

**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES**

TEMA:

Diseño de un sistema IoT para medición de variables ambientales de dispositivos TI alojados en racks de data centers mediante un módulo ESP32 y MQTT.

AUTOR:

Solórzano Maldonado, Erick Andrés

**Trabajo de Titulación previo a la obtención del título de
INGENIERO EN TELECOMUNICACIONES**

TUTOR:

Ing. Romero Rosero, Carlos Bolívar, M. Sc

**Guayaquil, Ecuador
28 de agosto del 2023**



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el Sr. **Solórzano Maldonado, Erick Andrés** como requerimiento para la obtención del título de **INGENIERO EN TELECOMUNICACIONES**.

TUTOR:

Ing. Romero Rosero, Carlos Bolívar, M. Sc

DIRECTOR DE CARRERA

M. Sc. Bohórquez Escobar, Celso Bayardo

Guayaquil, al 28 día del mes de agosto del año 2023



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

**FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES**

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Solórzano Maldonado, Erick Andrés**

DECLARO QUE:

El trabajo de titulación **Diseño de un sistema IoT para medición de variables ambientales de dispositivos TI alojados en racks de data centers mediante un módulo ESP32 y MQTT**, previo a la obtención del Título de **Ingeniero en Telecomunicaciones**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, al 28 día del mes de agosto del año 2023

EL AUTOR

Solórzano Maldonado, Erick Andrés



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

**FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES**

AUTORIZACIÓN

Yo, **Solórzano Maldonado, Erick Andrés**

Autorizo a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: **Diseño de un sistema IoT para medición de variables ambientales de dispositivos TI alojados en racks de data centers mediante un módulo ESP32 y MQTT**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

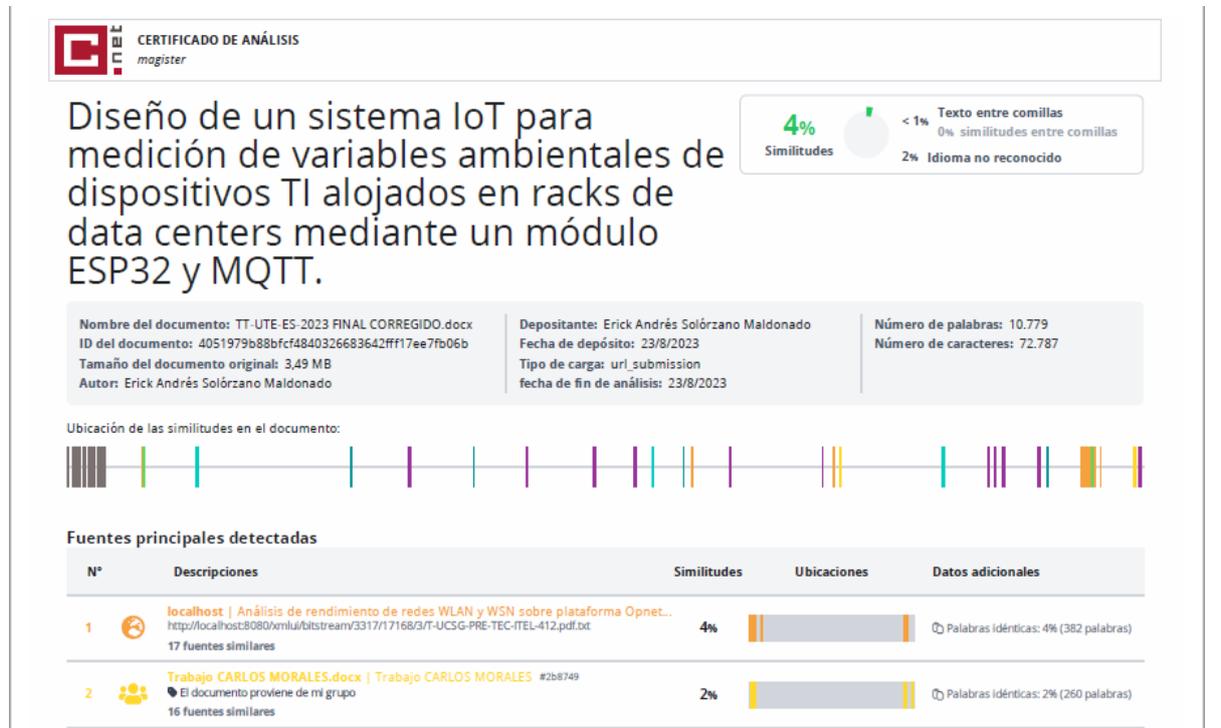
Guayaquil, al 28 día del mes de septiembre del año 2023

EL AUTOR

Solórzano Maldonado, Erick Andrés

INFORME SOFTWARE ANTIPLAGIO

REPORTE COMPILATIO



Reporte Compilatio del trabajo de titulación de la Carrera de Ingeniería en Telecomunicaciones: **“Diseño de un sistema IoT para medición de variables ambientales de dispositivos TI alojados en racks de data centers mediante un módulo ESP32 y MQTT”**, del estudiante Solórzano Maldonado, Erick Andrés se encuentra al **4 %** de coincidencias.

Atentamente,

Ing. Romero Rosero, Carlos Bolívar, M. Sc
DOCENTE-TUTOR

DEDICATORIA

Este trabajo de titulación va dedicado a Dios, porque él es mi luz y guía en todo este camino que es la vida, por regalarme esta familia que tengo, por los obstáculos que ha puesto a lo largo de mi carrera, que me permitieron ganar experiencia y crecer como persona y ahora como profesional.

A mi madre, Jeannette porque ha sido el pilar fundamental en mi formación como ser humano y ahora como profesional, por ese apoyo incondicional, consejos sabios, valores y motivación constante para que yo concluya mis estudios y esta etapa de mi vida.

A mi familia, por enseñarme que con esfuerzo y dedicación nada es imposible, por siempre mantenerse unida por ser esa familia bondadosa y amorosa, gracias por todo su amor y que Dios nos siga bendiciendo a todos.

A mi hermano, para que aprenda a discernir que las cosas que realmente valen la pena vienen de la mano del esfuerzo, trabajo, dedicación.

EL AUTOR

SOLÓRZANO MALDONADO, ERICK ANDRÉS

AGRADECIMIENTO

Agradezco primero que todo a Dios porque sin él nada de esto fuera posible, por dame esas fuerzas, energía, sabiduría y motivación cada día para pelear por mis sueños por mostrarme que siempre hay oportunidades si se trabaja de manera correcta, gracias por ese infinito amor y misericordia.

A mi madre, Jeannette por ser esa madre bondadosa, amorosa e incondicional apoyándome y depositando esa confianza en mí en todo momento, por ser mi claro ejemplo que con esfuerzo y trabajo todo se puede conseguir, por los regaños ya que me enseña hacer mejor ser humano.

A mis abuelos, por su eterno amor, por ese apoyo incondicional, por sus enseñanzas y también por sus regaños porque esas pequeñas cosas son las que nos hacen grandes y me han permitido ser una persona de bien.

A mi novia, agradecerle por esa motivación, por siempre estar presente de diferentes formas por ser amorosa y bondadosa, por aportar ese granito de arena para culminar esta etapa de mi vida.

Al Ingeniero Carlos Bolívar Romero Rosero por su asesoramiento y disposición desde el primer día, por su gran responsabilidad en cada reunión que hemos tenido, por sus sugerencias y por todas sus enseñanzas desinteresadas, muchas gracias por ayudarme a culminar con éxito mi trabajo de titulación.

EL AUTOR

SOLORZANO MALDONADO, ERICK ANDRES



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

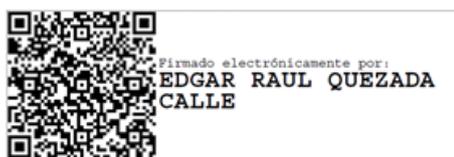
TRIBUNAL DE SUSTENTACIÓN

f. _____

Ing. Bohórquez Escobar, Celso Bayardo M. Sc.
DIRECTOR DE CARRERA

f. _____

Ing. Ubilla González, Ricardo Xavier
COORDINADOR DE ÁREA



f. _____

Ing. Quezada Calle, Edgar Raúl
OPONENTE

ÍNDICE GENERAL

| | |
|--|-----------|
| Índice de Figuras | XI |
| Índice de Tablas | XIII |
| Resumen..... | XIV |
| Abstract..... | XV |
| CAPÍTULO 1: DESCRIPCIÓN GENERAL..... | 2 |
| 1.1. Introducción..... | 2 |
| 1.2. Antecedentes..... | 3 |
| 1.3. Definición del Problema..... | 3 |
| 1.4. Justificación del Problema..... | 4 |
| 1.5. Objetivos del Problema de Investigación..... | 4 |
| 1.5.1. Objetivo General..... | 4 |
| 1.5.2. Objetivos Específicos..... | 5 |
| 1.6. Hipótesis..... | 5 |
| 1.7. Metodología de investigación..... | 5 |
| 1.8. Tipo de Investigación..... | 6 |
| CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA..... | 7 |
| 2.1. Introducción de IoT..... | 7 |
| 2.1.1. Definición y concepto de IoT..... | 7 |
| 2.1.2. Evolución histórica del IoT..... | 9 |
| 2.1.3. Aplicaciones y sectores clave del IoT..... | 12 |
| 2.1.4. Beneficios y desafíos del IoT..... | 15 |
| 2.1.5. Tecnologías clave del IoT..... | 18 |
| 2.1.6. Estándares y protocolos del IoT..... | 21 |
| 2.2. Datacenters y su importancia en la infraestructura tecnológica..... | 26 |
| 2.2.1. Definición y características de un datacenter..... | 26 |
| 2.2.2. Funciones y componentes de un datacenter..... | 27 |
| 2.2.3. Importancia de la temperatura y humedad en un datacenter..... | 29 |
| 2.3. Medición de temperatura y humedad en el entorno del datacenter..... | 30 |
| 2.3.1. Importancia de la medición de temperatura y humedad en un datacenter..... | 30 |
| 2.3.2. Parámetros y requisitos de temperatura y humedad en un datacenter..... | 32 |
| 2.3.3. Métodos y tecnologías tradicionales de medición de temperatura y humedad..... | 32 |
| 2.3.4. Limitaciones y desafíos de los métodos tradicionales de medición..... | 33 |
| 2.3.5. Infraestructura de red para la conectividad en un entorno de datacenter..... | 35 |
| CAPÍTULO 3: DISEÑO Y RESULTADOS..... | 37 |
| 3.1. Descripción general del sistema IoT para medición..... | 37 |
| 3.2. Primera etapa del sistema IoT para medición..... | 38 |
| 3.3. Segunda etapa del sistema IoT para medición..... | 41 |
| 3.4. Tercera etapa del sistema IoT para medición..... | 42 |

| | |
|--|-----------|
| 3.5. Resultados..... | 49 |
| CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES..... | 54 |
| 4.1. Conclusiones..... | 54 |
| 4.2. Recomendaciones | 55 |
| Referencias o bibliografías | 56 |
| Glosario General | 59 |
| Anexos | 60 |
| Anexo 1. Presupuesto aproximado del TT | 60 |

Índice de Figuras

| | |
|--|----|
| Figura 2.1: Internet de las cosas (IoT) | 12 |
| Figura 2.2: IoT en la industria..... | 14 |
| Figura 2.3: MQTT. | 22 |
| Figura 2.4: CoAP..... | 23 |
| Figura 2.5: Zigbee. | 24 |
| Figura 2.6: LoRaWAN..... | 25 |
| Figura 2.7: OPC UA. | 26 |
| Figura 2.8: Equipamientos de un datacenter..... | 27 |
| Figura 2.9: Límites operacionales..... | 29 |
| Figura 2.10: Límites operacionales. | 33 |
| Figura 2.11: Infraestructura para garantizar seguridad en el entorno. | 35 |
| Figura 3.1: Esquema del sistema IoT para medición..... | 37 |
| Figura 3.2: Módulo ESP32..... | 38 |
| Figura 3.3: Sensor DHT22. | 38 |
| Figura 3.4: Diagrama de conexión entre ESP32 y DHT22..... | 39 |
| Figura 3.5: Código cargado en la ESP32 1/2. | 40 |
| Figura 3.6: Código cargado en la ESP32 2/2. | 41 |
| Figura 3.7: Diagrama de envío de valores medidos a la nube..... | 42 |
| Figura 3.8: Clase “Datos_temp.dart”..... | 43 |
| Figura 3.9: Clase “Nodo.dart” Parte 1/3..... | 43 |
| Figura 3.10: Clase “Nodo.dart” Parte 2/3 | 44 |
| Figura 3.11: Clase “Nodo.dart” Parte 3/3. | 45 |
| Figura 3.12: Clase “Menu.dart” 1/2..... | 46 |

Figura 3.13: Clase “Menu.dart” 2/2.....47

Figura 3.14: Clase “Main.dart”.....48

Figura 3.15: Configuración “AndroidManifest.xml”.....49

Figura 3.16: Conexión entre ESP32 y DHT22.....49

Figura 3.17: ESP32 y DHT22 ubicados dentro del case.50

Figura 3.18: Nodo final.50

Figura 3.19: Nodo sensor ubicado en el rack.51

Figura 3.20: Aplicativo móvil desarrollado en Flutter – Pantalla de bienvenida...52

Figura 3.21: Aplicativo móvil desarrollado en Flutter –Menú principal.52

Figura 3.22: Aplicativo móvil desarrollado en Flutter – Nodo sensor.53

Figura 3.23: Visualización de valores medidos desde la plataforma en la nube. 53

Índice de Tablas

| | |
|--|----|
| Tabla 2.1: Principios claves del IoT..... | 8 |
| Tabla 2.2: Beneficios del IoT. | 16 |
| Tabla 2.3: Desafíos del IoT..... | 17 |
| Tabla 2.4: Funciones de un datacenters. | 28 |
| Tabla 2.5: Importancia de mediciones..... | 30 |
| Tabla 2.6: Limitaciones y desafíos. | 33 |

Resumen

La gestión de infraestructuras críticas, como las empleadas por los Proveedores de Servicios de Internet (ISP), enfrenta desafíos emergentes relacionados con la monitorización de condiciones ambientales en sus racks. Estas condiciones, especialmente las fluctuaciones de temperatura y humedad pueden comprometer la operatividad de dispositivos clave como enrutadores y conmutadores si no se mantienen dentro de rangos aceptables. Este trabajo aborda la necesidad de un sistema preciso y en tiempo real para supervisar estas variables. Nuestro principal objetivo es desarrollar un sistema IoT, aprovechando el potencial del módulo ESP32 y el sensor DHT22, para capturar, procesar y transmitir información sobre temperatura y humedad en racks de centros de datos. Mediante el uso del protocolo MQTT, aseguramos una transmisión de datos eficiente y confiable hacia una API en la nube, facilitando el acceso remoto y la toma de decisiones en tiempo real.

Dentro de este estudio, se profundiza en los fundamentos teóricos de las redes de sensores e IoT, ilustrando su relevancia y aplicabilidad en contextos industriales y empresariales. A través de la programación y configuración del ESP32 junto con el sensor DHT22, demostramos la eficacia de nuestra solución propuesta, destacando su precisión y fiabilidad en la monitorización ambiental. En conclusión, este trabajo presenta una solución innovadora y de bajo costo para abordar un problema operativo clave en los centros de datos, proponiendo un diseño IoT robusto que puede ser adaptado y escalado según las necesidades específicas de diferentes infraestructuras.

Palabras claves: Sensores, Redes de Sensores, Lot, Mqtt, Esp32, Temperatura, Humedad.

Abstract

The management of critical infrastructures, such as those used by Internet Service Providers (ISP), faces emerging challenges related to the monitoring of environmental conditions in their racks. These conditions, especially fluctuations in temperature and humidity, can compromise the operability of key devices such as routers and switches if not kept within acceptable ranges. This project addresses the need for an accurate and real-time system to oversee these variables. Our primary goal is to develop an IoT system, leveraging the potential of the ESP32 module and the DHT22 sensor, to capture, process, and transmit information about temperature and humidity in data center racks. Using the MQTT protocol, we ensure efficient and reliable data transmission to a cloud API, facilitating remote access and real-time decision-making. With in this study, we delve into the theoretical foundations of sensor networks and IoT, illustrating their relevance and applicability in industrial and business contexts. Through the programming and configuration of the ESP32 along with the DHT22 sensor, we demonstrate the effectiveness of our proposed solution, highlighting its accuracy and reliability in environmental monitoring.

In conclusion, this work presents an innovative and cost-effective solution to address a key operational issue in data centers, proposing a robust IoT design that can be adapted and scaled according to the specific needs of different infrastructures.

Keywords: Sensors, Sensor Networks, Lot, Mqtt, Esp32, Temperature, Humidity.

CAPÍTULO 1:

DESCRIPCIÓN GENERAL

1.1. Introducción.

El uso de IoT y redes de sensores inalámbricas de sensores ya no es una opción, sino una necesidad por parte de cualquier empresa que tenga visión, y se encuentre desarrollando proyectos tecnológicos. Lamentablemente, muchas compañías desconocen acerca de las nuevas tecnologías y prefieren mantenerse con lo que han venido trabajando, como suele ser sensores analógicos de medición, o sistemas muy costosos y generalizados. Por lo cual se limitan a conocer nuevas tecnologías de bajo consumo y las ventajas que ofrecen la implementación de las tecnologías IoT, tecnologías que son capaces de solventar todas las solicitudes de múltiples nodos sensores a la vez y que toda una red de sensores opere sin tener la mínima preocupación de que la red colapse.(Chaudhari et al., 2020)

Cuando se hablaba de topologías de redes de sensores, comenzaba la comparación y extenuante discusión sobre cual topología es mejor, hoy en día se dejó esa discusión a un lado para juntar y trabajar con diversas topologías a la vez. De tal manera que en el mismo ecosistema coexistan más de una, aprovechando los beneficios de cada topología y compensando sus desventajas entre sí.(Shrestha & Xing, 2007)

Actualmente, una de las aplicaciones de las redes de sensores son sistemas de mediciones para la obtención de las variables de interés, ya sea en sistemas agrícolas, acuícolas, entre otros. Esto como medida de seguridad, no solo para prevención sino para llevar un control, ya que existen lugares en los cuales la temperatura, humedad u otra variable debe ser controlada, por ejemplo, al momento

de almacenar mariscos, carnes, etc. En el caso de los dispositivos TI, como routers y switches alojados en racks, que al ser equipos que trabajan 24/7 estos deberían tener un sistema de monitoreo de sus temperatura y humedad para evitar fallas o daños irreversibles en los equipos. (Vélez, 2020)

1.2. Antecedentes.

Hoy en día los proveedores de servicio de internet enfrentan el gran reto de proporcionar al cliente un servicio de alta calidad, resiliencia y alta disponibilidad, es decir que no sea propenso a caídas y los enlaces se mantengan arriba para que el usuario pueda hacer uso de los servicios de voz, datos e internet que son distribuidos por la red. Para esto los nodos de distribución de red deben estar siempre funcionales a nivel lógico y físico, de tal manera que los factores externos no afecten el funcionamiento de estos. Todos estos dispositivos poseen infraestructuras metálicas para su protección sin embargo estos no otorgan una protección ni control total. Motivo por el cual surge la necesidad de diseño, implementación de un sistema para el monitoreo de variables que afectan la estructura de equipos, tales como la temperatura y humedad, las cuales causan daños irreversibles en los componentes electrónicos si se encontraran fuera del rango operacional. Con este antecedente encontramos variable dependiente que es la estructura de los equipos (conmutadores, Switch, enrutadores) y las causas que son el no monitoreo adecuado de temperatura y humedad.

1.3. Definición del Problema.

Actualmente los ISP no cuentan con un sistema de medición para racks por dispositivos el cual le proporcione los valores de temperatura y humedad de los enrutadores y/o conmutadores ubicados dentro de los racks debido a que valores

fuera del rango pueden causar daños en los equipos. Por lo cual nuestro problema de la investigación es:

¿Cómo afectan la temperatura y la humedad en la estructura de los equipos y sus componentes electrónicos en los racks y bastidores de centro de datos actualmente?

1.4. Justificación del Problema.

Los principales beneficiarios serán los ISP o empresas que desean obtener un mejor control de los valores ambientales como lo son la temperatura y humedad que experimentan los dispositivos ubicados en los racks, conmutadores y enrutadores, para evitar daños en los equipos por valores fuera del rango operacional de las variables ambientales.

El aporte académico del presente estudio ayudará para que estudiantes de la facultad puedan trabajar con tarjetas de desarrollo, sensores y Android Studio.

Está investigación tiene su relevancia ya que en estos últimos años el ecosistema, la tecnología y dispositivos electrónicos son afectados por cambios en las magnitudes de temperatura y humedad por el calentamiento global, es decir, cambios climáticos. Además, para mejorar la información acerca de sistemas IoT de bajo costo que les permita a las personas obtener el valor de las variables ambientales de interés en cualquier área a desarrollar.

1.5. Objetivos del Problema de Investigación.

1.5.1. Objetivo General.

Diseñar un sistema IoT mediante la programación del módulo ESP32, sensor DHT22 y envío de la información vía MQTT, para monitorear y medir variables ambientales en racks de centro de datos.

1.5.2. Objetivos Específicos.

1. Describir los fundamentos teóricos de las redes de sensores, IoT y sus aplicaciones.
2. Programar el sistema de medición mediante la tarjeta de desarrollo ESP32 y sensor DHT22.
3. Establecer la conectividad del sistema vía MQTT hacia la API en la nube para el envío de datos.
4. Diseñar un aplicativo móvil mediante Android Studio para la visualización de las mediciones realizadas.

1.6. Hipótesis.

Con este proyecto de graduación se diseña un sistema IoT que sea capaz de realizar la medición de temperatura y humedad de los dispositivos TI alojados en un ISP, monitoreando el estado de las variables importantes de los equipos ubicados dentro de los racks, evitando daños por valores ambientales fuera de lo establecido para la operación de los dispositivos.

1.7. Metodología de investigación.

Metodología Inductiva, Metodología de Análisis y Metodología de Síntesis.

Metodología Inductiva. - Este método permite la formulación de hipótesis, investigación y demostración. Donde se podrá generar una conclusión con los datos y requerimientos analizados.

Metodología de Análisis. - Se distinguen los elementos y se procede a revisar cada uno de ellos por separado para la efectiva descomposición de toda la información obtenida.

Metodología de Síntesis. - Combinación de elementos conceptos o partes de un todo, analizados exhaustivamente para llegar a una síntesis, es decir, reunir todos los elementos dispersos.

1.8. Tipo de Investigación.

El tipo de investigación será investigación aplicada ya que se tiene como objetivo resolver problemas prácticos y aplicar los conocimientos científicos y teóricos en situaciones reales. Buscar y desarrollar soluciones concretas y aplicables en el mundo real.

CAPÍTULO 2:

FUNDAMENTACIÓN TEÓRICA

2.1. Introducción de IoT.

El Internet de las cosas (IoT, por sus siglas en inglés) ha emergido como una de las tecnologías más innovadoras y disruptivas en la era digital actual. Su impacto se extiende más allá de los límites de la conectividad convencional, transformando la forma en que interactuamos con los objetos físicos y generando nuevas oportunidades en diversas áreas. En esta sección, exploraremos en detalle la definición y el concepto fundamental del IoT, sentando las bases para comprender su alcance y aplicaciones. (Mutlag et al., 2019)

2.1.1. Definición y concepto de IoT.

El Internet de las cosas se refiere a una red de objetos físicos interconectados que están equipados con sensores, actuadores y dispositivos electrónicos que les permiten recopilar y compartir datos a través de Internet. Estos objetos, que van desde dispositivos domésticos como electrodomésticos y sistemas de seguridad hasta maquinaria industrial y sensores ambientales, se comunican entre sí y con sistemas informáticos, generando información valiosa que se puede utilizar para tomar decisiones informadas y mejorar la eficiencia de diversos procesos.

El concepto fundamental del IoT se basa en la idea de que cualquier objeto físico puede convertirse en un "objeto inteligente" capaz de recopilar datos, transmitir información y responder a comandos o condiciones específicas. Estos objetos inteligentes están conectados a través de una infraestructura de red que les permite comunicarse de manera fluida y colaborativa. Además, el IoT se distingue por su capacidad para recopilar y analizar grandes cantidades de datos generados por los

dispositivos, lo que brinda una visión más profunda y en tiempo real de los sistemas físicos y su entorno.(Wang et al., 2021)

El IoT se basa en tres principios clave que son fundamentales para su funcionamiento, estos se describen a continuación en la tabla 2.1:

Tabla 2.1: Principios claves del IoT.

| | |
|-----------------------|--|
| Conectividad | El IoT se basa en la interconexión de objetos a través de redes de comunicación. Estas redes pueden ser inalámbricas, como Wi-Fi o Bluetooth, o utilizar tecnologías de comunicación de área amplia, como 5G. La conectividad permite que los objetos intercambien información y colaboren entre sí. |
| Sensores y actuadores | Los objetos del IoT están equipados con sensores y actuadores que les permiten capturar datos del entorno y tomar acciones en respuesta a ciertos estímulos. Los sensores recopilan información, como temperatura, humedad, luz o movimiento, mientras que los actuadores permiten que los objetos físicos realicen acciones físicas, como encender o apagar un dispositivo. |

| | |
|--------------------------|--|
| <p>Análisis de datos</p> | <p>El IoT se basa en el procesamiento y análisis de grandes volúmenes de datos generados por los objetos conectados. Estos datos se pueden utilizar para obtener información valiosa, identificar patrones, predecir eventos futuros y optimizar la toma de decisiones en tiempo real.</p> |
|--------------------------|--|

Fuente: (Wang et al., 2021)

2.1.2. Evolución histórica del IoT.

El Internet de las cosas (IoT) ha experimentado una evolución notable desde sus inicios hasta convertirse en una tecnología omnipresente en la sociedad actual. A lo largo de su historia, el IoT ha atravesado diversas etapas de desarrollo, impulsado por avances tecnológicos, cambios en la conectividad y una creciente demanda de interconexión de dispositivos. En esta sección, exploraremos la evolución histórica del IoT y los hitos importantes que han contribuido a su crecimiento.

Si bien el concepto del IoT se atribuye a Kevin Ashton, quien acuñó el término en 1999, sus raíces se remontan a décadas anteriores. En la década de 1970, el científico británico Mark Weiser propuso la idea de la "computación ubicua", donde los dispositivos electrónicos estarían presentes en todas partes y de forma invisible en nuestra vida cotidiana. Este concepto sentó las bases para la interconexión de objetos físicos(Wang et al., 2021).

2.1.2.1. La era de la identificación por radiofrecuencia (RFID).

En la década de 1990, la tecnología de identificación por radiofrecuencia (RFID) empezó a ganar popularidad. Los sistemas RFID permitían la identificación y seguimiento de objetos a través de etiquetas electrónicas. Esta tecnología allanó el camino para la interconexión de objetos físicos y fue un precursor importante del IoT.

2.1.2.2. Avances en la conectividad.

El desarrollo de tecnologías de comunicación inalámbrica fue un catalizador clave en la evolución del IoT. La adopción masiva de Wi-Fi y la proliferación de dispositivos móviles impulsaron la conectividad de dispositivos y permitieron una comunicación más fluida entre ellos. La disponibilidad de conexiones a Internet más rápidas y confiables también fue un factor determinante para el crecimiento del IoT.

2.1.2.3. Expansión de los sensores y dispositivos inteligentes.

A medida que la miniaturización de los componentes electrónicos avanzaba, los sensores y dispositivos inteligentes se volvieron más accesibles y económicos. Los avances en la tecnología de sensores permitieron la recopilación de datos en tiempo real sobre el entorno físico. Esto abrió la puerta a aplicaciones en áreas como la monitorización ambiental, la salud, la seguridad y la eficiencia energética.

2.1.2.4. Nacimiento de los ecosistemas IoT.

A medida que el IoT ganaba impulso, surgieron ecosistemas y estándares que promovían la interoperabilidad y la colaboración entre diferentes dispositivos y sistemas. Organizaciones como el Consorcio de Internet de las Cosas (IoT Consortium) y el Grupo de Trabajo de la Alianza Industrial de IoT (IIC) jugaron un

papel crucial en el desarrollo de marcos y directrices para la adopción del IoT en diversas industrias.

2.1.2.5. Integración con tecnologías emergentes.

El IoT se ha fusionado con otras tecnologías emergentes, lo que ha ampliado aún más su alcance y aplicaciones. La inteligencia artificial (IA) ha permitido la toma de decisiones autónoma y la automatización en tiempo real basada en los datos generados por el IoT. La computación en la nube ha proporcionado una plataforma escalable para el almacenamiento y análisis de grandes volúmenes de datos del IoT. Además, la llegada de la tecnología 5G ha impulsado la conectividad ultrarrápida y de baja latencia, allanando el camino para nuevas aplicaciones del IoT. (Lohachab & Jangra, 2019).

2.1.2.6. El IoT en la actualidad.

En la actualidad, el IoT está presente en una amplia gama de sectores y aplicaciones. La automatización del hogar, la agricultura inteligente, la industria 4.0, el cuidado de la salud, la logística y el transporte inteligentes son solo algunas de las áreas que se han beneficiado del IoT. A medida que la tecnología continúa avanzando, se espera que el IoT se integre aún más en nuestra vida diaria, mejorando la eficiencia, la comodidad y la calidad de vida, trabajando en conjunto con las tecnologías emergente como la IA, Nube hasta cubrir todas las áreas posibles como se muestra en la figura 2.1.



Figura 2.1: Internet de las cosas (IoT)

Fuente: (OpenMind/BBVA, 2023)

2.1.3. Aplicaciones y sectores clave del IoT.

El Internet de las cosas (IoT) ha transformado múltiples sectores de la sociedad, brindando una amplia gama de aplicaciones y oportunidades. Esta tecnología ha impulsado la interconexión de objetos físicos, permitiendo recopilar datos en tiempo real y tomar decisiones informadas para mejorar la eficiencia, la comodidad y la calidad de vida. En esta sección, exploraremos algunas de las aplicaciones y sectores clave en los que el IoT ha tenido un impacto significativo.

2.1.3.1. Automatización del hogar y edificios inteligentes.

Uno de los campos más destacados del IoT es la automatización del hogar y los edificios inteligentes. El IoT permite controlar y gestionar diversos dispositivos y sistemas en el hogar, como iluminación, climatización, electrodomésticos y sistemas de seguridad, a través de una red interconectada. Esto proporciona una mayor comodidad, seguridad y eficiencia energética, permitiendo a los usuarios controlar y monitorizar su hogar de forma remota.

2.1.3.2. Agricultura inteligente.

La agricultura inteligente se basa en la utilización de sensores y dispositivos conectados para optimizar la producción agrícola. El IoT permite monitorear y controlar factores como la humedad del suelo, la temperatura, la calidad del aire y la iluminación en tiempo real. Esto ayuda a los agricultores a tomar decisiones basadas en datos para maximizar los rendimientos, minimizar el uso de recursos y optimizar la gestión de plagas y enfermedades. (Asanza et al., 2021)

2.1.3.3. Salud y bienestar.

En el sector de la salud, el IoT ha abierto un abanico de posibilidades. Desde dispositivos portátiles para el monitoreo de la salud personal hasta sistemas de gestión de hospitales inteligentes, el IoT está transformando la forma en que se brinda atención médica. Los dispositivos conectados permiten el seguimiento continuo de parámetros vitales, la supervisión remota de pacientes, la gestión eficiente de inventarios de medicamentos y la mejora en la precisión y rapidez del diagnóstico médico. (Mutlag et al., 2019)

2.1.3.4. Industria manufacturera y logística.

El IoT ha revolucionado la industria manufacturera y la logística al permitir la monitorización en tiempo real de las operaciones y procesos. La interconexión de sensores y dispositivos inteligentes en las cadenas de producción proporciona información valiosa sobre el rendimiento, la calidad y el mantenimiento de las máquinas y equipos. Esto ayuda a optimizar la eficiencia de la producción, reducir los tiempos de inactividad y mejorar la planificación de la cadena de suministro, tal como se observa en la figura 2.2. (Khan et al., 2020)



Figura 2.2: IoT en la industria.
Fuente: (OpenMind/BBVA, 2023)

2.1.3.5. Ciudades inteligentes.

Las ciudades inteligentes utilizan el IoT para mejorar la calidad de vida de los ciudadanos y optimizar la gestión de los recursos. El monitoreo de sensores en tiempo real permite la administración eficiente de servicios como la iluminación pública, el estacionamiento, la gestión del tráfico y la recopilación de residuos. Además, el IoT se utiliza para mejorar la seguridad pública, la monitorización ambiental y la eficiencia energética en los entornos urbanos. (Aiello, 2022)

2.1.3.6. Transporte y logística inteligente.

El IoT está transformando la forma en que nos desplazamos y gestionamos la logística. Desde vehículos conectados y autónomos hasta sistemas de seguimiento y gestión de flotas, el IoT está mejorando la eficiencia y seguridad del transporte. Los datos en tiempo real sobre el tráfico, las condiciones de la carretera y el rendimiento de los vehículos permiten una planificación más precisa y una toma de decisiones basada en datos para optimizar las rutas y reducir los costos de operación.

2.1.3.7. Energía y gestión de recursos.

El IoT se utiliza en el sector energético para optimizar la generación, distribución y consumo de energía. Los sensores y dispositivos conectados permiten la monitorización y el control en tiempo real de la infraestructura energética, lo que ayuda a identificar oportunidades de ahorro energético y mejorar la eficiencia de los sistemas. Además, el IoT facilita la gestión inteligente del agua, la gestión de residuos y la conservación de recursos naturales. (Khan et al., 2020)

2.1.4. Beneficios y desafíos del IoT.

El Internet de las cosas (IoT) ha abierto un mundo de posibilidades y ha transformado múltiples aspectos de nuestra vida cotidiana. Sin embargo, junto con los beneficios que trae consigo, el IoT también presenta desafíos significativos. En esta sección, exploraremos tanto los beneficios, tabla 2.2, como los desafíos del IoT, tabla 2.3. (Lohachab & Jangra, 2019)

Tabla 2.2: Beneficios del IoT.

| | |
|-------------------|--|
| Beneficios | <p>Eficiencia mejorada: Una de las ventajas clave del IoT es la mejora de la eficiencia en diversos sectores. Mediante la recopilación y análisis de datos en tiempo real, el IoT permite optimizar procesos, reducir tiempos de inactividad y minimizar el desperdicio de recursos. Esto se traduce en una mayor productividad y una reducción de costos para las empresas.</p> |
| | <p>Toma de decisiones informadas: El IoT proporciona una gran cantidad de datos que pueden ser utilizados para tomar decisiones más informadas y estratégicas. Al recopilar información precisa y en tiempo real, el IoT permite comprender mejor los patrones, tendencias y necesidades de los usuarios y clientes, lo que facilita la personalización de productos y servicios, así como la identificación de oportunidades de mejora.</p> |
| | <p>Mejora de la calidad de vida: El IoT ha mejorado la calidad de vida de las personas al brindarles mayor comodidad, seguridad y bienestar. En el hogar, los dispositivos conectados permiten la automatización y control remoto de diversos sistemas, como iluminación, climatización y seguridad. Además, en el ámbito de la salud, el IoT ha permitido el desarrollo de dispositivos médicos inteligentes, monitoreo remoto de pacientes y sistemas de atención médica más eficientes.</p> |

| | |
|--|--|
| | <p>Optimización de la infraestructura urbana: El IoT desempeña un papel fundamental en el desarrollo de ciudades inteligentes. Mediante el uso de sensores y dispositivos conectados, las ciudades pueden recopilar datos en tiempo real sobre el tráfico, la gestión de residuos, el consumo de energía y otros aspectos clave de la infraestructura urbana. Esto permite una planificación más efectiva, una utilización más eficiente de los recursos y una mejora general de la calidad de vida en entornos urbanos.</p> |
|--|--|

Fuente: (Lohachab & Jangra, 2019)

Tabla 2.3: Desafíos del IoT.

| | |
|-----------------|--|
| Desafíos | <p>Seguridad: Uno de los principales desafíos del IoT es la seguridad. La interconexión de dispositivos y la recopilación de grandes cantidades de datos plantean riesgos significativos en términos de ciberseguridad. Los dispositivos IoT pueden ser vulnerables a ataques y hackeos, lo que podría comprometer la privacidad de los usuarios y poner en riesgo la integridad de los sistemas y la infraestructura.</p> |
| | <p>Privacidad de los datos: El IoT implica la recopilación y el análisis de datos personales y sensibles. Esto plantea preocupaciones sobre la privacidad y el uso adecuado de la información recopilada. Es fundamental establecer políticas y</p> |

| | |
|--|---|
| | regulaciones sólidas para proteger la privacidad de los usuarios y garantizar que los datos sean utilizados de manera ética y segura. |
| | Interoperabilidad: La falta de estándares y la heterogeneidad de los dispositivos y sistemas IoT pueden dificultar la interoperabilidad. Esto puede dificultar la integración y la comunicación efectiva entre diferentes dispositivos y plataformas, lo que limita el potencial del IoT y dificulta su adopción generalizada. |
| | Escalabilidad y gestión de la complejidad: A medida que el número de dispositivos conectados aumenta exponencialmente, surge el desafío de escalar las infraestructuras y gestionar la complejidad asociada. La gestión de grandes volúmenes de datos, la configuración y actualización de dispositivos, y el mantenimiento de la infraestructura requieren soluciones robustas y eficientes. |

Fuente: (Lohachab & Jangra, 2019)

2.1.5. Tecnologías clave del IoT.

El Internet de las cosas (IoT) se basa en la interconexión de dispositivos físicos a través de una red para recopilar y compartir datos. Para hacer posible esta interconexión, existen varias tecnologías clave que permiten el funcionamiento y la comunicación efectiva del IoT. En la tabla 2.4, exploraremos algunas de estas tecnologías fundamentales del IoT. (Lohachab & Jangra, 2019)

Tabla 2.4: Tecnologías claves del IoT.

| | |
|--|---|
| <p>Sensores y Actuadores</p> | <p>Los sensores y actuadores son componentes esenciales del IoT. Los sensores recopilan datos del entorno, como temperatura, humedad, luz, movimiento, y los convierten en señales electrónicas. Estos datos son fundamentales para monitorear y controlar diferentes aspectos del IoT. Los actuadores, por otro lado, toman señales electrónicas y generan acciones físicas, como encender o apagar dispositivos. Los sensores y actuadores permiten la recopilación de datos y la respuesta en tiempo real en el IoT.</p> |
| <p>Protocolos de Comunicación</p> | <p>Los protocolos de comunicación son los medios utilizados para que los dispositivos del IoT se comuniquen entre sí. Estos protocolos definen el formato y las reglas para el intercambio de información. Algunos de los protocolos de comunicación comunes en el IoT incluyen MQTT (Message Queuing Telemetry Transport), CoAP (Constrained Application Protocol) y HTTP (Hypertext Transfer Protocol). Estos protocolos permiten la transmisión eficiente de datos entre los dispositivos del IoT, garantizando una comunicación confiable y segura.</p> |

| | |
|--------------------------------------|---|
| <p>Redes Inalámbricas</p> | <p>Las redes inalámbricas son fundamentales para el funcionamiento del IoT, ya que permiten la conectividad y la comunicación sin necesidad de cables físicos. Algunas de las tecnologías de red inalámbrica utilizadas en el IoT incluyen Wi-Fi, Bluetooth, Zigbee y 4G/5G. Estas redes inalámbricas proporcionan la capacidad de transmitir datos de manera eficiente a largas distancias y conectar una amplia gama de dispositivos en el IoT.</p> |
| <p>Computación en la Nube</p> | <p>La computación en la nube juega un papel crucial en el IoT al proporcionar almacenamiento y procesamiento de datos a gran escala. Los datos recopilados por los dispositivos del IoT se envían a la nube para su almacenamiento y análisis. La computación en la nube permite la escalabilidad, flexibilidad y disponibilidad de datos, lo que permite realizar análisis avanzados y generar información valiosa para la toma de decisiones.</p> |
| <p>Edge Computing</p> | <p>Si bien la computación en la nube es fundamental para el IoT, el edge computing ha surgido como una tecnología complementaria. El edge computing implica procesar y analizar los datos del IoT en el extremo de la red, cerca de donde se generan los datos, en lugar de enviarlos a la</p> |

| | |
|--|--|
| | nube. Esto permite una mayor velocidad de procesamiento y una menor latencia, lo que es fundamental en aplicaciones que requieren una respuesta en tiempo real, como la industria manufacturera o los vehículos autónomos. |
| Inteligencia Artificial y Análisis de Datos | La Inteligencia Artificial (IA) y el análisis de datos desempeñan un papel importante en el IoT al permitir el procesamiento y análisis avanzados de los datos recopilados. La IA permite la automatización y la toma de decisiones inteligentes, mientras que el análisis de datos permite extraer información valiosa y patrones significativos de los grandes volúmenes de datos generados por el IoT. Estas tecnologías son fundamentales para aprovechar al máximo los datos del IoT y obtener información útil para mejorar la eficiencia y la toma de decisiones. |

Fuente: (Lohachab & Jangra, 2019)

2.1.6. Estándares y protocolos del IoT.

El Internet de las cosas (IoT) es una red compleja y diversa de dispositivos interconectados que requiere estándares y protocolos adecuados para garantizar una comunicación efectiva, segura y interoperable. En esta sección, exploraremos algunos de los estándares y protocolos clave del IoT que permiten la integración y la interacción fluida entre dispositivos y sistemas. (Dizdarević et al., 2019)

2.1.6.1. MQTT (Message Queuing Telemetry Transport).

MQTT es un protocolo de mensajería ligero y de bajo consumo de energía diseñado para entornos de IoT con ancho de banda limitado y dispositivos con recursos limitados. Es altamente eficiente y permite la comunicación bidireccional entre dispositivos a través de un intermediario llamado bróker, tal como se visualiza en la figura 2.3. MQTT se utiliza comúnmente en aplicaciones de monitoreo y control, y es ampliamente adoptado en el campo del IoT. (Bansal & Priya, 2021)

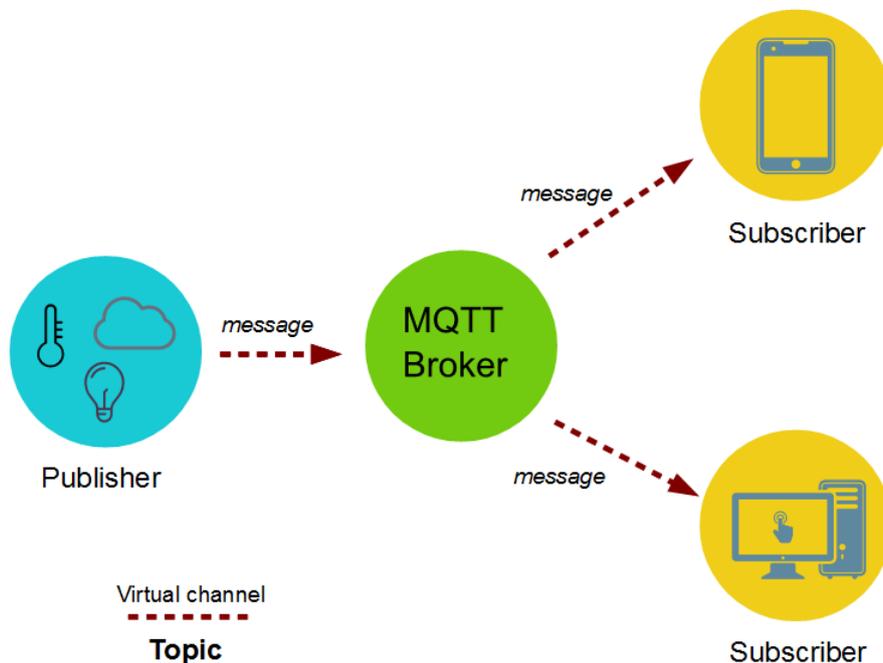


Figura 2.3: MQTT.
Fuente: (SWA, 2022)

2.1.6.2. CoAP (Constrained Application Protocol).

CoAP es un protocolo de aplicación diseñado específicamente para dispositivos con recursos limitados, como sensores y actuadores en el IoT. Es un

protocolo ligero y eficiente que permite la comunicación RESTful (Representational State Transfer) sobre protocolos de transferencia como UDP (User Datagram Protocol), tal como se muestra en la figura 2.4. CoAP es ideal para aplicaciones en las que la eficiencia energética y la baja sobrecarga de red son críticas, como la monitorización ambiental y la gestión de edificios.(Bansal & Priya, 2021)

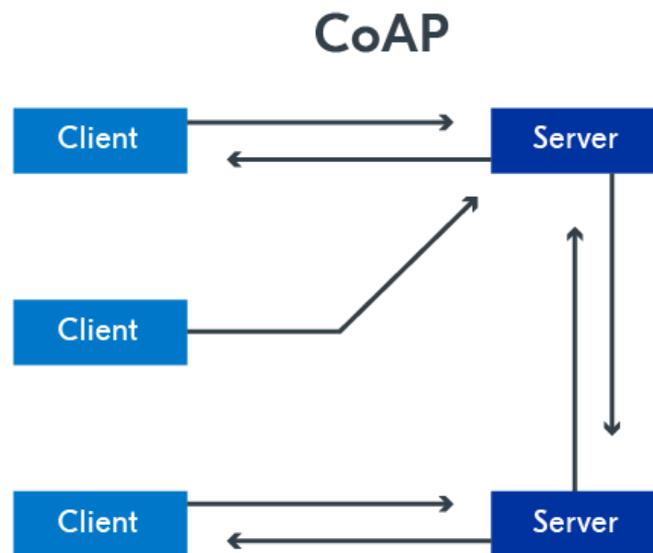


Figura 2.4: CoAP.
Fuente: (DevAcademy, 2022)

2.1.6.3. HTTP (Hypertext Transfer Protocol).

Aunque HTTP no fue diseñado específicamente para el IoT, sigue siendo ampliamente utilizado en aplicaciones de IoT, especialmente en aplicaciones basadas en la web. HTTP permite la comunicación entre dispositivos a través de solicitudes y respuestas basadas en el modelo cliente-servidor. Es flexible, ampliamente compatible y ofrece una amplia gama de funciones y características, lo que lo hace adecuado para aplicaciones de IoT que requieren interacciones más complejas y acceso a servicios web.

2.1.6.4. Bluetooth y Bluetooth Low Energy (BLE),

Bluetooth es una tecnología inalámbrica ampliamente utilizada para la comunicación de corto alcance entre dispositivos en el IoT. Bluetooth Low Energy (BLE) es una versión de bajo consumo de energía de Bluetooth que se ha convertido en un estándar en el IoT, especialmente en aplicaciones de dispositivos portátiles y de monitoreo de salud. BLE permite la comunicación eficiente y de bajo consumo de energía entre dispositivos cercanos, lo que lo hace ideal para escenarios de IoT en los que la vida útil de la batería es crítica.

2.1.6.5. Zigbee.

Zigbee es un estándar de comunicación inalámbrica de bajo consumo de energía diseñado específicamente para aplicaciones de IoT. Es ideal para redes de sensores inalámbricos y dispositivos de control de bajo costo, lo cual se evidencia en la figura 2.5. Zigbee ofrece una amplia cobertura, interoperabilidad y capacidad de autoorganización en redes de dispositivos IoT. (Ghotbou & Khansari, 2021)

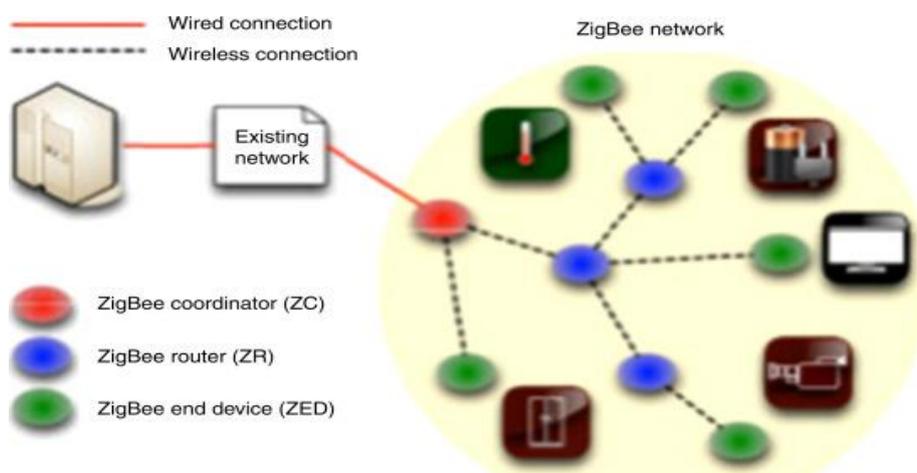


Figura 2.5: Zigbee.

Fuente: (Singh & Tomar, 2020)

2.1.6.6. LoRaWAN (Long Range Wide Area Network).

LoRaWAN es un protocolo de comunicación de largo alcance y baja potencia diseñado para permitir la conectividad de larga distancia en aplicaciones de IoT. Utiliza tecnología de modulación de espectro ensanchado para proporcionar una mayor cobertura y penetración de señal, lo que lo hace adecuado para aplicaciones como ciudades inteligentes, agricultura y monitoreo ambiental, tal como se muestra en la figura 2.6.(Ghotbou & Khansari, 2021)

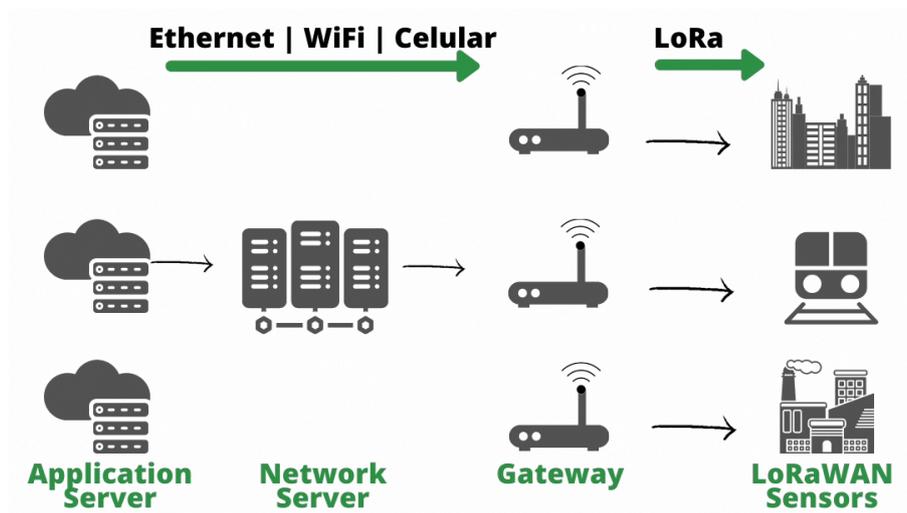


Figura 2.6: LoRaWAN.
Fuente: (Singh & Tomar, 2020)

2.1.6.7. OPC UA (Unified Architecture).

OPC UA es un estándar de comunicación industrial ampliamente utilizado en aplicaciones de IoT en entornos de automatización industrial. Proporciona una arquitectura unificada y segura para el intercambio de datos y la interoperabilidad entre dispositivos y sistemas de diferentes proveedores, tal como se muestra en la figura 2.7.

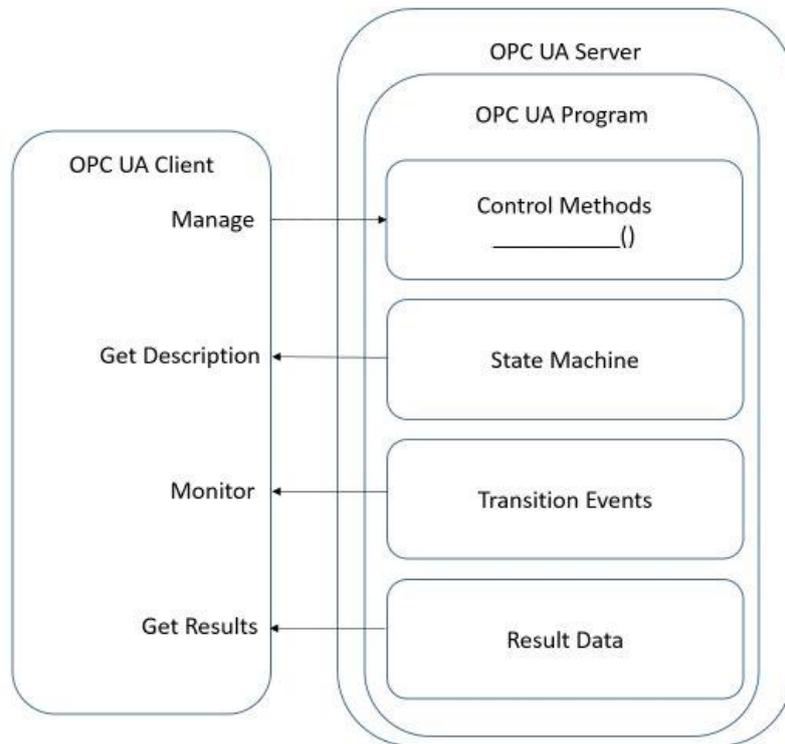


Figura 2.7: OPC UA.
Fuente: (Dorofeev & Zoitl, 2018)

2.2. Datacenters y su importancia en la infraestructura tecnológica.

Los datacenters son componentes fundamentales en la infraestructura tecnológica actual. Estos centros de procesamiento de datos albergan servidores, sistemas de almacenamiento y equipos de red que permiten el funcionamiento de una amplia gama de servicios y aplicaciones digitales. En esta sección, exploraremos la importancia de los datacenters en la infraestructura tecnológica y su papel crucial en la era digital.

2.2.1. Definición y características de un datacenter.

Un datacenter es una instalación física diseñada para alojar servidores, equipos de red y sistemas de almacenamiento. Está diseñado para proporcionar un entorno controlado y seguro para el procesamiento y almacenamiento de datos. Los datacenters varían en tamaño y capacidad, desde pequeñas salas de servidores hasta

grandes complejos de varios pisos. Estas instalaciones están equipadas con sistemas de enfriamiento, alimentación eléctrica ininterrumpida (UPS), sistemas de seguridad, monitoreo para garantizar un funcionamiento confiable y seguro entre otras herramientas que son necesarias para el correcto funcionamiento, tal como se puede observar en la figura 2.8.

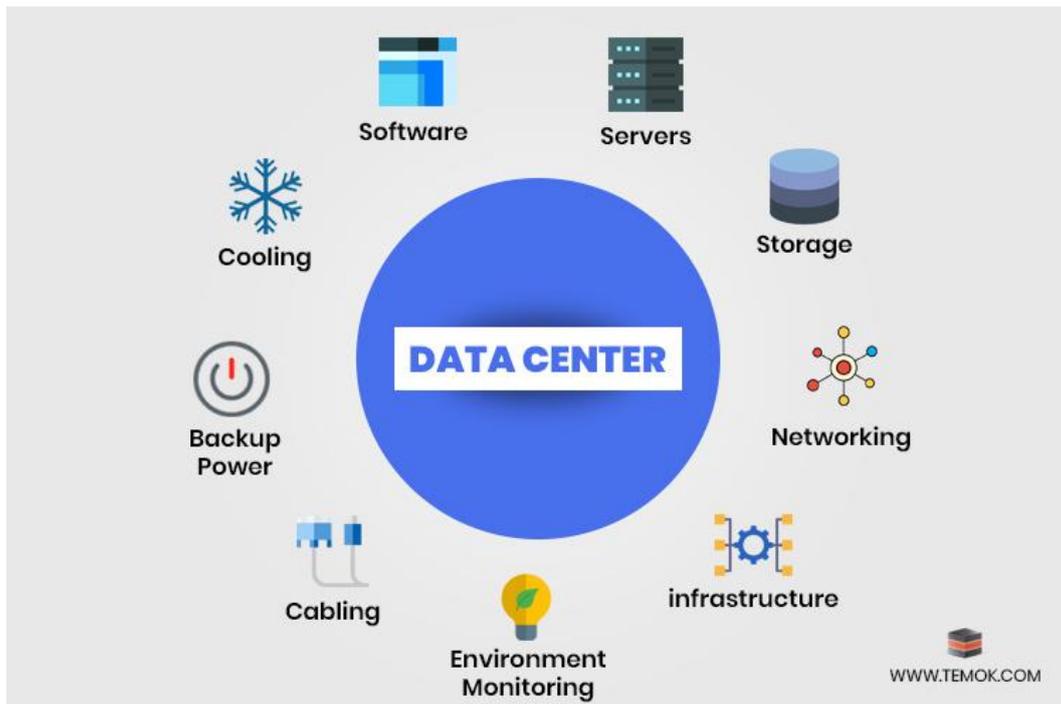


Figura 2.8: Equipamientos de un datacenter.
Fuente: (TEMOK, 2022)

2.2.2. Funciones y componentes de un datacenter.

Los datacenters desempeñan diversas funciones vitales para la infraestructura tecnológica. Algunas de las funciones se muestran en la tabla 2.4

Tabla 2.4: Funciones de un datacenters.

| | |
|---------------------------------------|---|
| <p>Procesamiento de datos</p> | <p>Realizan tareas de procesamiento intensivo de datos, como cálculos complejos, análisis de big data y ejecución de aplicaciones empresariales críticas.</p> |
| <p>Almacenamiento de datos</p> | <p>Proporcionan capacidad de almacenamiento para guardar y gestionar grandes volúmenes de información. Esto incluye datos de usuarios, bases de datos, archivos multimedia y documentos corporativos.</p> |
| <p>Conectividad</p> | <p>Actúan como centros neurálgicos para la conectividad de redes. Establecen y gestionan conexiones de alta velocidad para garantizar una comunicación fluida entre los servidores y los dispositivos de usuarios finales.</p> |
| <p>Seguridad de datos</p> | <p>Implementan medidas de seguridad avanzadas para proteger los datos almacenados. Esto incluye sistemas de cortafuegos, encriptación, autenticación y monitoreo constante para detectar posibles amenazas y garantizar la integridad de los datos.</p> |
| <p>Disponibilidad</p> | <p>Están diseñados para garantizar la disponibilidad continua de los servicios. Esto implica la redundancia de equipos, sistemas de respaldo y</p> |

| | |
|--|--|
| | <p>planes de recuperación ante desastres para minimizar el tiempo de inactividad y mantener los servicios en funcionamiento las 24 horas del día, los 7 días de la semana.</p> |
|--|--|

Fuente: (Lohachab & Jangra, 2019)

2.2.3. Importancia de la temperatura y humedad en un datacenter.

La temperatura y humedad son dos variables críticas que deben ser monitoreadas y controladas en un datacenter. Estos parámetros tienen un impacto significativo en el rendimiento, confiabilidad y vida útil de los equipos y sistemas informáticos presentes en el datacenter, por lo cual existen límites superiores e inferiores para la operación de dichos equipos, tal como se muestra en la figura 2.9.

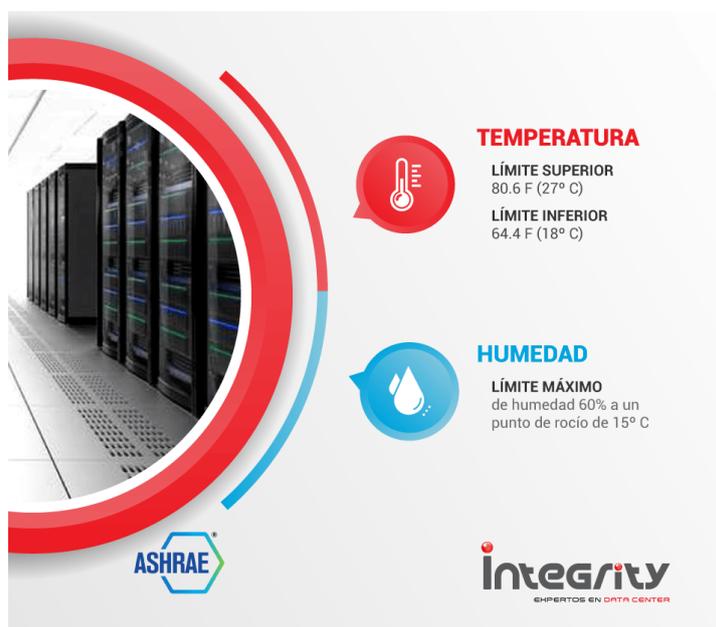


Figura 2.9: Límites operacionales.
Fuente: (Integrity, 2023)

2.3. Medición de temperatura y humedad en el entorno del datacenter

La temperatura y la humedad son dos variables críticas que desempeñan un papel fundamental en el funcionamiento y la confiabilidad de un datacenter. En esta sección, exploraremos la importancia de mantener condiciones óptimas de temperatura y humedad en un entorno de datacenter.

2.3.1. Importancia de la medición de temperatura y humedad en un datacenter.

La medición precisa y constante de la temperatura y humedad en un datacenter es esencial por varias razones, las cuales se describirán en la tabla 2.5.

Tabla 2.5: Importancia de mediciones.

| | |
|--|---|
| Rendimiento y confiabilidad de los equipos: | Los componentes electrónicos y servidores dentro de un datacenter generan calor durante su funcionamiento. Si la temperatura ambiente es demasiado alta, puede provocar un aumento en la temperatura interna de los equipos, lo que afecta su rendimiento y puede provocar fallas prematuras. Además, los niveles de humedad incorrectos pueden causar corrosión y daños en los componentes electrónicos, lo que también puede llevar a un mal funcionamiento y tiempos de inactividad. |
|--|---|

| | |
|---|---|
| <p style="text-align: center;">Eficiencia energética</p> | <p>La medición y el control precisos de la temperatura en un datacenter permiten optimizar la eficiencia energética. Mantener la temperatura adecuada evita el desperdicio de energía al evitar el sobre enfriamiento o el sobrecalentamiento. Además, la medición de la humedad puede ayudar a regular el uso de equipos de humidificación o deshumidificación, lo que también contribuye a la eficiencia energética del datacenter.</p> |
| <p style="text-align: center;">Cumplimiento de estándares y regulaciones</p> | <p>Existen estándares y regulaciones que establecen los rangos de temperatura y humedad aceptables en un datacenter. Cumplir con estos estándares es esencial para garantizar el correcto funcionamiento de los equipos y para cumplir con las normativas de seguridad y confiabilidad de los datos.</p> |

Fuente: (Lohachab & Jangra, 2019)

2.3.2. Parámetros y requisitos de temperatura y humedad en un datacenter.

Los parámetros y requisitos de temperatura y humedad en un datacenter pueden variar según la naturaleza de las operaciones y los equipos presentes. Sin embargo, algunos parámetros comunes incluyen:

a) Temperatura: Los datacenters generalmente mantienen una temperatura ambiente controlada entre 20°C y 25°C. Esto proporciona un equilibrio entre el rendimiento óptimo de los equipos y la eficiencia energética. Es importante evitar fluctuaciones bruscas de temperatura, ya que esto puede causar estrés térmico en los componentes.

b) Humedad relativa: La humedad relativa recomendada en un datacenter generalmente se encuentra entre el 40% y el 60%. Este rango permite prevenir problemas relacionados con la electricidad estática y la condensación, al tiempo que evita la corrosión y el daño a los componentes electrónicos.

2.3.3. Métodos y tecnologías tradicionales de medición de temperatura y humedad.

Existen varios métodos y tecnologías tradicionales utilizados para medir la temperatura y humedad en un datacenters, que incluyen:

a) Termómetros y termohigrómetros: Estos dispositivos analógicos o digitales miden la temperatura y la humedad relativa del entorno. Son ampliamente utilizados y ofrecen una medición básica y asequible.

b) Sensores de temperatura y humedad: Estos sensores electrónicos proporcionan mediciones más precisas y pueden ser integrados en sistemas de monitoreo y control automatizados, en especial los sensores que trabajan con

protocolos industriales, como el mostrado en la figura 2.10. Los sensores pueden ser conectados a través de cables o utilizar tecnologías inalámbricas, lo que permite una mayor flexibilidad en la ubicación y recopilación de datos.



Figura 2.10: Límites operacionales.
Fuente: (EtherPower, 2023)

2.3.4. Limitaciones y desafíos de los métodos tradicionales de medición.

Aunque los métodos tradicionales de medición de temperatura y humedad son ampliamente utilizados, presentan ciertas limitaciones y desafíos, como las que se muestran en la tabla 2.6:

Tabla 2.6: Limitaciones y desafíos.

| | |
|----------------------------------|---|
| <p>Precisión limitada</p> | <p>Algunos de estos métodos pueden tener una precisión limitada, lo que puede afectar la confiabilidad de las mediciones. Esto es especialmente importante en</p> |
|----------------------------------|---|

| | |
|--------------------------------|--|
| | entornos sensibles como los datacenters, donde una medición precisa es crucial. |
| Monitoreo manual | Algunos dispositivos de medición tradicionales requieren una intervención manual para realizar la medición. Esto puede ser incómodo y propenso a errores humanos. |
| Escalabilidad y gestión | La gestión de múltiples puntos de medición manualmente puede volverse compleja y difícil de escalar en grandes datacenters. La necesidad de recopilar y registrar datos de manera constante puede generar un costo y esfuerzo significativo. |

Fuente: (Lohachab & Jangra, 2019)

Sensores IoT para la medición de temperatura y humedad.

La medición de temperatura y humedad en un datacenter mediante sensores IoT implica la utilización de dispositivos especializados que pueden recopilar y transmitir datos de manera inalámbrica. Estos sensores son capaces de medir con precisión y consistencia los valores de temperatura y humedad en diferentes puntos del entorno del datacenter.

Existen diferentes tipos de sensores IoT utilizados para la medición de temperatura y humedad, que incluyen:

- a) **Sensores de temperatura:** Estos sensores utilizan elementos sensibles al calor, como termistores o termopares, para medir la temperatura ambiente. Pueden ser de contacto o sin contacto, y pueden ofrecer una alta precisión y respuesta rápida.

- b) Sensores de humedad: Los sensores de humedad miden la cantidad de vapor de agua presente en el aire. Utilizan diferentes tecnologías, como capacitancia, resistencia o resonancia, para detectar cambios en la humedad relativa. Los sensores de humedad pueden ser combinados con sensores de temperatura para medir ambas variables de manera simultánea.

2.3.5. Infraestructura de red para la conectividad en un entorno de datacenter.

La implementación de IoT en la medición de temperatura y humedad en datacenters requiere una infraestructura de red adecuada para garantizar la conectividad confiable y segura de los dispositivos IoT, tal como se muestra en la figura 2.11.

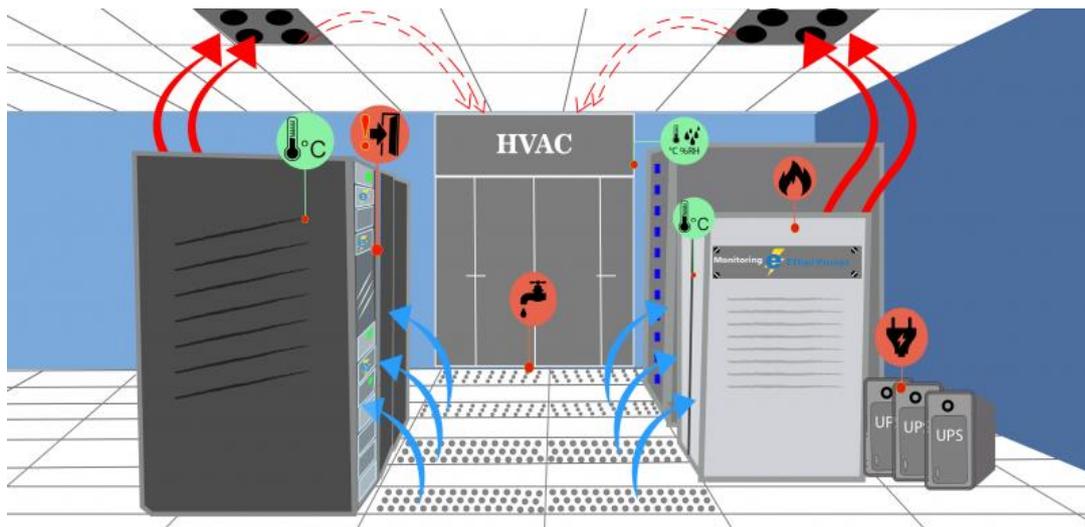


Figura 2.11: Infraestructura para garantizar seguridad en el entorno.
Fuente: (EtherPower, 2023)

Algunos aspectos importantes de la infraestructura de red en un entorno de datacenter incluyen:

- a) Conectividad inalámbrica: La utilización de tecnologías inalámbricas, como Wi-Fi o redes celulares, permite una mayor flexibilidad en la ubicación de los sensores IoT y evita la necesidad de cables físicos. Sin embargo, se debe

garantizar una cobertura adecuada y una configuración segura para proteger los datos transmitidos.

- b) Redes de área local (LAN): Las redes de área local dentro del datacenter proporcionan la conectividad local necesaria para los dispositivos IoT. Se debe garantizar un ancho de banda adecuado y una infraestructura de red confiable para soportar la comunicación entre los sensores IoT, los sistemas de gestión y los sistemas de monitoreo y control.
- c) Seguridad de red: La seguridad de la red es crucial en un entorno de datacenter, especialmente cuando se trata de la transmisión de datos sensibles. Se deben implementar medidas de seguridad, como autenticación de dispositivos, cifrado de datos y firewalls, para proteger la integridad y confidencialidad de la información transmitida.

CAPÍTULO 3: DISEÑO Y RESULTADOS

En este capítulo se presentarán los detalles para el diseño del prototipo del sistema de medición, finalmente se mostrará el código, hardware necesario, protocolos y tecnologías usadas.

3.1. Descripción general del sistema IoT para medición.

El sistema cuenta de 3 partes o etapas, tal como se puede visualizar en la figura 3.1, la primera que corresponde a la recolección de datos o medición de variables de temperatura y humedad, la segunda etapa que corresponde al envío de información a la nube, Ubidots, desde la cual se obtendrá la información, por último, se tiene la tercera etapa que es en la cual se obtendrán los valores y se visualizará en los dispositivos.

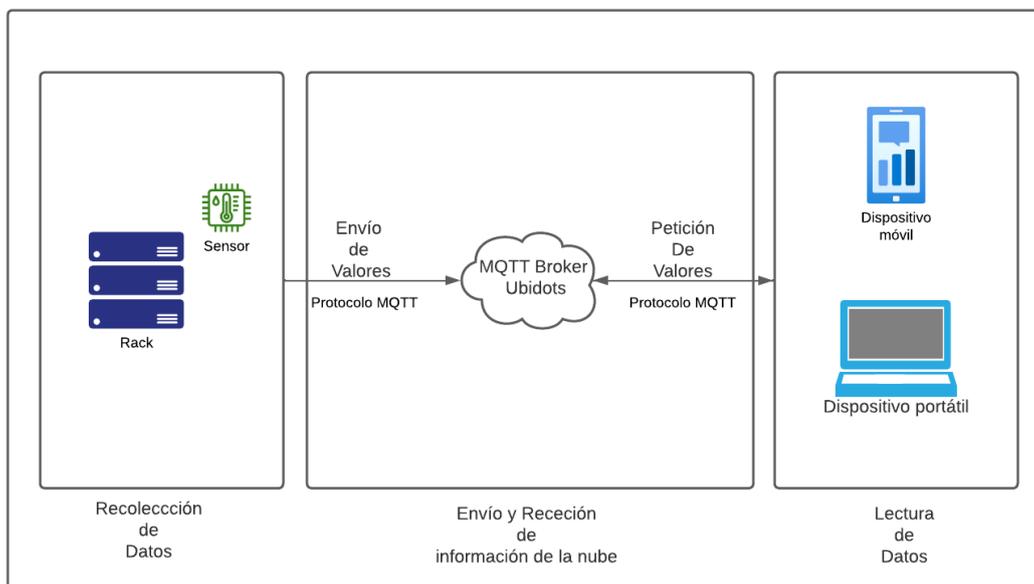


Figura 3.1: Esquema del sistema IoT para medición.
Elaborado por: Autor.

3.2. Primera etapa del sistema IoT para medición.

La primera etapa corresponde a la recolección de datos, esta etapa está conformada por la placa de desarrollo ESP32, figura 3.2, junto al sensor para obtener los valores de temperatura y humedad, figura 3.3.

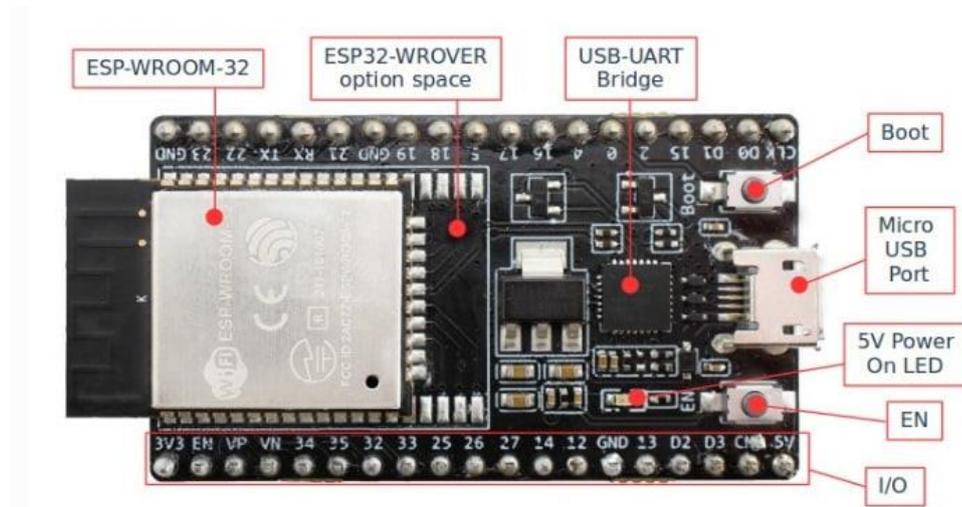


Figura 3.2: Módulo ESP32.
Elaborado por: (Electronics Mouser, 2022).



Figura 3.3: Sensor DHT22.
Elaborado por: (Electronics Mouser, 2022).

Estos dos elementos se conectan y comunican entre sí para obtener los valores de temperatura y humedad de los racks, tal como se observa en la figura 3.4.

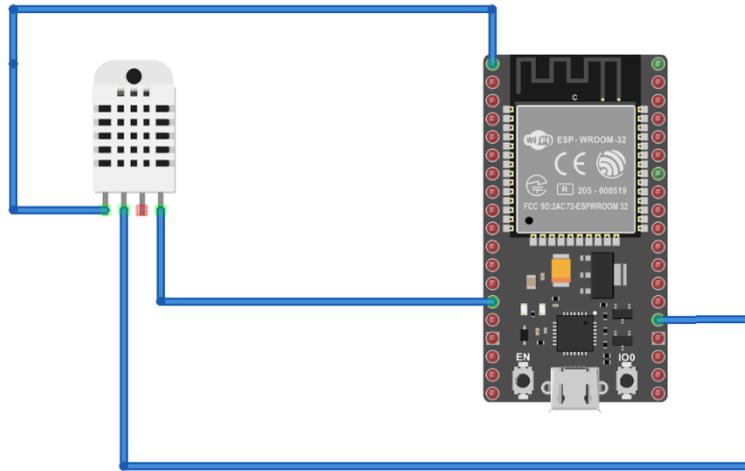


Figura 3.4: Diagrama de conexión entre ESP32 y DHT22.
Elaborado por: Autor.

Posteriormente el ESP32 se conecta mediante WIFI a la nube para enviar la información obtenida, esto se detallará a profundidad en el punto 3.3. Para que el ESP32 reciba los datos, se conectó a la red inalámbrica y realice el envío de la información a la nube se le debe establecer funciones mediante programación, esta programación se la hace mediante Arduino, el ESP32 debe ser conectado de forma cableada a una computadora y configurado mediante Arduino, el código utilizado se muestra en la figura 3.5. Los bloques de código mostrados en la figura 3.5 se establecen las librerías y funciones necesarias para que el ESP32 se conecte a internet.

```

#include <DHT.h>
#include "UbidotsEsp32Mqtt.h" //Cargamos la librería DHT
#define DHTTYPE DHT22 //Definimos el modelo del sensor (hay //otros DHT)
#define DHTPIN 2 // Se define el pin 2 del Arduino para conectar el sensor DHT11

DHT dht(DHTPIN, DHTTYPE, 22);

const char *UBIDOTS_TOKEN = "BBFF-mJErTxhd6IfCEAuHTPLYMxTFTI82zt";
const char *WIFI_SSID = "NotQuery";
const char *WIFI_PASS = "L4bLRD$@";
const char *DEVICE_LABEL = "esp32";
const char *VARIABLE_LABEL_1 = "temperatura";
const char *VARIABLE_LABEL_2 = "humedad";
const int PUBLISH_FREQUENCY = 5000;
int timer;

Ubidots ubidots(UBIDOTS_TOKEN);

void callback(char *topic, byte *payload, unsigned int length)
{
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++)
  {
    Serial.print((char)payload[i]);
  }
  Serial.println();
}

void setup()
{
  Serial.begin(9600); //Se inicia la comunicación serial
  dht.begin();
  delay(1000);
  ubidots.connectToWifi(WIFI_SSID, WIFI_PASS);
  ubidots.setCallback(callback);
  ubidots.setup();
  ubidots.reconnect();
  timer = millis();
}

```

Figura 3.5: Código cargado en la ESP32 1/2.
Elaborado por: Autor.

Por otro lado, en la figura 3.6 se establece un lazo que se repetirá cada 10 segundos, en el cual se lee las variables medidas desde el DHT22 y posteriormente las escribe en Ubidots.

```

void loop()
{
  if (!ubidots.connected())
  {
    ubidots.reconnect();
  }
  if (millis() - timer > PUBLISH_FREQUENCY)
  {
    /*Hacemos la medición de Temperatura y Humedad del sensor, y lo definimos en variables Float */
    float h = dht.readHumidity();
    float t = dht.readTemperature();

    /*Definimos que el valor de Temperatura será enviado por la variable 1, a la plataforma Ubidots*/
    ubidots.add(VARIABLE_LABEL_1, t);
    /*Definimos que el valor de Humedad será enviado por la variable 2, a la plataforma Ubidots*/
    ubidots.add(VARIABLE_LABEL_2, h);
    /*Hacemos la publicación de los datos en el dispositivo definido*/
    ubidots.publish(DEVICE_LABEL);
    /*Para mostrar los datos, los imprimimos en el terminal Serial*/
    Serial.println("Enviando los datos a Ubidots: ");
    Serial.println("Temperatura: " + String(t));
    Serial.println("Humedad: " + String(h));
    Serial.println("-----");
    timer = millis();
  }
  /*Agregamos un delay de 10 segundos para el envío, y que luego de este tiempo, se proceda a reiniciar el bucle*/
  delay(10000);
  ubidots.loop();

  //Se espera 2 segundos para seguir leyendo datos
}

```

Figura 3.6: Código cargado en la ESP32 2/2.
Elaborado por: Autor.

3.3. Segunda etapa del sistema IoT para medición.

La segunda etapa corresponde a la subida de información a Ubidots, nube, esto se lo realiza mediante el protocolo MQTT, en la figura 3.6 previamente se mostró el bloque de código correspondiente a esta tarea. El ESP32 luego de obtener los valores medidos desde el DHT22 se conecta a la red WIFI previamente configurada y publica los valores en el servidor remoto, Ubidots, de tal manera que la información queda disponible para que cualquier dispositivo IoT, teléfono o computadora pueda obtener dichos valores, tal como se muestra en la figura 3.7.

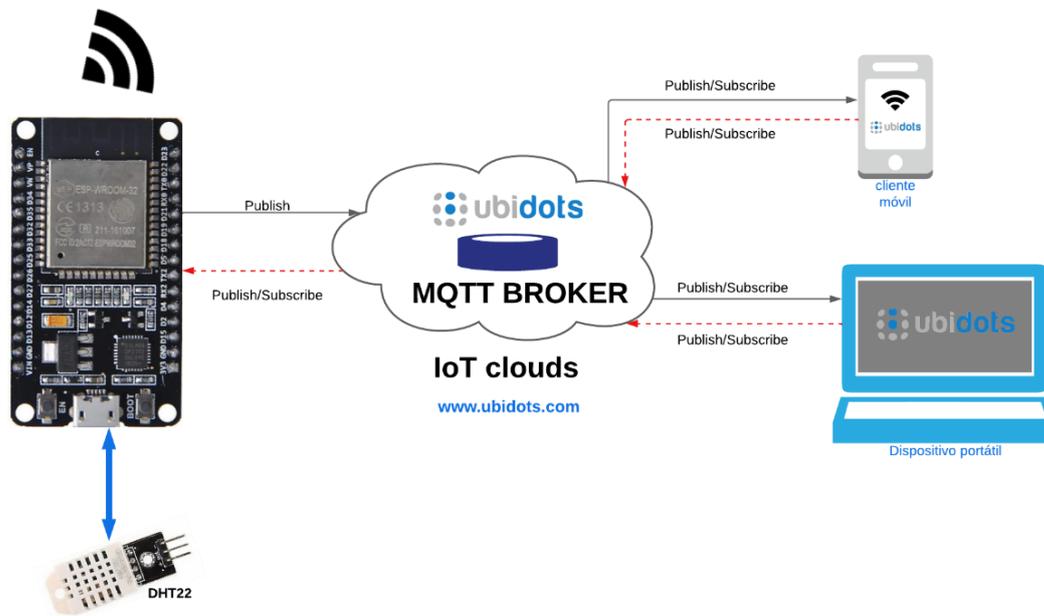


Figura 3.7: Diagrama de envío de valores medidos a la nube.
Elaborado por: Autor.

3.4. Tercera etapa del sistema IoT para medición.

En esa última etapa se realiza la lectura o visualización de los valores medidos, para esta tarea se tienen una aplicación móvil, la cual se desarrolló en Flutter, y por medio de esta se puede visualizar los valores de temperatura y humedad. A continuación, detallaré la arquitectura principal de dicha aplicación móvil. En la figura 3.8 se muestra la clase en la cual se define los métodos para la lectura de los valores de temperatura y humedad escritos en Ubidots.

```

import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

class Datos_temp {

  final int userId;
  final int id;
  final String title;
  final String body;

  Datos_temp({required this.userId, required this.id, required this.title, required this.body});

  factory Datos_temp.fromJson(Map<String, dynamic> json) {
    return Datos_temp(
      userId: json['userId'],
      id: json['id'],
      title: json['title'],
      body: json['body'],
    );
  }
}

```

Figura 3.8: Clase “Datos_temp.dart”.

Elaborado por: Autor.

En las figuras 3.9, 3.10 y 3.11 se muestra la clase que estructura la ventana y contenedores en los cuales se muestran las variables de temperatura y humedad.

```

import 'package:flutter/material.dart';
import 'package:tempapp/maps.dart';
import 'package:tempapp/modo.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';

String url = 'https://industrial.api.ubidots.com/api/v1.6/devices/esp32/humedad/lv/?token=BBFF-mJErTxhd6IFCEAuHTPLYMxTFTI82zt';
String url1 = "https://industrial.api.ubidots.com/api/v1.6/devices/esp32/temperatura/lv/?token=BBFF-mJErTxhd6IFCEAuHTPLYMxTFTI82zt";
var valor1;
var valorh;
Future<String> fetchPost() async {
  final response =
    await http.get(Uri.parse(url));

  if (response.statusCode == 200) {
    valor1 = json.decode(response.body).toString();
    // Si la llamada al servidor fue exitosa, analiza el JSON

    return valor1;
  } else {
    // Si la llamada no fue exitosa, lanza un error.
    throw Exception('Failed to load post');
  }
}

Future<String> fetchPost1() async {
  final response =
    await http.get(Uri.parse(url1));

  if (response.statusCode == 200) {
    valorh = json.decode(response.body).toString();
    // Si la llamada al servidor fue exitosa, analiza el JSON

    return valorh;
  } else {
    // Si la llamada no fue exitosa, lanza un error.
    throw Exception('Failed to load post');
  }
}
}

```

Figura 3.9: Clase “Nodo.dart” Parte 1/3.

Elaborado por: Autor.

```

class menu extends StatefulWidget {
  @override
  State<menu> createState() => _menuState();
}
class _menuState extends State<menu> {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        shadowColor: Colors.white,
        backgroundColor: Colors.white10,
        foregroundColor: Colors.white,
      ),
      body: Container(
        decoration: BoxDecoration(
          color: Colors.black12,
        ),
        padding: EdgeInsets.only(left: 20, right: 20),

        child: Center(
          child: Column(
            children: [
              SizedBox(height: 50),
              Text("Datos", style: TextStyle(
                color: Colors.teal,
                fontSize: 55,
              )),
              SizedBox(height: 80),
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: [
                  Container(
                    height: 80.0,
                    width: 180.0,
                    alignment: Alignment.center,
                    decoration: BoxDecoration(
                      image: DecorationImage(
                        image: AssetImage("assets/temp.png"),
                        fit: BoxFit.fill
                      ),
                    ),
                  ),
                  Container(
                    height: 80.0,
                    width: 180.0,

```

Figura 3.10: Clase "Nodo.dart" Parte 2/3
Elaborado por: Autor.


```

import 'package:flutter/material.dart';

import 'menu.dart';
class modo extends StatefulWidget {

  @override
  State<modo> createState() => _modoState();
}

class _modoState extends State<modo> {
  double raiting = 50.0;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        shadowColor: Colors.white,
        backgroundColor: Colors.white10,
        foregroundColor: Colors.white,
      ),
      body: Container(
        decoration: BoxDecoration(
          color: Colors.black12,
        ),
        padding: EdgeInsets.only(left: 20, right: 20),
        child: Center(
          child: Column(
            children: [
              SizedBox(height:50),
              Text("DISPOSITIVOS", style: TextStyle(
                color: Colors.teal,
                fontSize: 40,
              )),
              SizedBox(height:35),
              Container(
                margin: EdgeInsets.only(top: 70),
                width: 200,
                decoration: BoxDecoration(
                  color: Colors.green,
                  borderRadius: BorderRadius.circular(10)
                ),
                child: FlatButton(
                  color: Colors.green,
                  child: Text("Nodo 1", style: TextStyle(color: Colors.white, fontSize: 20)),
                  onPressed: (){
                    Navigator.push(context, MaterialPageRoute(builder: (_)=>menu()));
                  },
                ),
              ),
            ],
          ),
        ),
      ),
    );
  }
}

```

Figura 3.12: Clase "Menu.dart" 1/2.
Elaborado por: Autor.


```

import 'package:flutter/material.dart';
import 'menu.dart';
import 'modo.dart';

void main() {

  runApp(const MyApp());
}
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: _inicio(),
    );}
class _inicio extends StatefulWidget {
  const _inicio({Key? key}) : super(key: key);

  @override
  State<_inicio> createState() => _inicioState();}

class _inicioState extends State<_inicio> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        color: Colors.teal,
        child: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              SizedBox(height: 470), Image.asset("assets/logoq.png"),SizedBox(height: 50),
              Row(mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: <Widget> [
                  new RaisedButton(
                    color: Colors.green ,
                    textColor: Colors.white,
                    child: Text("Empezar", style: TextStyle(
                      fontSize: 20.0,)),),
                  shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(10),
                    side: BorderSide(color: Colors.white30,)),
                    onPressed: () {Navigator.push(
                      context,MaterialPageRoute(builder: (_)=>modo(),);)},),
                ],
              ],
            ),
          ),
        ),
      ),
    );}

```

Figura 3.14: Clase "Main.dart".
Elaborado por: Autor.

Por último, en la figura 3.15 tenemos la configuración del archivo "Manifest", en este archivo se establecen los permisos de la aplicación móvil, en este caso se debe autorizar el uso de internet y conexiones al ejecutar la aplicación en nuestro móvil.

```

<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.tempapp"
    android:versionCode="1"
    android:versionName="1.0.0" >
    <uses-sdk
        android:minSdkVersion="16"
        android:targetSdkVersion="31" />
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:name="android.app.Application"
        android:appComponentFactory="androidx.core.app.CoreComponentFactory"
        android:debuggable="true"
        android:icon="@mipmap/ic_launcher"
        android:label="tempapp" >
        <activity
            android:name="com.example.tempapp.MainActivity"
            android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenSize|locale|layoutDirection|fontScale|screenLayout|density|uiMode"
            android:exported="true"
            android:hardwareAccelerated="true"
            android:launchMode="singleTop"
            android:theme="@style/LaunchTheme"
            android:windowSoftInputMode="adjustResize" >
            <meta-data
                android:name="io.flutter.embedding.android.NormalTheme"
                android:resource="@style/NormalTheme" />

            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <meta-data
            android:name="flutterEmbedding"
            android:value="2" />
        <uses-library
            android:name="androidx.window.extensions"
            android:required="false" />
        <uses-library
            android:name="androidx.window.sidecar"
            android:required="false" />
    </application>
</manifest>

```

Figura 3.15: Configuración "AndroidManifest.xml".
Elaborado por: Autor.

3.5. Resultados.

El sensor DHT22 se comunica y alimenta con el ESP32, deben conectarse el pin de datos (cable azul), el pin de alimentación (cable rojo) y el pin de tierra (cable negro), quedando la conexión como se muestra en la figura 3.16.

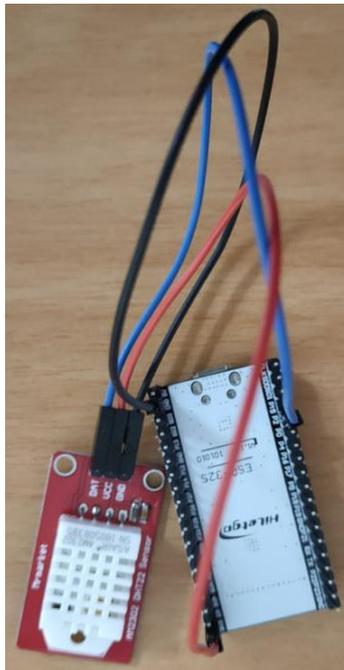


Figura 3.16: Conexión entre ESP32 y DHT22.
Elaborado por: Autor.

Posteriormente el sensor y ESP32, se ubicarán dentro del case que fue diseñado en una impresora 3D, quedando como figura 3.17. Una vez ubicado allí se conecta al cable de alimentación, figura 3.18, pudiendo ser alimentado con cualquier PC, dispositivo móvil o un cargador de teléfono.



Figura 3.17: ESP32 y DHT22 ubicados dentro del case.
Elaborado por: Autor.



Figura 3.18: Nodo final.
Elaborado por: Autor.

Una vez ensamblado el nodo sensor se lo ubica en el rack, de tal manera que se pueda obtener los valores de temperatura y humedad que experimenta el equipo, tal como se muestra en la figura 3.19.



Figura 3.19: Nodo sensor ubicado en el rack.
Elaborado por: Autor.

Para la visualización de los datos medidos se tiene un aplicativo móvil desarrollado en flutter, la pantalla de bienvenida se visualiza en la figura 3.20, el menú principal del aplicativo se muestra en la figura 3.21 y la información del nodo seleccionado se muestra en la figura 3.22.



Figura 3.20: Aplicativo móvil desarrollado en Flutter – Pantalla de bienvenida.
Elaborado por: Autor.



Figura 3.21: Aplicativo móvil desarrollado en Flutter –Menú principal.
Elaborado por: Autor.



Figura 3.22: Aplicativo móvil desarrollado en Flutter – Nodo sensor.
Elaborado por: Autor.

Otra opción que integra el proyecto es la facilidad de poder revisar la información del nodo sensor directamente de la página de www.ubidots.com, obteniendo un panel como el mostrado en la figura 3.23, pudiendo monitorear los valores desde cualquier dispositivo sin importar que no posea el aplicativo móvil.

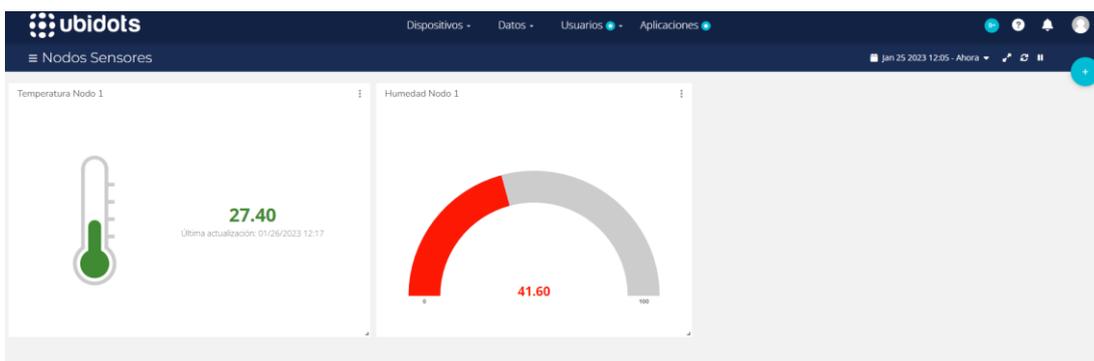


Figura 3.23: Visualización de valores medidos desde la plataforma en la nube.
Elaborado por: Autor.

CAPÍTULO 4:

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

1. Por medio de la descripción teórica y técnica de los fundamentos de sensores, IoT y redes inalámbricas, se pudo comprender la clasificación y tecnologías disponibles que pueden ser desarrolladas en la industria en busca del monitoreo y control de las variables de interés.
2. Se concluye con nuestro segundo y tercer objetivo, a través de la programación mediante los códigos y librerías se obtuvo los valores de temperatura y humedad experimentados por los dispositivos TI, para que posteriormente la tarjeta de desarrollo ESP32 los envíe a la API en la nube, Ubidots, usando el protocolo MQTT.
3. Con el uso de Android Studio y Flutter se pudo desarrollar el aplicativo móvil que integra las funciones necesarias para el monitoreo de las variables medidas mediante el nodo sensor.
4. Este trabajo de investigación servirá de guía a los alumnos de la facultad técnica para involucrarlos en el desarrollo de sistemas IoT para la recopilación y transmisión de datos a través de Internet y a su vez en el control de las distintas variables de interés. Además, se añade un presupuesto aproximado que se detalla en el Anexo 1.

4.2. Recomendaciones

- Se recomienda realizar una topología mesh entre los nodos sensores que se vayan a implementar en la medición, de tal manera que si alguno pierde conectividad lograría enviar la información que está midiendo hacia otro nodo que se comuniqué con la nube.
- Establecer una API privada de tal manera que los datos sean únicamente controlados por el personal de la empresa que emplee el sistema de monitoreo, evitando el comprometer la integridad de la información medida.
- Adicionalmente se recomienda implementar un sistema de respaldo de energía, para lo cual se puede usar baterías LiPo, para que en caso de una falla eléctrica los nodos sensores sigan midiendo y enviando los valores a la nube.
- Realizar capacitaciones en Android Studio para familiarizarse con las diferentes librerías y clases que se usa en el desarrollo de Apps para Android.

Referencias

- Aiello, M. (2022). IoT architectures: from data to smart systems. *Frontiers in the Internet of Things*, 1. <https://doi.org/10.3389/FRIOT.2022.959268>
- Asanza, V., Pico, R. E., Torres, D., Santillan, S., & Cadena, J. (2021). FPGA based meteorological monitoring station. *2021 IEEE Sensors Applications Symposium, SAS 2021 - Proceedings*. <https://doi.org/10.1109/SAS51076.2021.9530151>
- Bansal, M., & Priya. (2021). Performance comparison of mqtt and coap protocols in different simulation environments. *Lecture Notes in Networks and Systems*, 145, 549–560. https://doi.org/10.1007/978-981-15-7345-3_47
- DevAcademy. (2022). *CoAP protocol – Nordic Developer Academy*. <https://academy.nordicsemi.com/topic/lesson-5-coap-protocol/>
- Dizdarević, J., Carpio, F., Jukan, A., & Masip-Bruin, X. (2019). A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration. *ACM Computing Surveys*, 51(6). <https://doi.org/10.1145/3292674>
- Dorofeev, K., & Zoitl, A. (2018). Skill-based Engineering Approach using OPC UA Programs. *Proceedings - IEEE 16th International Conference on Industrial Informatics, INDIN 2018*, 1098–1103. <https://doi.org/10.1109/INDIN.2018.8471978>
- EtherPower. (2023). *Monitoreo DataCenter*. <https://www.etherpower.net/cms/index.php/monitoreo-datacenter/>
- Ghotbou, A., & Khansari, M. (2021). Comparing application layer protocols for video transmission in IoT low power lossy networks: an analytic comparison. *Wireless Networks*, 27(1), 269–283. <https://doi.org/10.1007/S11276-020-02453-6>

- Integrity. (2023). *Rango de Climatización de un Data Center – Integrity Perú*.
<https://integrity.pe/rango-de-climatizacion-de-un-data-center/>
- Khan, W. Z., Rehman, M. H., Zangoti, H. M., Afzal, M. K., Armi, N., & Salah, K. (2020). Industrial internet of things: Recent advances, enabling technologies and open challenges. *Computers and Electrical Engineering*, 81.
<https://doi.org/10.1016/J.COMPELECENG.2019.106522>
- Lohachab, A., & Jangra, A. (2019). Opportunistic Internet of Things (IoT): Demystifying the Effective Possibilities of Opportunistic Networks Towards IoT. *2019 6th International Conference on Signal Processing and Integrated Networks, SPIN 2019*, 1100–1105. <https://doi.org/10.1109/SPIN.2019.8711621>
- Mutlag, A. A., Abd Ghani, M. K., Arunkumar, N., Mohammed, M. A., & Mohd, O. (2019). Enabling technologies for fog computing in healthcare IoT systems. *Future Generation Computer Systems*, 90, 62–78.
<https://doi.org/10.1016/J.FUTURE.2018.07.049>
- OpenMind/BBVA. (2023). *Descubre las principales Tendencias IoT 2023 | OpenMind*.
<https://www.bbvaopenmind.com/tecnologia/mundo-digital/tendencias-iot-2023/>
- Singh, A. P., & Tomar, P. (2020). AI and IoT Capabilities: Standards, Procedures, Applications, and Protocols. *Artificial Intelligence to Solve Pervasive Internet of Things Issues*, 67–83. <https://doi.org/10.1016/B978-0-12-818576-6.00004-6>
- SWA. (2022). *MQTT Protocol Tutorial: Technical description with practical example*.
<https://www.survivingwithandroid.com/mqtt-protocol-tutorial/>
- TEMOK. (2022). *Datacenter or Data Center: Both Means Same Large Buildings for Servers | Temok Hosting Blog*. <https://www.temok.com/blog/datacenter-or-data-center/>

Wang, J., Lim, M. K., Wang, C., & Tseng, M. L. (2021). The evolution of the Internet of Things (IoT) over the past 20 years. *Computers and Industrial Engineering*, 155. <https://doi.org/10.1016/J.CIE.2021.107174>

Glosario General

API (Application Programming Interface): Conjunto de reglas y definiciones que permite que diferentes programas de software se comuniquen entre sí.

Android Studio: Entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android. Proporciona herramientas para desarrollar, testear y empaquetar aplicaciones.

Arduino: Plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar.

DHT22: Sensor digital que mide tanto la humedad relativa del aire como la temperatura.

ESP32: Módulo de microcontrolador con capacidades Wi-Fi y Bluetooth, ampliamente utilizado en proyectos IoT debido a su bajo costo y alta flexibilidad.

IDE (Entorno de Desarrollo Integrado): Software que proporciona herramientas esenciales para la programación, como editor de texto, compilador y depurador.

IoT (Internet de las Cosas): Concepto que se refiere a la interconexión de objetos cotidianos con la internet.

ISP (Proveedor de Servicios de Internet): Organización que ofrece servicios de acceso a Internet a clientes o suscriptores.

MQTT (Message Queuing Telemetry Transport): Protocolo de mensajería ligero y publicación/suscripción, optimizado para dispositivos con recursos limitados y redes con baja calidad.

Racks: Estructuras metálicas diseñadas para alojar equipos electrónicos, servidores, conmutadores y otros dispositivos relacionados con la informática y telecomunicaciones.

SDK (Kit de Desarrollo de Software): Conjunto de herramientas de desarrollo que permite a los programadores crear aplicaciones para un determinado software package, hardware platform, computer system o operating system.

Ubidots: Plataforma de IoT basada en la nube que permite a los desarrolladores capturar datos de sensores, visualizarlos y también transformarlos en información útil.

TT: Trabajo de titulación.

Apps: Aplicaciones

LiPo: (polímero de litio) tienen un gran rendimiento, son las baterías de última generación, son batería con una excelente relación entre capacidad, peso, volumen y tensión (voltaje) no obstante son más delicadas.

Anexos

Anexo 1. Presupuesto aproximado del TT

En este anexo se detalla un presupuesto aproximado para la elaboración del proyecto del TT.

| Ítem | Descripción | Costo Aproximado (USD) |
|--------------------------------|---|-----------------------------|
| Hardware | | |
| ESP32 | Módulo de microcontrolador con Wi-Fi y Bluetooth | \$8 - \$15 |
| Sensor DHT22 | Sensor de temperatura y humedad | \$8 - \$10 |
| Caja 3D | Contenedor plástico que contendrá el nodo sensor | \$15 - \$30 |
| Otros componentes | Resistencias, cables, protoboard, etc. | \$15 - \$30 |
| Software y Plataformas | | |
| Android Studio | IDE para desarrollo Android (Gratis) | \$0 |
| Arduino IDE | Entorno de programación para Arduino y ESP32 (Gratis) | \$0 |
| Servicios en línea | | |
| Ubidots (Plan Estándar) | Plataforma de IoT en la nube para visualización y análisis de datos | \$20/mes (varía según plan) |
| Total Aproximado | | \$66 - \$105 |

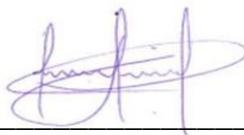
DECLARACIÓN Y AUTORIZACIÓN

Yo, **Solórzano Maldonado, Erick Andrés** con C.C: # 070584377-9 autor del Trabajo de Titulación: **Diseño de un sistema IoT para medición de variables ambientales de dispositivos TI alojados en racks de data centers mediante un módulo ESP32 y MQTT** previo a la obtención del título de **INGENIERO EN TELECOMUNICACIONES** en la Universidad Católica de Santiago de Guayaquil

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 28 de agosto del 2023



f. _____
Nombre: Solórzano Maldonado, Erick Andrés
C.C: 070584377-9



REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

| | | | |
|---|--|---|----|
| TÍTULO Y SUBTÍTULO: | Diseño de un sistema IoT para medición de variables ambientales de dispositivos TI alojados en racks de data centers mediante un módulo ESP32 y MQTT. | | |
| AUTOR(ES) | Solórzano Maldonado, Erick Andrés | | |
| REVISOR(ES)/TUTOR(ES) | Ing. Romero Rosero, Carlos Bolívar, M. Sc | | |
| INSTITUCIÓN: | Universidad Católica de Santiago de Guayaquil | | |
| FACULTAD: | Facultad de Educación Técnica para el Desarrollo | | |
| CARRERA: | Ingeniería en Telecomunicaciones | | |
| TITULO OBTENIDO: | Ingeniero en Telecomunicaciones | | |
| FECHA DE PUBLICACIÓN: | 28 de agosto del 2023 | No. DE PÁGINAS: | 59 |
| ÁREAS TEMÁTICAS: | Internet de las cosas, Protocolos de información, Tarjetas de desarrollo. | | |
| PALABRAS CLAVES | IoT, Redes de sensores, Sensores, MQTT, ESP32, Temperatura, Humedad | | |
| RESUMEN/ABSTRACT (150-250 palabras): | <p>La gestión de infraestructuras críticas, como las empleadas por los Proveedores de Servicios de Internet (ISP), enfrenta desafíos emergentes relacionados con la monitorización de condiciones ambientales en sus racks. Estas condiciones, especialmente las de temperatura y humedad comprometen la operatividad de dispositivos clave como enrutadores y conmutadores. Este trabajo aborda la necesidad de un sistema preciso y en tiempo real para supervisar estas variables. Nuestro principal objetivo es desarrollar un sistema IoT, aprovechando el potencial del módulo ESP32 y el sensor DHT22, para capturar, procesar y transmitir información sobre temperatura y humedad en racks de centros de datos. Mediante el uso del protocolo MQTT, aseguramos una transmisión de datos eficiente y confiable hacia una API en la nube, facilitando el acceso remoto y la toma de decisiones en tiempo real. Dentro de este estudio, se profundiza en los fundamentos teóricos de las redes de sensores e IoT, ilustrando su relevancia y aplicabilidad en contextos industriales y empresariales. A través de la programación y configuración del ESP32 junto con el sensor DHT22, demostramos la eficacia de nuestra solución propuesta, destacando su precisión y fiabilidad en la monitorización ambiental. Este trabajo presenta una solución innovadora y de bajo costo para abordar un problema operativo clave en los centros de datos, proponiendo un diseño IoT robusto que puede ser adaptado y escalado según las necesidades específicas de diferentes infraestructuras.</p> | | |
| ADJUNTO PDF: | <input checked="" type="checkbox"/> SI | <input type="checkbox"/> NO | |
| CONTACTO CON AUTOR/ES: | Teléfono: +593-98-291-0139 | E-mail: erick.solorzano@cu.ucsg.edu.ec | |
| CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE): | Nombre: Ricardo Xavier Ubilla González | | |
| | Teléfono: +593-99-952-8515 | | |
| | E-mail: ricardo.ubilla@cu.ucsg.edu.ec | | |
| SECCIÓN PARA USO DE BIBLIOTECA | | | |
| Nº. DE REGISTRO (en base a datos): | | | |
| Nº. DE CLASIFICACIÓN: | | | |
| DIRECCIÓN URL (tesis en la web): | | | |