



UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA,
ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**DISEÑO DE UNA ARQUITECTURA BASADA EN INTERNET DE
LAS COSAS PARA LA OPTIMIZACIÓN DE LA CALIDAD DEL
SUELO Y AMBIENTE DE LOS JARDINES DEL DISTRITO DE
PUNO**

TESIS

PRESENTADA POR:

JEHAN CARLOS COILA TICONA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS

PUNO – PERÚ

2024



Reporte de similitud

NOMBRE DEL TRABAJO

DISEÑO DE UNA ARQUITECTURA BASADA EN INTERNET DE LAS COSAS PARA LA OPTIMIZACIÓN DE LA CALIDAD DEL SUELO Y AMBIENTE DE LOS JARDINES DEL DISTRITO DE PUNO

AUTOR

Jehan Carlos Coila Ticona

RECuento DE PALABRAS

17287 Words

RECuento DE CARACTERES

99916 Characters

RECuento DE PÁGINAS

105 Pages

TAMAÑO DEL ARCHIVO

3.5MB

FECHA DE ENTREGA

Jan 18, 2024 1:05 PM GMT-5

FECHA DEL INFORME

Jan 18, 2024 1:06 PM GMT-5

● 18% de similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 16% Base de datos de Internet
- Base de datos de Crossref
- 7% Base de datos de trabajos entregados
- 2% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● Excluir del Reporte de Similitud

- Material bibliográfico
- Coincidencia baja (menos de 10 palabras)
- Material citado

Edgar H. Sotomayor

Dr. Guina C. Sotomayor Alzamora
INGENIERO DE SISTEMAS
CIP. 91234

Resumen



DEDICATORIA

A mi Madre, Mujer fuerte bella y luchadora quien me apoyo y guio en mi realizaci3n profesional y su constante apoyo. Su fortaleza y perseverancia han sido un ejemplo constante de lo que significa luchar por nuestros sueos y por un futuro mejor.

Coila Ticona Jehan Carlos



AGRADECIMIENTOS

A la Universidad Nacional del Altiplano, por sus grandes enseñanzas y experiencias, cuna de buenos profesionales para la región Puno y el país.

A mi Escuela Profesional de Ingeniería de Sistemas por su buena instrucción académica que contribuyo en mi desarrollo como profesional, a mi asesor de tesis M.Sc. Edgar Holguin Holguin y a los miembros del jurado integrado por M.Sc. Marga Isabel Ingaluque Arapa, M.Sc. Magali Gianina Gonzales Paco, y el M.Sc. Fidel Huanco Ramos.

Al Ing. Alberth Larry Mendizabal Flores, jefe de la Oficina de Tecnología e Informática y compañeros de trabajo de la Municipalidad Provincial de Puno, por su apoyo, inspiración y predisposición constante.

A mis compañeros de la Escuela Profesional de Ingeniería de Sistemas, quienes fueron fortaleza, apoyo amistad y alegría.

Coila Ticona Jehan Carlos



ÍNDICE GENERAL

	Pág.
DEDICATORIA	
AGRADECIMIENTOS	
ÍNDICE GENERAL	
ÍNDICE DE FIGURAS	
ÍNDICE DE ANEXOS	
ACRÓNIMOS	
RESUMEN.....	12
ABSTRACT	13
CAPÍTULO I	
INTRODUCCIÓN	
1.1. PLANTEAMIENTO DEL PROBLEMA.....	15
1.2. HIPOTESIS	15
1.3. OBJETIVO GENERAL	15
1.3.1. OBJETIVOS ESPECÍFICOS	15
1.4. METODOLOGÍA	16
CAPÍTULO II	
REVISIÓN LITERARIA	
2.1. ANTECEDENTES DE LA INVESTIGACIÓN	17
2.2. SISTEMA DE ADQUISICIÓN DE DATOS	23
2.3. SENSORES.....	25
2.4. SENSOR DHT22	26
2.5. HIGRÓMETRO.....	28
2.6. SENSOR DE GAS TÓXICO.....	30



2.6.1. SENSOR MQ135	31
2.7. SENSOR DS18B20	35
2.8. CONVERTOR ANALOGICO DIGITAL	36
2.9. CALIDAD DEL SUELO.....	38
2.10. CALIDAD DEL AMBIENTE.....	39
2.11. INTERNET DE LAS COSAS	40
2.12. ARDUINO.....	42
2.12.1. MICROCONTROLADOR.....	44
2.12.2. ATMEGA328P.....	47
2.13. LENGUAJE DE PROGRAMACION C++.....	49
2.13.1. ENTORNO DE PROGRAMACIÓN ARDUINO	50
2.14. SERVIDOR.....	51

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. LUGAR DE ESTUDIO.....	54
3.2. UBICACIÓN DE LA INVESTIGACIÓN.....	54
3.3. DISEÑO DE LA INVESTIGACIÓN.....	54
3.4. TIPO DE INVESTIGACIÓN.....	54
3.5. MATERIALES EMPLEADOS.....	55
3.5.1. Hardware	55
3.5.2. Software.....	55
3.6. TÉCNICAS PARA LA ADQUISICIÓN DE DATOS.....	55
3.6.1. SENSOR DHT22.....	57
3.6.2. SENSOR FC-28.....	59
3.6.3. SENSOR MQ135	62



3.6.4. SENSOR DS1820.....	64
3.6.5. CÓDIGO ARDUINO	66

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS	77
4.1.1. Análisis de variables para la optimización de suelo y ambiente.....	77
4.1.2. Casos de uso	78
4.1.3. Implementación del prototipo para la optimización del suelo y ambiente de los jardines de Puno	78
4.1.4. Diseño de software y hardware a nivel de cliente y servidor	80
4.1.5.Arquitectura del sistema basado en internet de las cosas para la optimización y calidad de suelo.....	83
4.1.6. Despliegue del servicio.....	85
4.1.7. Evaluación de la calidad del servicio implementados	86
4.2. DISCUSIÓN.....	86
4.2.1. En cuanto a los objetivos de nuestra investigación	86
4.3. PRUEBA DE HIPOTESIS	89
V. CONCLUSIONES.....	92
VI. RECOMENDACIONES	94
VII. REFERENCIAS BIBLIOGRÁFICAS.....	95
ANEXOS	99

Área: Sistemas de información

Tema: Internet de las Cosas

FECHA DE SUSTENTACIÓN: 19 de enero del 2024



ÍNDICE DE FIGURAS

	Pág.
Figura 1 Circuito de Hardware de adquisición de datos.....	14
Figura 2 Sensor DHT22.....	16
Figura 3 Diagrama de conexión electrónica DHT22.....	17
Figura 4 Sensor de humedad de suelo	19
Figura 5 MQ135 sensor	21
Figura 6 Circuito de Prueba MQ135	21
Figura 7 Curva de sensibilidad típica del sensor MQ135.....	22
Figura 8 Características típicas de temperatura/humedad	22
Figura 9 Curva de sensibilidad	22
Figura 10 Respuesta del sensor MQ135	23
Figura 11 Sensor DS18B20	24
Figura 12 Conversión analógica digital.....	26
Figura 13 Suelo de jardín.....	28
Figura 14 Arquitectura del Internet de las cosas	31
Figura 15 Arduino UNO.....	32
Figura 16 Microcontrolador.....	33
Figura 17 Diagrama general de un microcontrolador de 8 bits	35
Figura 18 Microcontrolador ATmega328 AVR	38
Figura 19 Entorno de programación Arduino.....	40
Figura 20 Arquitectura de la red de Internet de las Cosas	45
Figura 21 Conexión Arduino uno con sensor DHT.....	46
Figura 22 Pruebas del sensor DHT22 en el monitor serial.....	47
Figura 23 Conexión física del sensor DHT22	47



Figura 24 Diagrama de conexión del sensor FC-28	48
Figura 25 Prueba del sensor de humedad de suelo	49
Figura 26 Conexión física del sensor FC-28	50
Figura 27 Conexión de Arduino con el MQ - 135	51
Figura 28 Prueba del censored MQ 135	51
Figura 29 Conexión física del sensor MQ - 135	52
Figura 30 Conexión del sensor DS18B20	53
Figura 31 Datos obtenidos del sensor DS18B20 en el monitor serial	53
Figura 32 Librerías usadas para leer los sensores	54
Figura 33 Declaración de variables	56
Figura 34 Void setup del código	57
Figura 35 Void loop de código Arduino	59
Figura 36 Librerías para datos en internet	60
Figura 37 Configuración de IP, Gateway, dns, Ipsever	61
Figura 38 Código que posibilita el envío de datos a internet	63
Figura 39 Verificación de la conexión con el servidor	64
Figura 40 Resultados en la Web	69
Figura 41 Prototipo desarrollado	70
Figura 42 Arquitectura del prototipo desarrollado	71
Figura 43 Ubicación del parque José Salcedo	74
Figura 44 Prototipo en el parque José salcedo	75
Figura 45 Ubicación del parque Pedro Vilcapaza	75
Figura 46 Prototipo en el parque Pedro Vilcapaza	76
Figura 47 Porcentaje de optimización en cada uno de los parques	79



ÍNDICE DE ANEXOS

	Pág.
Anexo:1 Código Fuente.....	99
Anexo:2 Código Fuente.....	101
Anexo:3 Declaración jurada de autenticidad de tesis.....	109
Anexo:4 Autorización para el depósito de tesis o trabajo de investigación en el repositorio institucional	110



ACRÓNIMOS

IoT:	Internet de las cosas
OS:	Sistema Operativo
HTTP:	Protocolo de transferencia de hipertexto
IP:	Dirección del Protocolo de Internet
MAC:	Control de acceso a medios
UTP:	Par Trenzado sin Blindaje
OSI:	Modelo de interconexión de sistemas abiertos
ADC:	Convertidor Analógico-Digital
IUPAC:	Unión Internacional de Química Pura y Aplicada
CPU:	Unidad Central de Procesamiento
RAM:	Memoria de Acceso Aleatorio
E/S:	Entrada/Salida
GPIO:	Entrada/Salida de Propósito General
IDE:	Entorno de Desarrollo Integrado
DNS:	Sistema de nombres de dominio



RESUMEN

En la sociedad actual, la supervisión efectiva de áreas verdes se ha convertido en una necesidad esencial. Estas áreas verdes desempeñan un papel fundamental al absorber dióxido de carbono, promover el bienestar mental de las personas, regular la temperatura y fomentar prácticas sostenibles. En este contexto, nuestro enfoque se centra en la creación de un sistema basado en Internet de las Cosas (IoT) destinado a supervisar la calidad del suelo y del entorno en jardines, con el propósito de optimizar el cuidado de estos espacios verdes. En este proyecto se realiza la implementación de un sistema que monitorea diversos parámetros, como la temperatura y humedad ambiental, la temperatura y humedad del suelo, así como la presencia de gases tóxicos. Los datos recopilados por los sensores son transmitidos a través de Internet a un servidor central, lo que permite la visualización en tiempo real. La arquitectura de IoT utilizada para este propósito tiene varios componentes esenciales, incluyendo un Arduino que interpreta tanto datos digitales como analógicos. Estos datos se envían a través de una Raspberry Pi, que actúa como un enrutador para permitir la conectividad a Internet fuera de la red local. Este enfoque integral garantiza un monitoreo efectivo y una gestión eficiente de los jardines, contribuyendo así a la conservación y el cuidado de estos valiosos espacios urbanos. Como resultado, hemos logrado una verificación del estado de los jardines de manera efectiva y sin impacto negativo en el entorno, lo que conduce a una ciudad más limpia y un ambiente propicio para la salud de sus habitantes.

Palabras claves: Arduino, Internet de las cosas, jardín, servidor, salud.



ABSTRACT

In today's society, effective supervision of green areas has become an essential need. These green areas play a vital role in absorbing carbon dioxide, promoting people's mental well-being, regulating temperature and encouraging sustainable practices. In this context, our approach focuses on the creation of a system based on the Internet of Things (IoT) aimed at monitoring the quality of the soil and the environment in gardens, with the purpose of optimizing the irrigation and care of these green spaces. In this project, a system is implemented that monitors various parameters, such as environmental temperature and humidity, soil temperature and humidity, as well as the presence of toxic gases. The data collected by the sensors is transmitted over the Internet to a central server, allowing real-time visualization. The IoT architecture used for this purpose has several essential components, including an Arduino that interprets both digital and analog data. This data is sent through a Raspberry Pi, which acts as a router to allow Internet connectivity outside the local network. This comprehensive approach guarantees effective monitoring and efficient management of the gardens, thus contributing to the conservation and care of these valuable urban spaces. As a result, we have achieved a verification of the condition of the gardens effectively and without negative impact on the environment, which leads to a cleaner city and an environment conducive to the health of its inhabitants.

Keywords: Arduino, Internet of things, garden, server, health.



CAPÍTULO I

INTRODUCCIÓN

El creciente interés en la protección del entorno natural y la búsqueda de soluciones tecnológicas innovadoras han impulsado el desarrollo de sistemas de supervisión ambiental cada vez más avanzados. En este contexto, la incorporación de la Internet de las Cosas (IoT) ha surgido como una herramienta poderosa para monitorear y mejorar la calidad del suelo y del entorno en general. El presente estudio se concentra en la concepción y ejecución de un sistema IoT cuyo propósito principal es la supervisión en tiempo real de la calidad del suelo y del ambiente.

La calidad del suelo y del entorno desempeña un papel fundamental en la salud de los ecosistemas y la calidad de vida de las comunidades humanas. Factores como la degradación del suelo, la contaminación del aire y del agua, y otros aspectos ambientales pueden tener consecuencias devastadoras en la biodiversidad y la sostenibilidad de los recursos naturales. Por lo tanto, resulta imperativo contar con herramientas eficaces para observar y comprender estos fenómenos, lo que a su vez facilita la toma de decisiones basadas en información sólida y la implementación de estrategias destinadas a conservar y mejorar el medio ambiente.

En este contexto, el Internet de las Cosas ha demostrado ser una tecnología revolucionaria al permitir la recopilación eficiente y a gran escala de datos ambientales en tiempo real. La combinación de sensores avanzados, conectividad inalámbrica y capacidades de procesamiento de datos de vanguardia ofrece una amplia gama de posibilidades para supervisar de manera continua y precisa la calidad del suelo y del ambiente. Este enfoque no solo proporciona información valiosa para la toma de



decisiones, sino que también contribuye a la preservación de los recursos naturales y a la reducción de los impactos negativos en el medio ambiente.

A lo largo de esta investigación, exploraremos los elementos clave de nuestro sistema IoT, desde la selección de sensores especializados hasta la creación de una infraestructura de comunicación sólida. Además, analizaremos las aplicaciones prácticas de este sistema en la supervisión de la calidad del suelo y del ambiente, resaltando su potencial para mejorar la gestión de los recursos naturales y la capacidad de respuesta ante eventos ambientales adversos.

1.1. PLANTEAMIENTO DEL PROBLEMA

¿Cómo realizar el diseño de una arquitectura basada en Internet de las cosas que permita la optimización de la calidad del suelo y ambiente de los jardines del distrito de Puno?

1.2. HIPOTESIS

El diseño de una arquitectura basada en Internet de las cosas permitirá la optimización de la calidad del suelo y ambiente de los jardines del distrito de Puno.

1.3. OBJETIVO GENERAL

Diseñar una arquitectura basada en Internet de las cosas para la optimización de la calidad del suelo y ambiente de los jardines del distrito de Puno.

1.3.1. OBJETIVOS ESPECÍFICOS

- Determinar las variables fundamentales para la optimización la calidad de suelo y ambiente en los jardines del distrito de Puno.



- Implementar un prototipo para la monitorización de las variables de calidad de suelo y ambiente de los jardines del distrito de Puno.
- Evaluar cómo la arquitectura de Internet de las cosas permite la optimización del tratamiento de los jardines.

1.4. METODOLOGÍA

Esta investigación será experimental, ya que realizará pruebas y prototipos de los sistemas implementados en los parques de la ciudad de Puno, seguido de una evaluación de la eficiencia del sistema de internet de las cosas.



CAPÍTULO II

REVISIÓN LITERARIA

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

Según (Palak & Ferdous Ahmed, 2019) en su investigación titulada “Estación meteorológica basada en la nube que utiliza dispositivos IoT” concluyeron que este trabajo ha logrado su objetivo propuesto de desarrollar una estación meteorológica inteligente, modular y rentable que almacena de forma remota datos climáticos en un servidor de base de datos al que se puede acceder desde cualquier lugar a través de cualquier dispositivo. El uso de Raspberry Pi Zero W hizo que el producto fuera mucho más compacto y, por lo tanto, facilitó la instalación. El uso de múltiples módulos de este tipo ayudó a lograr la precisión y también resolvió el problema de las falsas alarmas. Usando Raspberry Pi 3 como elemento central de procesamiento y usando todas sus características, con tamaño físico reducido, bajo costo de energía y alta potencia de procesamiento.

Según (Chaoying, 2022) en su investigación titulada “Sistema de Monitoreo Ambiental Ecológico Multipunto con Internet de las Cosas” llegó a la conclusión que, si bien la ciencia y la tecnología brindan comodidad a las personas, también destruyen continuamente el entorno ecológico. El objetivo es principalmente construir un sistema de monitoreo ambiental ecológico multipunto con la plataforma Yeelink IoT. Por lo tanto, este trabajo utiliza medios de alta tecnología para monitorear los cambios en la contaminación ambiental ecológica y al mismo tiempo recuerda a los ciudadanos que siempre deben prestar atención a la protección del medio ambiente. En este trabajo se ha diseñado un sistema de monitoreo ambiental ecológico multipunto basado en Internet de las cosas. Este sistema es capaz de realizar un monitoreo inteligente en tiempo real del



entorno ecológico, logrando los objetivos de monitoreo del entorno ecológico e integración de parámetros de medición, modularización del sistema, transmisión de datos en tiempo real e integración de servicios de monitoreo. En resumen, este sistema de monitoreo ecológico es una solución integral y altamente eficiente para el monitoreo ambiental, lo que permitirá obtener datos precisos y en tiempo real sobre el estado de nuestro entorno ecológico, lo que es esencial para la protección y conservación del medio ambiente. Se construye un sistema de monitoreo ambiental ecológico multipunto con la plataforma Yeelink IoT. Se usó la placa de control principal y varios sensores para construir un sistema de hardware, escribir los programas de software correspondientes y recopilar varios parámetros ambientales ecológicos. Se construye un 1D-BN-CNN para evaluar la calidad del entorno ecológico. Evalúa la calidad del entorno ecológico actual a través de los datos ambientales recopilados en tiempo real por los sitios de monitoreo y a través de la plataforma de Internet de las cosas para mostrarlos en la terminal.

Según (Sanjeevi y otros, 2020) en su investigación titulada “Agricultura de precisión y ganadería con Internet de las cosas basado en una red de sensores inalámbricos” llegaron a la conclusión de que la agricultura de precisión basada en IoT y el sistema agrícola pueden confirmar que son extremadamente útiles para los agricultores, ya que una cantidad mayor o menor de riego es buena para la agricultura. Para optimizar el rendimiento del sensor, es importante ajustar los valores de los parámetros a las condiciones específicas del entorno en el que se encuentra. Por ejemplo, en el caso de la temperatura y la humedad, se debe considerar la influencia del clima y las características del campo agrícola para establecer un rango adecuado de humedad. De esta manera, se pueden obtener mediciones más precisas y útiles para la monitorización de las condiciones ambientales y el seguimiento del crecimiento de los cultivos. El sistema propuesto construirá una utilización optimizada de los recursos y resolverá el problema



de la escasez de riego. La representación gráfica del rendimiento significativo de WSN es mejor que las tecnologías existentes que podrían recuperarse y mostrarse gráficamente. En el futuro, se puede implementar un sistema de detección de intrusión de animales para proteger los cultivos y los agricultores de los ataques de animales. El monitoreo de campo en tiempo real es posible mediante el uso de tecnología IoT. El desperdicio de recursos agrícolas está altamente controlado en la metodología propuesta.

Según (Puranik y otros, 2019) en su investigación titulada “Automatización en Agricultura e IoT” concluyeron que la agricultura ha desempeñado un papel importante en el crecimiento económico de la India desde entonces y seguirá siéndolo en el futuro. Los agricultores, por otro lado, enfrentan desafíos en diferentes fases de la agricultura. El objetivo de este documento dado, trata de resolver el problema utilizando IoT y la automatización que puede gestionar la mayor parte del trabajo agrícola y los agricultores pueden diseñar estrategias sobre qué cultivos cultivar de acuerdo con el mercado en lugar de pasar la mayor parte del tiempo en el mantenimiento y la producción de cultivos. También puede ayudar a los agricultores a dar más tiempo para su vida personal, por lo tanto, aumentar el estándar social promedio de la sociedad.

Según (Gutierrez Perez & Narvaez Mellado, 2022) en su investigación titulada “Sistema de internet de las cosas para mejorar el proceso de riego en la comunidad de Matará” llegaron a la conclusión de que la implementación del sistema IoT ha resultado en una reducción del 26.05 % en el volumen utilizado en el riego de la comunidad de Matará. Los datos de preprueba de 38 m³ y posprueba de 28.1 m³ del volumen han servido como referencia. Además, se ha constatado que el sistema IoT ha disminuido el tiempo empleado en el abastecimiento de agua, pasando de 230 minutos a 24.3 minutos, lo que representa una disminución del 89.43 % cuando los usuarios solicitan agua de uno de los reservorios. Este sistema ha demostrado ser beneficioso en la reducción de costos



en el proceso de riego, reduciendo los costos de 163.00 soles a 47.96 soles, lo que equivale a un 70.57 % de los costos. Asimismo, también ha contribuido en la reducción de costos en la adquisición de agua, disminuyendo de 63.33 soles a 44.96 soles, lo que representa un 29.00 % de los costos. Es importante destacar que este proyecto tiene como objetivo administrar el agua en la comunidad, permitiendo que los comuneros interactúen con nuevas tecnologías y observen sus beneficios, lo que ha generado un gran interés en el uso de estas tecnologías en sus tareas diarias.

Según (Mora Magallanes & Rosas Pari, 2019) en la investigación denominada "Diseño, desarrollo e implementación de una red de sensores inalámbricos (WSN) para el control, monitoreo y toma de decisiones en la agricultura de precisión basada en Internet de las cosas (IoT) - un caso de estudio enfocado en el cultivo de frijol", se llegó a la conclusión de que se creó y puso en funcionamiento un dispositivo electrónico con la capacidad de establecer conexión a Internet, empleando el módulo ESP12E del MCU ESP8266, con el objetivo de fomentar la adopción de tecnologías emergentes en el ámbito de IoT. El equipo logró con éxito la conexión del dispositivo a Internet y la integración de varios sensores, que abarcaban desde la medición de la temperatura ambiente y la humedad del suelo hasta la detección de lluvia y la evaluación de niveles de humo. Esto permitió la creación de una red de sensores inalámbricos basada en la tecnología IoT, que monitoreó cinco factores ambientales y transmitió los datos a través de Internet, donde se almacenaron en la nube. Este registro de información resultó esencial para mejorar la toma de decisiones en la agricultura de precisión, específicamente en el contexto del cultivo de frijol. Las condiciones ambientales pudieron ser monitoreadas en tiempo real a través de una interfaz de usuario simple y de fácil comprensión. Además, el dispositivo electrónico funcionó de manera autónoma gracias al uso de baterías de iones de litio y paneles solares que proporcionaron la energía necesaria. Las pruebas realizadas



confirmaron el éxito de la conexión del módulo ESP12E a Internet con un consumo de energía reducido. Se estableció una plataforma de servicios mediante el uso de AWS y su herramienta EC2, lo que facilitó el almacenamiento y acceso a los datos, así como su visualización a través de una página web y Node-Red. Estos logros posibilitaron un mayor control, seguimiento y toma de decisiones en el ámbito de la agricultura de precisión, específicamente en lo que respecta al cultivo de frijol. Esto ha proporcionado a los agricultores una herramienta valiosa para mejorar el cuidado y la gestión de sus cultivos de frijol.

De acuerdo con la investigación realizada por (Guerra Guerra y otros, 2019) en su estudio titulado "Desarrollo de un servicio de monitoreo y control de áreas verdes en un municipio de Lima a través de la implementación de Internet de las Cosas (IoT)", se llega a la conclusión de que el proyecto actual permite la obtención de datos en tiempo real desde los puntos de gestión de áreas verdes. Estos datos son posteriormente enviados a un repositorio mediante el uso de una plataforma de nube pública. El objetivo principal del proyecto es establecer un servicio de monitoreo y control de áreas verdes en un municipio de Lima a través de la aplicación de Internet de las Cosas (IoT). Además, se pudo verificar que tanto la red WiFi como la red móvil son igualmente adecuadas para la solución propuesta. Sin embargo, se aconseja que, en caso de llevar a cabo una implementación a gran escala que abarque toda la ciudad de Lima, la red móvil sería la opción más apropiada para la comunicación entre los sensores y el depósito de almacenamiento en la nube. En el futuro, se tiene previsto incorporar paneles solares para la generación de energía. Al mismo tiempo, se está trabajando en la creación de un repositorio de datos en la Plataforma de Interoperabilidad del Estado (PIDE) con el fin de almacenar la información en una base de datos NoSQL (MongoDB) y, más tarde,



procesarla mediante técnicas de big data o análisis de datos, de acuerdo con lo señalado en el estudio realizado por Difallah, Benahmed, Draoui y Bounaama en el año 2018.

Según la investigación realizada por (Medina de la Cruz, 2020) en su estudio titulado "Desarrollo e implementación de un dispositivo de Internet de las Cosas para la medición y monitoreo de la calidad del aire en áreas urbanas," se concluye que Lima figura como la segunda ciudad más afectada por la contaminación en América Latina, lo que genera una inquietud considerable en la población local. El objetivo primordial de este proyecto es concebir y ejecutar un dispositivo basado en Internet de las Cosas (IoT) destinado a la medición y supervisión de la calidad del aire en entornos urbanos. A pesar de la existencia de regulaciones nacionales orientadas a la supervisión y control de emisiones con el fin de preservar el medio ambiente, estas medidas parecen insuficientes para abordar esta problemática. Se observa que las enfermedades respiratorias, como la disminución de la frecuencia respiratoria, el asma, la enfermedad pulmonar obstructiva crónica y el cáncer de pulmón, son las principales afecciones que pueden afectar la salud de la población, reflejándose en cambios en la apariencia pulmonar en distintas etapas de la vida de las personas. Además, se nota que los equipos utilizados por instituciones públicas y privadas para medir la contaminación son voluminosos y costosos, lo que conlleva gastos considerables tanto en la adquisición de los dispositivos como en la logística para la obtención de muestras. En consecuencia, se enfatiza la necesidad apremiante de implementar un dispositivo de medición de la calidad del aire que cumpla con requisitos técnicos específicos, tales como un bajo consumo energético, portabilidad, margen de error mínimo, un sistema abierto y la capacidad de evaluar parámetros climáticos y concentraciones de contaminantes en tiempo real a través de la conexión IoT. Además, se destaca la importancia de emplear este dispositivo en investigaciones ambientales debido a su naturaleza abierta que permite la incorporación de nuevos



componentes para futuras aplicaciones. Finalmente, se resalta que la plataforma de seguimiento de la calidad del aire opera en tiempo real y exhibe los resultados de los sensores integrados en el dispositivo IoT, lo que facilita la visualización y descarga de los datos recolectados durante el proceso de muestreo.

2.2. SISTEMA DE ADQUISICIÓN DE DATOS

A pesar de la ambigüedad en esta definición, la mayoría de los usuarios de computadoras, ingenieros y científicos coinciden en que una computadora personal (PC) se utiliza para diversas funciones, como programar equipos de prueba, manipular o almacenar datos, generar gráficos en tiempo real y crear informes. Esta PC puede estar conectada o no a un equipo de adquisición de datos, y el equipo de prueba puede incluir tarjetas enchufables para PC, chasis de tarjetas externas o instrumentos independientes conectados a la PC a través de puertos de comunicación estándar o una placa de interfaz patentada. A continuación, el equipo de prueba puede llevar a cabo procesos de medición y control utilizando diferentes combinaciones de entradas y salidas.

Los instrumentos modernos han evolucionado para incluir ranuras para tarjetas y microprocesadores, así como sistemas operativos de software. Estos instrumentos se asemejan más a computadoras que a los dispositivos tradicionales, lo que permite la creación de sistemas de prueba con una alta capacidad de almacenamiento y registro de datos en una computadora de control. Las tarjetas enchufables pueden convertir una computadora en dispositivos versátiles, como multímetros digitales de múltiple rango, osciloscopios y otros instrumentos con paneles frontales virtuales en pantalla que son fáciles de usar.

El hardware para la adquisición y control de datos se presenta en diversas formas, con diferentes niveles de funcionalidad, cantidad de canales, velocidad, resolución,



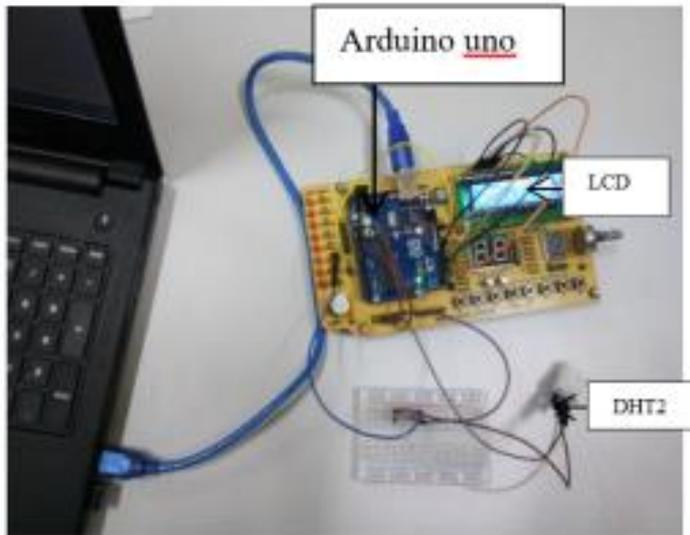
precisión y costo. En esta sección, proporcionamos una visión general de las características y beneficios comunes de estas categorías, basada en una amplia muestra de productos de diversos fabricantes.

Por lo general, los sistemas de adquisición de datos (DAQ) están compuestos por hardware y software DAQ, sensores y actuadores, y suelen requerir una infraestructura de soporte para la transferencia de datos entre el hardware y el software de adquisición de datos. El hardware generalmente se compone de componentes externos, como tarjetas de expansión, que se conectan a la computadora a través de una interfaz de comunicación como PCI o USB, o se pueden instalar directamente en la placa base. Este hardware se conecta a un dispositivo de entrada, como un escáner 3-D o un convertidor analógico a digital. La señal del dispositivo de entrada se procesa en el dispositivo/tarjeta de hardware y luego se envía al software DAQ, donde se almacena para su posterior revisión y análisis.

Un sistema de adquisición de datos representa una herramienta valiosa para recopilar, procesar y almacenar datos para su análisis posterior. Estos sistemas ofrecen diversas ventajas, como mejorar la productividad al acelerar la recopilación y el procesamiento de datos, reducir costos al evitar la necesidad de personal especializado y proporcionar flexibilidad para adaptarse a una variedad de entornos y aplicaciones (Roque y otros, 2013).

Figura 1

Circuito de Hardware de adquisición de datos



Nota: (Trias Prima y otros, 2020)

2.3. SENSORES

Un sensor se define como un dispositivo capaz de identificar o evaluar una característica particular o un cambio en su entorno físico y luego transformar esta información en una señal que puede ser aprovechada, generalmente en forma de señales eléctricas o digitales, y que otros dispositivos, sistemas o software pueden interpretar. Los sensores desempeñan un papel esencial en la obtención de datos del mundo real y se aplican en diversas situaciones en campos como la ciencia, la tecnología, la industria, la medicina y muchas otras áreas.

Los sensores pueden cuantificar una variada gama de variables, como la temperatura, la intensidad lumínica, la presión, la humedad, la velocidad, la posición, la proximidad, la aceleración, el magnetismo, la detección de gases, la medición de líquidos y otras más. Cada tipo de sensor se encuentra diseñado para una tarea específica y hace uso de diferentes tecnologías y principios físicos con el fin de llevar a cabo mediciones precisas.



Los sensores pueden cuantificar una variada gama de variables, como la temperatura, la intensidad lumínica, la presión, la humedad, la velocidad, la posición, la proximidad, la aceleración, el magnetismo, la detección de gases, la medición de líquidos y otras más. Cada tipo de sensor se encuentra diseñado para una tarea específica y hace uso de diferentes tecnologías y principios físicos con el fin de llevar a cabo mediciones precisas (Veneros y otros, 2020).

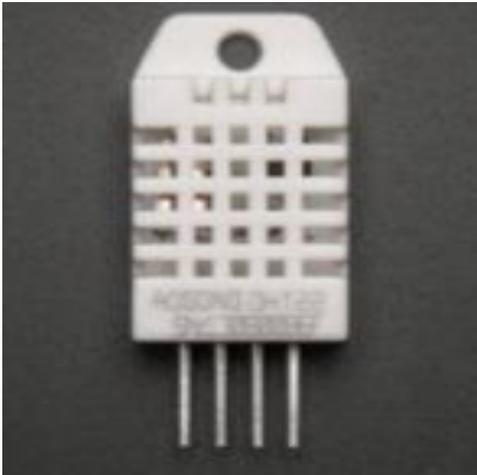
2.4. SENSOR DHT22

El DHT22, también conocido como AM2302, es un sensor versátil que mide la temperatura y la humedad relativa en su entorno. Utiliza un termistor y un material sensible a la humedad para realizar estas mediciones. Un microcontrolador interno procesa los datos y los envía a través de un protocolo de un solo cable, lo que lo hace fácil de conectar a microcontroladores y sistemas embebidos. Este sensor se utiliza ampliamente en proyectos de monitoreo ambiental y control de clima, ofreciendo mediciones precisas y una interfaz sencilla para obtener información valiosa sobre las condiciones del entorno.

Este sensor proporciona una señal digital de salida calibrada. Emplea una técnica única para recopilar señales digitales y utiliza tecnología de detección de humedad, lo que garantiza su confiabilidad y estabilidad. Los componentes de detección están integrados en un solo chip de 8 bits. Cada sensor de este modelo ha sido previamente calibrado en una cámara de calibración precisa, y los coeficientes de calibración se almacenan en la memoria OTP. Cuando el sensor realiza una detección, recupera automáticamente los coeficientes de la memoria. Su diseño compacto, bajo consumo de energía y capacidad de transmisión a larga distancia (hasta 20 metros) lo hacen adecuado para una amplia gama de aplicaciones en diversas condiciones (Trias Prima y otros, 2020)

Figura 2

Sensor DHT22



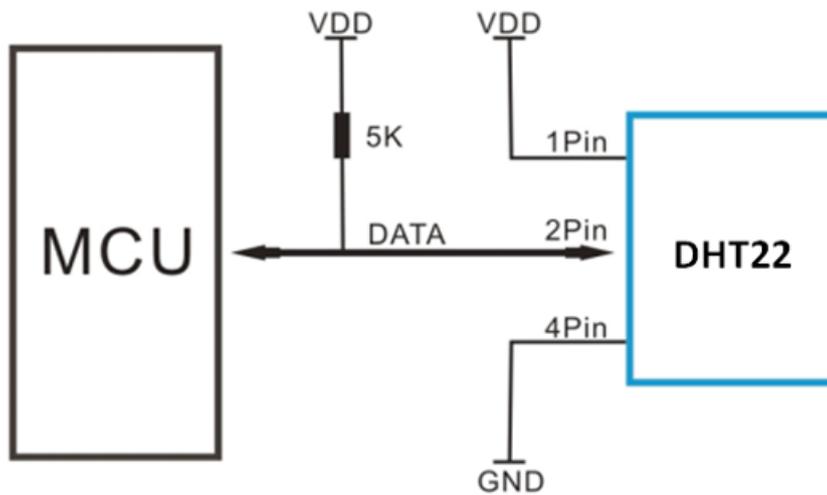
Nota: (Trias Prima y otros, 2020)

Las características del sensor son los siguientes:

- Alimentación de 3.3V a 5VDC
- Corriente máxima 2.5mA durante la conversión
- Lectura de humedad con un +/- 2% a 5% de precisión
- Lectura de temperatura con un +/- 0.5°C de precisión
- Capaz de medir humedad de 0% a 100%
- Capaz de medir temperatura de -40°C a 125°C
- No más de 0.5Hz en velocidad de muestreo (una vez cada dos segundos)
- Dimensiones: 15.1mm x 25mm x 7.7mm

Figura 3

Diagrama de conexión electrónica DHT22



Nota: (Electronics, 2020)

2.5. HIGRÓMETRO

El sensor de humedad de suelo diseñado para su uso con Arduino es un dispositivo empleado para medir la humedad presente en el suelo en una variedad de aplicaciones, como la agricultura, la horticultura y proyectos de jardinería inteligente. Su operación se basa en dos componentes esenciales: las sondas de detección y el módulo de interfaz. Las sondas, compuestas por materiales sensibles a la humedad, se insertan en el suelo y varían su conductividad eléctrica en función de la humedad circundante. Cuanto más húmedo esté el suelo, mayor será la conductividad eléctrica detectada entre las sondas. El módulo de interfaz se conecta a estas sondas y se encarga de convertir las mediciones de resistencia eléctrica en una señal digital comprensible para placas Arduino o microcontroladores similares, generalmente mediante un convertidor analógico a digital (ADC). Una vez que el sensor está conectado a una placa Arduino, es posible programarla para que tome decisiones basadas en los valores de humedad obtenidos, como activar un sistema de riego cuando la humedad del suelo desciende por debajo de un umbral predefinido, lo que contribuye a mantener un nivel óptimo de humedad para las plantas.

Estos sensores de humedad de suelo son herramientas valiosas para gestionar de manera eficiente el riego y el cuidado de las plantas, lo que a su vez puede contribuir a la conservación del agua y mejorar la salud de los jardines o cultivos.

El "módulo de interfaz" o "módulo sensor" desempeña un papel esencial como intermediario entre Arduino y un sensor de humedad del suelo. Su función principal consiste en facilitar la conexión y la adquisición de datos entre el sensor y la placa Arduino. Este módulo permite la conexión física mediante pines específicos que incluyen alimentación eléctrica, conexión a tierra y transmisión de la señal de datos. Además, realiza la amplificación y el procesamiento de la señal generada por el sensor, utilizando, en ocasiones, un convertidor analógico a digital (ADC) para obtener una señal digital representativa del nivel de humedad del suelo. También se encarga de nivelar los voltajes para garantizar una comunicación segura, y se conecta directamente a la placa Arduino para permitir la lectura de datos previamente procesados (Guijarro Rodríguez y otros, 2018).

Figura 4

Sensor de humedad de suelo



Nota: (Guijarro Rodríguez y otros, 2018)



2.6. SENSOR DE GAS TÓXICO

Cuando el entorno se ve afectado por la presencia de sustancias gaseosas perjudiciales, se desencadena una situación de contaminación atmosférica que puede tener consecuencias severas tanto para la salud de las personas como para el equilibrio general del ecosistema. Estos gases nocivos pueden originarse en diversas fuentes, que incluyen la actividad industrial, el transporte, la agricultura y la combustión de combustibles fósiles (Chica García y otros, 2020). Algunos de los efectos y problemas relacionados con la existencia de estas sustancias dañinas en el entorno abarcan:

- **Impacto en la salud humana:** La exposición prolongada o repentina a gases tóxicos puede provocar una serie de problemas de salud, como enfermedades respiratorias, cardiovasculares y neurológicas, así como la posibilidad de cáncer y trastornos del desarrollo en niños. La magnitud de los efectos puede variar según el tipo de gas y la concentración a la que las personas estén expuestas.
- **Daño al medio ambiente:** Los gases perjudiciales pueden tener efectos negativos sobre los ecosistemas, causando daños a la vegetación, impactando en la fauna y contaminando fuentes de agua. Esto puede llevar a la pérdida de biodiversidad y a la degradación del suelo.
- **Calidad del aire:** La presencia de gases tóxicos contribuye a la mala calidad del aire, lo que puede desencadenar alertas de calidad del aire y restricciones en las actividades al aire libre. Además, puede aumentar la formación de smog y la concentración de partículas finas, afectando la visibilidad y la calidad del aire.
- **Cambio climático:** Algunos de estos gases dañinos, como los gases de efecto invernadero (por ejemplo, el dióxido de carbono y el metano), también contribuyen al cambio climático, lo que conlleva efectos secundarios en el

ambiente, como el aumento de las temperaturas, la acidificación de los océanos y el derretimiento de los glaciares.

- Accidentes químicos: La liberación inesperada y descontrolada de gases tóxicos, ya sea debido a accidentes industriales o desastres naturales, puede tener consecuencias graves, como evacuaciones forzadas, lesiones y pérdidas de vidas humanas.

Para hacer frente a la contaminación atmosférica causada por estos gases nocivos, se requiere la implementación de medidas para controlar y reducir la contaminación.

2.6.1. SENSOR MQ135

Los sensores MQ de gases son una categoría de sensores químicos. De acuerdo con la descripción proporcionada por la Unión Internacional de Química Pura y Aplicada (IUPAC), un sensor químico es un dispositivo que convierte información de naturaleza química relacionada con una muestra en una señal que resulta analíticamente beneficiosa (S. Nagy y otros, 2020).

La operación básica del sensor de gas MQ135 se basa en el uso de un material sensible, el SnO₂, que exhibe una menor conductividad cuando se encuentra en un ambiente sin contaminación. Sin embargo, cuando se detecta la presencia del gas específico que se desea analizar, la conductividad del sensor aumenta en proporción al incremento de la concentración del gas (Technology, 2015). Se presentan las siguientes imágenes que corresponden al comportamiento del MQ135.

Figura 5

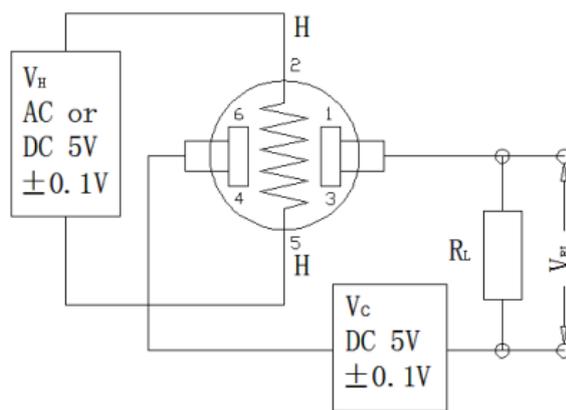
MQ135 sensor



Nota: (Technology, 2015)

Figura 6

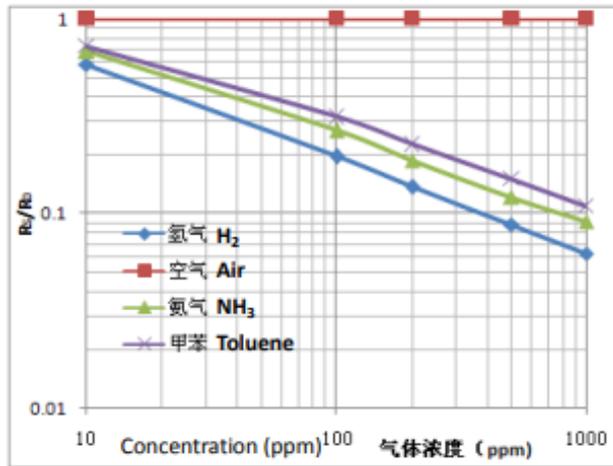
Circuito de Prueba MQ135



Nota: (Technology, 2015)

Figura 7

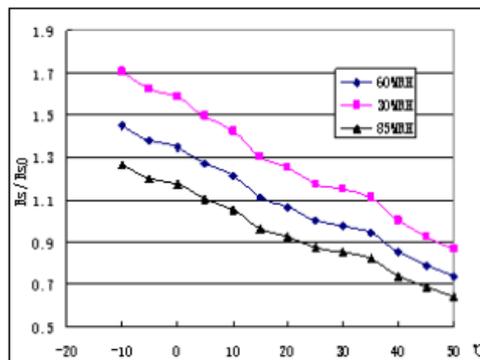
Curva de sensibilidad típica del sensor MQ135



Nota: (Technology, 2015)

Figura 8

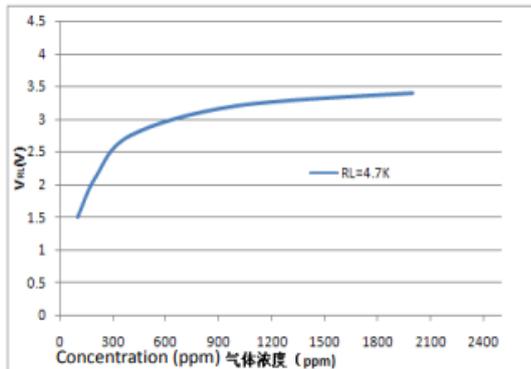
Características típicas de temperatura/humedad



Nota: (Technology, 2015)

Figura 9

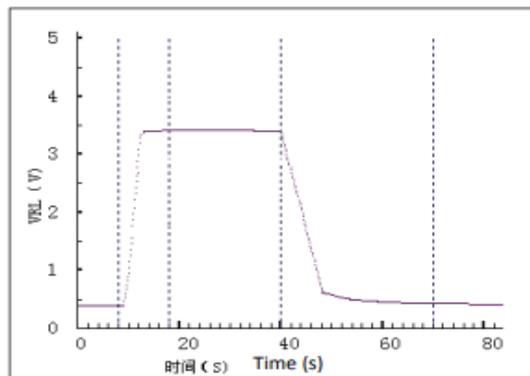
Curva de sensibilidad



Nota: (Technology, 2015)

Figura 10

Respuesta del sensor MQ135



Nota: (Technology, 2015)

El sensor de gas MQ-135 es conocido por ser sensible a varios gases, aunque su sensibilidad puede variar según las condiciones y la concentración de los gases.

Los principales gases que el MQ-135 puede detectar incluyen:

- Dióxido de carbono (CO₂)
- Amoníaco (NH₃)
- Vapores orgánicos volátiles (VOC)
- Humo y contaminantes del aire

2.7. SENSOR DS18B20

El sensor DS18B20 se destaca en proyectos de Arduino debido a su eficacia y precisión al medir la temperatura. Su facilidad de uso se refleja en la conexión a través de un solo cable digital, lo que simplifica su integración en diversos proyectos. Este enfoque de conexión se compone de tres cables: uno para la alimentación eléctrica, otro para la conexión a tierra y un tercero para transmitir datos mediante el protocolo OneWire, lo que asegura una comunicación efectiva con Arduino. Además de su sencillez, el sensor DS18B20 brinda la capacidad de ajustar la resolución de temperatura según las exigencias particulares del proyecto y permite conectar múltiples sensores, cada uno con una dirección única, facilitando así la creación de sistemas más complejos. La compatibilidad con librerías de Arduino, como "OneWire," simplifica aún más su implementación. Por su capacidad de funcionar en un amplio rango de temperaturas, el DS18B20 se presenta como una elección versátil para una variedad de aplicaciones que demandan mediciones precisas de temperatura, tales como sistemas de control de temperatura, termostatos y dispositivos de registro de datos climáticos. Su versatilidad y precisión le confieren un alto valor en el ámbito de la electrónica y la automatización (Mohamed & Dahoud, 2019).

Figura 11

Sensor DS18B20



Nota: (Mohamed & Dahoud, 2019)



2.8. CONVERSION ANALOGICO DIGITAL

Un conversor analógico-digital (ADC) es un dispositivo que transforma señales analógicas, como voltajes o corrientes, en datos digitales que una computadora u otro dispositivo digital puede comprender y procesar. Esto se logra tomando muestras de la señal analógica a intervalos regulares y asignándoles valores digitales que representan su amplitud. La precisión de la conversión depende de la resolución del ADC, que se expresa en bits. Cuanto mayor sea la resolución, mayor será la precisión en la representación de la señal analógica en formato digital. Los ADC son fundamentales en una amplia variedad de aplicaciones, desde la medición de sensores en electrónica hasta la grabación de audio digital y mucho más.

El fundamento de un conversor analógico-digital (ADC) se apoya en la teoría del muestreo y la cuantización, dos principios esenciales en el procesamiento de señales y la transformación de señales analógicas en digitales. A grandes rasgos, el proceso de conversión se lleva a cabo de la siguiente manera:

- **Muestreo:** En primera instancia, la señal analógica, que representa una magnitud continua y variable con el tiempo, es discretizada mediante el muestreo. Este procedimiento implica tomar muestras de la señal analógica en intervalos de tiempo regulares. El teorema de muestreo de Nyquist establece que la frecuencia de muestreo debe ser al menos el doble de la frecuencia máxima presente en la señal analógica para evitar la pérdida de información.
- **Cuantización:** Tras el muestreo, cada muestra se somete a la cuantización. Este proceso implica asignar un valor digital a cada muestra basándose en su amplitud. Para lograrlo, se divide el rango de amplitud de la señal en un número finito de niveles discretos. Cuanto mayor sea el número de bits empleados en la



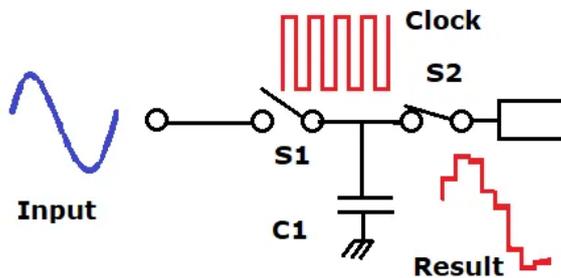
cuantización, mayor será la precisión de la conversión, ya que se dispondrán de más niveles discretos para representar la amplitud de la señal.

- **Codificación:** Una vez cuantizadas, las muestras se codifican en formato digital. Esto implica expresar cada valor cuantizado en un código binario, que se compone de una secuencia de 0s y 1s. La longitud de este código binario está determinada por la resolución del ADC, expresada en bits. Por ejemplo, en un ADC de 8 bits, cada muestra se codifica en una secuencia de 8 bits.
- **Proceso de conversión:** En pocas palabras, el proceso de conversión analógica a digital consiste en obtener muestras de una señal analógica, cuantizar cada muestra para asignarle un valor digital y finalmente codificar estos valores en un formato binario. El resultado es una representación digital discreta de la señal original, que puede ser procesada por dispositivos electrónicos, como microcontroladores o computadoras.

La elección de la frecuencia de muestreo, la resolución y otros parámetros del ADC depende de la aplicación específica y de la precisión requerida. La comprensión de estos fundamentos es esencial para una amplia gama de aplicaciones, desde la adquisición de datos en sistemas de medición hasta la conversión de señales analógicas en formatos digitales que pueden ser procesados y analizados por sistemas electrónicos (Waho, 2019).

Figura 12

Conversión analógica digital



Nota: (Waho, 2019)

2.9. CALIDAD DEL SUELO

El suelo es una capa de material que cubre la superficie terrestre y se forma debido a la interacción de factores geológicos, climáticos, biológicos y topográficos a lo largo del tiempo. Es un recurso natural vital para la vida en la Tierra, con un papel fundamental en los ecosistemas y la agricultura. Está compuesto por minerales, materia orgánica, agua, aire y microorganismos, y su composición varía ampliamente en términos de textura, estructura y composición química. Cumple diversas funciones, como proporcionar soporte a las plantas, participar en el ciclo de nutrientes, filtrar y purificar el agua, albergar microorganismos esenciales, almacenar carbono y servir como base para construcción e infraestructura. La calidad y salud del suelo son cruciales para la producción de alimentos, la conservación ambiental y el bienestar humano, por lo que su comprensión, conservación y gestión adecuada son imperativas para garantizar su sostenibilidad a largo plazo.

Cuando se habla de la calidad del suelo nos referimos a su capacidad para cumplir funciones específicas en un ecosistema o satisfacer necesidades humanas particulares. Se evalúa mediante múltiples parámetros, como la composición física, que incluye textura, estructura, retención de agua y permeabilidad; la composición química, que abarca

nutrientes, minerales, pH y concentraciones de sustancias químicas; la posible contaminación por metales pesados u otros tóxicos; la capacidad de retención y liberación de nutrientes para el crecimiento vegetal; la actividad biológica, relacionada con microorganismos y su contribución a la descomposición y formación de nutrientes; la erosión y degradación del suelo, influenciada por factores como la deforestación y la urbanización; y, por último, el tipo de uso previsto para el suelo, ya sea agricultura, jardinería, construcción o conservación. Mantener y evaluar la calidad del suelo es esencial para la producción de alimentos, la conservación ambiental y la planificación del uso de la tierra, ya que su degradación puede tener graves consecuencias para la productividad agrícola, la biodiversidad y la salud humana (Sharanappa, 2021).

Figura 13

Suelo de jardín



Nota: (Sharanappa, 2021)

2.10. CALIDAD DEL AMBIENTE

El término ambiente, también conocido como entorno o medio ambiente, hace referencia al conjunto de elementos y condiciones que rodean y ejercen influencia sobre un organismo o una comunidad de seres vivos. Estos componentes abarcan desde el aire, el agua y el suelo hasta la flora, la fauna, las actividades humanas y los factores climáticos,



geológicos y geográficos propios de un lugar específico. El ambiente constituye un sistema dinámico y complejo en el cual todos estos elementos interactúan, y desempeña un papel esencial en la vida en la Tierra al proporcionar los recursos necesarios para la supervivencia y el bienestar de los seres vivos. La preservación y el resguardo del ambiente son cruciales para mantener un equilibrio sostenible entre los sistemas naturales y las acciones humanas.

La calidad del ambiente hace referencia a la situación general y la salud de los componentes y las circunstancias que conforman el entorno natural que rodea a los organismos vivos. Esto abarca la calidad del aire, del agua, del suelo, así como la diversidad biológica presente en una región específica. Una alta calidad ambiental indica que estos elementos están en un estado saludable y equilibrado, lo cual beneficia tanto a los ecosistemas como a la salud de los seres humanos. Por el contrario, una baja calidad del ambiente se relaciona con la contaminación, la degradación del suelo, la pérdida de biodiversidad y otros factores que pueden tener consecuencias negativas en la vida en la Tierra. La preservación y mejora de la calidad del ambiente son metas esenciales para asegurar un entorno saludable y sostenible para las actuales y futuras generaciones (Dávila Dávila & Mena Dávila, 2022).

2.11. INTERNET DE LAS COSAS

Los dispositivos pueden adquirir capacidad de procesamiento de información a través de la integración de sensores, una posibilidad que ha sido habilitada gracias a la disponibilidad de chips de computadora económicos y la existencia de redes inalámbricas. Esto permite que los objetos sean capaces de recopilar y compartir datos de manera autónoma, sin requerir la intervención humana, lo que ha dado lugar al concepto de la Internet de las Cosas (IoT). Ejemplos de dispositivos IoT incluyen desde bombillas que



pueden ser controladas mediante una aplicación móvil hasta sensores de movimiento, termostatos inteligentes y farolas conectadas que ajustan su intensidad automáticamente en respuesta a las condiciones de luz solar. El IoT se aplica en una amplia variedad de contextos, desde dispositivos de entretenimiento para niños hasta vehículos autónomos, y proyectos destinados a crear ciudades inteligentes emplean sensores para supervisar y gestionar aspectos ambientales. Dispositivos portátiles como relojes inteligentes y pulseras de fitness también se consideran parte del universo del IoT. Aunque la tecnología subyacente al IoT se ha venido desarrollando desde las décadas de 1980 y 1990, su avance ha sido relativamente lento debido a limitaciones tecnológicas. No obstante, la reducción en el costo de los sensores y la conectividad a Internet ha permitido que la funcionalidad básica del IoT sea accesible prácticamente para cualquier tipo de objeto a un costo muy bajo. La adopción de IPv6, una versión más reciente del protocolo de direccionamiento en Internet, ha sido un requisito crucial para la expansión del IoT. El término "IoT" fue acuñado por Kevin Ashton en 1999, pero no fue hasta al menos una década después que la tecnología comenzó a materializar su visión de conectividad generalizada de dispositivos (Tovar Soto y otros, 2019).

En sus inicios, la Internet de las Cosas se centró en la comunicación entre máquinas, siendo principalmente relevante para empresas e industrias bajo la denominación de Máquina a Máquina (M2M). Sin embargo, en la actualidad, el enfoque se ha movido hacia la utilización de dispositivos inteligentes en hogares y entornos de oficina, lo que la hace pertinente para un público más amplio. A lo largo del tiempo, se exploraron diversas ideas para conectar objetos a Internet, como los "blogjects", la computación ubicua, la computación invisible y la computación generalizada, pero finalmente, fue la Internet de las Cosas la que logró consolidarse con éxito. Según las proyecciones de IDC, se espera que para el año 2025 haya una cifra considerable de

41.600 millones de dispositivos IoT, comúnmente conocidos como "cosas". La misma entidad señala que si bien los sistemas industriales y automotrices presentan una gran oportunidad para los dispositivos conectados, también anticipa una amplia adopción de dispositivos portátiles y dispositivos inteligentes para el hogar en un futuro cercano (Abdessamie, 2021).

Figura 14

Arquitectura del Internet de las cosas/



Nota: (Quiroga Montoya y otros, 2016)

2.12. ARDUINO

Arduino se presenta como una plataforma de código abierto que se utiliza de manera común en la elaboración de proyectos electrónicos interactivos, ofreciendo tanto una dimensión de hardware como de software. Esta plataforma se constituye a partir de una placa de circuito impreso que lleva incorporado un microcontrolador y un entorno de desarrollo integrado, lo cual simplifica notablemente tanto la programación como el control de dispositivos electrónicos.

El microcontrolador, que desempeña la función central en Arduino, se trata de un pequeño chip programable que actúa como el cerebro detrás de un proyecto. Arduino

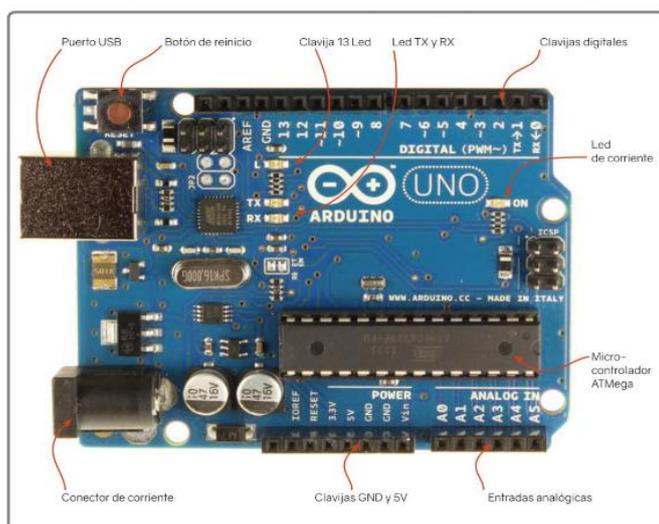
dispone de una amplia variedad de modelos de placas que incorporan diversos microcontroladores, siendo el Arduino Uno de los modelos más reconocidos y empleados.

La programación de Arduino se lleva a cabo mediante el uso del entorno de desarrollo de Arduino, el cual se basa en el lenguaje de programación C/C++. Destaca por su facilidad de aprendizaje y uso, lo que lo hace adecuado incluso para aquellos que están dando sus primeros pasos en el campo de la programación y la electrónica.

Los proyectos que pueden materializarse con Arduino abarcan una amplia gama de aplicaciones y pueden incluir desde el control de luces y motores hasta la creación de sistemas de domótica, dispositivos de monitoreo ambiental, robots y muchas otras opciones. Gracias a su carácter de código abierto, Arduino cuenta con una comunidad activa de usuarios que comparten proyectos, guías y recursos en línea, convirtiéndolo en una herramienta versátil y poderosa para la realización de proyectos electrónicos (Millahual , 2020).

Figura 15

Arduino UNO



Nota: (Millahual , 2020)

2.12.1. MICROCONTROLADOR

Un microcontrolador representa un dispositivo electrónico programable que concentra en un solo chip una unidad central de procesamiento (CPU), memoria, periféricos de entrada/salida y otros elementos esenciales necesarios para ejecutar aplicaciones específicas. Estos dispositivos se emplean en una amplia gama de aplicaciones que requieren control y automatización, como electrodomésticos, dispositivos médicos, sistemas de control automotriz, juguetes electrónicos, sistemas de control industrial y muchas otras (Marcillo Quimis, 2022).

Figura 16

Microcontrolador



Nota: (Marcillo Quimis, 2022)

Los microcontroladores se distinguen de las computadoras personales y otros sistemas informáticos más grandes por su tamaño más reducido, menor costo y capacidad de procesamiento adaptada a su propósito. Se diseñan para aplicaciones embebidas en las que se precisa un control preciso y específico, a menudo en tiempo real.

Los microcontroladores por lo general incorporan los siguientes componentes:



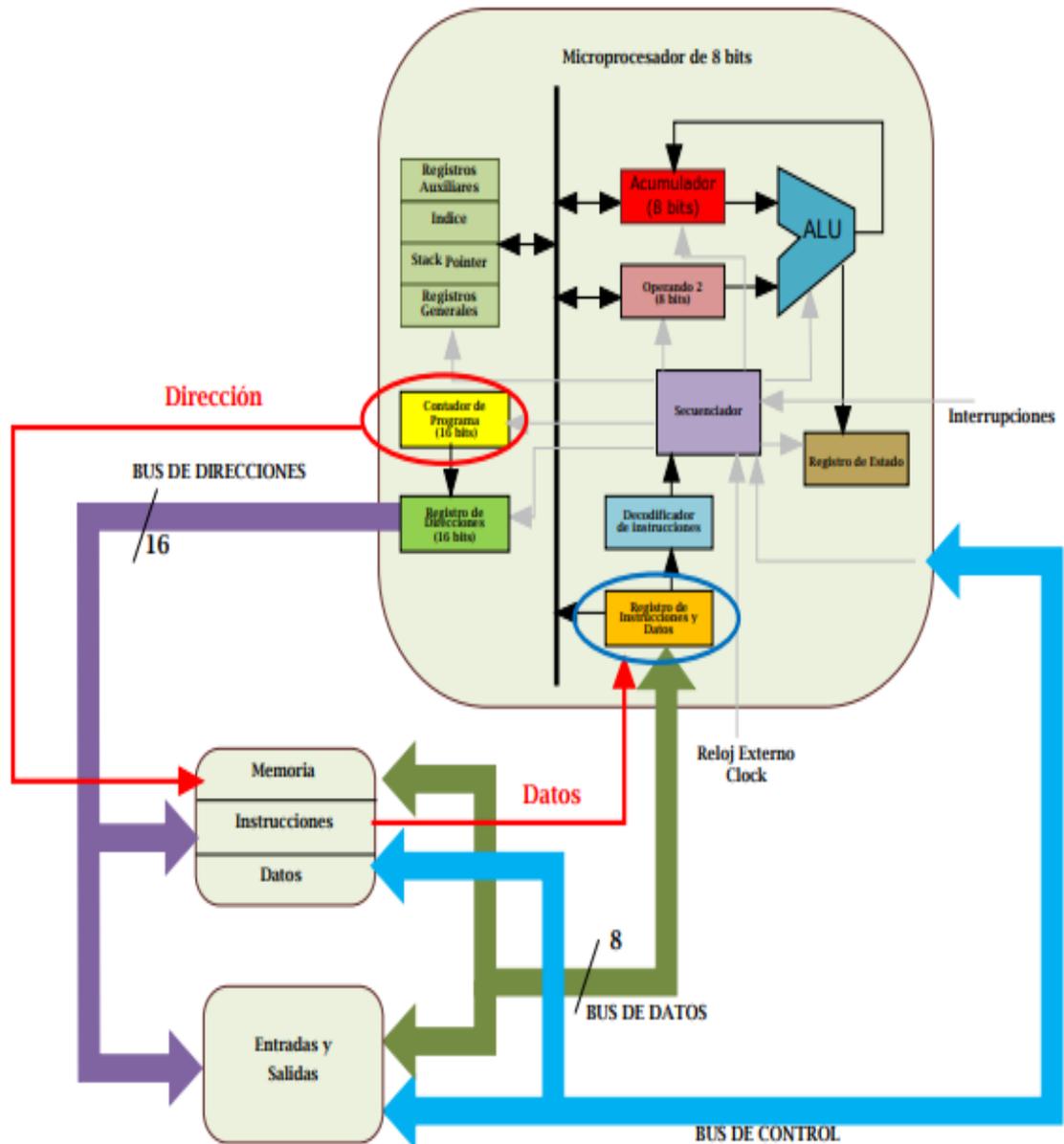
- **CPU:** La unidad central de procesamiento representa el núcleo del microcontrolador y ejecuta las instrucciones del programa.
- **Memoria:** Los microcontroladores disponen de memoria para almacenar el programa que rige su funcionamiento, así como datos temporales y permanentes. Esta memoria puede incluir flash, EEPROM (memoria de solo lectura programable y borrable eléctricamente) y RAM (memoria de acceso aleatorio).
- **Periféricos de E/S (Entrada/Salida):** Estos elementos permiten al microcontrolador interactuar con el entorno. Ejemplos de periféricos de E/S engloban pines GPIO (Entrada/Salida de Propósito General), convertidores analógico-digitales (ADC), convertidores digital-analógicos (DAC), temporizadores, puertos de comunicación (como UART, SPI e I2C) y otros.
- **Reloj:** Los microcontroladores integran un oscilador interno o externo que establece el ritmo de operación del dispositivo.
- **Unidad de control de interrupciones:** Esta unidad gestiona las interrupciones, permitiendo al microcontrolador responder a eventos externos o internos en tiempo real.
- **Interfaces de comunicación:** Estas interfaces habilitan la comunicación del microcontrolador con otros dispositivos o sistemas, como sensores, pantallas o sistemas de comunicación.

Los microcontroladores son programables, lo que implica que se les puede cargar un programa específico para llevar a cabo una tarea particular. Los lenguajes de programación usuales para microcontroladores abarcan C y ensamblador. Los

programadores de microcontroladores suelen emplear entornos de desarrollo integrados (IDE) para redactar, compilar y cargar programas en el microcontrolador.

Figura 17

Diagrama general de un microcontrolador de 8 bits



Nota: (Osio y otros, 2020)

La estructura interna y el diseño de hardware de un microcontrolador conforman su arquitectura, definiendo su funcionamiento. Esta arquitectura establece la forma en que las instrucciones se llevan a cabo, cómo se manejan los



datos y la interacción del microcontrolador con otros elementos en un sistema electrónico (Osio y otros, 2020).

2.12.2. ATMEGA328P

El ATmega328, un microcontrolador ampliamente utilizado y producido por la compañía Microchip Technology, es un componente común en varias placas Arduino, como el Arduino Uno. A continuación, se describen algunas de las características clave de este microcontrolador:

- **Arquitectura AVR:** El ATmega328 forma parte de la familia de microcontroladores AVR, conocida por su arquitectura de conjunto de instrucciones RISC (Reduced Instruction Set Computer). Esta arquitectura simplificada simplifica el proceso de programación y desarrollo de aplicaciones.
- **Frecuencia de Reloj:** Este microcontrolador es capaz de operar a velocidades de reloj de hasta 20 megahercios (MHz), lo que permite una rápida ejecución de las instrucciones.
- **Memoria Flash:** Ofrece una memoria flash de 32 kilobytes (KB) que almacena el programa de usuario. La capacidad de reprogramación de la memoria flash facilita las actualizaciones de software.
- **Memoria SRAM:** Incorpora 2 kilobytes (KB) de memoria SRAM que almacenan datos y variables temporales utilizadas durante la ejecución de programas.
- **Memoria EEPROM:** Incluye una memoria EEPROM de 1 kilobyte (KB) (Electrically Erasable Programmable Read-Only Memory) que permite el



almacenamiento no volátil de datos, lo que resulta útil para conservar configuraciones y datos importantes incluso cuando se corta la alimentación.

- **Periféricos de E/S (Entrada/Salida):** El ATmega328 cuenta con múltiples pines GPIO (Entrada/Salida de Propósito General) que se pueden configurar como entradas o salidas para conectar sensores, actuadores y otros dispositivos. También incorpora puertos de comunicación serie (UART) que facilitan la interacción con otros dispositivos.
- **Temporizadores y Contadores:** Incluye temporizadores y contadores integrados que pueden utilizarse para generar señales de temporización y medir intervalos de tiempo.
- **Convertidores Analógico-Digitales (ADC):** Dispone de un ADC de 10 bits que permite la conversión de señales analógicas a digitales, lo que resulta útil para la lectura de sensores analógicos.
- **Comunicación en Serie:** Admite diversos protocolos de comunicación en serie, como UART, SPI (Serial Peripheral Interface) e I2C (Inter-Integrated Circuit), lo que facilita la comunicación con otros dispositivos.
- **Alimentación:** Es capaz de funcionar con una amplia gama de tensiones de alimentación, generalmente entre 1.8V y 5.5V, lo que aumenta su versatilidad en términos de aplicaciones y fuentes de alimentación.
- **Bajo Consumo de Energía:** Ofrece modos de bajo consumo de energía que resultan útiles en aplicaciones con limitaciones energéticas, como dispositivos alimentados por batería.

- Programación y Depuración: Puede programarse mediante diversos programadores y entornos de desarrollo, incluido el Entorno de Desarrollo Integrado (IDE) de Arduino. Además, es compatible con herramientas de depuración y simulación que simplifican el proceso de desarrollo y depuración de código.

Figura 18

Microcontrolador ATmega328 AVR



Nota: (Marcillo Quimis, 2022)

2.13. LENGUAJE DE PROGRAMACION C++

/C++ es un lenguaje de programación de alto nivel con una sintaxis similar al inglés y se conoce ampliamente como una extensión del lenguaje C. Con C++, es posible crear una amplia variedad de aplicaciones, desde sistemas operativos hasta juegos y aplicaciones. Bjarne Stroustrup ideó este lenguaje en 1979 y posteriormente se estandarizó en 1998. Inicialmente, la visión de Stroustrup era agregar aspectos de programación orientada a objetos al lenguaje C existente, lo que llevó a que C++ se denominara en sus primeras etapas 'C con clases'. En 1983, se adoptó su nombre actual al agregar el operador ++ a C, lo que lo estableció como una extensión de C con características orientadas a objetos como clases, objetos, herencia y sobrecarga de operadores, entre otras. La primera versión del lenguaje se lanzó en 1985, pero las



características como las palabras clave "static" y "protected" se introdujeron en 1989. En 1990, se publicaron el manual de referencia de C++ y el compilador Turbo C++, lo que aceleró considerablemente su desarrollo. En 1998, el comité de estándares de C++ emitió la primera norma internacional, conocida como C++98. Luego, en 2011, se incorporaron más funcionalidades, como los bucles "for each", "auto", plantillas variadas y una nueva biblioteca de tiempo, en lo que se conoce como C++11. A pesar de los avances tecnológicos y la aparición de nuevos lenguajes de programación, C++ sigue siendo esencial en campos como la inteligencia artificial, el aprendizaje automático y el procesamiento de grandes cantidades de datos. Continúa siendo un lenguaje popular y de referencia para los desarrolladores, gracias a su capacidad para proporcionar un control preciso sobre los recursos del sistema y, cuando se utiliza de manera efectiva, un rendimiento excepcional en términos de velocidad. De hecho, se destaca como uno de los lenguajes de programación de más rápido crecimiento en la actualidad (Flores, 2022).

2.13.1. ENTORNO DE PROGRAMACIÓN ARDUINO

El lenguaje de programación principal utilizado en Arduino se basa en el lenguaje de programación C/C++. Sin embargo, se ha creado un conjunto de bibliotecas y funciones específicas de Arduino que simplifican la programación y el acceso a los periféricos de la placa Arduino, lo que lo hace más accesible para principiantes en la programación y la electrónica.

Aunque Arduino se basa en C/C++, la programación de Arduino suele ser más sencilla y legible que la programación en C/C++ estándar, debido a las bibliotecas y funciones predefinidas que facilitan la interacción con los componentes electrónicos.

Los usuarios de Arduino pueden escribir sus programas en el entorno de desarrollo integrado (IDE) de Arduino, que proporciona herramientas específicas para cargar el código en la placa Arduino y depurarlo. Esto hace que la programación de Arduino sea accesible para personas con diferentes niveles de experiencia en programación (Pérez Tavera, 2023).

Figura 19

Entorno de programación Arduino



Elaboración propia

2.14. SERVIDOR

Un servidor en la red de Internet es una robusta máquina o sistema de hardware destinado a almacenar y administrar datos, recursos y servicios, y se encuentra diseñado para responder a las solicitudes de otros dispositivos o usuarios conectados a través de una red. Su función principal radica en facilitar el acceso a servicios tales como páginas web, correo electrónico, archivos o bases de datos para usuarios remotos, desempeñando un papel central en el intercambio de información en la World Wide Web y en otras redes. Además de asegurar la disponibilidad y el rendimiento de estos servicios, los servidores



también tienen la responsabilidad de gestionar la seguridad y facilitar la comunicación de datos en línea (Bernstein, 2018).

El Internet de las Cosas (IoT) y los servidores están estrechamente relacionados en la infraestructura tecnológica que permite el funcionamiento del IoT. En el IoT, diversos dispositivos y sensores están conectados a Internet y recopilan datos del entorno o realizan acciones específicas (Smeenk, 2023). La relación entre el IoT y los servidores se puede explicar de la siguiente manera:

- **Recopilación de datos:** Los dispositivos de Internet de las Cosas (IoT) generan una gran cantidad de información, que incluye datos de sensores, métricas de uso y otros datos relevantes. Esta información se transmite a servidores con el propósito de ser procesada y almacenada, ya sea en entornos en la nube o en redes locales.
- **Almacenamiento y gestión de datos:** La responsabilidad de almacenar y administrar eficientemente los datos generados por los dispositivos IoT recae en los servidores. Esto implica la creación de bases de datos para organizar la información, realizar copias de seguridad y garantizar la disponibilidad de los datos cuando sea necesario.
- **Procesamiento y análisis:** Los servidores también desempeñan un papel crítico en el procesamiento y análisis de los datos provenientes del IoT. Tienen la capacidad de ejecutar algoritmos y aplicaciones analíticas para extraer información significativa de los datos recopilados, lo que permite tomar decisiones basadas en datos y detectar patrones o tendencias relevantes.
- **Gestión de dispositivos:** Los servidores pueden supervisar y administrar de manera remota los dispositivos IoT, lo que incluye la capacidad de configurar



parámetros, actualizar el firmware, monitorear el estado y recibir comandos de control desde un punto central.

- Seguridad: En el contexto del entorno IoT, los servidores tienen una función esencial en garantizar la seguridad. Se encargan de proteger los datos confidenciales y gestionar los procesos de autenticación y autorización para los dispositivos y usuarios que acceden a la red IoT.
- Interconexión: Los servidores permiten la conexión de los dispositivos IoT y su integración con otros sistemas y servicios. Esto es fundamental para lograr la interoperabilidad en un diverso ecosistema IoT.



CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. LUGAR DE ESTUDIO

En esta investigación se consideraron 10 parques de la ciudad de Puno ubicada en la sierra sudeste del país, cuyas coordenadas geográficas de Puno, Perú, en grados decimales Longitud: -70.0199000 y latitud: -15.8422000.

3.2. UBICACIÓN DE LA INVESTIGACIÓN

Esta investigación se realizó en la región Puno, en la ciudad de Puno, el cual posee las siguientes coordenadas geográficas: 15°50'36"S 70°01'25"O a 3818msnm. Para la investigación se tomó referencia los parques ubicados en los diferentes barrios de dicha ciudad.

3.3. DISEÑO DE LA INVESTIGACIÓN

En esta investigación, se realizará un estudio experimental con el propósito de implementar los principios científicos y la lógica en un prototipo de sistema de adquisición de datos. El análisis de este prototipo está relacionado con variables supervisadas por el sistema físico instalado en uno de los parques de la ciudad de Puno.

3.4. TIPO DE INVESTIGACIÓN

Se realizó una investigación experimental con el propósito de abordar las necesidades relacionadas con el problema en cuestión. Se elaboró una hipótesis predictiva y se diseñó un experimento destinado a ponerla a prueba. Este experimento se desarrolló en un entorno controlado o en un laboratorio, con el fin de evaluar tanto la hipótesis como el sistema propuesto.



3.5. MATERIALES EMPLEADOS

3.5.1. Hardware

- Laptop
- Arduino
- Microcontroladores
- Higrómetro
- Sensores
- Conversor analógico digital

3.5.2. Software

- Windows 10
- Microsoft office
- Google drive
- Arduino IDE
- Lenguaje de programación C++
- World Wide Web

3.6. TÉCNICAS PARA LA ADQUISICIÓN DE DATOS

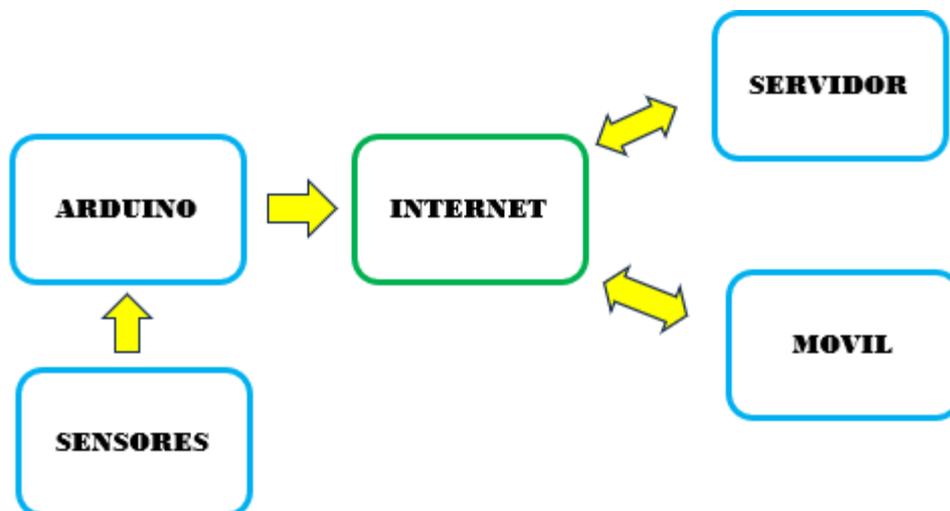
La adquisición de información o señales representa una fase crítica en el procesamiento de datos en diversas áreas, desde la medicina hasta la ingeniería. Este proceso involucra la recolección de muestras del entorno físico, como sonidos, imágenes, señales eléctricas y otros tipos de información analógica que son cuantificables, y su

conversión a un formato digital que pueda ser interpretado por dispositivos electrónicos. El propósito principal de la captura de señales es generar datos susceptibles de ser analizados, procesados y almacenados. Para lograr esto, se hacen uso de una variedad de sensores y transductores que convierten las señales analógicas en señales digitales que pueden ser manipuladas por una computadora. Los datos obtenidos tienen aplicaciones que van desde análisis, toma de decisiones, control de procesos hasta respuestas en tiempo real. La captura de señales es una tarea compleja que exige un conjunto de habilidades y conocimientos técnicos. Es fundamental contar con una sólida comprensión de los principios físicos subyacentes en las señales que se están midiendo, así como una familiaridad profunda con los diversos tipos de sensores y transductores disponibles y cómo se emplean. En resumen, la captura de señales es un proceso esencial en la recopilación de información y datos en múltiples campos, y desempeña un papel crucial en el éxito del análisis de datos y la toma de decisiones basadas en datos.

La arquitectura diseñada para este proyecto de Internet de las cosas se ve en la siguiente imagen:

Figura 20

Arquitectura de la red de Internet de las Cosas



Elaboración propia

Los sensores están conectadas a internet.

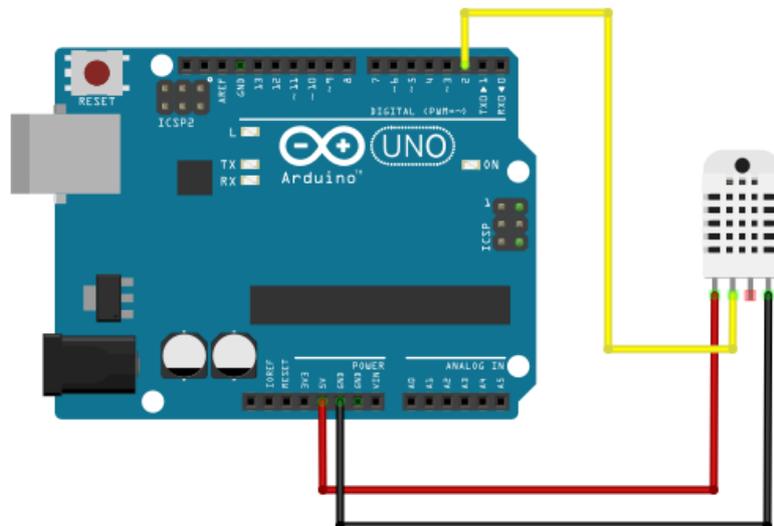
3.6.1. SENSOR DHT22

Este sensor conocido también como AM2302, es un dispositivo utilizado para medir la temperatura y la humedad relativa en el entorno. Las conexiones y el voltaje de funcionamiento son de la siguiente manera:

- Pin de Alimentación de 5V o conocido como VCC
- Pin de Datos, este va conectado a un pin GPIO del Arduino
- Tierra o conocido como GND

Figura 21

Conexión Arduino uno con sensor DHT



Elaboración propia

El monitor serial en Arduino es una herramienta que permite la comunicación entre el Arduino y la computadora a través del puerto serie (también conocido como puerto COM en Windows). Este monitor serial es una ventana de visualización de texto en la que puedes ver y enviar datos entre el Arduino y la computadora durante la ejecución de tu programa.

Es en el monitor serial donde se tiene que especificar los baudios o la velocidad de transmisión entre el Arduino y la computadora, en ambas deben estar configuradas a la misma velocidad, caso contrario encontraremos errores al enviar datos.

Al tener abierto el arduino IDE en nuestro computador, podemos ver el monitor serial en la barra de herramientas, para mejor entendimiento se presenta en la siguiente imagen el monitor serial.

Figura 22

Pruebas del sensor DHT22 en el monitor serial

```
sketch_sep18d | Arduino IDE 2.1.2-nightly-20230717
File Edit Sketch Tools Help
sketch_sep18d.ino
1 #include <DHT.h>
2 #define DHTPIN 2 // Define el pin al que está conectado el sensor DHT22
3 #define DHTTYPE DHT22 // Selecciona el tipo de sensor (DHT11 o DHT22)
4 DHT dht(DHTPIN, DHTTYPE);
5 void setup() {
6   Serial.begin(9600);
7   dht.begin();
8 }
9 void loop() {
10  delay(2000);
11  float humedad = dht.readHumidity();
12  float temperatura = dht.readTemperature();
13  if (isnan(humedad) || isnan(temperatura)) {
14    Serial.println("Error al leer el sensor DHT");
15  } else {
16    Serial.print("Humedad: ");
17    Serial.print(humedad);
18    Serial.print("% - ");
19    Serial.print("Temperatura: ");
20    Serial.print(temperatura);
21    Serial.println("°C");
22  }
23 }
Output Serial Monitor x
Message (Enter to send message to Arduino Uno on COM9)
Humedad: 24.20% - Temperatura: 29.30°C
Humedad: 25.00% - Temperatura: 29.20°C
Humedad: 25.00% - Temperatura: 29.20°C
Humedad: 24.90% - Temperatura: 29.20°C
Humedad: 24.90% - Temperatura: 29.20°C
```

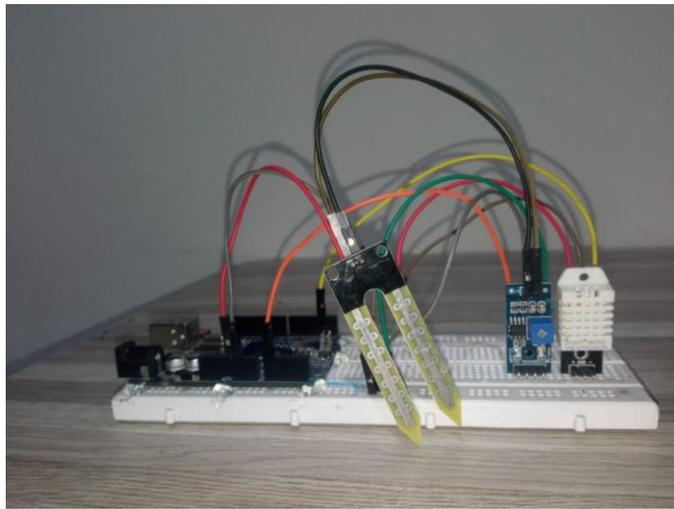
Elaboración propia

Los datos obtenidos del sensor DHT22 se muestran en el monitor serial. La peculiaridad de este sensor es que obtenemos dos valores como lo son: Temperatura y Humedad del ambiente.

La conexión física se hizo en protoboard, se ve en la siguiente imagen.

Figura 23

Conexión física del sensor DHT22



Elaboración propia

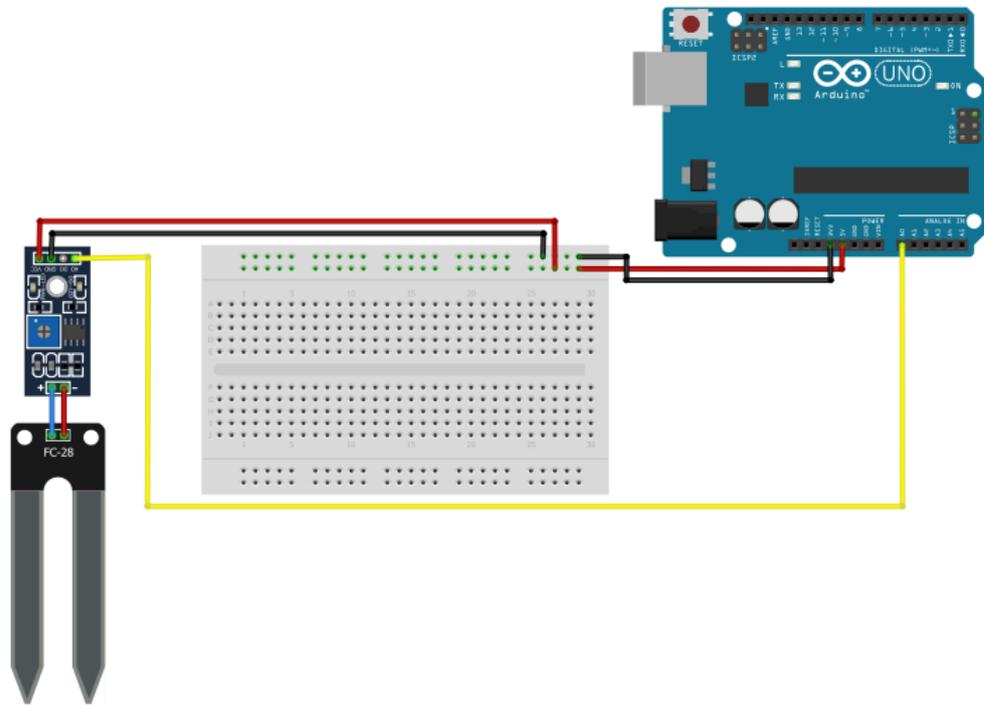
3.6.2. SENSOR FC-28

El FC – 28 es conocido como el sensor de detección de humedad de suelo, es muy utilizado en proyectos de automatización, especialmente en agricultura y la jardinería. El principio del funcionamiento se basa en sondas de detección y un módulo de amplificación de señales. Estas sondas son las que se insertan al suelo, y el módulo de amplificación se conecta al Arduino y esta a su vez procesa la señal para mostrar el porcentaje de humedad que existe en el suelo.

El diagrama de conexión de este sensor se muestra en la siguiente imagen:

Figura 24

Diagrama de conexión del sensor FC-28



Elaboración propia

Las señales analógicas son aquellas que pueden variar de manera continua en un rango específico de valores. A diferencia de las señales digitales, que solo pueden tomar valores discretos, las señales analógicas pueden tener un valor en cualquier punto dentro de su rango. Es en nuestro proyecto que el Arduino toma señales analógicas para este sensor.

El rango de valores digitales de 0 a 1023 se refiere al número de pasos discretos en los que se divide una señal analógica cuando se convierte en una señal digital. Esto se llama resolución. En el caso de Arduino, se utilizan 10 bits para representar la lectura analógica, lo que da como resultado $2^{10} = 1024$ valores discretos posibles.

Figura 25

Prueba del sensor de humedad de suelo

```
sketch_sep18a | Arduino IDE 2.1.2-nightly-20230717
File Edit Sketch Tools Help
sketch_sep18a.ino
1 int sensorPin = A0;
2 int sensorValue;
3 int porcentaje;
4 void setup() {
5   Serial.begin(9600);
6 }
7 void loop() {
8
9   sensorValue = analogRead(sensorPin);
10  porcentaje = map(sensorValue, 0, 1023, 0, 100);
11  Serial.print("Humedad del suelo: ");
12  Serial.print(porcentaje);
13  Serial.println("%");
14  delay(1000);
15 }
16 }

Output Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM9')
Humedad del suelo: 72%
```

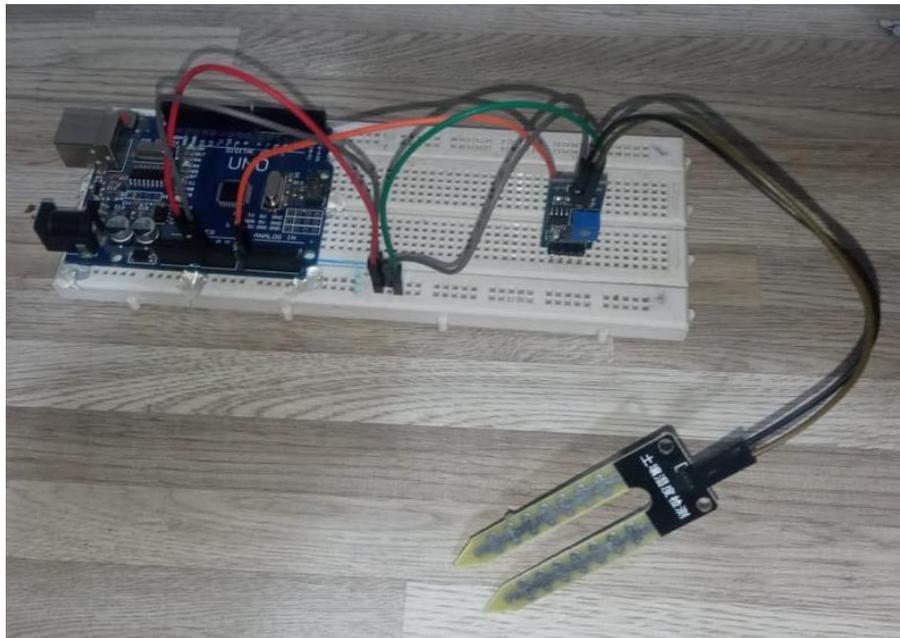
Elaboración propia

En la imagen anterior se presenta los resultados de las primeras mediciones de humedad, el monitor serial muestra el porcentaje de humedad que presenta el suelo.

La conexión física del sensor FC-28, se ve en la siguiente imagen.

Figura 26

Conexión física del sensor FC-28



Elaboración propia

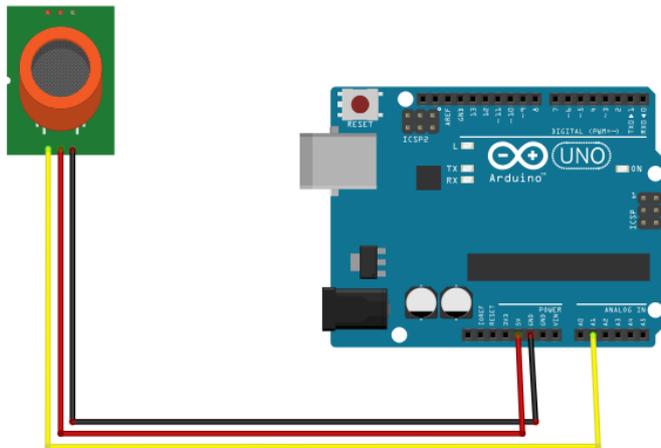
3.6.3. SENSOR MQ135

El sensor MQ-135 mide la calidad de aire y se usa para detectar gases tóxicos en el aire. Entre los gases que este puede detectar se encuentra:

- Dióxido de carbono (CO₂)
- Monóxido de carbono (CO)
- Amoníaco (NH₃)
- Metano (CH₄)
- Compuestos orgánicos volátiles (COV)

Figura 27

Conexión de Arduino con el MQ - 135



Elaboración propia

El sensor MQ – 135 va conectado a una entrada analógica del Arduino.

Figura 28

Prueba del censor MQ 135

```
sketch_sep19a.ino
1 int sensorpin_mq135 = A2; // Pin analógico al que está conectado el sensor MQ-135
2
3 void setup() {
4   Serial.begin(9600);
5 }
6
7 void loop() {
8   int sensorValue = analogRead(sensorpin_mq135);
9   int porcentaje_mq135= map(sensorValue,0,1023,0,100);
10  Serial.print("Valor del sensor: ");
11  Serial.print(porcentaje_mq135);
12  Serial.println(" %");
13  delay(1000);
14 }
15
16
17
18
```

Output Serial Monitor x

Message (Enter to send message to 'Arduino Uno' on 'COM3')

Valor del sensor: 7 %
Valor del sensor: 7 %

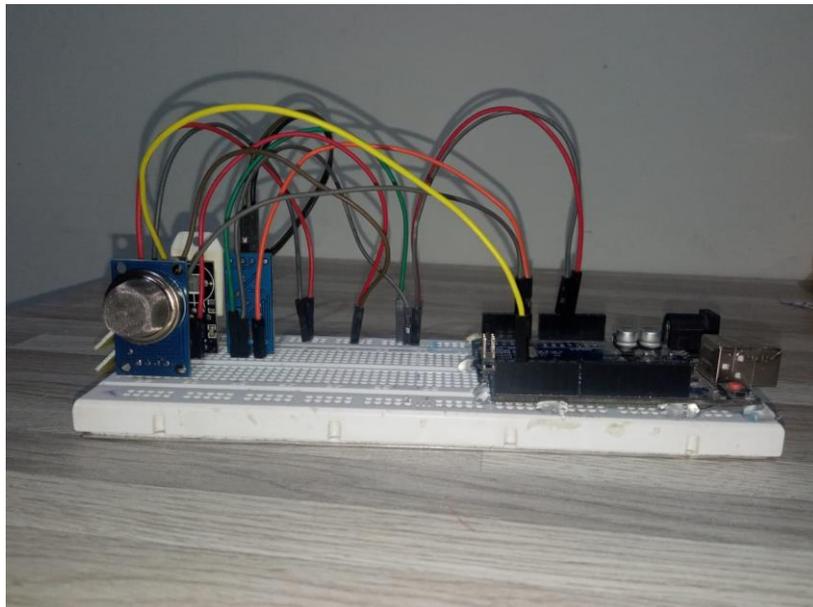
Elaboración propia

Las pruebas realizadas con el sensor MQ135 dio resultados en el monitor serial, el cual se observa en la imagen anterior.

La conexión física la podemos observar en la siguiente imagen.

Figura 29

Conexión física del sensor MQ - 135



Elaboración propia

3.6.4. SENSOR DS1820

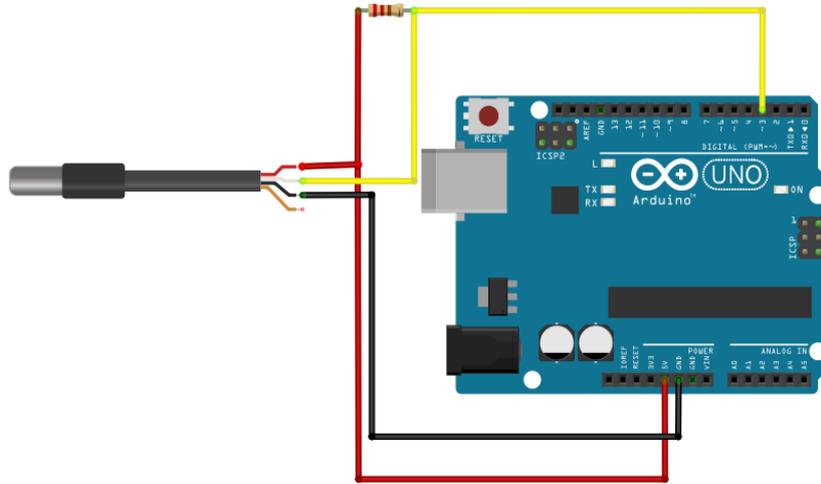
El DS18B20 es un sensor de temperatura digital de alta precisión que se utiliza comúnmente en proyectos de electrónica y automatización. Es capaz de medir temperaturas en un amplio rango y proporciona lecturas digitales precisas.

Se presentan algunas características importantes:

- Tiene alta precisión, con una tolerancia de +/- 5V
- Conexión por un solo cable
- Rango de temperatura de -55°C a +125°C
- Cuenta con un identificador de 64 bits
- Funciona con comunicación digital

Figura 30

Conexión del sensor DS18B20

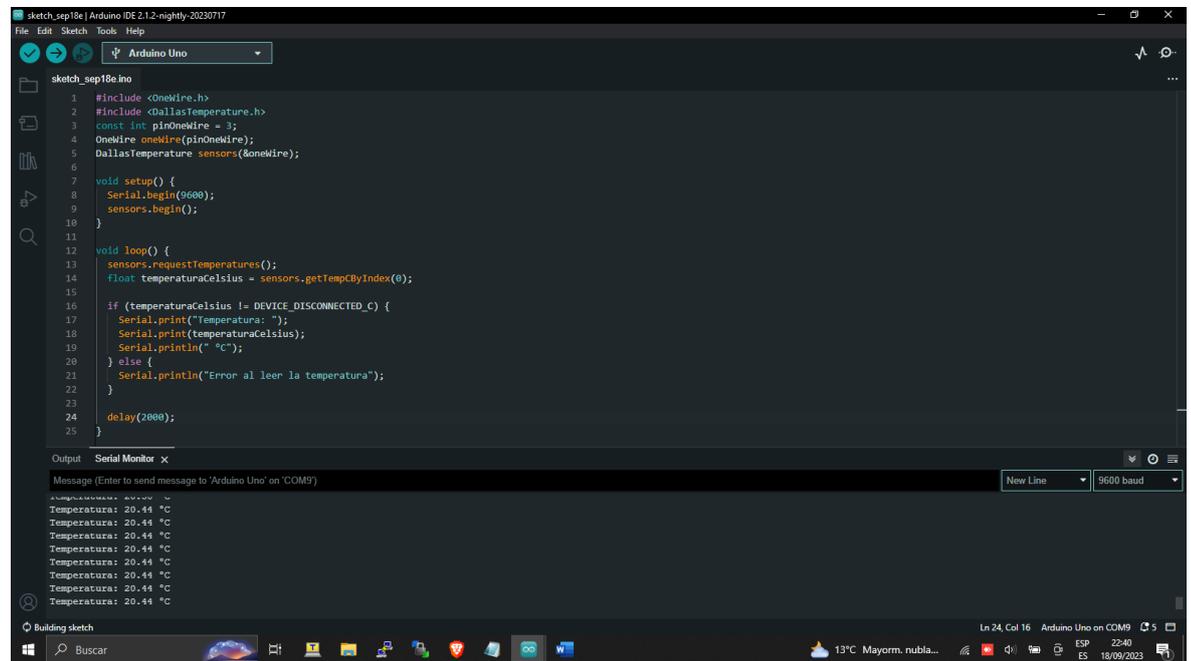


Elaboración propia

Los datos se envían del sensor al pin digital número 3 de Arduino.

Figura 31

Datos obtenidos del sensor DS18B20 en el monitor serial



Elaboración propia

3.6.5. CÓDIGO ARDUINO

El código de cada sensor es único, pero necesitamos que nuestro prototipo lea todos los valores al mismo tiempo, es por ello que se ha unificado y optimizado el código.

Las bibliotecas (también conocidas como "libraries" en inglés) son conjuntos de código preescrito que facilitan la programación y la interacción con hardware y sensores. Estas bibliotecas están diseñadas para simplificar tareas comunes y permiten a los usuarios de Arduino utilizar funciones predefinidas en lugar de tener que escribir todo el código desde cero. Las librerías que se usarán en esta investigación se pueden ver en la siguiente imagen:

Figura 32

Librerías usadas para leer los sensores

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

Elaboración propia

La biblioteca OneWire se emplea para establecer comunicación con dispositivos que utilizan el protocolo de comunicación OneWire. Este protocolo permite la transmisión digital de datos a través de un solo cable de datos. En el proyecto es usada para operar el sensor temperatura DS18B20, que se comunican mediante el protocolo OneWire.

La biblioteca DallasTemperature es un complemento a OneWire y está diseñada específicamente para trabajar con sensores de temperatura pertenecientes a la familia de dispositivos de Dallas Semiconductor, como el DS18B20.



La biblioteca DHT se emplea para interactuar con sensores de humedad y temperatura de la serie DHT, como el DHT11 y el DHT22. Esta biblioteca simplifica la obtención de datos de estos sensores y proporciona funciones para obtener la temperatura y la humedad relativa.

Las variables en Arduino, como en la mayoría de los lenguajes de programación, son recipientes utilizados para resguardar y manipular datos dentro de un programa. Estas variables varían en tipo y se escogen en función del tipo de información a almacenar y la manera en que se utilizará en el programa. A continuación, se presenta una breve descripción de los tipos de variables más comunes en Arduino:

- Variables Enteras (int): Las variables enteras sirven para almacenar números completos, ya sean positivos, negativos o cero.
- Variables de Punto Flotante (float o double): Las variables de punto flotante se emplean para guardar números que incluyen decimales.
- Variables de Texto (String): Las variables de tipo String se utilizan para almacenar texto o secuencias de caracteres.
- Variables Booleanas (bool): Las variables booleanas solo pueden adquirir dos valores: verdadero (true) o falso (false). Se emplean para representar condiciones lógicas.
- Variables de Caracteres (char): Las variables de tipo char se usan para contener un solo carácter. Por ejemplo, char letra = 'A'; almacena el carácter 'A' en la variable "letra".

- Variables de Tipo Byte (byte): Las variables de tipo byte se utilizan para guardar valores numéricos en un rango de 0 a 255. Son útiles cuando se trabajan con valores limitados y se desea ahorrar memoria.
- Arrays (Arreglos): Los arrays son colecciones de variables del mismo tipo que se almacenan en una secuencia.

La siguiente imagen muestra las variables que se usan en el proyecto.

Figura 33

Declaración de variables

```
#define DHTTYPE DHT22
const int DHTPIN = 2;
const int pinOneWire = 3;
DHT dht(DHTPIN, DHTTYPE);
int FC28 = A0;
int sensorpin_mq135 = A1;
int sensorValue_FC28;
int porcentaje_FC28;
OneWire oneWire(pinOneWire);
DallasTemperature sensors(&oneWire);
```

Elaboración propia

Se describe brevemente las líneas del código anterior:

- #define DHTTYPE DHT22: En esta línea, defines el tipo de sensor DHT que estás utilizando como DHT22. Esto se utiliza más adelante en el código para configurar el sensor.
- const int DHTPIN = 2: Esta línea establece el número de pin al que está conectado el sensor DHT. En este caso, está conectado al pin 2.
- const int pinOneWire = 3: Esta línea define el número de pin al que está conectado el bus OneWire, que se utiliza para comunicarse con dispositivos que siguen el protocolo OneWire. En este caso, está conectado al pin 3.

- DHT dht(DHTPIN, DHTTYPE): Aquí se crea una instancia de la clase DHT utilizando el número de pin definido anteriormente y el tipo de sensor DHT que es DHT22. Esto prepara el sensor para su uso posterior.
- int FC28 = A0: Se define una variable llamada FC28 con el propósito de asignar el pin de lectura en el sensor de humedad de suelo.
- int sensorpin_mq135 = A1: Similar al paso anterior, se define una variable llamada sensorpin_mq135 y se le asigna el valor del pin analógico A1.
- int sensorValue_FC28: Se declara una variable llamada sensorValue_FC28 que se utilizará más adelante para almacenar el valor leído del sensor asociado al pin A0.
- OneWire oneWire(pinOneWire): Se crea una instancia de la clase OneWire utilizando el número de pin definido anteriormente (pinOneWire).
- DallasTemperature sensors(&oneWire): Se crea una instancia de la clase DallasTemperature, que se utiliza para interactuar con dispositivos de temperatura de la familia Dallas Semiconductor.

El método `setup()` se ejecuta una sola vez al inicio del programa y se utiliza generalmente para realizar configuraciones iniciales y preparar el entorno antes de que comience la ejecución del bucle principal. Veamos la siguiente imagen:

Figura 34

Void setup del código

```
void setup() {  
  Serial.begin(9600);  
  dht.begin();  
  sensors.begin();  
}
```

Elaboración propia



Respecto a las líneas de código, la explicación es la siguiente:

- `Serial.begin(9600)`: Esta línea configura la comunicación serie (UART) a una velocidad de baudios de 9600. En otras palabras, establece la velocidad de transmisión de datos para la comunicación entre Arduino y un dispositivo externo, como una computadora a través del puerto serie. Esta velocidad debe coincidir con la velocidad configurada en el monitor serie de Arduino o en el software de comunicación serie en la computadora para que la comunicación sea exitosa.
- `dht.begin()`: Esta línea inicializa un sensor DHT que se ha configurado previamente en el código. La función `begin()` se utiliza para iniciar la comunicación con el sensor DHT y prepararlo para realizar lecturas de temperatura y humedad.
- `sensors.begin()`: Esta línea inicializa el bus OneWire y prepara el Arduino para comunicarse con dispositivos que siguen el protocolo OneWire.

Luego del `setup` viene el método `loop()` el cual es uno de los componentes clave de un programa de Arduino. Este método se ejecuta de forma continua después de que el método `setup()` se haya ejecutado una sola vez al inicio del programa. El `loop()` es responsable de controlar las operaciones y las acciones que se deben realizar de manera repetitiva mientras el Arduino está en funcionamiento

Figura 35

Void loop de código Arduino

```
void loop() {  
  
    delay(1000);  
  
    float hum_env = dht.readHumidity();  
    float temp_env = dht.readTemperature();  
    Serial.print("Humedad Ambiente: ");  
    Serial.print(hum_env);  
    Serial.print("% - ");  
    Serial.print("Temperatura Ambiente: ");  
    Serial.print(temp_env);  
    Serial.print("°C");  
  
    sensors.requestTemperatures();  
    float temp_land = sensors.getTempCByIndex(0);  
    Serial.print(" - Temperatura Suelo: ");  
    Serial.print(temp_land);  
    Serial.print(" °C");  
  
    int sensorValue = analogRead(sensorpin_mq135);  
    int per_mq135= map(sensorValue,0,1023,0,100);  
    Serial.print(" - Gases Tóxicos: ");  
    Serial.print(per_mq135);  
    Serial.print(" %");  
  
    sensorValue_FC28 = analogRead(FC28);  
    int per_FC28 = map(sensorValue_FC28, 0, 1023, 100, 0);  
    Serial.print(" - Humedad del suelo: ");  
    Serial.print(per_FC28);  
    Serial.println("%");  
}
```

Elaboración propia

Detallamos el código:

- El primer bloque conformado por `float hum_env = dht.readHumidity();` y `float temp_env = dht.readTemperature();` leen la humedad y la temperatura del ambiente utilizando un sensor DHT22 y almacenan los resultados en las variables `hum_env` y `temp_env`, respectivamente. En seguida las líneas

imprimen la humedad y la temperatura ambiental en el Monitor Serie utilizando Serial.print(). Se incluyen etiquetas descriptivas para identificar los datos.

- El segundo bloque conformado por sensors.requestTemperatures() solicita una lectura de temperatura a través de los dispositivos conectados al bus OneWire, configurados previamente con la biblioteca DallasTemperature. Tenemos también a float temp_land = sensors.getTempCByIndex(0) que lee la temperatura del suelo desde el primer dispositivo en el bus OneWire y almacena el resultado en la variable temp_land.
- El tercer bloque imprime los gases tóxicos y mapea según a los 10 bitad que Arduino manipula en sus pines.
- El cuarto bloque imprime la temperatura del suelo y también son mapeados según los 10 bits de arduino.

Para llevar a internet nuestros datos obtenidos, usamos un código adicional donde se incluyen librerías SPI y Ethernet.

Figura 36

Librerías para datos en internet

```
#include <SPI.h>  
#include <Ethernet.h>
```

Elaboración propia

#include <SPI.h>: La biblioteca SPI se utiliza para comunicarse con dispositivos que siguen el protocolo SPI, como sensores, pantallas, tarjetas de memoria y otros dispositivos de hardware que utilizan una interfaz serial síncrona. La biblioteca proporciona funciones y objetos que facilitan la comunicación a través de SPI. En nuestra investigación se usa para comunicación con el shield ethernet.

`#include <Ethernet.h>`: La biblioteca Ethernet se utiliza para habilitar la comunicación de red a través de una conexión Ethernet con el Arduino. Permite que el Arduino actúe como un cliente o un servidor en una red Ethernet y ofrece funciones y objetos para configurar la comunicación y transmitir datos a través de Ethernet.

El siguiente bloque permite identificar a nuestra red, se configuran parámetros como Gateway, IP, etc.

Figura 37

Configuración de IP, Gateway, dns, Ipserver

```
byte mac[] = { 0x54, 0xAB, 0x3A, 0x5B, 0xAD, 0x05 };
IPAddress server(108,62,123,70);
IPAddress ip(172,16,100,133);
byte gateway[] = { 172, 16, 100, 1 };
byte subnet[] = { 255, 255, 255, 0 };
byte myDns[] = { 8, 8, 8, 8 };
EthernetClient client;
bool printWebData = true;
char sensor_id[] = "JEH-01";
```

Elaboración propia

Detallamos a continuación:

- `byte mac[] = { 0x54, 0xAB, 0x3A, 0x5B, 0xAD, 0x05 };`: Esta línea define la dirección MAC (Media Access Control) del módulo Ethernet de Arduino. Cada dispositivo Ethernet debe tener una dirección MAC única para identificarse en una red. Esta dirección es en formato hexadecimal y se asigna manualmente.
- `IPAddress server (108,62,123,70)`: Aquí se especifica la dirección IP del servidor con el que el Arduino se comunicará. En este caso, la dirección IP del servidor es "108.62.123.70".



- `IPAddress ip (172,16,100,133)`: Esta línea configura la dirección IP que se asignará al Arduino en la red Ethernet. Es importante que esta dirección sea única dentro de la red en la que se encuentra el Arduino.
- `byte gateway [] = {172, 16, 100, 1}`: Se define la dirección IP del enrutador o gateway de la red Ethernet. El enrutador se utiliza para dirigir el tráfico de red fuera de la red local hacia otros destinos en la Internet.
- `byte subnet[] = {255, 255, 255, 0}`: Aquí se establece la máscara de subred que define cuál parte de la dirección IP corresponde a la red local y cuál parte corresponde a los dispositivos dentro de esa red. En este caso, es una máscara de subred típica de 255.255.255.0, que indica que los primeros tres octetos de la dirección IP forman la red local y el último octeto se utiliza para identificar dispositivos en esa red.
- `byte myDns[] = { 8, 8, 8, 8 }`: Se configura la dirección IP del servidor DNS que el Arduino utilizará para resolver nombres de dominio en direcciones IP. En este caso, se utiliza la dirección IP del servidor DNS de Google.
- `EthernetClient client`: Se crea una instancia de la clase `EthernetClient` que se utilizará para establecer una conexión de cliente Ethernet con un servidor remoto.
- `bool printWebData = true`: Se declara una variable booleana llamada `printWebData` que se utiliza para controlar si se imprimirán datos recibidos de un servidor web en el Monitor Serie. En este caso, se establece en `true`, lo que significa que los datos se imprimirán.

- `char sensor_id[] = "JEH-01"`: Aquí se define una cadena de caracteres llamada `sensor_id` que parece ser una identificación única para el sensor o dispositivo conectado nuestro Arduino.

Antes de enviar datos a internet, tenemos que asegurarnos que haya conexión con el servidor, y en caso de que, si hubiera, los datos de todos los sensores se envían y podremos visualizar desde un celular o una PC.

Figura 38

Código que posibilita el envío de datos a internet

```
void loop(){
  delay(1000);
  float hum_env = dht.readHumidity();
  float temp_env = dht.readTemperature();
  sensors.requestTemperatures();
  float temp_land = sensors.getTempCByIndex(0);
  int sensorValue = analogRead(sensorpin_mq135);
  int per_mq135 = map(sensorValue,0,1023,0,100);
  sensorValue_FC28 = analogRead(FC28);
  int per_FC28 = map(sensorValue_FC28, 0, 1023, 100, 0);
  if (client.connect(server, 80)) {
    client.print("GET /TEMHUMCO/formation_temhumco.php?hum_env=");
    client.print(hum_env);
    client.print("&temp_env=");
    client.print(temp_env);
    client.print("&temp_land=");
    client.print(temp_land);
    client.print("&per_mq135=");
    client.print(per_mq135);
    client.print("&per_FC28=");
    client.print(per_FC28);
    client.print("&sensor_id=");
    client.print(sensor_id);
    client.println(" HTTP/1.1");
    client.print("Host: ");
    client.println(server);
    client.println("Connection: close");
    client.println();
  } else {
```

Elaboración propia

Finalmente se tiene un bucle while en el código que verifica si la conexión con el servidor web todavía está activa. Y si la conexión con el servidor se cierra, se detiene la conexión, y tenemos un delay largo antes de que el ciclo comience nuevamente.

Figura 39

Verificación de la conexión con el servidor

```
while (client.connected()) {  
  int len = client.available();  
  if (len > 0) {  
    byte buffer[80];  
    if (len > 80) len = 80;  
    client.read(buffer, len);  
    if (printWebData) {  
      }  
    }  
  }  
}  
  
if (!client.connected()) {  
  client.stop();  
  delay(58358);  
}  
}
```

Elaboración propia



CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

En este capítulo presentaremos los resultados obtenidos a través del análisis de variables fundamentales que nos ayudan a optimizar las condiciones del suelo y ambiente en la ciudad de Puno. Recopilando y transformando datos.

4.1.1. Análisis de variables fundamentales para la optimización de suelo y ambiente

Durante el recorrido por la ciudad de Puno, hemos identificado parques que experimentan un deterioro evidente debido a la falta de atención por parte de las autoridades. Con el objetivo de abordar esta problemática, se han establecido variables esenciales que deben ser evaluadas para garantizar un riego apropiado en los parques y jardines, con la visión de transformar la estética del distrito de Puno y proporcionar diversos beneficios para la comunidad y el entorno urbano.

Para determinar las variables fundamentales para la optimización del suelo y ambiente de los jardines de Puno, se determinó que se usarán las siguientes variables:

- **Humedad del suelo:** Esta medida constante nos permite supervisar el nivel de humedad necesario para los parques y jardines en el distrito de Puno, asegurando un cuidado adecuado.
- **Temperatura ambiente:** Dada la variabilidad abrupta de las temperaturas en nuestro distrito, se ha reconocido la importancia de monitorear este parámetro para adaptar mejor las prácticas de cuidado y mantenimiento.



- **Temperatura del suelo:** La medición de la temperatura del suelo proporciona información precisa en grados Celsius, permitiendo ajustar las estrategias de manejo y asegurando condiciones óptimas para el crecimiento de las plantas.
- **Humedad del Ambiente:** Esta medida nos permite evaluar con precisión el porcentaje de humedad presente en nuestro entorno.
- **Calidad del Aire:** Mediante esta variable, llevamos a cabo mediciones para evaluar la presencia de contaminantes en los parques y sus alrededores, proporcionando una evaluación integral de la pureza del aire circundante.

4.1.2. Casos de uso

- Parques con suelos secos
- Parques con plantas que presentan falta de riego
- Jardines que no tienen el mantenimiento adecuado
- Parques con problemas de seguridad
- Parques con vegetación invasiva
- Parques con poca iluminación

4.1.3. Implementación del prototipo para la optimización del suelo y ambiente de los jardines de Puno

El diseño realizado del servicio web consta de una estructura LAMP (Linux Ubuntu Server, Apache, MariaDB y PHP), también se usa HTML y permite verificar los datos cada 60 segundos.

Los resultados se encuentran disponibles en la web y pueden ser visualizados tanto desde un dispositivo móvil como desde una computadora con conexión a Internet. En el contexto del parque, los datos revelan que la humedad ambiental se sitúa en torno al 30%, mientras que la temperatura ambiente alcanza

los 27 grados Celsius. Además, la temperatura del suelo registra unos 23 grados Celsius, la concentración de gases tóxicos se encuentra en un nivel del 25%, y la humedad del suelo alcanza un 73%. Esto demuestra como el sistema ayuda a la mejora del tratamiento de los jardines, pudiendo saber cuándo se debe regar los jardines.

Figura 40

Resultados en la Web

Conexión Satisfactoria!

Humedad Ambiente	Temperatura Ambiente	Temperatura Suelo	Calidad de Aire	Humedad Suelo	IP Cliente	ID Sensor	Fecha Hora
29.20	28.40	23.62	25.00	73.00	179.6.8.58	JEH-01	2023-09-25 12:46:57
29.20	28.40	23.62	25.00	73.00	179.6.8.58	JEH-01	2023-09-25 12:45:57
29.60	28.00	23.62	26.00	73.00	179.6.8.58	JEH-01	2023-09-25 12:44:56
29.10	27.70	23.62	25.00	73.00	179.6.8.58	JEH-01	2023-09-25 12:43:56
29.10	27.60	23.62	25.00	74.00	179.6.8.58	JEH-01	2023-09-25 12:42:55
29.00	27.50	23.62	25.00	73.00	179.6.8.58	JEH-01	2023-09-25 12:41:55
29.20	27.50	23.69	25.00	74.00	179.6.8.58	JEH-01	2023-09-25 12:40:54
29.40	27.60	23.69	25.00	74.00	179.6.8.58	JEH-01	2023-09-25 12:39:54
29.90	27.60	23.75	26.00	74.00	179.6.8.58	JEH-01	2023-09-25 12:38:53
29.50	27.60	23.75	26.00	74.00	179.6.8.58	JEH-01	2023-09-25 12:37:53
29.90	27.90	23.81	26.00	74.00	179.6.8.58	JEH-01	2023-09-25 12:36:52
30.20	27.30	23.81	26.00	74.00	179.6.8.58	JEH-01	2023-09-25 12:35:52
30.00	27.10	23.81	26.00	74.00	179.6.8.58	JEH-01	2023-09-25 12:34:51
30.70	26.80	23.81	26.00	74.00	179.6.8.58	JEH-01	2023-09-25 12:33:51
30.70	26.40	23.81	26.00	73.00	179.6.8.58	JEH-01	2023-09-25 12:32:50
30.50	26.50	23.87	27.00	75.00	179.6.8.58	JEH-01	2023-09-25 12:31:50
30.30	26.40	23.87	28.00	74.00	179.6.8.58	JEH-01	2023-09-25 12:30:49
30.10	26.60	23.94	28.00	75.00	179.6.8.58	JEH-01	2023-09-25 12:29:49

Elaboración propia

Figura 41

Prototipo desarrollado



Elaboracion propia

4.1.4. Diseño de la composición de software y hardware a nivel de cliente y servidor

En el proceso de adquisición de datos, Arduino desempeña un papel crucial al capturar tanto valores analógicos como digitales provenientes de los sensores.

Estas lecturas, que representan información específica sobre el entorno o los dispositivos monitorizados, son interpretadas y procesadas por Arduino a través de un código programado previamente. Este código no solo se encarga de la simple lectura de valores, sino que también incluye instrucciones detalladas para el acondicionamiento, conversión y análisis de los datos, lo que permite transformar las lecturas en información significativa y, en muchos casos, tomar decisiones automatizadas o generar respuestas específicas en función de las condiciones detectadas.



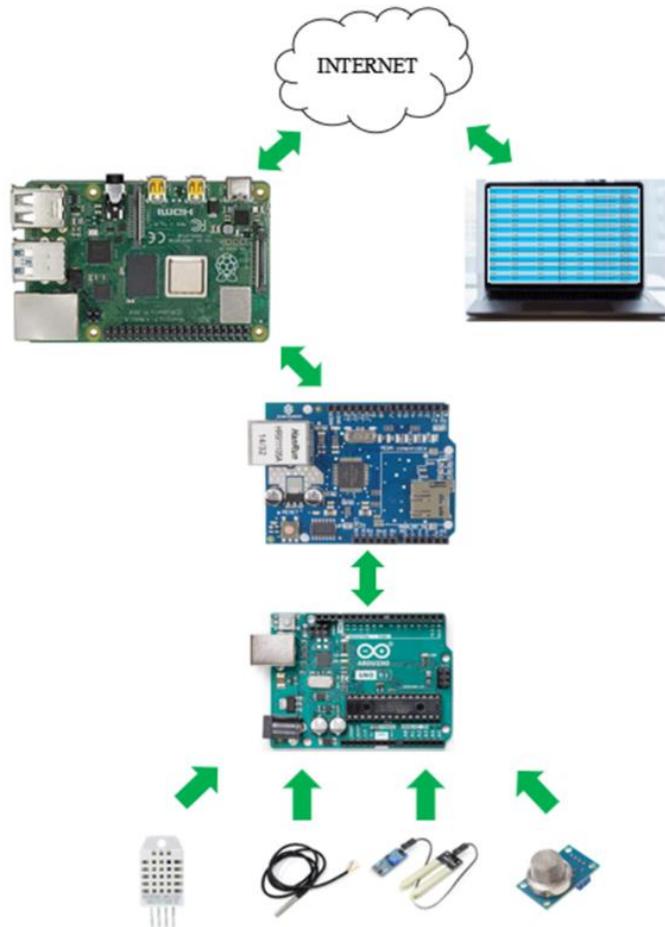
Esencialmente, Arduino actúa como un puente entre el mundo físico y la lógica de control, permitiendo una interacción inteligente y adaptativa con el entorno.

Los datos, una vez procesados por Arduino, se transfieren a través de la shield Ethernet. Esta valiosa extensión de hardware y funcionalidad se incorpora a la placa principal, habilitando a Arduino para el envío y recepción de información mediante conexiones Ethernet o redes. En otras palabras, los valores provenientes de los sensores se transmiten a través de cables Ethernet hacia un router u otro destino de red, permitiendo una comunicación efectiva y versátil en proyectos que requieren conectividad Ethernet.

Los datos de Arduino se canalizan hacia la Raspberry Pi, que cumple la función de un enrutador interno con una configuración que posibilita la salida de estos datos hacia Internet, es decir, fuera de la red local.

Figura 42

Arquitectura del prototipo desarrollado



Elaboración propia

La importancia de un prototipo reside en su capacidad para concretar y evaluar conceptos y diseños de forma palpable antes de invertir recursos considerables en el desarrollo definitivo de un producto o sistema. Al posibilitar la exploración de ideas, la detección temprana de desafíos y la revisión de soluciones, los prototipos se convierten en herramientas fundamentales en el proceso de diseño y desarrollo, mejorando la calidad, la eficacia y la factibilidad de proyectos. Teniendo en cuenta lo anterior, listamos los componentes más importantes de esta investigación.

- Arduino
- Shield ethernet

- Raspberry Pi 4B
- Sensor DHT22
- Sensor DS18B20
- Sensor FC-28
- Sensor MQ135

4.1.5. Arquitectura del sistema basado en internet de las cosas para la optimización y calidad de suelo

- Parque José Salcedo

Figura 43

Ubicación del parque José Salcedo



Elaboración propia

Figura 44

Prototipo en el parque José salcedo



Elaboración propia

- Parque Pedro Vilcapaza

Figura 45

Ubicación del parque Pedro Vilcapaza



Elaboración propia

Figura 46

Prototipo en el parque Pedro Vilcapaza



Elaboración propia

4.1.6. Despliegue del servicio

El despliegue de este servicio se da en 10 parques de la ciudad, los que se listan a continuación:

- Parque José Salcedo
- Parque de la Amistad
- Parque Dante Nava
- Parque Ciudad de los Niños
- Parque de la Cultura
- Parque del Periodista
- Parque Pedro Vilcapaza
- Parque Carácter
- Parque de la Madre
- Parque Sikuris



Los parques mencionados son parques que se identificaron en estado de descuido.

4.1.7. Evaluación de la calidad del servicio implementados

Los servicios implementados en los parques permitieron medir las variables con efectividad, teniendo éxitos en la calidad del servicio del prototipo desarrollado.

4.2. DISCUSIÓN

4.2.1. En cuanto a los objetivos de nuestra investigación

En este proyecto se usan las variables como: temperatura ambiente, humedad ambiente, temperatura de suelo, humedad del suelo y calidad de aire, pero en la investigación de (Mora Magallanes & Rosas Pari, 2019) se realiza la medición de las variables como temperatura ambiente, humedad de suelo y niveles de humo.

Según (Palak & Ferdous Ahmed, 2019) en su investigación titulada “Estación meteorológica basada en la nube que utiliza dispositivos IoT” concluyeron que este trabajo ha logrado su objetivo propuesto de desarrollar una estación meteorológica inteligente, modular y rentable; mientras que en la presente investigación se concluye que se ha logrado hacer una arquitectura que pueda tener varios módulos por distintas zonas geográficas en los parques del distrito de Puno.

Según (Chaoying, 2022) en su investigación titulada “Sistema de Monitoreo Ambiental Ecológico Multipunto con Internet de las Cosas” el objetivo es principalmente construir un sistema de monitoreo ambiental ecológico multipunto con la plataforma Yeelink IoT. Por lo tanto, este trabajo utiliza medios de alta tecnología para monitorear los cambios en la contaminación ambiental ecológica y al mismo tiempo recuerda a los ciudadanos que siempre deben prestar atención a la protección del medio ambiente. En la presente investigación se puede supervisar o



monitorear el estado de la calidad de suelo y ambiente de los jardines del distrito de Puno, con un modelo cliente servidor donde los clientes se encuentran en los jardines y el servidor en la nube.

Según (Sanjeevi y otros, 2020) en su investigación titulada “Agricultura de precisión y ganadería con Internet de las cosas basado en una red de sensores inalámbricos” llegaron a la conclusión de que la agricultura de precisión basada en IoT y el sistema agrícola pueden confirmar que son extremadamente útiles para los agricultores, ya que una cantidad mayor o menor de riego es buena para la agricultura.

En el Perú la agricultura es una actividad económica crítica y con un desarrollo que no ha cumplido la demanda del mercado actual y se presume que de no invertir en tecnología para el presente y futuro entonces el país no tendrá la producción suficiente y adecuada para el mercado nacional y mercado internacional, esto se valida con lo que (Puranik y otros, 2019) exponen en su investigación titulada “Automatización en Agricultura e IoT” concluyeron que la agricultura ha desempeñado un papel importante en el crecimiento económico de la India desde entonces y seguirá siéndolo en el futuro.

El diseño y la arquitectura de esta investigación está basada en Arduino UNO y Raspberry Pi 4B mientras que en la investigación de (Palak & Ferdous Ahmed, 2019) se usó la Raspberry Pi ZERO y la Raspberry Pi 3. Se obtiene resultados en ambos casos donde los costos económicos se ha reducido con el uso de estos dispositivos.

Según (Gutierrez Perez & Narvaez Mellado, 2022) en su investigación titulada “Sistema de internet de las cosas para mejorar el proceso de riego en la



comunidad de matará” llegaron a la conclusión de que la implementación del sistema IoT ha resultado en una reducción del 26.05 % en el volumen utilizado en el riego de la comunidad de Matará. En la presente investigación se trata de supervisar varias variables, una de ellas es la humedad ambiente y humedad del suelo, de esta forma se podrá optimizar la cantidad de agua que pueda ser usada para el riego de los jardines del distrito de Puno.

Según (Mora Magallanes & Rosas Pari, 2019) en la investigación denominada "Diseño, desarrollo e implementación de una red de sensores inalámbricos (WSN) para el control, monitoreo y toma de decisiones en la agricultura de precisión basada en Internet de las cosas (IoT) - un caso de estudio enfocado en el cultivo de frijol", se llegó a la conclusión de que se creó y puso en funcionamiento un dispositivo electrónico con la capacidad de establecer conexión a Internet, empleando el módulo ESP12E del MCU ESP8266, con el objetivo de fomentar la adopción de tecnologías emergentes en el ámbito de IoT. En la investigación actual que expone el autor se usa una arquitectura de Internet de las Cosas con el objetivo de supervisar variables con una conexión a un servidor ubicado en la nube, en Internet.

De acuerdo con la investigación realizada por (Guerra Guerra y otros, 2019) en su estudio titulado "Desarrollo de un servicio de monitoreo y control de áreas verdes en un municipio de Lima a través de la implementación de Internet de las Cosas (IoT)", se llega a la conclusión de que el proyecto actual permite la obtención de datos en tiempo real desde los puntos de gestión de áreas verdes. En la presente investigación se logra monitorear las variables de los puntos de gestión en áreas verdes en tiempo real.

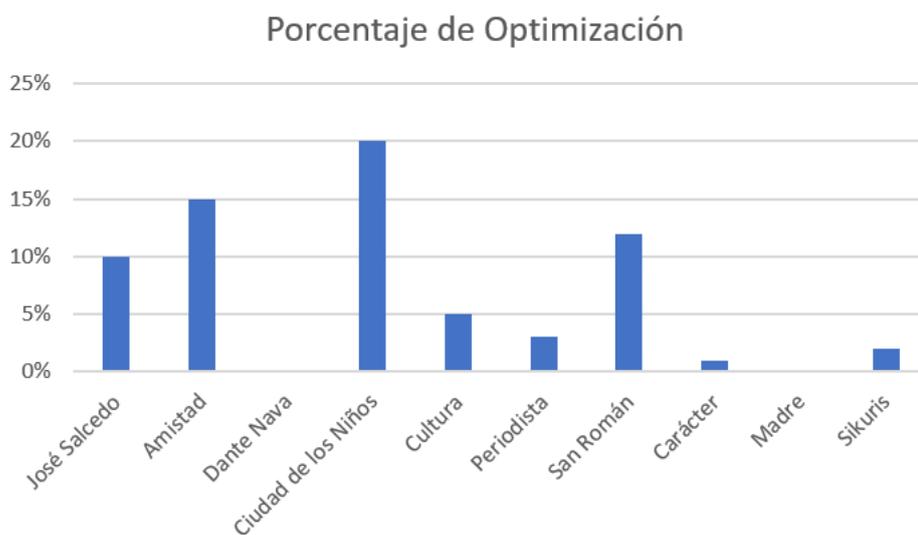
4.3. PRUEBA DE HIPOTESIS

Al llevar a cabo el monitoreo de los parques, hemos identificado tanto los avances en las mejoras implementadas como los posibles estancamientos en el proceso de mejoramiento de estos espacios verdes.

- Parque José Salcedo – 10%
- Parque de la Amistad – 15%
- Parque Dante Nava – 0%
- Parque Ciudad de los Niños - 20%
- Parque de la Cultura – 5%
- Parque del Periodista – 3%
- Parque San Román – 12%
- Parque Carácter – 1%
- Parque de la Madre – 0%
- Parque sikuris – 2%

Figura 47

Porcentaje de optimización en cada uno de los parques



Elaboración propia

La hipótesis nula (H_0) indica que no existe un efecto, mejora u optimización con el uso del sistema.

La hipótesis alternativa (H_1) sugiere que existe un efecto, mejora u optimización; y en esta investigación el diseño de una arquitectura basada en Internet de las cosas logró la optimización de la calidad del suelo y ambiente de los jardines del distrito de Puno en un 6.8% como promedio de la media poblacional.

La investigación llevada a cabo ha resultado en avances significativos que contribuyen a potenciar la calidad y el entorno de los jardines. El prototipo desarrollado, que facilita el monitoreo a través de la Internet de las Cosas (IoT), desempeña un papel crucial en la optimización del riego en parques y jardines. Este sistema no solo mejora la eficiencia del riego, sino que también contribuye a la creación de un entorno más amigable para la sociedad en general.

En relación a la prueba de normalidad, se establecieron las siguientes hipótesis:

- H_0 : El análisis de datos se procesan de manera eficiente ($p < \alpha$) y se utiliza T-Student.
- H_1 : El análisis de datos no se procesan de manera eficiente ($p \geq \alpha$) y se utiliza U de MannWhitney.

Es importante tener en cuenta que el valor de $\alpha = 0.05$ debe ser considerado.

Considerando que el valor $\alpha = 0.05$ y para nuestra toma de decisión para la prueba de normalidad:

- Si, $p < 0.05$, se acepta la hipótesis nula (H_0) se rechaza la hipótesis alterna (H_1).
- Si, $p \geq 0.05$, se acepta la hipótesis alterna (H_1) se rechaza la hipótesis nula (H_0).



Se considera una variable de valor 0 como resultado de prueba lo que significa que la transformación de datos no obedece a una distribución normal. Para ello se realizó la prueba métrica U de Mann Whitney.

Para la hipótesis y tiempo de medición de la temperatura se ha determinado medir los siguientes parámetros: temperatura y humedad del ambiente, temperatura y humedad del suelo y adicionalmente se mide los gases tóxicos. Ya que estos parámetros son usados para obtener información del estado, calidad y ambiente de los jardines del distrito de Puno.



V. CONCLUSIONES

PRIMERA: Al diseñar nuestra arquitectura basada en Internet de las cosas se pudo observar de manera significativa la mejora de calidad del suelo y ambiente de los jardines de puno. Implementando exitosamente nuestro sistema IoT lo que ha aumentado de manera significativa la frecuencia de monitoreo de la temperatura, humedad y entorno urbano. Aportando a la salud y calidad del medio ambiente.

SEGUNDA: Al realizar un análisis establecido de variables esenciales con el objetivo de evaluar y garantizar un riego apropiado en los parques y jardines, hemos proporcionado diversos beneficios para suelo y jardines y entorno urbano aportando variables importantes para el análisis y tratamiento de suelo y ambiente de los jardines de puno.

Este proyecto brindará variables precisas y en tiempo real sobre las condiciones del entorno y el estado de los parques y jardines, lo que permitirá tomar decisiones basadas en datos para mejorar la calidad de vida de la comunidad, promover la conservación de recursos naturales y embellecer aún más el distrito de Puno.

TERCERA: Con el fin de implementar un prototipo usando técnicas de análisis y transformación de datos el análisis arrojó un valor Z de -8,948, lo que confirma que los jardines tienen un impacto positivo en la ciudad de Puno y es de gran importancia. Por ello se ha determinado medir los siguientes parámetros: temperatura y humedad del ambiente, temperatura y humedad del suelo y adicionalmente se mide los gases tóxicos. Estos parámetros son



usados para obtener información del estado de los jardines y optimizando la calidad del suelo y el ambiente con el riego.

Debido a la implementación de nuestro sistema y esta muestra utilizando la estadística de prueba U de Mann-Whitney con un valor Z de -8,927, que indica claramente que la implementación de nuestro sistema optimiza el riego y calidad de suelo en los jardines del distrito de Puno. Es un paso fundamental hacia una gestión más eficiente y sostenible de estos espacios.

CUARTA: Finalmente se concluye que la arquitectura de Internet de las cosas permite la optimización del tratamiento de los jardines.

Analizando y transformando las variables ambientales que permiten identificar si un jardín esta con un buen tratamiento. Gracias a este sistema, tenemos la capacidad de visualizar en tiempo real los datos recopilados en el servidor. Los resultados obtenidos demuestran que la incorporación de la arquitectura IoT administra variables para el tratamiento de jardines representando un avance notable el análisis de calidad de suelo y ambiente de la ciudad de Puno.



VI. RECOMENDACIONES

PRIMERA: Para las posteriores investigaciones considerando los buenos resultados obtenidos podemos explorar una amplia variedad de sensores ambientales que puede tener un impacto considerable en la recopilación de datos a Corto plazo y Largo plazo. En este sentido, se recomienda la adquisición de sensores de calidad profesional o industrial, como medida para evitar la necesidad de reemplazarlos con frecuencia y garantizar una continuidad efectiva en la toma de datos.

SEGUNDA: Con el avance de la tecnología se puede discutir y analizar mejoras y posibles aplicaciones futuras del sistema de monitoreo, por ejemplo, la Incorporación de inteligencia artificial o el uso de drones con sensores de alta calidad y tecnología para la recopilación de datos.

TERCERA: Desarrollar nuevos sistemas de IoT que abarquen más variables para el preprocesamiento de datos para una mayor adaptabilidad y optimización de calidad de suelo y ambiente en los jardines.



VII. REFERENCIAS BIBLIOGRÁFICAS

- Abdessamie, H. (2021). *Diseño de cargador de baterías fotovoltaico y eólico para aplicaciones de IoT*. Escuela Técnica Superior de Ingeniería Industrial.
- Bernstein, J. (2018). *Networking Made Easy: Get Yourself Connected*. Independently published.
- Chaoying, C. (2022). Sistema de Monitoreo Ambiental Ecológico Multipunto con Internet de las Cosas. *Hindawi* , 12.
- Chica García, J. A., Rodríguez Aya, Á. A., Figueredo Luna, J. A., & Polanco Contreras, R. H. (2020). Sistema de monitoreo en tiempo real de gases CO, CO₂, NOX y PM_{2,5} en la ciudad de Villavicencio. . *EXPOTECH* , 7.
- Dávila Dávila, J. C., & Mena Dávila, J. H. (2022). Los impuestos verdes y su relación con el derecho fundamental a un medio ambiente saludable. *TecnoHumanismo*, 32.
- Electronics, A. (2020). *Digital-output relative humidity & temperature sensor/module* .
- Flores, W. G. (2022). *Programación Básica en Lenguaje C*. Cinvestav Unidad Tamaulipas.
- Guerra Guerra, J., Lunarejo Gonzales, Y., & Fermín Pérez, A. (2019). Implementación de un servicio de monitoreo. *Alicia*, 10.
- Guijarro Rodríguez, A., Cevallos Torres, L., Preciado Maila, D., & Zambrano Manzur, B. (2018). Sistema de riego automatizado con Arduino. *Espacios* , 15.



- Gutierrez Perez, W. J., & Narvaez Mellado, J. (2022). *Sistema de internet de las cosas para mejorar el proceso de riego en la comunidad de Matará*. Lima: Universidad Autónoma del Perú.
- Marcillo Quimis, G. J. (2022). *IMPLEMENTACIÓN DE CIRCUITOS ELECTRÓNICOS PROGRAMABLES EN PRÁCTICAS LABORATORIO PARA LA ASIGNATURA DE MICROCONTROLADORES DE LA CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN*. Ecuador : UNIVERSIDAD ESTATAL DEL SUR DE MANABÍ.
- Medina de la Cruz, M. R. (2020). *Diseño e implementación de un dispositivo de Internet de las cosas para la medición y monitoreo de la calidad de aire en zonas urbanas*. Lima: UCH.
- Millahual , C. (2020). *Descubriendo Arduino*. RedUsers.
- Mohamed , F., & Dahoud, A. A. (2019). Exploring One-wire Temperature sensor ³DS18B20' with Microcontrollers. *University of Al-Zaytoonah Faculty of IT*, 8.
- Mora Magallanes, H. V., & Rosas Pari, J. L. (2019). *Diseño, Desarrollo e Implementación de una red de sensores Inalambricos (WSN) Para el control, Monitoreo y toma de decisiones aplicado en la agricultura de precisión basado en internet de las cosas (IoT) – caso de estudio cultivo de frijol*. Lima: Universidad Ricardo Palma .
- Osio, J. R., Aróztegui, W. J., & Rapallini, J. A. (2020). *Sistemas digitales basados en microcontroladores*. Argentina - La plata: Libros de Cátedra .
- Palak, K., & Ferdous Ahmed, B. (2019). Estacion meteorologica basada en la nube que utiliza dispositivos de IoT. *IEEE*, 6.



- Pérez Tavera, I. H. (2023). Arduino IDE. *Vida Científica Boletín Científico de la Escuela Preparatoria*, 30 - 32 .
- Puranik, V., Sharmila, A. R., & A. K. (2019). Automatización en Agricultura e IoT. *IEEE*, 6.
- Quiroga Montoya, E. A., Jaramillo Colorado, S. F., Campo Muñoz, W. Y., & Chanchí Golondrino, G. E. (2016). Propuesta de una Arquitectura para Agricultura de Precisión Soportada en IoT. *RISTI*, 18.
- Roque, F. P., Zaldívar, E. V., & Fuentes, O. A. (2013). *Sistema de Adquisición de Datos con comunicación inalámbrica*. <http://scielo.sld.cu/>.
- S. Nagy, A., Polanco Risquet, A., Martínez de la Cotera, O. L., & Carralero Ibargollen, O. (2020). Medición simultánea de gases con sensores MQ. *RIELAC*, 10.
- Sanjeevi, P., S. P., Kumar, B. S., Gunasekaran, G., Alagir, I., & R. V. (2020). Agricultura de precisión y ganadería con Internet de las cosas basado en una red de sensores inalámbricos. *Wiley*.
- Sharanappa, S. (2021). *Fundamentals Of Agronomy*. New India Publishing Agency-Nipa.
- Smeenck, H. (2023). *Internet of Things for Smart Buildings: Leverage IoT for smarter insights for buildings in the new and built environments*. Packt Publishing.
- Technology, Z. W. (2015). *Air Quality Gas Sensor*. Winsen.
- Tovar Soto, J. P., José de los Santos , S. S., Badillo Rodríguez, A., & Rodríguez Cainaba, G. O. (2019). Internet de las cosas aplicado a la agricultura: estado actual. *Lámpsakos*, 20.



- Trias Prima , S., Unan Yusmaniar, O., Imam, F., & Hristina, P. (2020). Analisis Akurasi Sistem sensor DHT22 berbasis Arduino terhadap Thermohygrometer Standar. *Fisika dan aplikasinya*, 5.
- Veneros, J., García, L., Morales, E., Gomez, V., Torres, M., & Lopez Morales , F. (2020). Aplicación de sensores remotos para el análisis de cobertura vegetal y cuerpos de agua. *IDESIA* , 10.
- Waho, T. (2019). *Introduction to Analog-to-Digital Converters*. River Publishers.



ANEXOS

Anexo 1: Código Fuente

En este anexo se muestra los códigos en lenguaje Arduino IDE para las funciones de verificación de datos desde el prototipo a una computadora y transformación de variables analógicas a digitales.

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <DHT.h>
#define DHTTYPE DHT22
const int DHTPIN = 6;
DHT dht(DHTPIN, DHTTYPE);
const int pinOneWire = 7;
OneWire oneWire(pinOneWire);
DallasTemperature sensors(&oneWire);
int sensorpin_mq135 = A1;
int FC28 = A0;
int sensorValue_FC28;
int porcentaje_FC28;
void setup() {
  Serial.begin(9600);
  dht.begin();
  sensors.begin();
}
void loop() {
  delay(1000);
  float hum_env = dht.readHumidity();
  float temp_env = dht.readTemperature();
  Serial.print("Humedad Ambiente: ");
  Serial.print(hum_env);
  Serial.print("% - ");
  Serial.print("Temperatura Ambiente: ");
  Serial.print(temp_env);
```



```
Serial.print("°C");  
sensors.requestTemperatures();  
float temp_land = sensors.getTempCByIndex(0);  
Serial.print(" - Temperatura Suelo: ");  
Serial.print(temp_land);  
Serial.print(" °C");  
  
int sensorValue = analogRead(sensorpin_mq135);  
int per_mq135= map(sensorValue,0,1023,0,100);  
Serial.print(" - Gases Tóxicos: ");  
Serial.print(per_mq135);  
Serial.print(" %");  
sensorValue_FC28 = analogRead(FC28);  
int per_FC28 = map(sensorValue_FC28, 0, 1023, 100, 0);  
Serial.print(" - Humedad del suelo: ");  
Serial.print(per_FC28);  
Serial.println("%");  
  
}
```



Anexo 2 : Código Fuente

En este anexo se muestra los códigos en lenguaje Arduino IDE para las funciones de recolección de datos desde el cliente al servidor y transformación de variables analógicas a digitales.

```
#include <SPI.h>
#include <Ethernet.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <DHT.h>
#define DHTTYPE DHT22
const int DHTPIN = 6;
DHT dht(DHTPIN, DHTTYPE);
const int pinOneWire = 7;
OneWire oneWire(pinOneWire);
DallasTemperature sensors(&oneWire);
int sensorpin_mq135 = A1;
int FC28 = A0;
int sensorValue_FC28;
int porcentaje_FC28;
byte mac[] = { 0x54, 0xAB, 0x3A, 0x5B, 0xAD, 0x05 };
IPAddress server(108,62,123,70);
IPAddress ip(172,16,100,133);
byte gateway[] = { 172, 16, 100, 1 };
byte subnet[] = { 255, 255, 255, 0 };
byte myDns[] = { 8, 8, 8, 8 };
EthernetClient client;
bool printWebData = true;
char sensor_id[] = "JEH-01";
void setup(){
  Ethernet.begin(mac, ip, myDns, gateway, subnet);
  dht.begin();
  sensors.begin();
```



```
        delay(500);
    }
    void loop(){
        delay(1000);
        float hum_env = dht.readHumidity();
        float temp_env = dht.readTemperature();
        sensors.requestTemperatures();
        float temp_land = sensors.getTempCByIndex(0);
        int sensorValue = analogRead(sensorpin_mq135);
        int per_mq135 = map(sensorValue,0,1023,0,100);
        sensorValue_FC28 = analogRead(FC28);
        int per_FC28 = map(sensorValue_FC28, 0, 1023, 100, 0);
        if (client.connect(server, 80)) {
            client.print("GET /TEMHUMCO/formaction_temhumco.php?hum_env=");
            client.print(hum_env);
            client.print("&temp_env=");
            client.print(temp_env);
            client.print("&temp_land=");
            client.print(temp_land);
            client.print("&per_mq135=");
            client.print(per_mq135);
            client.print("&per_FC28=");
            client.print(per_FC28);
            client.print("&sensor_id=");
            client.print(sensor_id);
            client.println(" HTTP/1.1");
            client.print("Host: ");
            client.println(server);
            client.println("Connection: close");
            client.println();
        } else {
        }
        while (client.connected()) {
            int len = client.available();
```



```
if (len > 0) {  
    byte buffer[80];  
    if (len > 80) len = 80;  
    client.read(buffer, len);  
    if (printWebData) {  
  
        }  
    }  
}  
  
if (!client.connected()) {  
    client.stop();  
    delay(58358);  
}  
}
```



Anexo 3 – Declaración jurada de autenticidad de tesis



Universidad Nacional
del Altiplano Puno



Vicerrectorado
de Investigación



Repositorio
Institucional

DECLARACIÓN JURADA DE AUTENTICIDAD DE TESIS

Por el presente documento, Yo Jehan Carlos Coila Ticona , Identificado con DNI 72865330 en mi condición de egresado de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado

informo que he elaborado el/la Tesis o Trabajo de Investigación denominada:

“DISEÑO DE UNA ARQUITECTURA BASADA EN INTERNET DE LAS COSAS PARA LA OPTIMIZACIÓN DE LA CALIDAD DEL SUELO Y AMBIENTE DE LOS JARDINES DEL DISTRITO DE PUNO”

Es un tema original.

Declaro que el presente trabajo de tesis es elaborado por mi persona y **no existe plagio/copia** de ninguna naturaleza, en especial de otro documento de investigación (tesis, revista, texto, congreso, o similar) presentado por persona natural o jurídica alguna ante instituciones académicas, profesionales, de investigación o similares, en el país o en el extranjero.

Dejo constancia que las citas de otros autores han sido debidamente identificadas en el trabajo de investigación, por lo que no asumiré como tuyas las opiniones vertidas por terceros, ya sea de fuentes encontradas en medios escritos, digitales o Internet.

Asimismo, ratifico que soy plenamente consciente de todo el contenido de la tesis y asumo la responsabilidad de cualquier error u omisión en el documento, así como de las connotaciones éticas y legales involucradas.

En caso de incumplimiento de esta declaración, me someto a las disposiciones legales vigentes y a las sanciones correspondientes de igual forma me someto a las sanciones establecidas en las Directivas y otras normas internas, así como las que me alcancen del Código Civil y Normas Legales conexas por el incumplimiento del presente compromiso

Puno 16 de Enero del 2024

FIRMA (obligatoria)



Huella



Anexo 4 : Autorización para el depósito de tesis o trabajo de investigación en el repositorio institucional



Universidad Nacional
del Altiplano Puno



Vicerrectorado
de Investigación



Repositorio
Institucional

AUTORIZACIÓN PARA EL DEPÓSITO DE TESIS O TRABAJO DE INVESTIGACIÓN EN EL REPOSITORIO INSTITUCIONAL

Por el presente documento, Yo **Jehan Carlos Coila Ticona**, Identificado con DNI 72865330 en mi condición de egresado de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado

Escuela profesional de ingeniería de sistemas, informo que he elaborado el/la Tesis o Trabajo de Investigación denominada:

“DISEÑO DE UNA ARQUITECTURA BASADA EN INTERNET DE LAS COSAS PARA LA OPTIMIZACIÓN DE LA CALIDAD DEL SUELO Y AMBIENTE DE LOS JARDINES DEL DISTRITO DE PUNO”

para la obtención de Grado, Título Profesional o Segunda Especialidad.

Por medio del presente documento, afirmo y garantizo ser el legítimo, único y exclusivo titular de todos los derechos de propiedad intelectual sobre los documentos arriba mencionados, las obras, los contenidos, los productos y/o las creaciones en general (en adelante, los “Contenidos”) que serán incluidos en el repositorio institucional de la Universidad Nacional del Altiplano de Puno.

También, doy seguridad de que los contenidos entregados se encuentran libres de toda contraseña, restricción o medida tecnológica de protección, con la finalidad de permitir que se puedan leer, descargar, reproducir, distribuir, imprimir, buscar y enlazar los textos completos, sin limitación alguna.

Autorizo a la Universidad Nacional del Altiplano de Puno a publicar los Contenidos en el Repositorio Institucional y, en consecuencia, en el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto, sobre la base de lo establecido en la Ley N° 30035, sus normas reglamentarias, modificatorias, sustitutorias y conexas, y de acuerdo con las políticas de acceso abierto que la Universidad aplique en relación con sus Repositorios Institucionales. Autorizo expresamente toda consulta y uso de los Contenidos, por parte de cualquier persona, por el tiempo de duración de los derechos patrimoniales de autor y derechos conexos, a título gratuito y a nivel mundial.

En consecuencia, la Universidad tendrá la posibilidad de divulgar y difundir los Contenidos, de manera total o parcial, sin limitación alguna y sin derecho a pago de contraprestación, remuneración ni regalía alguna a favor mío; en los medios, canales y plataformas que la Universidad y/o el Estado de la República del Perú determinen, a nivel mundial, sin restricción geográfica alguna y de manera indefinida, pudiendo crear y/o extraer los metadatos sobre los Contenidos, e incluir los Contenidos en los índices y buscadores que estimen necesarios para promover su difusión.

Autorizo que los Contenidos sean puestos a disposición del público a través de la siguiente licencia:

Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

En señal de conformidad, suscribo el presente documento.

Puno 16 de Enero del 2024

FIRMA (obligatoria)



Huella