



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA TELECOMUNICACIONES

**“ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO
DE GESTIÓN PARA EL MONITOREO Y CONTROL DE
CULTIVOS DE PAPA A TRAVÉS DEL INTERNET DE LAS
COSAS”**

Trabajo de titulación

Tipo: Dispositivo Tecnológico

Presentado para optar al grado académico de:

**INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y
REDES**

AUTOR: DAVID ALEJANDRO MUÑOZ VALLE

DIRECTOR: Ing. DIEGO FERNANDO VELOZ CHERREZ MSC.

Riobamba-Ecuador

2022

© 2022, David Alejandro Muñoz Valle

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, David Alejandro Muñoz Valle, declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación. El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 21 enero de 2022.



David Alejandro Muñoz Valle

180443471-8

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA TELECOMUNICACIONES

El Tribunal del Trabajo de Titulación certifica que: El trabajo de titulación: Tipo: Dispositivo Tecnológico, “**ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE GESTIÓN PARA EL MONITOREO Y CONTROL DE CULTIVOS DE PAPA A TRAVÉS DEL INTERNET DE LAS COSAS**”, realizado por el señor: **DAVID ALEJANDRO MUÑOZ VALLE**, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, el mismo que cumple con los requisitos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA
Ing. Washington Gilberto Luna Encalada. PRESIDENTE DEL TRIBUNAL	 Firmado electrónicamente por: WASHINGTON GILBERTO LUNA ENCALADA	2022-01-21
Ing. Diego Fernando Veloz Cherrez DIRECTOR DEL TRABAJO DE TITULACIÓN	 Firmado electrónicamente por: DIEGO FERNANDO VELOZ CHERREZ	2022-01-21
Ing. Fabricio Javier Santacruz Sulca. MIEMBRO DEL TRIBUNAL	 Firmado electrónicamente por: FABRICIO JAVIER SANTACRUZ SULCA	2022-01-21

DEDICATORIA

A mis padres, que siempre me han brindado ese apoyo incondicional en toda mi vida estudiantil y sin ellos nada de esto fuera posible. A mi hijo, que es mi motor de vida para seguir adelante en todo el camino, mi esposa que siempre está en los momentos más difíciles y mis hermanos que nunca dejan de ser ese apoyo incondicional para salir adelante. En general a toda mi familia que siempre confío en mí.

David

AGRADECIMIENTO

A Dios por cuidar siempre de mí, darme la oportunidad de seguir luchando por mis sueños y regalarme sabiduría y paciencia para llegar a cumplir esta meta. A toda mi familia que han sido mi pilar fundamental en todo este trayecto.

Agradezco a la Escuela Superior Politécnica de Chimborazo y a todos mis profesores, que a lo largo de este trayecto han sabido impartir todos sus conocimientos de la mejor manera, en especial a mi tutor el ingeniero Diego Veloz Ch. Msc. Por toda su ayuda, dedicación y paciencia en el desarrollo de este trabajo de titulación

A las personas que han sido lo más comprensibles posible para poder hacer uso de sus propiedades y terrenos en la etapa de pruebas, de manera especial al ingeniero Adriano Núñez por su disposición, conocimientos y brindarme las posibilidades para hacer uso de su espacio en cualquier momento.

David

TABLA DE CONTENIDO

ÍNDICE DE TABLAS	ix
ÍNDICE DE FIGURAS	x
ÍNDICE DE ANEXOS	xii
RESUMEN	xiii
SUMMARY	xiv
INTRODUCCIÓN.....	1
1. MARCO TEÓRICO REFERENCIAL.....	5
1.1. Generalidades	5
1.2. Agricultura de precisión	5
1.3. Estado del arte IoT	7
1.3.1. <i>Definición</i>	7
1.3.2. <i>Arquitectura</i>	7
1.3.2.1. <i>Arquitectura SOA</i>	8
1.3.2.2. <i>Arquitectura basada en API</i>	9
1.3.3. <i>Componentes Básicos</i>	10
1.3.4. <i>Protocolos de comunicación</i>	11
1.3.5. <i>Cloud Computing</i>	12
1.3.6. <i>Fog Computing</i>	13
1.3.7. <i>Edge Computing</i>	14
1.3.8. <i>Relación en la agricultura</i>	15
1.4. Tecnologías inalámbricas.....	16
1.4.1. <i>Bluetooth</i>	16
1.4.2. <i>Zigbee</i>	17
1.4.3. <i>Wifi</i>	17
1.4.4. <i>RFID</i>	18
1.4.5. <i>LoraWan</i>	19
1.5. Requerimientos para un cultivo de papa.....	19
1.5.1. <i>Humedad</i>	19
1.5.2. <i>Temperatura</i>	19
1.5.3. <i>pH del suelo</i>	19
1.5.4. <i>Sistemas de riego</i>	19
1.6. Tarjetas de desarrollo	21
1.6.1. <i>Arduino</i>	21

1.6.2.	<i>Raspberry Pi</i>	22
1.6.3.	<i>NodeMCU</i>	23
1.6.4.	<i>BeagleBone Black</i>	23
1.7.	Plataformas para IoT	24
1.7.1.	<i>Google Web Services</i>	24
1.7.2.	<i>Amazon Web Services</i>	24
1.7.3.	<i>Microsoft Cloud Services</i>	25
2.	PROPUESTA Y DISEÑO DEL PROTOTIPO	26
2.1.	Introducción	26
2.2.	Análisis de requerimientos de diseño	26
2.2.1.	<i>Arquitectura General</i>	27
2.3.	Selección de dispositivos para el prototipo	28
2.3.1.	<i>Sensores</i>	28
2.3.1.1.	<i>Sensor de Humedad</i>	29
2.3.1.2.	<i>Sensor de Temperatura</i>	29
2.3.1.3.	<i>Sensor de pH</i>	31
2.3.2.	<i>Tarjetas de desarrollo</i>	34
2.3.3.	<i>Tarjeta Nodemcu</i>	36
2.3.3.1.	<i>Requisitos para la utilización</i>	38
2.3.4.	<i>Actuador</i>	38
2.3.4.1.	<i>Electroválvula</i>	38
2.4.	Software	39
2.4.1.	<i>IDE de desarrollo</i>	39
2.4.2.	<i>Nodemcu ESP32 con el IDE de desarrollo</i>	40
2.4.3.	<i>Servidores Web</i>	42
2.4.3.1.	<i>Thingier</i>	43
2.4.3.2.	<i>Comunicación</i>	45
2.4.3.3.	<i>Servidor de mensajería</i>	45
2.5.	Interfaz de usuario	46
2.5.1.	<i>Android Studio</i>	48
2.6.	Otros dispositivos para implementación	49
2.6.1.	<i>Batería 9V</i>	49
2.6.2.	<i>Convertidor TTL</i>	50
2.6.3.	<i>Cajas de paso herméticas</i>	51
2.7.	Indicadores ambientales del cultivo	52
2.8.	Esquema general del prototipo	53

3.	MARCO DE VALIDACION Y RESULTADOS	54
3.1.	Conectividad entre servidores	54
3.1.1.	<i>Servidor Thinger y Firebase</i>	54
3.1.2.	<i>Firestore (FCM) y app en Android Studio</i>	55
3.2.	Pruebas de funcionamiento de sensores	58
3.3.	Pruebas de anclaje a la red mediante Esp32	59
3.4.	Pruebas de envío y recepción de datos	60
3.5.	Pruebas de notificaciones en la aplicación móvil	62
3.6.	Mediciones reales	63
3.6.1.	<i>Terreno con cultivo de papa sin control de parámetros</i>	63
3.6.2.	<i>Terreno con cultivo de papa a cargo de un profesional</i>	66
3.6.3.	<i>Regadío por inundación con uso de agua potable</i>	68
3.6.4.	<i>Evaluación por quince días en terreno con cuidados</i>	69
4.	PRESUPUESTO ECONÓMICO	78
4.1.	Introducción	78
4.2.	Presupuesto físico	78
4.3.	Presupuesto total	79
	CONCLUSIONES	80
	RECOMENDACIONES	82
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-1: Frecuencias y velocidades de trabajo ZigBee.....	17
Tabla 2-1: Protocolos comúnmente usados en Wifi.	18
Tabla 1-2: Características del computador.....	28
Tabla 2-2: Especificaciones módulo sensor de pH.....	31
Tabla 3-2: Escala de Likert para los dispositivos.....	31
Tabla 4-2: Evaluación factor de alimentación.	32
Tabla 5-2: Evaluación de factor rango de medición.	32
Tabla 6-2: Evaluación de factor de precisión.....	32
Tabla 7-2: Evaluación de factor de tiempo de respuesta.....	33
Tabla 8-2: Evaluación de factor tamaño del módulo.	33
Tabla 9-2: Tipos de tarjetas de desarrollo para el prototipo.....	34
Tabla 10-2: Evaluación por escala de Likert a tarjetas de desarrollo.....	35
Tabla 11-2: Características técnicas Esp32.....	37
Tabla 12-2: Requisitos en software.....	38
Tabla 13-2: Entornos de desarrollo.	39
Tabla 14-2: Evaluación por escala de Likert a entornos de desarrollo	40
Tabla 15-2: Plataformas con servicios en la nube.....	43
Tabla 16-2: Servidores de mensajería.	46
Tabla 17-2: Herramientas de diseño de aplicaciones Android.....	47
Tabla 18-2: Rangos teóricos adecuados para el cultivo de papa.....	53
Tabla 1-3: Mediciones en un terreno sin control de parámetros.....	64
Tabla 2-3: Mediciones en un terreno a cargo de un profesional.	67
Tabla 3-3: Mediciones en un terreno con regado con agua potable.....	69
Tabla 4-3: Mediciones parámetro de temperatura.	70
Tabla 5-3: Mediciones parámetro de humedad.....	72
Tabla 6-3: Mediciones parámetro de pH del suelo.	74
Tabla 1-4: Presupuesto en materiales utilizados para su diseño y construcción.....	78
Tabla 2-4: Presupuesto total del prototipo.	79

ÍNDICE DE FIGURAS

Figura 1-1: Proceso de agricultura de precisión.....	6
Figura 2-1: Modelo de arquitectura para IoT.....	8
Figura 3-1: Capas en una arquitectura orientada API.....	9
Figura 4-1: Estructura y servicios de cloud computing.....	12
Figura 5-1: Trabajo de fog computing.....	13
Figura 6-1: Capas de trabajo edge computing.....	14
Figura 7-1: Comunicación Bluetooth.....	16
Figura 8-1: Riego por surcos.....	20
Figura 9-1: Riego por goteo.....	20
Figura 10-1: Riego por aspersión.....	21
Figura 11-1: Tipos de tarjeta de desarrollo Arduino.....	22
Figura 12-1: Modelos de Raspberry pi.....	22
Figura 13-1: NodeMCU ESP8266 v9.....	23
Figura 14-1: Tarjeta de desarrollo BeagleBone Black.....	24
Figura 1-2: Bloques de arquitectura general.....	30
Figura 2-2: Modulo HL-69 sensor de humedad del suelo.....	30
Figura 3-2: Sensor de temperatura.....	30
Figura 4-2: Modulo SEN0161 de pH del suelo.....	34
Figura 5-2: Modulo ESP32.....	36
Figura 6-2: Arquitectura de Nodemcu ESP32.....	37
Figura 7-2: Componentes de una electroválvula.....	38
Figura 8-2: Gestor de tarjetas en IDE de Arduino.....	41
Figura 9-2: Modulo a ser utilizado.....	41
Figura 10-2: Servicios gratuitos en Thinger.io.....	44
Figura 11-2: Interfaz de inicio en Thinger.....	45
Figura 12-2: Interfaz de usuario en un dispositivo móvil.....	48
Figura 13-2: Interfaz de usuario.....	49
Figura 14-2: Batería de 9V.....	50
Figura 15-2: Convertidor TTL a 3,3V.....	50
Figura 16-2: Caja hermética para circuitos.....	51
Figura 17-2: Caja hermética principal.....	52
Figura 18-2: Caja hermética secundaria y sensores.....	52
Figura 19-2: Esquema de trabajo.....	53
Figura 1-3: URL de conectividad hacia Firebase.....	56

Figura 2-3: Certificado SHA para conectar a la aplicación	56
Figura 3-3: Clave SHA-1 proporcionado por la app en Android Studio	56
Figura 4-3: Verificación de clave SHA-1 en Android Studio.....	56
Figura 5-3: Indicadores para prueba de notificación a firebase.	56
Figura 6-3: Verificación de notificaciones al teléfono móvil.	57
Figura 7-3: Verificación de alarmas a la aplicación.	57
Figura 8-3: Valores rango de mapeo del sensor de humedad.	58
Figura 9-3: Valores rango de mapeo del sensor de pH.....	58
Figura 10-3: Ventana de visualización de parámetros.....	59
Figura 11-3: Código para poder anclarse a una red.	59
Figura 12-3: Conexión de ESP32 a la red del dispositivo móvil.	60
Figura 13-3: Concordancia Parámetros de la red del dispositivo móvil.	60
Figura 14-3: Estado del dispositivo en el servidor.....	61
Figura 15-3: Pestañas de parámetros en transmisión y recepción de datos.	61
Figura 16-3: Transmisión en vivo de datos.....	62
Figura 17-3: Pestaña de ubicación del dispositivo.....	62
Figura 18-3: Verificación de alarma bajo del rango adecuado.	63
Figura 19-3: Monitorización de parámetros en cultivo de papa sin cuidados.	64
Figura 20-3: Prototipo en una plantación de papa sin monitorización de parámetros.	65
Figura 21-3: Plantación de papa con daños y sin nacimiento de flor.	65
Figura 22-3: Monitorización en un terreno con cuidados.	66
Figura 23-3: Prototipo en un cultivo de papa con cuidados.....	67
Figura 24-3: Medición de parámetros en riego con agua potable.....	68
Figura 25-3: Riego con agua potable usando el prototipo.	69
Figura 26-3: Florecimiento en espacio monitorizado.	77
Figura 27-3: Mediciones en el terreno utilizado.	77

ÍNDICE DE ANEXOS

ANEXO A: DATASHEET ESP32 WROOM.

ANEXO B: ESQUEMA DE CONEXIÓN CONVERTOR TTL 5V/3,3V.

ANEXO C: CÓDIGO DE PROCESAMIENTO DE DATOS EN ARDUINO IDE.

ANEXO D: ESQUEMA DE CONEXIÓN REGULADOR 7805.

ANEXO E: XML ACTIVITY_MENU.XML (MENÚ PRINCIPAL).

ANEXO F: CLASE JAVA MIFIREBASEINSTANCESERVICE.

RESUMEN

El objetivo del presente trabajo de titulación fue analizar, diseñar e implementar un prototipo de gestión que permita monitorizar y controlar un terreno con cultivo de papas a través del Internet de las cosas. Mediante la revisión bibliográfica se determinó todos los requerimientos de diseño e implementación aplicados a este cultivo en cielo abierto, así como también la composición y características de funcionamiento de la tecnología IoT (Internet Of Things). El método de análisis fue llevado a cabo mediante metodología experimental, realizando el diseño a través de desarrolladores ampliamente conocidos como Arduino IDE, Android Studio y la implementación con la tarjeta de desarrollo empleada para este tipo de prototipos Nodemcu ESP32 que incorpora un módulo Wifi para la conexión a la red, además se utilizó sensores de tipo climáticos como temperatura, humedad y pH del suelo conectados mediante cable de red (Ethernet) hacia la tarjeta de procesamiento. La comunicación en la nube fue realizada mediante el servidor HTTP Thingier y su comunicación hacia Firebase que permite la comunicación directa con la aplicación del usuario final teniendo una compatibilidad entre todos estos. Las pruebas de funcionamiento se experimentaron en distintos terrenos con plantación de papa, para poder validar sus diferentes resultados y observar si existen o no cambios en el cultivo. La evaluación constante realizada por el prototipo evidencia resultados muy llamativos en el cual se analiza parámetro por parámetro demostrando así la eficiencia de este, ayudando a que la plantación se encuentre dentro del rango de parámetros teórico y prácticos recomendados por distintas fuentes bibliográficas. Se recomienda realizar todo el análisis de monitoreo y control dentro de un invernadero ya que va a permitir garantiza un mejor control sin intervención de fenómenos naturales.

Palabras clave: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA> <INTERNET DE LAS COSAS (IOT)> <ARDUINO IDE> <ANDROID STUDIO> <TARJETA DE DESARROLLO NODEMCU ESP32>.



Firmado digitalmente por:
HOLGER GERMAN
RAMOS UVIDIA

0065-DBRA-UPT-2022

2022-01-17

SUMMARY

The objective of this degree work was to analyze, design and implement a prototype of management that allows monitoring and controlling a land with potato cultivation through the Internet of things. Through the bibliographic review, all the design requirements were determined and implementation applied to this crop in the open sky, as well as the composition and operating characteristics of IoT (Internet Of Things) technology. The analysis method was carried out through experimental methodology, carrying out the design through well-known developers such as Arduino IDE, Android Studio, and the implementation with the development card used for this type of Nodemcu ESP32 prototypes that incorporates a Wi-Fi module for connection to the network, in addition, climate-type sensors were used as soil temperature, humidity and pH connected via network cable (Ethernet) to the card processing. Communication in the cloud was done through the Thingier HTTP server and its communication to Firebase that allows direct communication with the application of the end user having a compatibility between all these. The performance tests are experimented on different lands with potato plantations, in order to validate their different results and observe whether or not there are changes in the crop. The constant evaluation carried out by the prototype shows very striking results in which parameter by parameter is analyzed thus demonstrating its efficiency, helping the plantation to be within the range of theoretical and practical parameters recommended by different bibliographical sources. It is recommended to carry out all the monitoring and control analysis inside a greenhouse since it is going to allow to guarantee a better control without the intervention of natural phenomena.

Keywords: <TECHNOLOGY AND ENGINEERING SCIENCES> <INTERNET OF THE THINGS (IOT)> <ARDUINO IDE> <ANDROID STUDIO> <DEVELOPMENT BOARD NODEMCU ESP32>.



Firmado electrónicamente por:
**WILSON GONZALO
ROJAS YUMISACA**

MSc. Wilson G. Rojas
NOMBRE Y FIRMA PROFESOR

INTRODUCCIÓN

La papa es un cultivo estratégico debido a su alta rentabilidad y su potencial en rendimiento. La papa es uno de los cuatro cultivos más importante del mundo, con una producción anual cercana a los 300 millones de toneladas, además es una importante fuente de carbohidratos (almidón), proteínas de alta calidad, vitamina C y minerales (Balladares, 2018, pp. 03-23). Conforme al paso de los años, el humano a detectado varias dificultades dentro de sus cultivos sean climatológicos, económicos o políticos que soporta cada región, haciendo que se genere un sin número de pérdidas.

Desde hace varios años, se ha tratado de encontrar una solución por medio de estudios e investigación de herramientas agroindustriales que vayan de la mano con la tecnología, buscando la manera de optimizar el recurso hídrico y su potencialización agrícola, así pues la agricultura ha venido desarrollándose cada vez más, adaptando a diferentes actores como la Telemática, y conjunto de procesos tecnológicos que lleva al campo de agricultura de precisión, uno de los actores en estos proyectos es INTA (Instituto Nacional de Tecnología Agropecuaria) que desde los años 90 vienen trabajando en proyectos de este tipo. Lo que permite tener como objetivo la generación de conocimientos agrícolas que proporcione el control o manejo de factores que influyen en la producción como insumos, información de tipo espacial, temporal y un factor de importancia que es el cambio en los diferentes ambientes, así poder optimizar su manejo (López y Quiñonez, 2018).

Un estudio tecnológico, desarrollado para el departamento de Sucre (Colombia) en el 2020, por la Corporación Universitaria del Caribe, muestra un control de riegos en cultivos de ‘ahuyama’ que es gestionado así mismo, por el internet de las cosas, consiste en un sistema diseñado que se compone de un controlador PID que consta de parámetros proporcional, integral y derivativo, y un prototipo que permite adquirir datos y enviar a la nube a través de herramientas utilizadas en Internet de las cosas. Realizado debido a que en este sector la agricultura es uno de sus principales ingresos económicos, además por los fuertes efectos del cambio climático, a la escasa pluviosidad y sobre todo por los precarios sistemas tecnológicos utilizados para la administración del agua.

El Ecuador es un país fundamentalmente agrícola que goza de un clima que no varía de forma abrupta, pero en la agricultura el clima depende mucho de su ubicación geográfica. Un sistema de información geográfica fue desplegado en la Universidad San Francisco de Quito el cual ayuda en la gestión global de cultivos de palma aceitera por medio de la aplicación de la agricultura de precisión, manejando una base de datos geográfica que permite contener las variables que tienen una correspondencia de esta plantación, incorporando datos específicos que ayuden al sistema de agricultura de precisión diseñado e implementado.

En el país, se goza de un clima estable todo el año, pese a sus diferencias regionales. Además de suelos aptos para cultivos y con una gran capacidad para trabajar la tierra, pese a esto, no se ha

logrado sacar una ventaja provechosa para que esta actividad sea considerada como generadora de miles puestos de trabajo y económicamente rentable (López y Quiñonez, 2018). En la región sierra, uno de los cultivos con mayor producción ha sido la papa, debido a la relevancia del cultivo de papa en el país, específicamente en zonas altas, se requiere de suma importancia un sistema que permita facilitar el monitoreo de parámetros generales para que este cultivo se lo realice de manera correcta, aplicando la agricultura de precisión y estos puedan ser revisados en cualquier parte del mundo por medio del internet.

FORMULACIÓN DEL PROBLEMA

¿La implementación de sistemas IoT en cultivos de papa permitirán un control más eficiente de los parámetros que determinan la calidad del producto?

SISTEMATIZACIÓN DEL PROBLEMA

¿Cuál es el estado actual de los sistemas IoT para la agricultura?

¿Qué tecnologías inalámbricas pueden ser utilizadas para controlar parámetros climáticos en cultivos de papa?

¿Qué tipo de variables o parámetros se pueden controlar para tener mayor calidad en estos cultivos?

¿Cuáles son los requerimientos para implementar un sistema IoT para el control y monitoreo en el cultivo de papa?

¿Cuáles son los parámetros que definen la eficiencia del prototipo IoT implementado?

JUSTIFICACIÓN TEÓRICA

Uno de los principales cultivos en el Ecuador es la papa, en la producción se vincula a 82 mil productores en un total de 90 cantones, radicándose alrededor del 70% de producción en provincias como Cotopaxi, Carchi, Chimborazo y Tungurahua. Con el paso de los años el volumen de producción de papa se ha mantenido, debido al incremento de rendimientos a pesar de que la superficie cosechada cayó, factores climáticos y no climáticos que posiblemente tienen incidencia sobre el cultivo de papa en las diferentes zonas productivas, han sido causantes de pérdidas económicas y productos de mala calidad.

Los pequeños productores (menos de 2Has en terreno) son el 50% de estos cultivos y poseen el 19% del área sembrada, la consolidación de estos pequeños productores con la ayuda de INIAP (Instituto Nacional de Investigaciones Agropecuarias) crearon un consorcio que lleva el nombre de CONPAPA, esta intervención permitió la investigación tecnológica para nuevas oportunidades

de mercado, la capacitación en el manejo del cultivo de papa a través de escuelas de campo de agricultores, pero también se tiene varios problemas como falta de semilla certificada para la producción que afecta los rendimientos y la producción, la incidencia de plagas y enfermedades en el cultivo, la falta de recursos para la producción, la desorganización, la falta de conocimientos técnicos para el cultivo, el desperdicio del recurso hídrico, falta de implementación de agricultura de precisión, etc.

En el sector agrícola al poseer un bajo grado de tecnificación y conocimientos tecnológicos, hace que los procesos de control de factores o recursos se los haga de manera manual influyendo directamente en el crecimiento y desarrollo del cultivo, principalmente el abastecimiento del recurso hídrico a cada una de las plantas, estos procesos son llevados a cabo de una manera individual a la deducción de la persona que esté a cargo de la zona en producción, sin poseer algún dispositivo o equipo de medición que permita observar los parámetros que hacen que su producción sea de buena o mala calidad, todo el cuidado que necesite se lo hace de manera empírica de forma que provee a esta zona lo que cree que haga falta, sin dejar un lado que tampoco lleva un registro detallado que le permita tener un mayor control de sustancias, cambios ambientales, o alteraciones en los suelos, para prevenir en etapas futuras la cosecha de cada uno de los cultivos.

JUSTIFICACIÓN APLICATIVA

En la Escuela Superior Politécnica de Chimborazo se presentó un prototipo que ayuda al control de cultivos en cielo abierto a través de una red inalámbrica con nodos inteligentes, integrando varias tarjetas de desarrollo, sensores climáticos, redes WSN, actuadores, etc. presentando conclusiones que beneficia a cualquier tipo de cultivo en cielo abierto, proporcionando los diferentes tipos de suelos aptos conforme a su pH, este prototipo evidencia errores en medición de temperatura de ± 1 °C. demostrando que el prototipo no agrega errores que pueden tener los sensores, siendo eficiente, además menciona que con el sistema desarrollado ASC se puede evidenciar reportes gráficos y una revisión auténtica del desarrollo de parámetros físico químicos en algún tipo de terreno, además permite ser descargado para su mejor análisis y tratamiento de datos.

De esta forma, en el presente trabajo se propone un sistema de gestión que permita al usuario el control y monitoreo de parámetros específicos que influyen en el cultivo de papa a través de la tecnología IoT, permitiendo visualizar datos de tipo climáticos, pH del suelo, humedad, etc. en cualquier lugar que se encuentre con tan solo tener acceso al internet, ayudando además a que se optimice el recurso del agua y su producción sea precisa obteniendo una mejor calidad del producto.

OBJETIVOS

OBJETIVO GENERAL

Analizar diseñar e implementar un prototipo de gestión para el monitoreo y control de cultivos de papa a través del internet de las cosas.

OBJETIVOS ESPECÍFICOS

- Analizar el estado del arte de la tecnología IoT para la agricultura a través de revisión bibliográfica disponible, para adquirir conocimientos necesarios que permitan desarrollar el sistema propuesto.
- Diseñar un prototipo IoT para el control y monitoreo de parámetros en cultivos de papa.
- Implementar el prototipo, para el control y monitoreo de parámetros en cultivos de papa, según los requerimientos de diseño.
- Demostrar el funcionamiento del prototipo propuesto, en base a los parámetros, para el control del cultivo de papa.

1. MARCO TEÓRICO REFERENCIAL

En el presente capítulo se detallarán los conceptos teóricos básicos para la realización de este proyecto de investigación, tales como la tecnología IoT, componentes de una red IoT, agricultura de precisión, topologías de trabajo, así como temas necesarios para la implementación como tarjetas de desarrollo, softwares aplicativos, infraestructura, etc. basados en modelos ya experimentados y bibliografía necesaria.

1.1. Generalidades

El escenario más nuevo que se tiene en el entorno con la seca de ríos y tanques, problemas del medio ambiente, calentamiento global, o el entorno impredecible presenta una necesidad urgente de una adecuada utilización del agua.

Para hacer frente a este uso desmedido, se puede ayudar de sensores de temperatura y humedad en condiciones adecuadas que se pueden implementar en ubicaciones precisas para el seguimiento de cultivos. Un algoritmo desarrollado con valores umbrales de temperatura y humedad del suelo puede ser programado en una puerta de enlace basada en un microcontrolador para controlar la cantidad de agua.

El desarrollo tecnológico de las Redes de Sensores Inalámbricos ha hecho posible su uso en la monitorización y control de los parámetros de algún cultivo específico en la agricultura de precisión.

Después de la investigación en el campo de la agricultura, los investigadores descubrieron que el rendimiento de la agricultura está disminuyendo día a día. Sin embargo, el uso de la tecnología en el campo de la agricultura juega un papel importante en el aumento de la producción, así como en la reducción de los esfuerzos adicionales de la mano de obra. Algunos de los intentos de investigación se hacen para mejorar a los agricultores que proporciona los sistemas que utilizan tecnologías útiles para aumentar el rendimiento agrícola (Gondchawar y kawitkar, 2016, p. 838).

1.2. Agricultura de precisión

La agricultura ha sido considerada la base de fuente alimenticia debido a su cantidad de vegetales, tubérculos y granos. El crecimiento del sector agrícola es necesario para el desarrollo de la condición económica del país. Por desgracia, muchos agricultores siguen utilizando los métodos tradicionales de cultivo que dan lugar a un bajo rendimiento de las cosechas y obtención de frutos. Pero en la mayoría de los casos donde se ha implantado la automatización y los seres humanos han sido sustituidos por máquinas automáticas, el rendimiento ha mejorado.

Gracias a los avances en el campo de las redes de sensores inalámbricos, así como a la miniaturización de las placas de sensores, ha empezado a surgir la agricultura de precisión. La agricultura de precisión se centra en proporcionar los medios para observar, evaluar y controlar las prácticas agrícolas. La agricultura de precisión es considerada como la aplicación y utilización de tecnologías en terrenos de pequeñas superficies, teniendo en cuenta los factores de dependencia como el suelo, el ámbito ambiental y las necesidades de cada planta, para lograr un fin en común que es la optimización de recursos como riego, aplicación de fertilizantes, etc. y así poder obtener una producción rentable, amigable con el medio ambiente y de mejor calidad (Ezcaray, 2012, pp. 08-09).

Una faceta de la agricultura de precisión se centra en la gestión de los cultivos, esto abarca diferentes aspectos, como la supervisión del suelo, los cultivos y el clima en un campo; la generalización de los resultados a un campo específico, para ello se utilizan experiencias de muchos ámbitos y se integran con las últimas herramientas de TIC (tecnologías de Información y Comunicación).

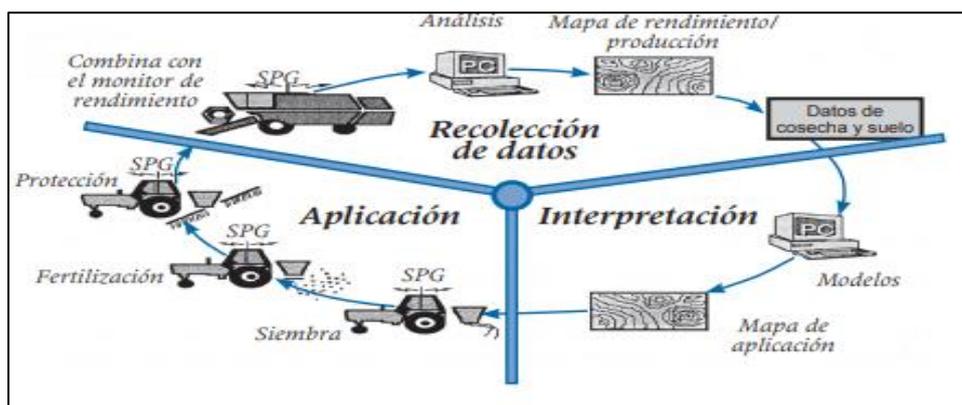


Figura 1-1: Proceso de agricultura de precisión

Fuente: Chartuni, 2007.

El desarrollo y el ataque asociado al cultivo dependen en gran medida de las condiciones climatológicas del campo. La humedad es un factor importante factor importante en el desarrollo de enfermedades o mal formaciones a los productos. De esta manera, se definen prácticas agrícolas de mayor precisión para ser reemplazadas por las recomendaciones habituales, conocida como agricultura tradicional, obteniendo un trabajo con manejo de parámetros y variables que permitirán un mejor rendimiento en toda el área escogida para su desempeño.

1.3. Estado del arte IoT

1.3.1. Definición

El paradigma de IoT se basa en el despliegue de miles de millones de objetos con conexión a internet, que tienen la capacidad de transmitir información sobre sus contextos y entorno para crear en tiempo real, una interacción eficiente entre lo real o cotidiano y el mundo virtual. IoT reveló ser una tecnología clave para resolver problemas sociales temas como ciudades digitales, transporte inteligente, vigilancia del medio ambiente y campos agrícolas o atención médica y seguimiento de personas mayores.

El grupo RFID define la IoT como: “La red mundial de objetos interconectados direccionables basado exclusivamente en estándares de protocolos de comunicación” (Jurado et al., 2014, p. 05)

En su desarrollo, se utilizó el Internet para protocolos orientados a la conexión de aplicaciones como es el caso de HTTP (Protocolo de transferencia de hipertexto) y SMTP (Simple Mail Transfer Protocol). En la actualidad varios de los dispositivos u objetos se pueden comunicar entre sí, dependiendo de las necesidades de su entorno, esta definición es conocida como M2M (Machine to Machine) o lo que es lo mismo, comunicación máquina a máquina. (Salazar y silvestre, p. 07)

IoT tiene fuertes conexiones con máquina a máquina (M2M), y a veces en la literatura, ambos se refieren a la misma idea, pero se puede decir que IoT cubre un alcance más amplio que incluye varias tecnologías de comunicación y a las aplicaciones a las que se puede enfrentar, por otro lado, M2M está enfocado a tecnologías que permiten a las máquinas u objetos comunicarse directamente. De manera global, se puede afirmar que el Internet de las cosas (IoT) facilitará a los usuarios de la red de Internet, comunicarse a cualquier hora y desde cualquier lugar del mundo a través de un conjunto de tecnologías de comunicación e información con el objetivo de proporcionar y utilizar los diferentes servicios para el control o monitoreo de agentes en la red en tiempo real o pospuesto de manera automática (Jurado et al., 2014, p. 06).

1.3.2. Arquitectura

En el campo de arquitectura IoT, no se tiene una estandarización debido a que la tendencia ha estado inclinada para solucionar dos problemáticas esenciales en esta tecnología: buscar una forma de acceso al medio y a los dispositivos, y la de buscar la integración de los dispositivos hacia el internet. Tomando como base la comunicación entre los dispositivos sobre la Internet: La capa de servicios Web.

Se presenta un patrón arquitectónico que puede ser aplicado por el diseño e implementación de aplicaciones y sistemas que transmitan eventos entre componentes de software que estén

emparejados libremente y servicios. Un sistema que conlleva eventos dispondrá de un emisor de eventos (agente) y un consumidor de estos (receptor) donde los consumidores tienen la responsabilidad de llevar la redacción del evento en suceso tan pronto sea posible. Por ejemplo, el receptor es dedicado solamente en filtrar, transformar y reenviar el evento a otro componente participe o proporcionar la acción propia a algún evento. En base a esta información, se muestra un modelo general de una arquitectura para IoT.

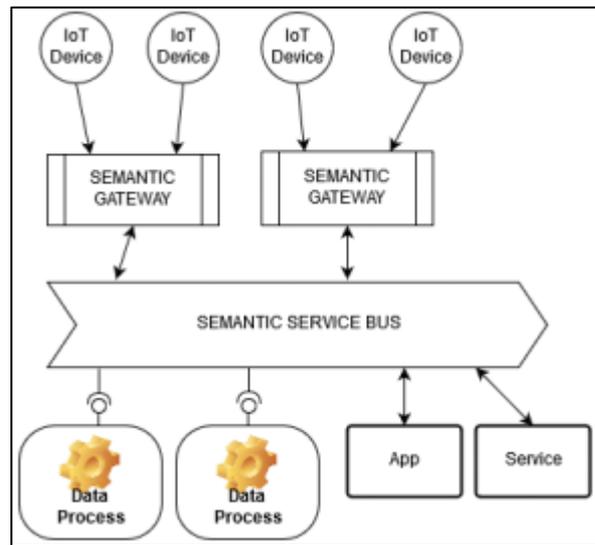


Figura 2-1: Modelo de arquitectura para IoT

Fuente: Barrera, 2018

Los servicios Web son basados en estándares como SOA y en arquitecturas REST para intercambiar datos entre aplicaciones de software desarrollados en distintos formatos.

1.3.2.1. *Arquitectura SOA*

La llegada de las tecnologías SOA están permitiendo un eficiente despliegue de soluciones que dan cobertura a la Interoperabilidad técnica y funcional, de hecho, la barrera es cada vez menos técnica, sino organizativa y procedural (González, 2007, p. 94).

La decisión de brindar un número cada vez mayor, de servicios a través del Internet se lo ha visto como precipitado, hoy en día se vincula constantemente a SOA (Arquitectura Basada en Servicios) en una definición netamente tecnológica. Dentro de esta arquitectura los servicios se construyen de manera que son captados por los responsables del negocio como responsables de los sistemas de información, por ello, la mayor dificultad para adoptar una arquitectura SOA radica en la orientación a servicios, es decir que se pueda tener una base como principio de diseño

donde pueda construir una capa de servicios intermedia entre negocio (capa de procesos) y tecnología (capa de aplicaciones) garantizando el alineamiento entre ambos. (González, 2007, p. 97) Gracias a la ayuda de esta arquitectura, se puede tener una capa de servicios de interoperabilidad entre infraestructuras de tipo heterogéneas (distintos). Además, su diseño está basado en estándares y agentes de aplicación para aprovechamiento informático basado en semántica con el objetivo de interpretar los datos de cada sensor de manera independiente e interactuar de forma mutua.



Figura 3-1: Capas en una arquitectura orientada API

Fuente: CA technologies, 2015.

1.3.2.2. Arquitectura basada en API

Se trata de una arquitectura que posee varias reglas para poder ser aplicada como: poseer una interfaz uniforme, realizar peticiones sin estado, separación de cliente y servidor, reducir el sistema de capas y complejidad que imponen otras técnicas, y código bajo de manda (opcional) (GitBook, 2015).

Como se observar en la figura 3-1, la arquitectura cuenta con cuatro capas específicas que engloban a más sistemas, reduciendo el exceso de cabeceras y ayudando que el consumo de potencia sea más eficiente.

Capa de seguridad: Las API se han visto vulnerables a varias amenazas que se encuentran fácilmente en la web, que hace necesario la aplicación de una seguridad sólida. Que se implemente seguridad en una arquitectura de API centralizada (en lugar de en la implementación de API) ayudará a evadir el problema anterior, como también facilitará el uso de tecnologías de administración de acceso flexible como OAuth y OpenID Connect (CA Technologies, 2015, pp. 17-18).

Capa de almacenamiento en caché: gracias a esta capa se puede maximizar la eficiencia de API. Esta capa debe permitir la entrega de respuestas en caché cuando son solicitudes comunes, lo que ayuda a la reducción de la presión colocada en las implementaciones de API y los recursos de administración reales (CA Technologies, 2015, pp. 17-18).

Capa de representación: la presentación de cualquier API deberá ser lo más sencilla posible para desarrolladores, haciendo que sea más fácil la presentación de sistemas administrativos complejos como interfaces ubicadas en dispositivos móviles y web que aprovecharan los desarrolladores para realizar aplicaciones mucho más sencillas y eficientes (CA Technologies, 2015, pp. 17-18).

Capa de organización: esta capa tiene la facultad de brindar combinaciones de diversas API, como también sintetizar el proceso de desarrollar nuevas implementaciones de diversos recursos administrativos (CA Technologies, 2015, pp. 17-18).

1.3.3. *Componentes Básicos*

- **Sensores**

Son dispositivos que tienen como función principal adquirir la información de distintos parámetros en forma de medida, como el pH del suelo, humedad, temperatura, gases, humo, luz, etc. Luego de obtener la información es enviada al sistema de control. Existen varios sensores con los cuales se puede trabajar en diferentes sistemas que se vayan a implementar como, por ejemplo: medidores de pH, medidor de presión del agua, detectores de humo, contacto magnético, detectores de movimiento y más (Rodas, 2013, p. 14).

- **Actuadores**

Son los encargados en cambiar el estado de equipos o factores con los que se está trabajando, luego de haber recogido la información. Los actuadores que más se utilizan en aplicaciones son: motores, electroválvulas de gas y agua, sirenas, relés y algunos elementos zumbadores (Rodas, 2013, p. 16).

- **Unidad de control**

Como su nombre lo indica, es la encargada en llevar el control del sistema, recibiendo las señales que son entregadas por los sensores y poder procesar a través del software con el que se desempeña, el mismo que debe definir las salidas que se van a activar o desactivar y la manera de trabajo de cada comando, además de comunicarse con los actuadores para que realicen su trabajo específico (Rodas, 2013, p. 17).

- **Estación base**

Encargada en la recepción de los datos de cada uno de los sensores, selecciona y procesa, y si es el caso, comunicarse con los demás nodos de la red (Rodas, 2013, p. 17).

1.3.4. *Protocolos de comunicación*

Debido a las limitaciones de energía y al estricto cálculo involucrados por los dispositivos de IoT hay muchos protocolos ligeros en la capa de aplicación como CoAP, MQTT, AMQP, y HTTP. Estos protocolos se pueden aumentar o disminuir según los requisitos del sistema.

Se ejecuta el protocolo CoAP en UDP y trabaja sobre el principio de solicitud o respuesta arquitectura. El protocolo AMQP se ejecuta sobre el protocolo TCP siguiendo la arquitectura de publicación / suscripción de forma asíncrona y utilice TSL / SSL para garantizar la seguridad. MQTT es un Protocolo de ancho de banda eficiente que utiliza poca energía de la batería y diseñado para recibir y transmitir información de sensores. HTTP es un protocolo de mensajería web muy conocido que se basa en la arquitectura de solicitud / respuesta. Carreras sobre TCP y no define ninguna QoS, utiliza TSL / SSL para propósito de seguridad (Farooq et al., 2019)

CoAP: CoAP proporciona un modelo de interacción solicitud / respuesta entre puntos finales de la aplicación, admite el descubrimiento integrado de servicios y recursos e incluye conceptos clave de la Web como URI y Tipos de medios de Internet. CoAP está diseñado para interactuar fácilmente con HTTP para la integración con la Web cumpliendo requisitos especializados como soporte de multidifusión, muy poca sobrecarga y simplicidad para entornos restringidos (Rfc 7252, 2014, p. 01).

MQTT: MQTT es un protocolo de comunicación bidireccional ligero emergente que está destinado a ser utilizado en un contexto de IoT para permitir la comunicación entre un dispositivo y un servidor, llamado broker. El protocolo MQTT utiliza TCP para garantizar la confiabilidad de la transmisión de paquetes y requiere una conexión entre el servidor y el cliente. Es un protocolo de publicación / suscripción, como se describe en la especificación resumen. Dado que es responsabilidad del corredor realizar un seguimiento de los clientes y de qué temas están suscritos, reduce la complejidad del lado del cliente. Se afirma que los protocolos de publicación / suscripción como MQTT se están volviendo importantes, especialmente para los dispositivos sensores en IoT, ya que los mensajes deben enviarse de la manera más eficiente posible sin consumir demasiada energía y necesita pocas capacidades de procesamiento (Deschambault, 2017, p. 27-28).

HTTP: El protocolo de transferencia de hipertexto (HTTP) es uno de los los protocolos de aplicación más utilizados en el Internet. Desde su publicación, RFC 2616 (HTTP 1.1) ha servido como base para el sin precedentes crecimiento de Internet: miles de millones de dispositivos de todas las formas y tamaños, desde computadoras de escritorio hasta la pequeña Web en dispositivos en nuestros bolsillos, habla HTTP todos los días para entregar noticias, videos y millones de otras aplicaciones web todos hemos llegado a depender en nuestra vida diaria. Lo que comenzó como un simple protocolo de una línea para recuperar hipertexto (es decir, "OBTENER / documento") evolucionó rápidamente a un transporte hipermedia genérico. Ahora, se utiliza para

alimentar casi cualquier caso de uso imaginable. Sin embargo, bajo el peso de su propio éxito, y a medida que más y más interacciones diarias continúan migrar a la Web: redes sociales, correo electrónico, noticias y videos, y, cada vez más, nuestros espacios de trabajo personales y laborales. HTTP 2.0 entra en escena mejorando aspectos de su antecesor, HTTP 2.0 hará aplicaciones más rápido, más simple y robusto por permitiendo una multiplexación eficiente y entrega de baja latencia a través de una sola conexión y que permite a los desarrolladores web deshacer muchas de las aplicaciones (Grigorik, 2013, p. 42-49).

1.3.5. Cloud Computing

Cloud computing hace referencia a aplicaciones y servicios que se ejecutan en una red distribuida utilizando recursos virtualizados y a los que se accede mediante protocolos de Internet y estándares de redes conocidos o comunes. Se distingue por la noción de que los recursos son virtuales e ilimitados y que los detalles de los sistemas físicos en los que se ejecuta el software se abstraen del usuario (Sosinsky, 2011, p. 04).

En un esfuerzo por describir mejor la computación en la nube, varios tipos de nubes han sido definidos. Las principales son las basadas en implementación y basadas en servicio, el modelo de implementación le indica dónde se encuentra la nube y para que propósito. Las nubes públicas, privadas, comunitarias e híbridas son modelos de implementación.

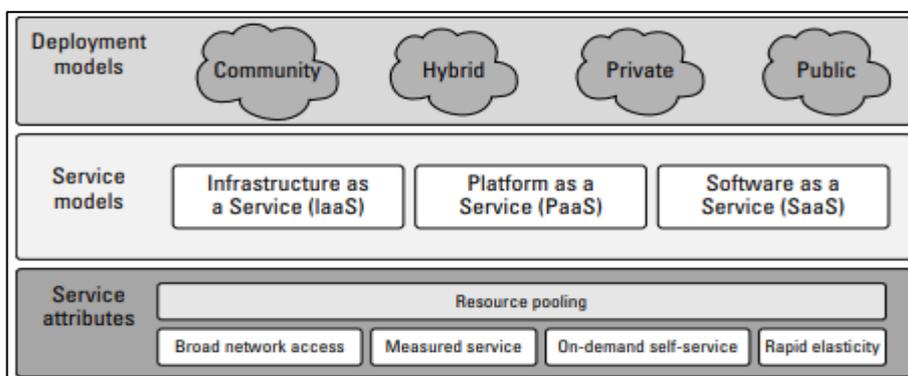


Figura 4-1: Estructura y servicios de cloud computing

Fuente: Sosinsky, 2018

Los modelos de servicio describen el tipo de servicio que ofrece el proveedor. Los modelos de servicio más conocidos son: el software como servicio (SaaS), la plataforma como Servicio (PaaS) y la Infraestructura como Servicio (IaaS). Los modelos de servicio se basan unos en otros y definen lo que el proveedor debe gestionar y lo que cliente tiene como responsabilidad. Se tiene varios ejemplos en cada una de estas: SaaS predifine una infraestructura, plataforma y software de aplicación, algunos son Gmail, Google Docs, Dropbox, en la actualidad Microsoft Azure y

Google Cloud. Para el caso de PaaS predefine una infraestructura y plataforma a través del cual puede implementar aplicaciones dependiendo de herramientas proporcionadas por el proveedor de servicio, como bases de datos, lenguajes de programación, entre otros. IaaS provee una virtualización de recursos físicos para que el usuario defina la infraestructura de acuerdo con sus necesidades aplicativos, los servicios más comunes son Amazon AWS Y Rackspace (Campoverde et al., 2015, p. 173).

La computación en nube representa un verdadero cambio de paradigma en la forma de desplegar los sistemas. La escala masiva de los sistemas de computación en nube fue de los sistemas de computación en nube fue posible gracias a la popularización de Internet y al crecimiento de algunas empresas de servicios. La computación en nube hace posible el viejo sueño de la computación de servicios de la computación de servicios públicos con un sistema de pago por uso, infinitamente escalable y universalmente disponible (Sosinsky, 2011, p. 03).

1.3.6. *Fog Computing*

Conocida como computación en la niebla, es un paradigma con capacidades limitadas, como servicios de computación, almacenamiento y redes de manera distribuida entre diferentes dispositivos finales y la computación clásica en la nube. Utilizado con mayor frecuencia en aplicaciones IoT que son sensibles a la latencia (Atlam et al., 2018, p. 03).

La computación en la niebla es una extensión de la nube, pero más cercana a las cosas que funcionan con datos de IoT. Como se puede observar en la Figura 5-1, la computación en la niebla actúa como un intermediario entre la nube y los dispositivos finales, lo que acerca los servicios de procesamiento, almacenamiento y redes a los propios dispositivos finales. Estos dispositivos se denominan nodos de niebla. Cualquier dispositivo con conectividad de red, almacenamiento y computación puede ser un nodo de niebla, como controladores industriales, conmutadores, enrutadores, servidores integrados y cámaras de videovigilancia (Atlam et al., 2018, p. 03).

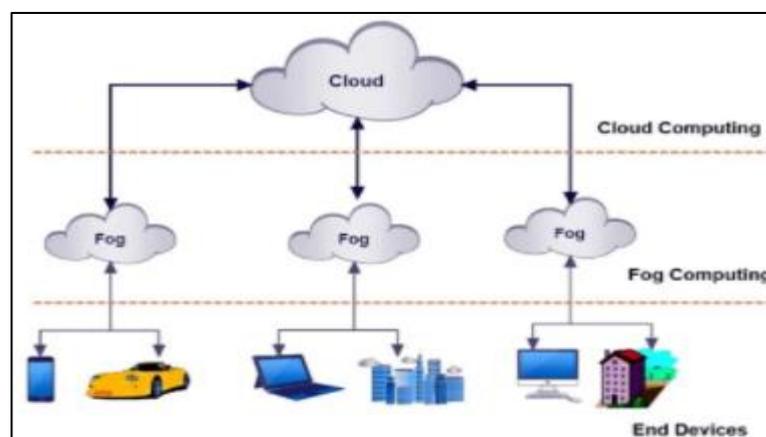


Figura 5-1: Trabajo de fog computing

Fuente: Atlam et al., 2018.

El objetivo del fog computing es reducir en lo posible las vías de comunicación entre la extensa nube y los dispositivos, para que se pueda menorar el caudal de datos en redes externas. La niebla tiene la capacidad de proporcionar una comunicación eficiente en tiempo real entre diferentes aplicaciones de IoT, como vehículos conectados, a través del proxy y los puntos de acceso ubicados de acuerdo con carreteras y caminos largos. La computación en la niebla se considera la mejor opción para aplicaciones con requisitos de baja latencia, como transmisión de video, juegos, realidad aumentada, etc. (IONOS, 2019).

La integración de la computación en la niebla con IoT traerá muchos beneficios a varias aplicaciones de IoT. La niebla admite interacciones en tiempo real entre dispositivos de IoT para reducir la latencia, especialmente para aplicaciones de IoT sensibles al tiempo.

1.3.7. *Edge Computing*

Edge Computing cuya base son las tecnologías habilitadoras que realizan computación en el borde de la red. De esta manera, los recursos informáticos y de red (bordes) están más cerca de la fuente de los datos que de los centros de datos en la nube. Edge Computing permite mejorar el rendimiento de los sistemas informáticos al reducir la latencia, reducir el costo de los recursos y aumentar la capacidad de respuesta, escalabilidad, confiabilidad, seguridad o privacidad. Las arquitecturas Edge Computing ofrecen una solución a las infraestructuras IoT porque son capaces de gestionar los datos heterogéneos generados por los dispositivos IoT. (Sittón-Candanedo et al., 2019, p. 02)

La Figura 6-1 muestra un pequeño ejemplo de una arquitectura de Computación Edge básica, de tres capas.

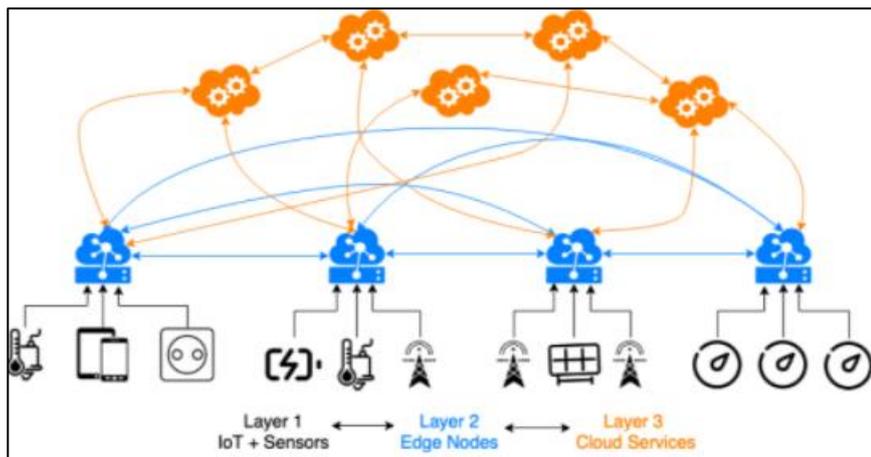


Figura 6-1: Capas de trabajo edge computing

Fuente: Sittón-Candanedo et al., 2019.

Capa 1: sensores de IoT: esta capa incluye dispositivos de IoT (sensores, medidores inteligentes, enchufes inteligentes, etc.), así como a los usuarios. La primera capa es responsable de la ingesta de datos y las operaciones involucradas.

Capa 2: nodos de borde: la segunda capa está formada por nodos de borde. Estos nodos son responsables de las operaciones de procesamiento, enrutamiento y computación de datos.

Capa 3: Servicios en la nube: esta capa está formada por múltiples servicios en la nube con mayores requisitos computacionales. Esta capa se encarga de la analítica de datos, la inteligencia artificial, el aprendizaje automático o la visualización, entre otras tareas.

Además, en muchos casos prácticos edge computing se utiliza debido que el procesamiento de datos es una necesidad fundamental más aún si se necesita una transmisión instantánea y permita evitar problemas de latencia, por ello, ofrecen diferentes interfaces para estándares de comunicación y tecnologías de transmisión que puede ser por cable (alámbrica) o radiofrecuencia (inalámbrica), como Ethernet, Wi-Fi, WLAN, Bluetooth, telefonía móvil 3G, LTE, Zigbee, Z-Wave, CAN Bus, Modbus, BACnet. o SCADA (Sittón-Candanedo et al., 2019, pp. 02-04).

1.3.8. *Relación en la agricultura*

La inserción de IoT en la agricultura se describe como una tecnología dedicada a formar redes a través de sensores los cuales puedan recolectar información de suelos o tierras aptas para cultivos, evaluando los datos que son proporcionados en tiempo real y almacenados, transmitidos hacia los agricultores para una posible toma de decisiones que sean adecuadas para el correcto desarrollo del cultivo (Pérez et al., 2019, p. 06)

Cuando se habla de IoT relacionado en la agricultura, el concepto no solo se refiere a dispositivos móviles como tabletas o drones, sino que incluye las interconexiones entre los sistemas, compatibles con información de tipo agrícola (sensores, cámaras, luces, estaciones meteorológicas, etc.) (Quiroga et al., 2016, p. 43). En el sentido tecnológico, esta relación es un gran aliado gracias a la computación en la nube para el procesamiento de datos que se pueden dar diferentes usos: uso eficiente de insumos como fertilizantes y pesticidas, correcta utilización del recurso hídrico, reducción de costos, control de ganado incluida su vigilancia, agricultura en interiores, invernaderos, peces, acuarios, todo el ámbito agropecuario industrial, tanques de combustible, asignación limitada de recursos a demanda, actualizaciones y mantenimientos que son realizadas en Back-end (parte lógica), añadiendo la colaboración con otros sistemas en la nube (Pérez et al., 2019, p. 06).

Para realizar las diferentes actividades de monitoreo se debe tener una red de este tipo, la red agrícola de IoT o la red de IoT para la agricultura (Farooq et al., 2019) es uno de los elementos vitales de IoT en la agricultura. Ayuda a monitorear los datos agrícolas y facilitar la transmisión y recepción de datos.

1.4. Tecnologías inalámbricas

1.4.1. Bluetooth

Bluetooth es una red de área personal de bajo consumo y bajo alcance que es la mejor para las comunicaciones móviles de corto alcance. La particularidad en esta tecnología es que no está diseñada para la comunicación entre redes de computadora, sino para la comunicación entre un computador y un periférico (López y Quiñonez, 2018, p. 13).

Al empezar el desarrollo (Aspiazu, 2015, pp. 16-17) se pensó que Bluetooth podía utilizarse para sustituir cables en varios dispositivos que se interconectan como impresoras, teclados, ordenadores, teléfonos móviles, etc., básicamente se trata de esta sustitución por un enlace de radio, que permite acceder o comunicarse mediante un ordenador, cámaras, u otro dispositivo a otro dispositivo Bluetooth. Las redes de este tipo tienen capacidad para comunicarse en topología punto a punto y punto multipunto, la conexión dependerá del rango que se encuentre el otro dispositivo, si se encuentra dentro del radio de cobertura, este puede establecer la conexión con cualquiera de los dispositivos.



Figura 7-1: Comunicación Bluetooth.

Fuente: López y Quiñonez, 2018.

Posee la característica de trabajar de modo maestro/esclavo donde un nodo maestro puede comunicarse hasta con siete esclavos, pudiendo comunicarse con otro esclavo y dejando al anterior en espera o parking. En definitiva, se puede trabajar dentro de dos redes, la que posibilita la comunicación de maestro y esclavos (piconet) y además donde un dispositivo puede ser maestro de una piconet y esclavo de otra, en este caso se conoce como Scatternet (López y Quiñonez, 2018, p. 13).

1.4.2. Zigbee

ZigBee está en la cima de los estándares IEEE 802 creados por ZigBee Alliance. Es un conjunto de especificaciones de dispositivo a dispositivo en la red con velocidades de datos de baja potencia. Con el avance de tecnología y debido al aumento de demanda en rendimiento, allí es donde se convirtió en una necesidad de tecnología más rápida y de bajo consumo de energía. Estos requisitos se cumplen con tecnologías más establecidas que proporcionan una transferencia de datos más rápida. En agricultura los sensores de entorno de IoT detectan los datos y los transfieren hacia servidor remoto. Después de la detección, los datos recopilados se analizan para toma de decisiones (Farooq et al., 2019).

Por ende, es una tecnología diseñada para sensores de bajo consumo, proporcionando conectividad de bajo costo, flexibilidad, escalabilidad y baja potencia, específicamente para proyectos que no consuman alta potencia. Es un conjunto definido en el estándar IEEE 802.15.4 dentro de las redes inalámbricas de área personal (WPAN) que ayudan a las comunicaciones entre dispositivos relacionados de corto alcance (López y Quiñonez, 2018, p. 16). Existen diferentes velocidades de datos que ofrece esta tecnología dependiendo de su frecuencia de operación.

Tabla 1-1: Frecuencias y velocidades de trabajo ZigBee.

Frecuencia de Operación	Velocidad de datos
2,4 GHz	250 kbps
915 MHz	40 kbps
868 MHz	20 kbps

Fuente: Loyola, 2015

Realizado por: Muñoz, D. 2021.

1.4.3. Wifi

Definido en el estándar IEEE 802.11 también conocido como WLAN (Wireless Local Área Network). Inicio como una conexión inalámbrica de mejor compatibilidad para diferentes dispositivos móviles (López y Quiñonez, 2018, p. 16). IEEE 802.11 es una compilación de estándares de comunicación de Red de área local inalámbrica que es 802.11a, 802.11b, 802.11g, 802.11n y 802.11ac. Todos estos estándares operan en diferentes anchos de banda que son 5 GHz, 2.2 GHz, 2,4 / 5 GHz, 60 GHz y 5 GHz. El rango de transferencia de datos de estos estándares es desde 1 Mb / s a 7 Gb / s. Su rango de comunicación es de 20 m hasta 100 m, todos los datos teóricos (Farooq et al., 2019) esta tecnología ha sido de gran aporte en la agricultura inteligente.

Trabaja con un sistema de seguridad denominado WPA (Acceso Protegido Wifi) que a más de corregir las deficiencias de WEP (Privacidad Equivalente a Cableado) el sistema permite la autenticación de los usuarios a través un servidor que guarda la información de cada uno (Rodas, 2013, p. 10)

En la tabla 2-1 se puede visualizar cada uno de los protocolos de mayor uso en la tecnología Wifi. Los años de diferencia entre publicaciones de protocolos es amplio debido a su desarrollo, desde la publicación de 802.11b a 802.11ac se tiene una diferencia de quince años. En su frecuencia de operación ronda entre 2.4 y 5 GHz, así mismo la velocidad de transmisión de datos tiene rangos desde 1 Mbps hasta 433 Mbps. Los alcances varían de manera amplia en los últimos años y la modulación utilizada en la mayor parte de protocolos es OFDM.

Tabla 2-1: Protocolos comúnmente usados en Wifi.

Protocolo	Año Publicado	Frecuencia	Velocidad	Distancia de Alcance	Tipo de Modulación
802.11b	1999	2,4 GHz	1-11 Mbps	50 m	DSSS
802.11 ^a	1999	5 GHz	6-54 Mbps	50 m	OFDM
802.11g	2003	2,4 GHz	54 Mbps	50 m	OFDM
802.11n	2009	2,4 GHz 5 GHz	300-600 Mbps (Teórico)	400 m (Teórico)	OFDM
802.11ac	2014	5 GHz	433 Mbps	250 m	MU-MIMO

Fuente: Navas, 2017.

Realizado por: Muñoz, D. 2021

1.4.4. *RFID*

RFID tiene una variedad de estándares que incluyen (ISO, IEC, ASTM International, DASH7 Alliance y EPC-global). Los sistemas RFID consisten en un dispositivo de lectura llamado lector y un pequeño transpondedor de radiofrecuencia llamado etiqueta RF. Esta etiqueta está programada electrónicamente con información única que tiene una característica de lectura a distancia. Hay dos tecnologías de sistemas de etiquetas RFID: la primera se llama sistema de etiquetas de lector activo y la otra se llama sistema de etiquetas de lector pasivo. Las etiquetas activas funcionan con baterías, son más caras y utilizan frecuencias más altas, mientras que la etiqueta pasiva utiliza frecuencias más bajas y no tiene una fuente de alimentación interna (Al-Sarawi, 2017, p. 686).

Debido a que la información de RFID es estática y debe programarse en la etiqueta, no se puede usar directamente para los datos de medición o diagnóstico. Algunas de las aplicaciones de IoT que usan RFID incluyen compras inteligentes, atención médica y seguridad nacional. RFID puede admitir la topología de red P2P (Al-Sarawi, 2017, p. 686).

1.4.5. *LoraWan*

LoraWan es un protocolo de comunicación de largo alcance que es desarrollado por una asociación abierta y sin fines de lucro llamada Lora TM Alliance. El propósito principal de este protocolo LoraWan es asegurar la interoperabilidad entre múltiples operadores. Para mejorar la productividad y anticipar los problemas (Farooq et al., 2019).

1.5. Requerimientos para un cultivo de papa

1.5.1. *Humedad*

La humedad relativa y moderada en el transcurso del cultivo dependerá del éxito en su cosecha. La humedad excesiva en el momento de la germinación del tubérculo y en el periodo desde la aparición de las flores hasta a la maduración del tubérculo resulta perjudicial, no debería exceder los límites de un 75%. Una humedad ambiental excesivamente alta favorece el ataque de Mildú, por tanto, esta circunstancia habrá que tenerla en cuenta (Punina, 2013, pp. 07-08).

1.5.2. *Temperatura*

Aunque hay diferencias de requerimientos térmicos según la variedad de que se trate, se puede generalizar, sin embargo, que temperaturas máximas o diurnas de 20 a 25°C y mínimas o nocturnas de 8 a 13°C son excelentes para un buen proceso en el cultivo (Punina, 2013, pp. 07-08).

1.5.3. *pH del suelo*

La papa crece mejor en suelos profundos con buen drenaje, de preferencia francos y franco arenosos, fértiles y ricos en materia orgánica. La papa puede ser sembrada en suelos arcillosos de buena preparación y buen drenaje. El pH ideal del suelo para el cultivo de papa esta entre 4,5 y 8,5. (Punina, 2013, pp. 07-08).

1.5.4. *Sistemas de riego*

Riego por infiltración o surcos: Este tipo de riego se basa en aplicar agua al terreno que por medio de surcos se lo deja escurrir, desde los que se filtra permitiendo al agua llegar hasta las raíces de la planta. Para que el agua llegue hasta las plantas es necesario solo de tubos PVC, siendo económicamente bueno para implementar. Requiere de suelos elevados pero nivelados, además de personal especializado en una distribución uniforme (Alvarado, 2015, pp. 17-18).



Figura 8-1: Riego por surcos.

Fuente: Alvarado, 2015

Riego por goteo: Este sistema consta en la acumulación de gotas de agua en el lugar de mayor concentración de la planta, este sistema lo conforman varios elementos como filtros, reguladores de presión, tubería de paso de agua y agujeros tipo goteo. La eficiencia de este sistema se encuentra entre un 90% a 95% con una emisión de agua entre 1 a 8 litros/seg/ha (Alvarado, 2015, pp. 20-21).



Figura 9-1: Riego por goteo

Fuente: Alvarado, 2015.

Riego por aspersión: este tipo de riego permite distribuir el agua a manera de lluvia, por medio de un dispositivo de gran aporte llamado aspersor. La presión del agua llega hasta el punto de salida que también se aprovecha para que se pueda esparcir de forma circular con movimientos constantes, se tiene así una transformación de energía de presión en energía cinética. Los aspersores tienen diferentes alcances y volumen de goteo, en la actualidad existe una gran cantidad de aspersores que dependerá del tipo de uso y la economía de riego, para seleccionar uno de ellos, con una red de aspersores se puede cubrir un largo terreno, teniendo un volumen de riego proporcional en todo el campo (Alvarado, 2015, p. 19).



Figura 10-1: Riego por aspersión

Fuente: Alvarado, 2015.

1.6. Tarjetas de desarrollo

Son circuitos electrónicos que contienen microcontroladores y dispositivos lógicos como puertos, reguladores o conectores, que permita a los usuarios el acceso más rápido a periféricos y el manejo de una manera menos compleja al momento de practicar en futuros proyectos y aplicaciones. La ventaja es que son compatibles con varios dispositivos de uso común como: sensores, módulos inalámbricos de distintas tecnologías, y tarjetas microSD. Además, están formadas por puertos de entradas y salidas sea analógicas, digitales o ambas, memoria de almacenamiento, puertos USB, puertos seriales, indicadores de transmisión y recepción, reguladores de voltaje, entrada para alimentación y una gran variedad de parámetros que dependen del diseño de cada fabricante (López y Quiñonez, 2018, p. 16).

1.6.1. *Arduino*

Su principal característica se define en ser hardware libre y de código abierto, posee un microcontrolador instalado, juntamente con un chip, circuitos para controlar periféricos, memoria de programa, memoria de datos y unidad central de procesamiento (CPU) como componente principal. Esta tarjeta de desarrollo permite incorporar hardware adicional como adaptador de entradas para mayor número de sensores, o componentes que permitan desarrollar el entorno de acuerdo con el requerimiento del prototipo que se vaya a implementar (Nacho, 2016, pp. 11-12).

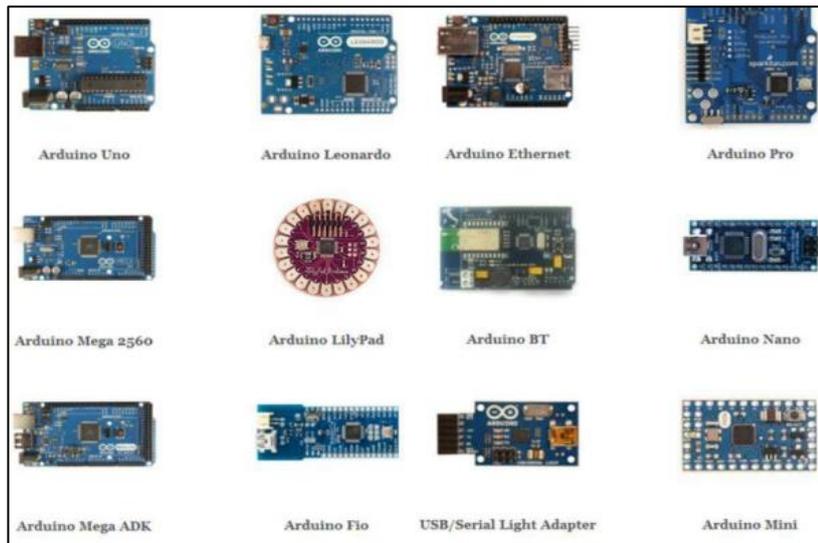


Figura 11-1: Tipos de tarjeta de desarrollo Arduino.

Fuente: López y Quiñonez, 2018.

1.6.2. *Raspberry Pi*

Es una minicomputadora de bajo costo, que no posee un disco duro instalado, pero es de importancia una tarjeta SD con una capacidad mínima de 2 Gb para poder cargar el SO Linux con la distribución Raspberry Wheezy. Cuenta con varios modelos todos incorporados memoria RAM, diferentes capacidades de procesamiento, conectores y costo, pero a partir del modelo B cuenta con un adaptador Ethernet integrado, que permite la conexión con otros dispositivos de red LAN, además que permite conectar a una red inalámbrica por medio de un adaptador USB (Arellano y Tomalá, 2015, p. 14).

En su hardware cuenta con varios accesorios como: pantallas LCD, videocámaras, también se añadió una tarjeta de comunicación celular GPRS los cuales permite la comunicación de voz por medio de bandas comerciales de los celulares (López y Quiñonez, 2018, pp. 23-24).

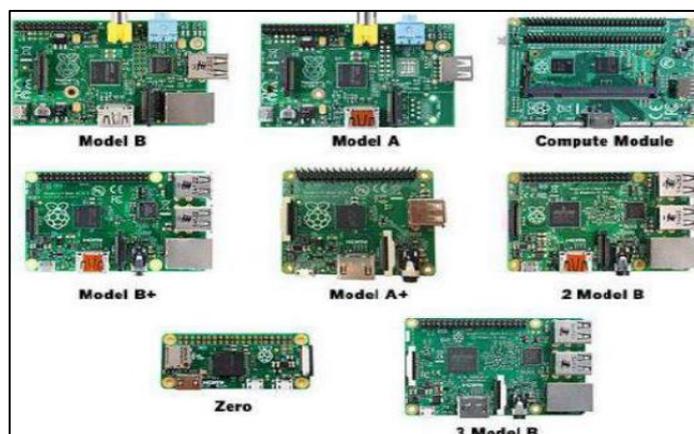


Figura 12-1: Modelos de Raspberry pi.

Fuente: López y Quiñonez, 2018.

1.6.3. *NodeMCU*

Node MCU es un módulo wifi que puede ser utilizado con GPIO, ADC, UART y PWM, 11 pines GPIO, 10 PWM, 1 pin ADC, 2 UART, WiFi 2,4GHz y WPA/ WPA2. En el caso de IoT, NodeMCU ESP8266 se convierte como cliente y controlador de equipos y objetos. NodeMCU ESP8266 puede ser una herramienta para control de sensores y enviar los datos del sensor al servidor para que se pueda controlar actuadores de diferente tipo. Si se trata, por ejemplo, de la información en el servidor también se puede notificar por correo electrónico. La NodeMCU se puede programar para el lenguaje LUA también puede ser programado en C para el IDE de Arduino, su microprocesador puede ser ESP32 o ESP8266 (Wicaksono, 2017, pp. 01-02).



Figura 13-1: NodeMCU ESP8266 v9.

Fuente: Wicaksono, 2017.

1.6.4. *BeagleBone Black*

Se puede definir como una combinación entre Raspberry Pi y Arduino debido que proporciona la potencia de Raspberry Pi y las alternativas externas que ofrece Arduino como puertos de entrada y salida, conectores y más. Consta de placas de extensión variables que, además, incluyen accesorios como: capas para conexión a impresoras en 3D, para pantallas táctiles, prototipos basados en FPGA y comunicaciones inalámbricas. Esta tarjeta se compone de un procesador que trabaja a 1 GHz, puertos de entradas y salidas, salidas PWM, entradas analógicas, puertos seriales, temporizadores y más (López y Quiñonez, 2018, p. 25).

servidores virtuales privados que puede ejecutar a Mundial, estos servidores se pueden aprovisionar con casi cualquier tipo de software de aplicación que pueda imaginar, y aprovechan una variedad de servicios de soporte que no solo hacen posibles las aplicaciones de computación en la nube distribuidas, sino también robustos (Sosinsky, 2011, pp. 179).

Amazon Web Services se basa en estándares SOA, incluidos HTTP, REST, y protocolos de transferencia SOAP, sistemas operativos comerciales y de código abierto, servidores de aplicaciones y acceso basado en navegador. Los servidores privados virtuales pueden aprovisionar nubes privadas virtuales conectadas a través de redes privadas virtuales que pueden proporcionar seguridad y control por parte del administrador del sistema (Sosinsky, 2011, p. 179).

AWS tiene una propuesta de gran valor: usted paga por lo que usa. Mientras puede que no ahorre mucho dinero con el tiempo utilizando AWS para empresas de clase de aplicaciones web, AWS tiene mucho que enseñarnos sobre el futuro de la computación en la nube y cómo se puede aprovechar mejor la infraestructura virtual como un activo comercial.

1.7.3. *Microsoft Cloud Services*

El enfoque de Microsoft es ver aplicaciones en la nube como software más servicio. En este modelo, la nube es otra plataforma y las aplicaciones pueden ejecutarse localmente y acceder a servicios en la nube o se ejecutan completamente en la nube y los navegadores pueden acceder a ellos usando estándares y protocolos de arquitectura orientada a servicios (SOA). Microsoft llama a su sistema operativo en la nube la plataforma Windows Azure. El servicio Windows Azure en sí es un entorno alojado de máquinas virtuales (Sosinsky, 2011, pp. 205-206).

Varios servicios interoperan con Windows Azure, incluido SQL Azure (una versión de SQL Server), SharePoint Services, Azure Dynamic CRM, y muchos de los servicios de Windows Live que comprenden lo que es Windows Azure Platform, que es un modelo de computación en la nube de Plataforma como servicio (Sosinsky, 2011, pp. 205-206).

2. PROPUESTA Y DISEÑO DEL PROTOTIPO

En este capítulo se describen todos los requerimientos para el diseño del prototipo, así como los dispositivos a utilizarse con características que los hacen adecuados, comunicación, software y hardware utilizados y todo lo necesario para su correcto desarrollo.

2.1. Introducción

Para el análisis de diseño e implementación del prototipo, se evalúa el escenario al que estamos inmersos, para ello se selecciona un espacio en cielo abierto de 4 metros cuadrados en un terreno con sembrío de papas ubicado a las afueras del cantón Ambato, parroquia Santa Rosa con una altitud de 3016 msnm (Pérez, 2015, p. 05).

Para continuar con los elementos en general se encuentran: sensores conectados entre sí, que evalúan los eventos del suelo como temperatura, humedad y pH, hardware libre para procesar la información, transmisor con tecnología inalámbrica que permita enviar los datos hasta el internet, servidor web con almacenamiento en la nube, conexión con notificador web que permita tener al usuario final notificaciones de los datos monitorizados, y su respectiva visualización en la aplicación móvil.

2.2. Análisis de requerimientos de diseño

En base a la información adquirida en el capítulo uno y la ayuda de la persona propietaria del terreno, se engloba los requerimientos para el diseño del prototipo, los requerimientos que se deben adaptar son:

- Debe cumplir con un diseño que permita recolectar datos de parámetros fisicoquímicos del suelo como temperatura, humedad y pH.
- Ser de bajo costo, un punto muy importante dentro de la agricultura.
- Intuitivo para el usuario, mediante una aplicación móvil.
- Presentar información de los parámetros a evaluar en la web.
- Funcionar de manera correcta, independientemente del país en el cual se implemente.
- Debe contar con un respaldo de energía.
- Los datos deben ser almacenados para su próximo análisis.
- Emitir avisos cada cierto tiempo más aun cuando los valores se encuentren fuera de los rangos de cada parámetro.

- Los dispositivos hardware del prototipo deben ocupar el menor espacio posible, para trabajos en la tierra.

Para la realización, se deberá disponer dispositivos electrónicos y eléctricos como son:

- Sensores de variables del suelo.
- Tarjeta de desarrollo.
- Actuador.
- Electroválvula.
- Cable de red.
- Dispositivos eléctricos.

Por otra parte, para la comunicación en la nube y la conexión con el usuario se requerirá de lo siguiente:

- Almacenamiento en la nube.
- Servicio de notificaciones.
- IDE de desarrollo.

2.2.1. *Arquitectura General*

Una vez analizado los requerimientos, se determinó una arquitectura general que represente al prototipo de manera organizada con cada uno de sus procesos en forma de bloques, así se puede tener los siguientes: Adquisición, Procesamiento, Transmisión, Almacenamiento, Notificación e Interfaz del usuario, como se puede observar en la Figura 1-2:



Figura 1-2: Bloques de arquitectura general.
Realizado por: Muñoz, D. 2021.

Cada uno de los bloques cumple una determinada tarea, integrando elementos y dispositivos que forman parte del proceso, a continuación, se presenta una tabla resumen de la descripción de cada bloque.

Tabla 3-2: Características del computador

BLOQUE	DESCRIPCIÓN GENERAL
ADQUISICIÓN	Es la primera etapa del prototipo en el cual los elementos (sensores) se encargan de obtener la información requerida de acuerdo con los parámetros estipulados.
PROCESAMIENTO	Corresponde a la segunda etapa que está conformado por la utilización de un hardware libre, que podrá recibir los datos recolectados por los sensores, para ser filtrados y preparar para su envío al internet.
TRANSMISION	Dispositivo hardware que ayuda a transmitir la información hacia la red, que en el caso de IoT tendrá que ser una tecnología inalámbrica como por ejemplo WiFi.
ALMACENAMIENTO	Plataforma que ayuda a almacenar los datos en la nube, además que permita la visualización y descarga de cada uno de estos.
NOTIFICACION	Software que tiene la capacidad de alertar al usuario final de todos los datos que proporciona el prototipo en base a los parámetros estipulados.
INTERFAZ DE USUARIO	A partir de una aplicación móvil, el usuario podrá tener una interfaz amigable, que muestre todos los datos que proporcione el prototipo, además de poder controlar, en este caso, el riego en el cultivo.

Fuente: Morocho y Quinapanta. 2016

Realizado por: Muñoz, D. 2021

2.3. Selección de dispositivos para el prototipo

A continuación, se presentan todos los elementos utilizados para la implementación del prototipo, así como las características principales de cada uno de estos, intervienen varios elementos como: sensores, tarjetas de desarrollo, controladores y otros dispositivos.

2.3.1. Sensores

Son utilizados para la medición de factores o parámetros, los cuales están diseñados específicamente para captar y arrojar valores del parámetro evaluado (Morocho y Quinapanta, 2016, p. 27).

Para cumplir con los objetivos del prototipo, los sensores utilizados para medir los eventos del suelo son:

- Humedad
- Temperatura
- Acidez del suelo

2.3.1.1. Sensor de Humedad

El módulo sensor de humedad del suelo de la familia HL, utiliza la conductividad entre sus dos terminales para especificar algunos parámetros en relación con el agua. Este sensor tiene la capacidad de medir la humedad mediante una pequeña tensión que se genera entre ambos terminales, la corriente depende mucho de la resistencia que se genera en el suelo y depende de la humedad, es así como, al aumentar la humedad la corriente crece y al bajar, la corriente disminuye (E-Elektronic, 2015).

Este dispositivo se compone de dos partes: el sensor, que consiste en dos terminales en forma de punta y el módulo de acondicionamiento de señales que procesa las señales provenientes del sensor para entregar a otro sistema analógico o digital. Es recomendado en la utilización de maquetas y proyectos que no sean permanentes, debido que la humedad y corriente puede corroer fácilmente los terminales, por esto se recomienda conectar la alimentación las veces que se realicen mediciones. (Electrónica Caribe, 2015).

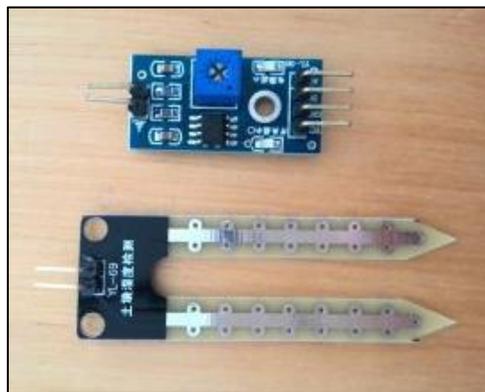


Figura 2-2: Modulo HL-69 sensor de humedad del suelo.

Fuente: E-Elektronic, 2015.

Algunas de sus características técnicas principales son:

- Salida: sin calibrar, el usuario deberá calibrar en estado óptimo dependiendo la aplicación.
- Voltaje de alimentación: 5 VDC.
- Sensor de humedad en suelo basado en conductividad eléctrica.
- Ideal para mediciones en maquetas no permanentes.
- Basado en el comparador LM393.
- Salida: analógica y digital.

2.3.1.2. Sensor de Temperatura

El sensor de temperatura y humedad relativa DHT11 es un sensor digital y uno de los más

sencillos de utilizar a nivel de software y hardware, está integrado por un termistor que permite medir el aire circundante en el lugar de uso, es uno de los más utilizados para aplicaciones académicas relacionados al control automático de temperatura, aire acondicionado, monitoreo del ambiente en general, temperaturas para monitoreo de agricultura, y más.

Se puede utilizar con varias tarjetas de desarrollo como Arduino, Raspberry pi, Nodemcu, debido a compatibilidad y sencillez en software ya que dispone librerías disponibles para el IDE de Arduino, en cuanto a hardware, se puede conectar el un pin para VCC de 3 a 5V de alimentación, el pin de GND a tierra y el pin de datos a un pin digital. Cada sensor utilizado, son calibrados de fábrica y sus coeficientes son almacenados en la memoria OTP, así puede asegurar su escalabilidad y fiabilidad en el tiempo de uso. Un parámetro de suma importancia es el protocolo de comunicación hasta el microcontrolador, que en este caso es un único hilo o cable de red, la distancia recomendada es máximo 20 metros (Naylamp Mechatronics, 2021).

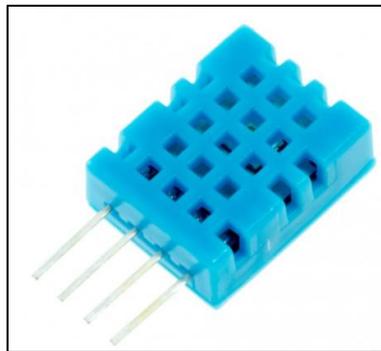


Figura 15-2: Sensor de temperatura DHT11.

Fuente: Naylamp Mechatronics, 2021.

Estas son las especificaciones técnicas del sensor DHT11:

- Voltaje de Operación: 3V - 5V DC
- Rango de medición: 0 a 50 °C
- Precisión de medición: ± 2.0 °C
- Resolución Temperatura: 0.1°C
- Tiempo de sensado: 1 seg.
- Interfaz digital: Single-bus (bidireccional)
- Modelo: DHT11
- Dimensiones: 16*12*5 mm
- Peso: 1 gr.
- Carcasa de plástico celeste

2.3.1.3. Sensor de pH

El pH del suelo es una medida de acides o alcalinidad, que permite que el suelo sea apto para el cultivo de ciertos productos. El pH (Calapiña, 2018, p. 61) nos indica la concentración de iones hidronio [H₃O⁺] presente en determinada sustancia. En la tabla 2-2 se muestra las características de dos sensores comerciales que se pueden obtener.

Tabla 4-2: Especificaciones módulo sensor de pH

Definiciones	SEN0161	GHL. Proflux pH electrode BNC
Voltaje de operación	5V	4.5 V
Medición	0-14 pH	0-14 pH
Medición de temperatura	0 – 60 °C	0 – 60 °C
Precisión	± 0.1 pH (25°C)	± 0.05 pH
Tiempo de respuesta	≤ 1min	≤ 1min
Conector	Con conector BNC	Con conector BNC
Tamaño de modulo	43 x 32mm	150mm

Fuente: Calapiña, 2016

Realizado por: Muñoz, D. 2021

Al contar con la posibilidad de presentar dos o más alternativas que pueden ser utilizadas, es recomendable utilizar una “Escala de Likert” que por medio de su método permita ponderar las variables de mayor relevancia, por ello, se proceden a ser evaluados por medio de esta escala de modo cualitativa. Esta escala de Likert estará ponderada en orden descendente con valores de 5 a 1, siguiendo sus propias recomendaciones, en la tabla 3-2 se presenta los valores y la escala de tipo importancia a utilizar para todos los dispositivos usados, en los que se pueda realizar una comparación.

Tabla 5-2: Escala de Likert para los dispositivos.

Muy Bueno	Bueno	Moderadamente Bueno	Regular	Malo
Sin dificultad	Básico	Intermedio	Avanzado	Muy Avanzado
Si				No
5	4	3	2	1
100%	80%	60%	40%	20%

Realizado por: Muñoz, D. 2021

Ponderación: se realiza de acuerdo con los requerimientos en implementación de este prototipo, a las variables individuales que presentan diferencias relevantes.

- **Alimentación:**

Tabla 6-2: Evaluación factor de alimentación.

Parámetro	SEN0161	GHL. Profilux pH electrode BNC
Voltaje	5 V	4.5 V
Likert	Muy Bueno	Regular
Puntuación	5	2

Realizado por: Muñoz, D. 2021

La alimentación de 5 V será la de mayor importancia debido a que la mayoría de los sensores la tienen como preferencia para entrar en funcionamiento, además pueden compartir la alimentación con la mayoría de las tarjetas de desarrollo, ya que trabajan con 5 voltios.

- **Rango de medición:**

Tabla 7-2: Evaluación de factor rango de medición.

Parámetro	SEN0161	GHL. Profilux pH electrode BNC
Rango	0-14 pH	0-14 pH
Likert	Muy Bueno	Muy bueno
Puntuación	5	5

Realizado por: Muñoz, D. 2021

El rango de medición de cada sensor se encuentra en el adecuado, debido que se mide valores en suelos que suponen ser ácidos, en este parámetro ambos sensores comparten el mismo puntaje.

- **Precisión:**

Tabla 8-2: Evaluación de factor de precisión.

Parámetro	SEN0161	GHL. Profilux pH electrode BNC
Rango	± 0.1 pH	± 0.05 pH
Likert	Bueno	Muy bueno
Puntuación	4	5

Realizado por: Muñoz, D. 2021

En todo tipo de dispositivo de medición mientras más preciso sea, mejores resultados reales se obtienen.

- **Tiempo de respuesta:**

Tabla 9-2: Evaluación de factor de tiempo de respuesta

Parámetro	SEN0161	GHL. Profilux pH electrode BNC
Rango	≤ 1min	≤ 1min
Likert	Muy Bueno	Muy bueno
Puntuación	5	5

Realizado por: Muñoz, D. 2021.

Ambos sensores, coinciden en este parámetro en donde su tiempo de respuesta es menor o igual a uno.

- **Tamaño de módulo:**

Tabla 10-2: Evaluación de factor tamaño del módulo.

Parámetro	SEN0161	GHL. Profilux pH electrode BNC
Rango	43mm x 32mm	150mm
Likert	Muy Bueno	Malo
Puntuación	5	1

Realizado por: Muñoz, D. 2021.

El tamaño del módulo tiene que ser relativamente pequeño, para que su ensamblaje en circuitería ocupe el menor espacio posible ya que funcionará en un cultivo donde la tierra es trabajada constantemente.

Debido a estas variables consideradas de mayor relevancia, se obtiene un porcentaje de resultado final evaluado sobre los 25 puntos que representara el 100% así se obtiene un puntaje final de cada opción a seleccionar. La opción SEN01601 obtiene un valor de 24 puntos sobre 25 representando un 96% en las variables evaluadas, mientras que GHL. Profilux pH electrode BNC obtiene una puntuación de 18 sobre 25 representando un 72% en las variables evaluadas.

Teniendo una evaluación con características muy similares, se seleccionó el sensor SEN0161 debido a que, a más de sus características revisadas y obteniendo el mayor puntaje, este sensor analógico fue desarrollado para que se lo utilice con el IDE de desarrollo de Arduino, debido a su alta utilización, tenido una gran ventaja a su favor.

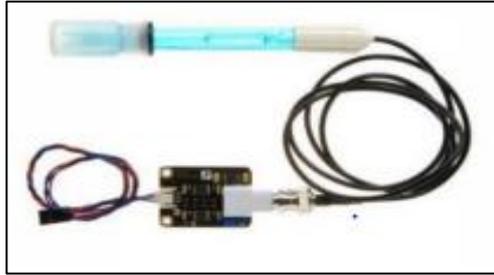


Figura 16-2: Modulo SEN0161 de pH del suelo.

Fuente: Calapiña, 2016.

2.3.2. Tarjetas de desarrollo

Constituye una parte fundamental del prototipo, debido que será el núcleo que recolecta la información enviada por los sensores para lograr procesarlos, filtrar y finalmente poder enviarlos hacia el internet. Además, la tarjeta de desarrollo tendrá la capacidad de comunicarse con servidores compatibles, que permitan una conexión directa (Vilañez, 2019, p. 44).

En la tabla 9-2 se muestra una comparación de las tarjetas de desarrollo más utilizadas en prototipos IoT, tomando en cuenta aspectos técnicos de importancia para su utilización.

Tabla 11-2: Tipos de tarjetas de desarrollo para el prototipo.

Tarjetas de desarrollo			
Parámetros técnicos	Arduino	Nodemcu	Raspberry Pi
Costo aproximado	\$22 a \$80	\$18 a \$40	\$60 a \$150
Microcontrolador	ATmega 2560	Risc 32 bits MCU ESP32 / ESP8266	Procesador ARM1176JZF-S core
Pines	54 DI/O, 12 AI, 2AO	16 pines GPIO	40 pines GPIO
Frecuencia de reloj	16 Hz	80 – 160 MHz	1.2 Ghz
Memoria de programa	256 Kb	96 Kb	512 Mb
Memoria RAM	8 Kb	64 Kb	1 Gb
Voltaje de operación	5V, 3.3V	3 a 3.3V	5V
Corriente de operación	40 – 50 mA	80 mA	500 mA
Modulo WiFi	No	Incorporado	De acuerdo con la versión
Programación	Basado en C++	Varios lenguajes	Varios lenguajes

Fuente: Calapiña, 2016

Realizado por: Muñoz, D. 2021.

A continuación, se presenta una tabla general que ayude a evaluar cada una de las opciones, en base a la escala presentada anteriormente.

Tabla 12-2: Evaluación por escala de Likert a tarjetas de desarrollo.

Variable	Arduino	Nodemcu	Raspberry pi
Microcontrolador			
Cualidad	Bueno	Muy bueno	Muy bueno
Ponderación	4	5	5
Puertos			
GPIO	Si	Si	Si
Ponderación	5	5	5
Frecuencia de reloj			
> 50 Hz	No	Si	Si
Ponderación	1	5	5
Rango Ghz	1	1	5
Memoria del programa			
Cualidad	Buena	Muy buena	Moderadamente Buena
Ponderación	4	5	3
Memoria RAM			
Cualidad	Moderadamente Buena	Buena	Muy buena
Ponderación	3	4	5
Alimentación			
< 5V	5	5	1
> 5V	5	1	5
Módulo			
Wifi	No	Si	Si
Ponderación	1	5	5
IDE desarrollo			
Soporte Windows 10	Si	Si	No
Ponderación	5	5	1
Dificultad	Intermedio	Avanzado	Avanzado
Ponderación	3	2	2
Costo			
Cualidad	Bueno	Muy bueno	Regular
Ponderación	4	5	2
TOTAL	41	48	44

Realizado por: Muñoz, D. 2021

La evaluación total se obtiene sumando cada uno de los puntajes por variable alcanzados, teniendo así los siguientes puntajes:

Arduino = 41 puntos

Nodemcu = 48 puntos

Raspberry Pi = 44 puntos

Para obtener un porcentaje se realiza una operación sobre 60 puntos en total, obteniendo así los siguientes porcentajes: para Arduino un 68,33% , Nodemcu un 80% y Raspberry Pi un 73,33%, teniendo un mayor porcentaje en todas las variables para la tarjeta Nodemcu llevando una pequeña ventaja de 6,67% frente a Raspberry Pi y 11,67% frente a Arduino, por lo que se decide trabajar con Nodemcu Esp32 para cumplir con los requerimientos propuestos para el prototipo, además, debido a su incorporación de modulo WiFi, la compatibilidad con servidores para IoT, la compatibilidad con IDE de desarrollo libre como Arduino y más características específicas que se muestra a continuación.

2.3.3. Tarjeta Nodemcu

NodeMCU es un microcontrolador que tiene más funciones en comparación con otros microcontroladores como Arduino. Es una tarjeta que se encuentra orientada a la creación de prototipos IoT, este microcontrolador tiene más pines de entrada y salida que se pueden utilizar y facilitar la creación de un sistema que utilice muchos pines. Además, está equipado con modulo Wi-Fi que tiene velocidades cada vez más rápidas. un Bluetooth de baja energía de dos modos, por lo que para hacer un dispositivo que requiera la función de wi-fi o Bluetooth no es necesario utilizar componentes por lo que no ocupa mucho espacio y ciertamente es rentable (Akbar, 2020, p. 33).

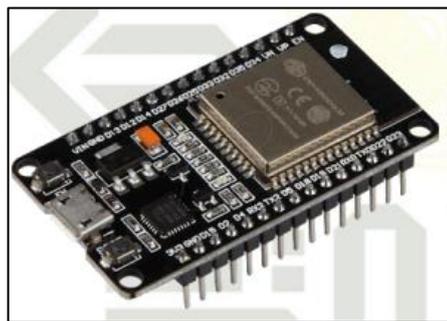


Figura 17-2: Modulo ESP32.

Fuente: Akbar, 2020

En el núcleo de este NodeMCU se encuentra un microprocesador Tensilica Xtensa LX6 de doble o único núcleo con interruptores y amplificador de potencia. Filtros, módulos de administración de energía y Amplificador de recepción de bajo ruido. Se puede utilizar para dispositivos móviles, dispositivos electrónicos, y actualmente se puede utilizar para aplicaciones de IoT (Akbar, 2020, p. 35-36).

En la Figura 6-2, se presenta una arquitectura de la tarjeta Nodemcu ESP32 a manera de diagrama de bloques, destacando sus principales componentes.

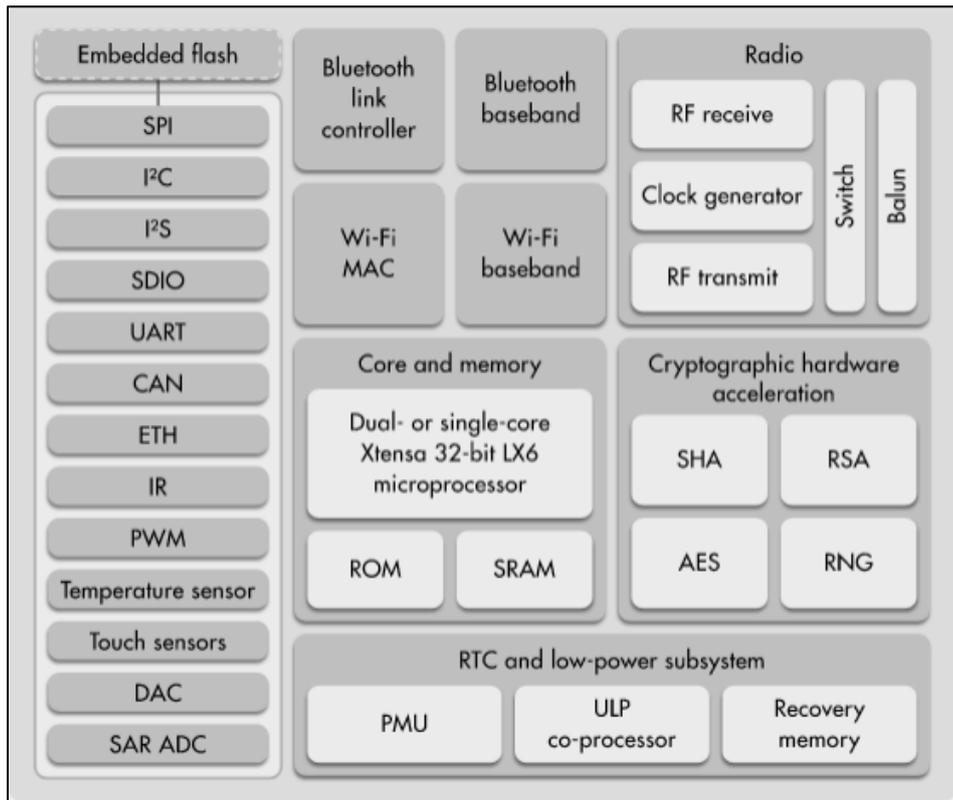


Figura 18-2: Arquitectura de Nodemcu ESP32.

Fuente: Akbar, 2020.

Las principales características técnicas que destacan a un microprocesador ESP32 son:

Tabla 13-2: Características técnicas Esp32.

CARACTERISTICAS	ESP32
Microprocesador	Xtensa Dual-Core 32-bit LX6 con 600 DMIPS
WiFi	HT 40 (802.11 b/g/n)
Frecuencia de operación	160 Mhz
SRAM	448 KB
Flash	520 KB
GPIO	34
PWM (software)	16 canales
ADC	12-bits de resolución
Interfaz MAC ethernet	Si
Temperatura de trabajo	-40°C to 125°C

Fuente: Guerra, 2021

Realizado por: Muñoz, D. 2021.

2.3.3.1. Requisitos para la utilización

Para el desarrollo del prototipo se presentan los requisitos que debe cumplir el software en la utilización de la tarjeta Nodemcu Esp32.

Tabla 14-2: Requisitos en software.

Dispositivos	Sistema operativo	Entorno de desarrollo	Lenguaje de programación
ESP32	-	IDE de desarrollo	C++
Computador	Windows 10	IDE de desarrollo	C++

Realizado por: Muñoz, D. 2021

2.3.4. Actuador

En base a la definición presentada en el marco teórico, se utiliza como actuador una electroválvula que permita el control del riego dentro del cultivo de papa, además, servirá para poder controlar el nivel de humedad y en cierta medida el nivel de temperatura.

2.3.4.1. Electroválvula

Para el desarrollo del prototipo Es un dispositivo que está diseñado para controlar el flujo de todo tipo de fluidos en este caso agua, a través de un conducto. El movimiento en este tipo de válvulas se produce por una bobina solenoide, que diferencia de las válvulas mecánicas. Se puede activar por medio de una señal eléctrica al solenoide de la electroválvula, la bobina se imanta y levanta el embolo (Arco, 2020).

Para una mejor visualización en la Figura 7-2, se puede observar los componentes de una electroválvula.

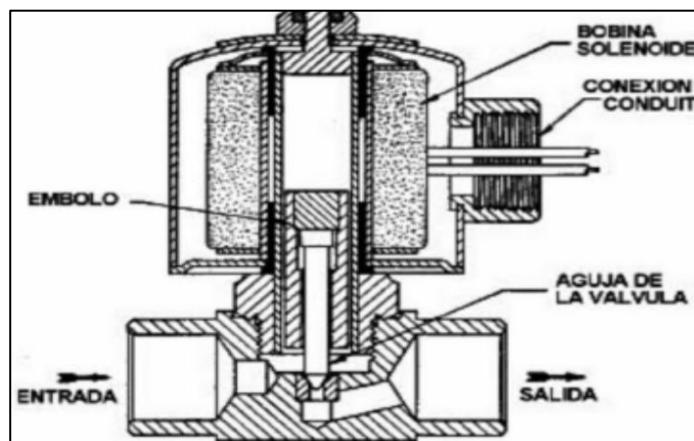


Figura 19-2: Componentes de una electroválvula.

Fuente: Arco, 2020.

2.4. Software

Es una etapa crucial para el desarrollo del prototipo, debido a que es necesario una plataforma para poder programar cada uno de los bloques de comunicación, como sensores, comunicación a la nube y códigos en los servidores a utilizar. Específicamente se necesita de un IDE de desarrollo, servidores en la nube, y un IDE de desarrollo para la interfaz final del usuario.

2.4.1. IDE de desarrollo

Una vez seleccionado el hardware en el cual se va a trabajar, la parte más importante es la selección del entorno de desarrollo, teniendo muy en cuenta la compatibilidad con la tarjeta Nodemcu ESP32 y que posea librerías disponibles para la comunicación hacia el internet, ya que será el medio por el cual se puede acceder a los servicios en la nube.

En la Tabla 13-2, se presenta una comparación de diferentes entornos de desarrollo que se enfocan en prototipos IoT, con sus características que hacen que uno sea mejor que otro entorno, específicamente los que se puede utilizar con la ESP32. De los cuales se escogerá la mejor opción.

Tabla 15-2: Entornos de desarrollo.

Especificaciones	Arduino IDE	Eclipse	WhiteCat	MonGoose
Lenguaje de programación	C++	C++ Java	Lua	C++, C Y JavaScript
Sistemas operativos	Windows Linux	Windows Linux	Windows	Mongoose
Hardware soportado	Arduino, Esp8266, Esp32, etc.	Arduino, Esp32, etc.	Esp8266, Esp32, etc	ST Microelectronicas Sistemas Empressif
Acceso a librerías	Si	Si	Si, básico	Si, básico
Tipo de acceso	Escritorio	Escritorio	Navegador	Escritorio
Gratuito	Si	Si	Si	Si
Complejidad	Media	Media	Alta	Alta

Realizado por: Muñoz, D. 2021

Como son variables a consideración y se pueden ponderar, se procede a presentar una tabla con la misma escala de referencia que se está utilizando

Tabla 16-2: Evaluación por escala de Likert a entornos de desarrollo

Variable	Arduino IDE	Eclipse	WhiteCat	MonGoose
Lenguaje de Programación				
Un lenguaje	5	1	5	1
Dos o más	1	5	1	5
Sistema operativo				
Windows	5	5	5	1
Linux	5	5	1	1
Soporte de hardware				
ESP32	5	5	5	1
Software libre				
Ponderación	5	5	5	5
Complejidad				
Ponderación	3	3	1	1
TOTAL	29	29	23	15

Realizado por: Muñoz, D. 2021

La evaluación presentada se lo hará sobre 35 puntos en total, dando así las siguientes puntuaciones para cada uno de los IDE de desarrollo:

Arduino IDE = 29 puntos

Eclipse = 29 puntos

WhiteCat = 23 puntos

MonGoose = 15 puntos

Realizando los porcentajes sobre el puntaje total, se tiene los siguientes resultados: para Arduino IDE y Eclipse se tiene un empate de 82,857% para ambos entornos de desarrollo frente a un 65,71% de WhiteCat que es el que sigue a continuación, y se llevan una ventaja amplia frente a MonGoose con un 40% de diferencia. De acuerdo con este análisis comparativo se decide trabajar con IDE de Arduino por todas sus características y facilidades como compatibilidad con la tarjeta que se está utilizando (Nodemcu Esp32), la complejidad de nivel intermedio, lo más importante que es un software de desarrollo libre y que existen librerías que se pueden utilizar con mayor facilidad en un lenguaje de programación C++ que es muy utilizado, amigable y conocido por varios usuarios.

2.4.2. Nodemcu ESP32 con el IDE de desarrollo

Luego de haber seleccionado nuestro IDE de desarrollo con sus respectivas razones, y el hardware a utilizar, se puede destacar que al momento de instalar el entorno de desarrollo Arduino no viene precargado estas tarjetas de desarrollo de la familia ESP, teniendo que realizar una instalación plugin con estos modelos de tarjeta, por ello, se presenta el paso principal para que puedan ser

compatibles sin ningún problema, se debe acceder al gestor de URL adicionales de Arduino, que proporciona la dirección donde se encuentran los archivos a descargarse para continuar con la instalación en el IDE.

En la Figura 8-2, se presenta el gestor de tarjetas de desarrollo en la ventana que procede a la instalación del microcontrolador del tipo ESP32, para que pueda ser utilizado.

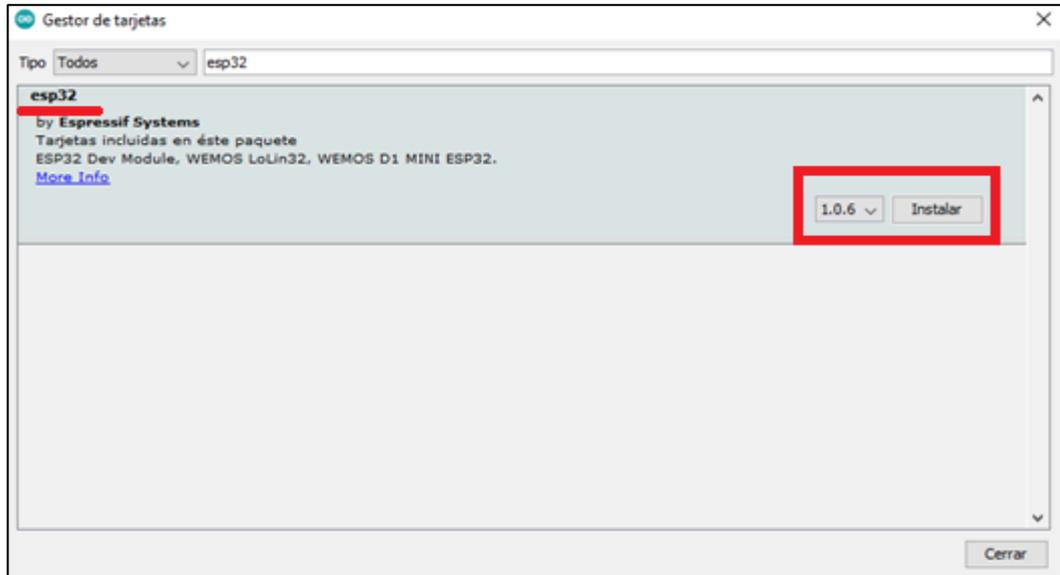


Figura 20-2: Gestor de tarjetas en IDE de Arduino.

Realizado por: Muñoz, D. 2021.

Una vez instalado el firmware, se puede observar que en la pestaña de placas se puede escoger la escogida para este prototipo y seleccionar entre todas las que ofrece como se muestra en la Figura 9-2.

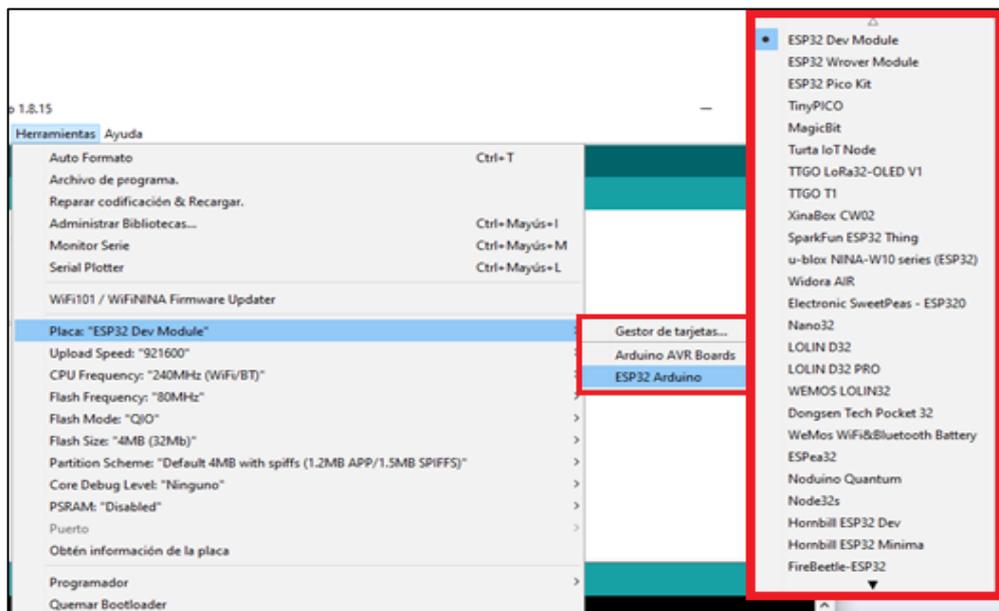


Figura 21-2: Modulo a ser utilizado.

Realizado por: Muñoz, D. 2021.

2.4.3. *Servidores Web*

Existen varios servidores Web que se pueden ocupar de acuerdo con las necesidades del prototipo, en este proyecto de investigación se trata de encontrar un servidor que nos permita almacenar datos y poder visualizarlos de la mejor manera, un punto de importancia es que el servidor de cualquier plataforma nos permita la compatibilidad con el entorno de desarrollo utilizado en este caso IDE de Arduino. Además, la gratuidad de su uso es otro parámetro de consideración, debido a los requerimientos de implementación.

El servidor de una plataforma se presenta como una aplicación que permite gestionar recursos, en donde el usuario podrá ser parte de esta plataforma siempre y cuando sea registrado en ella, el centro de operación será brindar respuesta con su hardware de manera instantánea. (Analuisa, 2018, p. 47)

En este punto es necesario conocer acerca de la capa de servidor, que está dividido en cinco subsistemas que son de análisis para el manejo del servidor que se puede escoger y son responsables de implementar la funcionalidad de un servidor Web, se presentan a continuación:

Subsistema de recepción: representa la primera línea de comunicación, su función es esperar las peticiones HTTP de los diferentes clientes que llegan a través de la red. Además, analizan las peticiones y determinan la capacidad como: tipo de navegador, compatibilidad, etc. Tiene la capacidad de manejar múltiples peticiones.

Analizador de peticiones: es el encargado en traducir el lugar del recurso de la red al nombre del archivo local.

Control de acceso: aquel que permite el ingreso o no a la plataforma (autenticación)

Manejador de recursos: subsistema de importancia al momento de solicitar el recurso el cual están solicitando, lo ejecuta y genera su respuesta.

Registro de transacción: como su nombre indica, encargado en registrar todas las peticiones y los resultados.

Existen varios servidores que permiten contar con características similares, pueden ser libres o pagados, la Tabla 15-2 muestra algunas de ellas con sus distintas características que lo hacen la mejor opción para que pueda ser utilizado. Se puede presentar a Temboo, ThingSpeak, Nimbits, Plot.ly, Thinger.io, entre otras.

Tabla 17-2: Plataformas con servicios en la nube.

Servidores	Temboo	ThingSpeak	Nimbits	Thingier.io
Licencias	14 días libres	Libre Restringida	Libre y Pagada	Libre y Pagada
Categoría	Internet de las cosas	Internet de las cosas	Artistas	Internet de las cosas
Software	Editable	No editable	Editable	No editable
Aplicaciones	Una visión del IoT aplicado a la industria	Domótica con Matlab	Control sobre su música	Monitoreo
Conexión	Ethernet, Wifi, Gsm	Ethernet, Wifi, Gsm	Ethernet, Wifi, Gsm	Ethernet, Wifi, Gsm
IDE de Integración	Arduino, Raspberry Pi, Servidor público	Arduino Raspberry Pi	Arduino Raspberry Pi	Arduino Raspberry Pi
Limitaciones	El alcance de 100 APIs (interfaz de programación de aplicaciones)	Para tres canales	500 MB	2 dispositivos y 4 Paneles

Fuente: Analuisa, 2018

Realizado por: Muñoz, D. 2021

Una vez revisada las alternativas, la opción más adecuada para el prototipo es Thingier.io debido a su aplicabilidad en monitoreo que es la base de este trabajo de investigación, ayudando a cumplir con sus objetivos, además, dispone de compatibilidad con el software de desarrollo Arduino y es libre para la utilización de hasta dos dispositivos, en este caso se utilizará un dispositivo la ESP32. En la parte de paneles de monitoreo soporta hasta cuatro en su versión libre y es suficiente para este prototipo que utilizará tres de ellos debido a sus tres sensores, por lo que se puede decir que esta plataforma de servidor es la más completa, e idónea en este tipo de prototipos.

2.4.3.1. Thingier

Thingier.io proporciona un servicio en la nube listo para usar para conectar dispositivos a Internet para realizar cualquier activación o detección remota a través de Internet. Ofrece un nivel gratuito para conectar un número limitado de dispositivos, pero también es posible instalar el software fuera de la nube para una gestión privada de los datos y dispositivos conectados a la plataforma, sin ninguna limitación.

Esta plataforma es independiente del hardware, por lo que es posible conectar cualquier dispositivo con conectividad a Internet, desde dispositivos Arduino, Raspberry Pi, Nodemcu dispositivos Sigfox, etc. La plataforma proporciona algunas funciones listas para usar, como el registro de dispositivos; comunicación bidireccional en tiempo real, almacenamiento de datos y

configuración, gestión de identidad y acceso, para permitir que entidades de terceros accedan a la plataforma y los recursos del dispositivo a través de API REST / WebSocket, para que los dispositivos puedan llamar fácilmente a otros servicios web, enviar correos electrónicos, SMS, enviar datos a otras nubes, etc. También proporciona una interfaz web para administrar todos los recursos y generar cuadros de mando para el monitoreo remoto (Bustamante et al., 2019).

Además, posee un beneficio único con las últimas tarjetas de desarrollo utilizadas para proyectos IoT como los basados en Arduino, ESP8266, ESP32, LinkitOne, Texas Instruments CC3200, Libellium Waspnotes, Raspberry Pi, etc. Ofreciendo una biblioteca cliente que proporcionan una forma integral de conectar dispositivos y enviar información a la nube, sin tener que lidiar con complejos protocolos de IoT. (Bustamante et al., 2019)

Los servicios que presta este servidor, en su versión libre se muestra en la Figura 10-2.

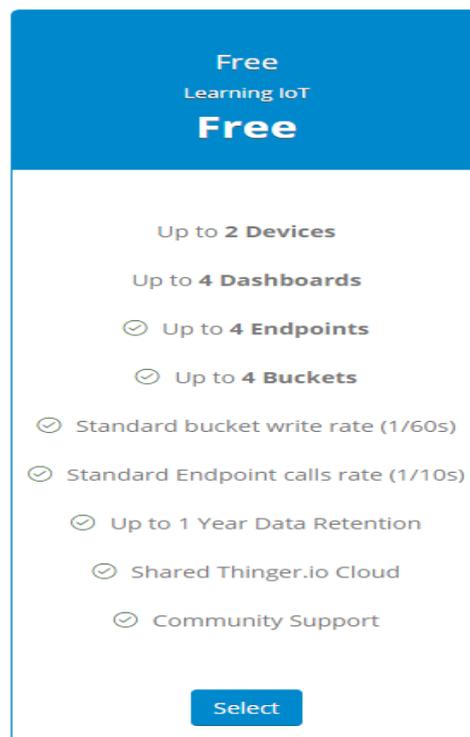


Figura 22-2: Servicios gratuitos en Thinger.io.

Fuente: Analuisa, 2018.

Una vez que, como usuarios del servidor, podamos ser registrados y obtener los servicios libres, se presentará una interfaz para desarrollar todo lo que refiere a comunicación y monitorización en la nube, como se muestra en la Figura 11-2.

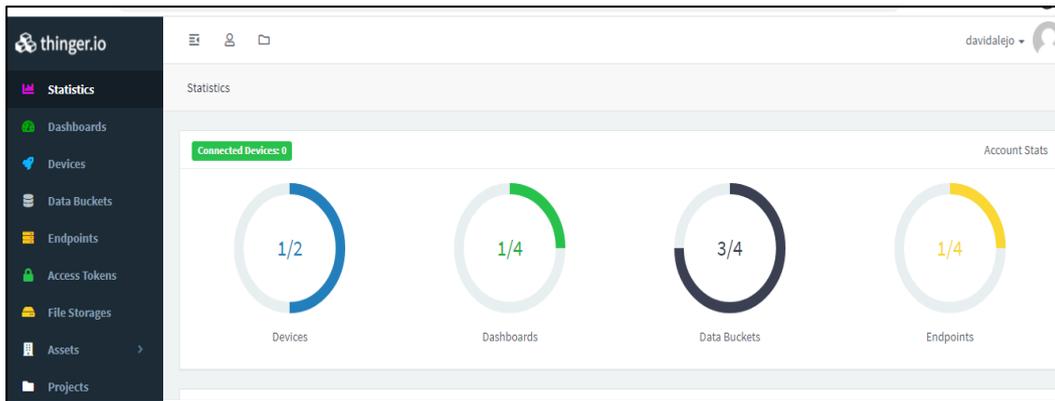


Figura 23-2: Interfaz de inicio en Thinger.

Realizado por: Muñoz, D. 2021.

2.4.3.2. Comunicación

Las plataformas basadas en HTTP pueden ser fáciles en desarrollo, y utilización, ya que operan como servidores HTTP regulares que reciben datos de sensores que utilizan tecnología conocida, como en este caso Wifi gracias al módulo incorporado en la tarjeta Nodemcu Esp32. Sin embargo, tales enfoques no pueden proporcionar un mecanismo eficiente para una comunicación bidireccional, es decir, cuando es necesario actuar sobre un dispositivo o configurarlo en tiempo real. Los servidores basados en HTTP se basan en un mecanismo de sondeo en el que los dispositivos comprueban periódicamente el servidor en busca de nuevos comandos o actualizaciones. Mientras tanto, Thinger.io proporciona un canal de comunicación bidireccional entre el dispositivo y el servidor en la nube, por lo que cualquier aplicación puede interactuar con el dispositivo en tiempo real, aumentando la cantidad de casos de uso que se pueden desarrollar con la plataforma.

Las otras soluciones basadas en MQTT también proporcionan comunicación bidireccional entre servidores y dispositivos. Sin embargo, este protocolo solo se enfoca en la capa de transporte, permitiendo que diferentes puntos finales se comuniquen, pero no define un lenguaje para modelar información; no proporciona un esquema de codificación eficiente como Protoson, o no permite una fácil interoperabilidad con aplicaciones y servicios de terceros, es decir, a través de una API REST para consultar información en tiempo real desde el dispositivo (Bustamante et al., 2019).

2.4.3.3. Servidor de mensajería

El servidor de mensajería se encarga de enviar solicitudes de mensajes a los dispositivos seleccionados para el usuario final, teniendo la capacidad de notificar en este dispositivo los datos enlazados.

Estos servidores de mensajería para notificación utilizan la tecnología Push, la cual tiene un modo de comunicación cliente/servidor, en donde se diferencia el servidor por ser donde se genera la petición y quien desea enviar los mensajes hasta su cliente. Además, la otra tecnología que trabaja de modo diferente es conocida como Pull, en este caso el cliente es donde se genera las peticiones. En base a estas características, se puede encontrar varios servidores de mensajería que en la actualidad son los más utilizados para proyectos de implementación y se presentan en la Tabla 16-2.

Tabla 18-2: Servidores de mensajería.

Servidores	Firestore Cloud Messaging	Meteor	Mean Stack
Definición	Base de datos para app en tiempo real. Notificación de Mensajería con compatibilidad en dispositivos.	Web Framework	Web framework
Base de datos (DB)	Real-tiempo sobre Mongo (DB basado en código abierto)	Interfaz de programación de aplicaciones (API) de DB de Mongo	Mongo DB
Paquetes	Empaquetamiento simplificado	Escala basada	Libre
Interfaz	Firestore, React.js y otras	Blaze, Angular, React.js	Mongo, Express, Angular
Límite de programación	Programación limitada	Programación Libre	Libre con compleja programación
Nivel de complejidad	Media, programación en la Web, iOS, Android apps.	Fácil Programación para Web Apps	Complejo en el lenguaje de programación

Fuente: Analuisa, 2018

Realizado por: Muñoz, D. 2021

2.5. Interfaz de usuario

Para la interfaz del usuario de este prototipo se han planteado algunos entornos de desarrollo oficiales para la aplicación del usuario final, del cual se presentan sus respectivas descripciones en la Tabla 17-2, se seleccionó la herramienta de Android Studio por su amplia información en diferentes fuentes para que la interfaz pueda ser diseñada.

Tabla 19-2: Herramientas de diseño de aplicaciones Android.

APLICACIÓN ANDROID		
Herramienta	Descripción	URL Descarga o Instalación
Android Studio	Android Studio es un nuevo entorno de desarrollo que fue presentado por Google como alternativa para la creación de apps.	https://android-studio.uptodown.com/windows
Java Platform (JDK) 8	Java Development Kit o (JDK), es un software que provee herramientas de desarrollo para la creación de programas en	http://www.oracle.com/technetwork/es/java/javase/downloads/index.htm
ADT (Android Development Tools) Plugin	(ADT) es un plugin para el IDE de Eclipse que está diseñado para darle un ambiente de gran alcance.	Se lo hace desde Eclipse: agregando repositorio: https://dl-ssl.google.com/android/eclipse/
Android SDK (Software Development Kit)	En el SDK podemos encontrar todas las librerías necesarias para poder trabajar con Android. Sea cualquier entorno, como es el caso de	http://developer.android.com/intl/es/sdk/index.html
Requerimientos para conexión a la web		
API Google Firebase Cloud Messaging	El API de Firebase Cloud Messaging es un servicio gratuito que te permite enviar notificaciones en tus aplicaciones o en tus páginas web.	Se necesita tener una clave para usar el API de Firebase Cloud Messaging se puede obtener: https://developers.google.com/console/help/?hl=es#generatingdevkeys
IIS (Internet Information Server 7)	Servidor web, conjunto de servicios para el S.O. M.Windows, convierte a una PC en un	Viene integrado en M. Windows, Hay activar características y escoger Internet Information Services en Panel de Control

Fuente: Cajilama, 2015

Realizado por: Muñoz, D. 2021

Una vez seleccionado el entorno de desarrollo para la aplicación, se muestra la interfaz del usuario final quién será el responsable de monitorear los datos obtenidos en el terreno a cielo abierto por medio de los sensores, la monitorización en estas condiciones no se podrá realizar las 24 horas del día, debido a varias adversidades ambientales y físicas que enfrenta el prototipo que se dan a conocer más adelante. En la Figura 12-2, se observa la interfaz que se presenta en el teléfono móvil.



Figura 24-2: Interfaz de usuario en un dispositivo móvil.

Realizado por: Muñoz, D. 2021.

Si el usuario necesita estar en constante monitorización se puede seleccionar el botón de suscribir para que las notificaciones se permitan comunicar al dispositivo cada minuto, teniendo la capacidad de estar pendiente de cualquier parámetro, sin embargo, en caso de no querer recibir las notificaciones cada instante, simplemente se puede dar click en cancelar.

2.5.1. *Android Studio*

Este entorno de desarrollo lanzado por Google nace para actualizar y reemplazar a Eclipse siendo uno de los IDE más utilizados en los últimos años. Este nuevo IDE tiene cambios avanzados en todo su sistema de construcción, una de sus herramientas más potentes es el elemento Smart Editing, que proporciona un código con mayor legibilidad. Otro elemento clave de Android Studio

es el nuevo sistema de construcción basado en Gradle, que a los desarrolladores servirá para aplicar distintas configuraciones dentro del mismo código, creando así diferentes versiones del mismo código o pudiendo utilizar códigos creados. Gradle proporciona una interfaz gráfica y de texto, pudiendo observar el diseño del entorno gráfico de la aplicación. Ahora el editor, tiene la capacidad de mostrar previamente la representación gráfica para distintas resoluciones, versiones de Android y características de georreferenciación (Hohensee, 2014, p.10).

Android studio posee servicios nuevos, y es una de las razones por la cual se utiliza en este prototipo, la compatibilidad y conexión mediante plataformas que poseen servicio de mensajería mediante Google Cloud Messaging el cual permitirá recibir y enviar mensajes de los servidores alojados en la nube. Además, posee un entorno de colaboración con Java IDE que permite que el ambiente sea amigable con el desarrollador, en este punto, se puede recalcar que el diseño y manejo en este entorno es de complejidad alta, debido a que congrega a varias herramientas como se menciona.

En la Figura 13-2, se puede observar la interfaz que esta desarrollada en Android Studio con sus diferentes ventanas auxiliares para su programación y visualización.

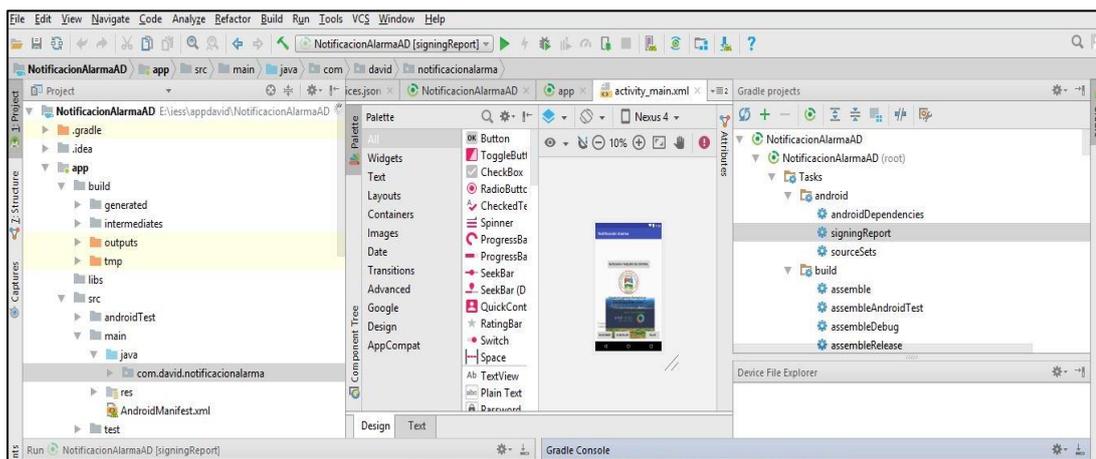


Figura 25-2: Interfaz de usuario.

Realizado por: Muñoz, D. 2021.

2.6. Otros dispositivos para implementación

Los dispositivos por presentar son de ayuda para lograr implementar el circuito con el que va a funcionar el prototipo.

2.6.1. Batería 9V

La idea de alimentar al prototipo por medio de una batería es la de evitar la instalación de tomacorrientes o en el caso, que no exista cerca al cultivo una fuente de energía eléctrica, como es este caso donde el prototipo funciona en un cultivo de papa de cielo abierto. Se puede

considerar una fuente secundaria y estas baterías son tomadas muy en cuenta por prototipos o aparatos que funcionan adecuadamente en diferencias de potencial de 5 a 9.5V.

Es adecuado informar que la batería puede ser utilizada en casos extremos debido a que el prototipo trabaja constantemente y se va a desgastar muy pronto, haciendo necesario que se remplace por una nueva, limitada por un máximo de dos horas.



Figura 26-2: Batería de 9V.

Fuente: Morocho y Quinapanta, 2016.

2.6.2. *Convertidor TTL*

Usado específicamente para alimentar a la tarjeta Nodemcu Esp32, debido que su alimentación máxima permite 3,3V, se usa un módulo convertidor TTL donde necesita ser alimentado por los dos voltajes a convertir aceptando solo señales en alto y bajo, no funciona con señales analógicas.

En la Figura 15-2, se presenta el convertidor utilizado.

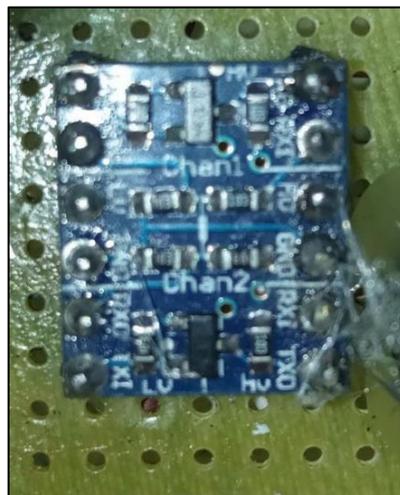


Figura 27-2: Convertidor TTL a 3,3V.

Realizado por: Muñoz, D. 2021.

2.6.3. Cajas de paso herméticas

El prototipo cuenta con dos cajas de paso protectoras de los módulos correspondientes, debido a que el prototipo toma datos en un cultivo de cielo abierto le permitirá cubrir los circuitos de las adversidades climáticas, sin embargo, no se podría garantizar la protección de sensores como temperatura debido a que debe estar constantemente en medición ambiente, es decir fuera de las protecciones de esta caja.

Algunas características de importancia se presentan a continuación:

- Realizados por material ignífugo ABS de alta calidad.
- Equipo hermético resistente al polvo y agua, con grado de protección IP65 (nivel 6 ante sólidos, nivel 5 ante líquidos)
- Fuertemente ignífugo y anticorrosivo.
- Rendimiento de sellado mediante tornillos del mismo material.
- No es tóxico.



Figura 28-2: Caja hermética para circuitos

Realizado por: Muñoz, D. 2021.

La caja principal será donde se procesan los datos, aquí se encuentra la tarjeta de desarrollo juntamente con su batería y convertidor a 3,3V y directamente conectada a la caja secundaria por medio de cable de red.

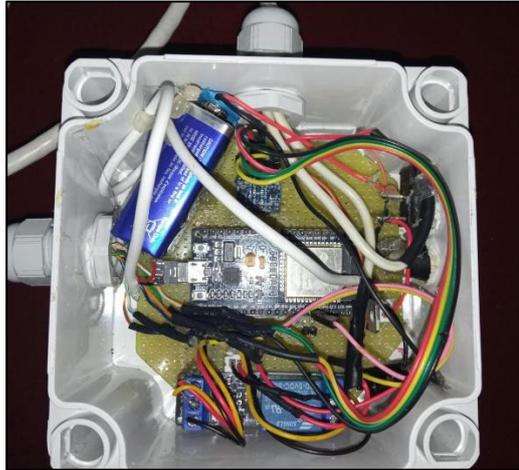


Figura 29-2: Caja hermética principal.

Realizado por: Muñoz, D. 2021.

La caja secundaria está conformada por los tres módulos de cada uno de los sensores, su tarea será la medición y recolección de los parámetros definidos para el cultivo de papa, gracias a los sensores. Cada una de las cajas tienen una medición de 10x10 cm.

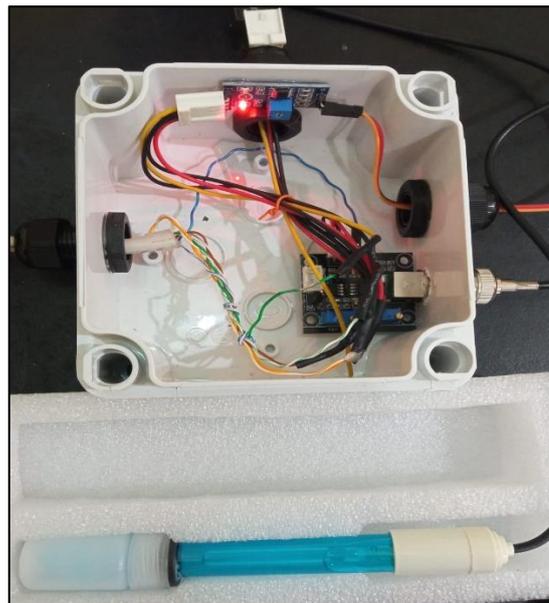


Figura 30-2: Caja hermética secundaria y sensores.

Realizado por: Muñoz, D. 2021.

2.7. Indicadores ambientales del cultivo

La papa posee indicadores que son de relevancia para su producción más aún en la etapa de producción de tubérculos, donde es aplicado, cabe destacar que el terreno tiene características adecuadas para considerar que su producto sea considerado de bajío, por lo que sus niveles

climáticos promedio como pH, temperatura ambiente y humedad han sido definidos como se muestra en la Tabla 18-2.

Tabla 20-2: Rangos teóricos adecuados para el cultivo de papa

Parámetros	Nivel mínimo	Nivel máximo
pH (práctica)	5.0	8.0
Temperatura ambiente óptima	10°	20°
Humedad	Su valor óptimo no debe disminuir el 50% ni sobrepasar el 80%	

Fuente: Franco, 2002

Realizado por: Muñoz, D. 2021

2.8. Esquema general del prototipo

Finalmente se presenta una esquematización de cómo se encuentra distribuido el prototipo con cada bloque tomado a consideración en su arquitectura general.

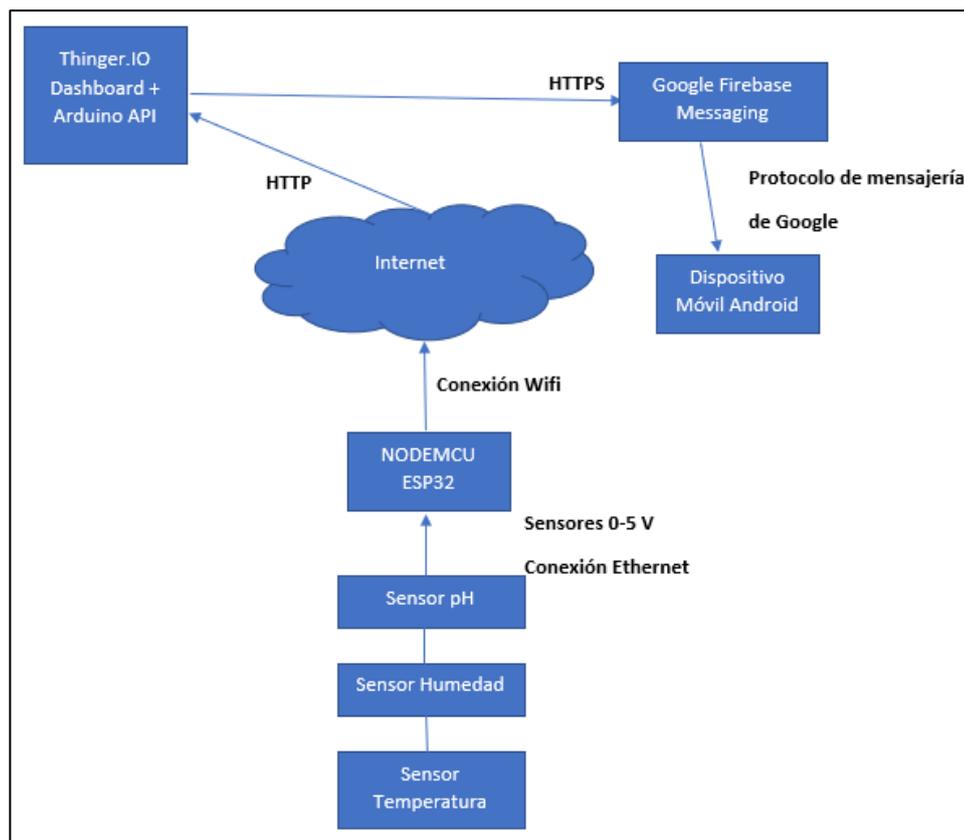


Figura 31-2: Esquema de trabajo.

Realizado por: Muñoz, D. 2021.

3. MARCO DE VALIDACION Y RESULTADOS

En el siguiente capítulo se evidencia el funcionamiento del prototipo en general, realizando pruebas de conexión, conectividad a servidores, transmisión de datos, etc. además, ser evaluado en diferentes ámbitos y terrenos con cultivos de papa de bajo, realizando una validación experimental y poder observar directamente los parámetros desde cualquier lugar que se disponga de una conexión a la red y en la recibiendo notificaciones en el dispositivo móvil que permite monitorizar al usuario final.

3.1. Conectividad entre servidores

Es un punto muy importante para el prototipo y en general para prototipos IoT, la conectividad que realicen en la nube todos los involucrados y que su comunicación sea compatible, ya que es el medio de transporte y envío de datos para que se pueda mostrar al usuario final.

3.1.1. Servidor Thinger y Firebase

Para evidenciar la manera en que ambos servidores pueden compartir la comunicación, se muestra una figura de la configuración en la que el servidor Firebase actúa como un end point, donde los dispositivos podrán llamar para realizar cualquier acción, así que será el punto de entrada hacia otro servicio como enviar un correo electrónico, enviar un SMS, llamar a una API, interactuar con IFFT, llamar a un dispositivo desde otra cuenta, o llamar a cualquier otro end poin (punto final) HTTP.

En este caso, la función clave es la de enviar mensajes desde el servidor Thinger a las aplicaciones cliente mediante Firebase Cloud Messaging. Cuando se usa el protocolo HTTP legacy, que es utilizado para aplicaciones Android, las solicitudes HTTP deberán dirigirse a la dirección <https://fcm.googleapis.com/fcm/send> como lo recomienda FCM y se observa en la Figura 1-3.



Figura 1-3: URL de conectividad hacia Firebase.

Realizado por: Muñoz, D. 2021.

La comunicación hasta el punto final será mediante peticiones HTTP, en la selección de la acción a cumplir por el servidor, necesariamente será POST que se encargará de enviar o postear los datos adquiridos apuntando a la dirección URL de FMC recomendada.

3.1.2. *Firebase (FCM) y app en Android Studio*

Este Para la comunicación de ambas partes, es necesario conocer un parámetro que es requisito para que se pueda hacer uso del servicio de mensajería y este es la clave SHA, que es una de las muchas funciones hash. Una función de este tipo hace referencia a una firma para textos o ficheros, la cual proporciona una huella digital de certificado de clave única con números hexadecimales. En este caso se está usando SHA de tipo 1, que se usa con el fin de crear un cliente tipo OAuth y una clave de API para la aplicación. El mismo servidor FMC proporciona una ID de la aplicación.



Figura 2-3: Certificado SHA para conectar a la aplicación.

Realizado por: Muñoz, D. 2021.

Para que esto sea válido se procede a verificar en el desarrollador de la aplicación (Android Studio) la firma digital de certificado que valide que es la correcta con los valores mostrados anteriormente:

01:f1:39:dc:69:ac:96:52:5f:1c:ba:fb:34:7d:1b:7d:ae:12:e3:cc

Se puede obtener accediendo a las ventanas de proyectos, mostrando las pestañas dentro de nuestro proyecto creado, y procediendo a acceder en las siguientes carpetas: Task, Android y Signing Report, donde se encontrará la firma digital específica para este proyecto, como se muestra en la Figura 3-3.

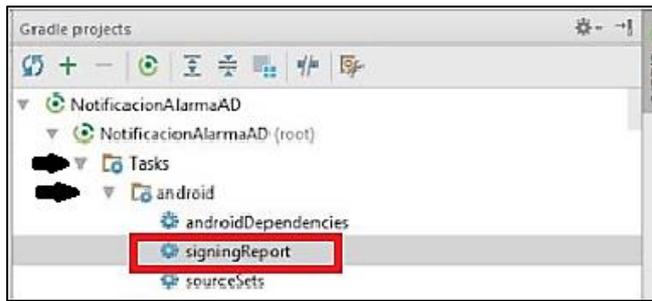


Figura 1-3: Clave SHA-1 proporcionado por la app en Android Studio

Realizado por: Muñoz, D. 2021.

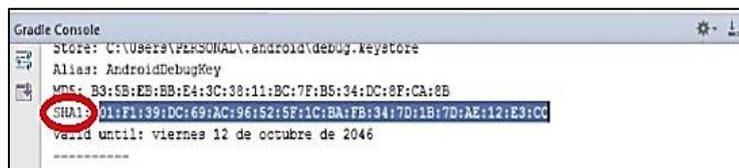


Figura 2-3: Verificación de clave SHA-1 en Android Studio.

Realizado por: Muñoz, D. 2021.

Con la conectividad adecuada el usuario puede hacer pruebas de conexión desde Thinger hasta el punto final, reflejándose directamente en el dispositivo móvil del usuario final, para poder realizarlo disponemos de una pestaña de prueba que facilita el mismo servidor, donde se puede observar los diferentes objetos en formato JSON y comprobar la respuesta a la llamada del end point informada por el servidor, en la parte de Headers.

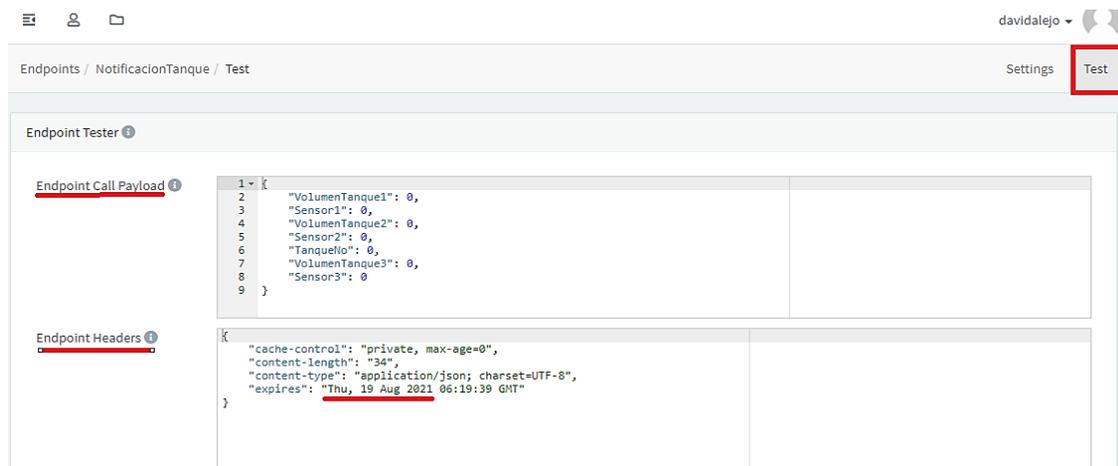


Figura 3-3: Indicadores para prueba de notificación a firebase.

Realizado por: Muñoz, D. 2021.

Se puede comprobar en ese instante, con la llegada de la notificación de mensajería enviado a la aplicación directamente como muestra la Figura 6-3.

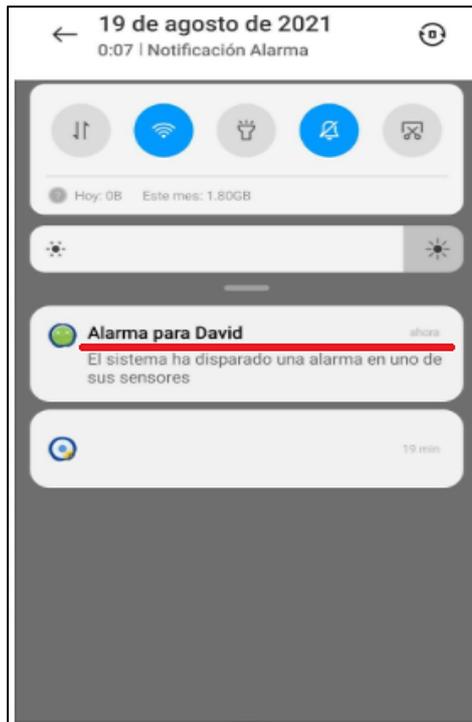


Figura 4-3: Verificación de notificaciones al teléfono móvil.

Realizado por: Muñoz, D. 2021.

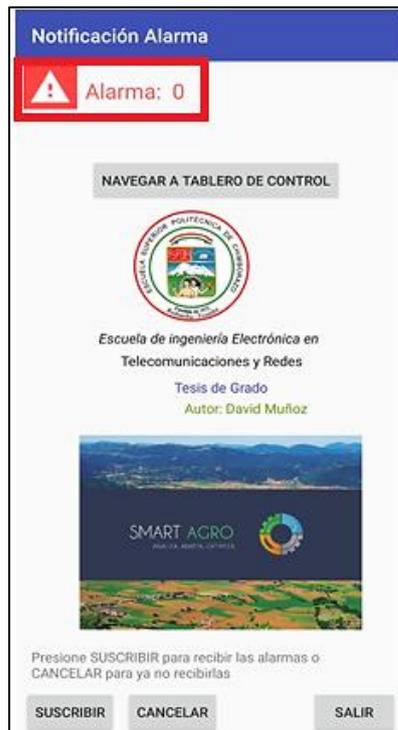


Figura 5-3: Verificación de alarmas a la aplicación.

Realizado por: Muñoz, D. 2021.

3.2. Pruebas de funcionamiento de sensores

Para comprobar el funcionamiento final de cada uno de los sensores, en este caso un sensor de temperatura, uno de temperatura y uno de pH, se debe evidenciar en el servidor Thinger cada una de sus mediciones a manera de comprobación, pero para que se pueda realizar esta prueba es necesario calibrar para que nos muestre en una ventana final, la medición real en el rango de valores que necesitamos. Los sensores adquiridos vienen previamente calibrados de fábrica, lo que resta por hacer es acoplar las señales con sus valores de voltaje para que se muestre en mediciones prácticas como por ejemplo la humedad que muestra sus valores en porcentaje.

Para cumplir con sus pruebas, se muestra el código realizado para que cada uno de ellos pueda medir dependiendo el rango de valores máximo y mínimo que tenga soporte, se muestra, además, el valor de referencia que pudo medir directamente los terminales del sensor.

```
h = analogRead(pin);  
float hum = map(h, 0, 4095, 100, 0);  
Serial.println("humedad es");
```

Figura 6-3: Valores rango de mapeo del sensor de humedad.

Realizado por: Muñoz, D. 2021.

Se puede observar los valores adimensionales que mide directamente el sensor de humedad y que serán acoplados en valores porcentuales en el código de programación de Arduino, en este caso el mínimo es 0 y el máximo 4095 que representa la medida directa en voltios, y se lo acopla a medidas de 0 a 100% de humedad, como se puede ver en la Figura 8-3.

Lo mismo sucede para el sensor de pH, en donde los valores de lectura directa serán acoplados a valores máximos que soporta el sensor, que como se vio en el capítulo anterior puede medir de 0 a 14, y el valor de medida máxima directa en voltios es 5095, como se muestra en el código de la Figura 9-3.

```
Serial.print(phVoltaje);  
phValor = map(phVoltaje, 0, 5095, 14, 0);  
Serial.print("el valor de ph");
```

Figura 7-3: Valores rango de mapeo del sensor de pH.

Realizado por: Muñoz, D. 2021.

Una vez acoplada las señales podemos verificar el funcionamiento con el rango de valores usados para este prototipo obteniendo así: valores de temperatura en grados centígrados en este caso el sensor DHT11 viene calibrado con estos valores, valor de humedad en porcentaje y valor de pH

de 0 a 14. Se puede observar en la Figura 10-3 las pizarras proporcionadas por el servidor para poder observar cada una de las mediciones.

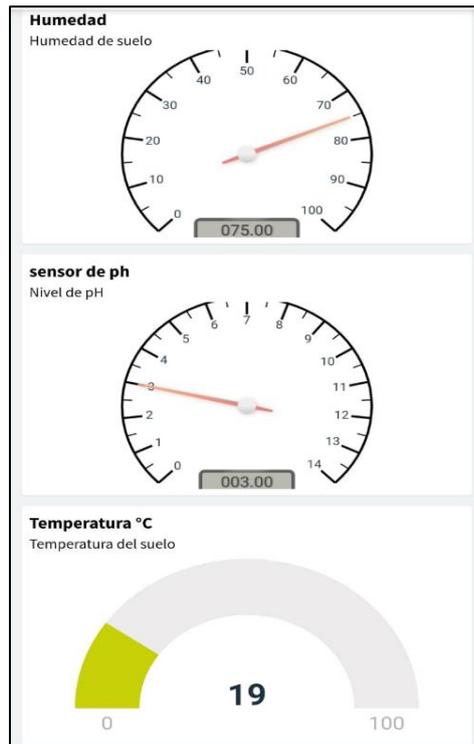


Figura 8-3: Ventana de visualización de parámetros.

Realizado por: Muñoz, D. 2021.

3.3. Pruebas de anclaje a la red mediante Esp32

Para que la comunicación sea posible hacia el internet, la tarjeta de desarrollo ESP32 debe conectarse mediante Wifi a la red, en este caso se lo hace mediante un dispositivo móvil que pueda funcionar como punto de acceso, debido a que como se informa en la introducción del capítulo dos los terrenos con cultivos de papa a evaluar se encuentran alejados de la ciudad y de difícil acceso a internet.

Para poder probar se tiene que configurar el código de programación de la tarjeta de desarrollo con los parámetros de SSID y el Password de la red que se vaya a utilizar como se muestra en la Figura 11-3.

```
#define SSID "Redmi 9 David"
#define SSID_PASSWORD "123456780"
ThingrESP32 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
```

Figura 9-3: Código para poder anclarse a una red.

Realizado por: Muñoz, D. 2021.

Una vez compilado se puede comprobar directamente en la red del dispositivo móvil que un dispositivo ya se encuentra conectado, en este caso la tarjeta de desarrollo.



Figura 10-3: Conexión de ESP32 a la red del dispositivo móvil.

Realizado por: Muñoz, D. 2021.



Figura 11-3: Concordancia Parámetros de la red del dispositivo móvil.

Realizado por: Muñoz, D. 2021.

3.4. Pruebas de envío y recepción de datos

Para que los datos proporcionados por los sensores y enviados a través de la tarjeta de desarrollo se comuniquen con el servidor es necesario verificar su conectividad, en este caso el dispositivo que entra en conexión con el servidor tiene el nombre de *dht111* que puede ser cualquiera, pudiendo observar su estado (connected / disconnected) para que se pueda recibir y enviar datos.

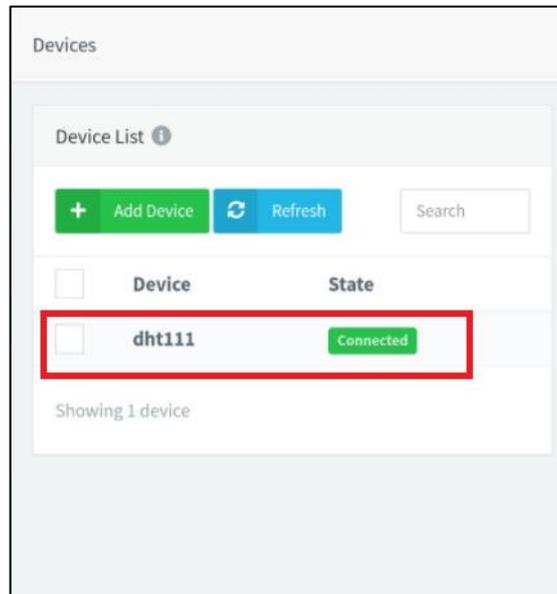


Figura 12-3: Estado del dispositivo en el servidor

Realizado por: Muñoz, D. 2021.

Dentro de la configuración del dispositivo disponemos de ventanas que nos ayudan a visualizar todos los parámetros de transmisión de datos como: *Transmisión, Recepción, dirección IP, estado del dispositivo, transmisión en vivo, y ubicación del dispositivo.*



Figura 13-3: Pestañas de parámetros en transmisión y recepción de datos.

Realizado por: Muñoz, D. 2021.



Figura 14-3: Transmisión en vivo de datos.

Realizado por: Muñoz, D. 2021.

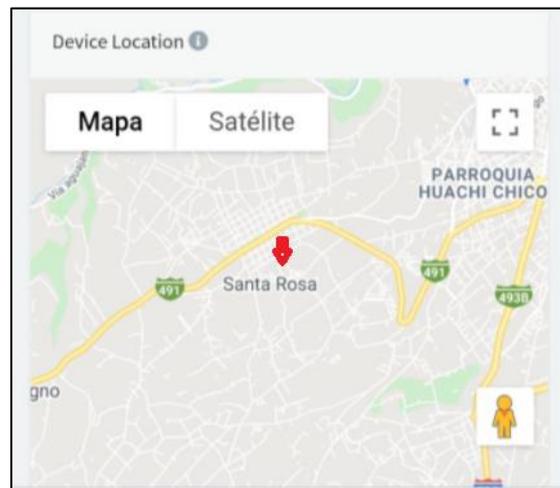


Figura 15-3: Pestaña de ubicación del dispositivo.

Realizado por: Muñoz, D. 2021.

3.5. Pruebas de notificaciones en la aplicación móvil

En este punto para poder evidenciar que las notificaciones estén configuradas correctamente, se tiene que enviar un valor por debajo del rango medio de cada sensor, para ello se tuvo que desconectar un pin del sensor de temperatura que es de menor complicación al momento de conectar, obteniendo así un valor de 0 voltios y observando la notificación directamente en el dispositivo móvil y también en la aplicación.



Figura 16-3: Verificación de alarma bajo del rango adecuado.

Realizado por: Muñoz, D. 2021.

3.6. Mediciones reales

El prototipo IoT de monitorización y control de parámetros para los cultivos de papa, se pone a prueba en varios escenarios donde los parámetros no se mantienen constantes, evidenciando la falta y existencia de control de cada uno de estos, por medio de abonos o agua ya sea de tipo potable o agua de regadío (acequia).

3.6.1. *Terreno con cultivo de papa sin control de parámetros*

Para este tipo de escenario se ha seleccionado un cultivo en donde la persona encarga o propietario de la siembra no posee los conocimientos ni cuidados que debe dar a la plantación, por ende, no tiene un sistema de riego, ni control de parámetros. La planta crece por medio de agua de lluvia y rara vez regada una vez en la semana a cualquier hora del día, sin abonos al suelo ni curaciones en la planta.

Estas mediciones fueron tomadas un día que existió lluvia en el lugar, por lo que la humedad es aceptable para el desarrollo de la plantación.

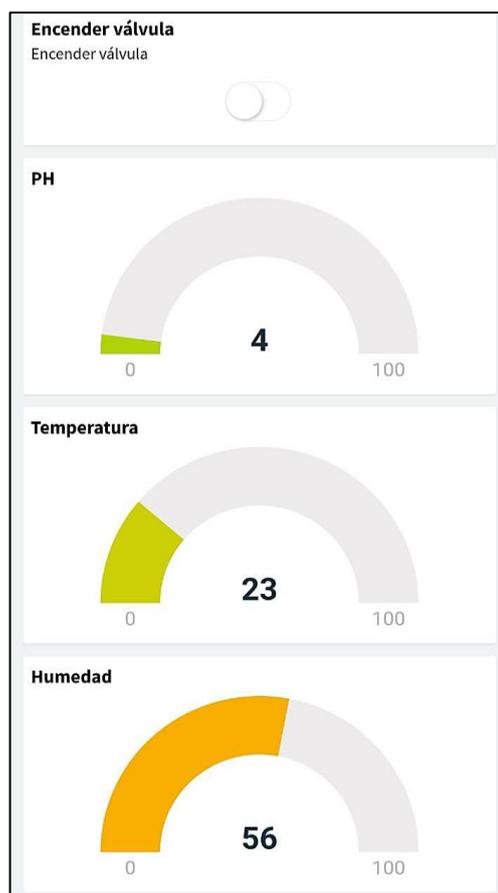


Figura 17-3: Monitorización de parámetros en cultivo de papa sin cuidados.

Realizado por: Muñoz, D. 2021.

En la siguiente tabla se puede observar los valores mínimo y máximo de temperatura, humedad y pH que se han podido tomar en este cultivo con las características mencionadas, pudiendo observar pequeñas variaciones tanto en temperatura como humedad.

Tabla 1-3: Mediciones en un terreno sin control de parámetros.

DATOS RECIBIDOS POR LOS SENSORES			
Sensor de temperatura			
Valor	Medición	Unidades Crudas del sensor	Alertas Generada
Máximo	19.60	19.600	Alarma Medio
Mínimo	18.79	18.799	Alarma Medio
Sensor de Humedad			
Máximo	74	3030	Sin Alerta
Mínimo	56	2293	Alarma Medio
Sensor de pH			
Máximo	6	2184	Alarma Bajo
Mínimo	4	1456	Alarma Bajo

Realizado por: Muñoz, D. 2021

De los datos se puede observar que tanto el pH como la humedad tendrá variaciones siempre y cuando se tenga presencia de agua, sea cual sea el tipo de esta, además la humedad tiene valores de variación de 18%, esto debido que no tiene un tipo de riego adecuado por lo que la acumulación de agua no permanecerá un buen tiempo como al utilizar un regadío por inundación o aspersión constante.



Figura 18-3: Prototipo en una plantación de papa sin monitorización de parámetros.

Realizado por: Muñoz, D. 2021.

Según los datos obtenidos en este tipo de terreno, se encontraría dentro del rango de pH del suelo que no es apto para un cultivo de papa, pese a que la plantación puede crecer e incluso desarrollar pequeños tubérculos, sufre daños e inclusive la muerte y retiro totalmente de esta plantación. En esta plantación en específico, luego de un par de semanas se pudo evidenciar que ya no podía crecer de manera adecuada, marchitándose, derrumbándose y sin mostrar rastro de nacimiento de flor pese a que su humedad no se encontraba por debajo del 50% que es mínimo y pocos valores por encima del 85% que es la referencia máxima.



Figura 19-3: Plantación de papa con daños y sin nacimiento de flor.

Realizado por: Muñoz, D. 2021.

Pese a que la tierra en la cual fue sembrado podría estar apta para sembrar este tubérculo, no significa que se tendrá la misma floración o crecimiento del tallo como se ha podido observar. La papa prefiere suelos con pH de 5,5 a 7,5 sin embargo, en la práctica se cultivan papas en suelos con pHs de 4,5 a 8,5 sin ningún inconveniente, además los pHs extremadamente altos se da por consecuencias de ciertos nutrientes más que todo el fósforo, por ello es recomendable tratar el suelo seis meses antes, ya que con la plantación es difícil bajar estos pHs (Yara.com).

3.6.2. Terreno con cultivo de papa a cargo de un profesional

Este terreno muestra una ubicación cercana al anterior, se encuentra en la misma parroquia y, por ende, la altura sobre el nivel del mar también es la misma, la diferencia radica en algo esencial, que demuestra ser tratado por una persona con conocimientos en el crecimiento adecuado de la papa, en este caso el propietario es un ingeniero agrónomo. El suelo y la plantación es tratado con ayuda de químicos como fungicidas, que ayudan al desarrollo adecuado y al cambio de pH del suelo a través del abono.

Se ha tomado mediciones en una parcela de terreno donde se encuentran alrededor de 8 plantas, que adquieren líquido vital a través de riego por inundación con agua de canal de riego o conocida comúnmente como agua de acequia, que se lo realiza una vez cada 15 días debido a la disponibilidad de agua de este tipo que es exclusiva para estos cultivos.

En la figura 22-3 se muestra los valores obtenidos presentados en el dashboard, que son de ayuda para una mejor visualización de los parámetros.



Figura 20-3: Monitorización en un terreno con cuidados.

Realizado por: Muñoz, D. 2021.

Se presenta también una tabla que muestra el valor máximo y mínimo alcanzado por cada uno de los sensores dentro de este terreno a evaluar obteniendo los resultados que se muestra en la tabla 2-3.

Tabla 2-3: Mediciones en un terreno a cargo de un profesional.

DATOS RECIBIDOS POR LOS SENSORES			
Sensor de temperatura			
Valor	Medición	Unidades Crudas del sensor	Alertas Generada
Máximo	22.20	1818	Sin Alerta
Mínimo	17.10	1400	Alarma Medio
Sensor de Humedad			
Máximo	80	3276	Sin Alerta
Mínimo	70	2866	Alarma Medio
Sensor de pH			
Máximo	7	2547	Sin alerta
Mínimo	4	1092	Alarma Bajo

Realizado por: Muñoz, D. 2021

Tanto en la medición que muestra las pizarras en el servidor, como los rangos de valores, muestra una clara diferencia entre el pH y la humedad del suelo, que son adecuados para que se lleve a cabo un correcto crecimiento del cultivo, esto se debe a que en el presente terreno se tiene los cuidados correctivos como abono del suelo, sistema de riego y agua por acequia, que como se puede observar influye mucho para tener un suelo ácido de pH con un valor mayor a 5.



Figura 21-3: Prototipo en un cultivo de papa con cuidados.

Realizado por: Muñoz, D. 2021.

3.6.3. *Regadío por inundación con uso de agua potable*

Para comprobar una diferencia experimental, se decidió enfrentar al cultivo de papa en un riego por inundación a través de agua potable que es la herramienta con la que varios terrenos considerados de bajío lo realizan siempre y cuando posean de este tipo de agua. A continuación, se muestra las mediciones tanto del dashboard como los rangos de mediciones.

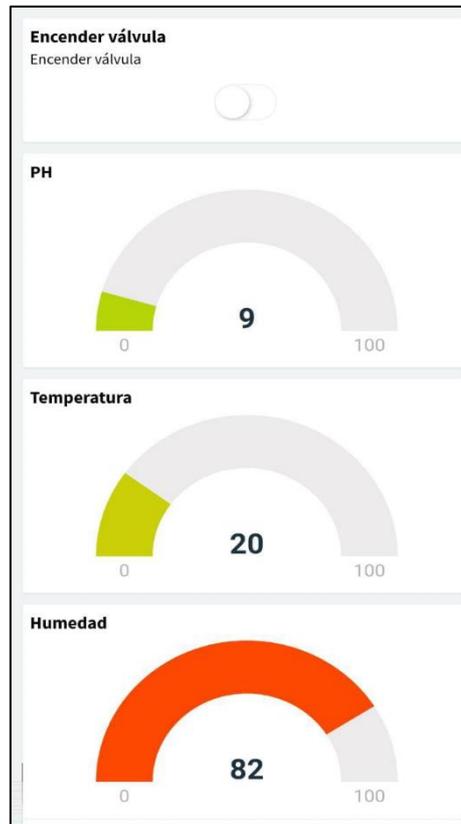


Figura 22-3: Medición de parámetros en riego con agua potable.

Realizado por: Muñoz, D. 2021.

Se puede diferenciar el cambio a gran escala por medio del sensor de pH, que en este caso aumenta a un valor de 9 o superior como se observa en la tabla 3-3, que muestra los rangos de valores tanto mínimo y máximos obtenidos. Además, se aclara que donde se realizó esta prueba en específico es en un suelo sin abono, es decir sin regular la acidez del suelo.

Tabla 3-3: Mediciones en un terreno con regado con agua potable.

DATOS RECIBIDOS POR LOS SENSORES			
Sensor de temperatura			
Valor	Medición	Unidades Crudas del sensor	Alertas Generada
Máximo	21.0	21.00	Sin Alerta
Mínimo	19.79	19.79	Alarma Medio
Sensor de Humedad			
Máximo	80	3276	Sin Alerta
Mínimo	70	2866	Alarma Medio
Sensor de pH			
Máximo	10	3639	Sin alerta
Mínimo	7	2547	Sin alerta

Realizado por: Muñoz, D. 2021

Se tiene diferencias de valor tanto en el valor mínimo de pH que en este caso es 7, como en el valor máximo que llega a tener valores de 10 debido al cloro que posee este tipo de agua, que si se mantuviera en este valor y con riego por inundación causaría que no se encuentre en un suelo óptimo para la plantación, quizás no cause lo peor que en este caso sería la perdida completa de la planta, pero si causase un retraso al crecimiento y aumento de los tubérculos.



Figura 23-3: Riego con agua potable usando el prototipo.

Realizado por: Muñoz, D. 2021.

3.6.4. Evaluación por quince días en terreno con cuidados

Para tener una medición diaria durante el tiempo de separación entre riego, se pone en funcionamiento durante los quince días en el terreno que se encuentra a cargo de un ingeniero agrónomo, para poder observar sus variaciones y si es factible el método de riego. Los valores

han sido escogidos en un rango de muestras diarias, anotando los valores máximos y mínimos, así como el valor promedio total de las mediciones realizadas.

Tabla 4-3: Mediciones parámetro de temperatura.

DATOS RECIBIDOS POR LOS SENSORES					
Día	Valores obtenidos	Valor mínimo	Valor máximo	Valor promedio	Presencia de lluvia
1	18,70 18,39 18,0 17,79 17,70 17,50 17,29 17,29	17,29	18,70	17,83	No
2	18,30 18,25 18,28 18,29 18,11 17,60 17,10 17,20	17,10	18,30	17,89	No
3	17,60 17,79 18,39 18,60 18,70 19,0 19,10 18,39	17,60	19,10	18,45	Leve
4	18,79 18,29 18,10 17,70 17,29 17,20 17,0 16,89	16,89	18,79	17,68	Si
5	16,10 16,29 16,5 16,89 17,0 17,0 16,70 16,70	16,10	17,0	16,65	No
6	16,10 16,10 16,10 15,80 15,69 15,80 15,80 15,89	15,69	16,10	15,91	No (Riego)
7	14,19 14,19 14,10 14,10 14,10 14,10 14,0 14,0	14,19	14,0	14,10	No

8	13,10 12,60 12,39 12,50 12,50 12,30 12,30 12,19	12,19	13,10	12,49	No
9	14,0 13,80 13,80 13,80 13,60 13,50 14,10 14,10	13,50	14,10	13,84	Leve
10	11,10 11,0 10,60 10,60 10,89 10,80 10,80 10,80	10,60	11,10	10,82	Fuerte
11	13,60 13,60 13,39 13,39 13,39 13,39 13,30 13,19	13,19	13,60	13,41	Leve
12	12,39 12,39 12,39 12,30 12,39 12,30 12,30 12,39	12,30	12,39	12,36	No
13	18,79 19,79 19,39 18,10 19,60 26,20 19,89 24,20	18,10	26,20	20,88	No (Riego)
14	13,80 13,30 13,10 13,10 13,10 13,0 12,60 12,50	12,50	13,80	13,06	No
15	13,40 13,40 13,40 14,10 14,10 14,50 13,90 13,90	13,40	14,50	13,84	Leve

Realizado por: Muñoz, D. 2021

Para el análisis se puede observar que debido a que el terreno es a cielo abierto, existen temperaturas que si sobre pasaron el máximo permitido ya que los rayos del sol van directamente hacia las plantaciones sin ningún material que las cubra, pese a que ha existido temperaturas sobre los 20°C en el día número 13, se tuvo que controlar el riego, activando la electroválvula y que pueda elevar su humedad manteniendo adecuado ese parámetro como se puede observar en la tabla 5-3, además se alcanzó temperaturas bajas registrando la temperatura promedio más baja en el día 10 con un valor de 10,83°C cerca del límite inferior, se dio debido a sus fuertes lluvias presentadas constantemente, y sin poder controlar este fenómeno, recalando que todas las mediciones se realizaron a horas de la tarde.

Tabla 5-3: Mediciones parámetro de humedad.

DATOS RECIBIDOS POR LOS SENSORES					
Día	Valores obtenidos	Valor mínimo	Valor máximo	Valor promedio	Presencia de lluvia
1	56,0 56,0 56,0 56,0 55,0 56,0 56,0	55,0	56,0	55,88	No
2	58,0 59,0 60,0 60,0 60,0 59,0 59,0 60,0	58,0	60,0	59,38	No
3	59,0 55,0 56,0 56,0 56,0 55,0 55,0	55,0	59,0	56,0	Leve
4	59,0 59,0 59,0 59,0 57,0 59,0 60,0 58,0	57,0	60,0	58,75	Si
5	70,0 67,0 68,0 70,0 64,0 70,0 68,0 68,0	64,0	70,0	68,13	No
6	59,0 60,0 59,0	58,0	60,0	58,88	No (Riego)

	59,0 59,0 58,0 58,0 59,0				
7	63,0 65,0 65,0 65,0 68,0 65,0 68,0	63,0	68,0	65,5	No
8	69,0 66,0 69,0 68,0 66,0 70,0 70,0 67,0	66,0	70,0	68,13	No
9	64,0 64,0 62,0 64,0 63,0 63,0 60,0 62,0	60,0	64,0	62,75	Leve
10	73,0 74,0 71,0 68,0 70,0 70,0 70,0 71,0	68,0	74,0	70,88	Fuerte
11	69,0 74,0 69,0 66,0 74,0 68,0 74,0 73,0	68,0	74,0	70,88	Leve
12	64,0 64,0 64,0 62,0 64,0 67,0 65,0 66,0	62,0	67,0	64,5	No
13	78,0 79,0 79,0 79,0 81,0 81,0 83,0 81,0	78,0	83,0	80,13	No (Riego)
14	65,0 69,0 64,0 68,0 66,0 65,0	64,0	69,0	66,13	No

	65,0 67,0				
15	62,0 62,0 64,0 59,0 60,0 60,0 62,0 64,0	59,0	64,0	61,63	Leve

Realizado por: Muñoz, D. 2021

Para este parámetro se puede observar que a medida que sube el volumen del agua así mismo lo hará su humedad al impactar con el suelo del terreno, por eso se puede observar que pese a ser objeto de lluvias fuertes la humedad se mantiene adecuada gracias al mismo suelo y al consumo de la plantación. Para los días de riego se tiene un aumento adecuado de humedad llegando en el día 13 hasta un promedio de 80,13% que fue medido después de unos minutos de haber sido regado, que es necesario por las altas temperaturas sabiendo que mientras pasan los días va a menorar el volumen del agua en el suelo, también se da el regadío cuando días anteriores los valores pueden estar cerca del límite que es por debajo del 55%.

Tabla 6-3: Mediciones parámetro de pH del suelo.

DATOS RECIBIDOS POR LOS SENSORES					
Día	Valores obtenidos	Valor mínimo	Valor máximo	Valor promedio	Presencia de lluvia
1	7,0 7,0 8,0 7,0 7,0 6,0 6,0 5,0	5,0	8,0	6,63	No
2	8,0 7,0 7,0 7,0 6,0 6,0 7,0 7,0	5,0	8,0	6,88	No
3	8,0 5,0 5,0 7,0 6,0 6,0 7,0 7,0	5,0	8,0	6,38	Leve
4	7,0 8,0 5,0 8,0 7,0 8,0 8,0 7,0	5,0	8,0	7,25	Si

5	9,0 9,0 8,0 9,0 9,0 7,0 8,0 8,0	7,0	9,0	8,38	No
6	8,0 7,0 8,0 7,0 7,0 7,0 8,0 5,0	5,0	8,0	7,13	No (Riego)
7	8,0 7,0 8,0 7,0 7,0 7,0 8,0 5,0	5,0	8,0	7,13	No
8	5,0 6,0 8,0 8,0 7,0 5,0 5,0 8,0	6,0	8,0	6,5	No
9	6,0 8,0 7,0 7,0 6,0 7,0 8,0 6,0	6,0	8,0	6,88	Leve
10	11,0 10,0 10,0 11,0 13,0 11,0 11,0 10,0	10,0	13,0	10,88	Fuerte
11	9,0 8,0 9,0 9,0 9,0 9,0 10,0	8,0	10,0	9,0	Leve
12	8,0 9,0 8,0 9,0 11,0 9,0 8,0 10,0	8,0	11,0	9,0	No
13	7,0 9,0	7,0	9,0	8,25	No (Riego)

	9,0 9,0 7,0 8,0 8,0 9,0				
14	9,0 9,0 9,0 9,0 8,0 8,0 8,0 9,0	8,0	9,0	8,62	No
15	9,0 10,0 9,0 9,0 8,0 8,0 8,0 9,0	8,0	10,0	8,75	Leve

Realizado por: Muñoz, D. 2021

En este parámetro se produce un fenómeno que causa impacto, ya que se puede observar que en el día 10 que anteriormente fue de análisis por su temperaturas bajas y su fuerte lluvia, ahora también se puede constatar que el nivel de pH se eleva mucho, llegando a un valor promedio de 10,88 que se encuentra por arriba del rango máximo, que puede ser causada por la mucha acumulación de agua constantemente, removiendo así nutrientes, abonos y más químicos con los cuales se trata a esta plantación, además como se puede ver en la práctica este suelo es apto para este cultivo manteniéndose en valores de pH dentro del rango de 4,5 a 8,5. Este valor cambiará también dependiendo la humedad del suelo ya que se necesita de un suelo mayormente húmedo para que el sensor pueda medir las partículas de iones de hidrógeno.

El riego realizado por las circunstancias ya descritas, se lo realizó con agua estancada (acequia) aunque dependiendo el sector se lo podría realizar con agua potable dependiendo el suelo ya que el pH que proporciona con agua potable no varía abruptamente en comparación con agua de acequia reservada, por lo que se podría usar de igual manera.

Luego de estos días se puede observar a simple vista (Figura 26-3) como ha ido creciendo el florecimiento de forma normal y por ende los tubérculos, en el espacio de terreno monitorizado, las plantaciones no muestran deformaciones, ni caída de tallos, lo único que pocas hojas se encuentran de diferente color a causa de que el sol llega directamente hasta ellas.



Figura 24-3: Florecimiento en espacio monitorizado.

Realizado por: Muñoz, D. 2021.



Figura 25-3: Mediciones en el terreno utilizado.

Realizado por: Muñoz, D. 2021.

4. PRESUPUESTO ECONÓMICO

4.1. Introducción

El presente prototipo conlleva a varios factores económicos que engloban el total no solo físico, sino también la parte lógica como programación y diseño de todos los sistemas tratados para llegar al final de su construcción.

4.2. Presupuesto físico

En base a la elaboración del prototipo, para llegar a su construcción final se necesitó de materiales como los que se presentan en la tabla 1-4.

Tabla 1-4: Presupuesto en materiales utilizados para su diseño y construcción.

Presupuesto físico del prototipo				
Ítem	Descripción	Cantidad	Precio Unitario	Valor Total
1	Modulo sensor de temperatura DHT11	1	10.00	10.00
2	Modulo sensor de humedad HL-69	1	18.00	18.00
3	Modulo sensor de pH del suelo SEN0161	2	50.00	100.00
4	Tarjeta Nodemcu ESP32	2	20.00	40.00
5	Convertor TTL a 3,3V	1	8.00	8.00
6	Cajas herméticas protectoras	2	4.00	8.00
7	Alimentación 12V 1.5A	1	10.00	10.00
8	Electroválvula Solenoide	1	20.00	20.00
9	Cables, conectores y demás accesorios para instalación en el campo		30.00	30.00
TOTAL				244.00
Imprevistos (5 %)				12.20
VALOR TOTAL				256.20

Realizado por: Muñoz, D. 2021.

El presupuesto utilizado en materiales para la elaboración, ensamblaje y la obtención del dispositivo final se encuentra en \$256.20 que no es el valor total, debido a que varios de los elementos no se encuentran disponibles en la ciudad, aplicando así valores de envío de cada uno de estos. Es un precio moderado pese a que los artículos que se encuentran con dos unidades se dan debido a que en la práctica se necesitó uno de repuesto porque no son muy resistentes, o como en el caso de la tarjeta de desarrollo es muy difícil encontrar un dispositivo nuevo que no contenga fallas de fábrica.

Además, el presupuesto final que se presenta en la tabla 2-4, será el incluido el diseño, la programación de la parte lógica y lo que conlleva a el acceso a la red por medio de datos móviles.

4.3. Presupuesto total

Tabla 2-4: Presupuesto total del prototipo.

Presupuesto total del prototipo			
Ítem	Descripción	Precio Unitario	Valor Total
1	Prototipo físico	250.95	256.20
2	Diseño	100.00	100.00
3	Programación lógica	200.00	200.00
4	Acceso a la red por datos móviles	10.00	10.00
5	Envíos de elementos a la ciudad	10.00	10.00
TOTAL			576.20
VALOR TOTAL			576.20

Realizado por: Muñoz, D. 2021

Debido a que es un dispositivo que trabaja con IoT específicamente en cultivos de papa en cielo abierto está diseñado para que no sufra daños sea por agua o tierra, y en el mercado aún no existe algún dispositivo similar que se use en el campo, así que pueden hacer uso de este presupuesto para futuras comparaciones en base a lo económico. Sin embargo, el precio por todo el prototipo incluyendo la parte lógica es razonable en comparación a prototipos de estudio que se han analizado en este mismo trabajo de titulación, por ejemplo, el prototipo para monitorización de tanques de Diesel en el Hospital Docente Ambato que trabaja con IoT y su presupuesto borda los 2000 dólares.

Por último, hay que tener en cuenta que la accesibilidad hacia el internet es una desventaja ya que en estos sectores no dan cobertura los proveedores de servicios de internet, por lo cual la única manera de hacerlo menos costoso y sencillo es poder activar la compatibilidad de acceso mediante un dispositivo móvil que navega a través de datos móviles.

CONCLUSIONES

- Mediante la revisión teórica y bibliográfica, se pudo analizar todas las características, y funcionamiento de la tecnología IoT, los requerimientos de cada sistema autónomo que lo conforma, en base a esta aplicación en cultivos de papa, así como análisis de los parámetros adecuados en forma general en el periodo de regadío de este cultivo.
- Se creó una aplicación para dispositivos móviles Android con una interfaz amigable y sencilla para el usuario final, en donde recibirá notificaciones de tipo alarma provenientes del servidor de mensajería Firebase, para el monitoreo de los tres parámetros importantes en un cultivo de papa como son temperatura, humedad y pH del suelo.
- En base a los requerimientos para el prototipo, se logró diseñar e implementar un dispositivo portátil que, trabaja con distintos bloques de procesamiento independientes que constituyen la tecnología Internet de las cosas, llegando a tener la capacidad de monitorear, adquirir y almacenar datos de temperatura, humedad, pH del suelo y controlar el regadío de agua en cualquier cultivo de papas, desde cualquier lugar que se encuentre con tan solo tener acceso a la red.
- Se diseñaron todas las diferentes configuraciones lógicas tanto en la tarjeta Nodemcu ESP32 como en los servidores Thingy y Firebase para que su comunicación pueda ser compatible sin ningún inconveniente y poder proporcionar todos los datos hasta la aplicación diseñada en el desarrollador de Android Studio.
- En base al método de evaluación constante realizado con el prototipo, se logró demostrar su funcionamiento y eficiencia del mismo, pudiendo constatar valores superiores al rango adecuado de florecimiento y crecimiento de tubérculos esto debido a que la plantación a cielo abierto se enfrenta a rayos directos del sol, lluvias fuertes e incluso pequeñas inundaciones que causan un cambio notorio al pH del suelo y a la humedad llegando a causar daños, caída de plantación o incluso pérdida completa de la planta, sin disponer de un monitoreo y control adecuado, como en un caso evidenciado en este trabajo de titulación.
- El prototipo creado es escalable y código abierto, por lo cual se puede aprovechar en trabajos futuros para que otros desarrolladores tengan la libertad de hacer nuevas modificaciones en los distintos sistemas utilizados y mejorar el funcionamiento en distintos ámbitos de la agricultura.
- A través de las pruebas realizadas en distintos entornos y riego, se evidencia que en un cultivo de este tipo se puede utilizar tanta agua de riego (acequia) o agua potable que, pese a que cuenta con pequeñas partículas de cloro, no proporciona una diferencia abrupta a cambios de pH por lo cual puede ser tomada en cuenta si se dispone de este tipo de líquido vital.

- En base al presupuesto, se pudo utilizar materiales que tengan el menor costo posible de acuerdo con los requerimientos y al lugar de aplicación que en este caso es la agricultura, llegando a obtener un costo de la estructura física de \$256.20 que ayuda a obtener un beneficio alto para personas que se dedican a esta cosecha, ayudando a que la siembra no se pierda y pase totalmente monitorizada el tiempo que se desee.

RECOMENDACIONES

- En cuanto a los sistemas independientes que conforman la tecnología IoT, para la adquisición de datos por medio de los sensores, se debe asegurar la transmisión inmediata por medio de cualquier comunicación sea alámbrica o inalámbrica teniendo en cuenta además los costos que deben ser lo menor posible y sobre todo el espacio en dimensión que ocupara el prototipo en el terreno, ya que en este tipo de cultivo también se trabaja la tierra y tendrá que ocupar un espacio lo más reducido que se pueda.
- La comunicación y conexión a la red mediante tecnología inalámbrica debe realizarse con la ayuda de tarjetas de desarrollo que sean diseñadas para prototipos IoT ya que tienen incorporado los módulos necesarios para realizar esta conexión como es el caso de Nodemcu que cuenta con un módulo Wifi incorporado.
- Para evitar la falta de accesibilidad a servicios como acceso a la red o energía eléctrica en estos sectores que existen cultivos de papa, se puede realizar un estudio con plantas sembradas en un área o parroquias cercanas a la ciudad que pueda tener cerca alguna red disponible para el acceso, envío y recepción de datos hasta la nube.
- En cuanto a la comunicación en la nube, se puede ahorrar económicamente y aprovechar servidores existentes que me permitan una compatibilidad de comunicación entre sí y con el desarrollador de la aplicación realizada para el usuario final, para evitar inconvenientes en envío y recepción de datos en tiempo real, con ello se evita la pérdida de información y alertas que llegan hasta el dispositivo móvil.
- En cuanto a los elementos electrónicos, en los más sensibles como el sensor de pH y tarjetas escasas como la ESP32 es preferible utilizar dos, una de pruebas y otra que funcione adecuadamente en el prototipo, ya que son muy propensas a que sufran daños.
- Para el análisis de un estudio a futuro con diseño e implementación de tipo Internet de las cosas en ámbitos de agricultura, se lo puede realizar en un invernadero ya que el control y monitoreo de parámetros en el cultivo se lo podrá realizar de mejor manera, evitando los fuertes cambios climáticos como se ha podido constatar (lluvia y temperaturas altas) y trabajando todo con el regadío proporcionado por el prototipo.

BIBLIOGRAFÍA

AKBAR, R. Sistema de bloqueo de vehículos de motor utilizando identificación de radiofrecuencia (RFID) y SIM ESP32 basado en Nodemcu [en línea] (tesis). Universidad islámica estatal de Sultan Sharif Kasim Riau, Facultad de ciencia y tecnología, Ingeniería eléctrica. Pekanbaru-Indonesia. 2020. pp. 33-36. [Consulta: 2021-06-11]. Disponible en: <http://repository.uin-suska.ac.id/30197/1/GABUNGAN%20KEQUALI%20BAB%20IV.pdf>

Al-Sarawi, S., Anbar, M., Alieyan, K., & Alzubaidi, M. "Internet of Things (IoT) communication protocols: Review". *2017 8th International Conference on Information Technology (ICIT)* [en línea], 2017, (Malasia). pp. 685-690. [Consulta: 02 agosto 2021]. Disponible en: doi: 10.1109/ICITECH.2017.8079928.

ALVARADO, S. El agua de riego y su influencia en el buen vivir de la comunidad Langualó grande, parroquia Mulaló, Cantón Latacunga, Provincia Cotopaxi [en línea] (tesis). Universidad Técnica de Ambato, Facultad de Ingeniería Civil y Mecánica, Ingeniería civil. Ambato-Ecuador. 2015. pp. 16-25. [Consulta: 2021-06-15]. Disponible en: <http://repositorio.uta.edu.ec/jspui/handle/123456789/10318>

ARELLANO PEDRAZZOLI, Edgar Gualberto, & TOMALA MIRANDA, Stalin Froilán. Implementación de un dispositivo que permita enrutar los datos adquiridos de una red ZigBee hacia la nube utilizando la minicomputadora Raspberry Pi y software de código abierto [en línea] (tesis). Escuela Superior Politécnica del Litoral, Facultad de Ingeniería en Electricidad y Computación. Guayaquil-Ecuador. 2015. pp. 14-15. [Consulta: 2021-07-01]. Disponible en: <http://www.dspace.espol.edu.ec/xmlui/bitstream/handle/123456789/30705/D-84580.pdf>

ASPIAZU, J. Aplicación de monitoreo cardiaco en Android con transmisión vía Bluetooth [en línea] (tesis). Escuela Superior Politécnica del Litoral, Facultad de ingeniería en Electricidad y computación. Guayaquil-Ecuador. 2015. pp.11-18. [Consulta: 2021-06-20]. Disponible en: <https://www.dspace.espol.edu.ec/retrieve/89038/D-84658.pdf>

Atlam, H., Walters, R., & Wills, G. "Fog Computing and the Internet of Things: A Review". *Big Data and Cognitive Computing* [en línea]. 2018, (Egypt) 2(2), p. 10. [Consulta: 06 junio 2021]. ISSN 2504-2289. Disponible en: <http://dx.doi.org/10.3390/bdcc2020010>

BALLADARES, V. Evaluación del contenido de antocianinas y micronutrientes (hierro, magnesio y calcio) en papas nativas (*Solanum andigena*) enteras y peladas de las variedades Puca Shungo, Yana Shungo y Yema de huevo (tesis). Universidad Técnica de Ambato, Facultad de ciencia e ingeniería en alimentos. Ambato-Ecuador. 2018. pp.03-23.

Barrera, Gabriel M. “Estilo arquitectónico para aplicaciones IoT”. *Econstor*, serie documentos de trabajo, n° 664 (2018), (Argentina) pp. 9-16.

Bustamante, L., Patricio, M. & Molina, J. “Thinger.io: An Open-Source Platform for Deploying Data Fusion Applications in IoT Environments”. *Sensors* [en línea], 2019, (España) 19(5), p. 1044. [Consulta: 28 julio 2021]. ISSN 1424-8220. Disponible en: <http://dx.doi.org/10.3390/s19051044>

CA TECHNOLOGIES. *Estrategia y arquitectura de API: Una estrategia coordinada* [en línea]. 2015. Sexta parte: Arquitectura de API, 17-18. [Consulta: 05 junio 2021]. Disponible en: http://www.idglat.com/afiliacion/whitepapers/453167_API%20Strategy%20and%20Architecture%20A%20Coordinated%20Approach-LAS.pdf?tk=/

Caja Hermética Conexiones [en línea]. Encaja mejor.com, 2021. [Consulta: 03 agosto 2021]. Disponible en: <https://www.encajamejor.com/caja-hermetica-conexiones/>

CAJILIMA, J. Desarrollo de una aplicación, para dispositivos móviles que permita administrar pedidos y controlar rutas de los vendedores, aplicada a la empresa: “Almacenes Juan Eljuri CÍA. LTDA. División perfumería” [en línea] (tesis). Universidad Politécnica Salesiana sede Cuenca, Ingeniería en sistemas. Cuenca-Ecuador. 2015. pp. 39-41. [Consulta: 2021-07-25]. Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/7951/1/UPS-CT004811.pdf>

Calapiña, J. Sistema Electrónico portable para la medición del índice de contaminación de suelos [en línea] (tesis). Universidad Técnica de Ambato. Facultad de ingeniería en sistemas Electrónica e Industrial, Ingeniería en Electrónica y comunicaciones. Ambato-Ecuador. 2018. pp. 60-65. [Consulta: 2021-07-10]. Disponible en: https://repositorio.uta.edu.ec/bitstream/123456789/28806/1/Tesis_%20t1483ec.pdf

Campoverde, A., Hernández R., L., & Mazón, B. “Cloud computing con herramientas open-source para Internet de las cosas” *Revista científica maskana* [en línea], 2015, (Ecuador) 6(especial), pp. 173-177. [Consulta: 08 junio 2021]. ISSN 2477-8893. Disponible en: <https://publicaciones.ucuenca.edu.ec/ojs/index.php/maskana/article/view/712/624>

Chartuni1, E., De Carvalho, F., Marçal, D., & Ruz, E. “Agricultura de precisión”. *Revista perspectivas* [en línea]. 2007, (Brasil), pp. 25-27. [Consulta: 16 mayo 2021]. Disponible en: <http://repiica.iica.int/docs/B0483e/B0483e.pdf>

Deschambault, O., Gherbi, A., & Légaré, C. “Efficient Implementation of the MQTT Protocol for Embedded Systems”. *Journal of information processing systems* [en línea], 2017, (Canada) 13(1), pp. 26-28. [Consulta: 29 junio 2021]. ISSN 2092-805X. Disponible en: <https://doi.org/10.3745/JIPS.04.0028>

E-Elektronic. Arduino Tutorial 20: Sensor de humedad de suelo [en línea]. 2015. [Consulta 2021-07-20]. Disponible en: <https://e-elektronic.com/arduino-tutorial-20-sensor-de-humedad-de-suelo/>

EZCARAY, I. Agricultura de precisión: elaboración de mapas de consumo y resbalamiento [en línea] (tesis). Universidad Pública de Navarra, Escuela Técnica Superior de ingenieros Agrónomos. Navarra-España. 2012. pp. 08-10. [Consulta: 2021-05-16]. Disponible en: <http://academica-e.unavarra.es/bitstream/handle/2454/6624/577934.pdf?sequence=1>

Farooq, M., Riaz, S., Abid, A., Abid, K., & Naeem, M. "A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming". *IEEE Access* [en línea], 2019, (Pakistan) 7, pp. 156237-156271. [Consulta: 28 junio 2021]. ISSN 2169-3536. Disponible en: doi: 10.1109/ACCESS.2019.2949703.

Fog computing: nuevo paradigma para las nubes del IoT [en línea]. España: IONOS by 1&1, 2019. [Consulta: 15 junio 2021]. Disponible en: <https://www.ionos.es/digitalguide/servidores/know-how/fog-computing/>

GitBook. *Arquitectura de una API REST*. 2015. [Consulta: 02 junio 2021]. Disponible en: <https://juanda.gitbooks.io/webapps/content/api/arquitectura-api-rest.html>

Gondchawar, N.; Kawitkar, S. "IoT based Smart Agriculture". *International Journal of Advanced Research in Computer and Communication Engineering* [en línea], 2016, (India) 5(6), pp. 838-841. [Consulta: 20 mayo 2021]. ISSN 2278-1021. Disponible en: <https://ijarccce.com/upload/2016/june-16/IJARCCCE%20188.pdf>

González, Fernando. "Tecnología". *Asociación Profesional de Cuerpos Superiores de Sistemas y Tecnologías de la Información de las Administraciones Públicas*, Boletín, n°44 (2007), (Barcelona) pp. 94-99.

Grigorik, Ilya. "Making the web faster with HTTP 2.0". *Commun. ACM*, vol. 56, n° 12 (2013), pp. 42-49.

Guerra Carmenate, José. *ESP32 Wifi y Bluetooth en un solo chip* [en línea]. 2021. [Consulta: 29 julio 2021]. Disponible en: <https://programarfacil.com/esp8266/esp32/>

Hohensee, B. *Introducción a Android Studio: Incluye proyectos reales y el código fuente* [en línea]. San Francisco: Babelcube Inc., 2014. [Consulta 01 agosto 2021]. Disponible en: <https://books.google.com.ec/books?hl=es&lr=&id=4dkuBQAAQBAJ&oi=fnd&pg=PT12&dq=android+studio&ots=YGJBBqSXzr&sig=eXi4-DKkbcdtzRF4i1YsTFyBFE0#v=onepage&q=android%20studio&f=false>

JURADO PÉREZ, Luis Alberto, VELASQUEZ VARGAS, Washington Adrián, & VINUEZA ESCOBAR, Nelson Fernando. Estado del Arte de las Arquitecturas de Internet de las Cosas (IoT) [en línea] (tesis). (Maestría) Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros de Telecomunicación. Madrid-España. 2014. pp. 06-07. [Consulta: 2021-05-15]. Disponible en: https://www.ecotec.edu.ec/material/material_2015B1_MAT130_11_43895.pdf

Laverde, J., Laverde, C. “Internet de las cosas aplicado en la agricultura ecuatoriana: Una propuesta para sistemas de riego”. *Dilemas contemp. educ. política valores* [en línea], 2021, (México) 8(2). [Consulta: 2021-06-17]. ISSN 2007-7890. Disponible en: <https://doi.org/10.46377/dilemas.v8i2.2542>.

LOACHAMÍN, R. Análisis de una red inalámbrica mallada autoconfigurable, utilizando el módulo Nodemcu ESP32 con el estándar 802.11 [en línea] (tesis). Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica. Quito-Ecuador. 2021. pp. 21-25. [Consulta: 2021-07-19]. Disponible en: <https://bibdigital.epn.edu.ec/bitstream/15000/21533/1/CD%2011027.pdf>

LÓPEZ LÓPEZ, Magaly Nataly, & QUIÑONEZ SINISTERRA, José Antonio. Diseño e implementación de un prototipo de una red inalámbrica con nodos inteligentes como ayuda para el manejo y selección de cultivos en cielo abierto [En línea] (tesis). Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería Electrónica en telecomunicaciones y redes. Riobamba-Ecuador. 2018. pp. 12-15. [Consulta: 2021-05-23]. Disponible en: <http://dspace.espoch.edu.ec/bitstream/123456789/9152/5/98T00205.pdf>

LOYOLA PINOS, María Augusta, & BECERRA DELGADO, Paúl Alonso. Manual para la aplicación de la tecnología ZigBee para edificios inteligentes en la ciudad de Cuenca [En línea] (tesis). Universidad Politécnica Salesiana sede Cuenca, Ingeniería electrónica. Cuenca-Ecuador. 2015. pp. 05-06. [Consulta: 2021-06-08]. Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/7986/1/UPS-CT004855.pdf>

MOROCHO SAILEMA, Vanessa Elena, & QUINAPANTA PÁEZ, Jonathan Ismael. Diseño e Implementación de un sistema multimodal que permita monitorear y controlar la recolección de desechos sólidos en tiempo real para Smart Cities [en línea] (tesis). Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería Electrónica, Telecomunicaciones y Redes. Riobamba-Ecuador. 2016. pp. 26-30. [Consulta: 2021-07-10]. Disponible en: <http://dspace.espoch.edu.ec/bitstream/123456789/6376/1/98T00108.pdf>

NACHO, R. Sistema de control domótico basado en Arduino, aplicación móvil y voz [en línea] (tesis). Universidad mayor de San Andrés, Facultad de Ciencias puras y Naturales. La Paz-

Bolivia. 2016. pp. 11-18. [Consulta: 2021-06-25]. Disponible en: <https://repositorio.umsa.bo/bitstream/handle/123456789/10674/T.3209.pdf?sequence=1>

Navas, Miguel. *¿Cuáles son los principales protocolos Wifi? Todo lo que necesitas saber* [blog]. 2017. [Consulta: 16 julio 2021]. Disponible en: <https://www.profesionalreview.com/2017/11/18/cuales-principales-protocolos-wifi/>

Pérez, M., Mendoza, M., & Suarez, M. “Paradigma IoT: desde su conceptualización hacia su aplicación en la agricultura”. *Revista Espacios* [en línea], 2019, (Venezuela) 40(18), pp. 02-07. [Consulta: 12 junio 2021]. ISSN 0798-1015. Disponible en: <https://www.revistaespacios.com/a19v40n18/a19v40n18p06.pdf>

PEREZ, E. Atención adecuada a los niños y niñas de 0 a5 años de los sectores periféricos de la parroquia de santa rosa del cantón Ambato (tesis). Universidad Técnica de Ambato, Facultad de ciencias humanas y de la Educación. Ambato-Ecuador. 2013. pp. 05-06.

Principios agronómicos en el cultivo de la papa [en línea]. Ecuador: Yara, 2021. [Consulta: 01 septiembre 2021]. Disponible en: <https://www.yara.com.ec/nutricion-vegetal/papa/principios-agronomicos-en-el-cultivo-de-la-papa/#:~:text=Las%20papas%20prefieren%20suelos%20de,0%20y%20de%20baja%20salinidad.>

PUNINA, E. Evaluación agronómica del cultivo de papa (*solanum tuberosum*) c.v. “Fripapa” a la aplicación de tres abonos completos [en línea] (tesis). Universidad Técnica de Ambato, Facultad de ciencias agropecuarias, Ingeniería agronómica. Ambato-Ecuador. 2013. pp. 07-08. [Consulta: 2021-06-10]. Disponible en: <https://repositorio.uta.edu.ec/bitstream/123456789/6532/1/Tesis69%20%20%20Ingenier%3%ADa%20Agron%3%B3mica%20-CD%20210.pdf>

Quiroga, E., Jaramillo, S., Campo, W., & Chanchí, G. “Propuesta de una Arquitectura para Agricultura de Precisión Soportada en IoT”. *Revista ibérica de Sistemas y Tecnologías de Información* [en línea], 2016, (Portugal) (24), pp. 41-43. [Consulta: 20 mayo 2021]. ISSN 1646-9895. Disponible en: <http://www.risti.xyz/issues/risti24.pdf>.

RFC 7252. *The Constrained Application Protocol (CoAP).*

RODAS, U. Diseño de una red inalámbrica de sensores para el monitoreo de una empresa en lima-perú [en línea] (tesis). Pontificia Universidad Católica del Perú, Facultad de ciencias de la Ingeniería, Ingeniería Electrónica. Lima-Perú. 2013. pp. 14-20. [Consulta: 29-05-2021]. Disponible en: https://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/5425/RODAS_URPI_DISE

%c3%91O_RED_INALAMBRICA_SENSORES_EMPRESA_PERU.pdf?sequence=1&isAllo
wed=y

Salazar, Jordi; & Silvestre, Santiago. *Internet de las cosas* [en línea]. Proyecto, 2017. [Consulta: 15 mayo 2021]. Disponible en: https://upcommons.upc.edu/bitstream/handle/2117/100921/LM08_R_ES.pdf

Sensor de temperatura y humedad relativa dht11 [en línea]. Perú: Naylamp Mechatronics, 2021. [Consulta: 02 agosto 2021]. Disponible en: <https://naylampmechatronics.com/sensores-temperatura-y-humedad/57-sensor-de-temperatura-y-humedad-relativa-dht11.html>

Sittón-Candanedo, I., Alonso, R., García, O., Muñoz, L., & Rodríguez-González, S. “Edge Computing, IoT and Social Computing in Smart Energy Scenarios”. *Sensors* [en línea]. 2019, (España) 19(15), p. 3353. [Consulta: 06 junio 2021]. ISSN 1424-8220. Disponible en: <http://dx.doi.org/10.3390/s19153353>

Sosinsky, B. *Cloud Computing Bible* [en línea]. Indiana-USA: Wiley Publishing, Inc., 2011. [Consulta: 06 junio 2021]. Disponible en: <https://arpitapatel.files.wordpress.com/2014/10/cloud-computing-bible1.pdf>

Válvulas Arco. *Electroválvulas: Qué es y para qué sirve* [blog]. 2020. [Consulta: 27 julio 2021]. Disponible en: <https://blog.valvulasarco.com>

VILAÑEZ, D. Implementación de un prototipo de sistema de seguridad doméstico basado en WPAN para una red IoT [en línea] (tesis). Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de ingeniería Electrónica, Telecomunicaciones y Redes. Riobamba-Ecuador. 2019. pp. 42-48. [Consulta: 2021-07-16]. Disponible en: <http://dspace.epoch.edu.ec/bitstream/123456789/13493/1/98T00262.pdf>

Wicaksono, M. “Implementasi Modul WIFI Nodemcu ESP8266 untuk smart home”. *Jurnal Teknik Komputer Unikom – Komputika*, vol. 6, n° 1 (2017), (Bandung) pp. 01-05.

ANEXOS

ANEXO A: DATASHEET ESP32 WROOM



ESP32-WROOM-32

Datasheet

NOT RECOMMENDED
FOR NEW DESIGNS
(NRND)



Version 3.2
Espressif Systems
Copyright © 2021



Contents

1	Overview	6
2	Pin Definitions	8
2.1	Pin Layout	8
2.2	Pin Description	8
2.3	Strapping Pins	9
3	Functional Description	12
3.1	CPU and Internal Memory	12
3.2	External Flash and SRAM	12
3.3	Crystal Oscillators	12
3.4	RTC and Low-Power Management	13
4	Peripherals and Sensors	14
5	Electrical Characteristics	15
5.1	Absolute Maximum Ratings	15
5.2	Recommended Operating Conditions	15
5.3	DC Characteristics (3.3 V, 25 °C)	15
5.4	Wi-Fi Radio	16
5.5	BLE Radio	17
	5.5.1 Receiver	17
	5.5.2 Transmitter	17
5.6	Reflow Profile	18
6	Schematics	19
7	Peripheral Schematics	20
8	Physical Dimensions	21
9	Recommended PCB Land Pattern	22
10	Learning Resources	23
10.1	Must-Read Documents	23
10.2	Must-Have Resources	23
	Revision History	25

[Not Recommended For New Designs \(NRND\)](#)

1 Overview

ESP32-WROOM-32 is a powerful, generic Wi-Fi+BT+BLE MCU module that targets a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as voice encoding, music streaming and MP3 decoding.

At the core of this module is the ESP32-D0WDQ6 chip*. The chip embedded is designed to be scalable and adaptive. There are two CPU cores that can be individually controlled, and the CPU clock frequency is adjustable from 80 MHz to 240 MHz. The chip also has a low-power co-processor that can be used instead of the CPU to save power while performing tasks that do not require much computing power, such as monitoring of peripherals. ESP32 integrates a rich set of peripherals, ranging from capacitive touch sensors, Hall sensors, SD card interface, Ethernet, high-speed SPI, UART, PS and PC.

Note:

* For details on the part numbers of the ESP32 family of chips, please refer to the document [ESP32 Datasheet](#).

The integration of Bluetooth®, Bluetooth LE and Wi-Fi ensures that a wide range of applications can be targeted, and that the module is all-around: using Wi-Fi allows a large physical range and direct connection to the Internet through a Wi-Fi router, while using Bluetooth allows the user to conveniently connect to the phone or broadcast low energy beacons for its detection. The sleep current of the ESP32 chip is less than 5 μ A, making it suitable for battery powered and wearable electronics applications. The module supports a data rate of up to 150 Mbps, and 20 dBm output power at the antenna to ensure the widest physical range. As such the module does offer industry-leading specifications and the best performance for electronic integration, range, power consumption, and connectivity.

The operating system chosen for ESP32 is freeRTOS with LwIP; TLS 1.2 with hardware acceleration is built in as well. Secure (encrypted) over the air (OTA) upgrade is also supported, so that users can upgrade their products even after their release, at minimum cost and effort.

Table 1 provides the specifications of ESP32-WROOM-32.

Table 1: ESP32-WROOM-32 Specifications

Categories	Items	Specifications
Certification	RF certification	FCC/CE-RED/IC/TELEC/KCC/SRRC/NCC
	Wi-Fi certification	Wi-Fi Alliance
	Bluetooth certification	BCB
	Green certification	RoHS/REACH
Test	Reliability	HTOL/HTSL/HAST/TCT/ESD
Wi-Fi	Protocols	802.11 b/g/n (802.11n up to 150 Mbps) A-MPDU and A-MSDU aggregation and 0.4 μ s guard interval support
	Frequency range	2.4 GHz ~ 2.5 GHz
Bluetooth	Protocols	Bluetooth v4.2 BR/EDR and BLE specification
	Radio	NZF receiver with -97 dBm sensitivity
		Class-1, class-2 and class-3 transmitter
		AFH
Audio	CVSD and SBC	

[Not Recommended For New Designs \(NRND\)](#)

Categories	Items	Specifications
Hardware	Module interfaces	SD card, UART, SPI, SDIO, PC, LED PWM, Motor PWM, PS, IR, pulse counter, GPIO, capacitive touch sensor, ADC, DAC, Two-Wire Automotive Interface (TWA [®]), compatible with ISO11828-1 (CAN Specification 2.0)
	On-chip sensor	Hall sensor
	Integrated crystal	40 MHz crystal
	Integrated SPI flash	4 MB
	Operating voltage/Power supply	3.0 V ~ 3.6 V
	Operating current	Average: 80 mA
	Minimum current delivered by power supply	500 mA
	Recommended operating temperature range	-40 °C ~ +85 °C
	Package size	(18.00±0.10) mm × (25.50±0.10) mm × (3.10±0.10) mm
	Moisture sensitivity level (MSL)	Level 3

2 Pin Definitions

2.1 Pin Layout

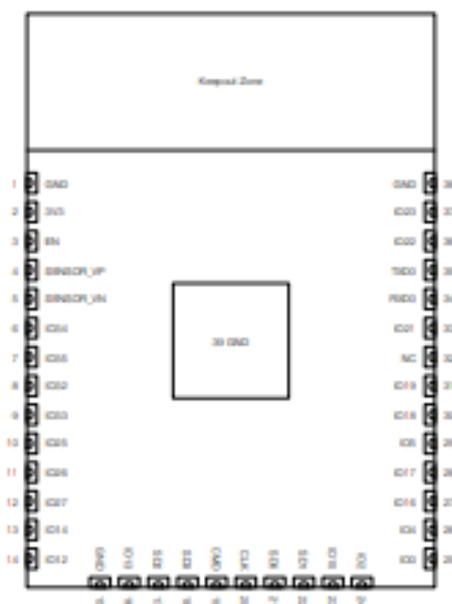


Figure 1: ESP32-WROOM-32 Pin Layout (Top View)

2.2 Pin Description

ESP32-WROOM-32 has 38 pins. See pin definitions in Table 2.

Table 2: Pin Definitions

Name	No.	Type	Function
GND	1	P	Ground
3V3	2	P	Power supply
EN	3	I	Module-enable signal. Active high.
SENSOR_VP	4	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, ADC1_CH3, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5
IO32	8	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8

[Not Recommended For New Designs \(NRND\)](#)

Name	No.	Type	Function
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
IO14	13	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12	14	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIO, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	P	Ground
IO13	16	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
SHD/SD2*	17	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SWP/SD3*	18	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SCS/CMD*	19	I/O	GPIO11, SD_CMD, SPICSS, HS1_CMD, U1RTS
SCK/CLK*	20	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SD0/SD0*	21	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SD1/SD1*	22	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
IO15	23	I/O	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICSS, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPWR, HS2_DATA0, SD_DATA0
IO0	25	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
IO4	26	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
IO16	27	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
IO17	28	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
IO5	29	I/O	GPIO5, VSPICSS, HS1_DATA6, EMAC_RX_CLK
IO18	30	I/O	GPIO18, VSPICLK, HS1_DATA7
IO19	31	I/O	GPIO19, VSPIO, U0CTS, EMAC_TXD0
NC	32	-	-
IO21	33	I/O	GPIO21, VSPHD, EMAC_TX_EN
RXD0	34	I/O	GPIO3, U0RXD, CLK_OUT2
TXD0	35	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	I/O	GPIO22, VSPWR, U0RTS, EMAC_TXD1
IO23	37	I/O	GPIO23, VSPID, HS1_STROBE
GND	38	P	Ground

Notice:

* Pins SCK/CLK, SD0/SD0, SD1/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on the module and are not recommended for other uses.

2.3 Strapping Pins

ESP32 has five strapping pins, which can be seen in Chapter 6 Schematics.

[Not Recommended For New Designs \(NRND\)](#)

- MTDI
- GPIO0
- GPIO2
- MTDO
- GPIO5

Software can read the values of these five bits from register "GPIO_STRAPPING".

During the chip's system reset release (power-on-reset, RTC watchdog reset and brownout reset), the latches of the strapping pins sample the voltage level as strapping bits of "0" or "1", and hold these bits until the chip is powered down or shut down. The strapping bits configure the device's boot mode, the operating voltage of VDD_SDIO and other initial system settings.

Each strapping pin is connected to its internal pull-up/pull-down during the chip reset. Consequently, if a strapping pin is unconnected or the connected external circuit is high-impedance, the internal weak pull-up/pull-down will determine the default input level of the strapping pins.

To change the strapping bit values, users can apply the external pull-down/pull-up resistances, or use the host MCU's GPIOs to control the voltage level of these pins when powering on ESP32.

After reset release, the strapping pins work as normal-function pins.

Refer to Table 3 for a detailed boot-mode configuration by strapping pins.

Table 3: Strapping Pins

Voltage of Internal LDO (VDD_SDIO)					
Pin	Default	3.3 V		1.8 V	
MTDI	Pull-down	0		1	
Booting Mode					
Pin	Default	SPI Boot		Download Boot	
GPIO0	Pull-up	1		0	
GPIO2	Pull-down	Don't-care		0	
Enabling/Disabling Debugging Log Print over U0TXD During Booting					
Pin	Default	U0TXD Active		U0TXD Silent	
MTDO	Pull-up	1		0	
Timing of SDIO Slave					
Pin	Default	FE Sampling FE Output	FE Sampling RE Output	RE Sampling FE Output	RE Sampling RE Output
MTDO	Pull-up	0	0	1	1
GPIO5	Pull-up	0	1	0	1

Note:

- Firmware can configure register bits to change the settings of "Voltage of Internal LDO (VDD_SDIO)" and "Timing of SDIO Slave" after booting.
- The module integrates a 3.3 V SPI flash, so the pin MTDI cannot be set to 1 when the module is powered up.

The strapping pins need a setup and hold time before and after the EN signal goes high. For details please refer

[Not Recommended For New Designs \(NRND\)](#)

3 Functional Description

This chapter describes the modules and functions integrated in ESP32-WROOM-32.

3.1 CPU and Internal Memory

ESP32-D0WDQ6 contains two low-power Xtensa® 32-bit LX6 microprocessors. The internal memory includes:

- 448 KB of ROM for booting and core functions.
- 520 KB of on-chip SRAM for data and instructions.
- 8 KB of SRAM in RTC, which is called RTC FAST Memory and can be used for data storage; it is accessed by the main CPU during RTC Boot from the Deep-sleep mode.
- 8 KB of SRAM in RTC, which is called RTC SLOW Memory and can be accessed by the co-processor during the Deep-sleep mode.
- 1 Kbit of eFuse: 256 bits are used for the system (MAC address and chip configuration) and the remaining 768 bits are reserved for customer applications, including flash-encryption and chip-ID.

3.2 External Flash and SRAM

ESP32 supports multiple external QSPI flash and SRAM chips. More details can be found in Chapter SPI in the [ESP32 Technical Reference Manual](#). ESP32 also supports hardware encryption/decryption based on AES to protect developers' programs and data in flash.

ESP32 can access the external QSPI flash and SRAM through high-speed caches.

- The external flash can be mapped into CPU instruction memory space and read-only memory space simultaneously.
 - When external flash is mapped into CPU instruction memory space, up to 11 MB + 248 KB can be mapped at a time. Note that if more than 3 MB + 248 KB are mapped, cache performance will be reduced due to speculative reads by the CPU.
 - When external flash is mapped into read-only data memory space, up to 4 MB can be mapped at a time. 8-bit, 16-bit and 32-bit reads are supported.
- External SRAM can be mapped into CPU data memory space. Up to 4 MB can be mapped at a time. 8-bit, 16-bit and 32-bit reads and writes are supported.

ESP32-WROOM-32 integrates a 4 MB SPI flash, which is connected to GPIO6, GPIO7, GPIO8, GPIO9, GPIO10 and GPIO11. These six pins cannot be used as regular GPIOs.

3.3 Crystal Oscillators

The module uses a 40-MHz crystal oscillator.

3.4 RTC and Low Power Management

With the use of advanced power-management technologies, ESP32 can switch between different power modes.

For details on ESP32's power consumption in different power modes, please refer to section "RTC and Low-Power Management" in [ESP32 Datasheet](#).

5 Electrical Characteristics

5.1 Absolute Maximum Ratings

stresses beyond the absolute maximum ratings listed in table 4 below may cause permanent damage to the device. These are stress ratings only, and do not refer to the functional operation of the device that should follow the recommended operating conditions.

Table 4: Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit
VDD33	Power supply voltage	-0.3	3.6	V
I_{load}^1	Cumulative IO output current	-	1,100	mA
T _{store}	Storage temperature	-40	150	°C

- The module worked properly after a 24-hour test in ambient temperature at 25 °C, and the ICs in three domains (VDD3P3_RTC, VDD3P3_CPU, VDD_SDIO) output high logic level to ground. Please note that pins occupied by flash and/or PSRAM in the VDD_SDIO power domain were excluded from the test.
- Please see Appendix IO_MUX of [ESP32 Datasheet](#) for IO's power domain.

5.2 Recommended Operating Conditions

Table 5: Recommended Operating Conditions

Symbol	Parameter	Min	Typical	Max	Unit
VDD33	Power supply voltage	3.0	3.3	3.6	V
I _{VDD}	Current delivered by external power supply	0.5	-	-	A
T	Operating temperature	-40	-	85	°C

5.3 DC Characteristics (3.3 V, 25 °C)

Table 6: DC Characteristics (3.3 V, 25 °C)

Symbol	Parameter	Min	Typ	Max	Unit	
C _{IN}	Pin capacitance	-	2	-	pF	
V _{IH}	High-level input voltage	0.75×VDD ³	-	VDD ³ +0.3	V	
V _{IL}	Low-level input voltage	-0.3	-	0.25×VDD ³	V	
I _{IH}	High-level input current	-	-	50	nA	
I _{IL}	Low-level input current	-	-	50	nA	
V _{OIH}	High-level output voltage	0.8×VDD ²	-	-	V	
V _{OOL}	Low-level output voltage	-	-	0.1×VDD ¹	V	
I _{OH}	High-level source current (VDD ¹ = 3.3 V, V _{OIH} >= 2.64 V, output drive strength set to the maximum)	VDD3P3_CPU power domain ^{1, 2}	-	40	-	mA
		VDD3P3_RTC power domain ^{1, 2}	-	40	-	mA
		VDD_SDIO power domain ^{1, 2}	-	20	-	mA

Symbol	Parameter	Min	Typ	Max	Unit
I_{OL}	Low-level sink current (VDD ¹ = 3.3 V, V _{OL} = 0.495 V, output drive strength set to the maximum)	-	28	-	mA
R _{PU}	Resistance of internal pull-up resistor	-	45	-	kΩ
R _{PD}	Resistance of internal pull-down resistor	-	46	-	kΩ
V _{IL,NOT}	Low-level input voltage of CHIP_PU to power off the chip	-	-	0.6	V

Notes:

1. Please see Appendix IO_MUX of [ESP32 Datasheet](#) for IO's power domain. VDD is the I/O voltage for a particular power domain of pins.
2. For VDD3P3_CPU and VDD3P3_RTC power domain, pin-pin current sourced in the same domain is gradually reduced from around 40 mA to around 29 mA, V_{OL} >= 2.84 V, as the number of current-source pins increases.
3. Pins occupied by flash and/or PSRAM in the VDD_SDIO power domain were excluded from the test.

5.4 Wi-Fi Radio

Table 7: Wi-Fi Radio Characteristics

Parameter	Condition	Min	Typical	Max	Unit
Operating frequency range ^{note 1}	-	2412	-	2484	MHz
Output impedance ^{note 2}	-	-	note 2	-	Ω
TX power ^{note 3}	11n, MCS7	12	13	14	dBm
	11b mode	17.5	18.5	20	dBm
Sensitivity	11b, 1 Mbps	-	-98	-	dBm
	11b, 11 Mbps	-	-89	-	dBm
	11g, 6 Mbps	-	-92	-	dBm
	11g, 54 Mbps	-	-74	-	dBm
	11n, HT20, MCS0	-	-91	-	dBm
	11n, HT20, MCS7	-	-71	-	dBm
	11n, HT40, MCS0	-	-89	-	dBm
	11n, HT40, MCS7	-	-69	-	dBm
Adjacent channel rejection	11g, 6 Mbps	-	31	-	dB
	11g, 54 Mbps	-	14	-	dB
	11n, HT20, MCS0	-	31	-	dB
	11n, HT20, MCS7	-	13	-	dB

1. Device should operate in the frequency range allocated by regional regulatory authorities. Target operating frequency range is configurable by software.
2. For the modules that use IPEX antennas, the output impedance is 50 Ω. For other modules without IPEX antennas, users do not need to concern about the output impedance.
3. Target TX power is configurable based on device or certification requirements.

5.5 BLE Radio

5.5.1 Receiver

Table 8: Receiver Characteristics – BLE

Parameter	Conditions	Min	Typ	Max	Unit
Sensitivity @30.8% PER	-	-	-97	-	dBm
Maximum received signal @30.8% PER	-	0	-	-	dBm
Co-channel CI	-	-	+10	-	dB
Adjacent channel selectivity CI	$F = F_0 + 1 \text{ MHz}$	-	-5	-	dB
	$F = F_0 - 1 \text{ MHz}$	-	-5	-	dB
	$F = F_0 + 2 \text{ MHz}$	-	-25	-	dB
	$F = F_0 - 2 \text{ MHz}$	-	-35	-	dB
	$F = F_0 + 3 \text{ MHz}$	-	-25	-	dB
	$F = F_0 - 3 \text{ MHz}$	-	-45	-	dB
Out-of-band blocking performance	30 MHz ~ 2000 MHz	-10	-	-	dBm
	2000 MHz ~ 2400 MHz	-27	-	-	dBm
	2500 MHz ~ 3000 MHz	-27	-	-	dBm
	3000 MHz ~ 12.5 GHz	-10	-	-	dBm
Intermodulation	-	-36	-	-	dBm

5.5.2 Transmitter

Table 9: Transmitter Characteristics – BLE

Parameter	Conditions	Min	Typ	Max	Unit
RF transmit power	-	-	0	-	dBm
Gain control step	-	-	3	-	dBm
RF power control range	-	-12	-	+9	dBm
Adjacent channel transmit power	$F = F_0 \pm 2 \text{ MHz}$	-	-52	-	dBm
	$F = F_0 \pm 3 \text{ MHz}$	-	-58	-	dBm
	$F = F_0 \pm > 3 \text{ MHz}$	-	-60	-	dBm
$\Delta f_{1\text{avg}}$	-	-	-	265	kHz
$\Delta f_{2\text{max}}$	-	247	-	-	kHz
$\Delta f_{2\text{avg}}/\Delta f_{1\text{avg}}$	-	-	-0.92	-	-
ICFT	-	-	-10	-	kHz
Drift rate	-	-	0.7	-	kHz/50 μ s
Drift	-	-	2	-	kHz

5.6 Reflow Profile

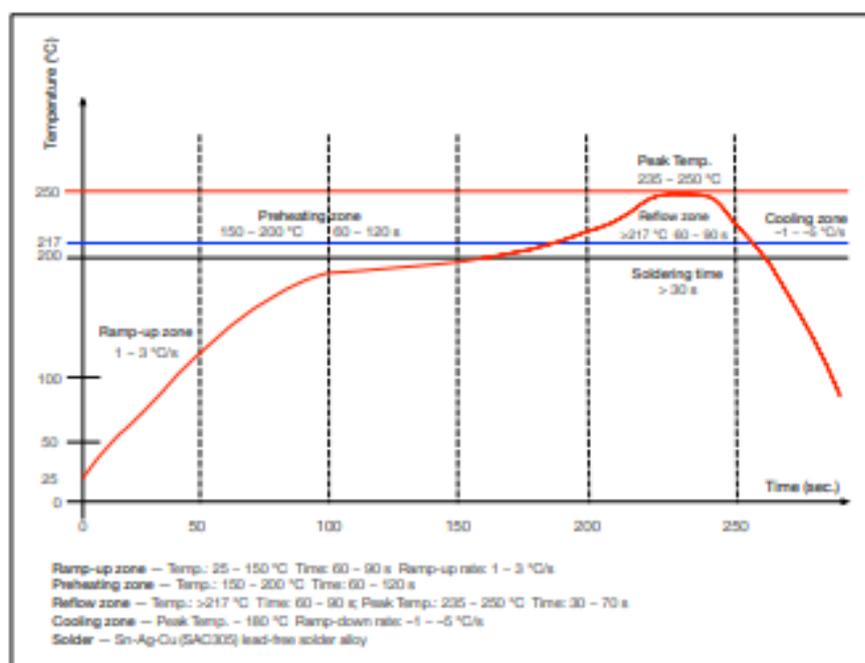


Figure 2: Reflow Profile

Note:

Solder the module in a single reflow.

6 Schematics

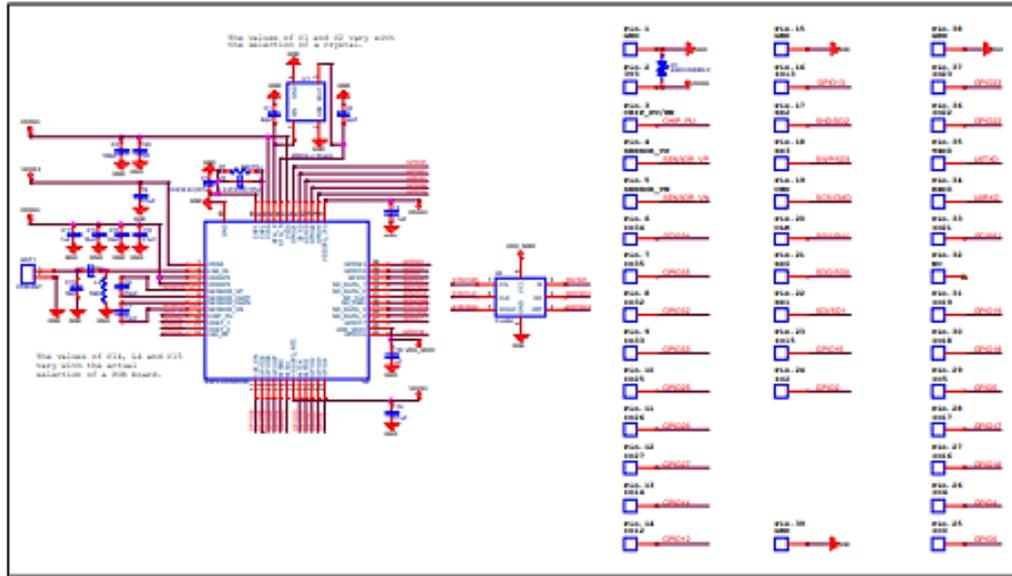


Figure 3: ESP32-WROOM-32 Schematics

7 Peripheral Schematics

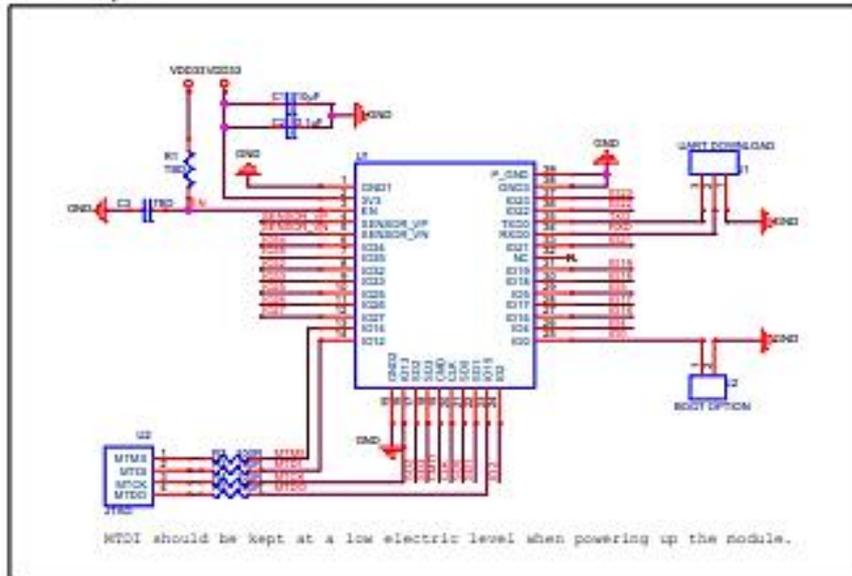


Figure 4: ESP32-WROOM-32 Peripheral Schematics

Note:

- Soldering Pad 39 to the Ground of the base board is not necessary for a satisfactory thermal performance. If users do want to solder it, they need to ensure that the correct quantity of soldering paste is applied.
- To ensure the power supply to the ESP32 chip during power-up, it is advised to add an RC delay circuit at the EN pin. The recommended setting for the RC delay circuit is usually $R = 10\text{ k}\Omega$ and $C = 1\ \mu\text{F}$. However, specific parameters should be adjusted based on the power-up timing of the module and the power-up and reset sequence timing of the chip. For ESP32's power-up and reset sequence timing diagram, please refer to Section Power Scheme in [ESP32 Datasheet](#).

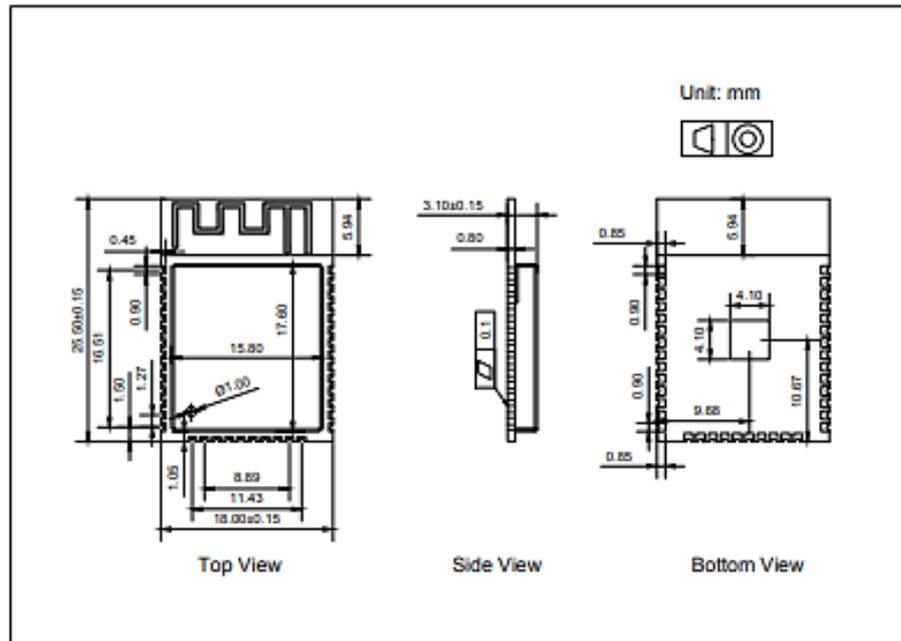
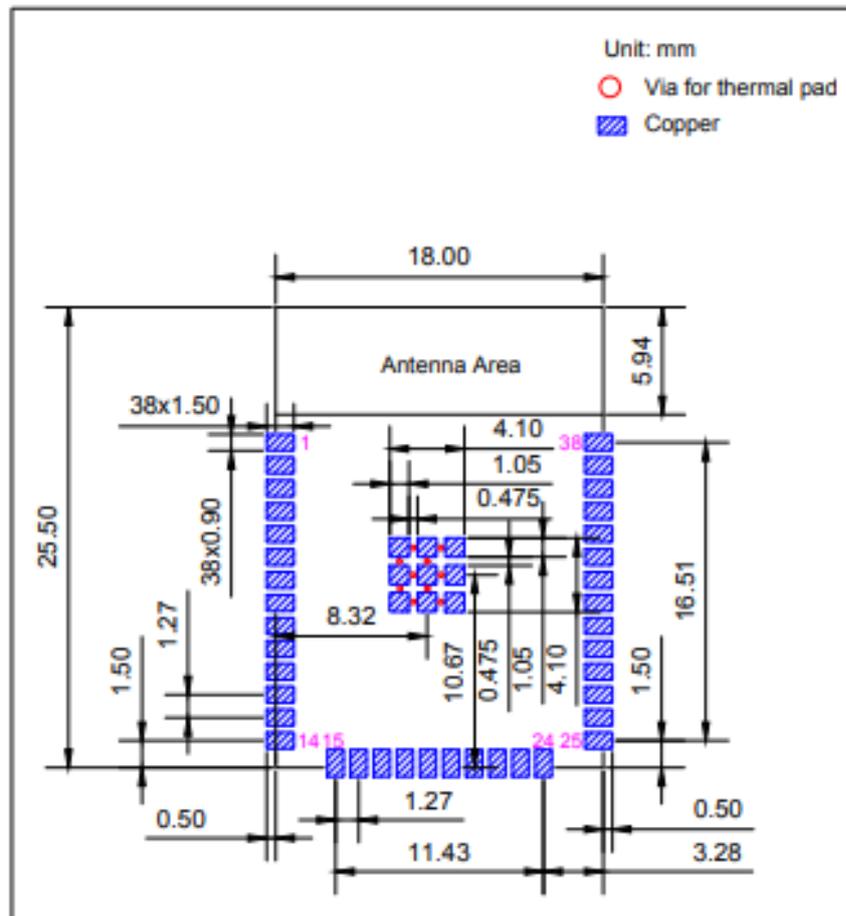


Figure 5: Physical Dimensions of ESP32-WROOM-32

8 Physical Dimensions

9 Recommended PCB Land Pattern



10 Learning Resources

10.1 Must-Read Documents

The following link provides documents related to ESP32.

- [ESP32 Datasheet](#)
This document provides an introduction to the specifications of the ESP32 hardware, including overview, pin definitions, functional description, peripheral interface, electrical characteristics, etc.
- [ESP32 ECO V3 User Guide](#)
This document describes differences between V3 and previous ESP32 silicon wafer revisions.
- [ECO and Workarounds for Bugs in ESP32](#)
This document details hardware errata and workarounds in the ESP32.
- [ESP-IDF Programming Guide](#)
It hosts extensive documentation for ESP-IDF ranging from hardware guides to API reference.
- [ESP32 Technical Reference Manual](#)
The manual provides detailed information on how to use the ESP32 memory and peripherals.
- [ESP32 Hardware Resources](#)
The zip files include the schematics, PCB layout, Gerber and BOM list of ESP32 modules and development boards.
- [ESP32 Hardware Design Guidelines](#)
The guidelines outline recommended design practices when developing standalone or add-on systems based on the ESP32 series of products, including the ESP32 chip, the ESP32 modules and development boards.
- [ESP32 AT Instruction Set and Examples](#)
This document introduces the ESP32 AT commands, explains how to use them, and provides examples of several common AT commands.
- [ESP Product Selector](#)

10.2 Must-Have Resources

Here are the ESP32-related must-have resources.

- [ESP32 BBS](#)
This is an Engineer-to-Engineer (E2E) Community for ESP32 where you can post questions, share knowledge, explore ideas, and help solve problems with fellow engineers.
- [ESP32 GitHub](#)
ESP32 development projects are freely distributed under Espressif's MIT license on GitHub. It is established to help developers get started with ESP32 and foster innovation and the growth of general knowledge about the hardware and software surrounding ESP32 devices.
- [ESP32 Tools](#)
This is a webpage where users can download ESP32 Flash Download Tools and the zip file "ESP32 Certification and Test".

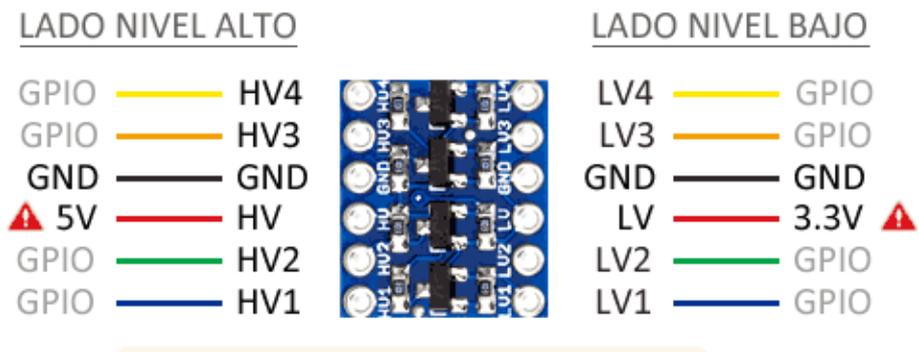
- [ESP-IDF](#)

This webpage links users to the official IoT development framework for ESP32.

- [ESP32 Resources](#)

This webpage provides the links to all available ESP32 documents, SDK and tools.

ANEXO B: ESQUEMA DE CONEXIÓN CONVERTOR TTL 5V/3.3V.



ANEXO C: CÓDIGO DE PROCESAMIENTO DE DATOS EN ARDUINO IDE.

```
#include <Arduino.h>
//#include <SPI.h>
#include <ThingyESP32.h>
#define USERNAME "davidalejo"
#define DEVICE_ID "dht111"
#define DEVICE_CREDENTIAL "esp32%"

#define SSID "Redmi 9 David"
#define SSID_PASSWORD "123456780"
ThingyESP32 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
#include <DHT.h>
#define DHTPIN 22 //GPIO 4
#define DHTTYPE DHT11 //Sensor dht 11
DHT dht(DHTPIN, DHTTYPE);
float t=0,h=0; //para auxiliar de thingy io
//::::::::::::::::::::::::::::::::::::::::::
const int analogInPinPH =35;
const int pin=34;
float phValor, temp, hum;
unsigned long timer1, timer2, timer3, timerAlarm;
    int alarmaBajo = 40;
    int alarmaMedio = 80;
    int alarmaBajoT=9;
    int alarmaMedioT=20;
    float alarmaBajoPh=5;
    float alarmaMedioPh=7;
//::::::::::::::::::::::::::::::::::::::::::
int valvula=2; //GPIO

float phVoltaje=0;

pson data;

void setup() {

    Serial.begin(9600);
    pinMode(valvula, OUTPUT);
    dht.begin();
    // WiFi.begin(ssid,pasw);
```

```

thing.add wifi(SSID, SSID PASSWORD);
thing["VolumenTanque3"] >> outputValue(calcularPh());
thing ["VolumenTanque2"]>> outputValue(Temp());
thing ["VolumenTanque1"]>> outputValue(Hum());
  //thing["VolumenTanque1"] >> outputValue(Hum());
  thing ["Valvula"]<<digitalPin(valvula);

timer1=millis();
timer2=millis();
timer3=millis();
timerAlarm=0;

//Timer1.initialize(50000);
//.attachInterrupt(modbusFunction);
}

void loop() {
  thing.handle();
  Temp();
  Hum();
  calcularPh();
}

//float TempHumLuz( float *temp_ext, float *hum_ext){
//  *temp_ext=dht.readTemperature();
//  *hum_ext=analogRead(Sensor pin);
//  return 0;
//}

float Temp(){
  float temp=dht.readTemperature();

  Serial.println("Temp");
  Serial.println(temp);
  data ["VolumenTanque2"]=temp;
  if(millis())>=timer1+timerAlarm+3000) //timer para verificar alarmas
  cada 3 segundos
  {
    if (temp<alarmaBajoT ||temp<alarmaMedioT){
      timerAlarm=57000;
    data ["VolumenTanque2"] =temp;

    if( temp<alarmaBajoT)
      {
        data["TanqueNo"] = "Nivel de temperatura bajo ";
      }
      else if (temp<alarmaMedioT )
      {
        data["TanqueNo"] = "Nivel de temperatura medio";

```

```

        }
        thing.call_endpoint("NotificacionTanque", data);
    }
    else
    {
        timerAlarm=0;
    }
    timer2=millis();
}
return(temp);
}
float Hum(){
    h=analogRead(pin);
    //valor prueba
    Serial.println("en valor h");
    Serial.println(h);
    float hum =map(h, 0,4095,100,0);

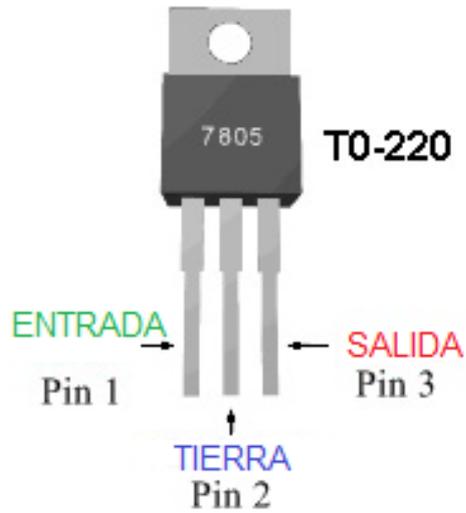
    Serial.println("humedad es");
    Serial.println(hum);
    if(millis())>=timer1+timerAlarm+3000) //timer para verificar alarmas cada 3 segundos
    {
        if(hum<alarmaBajo|| hum<alarmaMedio)
        {
            timerAlarm=57000; // si la alarma persiste se incrementa el delay a 60 segundos despues de la primera notificación para no saturar la nube con alarmas

            data ["VolumenTanque1"] = hum;
            Serial.println("humedad");
            Serial.println(hum);

            if( hum<alarmaBajo)
            {
                data["TanqueNo"] = "Nivel de humedad bajo ";
            }
            else if (hum<alarmaMedio )
            {
                data["TanqueNo"] = "Nivel de humedad medio";
            }
            thing.call_endpoint("NotificacionTanque", data);

```


ANEXO D: ESQUEMA DE CONEXIÓN REGULADOR 7805.



ANEXO E: XML ACTIVITY_MENU.XML (MENÚ PRINCIPAL).

```
Activity main
<?xml version="1.0" encoding="utf-8" ?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"

tools:context="com.oscaranaluisa.notificacionalarma.MainActivity">

    <TextView
        android:id="@+id/infoTextView"
        android:layout_width="283dp"
        android:layout_height="wrap_content"
        android:layout_gravity="start"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="76dp"
        android:textAlignment="viewStart"

        android:textAppearance="@style/TextAppearance.AppCompat.Headline"
        android:textColor="@color/colorAccent"
        android:textSize="22sp"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintLeft_toRightOf="@+id/imageViewAlarma"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/buttonSuscribe"
        android:layout_width="99dp"
        android:layout_height="53dp"
        android:layout_marginLeft="16dp"
        android:layout_marginTop="44dp"
        android:text="@string/stringButtonSuscribe"
```

```

        android:textColor="#009688"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/buttonNavegar"
    />

    <Button
        android:id="@+id/buttonCancela"
        android:layout_width="99dp"
        android:layout_height="53dp"
        android:layout_marginBottom="68dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="8dp"
        android:text="@string/stringButtonCancela"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/buttonSuscribe"
        app:layout_constraintVertical_bias="1.0" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="350dp"
        android:layout_height="35dp"
        android:layout_marginRight="24dp"
        android:layout_marginTop="32dp"
        android:text="@string/stringMessage"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toStartOf="@+id/buttonSuscribe"
        app:layout_constraintTop_toBottomOf="@+id/buttonCancela"
    />

    <Button
        android:id="@+id/buttonNavegar"
        android:layout_width="172dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="228dp"
        android:layout_marginStart="28dp"
        android:text="@string/stringButtonDashboard"
        android:visibility="visible"
        android:background="#00E5FF"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id/buttonSalir"
        android:layout_width="154dp"
        android:layout_height="40dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginTop="8dp"
        android:text="@string/stringButtonSalir"
        android:textColor="#F50057"
        app:layout_constraintBottom_toTopOf="@+id/textView2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/buttonNavegar"
        app:layout_constraintVertical_bias="1.0" />

    <TextView
        android:id="@+id/textViewBody"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="8dp"

```

```

        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"

    android:textAppearance="@style/TextAppearance.AppCompat.Body2"
        android:textSize="18sp"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/infoTextView" />

<ImageView
    android:id="@+id/imageViewAlarma"
    android:layout_width="54dp"
    android:layout_height="47dp"
    android:layout_marginBottom="8dp"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_red_light"
    android:contentDescription="@string/app_name"
    android:visibility="invisible"
    app:layout_constraintBottom_toTopOf="@+id/textViewBody"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.788"
    app:srcCompat="@drawable/ic_stat_name" />

<ImageView
    android:id="@+id/imageView7"
    android:layout_width="81dp"
    android:layout_height="44dp"
    app:layout_constraintEnd_toStartOf="@+id/textViewBody"
    app:layout_constraintHorizontal_bias="0.148"
    app:layout_constraintStart_toEndOf="@+id/textViewBody"
    app:srcCompat="@drawable/logo"
    tools:layout_editor_absoluteY="3dp" />

<ImageView
    android:id="@+id/imageView9"
    android:layout_width="85dp"
    android:layout_height="44dp"
    app:srcCompat="@drawable/maquinas"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="3dp" />

<TextView
    android:id="@+id/textView"
    android:layout_width="175dp"
    android:layout_height="19dp"
    android:layout_gravity="center_horizontal"
    android:text="MONITOREO DE NIVELES "

    android:textColor="#AA00FF"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.559"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.032" />

<android.support.constraint.Guideline

```

```

        android:id="@+id/guideline4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_begin="20dp" />

<android.support.constraint.Guideline
    android:id="@+id/guideline5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_begin="20dp" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="28dp"
    android:layout_gravity="center_horizontal"
    android:text="DE DIESEL"
    android:textColor="#AA00FF"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.528"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.072" />

<ImageView
    android:id="@+id/imageView2"
    android:layout_width="150dp"
    android:layout_height="200dp"

    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    app:layout_constraintBottom_toTopOf="@+id/buttonSalir"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textViewBody"
    app:srcCompat="@drawable/iess" />

</android.support.constraint.ConstraintLayout>

```

Descripción: Este código genera la interfaz para el usuario.

ANEXO F: CLASE JAVA MIFIREBASEINSTANCESERVICE.

Clase Java miFirebaseInstanceService

```
package com.davidalejo.notificacionalarma;
import android.util.Log;
import com.google.firebase.iid.FirebaseInstanceId;
import com.google.firebase.iid.FirebaseInstanceIdService;

/**
 * 7/26/2021.
 */

public class miFirebaseInstanceIdService extends
FirebaseInstanceIdService {
    public static final String TAG="DASHBOAR";

    @Override
    public void onTokenRefresh() {
        super.onTokenRefresh();

        String token =
FirebaseInstanceId.getInstance().getToken();
        // Log.d(TAG,"Token: " + token );

    }
}
```

Descripción:

- Se realiza el llamado a la instancia creada por firebase del archivo json.
- Enlaza la aplicación con la plataforma firebase.