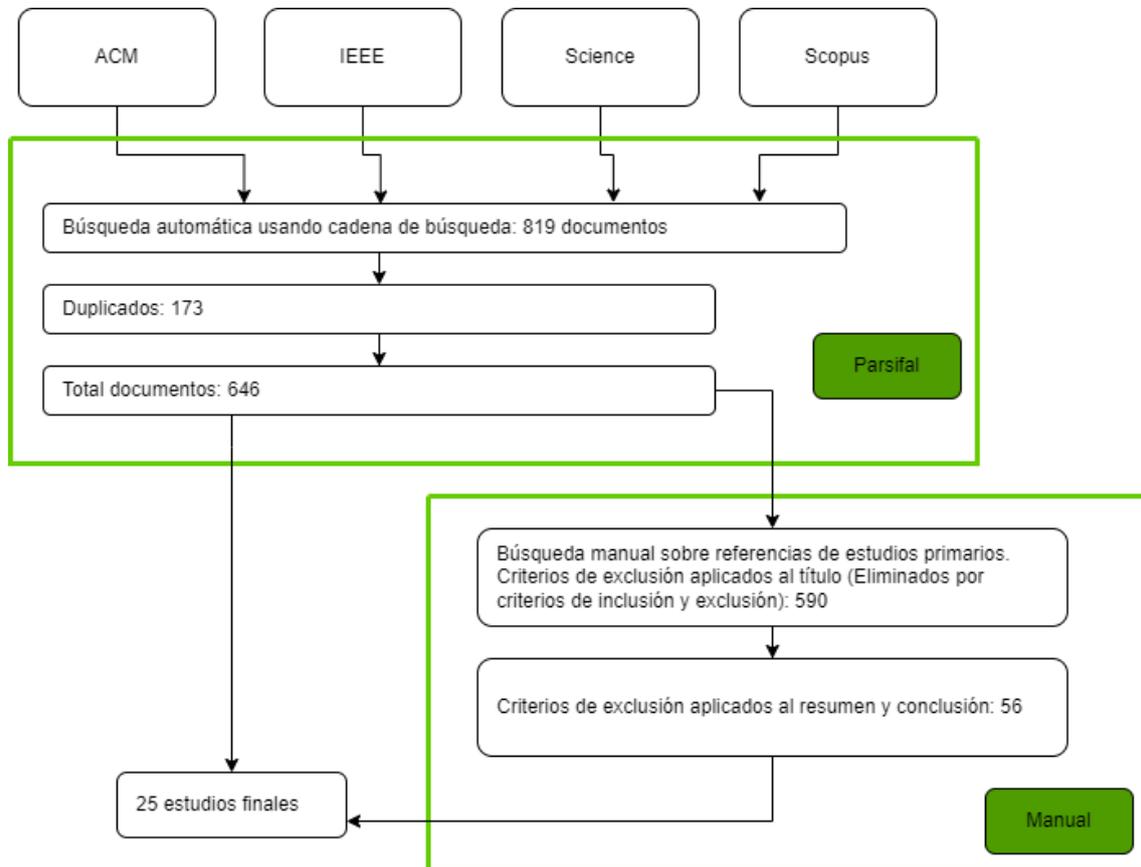


Figura 20. Diagrama de flujo paso de búsqueda

Cabe destacar que, para obtener los 25 estudios utilizados en la RSL, se analizó las arquitecturas, protocolos, herramientas, y problemas dentro de la construcción, necesarios para obtener un enfoque específico y conocer cómo se lleva a cabo la construcción de los sistemas de irrigación con IoT, así mismo, para conocer los factores que intervienen en el desarrollo de los mismos. En la tabla 6 se muestran los estudios seleccionados para el análisis completo.

Tabla 6. Artículos evaluados

Título	Autor / Año / Tipo Documento
An IoT Based Smart Irrigation System [65]	Stolojescu-Crisan, C. and Butunoi, B. and Crisan, C.; 2021; Artículo

IoT based autonomous percipient irrigation system using raspberry Pi[66]	Imteaj, Ahmed and Rahman, Tanveer and Hossain, Muhammad Kamrul and Zaman, Saika, 2016, Artículo
An automated irrigation system for optimized greenhouse using IoT[67]	E. Seethalakshmi and Meenakshi Shunmugam and R. Pavaiyarkarasi and Sneha Joseph and J. {Edward paulraj}, 2021, Artículo
Experimental and Mathematical Models for Real-Time Monitoring and Auto Watering Using IoT Architecture[68]	Yousif, J.H. and Abdalgader, K., 2022, Artículo
Smart System for Automated Irrigation Using Internet of Things Devices	Ferrarezi, R.S. and Peng, T.W., 2021, Artículo
Efficient iot-based control for a smart subsurface irrigation system to enhance irrigation management of date palm[69]	Mohammed, M. and Riad, K. and Alqahtani, N., 2021, Artículo
AgriSens: IoT-Based Dynamic Irrigation Scheduling System for Water Management of Irrigated Crops[70]	Roy, Sanku Kumar and Misra, Sudip and Raghuwanshi, Narendra Singh and Das, Sajal K., 2021, Artículo
Smart scheduling on cloud for IoT-based sprinkler irrigation[71]	Mannan J, M. and Kanimozhi Suguna, S. and Dhivya, M. and Parameswaran, T., 2021, Artículo
Low-cost IoT framework for irrigation monitoring and control[72]	Borah, S. and Kumar, R. and Mukherjee, S., 2021, Artículo
IoT based Smart Irrigation System by using ESP32 and Adafruit IO[73]	Shiva, S.J. and Palanivel, S. and Venkateswarlu, S.C., 2021, Conferencia
Internet of Things (IoT) Based Water Irrigation System[74]	Hamdi, M. and Rehman, A. and Alghamdi, A. and Nizamani, M.A. and Missen, M.M.S. and Memon, M.A., 2021, Artículo

<p>Evapotranspiration (ET)-Based Irrigation System with Internet of Things (IoT) Integration for Capsicum Annuum Farming: A Methodology[75]</p>	<p>Zulkifli, Nadiatulhuda and Iqbal, Farabi and Ramli, Arnidza and Idrus, Sevia M and Radzi, Raja Zahilah Raja Mohd and Ullah, Shafi and Hairon, Amirul Hakim and Ishak, Siti Hajar Asyikin and Mohommad, Munirah and Cruz, Jennifer Dela and Caya, Meo Vincent and Agulto, Ireneo, 2020, Artículo</p>
<p>IoT Soil Moisture Monitoring and Irrigation System Development[76]</p>	<p>Chew, Kim-Mey and Tan, Syvester Chiang-Wei and Loh, Gary Chee-Wyai and Bundan, Nancy and Yiong, Siew-Ping, 2020, Artículo</p>
<p>Review of Sensor Network-Based Irrigation Systems Using IoT and Remote Sensing[77]</p>	<p>Li, W. and Awais, M. and Ru, W. and Shi, W. and Ajmal, M. and Uddin, S. and Liu, C., 2020, Revisión</p>
<p>IoT Monitoring of Water Consumption for Irrigation Systems Using SEMMA Methodology[78]</p>	<p>López-Torres, S. and López-Torres, H. and Rocha-Rocha, J. and Aziz Butt, S. and Imran Tariq, M. and Collazos-Morales, C. and Piñeres-Espitia, G., 2020, Conferencia</p>
<p>Implementation of an Automated Irrigation System for Agriculture Monitoring using IoT Communication[79]</p>	<p>Rani, Deep and Kumar, Nagesh and Bhushan, Brij, 2019, Artículo</p>
<p>IoT based Irrigation and Water Logging monitoring system using Arduino and Cloud Computing[80]</p>	<p>Meher, Chandra Prakash and Sahoo, Adyasha and Sharma, Suraj, 2019, Artículo</p>
<p>Smart water management platform: IoT-based precision irrigation for agriculture[81]</p>	<p>Kamienski, C. and Soinenen, J.-P. and Taumberger, M. and Dantas, R. and Toscano, A. and Cinotti, T.S. and Maia, R.F. and Neto, A.T., 2019, Artículo</p>
<p>Internet of Things for Irrigation System[82]</p>	<p>Guzmán-Toloza, J.M. and Villafaña-Gamboa, D.F. and Peniche-Ruiz, L.J. and Gómez-Buenfil, R.A. and Molina-Puc, J.K. and Rodríguez-Morayta, M.J., 2019, Conferencia</p>

Automated Irrigation using Arduino sensor based on IOT[65]	Amu, D. and Amuthan, A. and Gayathri, S. Shiva and Jayalakshmi, A., 2019, Artículo
Testing topology adaptive irrigation IoT with circuits[83]	Hovhannisyan, D. and Eltawil, A. and Kurdahi, F., 2019, Conferencia
Wireless network based smart irrigation system using IOT[84]	Verma, A.K. and Bhagavan, K. and Akhil, V. and Singh, A., 2018, Artículo
Intelligent irrigation system - An IOT based approach[85]	Rajkumar, M.N. and Abinaya, S. and Kumar, V.V., 2017, COnferencia
Automation of irrigation system based on Wi-Fi technology and IOT[86]	Kranthi Kumar, M. and Srenivasa Ravi, K., 2016, Artículo
Smart Multi-Crop Irrigation System Using IOT[87]	Anbarasi M, Karthikeyan T, Ramanathan L, Ramani S, Nalini N, Artículo

6.1.2.1. Arquitecturas analizadas/utilizadas en el desarrollo de sistemas de irrigación con IoT

Dentro de la RSL se encontraron qué 19 artículos ha utilizado la arquitectura de tres capas, debido a que esta arquitectura se adapta a las necesidades de los proyectos para la automatización de sistemas de irrigación con IoT, ya que el alcance de estos proyectos solo se limita al control del riego de acuerdo a varios factores ambientales y no hacen uso de modelos inteligentes o manejo de datos con big data, para esos casos se utilizan arquitecturas como es el caso de 4 capas y de 5 capas. En la tabla 7 se presentan los artículos que utilizan las arquitecturas 3, 4 y 5 capas para el desarrollo de sistemas de irrigación con IoT.

Tabla 7. Tipos de arquitecturas

Arquitectura	Capas	Artículos que lo utilizan	# de Artículos
<i>Tres Capas</i>	Capa de sensores	[66][67][86][70][71][72][85][84]	19
	Capa de transmisión	[76][77][88][73][68][74][78][65]	
	Capa de centro de monitoreo	[82][80][87]	

	Capa de sensores		
	Capa de puerta de enlace		
<i>Cuatro Capas</i>	Capa de servicio de administración,	[69]	1
	Capa de aplicación		
	Capa de detección		
	Capa de acceso		
<i>Cinco Capas</i>	Capa de red	[68][75][81]	3
	Capa de middleware		
	Capa de aplicación		

6.1.2.2. Protocolos analizados/utilizadas en el desarrollo de sistemas de irrigación con IoT

El desarrollo de la RSL ayudó a identificar los protocolos utilizados en los sistemas de irrigación con IoT. Donde los protocolos más utilizados dentro de IoT son el protocolo ZigBee y el protocolo MQTT. El protocolo ZigBee es utilizado por los autores porque trabaja en redes de tamaño reducido, además que tiene un bajo consumo de red y baja tasa de transferencia de mensajes; este protocolo de red inalámbrico es el encargado del enlace de datos en las capas 2 y 3 del modelo OSI, que permite el enrutamiento de mensajes entre dispositivos mediante una topología malla de la red zigbee creada por el mismo protocolo. Por otro lado, el protocolo MQTT es el segundo más utilizado por los autores ya les permitió la comunicación mediante el modelo de publicador y suscriptor. Cabe destacar que este protocolo trabaja sobre TCP/IP, que permite la garantía de entrega de mensajes mediante tres niveles de calidad de servicio que pueden ser configurados de acuerdo a los requerimientos de la red, además posibilita una comunicación segura de mensajes por medio de SSL/TLS. Con este protocolo los autores implementaron comunicaciones con herramientas open-source, como es la configuración de un bróker en servidores privados, lo cual les permitió tener más libertad a la hora de realizar la configuración de varios dispositivos. En la figura 22 se presenta una vista global de los estudios que han utilizado los presentes protocolos en los sistemas de irrigación con IoT.

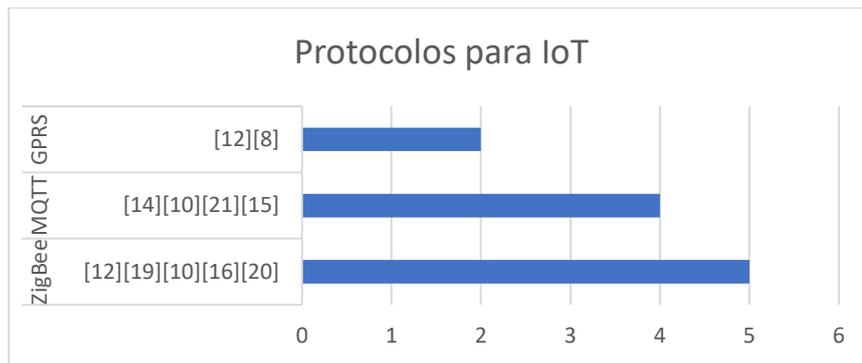


Figura 21. Protocolos para IoT

6.1.2.3. Herramientas analizadas/utilizadas en el desarrollo de sistemas de irrigación con IoT.

Para el desarrollo de sistemas de irrigación con IoT se encontró que se hacen uso de varios componentes eléctricos y electrónicos, que facilitan la obtención de datos, así como para activar el riego en horas programadas. Como es el caso de las tarjetas programables MCU Esp-32 y MCU Esp-8286, que son utilizados para proyectos con IoT, y a diferencia de Arduino estos tienen integrado wifi y bluetooth para la comunicación inalámbrica P2P. Además, la microcomputadora Raspberry Pi que es utilizada en estos proyectos para utilizarlo como servidor en donde se obtienen los datos ambientales enviados por los sensores. Y los sensores utilizados de acuerdo a la investigación realizada en el RSL son los sensores de humedad de suelo fc-28, sensor de ambiente DHT11 para capturar la humedad y temperatura ambiente. En la tabla 8 se presentan los estudios y los componentes que se han empleado en los sistemas de irrigación con IoT.

Tabla 8. Componentes electrónicos

Componentes	Tipo	# de Artículos	Descripción
<i>YL-69 y Fc-28</i>	Sensor		Sensor de humedad del suelo
<i>MCU-Esp</i>	Tarjeta		Tarjeta programable con modulo Wifi y Bluetooth, para la comunicación con los sensores,
<i>8266, ESP-32</i>	Programable		
<i>Arduino</i>	Tarjeta Programable	21	Utilizado la comunicación con los sensores, Mini computadora utilizada como servidor conectado con un
<i>Raspberry Pi</i>	Micro Computador		Microcomputador para la comunicación con los sensores y el usuario final
<i>Estación Meteorológica</i>	Sensor		Sensor de temperatura para calcular la evapotranspiración de las plantas
<i>Sensor de PH</i>	Sensor	2	Sensor de PH del suelo
<i>Relay, electroválvula</i>	Elementos electrónicos		Actuadores para habilitar el riego
<i>DTH11</i>	Sensor		Sensor de temperatura ambiente

6.1.2.4. Problemas analizados en el desarrollo de sistemas de irrigación con IoT

Los problemas o limitaciones encontradas en la revisión dentro de los sistemas de irrigación con IoT, tienen que ver principalmente con los dispositivos que se deterioran rápidamente por las condiciones climáticas, esto porque no cuentan con una protección extra, por lo que la mayoría de los autores han recurrido agregar recubierta con diferentes materiales (Impresiones 3d, cintas aislantes, recubiertas de metal) para que su tiempo de vida útil se prolongue por más tiempo. Así mismo otro de los problemas con estos sistemas es la seguridad, ya que según los autores utilizar servidores públicos, estos pueden capturar los datos enviados por los sensores y con esto pueden afectar estos datos para dañar los cultivos. De manera semejante otro problema que describen es el tiempo de implementación de estos sistemas es muy extenso por lo que se tiene que analizar el tipo de cultivos y las necesidades de estos. Lo que remite a otro problema es que implementar estos sistemas en terrenos pequeños con diferentes cultivos hacen

más complicada la implementación. En la figura 23 se muestran los problemas encontrados con los artículos seleccionados.

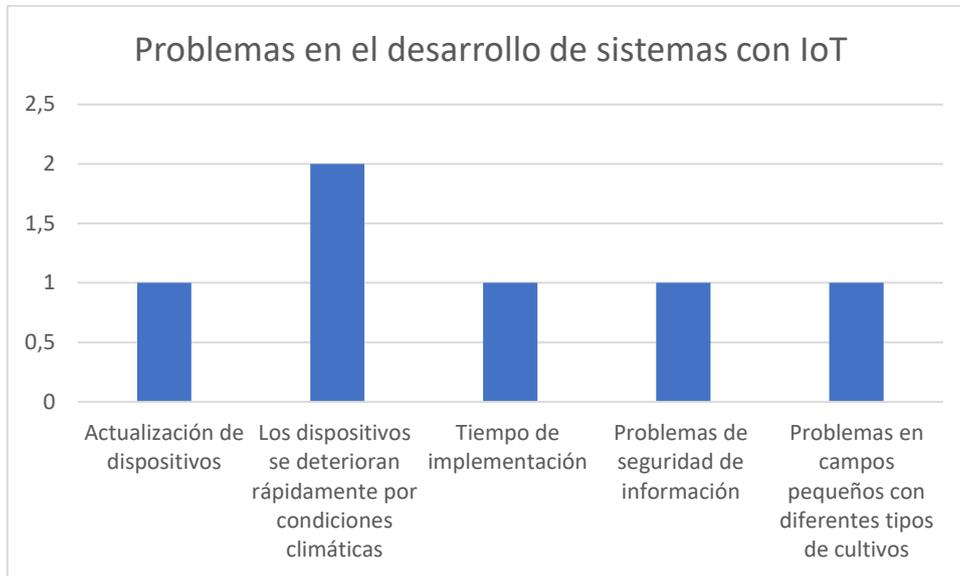


Figura 22. Problemas que se presentan en la Construcción

6.1.3. Documentar los resultados obtenidos sobre las técnicas de IoT para la construcción de aplicaciones de irrigación

Luego de dar contestación a las preguntas de investigación planteadas en la RSL, se procedió a documentar los resultados obtenidos en un artículo de revisión en el formato IEEE que está basado en IMRAD. El artículo se presenta en el Anexo 1.

Además, se documentó el proceso seguido en la aplicación web Parsifal, en donde se detallan el proceso de planificación, desarrollo y resultados obtenidos en la RSL, el documento final se presenta en el Anexo 2.

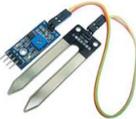
6.2. OBJETIVO 2: Diseñar la arquitectura de sistema de irrigación para recolectar los datos generados por los componentes IoT.

Para el desarrollo del segundo objetivo, se utilizó la arquitectura de tres capas, obtenida en base a la investigación realizada en el objetivo 1, el documento completo con todas las actividades y artefactos se detalla en el Anexo 1 sección RQ2: ¿Qué arquitecturas de IoT son utilizadas para el desarrollo de sistemas de irrigación?.

6.1.4. Identificar los componentes y materiales necesarios para funcionamiento del sistema de irrigación con IoT.

En la tabla 9, se presenta los materiales y componentes encontrados en la investigación realizada en el RSL objetivo 1 (Ver anexo 1 sección RQ1: ¿Qué componentes se necesitan para la construcción de aplicaciones de irrigación con IoT?), y se evidencio que los componentes usados para el desarrollo de un sistema de irrigación con IoT son:

Tabla 9. Componentes utilizados para el sistema de irrigación

Componentes y Materiales	Descripción	Figura
<i>FC-28</i>	Este sensor permite medir la humedad del suelo por medio de 2 electrodos resistivos.	
<i>Modulo Esp-32</i>	Es una tarjeta programable con microcontrolador que tiene diferentes señales analógicas, digitales, que se utiliza para programar los sensores.	
<i>Sensor Caudal YF-S2001</i>	Utilizado para medir la cantidad de líquido que se ha movido a través del mismo.	

Raspberry Pi Es un micro computador utilizado como servidor o bróker, necesario para la comunicación entre los sensores y la interfaz gráfica.



Para
la

Electroválvula Utilizada para abrir y cerrar el paso de agua.



Modulo Relay Utilizado para controlar la electroválvula por medio de pulsos eléctricos



comunicación entre los sensores y el servidor o broker, se optó por utilizar el protocolo MQTT, mediante mosquitto que es un servidor de mqtt para la comunicación entre sensores, usuarios y actuadores.



Figura 23. MQTT-Mosquitto

6.1.5. Diseñar la arquitectura del sistema de irrigación con IoT haciendo uso de los componentes y materiales anteriormente identificados

Dentro del RSL se encontró como resultado que la arquitectura utilizada en la mayoría de los proyectos con IoT, es la arquitectura de tres capas (sensores, red, aplicación), porque esta se adapta a las necesidades de los proyectos de sistemas de irrigación, en donde solo se requiere control de riego. Por lo que se optó por utilizar la arquitectura de 3 capas para el sistema de irrigación con IoT.

A continuación, se presentan los diseños de la arquitectura de tres capas, partiendo por el diseño arquitectónico de IoT, seguidamente de la arquitectura física desarrollada en fritz y finalmente el modelo arquitectónico en donde se presenta una vista global del sistema de irrigación con IoT.

6.1.5.1. Diseño Arquitectónico IoT

Aquí se presenta el diseño arquitectónico de IoT basado en tres capas, dado que esta se adapta a las necesidades de la automatización de un sistema de irrigación con IoT.

La arquitectura de tres capas empleada para el desarrollo del sistema de irrigación con IoT para cultivos de la Universidad Nacional de Loja, consta de la capa de sensores, que es la encargada de capturar los datos de humedad del suelo, que es una de las primeras características principales para programar el riego. Simultáneamente esta capa también contiene los actuadores necesarios para activar el riego de acuerdo a los datos obtenidos por los sensores.

En la capa de red, se encuentran los módulos (Esp-32 y Raspberry Pi) que realizan la comunicación mediante el protocolo MQTT. cabe destacar que el protocolo MQTT se basa en tres capas, publicador que es el que envía mensajes al broker que no es más que un servidor intermediario encargado de enviar los datos al suscriptor, este presenta los datos obtenidos desde el broker en un formato específico.

Finalmente, en la capa de aplicación se presentan los datos obtenidos por los sensores de forma atractivas mediante gráficos, en esta capa se desarrolla el aplicativo móvil, web que son los utilizados por los usuarios del sistema. En la figura 25 se muestra la representación gráfica de la arquitectura IoT empleada para el desarrollo del proyecto.

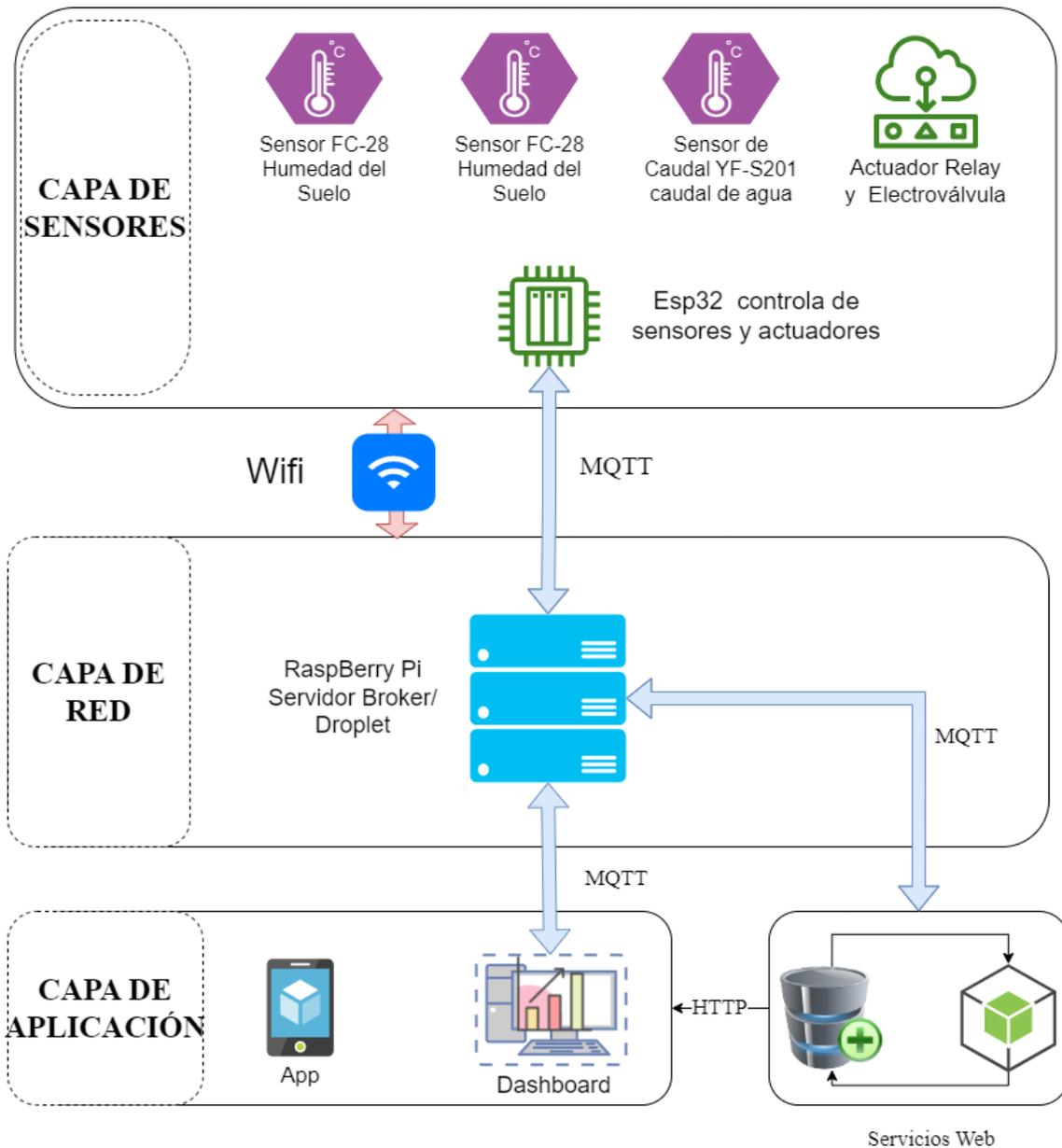


Figura 24. Diseño de la arquitectura en tres capas

6.1.5.2. Arquitectura Física

En base al diseño de la arquitectura de IoT basada en tres capas se desarrolla la arquitectura física del prototipo, utilizando los componentes seleccionados anteriormente.

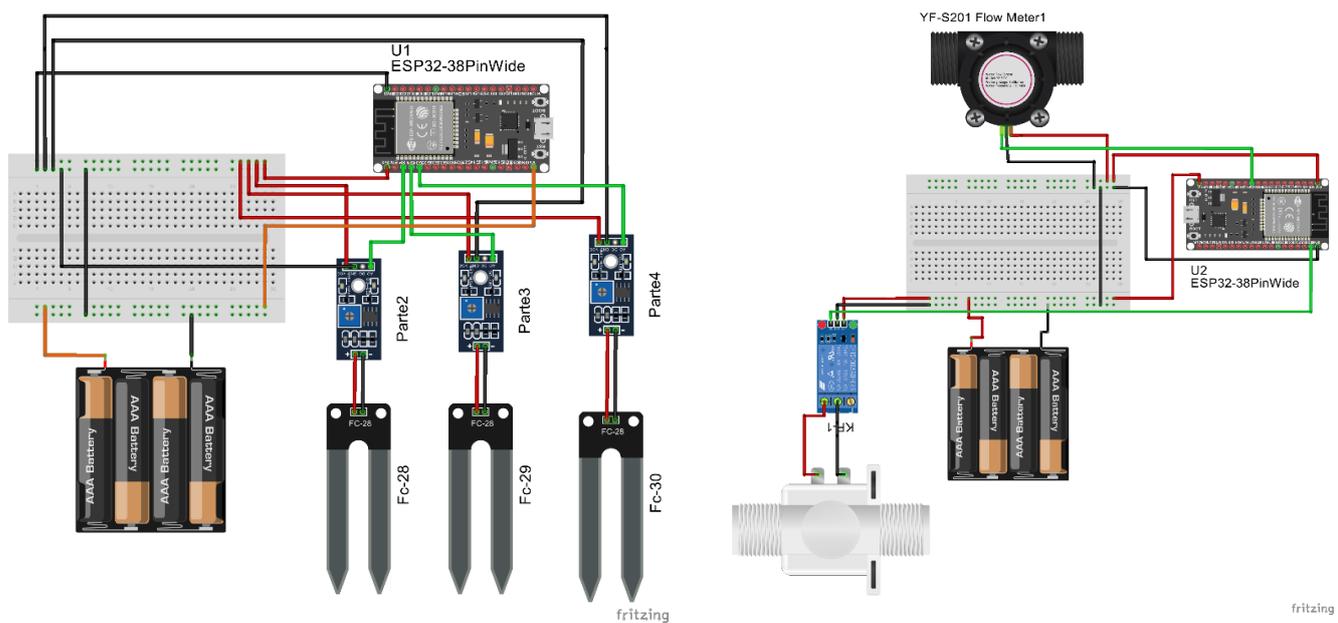
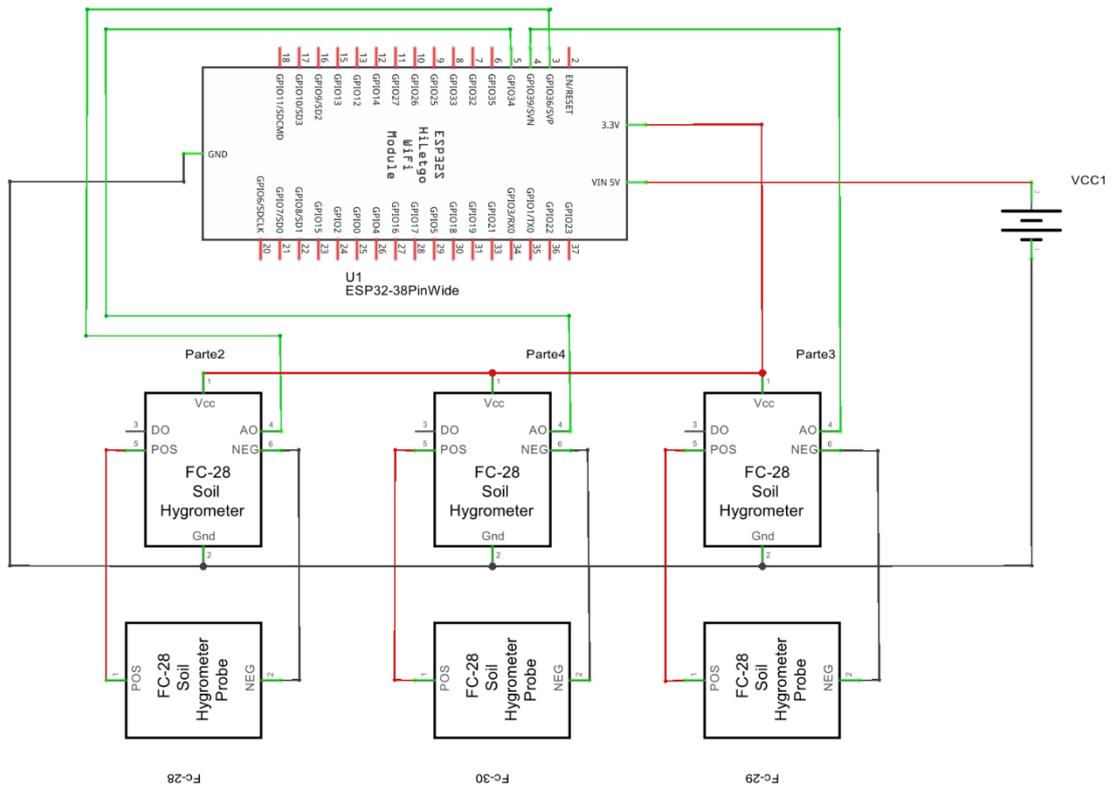
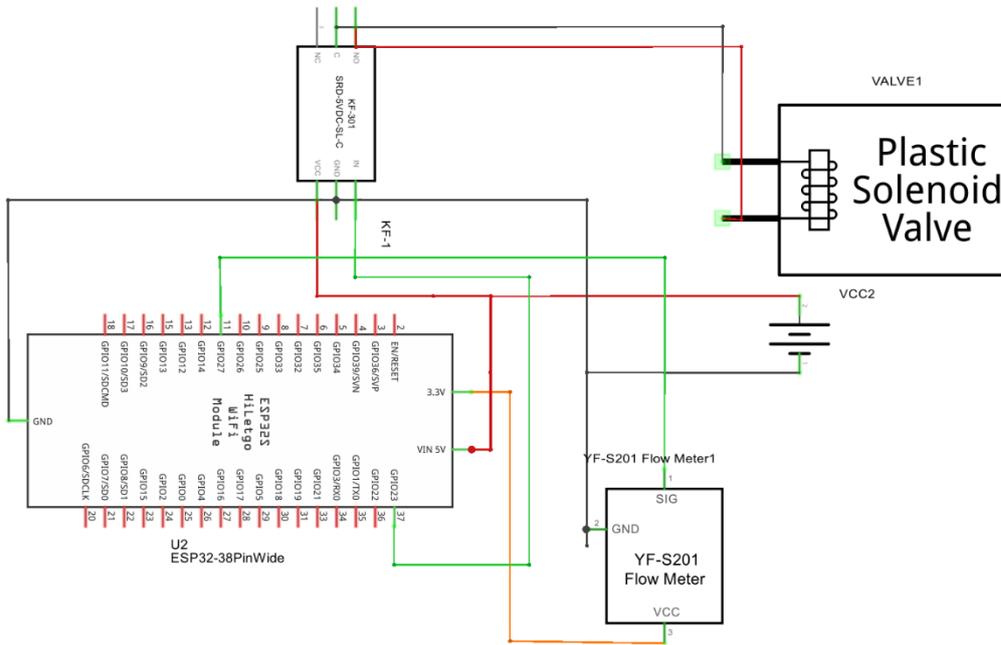


Figura 25. Arquitectura Física

Como se puede apreciar en la arquitectura física (Ver figura 26) se presentan los diferentes sensores y actuadores que se van a utilizar para el armado del prototipo, aquí el sensor FC-28 es el encargado de capturar la humedad de suelo y este se conecta al módulo Esp32 por medio de sus pines analógicos de igual manera el sensor de caudal está conectado al esp32 y es encargado de capturar los datos en litros del agua utilizada para el riego. EL módulo Esp32 es el encargado de recibir los datos de los sensores y enviarlos al servidor bróker que en este caso es el microcomputador raspberry pi, mediante wifi, de igual forma el módulo Esp32 es el encargado de recibir información del bróker para el control de riego que en este caso va a encender el relé, y esto está conectado a voltaje de 5V. En la figura 27 se presenta la vista esquemática de sistema en base a la arquitectura física.



fritzing



fritzing

Figura 26. Diagrama esquemático del sistema de irrigación

6.1.5.3. Estilo Arquitectónico de IoT (Broker)

En el estilo arquitectónico se presenta un modelo de referencia con el cual se puede desarrollar la solución informática.

El presente estilo arquitectónico está dividido en tres capas:

1. La primera capa se denomina de sensores o de percepción que es la encargada de obtener los datos mediante los sensores, para que esto suceda existen dos módulos esp-32 mismos que contienen las funciones necesarias para la comunicación de los sensores con el broker.
2. La segunda capa de red, se encuentra el servidor (Broker) que sirvió como intermediario en las comunicaciones entre el publicador y el suscriptor. Aquí se hace uso de mosquito que es un servidor basado en mqtt, este se instala en un raspberry pi o en un servidor externo de software libre. Mosquitto permite realizar la comunicación con dispositivos o clientes que se encuentren suscritos a este broker mediante topics. Cabe destacar que el protocolo mqtt permite la comunicación mediante los puertos 1883 para la comunicación de los dispositivos físicos (sensores de humedad de suelo, sensor de caudal y electroválvulas) y el 9001 para websocket utilizado por el aplicativo web.
3. La tercera capa de aplicación contiene el servicio web, que es utilizado para almacenar los datos obtenidos por los sensores, de igual forma cuenta con un aplicativo móvil para el control de riego y un dashboard para la representación de los datos almacenados en la base de datos.

En la figura 28 se presenta el estilo arquitectónico del sistema de irrigación con IoT para un mayor entendimiento de la arquitectura.

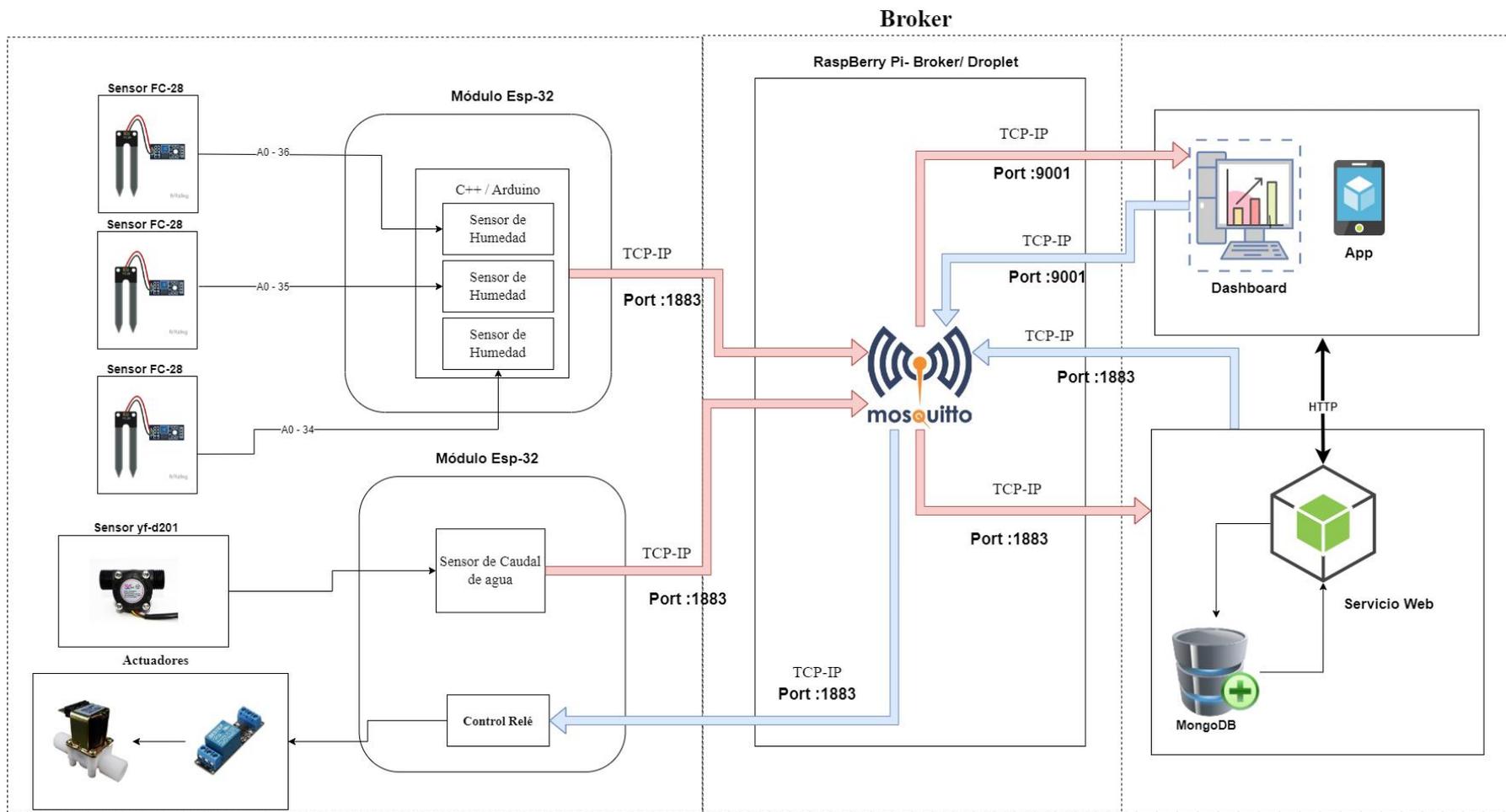


Figura 27. Diseño arquitectónico de IoT Brok

Finalmente, la arquitectura de software, planteada para la solución informática consta de cinco capas, que son: I) capa de sensores, II) capa de red, III) capa de comunicación, IV) capa de base de datos, V) capa de aplicación. En la figura 29, se presenta una representación gráfica de la solución.

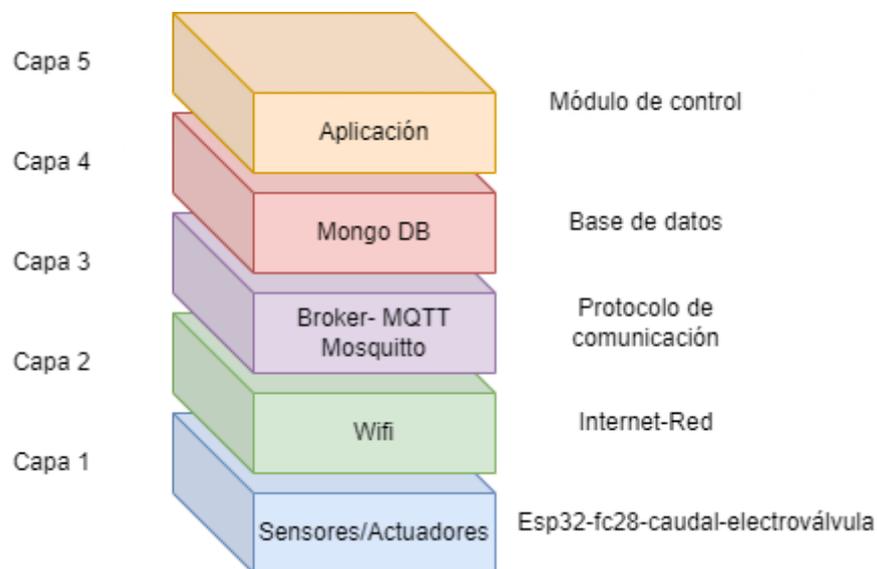


Figura 28. Arquitectura de software para IoT

6.3. OBJETIVO 3: Desarrollar un módulo de control que permitirá administrar el sistema de irrigación con IoT mediante la metodología XP.

6.1.6. Planificar las etapas para el desarrollo del módulo y administración

Como resultados de la fase de planificación para el desarrollo del módulo y administración para el sistema de irrigación con IoT, como primer paso se utilizó la técnica entrevista a los involucrados y expertos en sistemas de irrigación de la carrera de Ingeniería Agrícola de la Universidad Nacional de Loja. Los mismos expusieron cómo se realiza el proceso que se debe llevar para un sistema de riego y los factores que se tienen que tomar en cuenta para el desarrollo del mismo.

En la tabla 10, se detallan las preguntas realizadas en las entrevistas en donde describe el proceso que se lleva para desarrollar un sistema de irrigación:

Tabla 10. Preguntas de la entrevista

Pregunta	Desarrollo
----------	------------

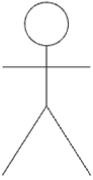
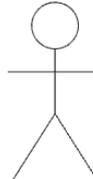
<p><i>¿Cuál es el proceso que se lleva a cabo para desarrollar un sistema de irrigación?</i></p>	<p>El proceso que se lleva para desarrollar un sistema de riego es extenso ya que se tiene que tomar en consideración varios factores ambientales, se parte por realizar un análisis del tipo de suelo en donde se va a instalar el sistema de riego, se determina el tipo de cultivo, con sus diferentes parámetros como por ejemplo evapotranspiración de la planta, ciclo del cultivo, fases de fenología, entre otros, así mismo se toma en consideración factores climáticos obteniendo datos de estaciones meteorológicas que ayudan a determinar cuándo se puede realizar el riego y realizar una programación del mismo.</p>
<p><i>¿Qué parámetros físicos del suelo son importantes tomar en cuenta para un sistema de irrigación?</i></p>	<p>Los parámetros físicos del suelo que se toman en consideración son:</p> <ul style="list-style-type: none"> Tipo Textura Capacidad de campo Punto de marchitez Permanente Perfil de suelo Velocidad de Infiltración
<p><i>¿De acuerdo a qué factores determinan la cantidad de agua que se va a utilizar en los sembríos?</i></p>	<p>Para determinar la cantidad de agua necesaria que se va a utilizar se debe de tomar en consideración:</p> <ul style="list-style-type: none"> Capacidad de campo Punto de marchitez permanente Humedad disponible Textura del suelo
<p><i>¿Cuáles son los métodos de riego que utilizan en los sistemas de irrigación?</i></p>	<p>Los métodos de riego utilizados son:</p> <ul style="list-style-type: none"> Por goteo Aspersión Inundación

6.1.6.1. Especificación de Requisitos:

Luego de obtener información de la entrevista realizada a los ingenieros de Ingeniería Agrícola de la UNL. Se desarrolló el documento de especificación de requisitos de software basados en el estándar IEEE-830 Anexo 3, en donde se describieron de manera formal los requisitos funcionales y no funcionales del sistema de administración para el sistema de irrigación con IoT.

Como primer paso, se determinaron los usuarios que intervienen en el desarrollo de la solución informática, de la misma forma en las tablas 11 se determina los roles que cada usuario va a realizar dentro del desarrollo.

Tabla 11. Roles de usuarios

Stakeholder	Descripción	Escenario	Vistas
 <i>Administrador</i>	Es la persona que encarga gestionar las cuentas del sistema de irrigación	Escenario de negocio	CU Gestionar Cuenta
 <i>Usuario</i>	Es la persona encargada de interactuar con el sistema de irrigación	Escenario de negocio	CU Abrir riego CU Cerrar Riego CU Analizar datos de sensores

Los requisitos funcionales de la solución informática se basan en el proceso que se llevan a cabo para el desarrollo de un sistema de irrigación tradicional, en la tabla 12 se muestran los requisitos funcionales del sistema.

Tabla 12. Requisitos funcionales del sistema

Referencia	Descripción	Categoría	Prioridad
------------	-------------	-----------	-----------

<i>RF 001</i>	El Administrador podrá registrar los datos (Nombres, Apellidos, Dirección, Teléfono, correo, contraseña).	Visible	Alta
<i>RF 002</i>	El administrador podrá visualizar los usuarios registrados	Visible	Media
<i>RF 003</i>	El usuario, administrador podrá actualizar sus datos (Nombres, Apellidos, Dirección, Teléfono, correo, contraseña).	Visible	Alta
<i>RF 004</i>	El usuario y administrador podrá iniciar sesión (correo, contraseña).	Visible	Alta
<i>RF 005</i>	El administrador podrá registrar el tipo de suelo (Tipo suelo, capacidad de campo, punto de marchitez)	Visible	Alta
<i>RF 006</i>	El administrador podrá actualizar el tipo de suelo (Tipo suelo, capacidad de campo, punto de marchitez)	Visible	Alta
<i>RF 007</i>	El administrador podrá visualizar los tipos de suelo registrados	Visible	Alta
<i>RF 008</i>	El usuario y administrador podrán controlar el riego (Encender y Apagar)	visible	Alta
<i>RF 009</i>	El usuario y administrador podrán visualizar de forma gráfica los datos obtenidos por los sensores.	Visible	Media
<i>RF 010</i>	El sistema permitirá registrar topics actuador y sensor para la conexión con el hardware	Visible	Alta
<i>RF 011</i>	El sistema permitirá visualizar los topics registrados	Visible	Media

<i>RF 012</i>	El sistema permitirá actualizar topics actuador y sensor para la conexión con el hardware	Visible	Alta
<i>RF 013</i>	El sistema permitirá registrar los datos obtenidos por los sensores.	Visible	Alta

Los requisitos no funcionales del sistema tienen que ver con las operaciones que va a realizar la solución informática internamente. En la tabla 13 se presentan los requisitos no funcionales empleados en el desarrollo de la solución informática.

Tabla 13. Requerimientos no funcionales

Referencia	Atributo	Descripción
<i>RNF 001</i>	Seguridad	El sistema validará las credenciales de los usuarios registrados para acceder al sistema.
<i>RNF 002</i>	Rendimiento	El sistema realizará los cálculos para determinar la cantidad de agua necesaria.
<i>RNF 003</i>	Rendimiento	El control de riego debe realizarse en tiempo real.
<i>RNF 004</i>	Seguridad	Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador del sistema.
<i>RNF 005</i>	Usabilidad	El sistema debe proporcionar alertas informativas para los usuarios finales.
<i>RNF 006</i>	Portabilidad	El sistema estará diseñado para funcionar en un entorno web de los navegadores de Google Chrome, Safari, Mozilla Firefox, Entre otros
<i>RNF 007</i>	Disponibilidad	El sistema estará disponible continuamente con un nivel de servicio para los usuarios de 7 días por 24 horas.

6.1.7. Diseñar el módulo de control en base de la especificación de requisitos utilizando notación UML (Unified Modeling Language).

Para la fase de diseño de la metodología de software XP, se optó por utilizar la arquitectura 4+1 para obtener una perspectiva amplia sobre el sistema.

El documento completo con todas las actividades y artefactos desarrolladas en el documento arquitectónico 4+1 se detalla en el Anexo 4.

6.1.7.1. Vistas. - escenarios

La vista de escenarios se utilizó para identificar y validar el diseño arquitectónico del sistema. La descripción de la vista de escenario se ilustra utilizando los diagramas de casos de USO. En la figura 30 se presenta el caso de uso general de la solución informática.

- a) Sistema
 - i. Dashboard de control para sistema de irrigación
 - ii. App móvil para control de riego
- b) Actores
 - i. Usuario
 - ii. Administrador
 - iii. Esp32
 - iv. Raspberry pi
- c) Casos de Uso
 - i. Abrir riego
 - ii. Cerrar riego
 - iii. Gestionar Cuenta
 - iv. Gestionar Datos sensores
 - v. Analizar datos sensores

Dentro de la vista de escenarios se presenta el diagrama de casos de uso general, así como los diagramas de casos de uso específicos con su respectiva descripción.

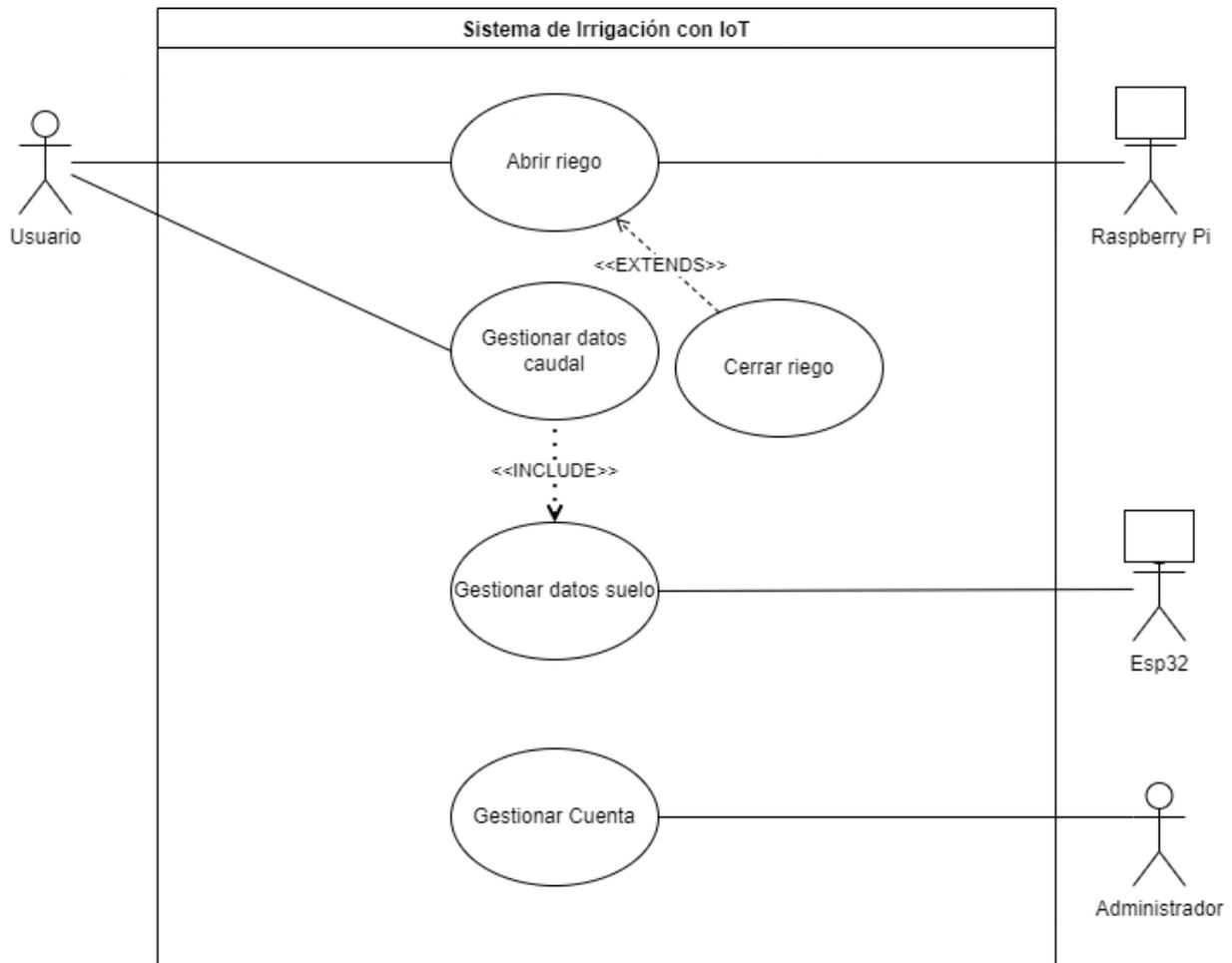


Figura 29. Caso de uso General

6.1.7.2. Vista. – Procesos

En esta vista se describen los procesos del sistema y cómo se relacionan entre sí. La vista de procesos se centra en el comportamiento del sistema durante la ejecución, utilizando diagramas de Actividad o Secuencia para ilustrar estos procesos. En las figuras 31, 32 y 33 se presentan los diagramas de actividad de encender, apagar riego y captura de datos.

Los diagramas de secuencia contienen los siguientes elementos:

- Actor
- Interfaz de usuario
- Controlador
- Base de datos

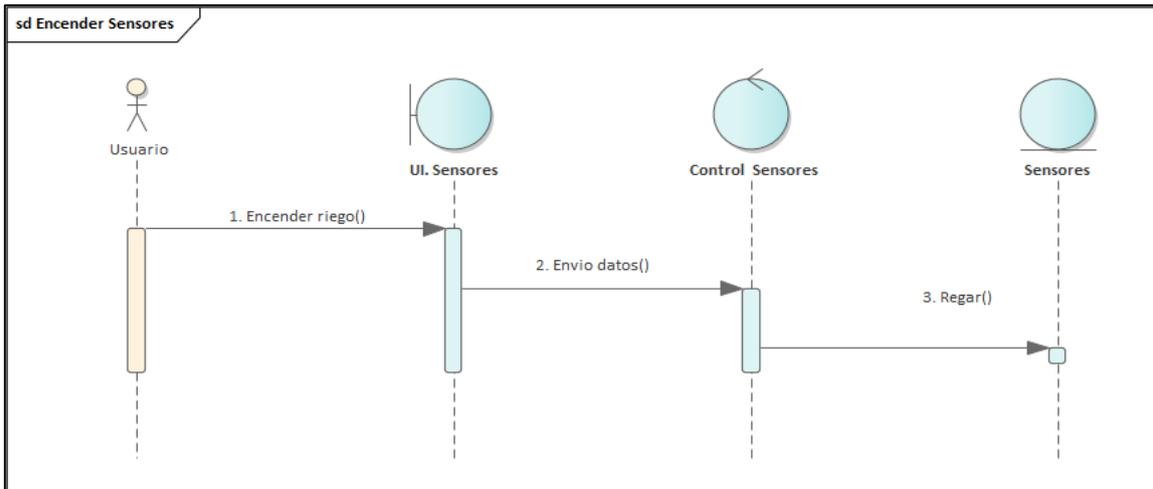


Figura 30. Diagrama de secuencia encender riego

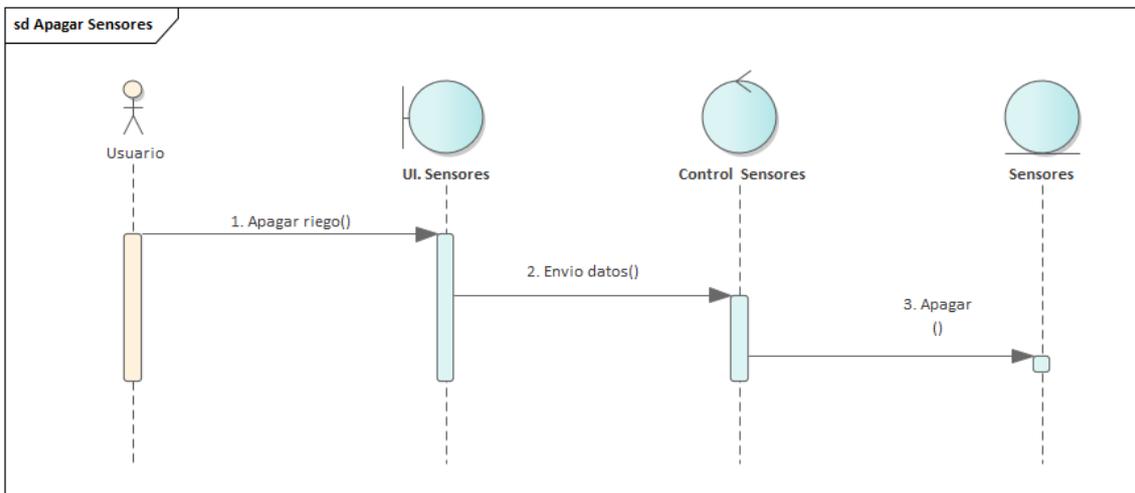


Figura 31. Diagrama de secuencia apagar riego

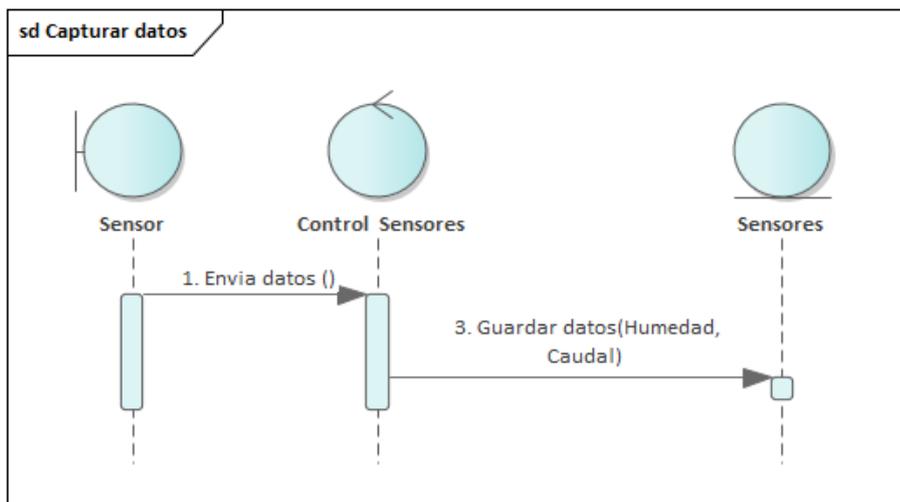


Figura 32. Diagrama de secuencia capturar datos

6.1.7.3. Vista. - Lógica

Presenta la funcionalidad que un sistema debe proporcionar a los usuarios. En la figura 34 se visualiza el diagrama de clases para la solución informática.

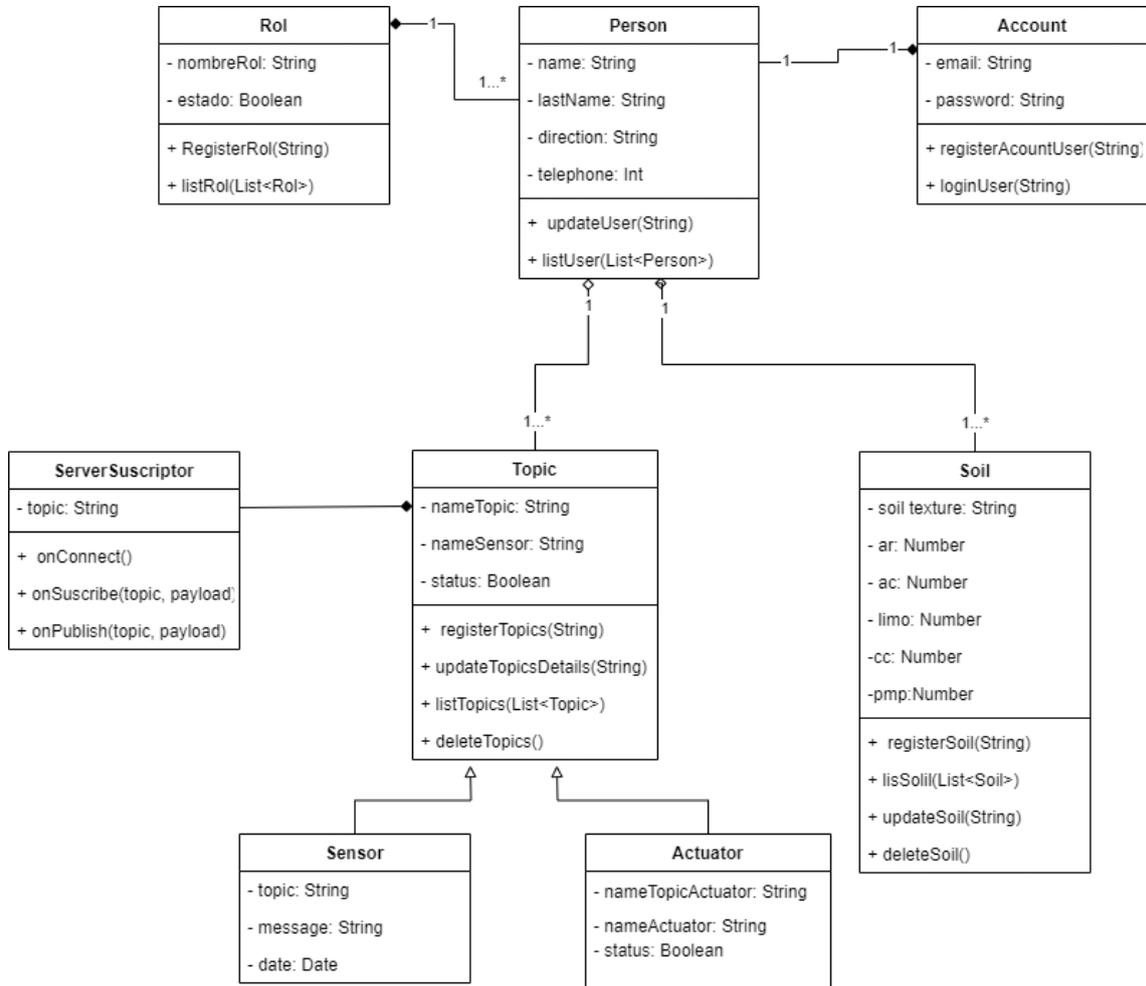


Figura 33. Diagrama de clases

6.1.7.4. Vista. - Desarrollo

Se utilizó diagrama de componentes para representar el sistema desde la perspectiva del programador, lo que permite enfocarse en la gestión de los artefactos de software y describir los diferentes componentes del sistema. En la figura 35 se presenta el diagrama de componentes de la aplicación.

- **Diagrama. - Componentes**

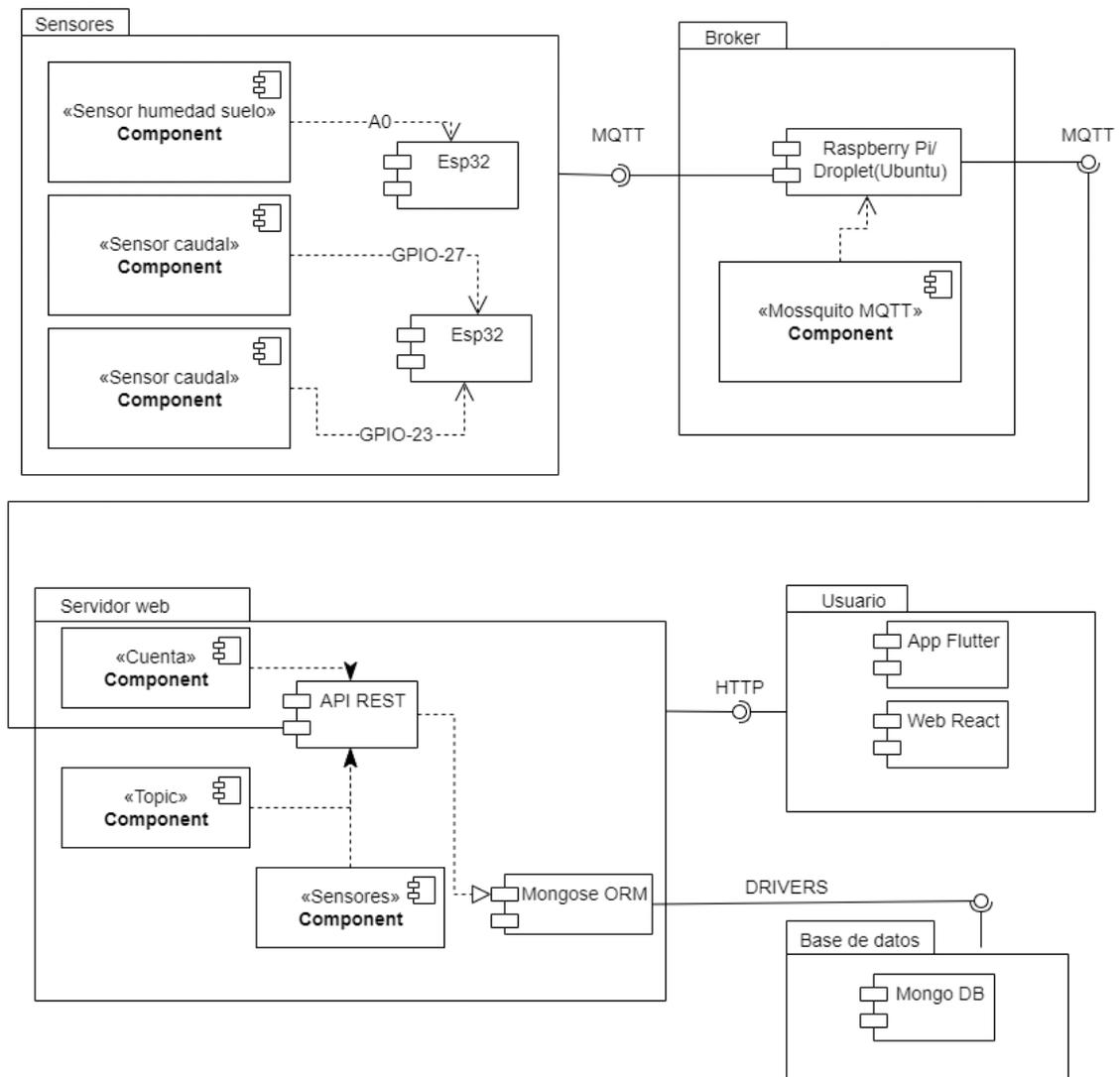


Figura 34. Diagrama de componentes

6.1.7.5. Vista. – Física

La vista física describe los componentes de software en la capa física, así como las conexiones físicas entre los componentes. En la figura 36 se visualiza el diagrama de despliegue de la solución informática.

- Nodo 1: Esp32 Sensores
 - <<FC-28>> Sensor de humedad
 - <<YF-S2001>> Sensor de caudal
- Nodo 2: Esp32 Actuadores
 - <<Relay>> Control de riego
- Nodo 3: Raspberry Pi
 - <<Mossquito>> Servidor MQTT
- Nodo 4: Servidor Web
 - <<NodeJs>> Servidor
- Nodo 5: Servidor de base de datos
 - <<Mongo DB>> Base de Datos
- Nodo 6: Usuario web
 - <<React Js>> Navegador Web
- Nodo 7: Usuario móvil
 - <<Flutter>> App

- **Diagrama. - Despliegue**

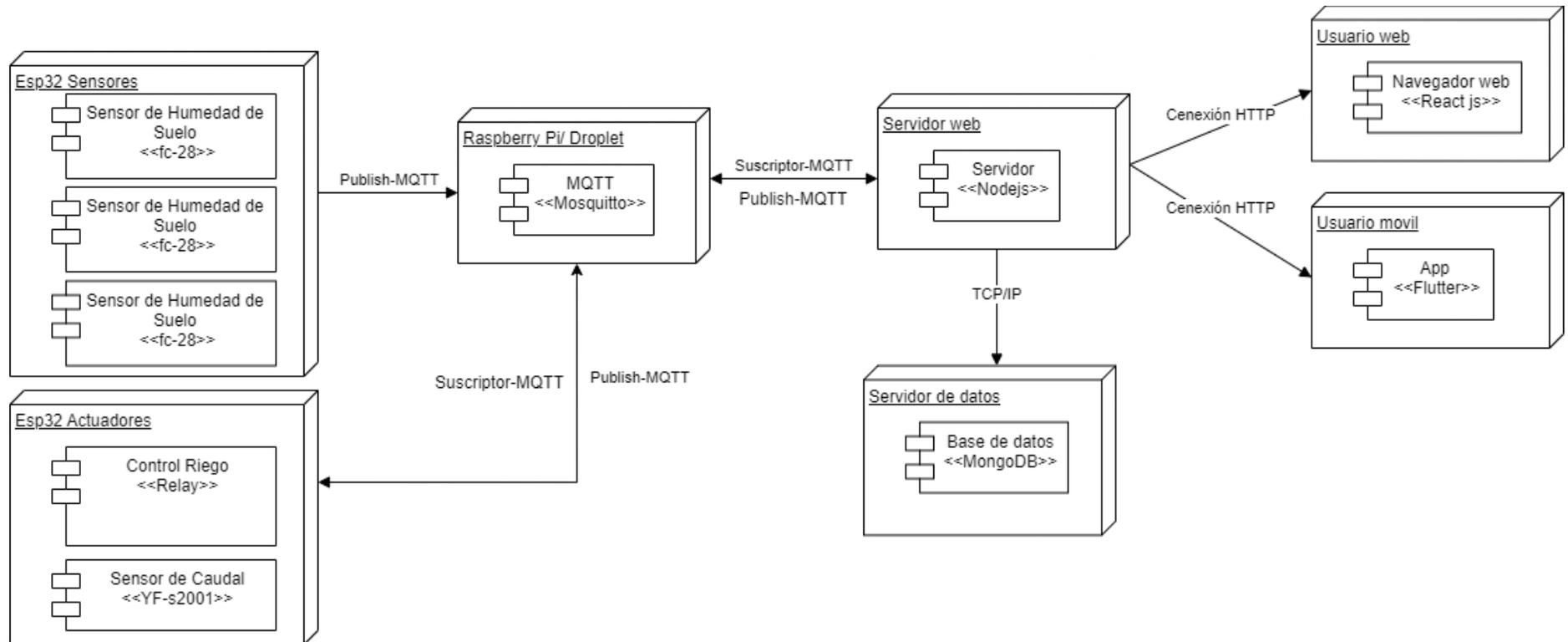


Figura 35. Diagrama de despliegue

6.1.8. Codificar el sistema de control y administración en base al diseño previamente establecido.

Luego de realizar la arquitectura del sistema de irrigación con IoT, se procedió a desarrollar los prototipos tanto de la aplicación web y móvil. Cabe destacar que estos diseños fueron cambiando de acuerdo a las necesidades de los stakeholders. A continuación, se muestran algunos diseños de la interfaz de usuario de la solución informática.

6.1.8.1. Diseño de la pantalla de Inicio de Sesión del aplicativo web

El diseño de la pantalla de inicio de sesión contiene en la parte izquierda el icono con el nombre de la aplicación denominado “Demeter”, en la parte derecha contiene un título del con el nombre de la página en la que se encuentra seguido por un formulario que permite ingresar el correo y contraseña antes registrado, así mismo se encuentra un botón que permite el ingreso de los usuarios al sistema principal.

Pantalla Inicio de Sesión		
Nro. Vista	001	
Estado de Prueba	Aceptado sin cambios	X
	Aceptado con cambios	
	Rediseñar	
Observaciones		



Figura 36. Pantalla login

6.1.8.2. Diseño de la pantalla principal del aplicativo web

El diseño de la pantalla principal contiene el nombre de la aplicación, nombre de usuario que ingresó al sistema. En la parte izquierda se encuentra un menú con las operaciones que puede realizar un administrador, en este caso puede registrar usuarios, tipos de suelo, topics de sensores y actuadores. En la parte central de la pantalla se presentan varias tarjetas con los tipos de suelo con sus características principales, seguidamente contiene indicadores de humedad de suelo enviados por los sensores. Finalmente, en la parte inferior contiene un indicador de caudal, el tipo de suelo con el que se va a trabajar y el botón de control de riego que permite encender y apagar el riego de forma manual.

Pantalla Principal-Home		
Nro. Vista	002	
Estado de Prueba	Aceptado sin cambios	X
	Aceptado con cambios	
	Rediseñar	
Observaciones		

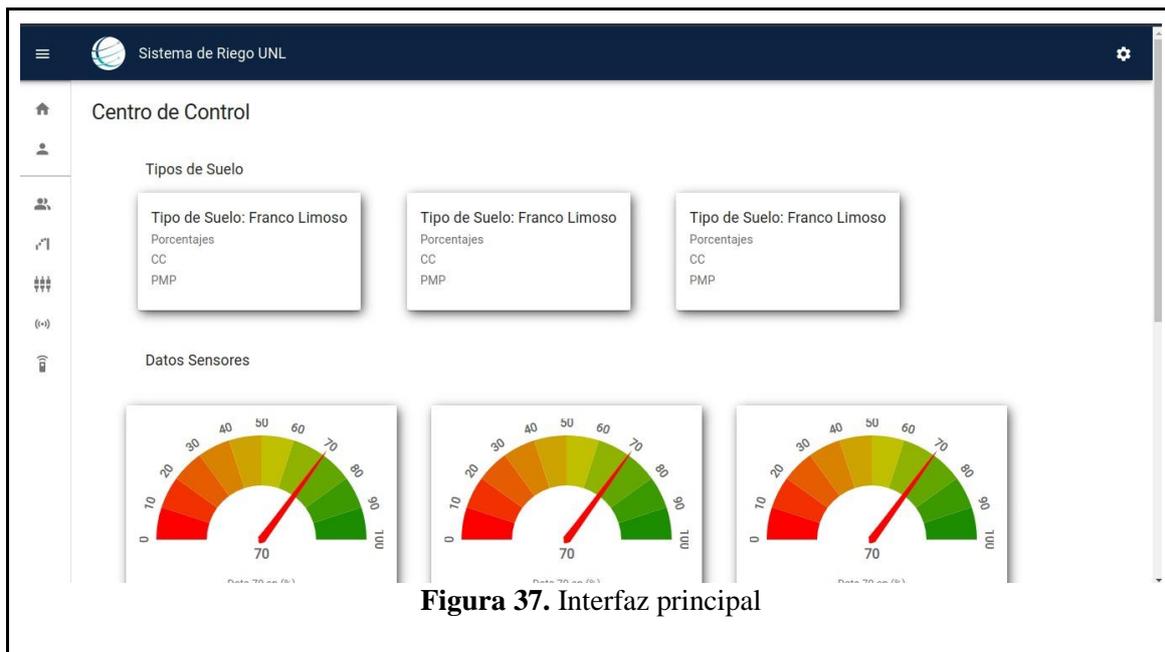
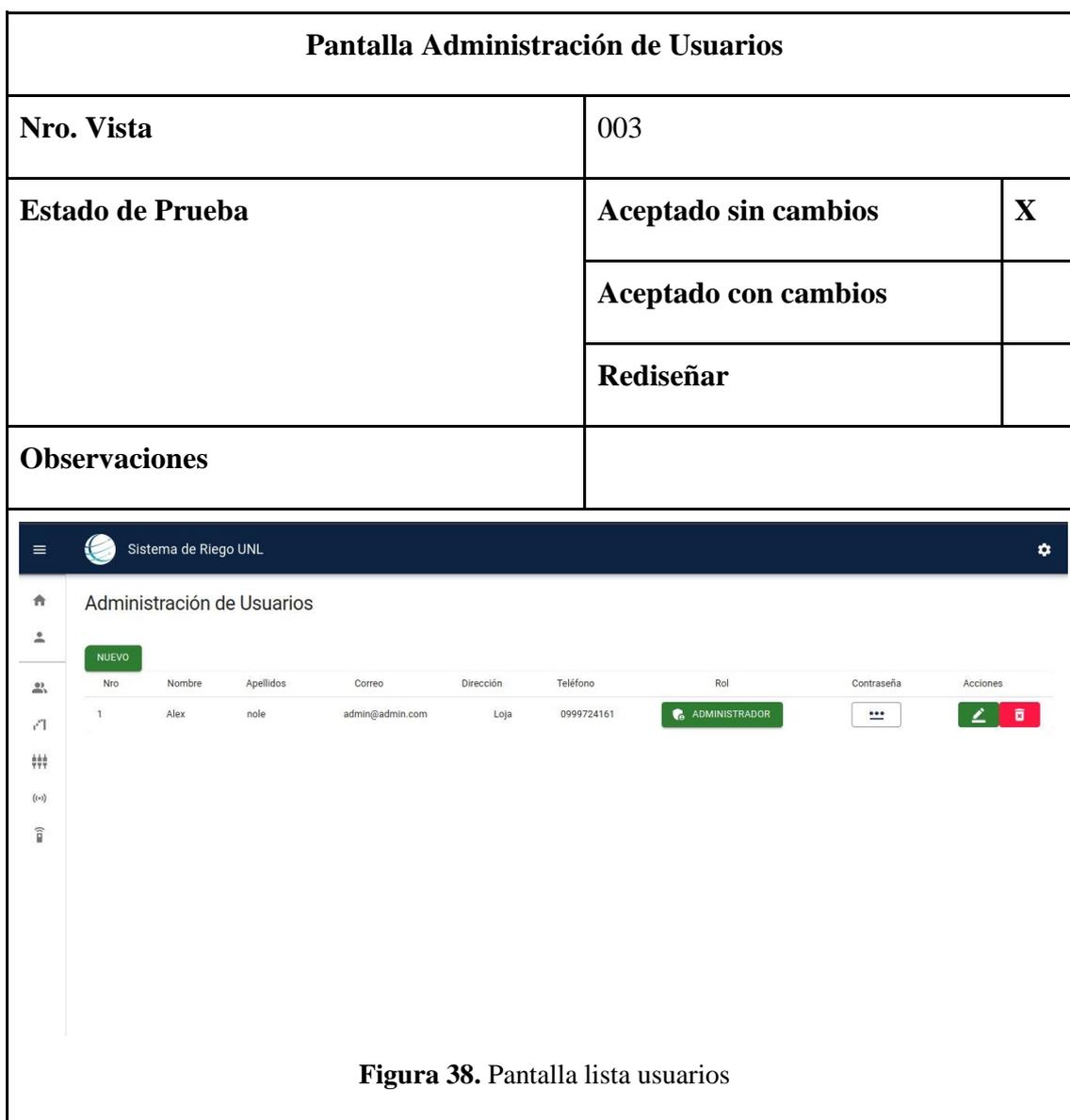


Figura 37. Interfaz principal

6.1.8.3. Diseño de pantalla para gestionar usuarios por el administrador del aplicativo web

El diseño de la pantalla para administrar usuario contiene en nombre descriptivo de la sección en donde se encuentra, seguidamente de un botón que habilita un modal con el formulario que permita ingresar los datos de los nuevos usuarios que utilizarán el sistema. A continuación, se presenta una tabla donde se listan los usuarios con la información relevante de cada uno, así mismo dentro de la misma tabla se encuentran botones iterativos que permiten llamar a formularios para asignar un rol diferente a los usuarios, cambios de contraseña o actualizar los datos de los usuarios.



6.1.8.4. Diseño de la pantalla de tipos de suelo del aplicativo web

El diseño de la pantalla de tipos de suelo contiene un formulario principal que permite ingresar las características de un tipo de suelo, en la parte inferior contiene un botón que permite realizar el registro del formulario. A continuación, se presenta una tabla que contiene la información de los tipos de suelo antes registrados y dentro de la misma tabla en cada registro presenta dos botones iterativos que permite realizar las operaciones de eliminar y actualizar los tipos de suelo según sea requerido.

Pantalla Características de Suelo		
Nro. Vista	004	
Estado de Prueba	Aceptado sin cambios	X
	Aceptado con cambios	
	Rediseñar	
Observaciones		

Figura 39. Pantalla tipo de suelo

6.1.8.5. Diseño de la pantalla de topic de sensores del aplicativo web

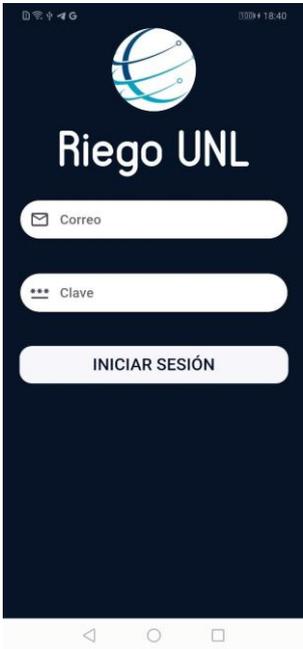
El diseño de la pantalla de topic de sensores es similar a la pantalla de tipos de suelo los datos del formulario son diferentes, este cuenta con un formulario principal que permite ingresar el nombre de topic y nombre de sensor, en la parte inferior contiene un botón que permite realizar el registro del formulario. A continuación, se presenta una tabla que contiene la información de los topics de sensores antes registrados y dentro de la misma tabla en cada registro presenta dos botones iterativos que permite realizar las operaciones de eliminar y actualizar los topics según sea requerido.

Pantalla Topics de Sensores-Actuadores		
Nro. Vista	005-006	
Estado de Prueba	Aceptado sin cambios	X
	Aceptado con cambios	
	Rediseñar	
Observaciones		

Figura 40. Pantalla topic sensores

6.1.8.6. Diseño de pantalla inicio de sesión aplicativo móvil

La pantalla contiene en la parte superior el icono con el nombre de la aplicación denominado “Demeter”, en la parte central se encuentra dos campos de texto que permite ingresar el correo y contraseña antes registrado, así mismo se encuentra un botón que permite el ingreso de los usuarios a la vista principal.

Pantalla Inicio sesion-app móvil		
Nro. Vista	001	
Estado de Prueba	Aceptado sin cambios	X
	Aceptado con cambios	
	Rediseñar	
Observaciones		
		
Figura 41. Pantalla inicio sesión		

6.1.8.7. Diseño de pantalla de control de riego (Encender/Apagar)

La pantalla de control en la parte superior se encuentra el icono de la aplicación, seguido de su nombre denominado “Demeter”, en la parte central se encuentran dos botones los cuales permiten encender y apagar el riego.

Pantalla Control de Riego-app móvil		
Nro. Vista	002	
Estado de Prueba	Aceptado sin cambios	X
	Aceptado con cambios	
	Rediseñar	
Observaciones		
		
Figura 42. Pantalla control riego		

6.1.8.8. Codificación del servicio web API-REST

El servidor web de la solución informática realiza las operaciones de registrar login, gestión de usuarios, registro de datos enviados por los sensores, gestión de tipos de suelo, gestión de topics. La estructura del servicio web tiene la estructura presentada en la figura 44.

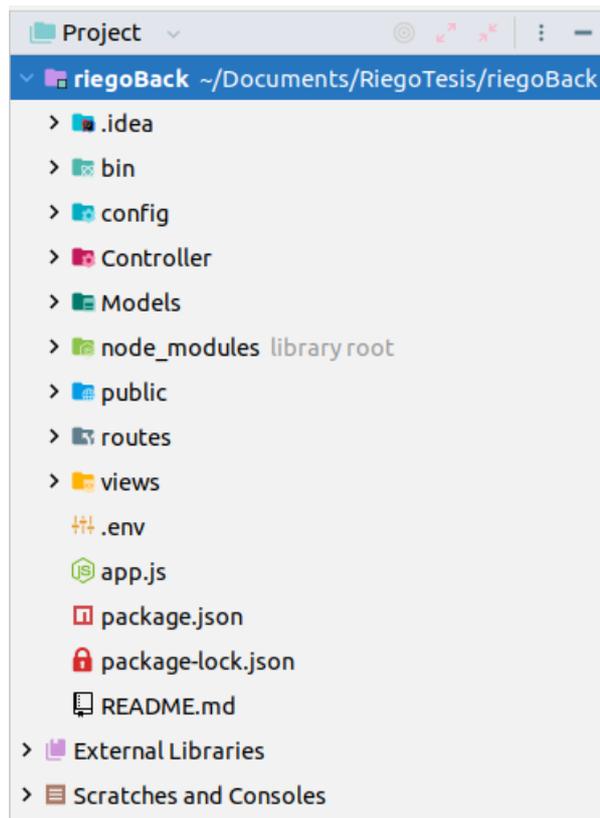


Figura 43. Estructura del servicio web

Entre los directorios más importantes del API-REST tenemos:

- **Config:** La carpeta config contiene la configuración del middleware de autenticación del sistema en donde se realizan las funciones de registro e inicio de sesión de los usuarios.
- **Controller:** La carpeta controller contiene todos los controladores que permiten realizar las operaciones como administrar los usuarios, registro de los datos enviados por los sensores, conexiones con el protocolo mqtt, entre otros.
- **Models:** La carpeta models contiene la estructura de la base de datos y va a ser el responsable de la comunicación entre la aplicación y la base de datos.
- **Routes:** La carpeta routes contiene las rutas para realizar las peticiones con el API-REST.

En la Figura 45 se muestra un fragmento de código del controlador “ServerSuscriptorController” que permite realizar la conexión con el hardware mediante el protocolo MQTT, para la captura de la información enviada por los sensores.

```

class ServerSuscriptorController {
  constructor() {
    this.mqttClient = null;
    this.host = '159.223.125.186';
    this.port = '9001';
    this.username = 'riego';
    this.password = '3435';
  }

  connect() {
    this.mqttClient = mqtt.connect( brokerUrl: `ws://${this.host}:${this.port}`, opts: {
      //this.mqttClient = mqtt.connect(`mqtt://159.223.182.77:1883`, {
        username: this.username,
        password: this.password
      });
    this.mqttClient.on('error', (error) => {
      console.log('Ocurrio un error al conectarse ', error)
      this.mqttClient.end();
    })
    this.mqttClient.on('connect', async () => {
      console.log('Conectado a MQTT Broker');
    })
  }
}

```

Figura 44. Código ServerSuscriptorController

Algunos de los servicios que contiene el API-REST se especifican en la tabla 14 en donde se muestra la URL y el controlador al cual pertenece.

Tabla 14. Controlador, router, url

Controlador	Router	URL
<i>AccountController</i>	router.get('/user/:email', accountController.findEmailUse rs);	router.get('/user/:email', accountController.findEmailUse rs);
<i>RolController</i>	http://157.230.82.173//users//use r/:email	http://157.230.82.173//users//use r/:email
<i>SensorDataControll er</i>	router.get('/rols', accountController.findRolsusers) ;	router.get('/rols', accountController.findRolsusers) ;
<i>MqttIrrigationContr oller</i>	http://157.230.82.173/users/rols	http://157.230.82.173/users/rols
<i>SoilController</i>	router.get('/data', SensorController.dataForSensor Detail);	router.get('/data', SensorController.dataForSensor Detail);
<i>TopicActuadorContr oller</i>	http://157.230.82.173/sensors/da ta	http://157.230.82.173/sensors/da ta

```

TopicSensorControll router.post('/riego',
er                    publishMqtt.MqttControlRiego);
                    publishMqtt.MqttControlRiego);

```

Seguidamente se presenta la ejecución de una petición HTTP haciendo uso de la aplicación postman el mismo que nos facilita realizar las peticiones con los servicios web. En la figura 46 se muestra cómo se realizó la búsqueda de los usuarios registrados en el sistema.

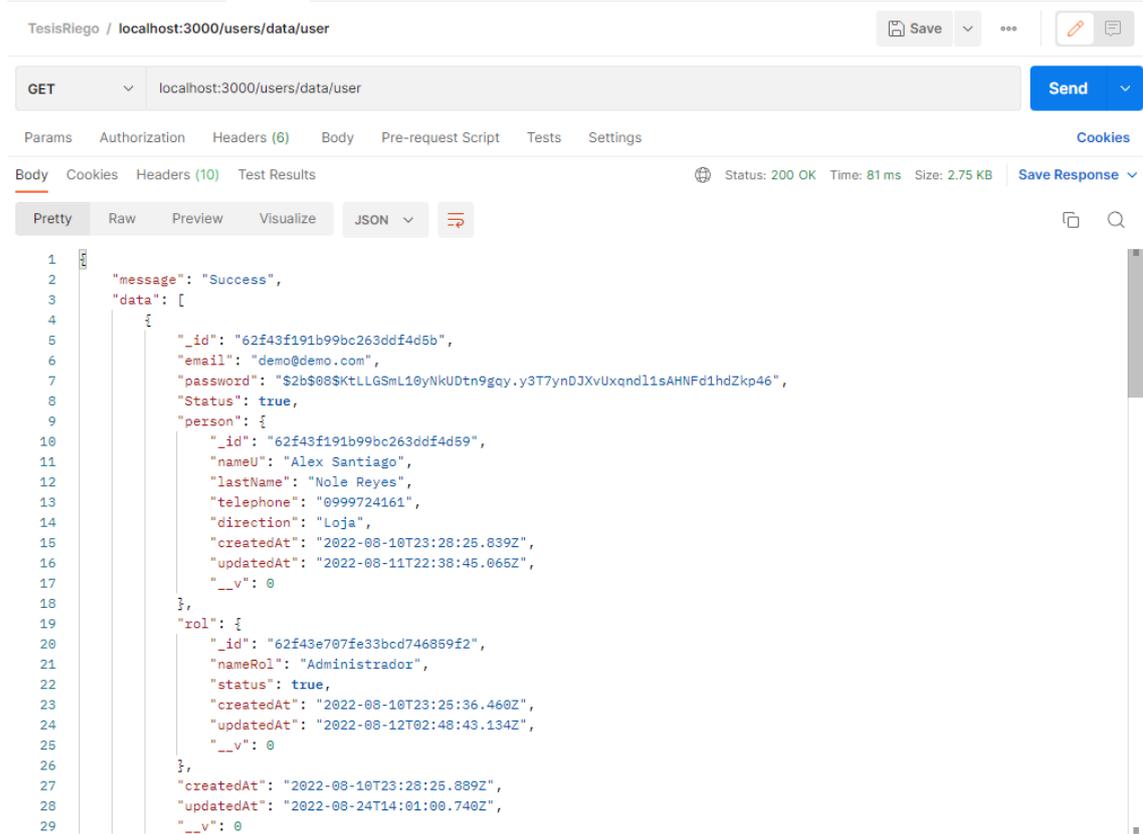


Figura 45. Buscar usuario

6.1.8.9. Codificación del aplicativo web

Para el desarrollo de la aplicación web se la realizó en la librería de JavaScript Reactjs, en donde se desarrollaron las funcionalidades del usuario como del administrador, que permite realizar las funcionalidades de control de riego, gestión de usuarios, gestión de tipos de suelo, gestión de topics tanto de sensores como de actuadores. En la figura 47 se puede observar la estructura del proyecto para Front-End.

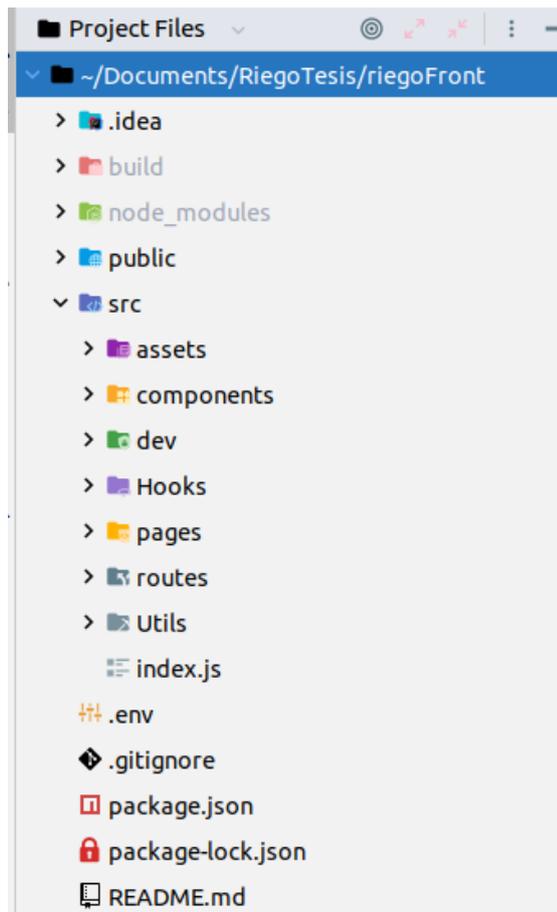


Figura 46. Estructura del aplicativo web

Entre los directorios más relevantes tenemos:

- **Src:** La carpeta src contiene los directorios necesarios para el desarrollo de la parte gráfica de la aplicación móvil, aquí están los componentes, paginas, rutas entre otros.
- **Components:** Contiene los componentes con extensión js o vistas con las cuales va a interactuar el usuario.
- **Assets:** Contiene los recursos adicionales para el maquetado, como iconos, imágenes, css necesarios para modelar los diseños establecidos.
- **Hooks:** Contiene funcionalidades que pueden ser invocadas en varios componentes.
- **Pages:** contiene las páginas principales o padres las cuales van hacer uso de varios componentes para completar una funcionalidad.
- **Routers:** Contiene las rutas o URL que va a ser presentada a los usuarios en el navegador.

6.1.8.10. Codificación del aplicativo móvil

Para el desarrollo de la aplicación móvil se la realizó con el framework Flutter en base al lenguaje de dart, en donde se desarrollaron las funcionalidades de inicio de sesión, y las funcionalidades de control de riego (encender/ apagar). En la figura 48 se puede observar la estructura del proyecto móvil.

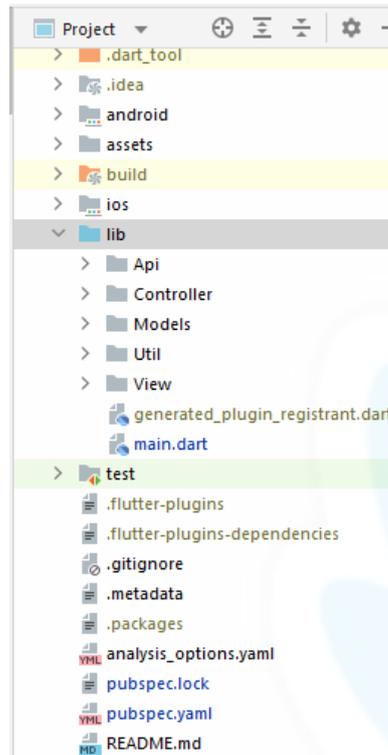


Figura 47. Estructura aplicativo móvil

Entre los directorios más relevantes tenemos:

- **assets:** Contiene las imágenes y los estilos de textos necesarios para el desarrollo de la parte gráfica de la aplicación móvil.
- **lib:** La carpeta lib, es la cual se crean los directorios necesarios para el desarrollo de la parte grafica de la aplicación móvil, aquí están los controladores, modelos, view, api, utilidades.
- **Api:** Contiene los archivos que permiten consumir los servicios rest.
- **Controller:** Contiene los providers que son los manejadores de estados, para la actualización de las variables que necesitan ser cambiadas en tiempo de ejecución.

- **Models:** Contiene los archivos de la estructura de los datos que retornan los recursos seleccionados.
- **Util:** Contiene los archivos globales que son necesarios para la aplicación, como son los colores principales de la app.
- **View:** Contiene las vistas necesarias con las cuales el usuario interactuara.

6.1.9. Fase de pruebas

6.1.9.1. Pruebas de Aceptación de Usuario

Las pruebas de aceptación de usuario fueron realizadas en primera instancia por los ingenieros Diego Chamba y el ingeniero Romel Ramon docentes y expertos de riego de la carrera de Ingeniería Agrícola de la Universidad Nacional de Loja, quienes verificaron el funcionamiento, la aceptación y satisfacción del prototipo (Ver anexo 8). Dicho proceso de verificación se lo realizó mediante un formulario realizado a los dos ingenieros.

Como resultado se obtuvo que la solución informática cumple con todos los parámetros y los requerimientos expuestos por los mismos. Por lo cual los docentes dan la aceptación del aplicativos web y móvil para el control del riego en base a la humedad del suelo.

6.1.9.2. Pruebas Unitarias del servicio web

Las pruebas unitarias del servicio web, se realizaron paralelamente con el desarrollo de cada uno de los módulos de la solución informática. Para ello se utilizó la herramienta postman, que permite realizar peticiones HTTP de tipo GET, POST, PUT, DELETE, al servicio web. Estas peticiones dan como resultado objetos en tipo JSON, que son útiles para consumir por la aplicación web. En la tabla 15 se presentan los resultados de las peticiones realizadas por postman al API-REST, aquí se puede observar el método, URL, resultado y el tiempo de ejecución (para más detalle ver anexo 6).

Tabla 15. Pruebas Unitarias Postman para Topic

Método	URL	Resultado	Tiempo ejecución
<i>GET</i>	134.122.19.64/topic/list/633c602266ee86df8d862f40	Busca el topic mediante el id	435 ms
	134.122.19.64/topic/list/{id}		

<i>POST</i>	134.122.19.64/topic/update/633c602266ee86df8d862f40	Actualiza el topic mediante id	488 ms
	134.122.19.64/topic/update/{id}		
<i>GET</i>	134.122.19.64/topicActuador/list/633c611366ee86df8d862f42	Busca el Topic de actuador mediante id	509 ms
	134.122.19.64/topicActuador/list/{id}		
<i>POST</i>	134.122.19.64/topicActuador/updateActuador/633c611366ee86df8d862f42	Actualiza el Topic de actuadores mediante el id	432 ms
	134.122.19.64/topicActuador/updateActuador/{id}		

En la figura 49, se presenta el resultado arrojado por una petición POST en la herramienta postman, en donde presentan el resultado en formato JSON para el login de la aplicación.

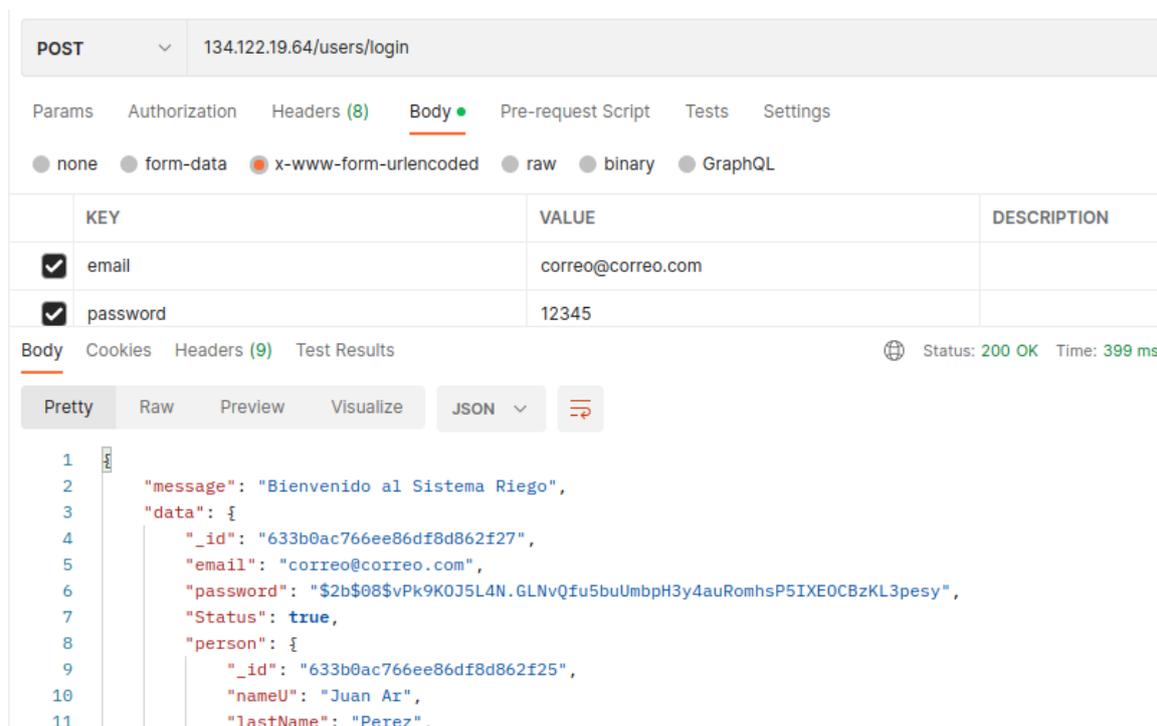


Figura 48. Petición Post Login

6.1.9.3. Pruebas de carga y estrés

Las pruebas de carga y estrés se las realizo a la solución informática ya desplegada en un servidor privado. En donde se realizaron 100 peticiones HTTP tipo GET para los endpoint del API-REST, para conocer hasta cuentas peticiones simultaneas puede soportar el uso del software desarrollado. En la tabla 16 se presenta los tiempos de respuesta y algunas url's, utilizadas para las pruebas. Esto se desarrolló con la ayuda del software Apache Jmeter, mismo que permitió ejecutar estas pruebas y nos arrojó el tiempo de respuesta de cada petición. En el Anexo 6 se detalla los resultados completos de las pruebas de carga y estrés realizadas a la solución informática.

Tabla 16. Prueba de carga y estrés

Método	URL	Resultado	Tiempo ejecución
<i>GET</i>	http://134.122.19.64/users/data/user	Listar usuarios	589 ms
<i>POST</i>	http://134.122.19.64 /soils/soil	Listar soils(Suelo)	237 ms
<i>GET</i>	http://134.122.19.64 /topic/topics	Listar sensores	268 ms
<i>POST</i>	http://134.122.19.64 /topicActuador/topicsActuador	Listar Actuadores	243 ms

En la figura 50 se visualiza las peticiones realizadas al recurso listar usuario, y se evidencia los resultados de las peticiones el tiempo de carga y latencia del mismo.

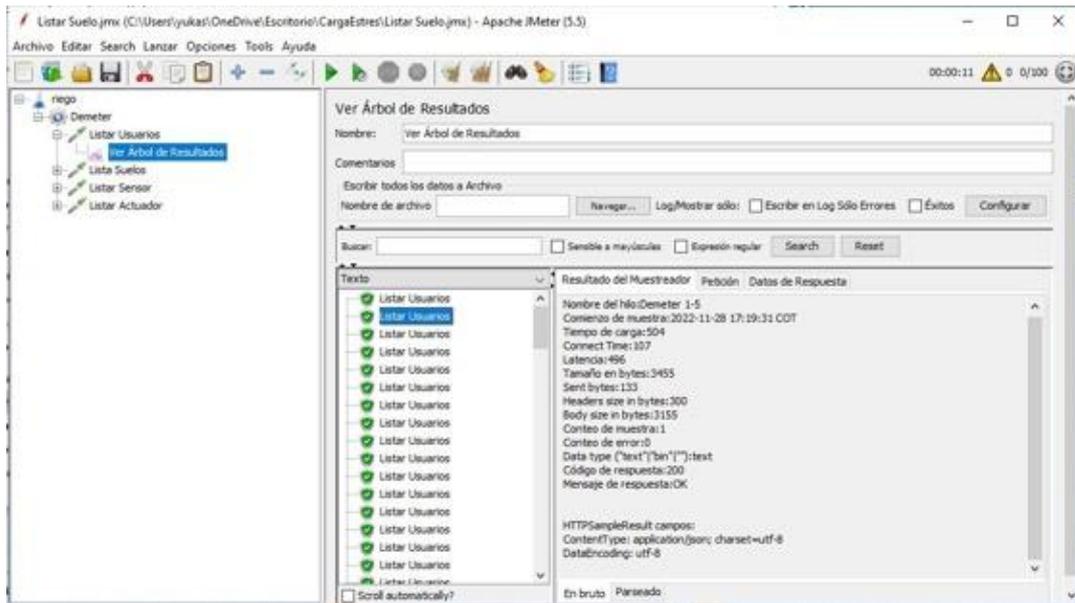


Figura 49. Prueba carga y estrés de listar usuario

6.1.10. Realizar el prototipo de sistema de irrigación, implementando hardware (sensores, actuadores, microcontroladores) y software (sistema de control y administración) haciendo uso de la arquitectura antes establecida

Para la fase de prototipado del sistema de irrigación, se detalla el documento manual de prototipado en el cual se establece las herramientas usadas, como se llevó a cabo la configuración e instalación del bróker, la habilitación de acceso remoto, la programación del suscriptor y publicador, la conexión de los sensores al esp 32, como también la conexión del API-REST con el bróker, aplicativos web y móvil (Véase Anexo 5).

6.1.10.1. Configuración de servidor broker

Para la configuración del broker en una microcomputadora o sistema operativo con mosquitto se siguieron los siguientes pasos:

- **Instalación de Mosquitto en Raspberry pi o Ubuntu:**

Los comandos utilizados para la instalación de mosquitto broker en la raspberry pi, se los debe ejecutar en el orden que se muestra en el presente documento:

Primeramente, se debe ingresar los comandos de actualización para la raspberry pi, cabe destacar que el sistema operativo utilizado se denomina RaspBian basado en el sistema operativo Debian, para lo cual permite actualizar con los mismos comandos:

sudo apt update && sudo apt dist-upgrade

Para la instalación de mosquitto broker se utilizó el siguiente comando:

sudo apt install-y mosquitto mosquitto-clients

Seguidamente para que mosquitto se inicie automáticamente cuando se inicie el raspberrypi se hace uso del comando:

sudo systemctl enable mosquitto.service

Seguidamente para verificar que la instalación es correcta y que se encuentra habilitado el servicio utilizamos el comando:

mosquitto -v

En la figura 51 se presenta la salida en consola del servicio de mosquitto en ejecución

```
root@Broker-new:~# mosquitto -v
1672884307: mosquitto version 2.0.11 starting
1672884307: Using default config.
1672884307: Starting in local only mode. Connections will only be possible from
clients running on this machine.
1672884307: Create a configuration file which defines a listener to allow remote
access.
1672884307: For more details see https://mosquitto.org/documentation/authentication-methods/
1672884307: Opening ipv4 listen socket on port 1883.
1672884307: Error: Address already in use
1672884307: Opening ipv6 listen socket on port 1883.
```

Figura 50. Servicio mosquitto

- **Configuración de Puertos para la comunicación de sensores y actuadores con MQTT:**

Luego de tener instalado el servidor mosquitto en un servidor, se procede a configurar los puertos para la comunicación de los sensores y actuadores utilizando mqtt, utilizando el siguiente comando:

sudo nano /etc/mosquitto/conf. d/default.conf

En la figura 52 se muestra la configuración final con los puertos habilitados para la comunicación.

```
GNU nano 6.4 /etc/mosquitto/conf.d/default.conf
#per_listener_settings true
allow_anonymous false
listener 1883
#protocol mqtt
listener 9001
protocol websockets
socket_domain ipv4
password_file /etc/mosquitto/passwd
```

Figura 51. Configuración puertos broker

- **Programación de sensores y actuadores en Esp-32**

En segunda instancia, se programó la comunicación de los sensores y actuadores en las tarjetas programables Esp-32. Dentro de lo que se programó dentro de los Esp-32, están los topics que son los nombres de los sensores, que son importantes para la comunicación por el protocolo mqtt ya que este reconoce a los dispositivos suscritos con dichos topics. En la figura 53 se muestran los topics utilizados para los sensores.

```
const char* humedytySolid_topic = "suelo1";
const char* humedytySolid_topic1 = "suelo2";
const char* humedytySolid_topic2 = "suelo3";
```

Figura 52. Topics configurados

Las librerías utilizadas para la conexión del broker (PubSubClient), son utilizadas para enviar y recibir mensajes enviados por los suscriptores mediante el protocolo mqtt. En la figura 54 se muestran las librerías utilizadas para la configuración de sensores.

```
#include <Arduino.h>
#include "PubSubClient.h" // Conectar y publicar con el broker
#include "WiFi.h" // Enables the ESP32 to connect to the local network (via WiFi)
#define TIME_DELAY 2000 // Cada cuánto leer del sensor
#define SENSOR 27 //sensor de caudal GPIO 27
```

Figura 53. Librerías de configuración

Seguidamente se declararon los puertos GPIO a los actuales para los sensores, cabe destacar que para los sensores FC-28 se los conecto a los puertos analógicos para tener un mejor control sobre la humedad del suelo. EL esp32 tiene 16 puertos analógicos con voltajes de 0-3.3v que traducido a los sensores va de 0-4095 cuando está saturado, en la figura 55 Se presenta la declaración de los puertos analógicos a los que van conectados los sensores de humedad del suelo.

```

/**
  * Variables Globales suelo
  */
const int SensorPin = 36; //Pin GPIO-36 es pin Analogico de 0-3.3V o 5V -> 0 - 4095 SP
const int SensorPin1 = 34; //Pin GPIO-36 es pin Analogico de 0-3.3V o 5V -> 0 - 4095 SP
const int SensorPin2 = 35; //Pin GPIO-36 es pin Analogico de 0-3.3V o 5V -> 0 - 4095 SP

```

Figura 54. Puertos analógicos

A continuación, se declararon las variables globales y credenciales para la conexión a la red wifi y broker para realizar la comunicación con publicador y servidor broker. Como se encuentra en la figura 56 se declaran el ssid y contraseña de la red a la que se va a conectar, la dirección del servidor broker, así como los topic. Es importante decir que los topic es el nombre del sensor con el cual se va a reconocer en la red y con estos se conectan los suscriptores para verificar los datos.

```

/**
  * Variables para configuracion de wifi y broker
  */
// WiFi
char* ssid = "Nettplus Electro"; // Nombre de la red del sembrio
const char* wifi_password = "1105639015Kevin"; // Contraseña de la red

// MQTT configuracion con el servidor Raspberry Pi
const char* mqtt_server = "192.168.0.104"; // Direccion Ip del Broker o Raspberry Pi
//const char* temperature_topic = "home/sembriol/temperature";
const char* humedytySolid_topic = "suelo1";
const char* humedytySolid_topic1 = "suelo2";
const char* humedytySolid_topic2 = "suelo3";
const char* mqtt_username = "riego"; // Nombre del Usuario del MQTT configurado en el broker
const char* mqtt_password = "3435"; // Nombre del Contraseña del MQTT configurado en el broker
const char* clientID = "riego";
const int mqtt_port = 1883;

```

Figura 55. Credenciales Wifi y Mqtt para la conexión

Seguidamente en la función void setup() se realizó la conexión con el broker y con la red wifi, esto para que cuando inicie el esp32 se conecte automáticamente. En la figura 57 se presenta el método para conexión a la red y al broker.

```

}void setup() {
  Serial.begin(115200);

  // Conexión WiFi
  WiFi.begin(ssid, wifi_password);
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.println('....');
    delay(500);
  }

  client.setServer(mqtt_server, mqtt_port);
  while (!client.connected())
  {
    if (!client.connect("ClienteESP32", mqtt_username, mqtt_password))
    {
      Serial.print("Conectado ");
      Serial.println(mqtt_username);
      delay(2000);
    }
  }
}
}

```

Figura 56. Código dentro de función setup, para conexión Wifi y broker

Finalmente se configuró los sensores de humedad del suelo en el loop para que reciban datos continuamente, aquí se definió la lectura de los sensores y se agrega una función map para presentar valores en porcentaje entendibles. En la figura 58 se presenta el fragmento de código utilizado para enviar datos por medio de mqtt.

El código completo del proyecto se encuentra en el repositorio de GitHub: https://github.com/Noreal-16/Tesis_Sistema_irrigacion_IoT.git

```

int moisturePercentage = analogRead(sensorPins) ;
moisturePercentage = map(moisturePercentage, 0, 4095, 100, 0);
if (isnan(moisturePercentage))
{
  env_msg_delay = millis();
  return;
}
Serial.print("Humedad del suelo-> 0 ");
Serial.print(moisturePercentage);
Serial.println("%");
String shs = String((float)moisturePercentage);
if (client.publish(humedytySolid_topic, String(shs).c_str())) {
  Serial.println("Humidity solid sent!");
} else {
  client.connect(clientID, mqtt_username, mqtt_password);
  delay(10); // This delay ensures that client.publish doesn't clash with the client.connect call
  client.publish(humedytySolid_topic, String(shs).c_str());
}

```

Figura 57. Código para envío de datos al broker

- **Conexión de sensores y actuadores a módulos Esp-32**

Para realizar la conexión correctamente, procedimos analizar cómo se compone el diagrama de puertos del esp32. En la figura 59 se muestran los puertos y pines de forma entendible para su posterior conexión.

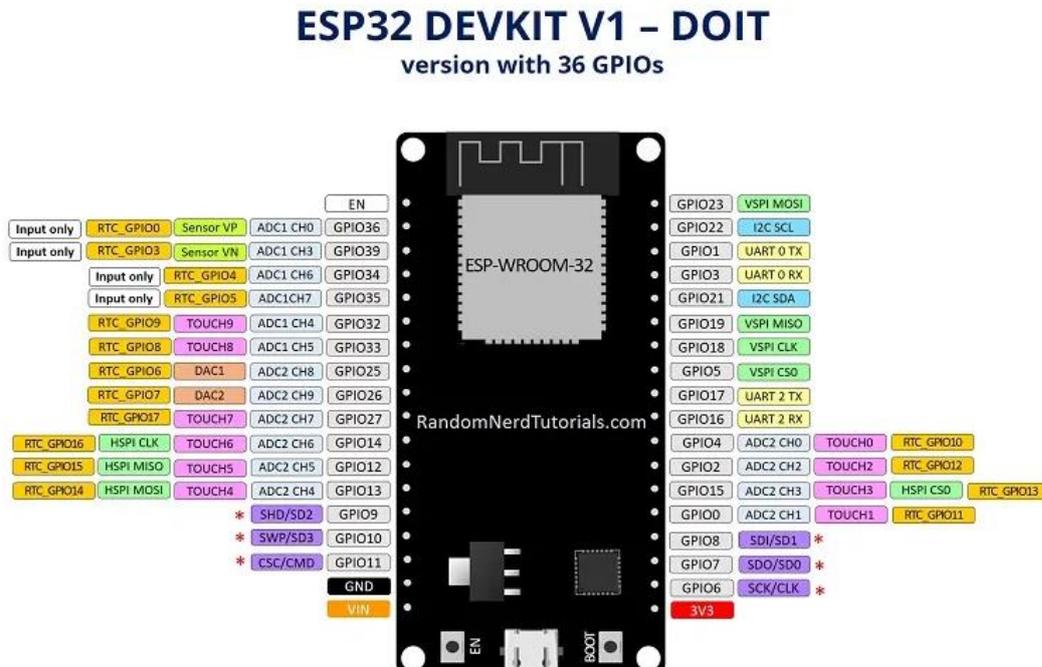


Figura 58. Diagrama de puertos del MCU Esp32

Seguidamente se procedió a la conexión de los sensores fc-28 a los puertos analógicos, antes programados en el esp-32 en los puertos 34,35 y 36. Cabe destacar que para la construcción del prototipo se utilizó la arquitectura física, desarrollada en el objetivo 2 (Ver anexo 4 sección 5.3 vistas de la arquitectura). En la figura 60, se muestra la conexión de los sensores a un módulo esp-32 que se encuentra conectado a un protoboard.

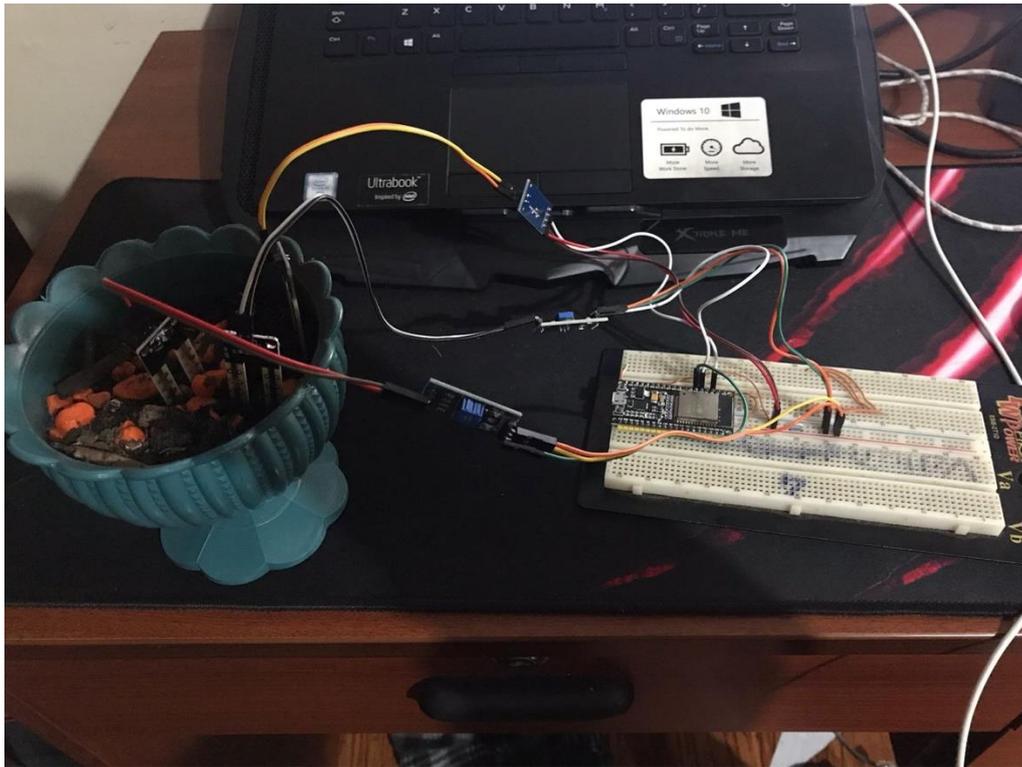


Figura 59. Conexión sensores a modulo esp-32

Una vez conectados los sensores y actuadores, se procedió a validar que estos envían datos en tiempo real para capturar la humedad de los sensores. En la figura 61 se puede ver cómo se actualizan los monitores de cada sensor. cuando estos envían datos de humedad de suelo.

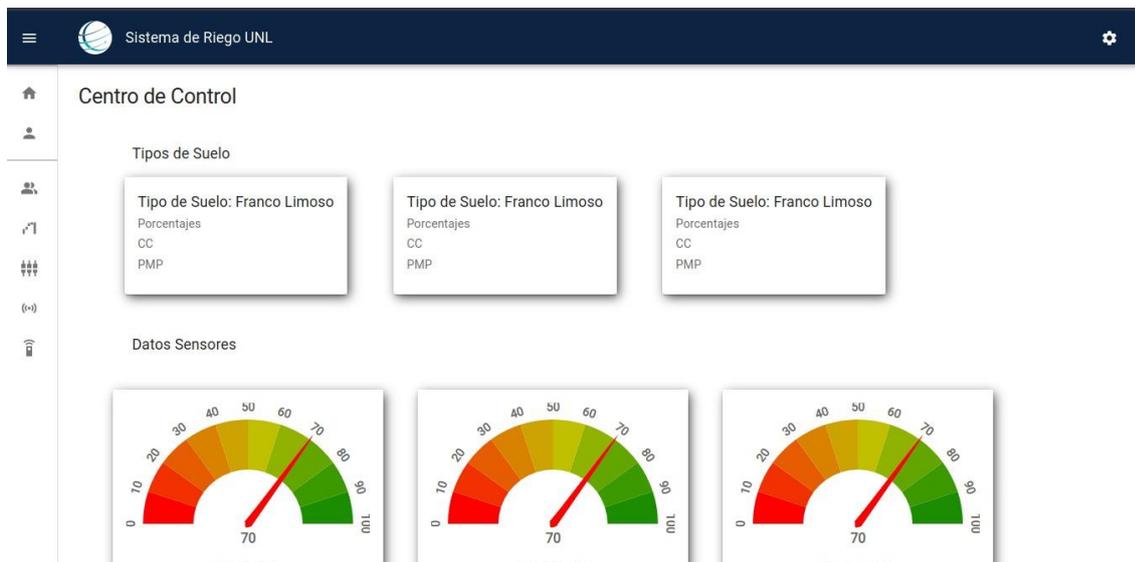


Figura 60. Presentación datos de humedad en vista

Es importante mencionar que configure que el riego se va a encender o apagar de acuerdo a los valores obtenidos en el experimento realizado para obtener la humedad del suelo en el transcurso de los días. Para profundizar sobre los valores utilizados ir a anexo 6.

Finalmente se construyó una estructura que contiene los circuitos electrónicos, así como una maceta que contiene un tipo de suelo franco limoso que es el más adecuado para sembrar cultivos. En la figura 62 se presenta la estructura final del prototipo desarrollado.



Figura 61. Estructura de prototipo final del sistema de irrigación

6.4. Experimentación

En el presente TT se desarrolló un experimento para validar que los sensores utilizados, detectan el cambio de humedad en el suelo. Para lo cual se utilizaron varios recipientes con dos tipos de suelo, arenoso y francos limosos, mismos que tiene diferentes tipos de estructuras y contienen de diferente forma la humedad. Para profundizar sobre el experimento realizado ir a anexo 6.

El experimento se desarrolló con el método gravimétrico, que consiste en pesar las muestras de suelo y calcular la humedad con el peso de las muestras. Con esto se puede tener la humedad gravimétrica de los tipos de suelo, para comparar con los datos obtenidos por los sensores. Los datos obtenidos en el transcurso del experimento se los presenta en la tabla 17, en donde tiene los pesos de las muestras, la humedad gravimétrica y la humedad por los sensores y un tiempo determinado.

Tabla 17. Datos obtenidos en la primera repetición del experimento

Días		0						4						8					
T. Suelo	Vaso	Peso Seco	Peso 1 o CC	Humedad	Sensor 1	Sensor 2	Sensor 3	Peso 2	Humedad	Sensor 1 %	Sensor 2 %	Sensor 3 %	Peso 2	Humedad	Sensor 1 %	Sensor 2 %	Sensor 3 %		
	AR-01	216,3	252,5	16,74	71	71	62	241,5	11,65	49	55	59	233,3	7,86	29	45	44		
Arenoso	AR-02	210,4	248,8	18,25	61	69	66	236,7	12,5	43	59	67	226,2	7,51	29	35	35		
	AR-03	196,9	233,2	18,44	67	61	67	221,5	12,49	39	55	64	211,6	7,47	30	38	35		
Franco	FL-01	121,4	142,2	17,13	72	70	82	131,9	8,65	64	74	73	122	0,49	36	48	39		
Limoso	FL-02	152,5	177,8	16,59	72	70	75	166,1	8,92	70	73	73	155,8	2,16	42	30	36		
	FL-03	124,5	149,8	20,32	68	72	85	139,2	11,81	73	75	78	131,5	5,62	61	63	55		
Fecha	1-oct-22	3-oct-22						5-oct-22				8-oct-2022							

Luego de obtener los datos de la experimentación, se generaron gráficas de dispersión para determinar cómo se da el cambio de humedad en el transcurso de los días. En la figura 63 se muestra el diagrama obtenido en la primera repetición del tipo de suelo arenoso y en la figura 64 el diagrama de dispersión de la humedad en el tipo de suelo franco limoso.

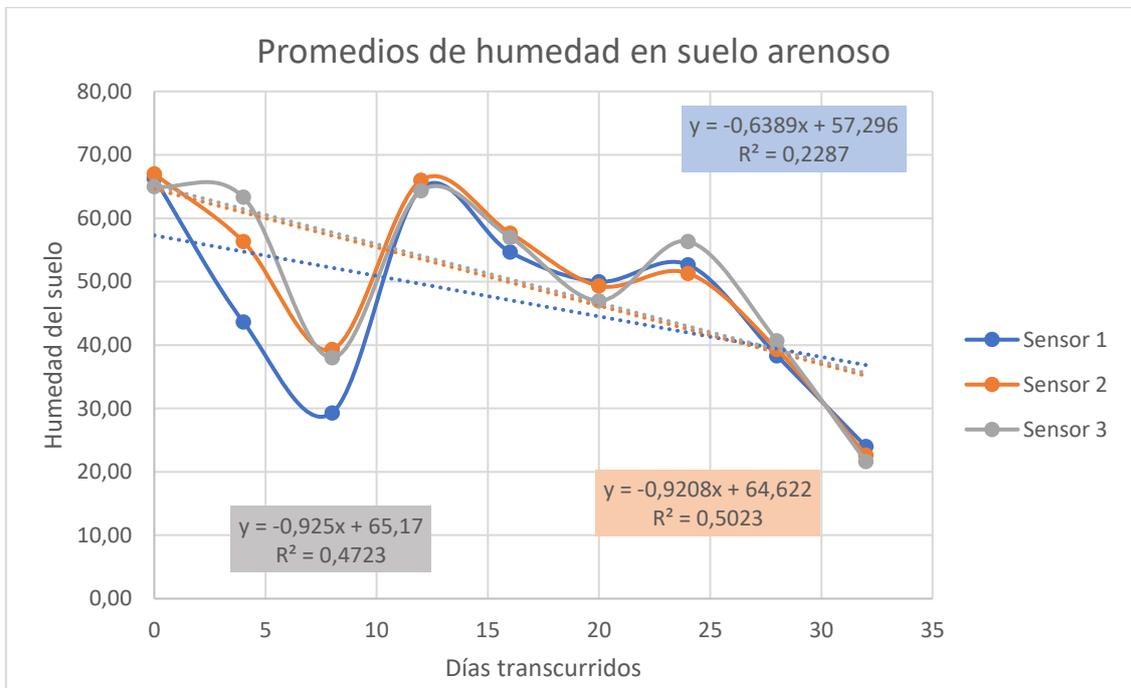


Figura 62. Diagrama de dispersión de repetición de suelo arenoso

Se encontró que los sensores son sensibles al cambio de humedad, esto por los datos obtenidos en el transcurso de 32 días que duró la experimentación con los dos tipos de suelos escogidos. Como se puede ver en las figuras 62 se muestran las gráficas de los datos obtenidos en cada repetición con el tipo de suelo arenoso. La humedad va disminuyendo en relación con el tiempo (días transcurridos) desde el día que se saturó las muestras.

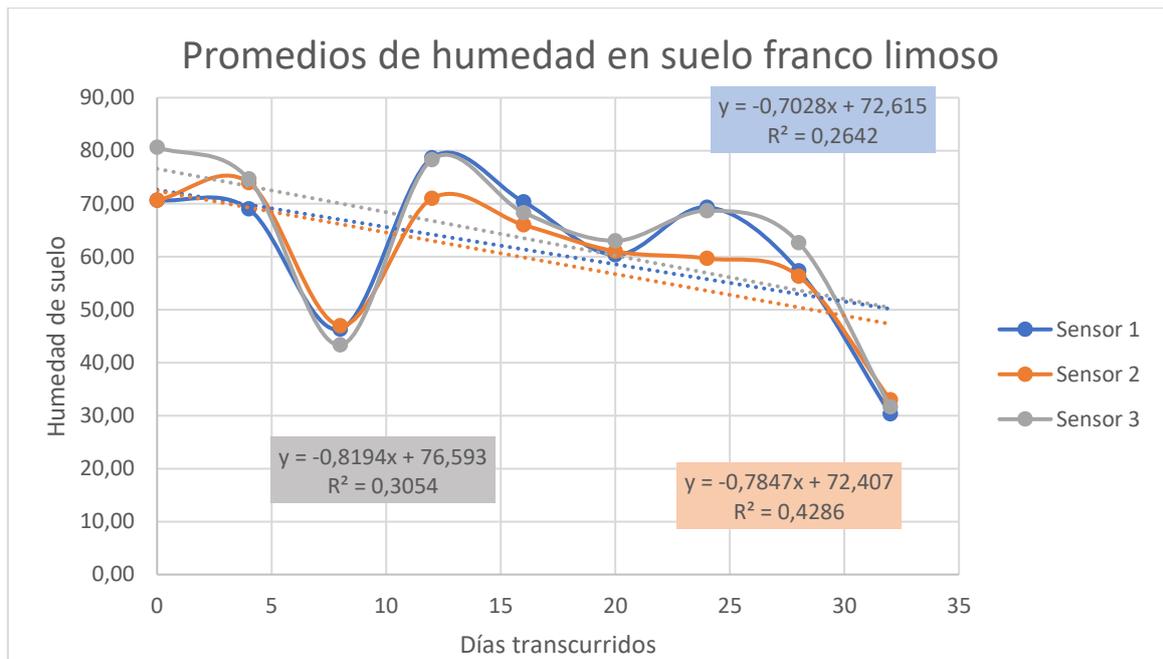


Figura 63. Diagrama de dispersión de repetición en suelo Franco Limoso

En las repeticiones realizadas para el tipo de suelo franco limoso se encontró de igual forma que los sensores perciben un cambio en la humedad de suelo, según van transcurriendo los días desde que se saturó el suelo. Cabe destacar que con este tipo de suelo los sensores obtienen datos de humedad más altos que del arenoso, esto porque el suelo franco limoso, retiene más humedad, por su estructura y los poros de este tipo de suelo son más pequeños en el arenoso. En las figuras 64, se presentan las gráficas obtenidas con las tres repeticiones realizadas al tipo de suelo.

Ahora bien, en el desarrollo del experimento se evidencio que los sensores, obtiene la humedad de suelo en el transcurso del tiempo determinado; pero estos sensores necesitan ser calibrados, porque los sensores toman en consideración las del suelo como humedad, lo que hace que los valores cambien constantemente, que afecta a la precisión de los datos.

7. Discusión

7.1. Desarrollo de la propuesta alternativa

El proyecto de titulación denominado Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja busca presentar una solución informática para automatizar el proceso de riego dentro de los cultivos de la universidad de Loja.

7.1.1. Objetivo Nro.1 Investigar las técnicas de IoT para la construcción de aplicaciones de irrigación.

Con la ejecución del RSL se identificó que existe gran investigación sobre la automatización de sistemas de irrigación, estas investigaciones tienen su procedencia de países de la India con un 80%, de los documentos utilizados para el desarrollo del TT. En cambio, en Ecuador se encontraron un 20% de investigaciones realizadas sobre la automatización de sistemas de irrigación. Siendo estos países agricultores, en Ecuador se ve reflejado el bajo porcentaje en investigación sobre esta temática, dado que con el desarrollo de estos sistemas se puede mejorar el crecimiento de los cultivos. Siendo así, a través de la información recolectada de los documentos seleccionados en el RSL (total fueron 25), se identificó que el desarrollo de sistemas de irrigación con IoT facilitan el proceso de riego y mejora la calidad de los cultivos.

De la misma manera, los protocolos empleados dentro de los sistemas de irrigación se encontró el protocolo MQTT que funciona en redes de pocos recursos y este permite tener comunicación por medio de suscriptores, publicadores y un broker. Este protocolo es idóneo para el proyecto porque permite emplear herramientas open-source, lo que hace que exista más libertad para configurar servidores privados para el uso exclusivo del proyecto. En [89] expresa que el protocolo ZigBee es ideal para este tipo de proyectos, porque crea redes internas con topología en malla para la comunicación de los dispositivos conectados a esta red; pero este protocolo necesita de un emisor y un receptor configurado con este protocolo, (que por lo general son privativos) para la comunicación además que los dispositivos conectados realizan el envío de datos de uno a uno. Por otro lado, existen otro protocolo que se emplea en IoT [38], como es el caso de Coap, es un protocolo de IoT utilizado en la automatización de sistemas de riego que trabaja sobre UDP, es utilizado para la comunicación con dispositivos de baja potencia como sensores y actuadores, este protocolo se asemeja a HTTP, lo que permite realizar peticiones REST para la comunicación, pero este protocolo carece la fiabilidad de mensajes ya que no utiliza procesos para determinar si los mensajes enviados por un emisor llegaron correctamente a un receptor. Por tanto, el protocolo MQTT se adaptó al presente proyecto, porque este permite tener fiabilidad a la hora de enviar mensajes ya que provee tres niveles de calidad de servicio,

además de dar libertad para la configuración y la creación de los servidores, ya que mediante el servidor mosquito que puede ser configurado en cualquier sistema operativo. Un punto importante de este protocolo es que permite la comunicación bidireccional entre dispositivos con el servidor y entre el servidor a dispositivos, así mismo permite la comunicación de muchos a muchos, lo que facilita la comunicación entre un grupo de dispositivos de forma simultánea.

Por último, los problemas que se denotan dentro del desarrollo de los sistemas de irrigación con IoT tienen bastante relevancia a los componentes utilizados ya que estos son de bajo costo y no cuentan con una protección adicional, haciéndolos vulnerables a daños por los cambios climáticos, que hacen que la vida útil de los dispositivos sea corta. Según [90] el desarrollo de estos proyectos ayuda a optimizar los procesos manuales que pueden llevar mucho tiempo, y especifica que los problemas con los componentes hardware se los puede solucionar cambiándolos ya que no tienen un valor elevado, pero los beneficios brindados por estos sistemas son mayores. Cabe destacar que, para extender la vida útil de estos dispositivos, se puede crear coberturas como cintas aislantes o mangueras que protejan sus circuitos de los cambios ambientales que suelen dañar su funcionamiento.

7.1.2. Objetivo Nro.2 Diseñar la arquitectura de sistema de irrigación para recolectar los datos generados por los componentes IoT.

Para dar cumplimiento del segundo objetivo se utilizó una arquitectura que permitió implementar la parte software y hardware dentro del mismo sistema. Dentro de la investigación realizada en el RSL del objetivo 1 según [91] no se puede especificar una única arquitectura para el desarrollo de aplicaciones con IoT, pero se puede buscar la que más se adapte al desarrollo del proyecto; pero presentan varias arquitecturas que se pueden adaptar al alcance de cada proyecto como es el caso de las arquitecturas multicapas de 3, 4 y 5 capas para el desarrollo de estos sistemas.

De este modo se desarrolló mediante la arquitectura de 3 capas, que como se evidencio en el objetivo 1; la arquitectura de 3 capas es utilizada por [92], [93], porque esta se adapta a las necesidades del proyecto en donde solo se necesita controlar el riego, además que es flexible porque permite emplear varios tipos de sensores y tarjetas programables en la capa de percepción así como utilizar servidores o transmisores de datos mediante varios protocolos en la capa de red y finalmente esta arquitectura se adapta para el desarrollo de una solución tecnológica basada en capas como es el caso de un api-rest que es consumida por aplicaciones web y móvil, en la capa de aplicación que es la capa más cercana al usuario. En [94] [95]

expresan que la mejor arquitectura para los sistemas de irrigación con IoT es la de 5 capas, porque esta tiene dos capas adicionales para el análisis de datos y manejo de información con modelos inteligentes. Es importante mencionar que la arquitectura de 5 capas no se utilizó en el presente TT no se implementaron modelos inteligentes para la toma de decisiones, puesto que el control de riego se lo puede realizar en base a los datos obtenidos por los sensores.

7.1.3. Objetivo Nro.3 Desarrollar un módulo de control que permitirá administrar el sistema de irrigación con IoT mediante la metodología XP.

En el presente objetivo se desarrolló el módulo de control que permite visualizar los datos enviados por los sensores. Además, los almacena en una base de datos no relacional MongoDB. A comparación de [92], [90] la cual utiliza herramientas web (ThingSpeak, Ubidots) que les permite utilizar visualizadores como gráficas y monitores, en donde muestran los datos obtenidos por los usuarios, y estos no almacenan información en una base de datos, además que utilizan servidores Broker públicos (HiveMQ, EMQ X) para sus sistemas. A comparación de la presente TT la solución informática desarrollada, permite tener un bróker privado con “mosquitto” para la transmisión de información, de igual forma almacena los datos obtenidos por los sensores en una base de datos, para luego utilizar dichos datos en trabajos futuros en lo que se pueden utilizar modelos inteligentes para analizar dicha información.

7.2. Valoración Técnica Económica Ambiental

En base a recursos técnicos, económicos y ambientales se realizó el presenta trabajo de titulación.

7.2.1. Valoración Técnica

Los recursos técnicos fueron indispensables en el desarrollo de la solución informática ya que con los componentes y herramientas seleccionadas permitieron desarrollar el aplicativo web y móvil, así como la parte de conexión de sensores y actuadores de manera eficiente. Además de facilitar el proceso de riego en cultivos de la Universidad Nacional de Loja.

7.2.2. Valoración Ambiental

La solución informática contribuye de manera positiva con el medio ambiente ya que permite utilizar recursos hídricos limitados para realizar el riego, ya que tomando en cuenta los factores de tipo de suelo se puede determinar qué cantidad de agua se necesita para humedecer el suelo, y con los sensores se puede determinar cuando este suelo se encuentre saturado.

7.2.3. Valoración Económica

En cuanto a los gastos que se realizaron para el desarrollo del presente proyecto, si existieron algunos con costos un poco elevados por los componentes electrónicos utilizados para el desarrollo del mismo.

En la Tabla 18, Tabla 19, se muestra el valor económico referente a servicios y recursos software, hardware y componentes para riego.

Tabla 18. Valoración económica-servicios

Servicio	Descripción	Precio Unitario	Valor Total
Internet	5 meses	\$25.00	\$125.00
Subtotal			\$125.00

Tabla 19: Recursos hardware, software y de riego

Recursos Hardware			
Nombre	Cantidad	Precio Unitario	Valor Total
Modulo Esp32	2	\$15.00	\$30.00
Sensor Fc-28	3	\$3.50	\$10.50
Sensor YF-s201	1	\$10.00	\$10.00
Raspberrypi	1	\$60.00	\$60.00
Relé	1	\$3.00	\$3.00
Electro válvula	1	\$18.00	\$18.00
Cables para modulo Esp32	2	\$6.00	\$12.00
Subtotal			143.50
Componentes para riego			
Manguera	3metros	\$0.29	\$0.87
Goteros	2	\$1.00	\$2.00
grifo	1	\$3.00	\$3.00
Abrazaderas	2	\$0.3125	\$0.63
Teflón	1	\$0.89	\$0.89
Adaptador Hebra	2	\$0.30	\$0.60
Subtotal			\$7.99
Recursos Software			
Nombre	Descripción	Valor Total	
Framework	Nodejs/ReactJs/Flutter	\$0.00	
Editor de Código	IntelliJ IDE	\$0.00	
Gesto de DB	Mongo DB	\$0.00	
Lenguaje de Programación	JavaScript/ Dart	\$0.00	
Firma electrónica	Firma Electrónica	\$15.00	
DigitalOcean	Servidores para web	\$5.00	
Subtotal			\$20.00

En la Tabla 20 se muestra el costo total del desarrollo del proyecto de titulación

Tabla 20. Valores totales de trabajo de titulación

Nombre	Cantidad
Servicio	\$125.00
Recursos Hardware	\$143.50
Componentes para riego	\$7.99
Recursos Software	\$20.00
Subtotal	\$296.49
Imprevistos	\$100
Total	\$396.49

8. Conclusiones

- El protocolo MQTT permitió realizar la comunicación entre el hardware y software del sistema de irrigación con IoT, permitiendo crear un servidor privado que permite el paso de paquetes enviados desde el publicador al suscriptor, además que permitió la comunicación simultanea entre varios dispositivos.
- La arquitectura de tres capas se utilizó en el desarrollo del proyecto ya que permitió definir las tecnologías y protocolos para la comunicación del sistema, además agilizo la construcción de prototipo, gracias a su estructura bien definida.
- La metodología ágil XP ofreció un enfoque estructurado para el desarrollo eficiente y efectivo del software. Proporcionó un marco organizado para el proyecto, comenzando con la fase de planificación donde se recopilaron los requisitos funcionales y no funcionales. Luego, se llevaron a cabo pruebas del sistema para verificar si la solución informática cumplía con los parámetros y especificaciones establecidos por los expertos en riego.
- Los sensores higrómetros son sensibles al cambio de humedad, pero envían datos variables de la humedad ya que asume que estos toman en cuenta las sales que contiene el suelo como parte de la humedad. Por lo cual es necesario realizar la calibración, y así tener mayor precisión con los datos obtenidos.
- El desarrollo del módulo de control para prototipo permitió administrar, el control de riego en base a los datos de humedad de suelo enviados por los sensores, además que permitió visualizar la humedad del suelo y el caudal en tiempo real. Lo que cumple con los requerimientos especificados por los stakeholders.
- Con el desarrollo de prototipo se pudo validar que se puede automatizar un sistema de riego mediante IoT, pero este tiene varias limitaciones, como es el caso de llevar a un ambiente real, se tiene que tomar en consideración una red de sensores, a que distancia se va a colocar y como se van a conectar para el envío de datos, otro factor a considerar es la alimentación de estos dispositivos, y finalmente los sensores a utilizar tienen que ser resistentes a los cambios climáticos, ya que los sensores empleados en el presente proyectos son utilizados para un ambiente de pruebas, que tienen varias limitaciones al llevar a un ambiente real.

9. Recomendaciones

- Para realizar correctamente un RSL se debe tener bien delimitadas las preguntas de investigación que se quieren resolver, para evitar divagar o buscar información que causen ruidos en los resultados que se desean obtener.
- Para implementar el sistema de irrigación con IoT en un ambiente real, se tiene que tomar en consideración varios factores como, crear una red de sensores conectados mediante un protocolo de comunicación, como zigbee o Wifi para el envío de datos, además utilizar una fuente de alimentación como paneles solares para mantener en funcionamiento la captura de sensores.
- Para determinar eficiencia de los recursos hídricos se debe escoger un tipo de cultivo específico, tipo de suelo para determinar el tiempo de riego para este tipo, ya que además de las características de suelo, se debe tomar en consideración la etapa del cultivo, y factores como evapotranspiración, temperatura, evaporación entre otros.
- Para obtener resultados más reales sobre la humedad del suelo por parte de los sensores, es necesario realizar la calibración de los mismos, ya que estos toman datos variados porque se asume que toman en consideración las sales que contiene el suelo, lo que altera los valores y puede ocasionar que el riego se encienda de forma tardía.
- Agregar un módulo para análisis de datos, o modelo inteligente para mejorar la eficiencia del riego, ya que los sensores generan gran cantidad de datos de humedad del suelo.

10. Bibliografía

- [1] F. Maria, “¿Qué es y para qué sirve la agricultura de irrigación? - Agroptima,” Apr. 04, 2020. <https://www.agroptima.com/es/blog/agricultura-de-irrigacion/> (accessed Mar. 12, 2022).
- [2] W. Loaiza Cerón, A. Reyes Trujillo, and Y. Carvajal Escobar, “Aplicación del Índice de Sostenibilidad del Recurso Hídrico en la Agricultura (ISRHA) para definir estrategias tecnológicas sostenibles en la microcuenca Centella,” *Ingeniería y Desarrollo*, vol. 30, no. 2, pp. 160–181, 2012, Accessed: Nov. 13, 2022.
- [3] C. Sociales, D. R. Presenta, Y. M. Jim, and V. S. Solano, “El Colegio de Sonora Cambio tecnológico para la optimización del uso agua en la producción de vid en el distrito de riego 051 Costa de Hermosillo . 1980-2005 .,” 2009.
- [4] OSÉ ROLDÁN CAÑAS, “ENTORNOALRIEGO_conferencia__151107”.
- [5] Victor Emilio Castillo Vásquez, “UNIVERSIDAD NACIONAL DE LOJA ÁREA AGROPECUARIA Y DE RECURSOS NATURALES RENOVABLES TESIS DE GRADO PREVIA A LA OBTENCION DEL TITULO DE MAGISTER EN RIEGO,” 2014. Accessed: Jan. 24, 2022.
- [6] C. Sociales, D. R. Presenta, Y. M. Jim, and V. S. Solano, “El Colegio de Sonora Cambio tecnológico para la optimización del uso agua en la producción de vid en el distrito de riego 051 Costa de Hermosillo . 1980-2005 .,” 2009.
- [7] K. W. Butzer, “Early hydraulic civilization in Egypt: a study in cultural ecology,” *Prehistoric archeology and ecology (USA)*, 1976.
- [8] OSÉ ROLDÁN CAÑAS, “ENTORNOALRIEGO_conferencia__151107,” 2007.
- [9] P.-L. Violette, “Water engineering and management in the early Bronze Age civilizations,” in *Water Engineering and Management through Time*, CRC Press New York, 2011, pp. 39–64. doi: 10.1201/b10560-7.
- [10] A. S. Carrazana, “Universidad Central ‘ Marta Abreu ’ de Las Villas,” *Los sistemas distribuidos. Una aplicación en la enseñanza.*, pp. 30–34, 2006.
- [11] R. Morales, “Sistemas inteligentes de irrigación en la agricultura,” *Universidad Central" Marta Abreu" de Las Villas. Facultad de Ingenier{\'i}a~...*, 2016.
- [12] E. Martin and C. Munoz, “Métodos para Medir la Humedad del Suelo para la Programación del Riego ¿Cuándo?,” *Extensión cooperativa de la Universidad de Arizona*, no. AZ1220s, p. 8, 2017,
- [13] R. J. Fernández and N. Trillo, “La textura del suelo como fuente de heterogeneidad ; sus efectos sobre la oferta de agua para las plantas,” *Researchgate*, no. January 2005, pp. 171–192,
- [14] L. Tamara and J. Ducuara, “Informe - CAPACIDAD DE CAMPO Y PUNTO DE MARCHITEZ PERMANENTE - fisiología vegetal.,” pp. 1–8, 2016,

- [15] O. Delgadillo and L. Pérez, “Medición de la infiltración del agua en el suelo,” p. 34, 2016.
- [16] J. D. P. de Leon, “Sistemas de Irrigación para Agricultura Sub-Tropical | TractorExport.com.” 2020.
- [17] O. Bhat, P. Gokhale, and S. Bhat, “Introduction to IOT,” *International Advanced Research Journal in Science, Engineering and Technology ISO*, vol. 3297, no. 1, 2007, doi: 10.17148/IARJSET.2018.517.
- [18] C. A. Icaza Guamba, “Análisis, diseño y desarrollo de un sistema inteligente y automatizado de monitoreo y control de cultivos con IoT.,” *Universidad Central Del Ecuador*, p. 258, 2021.
- [19] J. Fitzgerald, “A Service of zbw,” *CESifo Forum*, vol. 15, no. 2, pp. 8–13, 2014.
- [20] J. Alfaro, A. S. Corbacho Torregroza, and J. E. Prada Hernandez, “ANALISIS DEL INTERNET DE LAS COSAS (IoT) Y SU APLICABILIDAD EN EL CONTROL REMOTO DEL SISTEMA DE RIEGO DE LA FINCA LA JOSEFA UBICADA EN MACONDO ZONA BANANERA DEL DEPARTAMENTO DEL MAGDALENA.,” *Universidad Cooperativa De Colombia*, 2018.
- [21] J. Paolo et al., “Internet de las cosas aplicado a la agricultura: estado actual,” *Lámpsakos*, ISSN-e 2145-4086, ISSN 2145-4086, No. 22 (Julio-Diciembre), 2019, págs. 86-105, no. 22, pp. 86–105, 2019, doi: 10.21501/21454086.3253.
- [22] C. A. Navarrete Pinto, Sandra Marianela Icaza Guamba, “Repositorio Digital: Análisis, diseño y desarrollo de un sistema inteligente y automatizado de monitoreo y control de cultivos con IoT. Caso de estudio: huertos urbanos.”
- [23] L. García, L. Parra, J. M. Jimenez, J. Lloret, and P. Lorenz, “IoT-Based Smart Irrigation Systems: An Overview on the Recent Trends on Sensors and IoT Systems for Irrigation in Precision Agriculture,” *Sensors*, vol. 20, no. 4, p. 1042, Feb. 2020, doi: 10.3390/s20041042.
- [24] A. Glória, C. Dionisio, G. Simões, J. Cardoso, and P. Sebastião, “Water Management for Sustainable Irrigation Systems Using Internet-of-Things,” *Sensors*, vol. 20, no. 5, p. 1402, Mar. 2020, doi: 10.3390/s20051402.
- [25] C. Kishor, H. U. Sunil Kumar, H. S. Praveena, S. P. Kavya, G. B. Apeksha, and D. J. Nayaka, “Water usage approximation of Automated Irrigation System using IOT and ANN’s,” in *Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2018*, Aug. 2019, pp. 76–80. doi: 10.1109/I-SMAC.2018.8653692.
- [26] J. Karpagam, I. I. Merlin, P. Bavithra, and J. Kousalya, “Smart irrigation system using IoT,” in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020, pp. 1292–1295.
- [27] D. R. González, “Arquitectura y Gestión de la IoT,” *Telemática*, vol. 12, no. 3, pp. 49–60, 2013.

- [28] O. Debauche, S. Mahmoudi, P. Manneback, and F. Lebeau, “Cloud and distributed architectures for data management in agriculture 4.0 : Review and future trends,” *Journal of King Saud University - Computer and Information Sciences*, Oct. 2021, doi: 10.1016/j.jksuci.2021.09.015.
- [29] W. Honghui et al., “Research of the Hardware Architecture of the Geohazards Monitoring and Early Warning System Based on the Iot,” *Procedia Comput Sci*, vol. 107, pp. 111–116, 2017, doi: 10.1016/j.procs.2017.03.065.
- [30] F. M. Biot et al., “Las redes de sensores inalámbricos y el Internet de las cosas,” *Sensors (Switzerland)*, vol. 20, no. 137, pp. 49–60, 2017, doi: 10.3390/s20051402.
- [31] L. García, L. Parra, J. M. Jimenez, J. Lloret, and P. Lorenz, “IoT-based smart irrigation systems: An overview on the recent trends on sensors and iot systems for irrigation in precision agriculture,” *Sensors (Switzerland)*, vol. 20, no. 4, p. 1042, 2020, doi: 10.3390/s20041042.
- [32] E. Andrés et al., “Propuesta de una Arquitectura para Agricultura de Precisión Soportada en IoT Proposal of an Architecture for Precision Agriculture Supported in IoT”, doi: 10.17013/risti.24.39-56.
- [33] “Demostrador arquitectura publish/subscribe con MQTT”.
- [34] E. D. ACOSTA GALLO and C. R. CUAICAL ANGULO, “Desarrollo De Un Sistema Automático De Irrigación Para Huertos Caseros Por Medio De Iot Con Software Y Hardware Libre,” 2021.
- [35] F. Mahedero Biot, “Desarrollo de una aplicación IoT para el envío de imágenes mediante el protocolo MQTT,” 2020.
- [36] M. C. Collado and J. G. Llobera, “TRABAJO FINAL DE GRADO TÍTULO DEL TFG: Monitorización de sensores con arduino utilizando el protocolo MQTT TITULACIÓN: Ingeniería de Sistemas de Telecomunicaciones”.
- [37] D. Ruz, “Evaluacion del protocolo HTTP/2 para internet de las cosas,” p. 55, 2019.
- [38] G. Caiza, E. Alvarez-m, E. Remache, and A. Ortiz, “Comparación de AMQP y CoAP para la integración de las comunicaciones en el área de producción,” *Revista Ibérica de Sistemas e Tecnologías de Informação*, pp. 652–667, 2020.
- [39] A. Cama, E. De la Hoz, and D. Cama, “Las redes de sensores inalámbricos y el Internet de las cosas Palabras clave,” 2012.
- [40] N. Mechatronics, “Sensor De Humedad De Suelo,” 2021.
- [41] “YF-S401 pdf, YF-S401 Descripción Electrónicos, YF-S401 Datasheet, YF-S401 view ::: ALLDATASHEET :::”
- [42] m&m International, “Electroválvulas M & M International significa :,” 2016.
- [43] D. Bolaños, “Módulo de relés,” vol. 8, pp. 1–10, 2016.

- [44] Araujo Evelyn, “Implementación de un sistema de video vigilancia para los exteriores de la UPS, mediante mini computadores y cámaras Raspberry PI,” Universidad Politecnica Salesiana Ecuador, p. 108, 2015.
- [45] D. P. Mochales, “Universidad de Alcalá Escuela Politécnica Superior.”
- [46] J. E. Pérez, “Introducción a JavaScript,” Introducción a JavaScript. 2005.
- [47] J. Mohedano, J. M. Saiz, and P. Salazar Román, *Iniciación a javascript*. Ministerio de Educación, 2018.
- [48] L. Puciarelli, *Node JS-Vol. 1: Instalación-Arquitectura-node y npm*, vol. 1. RedUsers, 2020.
- [49] P. & G. & G. & M. & Catania Iacono et al., “Frontend y backend en Node . js para una aplicación IoT Trabajo Final de Grado Presentado en la facultad Escola Tècnica Superior d ’ Enginyeria de Telecomunicació de Barcelona Universitat Politècnica de Catalunya por Jonathan Camacho Torrens En cumplimien,” Universitat Politècnica de Catalunya, 2018.
- [50] K. Calvo, J. Durán, E. Quirós, and E. Malinowski, “MongoDB: alternativas de implementar y consultar documentos,” in *IX Congreso Internacional de Computación y Telecomunicaciones*, 2017, pp. 78–86. doi: ISSN 2519-0679.
- [51] A. R. Arcentales Viteri, “Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica, Automatización y Control,” pp. 1–141, 2021.
- [52] J. J. Villar, “Descubre React,” Leanpub, 2015.
- [53] L. Dagne, “Flutter for Cross-Platform App and SDK Development,” *Metropolia University of Applied Sciences*, no. May, pp. 1–28, 2019.
- [54] “Flutter for Beginners: An introductory guide to building cross-platform ... - Alessandro Biessek - Google Libros.”
- [55] Cedalo, “Eclipse Mosquitto,” Eclipse Mosquitto, 2018.
- [56] M. D. S. Soares, “Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software,” *Revista Eletrônica de Sistemas de Informação*, vol. 3, no. 1, 2004, doi: 10.21529/resi.2004.0301006.
- [57] J. Calderon, W. Diaz, N. Marquez, and Z. Angulo, “Metodologías para el desarrollo de software educativo,” *Universidad Autónoma del Caribe*, vol. 11, no. 2, p. 5, 2013.
- [58] J. Canos, P. Letelier, and C. Penades, “Metodologias Agiles en el Desarrollo de Software,” *Taller Metodologias Agiles en el Desarrollo de Software*, p. 59, 2003.
- [59] S. M. Meléndez, M. E. Gaitan, and N. N. Pérez, “Metologia Agil de Desenvolvimento de Software,” *Universidad Nacional Autónoma de Nicaragua, Managua*, 2016.
- [60] J. Ignacio et al., “Construcción De Un Prototipo De Un Sistema De Riego Y Ventilación De Un Invernadero Bajo El Concepto De Iot Construction Of A Prototype Of An Irrigation And Ventilation System Of A Greenhouse Under The Concept Of Iot,” *Tecnológico Nacional de México en Celaya Pistas Educativas*, vol. 2020, no. 137.

- [61] J. Alfaro, A. S. Corbacho Torregroza, and J. E. Prada Hernandez, “ANALISIS DEL INTERNET DE LAS COSAS (IoT) Y SU APLICABILIDAD EN EL CONTROL REMOTO DEL SISTEMA DE RIEGO DE LA FINCA LA JOSEFA UBICADA EN MACONDO ZONA BANANERA DEL DEPARTAMENTO DEL MAGDALENA.,” Universidad Cooperativa De Colombia, 2018.
- [62] J. E. Gómez, S. Castaño, T. Mercado, A. Fernandez, and J. Garcia, “Sistema de internet de las cosas (IoT) para el monitoreo de cultivos protegidos,” *Ingeniería e Innovación*, vol. 5, no. 1, 2018, doi: 10.21897/23460466.1101.
- [63] About Parsifal, “About · Parsifal.” <https://parsif.al/about/> (accessed May 21, 2022).
- [64] B. Kitchenham, “Procedures for performing systematic literature reviews,” Joint Technical Report, Keele University TR/SE-0401 and NICTA TR-0400011T.1, vol. 33, no. 2004, p. 33, 2004,
- [65] A. Ahmed and S. Rawal, “IOT based Smart Irrigation System Cite this paper IOT based Smart Irrigation System,” *Int J Comput Appl*, vol. 159, no. 8, pp. 975–8887, 2017.
- [66] A. Imteaj, T. Rahman, M. K. Hossain, and S. Zaman, “IoT based autonomous percipient irrigation system using raspberry Pi,” in 2016 19th International Conference on Computer and Information Technology (ICCIT), Dec. 2016, pp. 563–568. doi: 10.1109/ICCITECHN.2016.7860260.
- [67] E. Seethalakshmi, M. Shunmugam, R. Pavaiyarkarasi, S. Joseph, and J. Edward paulraj, “An automated irrigation system for optimized greenhouse using IoT,” *Mater Today Proc*, no. xxxx, Feb. 2021, doi: 10.1016/j.matpr.2020.12.636.
- [68] J. H. Yousif and K. Abdalgader, “Experimental and Mathematical Models for Real-Time Monitoring and Auto Watering Using IoT Architecture,” *Computers*, vol. 11, no. 1, 2022, doi: 10.3390/computers11010007.
- [69] M. Mohammed, K. Riad, and N. Alqahtani, “Efficient IoT-Based Control for a Smart Subsurface Irrigation System to Enhance Irrigation Management of Date Palm.,” *Sensors (Basel)*, vol. 21, no. 12, p. 3942, Jun. 2021, doi: 10.3390/s21123942.
- [70] S. K. Roy, S. Misra, N. S. Raghuvanshi, and S. K. Das, “AgriSens: IoT-Based Dynamic Irrigation Scheduling System for Water Management of Irrigated Crops,” *IEEE Internet Things J*, vol. 8, no. 6, pp. 5023–5030, Mar. 2021, doi: 10.1109/JIOT.2020.3036126.
- [71] M. Mannan J., K. S. S., D. M., and P. T., “Smart scheduling on cloud for IoT-based sprinkler irrigation,” *International Journal of Pervasive Computing and Communications*, vol. 17, no. 1, pp. 3–19, Feb. 2021, doi: 10.1108/IJPCC-03-2020-0013.
- [72] S. Borah, R. Kumar, and S. Mukherjee, “Low-cost IoT framework for irrigation monitoring and control,” *International Journal of Intelligent Unmanned Systems*, vol. 9, no. 1, pp. 63–79, Jun. 2020, doi: 10.1108/IJIUS-12-2019-0075.
- [73] S. J. Shiva, S. Palanivel, and S. C. Venkateswarlu, “IoT based Smart Irrigation System by using ESP32 and Adafruit IO,” *CEUR Workshop Proc*, vol. 3058, pp. 0–2, 2021.

- [74] M. Hamdi, A. Rehman, A. Alghamdi, M. A. Nizamani, M. M. S. Missen, and M. A. Memon, "Internet of Things (IoT) Based Water Irrigation System," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 17, no. 05, p. 69, May 2021, doi: 10.3991/ijoe.v17i05.22081.
- [75] N. Zulkifli et al., "Evapotranspiration (ET)-Based Irrigation System with Internet of Things (IoT) Integration for Capsicum Annuum Farming: A Methodology," in *2020 IEEE 5th International Symposium on Telecommunication Technologies (ISTT)*, Nov. 2020, pp. 88–93. doi: 10.1109/ISTT50966.2020.9279371.
- [76] K.-M. Chew, S. C.-W. Tan, G. C.-W. Loh, N. Bundan, and S.-P. Yiiiong, "IoT Soil Moisture Monitoring and Irrigation System Development," in *Proceedings of the 2020 9th International Conference on Software and Computer Applications*, Feb. 2020, pp. 247–252. doi: 10.1145/3384544.3384595.
- [77] W. Li et al., "Review of Sensor Network-Based Irrigation Systems Using IoT and Remote Sensing," *Advances in Meteorology*, vol. 2020, pp. 1–14, Sep. 2020, doi: 10.1155/2020/8396164.
- [78] S. López-Torres et al., "IoT Monitoring of Water Consumption for Irrigation Systems Using SEMMA Methodology," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11886 LNCS, 2020, pp. 222–234. doi: 10.1007/978-3-030-44689-5_20.
- [79] D. Rani, N. Kumar, and B. Bhushan, "Implementation of an Automated Irrigation System for Agriculture Monitoring using IoT Communication," in *2019 5th International Conference on Signal Processing, Computing and Control (ISPCC)*, Oct. 2019, vol. 2019-October, pp. 138–143. doi: 10.1109/ISPCC48220.2019.8988390.
- [80] C. P. Meher, A. Sahoo, and S. Sharma, "IoT based Irrigation and Water Logging monitoring system using Arduino and Cloud Computing," in *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, Mar. 2019, pp. 1–5. doi: 10.1109/ViTECoN.2019.8899396.
- [81] C. Kamienski et al., "Smart Water Management Platform: IoT-Based Precision Irrigation for Agriculture," *Sensors*, vol. 19, no. 2, p. 276, Jan. 2019, doi: 10.3390/s19020276.
- [82] J. A. B., *Telematics and Computing*, vol. 1053. Cham: Springer International Publishing, 2019. doi: 10.1007/978-3-030-33229-7.
- [83] D. Hovhannisyan, A. Eltawil, and F. Kurdahi, "Testing topology adaptive irrigation IoT with circuits," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 2019-May, 2019, doi: 10.1109/ISCAS.2019.8702559.
- [84] A. Kumar Verma, B. K. V. Akhil, and A. Singh, "Wireless network based smart irrigation system using IOT," *International Journal of Engineering & Technology*, vol. 7, no. 1.1, p. 342, Dec. 2017, doi: 10.14419/ijet.v7i1.1.9849.
- [85] M. N. Rajkumar, S. Abinaya, and V. V. Kumar, "Intelligent irrigation system — An IOT based approach," in *2017 International Conference on Innovations in Green Energy and Healthcare Technologies (IGEHT)*, Mar. 2017, pp. 1–5. doi: 10.1109/IGEHT.2017.8094057.

- [86] M. Kranthi Kumar and K. Srenivasa Ravi, "Automation of Irrigation System based on Wi-Fi Technology and IOT," *Indian J Sci Technol*, vol. 9, no. 17, May 2016, doi: 10.17485/ijst/2016/v9i17/93048.
- [87] A. Kumar, P. Ranjan, and V. Saini, "Smart Irrigation System Using IoT," in *Advanced Series in Management*, vol. 27, no. July, 2022, pp. 123–139. doi: 10.1108/S1877-636120220000027009.
- [88] D. Amu, A. Amuthan, S. S. Gayathri, and A. Jayalakshmi, "Automated Irrigation using Arduino sensor based on IOT," in *2019 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2019, pp. 1–6. doi: 10.1109/ICCCI.2019.8822147.
- [89] M. Kranthi Kumar and K. Srenivasa Ravi, "Automation of irrigation system based on Wi-Fi technology and IOT," *Indian J Sci Technol*, vol. 9, no. 17, May 2016, doi: 10.17485/ijst/2016/v9i17/93048.
- [90] Román Alvarado Rogger Estalin, "Sistema de monitoreo y control de irrigación usando Internet de las Cosas (IoT)".
- [91] W. Velásquez Vargas, L. Guerra, J. Smit Rueda, and J. Pérez Luis Alberto Velásquez Vargas Washington Adrián Vinueza Escobar Nelson Fernando, "Technological Innovation for Health and Personal Electromagnetic Safety and Francisco Falcone Similitudes y diferencias entre Redes de Sensores Inalámbricas e Internet de las Cosas: Hacia una po...."
- [92] E. Seethalakshmi, M. Shunmugam, R. Pavaiyarkarasi, S. Joseph, and J. Edward paulraj, "An automated irrigation system for optimized greenhouse using IoT," *Mater Today Proc*, no. xxxx, Feb. 2021, doi: 10.1016/j.matpr.2020.12.636.
- [93] A. Imteaj, T. Rahman, M. K. Hossain, and S. Zaman, "IoT based autonomous percipient irrigation system using raspberry Pi," in *2016 19th International Conference on Computer and Information Technology (ICCIT)*, Dec. 2016, pp. 563–568. doi: 10.1109/ICCITECHN.2016.7860260.
- [94] J. H. Yousif and K. Abdalgader, "Experimental and Mathematical Models for Real-Time Monitoring and Auto Watering Using IoT Architecture," *Computers*, vol. 11, no. 1, Jan. 2022, doi: 10.3390/computers11010007.
- [95] N. Zulkifli et al., "Evapotranspiration (ET)-Based Irrigation System with Internet of Things (IoT) Integration for Capsicum Annuum Farming: A Methodology," in *2020 IEEE 5th International Symposium on Telecommunication Technologies, ISTT 2020 - Proceedings*, Nov. 2020, pp. 88–93. doi: 10.1109/ISTT50966.2020.9279371.
- [96] I. G. C. Alexander, "Trabajo de titulación modalidad Proyecto Integrador, previo a la obtención del título de Ingeniero en Computación Gráfica".
- [97] V. Cortes, C. Marco, and F. V. Garcia, "Diseño E Implementación De Un Sistema De Riego Automatizado Y Monitoreo De Variables Ambientales Mediante Iot En Los Cultivos Urbanos De La Fundación Mujeres Empresarias Marie Poussepin."

- [98] L. García, L. Parra, J. M. Jimenez, J. Lloret, and P. Lorenz, “IoT-based smart irrigation systems: An overview on the recent trends on sensors and iot systems for irrigation in precision agriculture,” *Sensors (Switzerland)*, vol. 20, no. 4. MDPI AG, Feb. 02, 2020. doi: 10.3390/s20041042.
- [99] J. Camacho, K. Campos, P. Cedillo, B. Coronel, and A. Bermeo, “Forensics Analysis on Mobile Devices:A Systematic Mapping Study,” in *Advances in Intelligent Systems and Computing*, vol. 884, Springer Verlag, 2019, pp. 57–72. doi: 10.1007/978-3-030-02828-2_5.
- [100] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, “Lessons from applying the systematic literature review process within the software engineering domain,” *Journal of Systems and Software*, vol. 80, no. 4, pp. 571–583, Apr. 2007, doi: 10.1016/j.jss.2006.07.009.
- [101] J. Medeiros, M. Goulao, A. Vasconcelos, and C. Silva, “Towards a Model about Quality of Software Requirements Specification in Agile Projects,” in *2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC)*, Sep. 2016, pp. 236–241. doi: 10.1109/QUATIC.2016.058.
- [102] F. W. Neiva, J. M. N. David, R. Braga, and F. Campos, “Towards pragmatic interoperability to support collaboration: A systematic review and mapping of the literature,” *Inf Softw Technol*, vol. 72, pp. 137–150, Apr. 2016, doi: 10.1016/j.infsof.2015.12.013.
- [103] J. Popay, L. Arai, and N. Britten, “Guidance on the conduct of narrative synthesis in systematic reviews: A product from the ESRC Methods Programme,” 2006, doi: 10.13140/2.1.1018.4643.
- [104] M. S. Miguel and A. Villamagua, “CARRERA DE INGENIERIA AGRÍCOLA FISICA DE SUELOS.”

11. Anexo

Anexo 1 Revisión Sistemática de literatura

Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja: Revisión Sistemática de Literatura

Minga-Medina-Angel, Nole-Reyes-Alex

Universidad Nacional de Loja, Carrera de Ingeniería en Sistemas

Loja - Ecuador

{afmingam [alex.nole](mailto:alex.nole@unl.edu.ec)}@unl.edu.ec

Resumen — La agricultura es una de las principales fuentes de ingresos del Ecuador con el 38% de las exportaciones, sin embargo, en sectores de cultivos como la papa se lo realiza de forma ancestrales y empíricamente. Además, en zonas áridas, existen problemas con el abastecimiento del recurso hídrico y en otras zonas con mayor cantidad de agua, existe desperdicio de recursos hídricos

El trabajo actual ofrece un análisis sistemático de la literatura existente, en el que se llevó a cabo un mapeo y revisión para identificar, analizar y clasificar los artículos publicados sobre el uso de protocolos, arquitecturas, metodologías, herramientas y problemas relacionados con el desarrollo de sistemas de irrigación utilizando la tecnología del Internet de las Cosas (IoT).

Para realizar la revisión se utilizó la técnica propuesta por Barbara Kitchenham, con el uso de la herramienta Parsifal, la misma que permitió seleccionar 25 artículos desde el año 2016 en adelante, los cuales detallan información sobre, las principales arquitecturas, protocolos, metodologías, herramientas y problemas identificados en la revisión son: Arquitecturas de tres, cuatro y cinco capas para desarrollo de sistemas con IoT. El protocolo más utilizado es el MQTT. Las metodologías analizadas son propuestas por los autores de los artículos y proponen metodologías con bases matemáticas para el control optimizar al máximo el uso de agua en los cultivos. Las herramientas más utilizadas MCU Esp8266, Esp32, Arduino, Sensores de humedad del suelo, sensor de temperatura, electroválvulas, sensores de caudal, rasperry pi. Entre los problemas analizados encontró que cuando existen fallas en los dispositivos electrónicos, solo se puede solucionar reemplazándolos por otros.

Finalmente, Existen varios estudios relacionados con IoT dentro de la agricultura, específicamente en sistemas de irrigación, donde detallan que la implementación de estos sistemas mejora en la calidad de los cultivos ya que permite llevar un control sobre el proceso de riego, llevando el agua necesaria para las plantas y en el tiempo adecuado. Evitando el desperdicio de agua o que los cultivos se marchiten.

Palabras Clave — Internet de las cosas (IoT), sistemas de irrigación, protocolos, arquitecturas, metodologías, herramientas, problemas.

Abstract Agriculture is one of the main sources of income in Ecuador with 38% of exports, however, in crop sectors such as potatoes, it is done in an ancestral and empirical way. In addition, in arid areas, there are problems with the supply of water resources and in other areas with greater amounts of water, there is waste of water resources.

This paper presents a mapping and systematic literature review in order to identify, analyze and classify published study articles on the use of protocols, architectures, methodologies, tools and problems in the development of irrigation systems with the Internet of Things (IoT).

To carry out the review, the technique proposed by Barbara Kitchenham was carried out, with the use of the Parsifal tool, which turned out to select 25 articles from 2016 onwards, which detail information on the main architectures, protocols, methodologies, tools and problems identified in the review are: Architectures of three, four and five layers for the development of systems with IoT. The most widely used protocol is MQTT. The methodologies analyzed are proposed by the authors of the articles and they propose methodologies with mathematical bases for the control to optimize the use of water in crops to the maximum. The most used tools MCU Esp8266, Esp32, Arduino, Soil moisture sensors, temperature sensor, solenoid valves, flow sensors, rasperry pi. Among the problems analyzed, it was found that when there are faults in electronic devices, it can only be solved by replacing them with others.

Finally, there are several studies related to IoT within agriculture, specifically in irrigation systems, where they detail that the implementation of these systems improves the quality of crops since it allows control over the irrigation process, taking the water necessary for the plants and at the right time. Avoiding wasting water or crops wilting.

Keywords— Internet of things (IoT), irrigation systems, protocols, architectures, methodologies, tools, problems.

I. INTRODUCCIÓN

La irrigación es un conjunto de dispositivos capaces de proporcionar caudal de agua en las tierras más pobres de forma artificial y ordenada para una producción agrícola eficiente. Se trata de abastecer la tierra con la cantidad de agua necesaria, utilizando métodos artificiales para lograr una buena producción [1].

La agricultura logra los mejores resultados cuando se cumplen las condiciones ambientales adecuadas y se proveen eficientemente los recursos esenciales, como el agua. La humedad del suelo y el suministro de agua son de gran importancia en la agricultura, pero un exceso de humedad puede provocar la aparición de plagas y enfermedades que afectan negativamente el crecimiento de las plantas y la calidad del producto. Al utilizar sistemas de riego eficientes, se puede ahorrar significativamente en el uso de agua y optimizar el nivel de humedad del suelo necesario para el crecimiento de las plantas [6].

Ahora bien, Según [3] en su tesis expresa que la mayor parte del uso del agua en el Ecuador es para riego, que se estima en un 80% del uso total del agua, similarmente, las pérdidas en captación, conducción primaria, secundaria y terciaria, en el ámbito de parcela, hacen que la eficiencia varíe entre el 15% y 25%. De tal manera la irrigación representa el 82,10% del consumo de agua de Ecuador [4]. Sin embargo, la infraestructura de riego es extremadamente ineficiente, por lo tanto, se desperdicia mucha agua. Según [5], varios estudios han determinado que la eficiencia de la infraestructura de riego pública es del 30%, mientras que la eficiencia del sector privado está entre el 16% y el 50%. Solo los usuarios privados que cultivan productos de exportación cuentan con tecnologías de riego que ahorran agua. La mayoría de los canales de riego del sector privado no son impermeables, pero del sector público lo es, cabe destacar que el agua se desperdicia a menudo cuando se asignan pequeños terrenos que no cuentan con la tecnología suficiente para conservar el agua.

De tal manera, en la provincia de Loja según la información registrada en la Secretaría de Estado del Agua (SENAGUA), el 88,29% de las concesiones han sido otorgadas para riego y consumo domiciliario, y su caudal representa el 97,55% del total concesionado, concretamente 37,91 m³/s para riego, 0,92 m³/s para consumo doméstico. Por tanto, la agricultura bajo riego es la que más consume agua [6].

La correlación entre el agua y la producción agrícola es proporcional, ya que el recurso hídrico es imprescindible para el correcto desarrollo de los cultivos. No obstante, para lograr los resultados óptimos, es importante brindar los cuidados básicos adecuados. Investigaciones han evidenciado que cada especie vegetal tiene requerimientos únicos, como temperaturas ideales particulares y cantidades específicas de agua. De esta

forma, resulta crucial asegurarse de que el riego de los cultivos sea el adecuado para satisfacer las necesidades particulares de cada tipo de planta [7].

Sin embargo, el impacto ambiental de los sistemas de irrigación o de riego debe ser considerado un aspecto a mejorar. Dado que el riego tradicional puede ocasionar el uso excesivo de agua o, por el contrario, que la cantidad suministrada no sea suficiente. Además, esto generalmente se hace en horarios regulares, independientemente de las condiciones de cultivo diarias[8].

Por otra parte, en la actualidad, Internet ha dejado de ser solamente un medio de comunicación y se ha convertido en un elemento fundamental en la vida cotidiana. Los dispositivos inteligentes conectados a Internet [9], también conocidos como Internet de las cosas (IoT), son una de las últimas tendencias tecnológicas. Como resultado, los proyectos de ingeniería sustentable de las casas inteligentes son solo una de las muchas aplicaciones que la IoT puede proporcionar. Esto puede compararse con el cultivo de un sembradío, ya que es una alternativa sostenible que puede proporcionar mejores resultados en la agricultura.

Con lo cual el desarrollo de un sistema de irrigación con IoT, automatizará el proceso de riego proporcionando un control eficiente del mismo, ya que se obtendrán datos de humedad y temperatura por medios de sensores, lo que permitirá tener un control sobre la cantidad de agua y el tipo de riego que se implementará de acuerdo a cada fase de crecimiento de las plantas. Por lo que el sistema de irrigación con IoT ayudará en la contribución con el medio ambiente y de minimización e impactos negativos en el ahorro considerable de consumo de agua[5].

Es por esto que la presente RSL tiene como objetivo determinar las técnicas para la construcción de un sistema de irrigación con IoT, en donde se pretende obtener información concisa sobre las arquitecturas, protocolos, herramientas y problemas que se presentan en el desarrollo de estas soluciones informáticas.

En el presente documento se dará a conocer el proceso de la revisión sistemática de literatura (RSL), para la obtención de información relacionada con las técnicas para la construcción de un sistema de irrigación con IoT, presenta la siguiente estructura II. Metodología de investigación, III Resultados, IV. Discusión, V. Conclusiones.

II. MÉTODO DE INVESTIGACIÓN

En [9], se afirma que cualquier investigación debe fundamentarse en evidencias que permitan identificar cómo otros autores han abordado el objeto de estudio. No obstante, para agregar evidencia, es necesario recurrir a estudios secundarios, tales como las Revisiones Sistemáticas de la Literatura (RSL) y los Estudios de Mapeo Sistemático [10]. En el presente trabajo, se utilizó el protocolo establecido por [7] y se estructuró el documento en relación a trabajos previos de [11], [12].

A. Preguntas de Investigación

A partir del tema central “Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja”, se planteó cinco preguntas divididas en una para el mapeo sistemático (MQ) y cinco para la revisión sistemática (RQ), con el motivo de identificar los artículos más relevantes en la revisión sistemática.

- MQ1: ¿Cuántos estudios se publicaron a lo largo de los años sobre los sistemas de irrigación con IoT?
- RQ1: ¿Qué técnicas de IoT se pueden utilizar para construir aplicaciones de irrigación?
- RQ2: ¿Qué componentes se necesitan para la construcción de aplicaciones de irrigación con IoT?
- RQ3: ¿Qué arquitecturas de IoT son utilizadas para el desarrollo de sistemas de irrigación?
- RQ4: ¿Qué protocolos de red son idóneos para la construcción de aplicaciones de irrigación?
- RQ5: ¿Qué problemas se presentan en la construcción de aplicaciones de irrigación con IoT?

B. Proceso de Búsqueda

Se determinaron un conjunto de términos basados en las preguntas de investigación para construir la cadena de búsqueda. Para esto se utilizó la metodología PICOC propuesto por [11] para definir el ámbito del RSL:

- Población (P): “Irrigation system”.
- Intervención (I): “IoT”, Internet of things.
- Comparación (C): S/N
- Resultados (O): “Methodology, Standard, Method, Architecture, Protocol”.
- Contexto (C): IoT.

C. Definición de los criterios de inclusión (IC) y exclusión (EC)

Para poder clasificar los documentos obtenidos, fue necesario establecer criterios de inclusión y exclusión.

Se especificaron 6 Criterios de Inclusión (IC):

- IC1: Los artículos y procedimiento deben estar escritos en inglés.

- IC2: Los artículos y procedimiento detallan las técnicas de IoT que se pueden utilizar para la construcción de aplicaciones de irrigación.
- IC3: Los artículos y procedimiento deben contener información relacionada con sistemas de irrigación con IoT.
- IC4: Los artículos y procedimiento detallan arquitecturas de IoT para la construcción de sistemas de irrigación
- IC5: Los artículos y procedimiento detallan los componentes que se necesitan para la construcción de aplicaciones de irrigación con IoT.
- IC6: Los artículos y procedimiento detallan protocolos que se pueden utilizar para la construcción de aplicaciones de irrigación con IoT.

Se especificaron 11 criterios de exclusión (EC):

- EC1: Los documentos detallan sobre agricultura inteligente pero no habla sobre sistemas de irrigación con IoT
- EC2: Los documentos hablan sobre tecnologías en la agricultura inteligente pero no especifican información relacionada sobre irrigación con IoT
- EC3: Los documentos no contienen resumen.
- EC4: Los documentos no describen información relevante sobre el tema de estudio.
- EC5: Los documentos tienen relación sobre IoT, pero está alejado a los sistemas de irrigación
- EC6: Los documentos no detallan las arquitecturas de IoT para la construcción de sistema de irrigación.
- EC7: Los documentos no detallan las técnicas de IoT que se utilizan para la construcción de aplicaciones de irrigación
- EC8: Los documentos no detallan los componentes que se necesitan para la construcción de aplicaciones de irrigación con IoT.
- EC9: Los documentos no detallan protocolos que se pueden utilizar para la construcción de aplicaciones de irrigación con IoT.

D. Cadenas de Búsquedas

Se definieron palabras claves a través del método PICOC [13] que permitieron construir la cadena de búsqueda con una sinonimia de cada palabra en conjunto con operadores lógicos, “AND” para los conceptos complementarios y “OR” para los documentos similares.

Se utilizó la misma cadena de búsqueda en las bases de datos seleccionadas.

IEEE Library, ScienceDirect, ACM Digital Library y Scopus:

- ("IoT") AND ("Irrigation system" OR "Automated Irrigation" OR "Intelligent Irrigation System" OR "Irrigation monitoring" OR "Precision irrigation" OR "Smart Farming" OR "Smart Irrigation" OR "Smart drip irrigation system") AND ("Architecture" OR "Method" OR "Methodology" OR "Protocol" OR "Standard")

E. Evaluación de Calidad

Definir la cadena de búsqueda, se procedió a la evaluación la calidad de los documentos seleccionados de las cuatro bases de datos mediante las siguientes preguntas:

- QA1: ¿El autor identifica o detalla algún problema dentro de la construcción de aplicaciones de irrigación con IoT?
- QA2: ¿Existe en el artículo citas sobre protocolos para la construcción de aplicaciones de irrigación con IoT?
- QA3: ¿El autor describe o detalla al menos una arquitectura para la construcción de aplicaciones de irrigación con IoT?
- QA4: ¿El autor detalla o al menos utiliza una técnica para la construcción de aplicaciones de irrigación con IoT?

III. RESULTADOS

Los resultados obtenidos de la RSL se describen en los siguientes pasos:

- 1) Se ejecutó la cadena de búsqueda en cada base de datos y se obtuvieron 819 artículos que representan la muestra del total.
- 2) Al subir los 819 documentos a la herramienta Parsifal, está detectó que existen 173 duplicados que representan el 21.12% del total, los mismos que fueron descartados, quedando 646 artículos por revisar
- 3) Los 646 artículos fueron revisados y analizados en su título y resumen en relación con la temática central, tomando en consideración los criterios de inclusión y exclusión. Del total se rechazaron 590 artículos (72.03%), que son irrelevantes para el objetivo de estudio. Además, hacen referencia a ejemplos de sistemas de irrigación

basadas en aplicaciones con inteligencia artificial, así como en proyectos de agricultura 4.0 con IoT, por lo tanto, su argumentación referente a los sistemas de irrigación con IoT es débil y no dan contestación a las preguntas de investigación planteadas. En la figura 1 se detalla el proceso de selección de los estudios, de los cuales quedaron 25 artículos finales para extraer información relevante sobre la temática.

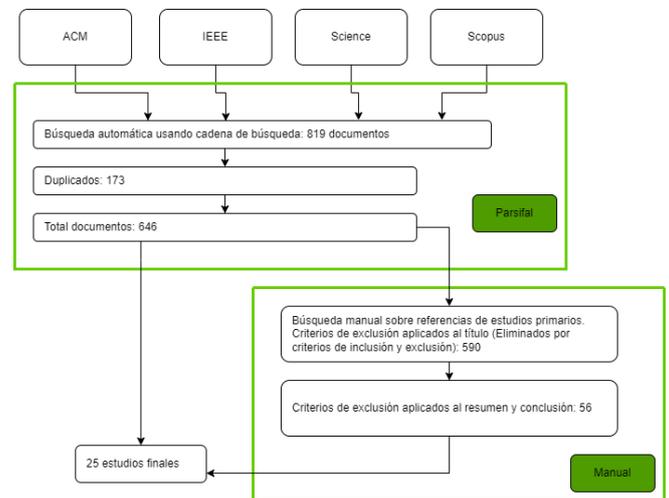


Figura 1: Pasos de búsqueda

4) Por último, cada una de las preguntas planteadas para el desarrollo del control de calidad tiene un puntaje de 1 si esta es calificada con “Si”, 0.5 si la respuesta es “Probablemente” y 0 si la respuesta fue evaluada con “No”. Cada artículo tuvo una calificación de 0 a 4 puntos. Si tiene un puntaje de igual o superior a 2, será seleccionado para extraer su información. Los que cumplieron con los criterios de calidad son 39 que representan (4.76%).

A. Informe del mapeo sistemático

MQ1: “¿Cuántos estudios se publicaron a lo largo de los años sobre los sistemas de irrigación con IoT?”.

En el lapso de tiempo entre 2016-2022 se han publicado 200 artículos. En la Figura 2 se ilustra el número de artículos por año, en el año 2021 tiene un porcentaje del 40% que corresponde a la mayoría de documentos extraídos generados por la cadena.



Figura 2: Publicaciones por año

En la Tabla I se presentan los 25 artículos seleccionados donde se detalla la información por título, autor, año de publicación y su fuente de publicación (congreso y revista), para identificar posibles escenarios de publicación.

Tabla I ARTÍCULOS EVALUADOS

Autor	Título	Año	Tipo Documento
Stolojescu-Crisan, C. and Butunoi, B. and Crisan, C.	An IoT Based Smart Irrigation System [14]	2021	Artículo
Imteaj, Ahmed and Rahman, Tanveer and Hossain, Muhammad Kamrul and Zaman, Saika,	IoT based autonomous percipient irrigation system using raspberry Pi[15]	2016	Artículo
E. Seethalakshmi and Meenakshi Shunmugam and R. Pavaiyarkarasi and Sneha Joseph and J. {Edward paulraj }	An automated irrigation system for optimized greenhouse using IoT[16]	2021	Artículo
Yousif, J.H. and Abdalgader, K.	Experimental and Mathematical Models for Real-Time Monitoring and Auto Watering Using IoT Architecture[17]	2022	Artículo
Ferrarezi, R.S. and Peng, T.W.	Smart System for Automated Irrigation Using Internet of Things Devices	2021	Artículo
Mohammed, M. and Riad, K. and Alqahtani, N.	Efficient iot-based control for a smart subsurface irrigation system to enhance irrigation management of date palm[18]	2021	Artículo
Roy, Sanku Kumar and Misra, Sudip and Raghuvanshi, Narendra Singh and Das, Sajal K.,	AgriSens: IoT-Based Dynamic Irrigation Scheduling System for Water Management of Irrigated Crops[19]	2021	Artículo
Mannan J. M. and Kanimozhi	Smart scheduling on cloud for IoT-based sprinkler	2021	Artículo

Suguna, S. and Dhivya, M. and Parameswaran, T.	irrigation[20]		
Borah, S. and Kumar, R. and Mukherjee, S.	Low-cost IoT framework for irrigation monitoring and control[21]	2021	Artículo
Shiva, S.J. and Palanivel, S. and Venkateswarlu, S.C.	IoT based Smart Irrigation System by using ESP32 and Adafruit IO[22]	2021	Conferencia
Hamdi, M. and Rehman, A. and Alghamdi, A. and Nizamani, M.A. and Missen, M.M.S. and Memon, M.A.	Internet of Things (IoT) Based Water Irrigation System[23]	2021	Artículo
Zulkifli, Nadiatulhuda and Iqbal, Farabi and Ramli, Arnidza and Idrus, Sevia M and Radzi, Raja Zahilah Raja Mohd and Ullah, Shafi and Hairon, Amirul Hakim and Ishak, Siti Hajar Asyikin and Mohommad, Munirah and Cruz, Jennifer Dela and Caya, Meo Vincent and Agulto, Ireneo	Evapotranspiration (ET)-Based Irrigation System with Internet of Things (IoT) Integration for Capsicum Annuum Farming: A Methodology[24]	2020	Artículo
Chew, Kim-Mey and Tan, Syvester Chiang-Wei and Loh, Gary Chee-Wyai and Bundan, Nancy and Yiiiong, Siew-Ping	IoT Soil Moisture Monitoring and Irrigation System Development[25]	2020	Artículo
Li, W. and Awais, M. and Ru, W. and Shi, W. and Ajmal, M. and Uddin, S. and	Review of Sensor Network-Based Irrigation Systems Using IoT and Remote Sensing[26]	2020	Revisión

Liu, C.			
López-Torres, S. and López-Torres, H. and Rocha-Rocha, J. and Aziz Butt, S. and Imran Tariq, M. and Collazos-Morales, C. and Piñeres-Espitia, G.	IoT Monitoring of Water Consumption for Irrigation Systems Using SEMMA Methodology[27]	2020	Conferencia
Rani, Deep and Kumar, Nagesh and Bhushan, Brij.	Implementation of an Automated Irrigation System for Agriculture Monitoring using IoT Communication[28]	2019	Artículo
Meher, Chandra Prakash and Sahoo, Adyasha and Sharma, Suraj.	IoT based Irrigation and Water Logging monitoring system using Arduino and Cloud Computing[29]	2019	Artículo
Kamienski, C. and Soininen, J.-P. and Taumberger, M. and Dantas, R. and Toscano, A. and Cinotti, T.S. and Maia, R.F. and Neto, A.T.	Smart water management platform: IoT-based precision irrigation for agriculture[30]	2019	Artículo
Guzmán-Toloza, J.M. and Villafaña-Gamboa, D.F. and Peniche-Ruiz, L.J. and Gómez-Buenfil, R.A. and Molina-Puc, J.K. and Rodríguez - Morayta, M.J.,	Internet of Things for Irrigation System[31]	2019	Conferencia
Amu, D. and Amuthan, A. and Gayathri, S. Shiva and Jayalakshmi, A.,	Automated Irrigation using Arduino sensor based on IOT[14]	2019	Artículo
Hovhannisyan, D. and Eltawil, A. and Kurdahi, F.	Testing topology adaptive irrigation IoT with circuits[32]	2019	Conferencia
Verma, A.K. and Bhagavan,	Wireless network based smart	2018	Artículo

K. and Akhil, V. and Singh, A.	irrigation system using IOT[33]		
Rajkumar, M.N. and Abinaya, S. and Kumar, V.V.	Intelligent irrigation system - An IOT based approach[34]	2017	Conferencia
Kranthi Kumar, M. and Srenivasa Ravi, K.,	Automation of irrigation system based on Wi-Fi technology and IOT[35]	2016	Artículo
Anbarasi M, Karthikeyan T, Ramanathan L, Ramani S, Nalini N,	Smart Multi-Crop Irrigation System Using IOT[36]	2019	Artículo

RQ1: ¿Qué componentes se necesitan para la construcción de aplicaciones de irrigación con IoT?

Como resultado se encontró que los componentes utilizados por los autores para el desarrollo de los sistemas de irrigación con IoT son los módulos FC-28 o YL-69 que permiten capturar la humedad del suelo, los sensores DHT-11 y DHT22 para capturar la temperatura ambiente y los sensores YF-S201 para calcular el caudal en litros del agua utilizada para el riego. Para la comunicación de los sensores y autores en su mayoría utilizan raspberry pi como servidor y los módulos Esp32 y Arduino para recibir los datos enviados por los sensores. Cabe destacar algunos autores han utilizado herramientas más sofisticadas como son estaciones meteorológicas profesionales para capturar los factores ambientales en tiempo real. Así mismo algunos autores implementaron sensores de PH para mejorar en gran medida los cálculos para regar los cultivos.

En la Tabla II se presenta una vista global de los componentes electrónicos utilizados por los autores para el desarrollo de sistemas de irrigación con IoT.

Tabla II COMPONENTES ELECTRONICOS

<i>Componentes</i>	<i>Tipo</i>	<i># de Artículo</i>	<i>Descripción</i>
YL-69 y Fc-28	Sensor	21	Sensor de humedad del suelo
MCU-Esp 8266, ESP-32	Placas programables		Placa programable con módulo Wifi y Bluetooth, para la comunicación

			con los sensores,
Arduino	Tarjeta Programable		Tarjeta Programable para la comunicación con los sensores,
Raspberry Pi	Microcomputador		Mini computadora utilizada como servidor conectado con una tarjeta programable para la comunicación con los sensores y el usuario final
Estación Meteorológica	Sensor	2	Sensor de temperatura para calcular la evapotranspiración de las plantas
Sensor de PH	Sensor		Sensor de PH del suelo
Relay, electroválvula	Elementos electrónicos		Actuadores para habilitar el riego
DTH11	Sensor		Sensor de temperatura ambiente

RQ2: ¿Qué arquitecturas de IoT son utilizadas para el desarrollo de sistemas de irrigación?

Como resultado se encontró que las arquitecturas para IoT describen una estructura de los componentes, organización y configuración funcional de los sistemas que se desean implementar, esto dependiendo del tipo de proyecto y las herramientas tecnológicas que se van a utilizar. Es por esto que se ha encontrado que la arquitectura de tres capas es la más utilizadas por los autores, porque esta se adapta a las necesidades de los sistemas de irrigación con IoT, dado que estos sistemas realizan operaciones de control de riego en tiempo real, de acuerdo a las necesidades de los factores ambientales, además que esta arquitectura le permite flexibilidad para que los protocolos de red puedan realizar varias operaciones en aplicaciones de red. Por otra parte, la arquitectura de 5 capas es utilizada por varios autores porque, esta arquitectura a diferencia de la de 3 capas les permite implementar tecnología más avanzadas para el

análisis y procesamiento de datos. Cabe destacar que la arquitectura de 5 capas ha sido utilizada por los autores para desarrollar sistemas de irrigación inteligentes o en sistemas que realizan análisis de datos para la toma de decisiones dentro de los cultivos. Por último, la arquitectura de 4 capas es utilizada por pocos autores, al igual que la arquitectura de 5 capas que utiliza herramientas tecnológicas para el análisis y procesamiento de datos obtenidos por los sensores.

En la Tabla III se presentan varios tipos de arquitecturas multicapas empleadas en el desarrollo de sistemas de irrigación con IoT. En donde la arquitectura más utilizada es la arquitectura 3 capas, seguido de la arquitectura de cinco capas y al final un autor presenta la utilización de la arquitectura de cuatro capas.

Tabla III TIPOS DE ARQUITECTURAS

Arquitectura	Capas	Artículos que lo utilizan	# de Artículos
Tres Capas	Capa de sensores, Capa de transmisión y Capa de centro de monitoreo	[15][16][35][19] [20][21][34][33] [25][26][37][22] [17][23][27][14] [31][29][36]	19
Cuatro Capas	Capa de sensores, Capa de puerta de enlace, Capa de servicio de administración, Capa de aplicación	[14]	1
Cinco Capas	Capa de detección Capa de acceso Capa de Red Capa de middleware	[20][13][26]	3

	Capa de aplicación		
--	--------------------	--	--

RQ3: ¿Qué protocolos de red son idóneos para la construcción de aplicaciones de irrigación?

Como resultado se encontró que muchos de los autores hacen uso del protocolo Zigbee para IoT, porque es un protocolo de bajo consumo y baja tasa de transferencia de mensajes, así como es utilizada en tamaños reducidos, permite realizar una comunicación mediante una topología en malla, permite la conexión de dispositivos dentro de una red. Este protocolo tiene una limitante necesita de un hub o puente para realizar la conexión con los diferentes dispositivos. Cabe destacar que la mayoría de autores han utilizado el protocolo Zigbee porque les permite tener una comunicación rápida con varios sensores mediante una dirección IP lo que permite que tengan varios dispositivos conectados dentro de una misma red creada por el mismo protocolo lo que evita saturar la red a la que está conectada el protocolo, pero para realizar esta acción los autores han utilizado un componente adicional que les permita utilizar el protocolo para realizar la comunicación. Por otro lado, MQTT es un protocolo de mensajería ligera de publicación y suscripción en entornos de bajo ancho de banda. Es bastante utilizado en diferentes proyectos de IoT, ya que permite conectarse o comunicarse con distintos sensores, está basando en tres capas denominadas publicador, suscriptor y bróker. Este protocolo trabaja sobre Tcp y provee calidad de servicio en 3 niveles, para la comunicación de los dispositivos. Es preciso argumentar que el protocolo MQTT es el segundo protocolo más utilizado por los autores de los artículos seleccionados, ya que este permite realizar comunicaciones seguras dentro de los proyectos de irrigación con IoT, además que permite implementar dicho protocolo con herramientas open source y es libre configuración, lo que permite tener más libertad a la hora de realizar la configuración de varios sensores.

En la figura 3 se presentan los diferentes protocolos utilizados en el desarrollo de sistemas de irrigación con IoT. En donde la mayoría de los autores han utilizado el protocolo ZigBee, seguido del protocolo MQTT que en su mayoría los autores también lo han utilizado.

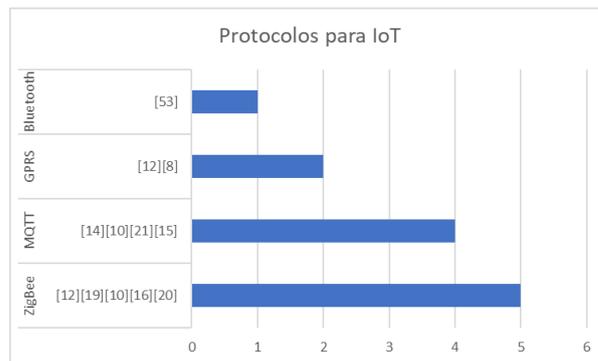


Figura 3: Protocolos para IoT

RQ4: ¿Qué problemas se presentan en la construcción de aplicaciones de irrigación con IoT?

Dentro de los problemas señalados por los autores, tienen que ver con la actualización de dispositivos ya que si uno de esta falla, no se los podría arreglar, dado que estos no cuentan con una protección adicional que eviten deterioro de los mismos. Por lo que los autores recomiendan crear carcasas o recubrimientos para que estos dispositivos tengan una vida útil más prolongada. Otro de los problemas detallados tiene que ver con el tiempo de implementación ya que se necesita realizar estudios y calibración de los sensores para determinar los valores con los que se va a trabajar. Por último, un problema que va ya a entornos reales es la seguridad de la información, ya que si esta es corrompida puede afectar la calidad de los cultivos. Por lo cual recomiendan cambiar configuración por defecto.

En la figura 4 se presenta una gráfica con los problemas o limitaciones encontradas al momento de desarrollar un sistema de irrigación con IoT.

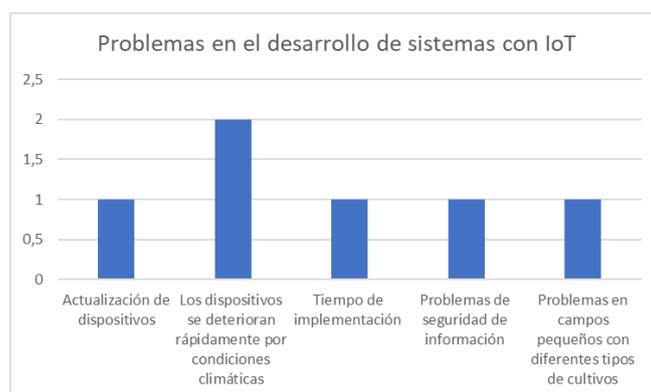


Figura 4: Problemas que se presentan en la Construcción

IV. DISCUSIÓN

En los resultados obtenidos en el RSL se pudo apreciar que la arquitectura de tres capas es utilizada para el desarrollo de sistemas irrigación con IoT según [11] [12] [31] [15] [16] [17] [30] [29] [21] [22] [33] [18] [13] [19] [23] [10] [27] [25] [32], que son el 82% de los artículos utilizados para la revisión. En donde los autores hacen uso

de esta arquitectura, porque está se adapta a las necesidades de los sistemas de irrigación con IoT, dado que estos sistemas realizan operaciones de control de riego en tiempo real, de acuerdo a las necesidades de los factores ambientales, además que esta arquitectura le permite flexibilidad para que los protocolos de red puedan realizar varias operaciones en aplicaciones de red. Por otra parte, las arquitecturas de cuatro y cinco capas son utilizadas en un 18% de los artículos revisados, en donde, los autores [20] [13] [14] [26] prefieren estas arquitecturas porque a diferencia de la de 3 capas les permite implementar tecnología más avanzadas para el análisis y procesamiento de datos. Cabe destacar que la arquitectura de 4 y 5 capas han sido utilizadas por los autores para desarrollar sistemas de irrigación inteligentes o en sistemas que realizan análisis de datos para la toma de decisiones dentro de los cultivos.

La investigación también ayudo a determinar que el protocolo MQTT, es idóneo para el desarrollo de sistemas con IoT, esto porque el protocolo trabaja sobre TCP y presta 3 niveles de calidad de servicio, así mismo, este cuenta con una arquitectura de tres capas denominadas bróker, suscriptor y publicador, que permiten la comunicación de los dispositivos (sensores o actuadores) con el usuario, esto coincide con los artículos [17] [13] [24] [18], en donde los autores prefieren el uso de este protocolo porque además de los servicios que presenta, se adapta a las arquitectura de tres capas, brindando además confiabilidad en la entrega de paquetes dentro de redes con bajos recursos. Por otra parte, los autores [15] [22] [13] [19] [23], prefieren la utilización del protocolo Zigbee, que es otro protocolo utilizado en proyectos IoT. Este protocolo es de bajo consumo y baja tasa de transferencia de mensajes; es utilizado en redes de tamaños reducidos, permitiendo realizar una comunicación mediante una topología en malla. Este protocolo es eficiente, pero tiene una limitante, y es que necesita de un hub o puente para realizar la conexión con los diferentes dispositivos y salir hacia internet.

Dentro de los resultados, los problemas que se presentan en el desarrollo de sistemas de irrigación con IoT en su mayoría tienen que ver con los componentes utilizados ya que son de bajo costo, lo cual tienden a deteriorarse con rapidez y tienen que ser reemplazados, esto coincide con los autores [14] [34] [15] que describen que los sistemas de riego basados en IoT, no son sistemas independientes y simples, estos se pueden administrar, mejorar y probar fácilmente durante la práctica, pero requieren validación, pruebas de integración y verificación antes de la implementación. Los dispositivos en red deben operar sin supervisión, por lo tanto, deben ser prioritariamente y dinámicamente configurables para adaptarse a las necesidades funcionales del sistema. Sin embargo, no siempre es así y las fallas de los dispositivos en muchos casos solo pueden resolverse mediante el reemplazo, para poder identificar y localizar ese dispositivo es un desafío para distinguir en un campo de muchos dispositivos, a menos que el resto de los dispositivos puedan ubicar e insinuar el nodo supervisor de las fallas. Por otra parte, otro problema encontrado tiene que ver el tiempo de implementación esto porque en [28] presenta que tuvieron problemas al implementar el sistema de irrigación en un campo pequeño, que tenía varios tipos de cultivos en el mismo, lo que dificulta tener un tipo de riego diferente y un espacio tan pequeño. Otro de los problemas encontrados tiene que ver con la seguridad ya que si se altera la información puede tener consecuencias

graves en los diferentes cultivos. En [20] identifica que pueden existir problemas de seguridad de datos en los sistemas de irrigación lo que puede causar caos en los sistemas ya que terceros podrían manipular los datos a su conveniencia.

Las herramientas o componentes utilizados para el desarrollo de sistemas de irrigación con IoT, son dispositivos de bajo costo pero que cumplen con las funcionalidades requeridas en cada sistema es por ello que los autores [11] [12] [14] [31] [15] [16] [30] [17] [29] [20] [13] [21] [22] [10] [34] [18] [35] [10] [26] [25] [32] presentan que los dispositivos electrónicos para la creación de sistemas de irrigación con IoT son: Sensor de humedad FC-28 y YL-69 necesarios para detectar la humedad relativa del suelo enviando señales digitales o analógicas, de acuerdo a la programación y a los requerimientos del usuario, para evaluar cuándo abrir la electroválvula para regar los cultivos. MCU-Esp32, MCU Esp8266 y Arduino, estos componentes son usados de manera similar, aunque en su arquitectura cambian algunos componentes, pero la programación se la puede realizar mediante “Arduino IDE”, cabe destacar que en la mayoría de artículos son más utilizados los módulos MCU-Esp32, MCU Esp8266 ya que tiene integrado un módulo de wifi necesario para la conexión inalámbrica en una red LAN, en cambio con Arduino es necesario utilizar un módulo wifi aparte para la comunicación. La microcomputadora Raspberry Pi es utilizada

En conclusión se puede utilizar la arquitectura de tres capas (capa de sensores, red, aplicación) para el desarrollo de un sistema de irrigación con IoT, por su fácil implementación y entendimiento, esto con la utilización del protocolo MQTT que trabaja sobre TCP brindando calidad de servicios en tres niveles, permitiendo el envío y recepción de paquetes de manera fiable, así mismo los componentes para el desarrollo del sistema son de bajo costo y se los puede adquirir en la localidad sin problema alguno, estos son: esp32, sensor de humedad de suelo fc-28, sensor de caudal YF-S201, raspberry pi como servidor. Los mismo que serán utilizados para el desarrollo del prototipo de sistema de irrigación con IoT para la Universidad Nacional de Loja.

V. CONCLUSIONES

Se seleccionaron 25 artículos y los resultados muestran que existen varios tipos de arquitecturas para la implementación de sistemas de irrigación con IoT, pero entre las más utilizadas se encuentra la arquitectura de tres capas que tuvo mayor impacto dentro de los investigadores por se compone: i) Capa de sensores, ii) capa de servidores o capa de transmisión de datos, iii) La capa de aplicación o centro de monitoreo. La revisión muestra que hay un gran número de estudios relacionados con los sistemas de irrigación con IoT, donde utilizan los protocolos MQTT y ZigBee para la comunicación. Así mismo utilizan diferentes herramientas electrónicas para la extracción de datos del medio. Los problemas que detallan los autores van relacionados con la falla de los dispositivos electrónicos que para mitigar errores de estos se los tiene que cambiar. Por último, los sistemas de irrigación con IoT ayudan a varios aspectos como es el optimizar el uso de agua en los sembríos, automatiza el proceso de riego y realiza cálculos de acuerdo a los factores ambientales de cuanta cantidad de agua necesitan los cultivos.

VI. REFERENCIAS BIBLIOGRÁFICA

- [1] Diana Mata Hernandez, "Convertidor Analógico Digital Pipeline," Mar. 2015, Accessed: Jun. 03, 2015. [Online]. Available: <https://hdl.handle.net/20.500.12371/8257>
- [2] J. Ignacio, "Conversores Análogo-Digital y Digital-Análogo: Conceptos Básicos," 2012.
- [3] R. Saito, C. Rios, J. Lluís, and A. Ferrer, "UNIVERSIDAD POLITECNICA DE CATALUNYA ESTUDIO E IMPLEMENTACIÓN DE UN CONVERTIDOR ANALÓGICO DIGITAL Y DIGITAL ANALÓGICO SIGMA DELTA," 2015. Accessed: Jun. 04, 2015. [Online]. Available: <http://hdl.handle.net/2117/89044>
- [4] B. Gimeno and E. Tutores, "Diseño de un Convertidor Analógico-Digital de Aproximaciones Sucesivas de bajo consumo y área reducida Proyecto Fin de Carrera," 2017.
- [5] A. Glória, P. Sebastião, C. Dionísio, G. Simões, and J. Cardoso, "Water management for sustainable irrigation systems using internet-of-things," *Sensors (Switzerland)*, vol. 20, no. 5, Mar. 2020, doi: 10.3390/s20051402.
- [6] J. Ignacio et al., "Construcción De Un Prototipo De Un Sistema De Riego Y Ventilación De Un Invernadero Bajo El Concepto De Iot Construction Of A Prototype Of An Irrigation And Ventilation System Of A Greenhouse Under The Concept Of Iot," *Tecnológico Nacional de México en Celaya Pistas Educativas*, vol. 2020, no. 137. [Online]. Available: <http://itcelaya.edu.mx/ojs/index.php/pistas>
- [7] V. Cortes, C. Marco, and F. V. Garcia, "Diseño E Implementación De Un Sistema De Riego Automatizado Y Monitoreo De Variables Ambientales Mediante Iot En Los Cultivos Urbanos De La Fundación Mujeres Empresarias Marie Poussepin."
- [8] I. G. C. Alexander, "Trabajo de titulación modalidad Proyecto Integrador, previo a la obtención del título de Ingeniero en Computación Gráfica".
- [9] L. García, L. Parra, J. M. Jimenez, J. Lloret, and P. Lorenz, "IoT-based smart irrigation systems: An overview on the recent trends on sensors and iot systems for irrigation in precision agriculture," *Sensors (Switzerland)*, vol. 20, no. 4. MDPI AG, Feb. 02, 2020. doi: 10.3390/s20041042.
- [10] J. Camacho, K. Campos, P. Cedillo, B. Coronel, and A. Bermeo, "Forensics Analysis on Mobile Devices: A Systematic Mapping Study," in *Advances in Intelligent Systems and Computing*, vol. 884, Springer Verlag, 2019, pp. 57–72. doi: 10.1007/978-3-030-02828-2_5.
- [11] J. Medeiros, M. Goulao, A. Vasconcelos, and C. Silva, "Towards a Model about Quality of Software Requirements Specification in Agile Projects," in 2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC), Sep. 2016, pp. 236–241. doi: 10.1109/QUATIC.2016.058.
- [12] F. W. Neiva, J. M. N. David, R. Braga, and F. Campos, "Towards pragmatic interoperability to support collaboration: A systematic review and mapping of the literature," *Inf. Softw. Technol.*, vol. 72, pp. 137–150, Apr. 2016, doi: 10.1016/j.infsof.2015.12.013.
- [13] J. Popay, L. Arai, and N. Britten, "Guidance on the conduct of narrative synthesis in systematic reviews: A product from the ESRC Methods Programme," 2006, doi: 10.13140/2.1.1018.4643.
- [14] A. Ahmed and S. Rawal, "IOT based Smart Irrigation System Cite this paper IOT based Smart Irrigation System," *Int. J. Comput. Appl.*, vol. 159, no. 8, pp. 975–8887, 2017.
- [15] A. Imteaj, T. Rahman, M. K. Hossain, and S. Zaman, "IoT based autonomous percipient irrigation system using raspberry Pi," in 2016 19th International Conference on Computer and Information Technology (ICIT), Dec. 2016, pp. 563–568. doi: 10.1109/ICCITECHN.2016.7860260.
- [16] E. Seethalakshmi, M. Shunmugam, R. Pavaiyarkarasi, S. Joseph, and J. Edward paulraj, "An automated irrigation system for optimized greenhouse using IoT," *Mater. Today Proc.*, no. xxxx, Feb. 2021, doi: 10.1016/j.matpr.2020.12.636.
- [17] J. H. Yousif and K. Abdalgader, "Experimental and Mathematical Models for Real-Time Monitoring and Auto Watering Using IoT Architecture," *Computers*, vol. 11, no. 1, 2022, doi: 10.3390/computers11010007.
- [18] M. Mohammed, K. Riad, and N. Alqahtani, "Efficient IoT-Based Control for a Smart Subsurface Irrigation System to Enhance Irrigation Management of Date Palm.," *Sensors (Basel)*, vol. 21, no. 12, p. 3942, Jun. 2021, doi: 10.3390/s21123942.
- [19] S. K. Roy, S. Misra, N. S. Raghuvanshi, and S. K. Das, "AgriSens: IoT-Based Dynamic Irrigation Scheduling System for Water Management of Irrigated Crops," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 5023–5030, Mar. 2021, doi: 10.1109/JIOT.2020.3036126.
- [20] M. Mannan J., K. S. S., D. M., and P. T., "Smart scheduling on cloud for IoT-based sprinkler irrigation," *Int. J. Pervasive Comput. Commun.*, vol. 17, no. 1, pp. 3–19, Feb. 2021, doi: 10.1108/IJPC-03-2020-0013.
- [21] S. Borah, R. Kumar, and S. Mukherjee, "Low-cost IoT framework for irrigation monitoring and control," *Int. J. Intell. Unmanned Syst.*, vol. 9, no. 1, pp. 63–79, Jun. 2020, doi: 10.1108/IJIUS-12-2019-0075.
- [22] S. J. Shiva, S. Palanivel, and S. C. Venkateswarlu, "IoT based Smart Irrigation System by using ESP32 and Adafruit IO," *CEUR Workshop Proc.*, vol. 3058, pp. 0–2, 2021.
- [23] M. Hamdi, A. Rehman, A. Alghamdi, M. A. Nizamani, M. M. S. Missen, and M. A. Memon, "Internet of Things (IoT) Based Water Irrigation System," *Int. J. Online Biomed. Eng.*, vol. 17, no. 05, p. 69, May 2021, doi: 10.3991/ijoe.v17i05.22081.
- [24] N. Zulkifli et al., "Evapotranspiration (ET)-Based Irrigation System with Internet of Things (IoT) Integration for Capsicum Annuum Farming: A Methodology," in 2020 IEEE 5th International Symposium on Telecommunication Technologies (ISTT), Nov. 2020, pp. 88–93. doi: 10.1109/ISTT50966.2020.9279371.
- [25] K.-M. Chew, S. C.-W. Tan, G. C.-W. Loh, N. Bundan, and S.-P. Yiiiong, "IoT Soil Moisture Monitoring and Irrigation System Development," in *Proceedings of the 2020 9th International Conference on Software and Computer Applications*, Feb. 2020, pp. 247–252. doi: 10.1145/3384544.3384595.
- [26] W. Li et al., "Review of Sensor Network-Based Irrigation Systems Using IoT and Remote

Sensing,” *Adv. Meteorol.*, vol. 2020, pp. 1–14, Sep. 2020, doi: 10.1155/2020/8396164.

[27] S. López-Torres et al., “IoT Monitoring of Water Consumption for Irrigation Systems Using SEMMA Methodology,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11886 LNCS, 2020, pp. 222–234. doi: 10.1007/978-3-030-44689-5_20.

[28] D. Rani, N. Kumar, and B. Bhushan, “Implementation of an Automated Irrigation System for Agriculture Monitoring using IoT Communication,” in *2019 5th International Conference on Signal Processing, Computing and Control (ISPCC)*, Oct. 2019, vol. 2019-October, pp. 138–143. doi: 10.1109/ISPCC48220.2019.8988390.

[29] C. P. Meher, A. Sahoo, and S. Sharma, “IoT based Irrigation and Water Logging monitoring system using Arduino and Cloud Computing,” in *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, Mar. 2019, pp. 1–5. doi: 10.1109/ViTECoN.2019.8899396.

[30] C. Kamienski et al., “Smart Water Management Platform: IoT-Based Precision Irrigation for Agriculture,” *Sensors*, vol. 19, no. 2, p. 276, Jan. 2019, doi: 10.3390/s19020276.

[31] J. A. B. Telematics and Computing, vol. 1053. Cham: Springer International Publishing, 2019. doi: 10.1007/978-3-030-33229-7.

[32] D. Hovhannisyanyan, A. Eltawil, and F. Kurdahi, “Testing topology adaptive irrigation IoT with circuits,” *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2019-May, 2019, doi: 10.1109/ISCAS.2019.8702559.

[33] A. Kumar Verma, B. K. V. Akhil, and A. Singh, “Wireless network based smart irrigation system using IOT,” *Int. J. Eng. Technol.*, vol. 7, no. 1.1, p. 342, Dec. 2017, doi: 10.14419/ijet.v7i1.1.9849.

[34] M. N. Rajkumar, S. Abinaya, and V. V. Kumar, “Intelligent irrigation system — An IOT based approach,” in *2017 International Conference on Innovations in Green Energy and Healthcare Technologies (IGEHT)*, Mar. 2017, pp. 1–5. doi: 10.1109/IGEHT.2017.8094057.

[35] M. Kranthi Kumar and K. Srenivasa Ravi, “Automation of Irrigation System based on Wi-Fi Technology and IOT,” *Indian J. Sci. Technol.*, vol. 9, no. 17, May 2016, doi: 10.17485/ijst/2016/v9i17/93048.

[36] A. Kumar, P. Ranjan, and V. Saini, “Smart Irrigation System Using IoT,” in *Advanced Series in Management*, vol. 27, no. July, 2022, pp. 123–139. doi: 10.1108/S1877-636120220000027009.

[37] D. Amu, A. Amuthan, S. S. Gayathri, and A. Jayalakshmi, “Automated Irrigation using Arduino sensor based on IOT,” in *2019 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2019, pp. 1–6. doi: 10.1109/ICCCI.2019.8822147.

Anexo 2 Manual RSL

FACULTAD DE ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS

NATURALES NO RENOVABLES

CARRERA DE INGENIERÍA EN SISTEMAS / COMPUTACIÓN

*MANUAL RSL PARA “EL SISTEMA DE IRRIGACIÓN
CON IoT PARA CULTIVOS DE LA UNIVERSIDAD
NACIONAL DE LOJA”*

Estudiantes:

- Alex Santiago Nole Reyes
- Angel Favian Minga Medina

El presente manual contiene información del proceso realizado en el RSL, mediante el uso de la aplicación web “Parsifal”

1. Review

a. Registro de Título del proyecto

Primeramente, se registra el título del proyecto para iniciar la investigación, en la figura 1, se presenta una ilustración del título registrado para el RSL.

Review details

Title

Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja

Description

Figura 1 Titulo del Proyecto RSL

b. Integrantes de la Investigación

Para ingresar los integrantes que formarán parte de la investigación se los agrega, dando click en el botón “add 130utor”, mismo que despliega un formulario para agregar el correo electrónico del integrante. En la figura 2 se presentan los usuarios registrados para la presente investigación.

Authors

Alex Nole	main author
Angel Minga	✖
René Guamán Quinche	✖

+ Add author

Figura 2 Autores del RSL

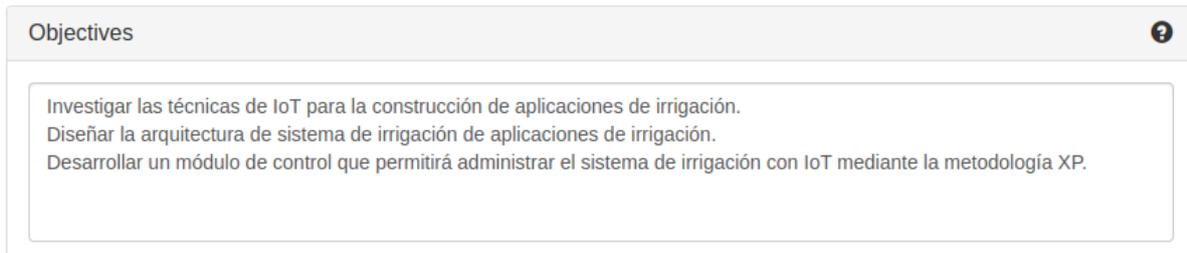
2. Planning

En la sección de planning se ingresan los criterios relevantes para el proceso de selección de los documentos idóneos para la investigación, así como las fuentes de donde serán extraídos.

a. Protocolo

i. Objetivos.

Se ingresan los objetivos a los que se pretende llegar con la presente investigación, en la figura 3 se presentan los objetivos que ayudan al proceso de la investigación.



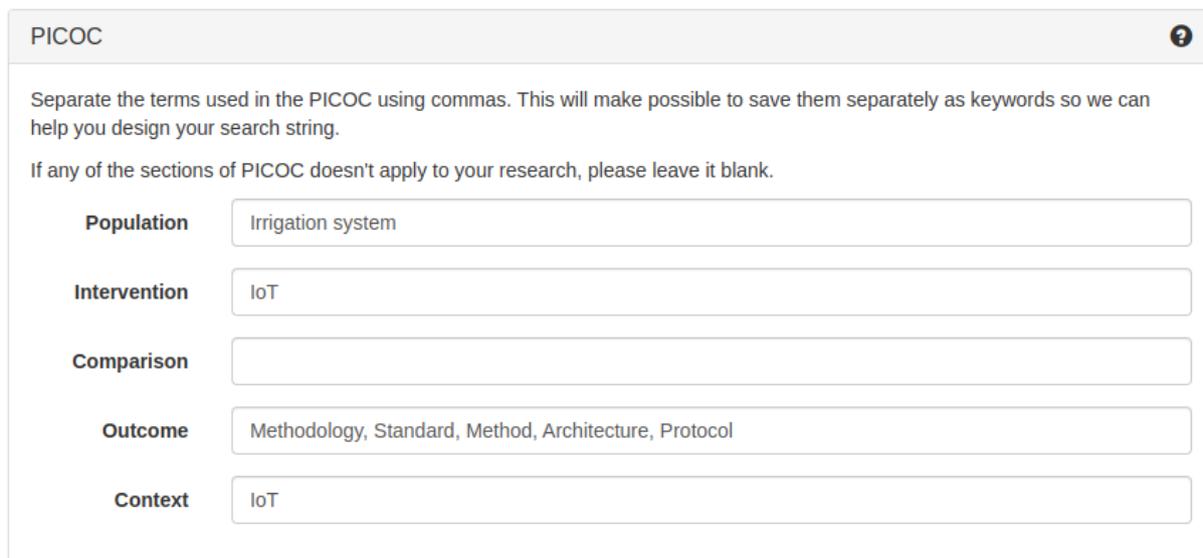
Objectives

Investigar las técnicas de IoT para la construcción de aplicaciones de irrigación.
Diseñar la arquitectura de sistema de irrigación de aplicaciones de irrigación.
Desarrollar un módulo de control que permitirá administrar el sistema de irrigación con IoT mediante la metodología XP.

Figura 3 Objetivos de la investigación

b. PICOC

El desarrollo del PICOC (Population, Intervention, Comparison, Outcome, Context), permite ingresar las palabras claves, mismas que permiten construir la cadena de búsqueda para obtener los documentos de diferentes fuentes necesarios para la investigación. En la figura 4 se presentan las palabras ingresadas al PICOC para el desarrollo del RSL.



PICOC

Separate the terms used in the PICOC using commas. This will make possible to save them separately as keywords so we can help you design your search string.

If any of the sections of PICOC doesn't apply to your research, please leave it blank.

Population	Irrigation system
Intervention	IoT
Comparison	
Outcome	Methodology, Standard, Method, Architecture, Protocol
Context	IoT

Figura 4 Palabras ingresadas al PICOC

c. Preguntas de Investigación

Las preguntas de investigación, son la parte fundamental de la investigación, estas deben ser bien desarrolladas para encaminarse a lo que se quiere llegar a resolver con el proceso de la RSL. En la figura 5 se presentan las preguntas de investigación planteadas para el desarrollo de la RSL.

Research Questions			
^ v	RQ1: ¿Qué componentes se necesitan para la construcción de aplicaciones de irrigación con IoT?	edit	remove
^ v	RQ2: ¿Qué arquitecturas de IoT son utilizadas para el desarrollo de sistemas de irrigación?	edit	remove
^ v	RQ3: ¿Qué protocolos de red son idóneos para la construcción de aplicaciones de irrigación?	edit	remove
^ v	RQ4: ¿Qué problemas se presentan en la construcción de aplicaciones de irrigación con IoT?	edit	remove
^ v	MQ1: ¿Cuántos estudios se publicaron a lo largo de los años sobre los sistemas de irrigación con IoT?	edit	remove
+ Add Question			

Figura 5 Preguntas de Investigación

d. Palabras claves y sinónimos

Las palabras claves y sinónimos para el desarrollo de la investigación se las ilustra en la figura 6.

Keywords and Synonyms				
To edit or remove a certain keyword or synonym you may click on its description to enable the field.				
Keyword	Synonyms	Related to		
Architecture		Outcome	edit	remove
IoT		Population	edit	remove
Irrigation system	Automated Irrigation Intelligent Irrigation System Irrigation monitoring Precision irrigation Smart Farming Smart Irrigation Smart drip irrigation system	Intervention	edit	remove
Method		Outcome	edit	remove
Methodology		Outcome	edit	remove
Protocol		Outcome	edit	remove
Standard		Outcome	edit	remove

Figura 6 Palabras claves y sinónimos

e. Cadena de Búsqueda

La cadena de búsqueda utilizada en las diferentes bases de datos científicas, se la generó en base al PICOC y las palabras claves antes descritas, en la figura 7, se presenta la cadena de búsqueda utilizada.

Search String ?

i Use uppercase for boolean operators (**AND, OR**), double quotes for composite words and parentheses to logically separate the keywords and synonyms.

("IoT") AND ("Irrigation system" OR "Automated Irrigation" OR "Intelligent Irrigation System" OR "Irrigation monitoring" OR "Precision irrigation" OR "Smart Farming" OR "Smart Irrigation" OR "Smart drip irrigation system") AND ("Architecture" OR "Method" OR "Methodology" OR "Protocol" OR "Standard")

Figura 7 Cadena de Búsqueda para la ingresar a las fuentes

f. Fuentes

Las fuentes bibliográficas utilizadas para obtener los diferentes artículos, conferencias, revistas, son IEEE, ACM Digital Library, Science@Direct y Scopus, mismas que se las puede apreciar en la figura 8.

Sources		
Name	URL	
ACM Digital Library	http://portal.acm.org	✎ edit 🗑 remove
IEEE Digital Library	http://ieeexplore.ieee.org	✎ edit 🗑 remove
Science@Direct	http://www.sciencedirect.com	✎ edit 🗑 remove
Scopus	http://www.scopus.com	✎ edit 🗑 remove

+ Add Source
Add a Digital Library

Figura 8 Fuentes bibliográficas

g. Criterios de selección

Los criterios de inclusión y exclusión, son importantes en el desarrollo del RSL, ya que permite seleccionar qué documentos son relevantes al tema de estudio y cuales causan ruido a la investigación, en la figura 9, se presentan los criterios de inclusión y exclusión utilizados para el desarrollo de la RSL.

Selection Criteria

Inform your inclusion or exclusion criteria and press **Enter** to add.

Inclusion Criteria

Los Documentos deben estar escritos en ingles

Los Documentos detallan las técnicas de IoT que se se pu

Los documentos deben contener información relacionada

Los documentos detallan las arquitecturas de IoT para la c

Los documentos detallan los componentes que se necesiti

Los documentos detallan protocolos que se pueden utiliza

🗑 remove selected

Exclusion Criteria

Los Documentos hablan sobre agricultura pero no especifi

Los Documentos hablan sobre agricultura inteligente pero

Los Documentos hablan sobre tecnologías en la agricultur

Los Documentos no contienen resumen

Los Documentos no describen información relevante sobre

Los Documentos tienen relación sobre IoT pero esta alejac

🗑 remove selected

Figura 9 Criterios de inclusión y exclusión

b. Lista de verificación de evaluación de calidad

i. Preguntas

Las preguntas de calidad, se utilizan para filtrar entre los documentos, los cuales cumplen los criterios planteados en la investigación, en la figura 10 se presentan las preguntas de calidad utilizadas en la RSL.

Questions

^	v	QA:1¿El autor identifica o detalla algún problema dentro de la construcción de aplicaciones de irrigación con IoT?	edit	remove
^	v	QA:2¿Existe en el artículo citas sobre protocolos para la construcción de aplicaciones de irrigación con IoT?	edit	remove
^	v	QA:3¿El autor describe o detalla al menos una arquitectura para la construcción de aplicaciones de irrigación con IoT?	edit	remove
^	v	QA:4 ¿El autor detalla o al menos utiliza una técnica para la construcción de aplicaciones de irrigación con IoT?	edit	remove

+ Add Question

Figura 10 Preguntas de calidad

ii. Respuestas

La sección de respuestas de las preguntas de calidad, constan de tres, las mismas que son: si para los documentos que contienen la información relevante para la investigación, en parte para documentos que cuentas con algunos rasgos importantes, y no para los documentos que no aportan al desarrollo de la investigación, en la figura 11, se presentan las respuestas y la puntuación que contendrá cada respuesta.

Answers		
Description	Weight	
Si	1.0	 edit  remove
Probablemente	0.5	 edit  remove
No	0.0	 edit  remove

[+ Add Answer](#)

Figura 11 Respuestas para las preguntas de investigación

iii. Puntuaciones de evaluación de calidad

La puntuación de evaluación de calidad es el valor máximo que tiene que contener un documento para ser seleccionado extraer información sobre las preguntas de investigación, en la figura 12 se presentan las puntuaciones para seleccionar los documentos.

Quality Assessment Scores		
Max Score	<input type="text" value="4.0"/>	Calculated based on the number of questions and on the answer of greater weight
Cutoff Score	<input type="text" value="1.9"/>	 save

Figura 12 Calificación para las preguntas de calidad

c. Formulario de extracción de datos

El formulario de extracción de datos, son los criterios que se va a extraer de la investigación, en la figura 13 se presentan los criterios que se desean obtener en el desarrollo de la RSL.

Data Extraction Form			
Description	Type	Values	
  Protocolos	String Field	n/a	 edit  remove
  Arquitecturas	String Field	n/a	 edit  remove
  Metodología	String Field	n/a	 edit  remove
  Problemas en la construcción de aplicaciones de irrigación	String Field	n/a	 edit  remove
  Herramientas	String Field	n/a	 edit  remove

[+ Add Field](#)

Figura 13 Formulario de extracción de datos

3. Conducting

En la sección de “conducting”, se realiza el proceso de selección de los documentos que son importantes para extraer información sobre las preguntas de investigación.

a. Cadena de Búsqueda

En la figura 13 se presenta la cadena de búsqueda utilizada para la extracción de documentos, de las diferentes fuentes bibliográficas.

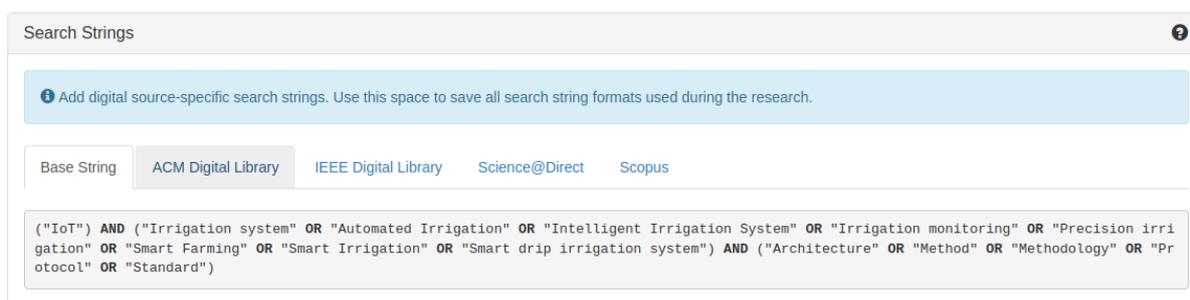


Figura 14 Cadena de búsqueda y las diferentes fuentes

b. Estudios de Importación

Los estudios de importación, son los documentos encontrados en cada fuente bibliográfica, relacionada con el tema de investigación. Para el desarrollo del presente proyecto se encontraron que en Scopus se encontraron 578 documentos, en IEEE 168, en Scencedirect 22 y en ACM 51, como se puede apreciar en la figura 14.

Import Studies		
Source	Imported Studies	
ACM Digital Library	51	
IEEE Digital Library	168	
Science@Direct	22	
Scopus	578	

Figura 15 Importación de estudios de diferentes fuentes

c. Selección de estudios

En la selección de estudios se filtraron primeramente los duplicados, esto por medio de la aplicación Parsifal que lo realiza en automático para lo cual se encontró que existen 173 documentos duplicados, como se puede ver en la figura 15, se presentan los documentos seleccionados por la aplicación como duplicados.

Action: 0 of 173 selected

Show: All Accepted Rejected Unclassified Duplicated

<input type="checkbox"/>	Bibtex Key	Title	Author	Journal	Year	Added by	Added at	Status
<input type="checkbox"/>		IoT enabled smart farming and irrigation system	Rohith, M. and Sainivedhana, R. and Sabiyath Fatima, N.	Proceedings - 5th International Conference on Intelligent Computing and Control Systems, ICICCS 2021	2021	Alex Nole	05 May 2022 17:11:27	Duplicated
<input type="checkbox"/>		Agriculture Improvement Using Iot	Lavanya, E. and Indra, E. and Priyadarshikadevi, T.	2021 International Conference on System, Computation, Automation and Networking, ICSCAN 2021	2021	Alex Nole	05 May 2022 17:11:27	Duplicated
<input type="checkbox"/>		IoT-Agro: A smart farming system to Colombian coffee farms	Rodríguez, J.P. and Montoya-Munoz, A.I. and Rodríguez-Pabon, C. and Hoyos, J. and Corrales, J.C.	Computers and Electronics in Agriculture	2021	Alex Nole	05 May 2022 17:11:27	Duplicated
<input type="checkbox"/>		Smart collective irrigation: Agent and internet of things based system	Ikididi, A. and Fazziki, A.E. and Sadgal, M.	ACM International Conference Proceeding Series	2021	Alex Nole	05 May 2022 17:11:27	Duplicated
<input type="checkbox"/>		Intelligent IoT-multiagent precision irrigation approach for improving water use efficiency in irrigation systems at farm and district scales	Jiménez, A.-F. and Cárdenas, P.-F. and Jiménez, F.	Computers and Electronics in Agriculture	2022	Alex Nole	05 May 2022 17:11:27	Duplicated

Figura 16 Documentos duplicados

Luego de seleccionar los documentos duplicados, se procedió a la selección de los documentos que contienen información relevante sobre la temática de estudio, mediante los criterios de inclusión y exclusión. Esto basado en la lectura de los títulos de los artículos seleccionados. Cabe destacar que se seleccionaron 42 documentos para la siguiente selección, en la figura 16 se presentan algunos de los documentos aceptados en el desarrollo de la RSL.

Action: 0 of 42 selected

Show: All Accepted Rejected Unclassified Duplicated

<input type="checkbox"/>	Bibtex Key	Title	Author	Journal	Year	Added by	Added at	Status
<input type="checkbox"/>		Smart multi-crop irrigation system using IOT	Anbarasi, M. and Karthikeyan, T. and Ramanathan, L. and Ramani, S. and Nalini, N.	International Journal of Innovative Technology and Exploring Engineering	2019	Alex Nole	05 May 2022 17:11:28	Accepted
<input type="checkbox"/>		Smart Farming using IoT, a solution for optimally monitoring farming conditions	Jash Doshi and Tirthkumar Patel and Santosh kumar Bharti	Procedia Computer Science	2019	Alex Nole	02 May 2022 15:06:45	Accepted
<input type="checkbox"/>		A new Kappa Architecture for IoT Data Management in Smart Farming	Jean Bertin (Nkamla Penka) and Said Mahmoudi and Olivier Debauche	Procedia Computer Science	2021	Alex Nole	02 May 2022 15:06:45	Accepted
<input type="checkbox"/>		IoT Monitoring of Water Consumption for Irrigation Systems Using SEMMA Methodology	López-Torres, S. and López-Torres, H. and Rocha-Rocha, J. and Aziz Butt, S. and Imran Tariq, M. and Collazos-Morales, C. and Piñeres-Espitia, G.	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	2020	Alex Nole	05 May 2022 17:11:28	Accepted
<input type="checkbox"/>		IoT Powered Portable Aquaponics System	Jacob, Nikhil Kurian		2017	Alex Nole	02 May 2022 15:20:35	Accepted
<input type="checkbox"/>		Chapter 8 - Smart Irrigation and crop security in agriculture using IoT	Sugamya Katta and Sangita Ramatenki and Harika Sammeta		2022	Alex Nole	02 May 2022 15:06:45	Accepted

Figura 17 Documentos Aceptados para revisión

d. Evaluación de calidad

Para el desarrollo de calidad se utilizaron los 42 documentos seleccionados en la sección de selección de estudios. Aquí se realizó una revisión más completa, comprendiendo el resumen y los resultados de cada investigación, así mismo dando una calificación a cada documento para validar que estos documentos contienen información relevante para el desarrollo de la investigación. En la figura 17 se presentan los documentos aceptados.

Quality Assessment

Show: All Done Pending Score higher than 1.9 Score lower or equal to 1.9
 Order by:

To answer the form you may click on the desired answer on the following tables.

A Precise and Autonomous Irrigation System For Agriculture: IoT Based Self Propelled Center Pivot Irrigation System (2019)				0.5
QA:1¿El autor identifica o detalla algún problema dentro de la construcción de aplicaciones de irrigación con IoT?	Si	Probablemente	No	
QA:2¿Existe en el artículo citas sobre protocolos para la construcción de aplicaciones de irrigación con IoT?	Si	Probablemente	No	
QA:3¿El autor describe o detalla al menos una arquitectura para la construcción de aplicaciones de irrigación con IoT?	Si	Probablemente	No	
QA:4 ¿El autor detalla o al menos utiliza una técnica para la construcción de aplicaciones de irrigación con IoT?	Si	Probablemente	No	

A new Kappa Architecture for IoT Data Management in Smart Farming (2021)				1.5
QA:1¿El autor identifica o detalla algún problema dentro de la construcción de aplicaciones de irrigación con IoT?	Si	Probablemente	No	
QA:2¿Existe en el artículo citas sobre protocolos para la construcción de aplicaciones de irrigación con IoT?	Si	Probablemente	No	
QA:3¿El autor describe o detalla al menos una arquitectura para la construcción de aplicaciones de irrigación con IoT?	Si	Probablemente	No	
QA:4 ¿El autor detalla o al menos utiliza una técnica para la construcción de aplicaciones de irrigación con IoT?	Si	Probablemente	No	

Figura 18 Evaluación de calidad de los documentos

Aquí se seleccionan los documentos con una calificación mayor a 1,9, en donde quedaron para la revisión final 25 documentos que contienen información relevante para la investigación.

e. Extracción de datos

En esta sección se extrae la información de acuerdo al formulario de extracción de datos, en donde se obtiene la información que será utilizada para responder las preguntas de investigación, en la figura 18 se presenta la extracción de información de algunos documentos seleccionados.

To-do

Done

All

An IoT Based Smart Irrigation System **3.0** [mark as undone](#)

Protocolos	No detalla protocolos en el informe
Arquitecturas	Presenta una arquitectura de tres capas capa de sensores en donde se ubican los sensores de humedad, la capa de control en donde se encuentran los componentes Arduino y GSM y la capa de aplicación en donde se encuentra la web para el usuario
Metodología	La metodología a utilizar se basa en tres partes el análisis de los datos para el calculo del caudal del agua, el desarrollo del sistema de irrigación y las pruebas del sistema de irrigación en un ambiente controlado
Problemas en la construcción de aplicaciones de irrigación	No detallan problemas sobre el sistema de irrigación
Herramientas	Sensor de humedad YL-69, módulos de comparadores LM393, módem GSM SIM900A, Arduino UNO

Figura 19 Extracción de datos de los documentos seleccionados

f. Análisis de datos

En la sección de análisis de datos se presentan gráficas atractivas de los documentos seleccionados, por año, por la base de datos, entre otros. En la figura 19 se presenta la ilustración de las gráficas presentadas por la aplicación.

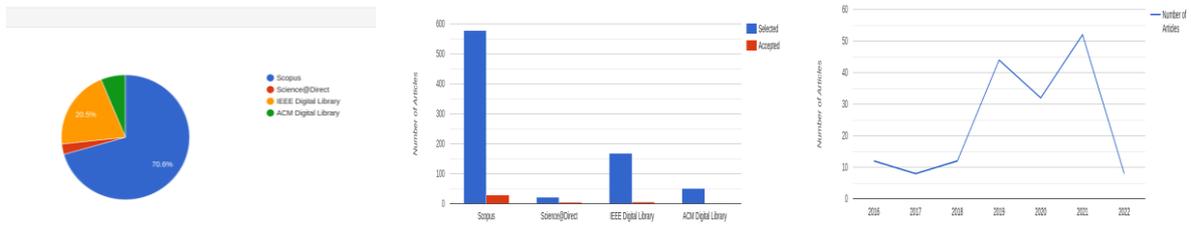


Figura 20 Graficas de análisis de datos de los documentos seleccionados

Anexo 3 IEEE 830

Especificación de requisitos de software

Proyecto: Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja

Revisión 1.0

Angel Favian Minga Medina

Alex Santiago Nole Reyes

Agosto 2022

Ficha del documento:

Fecha	Revisión	Autor
27-08-2022		<ul style="list-style-type: none">• Angel Minga• Alex Nole



Contenido

1. Introducción.....	143
1.1. Propósito.....	143
1.2. Alcance.....	143
1.3. Personal involucrado	143
1.3.1. Responsables del proyecto a desarrollar	143
1.3.2. Responsables del control de Riego.....	143
1.3.3. Director de Trabajo de titulación	144
1.4. Definiciones, acrónimos y abreviaturas	144
1.5. Referencias	144
1.6. Resumen	144
2. Descripción general	145
2.1. Perspectiva del producto	145
2.2. Funcionalidad del producto.....	145
2.3. Características de los usuarios.....	146
2.4. Restricciones	146
2.5. Suposiciones y dependencias	147
2.6. Evolución previsible del sistema.....	147
3. Requisitos específicos	147
3.1. Requisitos Funcionales.....	147
3.2. Requisitos No Funcionales.....	152
3.3. Requisitos comunes de los interfaces.....	155
3.3.1. Interfaces de usuario.....	155
3.3.2. Interfaces de hardware.....	155



1. Introducción

En el presente documento se detalla la Especificación de Requisitos de Software (SRS) del proyecto de titulación “**Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja**”. Para el desarrollo del documento de SRS, se ha basado en las directrices dadas por el estándar IEEE-830.

1.1. Propósito

El propósito del presente documento de SRS es definir las de manera clara y precisa todas las funcionalidades y restricciones del sistema web y aplicativo móvil para el “**Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja**”.

1.2. Alcance

Este documento de SRS está dirigida para los usuarios del sistema (ingenieros de ingeniería agrícola de la Universidad Nacional de Loja y agricultores), con la finalidad de realizar el desarrollo del sistema de control de riego para los cultivos de la Universidad Nacional de Loja basado en IoT.

1.3. Personal involucrado

1.3.1. Responsables del proyecto a desarrollar

Nombre	Angel Favian Minga Medina
Rol	Desarrollador de Software
Categoría profesional	Estudiante de Ingeniería en Sistemas
Información de contacto	afmingam@unl.edu.ec

Nombre	Alex Santiago Nole Reyes
Rol	Desarrollador de Software
Categoría profesional	Estudiante de Ingeniería en Sistemas
Información de contacto	alex.nole@unl.edu.ec

1.3.2. Responsables del control de Riego

Nombre	Ingenieros de la carrera de Ingeniería Agrícola de la UNL
--------	---



Rol	Responsables de los sistemas de riego
Categoría profesional	Ingenieros, estudiantes de la carrera de ingeniería Agrícola de la UNL

1.3.3. Director de Trabajo de titulación

Nombre	Edwin Rene Guamán Quinche
Rol	Director de Trabajo de Titulación
Responsabilidad	Supervisar y asesorar en el desarrollo del Trabajo de Titulación
Información de contacto	de rguaman@unl.edu.ec

1.4. Definiciones, acrónimos y abreviaturas

Nombre	Descripción
Usuario	Persona que usará el sistema para gestionar procesos
SRS	Especificación de Requisitos de Software
OCL	Lenguaje de Restricciones de Objetos
RF	Requisito Funcional
RNF	Requisito no Funcional
NL	Lenguaje Natural
MQTT	Protocolo de comunicación para sistemas IoT
IoT	Internet de las cosas
Topic	Nombre de sensor

1.5. Referencias

Referencia	Título	Ruta	Fecha	Autor
1	IEEE-830	http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf	1998	IEEE-SA Standards Board

1.6. Resumen

El presente documento consta de tres secciones, las cuales se las detalla a continuación:

En la primera sección se muestra una introducción al documento de especificación de requisitos de software en donde se detallan los participantes, propósito y alcances del mismo.

En la segunda sección se presenta la descripción general en donde especifica los factores principales del dominio, así también se detalla la funcionalidad del producto representado por un diagrama de casos de uso del sistema y las restricciones que el mismo tendrá a la hora de desarrollarlo.

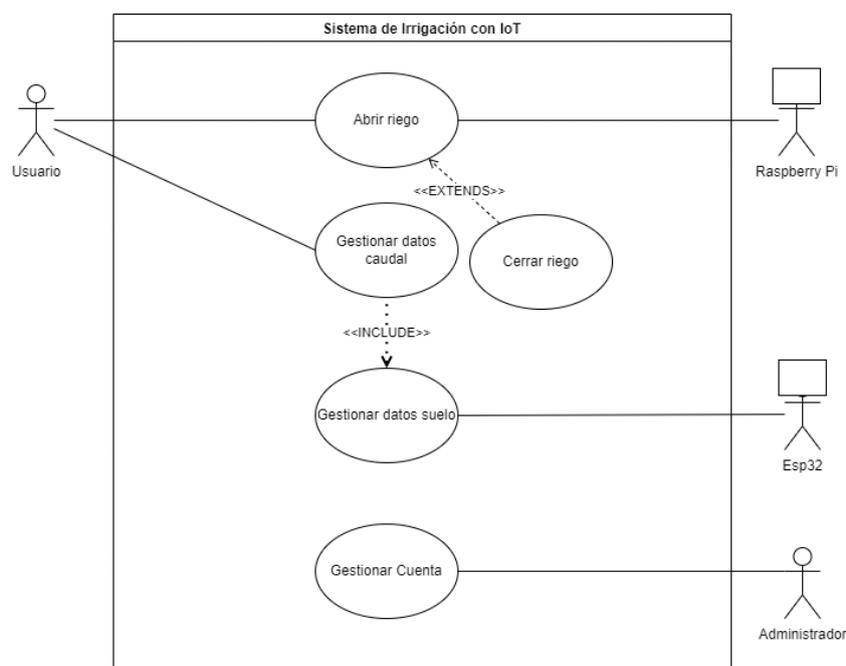
Por último, en la tercera sección se presenta detalladamente los requisitos funcionales y no funcionales que debe satisfacer las necesidades de los stakeholder.

2. Descripción general

2.1. Perspectiva del producto

El Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja que se va a desarrollar es independiente, diseñado para trabajar en entorno web y móvil, tendrá un diseño modular para el control de riego, el mismo que pretende apoyar en el proceso de irrigación para cultivos de la Universidad Nacional de Loja por parte de los ingenieros de la carrera de Ingeniería Agrícola.

2.2. Funcionalidad del producto





Abrir y Cerrar riego: En el presenta caso de uso el Usuario y Administrador puede encender y apagar el riego.

Analizar datos de sensores: En este caso de uso el usuario y administrador podrá visualizar mediante graficas estadísticas los datos obtenidos por los sensores.

Gestionar datos de sensores: En este caso de uso el sistema permitirá registrar los datos obtenidos por los sensores.

Gestionar Cuenta: En este caso de uso el administrador podrá agregar, modificar y buscar información sobre los usuarios del sistema.

Gestionar Topic: En este caso de uso el administrador podrá agregar, modificar y buscar información sobre los Topics o nombres de los sensores que estarán funcionando en el sistema.

Gestionar Tipo Suelo: En este caso de uso el administrador podrá agregar, modificar y buscar información sobre el tipo de suelo con los que se va a trabajar con el sistema.

2.3. Características de los usuarios

Tipo de usuario	Usuario
Formación	NA
Actividades	Interactuar activamente en el control de riego de acuerdo a las especificaciones del suelo

Tipo de usuario	Administrador
Formación	NA
Actividades	<ul style="list-style-type: none">• Control y manejo del sistema en general• Administrar Usuarios• Administrar Topics para sensores y actuadores• Administrar Tipos de Suelo• Control de Riego

2.4. Restricciones

- ✓ El sistema se diseñará según un modelo cliente/servidor.
- ✓ El sistema será desarrollado con el framework Nodejs para el servicio web.
- ✓ El sistema web será desarrollado con la librería Reactjs.
- ✓ El aplicativo móvil será desarrollado con la librería de Google Flutter.
- ✓ El sistema utilizará motor de la base de datos No relacional Mongo DB.
- ✓ La metodología para el desarrollo del sistema se basará en las metodologías ágiles.



2.5. Suposiciones y dependencias

- ✓ Se asume que los requisitos descritos en el presente documento son estables y se basan de acuerdo a los reglamentos de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja.

2.6. Evolución previsible del sistema

- ✓ La presente solución informática podrá evolucionar, agregándole más funcionalidades para el control de factores climáticos que intervienen para un mejor control de riego, lo cual permitirá que se consolide en un ecosistema para la irrigación.

3. Requisitos específicos

3.1. Requisitos Funcionales

Identificación del Requerimiento	RF001
Nombre del Requerimiento	Registro de Usuarios
Características	El Administrador podrá registrar los datos (Nombres, Apellidos, Dirección, Teléfono, correo, contraseña).
Descripción del Requerimiento	El administrador podrá registrar usuarios que interactuaran con el sistema de control de riego, en donde será necesarios los siguientes campos: Nombres, Apellidos, Dirección, Teléfono, Correo y Contraseña, además de su por defecto en cada registro será de Usuario
Requerimiento NO Funcional	RNF001 RNF004 RNF005 RNF006 RNF007
Prioridad del Requerimiento	Alta

Identificación del Requerimiento	RF002
Nombre del Requerimiento	Visualizar Usuario Registrados
Características	El administrador podrá visualizar los usuarios registrados



Descripción del Requerimiento	EL administrador podrá visualizar mediante una tabla los usuarios registrados en el sistema, y se mostrará toda la información relevante de cada usuario.
Requerimiento NO Funcional	RNF001 RNF004 RNF006 RNF007
Prioridad del Requerimiento	Media

Identificación del Requerimiento	RF003
Nombre del Requerimiento	Actualizar Usuarios
Características	El usuario, administrador podrá actualizar sus datos (Nombres, Apellidos, Dirección, Teléfono, correo, contraseña).
Descripción del Requerimiento	El administrados y los usuarios registrados podrán actualizar su información persona mediante un formulario que contenga la información antes registrada de los mismos
Requerimiento NO Funcional	RNF001 RNF004 RNF005 RNF006 RNF007
Prioridad del Requerimiento	Alta

Identificación del Requerimiento	RF004
Nombre del Requerimiento	Iniciar Sesión
Características	El usuario y administrador podrá iniciar sesión (correo, contraseña).
Descripción del Requerimiento	El administrados y usuarios registrados podrán ingresar al sistema mediante un inicio de sesión en donde se agregará correo y contraseña.
Requerimiento NO Funcional	RNF001 RNF004 RNF005 RNF006



	RNF007
Prioridad del Requerimiento	Alta

Identificación del Requerimiento	RF005
Nombre del Requerimiento	Registro de Tipos de Suelo
Características	El administrador podrá registrar el tipo de suelo (Tipo suelo, capacidad de campo, punto de marchitez)
Descripción del Requerimiento	El administrador podrá registrar los tipos de suelo mediante un formulario con los campos Tipo de textura de suelo, Capacidad de campo y punto de marchitez
Requerimiento NO Funcional	RNF001 RNF004 RNF005 RNF006 RNF007
Prioridad del Requerimiento	Alta

Identificación del Requerimiento	RF006
Nombre del Requerimiento	Actualización de Tipos de Suelo
Características	El administrador podrá actualizar el tipo de suelo (Tipo suelo, capacidad de campo, punto de marchitez)
Descripción del Requerimiento	EL administrador podrá actualizar la información sobre el tipo de suelo mediante un formulario con la información de las características de suelo antes registradas
Requerimiento NO Funcional	RNF001 RNF004 RNF005 RNF006 RNF007
Prioridad del Requerimiento	Alta



Identificación del Requerimiento	RF007
Nombre del Requerimiento	Visualizar Tipos de Suelo
Características	El administrador podrá visualizar los tipos de suelo registrados
Descripción del Requerimiento	El administrador podrá visualizar mediante una tabla los tipos de suelos registrados
Requerimiento NO Funcional	RNF001 RNF004 RNF006 RNF007
Prioridad del Requerimiento	Media

Identificación del Requerimiento	RF008
Nombre del Requerimiento	Control de Riego
Características	El usuario y administrador podrán controlar el riego (Encender y Apagar)
Descripción del Requerimiento	El administrador y los usuarios registrados podrán encender y apagar el riego mediante un botón interactivo.
Requerimiento NO Funcional	RNF001 RNF002 RNF003 RNF004 RNF005 RNF006 RNF007
Prioridad del Requerimiento	Alta

Identificación del Requerimiento	RF009
Nombre del Requerimiento	Visualizar datos de sensores
Características	El usuario y administrador podrán visualizar de forma gráfica los datos obtenidos por los sensores.



Descripción del Requerimiento	El administrador y los Usuarios podrán visualizar mediante gráficos lineales los datos capturados por los sensores.
Requerimiento NO Funcional	RNF001 RNF004 RNF006 RNF007
Prioridad del Requerimiento	Media

Identificación del Requerimiento	RF010
Nombre del Requerimiento	Registro de Topics
Características	El sistema permitirá registrar topics actuador y sensor para la conexión con el hardware
Descripción del Requerimiento	El sistema le permitirá al administrador registrar los topics o nombres de sensores necesarios para la captura de información de los mismos
Requerimiento NO Funcional	RNF001 RNF004 RNF005 RNF006 RNF007
Prioridad del Requerimiento	Alta

Identificación del Requerimiento	RF011
Nombre del Requerimiento	Visualizar Topics
Características	El sistema permitirá visualizar los topics registrados
Descripción del Requerimiento	El sistema permitirá visualizar los topics registrados mediante tablas en donde se presentará la información relevante de cada topic.
Requerimiento NO Funcional	RNF001 RNF004 RNF006 RNF007
Prioridad del Requerimiento	Media



Identificación del Requerimiento	RF012
Nombre del Requerimiento	Actualización de Topics
Características	El sistema permitirá actualizar topics actuador y sensor para la conexión con el hardware
Descripción del Requerimiento	El sistema permitirá la actualización de topics registrados en el sistema mediante un formulario cargando los datos antes registrados
Requerimiento NO Funcional	RNF001 RNF004 RNF005 RNF006 RNF007
Prioridad del Requerimiento	Alta

Identificación del Requerimiento	RF013
Nombre del Requerimiento	Registro de datos de sensores
Características	El sistema permitirá registrar los datos obtenidos por los sensores.
Descripción del Requerimiento	El sistema registrara automáticamente los datos enviados por los sensores
Requerimiento NO Funcional	RNF001 RNF004 RNF006 RNF007
Prioridad del Requerimiento	Alta

3.2. Requisitos No Funcionales

Identificación del Requerimiento	RNF001
Nombre del Requerimiento	Validación de credenciales



Características	El sistema validará las credenciales de los usuarios registrados para acceder al sistema.
Descripción del Requerimiento	El sistema validará las credenciales de los usuarios registrados mediante un inicio de sesión
Prioridad del Requerimiento	Alta

Identificación del Requerimiento	RNF002
Nombre del Requerimiento	Cálculos de la cantidad de agua
Características	El sistema realizará los cálculos para determinar la cantidad de agua necesaria.
Descripción del Requerimiento	El sistema deberá realizar los cálculos necesarios para determinar la cantidad de agua que se va a utilizar en los sembríos
Prioridad del Requerimiento	Alta

Identificación del Requerimiento	RNF003
Nombre del Requerimiento	Control de riego en tiempo real
Características	El control de riego debe realizarse en tiempo real.
Descripción del Requerimiento	El sistema permitirá que se realiza el control de riego en tiempo real haciendo uso del protocolo MQTT
Prioridad del Requerimiento	Alta

Identificación del Requerimiento	RNF004
Nombre del Requerimiento	Permisos de Acceso
Características	Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador del sistema.
Descripción del Requerimiento	El sistema le permitirá únicamente al administrador realizar cambios en los roles que permitirán definir las funcionalidades que realizarán cada usuario



Prioridad del Requerimiento	Alta
------------------------------------	------

Identificación del Requerimiento	RNF005
Nombre del Requerimiento	Alertas informativas
Características	El sistema debe proporcionar alertas informativas para los usuarios finales.
Descripción del Requerimiento	El sistema proporcionara alertas informativas después de cada acción realizada por los usuarios.
Prioridad del Requerimiento	Alta

Identificación del Requerimiento	RNF006
Nombre del Requerimiento	Portabilidad
Características	El sistema estará diseñado para funcionar en un entorno web de los navegadores de Google Chrome, Safari, Mozilla Firefox, Entre otros
Descripción del Requerimiento	El sistema será portable y podrá ejecutarse en cualquier entorno web
Prioridad del Requerimiento	Alta

Identificación del Requerimiento	RNF007
Nombre del Requerimiento	Disponibilidad
Características	El sistema estará disponible continuamente con un nivel de servicio para los usuarios de 7 días por 24 horas
Descripción del Requerimiento	El sistema estará disponible continuamente con un nivel de servicio para los usuarios de 7 días por 24 horas
Prioridad del Requerimiento	Alta



3.3. Requisitos comunes de los interfaces

3.3.1. Interfaces de usuario

- ✓ Las interfaces de usuario tendrán colores claro llamativos en lo que se refiere al fondo de pantalla, contará con un dashboard que permitirá interactuar con varias funcionalidades del sistema.
- ✓ Las interfaces de usuario estarán relacionadas con las pantallas, de registro, inicio de sesión, formularios, graficas lineales para los datos de los sensores, con las va a interactuar mediante el uso de los periféricos de entrada y salida (Mouse y Teclado).
- ✓ Las interfaces de usuario ayudarán al usuario y administrador a trabajar en un ambiente con Formularios, por lo que se dichas interfaces incluirán:
 - Botones iterativos
 - Menús desplegados
 - Mensajes informativos
 - Mensajes de error
 - Cuadros de diálogo
 - Tablas
 - Graficas lineales
 - Formularios para el ingreso, modificación, actualización y eliminación de datos. Así como para las operaciones.

3.3.2. Interfaces de hardware

- ✓ El sistema se debe alojar en un Host con un dominio que cuente con el nombre del sistema de riego.
- ✓ Para la utilización del entorno web se requerirá de equipos de cómputo con un procesador Intel Celeron o superiores con 4 Gb de ram.
- ✓ Para la utilización del aplicativo móvil se requerirá de dispositivos con Android 7 y con 2gb de memoria Ram.

1.2.1 Interfaces de software

- ✓ Aplicativo web: Multiplataforma con navegadores web (Google Chrome, Mozilla, Safari, Opera, etc.)
- ✓ Aplicativo Móvil: Android

1.2.2 Interfaces de comunicación

- ✓ Los servidores, clientes se comunicarán mediante protocolos de internet, para realizar las operaciones necesarias en el sistema.
- ✓ El sistema trabajará con el protocolo TCP el cual definirá como se mueve la información desde el remitente hasta el destinatario.

4. Anexos

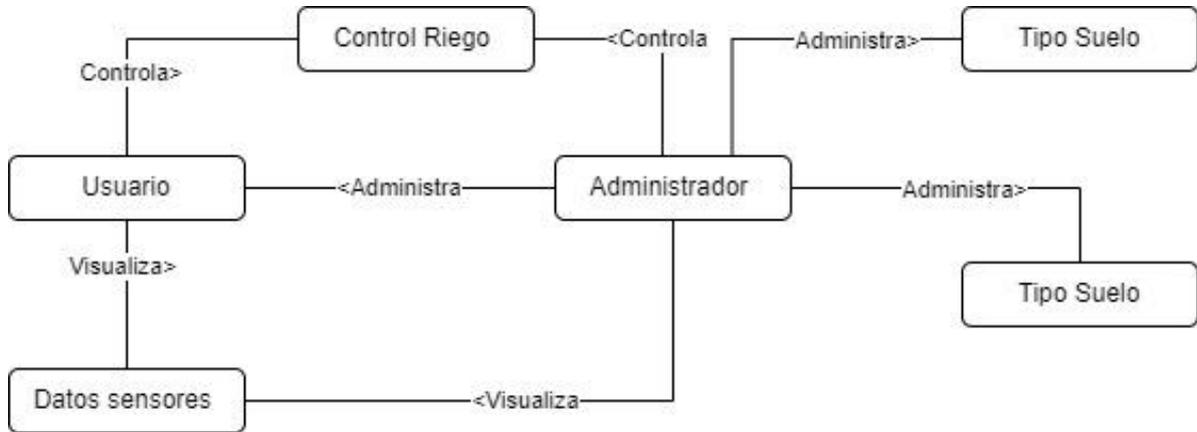


Figura 1: Diagrama de dominio



Anexo 4 Arquitectura 4+1

FACULTAD DE ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES

CARRERA DE INGENIERÍA EN SISTEMAS / COMPUTACIÓN

***ARQUITECTURA 4+1 PARA “EL SISTEMA DE IRRIGACIÓN
CON IoT PARA CULTIVOS DE LA UNIVERSIDAD NACIONAL
DE LOJA”***

Estudiantes:

- Alex Santiago Nole Reyes
- Angel Favian Minga Medina

Control del documento

1. Introducción.....	158
1.1. Propósito.....	158
1.2. Alcance	158
1.3. Usuarios Interesados.....	158
2. Referencias.....	158
3. Definiciones, acrónimos y abreviaciones.....	159
4. Framework Conceptual	160
4.1. Descripción de la arquitectura en contexto.....	160
4.2. Stakeholders y sus roles.....	160
5. Descripciones prácticas de arquitectura	160
5.1. Identificación de los Stakeholders y sus responsabilidades	160
5.2. Selección de puntos de vista de la arquitectura	161



5.3. Vistas de la arquitectura 161

1. Introducción

1.1. Propósito

Este documento detalla una descripción comprensiva arquitectónica del sistema, haciendo uso de modelos, diagramas y vistas, para representar las distintas necesidades que se anhelan capturar y así tomar decisiones significativas para el sistema.

1.2. Alcance

El presente documento contiene el diseño elaborado para el proyecto “**Sistema de irrigación con IoT para cultivos de la Universidad Nacional de Loja**”, el cual es producto de un análisis minucioso de los requisitos del sistema, por lo cual se determinó que estos pueden ser satisfechos con las tecnologías y características discutidas entre analistas, desarrolladores, clientes y usuarios.

El documento está organizado alrededor de tres ideas principales.

1. Las características generales del diseño
2. Los requisitos necesarios para poder diseñar el sistema
3. Los diagramas, modelos y vistas del sistema

1.3. Usuarios Interesados

Este documento de Arquitectura de Software (DAS), puede ser usado por todos aquellos usuarios que deseen comprender el diseño y construcción de la aplicación “**Sistema de irrigación con IoT para cultivos de la Universidad Nacional de Loja**”, y sirve como base para que los desarrolladores de software puedan construir el bajo nivel de la aplicación usando el lenguaje que más les acomode.

2. Referencias

Las referencias aplicables a este documento son:



- Documento de Arquitectura de Software IEEE-1471-2000
- Solución Informática para la administración de procedimientos odontológicos, Dennys Andrés Camacho Velíz
- Architecture Tradeoff Analysis Method
- The 4+1 View .Kruchten - 1009
- -IEEE 1471-2000

3. Definiciones, acrónimos y abreviaciones.

IoT: Internet de las Cosas, interactúa Hardware, software y redes

Autenticación: Es cuando el usuario se identifica en forma inequívoca; es decir, sin duda o equivocación de que es quien dice ser.

MQTT: Protocolo de Transporte de telemetría de Mensaje Server (Message Queuing Telemetry Transport) utilizado para la comunicación con los sensores y el una tarjeta programable que contiene un microcontrolador

Broker: Broker o servidor es el encargado de transmitir los mensajes recibidos por el publicador al suscriptor

Publicador: Es aquel que emite los datos obtenidos por los sensores.

Suscriptor: Es donde se presenta los datos obtenidos de los sensores.

Módulo: Es un componente de software o parte de un programa que contiene una o más rutinas.

Parámetro: Elemento o dato importante desde el que se examina un tema, cuestión o asunto.

React: Librería de JavaScript para el desarrollo de frontend

Flutter: Framework para el desarrollo de aplicaciones móviles

NodeJs: Framework del lado del servidor para la creación de servicios del sistema

MongoDB: Base de datos no relacional para el almacén de datos.

Raspberry PI: Microcomputador utilizado como servidor MQTT

Mosquito: Servidor MQTT instalado en raspberry Pi

Esp32: Tarjeta programable que contiene un Microcontrolador utilizado para la comunicación entre sensores y MQTT

DAS: Documento de Arquitectura de Software



4. Framework Conceptual

4.1. Descripción de la arquitectura en contexto

En el presente documento se presenta la arquitectura de Kruchten, el mismo que es un modelo de vista que cumple con el estándar "IEEE 1471-2000". Estas vistas son:

- **Vista lógica.** - Apoya principalmente los requisitos funcionales. Lo que el sistema debe brindar en términos de servicios a sus usuarios.
- **Vista de desarrollo.** - Se centra en la organización real de los módulos de software en el ambiente de desarrollo del software.
- **Vista de procesos.** - Toma en cuenta algunos requisitos no funcionales tales como la performance y la disponibilidad. Se enfoca en asuntos de concurrencia y distribución, integridad del sistema, de tolerancia a fallas.
- **Vista física.** - Toma en cuenta primeramente los requisitos no funcionales del sistema tales como la disponibilidad, confiabilidad (tolerancia a fallas), performance (throughput), y escalabilidad. El software ejecuta sobre una red de computadoras o nodos de procesamiento (o tan solo nodos).
- **Escenarios.** - Los escenarios son de alguna manera una abstracción de los requisitos más importantes. Su diseño se expresa mediante el uso de diagramas de escenarios y diagramas de interacción de objetos

Los elementos de las cuatro vistas trabajan conjuntamente en forma natural mediante el uso de un conjunto pequeño de escenarios relevantes.

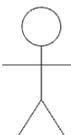
4.2. Stakeholders y sus roles

En este documento se presentan los Stakeholders y sus roles identificados a partir de la interpretación de los casos de uso del Negocio, descritos en la siguiente sección.

5. Descripciones prácticas de arquitectura

5.1. Identificación de los Stakeholders y sus responsabilidades

Stakeholder	Descripción	Escenario	Vistas
-------------	-------------	-----------	--------

 Administrador	Es la persona que encarga gestionar las cuentas del sistema de irrigación	<ul style="list-style-type: none"> ● Escenario de negocio 	<ul style="list-style-type: none"> ● CU Gestionar Cuenta
 Usuario	Es la persona encargada de interactuar con el sistema de irrigación	<ul style="list-style-type: none"> ● Escenario de negocio 	<ul style="list-style-type: none"> ● CU Abrir riego ● CU Cerrar Riego ● CU Analizar datos de sensores

5.2. Selección de puntos de vista de la arquitectura

Vistas	UML
Escenarios	Casos de uso
Lógica	Clases
Desarrollo	Componentes
Física	Despliegue
Procesos	Actividades

5.3. Vistas de la arquitectura

Vistas. - escenarios

La vista de escenarios se utilizó para identificar y validar el diseño arquitectónico del sistema. La descripción de la vista de escenario se ilustra utilizando los diagramas de casos de USO.

- a. Sistema
 - i. Dashboard de control para sistema de irrigación
 - ii. App móvil para control de riego
- b. Actores
 - i. Usuario
 - ii. Administrador
 - iii. Esp32

- iv. Raspberry pi
- c. Casos de Uso
 - i. Abrir riego
 - ii. Cerrar riego
 - iii. Gestionar Cuenta
 - iv. Gestionar Datos sensores
 - v. Analizar datos sensores

Dentro de la vista de escenarios se presenta el diagrama de casos de uso general, así como los diagramas de casos de uso específicos con su respectiva descripción.

Diagrama. - Caso de uso del negocio – Caso de uso de diseño

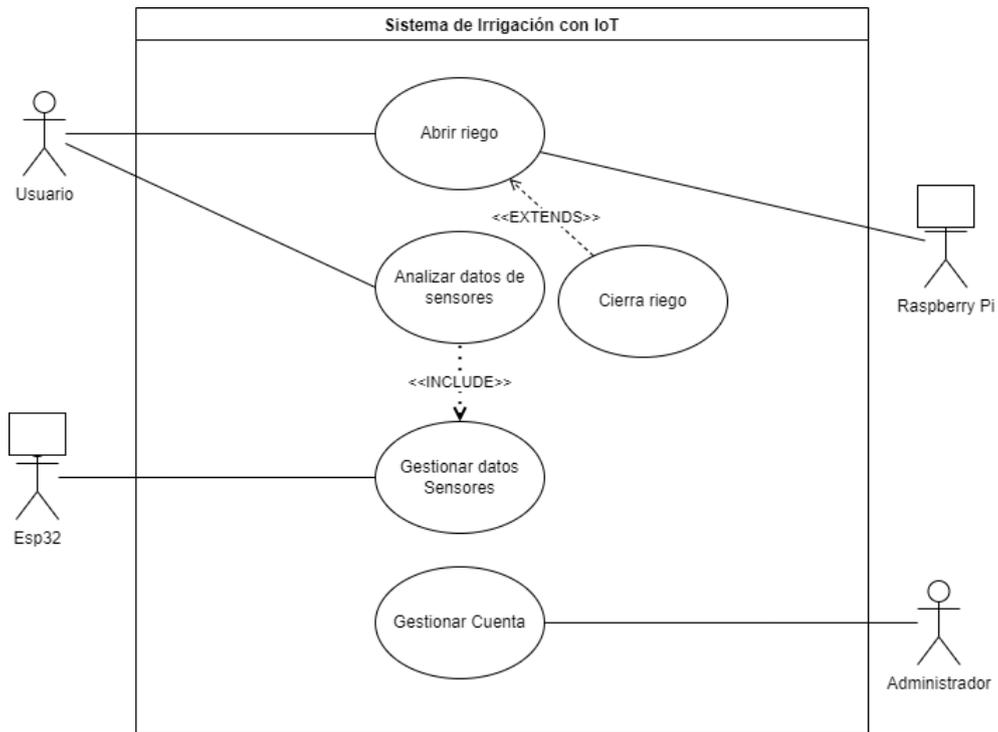


Figura 1: Caso de uso General

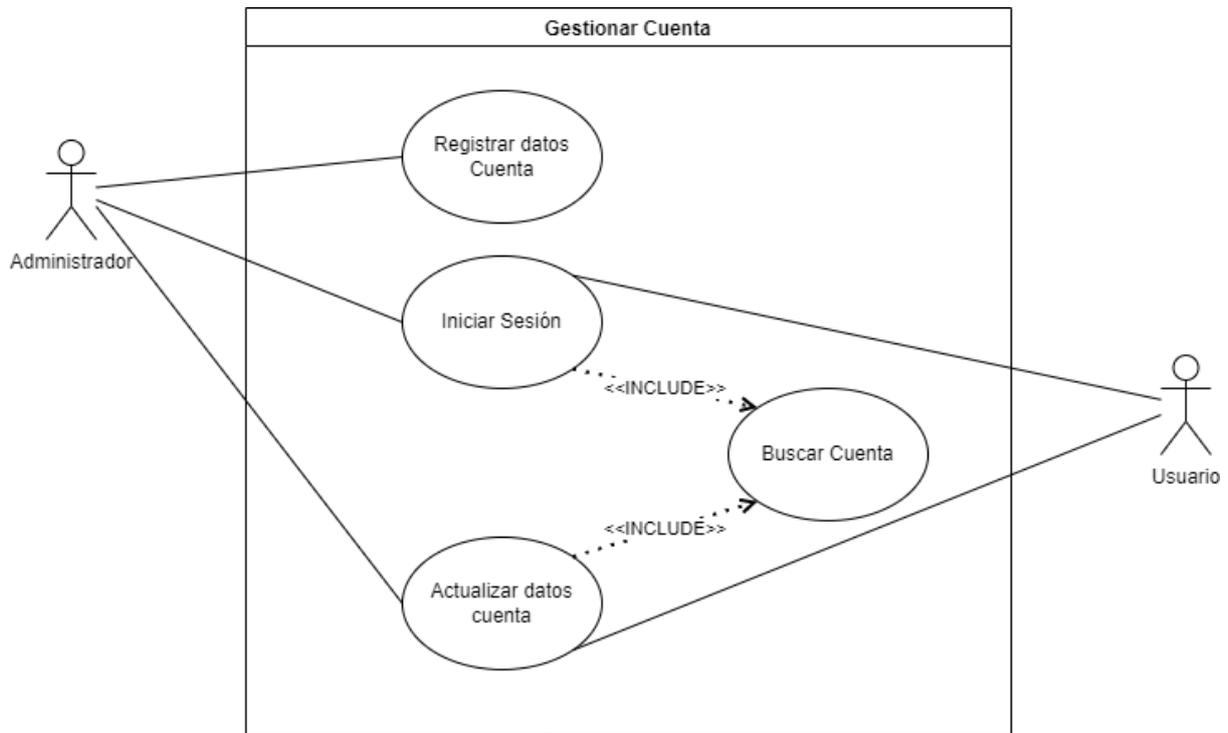


Figura 2: Caso de uso gestionar cuenta



Figura 3: Prototipo vista login

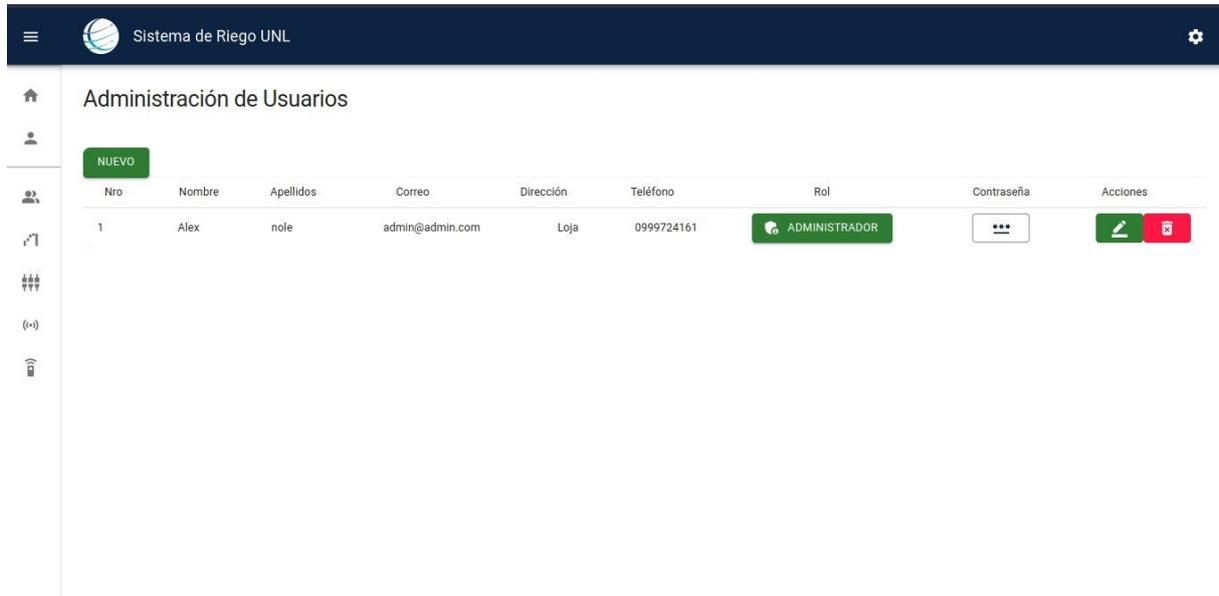


Figura 4: Prototipo vista lista de usuarios

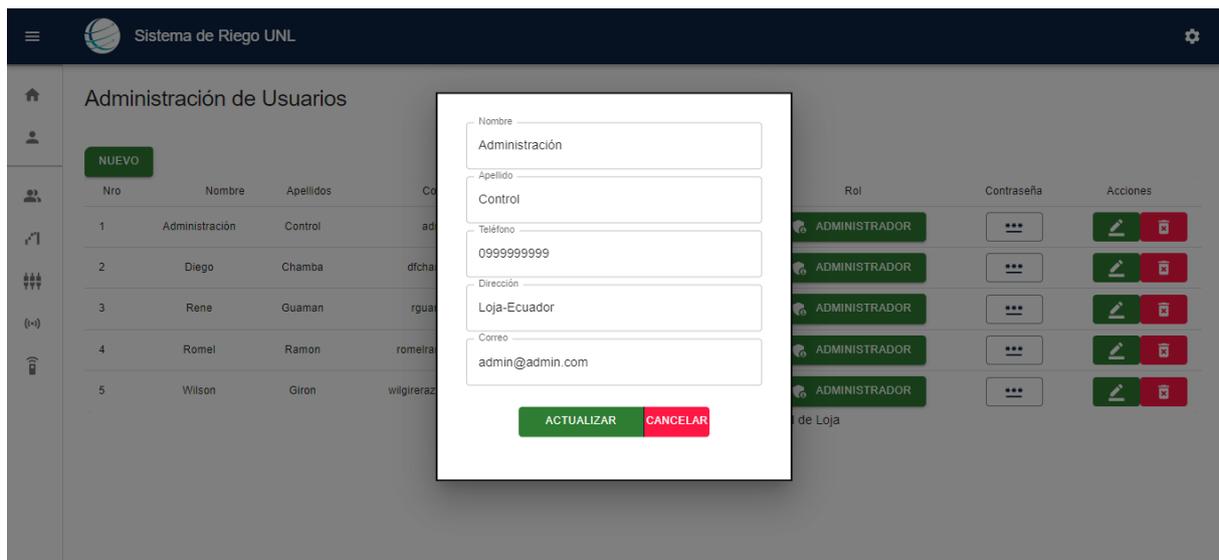


Figura 5: Prototipo vista registro

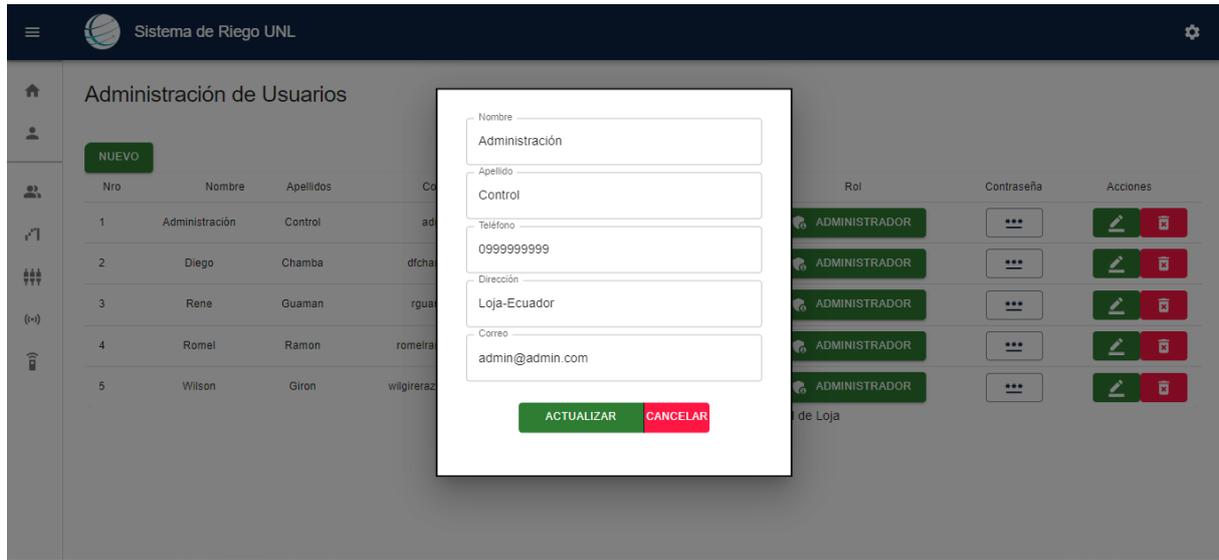


Figura 6: Prototipo vista actualizar

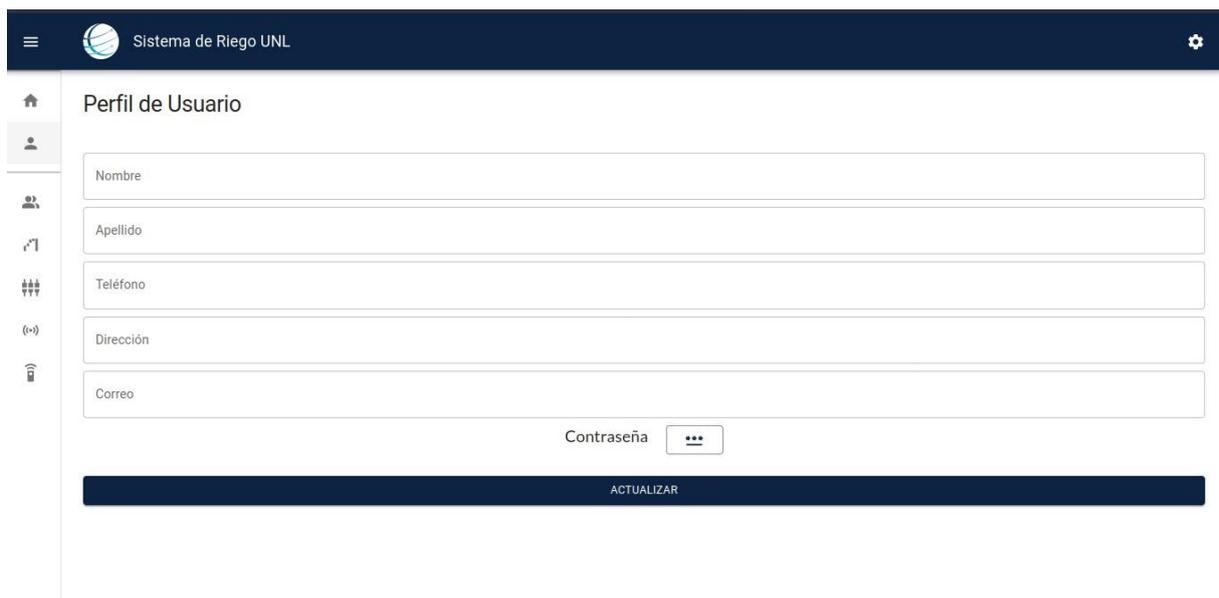


Figura 6: Prototipo vista perfil usuario

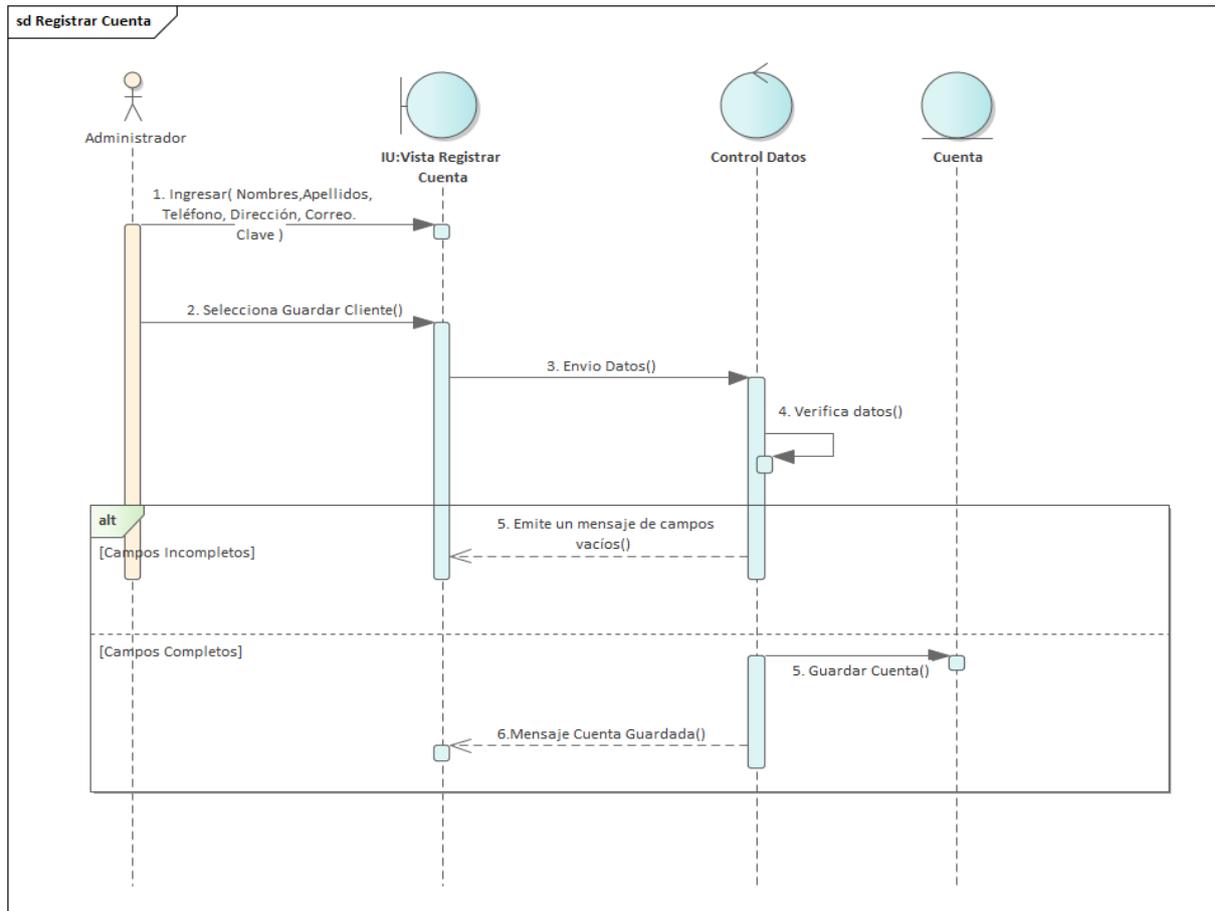


Figura 7: Diagrama de secuencia, registro de cuenta

Tabla 1: Descripción de caso de uso, Registrar Datos cuenta.

Nombre:	Registrar datos Cuenta
Descripción:	El UC Registrar datos cuenta permite al Administrador registrar datos de Usuarios en el sistema.
Actores:	Administrador
Precondiciones:	No estar registrado en el sistema. Tener un correo activo. Estar en la interfaz crear cuenta nueva.
Flujo Normal:	



Acciones del ADMINISTRADOR	Acciones del Sistema
1. Ingresar los datos(Nombre, Apellido, teléfono, Dirección, Correo, Clave) en el formulario de la interfaz REGISTRAR CUENTA	
2. Selecciona la opción [Registrar]	3. Válida los campos ingresados
	4. Guarda los datos de la cuenta
	5. Presenta un mensaje de confirmación “Usuario registrado correctamente”.
	6. fin de UC.
Flujo Alternativo:	
	A. Datos del cliente Incorrecto
	A.4. Presenta mensaje de error “Los datos ingresados son incorrectos”.
	A.5. El U.C. continúa en el paso 1 del flujo normal de eventos.
	A.6. Fin de UC
A. El USUARIO selecciona la opción cancelar	
	B.3. Muestra un mensaje emergente “Seguro que desea cancelar el registro”.
B.4. Selecciona la opción “SI”	B.5. Redirecciona a la interfaz PRINCIPAL
	B.6. Fin del UC

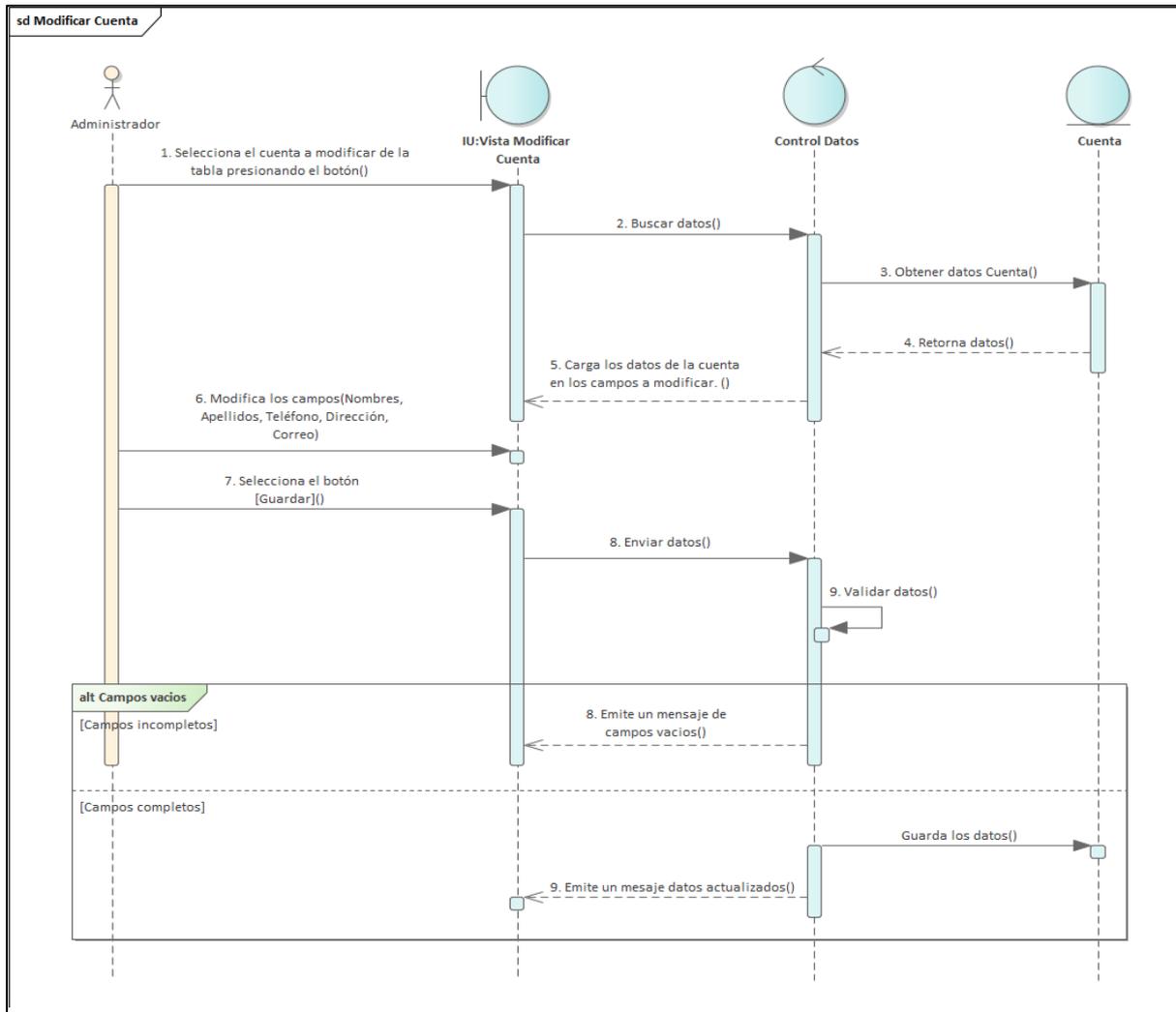


Figura 8: Diagrama de secuencia modificar cuenta

Tabla 2: Descripción de caso de uso, modificar datos cuenta.

Nombre:	Modificar datos de cuenta
Descripción:	El UC permite al Administrador y Usuario actualizar sus datos de cuenta
Actores:	Administrador Cliente
Precondiciones:	El Usuario debe haber iniciado sesión en el sistema. Estar en la interfaz Actualizar datos de cuenta.



Flujo Normal:	
Acciones del USUARIO	Acciones del Sistema
1. Selecciona la Opción [Actualizar Datos]	2. Busca información de cuenta
	3. Presenta los campos llenos con la información de la cuenta
4. Selecciona el campo que desea actualizar	
5. Ingresa nueva información en los campos solicitados.	
6. Selecciona la opción [Actualizar]	7. Válida los campos ingresados
	8. Guarda los datos del cliente
	9. Presenta un mensaje de confirmación “Los datos de Usuario actualizados correctamente”.
	10. fin de UC.
Flujo Alternativo:	
A. Ingresa información errónea al sistema (caracteres desconocidos)	A.8 Presenta mensaje de error “Los datos ingresados son incorrectos”.
	A.9. El U.C. continúa en el paso 1 del flujo normal de eventos.
	A.10. Fin de UC
B. Selecciona la Opción [Cancelar]	B.7. Presenta un mensaje de confirmación “Está seguro que desea cancelar la actualización”
B.8 Selecciona la opción [Si]	B.9 Redirecciona a la interfaz PRINCIPAL
	B.10 El UC finaliza

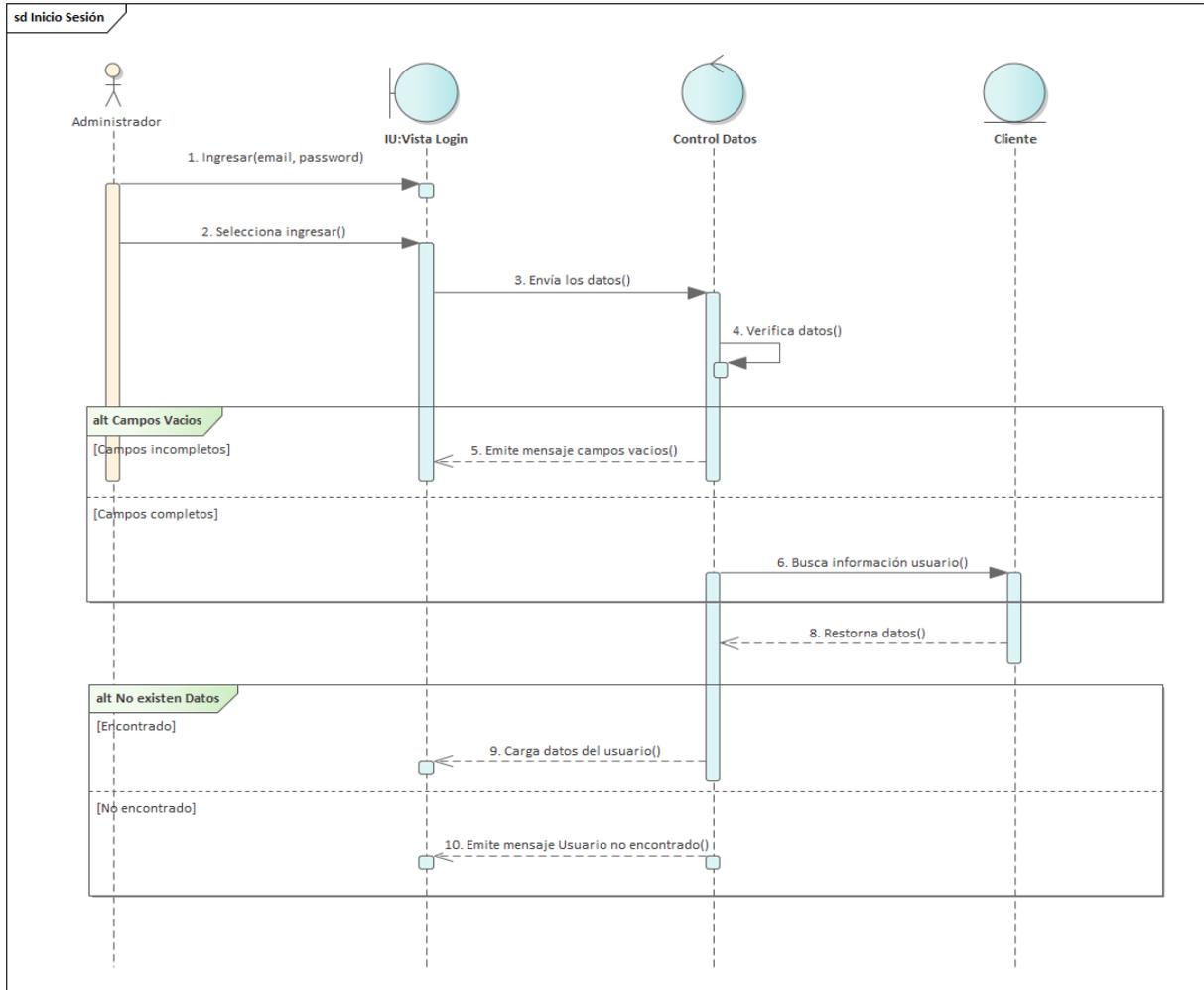


Figura 9: Diagrama de secuencia inicio de sesión

Tabla 3: Descripción de caso de uso, Iniciar Sesión.

Nombre:	Iniciar Sesión
Descripción:	El UC permite al Administrador y Usuario iniciar sesión en el sistema.
Actores:	Administrador Cliente
Precondiciones:	Estar registrado en el sistema. Tener un correo activo. Estar en la interfaz Iniciar sesión.
Flujo Normal:	



Acciones del USUARIO	Acciones del Sistema
1. Ingresar los datos (Correo, Clave) en el formulario de la interfaz INICIAR SESIÓN	
2. Selecciona la opción [Ingresar]	3. Válida los campos estén completos
	4. Busca las credenciales del usuario en el sistema
	5. Redirecciona a la interfaz de Inicio del sistema
	6. fin de UC.
Flujo Alternativo:	
A. Ingresar al formulario campos incompletos	A. 4. Presenta un mensaje de error “Los campos están incompletos”
	A.5. El U.C. continúa en el paso 1 del flujo normal de eventos.
B. Los datos ingresados son incorrectos o no existen	B.5. Presenta un mensaje de error “Las credenciales ingresadas son incorrectas o no está registrado en el sistema”
	B.6. El U.C. continúa en el paso 1 del flujo normal de eventos.
	B.7. Fin de UC

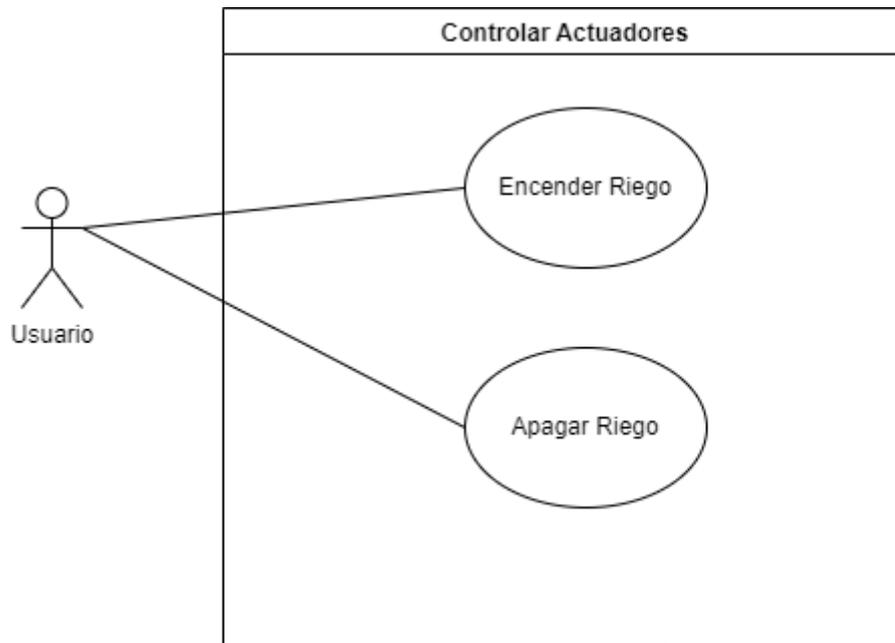


Figura 10: Controlar Actuadores

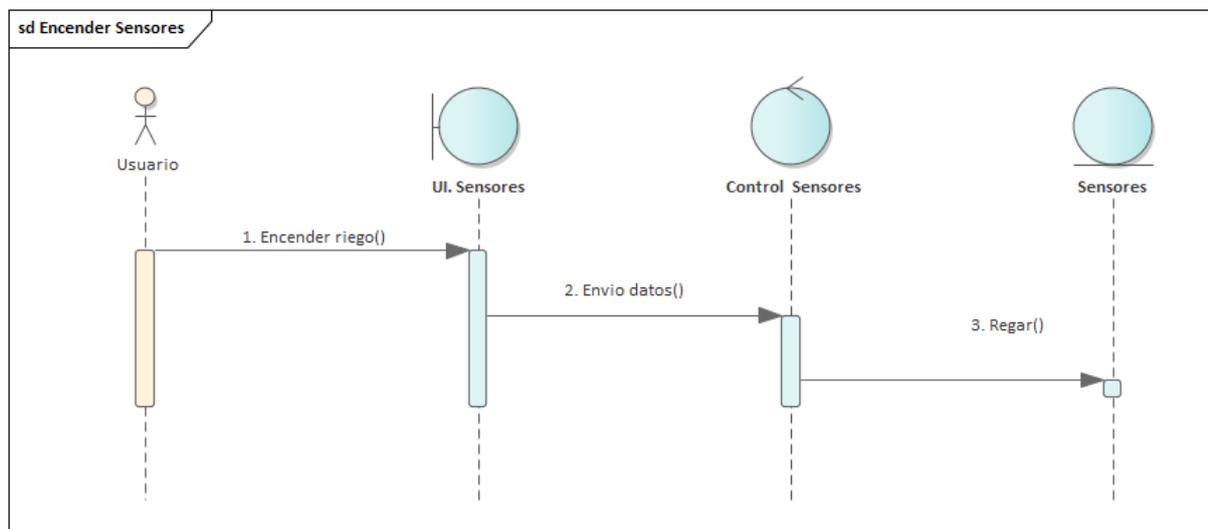


Figura 11: Diagrama de secuencia encender riego

Tabla 4: Descripción de caso de uso, Encender riego.

Nombre:	Encender Riego
Descripción:	El UC permite al Usuario y Administrador encender el riego



Actores: Usuario Administrador	
Precondiciones: Tener conectados el actuador Estar energizado Haber iniciado sesión Estar en la vista de encender riego	
Flujo Normal:	
Acciones del USUARIO- ADMINISTRADOR	Acciones del Sistema
1. Selecciona el botón encender riego	
	2. Abre Riego
	3. fin de UC.

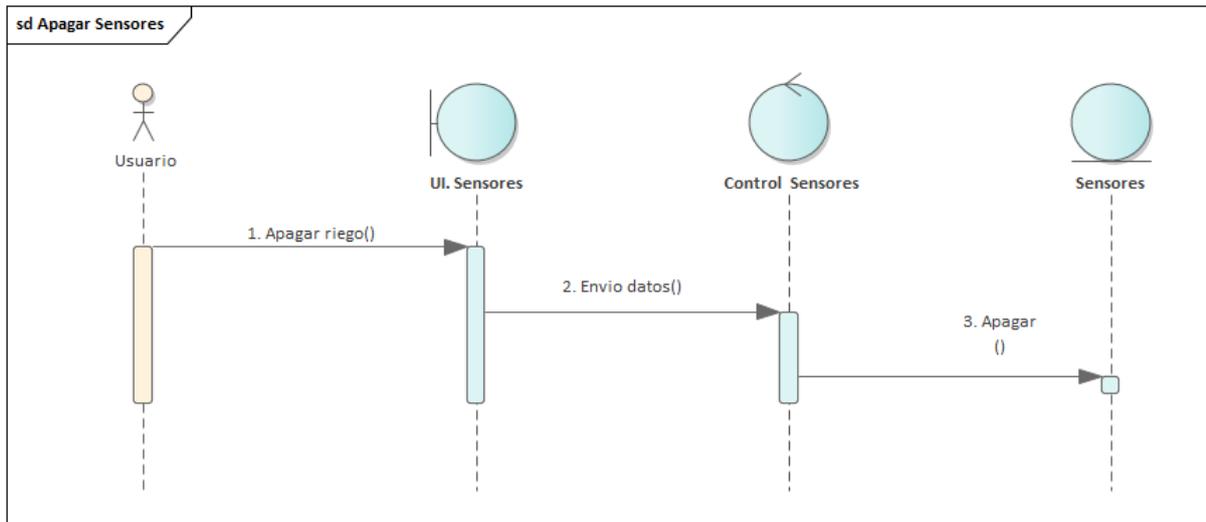


Figura 12: Diagrama de secuencia apagar riego

Tabla 5: Descripción de caso de uso, Apagar riego.

Nombre:	Apagar Riego
Descripción: El UC permite al Usuario y Administrador apagar el riego	
Actores: Usuario Administrador	



Precondiciones: Tener conectados el actuador Estar energizado Haber iniciado sesión Estar en la vista de encender riego	
Flujo Normal:	
Acciones del USUARIO- ADMINISTRADOR	Acciones del Sistema
1. Selecciona el botón apagar riego	
	2. Cierra el Riego
	3. fin de UC.

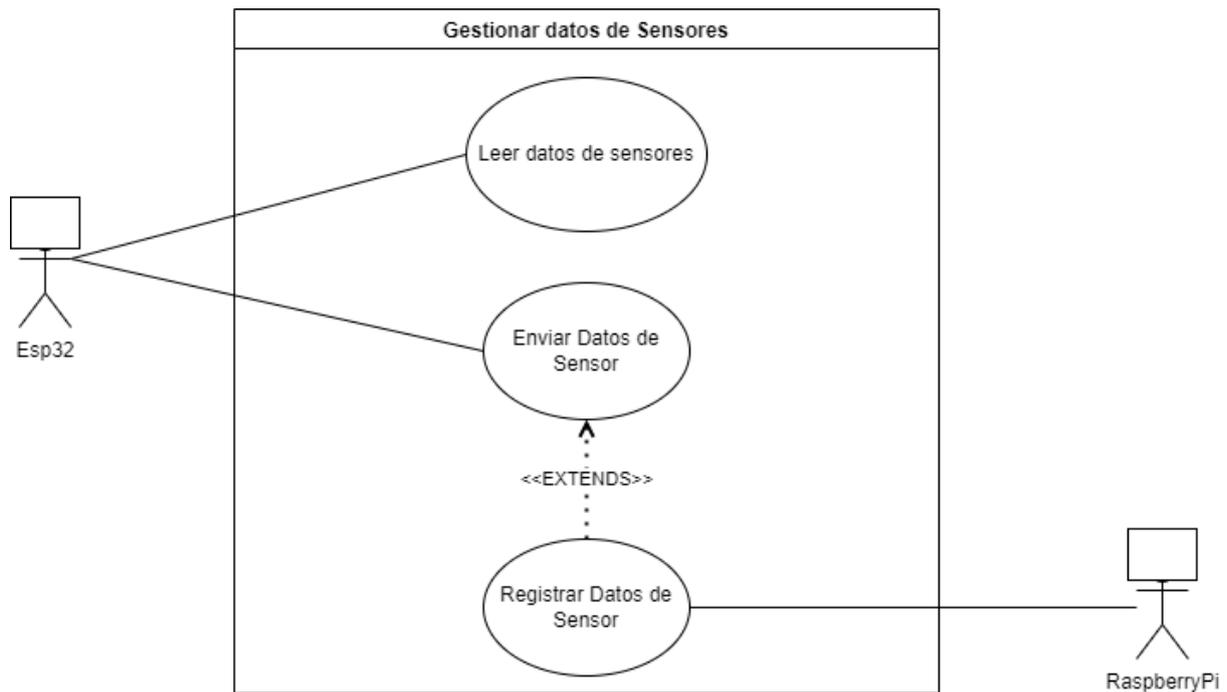


Figura 13: Caso de uso gestionar datos de sensores

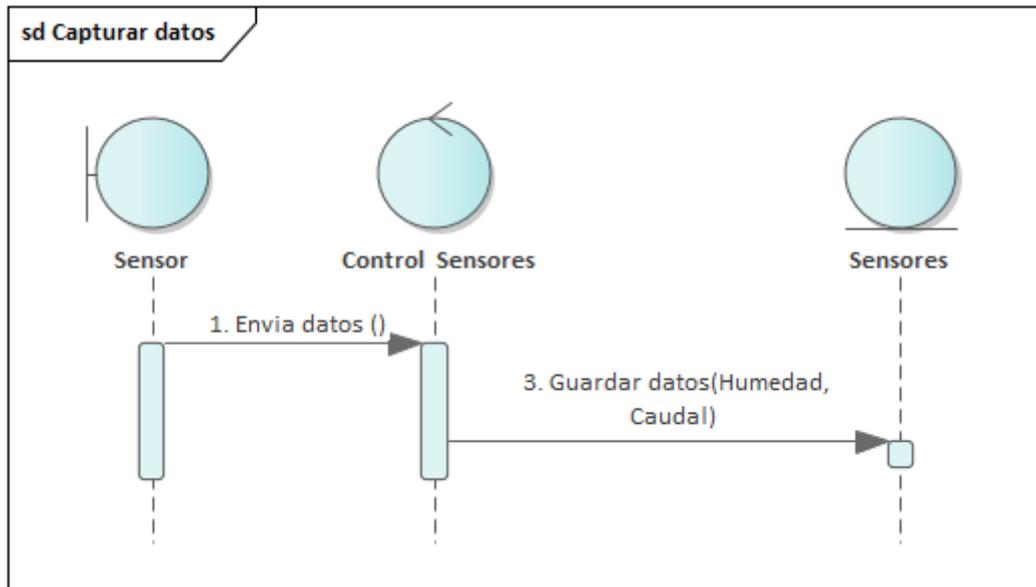


Figura 14: Diagrama de secuencia capturar datos

Tabla 4: Descripción de caso de uso, Registrar Datos de Sensores.

Nombre:	Registrar Datos de Sensores
Descripción:	El UC permite al Actor (Esp32, Raspberry Pi o Broker) registrar los datos obtenidos por los sensores.
Actores:	Raspberry Pi o Broker
Precondiciones:	El servidor debe estar levantado Estar escuchando por el puerto 1883 o 8883 Estar conectado con el broker para la transmisión de datos
Flujo Normal:	
Acciones del Raspberry Pi o Broker	Acciones del SIstema
1. Recoge datos ambientales brindados por el Esp32	
2. Envía datos recibidos mediante el protocolo MQTT	3. Recibe los datos enviados
	4. Válida que contengan datos de cada



	sensor
	5. Guarda los datos ambientales
	6. Fin U.C.
Flujo Alternativo:	
A. Los datos de los sensores no son recibidos	A.4. Presenta un mensaje de error “Verificar sensor datos no recibidos”
	A.5. El U.C. continúa en el paso 1 del flujo normal de eventos.
	A.6. Fin de UC

Tabla 5: Descripción de caso de uso, Enviar, Datos de sensores.

Nombre:	Enviar Datos de sensores
Descripción: El UC permite al Actor (Esp32) envía los datos ambientales obtenidos por los sensores.	
Actores: Esp32	
Precondiciones: El código debe estar ingresado en el Esp32 Estar conectado con el broker para la transmisión de datos	
Flujo Normal:	
Acciones del Esp32	Acciones del Sistema
1. Recoge datos ambientales obtenidos por los sensores	
2. Envía datos recibidos mediante el protocolo MQTT hacia el broker	3. Recibe los datos enviados
	4. Válida que contengan datos de cada sensor
	5. Fin U.C.
Flujo Alternativo:	



A. Los datos de los sensores no son recibidos	A.4. Presenta un mensaje de error “Verificar sensor datos no recibidos”
	A.5. El U.C. continúa en el paso 1 del flujo normal de eventos.
	A.6. Fin de UC

Tabla 6: Descripción de caso de uso, Leer datos de Sensores.

Nombre:	Leer datos de Sensores
Descripción: El UC permite al Actor (Esp32) leer los datos de los sensores.	
Actores: Esp32	
Precondiciones: Tener conectados los sensores de caudal y suelo Estar energizado	
Flujo Normal:	
Acciones del Esp32	Acciones del Sistema
1. Captura datos de los sensores	
	2. Envía datos de sensores a Broker
	3. fin de UC.

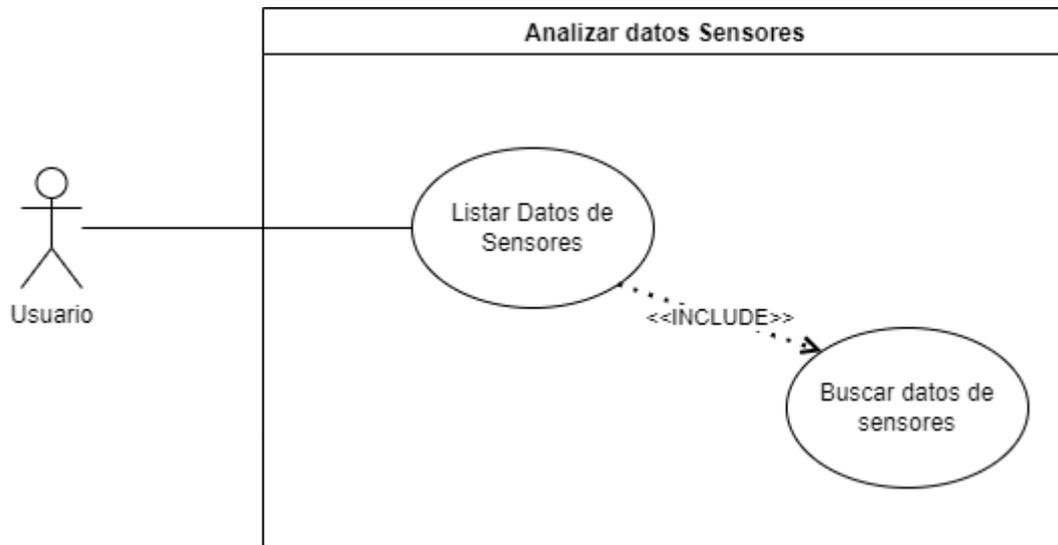


Figura 15: Caso de uso analizar datos de sensores

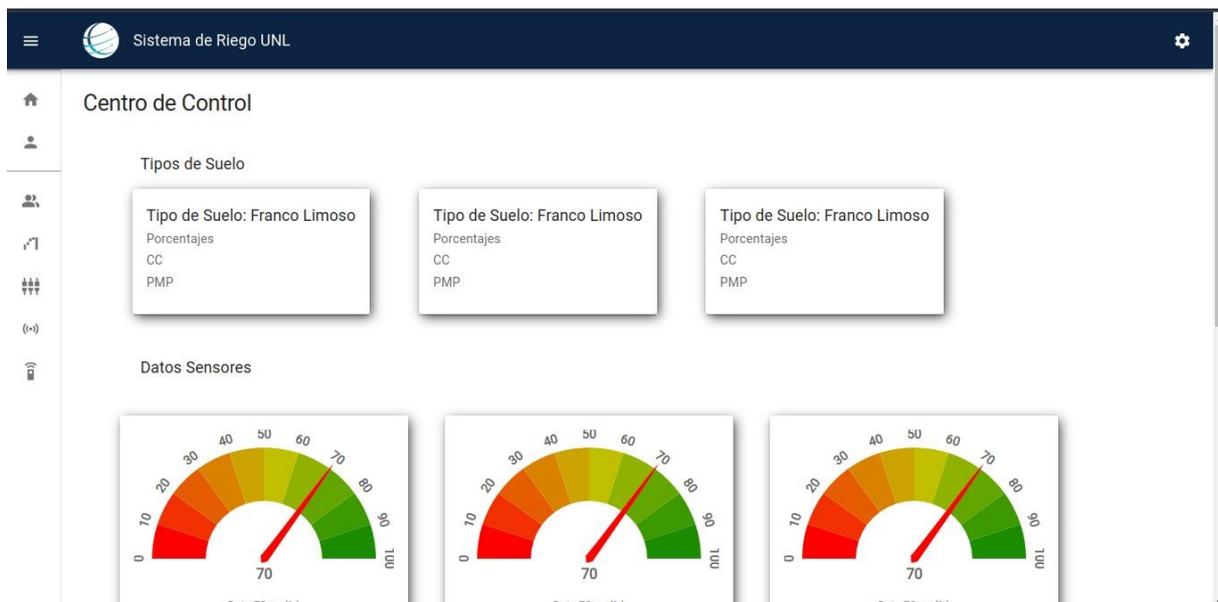


Figura 16: Prototipo de vista principal

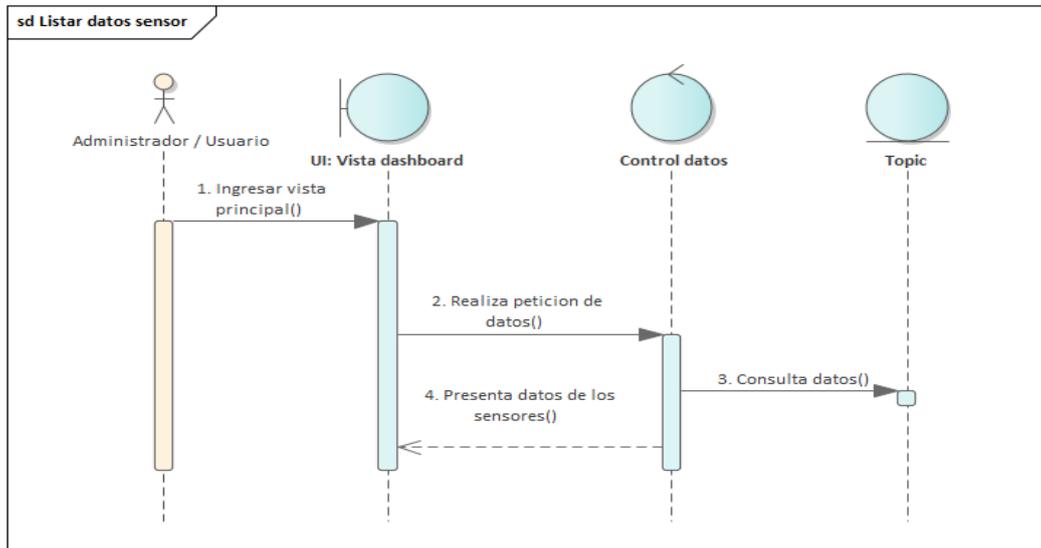


Figura 17: Diagrama de secuencia listar datos de sensor

Tabla 7: Descripción de caso de uso, Listar Datos de Sensores.

Nombre:	Listar Datos de Sensores
Descripción:	El UC permite al Administrador y Usuario Listar los datos obtenidos por los sensores
Actores:	Administrador Usuario
Precondiciones:	El Administrador o Usuario debe haber iniciado sesión en el sistema.
Flujo Normal:	
Acciones del Administrador o Usuario	Acciones del Sistema
1. Ingresa a la interfaz Dashboard	
	2. Carga información de los datos obtenidos por los sensores
	3. Presenta la información de los sensores existentes
	4. fin de UC.

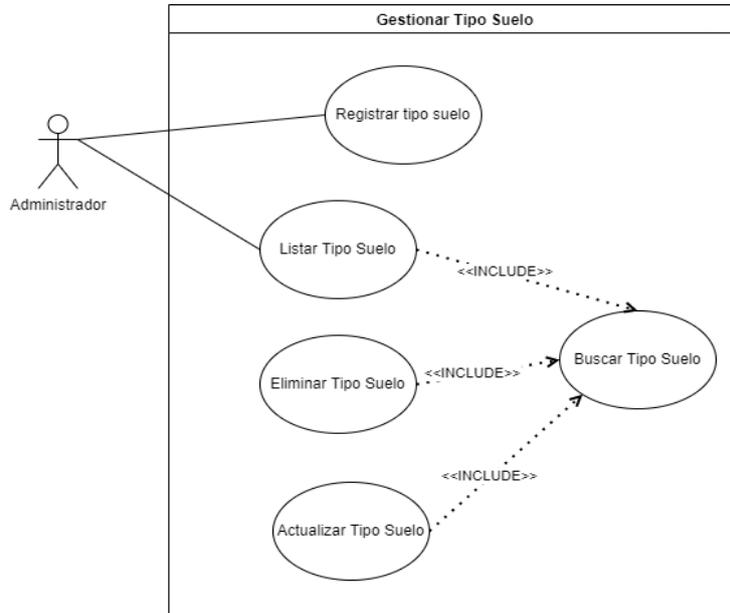


Figura 18: Caso de uso gestionar tipo de suelo

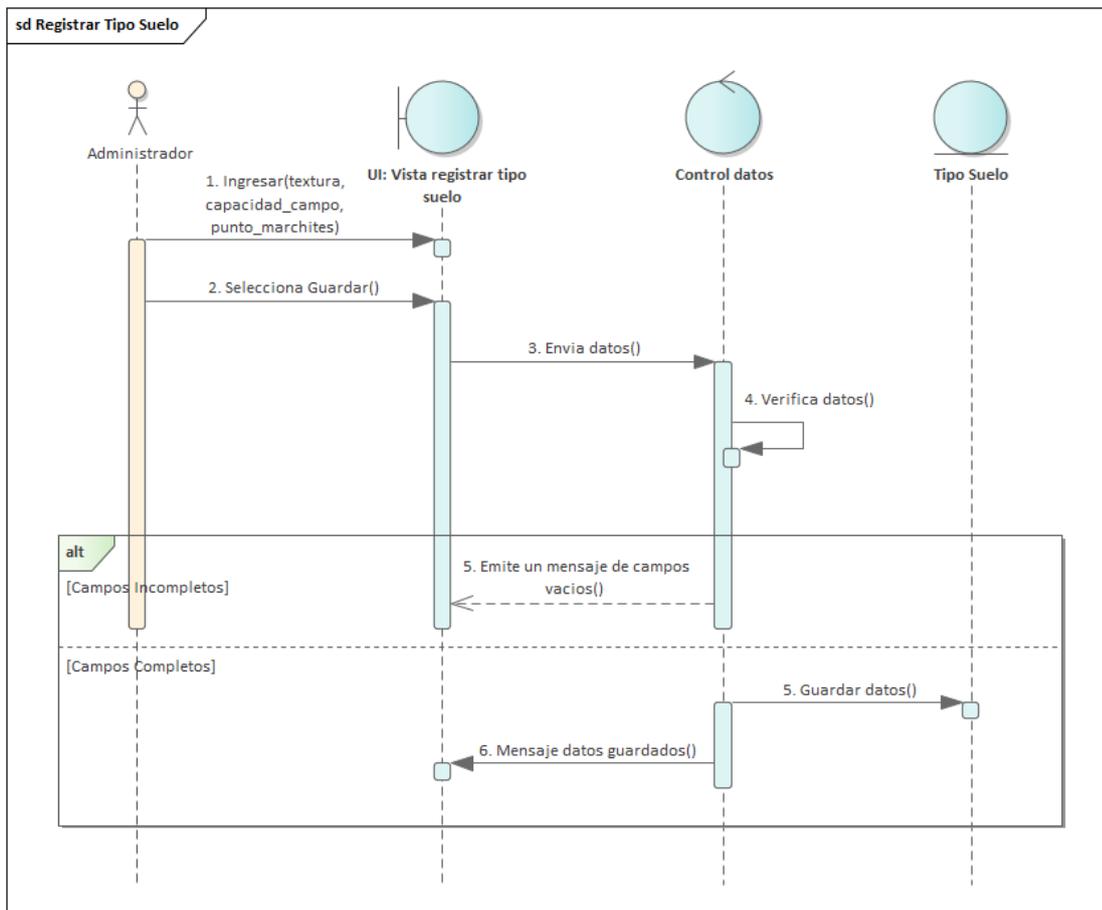


Figura 19: Diagrama de secuencia registrar tipo de suelo



Tabla 8: Descripción de caso de uso, registrar tipo de suelo.

Nombre:	Registrar Tipo de suelo
Descripción: El UC Registrar Tipo Suelo permite al Administrador registrar características de tipo de suelo en el sistema.	
Actores: Administrador	
Precondiciones: Haber iniciado sesión Estar en la interfaz registrar tipo de suelo	
Flujo Normal:	
Acciones del ADMINISTRADOR	Acciones del Sistema
1. Ingresar los datos (tipo suelo, capacidad de campo, punto de marchitez) en el formulario de la interfaz REGISTRAR SUELO	
2. Selecciona la opción [Registrar]	3. Válida los campos ingresados
	4. Guarda los datos de tipo de suelo
	5. Presenta un mensaje de confirmación “Tipo de suelo registrado correctamente”.
	6. fin de UC.
Flujo Alternativo:	
A. El ADMINISTRADOR selecciona la opción cancelar	
	A.3. Redirecciona a la interfaz PRINCIPAL
	A.4. Fin del UC

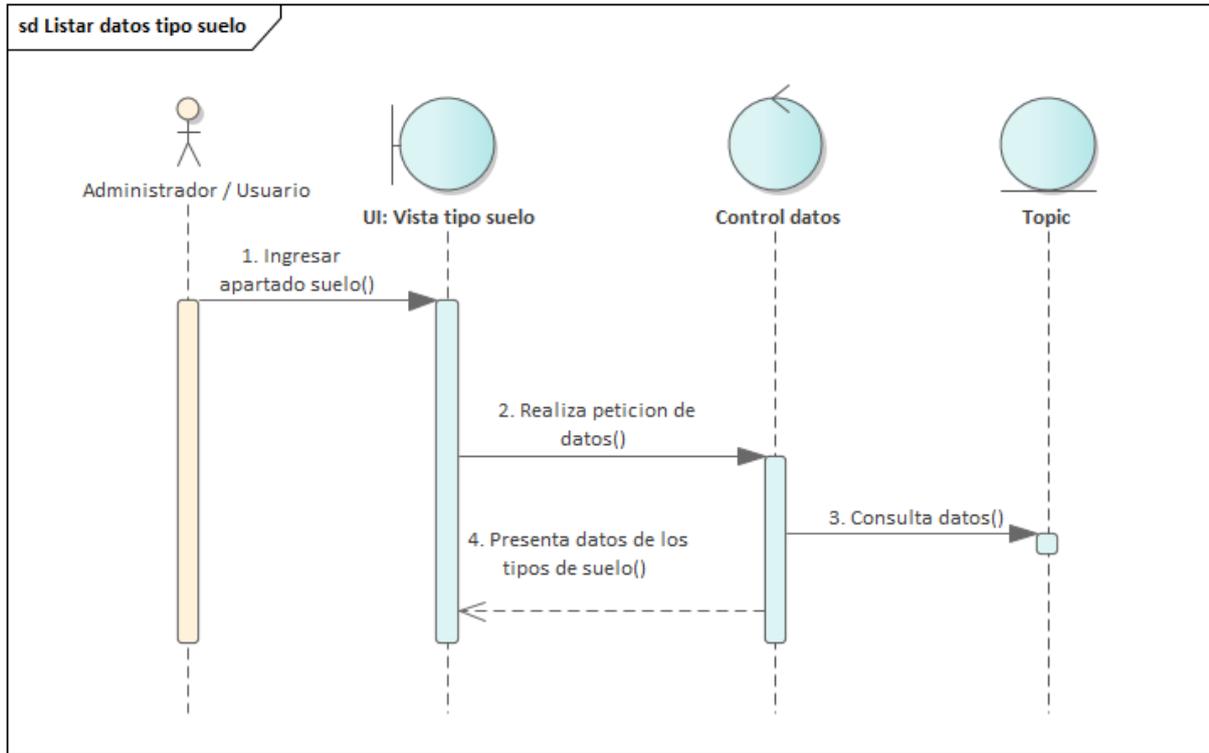


Figura 20: Diagrama de secuencia listar tipos de suelo

Tabla 9: Descripción de caso de uso, Listar tipo de suelo.

Nombre:	Listar Tipo Suelo
Descripción:	El UC Listar Tipo Suelo permite al Administrador listar los tipos de suelo registrados
Actores:	Administrador
Precondiciones:	Haber iniciado sesión Estar en la interfaz tipo de suelo
Flujo Normal:	
Acciones del ADMINISTRADOR	Acciones del Sistema
1. El sistema visualiza los datos de los tipos de suelo	2. El sistema busca los datos de los tipos de suelo
	3. El sistema carga los datos del tipo de suelo

	4. fin de UC.
--	---------------

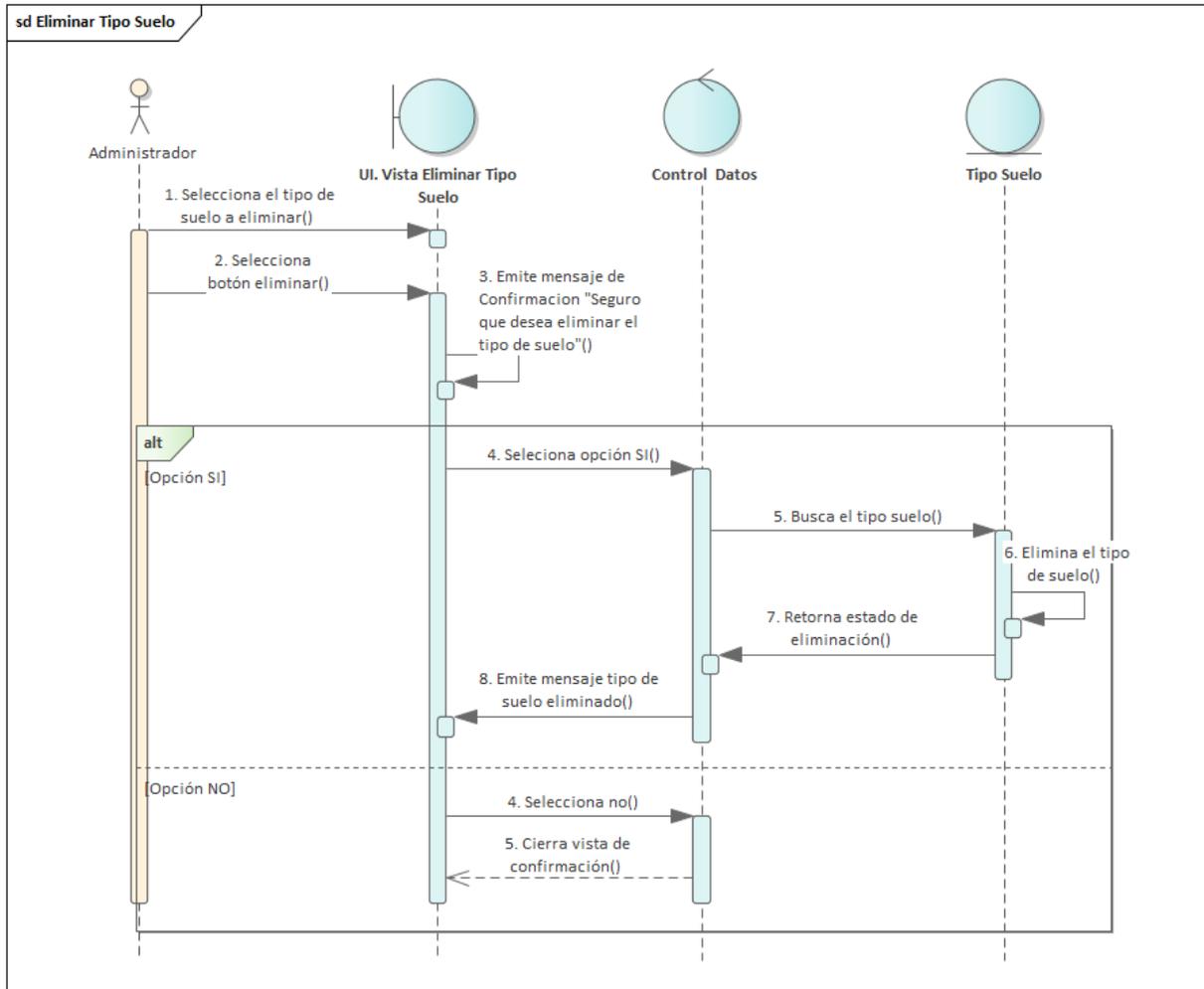


Figura 21: Diagrama de secuencia eliminar tipo de suelo

Tabla 10: Descripción de caso de uso, Eliminar tipo de suelo.

Nombre:	Eliminar Tipo de suelo
Descripción:	El UC Eliminar Tipo Suelo permite al Administrador eliminar características de tipo de suelo en el sistema.
Actores:	Administrador



Precondiciones: Haber iniciado sesión Estar en la interfaz Eliminar tipo de suelo	
Flujo Normal:	
Acciones del ADMINISTRADOR	Acciones del Sistema
1. Selecciona el botón eliminar	
	2. ¿Presenta mensaje de confirmación “Seguro que desea eliminar el tipo de suelo?”
3. Selecciona la opción SÍ	4. Busca el tipo de suelo
	5. Elimina el registro de tipo de suelo
	6. fin de UC.
Flujo Alternativo:	
A. El ADMINISTRADOR selecciona la opción NO	
	A.4. Redirecciona a la interfaz PRINCIPAL
	A.5. Fin del UC

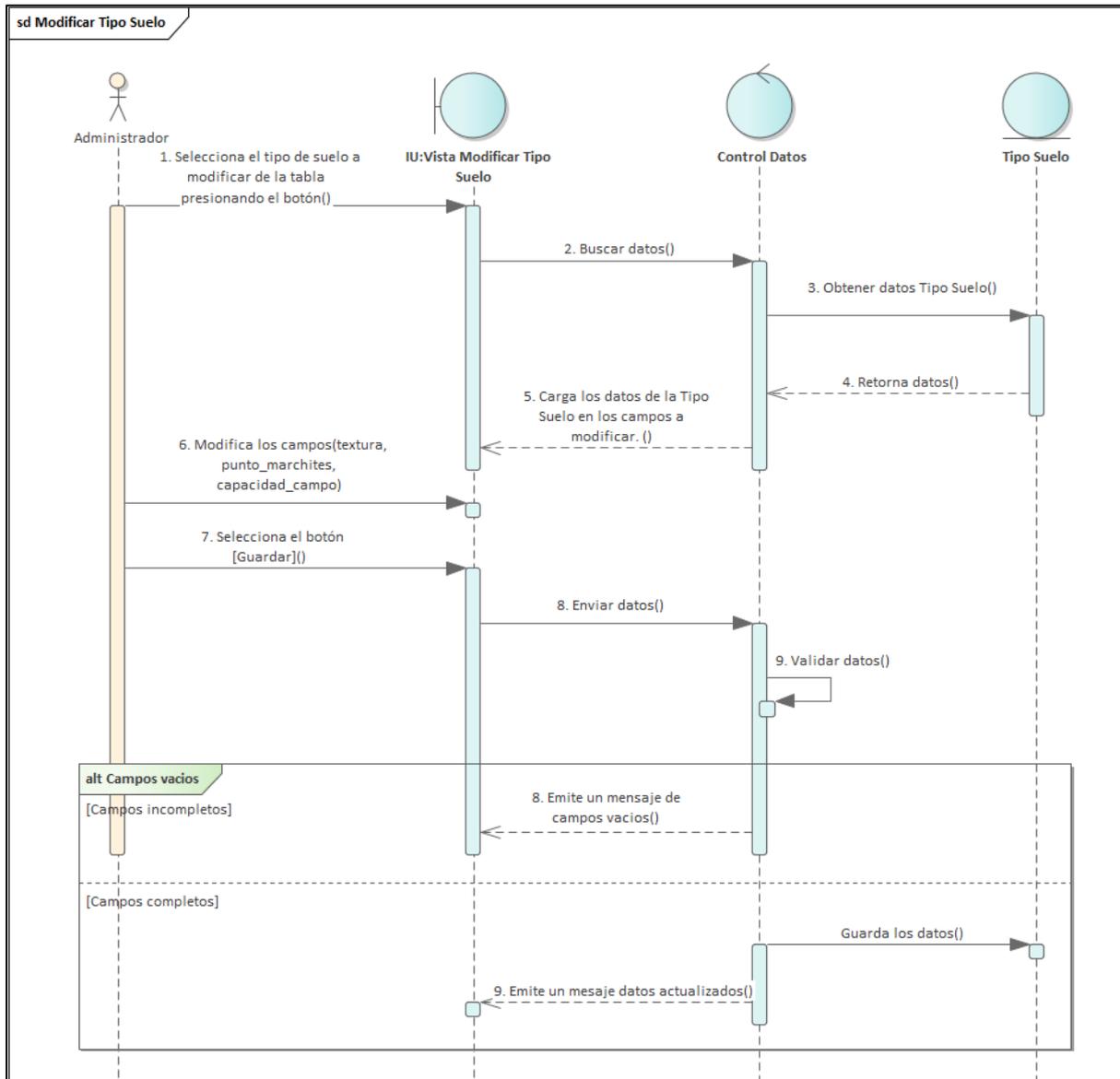


Figura 19: Diagrama de secuencia modificar tipo de suelo

Tabla 10: Descripción de caso de uso, modificar tipo de suelo.

Nombre:	Actualizar Tipo de suelo
Descripción:	El UC Actualizar Tipo Suelo permite al Administrador modificar características de tipo de suelo en el sistema.
Actores:	Administrador



Precondiciones: Haber iniciado sesión Estar en la interfaz tipo de suelo	
Flujo Normal:	
Acciones del ADMINISTRADOR	Acciones del Sistema
1. Selecciona el botón actualizar	
	2. Busca los datos del tipo de suelo a modificar
	3. Carga los datos en la vista actualizar
4. Actualiza lo datos necesarios a actualizar	
5. Seleccione el botón actualizar	6. Valida que lo campos estén correctos
	7. guarda nuevos datos de sensores
	8. presenta mensaje de confirmación “Tipo de suelo actualizado con éxito”
	9. fin de UC.
Flujo Alternativo:	
B. El ADMINISTRADOR selecciona la opción cancelar	
	A.6. Redirecciona a la interfaz PRINCIPAL
	A.7. Fin del UC

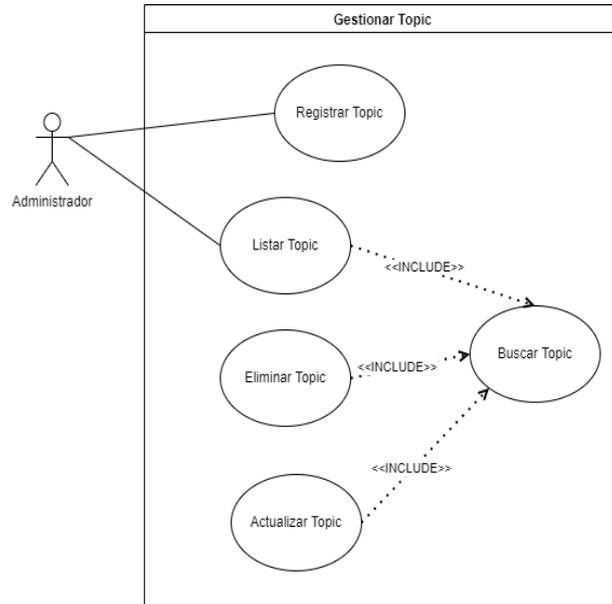


Figura 19: Caso de uso gestionar topic

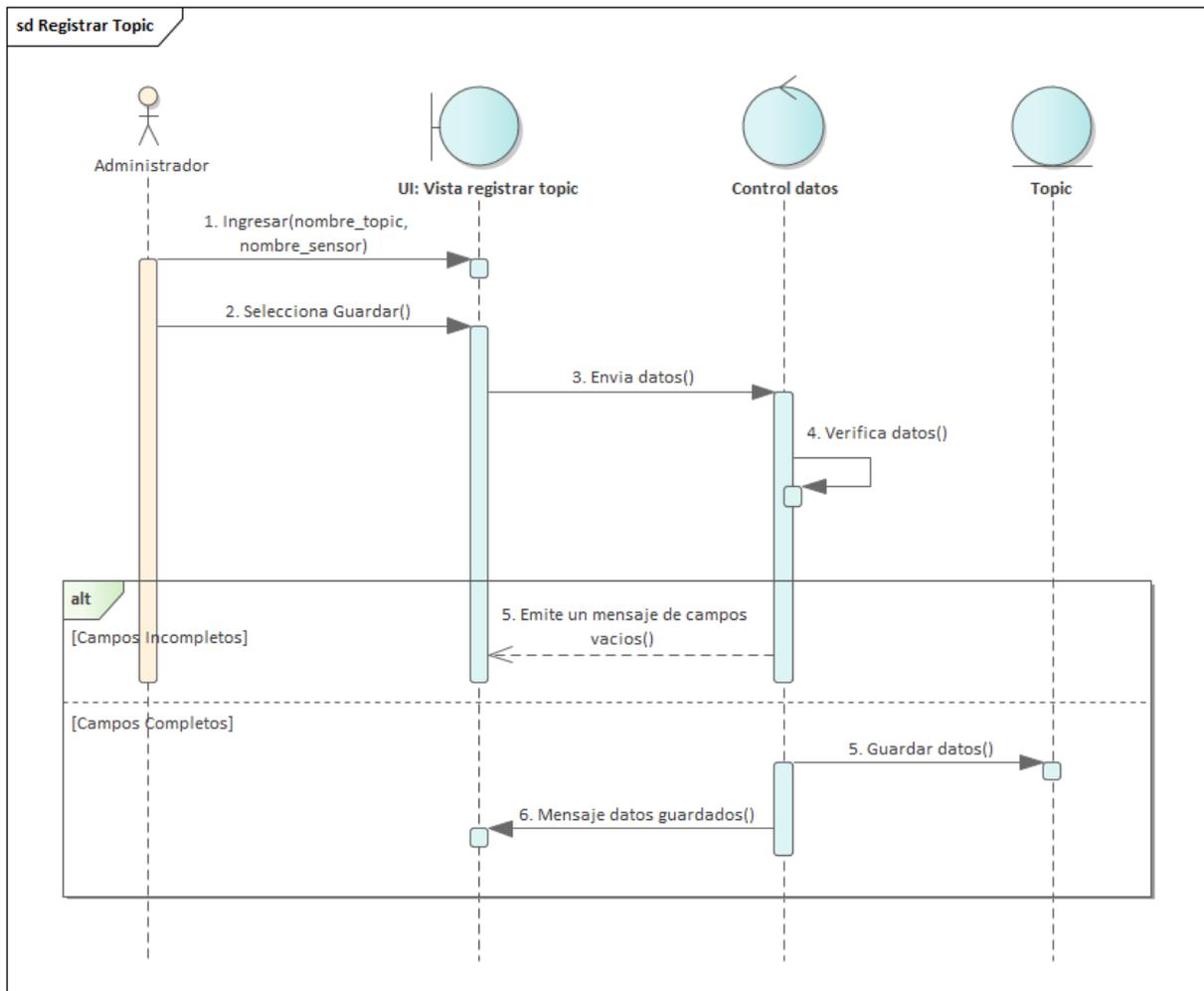


Figura 20: Diagrama de secuencia registrar topic



Tabla 11: Descripción de caso de uso, Registrar topic.

Nombre:	Registrar Topic
Descripción: El UC Registrar Topic permite al Administrador registrar topics o nombres de sensores en el sistema.	
Actores: Administrador	
Precondiciones: Haber iniciado sesión Estar en la interfaz registrar topic	
Flujo Normal:	
Acciones del ADMINISTRADOR	Acciones del Sistema
1. Ingresar los datos (nombre topic, nombre sensor) en el formulario de la interfaz REGISTRAR TOPIC	
2. Selecciona la opción [Registrar]	3. Válida los campos ingresados
	4. Guarda los datos de Topic
	5. Presenta un mensaje de confirmación "Topic registrado correctamente".
	6. fin de UC.
Flujo Alternativo:	
B. El ADMINISTRADOR selecciona la opción cancelar	
	A.3. Redirecciona a la interfaz PRINCIPAL
	A.4. Fin del UC

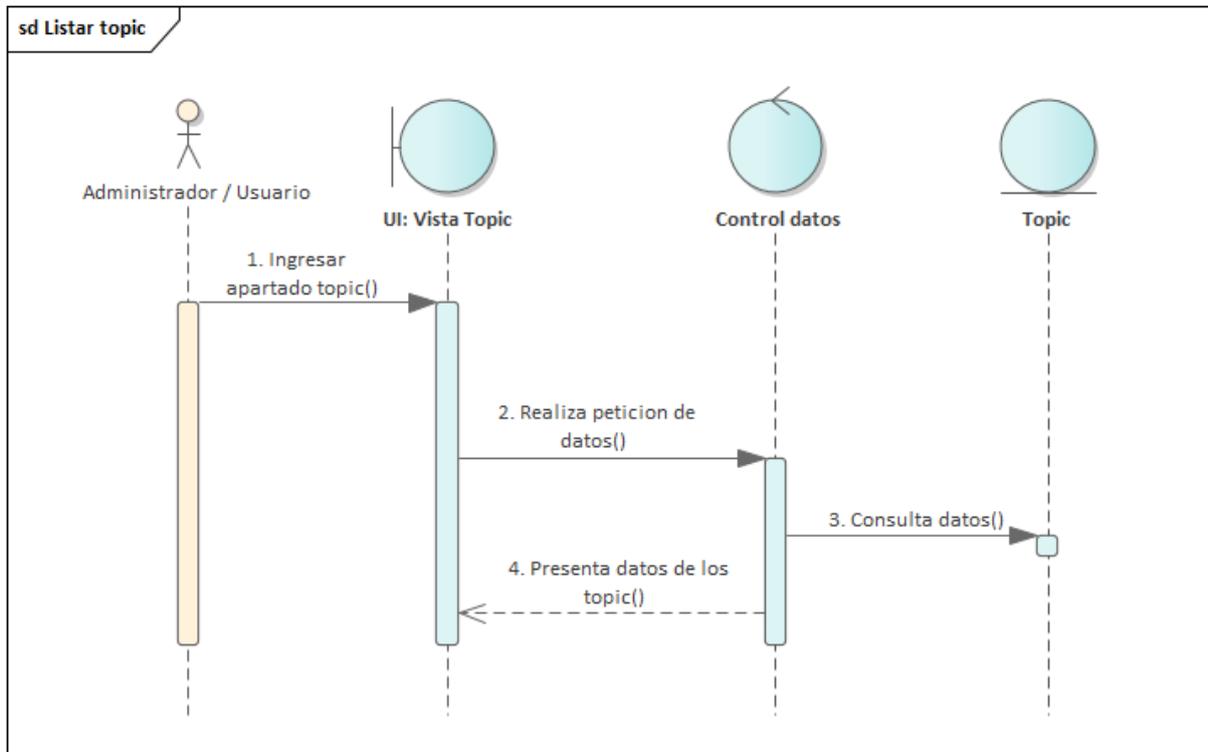


Figura 21: Diagrama de secuencia listar topic

Tabla 12: Descripción de caso de uso, Listar topic.

Nombre:	Listar Topics
Descripción:	El UC Listar Topic permite al Administrador listar los topics registrados
Actores:	Administrador
Precondiciones:	Haber iniciado sesión Estar en la interfaz topics
Flujo Normal:	
Acciones del ADMINISTRADOR	Acciones del Sistema
1. El sistema visualiza los datos de los tipos de suelo	2. El sistema busca los datos de Topics
	3. El sistema carga los datos de topics registrados
	4. fin de UC.

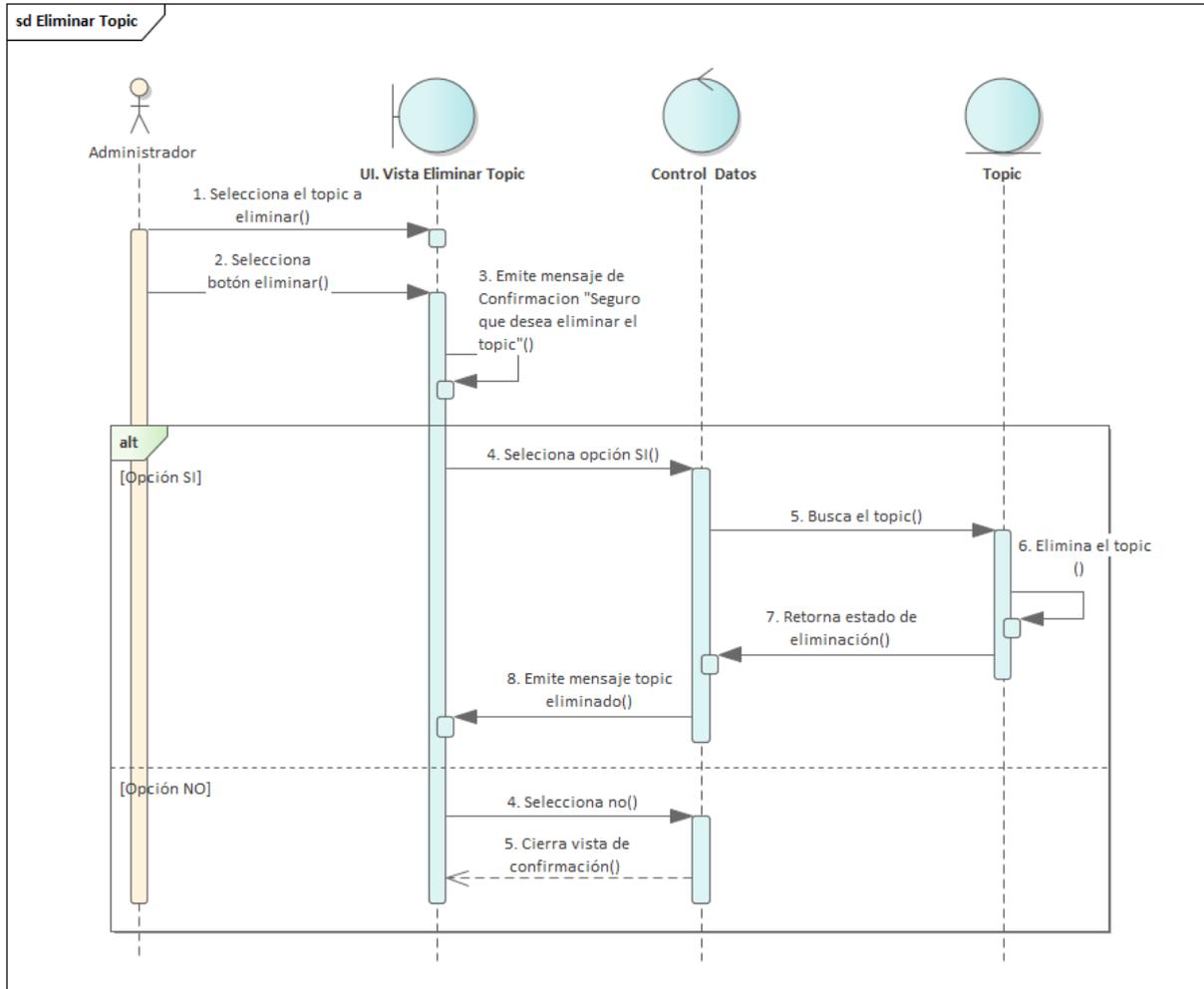


Figura 22: Diagrama de secuencia eliminar topic

Tabla 13: Descripción de caso de uso, Eliminar topic.

Nombre:	Eliminar Topic
Descripción:	El UC Eliminar Topic permite al Administrador eliminar topics en el sistema.
Actores:	Administrador



Precondiciones: Haber iniciado sesión Estar en la interfaz Eliminar Topic	
Flujo Normal:	
Acciones del ADMINISTRADOR	Acciones del Sistema
1. Selecciona el botón eliminar	
	2. ¿Presenta mensaje de confirmación “Seguro que desea eliminar el Topic?”
3. Selecciona la opción SÍ	4. Busca el Topic
	5. Elimina el registro de topic
	6. fin de UC.
Flujo Alternativo:	
C. El ADMINISTRADOR selecciona la opción NO	
	A.4. Redirecciona a la interfaz PRINCIPAL
	A.5. Fin del UC

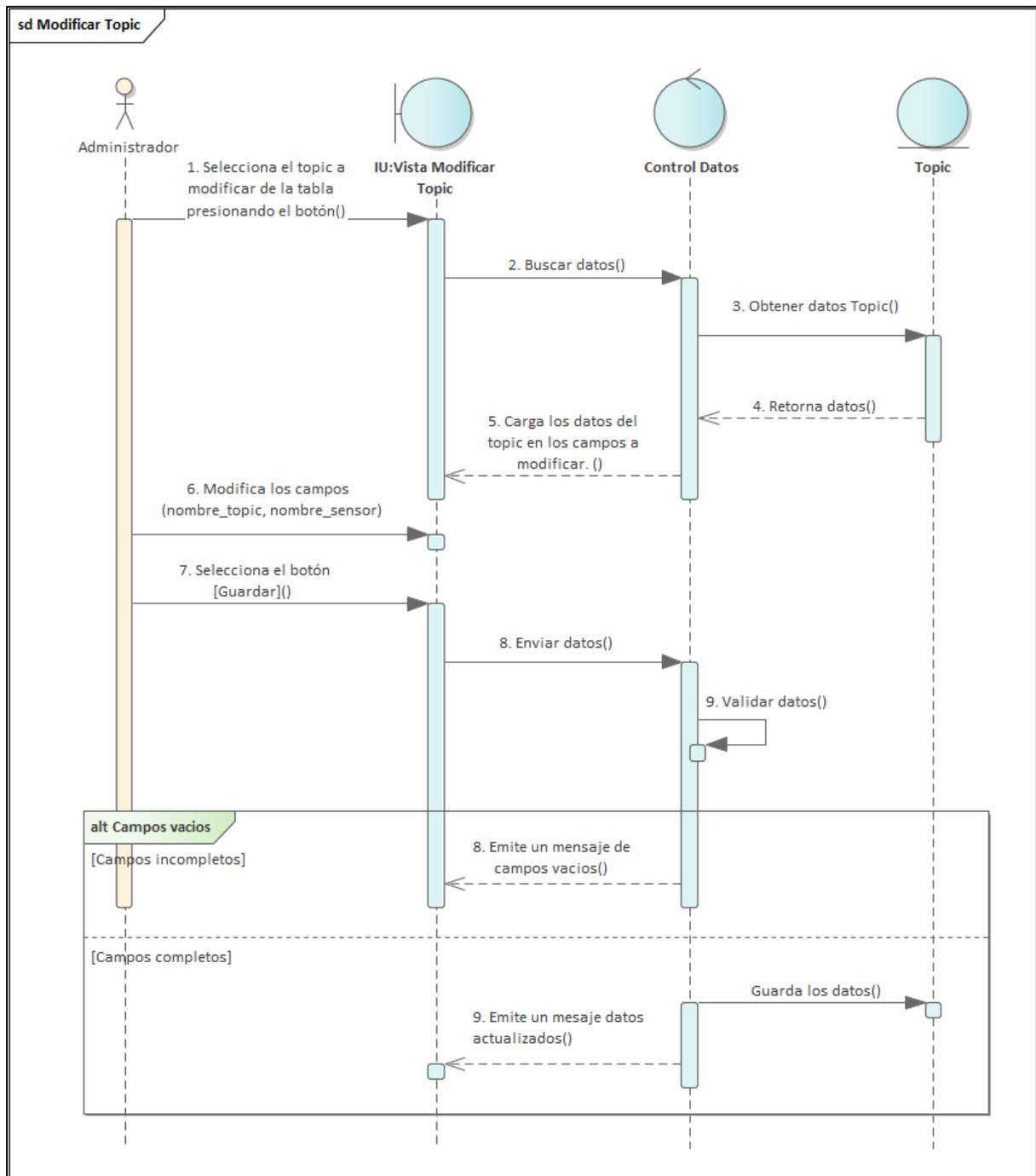


Figura 23: Diagrama de secuencia modificar topic

Tabla 14: Descripción de caso de uso, Actualizar topic.

Nombre:	Actualizar Topic
Descripción:	El UC Actualizar Topic permite al Administrador modificar los topic en el sistema.



Actores: Administrador	
Precondiciones: Haber iniciado sesión Estar en la interfaz tipo de suelo	
Flujo Normal:	
Acciones del ADMINISTRADOR	Acciones del Sistema
1. Selecciona el botón actualizar	
	2. Busca los datos del topic a modificar
	3. Carga los datos en la vista actualizar
4. Actualiza lo datos necesarios a actualizar	
5. Seleccione el botón actualizar	6. Valida que lo campos estén correctos
	7. guarda nuevos datos de sensores
	8. presenta mensaje de confirmación “Topic actualizado con éxito”
	9. fin de UC.
Flujo Alternativo:	
D. El ADMINISTRADOR selecciona la opción cancelar	
	A.6. Redirecciona a la interfaz PRINCIPAL
	A.7. Fin del UC

Vista. - Lógica

Diagramas. - Clases

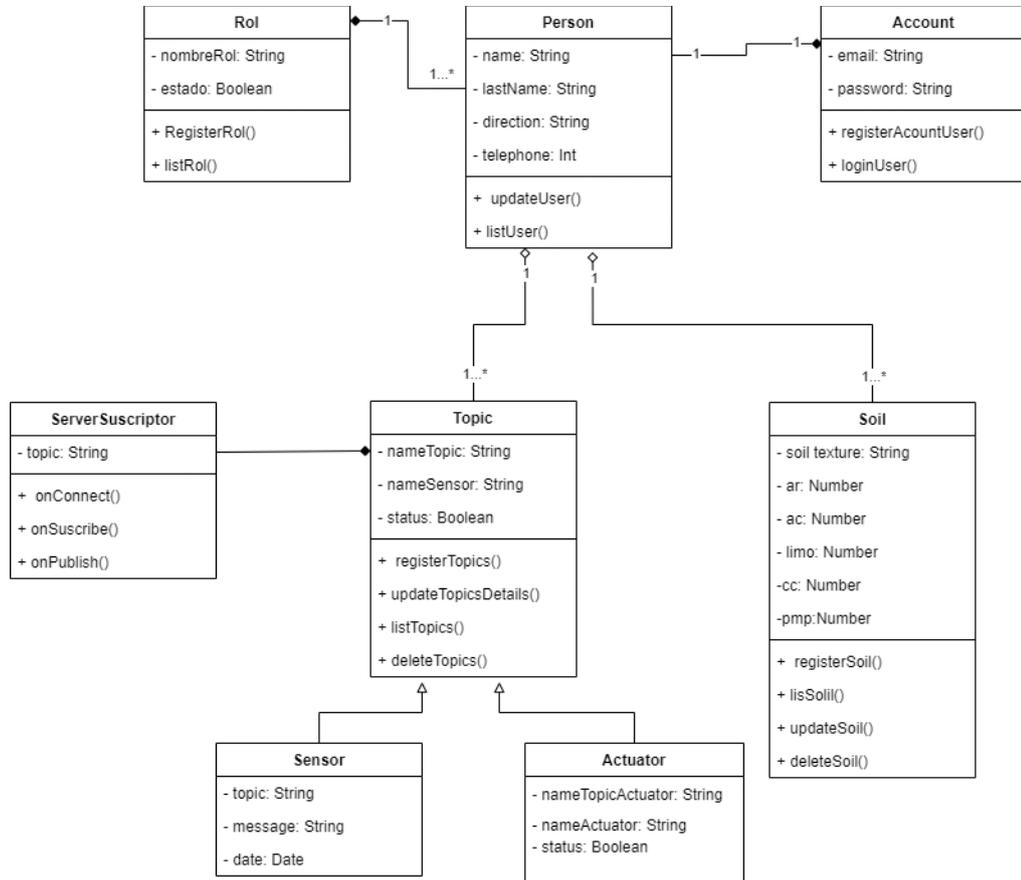


Figura 15: Diagrama de clases del sistema de irrigación



Vista. - Desarrollo

Se utilizó el diagrama de componentes para ilustrar el sistema desde la perspectiva del programador y de esta manera enfocarlo en la administración de los artefactos de software, por ende, nos permite describir los componentes del sistema.



Diagrama. – Componentes

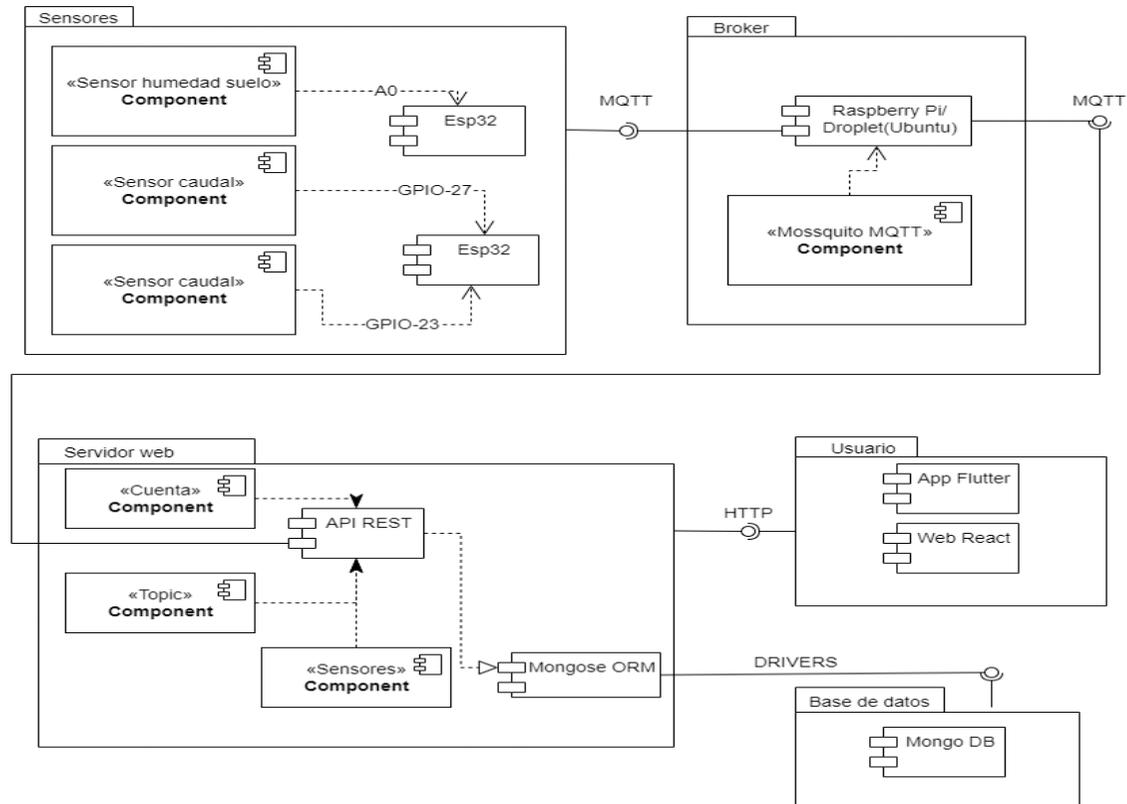


Figura 16: Diagrama de componentes del sistema de irrigación



Vista. - Física

La vista física describe los componentes de software en la capa física, así como las conexiones físicas entre los componentes.

- **Nodo 1: Esp32 Sensores**
 - <<FC-28>> Sensor de humedad
 - <<YF-S2001>> Sensor de caudal
- **Nodo 2: Esp32 Actuadores**
 - <<Relay>> Control de riego
- **Nodo 3: Raspberry Pi**
 - <<Mossquito>> Servidor MQTT
- **Nodo 4: Servidor Web**
 - <<NodeJs>> Servidor
- **Nodo 5: Servidor de base de datos**
 - <<Mongo DB>> Base de Datos
- **Nodo 6: Usuario web**
 - <<React Js>> Navegador Web
- **Nodo 7: Usuario móvil**
 - <<Flutter>> App

Diagrama. – Despliegue

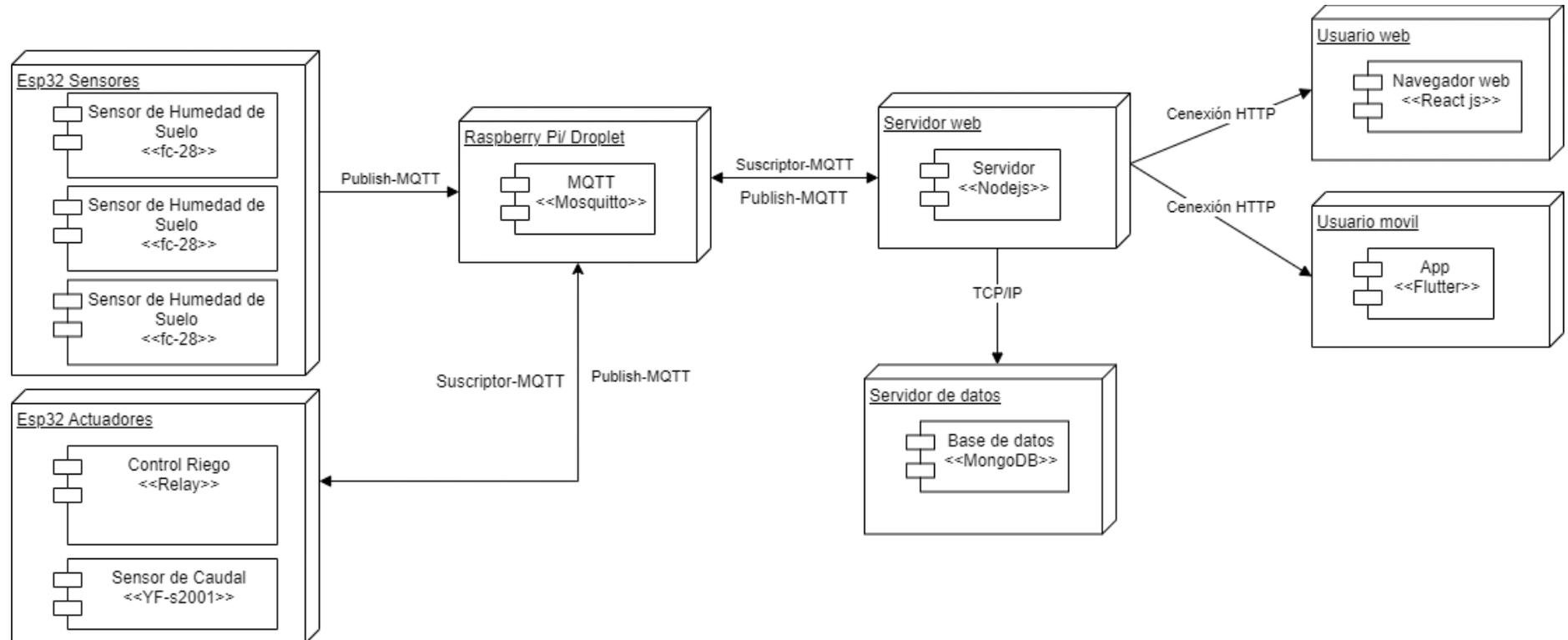


Figura 17: Diagrama de despliegue del sistema de irrigación

Diagrama Físico

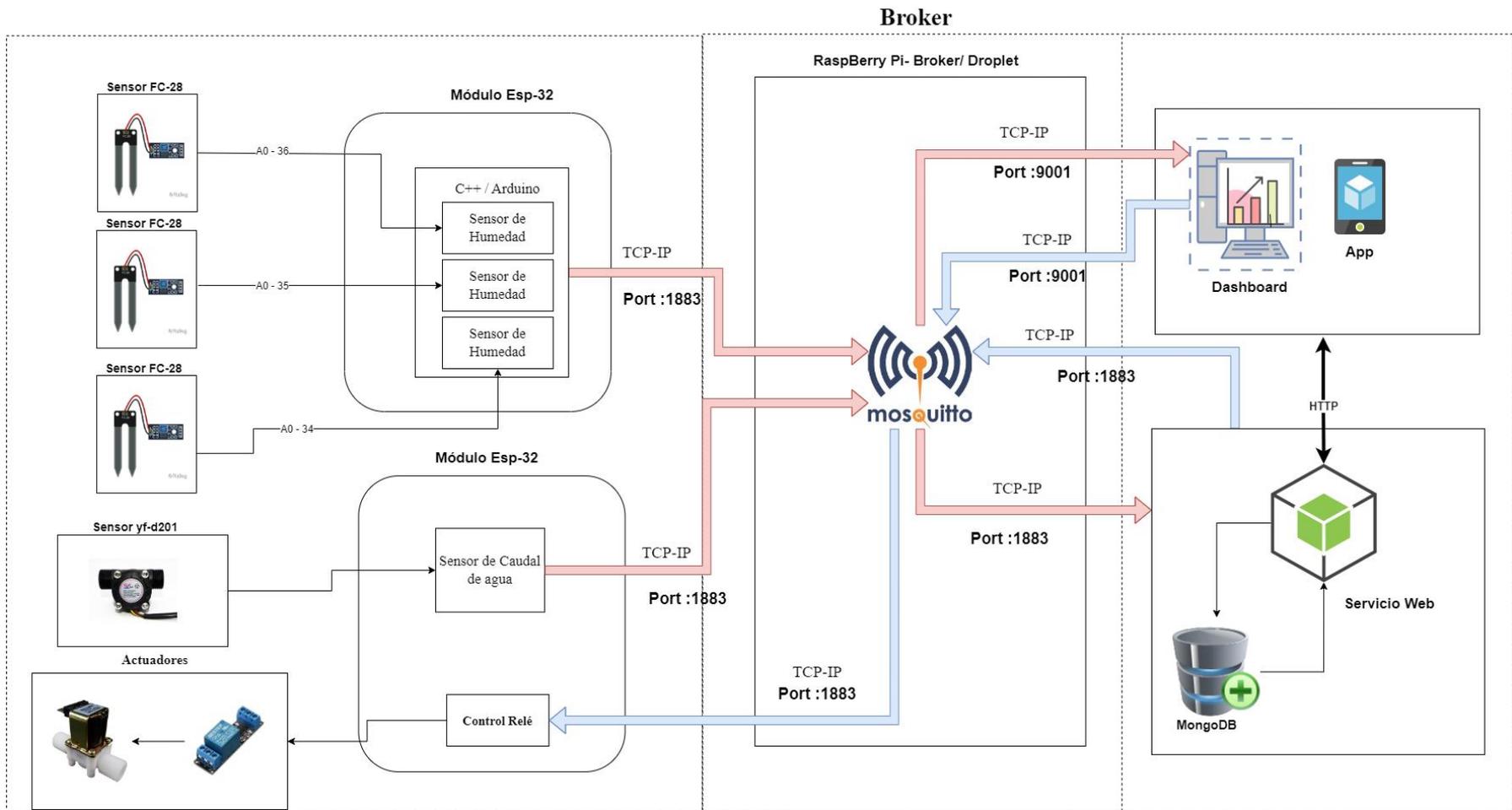


Figura 18: Diseño arquitectónico de IoT (Broker)

Diagrama arquitectónico:

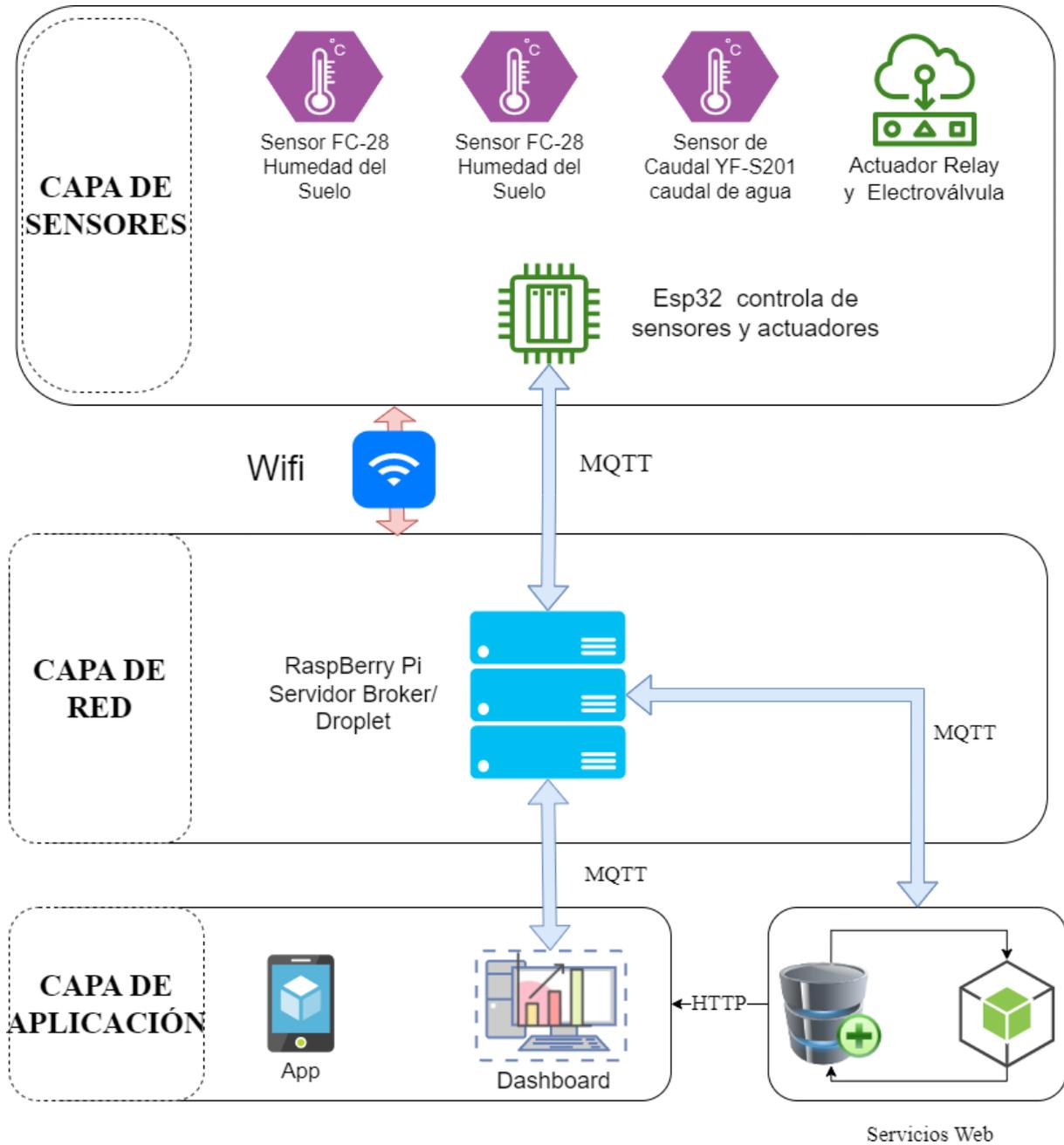


Figura 19: Diseño Arquitectónico de 3 capas IoT

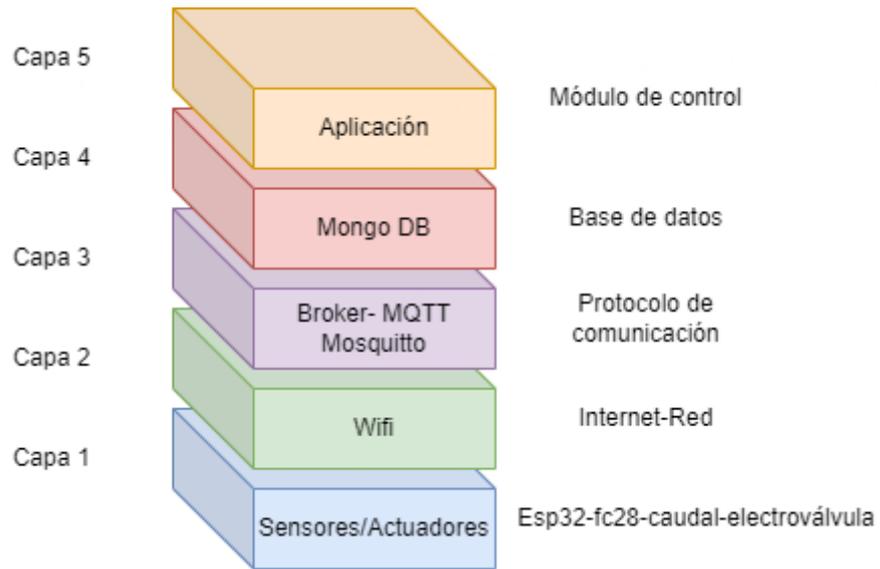


Figura 20 Arquitectura de Software para IoT

Anexo 5 Manual de prototipo

**FACULTAD DE ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES
CARRERA DE INGENIERÍA EN SISTEMAS / COMPUTACIÓN**

**MANUAL DE PROTOTIPO PARA “EL SISTEMA DE
IRRIGACIÓN CON IoT PARA CULTIVOS DE LA
UNIVERSIDAD NACIONAL DE LOJA”**

Estudiantes:

- Alex Santiago Nole Reyes
 - Angel Favian Minga Medina
-

Control del documento

1. Introducción	203
1.1. Propósito	203
2. Definiciones, acrónimos y abreviaciones.....	203
3. Herramientas Utilizadas.....	203
4. Construcción del Prototipo de Sistema de Irrigación para Cultivos de La Universidad Nacional de Loja.....	204
4.1. Configuración del servidor o Broker en Raspberry Pi	204
4.1.2. <i>Habilitar acceso Remoto con Autenticación en broker mosquito</i>	205
4.2. Programación del suscriptor publicador o módulo Esp32 para sensores	207
4.2.1. <i>Librerías utilizadas para la programación del Esp32</i>	207
4.3. Conexión de Sensores y actuadores a modulo Esp32.	209
4.4. Programación del suscriptor publicador o módulo Esp32 para actuadores	212
4.5. Conexión de broker con API-REST	214
4.6. Conexión con aplicativo Front-end y aplicación Móvil	216
5. Anexos	217

1. Introducción

1.1. Propósito

Este documento detalla los pasos que se siguieron para la configuración de servidores, sensores, actuadores y conexión con el aplicativo web y móvil para la solución informática “Sistema de irrigación para cultivos de la Universidad Nacional de Loja”

2. Definiciones, acrónimos y abreviaciones.

- **IoT:** Internet de las Cosas, interactúa Hardware, software y redes
- **Autenticación:** Es cuando el usuario se identifica en forma inequívoca; es decir, sin duda o equivocación de que es quien dice ser.
- **MQTT:** Protocolo de Transporte de telemetría de Mensaje Server (Message Queuing Telemetry Transport) utilizado para la comunicación con los sensores y el microcontrolador
- **Broker:** Broker o servidor es el encargado de transmitir los mensajes recibidos por el publicador al suscriptor
- **Publicador:** Es aquel que emite los datos obtenidos por los sensores.
- **Suscriptor:** Es donde se presentan los datos obtenidos de los sensores.
- **Módulo:** Es un componente de software o parte de un programa que contiene una o más rutinas.
- **React:** Librería de JavaScript para el desarrollo de frontend
- **Flutter:** Framework para el desarrollo de aplicaciones móviles
- **NodeJs:** Framework del lado del servidor para la creación de servicios del sistema
- **MongoDB:** Base de datos no relacional para el almacén de datos.
- **Raspberry PI:** Microcomputador utilizado como servidor MQTT
- **Mosquitto:** Servidor MQTT instalado en raspberry Pi
- **Esp32:** Tarjeta programable que contiene un Microcontrolador utilizado para la comunicación entre sensores y MQTT

3. Herramientas Utilizadas

- **Raspberry pi:** Microcomputador que servirá como servidor o broker, el cual servirá de intermediario entre el publicador y suscriptor mediante el protocolo MQTT.

- **Esp32:** Se utiliza para la conexión de sensores, además es el que se conecta con el broker para publicar los datos recibidos por los sensores, así mismo, permitirá enviar información al actuador para que abra y cierre el riego.
- **Sensor de suelo FC-28:** Es el encargado de recolectar la humedad del suelo y encaminarla al módulo Esp32.
- **Sensor de caudal YF-S201:** Es el encargado de medir la cantidad de agua utilizada cuando se abre el riego
- **Electroválvula:** Permite abrir el y cerrar el riego cada vez que el relé le envíe voltaje
- **Relé:** Es aquel que recibe la señal enviada por los usuarios para encender el riego.

4. Construcción del Prototipo de Sistema de Irrigación para Cultivos de La Universidad Nacional de Loja.

4.1. Configuración del servidor o Broker en Raspberry Pi

Para el desarrollo del sistema de irrigación con IoT para los cultivos de la Universidad Nacional de Loja se procedió primeramente a realizar la conexión del broker con la herramienta mosquitto en el microcomputador RaspberryPi para lo cual se realizaron los siguientes pasos:

4.1.1. Instalación de Mosquitto Broker

Los comandos utilizados para la instalación de mosquitto broker en la raspberrypi, se los debe ejecutar en el orden que se muestra en el presente documento:

Primeramente, se debe ingresar los comandos de actualización para la raspberrypi, cabe destacar que el sistema operativo utilizado se denomina RaspBian basado en el sistema operativo Debian, para lo cual permite actualizar con los mismos comandos:

sudo apt update && sudo apt dist – upgrade

Para la instalación de mosquitto broker se utilizó el siguiente comando:

sudo apt install – y mosquitto mosquitto – clients

Seguidamente para que mosquitto se inicie automáticamente cuando se inicie el raspberrypi se hace uso del comando:

sudo systemctl enable mosquitto.service

Seguidamente para verificar que la instalación es correcta y que se encuentra habilitado el servicio utilizamos el comando:

mosquitto -v

En la Figura 1 se presenta la salida en consola del servicio de mosquitto en ejecución

```
root@Broker-new:~# mosquitto -v
1672884307: mosquitto version 2.0.11 starting
1672884307: Using default config.
1672884307: Starting in local only mode. Connections will only be possible from
clients running on this machine.
1672884307: Create a configuration file which defines a listener to allow remote
access.
1672884307: For more details see https://mosquitto.org/documentation/authentication-methods/
1672884307: Opening ipv4 listen socket on port 1883.
1672884307: Error: Address already in use
1672884307: Opening ipv6 listen socket on port 1883.
```

Figura 64 Servicio de mosquitto en ejecución

4.1.2. Habilitar acceso Remoto con Autenticación en broker mosquitto

Para habilitar el acceso remoto para conectarse al broker se lo puede realizar de dos formas, una con autenticación con la cual se realizado el proyecto y la forma sin autenticación que permite que cualquier suscriptor o publicador se conecte sin necesidad de credenciales.

Para habilitar el acceso remoto con autenticación se debe utilizar los siguientes comandos:

Primeramente, se va a crear un usuario y contraseña con la cual se podrá acceder al servidor para lo cual se utiliza el comando:

`sudo mosquitto_passwd -c /etc/mosquitto/passwd YOUR_USERNAME`

Luego de ingresar este comando le pedirá que ingrese una contraseña, como se muestra en la Figura 2.

```
root@Broker-new:~# sudo mosquitto_passwd -c /etc/mosquitto/passwd riego
Password:
Reenter password:
root@Broker-new:~#
```

Figura 65 Configuración de credenciales a servidor mosquitto

Ahora se tiene que editar el archivo de configuración, para que permita a los suscriptores y publicadores conectarse con las credenciales que se crearon anteriormente para lo cual se debe ingresar:

sudo nano /etc/mosquitto/mosquitto.conf

Luego de ejecutar el comando anterior nos abrirá el archivo de configuración de mosquitto el cual por defecto contiene la configuración que se muestra en la Figura 3.

```
GNU nano 6.4 /etc/mosquitto/mosquitto.conf
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

pid_file /run/mosquitto/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d
```

Figura 66 Archivo de configuración de mosquitto por defecto

En la Figura 4 se muestra la configuración final del archivo de configuración del broker en donde se agregó que las conexiones no sean anónimas, que se escuche en el puerto 1883 y que las credenciales están configuradas en el archivo **pwfile**.

```
GNU nano 6.4 /etc/mosquitto/conf.d/default.conf
#per_listener_settings true
allow_anonymous false
listener 1883
#protocol mqtt
listener 9001
protocol websockets
socket_domain ipv4
password_file /etc/mosquitto/passwd
```

Figura 67 Archivo de configuración actualizado

Luego de ingresar la nueva configuración al nuestro servidor lo tenemos que reiniciar usando el comando:

sudo systemctl restart mosquitto

Para verificar que todo esté funcionando correctamente se puede verificar el estado del servidor con el comando:

sudo systemctl status mosquitto

En la Figura 5 se muestra el estado del servidor mosquitto ejecutándose correctamente

```
root@Broker-new:~# sudo systemctl status mosquitto
● mosquitto.service - Mosquitto MQTT Broker
  Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; preset: en
  Active: active (running) since Thu 2023-01-05 02:03:38 UTC; 8min ago
    Docs: man:mosquitto.conf(5)
          man:mosquitto(8)
  Main PID: 18538 (mosquitto)
    Tasks: 1 (limit: 512)
  Memory: 1.9M
    CPU: 373ms
  CGroup: /system.slice/mosquitto.service
          └─18538 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

Jan 05 02:03:38 Broker-new systemd[1]: Starting Mosquitto MQTT Broker...
Jan 05 02:03:38 Broker-new systemd[1]: Started Mosquitto MQTT Broker.
lines 1-14/14 (END)
```

Figura 68 Status de servicio de Mosquitto

4.2. Programación del suscriptor publicador o módulo Esp32 para sensores

Luego de haber configurado el servidor mosquitto en raspberrypi se procede a realizar la programación del módulo Esp32 para controlar los sensores y a su vez enviar la información al servidor mosquito, en este caso el Esp32 va a ser configura como como publicador, ya que este va a enviar información receptada por los sensores.

4.2.1. Librerías utilizadas para la programación del Esp32

Como se muestra en la Figura 6 se muestran las librerías utilizadas para la conexión del broker (PubSubCLiet), para la conexión a la red se utiliza la librería WiFi.

```
#include <Arduino.h>
#include "PubSubClient.h" // Conectar y publicar con el broker
#include "WiFi.h" // Enables the ESP32 to connect to the local network (via WiFi)
#define TIME_DELAY 2000 // Cada cuánto leer del sensor
#define SENSOR 27 //sensor de caudal GPIO 27
```

Figura 69 Librerías necesarias para sensores, cliente mqtt y Wifi

Seguidamente se declaran los puertos GPIO a los actuales van a ir conectados los sensores, cabe destacar que para los sensores FC-28 se los debe conectar a los puertos analógicos para tener un mejor control sobre la humedad del suelo. EL esp32 tiene 16 puertos analógicos con voltajes de 0-3.3v que traducido a los sensores va de 0-4095 cuando está saturado, en la Figura 7 se presenta la declaración de los puertos analógicos a los que va conectados los sensores de humedad del suelo.

```

/**
  Variables Globales suelo
*/
const int SensorPin = 36; //Pin GPIO-36 es pin Analogico de 0-3.3V o 5V -> 0 - 4095 SP
const int SensorPin1 = 34; //Pin GPIO-36 es pin Analogico de 0-3.3V o 5V -> 0 - 4095 SP
const int SensorPin2 = 35; //Pin GPIO-36 es pin Analogico de 0-3.3V o 5V -> 0 - 4095 SP

```

Figura 70 Declaración de puertos GPIO de sensores con Esp32

A continuación, se declaran las variables globales y credenciales para la conexión a la red wifi y broker para realizar la comunicación con publicador y servidor broker. Como se encuentra en la Figura 8 se declaran el ssid y contraseña de la red a la que se va a conectar, la dirección del servidor broker, así como los topic. Es importante decir que los topic es el nombre del sensor con el cual se va a reconocer en la red y con estos se conectan los suscriptores para verificar los datos.

```

/**
  Variables para configuracion de wifi y broker
*/
// WiFi
char* ssid = "Nettplus Electro"; // Nombre de la red del sembrio
const char* wifi_password = "1105639015Kevin"; // CONtraseña de la red

// MQTT configuracion con el servidor Raspberry Pi
const char* mqtt_server = "192.168.0.104"; // Direccion Ip del Broker o Raspberry Pi
//const char* temperature_topic = "home/sembriol/temperature";
const char* humedytySolid_topic = "suelo1";
const char* humedytySolid_topic1 = "suelo2";
const char* humedytySolid_topic2 = "suelo3";|
const char* mqtt_username = "riego"; // Nombre del Usuario del MQTT configurado en el broker
const char* mqtt_password = "3435"; // Nombre del Contraseña del MQTT configurado en el broker
const char* clientID = "riego";
const int mqtt_port = 1883;

```

Figura 71 Credenciales Wifi y Mqtt para la conexión

Seguidamente en la función void setup() se realiza la conexión con el broker y con la red wifi, esto para que cuando inicie el esp32 se conecte automáticamente. En la Figura 9 se presenta el método para conexión a la red y al broker.

```

} void setup() {
  Serial.begin(115200);

  // Conexión WiFi
  WiFi.begin(ssid, wifi_password);
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.println('....');
    delay(500);
  }

  client.setServer(mqtt_server, mqtt_port);
  while (!client.connected())
  {
    if (!client.connect("ClienteESP32", mqtt_username, mqtt_password))
    {
      Serial.print("Conectado ");
      Serial.println(mqtt_username);
      delay(2000);
    }
  }
}
}

```

Figura 72 Código dentro de función setup, para conexión Wifi y broker

Finalmente se realiza la configuración de los sensores de humedad del suelo en el loop para que reciban datos continuamente, aquí se define la lectura de los sensores y se agrega una función map para presentar valores en porcentaje entendibles para enviar al broker.

4.3. Conexión de Sensores y actuadores a modulo Esp32.

A continuación, se muestran cómo se realizó la conexión de los sensores de suelo al Esp32.

Para realizar la conexión correctamente, procedimos a comprender como es el diagrama de puertos del esp32. En la Figura 10 se muestra los puertos y pines de forma entendible para su posterior conexión.

ESP32 DEVKIT V1 – DOIT

version with 36 GPIOs

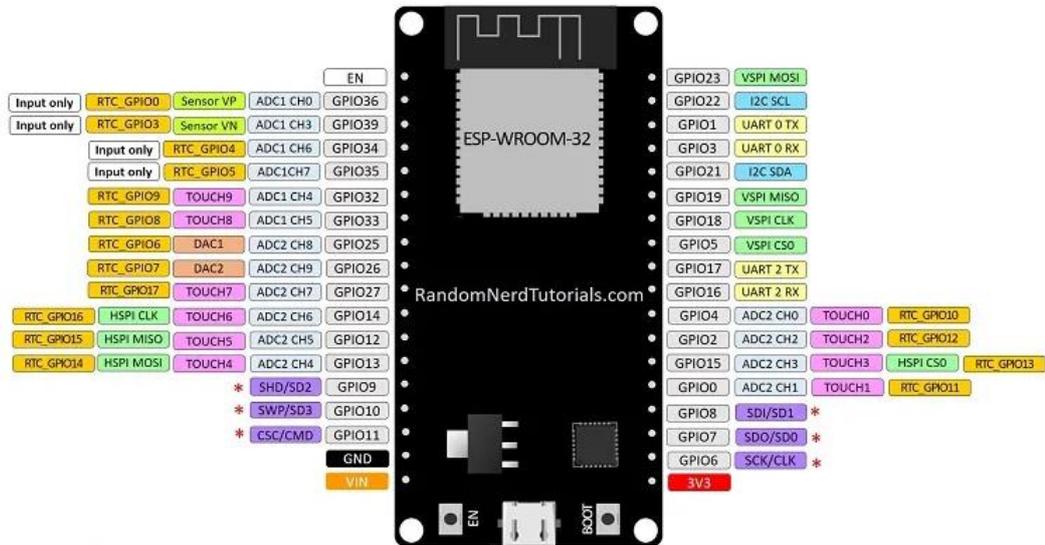


Figura 73 Diagrama de puertos del MCU Esp32

Luego de entender como están definidos los pines en el Esp32, se realizó la simulación de conexión en fritzig, como se muestra en la Figura 11, se puede observar la conexión de todos los sensores y actuadores al esp32, con lo cual se procedió a realizar la conexión física.

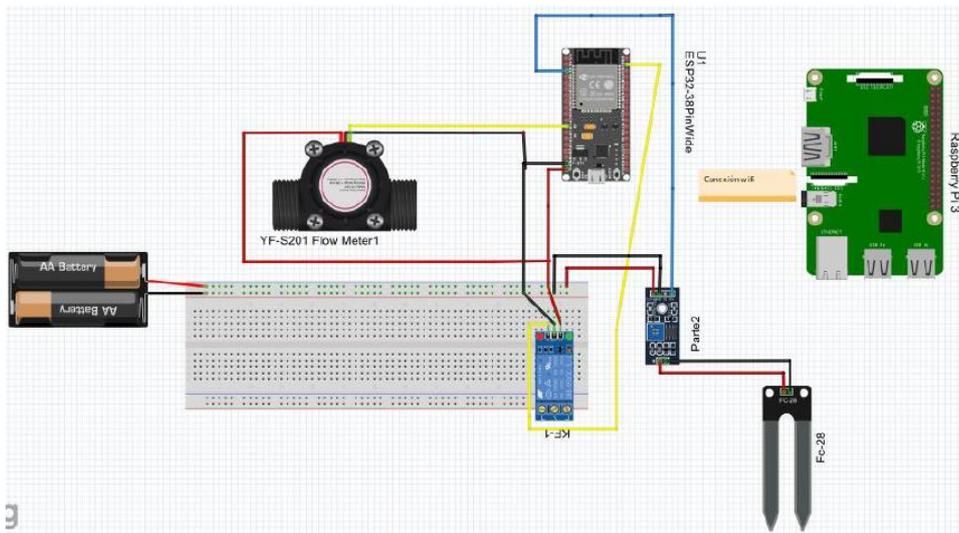


Figura 74 Conexión en fritzig de sensores y actuadores del sistema de irrigación

Cabe destacar que para la conexión de los sensores FC-28, para la humedad de suelo se los conecto a los puertos GPIO 34, 35, 36 que son puertos analógicos. Para la conexión del sensor de caudal se lo realizo al puerto GPIO 27 y el relé al puerto GPIO 23, para el

envió de datos. Todos los componentes están conectados a 3.3V a excepción del relé que está conectado a 5V.

En la Figura 12, se puede visualizar la conexión final de los sensores de humedad de Suelo al módulo Esp32, y en la Figura 13 se muestra como quedo la conexión final al módulo Esp32 para actuadores.

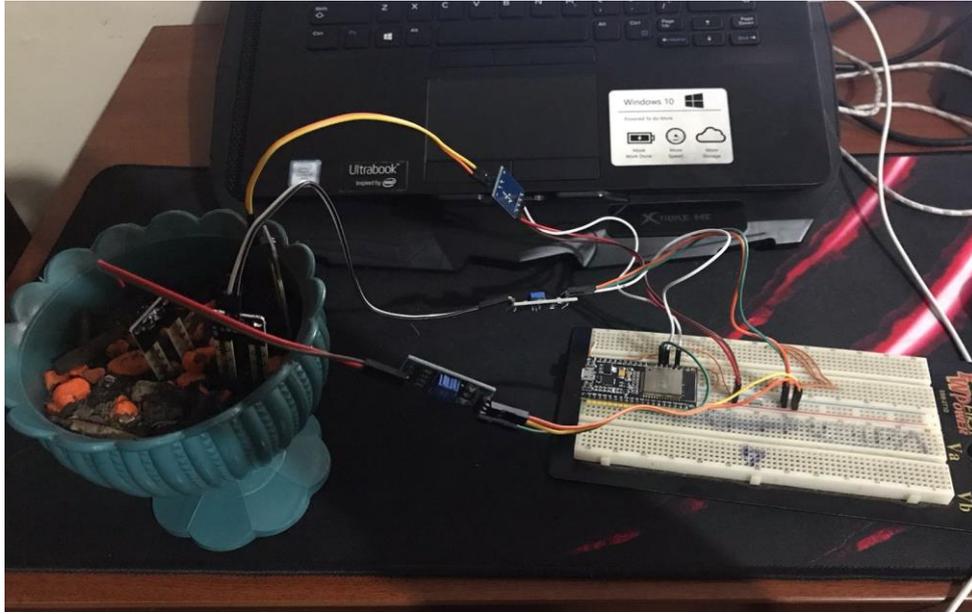


Figura 75 Conexión física de sensores de humedad de suelo

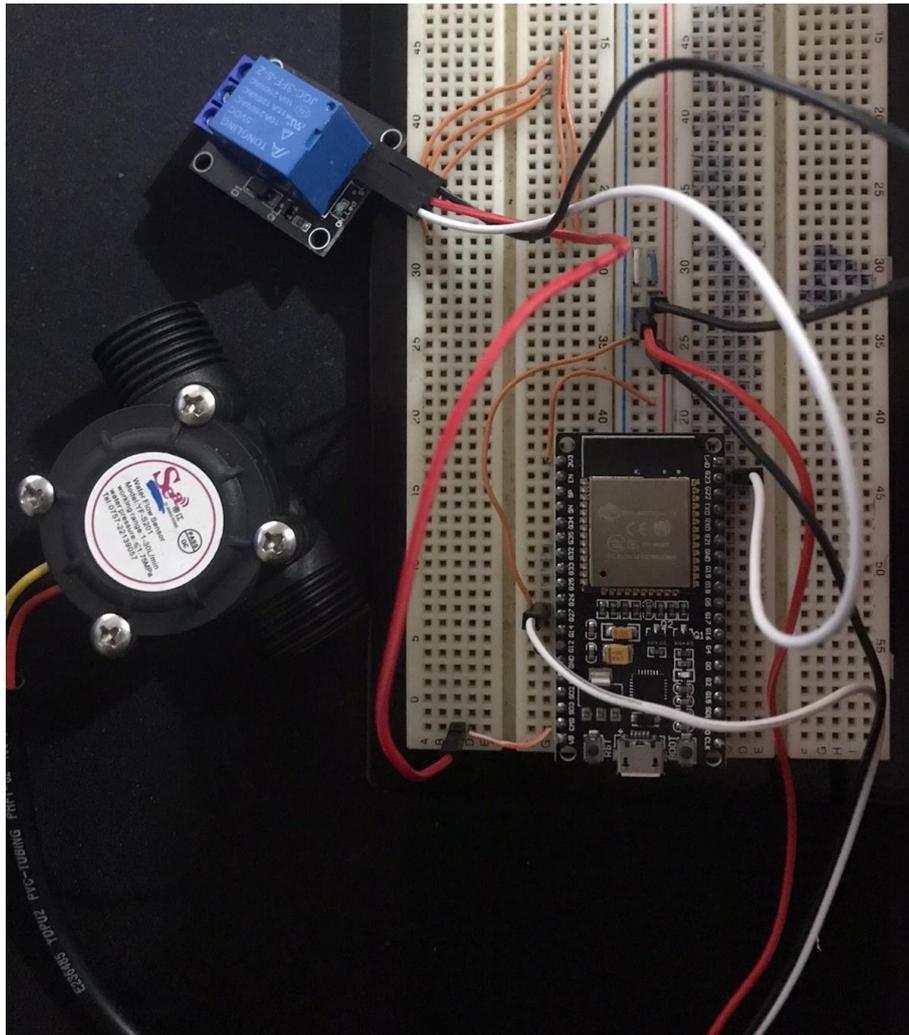


Figura 76 Conexión de sensor de caudal y relé para habilitar riego

4.4. Programación del suscriptor publicador o módulo Esp32 para actuadores

Para la programación del actuador y el sensor de caudal, se utilizó la misma configuración que el módulo Esp32 para sensores, aquí la diferencia se resalta en la configuración de los pines y se añadió una función para el callback que permite recibir información desde el broker. En la Figura 14 se muestra las variables y puertos utilizados para la conexión del relé y el sensor de caudal

```

#include <Arduino.h>
#include "PubSubClient.h" // Conectar y publicar con el broker
#include "WiFi.h" // Enables the ESP32 to connect to the local network (via WiFi)
#define TIME_DELAY 2000 // Cada cuánto leer del sensor
#define SENSOR 27 //sensor de caudal GPIO 27
/**
 * Variables sensores
 */
const int outPin = 23; //Pin GPIO-23 Para control de Relay
String outState = "Apagado";
String dataSensor;
String dataSensor1;
String dataSensor2;
/**
 * Valores de calibracion de sensor de caudal
 */
long currentMillis = 0;
long previousMillis = 0;
int interval = 1000;
boolean ledState = LOW;
float calibrationFactor = 4.5;
volatile byte pulseCount;
byte pulse1Sec = 0;
float flowRate;
unsigned int flowMilliLitres;
unsigned long totalMilliLitres;

```

Figura 77 Variables y puertos GPIO para sensor de caudal y Relé

En la Figura 15 se muestra la función callback que se utilizó para recibir mensajes desde el broker.

```

/**
 * Función Callback para recibir un mensaje desde el broker
 */
void callback(char *topic, byte *payload, unsigned int length) {
    if (length <= 0)
        return;
    char primerCaracter = (char)payload[0];
    if (primerCaracter == '0')
        apagarLed();
    else if (primerCaracter == '1')
        encenderLed();

    delay(1000);
}

```

Figura 78 Función callback para recibir mensajes desde el broker

Seguidamente en la Figura 16 se muestra la programación del sensor de caudal mismo que nos medirá en mili litros la cantidad de agua utiliza por el sistema de irrigación.

```

void loop() {
  client.loop();
  if (millis() - env_msg_delay > TIME_DELAY) {
    currentMillis = millis();
  }
  if (currentMillis - previousMillis > interval) {
    pulseSec = pulseCount;
    pulseCount = 0;
    flowRate = ((1000.0 / (millis() - previousMillis)) * pulseSec) / calibrationFactor;
    previousMillis = millis();
    flowMilliLitres = (flowRate / 60) * 1000;
    totalMilliLitres += flowMilliLitres;
  }
  // En ocasiones puede devolver datos erróneos; por eso lo comprobamos
  if (isnan(flowRate) || isnan(totalMilliLitres))
  {
    env_msg_delay = millis();
    return;
  }
  // MQTT can only transmit strings
  String caudalAgua = String((int)totalMilliLitres);
  if (client.publish(caudalAgua_topic, String(caudalAgua).c_str())) {
    Serial.println("Humidity solid sent!");
  } else {
    client.connect(clientID, mqtt_username, mqtt_password);
    delay(10); // This delay ensures that client.publish doesn't clash with the client.connect call
    client.publish(caudalAgua_topic, String(caudalAgua).c_str());
  }
}

```

Figura 79 Programación del sensor de caudal

4.5. Conexión de broker con API-REST

Luego de haber configurado Mosquitto Broker en raspberrypi, y la configuración de los sensores y actuadores a los microcontroladores, se procede a la conexión del Api-REST con el servidor broker, esto se lo realizó utilizando la librería mqtt, instalada con el siguiente comando.

npm install mqtt

El mismo que es un cliente de mqtt que permite realizar la conexión de Nodejs con raspberrypi.

Para realizar la conexión entre el broker y el servicio web, se procedió a crear el controlador, ServerSuscriptorController, en donde se creó la clase llamada con el mismo nombre, aquí se realizó la escucha del suscriptor mqtt con el broker, para que cuando los sensores envíen sus datos este los almacene en la base de datos. En la Figura 17 se muestra un fragmento de código en donde se creó un constructor para definir las credenciales y direcciones para la conexión del broker.

```

constructor() {
  this.mqttClient = null;
  this.host = '192.168.0.104';
  this.port = '1883';
  this.username = 'riego';
  this.password = '3435';
}

```

Figura 80 Definir controlador con credenciales para la conexión

Una vez definido los parámetros de conexión se procede a crear la función connect, en donde se alojarán las configuraciones de suscriptor, publicador, error y desconec. En la Figura 18 se presenta el fragmento de código para la conexión y suscripción al broker o servidor antes configurado.

```

this.mqttClient.on('connect', async () => {
  console.log('Conectado a MQTT Broker');
  let topic = [];
  await Topic.find({}).then( onfulfilled: (result :... ) => {...}).catch((error) => {
    res.send({
      message: 'Error en la consulta Topic',
      error: error
    })
  });
  this.mqttClient.subscribe(topic, () => {
    console.log('Conectado al topic ' + topic);
  });
});
this.mqttClient.on('message', (topic, payload) => {
  console.log("El topic ", topic, " y el mensaje ", payload.toString());
  Topic.find({}).then( onfulfilled: (result :... )=>{...}).catch((error)=>{
    console.log(error);
  })
})
}

```

Figura 81 Función para conexión, suscripción y message del API-REST con Broker

Como se puede apreciar en la función anterior existen dos funciones una que se utiliza para conexión y suscripción al broker mediante mqtt, la función message es utilizada para mostrar los mensajes recibidos por el broker, el cual presenta el topic que es el nombre del sensor y el payload que es el mensaje enviado por los sensores, internamente aquí se realiza el guardado de dos latos de los sensores.

4.6. Conexión con aplicativo Front-end y aplicación Móvil

Luego de realizar la conexión del API-REST con el servidor Broker Mosquitto, se procede a consumir el api con aplicativo web y móvil los mismos que realizan las operaciones de control de riego. En la Figura 19 se presenta la página en donde se muestran los datos capturados por los sensores y el botón de riego. Y en la Figura 20 se ilustra la pantalla del aplicativo móvil para control de riego.

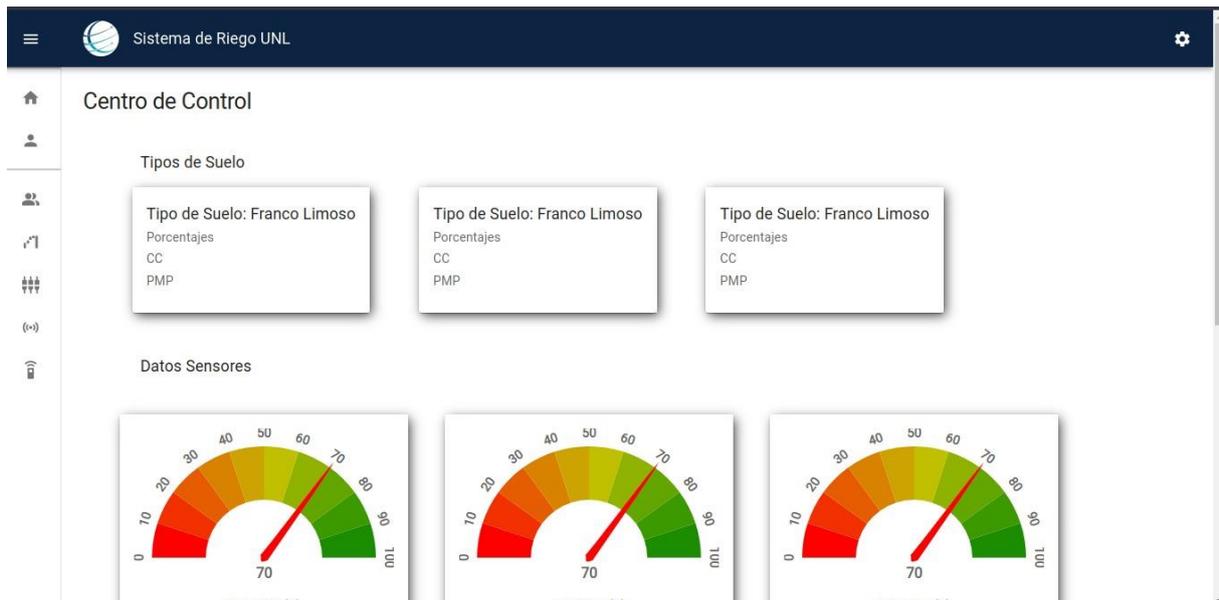


Figura 82 Aplicación web con los datos recogidos por sensores



Figura 83 Aplicación móvil para el control de Riego

Por último, en la Figura 21 se muestra una vista global del prototipo finalizado con la conexión de los sensores y actuadores, así como el aplicativo web y móvil del mismo

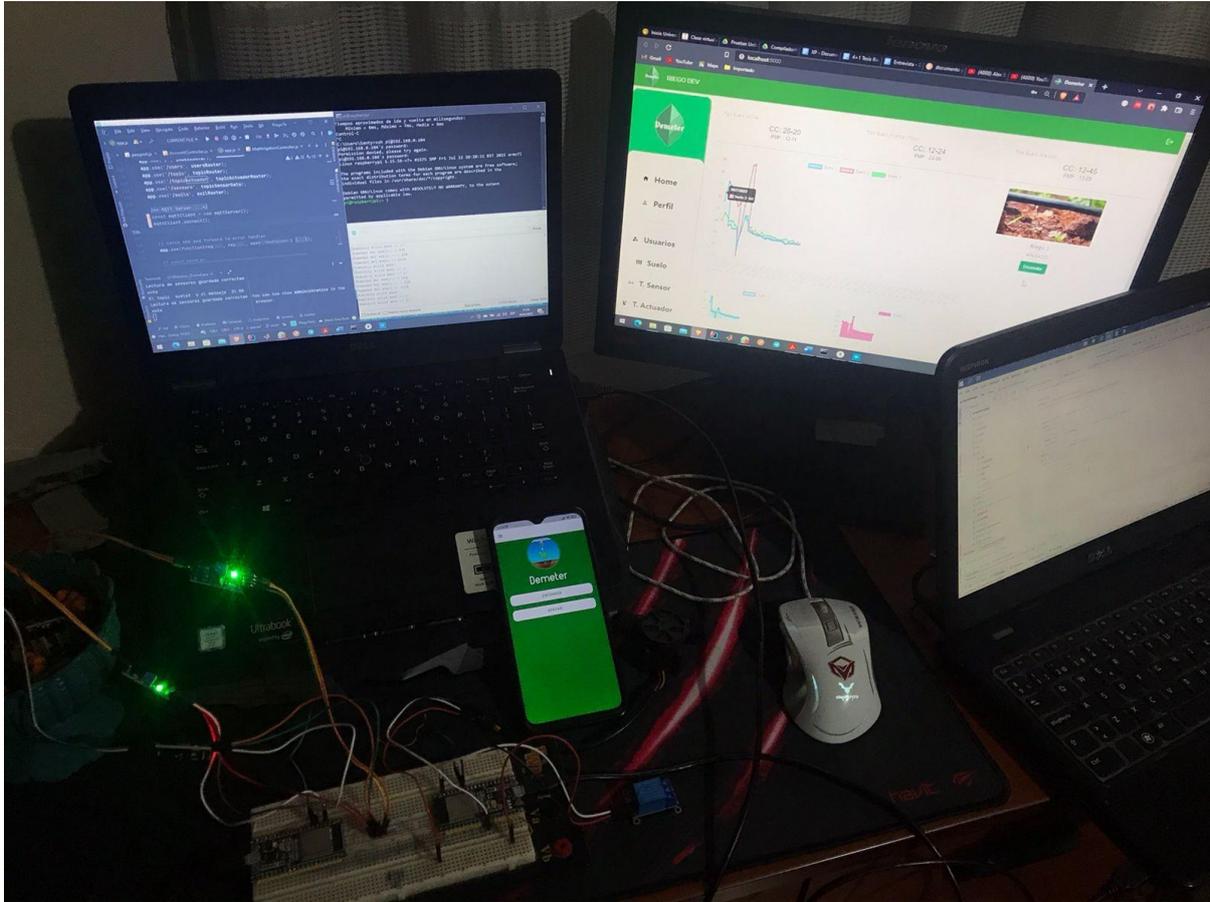


Figura 84 Vista global de prototipo de sistema de Irrigación con IoT

5. Anexos

Enlace de imágenes de construcción de sistema de irrigación con IoT.

https://drive.google.com/drive/folders/14av67bNrGU91t7j9b7iDw8_peqQxp1mw?usp=sharing

Anexo 6 Pruebas De software (Unitarias, Carga y estrés y De Usabilidad)

**FACULTAD DE ENERGÍA, LAS INDUSTRIAS Y
LOS RECURSOS NATURALES NO RENOVABLES**

**CARRERA DE INGENIERÍA EN SISTEMAS /
COMPUTACIÓN**

PERIODO OCTUBRE 2021- MARZO 2022

**SISTEMA DE IRRIGACIÓN UTILIZANDO
IOT PARA CULTIVOS DE LA UNIVERSIDAD
NACIONAL DE LOJA**

Versión [1.0]

Estudiantes:

- Alex Santiago Nole Reyes
- Angel Favian Minga Medina

Contenido

1. Casos y procedimientos de Pruebas Unitarias

En la presente sección se detallan los casos de pruebas y procedimientos que se realizan en cada iteración del sistema de irrigación con IoT para los cultivos de la Universidad de Loja.

1.1. “AccountController” Versión 1

Implementador: Alex Nole, Angel Minga

1.1.1. Requerimientos funcionales

1.1.1.1.El Administrador podrá registrar los datos (Nombres, Apellidos, Dirección, Teléfono, correo, contraseña).

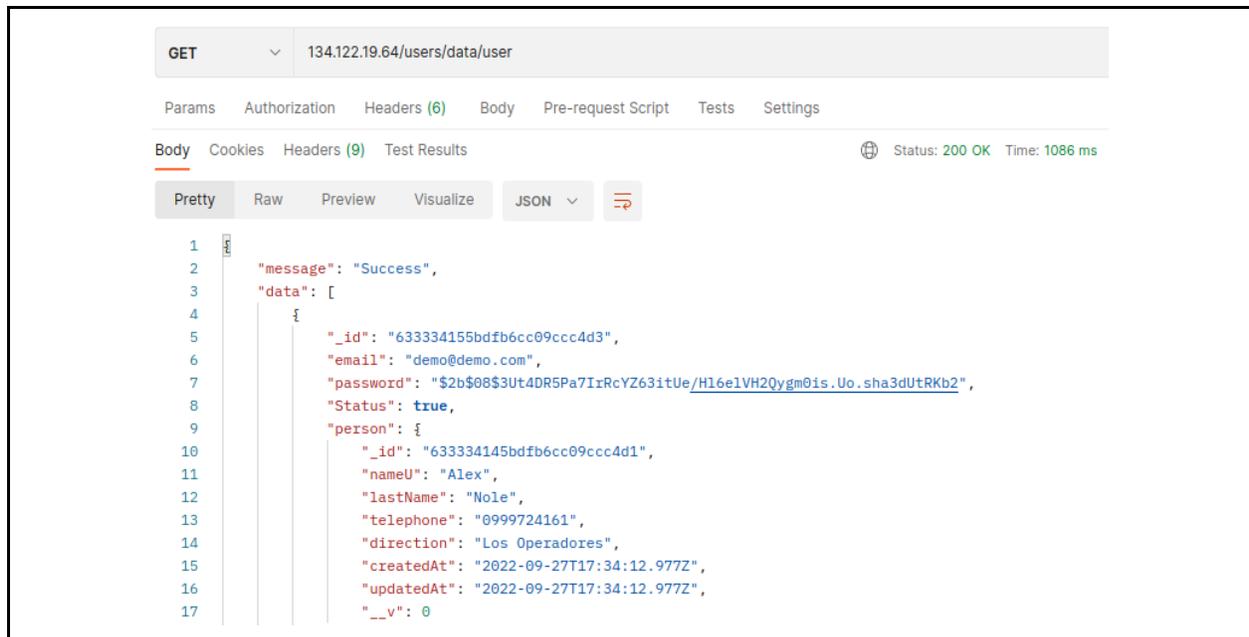
Método	URL	Resultado	Tiempo ejecución
---------------	------------	------------------	-------------------------

POST	134.122.19.64/users/signup	Registra los datos de los usuarios: Nombre, Apellido, Dirección, Teléfono, Correo y contraseña	109 ms
------	----------------------------	--	--------

The screenshot shows a REST client interface for a POST request to 134.122.19.64/users/signup. The request is form-urlencoded with a 'telephone' field set to '0999999999'. The response body is JSON: {"message": "El usuario se registro correctamente"}.

1.1.1.2.El administrador podrá visualizar los usuarios registrados

Método	URL	Resultado	Tiempo ejecución
GET	134.122.19.64/users/data/user	Lista Todos los usuarios registrados al sistema	1086 ms



1.1.1.3.El usuario o administrador podrá actualizar sus datos (Nombres, Apellidos, Dirección, Teléfono, correo, contraseña).

Método	URL	Resultado	Tiempo ejecución
GET	134.122.19.64/users/updateList/633b0ac766ee86df8d862f27	Lista Al usuario seleccionado por su ID	467 ms

GET 134.122.19.64/users/updateList/633b0ac766ee86df8d862f27

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (9) Test Results Status: 200 OK Time: 467 ms

Pretty Raw Preview Visualize JSON ≡

```

1  {
2    "message": "Success",
3    "data": {
4      "_id": "633b0ac766ee86df8d862f27",
5      "email": "correo@correo.com",
6      "password": "$2b$08$vPk9K0J5L4N.GLNvQfu5buUmbpH3y4auRomhsP5IXE0CBzKL3pesy",
7      "Status": true,
8      "person": {
9        "_id": "633b0ac766ee86df8d862f25",
10       "nameU": "Juan",
11       "lastName": "Perez",
12       "telephone": "0999999999",
13       "direction": "Loja-Ecuador",
14       "createdAt": "2022-10-03T16:16:07.327Z",
15       "updatedAt": "2022-10-03T16:16:07.327Z",
16       "__v": 0
17     }
18   }
19 }

```

POST	134.122.19.64/users/updateAccount	Actualiza los datos del usuario o perfil	694 ms
------	-----------------------------------	--	--------

POST 134.122.19.64/users/updateAccount

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY VALUE DESCRIPTION

Body Cookies Headers (9) Test Results Status: 200 OK Time: 694 ms

Pretty Raw Preview Visualize JSON ≡

```

1  {
2    "message": "Usuario actualizado con éxito",
3    "data": {
4      "_id": "633b0ac766ee86df8d862f27",
5      "email": "correo@correo.com",
6      "password": "$2b$08$vPk9K0J5L4N.GLNvQfu5buUmbpH3y4auRomhsP5IXE0CBzKL3pesy",
7      "Status": true,
8      "person": {
9        "_id": "633b0ac766ee86df8d862f25",
10       "nameU": "Juan",
11       "lastName": "Perez",
12       "telephone": "0999999999",
13       "direction": "Loja-Ecuador",
14       "createdAt": "2022-10-03T16:16:07.490Z",
15       "updatedAt": "2022-10-03T16:16:07.490Z",
16       "__v": 0
17     }
18   }
19 }

```

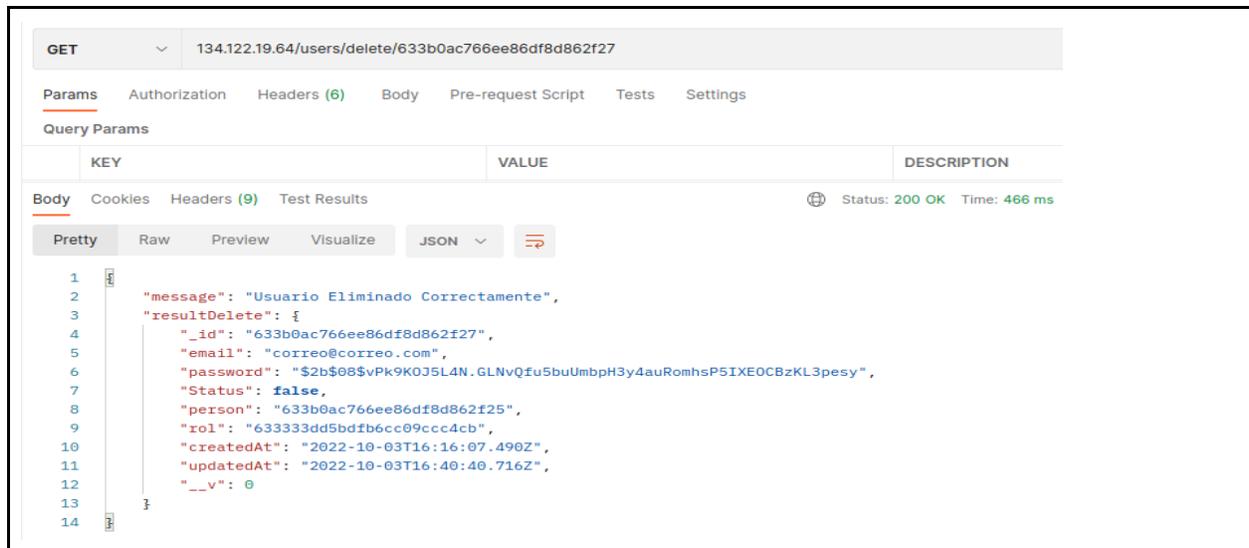
1.1.1.4.El usuario y administrador podrá iniciar sesión (correo, contraseña).

Método	URL	Resultado	Tiempo ejecución
--------	-----	-----------	------------------

POST	134.122.19.64/users/login	Los Usuarios y Administrador pueden iniciar sesión para ingresar al sistema	399 ms									
<p>The screenshot shows a REST client interface for a POST request to 134.122.19.64/users/login. The request body is form-urlencoded with the following data:</p> <table border="1"> <thead> <tr> <th>KEY</th> <th>VALUE</th> <th>DESCRIPTION</th> </tr> </thead> <tbody> <tr> <td>email</td> <td>correo@correo.com</td> <td></td> </tr> <tr> <td>password</td> <td>12345</td> <td></td> </tr> </tbody> </table> <p>The response body is a JSON object:</p> <pre> 1 { 2 "message": "Bienvenido al Sistema Riego", 3 "data": { 4 "_id": "633b0ac766ee86df8d862f27", 5 "email": "correo@correo.com", 6 "password": "\$2b\$08\$vPk9K0J5L4N.GLNvQfu5buUmbpH3y4auRomhsP5IXE0CbzKL3pesy", 7 "Status": true, 8 "person": { 9 "_id": "633b0ac766ee86df8d862f25", 10 "nameU": "Juan Ar", 11 "lastName": "Perez", </pre>				KEY	VALUE	DESCRIPTION	email	correo@correo.com		password	12345	
KEY	VALUE	DESCRIPTION										
email	correo@correo.com											
password	12345											

1.1.1.5.El administrador podrá dar de baja a los usuarios registrados

Método	URL	Resultado	Tiempo ejecución
GET	134.122.19.64/users/delete/633b0ac766ee86df8d862f27	El administrador da de baja a los usuarios registrados cambiando su estado a false	466 ms

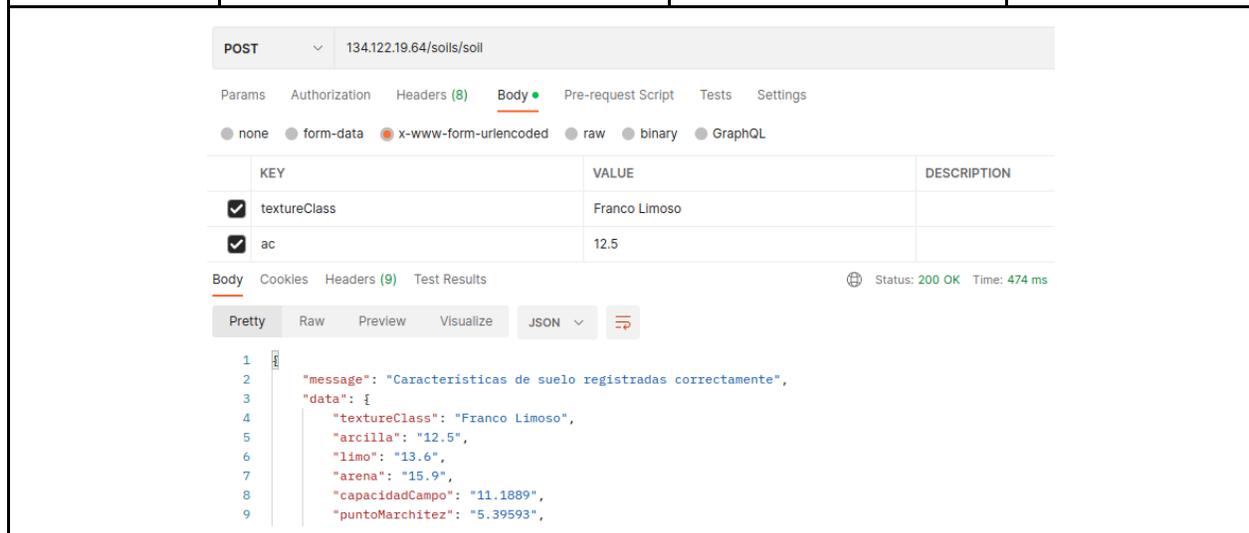


1.2. “SoilController” Versión 1

1.2.1. Requerimientos Funcionales:

1.2.1.1. El administrador podrá registrar el tipo de suelo (Tipo suelo, capacidad de campo, punto de marchitez)

Método	URL	Resultado	Tiempo ejecución
POST	134.122.19.64/soils/soil	Registra las características de suelo	474 ms



1.2.1.2. El administrador podrá actualizar el tipo de suelo (Tipo suelo, capacidad de campo, punto de marchitez)

Método	URL	Resultado	Tiempo ejecución
GET	134.122.19.64/soils/soilData/633c5d7366ee86df8d862f38 134.122.19.64/soils/soilData/{id}	Busca el tipo de suelo según el id	518 ms

GET 134.122.19.64/soils/soilData/633c5d7366ee86df8d862f38

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
-----	-------	-------------

Body Cookies Headers (9) Test Results Status: 200 OK Time: 518 ms

Pretty Raw Preview Visualize JSON

```

1  {
2    "message": "Consulta Correcta",
3    "data": {
4      "_id": "633c5d7366ee86df8d862f38",
5      "textureClass": "Frnaco Arcilloso",
6      "arcilla": "19.8",
7      "limo": "13.6",
8      "arena": "25.8",
9      "capacidadCampo": "14.9206",
10     "puntoMarchitez": "7.746060000000001",
11     "createdAt": "2022-10-04T16:21:07.181Z",
12     "updatedAt": "2022-10-04T16:24:24.366Z",
13     "__v": 0
14   }
15 }

```

POST	134.122.19.64/soils/updateSoil/633c5d7366ee86df8d862f38 134.122.19.64/soils/updateSoil/{id}	Actualiza las características de suelo	409 ms
------	--	--	--------

POST 134.122.19.64/soils/updateSoil/633c5d7366ee86df8d862f38

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> textureClass	Frnaco Arcilloso	
<input checked="" type="checkbox"/> ac	19.8	

Body Cookies Headers (9) Test Results Status: 200 OK Time: 409 ms

Pretty Raw Preview Visualize JSON

```

1  {
2    "message": "Data Actualizada correctamente",
3    "data": {
4      "_id": "633c5d7366ee86df8d862f38",
5      "textureClass": "Frnaco Arcilloso",
6      "arcilla": "19.8",
7      "limo": "13.6",
8      "arena": "25.8",
9      "capacidadCampo": "14.9206",
10     "puntoMarchitez": "7.746060000000001",
11     "createdAt": "2022-10-04T16:21:07.181Z",
12     "updatedAt": "2022-10-04T16:24:24.366Z",
13     "__v": 0
14   }
15 }

```

1.2.1.3.El administrador podrá visualizar los tipos de suelo registrados

Método	URL	Resultado	Tiempo ejecución
GET	134.122.19.64/soils/soil	Lista los tipos de suelo registrados por el administrador	226 ms

GET 134.122.19.64/soils/soil

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
-----	-------	-------------

Body Cookies Headers (9) Test Results Status: 200 OK Time: 226 ms

Pretty Raw Preview Visualize JSON

```

1  {
2    "message": "Success",
3    "data": [
4      {
5        "_id": "633467b83e32243b419d7de5",
6        "textureClass": "Franco Limoso",
7        "arcilla": "24.56",
8        "limo": "54",
9        "arena": "12.44",
10       "capacidadCampo": "23.44292",
11       "puntoMarchitez": "13.187987999999999",
12       "createdAt": "2022-09-28T15:26:48.450Z",
13       "updatedAt": "2022-09-28T15:26:48.450Z",

```

1.2.1.4.El administrador podrá eliminar los tipos de suelo registrados

Método	URL	Resultado	Tiempo ejecución
DELETE	134.122.19.64/soils/deleteSoil/633c5d7366ee86df8d862f38 134.122.19.64/soils/deleteSoil/{id}	Elimina el tipo de suelo según su ID	406 ms

```

1  "message": "Tipo suelo eliminado correctamente",
2  "data": {
3    "_id": "633c5d7366ee86df8d862f38",
4    "textureClass": "Frnaco Arcilloso",
5    "arcilla": "19.8",
6    "limo": "13.6",
7    "arena": "25.8",
8    "capacidadCampo": "14.9286",
9    "puntoMarchitez": "7.746868000000001",
10   "createdAt": "2022-10-04T16:21:07.181Z",
11   "updatedAt": "2022-10-04T16:24:24.366Z",
12   "__v": 0
13 }

```

1.3. “TopicActuadorController y TopicSensorController”

1.3.1. Requerimientos Funcionales

1.3.1.1. El sistema permitirá registrar topics actuador y sensor para la conexión con el hardware

Método	URL	Resultado	Tiempo ejecución
POST	134.122.19.64/topic/registroTopic	Registra Topic para los sensores que se comunicaran con el hardware	420 ms

POST 134.122.19.64/topic/registroTopic

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data **x-www-form-urlencoded** raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nameTopic	Suelo1	
<input checked="" type="checkbox"/> nameSensor	sensor para sensor superior	

Body Cookies Headers (9) Test Results Status: 200 OK Time: 420 ms

Pretty Raw Preview Visualize JSON

```

1  {
2    "message": "TopicSensor registrado correctamente",
3    "data": {
4      "nameTopic": "Suelo1",
5      "nameSensor": "sensor para sensor superior",
6      "status": true,
7      "_id": "633c602266ee86df8d862f40",
8      "createdAt": "2022-10-04T16:32:34.108Z",
9      "updatedAt": "2022-10-04T16:32:34.108Z",
10     "updatedAt": "2022-10-04T16:32:34.108Z",

```

POST	134.122.19.64/topicActuador/registroTopicActuador	Registra Topic de actuadores para encender o apagar el riego	387 ms
------	---	--	--------

POST 134.122.19.64/topicActuador/registroTopicActuador

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data **x-www-form-urlencoded** raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nameTopicActuador	ControlRiego	
<input checked="" type="checkbox"/> nameActuador	Control de riego	

Body Cookies Headers (9) Test Results Status: 200 OK Time: 387 ms

Pretty Raw Preview Visualize JSON

```

1  {
2    "message": "TopicActuador de Actuador registrado correctamente",
3    "data": {
4      "status": true,
5      "_id": "633c611366ee86df8d862f42",
6      "createdAt": "2022-10-04T16:36:35.661Z",
7      "updatedAt": "2022-10-04T16:36:35.661Z",
8      "updatedAt": "2022-10-04T16:36:35.661Z",
9      "updatedAt": "2022-10-04T16:36:35.661Z",
10     "updatedAt": "2022-10-04T16:36:35.661Z",

```

1.3.1.2.El sistema permitirá visualizar los topics registrados

Método	URL	Resultado	Tiempo ejecución
GET	134.122.19.64/topic/topics	Lista los Topics	655 ms

Registrados en el sistema

GET 134.122.19.64/topic/topics

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
-----	-------	-------------

Body Cookies Headers (9) Test Results Status: 200 OK Time: 655 ms

Pretty Raw Preview Visualize JSON

```
1 1
2 2 "message": "Consulta Correcta",
3 3 "data": [
4 4   {
5 5     "_id": "6338893a6ae3b4ef94c6baec",
6 6     "nameTopic": "sensor1",
7 7     "nameSensor": "suelo",
8 8     "status": true,
9 9     "createdAt": "2022-10-01T18:38:50.166Z",
10 10    "updatedAt": "2022-10-01T18:38:50.166Z",
11 11    "__v": 0
12 12   },
13 13   {
```

GET

134.122.19.64/topicActuador/topicsActuador

Lista los Topics de actuadores registrados en el sistema

444 ms

GET 134.122.19.64/topicActuador/topicsActuador

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

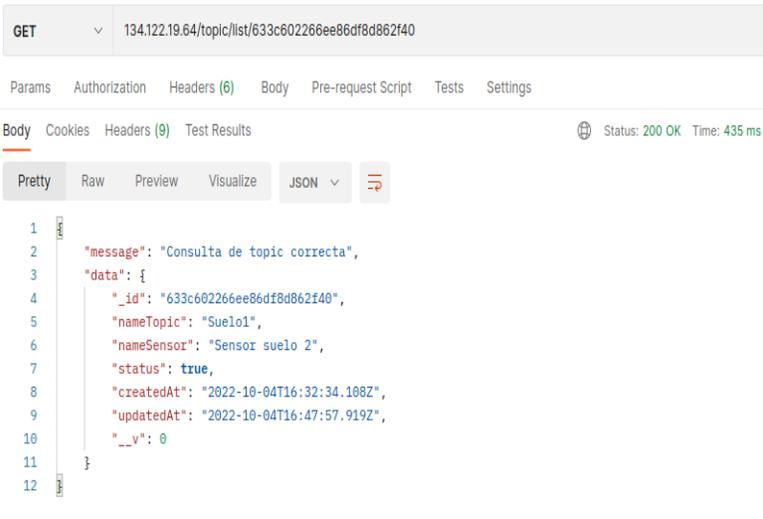
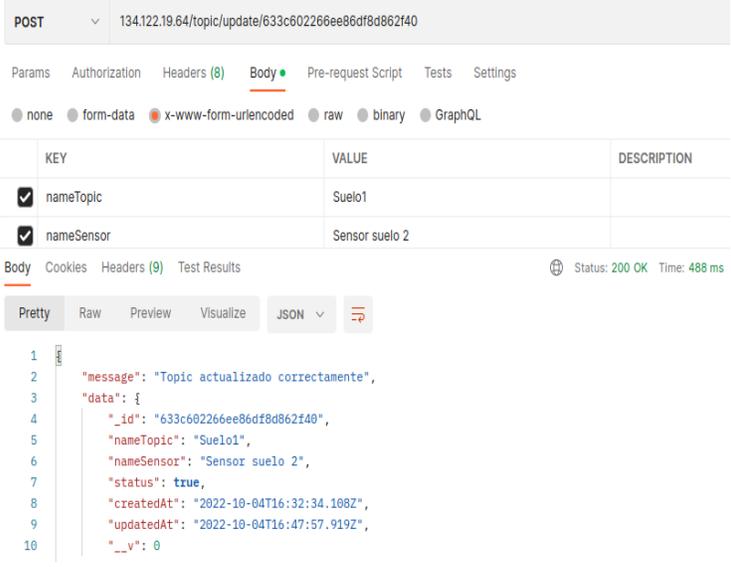
KEY	VALUE	DESCRIPTION
-----	-------	-------------

Body Cookies Headers (9) Test Results Status: 200 OK Time: 444 ms

Pretty Raw Preview Visualize JSON

```
1 1
2 2 "message": "Consulta Correcta",
3 3 "data": [
4 4   {
5 5     "_id": "633c611366ee86df8d862f42",
6 6     "status": true,
7 7     "createdAt": "2022-10-04T16:36:35.661Z",
8 8     "updatedAt": "2022-10-04T16:36:35.661Z",
9 9     "__v": 0
10 10   }
11 11 ]
12 12
```

1.3.1.3.El sistema permitirá actualizar los topics registrados

Método	URL	Resultado	Tiempo ejecución									
GET	134.122.19.64/topic/list/633c602266ee86df8d862f40 134.122.19.64/topic/list/{id}	Busca el topic mediante el id	435 ms									
 <pre> 1 "message": "Consulta de topic correcta", 2 3 "data": { 4 "_id": "633c602266ee86df8d862f40", 5 "nameTopic": "Suelo1", 6 "nameSensor": "Sensor suelo 2", 7 "status": true, 8 "createdAt": "2022-10-04T16:32:34.108Z", 9 "updatedAt": "2022-10-04T16:47:57.919Z", 10 "_v": 0 11 12 </pre>												
POST	134.122.19.64/topic/update/633c602266ee86df8d862f40 134.122.19.64/topic/update/{id}	Actualiza el topic mediante id	488 ms									
 <table border="1" data-bbox="446 1365 1177 1480"> <thead> <tr> <th>KEY</th> <th>VALUE</th> <th>DESCRIPTION</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/> nameTopic</td> <td>Suelo1</td> <td></td> </tr> <tr> <td><input checked="" type="checkbox"/> nameSensor</td> <td>Sensor suelo 2</td> <td></td> </tr> </tbody> </table> <pre> 1 "message": "Topic actualizado correctamente", 2 3 "data": { 4 "_id": "633c602266ee86df8d862f40", 5 "nameTopic": "Suelo1", 6 "nameSensor": "Sensor suelo 2", 7 "status": true, 8 "createdAt": "2022-10-04T16:32:34.108Z", 9 "updatedAt": "2022-10-04T16:47:57.919Z", 10 "_v": 0 </pre>				KEY	VALUE	DESCRIPTION	<input checked="" type="checkbox"/> nameTopic	Suelo1		<input checked="" type="checkbox"/> nameSensor	Sensor suelo 2	
KEY	VALUE	DESCRIPTION										
<input checked="" type="checkbox"/> nameTopic	Suelo1											
<input checked="" type="checkbox"/> nameSensor	Sensor suelo 2											
GET	134.122.19.64/topicActuador/list/633c611366ee86df8d862f42	Busca el Topic de	509 ms									

	134.122.19.64/topicActuador/list/{id}	actuador mediante id	
--	---------------------------------------	----------------------	--

GET 134.122.19.64/topicActuador/list/633c611366ee86df8d862f42

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION

Body Cookies Headers (9) Test Results Status: 200 OK Time: 509 ms

Pretty Raw Preview Visualize JSON

```

1  {
2    "message": "Busqueda de topics correcta",
3    "data": {
4      "_id": "633c611366ee86df8d862f42",
5      "status": true,
6      "createdAt": "2022-10-04T16:36:35.661Z",
7      "updatedAt": "2022-10-04T16:36:35.661Z",
8      "__v": 0
9    }
  }

```

POST	134.122.19.64/topicActuador/updateActuador/633c611366ee86df8d862f42 134.122.19.64/topicActuador/updateActuador/{id}	Actualiza el Topic de actuadores mediante el id	432 ms
------	--	---	--------

POST 134.122.19.64/topicActuador/updateActuador/633c611366ee86df8d862f42

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none
 form-data
 x-www-form-urlencoded
 raw
 binary
 GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nameTopicActuator	ControlRiego1	
<input checked="" type="checkbox"/> nameActuator	Manejo de riego	

Body Cookies Headers (9) Test Results Status: 200 OK Time: 432 ms

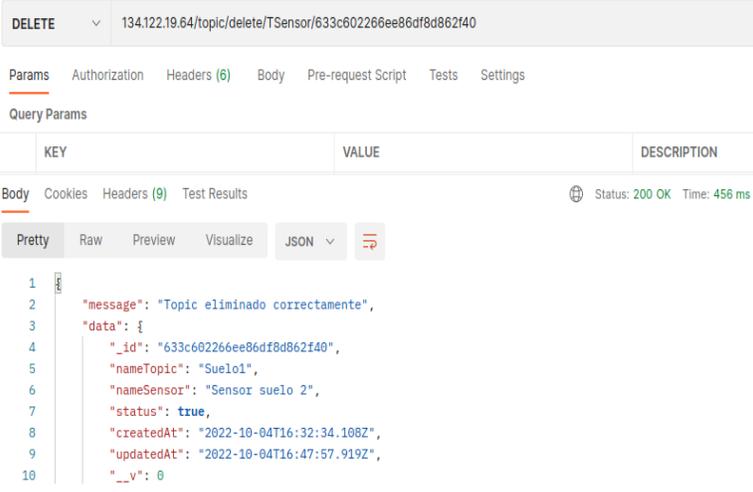
Pretty Raw Preview Visualize JSON

```

1  {
2    "message": "Topic actualizado correctamente",
3    "data": {
4      "_id": "633c611366ee86df8d862f42",
5      "status": true,
6      "createdAt": "2022-10-04T16:36:35.661Z",
7      "updatedAt": "2022-10-04T16:36:35.661Z",
8      "__v": 0
9    }
  }

```

1.3.1.4.El sistema permitirá eliminar los topics registrados

Método	URL	Resultado	Tiempo ejecución
DELETE	134.122.19.64/topic/delete/TSensor/633c602266ee86df8d862f40 134.122.19.64/topic/delete/TSensor/{id}	Elimina el Topic de sensores mediante su id	456 ms
 <p>The screenshot shows a REST client interface for a DELETE request. The URL is 134.122.19.64/topic/delete/TSensor/633c602266ee86df8d862f40. The response status is 200 OK with a time of 456 ms. The response body is a JSON object:</p> <pre> 1 { 2 "message": "Topic eliminado correctamente", 3 "data": { 4 "_id": "633c602266ee86df8d862f40", 5 "nameTopic": "Suelo1", 6 "nameSensor": "Sensor suelo 2", 7 "status": true, 8 "createdAt": "2022-10-04T16:32:34.108Z", 9 "updatedAt": "2022-10-04T16:47:57.919Z", 10 "_v": 0 </pre>			
DELETE	134.122.19.64/topicActuador/deletetopic/633c611366ee86df8d862f42 134.122.19.64/topicActuador/deletetopic/{id}	Elimina el Topic de actuador mediante el id	478 ms

DELETE 134.122.19.64/topicActuador/deletetopic/633c611366ee86df8d862f42

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
-----	-------	-------------

Body Cookies Headers (9) Test Results Status: 200 OK Time: 478 ms

Pretty Raw Preview Visualize JSON

```

1  {
2    "message": "Topic eliminado correctamente",
3    "data": {
4      "_id": "633c611366ee86df8d862f42",
5      "status": true,
6      "createdAt": "2022-10-04T16:36:35.661Z",
7      "updatedAt": "2022-10-04T16:36:35.661Z",
8      "__v": 0
9    }
10 }

```

1.4. “RolController”

1.4.1. Requerimientos Funcionales

1.4.1.1. El sistema permitirá registrar los roles para los usuarios

Método	URL	Resultado	Tiempo ejecución
POST	134.122.19.64/users/rol	Registra roles para las funciones de los usuarios	965 ms

POST 134.122.19.64/users/rol

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> nombreRol	Usuario	

Body Cookies Headers (9) Test Results Status: 200 OK Time: 965 ms

Pretty Raw Preview Visualize JSON

```

1  {
2    "message": "Rol registrado correctamente",
3    "data": {
4      "nameRol": "Usuario",
5      "status": true,
6      "_id": "633db33b66ee86df8d862f51",
7      "createdAt": "2022-10-05T16:39:23.705Z",
8      "updatedAt": "2022-10-05T16:39:23.705Z",
9      "__v": 0
10   }
11 }

```

1.4.1.2.EL sistema permitirá visualizar los roles registrados

Método	URL	Resultado	Tiempo ejecución
GET	134.122.19.64/users/rols	Lista los roles registrados en el sistema	234


```

1  {
2    "message": "Consulta de roles correcta",
3    "data": [
4      {
5        "_id": "633333e85bdf6cc09ccc4cd",
6        "nameRol": "Administrador",
7        "status": "tzue",
8        "createdAt": "2022-09-27T17:33:28.844Z",
9        "updatedAt": "2022-09-27T17:33:28.844Z",
10       "__v": 0
11      },
12      {
13        "_id": "633db33b66ee86df8d862f51",

```

1. Prueba carga y estrés

Listar Usuario

Muestra #	Tiempo de ...	Nombre del...	Etiqueta	Tiempo de ...	Estado	Bytes	Sent Bytes	Latency	Connect Tl...
1	00:44:21.315	Demeter 1-1	Listar Usua...	432	🟢	3455	131	432	98
2	00:44:21.419	Demeter 1-2	Listar Usua...	439	🟢	3455	131	439	97
3	00:44:21.519	Demeter 1-3	Listar Usua...	430	🟢	3455	131	430	95
4	00:44:21.620	Demeter 1-4	Listar Usua...	436	🟢	3455	131	436	95
5	00:44:21.721	Demeter 1-5	Listar Usua...	437	🟢	3455	131	437	96
6	00:44:21.821	Demeter 1-6	Listar Usua...	442	🟢	3455	131	442	102
7	00:44:21.919	Demeter 1-7	Listar Usua...	440	🟢	3455	131	440	102
8	00:44:22.019	Demeter 1-8	Listar Usua...	434	🟢	3455	131	434	96
9	00:44:22.119	Demeter 1-9	Listar Usua...	443	🟢	3455	131	443	100
10	00:44:22.219	Demeter 1-10	Listar Usua...	441	🟢	3455	131	441	100
11	00:44:22.314	Demeter 1-11	Listar Usua...	431	🟢	3455	131	431	96
12	00:44:22.419	Demeter 1-12	Listar Usua...	431	🟢	3455	131	431	95
13	00:44:22.519	Demeter 1-13	Listar Usua...	430	🟢	3455	131	430	94
14	00:44:22.620	Demeter 1-14	Listar Usua...	435	🟢	3455	131	435	95
15	00:44:22.715	Demeter 1-15	Listar Usua...	424	🟢	3455	131	424	93
16	00:44:22.819	Demeter 1-16	Listar Usua...	461	🟢	3455	131	461	97
17	00:44:22.920	Demeter 1-17	Listar Usua...	432	🟢	3455	131	432	95
18	00:44:23.020	Demeter 1-18	Listar Usua...	439	🟢	3455	131	439	96
19	00:44:23.119	Demeter 1-19	Listar Usua...	436	🟢	3455	131	436	96
20	00:44:23.218	Demeter 1-20	Listar Usua...	441	🟢	3455	131	441	106
21	00:44:23.319	Demeter 1-21	Listar Usua...	445	🟢	3455	131	445	96
22	00:44:23.421	Demeter 1-22	Listar Usua...	446	🟢	3455	131	446	98
23	00:44:23.517	Demeter 1-23	Listar Usua...	431	🟢	3455	131	431	95
24	00:44:23.619	Demeter 1-24	Listar Usua...	434	🟢	3455	131	434	96

Scroll automatically?
 Child samples?
 No. de Muestras 100
 Última Muestra 425
 Media 434
 Desviación 6

Figura 1 Resultados carga y estrés listar usuario

El la figura 1 se visualiza la prueba realizada de peticiones al recurso lista de usuarios en el cual se visualiza el tiempo de espera, latencia, entre otros datos entre cada petición.

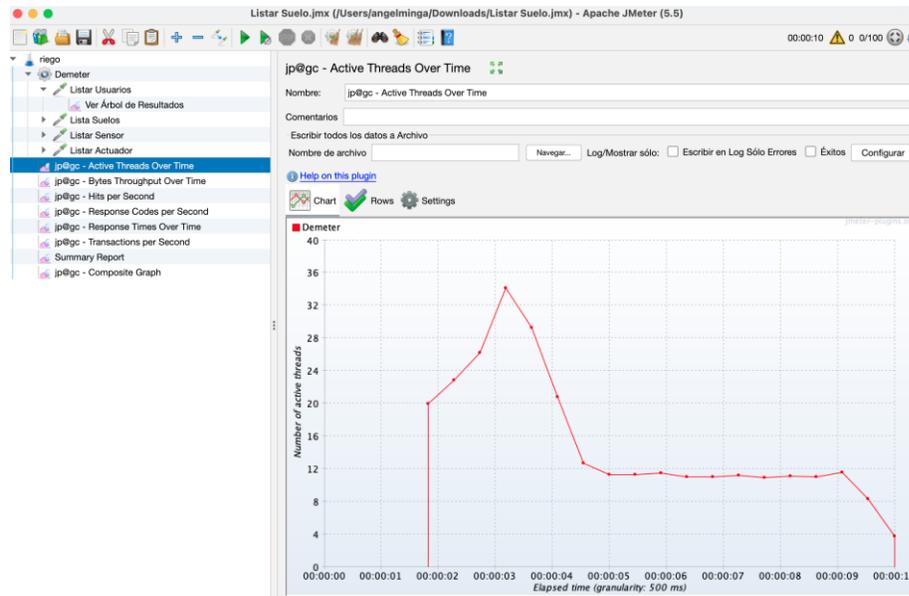


Figura 2 Resultados Active Threads Over Time (Subprocesos activos a lo largo del tiempo)

En la figura 2 se visualiza cuántos subprocesos activos hay en cada grupo de subprocesos durante la ejecución de la peticiones, se observa que al iniciar inicia con 20 subprocesos en un tiempo de 2 segundos, y así hasta llegar a un 35 subprocesos, y va disminuyendo de acuerdo al tiempo, hasta culminar con las 100 peticiones.

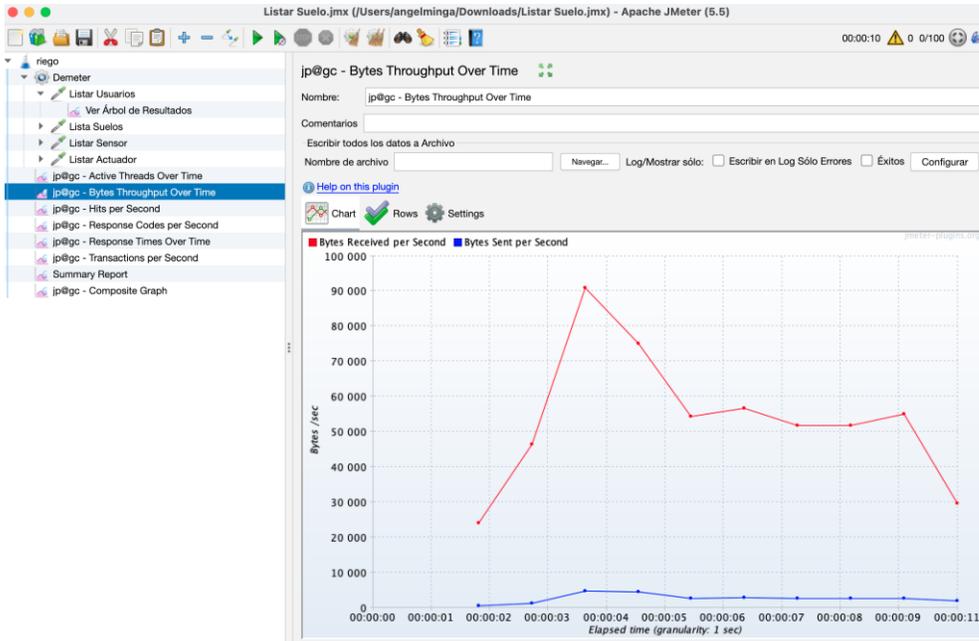


Figura 3 Resultados Bytes Throughput Over Time (Rendimiento de bytes a lo largo del tiempo)

En la figura 3 se observa que cuando aumenta la cantidad de subprocesos, JMeter generará más subprocesos nuevos y todos ejecutarán la solicitud que tiene en su prueba en paralelo, por lo tanto, más subprocesos, más solicitudes enviadas simultáneamente que es un aumento total de bytes recibidos por segundo.

Lista Suelo

Muestra #	Tiempo de ...	Nombre def...	Etiqueta	Tiempo de ...	Estado	Bytes	Sent Bytes	Latency	Connect T...
1	00:46:20.790	Demeter 1-1	Lista Suelos	216	✓	851	126	216	0
2	00:46:20.852	Demeter 1-2	Lista Suelos	217	✓	851	126	217	0
3	00:46:20.882	Demeter 1-3	Lista Suelos	218	✓	851	126	218	0
4	00:46:21.051	Demeter 1-4	Lista Suelos	216	✓	851	126	216	0
5	00:46:21.144	Demeter 1-5	Lista Suelos	213	✓	851	126	213	0
6	00:46:21.292	Demeter 1-6	Lista Suelos	220	✓	851	126	220	0
7	00:46:21.351	Demeter 1-7	Lista Suelos	214	✓	851	126	214	0
8	00:46:21.442	Demeter 1-8	Lista Suelos	215	✓	851	126	215	0
9	00:46:21.549	Demeter 1-9	Lista Suelos	220	✓	851	126	220	0
10	00:46:21.647	Demeter 1...	Lista Suelos	217	✓	851	126	217	0
11	00:46:21.745	Demeter 1...	Lista Suelos	215	✓	851	126	215	0
12	00:46:21.853	Demeter 1...	Lista Suelos	214	✓	851	126	214	0
13	00:46:21.947	Demeter 1...	Lista Suelos	216	✓	851	126	216	0
14	00:46:22.053	Demeter 1...	Lista Suelos	219	✓	851	126	219	0
15	00:46:22.154	Demeter 1...	Lista Suelos	218	✓	851	126	218	0
16	00:46:22.266	Demeter 1...	Lista Suelos	217	✓	851	126	217	0
17	00:46:22.348	Demeter 1...	Lista Suelos	214	✓	851	126	214	0
18	00:46:22.479	Demeter 1...	Lista Suelos	215	✓	851	126	215	0
19	00:46:22.554	Demeter 1...	Lista Suelos	217	✓	851	126	217	0
20	00:46:22.645	Demeter 1...	Lista Suelos	214	✓	851	126	214	0
21	00:46:22.763	Demeter 1...	Lista Suelos	219	✓	851	126	219	0
22	00:46:22.849	Demeter 1...	Lista Suelos	213	✓	851	126	213	0
23	00:46:22.962	Demeter 1...	Lista Suelos	217	✓	851	126	217	0
24	00:46:23.041	Demeter 1...	Lista Suelos	220	✓	851	126	220	0

Figura 4 Resultados carga y estrés listar suelo

El la figura 4 se visualiza la prueba realizada de peticiones al recurso lista de suelo en el cual se visualiza el tiempo de espera, latencia, entre otros datos entre cada petición.

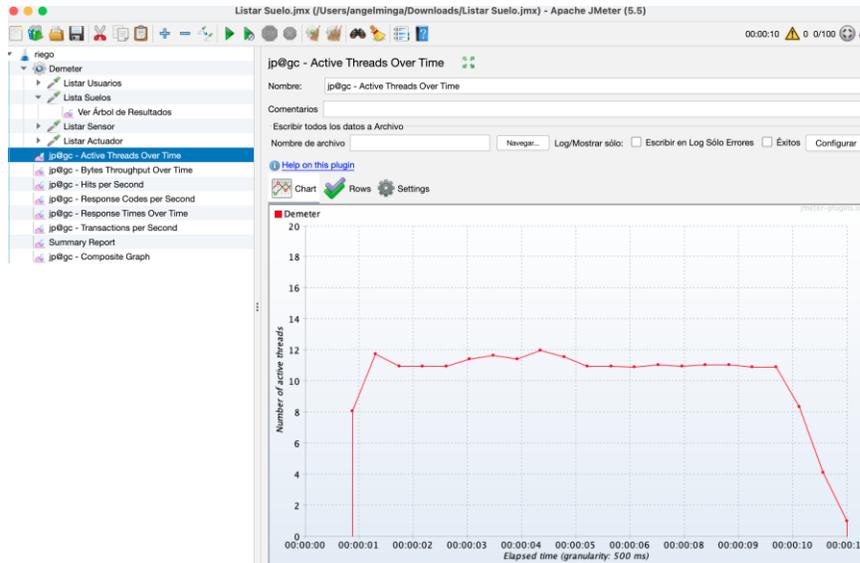


Figura 5 Resultados Active Threads Over Time (Subprocesos activos a lo largo del tiempo)

En la figura 5 se visualiza Active Threads Over Time los subprocesos que inicien con 8 subprocesos en 1 segundo, y existe un incremento de 4 subprocesos asta llegar a 12 subprocesos, y se mantiene hasta finalizar en 11 segundos todo los subprocesos.

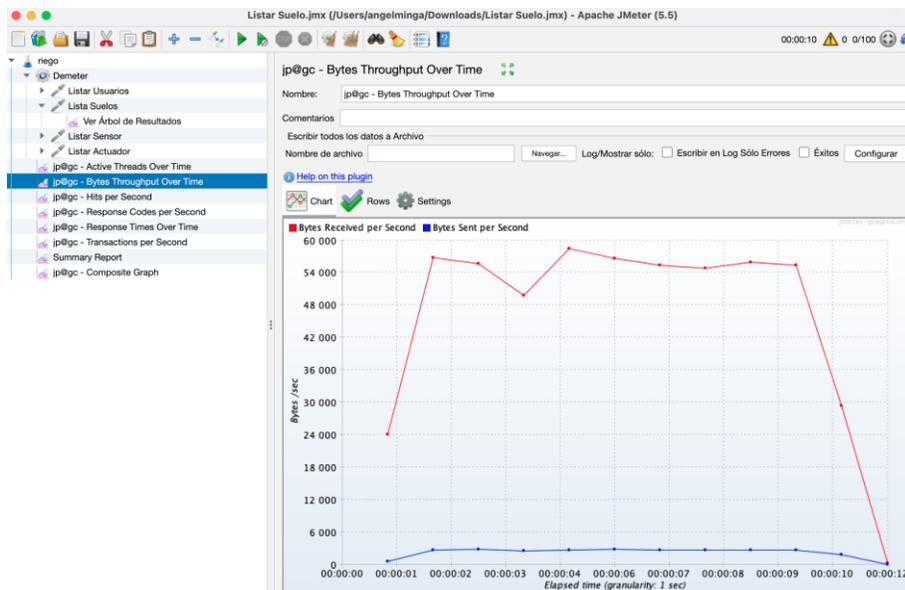


Figura 6 Resultados Bytes Throughput Over Time

En la figura 6 se visualiza cuando aumenta la cantidad de subprocesos, JMeter generará más subprocesos nuevos y todos ejecutarán la solicitud que tiene en su prueba en

paralelo, entre más solicitudes enviadas simultáneamente que es un aumento total de bytes recibidos por segundo.

Lista sensor

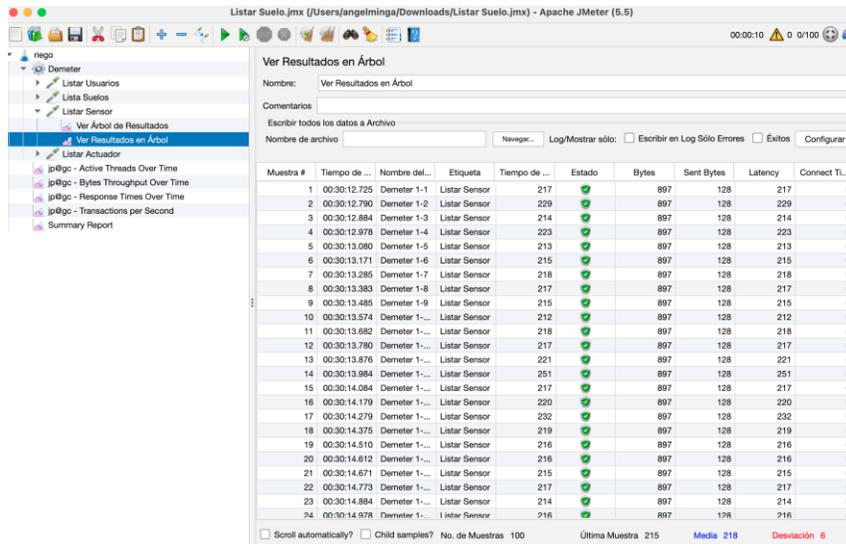


Figura 7 Resultados carga y estrés listar sensor

En la figura 7 se visualiza la prueba realizada de peticiones al recurso lista de sensor en el cual se visualiza el tiempo de espera, latencia, entre otros datos entre cada petición.

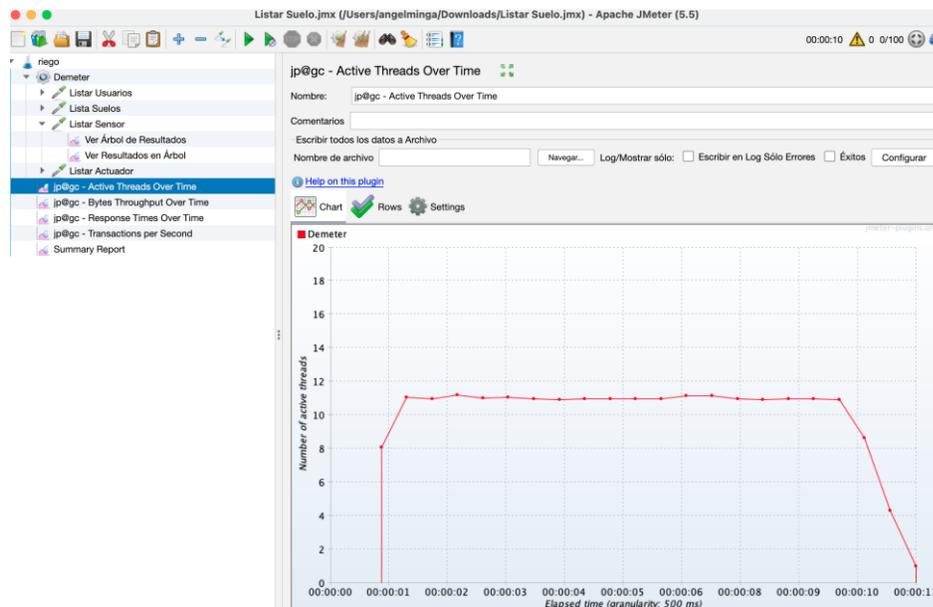


Figura 8 Resultados Active Threads Over Time

En la figura 8 se visualiza cuántos subprocesos activos hay en cada grupo de subprocesos durante la ejecución de la prueba, inicia con 8 subprocesos en 1 segundo, y existe un incremento de subprocesos y se mantiene asta finalizar todos los subprocesos en 11 segundos.

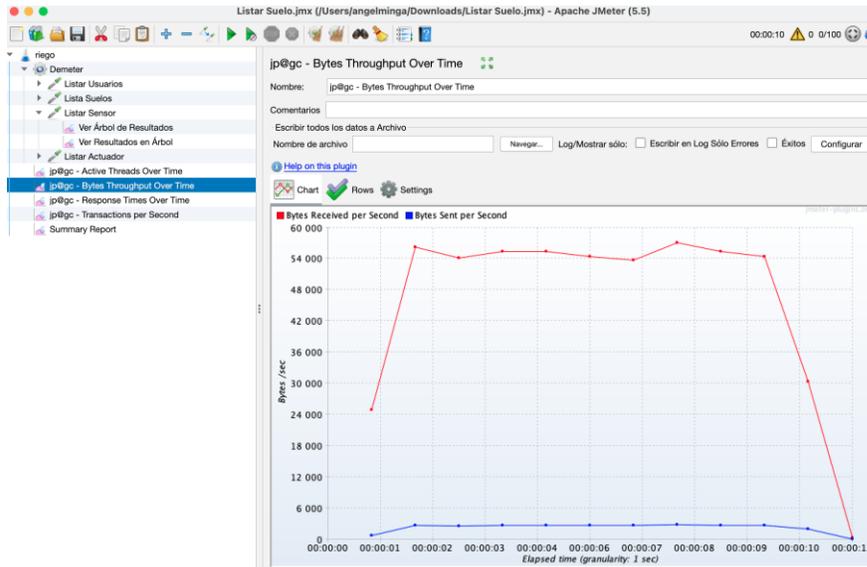


Figura 9 Resultados Bytes Throughput Over Time

En la figura 9 se visualiza que al aumentar la cantidad de subprocesos, JMeter generará más subprocesos nuevos y todos ejecutarán la solicitud que tiene en su prueba en paralelo, por lo tanto, entre más solicitudes enviadas simultáneamente que es un aumento total de bytes recibidos por segundo.

Lista actuador

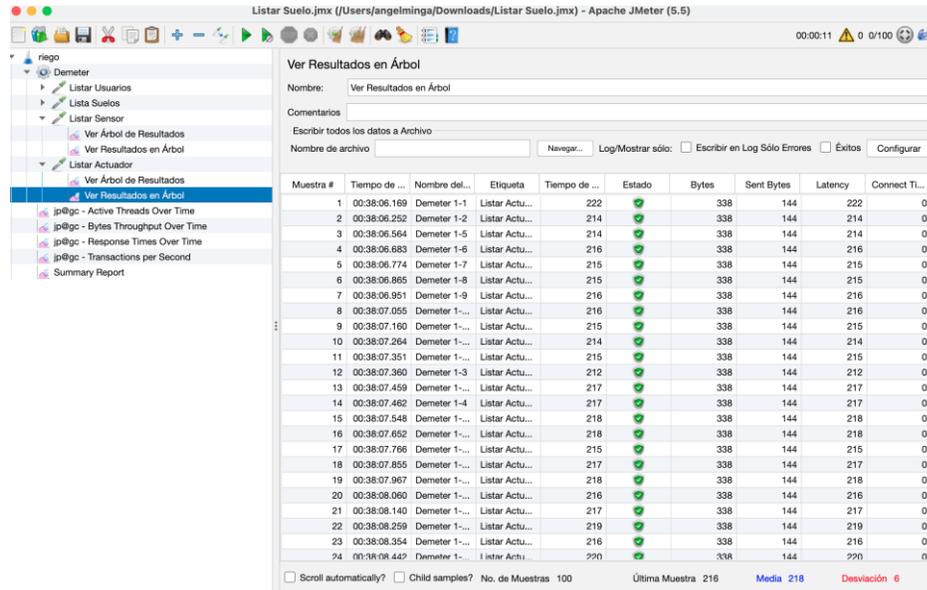


Figura 10 Resultados carga y estrés listar actuador

En la figura 10 se visualiza la prueba realizada de peticiones al recurso lista de actuador en el cual se visualiza el tiempo de espera, latencia, entre otros datos entre cada petición.

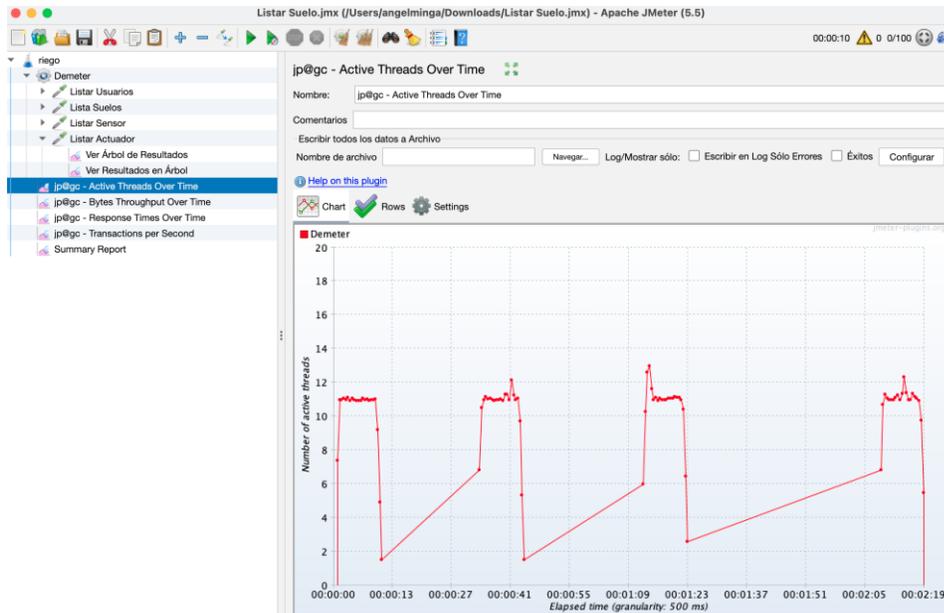


Figura 11 Resultados Active Threads Over Time

En la figura 11 se visualiza cuántos subprocesos activos hay en cada grupo de subprocesos durante la ejecución de la prueba, antes del primeros 13 segundos se visualiza que existe inica con 8 subprocesos, y al transcurrir el tiempo se visualiza que existe un aumento de

subprocesos, y disminucion del mismo, asta culminar toda los subproces en 2 minutos y 19 segundos.

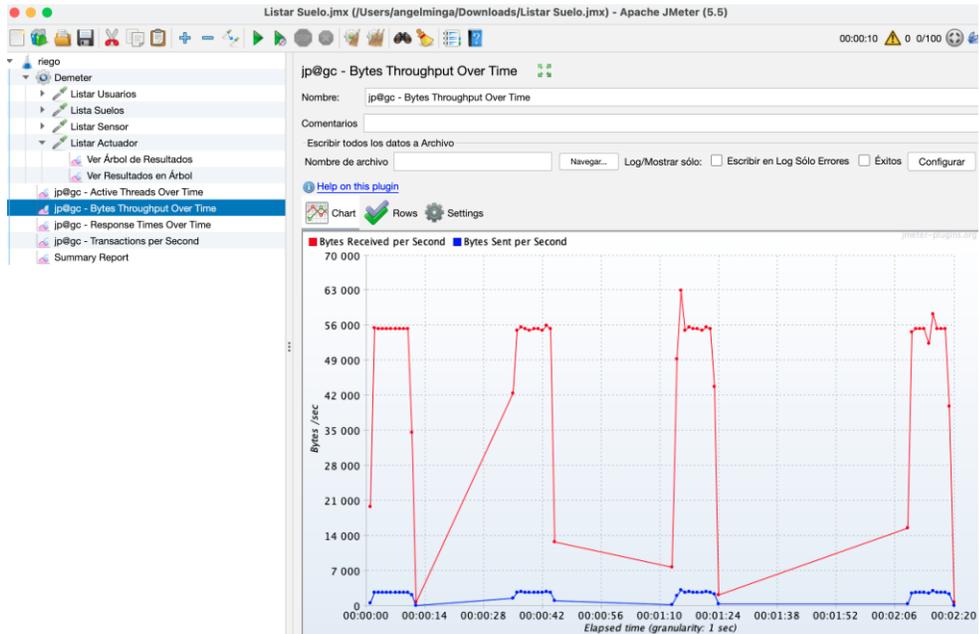


Figura 12 Resultados Bytes Throughput Over Time

En la figura 12 se visualiza que al aumenta de subprocesos, JMeter generará más subprocesos nuevos y todos ejecutarán la solicitud que tiene en su prueba en paralelo, por lo tanto, más subprocesos, más solicitudes enviadas simultáneamente que es un aumento total de bytes recibidos por segundo.

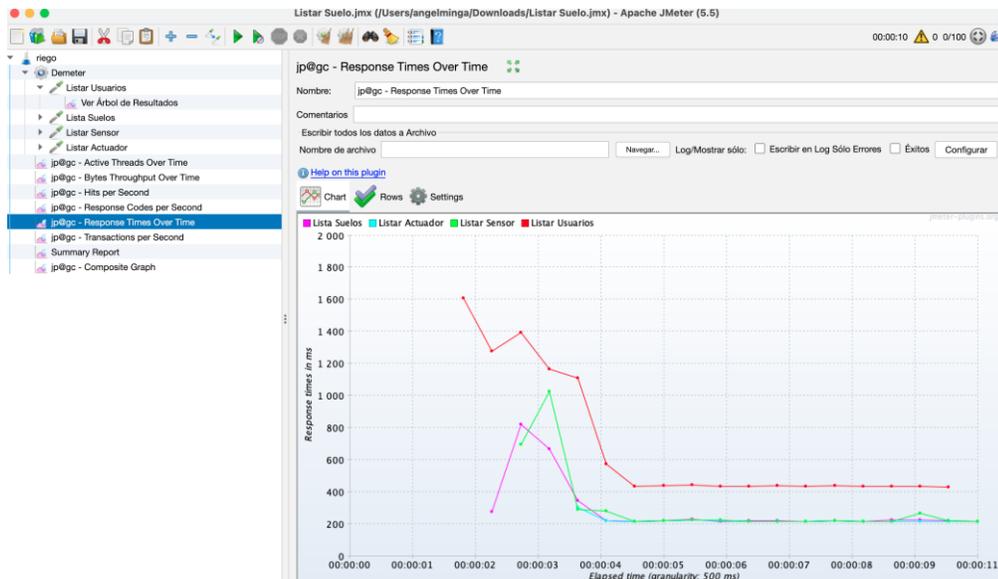


Figura 13 Response Times Over Time

En la figura 13 se visualiza JMeter mide el tiempo transcurrido desde justo antes de enviar la solicitud hasta justo después de recibir la última respuesta, el rendimiento se calcula como solicitudes/unidad de tiempo. Esto incluye cualquier intervalo entre muestras, ya que se supone que representa la carga en el servidor. La fórmula es: Rendimiento = (número de solicitudes) / (tiempo total).

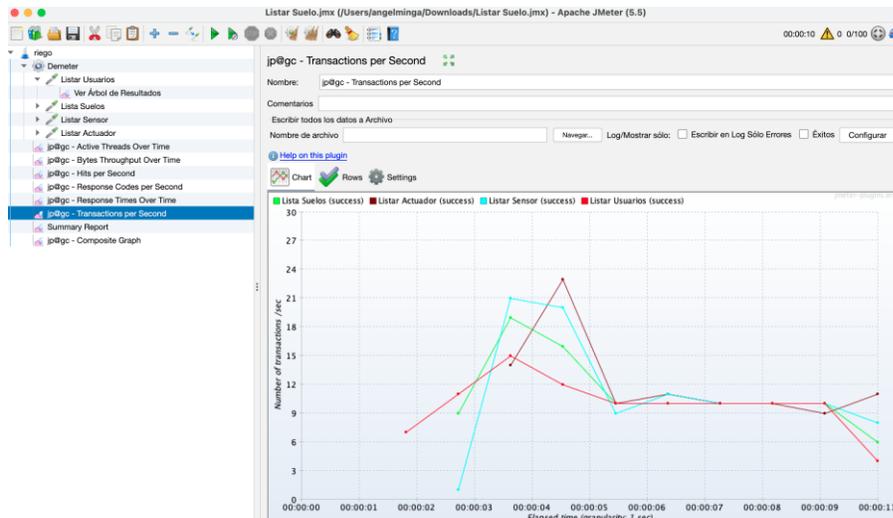


Figura 14 Transactions por Second

Transactions por Second (Transacciones por segundo)

En la figura 14 se presenta el rendimiento total de transacciones del servidor para subprocesos de prueba activos. Básicamente, muestra el máximo número estadístico posible de transacciones en función del número de usuarios que acceden al recurso.

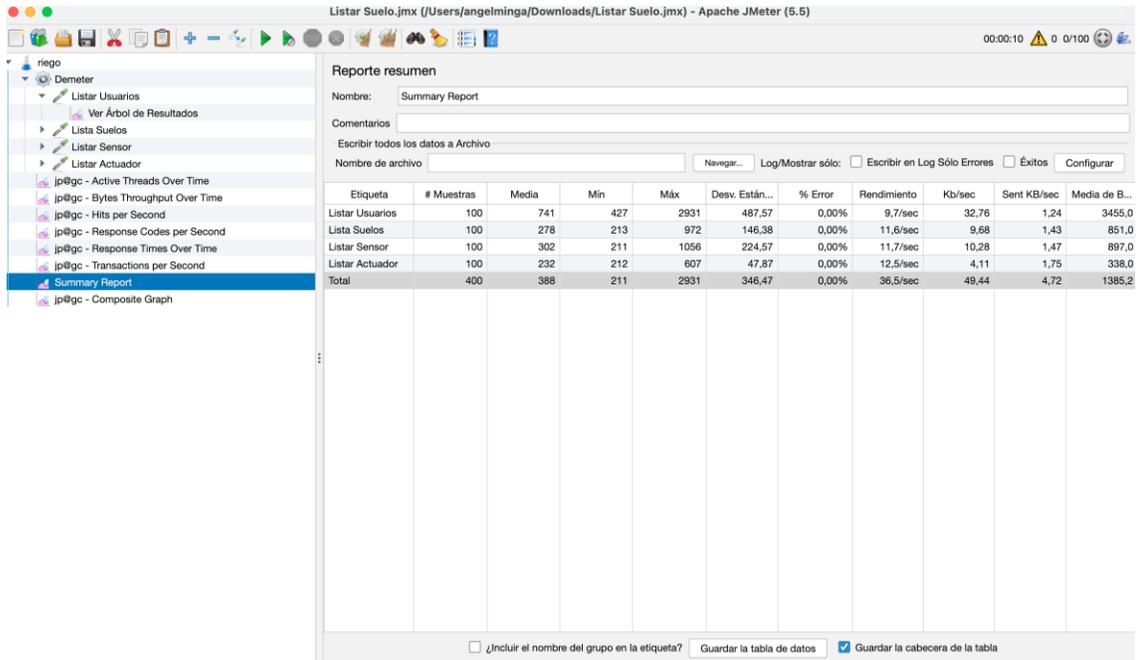


Figura 15 Summary Report

Summary Report

Permite obtener el tiempo absoluto durante el cual se han producido las solicitudes. Como se visualiza en la figura 15 se presenta el tiempo total durante el cual se han generado las solicitudes. Si hay otros muestreadores y temporizadores en el mismo subproceso, aumentarán el tiempo total y, por lo tanto, reducirán el valor de rendimiento.

2. Anexos

Se presenta en enlace de JSON exportado por la herramienta de Google Postman en donde se realizaron todas las pruebas unitarias del sistema de irrigación con IoT para la Universidad Nacional de Loja.

<https://drive.google.com/file/d/1UfhSj5hyMSjdiO997WYba8tnSse0EQUC/view?usp=sharing>

Anexo 7 Experimentación

“Validación del funcionamiento de sensores de humedad de suelo para sistemas de riego de precisión utilizando IoT”

Nole-Reyes, Alex Santiago. Minga-Median, Angel Favian. Guamán-Quinche, Edwin René. Chamba-Zaragocin, Diego Fernando. Ramón-Hurtado, Manuel Romel. Giron-Eraza, Wilson Josue

{alex.nole,afmingam, rguaman, diego.f.chamba, manuel.r.ramon, wilson.giron}@unl.edu.ec

Universidad Nacional de Loja
CIS

Resumen— El presente estudio evaluó el funcionamiento de sensores higrómetros [fc-28], para obtener la humedad del suelo, en relación con dos tipos de texturas de suelo, mediante el método gravimétrico, con el cual se obtuvo el valor de la humedad de acuerdo a los pesos de los sustratos por medio de los sensores en el transcurso de 32 días. Con el desarrollo del experimento y las diferentes repeticiones se vio que los sensores tienen una gran variabilidad en la recolección de datos, esto puede ser asumido por la diferente composición de los suelos o a la presencia de sales que puede afectar la conductividad eléctrica, por lo que es de importancia la realización de la curva de calibración, previo a la instalación definitiva en suelo, logrando obtener así la regresión y características de los sensores para dos clases de suelos texturales.

I. DEFINICIÓN DEL ALCANCE

Validar la recepción de datos de los sensores de humedad de suelo [fc-28] con el propósito de evaluar los datos arrojados y la transmisión de los sensores con respecto al contenido de humedad según la textura de suelos franco-limosos y arenosos, desde el punto de vista experimental en el contexto de un prototipo de sistema de riego con IoT.

II. PLANIFICACIÓN

A. Selección del contexto

El objeto experimental consiste en validar que los sensores [fc-28] perciban el cambio de humedad del suelo, utilizando el método gravimétrico para el sistema de irrigación con IoT, en primera instancia realizados a nivel de Laboratorio para luego trasladar la metodología a cultivos de la Universidad Nacional de Loja.

La humedad del suelo se obtuvo de dos orígenes diferentes:

- a) Humedad del suelo obtenida de forma manual [1](método gravimétrico en laboratorio).
- b) Humedad del suelo obtenida con los sensores de humedad del suelo fc-28

En el primer caso se utilizó varios recipientes con diferentes tipos de suelo, con la finalidad de obtener la humedad mediante el método gravimétrico, previamente se determinó la densidad del suelo, la Capacidad de campo [CC], punto de marchitez permanente [PMP], para luego saturar el suelo e ir tomando medidas del peso de recipiente (g) hasta que llega el suelo a secarse completamente. En el segundo caso se utiliza los sensores para determinar el porcentaje de humedad del suelo en un tiempo determinado y compararlo con los pesos obtenidos por el método anterior, en el día y hora de medidas.

Para este experimento, se decidió utilizar solo dos tipos texturales de suelos, mismos que tiene una velocidad de infiltración del agua diferente.

B. Selección de las muestras

En el experimento se utilizó dos tipos de suelo, Franco limoso y arenoso, que cuentan con propiedades físicas como la infiltración, contenido de humedad a capacidad de campo y punto de marchitez permanente diferentes, lo que permite validar que los sensores detecten la humedad en varias condiciones de suelo.

C. Selección de variables

- **Variable dependiente**

La variable dependiente es la precisión de los sensores al capturar la humedad del suelo en diferentes texturas.

- **Variables independientes**

Las variables independientes son los métodos que se utilizan para obtener la humedad del suelo.

D. Elección del diseño

Se seleccionó el diseño estadístico experimental para la obtención de datos apropiados y luego fueron analizados mediante ANOVA con nivel de confianza 95% y correlaciones lineales con el fin de obtener la variabilidad en los datos de humedad y que pueden atribuirse a diferentes causas, ya sea de tipo de suelo, o la precisión de medición del

sensor [2]. Para el desarrollo del experimento se seleccionó el método gravimétrico para realizar la evaluación del funcionamiento de los sensores [3], el cual se utiliza agregando en 6 contenedores dos tipos de suelo. En donde se obtiene el peso de los contenedores con el suelo seco. Seguidamente se les agrega 7 oz de agua y se deja reposar por 48 horas que es cuando el suelo está en capacidad de campo [4]. Luego de haber transcurrido 48 horas en reposo se realiza la medición del peso de cada recipiente con sustratos para determinar la humedad mediante la humedad gravimétrico(wg). Del mismo modo se realiza la medición con los sensores de humedad en capacidad de campo del suelo. Posteriormente se realizó el mismo proceso por 2 repeticiones más, esto saltando 3 días para verificar que va disminuyendo la humedad en cada muestra y con ello se puede constatar que los sensores están detectando correctamente la humedad.

Los valores [1] gravimétricos se obtuvieron mediante la fórmula:

$$wg = \left(\frac{\text{peso húmedo} - \text{peso seco}}{\text{peso seco}} \right) * 100$$

E. Instrumentación

En esta tarea, se prepararon los sensores de higrómetros [fc-28], conectándolos a una tarjeta programable que contienen un microcontrolador programable [Esp-32] que se enlaza mediante wifi al bróker que está instalado en la microcomputadora Raspberry Pi. Del mismo modo se tiene preparado los recipientes con los tipos de suelo que se va a utilizar para comprobar que los sensores envían los datos de acuerdo a la humedad del suelo, los sensores se encuentran instalados a una profundidad de 4.7cm.

Cabe destacar que durante la instrumentación se diseñó una plantilla en Excel para almacenar los datos obtenidos por los sensores y realizar los cálculos de manera manual para comparar la humedad en un tiempo específico. En la tabla I y tabla II se puede apreciar las tablas creadas para llevar registro de los valores obtenidos por los sensores y por el método gravimétrico.

Tabla 21
Valores de humedad de suelo Arenoso

Días		0						4						8					
T. Suelo	Vaso	Peso Seco	Peso 1 o CC	Humedad	Sensor 1	Sensor2	Sensor 3	Peso 2	Humedad	Sensor 1 %	Sensor2 %	Sensor 3 %	Peso 2	Humedad	Sensor 1 %	Sensor2 %	Sensor 3 %		
	AR-01	216,3	252,5	16,73	71	71	62	241,5	11,65	49	55	59	233,3	7,85	29	45	44		
Arenoso	AR-02	210,4	248,8	18,25	61	69	66	236,7	12,5	43	59	67	226,2	7,50	29	35	35		
	AR-03	196,9	233,2	18,43	67	61	67	221,5	12,49	39	55	64	211,6	7,46	30	38	35		
				17,80	66,3	67	65		12,21	43,66	56,33	63,33		7,61	29,33	39,33	38		
Fecha	1-oct-22				3-oct-22					05-oct-22					8-oct-2022				

Tabla 22
Valores de humedad de suelo Franco Limoso

Días		0						4						8					
T. Suelo	Vaso	Peso Seco	Peso 1 o CC	Humedad	Sensor 1	Sensor2	Sensor 3	Peso 2	Humedad	Sensor 1 %	Sensor2 %	Sensor 3 %	Peso 2	Humedad	Sensor 1 %	Sensor2 %	Sensor 3 %		
Franco Limoso	FL-01	121,4	142,2	17,13	72	70	82	131,9	8,64	64	74	73	122	0,49	36	48	39		
	FL-02	152,5	177,8	16,59	72	70	75	166,1	8,91	70	73	73	155,8	2,16	42	30	36		

	FL-03	124,5	149,8	20,32	68	72	85	139,2	11,80	73	75	78	131,5	5,62	61	63	55
				18,01	70,6	70,6	80,66		9,791	69	74	74,66		2,76	46,33	47	43,33
Fecha	1-oct-22				3-oct-22					05-oct-22						8-oct-2022	

III. OPERACIÓN

A. *Preparación*

Previo a la ejecución del experimento se realizó la búsqueda de cómo realizar el método gravimétrico para la calibración de sensores con la finalidad de utilizar de forma correcta las fórmulas y conocer los tiempos para obtener los datos de los sensores. Para ello, ante todo, se realizó un estudio piloto utilizando una maceta con tipo de suelo franco limoso para ver el funcionamiento de los sensores.

Para realizar el experimento se utilizó los tipos de suelo arenoso y franco limoso. Estos dos tipos de suelo cuentan con características diferentes, que son importantes para el experimento ya que cada uno puede almacenar diferentes cantidades de humedad dentro de su textura, como es el ejemplo del tipo de suelo franco limoso, este contiene la humedad por más tiempo porque sus poros son más pequeños en relación al arenoso, por lo cual la infiltración es menor; en el caso del arenoso sus poros son más grandes por ende tiene una infiltración elevada.

B. *Ejecución*

El experimento se realizó en un ambiente controlado. Realizando 3 repeticiones con cada sensor, para obtener la humedad del suelo en un tiempo determinados. Es importante mencionar que se utilizó la misma cantidad de agua en cada interacción para evitar sesgos en los resultados. Los datos obtenidos por los sensores se los agregaron al Excel antes preparado para luego comparar los resultados con cada uno de los sensores.

Cabe destacar que adicionalmente se realizó un ensayo donde se utilizaron dos macetas con los tipos de suelo escogidos anteriormente, aquí se recolectaron los datos cada 48h. En capacidad de campo se agregó 16 ml de agua y con ello se verifican que los sensores percibían un cambio de humedad cada vez que se le agrega más agua. En la Tabla III se presentan los valores obtenidos en el proceso de recolección de datos de humedad de suelo de la última repetición.

Tabla 23

Valores de datos de última repetición

Días		0									2									4									6								
T. Suelo	Peso Recipiente	Peso Seco	Peso Saturado	Humedad-Seco			Humedad-Saturado			Peso CC.	Peso Saturado	Humedad-CC			Humedad-Saturado			Peso CC.	Peso Saturado	Humedad-CC			Humedad-Saturado			Peso CC.	Peso Saturado	Humedad-CC			Humedad-Saturado						
				S1	S2	S3	S1	S2	S3			S1	S2	S3	S1	S2	S3			S1	S2	S3	S1	S2	S3			S1	S2	S3	S1	S2	S3	S1	S2	S3	
Arenoso	106,7	181 4,37	2267 ,96	0	0	0	7	7	7	220 8,99	2317 ,86	6	6	6	5	5	5	244 0,33	2503 ,83	5	5	5	5	5	5	242 2,18	2494 ,76	4	3	4	5	5	4				
Francolimoso	109,6	771. 11	879. 97	3	0	2	6	5	6	115 6,66	1260 ,99	6	4	6	6	6	6	134 7,17	1387 ,99	7	6	6	7	7	7	132 4,49	1419 ,74	6	6	7	7	6	6				
Fecha		15-nov-22									17-nov-22									19-nov-22									21-nov-22								

C. *Validación de datos*

Después de la ejecución del experimento se obtuvieron los datos de la humedad del suelo por medio del método gravimétrico y mediante los datos obtenidos por los sensores. Aquí se realizó un promedio de la humedad obtenida manualmente y con ello se realizó una comparativa mediante correlaciones con los valores obtenidos con los sensores, para así asignar un valor de porcentaje con el cual se tiene que abrir el riego.

IV. ANÁLISIS E INTERPRETACIÓN

A. *Estadísticos descriptivos*

El procedimiento seguido para analizar los datos es:

- a) Se unió los datos obtenidos en las diferentes repeticiones, para así formar un todo de los datos obtenidos en los experimentos y estos fueron separados por tipos de suelo.
- b) Se realizaron análisis de varianza y gráficos de regresión para cada repetición con la finalidad de realizar el estudio de los datos.
- c) Comparar los datos en el transcurso de los días para verificar que los sensores envían valores con los cambios de humedad.

B. *Análisis de los datos se humedad de suelo*

Se encontró que los sensores son sensibles al cambio de humedad, esto por los datos obtenidos en el transcurso de 32 días que duró la experimentación con los dos tipos de suelos escogidos. Como se puede ver en las figuras 1 se muestran las gráficas de los promedios en base a los datos obtenidos en cada repetición con el tipo de suelo arenoso. La humedad va disminuyendo en relación con el tiempo (32 días transcurridos) desde el día que se saturó las muestras.

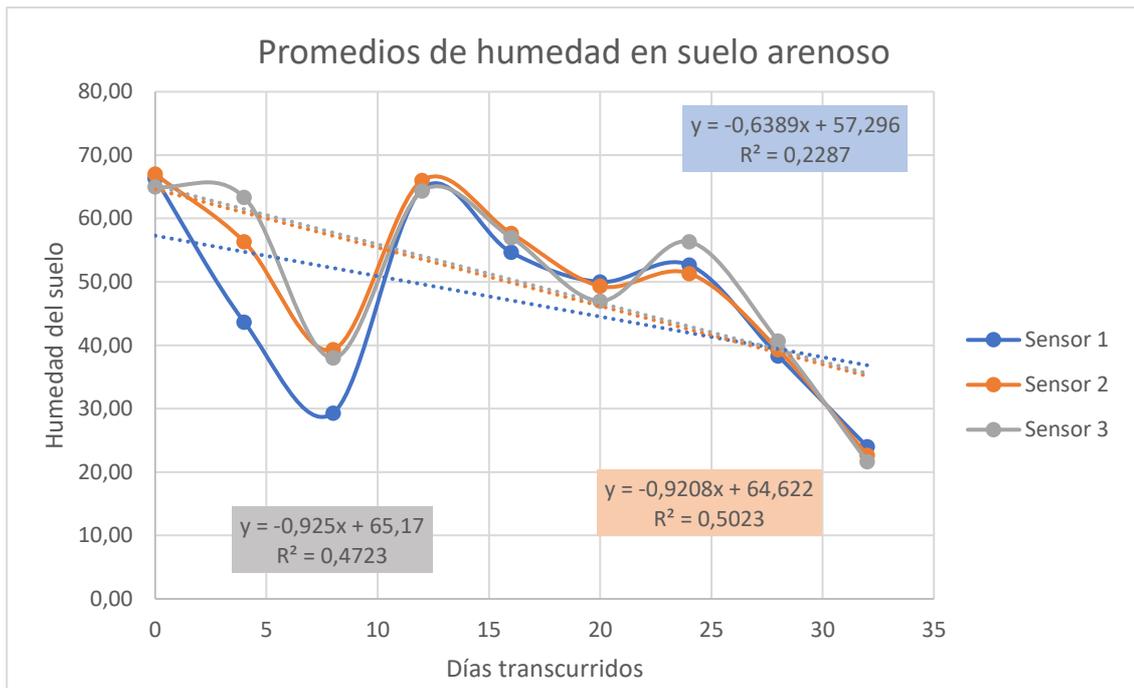


Figura 85 Gráfica de dispersión de los promedios de humedad obtenidas en el tipo de suelo arenoso

En las repeticiones realizadas para el tipo de suelo franco limoso se encontró de igual forma que en las repeticiones de suelo arenoso, que los sensores perciben un cambio en la humedad de suelo, según van transcurriendo los días desde que se saturó el suelo. Cabe destacar que con este tipo de suelo los sensores obtienen datos de humedad más altos que del arenoso, esto porque el suelo franco limoso, retiene más humedad. En las figuras 2, se presentan las gráficas de los promedios de humedad obtenidas con las tres repeticiones realizadas al tipo de suelo.

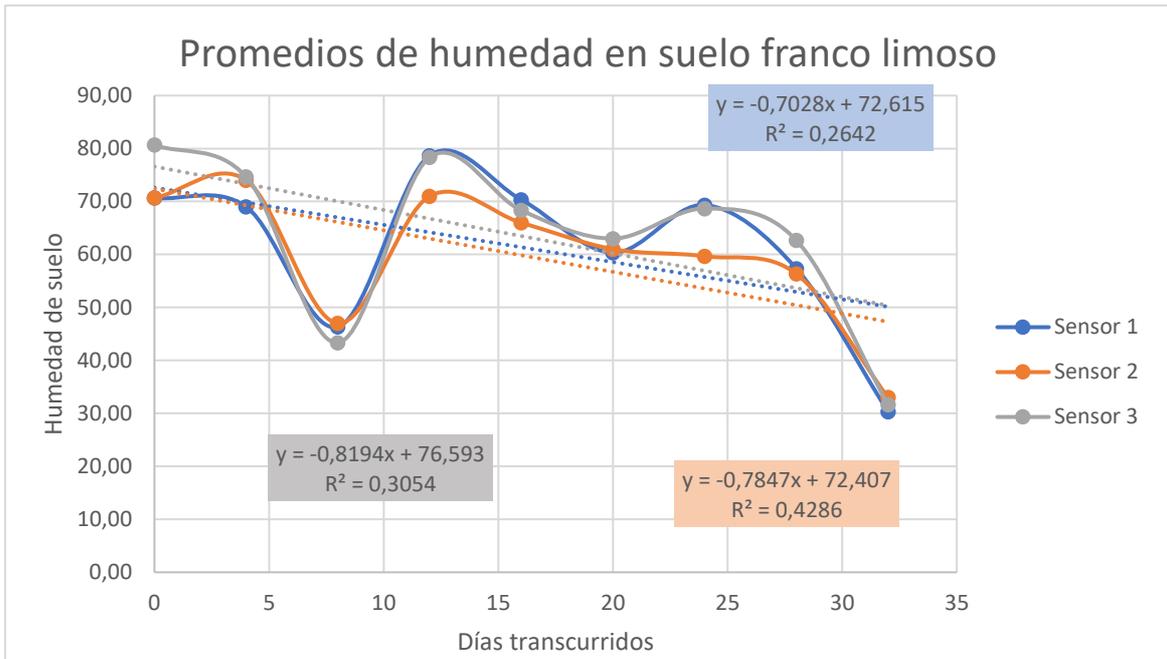


Figura 86 Gráfica de dispersión de los promedios de humedad obtenidas en el tipo de suelo franco limoso

En las últimas repeticiones realizadas cada 48 horas, utilizando dos macetas en donde se las llenó con los dos tipos de suelo, y agregando 16 ml de agua cada 48 horas, que es cuando el suelo está en capacidad de campo. Se realizaron estas últimas repeticiones para corroborar que los sensores determinan la humedad en capacidad de campo y cuando está saturado. En las figuras 3,4 se presentan las gráficas obtenidas por los datos de los sensores.

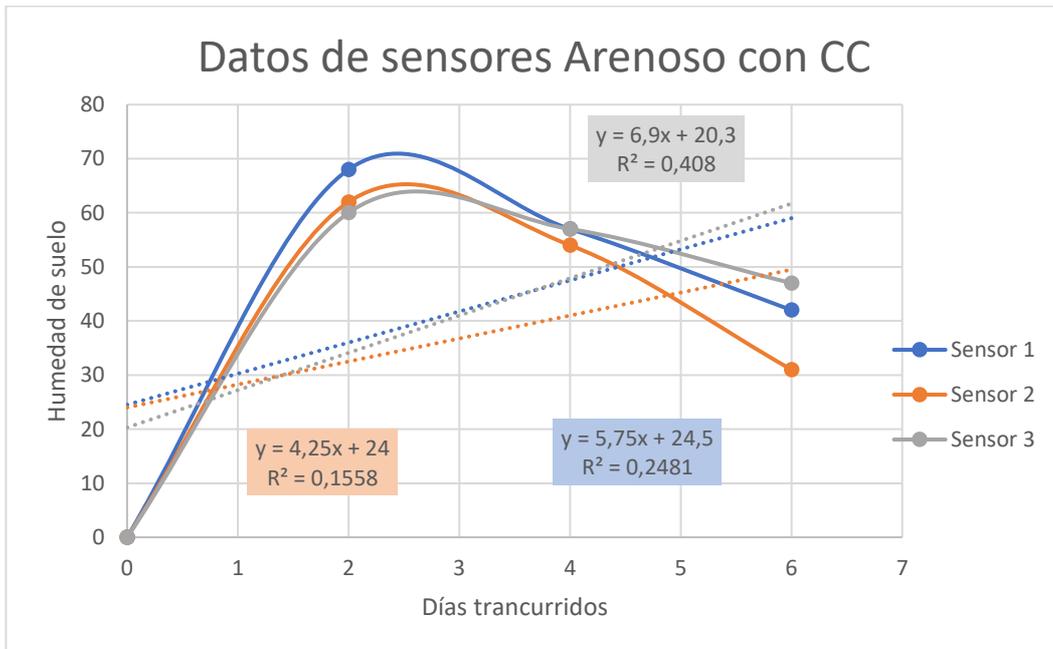


Figura 87 Gráfica de dispersión de repetición final en suelo Arenoso en Capacidad de Campo

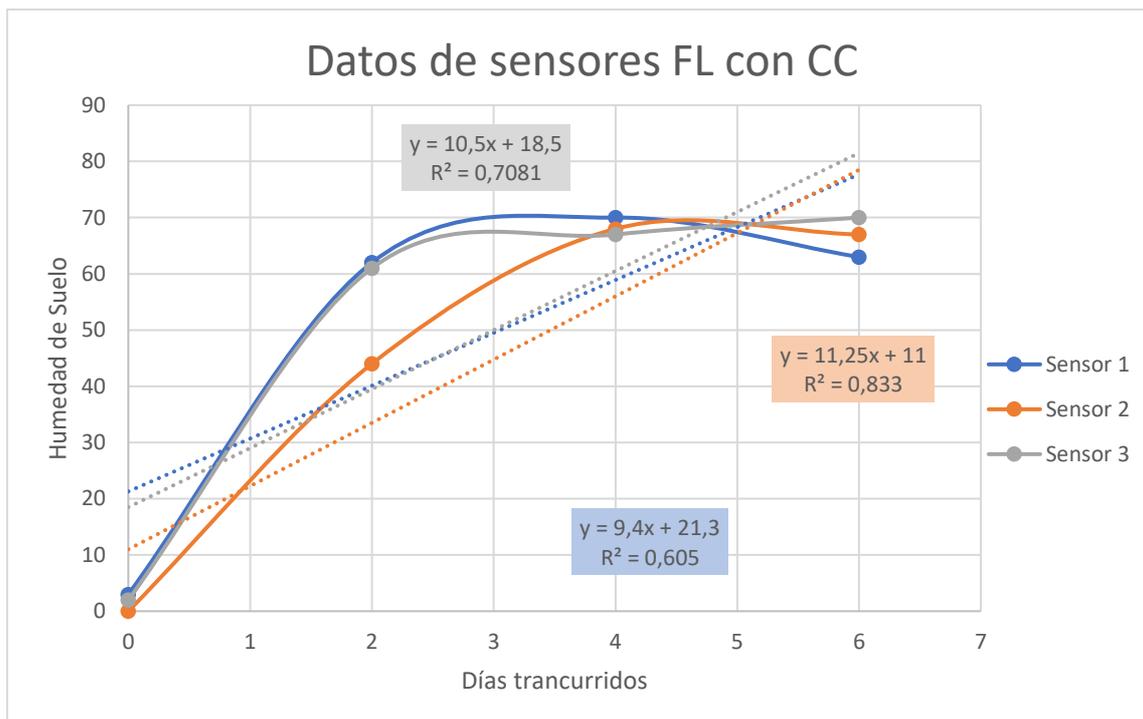


Figura 88 Gráfica de dispersión de repetición final en suelo Franco Limoso en Capacidad de Campo

Como se puede apreciar en las gráficas obtenidas en cada repetición realizada en el experimento. Los sensores son sensibles a los cambios de humedad de suelo, lo que permite determinar cuándo se puede regar, esto realizando una comparativa entre los valores teóricos cuando un suelo está en capacidad y los valores arrojados por los sensores.

Cabe destacar que los sensores envían datos variables de la humedad ya que los sensores toman en cuenta las sales que contiene el suelo como parte de la humedad. Por lo cual es necesario realizar la calibración de estos, y así tener mayor precisión con los datos obtenidos.

V. CONCLUSIÓN

Los sensores utilizados para el desarrollo del prototipo son sensibles a la humedad del suelo, lo que arrojan buenas respuestas, pero estos necesitan de una calibración para obtener mayor eficiencia al momento de capturar los datos, ya que estos toman en cuenta los nutrientes del suelo como humedad, lo que hace que los datos varíen en un gran porcentaje.

El desarrollo de la experimentación permitió obtener un promedio de la humedad del suelo mediante una comparativa de correlaciones con los valores obtenidos con los sensores, estos datos son necesarios para asignar un valor de porcentaje con el cual se tiene que abrir el riego en la aplicación.

VI. REFERENCIAS

[1] V. Serman, C. Schugurensky, R. Carrión, and S. Rodríguez, “EVALUACIÓN DEL COMPORTAMIENTO DE SENSORES DE HUMEDAD DE SUELO DEL TIPO (FDR) DE DESARROLLO LOCAL, EN RELACIÓN AL CONTENIDO DE AGUA Y A LA TEXTURA DE SUELO.”

[2] Francisco Jiménez Otárola, “Planificacion_y_manejo_de_los_recursos”.

[3] R. Radulovich, “Método gravimétrico para determinar in situ la humedad volumétrica del suelo.” [Online]. Available: www.cia.ucr.ac.cr

[4] M. M. Valdivieso, “Producción Agropecuaria sostenible en suelos Arcillosos del piso temperado Andino del Sur del Ecuador.”

Anexo 8 Certificados de Pruebas de Usabilidad



Loja 10 de enero del 2023

Ing. Manuel Romel Ramón Hurtado
Docente de la Carrera de Ingeniería Agrícola

Certifico

Que el Señor Egresado de la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, Alex Santiago Nole Reyes con cedula de identidad 1105642076, realizo las pruebas de la aplicación web y aplicación móvil para el proyecto de Integración Curricular denominado **“Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja”**, por lo que se da fe de la realización de las pruebas, el cual concluyó que cumple con los requerimientos del usuario y hay la aceptación satisfactoria de la solución informática.

Es todo cuanto puedo certificar en honor a la verdad.

 Firmado digitalmente por:
MANUEL ROMEL
RAMON HURTADO
Ing. Manuel Romel Ramón Hurtado
Docente de la Carrera de Ingeniería Agrícola
C.I. 1102367248



unl

Universidad
Nacional
de Loja

Loja, 10 de enero del 2023

Ing. Diego Fernando Chamba Zaragocín, PhD.

Docente de la carrera de Ingeniería Agrícola de la Universidad Nacional de Loja

Certifico

Que el Señor Egresado de la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, **Angel Favian Minga Medina** con cedula de identidad 1104695190, realizó las pruebas de la aplicación web y aplicación móvil para el proyecto de trabajo de titulación denominado “**Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja**”, lo que se da fe de la realización de las pruebas, con el cual concluyó que cumple con los requerimientos del usuario y hay la aceptación satisfactoria de la solución informática y su aplicación referente al manejo de riego.

Es todo cuanto puedo certificar en honor a la verdad.



Firmado electrónicamente por:
DIEGO FERNANDO
CHAMBA ZARAGOCIN

Ing. Diego Fernando Chamba Zaragocín, PhD

Docente de la carrera de Ingeniería Agrícola de la Universidad Nacional de Loja

C.I. 1104011588

Loja, 10 de enero del 2023

Ing. Diego Fernando Chamba Zaragocín, PhD.

Docente de la carrera de Ingeniería Agrícola de la Universidad Nacional de Loja

Certifico

Que el Señor Egresado de la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, **Alex Santiago Nole Reyes** con cedula de identidad 1105642076, realizó las pruebas de la aplicación web y aplicación móvil para el proyecto de trabajo de titulación denominado “**Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja**”, lo que se da fe de la realización de las pruebas, con el cual concluyó que cumple con los requerimientos del usuario y hay la aceptación satisfactoria de la solución informática y su aplicación referente al manejo de riego.

Es todo cuanto puedo certificar en honor a la verdad.



Firmado electrónicamente por:
DIEGO FERNANDO
CHAMBA ZARAGOCIN

Ing. Diego Fernando Chamba Zaragocín, PhD

Docente de la carrera de Ingeniería Agrícola de la Universidad Nacional de Loja

C.I. 1104011588



unl

Universidad
Nacional
de Loja

Loja 10 de enero del 2023

Ing. Manuel Romel Ramón Hurtado
Docente de la Carrera de Ingeniería Agrícola

Certifico

Que el Señor Egresado de la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, Ángel Favian Minga Medina con cedula de identidad 1104695190, realizo las pruebas de la aplicación web y aplicación móvil para el proyecto de Integración Curricular denominado **“Sistema de irrigación utilizando IoT para cultivos de la Universidad Nacional de Loja”**, por lo que se da fe de la realización de las pruebas, el cual concluyó que cumple con los requerimientos del usuario y hay la aceptación satisfactoria de la solución informática.

Es todo cuanto puedo certificar en honor a la verdad.



Firmado electrónicamente por:
MANUEL ROMEL
RAMON HURTADO

Ing. Manuel Romel Ramón Hurtado
Docente de la Carrera de Ingeniería Agrícola
C.I. 1102367248

Anexo 10. Manual de Usuario

**FACULTAD DE ENERGÍA, LAS INDUSTRIAS Y
LOS RECURSOS NATURALES NO
RENOVABLES**

**CARRERA DE INGENIERÍA EN
SISTEMAS / COMPUTACIÓN**

Manual de usuario

Sistema de riego UNL

Tabla de contenido

1. OBJETIVO	264
2. DEFINICIONES	264
3. DESARROLLO DEL MANUAL DE USUARIO	265
3.1. Formulario Inicio de sesión	265
3.2. Menú	266
3.3. Formulario perfil de usuario	267
3.4. Formulario cambio de contraseña	267
3.5. Administración de usuarios	268
3.6. Formulario registro nuevo usuario	269
3.7. Formulario cambio de perfil	269
3.8. Formulario cambio de contraseña	270
3.9. Formulario editar datos de usuario	271
3.10. Formulario confirmación de eliminar usuario	271
3.11. Formulario presentación histórico de datos	272
3.12. Formulario características de suelo	273
3.13. Formulario editar características del suelo	273
3.14. Formulario confirmación de eliminar el suelo	274
3.15. Formulario presentación topics sensores	275
3.16. Formulario editar topic sensor	275
3.17. Formulario confirmación de eliminación de topic sensor	276
3.18. Formulario presentación topics actuadores	277
3.19. Formulario editar topic sensor	277
3.20. Formulario confirmación de eliminación de topic sensor	278

OBJETIVO

Establecer los pasos específicos para el uso del sistema de administración de riego UNL, con el fin de promover la interacción del usuario con el sistema

DEFINICIONES

Sistema de riego UNL: Es una aplicación que facilita el registro, actualización y administración de la información de los usuarios, topic, tipo de suelo, permite visualizar el histórico de los datos obtenidos por los sensores.

DESARROLLO DEL MANUAL DE USUARIO

3.1. Formulario Inicio de sesión



Inicio Sesión

Correo **1**

Contraseña **2**

INICIAR SESIÓN **3**

© 2023 Todos los derechos reservados. Universidad Nacional de Loja

1. Ingrese correo electrónico
2. Ingrese contraseña
3. Presiones ingresar

3.2. Menú



1. Vista principal
2. Perfil del usuario
3. Administración de usuarios
4. Histórico de datos
5. Administración de tipos de suelos
6. Administración de topic de sensores
7. Administración de topic de actuadores

Formulario perfil de usuario

Perfil de Usuario

Nombre	Administración
Apellido	Control
Teléfono	0999999999
Dirección	Loja-Ecuador
Correo	admin@admin.com
Contraseña	*** 1
ACTUALIZAR 2	

© 2023 Todos los derechos reservados. Universidad Nacional de Loja

En el formulario se presenta la información del usuario, que puede ser editada, también se presenta dos botones que permiten:

1. Permite abrir una vista de edición de contraseña.
2. Permite actualizar los datos del perfil del usuario.

Formulario cambio de contraseña

Contraseña Actual

Contraseña Nueva

1 ACTUALIZAR ✎ CANCELAR ✖ 2

El siguiente formulario permite ingresar la contraseña actual, la nueva contraseña, presenta dos botones que permiten:

1. Actualiza la contraseña
2. Cancela la actualización de la contraseña

Administración de usuarios

Administración de Usuarios

NUEVO 1

Nro	Nombre	Apellidos	Correo	Dirección	Teléfono	Rol	Contraseña	Acciones
1	Administración	Control	admin@admin.com	Loja-Ecuador	0999999999	ADMINISTRADOR	***	✎ ✖
2	Diego	Chamba	dfchamba@gmail.com	Loja-Ecuador	0999999999	ADMINISTRADOR	***	✎ ✖
3	Rene	Guaman	rguaman@unl.edu.ec	Loja-Ecuador	0999999999	ADMINISTRADOR	***	✎ ✖
4	Romel	Ramon	romelramon@gmail.com	Loja-Ecuador	0999999999	ADMINISTRADOR	***	✎ ✖
5	Wilson	Giron	wilgirerazojos@gmail.com	Loja-Ecuador	0999999999	ADMINISTRADOR	***	✎ ✖
6	Alex Santiago	Nole Reyes	snole7@gmail.com	Cdla. Los Operadores	0999724161	USUARIO	***	✎ ✖

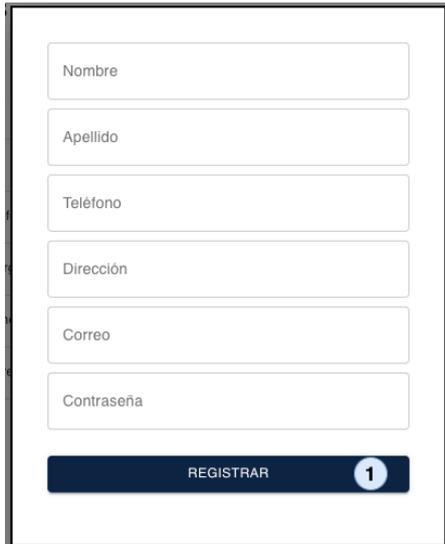
© 2023 Todos los derechos reservados. Universidad Nacional de Loja

En el siguiente formulario se visualiza la lista de todos los usuarios registrados, se presenta cinco botones que permiten:

1. Permite visualizar el formulario de nuevo registro
2. Presenta el formulario de cambio de perfil
3. Presenta el formulario de actualizar contraseña

4. Presenta el formulario de editar usuario
5. Presenta el formulario de confirmación de eliminar usuario

Formulario registro nuevo usuario



Formulario de registro de nuevo usuario con los siguientes campos de texto:

- Nombre
- Apellido
- Teléfono
- Dirección
- Correo
- Contraseña

Botón de acción: REGISTRAR (1)

En el siguiente formulario se debe registrar la información solicitada para el nuevo usuario, presenta un botón que permite:

1. Permite registrar un usuario.

Formulario cambio de perfil



Formulario de cambio de perfil con los siguientes elementos:

- Campo de texto: Nombre Rol Actual (Administrador)
- Lista desplegable: Select... (▼)
- Botón de acción: ACTUALIZAR (1) (2)
- Botón de acción: CANCELAR (2)

El siguiente formulario permite seleccionar el perfil del usuario (Administrador o Usuario), presenta dos botones que permiten:

1. Permite actualizar el perfil
2. Cancela la actualización del perfil

Formulario cambio de contraseña

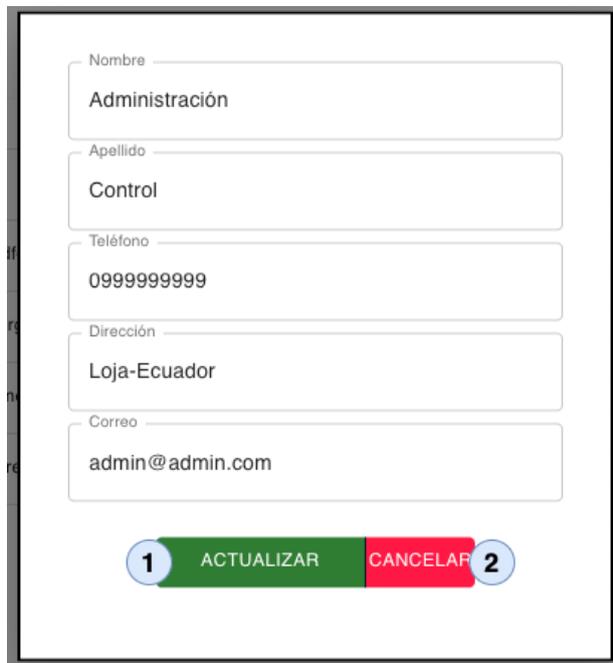


El formulario muestra dos campos de entrada de texto. El primer campo está etiquetado como 'Contraseña Actual' y el segundo como 'Contraseña Nueva'. Debajo de los campos, hay una barra de botones. El primer botón, etiquetado como '1', es verde y contiene el texto 'ACTUALIZAR' y un ícono de una flecha hacia abajo y a la izquierda. El segundo botón, etiquetado como '2', es rojo y contiene el texto 'CANCELAR' y un ícono de una 'X' roja.

El siguiente formulario permite ingresar la contraseña actual, la nueva contraseña, presenta dos botones que permiten:

1. Actualiza la contraseña
2. Cancela el cambio de la contraseña

Formulario editar datos de usuario



Formulario de edición de datos de usuario. El formulario contiene los siguientes campos de texto:

- Nombre: Administración
- Apellido: Control
- Teléfono: 0999999999
- Dirección: Loja-Ecuador
- Correo: admin@admin.com

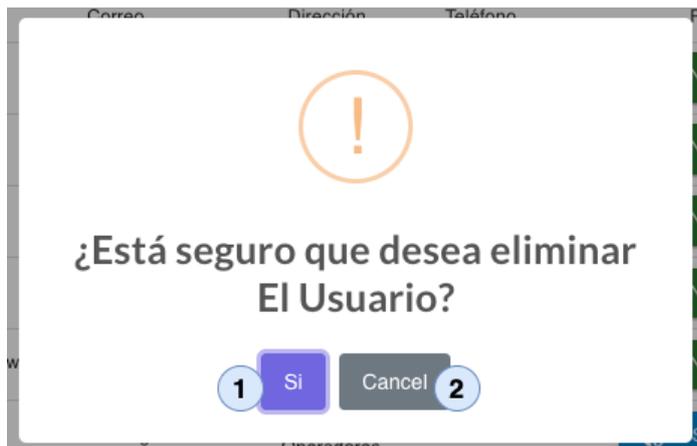
En la parte inferior del formulario hay dos botones:

- 1 ACTUALIZAR (botón verde)
- CANCELAR 2 (botón rojo)

El siguiente formulario permite editar la información del usuario, presenta dos botones que permiten:

1. Actualiza la información del usuario
2. Cancela la actualización del usuario

Formulario confirmación de eliminar usuario

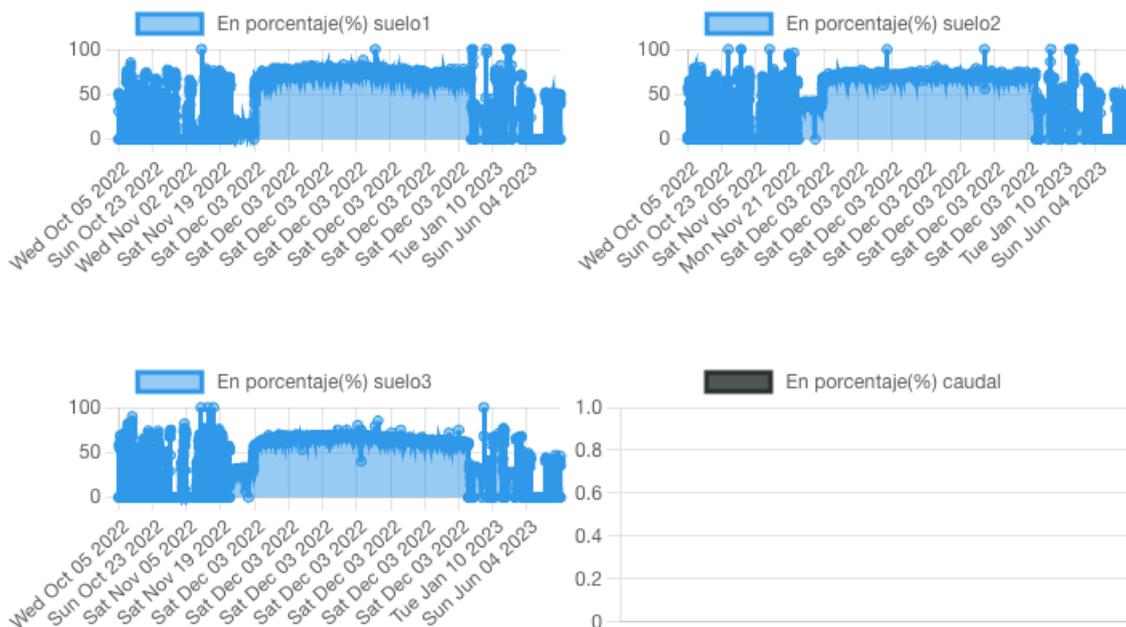


El siguiente formulario permite confirmar si desea eliminar el usuario, presenta dos botones que permiten:

1. Elimina la información
2. Cancela la eliminación del usuario

Formulario presentación histórico de datos

Historicos de Datos



El siguiente formulario presenta el histórico de captura de datos del suelo por medio de los sensores.

Formulario características de suelo

Características de Suelo

REGISTRAR 1

Nro	CLase Textural	Arcilla (%)	Limo (%)	Arena (%)	CC	PMP	Acciones
1	Franco Limoso	24.56	54	12.44	23.44	13.11	2   3
2	Franco Arcilloso	70	20	10	39.69	23.33	 

El siguiente formulario permite visualizar las características del tipo de suelo registrados, permite registrar las características de suelo, presenta tres botone que permiten:

1. Registrar las características del suelo
2. Permite cargar los datos del suelo en el formulario para editarlos
3. Presenta una alerta de confirmación para eliminar el tipo de suelo registrado

Formulario editar características del suelo

Características de Suelo

Clase Textural	Franco Limoso
(%) Arcilla	24.56
(%) Limo	54
(%) Arena	12.44



El siguiente formulario permite editar las características del tipo de suelo seleccionado, presenta dos botone que permiten:

1. Actualizar las características del suelo
2. Cancela la actualización

Formulario confirmación de eliminar el suelo

¿Está seguro que desea eliminar el tipo de suelo?

1 Si Cancel 2

El siguiente formulario permite confirmar si desea eliminar el tipo de suelo, presenta dos botones que permiten:

1. Elimina la información

2. Cancela la eliminación del tipo de suelo

Formulario presentación topics sensores

Topics Sensores

REGISTRAR 1

Nro	Nombre Topic	Descripción Sensor	Acciones
1	suelo1	sensor suelo 1	2   3
2	suelo2	sensor suelo 2	 
3	suelo3	sensor suelo 3	 
4	caudal	sensor de agua	 

El siguiente formulario permite visualizar los topics de sensores, permite realizar un nuevo registro de un topic sensor, presenta tres botone que permiten:

1. Registrar un nuevo topic sensor
2. Permite cargar los datos del topic sensor en el formulario para editarlos
3. Presenta una alerta de confirmación para eliminar el topic sensor

Formulario editar topic sensor

Topics Sensores

Nombre de topic

Nombre de Sensor

1 ACTUALIZAR **CANCELAR** **2**

El siguiente formulario permite editar los datos del topic sensor, presenta dos botone que permiten:

1. Actualizar los datos del topic sensor
2. Cancela la actualización

Formulario confirmación de eliminación de topic sensor



A confirmation dialog box with a white background and a thin grey border. At the top center is an orange circle containing a white exclamation mark. Below the icon, the text reads "¿Está seguro que desea eliminar el topic del Sensor?". At the bottom, there are two buttons: a blue button labeled "Si" with a "1" in a circle to its left, and a grey button labeled "Cancel" with a "2" in a circle to its right.

El siguiente formulario permite confirmar si desea eliminar el topic sensor, presenta dos botones que permiten:

1. Elimina la información
2. Cancela la eliminación del topic sensor

Formulario presentación topics actuadores

Topics Actuadores

Nombre de topic			
Nombre de Sensor			
REGISTRAR			1
Nro	Nombre Topic	Descripción Actuador	Acciones
1	controlRiego	Actuador de riego	2   3

El siguiente formulario permite visualizar los topics actuador, permite realizar un nuevo registro de un topic actuador, presenta tres botone que permiten:

1. Registrar un nuevo topic actuador
2. Permite cargar los datos del topic actuador en el formulario para editarlos
3. Presenta una alerta de confirmación para eliminar el topic actuador

Formulario editar topic sensor

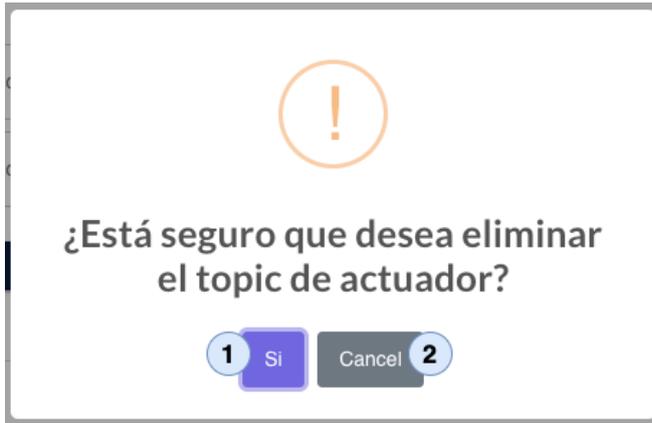
Topics Actuadores

Nombre de topic	controlRiego
Nombre de Sensor	Actuador de riego
1 ACTUALIZAR	CANCELAR 2

El siguiente formulario permite editar los datos del topic actuador, presenta dos botone que permiten:

1. Actualizar los datos del topic actuador
2. Cancela la actualización

Formulario confirmación de eliminación de topic sensor



El siguiente formulario permite confirmar si desea eliminar el topic actuador, presenta dos botones que permiten:

1. Elimina la información
2. Cancela la eliminación del topic actuador

Anexo 11 Certificado de traducción del resumen del trabajo de titulación

Loja, 15 de junio de 2023

Magister

KARINA CELI JARAMILLO

CATEDRÁTICA DE LA CARRERA DE PEDAGOGÍA DE LOS IDIOMAS
NACIONALES YEXTRANJEROS - UNL

CERTIFICADO:

Que el resumen del Trabajo de Integración Curricular de los aspirantes, Alex Santiago Nole Reyes con C.I. 1105642076 y Angel Favian Minga Medina con C.I. 1104695190 pertenecientes a la Carrera de Ingeniería en Sistemas; traducido al inglés cumple con las características propias del idioma extranjero.

Resumen:

En la actualidad Internet de las cosas (IoT) ha tenido un gran impacto en el desarrollo de soluciones tecnológicas permitiendo que varios componentes del mundo real se conecten a internet posibilitando interactuar con ellas. Existen grandes aplicaciones para la automatización de procesos en distintas áreas. Una de las aplicaciones en las que se puede implementar el IoT es la agricultura conocida como Smart Farming, que permite mejorar los procesos agrícolas, como el riego que es un proceso fundamental para los cultivos, ya que se puede monitorear las necesidades hídricas de las plantas en base a los datos enviados por los sensores y con esto llevar un control del agua utilizada. En este contexto el presente trabajo de titulación tiene como objetivo desarrollar un prototipo de sistema irrigación con IoT para cultivos de la Universidad Nacional de Loja. Este trabajo introduce el uso de sensores higrómetros que permiten capturar los cambios de humedad del suelo, para encender el riego cuando la humedad del suelo este en punto de marchitez permanente y se cierre el riego cuando la humedad este en Capacidad de campo. En tal sentido el marco metodológico utilizado consto de tres fases. I) Se realizó una RSL para obtener información relevante sobre la temática y responder las preguntas de investigación. II) Se desarrolló la solución informática haciendo uso de la metodología XP, aplicando estándares como el documento IEEE-830 para especificar los requisitos de software y el documento arquitectónico 4+1. III)

Se realizó la experimentación para conocer el funcionamiento de los sensores y los datos que envían estos en base a la humedad del suelo en un tiempo específico. Por último, el desarrollo del prototipo se ejecutó en un ambiente académico, partiendo con el desarrollo del RSL en donde se obtuvieron 25 artículos de los cuales la mayoría de autores utilizaron la arquitectura de tres capas para la automatización de sistemas de irrigación, empleando el protocolo de comunicación MQTT mediante la herramienta

mosquitto que funciona como servidor o bróker por donde pasan los paquetes enviados desde los publicadores hasta los suscriptores. En el diseño de la arquitectura se utilizó la arquitectura de tres capas que se adaptó al prototipo para el control del riego y finalmente se realizó el desarrollo del módulo de control utilizando la metodología ágil de software XP y armado del prototipo con la información recabada en las fases anteriores. De esta manera desarrolló un prototipo de sistema de irrigación con IoT, que permitió capturar la humedad del suelo mediante sensores higrómetros, para determinar cuándo es necesario abrir y cerrar el riego.

Palabras Claves: Internet de las Cosas, software, prototipo, riego, automatización, Smart farming

Abstract:

Currently, the Internet of Things (IoT) has had a significant impact on the development of technological solutions, enabling various real-world components to connect to the Internet and allowing interaction with them. There are extensive applications for process automation in various fields. One such application where IoT can be implemented is agriculture, known as Smart Farming, which improves agricultural processes, such as irrigation—a vital process for crops. With IoT, it becomes possible to monitor the water needs of plants based on data sent by sensors, enabling better control of water usage. In this context, the objective of this research work is to develop a prototype IoT irrigation system for crops at the Universidad Nacional de Loja. This work introduces the use of hygrometer sensors to capture soil moisture changes, triggering irrigation when the soil moisture reaches the permanent wilting point and shutting off irrigation when it reaches field capacity. The methodological framework used consisted of three phases. I) A systematic literature review (SLR) was conducted to gather relevant information on the topic and address the research questions. II) The software solution was developed using the XP methodology, applying standards such as the IEEE-830 document to specify software requirements and the 4+1 architectural document. III) An experimentation phase was carried out to understand the functioning of the sensors and the data they send based on soil moisture at specific times. Finally, the prototype development was executed in an academic environment, starting with the SLR phase, which yielded 25 articles, most of which utilized a three-layer architecture for irrigation system automation, employing the MQTT communication protocol through the mosquitto tool, which functions as a server or broker for packets sent from publishers to subscribers. The three-layer architecture was adapted to the prototype design for irrigation control, and

the control module was developed using the agile XP software methodology, followed by the assembly of the prototype using information gathered in the previous phases. In this way, a prototype IoT irrigation system was developed, capable of capturing soil moisture using hygrometer sensors to determine when it is necessary to open and close the irrigation system.

Keywords: Internet of Things, software, prototype, irrigation, automation, Smart Farming.

Lo certifico.



KARINA CELI JARAMILLO M.Ed.
CATEDRÁTICA DE LA CARRERA DE PEDAGOGÍA DE LOS IDIOMAS
NACIONALES Y EXTRANJEROS - UNL