

**UNIVERSIDAD INTERNACIONAL DE LAS AMÉRICAS  
FACULTAD DE INGENIERÍA ELECTROMECAÁNICA**

**TRABAJO FINAL DE GRADUACIÓN PARA OPTAR POR EL GRADO DE  
BACHILLERATO EN INGENIERÍA ELECTROMECAÁNICA**

**“Sistema de monitorización remoto de variables para cuartos de ambiente controlado para la fabricación de dispositivos médicos, mediante la plataforma Arduino”**

**David Gerardo Herrera Mora**

**Autor**

**Dennis Sánchez Fallas**

**Tutor**

**Sede Aranjuez**

**Septiembre, 2024**

## ***Dedicatoria***

*A Dios por permitirme llegar hasta este momento de mis estudios, por la sabiduría y salud que me permite culminar esta etapa.*

*A mi padre, por ser un pilar importante en la formación académica mía y de mis hermanos, ya que sabemos que realizó un gran esfuerzo para poder brindarnos una educación superior a los tres.*

*A mi madre, por ser ese motor de apoyo incondicional durante cada etapa, siempre brindando su amor y cariño para que nunca desistiéramos de nuestras metas.*

*A mis hermanos, por ser una fuente de inspiración para mí, siempre guiándome por el camino de la superación.*

*Les dedico este pequeño esfuerzo que no se compara con el que ustedes realizaron para yo pudiera culminar esta formación académica.*

## ***Agradecimientos***

*A Dios y a mi familia por acompañarme en todo momento de mi carrera.*

*A mis padres por siempre educarme para cumplir las metas que uno se propone.*

*A mis hermanos por siempre mostrarme su cariño, gracias, por los consejos para seguir siempre adelante de la mejor manera.*

*A mi tío Yanuario y Alex por apoyarme en este proyecto para culminarlo, siempre con palabras de aliento para no desistir en el camino.*

*A mi jefe Christian y la empresa actual donde trabajo por brindarme la oportunidad de dedicar tiempo al presente proyecto de graduación, además de facilitarme la posibilidad de consultar con expertos en la industria médica.*

*A mi profesor tutor, Dennis, a quien agradezco profundamente por su apoyo y conocimiento desde el inicio de este proyecto, además de ser una persona y profesional ejemplar.*

*A mis compañeros de trabajo y carrera con los que en muchas ocasiones compartí el día a día, los cuales siempre me mostraron su apoyo para el crecimiento profesional.*

*Finalmente, agradecer a todos los profesores de la universidad y a David, el director de carrera, que de alguna manera cada uno aportó su granito de arena para formarme como profesional.*

## CONTENIDO

Resumen.....	11
Planteamiento del problema .....	12
Antecedentes .....	18
Nacionales.....	18
Internacionales.....	22
Automatización de sistemas de monitoreo .....	29
Puntos de muestreo en un cuarto limpio.....	32
Establecimiento de puntos de muestreo .....	33
Consideraciones para la ubicación de puntos de muestreo.....	34
Requerimientos mínimos para un plan de monitoreo .....	35
Características de los sistemas de medida .....	38
Características estáticas.....	38
Exactitud.....	39
Fidelidad.....	39
Repetibilidad .....	39
Reproducibilidad .....	39
Sensibilidad .....	40
Linealidad .....	40
Características dinámicas .....	40
Tiempo de respuesta .....	40
Histéresis.....	41
Linealidad dinámica.....	41
Error dinámico.....	41
Tipos de sensores .....	41
Sensores de temperatura y humedad .....	41
Sensor DHT11 .....	42
Módulo AM2301 .....	43
SHT31-D.....	45
Si7021-A20 .....	45
Sensores de presión .....	46

BMP280.....	47
BME280.....	48
BMP390.....	49
LPS22HB .....	50
Sensores de medición de partículas .....	50
PMSA003I.....	51
Sensor Láser SDS011 .....	53
PMS9003M.....	53
HPMA115CO-004 .....	54
IPS-7100 .....	55
SPS30.....	56
Consumo de eléctrico .....	57
Plataforma Arduino .....	58
Historia.....	58
Características de la plataforma Arduino.....	58
Placas de Arduino .....	59
Arduino IDE .....	61
Microcontroladores compatibles con la plataforma Arduino .....	62
Internet de las cosas (IoT) .....	65
Introducción al internet de las cosas .....	65
Arquitectura de una red IoT .....	66
Componentes de una red IoT .....	68
Tecnologías de Acceso Inalámbrico .....	73
Estándar IEEE 802.11 .....	76
Funciones de las capas físicas (PHY) del estándar IEEE 802.11 .....	77
Capa de enlace de datos (DLL) y subcapa MAC del estándar IEEE 802.11 .....	77
Bases de datos .....	79
Historia de aplicaciones de base datos.....	79
Aplicación en sistemas de monitoreo.....	80
Sistema de administración de base de datos (DBMS).....	81
Modelo de datos relacionales .....	82

Lenguaje SQL .....	83
Base de datos web utilizando PHP.....	84
Aplicaciones en sistemas de monitoreo con la ayuda de bases de datos .....	84
Sistema de visualización .....	85
Enfoque de la investigación.....	87
Fuentes de información .....	87
Unidades de análisis .....	88
Análisis de componentes para el sistema de monitoreo .....	92
Sensores considerados para el sistema de monitoreo.....	92
Sensores de temperatura, humedad y presión.....	92
Análisis sensores de temperatura.....	94
Análisis sensores de humedad .....	95
Análisis sensores de presión.....	97
Análisis características eléctricas de sensores de temperatura, humedad y presión.....	99
Sensores de medición de concentración de partículas en el aire .....	102
Sensores de calidad de aire, medición en ( $\mu\text{g}/\text{m}^3$ ) .....	103
Sensores de calidad de aire, medición de conteo de partículas .....	105
Análisis características eléctricas/físicas de sensores medidores de partículas .....	107
Microcontroladores considerados para el sistema de monitoreo .....	108
Análisis características eléctricas del microcontrolador .....	112
Programación de los sensores utilizando Arduino IDE.....	114
Conexión física de los sensores al microcontrolador .....	114
Respaldo de instalación eléctrica de los circuitos .....	117
Programación de los sensores .....	118
Programación de los sensores SHT31-D y BMP390 .....	118
Programación del sensor IPS-7100 .....	121
Programación de periodicidad de registros de sensores .....	124
Topología de conectividad a la red para el sistema de monitoreo .....	125
Tecnología de acceso a la red .....	125
Aspectos por tomar en cuenta en caso de que el Router falle .....	128
Topología de conectividad.....	131

Protocolo de comunicación .....	133
Sistema de almacenamiento .....	135
Visualización de los datos .....	138
Análisis financiero .....	139
Materiales requeridos para cada nodo o punto de muestreo .....	139
Costos de programación e instalación del sistema de monitoreo .....	141
Costos por consumo eléctrico del sistema de monitoreo .....	142
Costo actual del sistema de monitoreo .....	142
Costo de mantenimiento del sistema de monitoreo .....	143
Cálculo del VAN y TIR.....	144
Conclusiones .....	146
Recomendaciones .....	147
Bibliografía .....	150
Anexo 1 .....	155
Anexo 2 .....	162
Anexo 3 .....	164
Anexo 4 .....	165

## ÍNDICE DE FIGURAS

Figura 1: Costa Rica: Exportaciones de dispositivos médicos por nivel de sofisticación, 2007-2020	12
Figura 2: Conexiones básicas antecedente internacional n°6.....	27
Figura 3: Modelo estructural de un sistema automatizado.....	30
Figura 4: Sensor de Temperatura y Humedad DT11 .....	43
Figura 5. Módulo de Temperatura y Humedad AM2301 .....	43
Figura 6. Error de humedad relativa a 25 °C .....	44
Figura 7. Sensor de temperatura y Humedad SHT31-D. ....	45
Figura 8. Sensor de temperatura y Humedad Si7021-A20.....	46
Figura 9. Clasificación de los sensores de presión.....	47
Figura 10. Sensor BMP280 .....	48
Figura 11. Sensor BME280 .....	49
Figura 12. Sensor BMP390 .....	49
Figura 13. Sensor LPS22HB .....	50
Figura 14. Sensor Adafruit PMSA003L. ....	52
Figura 15. Sensor Láser SDS011.....	53
Figura 16. Sensor PMS9003M. ....	54
Figura 17. Sensor HPM115CO-004.....	55
Figura 18. Sensor IPS-7100.....	56
Figura 19. Sensor SPS30.....	57
Figura 20. Placa de Arduino Uno. ....	60
Figura 21. Diagrama Arduino UNO .....	61
Figura 22. Arduino IDE 2.2.1 .....	62
Figura 23. Placa ESP32-WROOM-32.....	63
Figura 24. IoT modelo de referencia publicado por IoT World Forum .....	66
Figura 25. Modelo simplificado IoT.....	68
Figura 26. Topología estrella. ....	71
Figura 27. Topología en malla. ....	71
Figura 28. Rango de tecnologías de acceso inalámbrico. ....	75
Figura 29. Modelo OSI para IEEE 802.11 .....	77
Figura 30. Formato de la trama MAC .....	78
Figura 31. Ejemplo de Dashboard en Grafana .....	86

Figura 32. Placa ESP32-WROOM-32.....	112
Figura 33. Diseño de Pines ESP32-DevKit C.....	115
Figura 34. Conexión física del microcontrolador y sensores. ....	117
Figura 35. Librerías de sensores BMP390 y SHT31-D. ....	119
Figura 36. Conexión inalámbrica del microcontrolador.....	128
Figura 37. Topología de red del sistema de monitoreo .....	132
Figura 38. Diagrama de flujo de conectividad HTTP cliente-servidor local.....	135
Figura 39. Esquema de base de datos “Sistema_monitoreo”.....	136
Figura 40. Diseño de base de datos. ....	136
Figura 41. Distribución de base de datos.....	138
Figura 42. Dashboard de Promedios para nodo 1. ....	139
Figura 43. Tarifas de consumo eléctrico, ICE.....	142
Figura 44. Propuesta de sistema de monitoreo. ....	149

## ÍNDICE DE TABLAS

Tabla 1. ISO clases de calidad del aire de acuerdo con la concentración de partículas.....	31
Tabla 2. Puntos de muestreo relacionados con el área del cuarto limpio. ....	34
Tabla 3. Características sensor DHT11.....	43
Tabla 4. Características de humedad, módulo AM2301.....	44
Tabla 5. Características de temperatura, módulo AM2301. ....	44
Tabla 6. Características de eléctricas, módulo AM230. ....	45
Tabla 7. Ejemplo de base de datos relacionales. ....	83
Tabla 8. Matriz de conceptualización, investigación de enfoque cuantitativo. ....	89
Tabla 9. Sensores de temperatura, humedad y presión.....	93
Tabla 10. Comparación de características para la variable de temperatura. ....	94
Tabla 11. Comparación de características para la variable de humedad.....	96
Tabla 12. Comparación de características para la variable de presión. ....	97
Tabla 13. Resumen de características eléctricas de los sensores de temperatura, humedad y presión.....	100
Tabla 14. Cálculo consumo eléctrico sensores de temperatura, humedad y presión.....	101
Tabla 15. Sensores capaces de medir la concentración de partículas en el aire.....	102
Tabla 16. Sensores de concentración de partículas en el aire, medición en ( $\mu\text{g}/\text{m}^3$ ). ....	103
Tabla 17. Sensores de conteo de partículas en el aire. ....	105
Tabla 18. Características eléctricas/físicas del sensor IPS-7100.....	107
Tabla 19. Microcontroladores considerados para el sistema de monitoreo. ....	109
Tabla 20. Características de los microcontroladores.....	111
Tabla 21. Características eléctricas ESP32-DevKit C.....	112
Tabla 22. Potencia Total(W) por punto de muestreo.....	113
Tabla 23. Costo de componentes requeridos para el diseño del sistema de monitoreo.....	140
Tabla 24. Computadora utilizada como servidor local.....	140
Tabla 25. Costos de programación e instalación del sistema de monitoreo. ....	141
Tabla 26. Costos actuales totales del sistema de monitoreo.....	143
Tabla 27. Costos de mantenimiento bimensual del sistema de monitoreo. ....	144

## Resumen

En la actualidad la adquisición de datos ha tomado gran relevancia en el ámbito industrial, permitiendo un mayor control de los procesos que se pueden llevar a cabo en una organización. El acceso a los datos obtenidos de alguna determinada actividad o proceso permite tener un mejor entendimiento de este y que podrá ayudar al usuario en la toma de decisiones basadas en la información obtenida.

Con el avance de las tecnologías, cada vez es más fácil tener acceso a distintos tipos de sistemas de adquisición de datos que permiten conocer los valores obtenidos en tiempo real por medio de una interfaz. Con el gran auge de los sistemas electrónicos, y la mejora de los elementos de medición, es posible encontrar en el mercado actual una gran variedad de microprocesadores a un precio bastante accesible. Esto ha hecho que surgieran distintos sistemas de monitorización con capacidad de implementarse en múltiples sectores, con una infinidad de usos.

Como parte de este proyecto de investigación se pretende proponer un sistema de monitoreo remoto de variables para cuartos de ambiente controlado, utilizando la plataforma Arduino. Este sistema electrónico, ayudará a la adquisición de datos de temperatura, humedad, presión y cantidad de partículas en el aire, específicamente para cuartos de ambiente controlado.

El sistema de monitoreo propuesto en el presente trabajo ofrecerá un sistema capaz de recopilar en una base de datos, todas las mediciones obtenidas por los sensores que conforma el circuito. Esto para que luego puedan ser analizadas en una interfaz gráfica que permita al usuario visualizar los datos registrados.

## CAPÍTULO I: INTRODUCCIÓN

### Planteamiento del problema

Actualmente, una de las industrias que más se ha beneficiado con el avance en las nuevas tecnologías es el sector médico, como principal área de operación la fabricación de dispositivos médicos, ofreciendo cada día a la sociedad en general una serie de dispositivos con el fin de mejorar el bienestar de las personas.

A nivel nacional, Costa Rica ha visto un incremento en el sector de la fabricación de dispositivos médicos, convirtiéndose en el principal producto de exportación del país, según el estudio realizado en diciembre del 2022 por el Banco Interamericano de Desarrollo, “El sector/clúster de dispositivos médicos de Costa Rica”.

En la figura 1, es posible apreciar el incremento de exportaciones a nivel nacional de dispositivos médicos entre el 2007-2020.

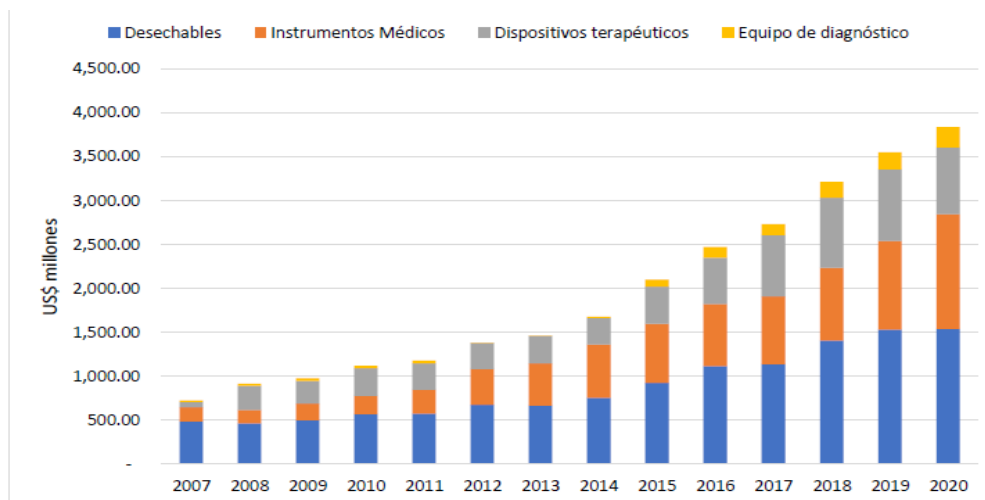


Figura 1: Costa Rica: Exportaciones de dispositivos médicos por nivel de sofisticación, 2007-2020

Fuente: BID. José Salazar Xirinachs, (2022)

Producto del crecimiento de la industria, es necesario cada vez más la habilitación de infraestructura, que pueda albergar los procesos de manufactura de equipos médicos. Puesto que la industria médica obedece al cumplimiento de normativas internacionales emitidas

por autoridades reguladoras, los procesos de manufactura de dispositivos médicos deben cumplir con altos estándares de calidad, con la finalidad de garantizar la seguridad de los pacientes.

Debido a los estándares de calidad que deben seguirse para la fabricación de los dispositivos médicos es importante mantener un ambiente controlado en las instalaciones, donde se garantice la protección de procesos o productos sensibles a las partículas u otros tipos de contaminantes. Una de las medidas fundamentales que aplica la industria de fabricación de equipo médico es la implementación de cuartos de ambientes controlados.

Cuando se habla de cuartos de ambientes controlados, un claro ejemplo, pueden ser los cuartos limpios. La Norma ISO 1464-1: 2015, define a un cuarto limpio como aquel donde el número de concentración de partículas transportadas en el aire es controlado y clasificado, además que es diseñado, construido y operado de manera que se controle la introducción, generación y retención de partículas dentro del cuarto. La clasificación de los cuartos limpios se determina de acuerdo con la calidad del aire, tomándose como referencia la cantidad de partículas en un volumen de aire.

La clasificación ISO de los cuartos limpios, se realiza de 1 a 8, en donde la clasificación 1 corresponde al más estricto en cuanto a los límites permitidos de concentración de partículas.

En los cuartos limpios es importante controlar las condiciones ambientales como la temperatura, presión, concentración de partículas en el aire, humedad relativa, entre otras. El monitoreo de las condiciones ambientales permitirá tener un control sobre los parámetros que deben cumplirse en un cuarto limpio, de manera que las condiciones ambientales no se vuelvan un factor que perjudique al proceso de producción y su calidad en el producto.

En muchas ocasiones, los cuartos limpios cuentan con un sistema de sensores que permite visualizar en un *display* (pantalla digital) la temperatura y humedad relativa del cuarto, donde una persona debe ser la encargada de anotar y registrar esos parámetros en

el momento exacto que se tomó la medición, lo que provoca que se deba designar un recurso que sea encargado de monitorear estos parámetros de manera presencial en el cuarto, condicionando la obtención de los datos a la disponibilidad de la persona encargada.

Para la obtención y análisis de los datos, se vuelve poco provechoso que solo se logren tomar al menos 2 mediciones al día de cada variable monitoreada en el cuarto, ya que estos datos se pueden ver influenciados por situaciones específicas que se pudieron haber presentado durante la medición. Por ejemplo, puede suceder que la persona encargada de tomar la medición de temperatura siempre lo haga al principio del turno laboral, cuando no hay ninguna o no están todas las máquinas operando, o bien que cuando tome la medición de la presión diferencial del cuarto, varias personas estén en alguna sección del cuarto abriendo una puerta, causando fluctuaciones en la presión.

El hecho que el registro de todas estas variables no se encuentre automatizado, abre la posibilidad para que los datos proporcionen información errónea y creen un panorama poco realista del comportamiento de las variables del cuarto durante los procesos que se llevan a cabo en este, provocando poca seguridad en los datos obtenidos y siendo pocos útiles para la toma de decisiones en el sistema de calidad.

Por otro lado, para el control de la cantidad de partículas viables en el ambiente, es común que se realice la medición posicionando en puntos estratégicos el sensor, según lo establece la norma ISO 1464-1: 2015. La poca frecuencia del monitoreo de partículas puede provocar que no se logren identificar los momentos del proceso productivo, en donde la generación de partículas es mayor o incluso pueda estar en riesgo de encontrarse fuera de los criterios de aceptación, afectando la calidad del producto final.

Como parte de este proyecto, se pretende dar una propuesta de diseño de un sistema de monitoreo que permita automatizar y mantener un control constante de las variables de temperatura, humedad relativa, presión y monitoreo de la cantidad de partículas en el aire, de modo que se pueda llevar un registro de las mediciones, con acceso a estas en cualquier momento del día, proporcionando un sistema escalable que permita la incorporación de

sensores para los puntos estratégicos del cuarto limpio, permitiendo conocer mejor el comportamiento del cuarto durante el proceso de manufactura.

Es por esto que este proyecto plantea: ¿De qué manera se puede implementar un sistema que monitoree la temperatura, humedad relativa, presión y cantidad de partículas en el aire, de forma remota para cuartos de ambiente controlado para la fabricación de equipo médico utilizando Arduino?

## **Objetivo general**

Diseñar una propuesta de un sistema electrónico que permita monitorear de manera remota la temperatura, humedad relativa, presión y cantidad de partículas en el aire mediante la adquisición de datos provenientes de cuartos de ambiente controlado utilizados para la fabricación de dispositivos médicos.

## **Objetivos específicos**

- Proponer una red de monitoreo por sensores remotos de temperatura, humedad relativa, presión y cantidad de partículas en el aire, utilizando la plataforma Arduino.
- Proporcionar un análisis técnico para la selección de los sensores a utilizar en el sistema de monitoreo.
- Determinar los requerimientos para una red de internet de las cosas (IoT, Internet of Things) que satisfaga las necesidades tecnológicas del sistema de monitoreo.
- Elaborar una propuesta de un sistema de almacenamiento y visualización de los datos, transmitidos por una red de sensores de temperatura, humedad relativa, presión y cantidad de partículas en el aire.
- Elaborar el análisis financiero para futuras instalaciones del sistema de monitoreo en cuartos de ambiente controlado, tomando en cuenta el costo económico de su implementación.

## **Justificación**

Este proyecto pretende desarrollarse específicamente hacia la industria de fabricación de equipos médicos. La industria médica contribuye al mejoramiento de la calidad y esperanza de vida de las personas, innovando cada día con nuevos dispositivos capaces de tratar y diagnosticar enfermedades. Sin embargo, para la fabricación de estos dispositivos si no se realizan siguiendo todos los estándares de calidad estipulados por las normativas nacionales e internacionales, pueden llegar a representar un riesgo importante para la salud de las personas que hacen uso de estos.

Es por esto que no solo los dispositivos médicos deben estar en una mejora continua, sino que también todos los sistemas de control y monitoreo, que influyan de manera directa o indirecta, con la calidad de los productos.

Para la fabricación de los dispositivos médicos, los cuartos limpios representan una herramienta muy importante para mantener controladas las condiciones ambientales, sobre las cuales se llevan a cabo las actividades productivas. La justificación de este proyecto radica en ofrecer un sistema de monitoreo automatizado que permita la implementación de diferentes componentes electrónicos y paquetes informáticos, que faciliten la obtención de mediciones como la temperatura, humedad relativa, presión diferencial y cantidad de partículas en el aire en puntos estratégicos del cuarto limpio, de una manera continua y en tiempo real, presentándose como una solución económica.

El sistema de monitoreo propuesto dará solución a que se lleve un registro continuo de las condiciones ambientales del cuarto limpio, permitiendo así, que la obtención de estos datos no dependa de la disponibilidad de alguna persona. Además, permitirá generar mejor trazabilidad de los datos, y seguridad de la veracidad de estos.

La problemática de que los datos obtenidos a ciertas horas del día, no representen las condiciones reales del cuarto limpio, se va a reducir dado que la monitorización continua, permitirá detectar tendencias de datos a través del tiempo, de manera que se pueda procesar dicha información, y se identifiquen todas aquellas actividades del proceso productivo que puedan influir a que los parámetros del cuarto limpio no cumplan los

criterios de aceptación, convirtiéndose una posible causa de inconformidades en el producto final.

## **Antecedentes**

### **Nacionales**

Antecedente n°1

Institución: Instituto Tecnológico de Costa Rica.

Título: Sistema electrónico para el control de riego y adquisición de datos en la monitorización de temperatura, humedad y luminosidad de un invernadero.

Autor: Claudio Carvajal González

Año: 2015

En este estudio Carvajal (2015) tiene como objetivo realizar un sistema que permita programar el riego por mesas y secciones de un invernadero, además que genere un historial de las condiciones ambientales del lugar dadas por las variables temperatura, humedad y luminosidad.

El autor toma en cuenta las variables que pueden favorecer la creación de su sistema uno que sea eficiente para la adquisición y almacenamiento de los datos recibidos por los sensores, dado que analiza cada aspecto técnico de los equipos e implementos a utilizar en su circuito y como se pueden interconectar entre sí. En la sección 4.3 es posible observar la evaluación de alternativas para el diseño del sistema electrónico de acuerdo con la distribución de los sensores en el invernadero.

Para el desarrollo de la base de datos en el punto 5.5.3, utiliza el *software* llamado MySQL Workbench, el cual le permite el modelado de datos, desarrollo de código SQL, administración de los usuarios, configurar y administrar las bases de datos con el servidor web. Una vez desarrollada la base de datos Carvajal (2015) logra desarrollar una página web para la visualización de los datos obtenidos por los sensores.

En este proyecto finalmente el autor logra diseñar el sistema electrónico de monitoreo y control de riego para las mesas del invernadero.

La información y metodología que se encuentra en esta tesis se utilizará como referencia para la propuesta de un sistema de monitoreo remoto de variables para cuarto de ambiente controlado mediante la plataforma Arduino.

Antecedente n°2

Institución: Universidad de Costa Rica.

Título: Implementación de microcontroladores electrónicos de bajo costo en la adquisición de deformaciones unitarias en estructuras.

Autor: Josué Berrocal Flores

Año: 2021

Como parte de este trabajo de graduación Berrocal (2021) propone la implementación de microcontroladores electrónicos de bajo costo para medir las deformaciones unitarias en estructuras. En la sección 2.3 define las características y composición del dispositivo Arduino, así como la comparación entre cada uno de los distintos modelos. Además, en la sección 3.1 desarrolla un esquema de un sistema de adquisición de datos por sus siglas en inglés (DAQ), que se compone de, sensores, un módulo de detección, un dispositivo DAQ, *software* y un ordenador.

La manera como el autor logró desarrollar el DAQ, se utilizará de referencia, como un método que podría implementarse en el sistema electrónico del presente trabajo de graduación.

Antecedente n°3

Institución: Universidad de Costa Rica

Título: Implementación de un sistema de monitoreo de partículas no viables en un ambiente controlado de Boston Scientific.

Autor: Marcos Waterhouse Morales

Año: 2021

En este trabajo final de graduación, Waterhouse (2021) tiene como objetivo general implementar un sistema automatizado de monitoreo de partículas no viables en un cuarto de

ambiente controlado de Boston Scientific. Esta investigación hace referencia a los conceptos básicos necesarios para entender la definición de un cuarto de ambiente controlado CER, por sus siglas en inglés.

En la sección 2.2.1 hace referencia a la norma ISO 14644-1, la cual determina la clasificación de los cuartos según la calidad del aire, utilizando para esto, la cantidad de partículas en un volumen de aire. Así como la selección de puntos de muestreo y como determinar la clasificación de manera periódica.

La información que se estudia en la tesis de Waterhouse con respecto a la monitorización de partículas en el aire se utilizará como referencia para el presente trabajo de investigación en la implementación del sistema de monitoreo remoto de variables para cuartos de ambiente controlado, mediante la plataforma Arduino.

Antecedente n°4.

Institución: Instituto Tecnológico de Costa Rica

Título: Diseño de un sistema electrónico inalámbrico de adquisición de datos para la toma de decisiones en un sistema hidropónico de lechuga en un invernadero del TEC sede San Carlos.

Autor: Anthony Amador Álvarez

Año: 2017

En esta investigación el autor, se propone como objetivo implementar la transmisión y adquisición inalámbrica de datos para la definición de un umbral en el control de riego y la visualización gráfica de la información mediante la utilización de transmisores inalámbricos. Como parte de la metodología implementada, desarrolla cuatro encapsulados, para dividir por secciones su circuito, para luego enlazarlos entre sí. El autor llegó a esta solución, debido a que encontró inconvenientes en la estructura que pensaba utilizar con los Arduinos. Cito textualmente al autor que expone que “se tenía pensado utilizar un Arduino Mega, Nano o UNO en cada módulo, pero durante el desarrollo del proyecto empezaron a surgir problemas en cuanto a la decodificación de los datos antes de subirlos a internet, ya que utilizando un Arduino para hacer

esta tarea implicaba un algoritmo complejo para decodificar los datos y además contar con un módulo de ethernet para subirlos” (pp. 27-28, 2017)

La solución implementada en la metodología de esta tesis Amador (2017) se utilizará de referencia para la implementación del sistema de monitoreo remoto de variables para cuartos de ambiente controlado, mediante la plataforma Arduino.

#### Antecedente n°5

Institución: Universidad Técnica Nacional, Costa Rica.

Título: Aprovechamiento de tecnologías basadas en internet de las cosas y su relación con el agotamiento de protocolos IPV4 en la zona de San Carlos hasta el año 2019

Autor: Freddy Gerardo Rocha Boza

Año: 2019

En esta investigación Rocha tiene como objetivo específico caracterizar la estructura de un ecosistema IoT referente a la infraestructura de *hardware* y *software* en el contexto de las empresas. Durante su investigación demuestra lo importante que es para las empresas tener herramientas de *software* con características que permitan la centralización de la información. Al mismo tiempo llega a la siguiente conclusión, ya que “al caracterizar la estructura de un ecosistema IoT referente a la infraestructura de *hardware* y *software* en el contexto de las empresas, los elementos claves en este ámbito son la conectividad y la nube” (p. 90, 2019).

Parte del estudio realizado por Rocha de los requerimientos para una estructura IoT, serán tomados como referencia para la implementación del sistema de monitoreo remoto de variables para cuartos de ambiente controlado, mediante la plataforma Arduino.

#### Antecedente n°6.

Institución: Instituto Tecnológico de Costa Rica.

Título: Diseño de sistema de medición y monitorización de la cantidad de material de un basurero de reciclaje.

Autor: Steven Josué Quiel Hidalgo

Año: 2021

En esta investigación el autor diseña una arquitectura IoT para el envío, almacenamiento y visualización de los datos de la cantidad de material en el tiempo.

La metodología utilizada en esta tesis para la creación de la base de datos e interfaz de visualización será utilizada como referencia para el manejo de los datos obtenidos en el sistema de monitoreo de variables para cuartos de ambiente controlado, utilizando la plataforma Arduino.

### **Internacionales**

Antecedente n° 1

Institución: Tecnológico de Monterrey, México.

Título: Sistema de administración de energía utilizando Internet de las Cosas (IoT)

Autor: David Lizárraga Osuna

Año: 2018

Como parte de esta investigación Lizárraga (2018) tiene como objetivo realizar un sistema de administración de energía con *hardware* y *software* de código abierto completamente configurable que sea capaz de monitorear y registrar temperatura y humedad en múltiples zonas de los edificios de una empresa.

La arquitectura del sistema propuesto por el autor implementa el uso de *hardware* y *software* de código abierto. Donde el dispositivo central es un Raspberry Pi, el *gateway* son dos Arduinos Mega 2560 comunicados por Modbus RTU, y los módulos remotos son NodemCU, los cuales tienen un microcontrolador que es capaz de conectarse a la red vía Wifi. Estos últimos, cuentan con sensores y relevadores, con el fin de estar registrando la temperatura y la humedad de cada espacio refrigerado de las sucursales.

En esta investigación el autor se encarga de explicar la topología de la red que propone, así como los algoritmos de programación para cada uno de los componentes que conforman el sistema. Finalmente logra basar la arquitectura del sistema mediante el uso de tres componentes principales: un dispositivo central; módulos remotos y *gateways*.

La información y metodología que se encuentra en esta tesis se utilizará como referencia para elaborar la estructura del sistema de monitoreo remoto de variables para cuarto de ambiente controlado mediante la plataforma Arduino.

Antecedente n°2

Institución: Universidad Nacional De San Agustín De Arequipa, Perú.

Título: Modelo de implementación para el monitoreo y control de condiciones ambientales basados con tecnologías de Arduino y Raspberry.

Autor: Irani Guillen Gomez

Año: 2021

En esta investigación Irani Guillen (2021) realiza un modelo de implementación de *software* que integra recursos electrónicos libres y de fácil adquisición como Raspberry, Arduino y sensores de detección ambiental que hacen uso de las Tecnologías de la Información y Comunicación (TIC) para realizar los procedimientos de transmisión de información entre los diferentes dispositivos y sistemas correspondientes. Además, el desarrollo del proyecto se realizó con base en la aplicación de la tecnología del Internet de las Cosas (IoT), tecnología que resalta por integrar dispositivos tecnológicos inteligentes para detectar y recoger datos para su gestión.

El autor realiza un análisis de las posibles plataformas digitales que se pueden utilizar para el procesamiento y gestión de datos recolectados por los sensores, así como los requerimientos para la implementación de una red IoT.

En esta investigación, el autor logra:

- El diseño de la plataforma de IoT que permite integrar el *software* con el *hardware* que permite ofrecer la información climatológica de temperatura, humedad, calidad del aire y radiación ultravioleta a partir de los sensores inteligentes de detección, placa Arduino uno y el ordenador de placa reducida Raspberry.
- Conseguir contar con un historial climatológico, el mismo que es almacenado en un repositorio digital de base de datos en MySQL para la generación de un sistema web interactivo que permita la visualización dinámica de los datos. (2021, p.125).

La información y metodología que se encuentra en esta tesis se utilizará como referencia para implementar una red IoT que satisfaga los requerimientos del sistema de monitoreo remoto de variables para cuarto de ambiente controlado mediante la plataforma Arduino.

Antecedente n°3

Institución: Universidad Politécnica de Madrid, España.

Título: Sistema de monitorización de variables ambientales de bajo coste basado en Arduino.

Autor: Alejandro Payan de Tejada Alonso

Año: 2021

El autor de esta tesis de doctorado se propone como objetivo principal diseñar, desarrollar, implementar y poner a prueba un sistema de monitorización de bajo coste, basado en la plataforma Arduino. En esta investigación se ofrece una detallada información respecto a la historia, funcionalidad y aplicaciones de la plataforma Arduino a distintos sistemas electrónicos.

En la sección 4.2 detalla a cada uno de los sensores que se pueden utilizar para la adquisición de datos, así como la explicación técnica para la selección de cada sensor. También en la sección 4.4.2.1 explica de una manera clara el funcionamiento del IDE Arduino, al servir como editor de código, compilador, depurador y constructor de interfaz gráfica, posibilitando el proceso completo de tareas de programación del controlador.

El autor de esta tesis también realiza un estudio para la selección de la placa de Arduino, la cual de acuerdo con las necesidades técnicas del sistema tendrá que satisfacer los requisitos de:

- Capacidad de conexión física de señal: la placa debe contar con suficientes pines preparados para la conexión de los sensores.
- Capacidad de comunicación
- Suficiente espacio en la memoria *flash* y SRAM.

Finalmente, el autor realiza un análisis del coste económico del sistema, cuantificando económicamente los distintos elementos que componen el sistema, utilizando un precio medio de las tiendas de referencia consultadas.

Para la presente investigación, se utilizará la información que se encuentra en esta tesis como base para el estudio de la plataforma Arduino, que será utilizada en el sistema de monitoreo remoto de variables para cuarto de ambiente controlado.

Antecedente n°4

Institución: Universidad UNAPEC, República Dominicana.

Título: Diseño de un sistema Automático de supervisión de parámetros de un cuarto limpio.

Autor: Emmanuel Morel Linares

Año: 2017

En este proyecto de investigación el autor toma un enfoque de la monitorización de variables para cuartos limpios, específicamente cuartos limpios utilizados en la industria médica. El autor pretende diseñar un sistema con el fin de realizar una supervisión continua y en tiempo real de los parámetros del cuarto limpio y que a la vez tenga la capacidad de almacenar en un registro todas las actividades ejecutadas dentro del recinto controlado.

En este trabajo de investigación, se mencionan diferentes tipos de cuartos limpios, de los cuales cada uno tiene distintas variables que se deben monitorear y controlar de manera constante. Por esto se expone un sistema de monitoreo utilizando la plataforma Arduino. Además, el sistema propuesto implementa el uso de un sensor de partículas capaz de detectar partículas de polvo y determinar el nivel de particulado del aire.

Para nuestra investigación, se utilizará la información y metodología que se encuentra en esta tesis como referencia para la implementación de un sistema de monitoreo remoto de variables mediante Arduino, en un cuarto de ambiente controlado, como lo son los cuartos limpios.

Antecedente n°5

Institución: Universidad Tecnológica de Bolívar, Colombia

Título: Control seguro de una red de sensores/actuadores domésticos.

Autor: Maryori Del Carmen Sabalza Mejia

Año: 2016

En esta investigación la autora propone un sistema de *hardware-software* que permita controlar dispositivos electrónicos desde una aplicación web, implementado una red de sensores-actuadores que permita su monitoreo y control remoto.

Como parte de este trabajo de investigación se tomarán en cuenta los protocolos de comunicación y las plataformas de *hardware* abierto, utilizados para conectar los dispositivos a internet de forma segura.

La información y metodología que se encuentra en esta tesis se utilizará como referencia para la propuesta del sistema de monitoreo remoto de variables para cuarto de ambiente controlado mediante la plataforma Arduino.

Antecedente n°6

Institución: Universitat Politècnica de Valencia, España.

Título: Diseño de un sistema de captura y procesamiento de señales.

Autor: Sergio Villanueva Martinez

Año: 2014

En esta investigación Villanueva (2014) propone un muestreo de sensores mediante un microcontrolador Arduino. En la siguiente figura el autor representa con un esquema el alcance de su proyecto, donde las líneas discontinuas representan conexiones inalámbricas entre dispositivos.

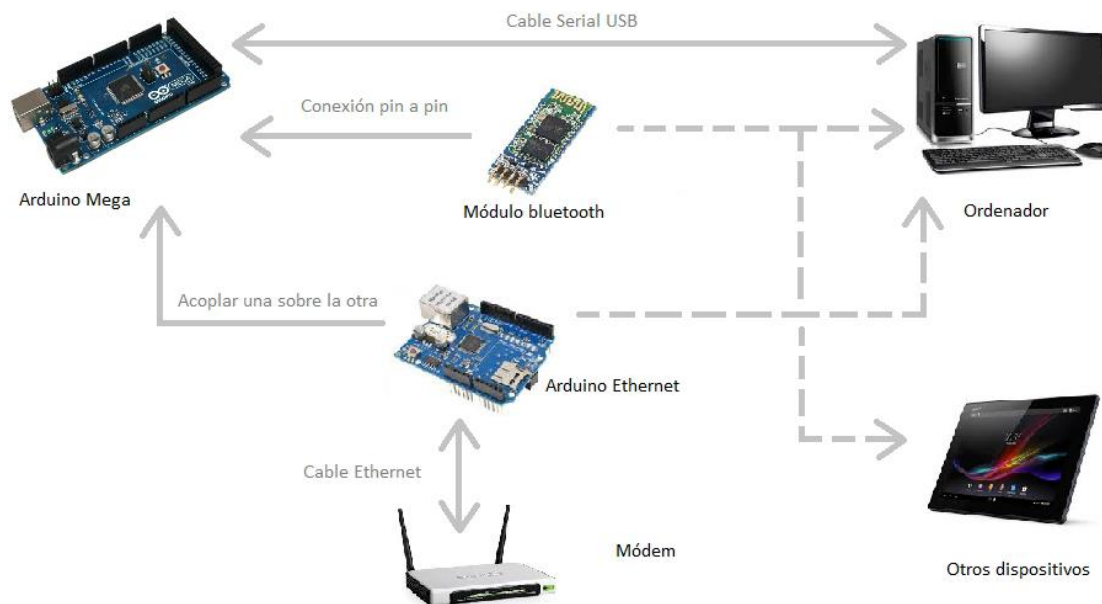


Figura 2: Conexiones básicas antecedente internacional n°6.

Fuente: Villanueva, 2014.

En esta investigación el autor desarrolla de manera clara cómo es posible realizar una programación en el *software* Arduino IDE, que está basado en el lenguaje de programación C. Explica cada una de las secciones del *software* Arduino IDE y que instrucciones se pueden ejecutar en cada sección.

En la sección 3.3, el autor utiliza un sensor digital de temperatura / humedad, donde explica su funcionamiento teórico y realiza un análisis de la señal que es enviada desde el sensor hasta el microcontrolador. En este mismo apartado propone 2 soluciones para capturar la información del sensor.

El esquema utilizado en esta investigación, tanto como la metodología empleada para la captura de datos y posterior procesamiento de los datos capturados, se utilizará como referencia para el sistema de monitoreo de variables para cuartos de ambiente controlado, mediante la plataforma Arduino.

#### Limitaciones

- Existencia de sensores disponibles en el mercado capaces de medir la cantidad de partículas dispersas en el aire en las unidades de cantidad de partículas/m<sup>3</sup> y que además

sean compatibles con los protocolos de comunicación I2C, UART o SPI y la plataforma de Arduino IDE.

- Para este trabajo final de graduación no será parte del alcance el diseño mecánico requerido para la instalación del sistema de monitoreo.

## CAPÍTULO II: MARCO TEÓRICO

El propósito de este capítulo es exponer los conceptos que abarcan el área de la automatización mediante la adquisición de datos, específicamente para el sistema de monitoreo de variables en cuartos de ambiente controlado utilizados para la fabricación de equipos médicos, propuesto en el presente proyecto de graduación. Además de identificar los conceptos necesarios para la selección de sensores, unidades de procesamiento de la información y plataformas de presentación de datos.

### **Automatización de sistemas de monitoreo**

La automatización de procesos es uno de los mayores objetivos de las empresas hoy debido a la alta competitividad. La reducción de tiempos de operación, optimización de las actividades realizadas por el personal y equipo utilizado, representan una reducción significativa en los costes de operación y un impacto positivo en la calidad final del producto.

Cuando se habla de automatización, no solo depende de un grupo de personas o actividades, sino de un conjunto de factores que favorecen la creación de un sistema automatizado. Para llegar a un sistema automatizado es necesario realizar un análisis del entorno del proceso, considerando la maquinaria, el personal, procesos, etc, de manera que cada factor tenga un impacto positivo sobre los demás, dando como resultado un proceso automatizado.

García (2001) define la automática como “la ciencia y técnica de la automatización, que agrupa el conjunto de las disciplinas teóricas y tecnológicas que intervienen en la concepción, la construcción y el empleo de los sistemas automáticos” (p.7).

Con los avances de las tecnologías, la incorporación de microcontroladores, sensores y actuadores ha permitido una mayor integración de la automatización a los distintos procesos.

En la siguiente figura se muestra un ejemplo de modelo estructurado de un sistema automatizado, en donde es posible observar la interacción que existe entre la parte operativa y la parte de control. Entendiéndose que los captadores pueden representar los sensores del sistema de monitoreo que recogen los valores de las magnitudes físicas a controlar, mientras que los preaccionadores son los encargados de ejecutar en la parte operativa las órdenes del dispositivo de control.

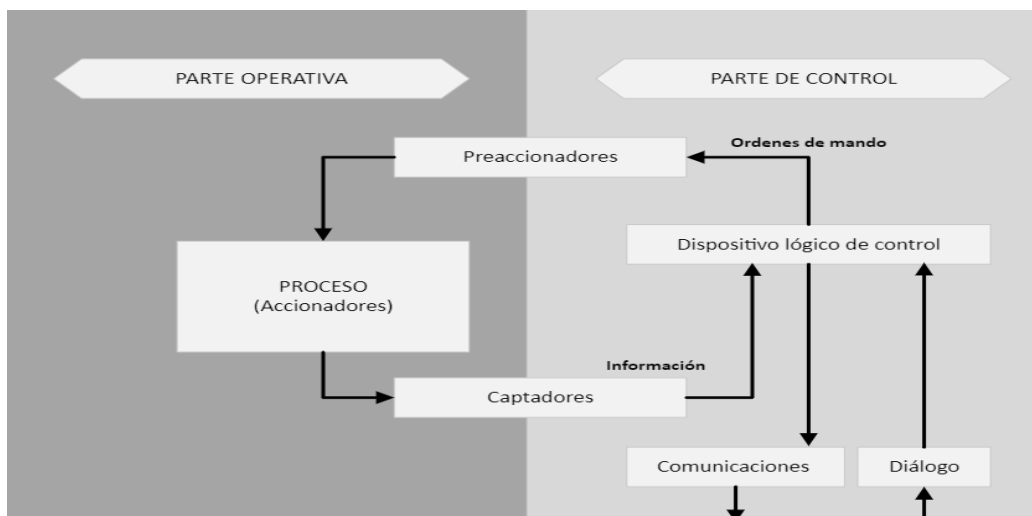


Figura 3: Modelo estructural de un sistema automatizado.

Fuente: Propia.

Si se llevara la definición de un sistema automatizado a un sistema de monitoreo, será posible implementar un modelo capaz de medir las magnitudes físicas del cuarto de ambiente controlado, de manera que permita la obtención y análisis de datos, para asegurar un buen comportamiento de las condiciones ambientales dentro de las instalaciones.

En la industria de fabricación de dispositivos médicos, la implementación de cuartos limpios es una de las principales medidas utilizadas para reducir de manera significativa los niveles de contaminación durante los procesos de producción.

Como se mencionó anteriormente, la Organización Internacional para la Estandarización (ISO), por sus siglas en inglés, ISO 14644-1:2015, define a un cuarto limpio como aquel donde el número de concentración de partículas transportadas en el aire, es controlado y clasificado, además que es diseñado, construido y operado de manera que se controle la introducción, generación y retención de partículas dentro del cuarto.

Esta norma define un número de clase ISO(N) de acuerdo con el nivel de limpieza del aire por concentración de partículas en el cuarto limpio. En la tabla 1, se puede observar la clasificación de la calidad del aire de acuerdo con la concentración de partículas.

Numero de clase ISO (N)	Máximas concentraciones (partículas/m <sup>3</sup> ) para partículas iguales o mayores que el tamaño considerado, mostrado en la tabla de abajo <sup>a</sup> .					
	0.1 µm	0.2 µm	0.3 µm	0.5 µm	1 µm	5 µm
1	10 <sup>b</sup>	d	d	d	d	e
2	100	24 <sup>b</sup>	10 <sup>b</sup>	d	d	e
3	1000	237	102	35 <sup>b</sup>	d	e
4	10 000	2370	1 020	352	83 <sup>b</sup>	e
5	100 000	23700	10 200	3 520	832	d,e,f
6	1 000 000	237 000	102 000	35 200	8 320	293
7	c	c	c	352 000	83 200	2 930
8	c	c	c	3 520 000	832 000	29 300
9g	c	c	c	35 200 000	8 320 000	293 000

a. Todas las concentraciones en la tabla son acumulativas, e.g. para ISO clase 5, las 10 200 partículas mostradas a 0.3 µm incluyen todas las partículas iguales o mayores a este tamaño.

b. Estas concentraciones van a liderar muestras largas de aire para clasificación. Procedimiento de muestreo secuencial debe ser aplicado.

c. Límites de concentración no son aplicables en esta región de la tabla debido a la alta concentración de partículas.

d. Muestreo y limitaciones estadísticas para partículas en bajas concentraciones hace la clasificación inapropiada.

e. Limitaciones de la recolección de muestras para partículas en bajas concentraciones y en tamaños mayores a 1 µm hacen la clasificación para este tamaño de partículas inapropiado, debido a pérdidas potenciales de partículas en el sistema de muestreo.

f. Con el fin de especificar este tamaño de partícula en asociación con ISO clase 5, las micropartículas descripción M, debe ser adoptado y ajustado en conjunción con al menos otro tamaño de partícula.

g. Esta clase es únicamente aplicable para el estado en operación.

*Tabla 1. ISO clases de calidad del aire de acuerdo con la concentración de partículas*

*Fuente: Norma ISO 14644-1:2015.*

Tomando en cuenta la información de la tabla anterior, se puede observar que cada vez que se desciende en la clase de un cuarto limpio número de clase ISO(N), la concentración de partículas por metro cúbico y tamaño permitido en su interior es menor.

Para determinar la clase de cuarto limpio adecuado para cada proceso de manufactura es importante tomar en cuenta los requerimientos del cliente, proceso o regulaciones existentes para cada industria. Los cuartos limpios por lo general son implementados en industrias como la farmacéutica, aeroespacial, electrónica, médica y de laboratorios, entre otros. Para todas estas industrias el requerimiento de un cuarto de ambiente controlado es de vital importancia para evitar

impactos en la calidad del producto debido a contaminaciones en el aire. En el caso de la industria de dispositivos médicos no es la excepción, y el nivel de requerimiento de clasificación del cuarto limpio, número de clase ISO(N), utilizado para desarrollar todas las actividades de manufactura es determinado de acuerdo con la evaluación del riesgo del producto, donde se evalúa los impactos en el producto/paciente, por contaminación en el aire durante el proceso de manufactura. Las evaluaciones del nivel de riesgo del producto son un requerimiento regulatorio en los sistemas de calidad, de acuerdo con la FDA (*Food and Drug Administration*), en el título 21, CFR, Part 820.30(g).

La idea de implementar un sistema de monitoreo, utilizando un modelo estructural de automatización y su relación con el espacio físico donde se desea implementar, en este caso un cuarto limpio, permite entender mejor las cualidades que se buscan tener en el sistema de monitoreo.

### **Puntos de muestreo en un cuarto limpio**

Los puntos de muestreo se pueden definir como ubicaciones específicas que fueron estudiadas con anterioridad para obtener y recopilar las mediciones de la concentración de partículas en el aire. Estos puntos de muestreo deben ser seleccionados estratégicamente en diferentes áreas del cuarto limpio, para obtener mediciones representativas de la calidad del aire dentro del cuarto limpio.

Para realizar la selección estratégica de los puntos de muestreo, deben identificarse áreas del cuarto limpio donde el control de partículas en el aire es importante para la operación que se lleve a cabo, como por ejemplo cerca de operaciones críticas, áreas de transferencia de material, ingreso de personal, entre otros, que sean puntos representativos para la toma de decisiones dirigidas a la calidad del aire dentro del cuarto limpio.

La ISO 1464-1:2015 también explica en su anexo A como calcular el número mínimo de puntos de muestreo requeridos en el cuarto limpio, para obtener un 95% de garantía de que el 90% del área del cuarto limpio no excede la concentración de partículas/m<sup>3</sup> de la clase.

### Establecimiento de puntos de muestreo

Para determinar la cantidad de puntos de muestreo requeridos para un cuarto limpio, es necesario contar con el dato del área total del mismo. En la siguiente tabla, se muestran los puntos mínimos de locaciones de muestreo,  $N_L$ , respecto al área del cuarto limpio.

Área del cuarto limpio(m <sup>2</sup> ) menor o igual a	Número mínimo de puntos de muestreo ( $N_L$ )
2	1
4	2
6	3
8	4
10	5
24	6
28	7
32	8
36	9
52	10
56	11
64	12
68	13
72	14
76	15
104	16
108	17
116	18
148	19
156	20
192	21
232	22
276	23
352	24
436	25
636	26
1 000	27
>1000	Ver fórmula (1) del presente documento.
NOTA 1	Si el área considerada se encuentra entre dos valores de la tabla, se debe seleccionar el mayor de los dos.

Área del cuarto limpio(m <sup>2</sup> ) menor o igual a	Número mínimo de puntos de muestreo (N <sub>L</sub> )
NOTA 2	En el caso de flujo de aire unidireccional, el área puede considerarse como la sección transversal del aire en movimiento perpendicular a la dirección del flujo de aire. En todos los demás casos, el área podrá considerarse como el área en planta horizontal del cuarto limpio o zona limpia.

Tabla 2. Puntos de muestreo relacionados con el área del cuarto limpio.

Fuente: Norma ISO 14644-1:2015

Tomando en cuenta los puntos de muestreo mínimos requeridos, utilizando la tabla 2, para distribuirlos de manera equitativa en el área del cuarto limpio, basta con dividir el número de puntos de muestreo N<sub>L</sub> entre el área total del cuarto limpio.

En los casos donde el área del cuarto limpio es superior a los 1000 m<sup>2</sup>, es posible aplicar la fórmula (1), para determinar la cantidad de puntos de muestreo.

$$N_L = 27 \times \left( \frac{A}{1000} \right) \quad (1)$$

Donde:

N<sub>L</sub>: Corresponde al número mínimo de puntos de muestreo a evaluar, redondeado al siguiente número entero.

A: Área del cuarto limpio en m<sup>2</sup>.

### Consideraciones para la ubicación de puntos de muestreo

De acuerdo con la norma ISO 1464-1:2015, para la medición de partículas se debe tomar en cuenta lo siguiente:

- seleccionar dentro de cada sección un punto de muestreo que se considere representativo de las características de la sección.
- En cada punto colocar el medidor contador de partículas en el plano de la actividad de trabajo o en otro punto especificado.
- Los puntos de muestreo deben ser seleccionados para ubicaciones consideradas como críticas.

- En el caso de cuartos limpios con flujo de aire no unidireccional, los puntos de muestreo no serán representativos si se colocan directamente debajo de fuentes de suministro de aire.
- Con respecto al volumen de aire, para que este sea representativo para la medición debe ser lo suficiente para detectar un mínimo de 20 partículas para el tamaño de partícula más grande seleccionado a monitorear de acuerdo con el número de la clase ISO (N).
- El sensor o la sonda de prueba debe estar posicionada apuntando hacia el flujo de aire. Si la dirección del flujo de aire no es controlado o predecible (no es unidireccional), la entrada de la sonda de medición o del sensor debe estar directamente vertical hacia arriba.

De acuerdo con la norma ISO 1464-3:2019, para la medición de presión se debe tomar en cuenta lo siguiente:

- La diferencia de presión aceptable entre cuartos debe estar definida
- Se debe asegurar que los componentes que puedan afectar las mediciones de presión, como puertas, ventanas, zonas de paso, etc, deben permanecer cerrados.
- Se deben establecer puntos fijos para las mediciones
- Los puntos de medición no pueden estar cerca de entradas de aire, retornos, dispositivos con movimiento de aire, puertas u otras zonas con altas velocidades de aire que puedan influenciar con la presión local en el punto de medición.

Con respecto a la posición de los puntos de muestreo para la medición de temperatura y humedad, la norma ISO 1464-3:2019 de cuartos limpios y ambientes controlados asociados, menciona que para los métodos de medición de temperatura y humedad se debe seguir la ISO 7726. De acuerdo con la ISO 7726 de 1998 de ergonomía del entorno térmico-Instrumentos para medir magnitudes físicas, menciona que el sensor de temperatura y humedad debe estar efectivamente protegido de cualquier efecto de la radiación térmica proveniente de paredes calientes o frías.

### **Requerimientos mínimos para un plan de monitoreo**

En la norma ISO 1464-2:2015 establece los lineamientos con los que debe contar un sistema de monitoreo, para ofrecer evidencia del rendimiento del cuarto limpio con relación a la limpieza del aire por concentración de partículas.

Entre estos lineamientos se encuentra que se debe crear un plan de monitoreo, el cual debe ser implementado y mantenido, tomando en cuenta el nivel de limpieza del aire requerido para el proceso llevado a cabo en el cuarto, además de las ubicaciones críticas y atributos que puedan afectar el desempeño de la instalación.

A continuación, se listan los requerimientos mínimos que dicta la norma:

- a) Listar y justificar todos los parámetros que van a ser monitoreados, incluyendo aquellos que puedan afectar la concentración de partículas en el aire.
- b) Descripción y justificación de los métodos de medición
- c) Precisión, mantenimiento y calibración de los instrumentos de monitoreo
- d) Identificación y justificación de las ubicaciones seleccionadas de monitoreo. Ubicaciones de monitoreo deben ser definidas en 3 dimensiones.
- e) Identificación y justificación de los criterios de aceptación o límites del sistema de monitoreo, incluyendo un nivel de alarma único o dual y niveles de acción. El requerimiento mínimo es un nivel de acción de alarma único. Adicional, un nivel de alarma puede ser establecido para proveer una advertencia temprana a alguna desviación del desempeño.
- f) Especificación de la respuesta requerida en caso de que los datos queden fuera de los límites especificados.
- g) La necesidad y frecuencia de una clasificación periódica de la limpieza del aire del cuarto limpio por concentración de partículas de acuerdo con ISO 1464-1:2015, 5.1.
- h) El formato para registrar datos
- i) Los métodos, incluidos los métodos estadísticos que se utilizarán para determinar las tendencias de los datos u otros métodos apropiados.
- j) Los requisitos de presentación de informes
- k) La política y los medios que se utilizarán para la retención de registros
- l) La frecuencia de revisión del plan de monitoreo

Tomando en cuenta los puntos anteriores como referencia, los cuales dictan cuales deben ser los aspectos mínimos que se deben cumplir para obtener un plan de monitoreo exitoso para un cuarto limpio. El presente proyecto los tomará únicamente como referencia, dado que la propuesta de diseño se enfocará en el proceso de adquisición, almacenamiento y visualización de las variables de temperatura, humedad relativa, presión y cantidad de partículas dispersas en el aire,

para que pueda ser adaptado por cualquier fabricante de dispositivos médicos, dado que varios de los requisitos anteriores, formatos, justificaciones y políticas van a depender de cada fabricante.

De acuerdo con el requerimiento a), de un plan de monitoreo, indica que se deben incluir todos aquellos parámetros que puedan afectar a la concentración de partículas. De los principales parámetros que se monitorean en los cuartos limpios, se encuentran la temperatura, humedad relativa y presión absoluta.

Los parámetros de la temperatura y presión pueden tener efectos directos en la concentración de partículas en el aire, dado que al existir una mayor temperatura dentro del cuarto provocaría que la densidad del aire disminuyera, teniendo como efecto una mayor suspensión de partículas en el aire durante más tiempo.

Esto se puede ver expresado mediante la ecuación de los gases ideales:

$$P = \rho RT \quad (3)$$

Donde:

P: Presión

$\rho$ : densidad

R: Constante de gases

T: Temperatura (°K)

En cuanto a la presión absoluta del cuarto limpio, es importante mantener bajo monitoreo este parámetro dado que los cuartos limpios operan a una mayor presión respecto al exterior, para evitar el ingreso de partículas que podrían llegar a provocar efectos adversos sobre las condiciones de operación del cuarto.

La norma ISO 1464-2:2015, detalla que el monitoreo continuo de todos estos parámetros obtenidos del cuarto limpio puede traer potenciales beneficios como:

- a) Una rápida respuesta a condiciones y eventos adversos
- b) Capacidad para desarrollar tendencias de datos a lo largo del tiempo
- c) Integración de datos de múltiples instrumentos
- d) Adquirir un conocimiento mejorado de la instalación y el proceso, permitiendo así evaluaciones de riesgos más efectivas.

e) Mejor control de los costos operativos y las pérdidas de productos

El cumplimiento de los requerimientos mínimos de un plan de monitoreo permitirá tener bajo control todas aquellas variables que puedan influir en la concentración de partículas en el aire dentro del cuarto limpio, y a su vez garantizará que los datos sean obtenidos, registrados, analizados y presentados de manera que sean representativos para las condiciones ambientales del cuarto limpio.

### **Características de los sistemas de medida**

Los sistemas de medida se pueden definir como

la combinación de dos o más elementos necesarios para realizar una o varias funciones. la realización de un sistema de medida implica, la adquisición de la información, realizada por algún elemento, también el procesamiento de dicha información y la presentación de resultados que puedan ser percibidos por nuestros sentidos. (Pallas Areny, 2003, pág. 1)

Por lo general los elementos que componen los sistemas de medida dispuestos a la obtención de información, se denominan sensores. Corona et al., (2014) “define los sensores como dispositivos de entrada que proveen una salida manipulable de la variable física medida” (p.17).

Según Pallas Areny (2003),

añade que un sistema de medida viene condicionado por el sensor empleado, es por esto que considera importante describir las características de los sensores. Las características estáticas influyen también en el comportamiento dinámico del sensor, es decir, en el comportamiento que presenta cuando la magnitud medida varía a lo largo del tiempo (p. 12).

#### **Características estáticas**

Un punto importante por considerar en los sensores, son las características estáticas, ya que representan el comportamiento de los sensores en condiciones constantes. A continuación, se listan algunas de ellas:

**Exactitud**

La exactitud es la cualidad que caracteriza la capacidad del instrumento de medida de dar indicaciones que se aproximen al verdadero valor de la magnitud medida. [...]La exactitud de un sensor puede ser determinada mediante la calibración estática”, que consiste en definir un patrón de referencia que debe conocerse con una exactitud de al menos diez veces mayor que la del sensor que se calibra. “La discrepancia entre la indicación del instrumento y el verdadero valor de la magnitud medida se denomina; error absoluto. (Pallas Areny, 2003, pág. 12)

**Fidelidad**

La fidelidad es la cualidad que caracteriza la capacidad de un instrumento de medida de dar el mismo valor de la magnitud medida, al medir varias veces en unas mismas condiciones determinadas, prescindiendo de su concordancia o discrepancia con el valor real de dicha magnitud. La fidelidad implica que se tenga simultáneamente una conformidad en las sucesivas lecturas y un número alto de cifras significativas y es, por tanto, una condición necesaria pero no suficiente para la exactitud. (Pallas Areny, 2003, pág. 13)

**Repetibilidad**

Se refiere a cuando las medidas se realizan en un intervalo de tiempo corto, bajo las mismas condiciones. Cuantitativamente es el valor por debajo del cual se encuentra, con una probabilidad especificada, el valor absoluto de la diferencia entre dos resultados individuales obtenidos en las condiciones antedichas. Si no se dice lo contrario, la probabilidad se toma del 95%. (Pallas Areny, 2003, pág. 14)

**Reproducibilidad**

La reproducibilidad se refiere también al grado de coincidencia entre distintas lecturas individuales cuando se determina el mismo parámetro con un método concreto, pero con un conjunto de medidas a largo plazo o realizadas por

personas distintas o con distintos aparatos o en diferentes laboratorios. (Pallas Areny, 2003, pág. 14)

### **Sensibilidad**

Según Corona et al. (2014)

la sensibilidad de un sensor se define como la entrada mínima que requiere este para provocar una salida detectable. La representación gráfica de cómo cambia la salida del sensor con respecto a la entrada se conoce como curva de salida, donde la pendiente de la recta tangente a esta curva constituye la sensibilidad del sensor (p.19).

### **Linealidad**

La linealidad depende principalmente de factores ambientales, por lo que se define como la desviación que representa el sensor entre la curva proporcionada por el fabricante en condiciones controladas y la curva de salida actual. La no-linealidad estática se expresa en porcentaje, el cual refleja que tanto se aleja el sensor de la curva ideal y el valor máximo a escala completa. (Corona et al., 2014, p. 19)

Se puede expresar mediante la ecuación 4:

$$\% \text{ no linealidad} = \frac{\text{desviación máxima}}{\text{valor máximo a escala completa}} \times 100 \quad (4)$$

### **Características dinámicas**

A continuación, se listan las características dinámicas que describen como un sensor responde a cambios en las condiciones de entrada a lo largo del tiempo. A continuación, se listan algunas de ellas:

#### **Tiempo de respuesta**

Se define como “el periodo que transcurre desde que la variable medida presenta un cambio de estado y el sensor lo registra. El tiempo de respuesta depende del tipo de magnitud que se esté midiendo y el sensor utilizado” (Corona et al., 2014, p. 20). El tiempo requerido de

respuesta del sensor, dependerá de la variable para la cual fue diseñado para medir y con qué velocidad se requiere medir la variable.

### **Histéresis**

“Es la capacidad que tiene el sensor para seguir la curva de salida ideal, debido a los cambios en la salida de la variable física”. (Corona et al., 2014, p. 20). “La histéresis sino es especificada en la precisión del sensor, puede llegar a representar una causa de incertidumbre en la medición”. (Ruonala, 2023)

### **Linealidad dinámica**

“Es la capacidad de un sensor para seguir correctamente la curva de salida dada por el fabricante cuando la variable física experimenta cambios repentinos y rápidos”. (Corona et al., 2014, p. 21)

### **Error dinámico**

“La posibilidad de estos errores se entiende si se considera que en el resultado de una medida influye no sólo el aparato empleado para efectuarla sino también el método, operario o circunstancia ambientales que no son ideales, constantes o conocidas”. (Pallas Areny, 2003, p. 16)

El conocimiento de todas estas características estáticas y dinámicas de los sensores permitirá tener un conocimiento técnico de las características que posee cada sensor, para así realizar un análisis de sus capacidades, cualidades y funciones que puedan desempeñar de acuerdo con sus capacidades físicas y de programación de interpretar los datos de las variables físicas medidas.

## **Tipos de sensores**

En este apartado se van a listar algunos de los sensores que podrían implementarse en el sistema de monitoreo de acuerdo con los objetivos de este proyecto, clasificándose con relación al tipo de variable capaz de medir y sus características técnicas. Todos los sensores listados corresponden a sensores compatibles con la plataforma de Arduino.

### **Sensores de temperatura y humedad**

Respecto a las variables de temperatura y humedad, Corona expone que “la variable de temperatura corresponde a la intensidad de calor de un objeto. Es la energía asociada a los movimientos de las moléculas del sistema”. (Corona et al., 2014, p. 174).

Por otro lado, cuando nos referimos a humedad,

esta se relaciona con las moléculas de agua que pueden encontrarse en algunas sustancias. Esta variable se puede expresar como humedad relativa, específica y absoluta. La humedad absoluta; representa la relación entre la cantidad de moléculas de agua presentes en una sustancia y el volumen de esta. Humedad relativa; representa la relación entre la cantidad actual de agua que contiene un gas y la cantidad que el gas tendría en estado de saturación sin que se produzca condensación. Mientras que la humedad específica, es la relación entre la masa de agua y la masa de la sustancia seca (Corona et al., 2014, p. 200).

Para el presente proyecto se tomará en cuenta solamente la medición de la humedad relativa, para la selección de los sensores de humedad.

### **Sensor DHT11**

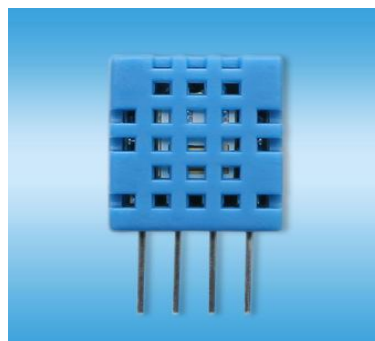
El sensor DHT11 se presenta como una opción para la medición de la temperatura y la humedad, dada su fiabilidad y bajo precio. Según el fabricante AOSONG, en su manual técnico explica al DHT11 como un sensor digital de temperatura y humedad, el cual incluye un sensor resistivo de componentes húmedos y dispositivos de medición de temperatura NTC (Coeficiente de Temperatura Negativa, por sus siglas en inglés), conectados a un microcontrolador de 8 bits de alto rendimiento. En la siguiente tabla se muestra sus especificaciones técnicas:

<b>Características</b>	<b>Humedad</b>	<b>Temperatura</b>
Alimentación	3.5 -5.5 V DC	
Corriente de alimentación	0.3 mA	
Rango de medición	20% a 90%RH	0 a 50 °C
Precisión	a 25 °C ± 5% RH	± 2.0 °C
Resolución o sensibilidad	16 Bit, 1% RH	16 Bit, 0.1 °C
Repetibilidad	± 1% RH	± 0.2 °C
Histéresis de humedad	<± 0.3% RH	N/A
Estabilidad a largo plazo	<± 0.5% RH /year in	N/A

Dimensiones	12 x 15.5 x 5.5mm
-------------	-------------------

*Tabla 3. Características sensor DHT11.*

*Fuente: Propia. Obtenido de DHT 11 manual del producto (AOSONG)*



*Figura 4: Sensor de Temperatura y Humedad DT11*

*Fuente: Guangzhou Aosong Electronics Co., 2021*

### **Módulo AM2301**

El módulo AM2301 es capaz de medir la temperatura y humedad relativa. Este módulo utiliza un sensor de humedad capacitivo y un termistor para medir el aire circundante. (Novatronic, 2020). Este sensor es calibrado en fábrica para obtener coeficientes de calibración grabados en su memoria OTP, asegurando alta estabilidad y fiabilidad a lo largo del tiempo. El fabricante AOSONG también añade que su tamaño pequeño, bajo consumo y una distancia de transmisión de señal de 20 m, lo vuelve útil para aplicaciones como control automático, equipo de inspección, registro de datos, regulador de humedad, entre otros (AOSONG, 2020).

El uso de este sensor representa una buena opción en casos donde por cuestiones de dimensiones de cuarto limpio sea requerido recorrer una larga distancia entre la ubicación del sensor y el microcontrolador que reciba la información.



*Figura 5. Módulo de Temperatura y Humedad AM2301*

Fuente: AOSONG, 2020

En las siguientes tablas se muestra las especificaciones técnicas que son de principal interés:

Parámetro	Condición	min.	typ	máx.	Unidad
Resolución			0,1		%RH
Rango de medición		0		99,9	%RH
Precisión	25 °C		±3		%RH
Repetibilidad			±1		%RH

Tabla 4. Características de humedad, módulo AM2301.

Fuente: Propia, obtenido de AM2301 manual del producto (AOSONG)

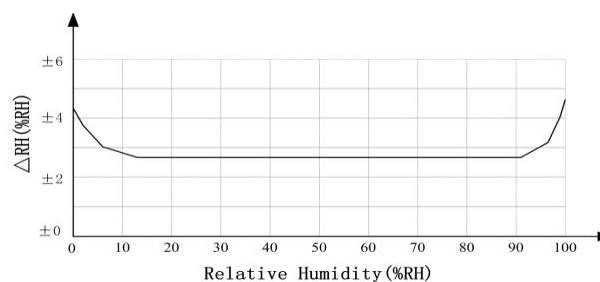


Figura 6. Error de humedad relativa a 25 °C

Fuente: AOSONG, 2020

Parámetro	min.	typ	máx.	unidad
Resolución		0,1		°C
		16		bit
Precisión		± 0,3	± 1	°C
Rango de medición	-40		80	°C
Repetibilidad		± 0,2		°C

Tabla 5. Características de temperatura, módulo AM2301.

Fuente: Propia. Obtenido de AM2301 manual del producto (AOSONG)

Parámetro	Condición	min.	typ	máx.	unidad
Voltaje		3,3	5	5,2	V
Consumo	Inactivo	10	15		μA
	Midiendo		500		μA
	Promedio		300		μA
Dimensiones		26,7 x 58,8 x 13,8			mm

*Tabla 6. Características de eléctricas, módulo AM230.*

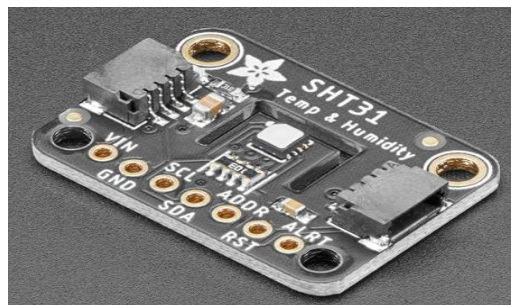
*Fuente: Propia. AM2301 manual del producto (AOSONG).*

Con base a la información técnica anterior, es posible identificar tanto las características estáticas como dinámicas del sensor, así como su comportamiento en condiciones óptimas de funcionamiento. Incluyendo sus capacidades para operar dentro de un rango específico y las posibles condiciones de operación que podrían ocasionar errores en las mediciones.

### **SHT31-D**

El sensor SHT31-D, fabricado por SENSIRION, es capaz de medir las variables de temperatura y humedad relativa con una precisión de  $\pm 0.3$  °C y  $\pm 2\%$  de RH, respectivamente. Destaca por su capacidad de comunicación a través del protocolo I2C (Circuito inter-integrado, por sus siglas en inglés) y su capacidad para operar con hasta 2 direcciones I2C distintivas, seleccionables por el usuario, con una comunicación de hasta 1 Mhz.

Además, su amplio rango de voltaje de alimentación, que va desde 2.4 hasta 5.5 V, garantiza su compatibilidad con diversas situaciones de ensamblado y aplicación. El sensor también cuenta con funciones que le permiten entrar automáticamente en modo reposo cuando está encendido, pero no recibe ninguna instrucción para realizar mediciones, con la finalidad de ahorrar energía. (SENSIRION, 2015)



*Figura 7. Sensor de temperatura y Humedad SHT31-D.*

*Fuente: Adafruit Sensirion SHT31-D, 2021.*

### **Si7021-A20**

El sensor Si7021-A20, fabricado por Silicon Labs, corresponde a un sensor digital de temperatura y humedad que integra un convertidor analógico a digital, procesamiento de señales, calibración e interfaz de comunicación I2C. El sensor Si7021-A20 está calibrado individualmente en fábrica tanto para temperatura como humedad, con los datos almacenados en la memoria no volátil del chip. Esto asegura que el sensor sea completamente intercambiable, sin necesidad de recalibración ni cambios en el *software*.

Otra característica de este dispositivo corresponde a que

este sensor ofrece excelente estabilidad a largo plazo e inmunidad a contaminantes con baja deriva e histéresis, siendo una solución de baja potencia, alta precisión, calibrada y estable para aplicaciones de temperatura y humedad, en áreas médicas, automotrices, de instrumentación e industriales (LABS, 2022).

El sensor Si7021-A20 tiene las capacidades de medir la humedad relativa en un rango de 0 a 80% con una precisión de  $\pm 2\%$  de RH, y la temperatura ambiental en un rango de  $-10$  a  $85$  °C con una precisión de  $\pm 0.3$  °C. El voltaje de operación de este sensor es de 1.9 V a 3.6 V y un consumo mínimo de corriente mientras realiza las mediciones de en promedio de  $150 \mu\text{A}$ . (LABS, 2022)

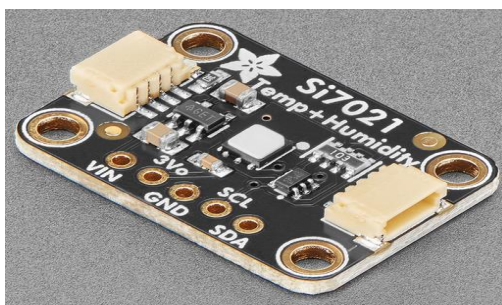


Figura 8. Sensor de temperatura y Humedad Si7021-A20.

Fuente: Adafruit Si7021 Temperature & Humidity Sensor Breakout Board - STEMMA QT, 2024.

### **Sensores de presión**

Como parte de las variables a medir en el cuarto limpio, una de ellas corresponde a la presión. Es importante especificar que un sensor de presión corresponde a un transductor de voltaje, entre la fuerza aplicada en un área determinada y una señal

eléctrica. Por lo general la unidad de medida utilizada en los sensores de presión corresponde a los bares, ya que su equivalencia en atmosferas de presión se expresa en  $1 \text{ atm} = 1.03525 \text{ bares}$ . (Corona et al., 2014, p. 236)

Los sensores de presión se clasifican de acuerdo con el esquema de la figura 9. En esta imagen es posible ver que hay dos tipos; presión absoluta y presión diferencial, que esta a su vez puede ser relativa. Los sensores de presión absoluta tienen una cámara sellada, la cual contiene la presión de referencia, con un valor de cero como punto de referencia absoluto. “Los sensores de presión diferenciales funcionan bajo una diferencia de presiones, entre una presión de referencia dada y una presión detectada. Si se utiliza una presión barométrica de referencia, entonces el sensor se clasifica de presión relativa”. (Corona et al., 2014, p. 236)

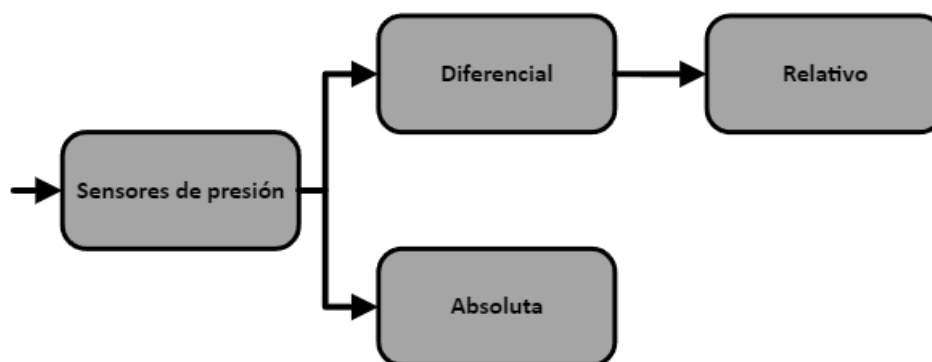


Figura 9. Clasificación de los sensores de presión

Fuente: Propia, basado en Corona et al., 2014.

### **BMP280**

El sensor BMP280, corresponde a un sensor de presión barométrico digital, del fabricante BOSCH. El sensor utiliza la tecnología de presión piezorresistivo, brindando una alta precisión, linealidad y estabilidad a largo plazo. Además, cuenta con las interfaces digitales de comunicación mediante los protocolos I2C y SPI (Interfaz Periférica en Serie, por sus siglas en inglés).

“Este sensor es bastante compacto, con un bajo consumo de  $2.7\mu\text{A}$  @1 Hz, permitiéndole ser incluso implementado en dispositivos móviles, como celulares o relojes” (Bosch Sensortec, 2021). Además, cuenta con un conjunto de configuraciones para el muestreo de la presión que va desde un consumo bajo de energía hasta una resolución ultra alta dependiendo de la aplicación que se le quiera dar.



*Figura 10. Sensor BMP280*

*Fuente: Adafruit, 2022*

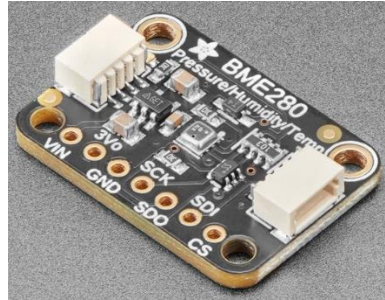
El sensor BMP280 cuenta con un rango de medición de la presión barométrica de 300 a 1100 hPa con una precisión de  $\pm 1.0$  hPa, además tiene la capacidad de medir la variable de temperatura en el rango de  $-40$  a  $85$  °C con una precisión de  $\pm 1.0$  °C, requiriendo un voltaje de alimentación de 1.71 V a 3.6 V.

### **BME280**

El sensor BME280 corresponde a un sensor digital de humedad, presión y temperatura, fabricado por BOSCH. Este sensor tiene aplicaciones en medición, la medición de volumen y flujo de aire, control domótico en el ámbito de climatización para sistemas de calefacción, ventilación y aire acondicionado (HVAC, por sus siglas en inglés), implementación en sistemas IoT y aplicaciones climáticas. ( BOSCH SENSORTEC, 2022)

El sensor cuenta con las interfaces de comunicación de I2C y SPI que puede trabajar con un voltaje de alimentación de 1.71 V a 3.6 V, con un consumo de corriente promedio de  $3.6 \mu\text{A}$  durante la medición de la presión, humedad y temperatura y de  $0.2 \mu\text{A}$  mientras el sensor se encuentra en modo inactivo.

A diferencia del sensor BMP280, tiene la capacidad para monitorear la humedad en un rango de 0 a 100 %RH con una precisión de  $\pm 3\%$  RH, además que para la medición de presión reduce drásticamente el ruido. ( BOSCH SENSORTEC, 2022)



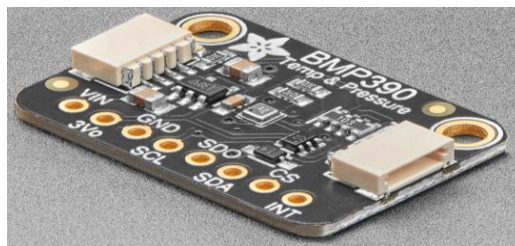
*Figura 11. Sensor BME280*

*Fuente: Adafruit, 2023*

### **BMP390**

El sensor BMP390 corresponde a un sensor digital de presión y temperatura. Es capaz de medir la variable de presión en un rango de 300 hPa a 1250 hPa. Entre sus aplicaciones se encuentra el monitoreo de condiciones ambientales, aplicaciones IoT, navegación en interiores gracias a la capacidad de medir la altitud a la que se encuentra el sensor.

Su funcionamiento consiste en ser un sensor de presión piezorresistivo que provee los resultados mediante una interfaz digital. Este sensor soporta las interfaces digitales I2C y SPI, comportándose como un esclavo para ambos modos de comunicación de los datos. Este sensor funciona con un voltaje de alimentación de 1.65 V a 3.6 V, con un consumo promedio de 3.2  $\mu\text{A}$  y con la capacidad de reducir su consumo hasta 1.4  $\mu\text{A}$  en el modo reposo. (Bosch Sensortec, 2021)



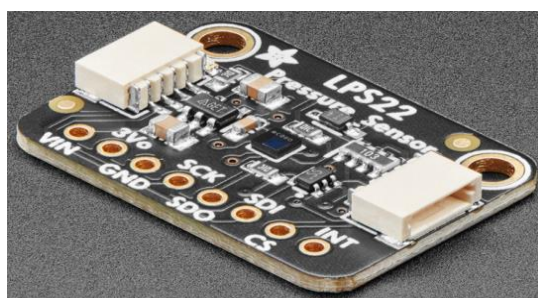
*Figura 12. Sensor BMP390*

*Fuente: Adafruit, 2023*

## LPS22HB

El sensor LPS22HB fabricado por STMicroelectronics, corresponde a un sensor digital ultra compacto piezorresistivo de presión absoluta. Este sensor consta de una membrana suspendida fabricada bajo un proceso desarrollado por ST, la deflexión de la membrana induce un desequilibrio en las piezorresistencias para que luego la señal de salida sea convertida en la interfaz que consta de un convertidor de analógico a digital. Este sensor tiene interfaces I2C y SPI las cuales permiten la comunicación serial con la aplicación. (STMicroelectronics, 2017)

El sensor LPS22HB es capaz de medir la presión absoluta en un rango de 260 a 1260 hPa, además de ser capaz de medir la temperatura en un rango de -40 a 85 °C, requiriendo un voltaje de alimentación de 1.7 V a 3.6 V, con un consumo promedio de corriente de 3  $\mu$ A. Este sensor además incluye la calibración de fábrica, que permite la utilización del dispositivo sin requerir una calibración adicional. (STMicroelectronics, 2017)



*Figura 13. Sensor LPS22HB*

*Fuente: Adafruit, 2022*

## Sensores de medición de partículas

Una de las principales características a monitorear en el cuarto limpio es la cantidad de partículas en el aire para mantener la calidad de este, dentro de los parámetros establecidos. Los sensores de partículas pueden utilizar diversas tecnologías y métodos para llevar a cabo sus mediciones. Los sensores que van a ser analizados en este proyecto son los contadores de partículas ópticas que pueden utilizarse con la plataforma de Arduino.

De acuerdo con Allen (2003), los sensores contadores de partículas ópticas pueden llegar a determinar la distribución de tamaño de partículas, mediante una de las siguientes formas:

- Por la cantidad de luz cortada por una partícula cuando pasa a través de una zona sensible en un haz de luz.
- Recolectando y midiendo la luz que se dispersa sobre un ángulo sólido específico en la dirección de avance.
- Recolectando y midiendo la luz que se dispersa en un ángulo sólido específico con respecto al haz incidente (normalmente un ángulo recto).
- Midiendo el desplazamiento de fase cuando una partícula pasa a través de un rayo láser cruzado.
- Midiendo el "tiempo de vuelo" entre dos rayos láser (p. 468).

Los sensores contadores de partículas ópticas pueden ir desde simples hasta muy sofisticados dependiendo del rango y la amplia gama de aplicaciones que existen, que pueden generar cambios en el tamaño y diseño. “La respuesta del instrumento puede depender de factores como tamaño de partículas, orientación de partículas, longitud de onda de luz e incluso geometría del sistema colector”. (Allen, 2003, p. 468)

Los sensores de partículas presentes en el mercado pueden dar los resultados de las mediciones en conteo de partículas por determinados volúmenes de medición como, por ejemplo, partículas/m<sup>3</sup>, partículas/cm<sup>3</sup>, partículas/litro o concentración de partículas en microgramos por determinados volúmenes de medición como por ejemplo, µg/m<sup>3</sup>, µg/cm<sup>3</sup>, µg/litros.

El total de partículas en suspensión (TSP, por sus siglas en inglés) se utiliza para referirse al total de partículas en el aire, que a menudo se miden en alto volumen sin una entrada de tamaño selectivo. Mientras que la materia de partículas (PM, por sus siglas en inglés) corresponde a la medida de concentración de masa de partículas que pasan a través de una entrada de tamaño selectivo. Por ejemplo: PM10 se refiere a la concentración de partículas para un diámetro de 10 µm. (Tiwary y Williams, 2019)

### **PMSA003I**

El sensor de calidad del aire PMSA003I es fabricado por PANTOWER, y este a su vez es implementado en la placa de la familia de sensores de la empresa Adafruit, la cual se dedica al diseño y fabricación de productos electrónicos y *hardware* de código abierto.

El sensor PMSA003I se puede utilizar para obtener la concentración de partículas suspendidas en el aire, y generar los resultados en forma de interfaz digital.

Además,

su funcionamiento consiste en el principio de dispersión láser, mediante el uso de un láser se irradian partículas en suspensión en el aire, luego con un cierto grado se recoge la luz dispersada y finalmente se obtiene la curva de cambio de la luz dispersada con el tiempo. El microprocesador es quien se encarga de calcular el diámetro de partícula equivalente y el número de partículas con diferentes diámetros por unidad de volumen (Yong, 2018,p.2).

Adafruit incorpora en el sensor la *interfaz* I2C, lo cual lo convierte en una buena opción para computadoras Linux, placas Raspberry Pi, y la plataforma de Arduino que utiliza *software* en serie (Rembor, 2022). Adafruit además añade que el flujo de datos I2C al actualizarse una vez por segundo se obtendrá:

- Concentración de partículas PM1.0, PM2.5 y PM10.0 en unidades estándar y ambientales.
- Partículas por 0.1 litros de aire, se pueden actualizar en tamaños de 0.3  $\mu\text{m}$ , 0.5  $\mu\text{m}$ , 1.0  $\mu\text{m}$ , 2.5  $\mu\text{m}$ , 5.0  $\mu\text{m}$  y 10  $\mu\text{m}$ .

Este sensor se caracteriza por ser capaz de medir la concentración de partículas en los rangos de partículas de 0.3~1.0, 1.0~2.5 y 2.5~10  $\mu\text{m}$ , con una resolución de 1  $\mu\text{g}/\text{m}^3$ .



Figura 14. Sensor Adafruit PMSA003I.

Fuente: Rembor, 2022

### **Sensor Láser SDS011**

El sensor SDS011 fabricado por Nova Fitness, al igual que el sensor PMSA003I mencionado anteriormente, utiliza el principio de dispersión láser, donde la luz dispersada se transforma en señales eléctricas que posteriormente serán amplificadas y procesadas, lo cual le permite medir la concentración de partículas en el aire entre los 0.3 y 10  $\mu\text{m}$ . (Nova Fitness Co., 2021)



*Figura 15. Sensor Láser SDS011.*

*Fuente:* Nova Fitness Co., 2021

El sensor SDS011 cuenta con una capacidad de medir la concentración de partículas de tamaño de 2.5  $\mu\text{m}$  y 10  $\mu\text{m}$  en el rango de 0.0 a 999.9  $\mu\text{g}/\text{m}^3$  con una resolución de 0.3  $\mu\text{g}/\text{m}^3$ . Se caracteriza también por utilizar únicamente la interfaz de comunicación receptor/transmisor asíncrono universal (UART, por sus siglas en inglés).

### **PMS9003M**

El sensor PMS9003M fabricado por PLANTOWER, utiliza el principio de dispersión láser, para obtener la concentración de partículas suspendidas en el aire, y generar los resultados en forma de interfaz digital.

Este sensor tiene la capacidad de medir la concentración de partículas de tamaño 0.3~1.0, 1.0~2.5 y 2.5~10  $\mu\text{m}$  con una resolución de 1  $\mu\text{g}/\text{m}^3$ . A diferencia del modelo PMSA003I, utiliza la interfaz de comunicación UART, la cual puede trabajar en los valores de 3.3 V y su consumo

de corriente puede variar dependiendo de los modos activos o de reposo en las que se configure. (Zhendong y Yong, 2019)



*Figura 16. Sensor PMS9003M.*

*Fuente: Zhendong y Yong, 2019*

#### **HPMA115CO-004**

El sensor HPMA115CO-004 fabricado por Honeywell corresponde a un sensor de partículas basado en un láser que detecta y cuenta partículas mediante dispersión de luz. El rango de concentración de detección es de  $0 \mu\text{g}/\text{m}^3$  a  $1000 \mu\text{g}/\text{m}^3$ . La fuente de luz láser ilumina la partícula a medida que pasa a través de la cámara de detección, la luz se refleja en las partículas y se registra en la fotografía o detector de luz. El sensor analiza la luz y la convierte en una señal eléctrica para calcular la concentración de partículas.

El sensor puede ser implementado en aplicaciones de sistemas de calefacción, ventilación y aire acondicionado (HVAC, por sus siglas en inglés), monitoreo de la calidad del aire en interiores y como monitor portátil de calidad del aire, con una vida útil de uso continuo durante 10 años y totalmente calibrado de fábrica (Honeywell, 2021).

Este sensor es capaz de medir la concentración de partículas para los tamaños de 1.0, 2.5, 4.0 y  $10 \mu\text{g}/\text{m}^3$ , su interfaz de comunicación corresponde a UART capaz de trabajar en los valores de 3.3 V y un consumo de corriente menor a los 80 mA y 20 mA en el caso de modo reposo.



*Figura 17. Sensor HPMA115CO-004.*

*Fuente:* Honeywell, 2021

### **IPS-7100**

El sensor IPS-7100 es un sensor fabricado por Piera Systems, el cual corresponde a un sensor bastante preciso para el monitoreo de la calidad del aire.

Es un sensor optoelectrónico altamente sensible basado en la lectura de conteo de fotones, su procesador se encarga de la lectura y clasificación de partículas según su tamaño. Su algoritmo único para identificar diferentes partículas lo hace adecuado para el monitoreo preciso y en tiempo real de partículas en el aire (Piera Systems, 2021, párr. 1).

El sensor ha sido calibrado con GRIMM 11D en un entorno de calibración controlado y cuenta con una vida útil mayor a 8 años, operando continuamente las 24 horas del día.

Según el fabricante el sensor puede ser implementado en sistemas de gestión y monitoreo de la calidad del aire interior en espacios inteligentes en oficinas o a nivel industrial, sistema de ventilación e incluso detección de humo y vapeo. Entre sus características destaca su capacidad de monitorear no solo la concentración de partículas en  $\mu\text{g}/\text{m}^3$  sino que también es capaz de medir el conteo de partículas en el aire para los tamaños de 0.1, 0.3, 0.5, 1.0, 2.5, 5.0, y 10  $\mu\text{m}$  en un rango máximo de 1 000 000 partículas/litro. (Piera Systems, 2021)

Este sensor puede utilizar además las interfaces de comunicación I2C y UART las cuales pueden funcionar a un voltaje de 3.3 V y un consumo de corriente de 65 mA mientras realiza las mediciones.



*Figura 18. Sensor IPS-7100.*

*Fuente: Piera Systems, 2021*

### **SPS30**

El sensor SPS30 fabricado por SENSIRION

monitorea la calidad del aire y control, donde su principio de medición se basa en la dispersión del láser y una innovadora tecnología de resistencia a la contaminación. Sus algoritmos avanzados permiten una alta precisión para diferentes tipos de partículas y agrupación de tamaño de partículas, permitiendo realizar mediciones continuas durante sus 10 años de vida útil, con dimensiones de 41 mm x 41 mm x 12mm facilitando su capacidad para ser colocados en paredes, además que incluye una calibración de fábrica. (SENSIRION, 2020)

Este sensor es capaz de medir la concentración de partículas en  $\mu\text{g}/\text{m}^3$  para los tamaños de 1.0, 2.5, 4, 10  $\mu\text{m}$ , además de ser capaz de medir el conteo de partículas suspendidas en el aire para el tamaño de partículas de 0.5, 1.0, 2.5, 4.0 y 10.0  $\mu\text{m}$  en las unidades de partículas/ $\text{cm}^3$ . (SENSIRION, 2020)

El sensor SPS30 utiliza las interfaces de comunicación I2C y UART las cuales pueden funcionar a un voltaje de 3.3 V con un consumo de corriente medio durante la medición de 55mA y con un consumo que disminuye hasta los 38  $\mu\text{A}$  en el modo reposo.



*Figura 19. Sensor SPS30.*

*Fuente: SENSIRION, 2020*

### **Consumo de eléctrico**

Con el fin de identificar el consumo eléctrico de los sensores y en sí del sistema de monitoreo es importante conocer las definiciones de potencia y energía.

La potencia se puede definir como la variación respecto al tiempo de gasto o absorción de energía, medida en Watts (W). Mientras que la energía es la capacidad para realizar trabajo, medida en Joules (J). La potencia es una cantidad que varía con el tiempo, así la potencia suministrada por un elemento es el producto de la tensión entre los extremos del elemento y la corriente a través de él. (Charles y Matthew, 2013, p. 10)

Para el cálculo de la potencia eléctrica se puede utilizar la siguiente ecuación:

$$P = V \times I \quad (5)$$

Donde:

P: Potencia eléctrica (W)

I: Corriente eléctrica (A)

V: Diferencia de potencial (V)

Las compañías abastecedoras de electricidad por lo general cobran a sus clientes el monto de energía consumida en Kilowatts-hora (KWh). Para el cálculo del consumo eléctrico en Kilowatts-hora se puede utilizar la siguiente fórmula:

$$E_{(KWh)} = P_{(KW)} \times t_{(h)} \quad (6)$$

Donde:

P: Potencia eléctrica (KW)

t: Tiempo de uso en horas (h)

## **Plataforma Arduino**

### **Historia**

Arduino nació en el año 2005 en el Instituto de Diseño Interactivo de Ivrea, Italia. Este proyecto surgió por la necesidad de contar con un dispositivo para utilizar en las aulas que fuera de bajo costo y fácil de implementar para que los estudiantes pudieran trabajar en proyectos de diseño interactivo. Los principales responsables de la idea y diseño de Arduino fueron Massimo Banzi, David Cuartielles, David Mellis, Tom Igoe y Gianluca Martino. (Arduino, 2023). Arduino pasó de ser un proyecto dirigido a estudiantes sin experiencia en electrónica y programación, a formar una comunidad amplia, provocando que la placa de Arduino se adaptase a nuevas necesidades y desafíos, diferenciando su oferta desde simples placas de 8 bits hasta productos para aplicaciones de IoT, dispositivos portátiles, impresión 3D y entornos integrados. (Arduino, 2018).

### **Características de la plataforma Arduino**

De acuerdo con la página oficial de Arduino, las principales características de la plataforma corresponden a:

- Bajo costo: las placas Arduino son relativamente económicas en comparación con otras plataformas de microcontroladores. La versión menos costosa del módulo Arduino se puede ensamblar a mano, e incluso los módulos Arduino preensamblados cuestan menos de 50 US dólares.

- **Multiplataforma:** el *software* Arduino (IDE) se ejecuta en los sistemas operativos Windows, Mac OS y Linux. La mayoría de los sistemas de microcontroladores de otros fabricantes están limitados a Windows.
- **Entorno de programación simple y claro:** el *software* Arduino (IDE) es fácil de usar para principiantes, pero lo suficientemente flexible como para que los usuarios avanzados lo puedan aprovechar.
- ***Software* de código abierto y extensible:** el *software* Arduino se publica como herramientas de código abierto, disponibles para que programadores experimentados puedan ampliarlo. El lenguaje se puede ampliar a través de bibliotecas C++.
- ***Hardware* extensible y de código abierto:** los planos de las placas Arduino se publican bajo una licencia Creative Commons, por lo que los diseñadores de circuitos experimentados pueden crear su propia versión del módulo, ampliándolo y mejorándolo (Arduino, 2018).

Parte de las características de esta plataforma, la convierten en una herramienta bastante provechosa, ya que no se limita a un solo *hardware* o *software* de programación, sino que pueda ser adaptable a las necesidades del usuario, por lo cual es una de las principales razones por las que el presente proyecto se basa en esta plataforma, para que pueda ser adaptado las condiciones de cada cuarto limpio o preferencias del usuario.

### **Placas de Arduino**

La plataforma de Arduino se puede definir como una plataforma informática física de código abierto para la creación de objetos interactivos que son independientes o colaboran con *software* en las computadoras. “Las placas de Arduino son pequeñas placas microcontroladoras, que contienen una computadora en un pequeño chip (microcontrolador), si bien no es una computadora tan poderosa como una MacBook, son mucho más baratas y útiles para construir dispositivos interesantes”. (Banzi y Shiloh, 2015, p. 15-16).



o una batería (Arduino, 2023). En la siguiente figura 15, se muestra el diagrama para la placa Arduino Uno.

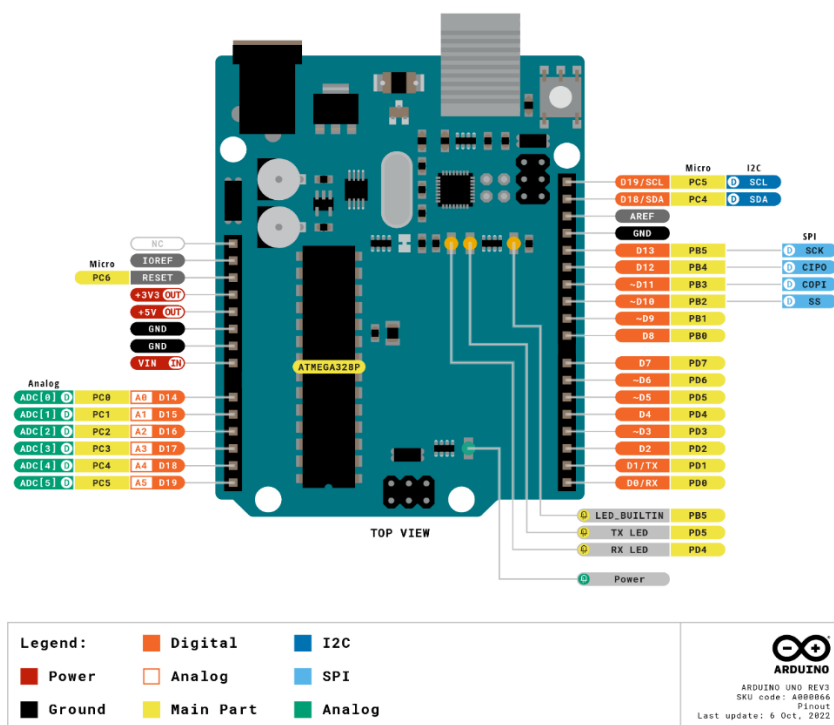


Figura 21. Diagrama Arduino UNO

Fuente: Arduino, 2023.

## Arduino IDE

Peña (2020) define a Arduino como una placa de desarrollo basada en un microcontrolador Atmel. Los microcontroladores son circuitos integrados en los que es posible grabar instrucciones, que deben escribirse con un lenguaje de programación y utilizando un entorno de desarrollo compatible. En el caso del Arduino, este entorno se llama Arduino IDE.

El entorno de desarrollo integrado o IDE de Arduino es una aplicación multiplataforma que se puede utilizar para escribir y cargar programas en placas Arduino y también en aquellas que sean compatibles. El IDE de Arduino permite no solo escribir el código, sino también depurar, editar y grabar los programas en la placa de desarrollo, de manera rápida y sencilla. (Peña, 2020, pp. 1-2)

Peña añade que el IDE corresponde a

un *software* que se distribuye con una licencia libre, de modo que es posible acceder al código fuente del IDE y construir el instalador desde él o realizar las modificaciones que se consideren necesarias. Al ser una multiplataforma, permite que sea posible ser instalada y utilizada en diferentes sistemas operativos, por ejemplo, Microsoft Windows, GNU/Linux o Mac OS, entre otros. (Peña, 2020, p. 3).

Los programas escritos utilizando la plataforma Arduino Software (IDE) son llamados “Sketches”.

Los Sketches son escritos en el editor de texto, el cual permite cortar/pegar y buscar/reemplazar texto. Además, en el *software* IDE se puede contar con librerías que permitirán una funcionalidad extra para el uso de Sketches, en trabajos con *hardware* o manipulación de data. (Arduino, 2023)

Al momento de este trabajo de investigación, la versión más reciente de Arduino IDE corresponde a la versión 2.2.1.

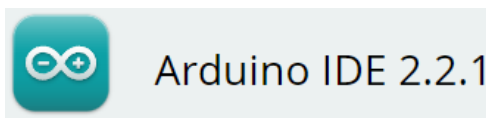


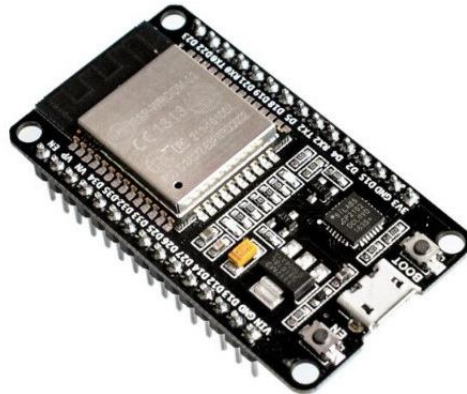
Figura 22. Arduino IDE 2.2.1

Fuente: Arduino, 2023

### **Microcontroladores compatibles con la plataforma Arduino**

En el mercado existen diversos microcontroladores que son capaces de trabajar en la plataforma de Arduino mediante el uso de las librerías. Una de las placas que se tomará en consideración para este proyecto es la placa ESP32. Esta placa corresponde a un módulo MCU, potente y genérico de Wi-Fi + Bluetooth LE, que se dirige a una amplia variedad de aplicaciones, que van desde redes de sensores de bajo consumo hasta tareas exigentes como transmisión de música y codificación de voz. El núcleo de este módulo se encuentran varios modelos de chip desarrollados por el fabricante ESPRESSIF. Estos chips están diseñados para ser escalables y adaptados, junto son sus características de integración de Bluetooth, Bluetooth LE and Wi-Fi. El

uso de Wi-Fi permite un gran alcance físico y de conexión directa a Internet a través de un enrutador con una capacidad de admitir una velocidad de datos de hasta 150 Mbps. También tiene la capacidad de ser alimentado por baterías, lo que facilita aplicaciones portátiles. (Espressif, 2023)



*Figura 23. Placa ESP32-WROOM-32*

*Fuente: CRCibernética, 2023*

Las capacidades de esta placa representan una interesante alternativa para la interconexión de sensores a una red de monitoreo, y que facilitará la obtención y extracción de los datos, mediante el acceso a una red internet, posibilitando así la implementación del internet de las cosas.

Entre las características más importantes a considerar en los microcontroladores se encuentran:

- **Memorias NO-Volátiles:** “en este tipo de memorias cuando se corta la energía, el contenido de la memoria no se pierde y se vuelve a habilitar cuando se enciende el procesador” (Beuchat, Depraz, Guerrieri, y Kashani, 2021, p. 21). En este tipo de memorias se encuentran: ROM (memoria de solo lectura, por sus siglas en inglés), Flash, entre otras.
- **Memorias volátiles:** “tradicionalmente se representan por la memoria RAM (memoria de acceso aleatorio, por sus siglas en inglés), cuando hay un corte de energía, el

contenido de la memoria volátil se pierde. Esta memoria a su vez puede ser estática (SRAM) o dinámica (DRAM)”. (Beuchat et al., 2021, p. 21)

- Señal de reloj: “los circuitos de proceso digital requieren de una señal que les indique la velocidad en la que se ejecutarán sus cálculos internos, esta señal también sirve como sincronía entre los diversos bloques periféricos necesarios para cierta aplicación” (Parra, 2012, p. 28).
- Interfaces de comunicación: estas ayudan a la comunicación de los microcontroladores con otros dispositivos externos para la transferencia de datos entre ellos se encuentran:

### UART

Por sus siglas en inglés se refiere a Receptor-Transmisor Asíncrono Universal,

su principal función es transferir un byte de información a la vez a lo largo de un solo cable. Para controlar la velocidad de bits, una interfaz programable utiliza una frecuencia de reloj que es 16 veces superior a la velocidad de bits seleccionada para garantizar la recuperación de datos con buena precisión. El carácter de transmisión de UART consiste en: un bit de inicio, bits de datos, bit de parada y una velocidad de determinada (Beuchat et al., 2021, p. 337-339).

Esta interfaz utiliza dos líneas de transmisión de datos TxD y RxD y puede ser utilizado en una corta distancia con un voltaje de alimentación típico de 3.3 V y 5V.

### SPI

Es una interfaz para la transferencia de datos serial síncrona utilizando un reloj. SPI implementa una arquitectura maestro-esclavo con un solo maestro y múltiples esclavos. Utiliza cuatro cables que involucran cuatro señales lógicas; SCK, MOSI, MISO y CS. Si hay más de un esclavo conectado a tres líneas básicas, ósea, SCK, MOSI y MISO, la línea CS se utiliza físicamente separada para seleccionar uno esclavo la vez. (Beuchat et al., 2021, p. 349-350)

## I2C

Es un protocolo de bus de comunicación serial síncrono, es más lento que SPI pero también físicamente más simple, solo utiliza dos líneas para la transferencia y sincronización de datos y se usa ampliamente para conectar periféricos a microcontroladores en distancias cortas. Utiliza las dos líneas de transferencia y sincronización SCL el cual es siempre controlado por el maestro y SDA. (Beuchat et al., 2021, págs. 354-355)

## **Internet de las cosas (IoT)**

### **Introducción al internet de las cosas**

Hanes, Salgueiro, Grossetete, Barton, Jerome (2017) definen al internet de las cosas como

una transición tecnológica en la que los dispositivos nos permitirán sentir y controlar el mundo físico haciendo que los objetos sean más inteligentes conectándolos a través de una red. Cuando los objetos y las máquinas puedan ser detectados y controlados de manera remota a través de una red, se habilita una integración más estrecha entre el mundo físico y las computadoras. Esto permite las mejoras en áreas como: precisión, automatización y eficiencia de procesos. (p. 3).

Es importante diferenciar los términos IoT y digitalización, ya que a menudo se usan indistintamente. Según Hanes et al. (2017)

IoT se enfoca en conectar "cosas", como objetos y máquinas, a una red informática, como Internet, mientras que digitalización se define en su forma más simple, como la conversión de información en un formato. La digitalización ha estado ocurriendo de una forma u otra durante varias décadas. Por ejemplo, en el pasado, la gente iba a una tienda a alquilar o comprar cintas de video o DVD de películas. Con la digitalización, casi todos transmiten contenido de video o compran películas como archivos descargables. (Hanes et al., 2017, p. 6-7).

En resumen, se puede definir el internet de las cosas (IoT) como aquella tecnología que trata de conectar lo desconectado, habilitando objetos inteligentes para comunicarse con otros

objetos, sistemas y personas. El resultado final es una inteligente red que permite un mayor control del mundo físico con la habilitación de avanzadas aplicaciones.

### Arquitectura de una red IoT

En 2014, el comité de arquitectura de IoTWF (dirigido por Cisco, IBM, Rockwell Automation, y otros) publicó un modelo de referencia de IoT de siete capas. El modelo presentado por el IoT World Forum ofrece una perspectiva limpia y simplificada sobre IoT que incluye computación de borde, almacenamiento de datos y acceso. Cada una de las siete capas se descompone en funciones específicas, y la seguridad abarca todo el modelo. (Hanes et al., 2017, p. 35)

En la siguiente figura se detalla el modelo de referencia de IoT publicado por IoTWF.

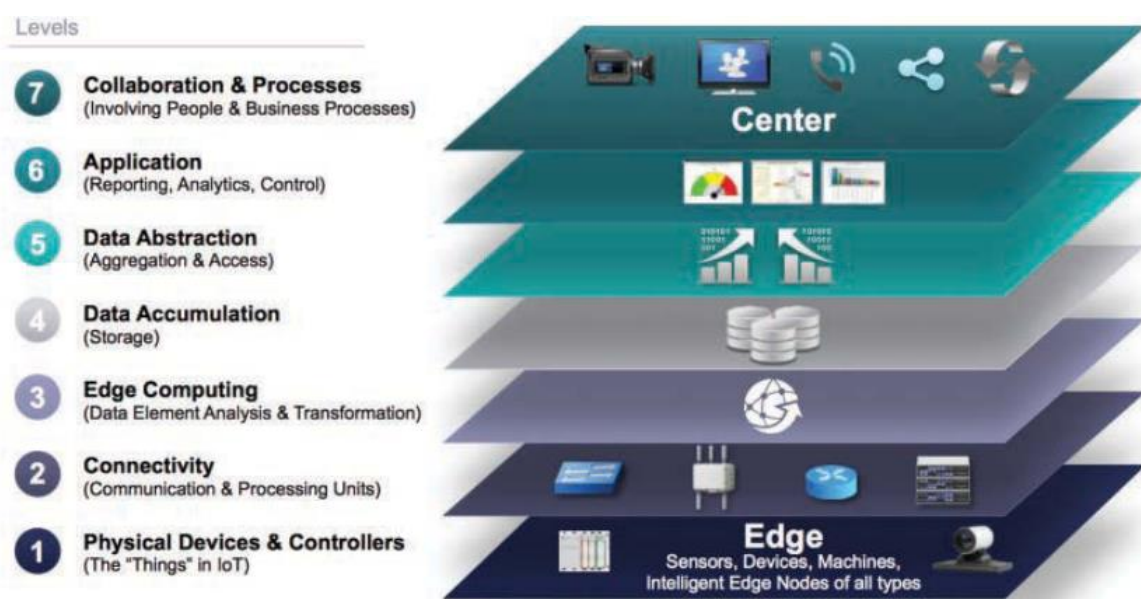


Figura 24. IoT modelo de referencia publicado por IoT World Forum

Fuente: IoT Fundamentals (2017).

#### Capa 1: Elementos físicos y controladores

La primera capa del modelo de referencia de IoT son los dispositivos físicos y los controladores. Esta capa es el hogar de las "cosas" en Internet de las cosas, incluidos los diversos dispositivos terminales y sensores que envían y reciben información.

El tamaño de estas "cosas" puede variar desde sensores casi microscópicos hasta máquinas gigantes en una fábrica. Su función principal es generar datos y ser capaz de ser consultados y/o controlados sobre una red. (Hanes et al., 2017, p. 36).

### Capa 2: Conectividad

La función más importante de esta capa de IoT es la transmisión confiable y oportuna de datos. Más específicamente, esto incluye transmisiones entre dispositivos de Capa 1 y la red y entre la red y el procesamiento de información que ocurre en la Capa 3. (Hanes et al., 2017, p. 36)

### Capa 3: Computación de borde

En esta capa, el énfasis está en la reducción de datos y la conversión de flujos de datos de red en información que está lista para ser almacenada y procesada por capas superiores. Cumple las funciones de: filtrar datos para reducir el tráfico procesado en el nivel superior, evaluar datos para generar alertas u otras acciones. (Hanes et al., 2017, p. 37).

### Capa 4: Almacenamiento de datos

“Esta capa se encarga de capturar los datos y almacenarlos para que las aplicaciones puedan usarlos cuando sea necesario. El procesamiento realizado parte de datos de eventos y datos de consultas”. (Hanes et al., 2017, pág. 38).

### Capa 5: Abstracción de información

“La capa 5 se encarga de confirmar que el conjunto de datos está completo y consolida los datos en un solo lugar o en varios almacenes de datos mediante la virtualización, además asegura una semántica consistente de varias fuentes”. (Hanes et al., 2017, p. 38)

### Capa 6: Aplicaciones

“Esta capa interpreta datos usando aplicaciones de *software*. Las aplicaciones pueden monitorear, controlar y proporcionar informes basados en el análisis de los datos recibidos” (Hanes et al., 2017, p. 38).

### Capa 7: Colaboración y procesos

En esta capa se consume y comparte la información obtenida de la aplicación. La colaboración y comunicación de la información de IoT con frecuencia requiere de varios pasos, y precisamente es lo que la vuelve útil. Es capaz de cambiar los procesos comerciales y ofrece beneficios para el control de la red IoT. (Hanes et al., 2017, p. 38).

### Componentes de una red IoT

Para poder comprender como se compone una red IoT y cada una de sus capas, se puede analizar por bloques donde se observan los componentes fundamentales que son comunes en las mayorías de los sistemas IoT. En la siguiente figura se muestra el diagrama de bloques.

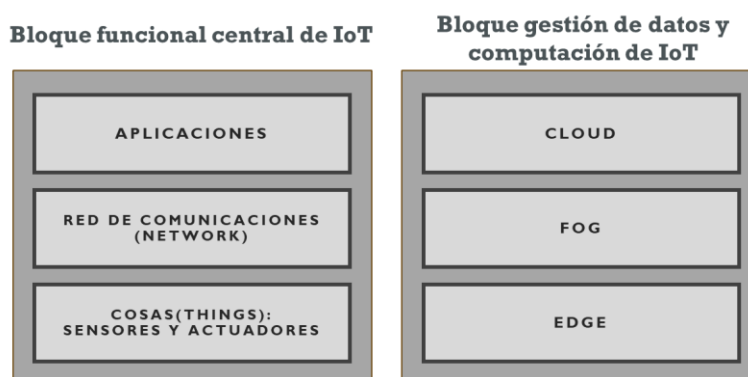


Figura 25. Modelo simplificado IoT.

Fuente: propia, basado en IoT de Hanes et al., 2017.

#### 1. Capa de cosas (Things en inglés)

Se refiere a todos los dispositivos físicos, los cuales deben adaptarse a las limitaciones del entorno en el que se implementan y al mismo tiempo ser capaces de proporcionar la información necesaria. En esta capa podemos encontrar todos los sensores que van a ser implementados en el sistema de monitoreo.

#### 2. Capa de red de comunicaciones (Network en inglés)

Se pueden observar los medios por los cuales los objetos inteligentes solventan la necesidad de comunicarse con un sistema externo. La tecnología de comunicación utilizada en muchas ocasiones es inalámbrica. Esta capa se divide en 4 subcapas:

### 2.1. Acceso a la subcapa de red

Respecto a la subcapa de red

es importante entender que existe una relación directa entre la tecnología de red IoT seleccionada y el tipo de topología de conectividad, tomando en cuenta que cada tecnología fue diseñada para ciertos casos como: que tipo de dispositivo se va a conectar, donde se va a conectar, cuantos datos va a transportar y en que intervalo y sobre que distancia. Para cada caso es importante considerar la banda de frecuencia más adecuada, la estructura de trama que coincida con el patrón de datos esperado (tamaño de paquete e intervalos de comunicación) y la topología. Un punto importante para determinar la tecnología de acceso a la red, es conocer el rango de la distancia entre el objeto inteligente (sensores) y el colector de información. (Hanes, et al., 2017, p. 48)

En el mundo de IoT de acuerdo con las distancias de transmisión para las tecnologías de acceso se pueden encontrar:

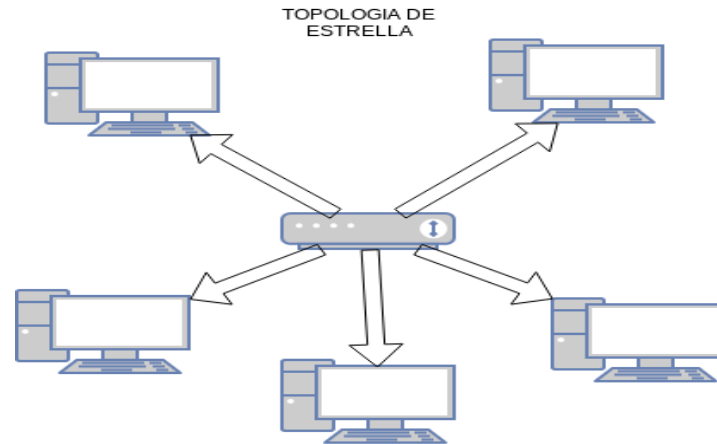
- PAN: Red de área personal (Personal Area Network, por sus siglas en inglés), tiene una escala de pocos metros, utilizada para espacios personales. Un ejemplo de esta tecnología inalámbrica es el Bluetooth.
- HAN: Red de área doméstica (Home Area Network, por sus siglas en inglés), tiene una escala de unas decenas de metros, y un ejemplo de esta tecnología corresponde a ZigBee y Bluetooth Low Energy (BLE, por sus siglas en inglés).
- LAN: Red de área local (Local Aarea Network, por sus siglas en inglés), tiene una escala de hasta 100 metros. Está tecnología es común en las redes, y por lo tanto también en el espacio de IoT cuando se utilizan las tecnologías de red estándar como Ethernet o IEEE 802.11. (Hanes et al., 2017, p. 48-50)

Cada una de las tecnologías de acceso a la red, puede ofrecer una estructura de conectividad flexible para ampliar las posibilidades de comunicación, algunas de las topologías de conectividad son:

- Topología punto a punto: “esta topología permite comunicar un punto con otro punto” (Hanes, et al., 2017, p. 52). Implica que un único objeto puede comunicarse con la puerta de enlace (*gateway*, en inglés).
- Topología punto a multipunto: esta topología permite comunicar un punto con varios otros puntos. En las redes IoT el uso de nodos (ejemplo, sensores) pueden admitir funciones de recopilación y reenvío de datos, mientras otros nodos (ejemplo, *gateways*) recopilan datos de los objetos inteligentes, a veces controlan al sensor para que realice operaciones específicas y al mismo tiempo interactúan con otras redes o *gateways*. (Hanes, et al., 2017, p. 52)

Un ejemplo de una topología punto a multipunto es la topología estrella, la cual utiliza una base central o controlador que permite las comunicaciones con los puntos finales.

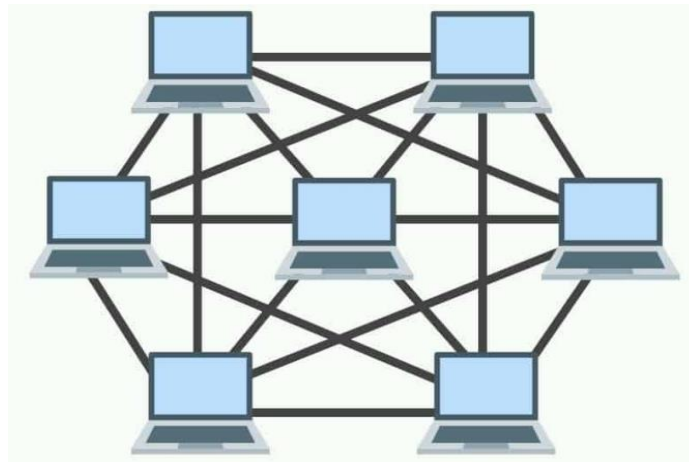
La topología estrella se puede implementar en las redes de sensores donde se utiliza un punto central que puede estar a cargo de toda la coordinación de la red, transmisiones y conexiones a cada sensor. Es decir, cada sensor está destinado únicamente a comunicarse con el coordinador en una relación llamada maestro/esclavo. Esta topología se puede evidenciar cuando se implementa una red de sensores para el monitoreo de la temperatura en una casa, donde cada sensor se comunica con un punto central donde se muestra y controla la temperatura. Cada sensor que se encuentra en distintas locaciones de la casa no necesita comunicarse con algún otro sensor para enviar la información. (Hanes, et al., 2017, p. 52)



*Figura 26. Topología estrella.*

*Fuente: Moreno, 2023*

Otro ejemplo de topología punto a multipunto es la topología en malla. En esta topología cada nodo tiene más de una ruta para comunicarse con otro, formando una malla. Esta redundancia significa que cada nodo puede comunicarse con más de un nodo para intercambiar información o ampliar el alcance del enlace de comunicación.



*Figura 27. Topología en malla.*

*Fuente: Bhatti, 2022*

## 2.2. Gateways y subcapa de Backhaul:

Toda información recolectada de los objetos inteligentes necesita ser enviada a una estación central donde se procese la información. En muchas ocasiones la estación puede estar en una ubicación distinta a la del objeto inteligente, por lo que los datos recibidos directamente del sensor a través de una tecnología de acceso deben reenviarse a otro medio (el *backhaul*) y transportarse a la estación central. El *gateway* es el encargado de esta comunicación intermedia. La elección del *backhaul* depende de la distancia y la cantidad de datos que deben reenviarse. Cuando el funcionamiento del objeto inteligente se controla desde un sitio local, con un entorno estable, (por ejemplo, como una fábrica) es posible utilizar Ethernet, mientras que si el uso de cables se vuelve poco práctico o inseguro, se pueden utilizar tecnologías inalámbricas como Wi-Fi. (Hanes et al., 2017, p. 55)

## 2.3. Subcapa de transporte de red

En las redes IoT, para lograr una estructura de comunicación, es necesario implementar un protocolo con características específicas. Este protocolo debe ser abierto y estar basado en estándares para adaptarse a múltiples industrias y múltiples medios. IP (internet protocol, por sus siglas en inglés) es un protocolo que permite que sea integrado en objetos de distintas naturalezas permitiendo el intercambio de información.

Los protocolos de la capa de transporte creados sobre IP (UDP y TCP) se aprovechan para decidir si la red debe controlar la entrega de paquetes de datos con TCP o si la tarea de control debe dejarse en manos de la aplicación (UDP). (Hanes et al., 2017, p. 58)

## 2.4. Subcapa de gestión de red IoT

Los protocolos IP, TCP y UDP brindan conectividad a las redes IoT. Los protocolos de capa superior deben encargarse de la transmisión de datos entre los objetos inteligentes y otros sistemas. Para la transmisión de datos es común observar el uso de HTTP, el cual tiene un componente de cliente y servidor, donde el sensor puede usar la parte del cliente para establecer una conexión con la aplicación central de IoT (el servidor), y luego poder intercambiar datos (Hanes et al., 2017, p. 58).

Otros protocolos derivados de la web sugeridos para el espacio IoT, pueden ser: WebSocket que proporciona una conexión bidireccional simple a través de una única conexión y MQTT (Message Queue Telemetry Transport, por sus siglas en inglés).

### 3. Capa de aplicaciones

En esta capa las aplicaciones son las que hacen uso de la información intercambiada con los objetos inteligentes. Una clasificación básica de las aplicaciones puede ser:

- Aplicaciones de análisis: este tipo de aplicación recopila los datos desde múltiples objetos inteligentes, procesa los datos y muestra la información resultante. Se pueden realizar informes históricos, estadísticas o tendencias. La aplicación es la que se encarga de mostrar los datos de la red, que no se pueden obtener mirando únicamente un único objeto inteligente. (Hanes et al., 2017, p. 59)
- Aplicaciones de control: “estas aplicaciones permiten controlar el comportamiento de los objetos inteligentes de la red. Un ejemplo de arquitectura de sistema de control es SCADA, el cual fue diseñado como un método universal para acceder a sistemas remotos y enviar instrucciones” (Hanes et al., 2017, p. 60).

Todas las capas definidas en el modelo por IoT World Forum, permite comprender mejor las fases y componentes que se requieren, para poder cumplir con una red del internet de las cosas. Cada capa cumple con una función específica que permitirá la obtención, interconexión, almacenamiento, análisis y presentación de la información, conformando así la red IoT.

### **Tecnologías de Acceso Inalámbrico**

Durante las últimas 3 décadas, el mundo ha visto cambios significativos en la industria de las telecomunicaciones. Ha habido algunos aspectos notables en el rápido crecimiento de las comunicaciones inalámbricas. Los sistemas inalámbricos constan de redes de área amplia (WWAN), como los sistemas celulares, redes de área local inalámbricas (WLAN) y redes inalámbricas de área personal (WPAN). (Garg, 2007, p. 1)

El Internet ha cambiado las expectativas sobre lo que significa el acceso a los datos, ya que la capacidad de recuperar información a través del internet ha incrementado la demanda de aplicaciones inalámbricas.

Con el uso de las tecnologías inalámbricas, el término de IoT ha cobrado relevancia, teniendo como objetivo la interconectividad de objetos y máquinas a una red informática como internet. Para formar una red de monitoreo por sensores, los dispositivos IoT deben estar conectados a la red para que se puedan utilizar sus datos. Dado que existen distintos protocolos para lograr la interconexión entre sí, es importante comprender características y criterios de comunicación que emplean los objetos inteligentes para su conectividad.

A continuación, se listan algunos criterios de comunicación:

- Rango: es importante tomar en cuenta condiciones como que tan largo se requiere propagar una señal, cual pretende ser el área de cobertura, si se utilizará en interiores o exteriores, etc. Los rangos se pueden clasificar como de corto, medio o largo alcance. En el primer tipo de corto alcance, es posible observar conexiones por cable que soportan decenas de metros de distancia o conexiones inalámbricas, un ejemplo claro de esta tecnología inalámbrica de corto alcance es IEEE 802.15.1 Bluetooth. Por otro lado, las tecnologías de acceso de medio alcance permiten conexiones a cientos de metros de distancia. Algunos ejemplos de tecnologías inalámbricas en este rango son; IEEE 802.11 Wi-Fi, IEEE 802.15.4, and 802.15.4g WPAN, mientras que en este mismo rango en tecnologías alámbricas están IEEE 802.3 Ethernet and IEEE 1901.2 Narrowband Power Line Communications (PLC). Finalmente, para tecnologías de accesos en el rango de largo alcance, permite la comunicación entre dos dispositivos en distancias que superan los 1600 metros, algunos ejemplos de tecnologías inalámbricas en este rango son: celulares (2G, 3G, 4G) y algunas aplicaciones exteriores de IEEE 802.11 Wi-Fi. (Hanes et al., 2017, p. 95-98)

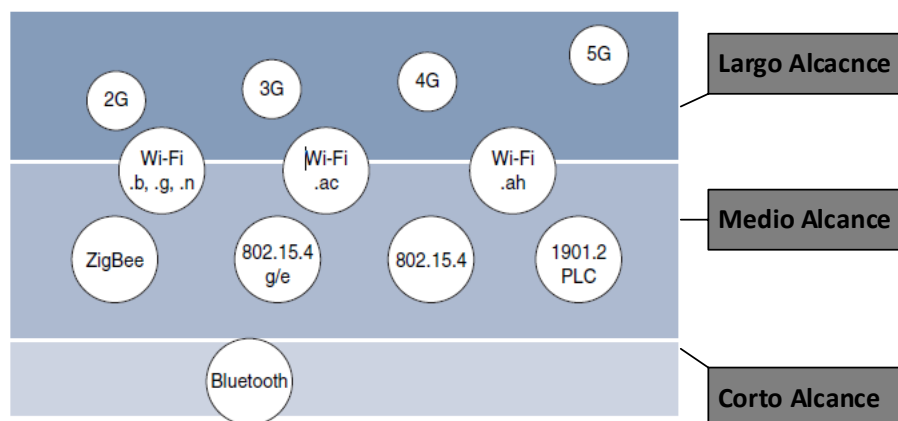


Figura 28. Rango de tecnologías de acceso inalámbrico.

Fuente: propia, basado en IoT de Hanes et al., 2017

- Banda de frecuencia: en el mundo de IoT, las bandas de frecuencia aprovechadas por las comunicaciones inalámbricas se dividen entre bandas con licencia y sin licencia.

La frecuencia sin licencia suele ser más sencilla de implementar que la frecuencia con licencia porque no requiere de un proveedor de servicios. Sin embargo, puede sufrir más interferencias porque otros dispositivos pueden estar compitiendo por la misma frecuencia en un área específica. La frecuencia de transmisión afecta directamente la forma en que se propaga una señal y su alcance máximo práctico. (Hanes et al., 2017, p. 98-101)

La frecuencia sub-GHz, es útil para aplicaciones en interiores o exteriores, permitiendo mayores distancias entre dispositivos, dado que tienen una mejor capacidad de penetrar infraestructuras de edificios o sortear obstáculos manteniendo la misma potencia de transmisión. Las bandas sub-GHz son utilizadas por protocolos como IEEE 802.15.4, 802.15.4g y 802.11ah.

- Consumo de energía: en el desarrollo de redes IoT, las tecnologías de acceso inalámbrico deben abordar las necesidades de bajo consumo de energía y conectividad para nodos alimentados por baterías, dado que no siempre se cuenta con una fuente de alimentación directa. “En muchas ocasiones el despliegue de

nodos alimentados está limitado por la disponibilidad de una fuente de energía, lo que hace que la movilidad sea más compleja”. (Hanes et al., 2017, p. 101)

- Topología: en las tecnologías de acceso, las principales topologías dominantes son: topología en estrella, topología uno a uno (peer-to-peer, por sus siglas en inglés) y topología en malla. Para tecnologías de largo y corto alcance, prevalece la topología en estrella. Mientras que para tecnologías de medio alcance se pueden usar las tres topologías dominantes. (Hanes et al., 2017, p. 102)

Tomado en cuenta los criterios de comunicación, es más fácil identificar cuál tipo de tecnología de acceso inalámbrico se adapta mejor a las condiciones del lugar en que se pretende interconectar dispositivos. Identificar las condiciones en las que va a operar el sistema, reducirá las probabilidades de que este se vea limitado por una mala elección de la tecnología a implementar.

### **Estándar IEEE 802.11**

Como parte de las tecnologías de acceso inalámbrico destaca la tecnología IEEE 802.11 Wi-Fi la cuál es sin duda la más exitosa tecnología inalámbrica desplegada. “Este estándar es clave para conectar puntos finales de nodos de computación, sensores con tasa de datos elevadas, dispositivos de análisis de audio y video, entre otros”. (Hanes et al., 2017, p. 130)

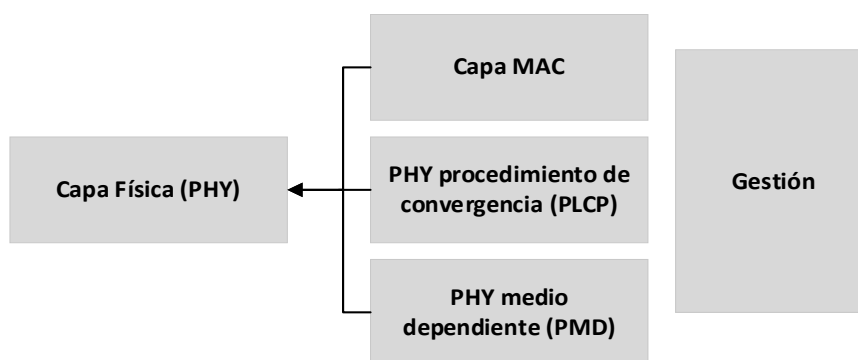
De acuerdo con Garg (2007), en 1997 el Instituto de Ingenieros Eléctricos y Electrónicos (IEE, por sus siglas en inglés) desarrolló un estándar internacional 802.11 centrándose en dos capas del modelo OSI, siendo la capa física (PHY, por sus siglas en inglés) y la capa de enlace de datos (DLL, por sus siglas en inglés). El principal objetivo del estándar IEEE 802.11 fue definir una subcapa de control de acceso medio (MAC, por sus siglas en inglés). Brindando protocolos y servicios de gestión de MAC y tres capas PHYs para la conectividad inalámbrica dentro de un área local. Las 3 capas físicas corresponden a IR (infrarrojo, por sus siglas en inglés), base de banda de infrarrojos, FHSS (espectro extendido por salto de frecuencias, por sus siglas en inglés) radio en la banda de 2.4GHz y DSSS (espectro extendido de secuencia directa, por sus siglas en

inglés) radio en la banda de 2.4GHz. Siendo capaces de admitir las 3 capas operaciones de 1 a 2 Mbps.

### **Funciones de las capas físicas (PHY) del estándar IEEE 802.11**

Como se mencionó anteriormente las 3 capas físicas corresponde a Infrarrojo (banda base), DSSS y FHSS. De acuerdo con Garg (2007), estas 3 capas proveen 3 niveles de funcionalidad.

1. Intercambio de tramas entre la capa MAC y la capa física PHY, bajo el control del procedimiento de convergencia de capa física (PLCP, por sus siglas en inglés).
2. Transmitir trama de datos sobre el medio, bajo el control de la subcapa PMD.
3. Proporcionar una indicación de detección de portadora a la capa MAC para verificar la actividad en el medio (p. 723-724).



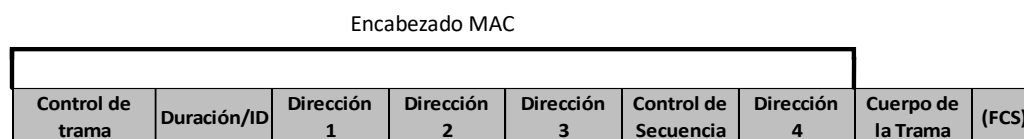
*Figura 29. Modelo OSI para IEEE 802.11*

*Fuente: propia, basado en Garg, 2007*

### **Capa de enlace de datos (DLL) y subcapa MAC del estándar IEEE 802.11**

La capa de enlace de datos consiste en 2 subcapas: la subcapa de control de enlace lógico (LLC, por sus siglas en inglés) y la subcapa de control de acceso medio, MAC. La subcapa LLC inserta ciertos campos en la trama, como la dirección de origen y la dirección de destino en la cabecera de la trama. Las redes locales

inalámbricas (WLAN, por sus siglas en inglés) utilizan un medio de transmisión compartido de alta velocidad al que todos los dispositivos están conectados y se transmiten tramas de información. La subcapa MAC define como un usuario obtiene un canal cuando lo necesita. La trama MAC contiene información de direccionamiento, información para establecer el vector de asignación de red y una secuencia de comprobación de la trama para verificar la integridad de este. En la siguiente figura se muestra el formato de la trama MAC. (Garg, 2007, p. 743-744)



*Figura 30. Formato de la trama MAC*

*Fuente: propia, basado en Garg, 2007.*

Cada espacio de la trama contiene:

- Dirección de transmisión: es la dirección de MAC que transmitió la trama en el medio inalámbrico, siendo siempre una dirección individual. Las estaciones utilizan la dirección de transmisión para identificar la estación a la cual se enviarán las respuestas.
- Dirección del receptor (RA, por sus siglas en inglés): es la dirección de MAC a la cual se envía la trama a través del medio inalámbrico.
- Dirección de origen (SA, por sus siglas en inglés): es la dirección MAC que originó la trama. Se utiliza para identificar la fuente de una trama.
- Dirección de destino (DA, por sus siglas en inglés): corresponde a la dirección de destino a la cual se envía la trama.

- Control de secuencia: es un espacio de 16 bit, este espacio de la trama se utiliza para que una estación receptora elimine tramas recibidas duplicadas.
- Cuerpo de la trama: contiene la información específica para la trama de datos. Este espacio tiene una longitud variable y puede alcanzar como máximo los 2034 bytes.
- Secuencia de comprobación de trama (FCS, por sus siglas en inglés): tiene una longitud de 32 bits y contiene el resultado de aplicar el polinomio C-32 al encabezado MAC y al cuerpo de la trama. Este espacio se utiliza para garantizar que la trama de datos no se haya corrompido durante la transmisión.

La tecnología de acceso inalámbrico WiFi cada día continúa evolucionando, logrando la mejora en aspectos como la velocidad y rango de transmisión de datos, aumentando la capacidad de dispositivos conectados y conexiones más eficientes en el medio.

## **Bases de datos**

### **Historia de aplicaciones de base datos**

Muchas de las primeras aplicaciones de las bases de datos mantenían registros en grandes organizaciones como corporaciones, universidades, hospitales y bancos. En muchas de estas aplicaciones había un gran número de registros de estructura similar. Uno de los principales problemas de los primeros sistemas de base de datos fue la mezcla de conceptos relacionados con el almacenamiento físico y la colocación de registros en el disco.

Las bases de datos relacionales fueron propuestas originalmente para separar el almacenamiento físico de su representación conceptual y además proporcionar una base matemática para la representación y consulta de datos. El modelo de datos relacionales también introdujo lenguajes de consulta de alto nivel que proporcionaron una alternativa a las interfaces de los lenguajes de programación, haciendo que escribir nuevas consultas fuera mucho más rápido.

Los primeros sistemas relacionales experimentales se desarrollaron a finales de la década de 1970. Con el desarrollo de nuevas técnicas de almacenamiento y un mejor procesamiento y optimización de consultas su desempeño mejoró de tal manera que hasta el día de hoy se encuentran desde computadoras personales hasta grandes servidores. [...]

Con el uso del internet, los usuarios hoy pueden crear páginas web estáticas que utilizan un lenguaje de publicación web como Hyper-Text Markup Language (HTML) y almacenar estos documentos en servidores web donde otros usuarios (clientes) pueden acceder a ellos. Gran parte de la información crítica de las páginas web, son datos extraídos dinámicamente de los DBMS. (Elmasri y Navathe, p. 24-25)

Gracias al éxito de las bases de datos, se han podido adaptar a aplicaciones como:

- Almacenamiento de datos provenientes de experimentos en el área científica
- Almacenamiento de todo tipo de imágenes, videos y películas
- Análisis de datos en áreas financieras para la búsqueda de patrones
- Almacenamiento de datos obtenidos que cambian con el tiempo, como datos diarios, etc.

### **Aplicación en sistemas de monitoreo**

El principal objetivo de los sistemas de monitoreo por sensores es la obtención de datos correspondientes a mediciones de fenómenos en un medio. Los resultados obtenidos en las mediciones muchas veces requieren ser almacenados para llevar un registro, realizar estudios de análisis de datos para cálculos estadísticos o análisis de tendencias, etc. En el ámbito de la computación la tecnología de las bases de datos hace esto posible. De acuerdo con Elmasri y Navathe (2016) “se puede definir de manera general una base de datos como una colección de datos relacionados, que provienen de fuentes con cierto grado de interacción con eventos del mundo real y una audiencia que está activamente interesada en su contenido” (p.5).

Para definir una base de datos, se debe especificar los tipos de datos, las estructuras y las restricciones de los datos que se almacenarán en la base de datos. “Los diseñadores de las bases de datos son responsables de identificar la data que va a ser almacenada en la base de datos y de

escoger la estructura adecuada para representar la data, entendiendo los requerimientos de los usuarios”. (Elmasri y Navathe, p. 15)

En una base de datos se tiene como propiedades implícitas:

- Representar un aspecto de la vida real llamado como *miniworld*. Los cambios que hay en el *miniworld*, se deben ver reflejados en la base de datos.
- Una mezcla aleatoria de datos no puede ser correctamente denominada como una base de datos.
- Una base de datos está diseñada, construida y poblada con datos para un propósito específico. Tiene un grupo previsto de usuarios y algunas aplicaciones preconcebidas en las cuales los usuarios están interesados.

En otras palabras, la base de datos tiene una fuente de la cual se derivan los datos, cierto grado de interacción con eventos del mundo real y una audiencia que está activamente interesada en su contenido. (Elmasri y Navathe, p. 5)

### **Sistema de administración de base de datos (DBMS)**

Una base de datos computarizada puede ser creada y mantenida por un grupo de programas de aplicación escritos específicamente para esa tarea o por un sistema de administración de base de datos (DBMS, por sus siglas en inglés). Un DBMS es un sistema computarizado que permite a los usuarios crear y mantener una base de datos.

Definir una base de datos como se mencionó anteriormente involucra especificar el tipo de datos, estructura y restricciones de los datos que van a ser almacenados en esta. La definición de la base de datos se almacena por el DBMS y es llamado como meta-data. El proceso de construir una base de datos consiste en almacenar datos por un medio controlado por el DBMS, al igual que el proceso de manipulación consiste en consultar a la base de datos para recuperar datos

específicos, actualizar datos o generar informes a partir de la información almacenada. [...]

Los programas de aplicaciones acceden a la base de datos enviando consultas o solicitudes de datos al DBMS. Se puede definir una consulta, como la acción de recuperar algunos datos de la base, y una transacción como provocar que algunos datos se lean y otros se escriban en la base de datos (Elmasri y Navathe, pp. 5- 6).

Los sistemas de administración de base de datos (DBMS), desempeñan un papel fundamental, al ser un medio eficiente y organizado para almacenar, gestionar y recuperar datos. En la actualidad sobresalen como DBMS; MySQL, Microsoft SQL Server, Oracle Database, SQLite, entre otros.

Los sistemas DBMS, muchas veces poseen interfaces que pueden incluir: interfaces para clientes web o navegación, aplicaciones para dispositivos móviles que permiten acceder a la información desde un celular, interfaces gráficas para el usuario, gráficos de los datos almacenados, entre otros.

### **Modelo de datos relacionales**

El modelo de datos relacional es uno de los modelos más utilizados en los DBMS del comercio y fuentes de código abierto.

Este modelo representa una base de datos como un conjunto de relaciones, donde cada relación forma una tabla de valores, cada registro tiene una estructura lineal. En una tabla de valores, cada fila representa una recopilación de valores provenientes del mundo real. En este modelo se asigna un nombre a la tabla, y los nombres de las columnas se utilizan para ayudar a interpretar el significado de los valores en cada fila. (Elmasri y Navathe, p. 150)

# de Registro	Temperatura (°C)	Humedad (%)	Presión (hPa)	Partículas en el aire (µg/m <sup>3</sup> )
1	25	70%	900	50
2	24	66%	902	65
3	22	64%	901	80

*Tabla 7. Ejemplo de base de datos relacionales.*

*Fuente: Propia.*

En la tabla anterior es posible apreciar un ejemplo de una base de datos relacionales, donde cada fila representa un registro de variables obtenidos del mundo real. En una tabla relacional, existen dominios con un nombre, por ejemplo; según la tabla 7, un dominio es la “Temperatura”, a cada dominio se le debe especificar el tipo de dato, en este caso para “Temperatura” el tipo de dato podría ser “Integer”, que se utiliza para tipos de datos enteros. Si se tratara de una variable como la fecha en que se obtuvo el dato, el tipo de dato podría ser “String” que se utiliza para almacenar caracteres como texto y su formato se podría ajustar a que solo acepte cierta cantidad de caracteres, dado que una fecha no puede tener una cantidad no definida de caracteres. Por lo tanto, un dominio deber recibir un nombre, tipo de dato y formato, en una base de datos relacional.

## **Lenguaje SQL**

El lenguaje SQL es uno de los estándares para las bases de datos relacionales. Las siglas SQL corresponde al idioma en inglés como Structured Query Language, el cual se traduce como lenguaje de consulta estructurado, el cual fue diseñado e implementado por IBM Research como la interfaz para un experimento de base de datos relacional llamado SYSTEM R. La estandarización de SQL fue un esfuerzo en conjunto con el Instituto Nacional Americano de Estándares (ANSI, por sus siglas en inglés) y la Organización Internacional de Estándares (ISO, por sus siglas en inglés). [...]

Se puede definir SQL como un lenguaje de base de datos integral, que tiene declaraciones para definiciones de datos, consultas y actualizaciones. Además, tiene funciones para definir vistas en la base de datos, para especificar seguridad y autorización, para definir restricciones de integridad y para especificar controles de transacciones. También tiene reglas para incorporar declaraciones SQL en un lenguaje de programación de propósito general como Java o C/C++. (Elmasri y Navathe, p. 178-179)

Este lenguaje tiene sus propios comandos para crear esquemas y tablas, inserción, eliminación y actualización de datos. Los tipos de datos de atributos y dominios en SQL corresponden a: datos numéricos, cadenas de caracteres, datos booleanos, fechas y tiempo.

### **Base de datos web utilizando PHP**

Muchas aplicaciones de internet que proporcionan interfaces web para acceder a información almacenada en una o más bases de datos utilizan lenguajes de secuencias de comandos. Estos se utilizan a menudo para generar documentos HTML, que luego el navegador web muestra para interactuar con el usuario.

PHP originalmente significaba “Personal Home Page” que en español sería, página de inicio personal. Hoy en día significa “PHP Hypertext Processor”, procesador de hipertexto. “PHP de código abierto, es un intérprete que proporciona un preprocesador de hipertexto, que ejecutará comandos PHP en un archivo de texto y creará el archivo HTML deseado” (Elmasri y Navathe, p. 343-344).

El motor de interpretación de PHP está escrito en el lenguaje de programación C, por lo que puede usarse en casi todos los tipos de computadoras y sistemas operativos como Windows, Linux o Mac OS. Además, PHP tiene bibliotecas de funciones para acceder a bases de datos almacenadas en varios tipos de sistemas de datos relacionales como Oracle, MySQL, SQLServer.

### **Aplicaciones en sistemas de monitoreo con la ayuda de bases de datos**

Los sistemas de monitoreo se ven muy beneficiados con el uso de bases de datos, ya que permite almacenar, gestionar y analizar grandes cantidades de datos adquiridos del medio con el uso de sensores y dispositivos de medición. Entre las aplicaciones se puede encontrar:

- Uso de bases de datos para el almacenamiento de los datos obtenidos del monitoreo ambiental de un cuarto limpio, con datos temporales, históricos y en tiempo real para un análisis continuo.
- Uso de bases de datos para almacenar información correspondiente a mediciones o registros de consumo eléctrico, para evaluar la eficiencia y optimización de recursos.

- Uso de bases de datos para almacenar notificaciones y datos de alarmas del sistema de monitoreo, para mejorar las respuestas a acciones correctivas y análisis de eventos que se presentan en el sistema, que puedan ayudar predecir análisis de modos de falla.

En términos generales, los sistemas de monitoreo buscan tener un complemento con las bases de datos, para así hacer de los sistemas de monitoreo una fuente valiosa de información que pueda ser almacenada, actualizada, y conservada a lo largo del tiempo, asegurando que la información sea accesible en cualquier momento, y facilite el análisis de los datos recopilados.

### **Sistema de visualización**

En una red IoT se pueden utilizar múltiples aplicaciones para incrementar su eficiencia.

Cada aplicación puede recolectar los datos de manera que proporciona una gama de funciones basada en el análisis de los datos. Las aplicaciones de análisis pueden recopilar datos de múltiples dispositivos inteligentes, procesar los datos y brindar una visualización de estos como informes históricos, estadísticas o tendencias de los datos. (Hanes et al., 2017, p. 59)

El análisis de datos es fundamental para los sistemas de monitoreo, dado que se procesan los datos obtenidos para proveer una vista inteligente del sistema. Los niveles de temperatura, humedad, presión, partículas suspendidas en el aire recolectados de cientos de sensores pueden ser procesados de manera que sea posible representarse en un *dashboard* o tablero para sí tener indicadores claves del desempeño del cuarto limpio.

Un ejemplo de un sistema de visualización corresponde a la plataforma Grafana, la cual está diseñada para la monitorización y la visualización de datos en tiempo real. Facilita la creación y uso de paneles interactivos, que reflejan las métricas y análisis de datos obtenidos de una base de datos.



*Figura 31. Ejemplo de Dashboard en Grafana.*

*Fuente: LoRa 2024.*

## CAPÍTULO III: MARCO METODOLÓGICO

### Enfoque de la investigación

De acuerdo con Hernández, Fernández y Baptista (2014), en el libro de *Metodología de la investigación*, desglosa los tipos de enfoques de la investigación como: cuantitativo, mixto y cualitativo, cada uno con sus características y proceso.

El enfoque cuantitativo representa un conjunto de procesos secuenciales y probatorios. A partir de objetivos y preguntas de investigación, se revisa la literatura y se construye un marco o una perspectiva teórica, donde se determinan variables, trazando un plan para probarlas (diseño). Una de las características del enfoque de investigación cuantitativo, es que debe ser objetiva. Los fenómenos que se observan o miden no deben ser afectados por el investigador, en este enfoque se intenta generalizar los resultados encontrados en un grupo a una colectividad mayor, buscando que los estudios efectuados puedan ser replicados. (Hernandez et al., 2014, p. 4)

Como parte de este proyecto se utilizará el enfoque cuantitativo, el cual permitirá una aproximación al estudio de variables físicas medidas en cuartos de ambiente controlado, tomando en cuenta la estructura del sistema para poder llevar a cabo un monitoreo de conteos y magnitudes.

### Fuentes de información

El propósito de esta investigación es la propuesta de un sistema de monitoreo de variables para cuartos de ambientes controlado. Para lograr esto se debe obtener información sobre los dispositivos actuales que pueden ser capaces de cumplir con las funciones de monitoreo de las variables de una forma acertada, que puedan ser adaptados a las normativas y características de un ambiente controlado y que a su vez sean compatibles entre sus tecnologías para formar una red IoT. Para obtener esta información se hizo una exhaustiva búsqueda en el mercado de dispositivos y *software* actuales compatibles con la plataforma de Arduino, revisando fichas técnicas de fabricantes y desarrolladores de *software*. Además de un análisis de la normativa y regulaciones existentes en los sistemas de monitoreo de cuartos limpios, que facilitó la comprensión de los requerimientos mínimos con los que debe contar el sistema de monitoreo.

## Unidades de Análisis

Matriz de conceptualización: investigaciones con enfoque cuantitativo.

Objetivo	Variable	Indicador	Definición conceptual	Definición operacional	Definición instrumental
Selección de sensores y microcontroladores para el sistema de monitoreo.	Modelos y fabricantes seleccionados.	Los sensores y microcontroladores son compatibles con la plataforma Arduino y cumplen con las características requeridas para el sistema.	Características estáticas y dinámicas de los sensores, además de características operativas de los microcontroladores.	Consultar las características técnicas de los sensores y microcontroladores para comparar sus características.	Consulta de fichas técnicas brindadas por el proveedor.
Programación de sensores utilizando la plataforma Arduino.	Disponibilidad física de <i>hardware</i> y <i>software</i> .	Interconexión de los sensores con las placas microcontroladoras.	Programación que permita la conexión de sensores al microcontrolador.	Elaboración de código para la interconexión de los sensores al microcontrolador mediante la programación del Arduino.	Uso del microcontrolador Arduino, sensores de: Humedad relativa, temperatura, presión y cantidad de partículas en el aire.
Determinar la topología de conectividad para la red de sensores.	Tipos de dispositivos conectados, localidad, distancia de comunicación.	Topología de red que permita el acceso a la red para la transmisión de datos.	Tecnologías de acceso a la red.	Elaborar esquema de Topología que permita el acceso a la red para la transmisión de datos.	Tecnologías de acceso a la red.
Determinar el protocolo de comunicación y servidor a utilizar para la red IoT.	Protocolo de comunicación y servidor utilizado.	Comunicación entre la placa microcontroladora y el servidor.	Protocolo de comunicación encargado de enviar los datos del microcontrolador al servidor.	Protocolo de comunicación que envíe los datos obtenidos por los sensores.	Código de programación en Arduino.

<b>Objetivo</b>	<b>Variable</b>	<b>Indicador</b>	<b>Definición conceptual</b>	<b>Definición operacional</b>	<b>Definición instrumental</b>
Determinar el sistema de almacenamiento de los datos obtenidos por los sensores.	Capacidad de almacenar los datos.	Los datos obtenidos por los sensores se almacenan de manera correcta.	Sistema de almacenamiento de datos capaz de llevar registro de datos obtenidos.	Diseño y programación de un sistema de almacenamiento.	Tipo de sistema de almacenamiento.
Sistema de visualización de los datos obtenidos.	Capacidad de visualizar los datos obtenidos.	Se proyectan de manera correcta los datos reales obtenidos por el sistema.	Interfaz capaz de mostrar los datos contenidos en la base de datos.	Desarrollo de interfaz que facilite la visualización en tiempo real.	Plataforma que despliegue la interfaz.
Análisis financiero	Costos de implementación.	Evaluación Beneficio Costo B/C.	Cálculo de la evaluación Beneficio Costo B/C con el fin de analizar respecto al costo del sistema de monitoreo, si es lo suficientemente confiable.	Comparar los costos de la implementación del sistema de monitoreo con respecto al beneficio que ofrece el sistema.	Cotizaciones de componentes utilizados en el sistema, ficha técnica de componentes. Datos de costo total y beneficios obtenidos.

*Tabla 8. Matriz de Conceptualización, Investigación de Enfoque Cuantitativo.*

*Fuente: Propia.*

## CAPÍTULO IV: ANÁLISIS DE RESULTADOS

En el presente capítulo se desarrollará el diseño del sistema de monitoreo que como según se explicó anteriormente pretende facilitar una solución alcanzable y fácil de producir por cualquier empresa de manufactura de dispositivos médicos que desarrolle sus procesos productivos en ambientes controlados, como lo son los cuartos limpios, en el que el monitoreo de las condiciones ambientales del cuarto es de suma importancia para garantizar la calidad del producto.

Entre las principales deficiencias o puntos de mejora que busca dar solución este proyecto se encuentran:

- Instalaciones en donde un único *display* colocado a las afueras del cuarto limpio es quien indica la temperatura general en la que se encuentra operando. Este sistema de medición tiene las siguientes deficiencias:
  - a) En ciertas configuraciones de cuartos limpios es común encontrar diversas máquinas como soldadoras, moldeadoras, incluso hornos que generan altas temperaturas agrupadas en una misma sección. Esta concentración de equipos puede provocar un aumento significativo de la temperatura en esas áreas comprometiendo los límites permitidos. Además, si se trata de un cuarto que cuenta con un área amplia y solo dispone de un único indicador de temperatura general para todo el espacio, los usuarios no tienen forma de saber si se están excediendo los límites en esas secciones en específico.
  - b) Debido que el *display* solo muestra los niveles de temperatura, la gestión de registros de las temperaturas depende de la presencia de personal designado para registrar los datos. Esto plantea una limitación, ya que el registro de la temperatura solo se realiza si hay disponibilidad del personal designado. Además, no se tiene control sobre los momentos o condiciones en los que se tomó el registro, ya que generalmente se realiza un único registro al día.
- Las instalaciones que no cuentan con un sistema de monitoreo de partículas continuo cuentan con las siguientes deficiencias:

- a) En muchas ocasiones, las empresas cuentan con un dispositivo capaz de medir la cantidad de partículas suspendidas en el aire. Este dispositivo debe ser colocado por el personal encargado en cada uno de los puntos de muestreo que se identificaron según la norma ISO 14644-1:2015. El proceso de medición implica colocar el dispositivo medidor y luego registrar manualmente la medición que muestra el dispositivo, repitiendo este procedimiento en cada punto de muestreo, significando un consumo de tiempo importante en los casos donde el área del cuarto limpio es bastante amplia. Por lo general este proceso de medición se realiza únicamente una vez al mes lo que significa que las condiciones en las que operó el cuarto en los días sin medición quedan desconocidas. Además, que el registro de los datos no se realiza de una manera automatizada.

Con el fin de dar solución a todas estas situaciones y más que puedan presentarse en otras instalaciones, el presente diseño pretende resolver mediante la implementación de un sistema de monitoreo capaz de medir las variables ambientales de temperatura, humedad relativa, presión y cantidad de partículas suspendidas en el aire de una manera focaliza por cada punto de muestreo que se desee en el interior de cuarto, y que además los resultados de las mediciones queden registrados automáticamente en una base de datos, para que luego puedan ser consultados en cualquier momento, mediante una interfaz que muestre e interprete los resultados obtenidos por los sensores. Convirtiendo el sistema de monitoreo en una herramienta valiosa para conocer las condiciones ambientales bajo las que opera el cuarto durante los días del año.

Entre las principales ventajas de la implementación del presente diseño, se encuentran:

- Su facilidad de implementación es un punto clave, ya que la plataforma Arduino al ser una plataforma de código abierto, no representa ningún costo económico, además que por su popularidad cuenta con gran variedad de herramientas que facilitan su implementación con otros dispositivos, siendo este un punto muy importante en el mundo IoT. Por esta misma razón, hay una gran variedad de sensores y microcontroladores disponibles en el mercado y fáciles de adquirir debido a sus tamaños y precios.
- La capacidad de poder monitorear constantemente las condiciones ambientales del cuarto limpio, además de tener la posibilidad de colocar sensores en lugares específicos lo vuelve una herramienta muy versátil que permite el monitoreo por zonas de una manera constante.

- La posibilidad de obtener automáticamente los resultados de los sensores en una base de datos garantiza que se guarden los registros de las mediciones sin la dependencia de la disponibilidad de personal. Además, con la implementación de la base de datos se facilita la exportación de datos para análisis estadísticos que ayuden a la toma de decisiones o consulta de información histórica del cuarto.
- La implementación de una interfaz permite el monitoreo en tiempo real de las condiciones ambientales del cuarto, además que ayuda a la interpretación de los registros de las mediciones que se encuentran en la base de datos, ya sea de condiciones ambientales actuales o consulta de condiciones ambientales en fechas determinadas.

### **Análisis de componentes para el sistema de monitoreo**

En esta sección del presente proyecto se pretende listar las características de los dispositivos que podrían ser considerados para su implementación en el sistema de monitoreo. El objetivo es comparar los elementos listados y establecer una base técnica que respalde la propuesta del proyecto.

#### **Sensores considerados para el sistema de monitoreo**

Para determinar cuáles de los sensores disponibles en el mercado se adapta mejor a los objetivos del sistema de monitoreo del presente proyecto es necesario realizar una comparación donde se analice cada una de las características técnicas de los sensores, considerando aspectos como su facilidad de implementación en los puntos de muestreo y sus capacidades de integración en una red IoT, entre otros.

#### **Sensores de temperatura, humedad y presión**

En la siguiente tabla se listan los sensores capaces de medir temperatura, humedad y presión que se encuentran disponibles en el mercado, compatibles con la plataforma y tecnología de Arduino.

Nombre del sensor	Fabricante	Precio aproximado (\$)	Variable capaz de medir	Enlace del fabricante, suplidor
DHT11	AOSONG	5.95	Temperatura, humedad	<a href="https://www.crcibernetica.com/dht11-temperature-and-humidity-sensor-module/">https://www.crcibernetica.com/dht11-temperature-and-humidity-sensor-module/</a>
DHT21, Module AM2301	AOSONG	9.95	Temperatura, humedad	<a href="https://www.crcibernetica.com/dht21-digital-temperature-and-humidity-sensor-module/">https://www.crcibernetica.com/dht21-digital-temperature-and-humidity-sensor-module/</a>
SHT31-D	SENSIRION PCB: Adafruit	13.95	Temperatura, humedad	<a href="#">Adafruit Sensirion SHT31-D [Temperature &amp; Humidity Sensor] : ID 2857 : \$13.95 : Adafruit Industries, Unique &amp; fun DIY electronics and kits</a>
Si7021-A20	SILICON LABS PCB: Adafruit	9.95	Temperatura, humedad	<a href="https://www.adafruit.com/product/3251">https://www.adafruit.com/product/3251</a>
BMP280	BOSCH PCB: Adafruit	9.95	Presión Barométrica Temperatura	<a href="https://www.adafruit.com/product/2651">https://www.adafruit.com/product/2651</a>
BME280	BOSCH PCB: Adafruit	14.95	Presión Barométrica Temperatura Humedad	<a href="https://www.adafruit.com/product/2652">https://www.adafruit.com/product/2652</a>
BMP390	BOSCH PCB: Adafruit	10.95	Presión Barométrica Altitud	<a href="https://www.adafruit.com/product/4816">https://www.adafruit.com/product/4816</a>
LPS22HB	STMicroelectronics PCB: Adafruit	6.95	Presión Barométrica, Temperatura	<a href="https://www.adafruit.com/product/4633">https://www.adafruit.com/product/4633</a>

*Tabla 9. Sensores de temperatura, humedad y presión*

*Fuente: Propia.*

Con el fin de realizar un análisis más exhaustivo de las características de cada sensor listado en la tabla anterior se emplearán tablas comparativas que resalten las ventajas individuales de cada sensor con respecto a los demás. Dicha información de las tablas será obtenida directamente de las

fichas técnicas de cada sensor. Esto permitirá una evaluación más precisa y detallada, facilitando la identificación de las fortalezas específicas de cada sensor en comparación con sus contrapartes.

### **Análisis sensores de temperatura**

A continuación, se muestra la tabla comparativa de sensores capaces de medir la variable de temperatura utilizando los sensores listados en la tabla 9, tomando en cuenta características como rango de medición, precisión, resolución y repetibilidad.

<b>Sensores de temperatura °C</b>							
<b>Características</b>	<b>DHT11</b>	<b>DHT21</b>	<b>SHT31-D</b>	<b>Si7021-A20</b>	<b>BMP280</b>	<b>BME280</b>	<b>LPS22HB</b>
Rango de medición	0 a 50 °C	-40°C a 80 °C	-40°C a 125 °C	-40°C a 125 °C	-40°C a 85 °C	-40°C a 85 °C	-40°C a 85 °C
Precisión	± 2.0 °C	± 0.3 °C	±0.3 °C	± 0.3 °C	± 0,5 °C	± 0,5 °C	± 1.5 °C
Resolución	16 Bit, 0.1 °C	16 Bit, 0.1 °C	0,015 °C	14 Bit	0.01 °C	0.01 °C	16 Bit
Repetibilidad	± 0.2 °C	± 0.2 °C	0.04 °C	0.01 °C	No especifica	No especifica	No especifica

*Tabla 10. Comparación de características para la variable de temperatura.*

*Fuente: Propia.*

De acuerdo con la tabla 10, y después de evaluar los sensores que son capaces de medir la temperatura se observa que para la característica “rango de medición”, el sensor SHT31-D y Si7021-A20 cuentan con un mayor rango de operación que va desde los -40°C a 125 °C, lo cual presenta una ventaja respecto a los otros sensores, dado que con un amplio rango de operación permite utilizar el sensor en una variedad de entornos y situaciones.

Con respecto a la característica de “precisión”, el sensor SHT31-D y Si7021-A20 también cuentan con uno de los mejores rangos de precisión respecto a los demás sensores listados en la tabla 10. Con una precisión de ±0.3 °C es posible conocer la capacidad del sensor de proporcionar lecturas cercanas al valor real de la magnitud física.

Para la característica de “resolución”, esta se refiere a la cantidad más pequeña de cambio en la temperatura que el sensor puede detectar y registrar. Algunos fabricantes detallan la resolución del

sensor en términos de bits, lo cual se utiliza para sensores que producen señales analógicas y luego son convertidas a valores digitales, a mayor cantidad de bits resulta a una mayor precisión en la conversión analógica a digital. En cuanto a los sensores que expresan su resolución en bits, los sensores DHT11, LPS22HB y DHT21, tienen mejor resolución respecto al sensor Si7021-A20, además suponiendo que el sensor de 16 Bits de resolución LPS22HB tienen una resolución de 0.1 °C como indica el fabricante del sensor DHT21 y DHT11, de los sensores listados en la tabla 10, es posible observar que los sensores que cuentan con una mejor resolución de 0.01 °C son: BMP280 y BME280, seguidos por el sensor SHT31-D con una resolución de 0,015 °C. Es importante considerar que a diferencia del sensor SHT31-D estos operan en el rango de medición de -40°C a 85 °C.

Finalmente, para la característica de “repetibilidad”, según se explicó en capítulos anteriores, la repetibilidad mide la capacidad de un sensor para proporcionar resultados consistentes cuando se mide la misma magnitud en condiciones idénticas. Entre menor sea la variación de las medidas, mejor repetibilidad. Al analizar la tabla 10, se observa que el sensor Si7021-A20, cuenta con una capacidad de repetibilidad de 0.01 °C, demostrando la capacidad del sensor para proporcionar mediciones consistentes y reproducibles en condiciones similares. Seguido del sensor Si7021-A20, de acuerdo con la tabla 10, el segundo sensor con mejores características de repetibilidad corresponde al sensor SHT31-D con 0.04 °C.

Como parte de este análisis se puede concluir que el sensor SHT31-D, es el sensor que cuenta con mejores características, seguido por el sensor Si7021-A20, para la medición de la variable temperatura respecto a los sensores listados en la tabla 10.

### **Análisis sensores de humedad**

A continuación, se muestra la tabla comparativa para medir la variable de humedad utilizando los sensores listados en la tabla 9, tomando en cuenta características como rango de medición, precisión, resolución y repetibilidad.

<b>Humedad</b>					
<b>Características</b>	<b>DHT11</b>	<b>DHT21</b>	<b>SHT31-D</b>	<b>Si7021-A20</b>	<b>BME280</b>
Rango de medición	20% a 90%RH	0% a 99,9 %RH	0% a 100 %RH	0% a 100 %RH	0% a 100 %RH
Histéresis	<± 0.3% RH	No se especifica	a 25 °C ± 0,8% RH	± 1% RH	a 25 °C ± 1 %RH
Precisión	a 25 °C ± 5% RH	a 25 °C ± 3% RH	± 2% RH	± 2% RH	± 3% RH

Humedad					
Características	DHT11	DHT21	SHT31-D	Si7021-A20	BME280
Resolución o sensibilidad	16 Bit, 1% RH	0,1 % RH	0.01 %RH	12 Bit	0,008 %RH
Repetibilidad	± 1% RH	± 1% RH	0.15 %RH	0.025 %RH	No específica

Tabla 11. Comparación de características para la variable de humedad

Fuente: Propia.

Con respecto a la tabla 11, se muestran los sensores que son capaces de medir la variable de humedad. El rango de operación de estos sensores va desde 20% a 90%RH o incluso desde 0% a 100 %RH. Los sensores capaces de medir en el rango de 0% a 100 %RH, corresponden a: SHT31-D, Si7021-A20 y BME280.

Como parte de la característica de “histéresis”, la cual se explicó en capítulos anteriores, al ser una característica dinámica del sensor, representa la capacidad que tiene el sensor para seguir la curva de salida ideal, debido a los cambios en la salida de la variable física, a una mayor histéresis el sensor podría presentar una mayor discrepancia entre las lecturas. Por lo tanto, entre más baja sea la histéresis esto indica una mayor precisión en la respuesta del sensor a cambios en la humedad.

Después de comparar los sensores de la tabla 11, la histéresis de  $\pm 0,8\%$  RH del sensor SHT31-D es bastante precisa y puede ser suficiente para muchas aplicaciones de medición de humedad. Sin embargo, si se requiere una mayor precisión o se trabaja en un rango muy estrecho de humedad, una histéresis de  $\leq \pm 0.3\%$  RH del sensor DHT11 podría ser preferible, ya que proporciona una respuesta más precisa a los cambios en la humedad relativa.

Para la característica de “precisión” se observa que los sensores SHT31-D y Si7021-A20 son los que tienen mejores rangos de precisión respecto a los demás sensores listados en la tabla 11. Con una precisión de  $\pm 2\%$  RH, seguidos por una precisión de  $\pm 3\%$  RH del sensor BME280 y DHT21.

Con respecto a la característica de “resolución”, el sensor que tiene mayor resolución corresponde al sensor BME280 con una resolución de 0,008 %RH, seguido por el sensor SHT31-D con una resolución de 0.01 %RH.

Finalmente, para la característica de “repetibilidad” para la medición de la variable de humedad relativa, se observa que el sensor Si7021-A20, es el sensor que tiene mejor repetibilidad con un valor de 0.025 %RH, seguido por el sensor SHT31-D con un valor de 0.15 %RH.

Como parte del análisis se puede concluir que para medir la variable de humedad los sensores SHT31-D y Si7021-A20 poseen mejores combinaciones de características respecto a los demás sensores listados en la tabla 11.

### Análisis sensores de presión

A continuación, se muestra la tabla comparativa de sensores capaces de medir la variable de presión utilizando los listados en la tabla 9, tomando en cuenta características como rango de medición, precisión y resolución.

Presión								
Características	BMP280		BME280		BMP390		LPS22HB	
Rango de medición	300 hPa a 1100 hPa		300 hPa a 1100 hPa		300 hPa a 1250 hPa		260 hPa a 1260 hPa	
Precisión de presión absoluta	300 hPa a 1100 hPa	±1.0 hPa	Condiciones	Valor	Condiciones	Valor	260 a 1260 hPa	±0,1 hPa
			300 a 1100 hPa	±1,7 hPa	900 hPa	±0,33 hPa		
			-20 a 0°C		25°C	±0,50 hPa		
	0 a 65 °C		300 a 1100 hPa	±1,0 hPa	900 a 1100 hPa	±0,40 hPa		
			0 a 65°C		25 a 40°C	±0,50 hPa		
			1100 a 1250 hPa	±1,5 hPa	1100 a 1250 hPa	±0,50 hPa		
			25 a 40°C		0 a 65°C	±0,50 hPa		
Precisión de presión relativa	700 a 900 hPa	±0.12 hPa	700 a 900 hPa	±0.12 hPa	700 a 1100 hPa	±0.03 hPa	800 a 1100 hPa	±0.1 hPa
	25 a 40°C		25 a 40°C		25 a 40°C		25°C	
Resolución	0.0016 hPa		0.18 Pa ≈ 0.0018 hPa		0.016 Pa ≈ 0,00016 hPa		0.01 hPa	

Tabla 12. Comparación de características para la variable de presión.

Fuente: Propia.

Con respecto a la tabla 12, el rango de operación de los sensores listados se encuentra entre los valores de 300 hPa a 1100 hPa para los sensores BMP280 y BME280, 300 hPa a 1250 hPa para el sensor BMP390 y 260 hPa a 1260 hPa para el sensor LPS22HB. En el caso de los sensores listados, el sensor LPS22HB es quien tiene un rango de operación mayor lo cual permite que sea más flexible para su aplicación y adaptación a un sistema de monitoreo.

En el caso de los sensores de presión listados en la tabla 12, estos son capaces de medir la presión absoluta y la presión relativa, tal y como se explicó en capítulos anteriores. La diferencia entre presión absoluta y relativa radica en que la presión absoluta se refiere a la presión medida con respecto al cero absoluto, mientras que la presión relativa se refiere a la diferencia entre la presión medida y la presión de referencia que puede ser la presión atmosférica local.

De acuerdo a lo anterior y después de analizar los sensores, para la característica de “precisión” en la medición de presión absoluta se observa que los sensores cuentan con distintos niveles de precisión ya establecidos por el fabricante de acuerdo con las condiciones ambientales en las que se encuentren operando. En el caso de los sensores BME280, BMP390 y LPS22HB, son los que presentan mejores rangos de precisión para cada distinta condición de operación, volviéndose bastante versátiles para las distintas condiciones de presión que pueden presentarse en un cuarto de ambiente controlado.

El sensor LPS22HB mantiene un mismo rango de precisión a lo largo de las condiciones de operación de 260 a 1260 hPa con una precisión de  $\pm 0,1$  hPa, lo cual lo vuelve más estable para medir la presión absoluta dada su capacidad para producir mediciones consistentes y precisas bajo diversas condiciones ambientales y de operación.

Para la característica de precisión en la medición de presión relativa, el sensor BMP390 tiene una mejor precisión de  $\pm 0.03$  hPa en condiciones de medición de 700 a 1100 hPa. Seguido por el sensor LPS22HB, el cual en condiciones de medición de 800 a 1100 hPa tiene una precisión de  $\pm 0.1$  hPa para medir la presión relativa.

Finalmente, para la característica de “resolución”, el sensor que tiene mayor resolución corresponde al sensor BMP390 con una resolución de 0,00016 hPa, seguido por el sensor BMP280 con una resolución de 0.0016 hPa.

Como parte del análisis de los sensores listados en la tabla 12, se puede concluir que cada sensor presenta ventajas en distintas características y cada uno puede resultar mejor al otro

dependiendo de las características que se requieran para el sistema de monitoreo que se pretenda diseñar. Por ejemplo si lo que se desea es tener una mayor precisión y resolución en la medición de la presión relativa, se recomienda utilizar el sensor BMP390. Sin embargo, si lo que se desea es tener la posibilidad de un mayor rango de medición y una mayor estabilidad a lo largo del rango de medición en la precisión de la medición de la presión absoluta, se recomienda utilizar el sensor LPS22HB.

Para el presente proyecto, la presión que se desea monitorear en el cuarto de ambiente controlado corresponde a la presión diferencial, la cual se obtiene de comparar la presión del interior del cuarto con otras secciones del cuarto, como por ejemplo la presión diferencial del cuarto de producción respecto a la sección de empaque, o la presión diferencial del cuarto de producción respecto a la sección de vestimenta para ingreso al cuarto, entre otras. En general, para obtener la presión diferencial, se requiere que cada sensor obtenga la presión relativa, por esta razón, de los sensores listados en la tabla 12 el sensor con mejores características para medir la presión relativa en el interior del cuarto controlado corresponde al sensor BMP390.

### **Análisis características eléctricas de sensores de temperatura, humedad y presión**

Después de analizar las características que describen las capacidades de los sensores para medir las variables para las que fueron diseñados, en este caso variables de temperatura, humedad relativa y presión. Se observó que los sensores que presentan mejores características para el sistema de monitoreo del presente proyecto corresponden a los sensores: SHT31-D y BMP390.

Como parte del análisis de las características eléctricas de los sensores se tomará en cuenta el voltaje de alimentación y consumo de corriente con el fin de determinar la potencia que requieren para su funcionamiento, además de los protocolos de comunicación que utilizan para transmitir los datos obtenidos de las mediciones.

<b>Características eléctricas</b>				
<b>Características</b>	<b>SHT31-D</b>		<b>BMP390</b>	
Voltaje de alimentación	2,4 V a 5,5 V		1,65 V a 3,6 V	
Consumo de corriente	<b>Condiciones</b>	<b>Valor Promedio</b>	<b>Condiciones</b>	<b>Valor Promedio</b>
	Durante medición	800 $\mu$ A	Durante medición	660 $\mu$ A

Características eléctricas				
Características	SHT31-D		BMP390	
	Promedio	2 $\mu$ A	Promedio	3,2 $\mu$ A
	Modo reposo	0,2 $\mu$ A	Modo reposo	1,4 $\mu$ A
Interfaz de comunicación	1 MHz	I2C	3,4 MHz	I2C
			10 MHz	SPI

Tabla 13. Resumen de características eléctricas de los sensores de temperatura, humedad y presión.

Fuente: Propia.

Dentro de las especificaciones eléctricas de los sensores, es notable que cada uno está diseñado para funcionar dentro de un rango de voltaje específico, como indican las recomendaciones del fabricante, con el fin de asegurar su correcto funcionamiento. En el caso del sensor SHT31-D, este admite un voltaje de alimentación de 2.4V a 5.5V, mientras que el sensor BMP390 admite un rango de voltaje de alimentación de 1.65 a 3.6 V.

Con respecto a las corrientes de operación, cada sensor tiene distintos niveles de consumo, dependiendo de las condiciones en las que se encuentre. Por lo general, el máximo pico de corriente se registra al iniciar los procesos de medición en cada sensor, siendo este el momento de consumo máximo. Los fabricantes también incorporan funciones para optimizar el consumo de corriente cuando el sensor está conectado a la fuente de alimentación y no es requerido realizar ninguna medición. Esta opción, conocida como modo reposo, puede ser configurada en cada sensor según la periodicidad requerida en las mediciones.

Además, los fabricantes proporcionan información sobre el consumo promedio del sensor, que representa la cantidad de corriente eléctrica esperada en condiciones normales de funcionamiento. Este valor es importante ya que ayuda al usuario a calcular el costo de energía y planificar sus periodos de funcionamiento.

Para el cálculo del consumo eléctrico de cada sensor es importante tomar en cuenta el voltaje de alimentación y la corriente promedio que requiere el sensor para tomar las mediciones. Utilizando la ecuación de la Ley de Ohm, la cual establece que la potencia es igual al producto de la corriente eléctrica y la diferencia de potencial, es posible determinar el consumo de cada sensor.

$$P = V \times I \quad (5)$$

Para el cálculo del consumo eléctrico de cada sensor listado en la tabla anterior, se tomó en cuenta utilizar los valores superiores de los rangos de voltaje permitidos en el que puede operar cada

sensor para determinar su consumo máximo, obviando los picos de consumo de corriente eléctrica y tomando el valor promedio de corriente establecido por el fabricante. En la siguiente tabla se resumen los consumos eléctricos para cada sensor:

Consumo eléctrico		
Características	SHT31-D	BMP390
Cálculo	$P = 5.0 \times 2 \times 10^{-6}$	$P = 3.3 \times 3.2 \times 10^{-6}$
Potencia (W)	$P = 0.00001$	$P = 0.00001056$

Tabla 14. Cálculo consumo eléctrico sensores de temperatura, humedad y presión

Fuente: Propia.

Para el sistema de monitoreo, al ser necesario el uso de los sensores SHT31-D y BMP390 para la medición de la temperatura, humedad y presión en los puntos de muestreo será indispensable tomar en cuenta el consumo eléctrico calculado en la tabla anterior para el diseño y cálculos de costos de implementación.

Con respecto a las interfaces de comunicación utilizadas por los sensores listados en la tabla 13, el sensor SHT31-D es capaz de transmitir sus datos utilizando el protocolo I2C, que según se explicó en capítulos anteriores, permite la comunicación serial entre dispositivos utilizando solo dos líneas de señal, conocidas como SDA y SCL, siendo una tecnología que permite reducir la cantidad de cables y permite la conexión de múltiples dispositivos a un bus I2C utilizando direcciones únicas asignadas a cada sensor, lo cual lo vuelve útil en sistemas que requieren de varios sensores.

En el caso del sensor SHT31-D, este cuenta con una velocidad de transmisión de 1 MHz, esta velocidad es considerablemente buena, ya que en aplicaciones que requieran una alta velocidad de procesamiento representa una ventaja.

Para el sensor BMP390, este cuenta con una velocidad de transmisión de datos mayor al sensor SHT31-D de 3.4 MHz utilizando el protocolo I2C y además cuenta con la capacidad de transmitir datos utilizando el protocolo SPI que a diferencia de I2C utiliza 4 líneas de señal para la comunicación, lo cual le permite adquirir velocidades de transmisión más altas de hasta 10 MHz y admitir un mayor número de dispositivos conectados en un mismo bus.

Para el presente diseño, las altas velocidades de transferencia de datos utilizando el protocolo I2C de ambos sensores representan una ventaja para el diseño del sistema ya que con poca cantidad de

cables utilizados para la transmisión de datos es posible alcanzar buenas velocidades. Es por esta razón que ambos sensores serán configurados para transmitir sus datos al microcontrolador utilizando el protocolo I2C.

### **Sensores de medición de concentración de partículas en el aire**

Después de realizar una exhaustiva búsqueda de los sensores disponibles en el mercado para medir la concentración de partículas en el aire, se identificaron como compatibles con la plataforma y tecnología Arduino los siguientes sensores listados en la tabla 15.

<b>Nombre del sensor</b>	<b>Fabricante</b>	<b>Precio aproximado (\$)</b>	<b>Variable capaz de medir</b>	<b>Enlace del fabricante, suplidor</b>
PMSA003I	PLANTOWER Placa: Adafruit	44.95	Concentración de partículas en el aire	<a href="https://www.adafruit.com/product/4632">https://www.adafruit.com/product/4632</a>
SDS011	Nova Fitness Co	39		<a href="https://www.amazon.com/-/es/detecci%C3%B3n-precisi%C3%B3n-Sensores-compatible-Arduino/dp/B0899V46SS">https://www.amazon.com/-/es/detecci%C3%B3n-precisi%C3%B3n-Sensores-compatible-Arduino/dp/B0899V46SS</a>
SPS30	SENSIRION	41.15		<a href="https://www.sensirion.com/products/catalog/SPS30/">https://www.sensirion.com/products/catalog/SPS30/</a>
HPMA115CO-004	Honeywell	81		<a href="https://www.mouser.com/ProductDetail/Honeywell/HPMA115CO-004?qs=PzGy0jfpSMv%252BXuZmF5yeTA%3D%3D">https://www.mouser.com/ProductDetail/Honeywell/HPMA115CO-004?qs=PzGy0jfpSMv%252BXuZmF5yeTA%3D%3D</a>
PMS9003M	PLATOWER	23		<a href="https://www.plantower.com/en/products_33/99.html">https://www.plantower.com/en/products_33/99.html</a>
IPS-7100	Piera Systems	71.10		<a href="https://pierasystems.com/intelligent-particle-sensors/">https://pierasystems.com/intelligent-particle-sensors/</a>

*Tabla 15. Sensores capaces de medir la concentración de partículas en el aire.*

*Fuente: Propia.*

Según se explicó anteriormente existen sensores capaces de medir el conteo de partículas o concentración de partículas en microgramos, lo cual representa la cantidad de materia PM para cada tamaño de partículas. La medida de concentración de partículas resulta útil en el monitoreo del aire ya que permite conocer la cantidad total de materia de partículas por cada tamaño permitiendo evaluar la presencia de contaminantes en el aire. Cabe recalcar que de los sensores listados en la tabla 15, algunos son capaces de medir únicamente la concentración de partículas en el aire y en las unidades de  $\mu\text{g}/\text{m}^3$ .

En el mercado también existen sensores capaces de medir el conteo de partículas totales en el aire para cada determinado tamaño de partícula para el que fue diseñado según el fabricante. Es importante destacar que de los sensores listados en la tabla 15, algunos son capaces de medir la cantidad de las partículas en el aire en las unidades de partículas/ $\text{cm}^3$  o partículas/litro.

Para el análisis de los sensores medidores de partículas en el aire estos se analizarán en tablas distintas con el fin de comparar las capacidades de los sensores por cada tipo:

- Sensores de concentración de partículas
- Sensores de conteo de partículas

#### Sensores de calidad de aire, medición en ( $\mu\text{g}/\text{m}^3$ )

Parámetro	PMSA003I		SDS011		HPMA115CO-004		PMS9003M		
	Índice	Unidad	Índice	Unidad	Índice	Unidad	Índice	Unidad	
Rango de medición de partículas	0.3~1.0	$\mu\text{m}$	PM 2.5	$\mu\text{m}$	1.0	$\mu\text{m}$	0.3~1.0	$\mu\text{m}$	
	1.0~2.5	$\mu\text{m}$			2.5	$\mu\text{m}$	1.0~2.5	$\mu\text{m}$	
	2.5~10	$\mu\text{m}$	PM 10	$\mu\text{m}$	4.0	$\mu\text{m}$	2.5~10	$\mu\text{m}$	
				10.0	$\mu\text{m}$				
Eficiencia/precisión de conteo de partículas	50% @0,3	$\mu\text{m}$	70% @0,3	$\mu\text{m}$	PM 2.5	$\pm 15\%$	$\mu\text{g}/\text{m}^3$	50% @0,3	$\mu\text{m}$
	98% @ $\geq 0,5$	$\mu\text{m}$	98% @ $\geq 0,5$	$\mu\text{m}$	PM 1.0, PM 4.0, PM 10	$\pm 25\%$	$\mu\text{g}/\text{m}^3$	98% @ $>0,5$	$\mu\text{m}$
Rango de medición	0~500	$\mu\text{g}/\text{m}^3$	0.0~999.9	$\mu\text{g}/\text{m}^3$	0~1000	$\mu\text{g}/\text{m}^3$	0~500	$\mu\text{g}/\text{m}^3$	
Resolución de partículas	1	$\mu\text{g}/\text{m}^3$	0,3	$\mu\text{g}/\text{m}^3$	1	$\mu\text{g}/\text{m}^3$	1	$\mu\text{g}/\text{m}^3$	

Tabla 16. Sensores de concentración de partículas en el aire, medición en ( $\mu\text{g}/\text{m}^3$ ).

*Fuente: Propia.*

Antes de realizar la comparación de sensores es importante entender que es lo que pretende evaluar cada característica del sensor determinada por el fabricante, dado que no todos utilizan o detallan los mismos parámetros. Con respecto a la tabla 16, cada característica corresponde a:

- Rango de medición de partícula: este término se refiere al rango de tamaño de partículas que el sensor es capaz de detectar y medir su concentración en masa.
- Eficiencia/precisión de conteo de partículas: se refiere a la capacidad del sensor para contar con precisión las partículas presentes en el aire.
- Rango de medición: se refiere al rango de concentraciones de partículas que el sensor puede medir de manera efectiva y precisa.
- Resolución de partículas: se refiere a la capacidad del sensor para distinguir pequeñas diferencias en la concentración de partículas.

Con respecto a la tabla 16, se puede observar que los sensores PMSA003I y PMS9003M tienen características idénticas en lo que respecta a las capacidades para medir la concentración total de masa de las partículas suspendidas en el aire, teniendo el mismo rango de medición de partículas, misma eficiencia de conteo para partículas de  $0.3 \mu\text{m}$  y mayores a  $0,5 \mu\text{m}$  y además tienen una resolución de  $1 \mu\text{g}/\text{m}^3$ .

El sensor HPMA115CO-004 a diferencia de los sensores PMSA003I y PMS9003M, este sensor es capaz de medir la concentración de partículas con tamaños a partir de los  $1.0 \mu\text{g}/\text{m}^3$  y en vez de utilizar porcentajes de eficiencia, brinda rangos de porcentaje de precisión para cada medida obtenida a cada tamaño de partícula, además que su rango de medición de  $0\sim 1000 \mu\text{g}/\text{m}^3$  lo vuelve más versátil.

El sensor SDS011 a diferencia de los anteriores cuenta con una mayor eficiencia, ya que es capaz de medir la concentración únicamente para los tamaños de partículas de  $2.5 \mu\text{m}$  y  $10 \mu\text{m}$ , pero con mejores eficiencias, un rango de medición igual al del sensor HPMA115CO-004, y una resolución mayor a los demás sensores de  $0.3 \mu\text{g}/\text{m}^3$ .

Después de realizar el análisis de los sensores de la tabla 16, se concluye que estos son herramientas útiles para el monitoreo de la concentración de la masa total para partículas de  $0.3 \mu\text{m}$ ,  $1.0 \mu\text{m}$ ,  $2.5 \mu\text{m}$  y  $10 \mu\text{m}$ . Sin embargo, para la implementación en cuartos de ambiente controlado utilizados para la fabricación de dispositivos médicos, donde la gran mayoría de cuartos se clasifican

ISO Clase 8, según la norma ISO 1464-1:2015, estos sensores no se consideran adecuados ya que las partículas que se monitorean en los cuartos ISO clase 8, corresponde a 0.5  $\mu\text{m}$ , 1  $\mu\text{m}$  y 5  $\mu\text{m}$ .

### Sensores de calidad de aire, medición de conteo de partículas

Parámetro	IPS-7100			SPS30			
	Condiciones	Valor	Unidad	Condiciones	Valor	Unidad	
Rango de conteo de partículas	0.1	$\leq 0.1$	$\mu\text{m}$	PM 0.5	0.3 - 0.5	$\mu\text{m}$	
	0.3	0.1-0.3	$\mu\text{m}$				
	0.5	0.3-0.5	$\mu\text{m}$	PM 1.0	0.3 - 1.0	$\mu\text{m}$	
	1.0	0.5-1.0	$\mu\text{m}$				
	2.5	1.0-2.5	$\mu\text{m}$	PM 2.5	0.3 - 2.5	$\mu\text{m}$	
	5.0	2.5-5.0	$\mu\text{m}$	PM 4	0.3 - 4.0	$\mu\text{m}$	
	10	5.0-10	$\mu\text{m}$	PM 10	0.3 - 10	$\mu\text{m}$	
Precisión conteo de partículas	N/A	$\pm 10$	%	Para partículas de 0.5, 1.0 y 2.5 $\mu\text{m}$	0 a 1000 $\#/\text{cm}^3$	$\pm 100$	$\#/\text{cm}^3$
					1000 a 3000 $\#/\text{cm}^3$	$\pm 10$	% m.v.
				Para partículas de 4.0 y 10 $\mu\text{m}$	0 a 1000 $\#/\text{cm}^3$	$\pm 250$	$\#/\text{cm}^3$
					1000 a 3000 $\#/\text{cm}^3$	$\pm 25$	% m.v.
Resolución de conteo de partículas	N/A	1	$\#/\text{Litro}$	No especifica			
Rango de tamaño de concentración de masa	PM 0.1	$\leq 0.1$	$\mu\text{m}$	PM 1.0	0.3 - 1.0	$\mu\text{m}$	
	PM 0.3	$\leq 0.3$	$\mu\text{m}$				
	PM 0.5	$\leq 0.5$	$\mu\text{m}$	PM 2.5	0.3 - 2.5	$\mu\text{m}$	
	PM 1.0	$\leq 1.0$	$\mu\text{m}$				
	PM 2.5	$\leq 2.5$	$\mu\text{m}$	PM 4	0.3 - 4.0	$\mu\text{m}$	
	PM 5.0	$\leq 5.0$	$\mu\text{m}$				
	PM 10	$\leq 10$	$\mu\text{m}$	PM 10	0.3 - 10	$\mu\text{m}$	
Límite de conteo de partículas	@ 2,5 $\mu\text{m}$	1 000 000	$\#/\text{Litro}$	0 - 3 000		$\#/\text{cm}^3$	
Límite de tamaño de concentración de masa	@ $\leq 2,5 \mu\text{m}$	6 000	$\mu \text{g}/\text{m}^3$	0 - 1 000		$\mu \text{g}/\text{m}^3$	

Tabla 17. Sensores de conteo de partículas en el aire.

Fuente: Propia.

En la tabla anterior se listan los sensores disponibles en el mercado actual capaces de medir el conteo de partículas en el aire. Los sensores IPS-7100 y SPS30, además de contar con la capacidad de medir el conteo de partículas en el aire, también son capaces de medir la concentración de estas en el aire, por lo que en definitiva se presentan como una mejor opción que los sensores listados en la tabla 16.

Para la característica de conteo de concentración de partículas en el aire, el sensor IPS-7100 cuenta con la capacidad de un rango de medición de  $6\ 000\ \mu\text{g}/\text{m}^3$ , siendo mayor a los  $1\ 000\ \mu\text{g}/\text{m}^3$ , del sensor SPS30, además que el sensor IPS-7100 cubre la medición de concentración de partículas para los tamaños de 0.1, 0.3, 0.5, 1.0, 2.5, 5.0, 10  $\mu\text{m}$ .

El sensor IPS-7100 es capaz de medir la cantidad de partículas en el aire para los tamaños de 0.1, 0.3, 0.5, 1.0, 2.5, 5.0, 10  $\mu\text{m}$ , mientras que el sensor SPS30 solamente es capaz de medir el conteo de partículas para los tamaños 0.5, 1.0, 2.5, 4.0 y 10.0. Según se explicó anteriormente, de acuerdo con la norma ISO 1464-1:2015, para los cuartos clase 8, las partículas que se monitorean corresponden a 0.5  $\mu\text{m}$ , 1  $\mu\text{m}$  y 5  $\mu\text{m}$ . De acuerdo con esta información el sensor IPS-7100, se presenta como la mejor opción a implementar en el sistema de monitoreo para el conteo de partículas.

Es importante considerar que las unidades utilizadas por el sensor IPS-7100 para la medición del conteo de partículas corresponde a número de partículas por litro ( $\#/ \text{Litro}$ ). Esta unidad de medida difiere un poco con la cantidad establecida en la norma ISO 1464-1:2015, la cual establece la cantidad permitida de partículas para cada tamaño en las unidades de número de partículas por metro cúbico ( $\#/\text{m}^3$ ). Aunque el sensor IPS-7100 no brinde el dato directamente en las mismas unidades a la norma, sigue siendo una herramienta funcional para el diseño, dado que los datos obtenidos por el sensor IPS-7100, pueden ser sometidos a una conversión o a un estudio que permita tener un estimado de a qué nivel de partículas por litro ( $\#/ \text{Litro}$ ) estaría en riesgo la calidad del aire en el interior del cuarto limpio.

En resumen, se considera apto para el sistema de monitoreo el sensor IPS-7100, dado que es capaz de medir la cantidad de partículas en el aire para los tamaños de partículas que establece la norma ISO 1464-1:2015, además que es capaz de medir la concentración de partículas en el aire para los tamaños de partículas que son de principal interés en los cuartos de ambiente controlado, clase ISO 8, utilizados generalmente para la fabricación de dispositivos médicos.

### Análisis características eléctricas/físicas de sensores medidores de partículas

Como parte del análisis de los sensores a implementar en el sistema de monitoreo es importante destacar las características eléctricas/físicas del sensor medidor de partículas, tomando en cuenta aspectos como el consumo de corriente, voltaje de alimentación, protocolo de comunicación que utiliza e incluso dimensiones físicas del sensor, los cuales son aspectos importantes a la hora de implementar el sistema de monitoreo.

A continuación, se muestra la tabla que resume las características eléctricas/físicas del sensor IPS-7100.

Características	IPS-7100	
Voltaje de alimentación	4.5 ~ 5.5 V	
Consumo de corriente	Condiciones	Valor
	En estado activo	65 mA
	Modo reposo (UART)	67 $\mu$ A
	Modo reposo (I2C)	273 $\mu$ A
Protocolo de interfaz	UART_TTL /I2C (3,3V)	
Dimensiones	4.6 (ancho) x 4.15 (largo) x 1.24 (espesor) cm	
Peso	26 gramos	
Vida útil / Tiempo medio hasta la falla	Operando las 24/día, > 8 años	

Tabla 18. Características eléctricas/físicas del sensor IPS-7100.

Fuente: Propia.

Tal y como se explicó anteriormente, si se desea conocer el consumo eléctrico del sensor IPS-7100, basta con utilizar la ecuación de la Ley de Ohm, como se muestra a continuación:

$$P = V \times I$$

$$P = 5 V * 65 \times 10^{-3}$$

$$P = 0.325 W$$

De acuerdo con la ecuación anterior, si el sensor IPS-7100, se conecta a una fuente de alimentación de 5 V, y manteniendo un consumo de corriente promedio de 65 mA, su consumo eléctrico corresponde a 0.325 W.

De acuerdo con los datos del fabricante, el sensor IPS-7100, es capaz de utilizar los protocolos de comunicación I2C y UART para la transferencia de los datos obtenidos por el sensor al microcontrolador, además los puertos de salida de las interfaces son capaces de operar en voltajes de 3.3 V y 5.5 V, lo cual es una característica importante para la selección del microcontrolador más adelante.

Con respecto a las características físicas del sensor este cuenta con las dimensiones de 4.6 cm (ancho) x 4.15 cm (largo) x 1.24 cm (espesor), lo cual lo vuelve un dispositivo bastante versátil y fácil de implementar en los puntos de muestreo por sus dimensiones compactas, además el proporciona el sensor calibrado y con una vida útil de aproximadamente 8 años en uso continuo.

### **Microcontroladores considerados para el sistema de monitoreo**

En el sistema de monitoreo, además de los sensores encargados de medir las variables específicas para las que fueron diseñados, se necesitará conectarlos a un microcontrolador. Este dispositivo se encarga de recibir los datos de los sensores, procesar la información obtenida y realizar las acciones correspondientes según las instrucciones programadas.

Dado que el sistema de monitoreo pretende realizar una digitalización de los datos de temperatura, humedad, presión y cantidad de partículas suspendidas en el aire en un cuarto limpio, la implementación de varios puntos de muestreo mediante sensores es vital para monitorear de manera eficiente toda el área del cuarto limpio.

Tomando en cuenta que el sistema requiere de varios sensores, es importante para el sistema elegir un microcontrolador que se encargue de comunicarse con los sensores y con el servidor al que va a enviar los datos de una manera estrictamente remota y sincronizada, con el fin de reducir el cableado y facilitar la implementación de la conexión de los sensores a la red de internet.

Debido a lo anterior, los microcontroladores que van a ser considerados para el sistema deben cumplir con las siguientes características:

- Compatible con el entorno de programación de Arduino y sensores seleccionados anteriormente.
- Capacidad de comunicación integrada por medio de Wi-Fi
- Suficientes pines de entrada y salida para conectar al menos un sensor de temperatura y humedad, un sensor de presión relativa y un sensor medidor de partículas en el aire.
- Suficientes interfaces de comunicación I2C, SPI o UART

En la siguiente tabla se listan los microcontroladores disponibles en el mercado que cumplen con las características anteriores y que se adecuan al sistema de monitoreo respecto a las funcionalidades que se requieren.

Nombre de la placa	Fabricante	Precio aproximado (\$)	Enlace del fabricante, suplidor
ESP8266 Dev Board	ESPRESSIF	9.95	<a href="https://www.crcibernetica.com/nodemcu-v3-lua-lolin-esp8266-dev-board/">https://www.crcibernetica.com/nodemcu-v3-lua-lolin-esp8266-dev-board/</a>
ESP32-DevKit C	ESPRESSIF	14.95	<a href="https://www.crcibernetica.com/esp32-wifi-and-bluetooth-dev-board/">https://www.crcibernetica.com/esp32-wifi-and-bluetooth-dev-board/</a>
ESP32-DevKitM-1	ESPRESSIF	8	<a href="https://n9.cl/3thzvc">https://n9.cl/3thzvc</a>
ESP32-PICO-KIT-1	ESPRESSIF	10	<a href="https://www.amazon.com/dp/B0CPP8876C?ref=myi_title_dp">https://www.amazon.com/dp/B0CPP8876C?ref=myi_title_dp</a>
ESP32-S3-DevKitC-1	ESPRESSIF	15~17	<a href="https://n9.cl/tmrf22">https://n9.cl/tmrf22</a> <a href="https://www.amazon.com/dp/B09R4GSDJM?ref=myi_title_dp">https://www.amazon.com/dp/B09R4GSDJM?ref=myi_title_dp</a>
Arduino Nano ESP32	Arduino	20	<a href="https://n9.cl/it9kp">https://n9.cl/it9kp</a>
Arduino UNO R4 Wi-Fi	Arduino	49.95	<a href="https://www.crcibernetica.com/arduino-uno-r4-wifi/">https://www.crcibernetica.com/arduino-uno-r4-wifi/</a>

Tabla 19. Microcontroladores considerados para el sistema de monitoreo.

Fuente: Propia.

Después de realizar una exhaustiva revisión de los microcontroladores/placas de desarrollo disponibles en el mercado actual que cumplen con las especificaciones establecidas anteriormente es indispensable analizar las funcionalidades y características disponibles en los microcontroladores.

A continuación, se explica únicamente las características que son importantes a tomar en cuenta para la selección del microcontrolador adecuado para el sistema de monitoreo, las cuales corresponden a:

- **GPIO Pines:** se refiere a los pines de entrada/salida de propósito general, la cantidad de pines GPIO indica cuantos pines de la placa del microcontrolador pueden ser programados para varias funciones como entradas/salidas digitales, comunicación serial, lectura de datos, controlador de dispositivos, entre otros.
- **Interfaces SPI, I2C, UART:** indica la capacidad del microcontrolador para soportar cierta cantidad de dispositivos que se comuniquen simultáneamente utilizando el protocolo de comunicación serial SPI, I2C y UART.
- **Memoria *Flash*:** es una memoria no volátil, que retiene la información del *firmware* del dispositivo y el programa del usuario (ejemplo: el sketch de Arduino IDE).
- **Memoria ROM:** corresponde a una memoria no volátil y contiene el código para el funcionamiento del dispositivo, como inicios del sistema operativo y la carga del programa del usuario.
- **Memoria SRAM:** es la encargada de almacenar datos temporales y variables cuando se ejecuta el programa, siendo una memoria volátil.
- **Velocidad del reloj del procesador:** se mide en hercios (Hz) e indica la velocidad del reloj para la ejecución de instrucciones, accesos a la memoria, entre otros. Indica la velocidad en que se ejecutarán las tareas internas.
- **Procesador:** corresponde al modelo y arquitectura del procesador central, los bits del procesador indica que puede manejar datos y operaciones en unidades de bits a la vez.
- **WiFi:** se refiere al estándar de la red inalámbrica, determinando la banda de frecuencia y capacidad de velocidades de transmisión de datos.

En la siguiente tabla se muestran los valores de las características anteriores a comparar entre los sensores listados de la tabla 19.

Modelo de Placa	Serie/Módulo	GPIO	SPI	I2C	UART	Flash	ROM	SRAM	RAM	Procesador	WiFi	Dimensiones de la placa
ESP8266	ESP-WROOM-02D/ESP-WROOM-02U	17	2	1	2	2/4/16 MB	N/A	160 KB		Tensilica L106 32-bit RISC	802.11. b/g/n	44,9mm x 25,4 mm
ESP32-DevKit C	ESP32-WROOM-32 ESP32-WROOM-32E ESP32-WROOM-32UE ESP32-WROOM-DA ESP32-WROVER-E ESP32-WROVER-IE	34	4	2	3	4/8/16 MB	448 KB	520 KB		Xtensa LX6 dual-core 32-bit	802.11. b/g/n	54,4 mm x 27,9 mm
ESP32-DevKitM-1	ESP32-MINI-1 ESP32-MINI-1U	34	4	2	3	4 MB	448 KB	520 KB		Xtensa LX6 dual-core 32-bit	802.11. b/g/n	43,81 mm x 25,40 mm
ESP32-PICO-KIT-1	ESP32-PICO-V3 ESP32-PICO-D4 ESP32-PICO-V3-02	31/34	4	2	3	4/8 MB	448 KB	520 KB		Xtensa LX6 dual-core 32-bit	802.11. b/g/n	47,00 mm x 20,30 mm
ESP32-S3-DevKitC-1	ESP32-S3-WROOM-1 ESP32-S3-WROOM-1U ESP32-S3-WROOM-2 ESP32-S3-WROOM-2U	45	4	2	3	8/16/32 MB	384 KB	512 KB		Xtensa LX7 dual-core 32-bit	802.11. b/g/n	62,74 mm x 25,40 mm
Arduino Nano ESP32	NORA-W106-10B u-blox	14 Digital 8 Analog	4	1	2	16 MB	384 KB	512 KB		Xtensa LX7 dual-core 32-bit	802.11. b/g/n	43,18 mm x 17,78 mm
Arduino UNO R4 WiFi	ESP32-S3-MINI-1-N8	14 Digital 6 Analog	1	1	1	8 MB	384 KB	512 KB		Xtensa LX7 dual-core 32-bit	802.11. b/g/n	68,58 mm x 53,34 mm

Tabla 20. Características de los microcontroladores.

Fuente: Propia.

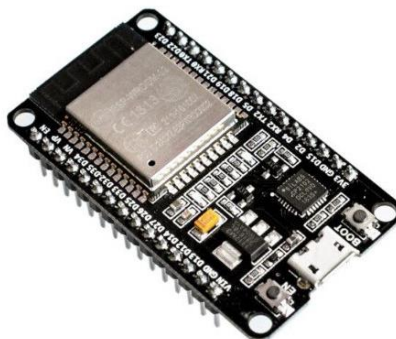
Como parte del análisis para la selección del microcontrolador es importante tener en cuenta que los sensores considerados para el sistema de monitoreo corresponden a los sensores SHT31-D, BMP390 e IPS-7100. Tomando en cuenta los protocolos de comunicación que utilizan, estos sensores para funcionar simultáneamente requerirán de al menos un microcontrolador capaz de utilizar dos (2) interfaces I2C, una (1) interfaz UART y una (1) interfaz SPI.

Con relación a la cantidad de sensores e interfaces de comunicación requeridos se necesitaría como mínimo seis (6) pines GPIO para la conexión de todos los sensores al microcontrolador.

Después de analizar los requerimientos mínimos de los sensores para comunicarse con el microcontrolador, de acuerdo con la tabla 20, los microcontroladores que cumplen con estos requerimientos son; ESP32-DevKit C, ESP32-DevKitM-1, ESP32-PICO-KIT-1, ESP32-S3-DevKitC-1.

Entre los microcontroladores mencionados anteriormente el que mejor se adapta al sistema de monitoreo corresponde a microcontrolador ESP32-DevKit C, dado que no se requieren de muchos pines GPIO, lo 34 pines del ESP32-DevKit C son suficientes, puede tener una capacidad de memoria *flash* de hasta 16 MB, ROM de 448 KB, SRAM de 520 KB, utilizando un microprocesador de 32 bit con la capacidad de conexión a WiFi mediante los estándares 802.11. b/g/n.

Dado que para el sistema de monitoreo la implementación se requiere de la colocación de puntos de muestreo, el tamaño de los componentes resulta importante para mejorar la facilidad de implementación, el microcontrolador ESP32-DevKit C desarrollado por Espressif ofrece todas las características mínimas requeridas por el sistema de monitoreo con dimensiones de la placa de 54,4 mm x 27,9 mm.



*Figura 32. Placa ESP32-WROOM-32*

*Fuente: CRCibernética, 2023*

### **Análisis características eléctricas del microcontrolador**

Para la conexión de los sensores al microcontrolador es importante entender las características eléctricas de este, tomando en consideración su voltaje de alimentación, su voltaje de operación, corriente de consumo e incluso su capacidad de proveer alimentación a los sensores conectados a él. En la siguiente tabla se detallan sus características:

<b>Características eléctricas</b>	
<b>Características</b>	<b>ESP32-DevKit C</b>
Voltaje de alimentación por USB	5 V
Voltaje de operación	3,0 ~3,6 V
Consumo de corriente	240 mA
Pines de voltaje de salida = 5V	1
Corriente de salida	500 mA

*Tabla 21. Características eléctricas ESP32-DevKit C*

*Fuente: Propia.*

De acuerdo con la tabla 21, la conexión USB ofrece una solución conveniente para alimentar la placa del microcontrolador con un voltaje de 5V. Aunque el microcontrolador funciona dentro de un rango de 3.0 a 3.6V, es posible gracias a que la placa cuenta con un regulador de voltaje, que adapta la tensión de entrada de 5V a los 3.3 V necesarios.

Con respecto al consumo promedio de corriente del microcontrolador, este equivale a 240 mA, por lo que, utilizando la ley de Ohm, el consumo eléctrico corresponde a:

$$P = V \times I$$

$$P = 5 V * 240 \times 10^{-3} A$$

$$P = 1.2 W$$

Conociendo el consumo eléctrico promedio del microcontrolador ESP32-DevKit C y el consumo eléctrico de los sensores SHT31-D, BMP390 e IPS-7100 es posible determinar el consumo eléctrico total de cada punto de medición utilizado en el sistema de monitoreo, tal y como se muestra en la siguiente tabla:

Consumo eléctrico					
Características	SHT31-D	BMP390	IPS-7100	ESP32-DevKit C	Total
Potencia (W)	0.00001	0.00001056	0.325	1.2	1.525

Tabla 22. Potencia Total(W) por punto de muestreo

Fuente: Propia.

Después de haber analizado cada uno de los dispositivos encargados de la recopilación de los datos de temperatura, humedad, presión relativa y conteo de partículas, aspectos técnicos, dimensionales y eléctricos, fue posible determinar los dispositivos disponibles en el mercado que mejor se adaptan al diseño del sistema de monitoreo, llegándose a conocer que cada punto de muestreo tendrá un consumo promedio de 1.525 W.

## **Programación de los sensores utilizando Arduino IDE**

En esta sección del presente proyecto, se pretende explicar los aspectos importantes a tomar en cuenta para la configuración de los sensores y del microcontrolador para lograr la interconexión y comunicación de los datos obtenidos por los sensores hacia el microcontrolador.

### **Conexión física de los sensores al microcontrolador**

Primeramente, antes de iniciar con la programación del código que será ejecutado en el microcontrolador para la configuración de las tareas a ejecutar cada sensor, se debe establecer que pines del microcontrolador serán utilizados.

Como se explicó anteriormente, los sensores tienen la capacidad de transmitir sus datos mediante interfaces de comunicación I2C, UART y SPI, en el caso de los sensores seleccionados anteriormente para el sistema de monitoreo, se utilizará para cada sensor los siguientes protocolos de comunicación con el microcontrolador:

- Sensor SHT31: protocolo I2C
- Sensor BMP390: protocolo SPI
- Sensor IPS-7100: protocolo I2C

Con los protocolos definidos por cada sensor es posible determinar la cantidad de pines que requerirá cada sensor. Es importante recordar que cada sensor necesitará un Pin VCC y GND, los cuales corresponden a los pines de alimentación y tierra respectivamente. Con base al diagrama de distribución de pines facilitado por el fabricante es posible determinar que pin será utilizado para la conexión de cada sensor y por ende el pin que será tomado en cuenta para la escritura del código en Arduino IDE.

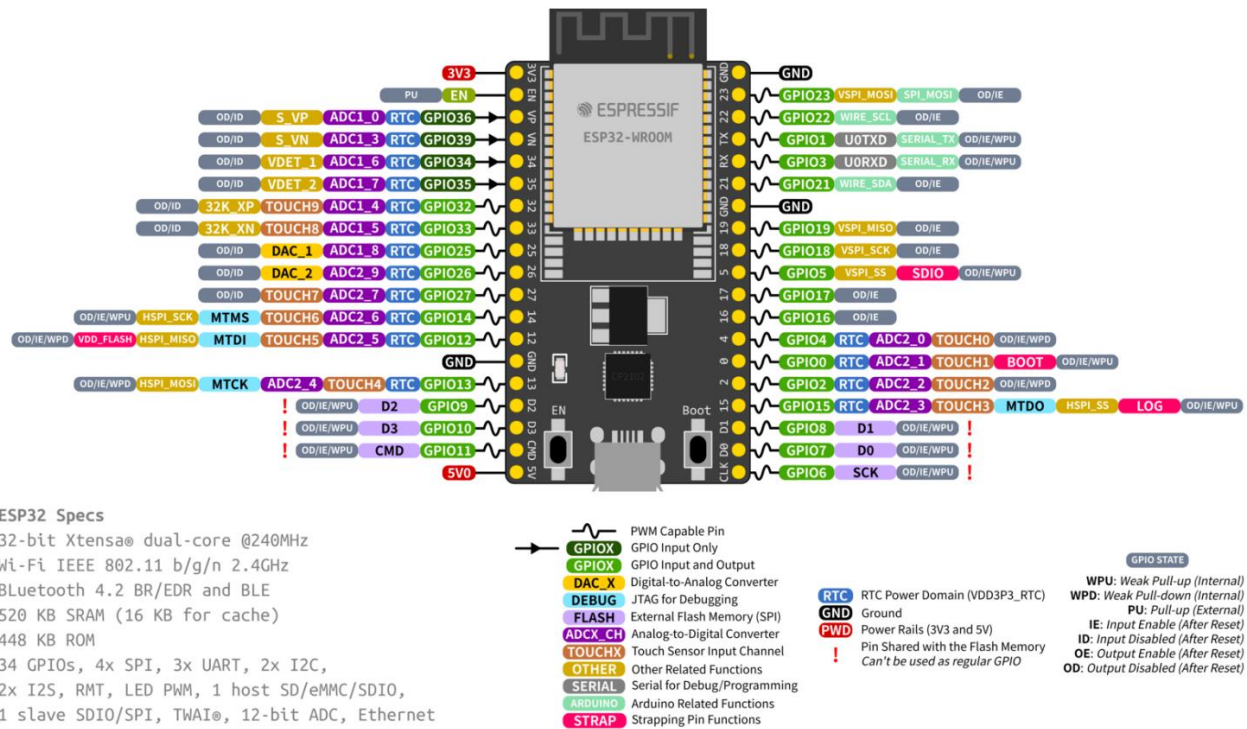


Figura 33. Diseño de Pines ESP32-DevKit C

Fuente: ESPRESSIF, s.f.

En la figura 33, es posible observar que en la esquina inferior izquierda tiene un pin llamado “5VO” este pin será quien dará alimentación a los dos sensores SHT31-D e IPS-7100 ya que ambos operan dentro del rango de 5V. Mientras que el pin llamado “3V3” ubicado en la esquina superior izquierda será el pin de alimentación del sensor BMP390 que opera dentro del rango de los 3.3 V. Además, según se requiera se puede utilizar como tierra cualquiera de los pines demarcados en la placa como “GND” para cada sensor.

Seguidamente para la conexión de los sensores SHT31-D e IPS-7100, estos serán conectados a la placa utilizando ambos sensores los pines GPIO 22, demarcado en la placa como “22” este pin corresponde al SCL, y el pin GPIO 21, demarcado en la placa como “21” este pin corresponde al SDA. Esto es posible dado que el protocolo I2C permite que dispositivos

compartan el mismo bus de comunicación ya que cada sensor en el bus tiene una dirección única. Según el fabricante, el SHT31-D tiene una dirección por defecto I2C de 0x44, mientras que la dirección I2C del sensor IPS-7100 corresponde a 0x4b.

El sensor BMP390 será conectado al ESP32 mediante el protocolo SPI, por lo que se requiere conectar el sensor a los pines GPIO18 demarcado en la placa como “18”, el cual corresponde al pin SCL (SCK), el pin GPIO19 demarcado en la placa como “19” que corresponde al pin SDO (MISO), el pin GPIO23 demarcado en la placa como “23” que corresponde al pin SDA (MOSI) y finalmente el pin GPIO5, demarcado en la placa como “5” que corresponde al pin CS.

Finalmente, para la conexión del microcontrolador a la fuente de alimentación, dado que este tiene la posibilidad de ser alimentado mediante USB con un voltaje de 5V, para facilitar la implementación del sistema de monitoreo, se pretende alimentar al microcontrolador con una fuente de energía fija en vez del uso de baterías, esto con el fin de evitar la pérdida de datos por la descarga de baterías en cada punto de muestreo. El uso de un adaptador de voltaje, como el que se utiliza en la carga de celulares (cargador de pared USB), los cuales convierten el voltaje de la red eléctrica generalmente de 120V o 240 V a una salida de 5V DC con una corriente aproximada a los 2.0 A, lo cual es suficiente para la alimentación del microcontrolador y sus sensores.

En la siguiente figura se muestra el esquema de conexión física del microcontrolador y los sensores.

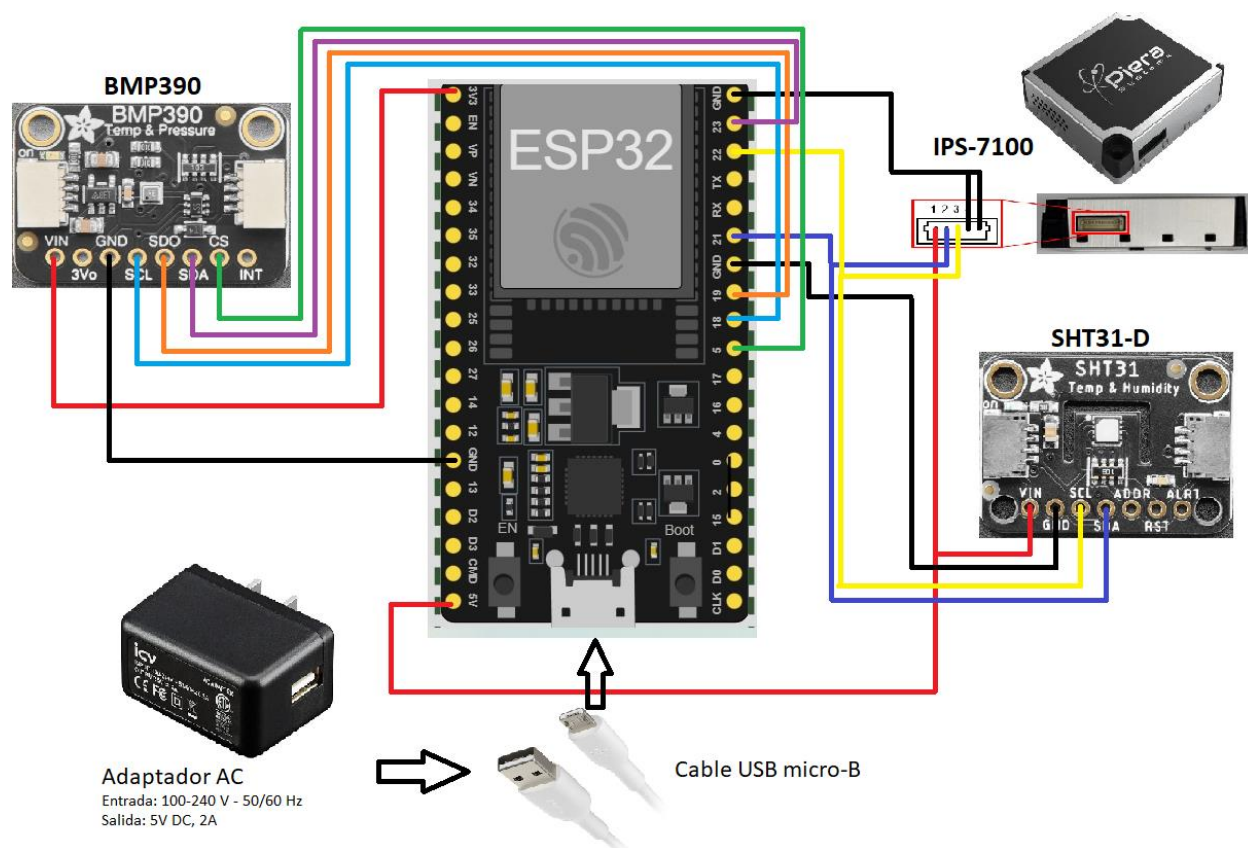


Figura 34. Conexión física del microcontrolador y sensores.

Fuente: Propia

### Respaldo de instalación eléctrica de los circuitos

Es importante considerar el estado de los circuitos donde serán colocados los puntos de muestreo y el servidor local. Para garantizar la continuidad operativa del sistema de monitoreo, se deberá asegurar que los circuitos correspondientes a cada nodo y al servidor local cuenten con un respaldo eléctrico adecuado. El presente proyecto tomará en cuenta que los circuitos de cada nodo se encuentran respaldados por la planta de alimentación de las facilidades donde se encuentra el cuarto limpio. Esta planta de emergencia se encarga de mantener durante un tiempo determinado el uso de los sistemas de ventilación del cuarto limpio con el fin de prevenir el aumento de partículas suspendidas en el aire durante un apagón. A su vez dicha planta tendrá la capacidad de alimentar a los circuitos donde se encuentran conectados los nodos y el servidor local.

En el caso del servidor local este estará respaldado por una UPS (sistema de alimentación ininterrumpida, por sus siglas en inglés) que proporcionará energía eléctrica en casos de un corte de

energía o fluctuaciones en el suministro eléctrico con el fin de proteger y a su vez mantener por un periodo de tiempo la comunicación en el sistema de monitoreo.

En situaciones donde no se disponga de una planta eléctrica de respaldo para los circuitos del sistema de monitoreo, se recomienda por buena práctica en la industria médica (esta práctica se implementa por lo general aunque se cuente con circuitos respaldados) siempre que suceda un corte de energía se verifiquen las condiciones ambientales del cuarto para confirmar si este se encuentra dentro de los parámetros antes de continuar con las actividades de producción, lo que conlleva a que una persona encargada verifique en la base de datos el estado de las mediciones posteriores al corte de energía.

### **Programación de los sensores**

Como parte de la programación de los sensores, se utilizará Arduino IDE, la cual como se explicó anteriormente es una multiplataforma que puede utilizarse para escribir y cargar programas en placas Arduino y también en aquellas que sean compatibles.

Primeramente, para iniciar con la configuración de los sensores, se debe instalar en el *sketch* de Arduino las librerías de los sensores. Las librerías proveen funciones predefinidas que manejan la comunicación y el procesamiento de datos, facilitando la interacción con cada sensor/microcontrolador o protocolos en específico mediante una serie de comandos.

Es posible añadir las librerías mediante la pestaña de Arduino IDE “Manejo de librerías” la cual permite buscar todas las librerías disponibles publicadas por fabricantes y comunidad de desarrolladores.

### **Programación de los sensores SHT31-D y BMP390**

De acuerdo con las recomendaciones del fabricante Adafruit para el sensor de presión BMP390 y el sensor SHT31-D, se deben seguir los siguientes pasos:

Paso 1: incluir las librerías “BMP3XX” y “Adafruit SHT31” de acuerdo con la siguiente figura.

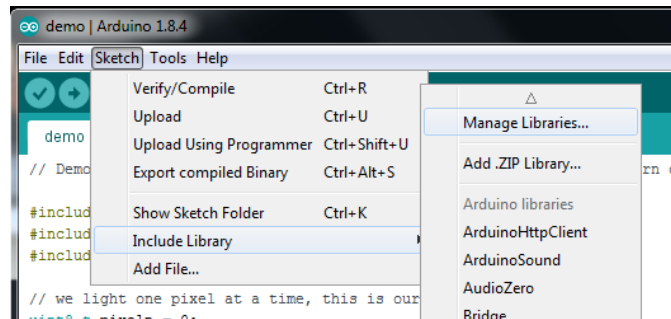


Figura 35. Librerías de sensores BMP390 y SHT31-D.

Fuente: Adafruit, 2015

Paso 2: dado que los sensores serán conectados al microcontrolador utilizando el protocolo I2C y SPI, se debe incluir en el *Sketch* la librería “Wire.h” y “SPI.h”. Estas librerías ayudan al microcontrolador a enviar comandos al sensor y leer los datos que este proporciona.

Ejemplo de código:

```
#include <Adafruit_SHT31.h> //Librería del sensor de temperatura y humedad
SHT31-D
#include <Adafruit_BMP3XX.h> //Librería del sensor de presión BMP390
#include <Wire.h> //funciones para comunicación I2C
#include <SPI.h> //funciones para comunicación SPI
```

Paso 3: definir las variables que serán medidas por cada sensor y los pines a utilizar en el microcontrolador, de manera que sea posible leer a lo largo del código a que variable corresponde la programación.

Ejemplo de código:

```
#define BMP_SCK 18 //Definir pin SCL(SCK) para BMP390
#define BMP_MISO 19 //Definir pin SDO (MISO) para BMP390
#define BMP_MOSI 23 //Definir pin SDA (MOSI) para BMP390
#define BMP_CS 5 //Definir pin CS para BMP390
```

```
Adafruit_BMP3XX bmp; // Se declara como "bmp" para BMP3XX
Adafruit_SHT31 SHT31 = Adafruit_SHT31(); //Se declara como "SHT31" para
sensor SHT31, además de inicializarse el sensor.
```

```
float temperatura = 0; // Variable equivale a cero cuando el sensor no
realiza la medición
```

```
float humedad = 0; // Variable equivale a cero cuando el sensor no realiza
la medición
float presion = 0; // Variable equivale a cero cuando el sensor no realiza
la medición
```

Paso 4: en la sección del *Sketch* “void setup ()” se debe inicializar el sensor y configurar los sensores. En la sección de “void loop ()” se debe programar los comandos para que el microcontrolador lea los datos proporcionados por los sensores.

Ejemplo de código:

#### Sección de “void setup ()”

```
// Se inicializan los sensores
```

```
SHT31.begin(0x44); //inicializar sensor SHT31 con la dirección I2C
bmp.begin_SPI (BMP_CS, BMP_MOSI, BMP_MISO, BMP_SCK); //inicializar sensor
BMP390
```

```
// Configuración del sensor BMP390
```

```
bmp.setIIRFilterCoeff (BMP3_IIR_FILTER_COEFF_3); //función que configura al sensor
para filtrar las variaciones rápidas
```

#### Sección de “void loop ()”

Acá se define el bloque llamado “void variables” y se colocan las equivalencias para la lectura de cada variable.

```
void variables () {
```

```
temperatura = SHT31.readTemperature(); // Leer el valor de temperatura en el sensor
humedad = SHT31.readHumidity(); // Leer el valor de humedad en el sensor
presion = (bmp.pressure / 100.0); // Leer el valor de presión en hPa en el
sensor
delay(21600000); // pausa la ejecución del bloque "void variables" por 6
horas
```

```
}
```

```
//En caso de que fallen los sensores en la medición se registrará un valor
de cero.
```

```
if (isnan(temperatura) || isnan(humedad) || isnan(presion)) { // Sino se
obtienen los valores de las variables, estas toman el valor de cero.
```

```
Serial.println("Error en leer variables!");
```

```
temperatura = 0;
```

```
humedad = 0;
```

```

presion = 0;
}

```

Finalmente, con las variables definidas para cada una de las funciones correspondientes del sensor, es posible utilizar las variables a lo largo del código, para las distintas funcionalidades que se le desee dar. Es importante aclarar que en el caso que por alguna circunstancia alguno de los sensores presente alguna falla que no le permita realizar la medición, automáticamente cuando se ejecute el código se le asignará a la variable medida un valor de cero. Con respecto a la base de datos esta variable seguirá registrándose con un valor de cero para el sensor que presenta la falla hasta que sea reparado. Para más detalle ver en el anexo 1 el código utilizado para la programación del sistema de monitoreo y sus secuencias en Arduino IDE.

### **Programación del sensor IPS-7100**

Para la programación del sensor IPS-7100 los pasos a utilizar son similares al de los sensores SHT31-D y BMP390. De acuerdo con el fabricante Piera System se debe:

Paso 1: incluir la librería “IpsI2C.h” y asegurarse de que esté incluido en el *sketch* la librería “Wire.h” para la comunicación I2C entre el sensor y el microcontrolador.

Ejemplo de código:

```

#include <IpsI2C.h> //Librería del sensor de partículas en el aire IPS-
7100

```

Paso 2: definir las variables que serán medidas por el sensor de partículas, ya sea concentración de partículas en el aire o conteo de partículas en el aire, además de los pines a utilizar en el microcontrolador, de manera que sea posible leer a lo largo del código a que variable corresponde la programación.

Ejemplo de código:

```

#define SDA_PIN 21 //Definir pin SDA para IPS-7100
#define SCL_PIN 22 //Definir pin SCL para IPS-7100

```

```

IpsSensor ips_sensor; // Se declara como "ips sensor" para sensor IPS7100

```

```

// Las siguientes variables equivalen a cero cuando el sensor no realiza
la medición

```

```
float PM0p5 = 0; // Concentración de partículas de 0.5 micrómetros
float PM1p0 = 0; // Concentración de partículas de 1.0 micrómetros
float PM5 = 0; // Concentración de partículas de 5 micrómetros
float PC0p5 = 0; // Conteo de partículas de 0.5 micrómetros
float PC1p0 = 0; // Conteo de partículas de 1.0 micrómetros
float PC5 = 0; // Conteo de partículas de 5 micrómetros
```

Paso 3: en la sección del *Sketch* “void setup ()” se debe inicializar el sensor y configurar. En la sección de “void loop ()” se debe programar los comandos para que el microcontrolador lea los datos proporcionados por los sensores.

Ejemplo de código:

#### Sección de “void setup ()”

```
// Se inicializan los sensores
```

```
ips_sensor.begin(SDA_PIN, SCL_PIN); // inicializar sensor IPS7100
```

#### Sección de “void loop ()”

Acá se define el bloque llamado “void particulas” y se colocan las equivalencias para la lectura de cada variable.

```
void particulas () {
{
ips_sensor.update(); // Actualiza las variables y mediciones internas del
sensor IPS-7100

PM0p5 = ips_sensor.getPM05(); // Concentración de partículas de 0.5
micrometros
PM1p0 = ips_sensor.getPM10(); // Concentración de partículas de 1.0
micrometros
PM5 = ips_sensor.getPM50(); // Concentración de partículas de 5
micrometros
PC0p5 = ips_sensor.getPC05(); // Conteo de partículas de 0.5 micrometros
PC1p0 = ips_sensor.getPC10(); // Conteo de partículas de 1.0 micrometros
PC5 = ips_sensor.getPC50(); // Conteo de partículas de 5 micrometros

//En caso de que fallen los sensores en la medición se registrará un valor
de cero.
if (isnan(PM0p5) || isnan(PM1p0) || isnan(PM5) || isnan(PC0p5) ||
isnan(PC1p0) || isnan(PC5)) {
```

```

Serial.println("Error en leer partículas!");
PM0p5 = 0;
PM1p0 = 0;
PM5 = 0;
PC0p5 = 0;
PC1p0 = 0;
PC5 = 0;
// Imprimir valor de concentración de partículas obtenido por IPS-7100

Serial.print("PM0.5: "); // Se define nombre a imprimir para la variable
"PM0p5"
Serial.println(PM0p5); // imprimir valor leído por el sensor.
Serial.print("PM1.0: "); // Se define nombre a imprimir para la variable
"PM1p0"
Serial.println(PM1p0); // imprimir valor leído por el sensor.
Serial.print("PM5.0: "); // Se define nombre a imprimir para la variable
"PM5"
Serial.println(PM5); // imprimir valor leído por el sensor.

// Imprimir valor de conteo de partículas obtenido por IPS-7100

Serial.print("PC0.5: "); // Se define nombre a imprimir para la variable
"PC0p5"
Serial.println(PC0p5); // imprimir valor leído por el sensor.
Serial.print("PC1.0: "); // Se define nombre a imprimir para la variable
"PC1p0"
Serial.println(PC1p0); // imprimir valor leído por el sensor.
Serial.print("PC5.0: "); // Se define nombre a imprimir para la variable
"PC5"
Serial.println(PC5); // imprimir valor leído por el sensor.
}
delay(21600000); // pausa la ejecución del bloque "void partículas" por 6
horas
}
}

```

Al igual que los sensores de temperatura, humedad y presión, en el caso del sensor medidor de partículas si este presenta un error que no le permita realizar la medición de todas o algunos de los tamaños de partículas suspendidas en el aire, se asignará automáticamente un valor de cero para la variable del tamaño de partícula. Con respecto a la base de datos esta variable seguirá registrándose con un valor de cero para la variable que presenta la falla hasta que sea reparado. Para más detalle ver

en el anexo 1 el código utilizado para la programación del sistema de monitoreo y sus secuencias en Arduino IDE.

Una vez realizada la conexión física y configuración de los sensores mediante la plataforma de Arduino IDE, es posible cargar el código a la placa del microcontrolador ESP32-DevKit C para que este quede ejecutándose de manera permanente en la memoria del microcontrolador, mediante el uso de un conector de cable USB micro-B entre el computador que ejecuta el programa de Arduino IDE y la placa.

### **Programación de periodicidad de registros de sensores**

Para determinar el tiempo transcurrido entre cada medición de variables de temperatura, humedad, presión y cantidad de partículas suspendidas en el aire, se debe especificar en el código de Arduino IDE. El tiempo definido entre cada medición será el que determine la periodicidad de las mediciones del sistema de monitoreo y por lo tanto cada cuanto se realizará un registro en la base de datos. Para definir la periodicidad se deben seguir los siguientes pasos:

Paso 1: utilizar la función “millis” la cual devuelve el número de milisegundos transcurridos desde que la placa comenzó a ejecutar el programa. Mediante esta función se pueden generar intervalos entre las mediciones. Se deben crear las variables que contengan los valores.

Ejemplo de código:

```
//Función "millis" para verificar que cuando transcurran 6 horas se genere
un nuevo registro
unsigned long millisAnteriores= 0; // En la variable "millisAnteriores" se
almacena el tiempo anterior.
const long periodicidad = 21600000; // En la variable "periodicidad" se
realiza un registro cada 6 horas.
```

Paso 2: en la sección del *Sketch* “void loop ()” se realiza la captura del tiempo actual y se verifica la periodicidad. De esta manera se realiza un registro de acuerdo con la periodicidad que según se definió anteriormente a manera de ejemplo corresponde a 6 horas.

Ejemplo de código:

#### Sección de “void loop ()”

```
//Función millis para obtener el tiempo actual
unsigned long millisActuales = millis(); //se obtiene el tiempo actual
if (millisActuales - millisAnteriores >= periodicidad)
```

```
{millisAnteriores = millisActuales; //Actualizar el tiempo actual
```

Mediante la variable periodicidad el usuario puede determinar la frecuencia con la que se tomen los registros de variables mediante el sistema de monitoreo en el cuarto limpio.

### **Topología de conectividad a la red para el sistema de monitoreo**

Como se ha explicado en capítulos anteriores, una de las capas importantes para la arquitectura de una red IoT corresponde a la capa de red de comunicaciones, la cual es donde los sensores solventan la necesidad de comunicarse con un sistema externo.

Para el presente diseño del sistema de monitoreo, una de las condiciones más importantes para seleccionar la topología de conectividad es el hecho que se requiere que cada uno de los microcontroladores utilizados en el sistema de monitoreo transmita los datos obtenidos por los sensores hacia el servidor de una manera inalámbrica con la finalidad de reducir el cableado y facilitar la implementación del sistema de monitoreo.

Dado que existe una relación directa entre la tecnología de red IoT seleccionada y el tipo de topología de conectividad, primeramente, se debe seleccionar la tecnología de acceso a la red.

### **Tecnología de acceso a la red**

Dado que el presente diseño del sistema de monitoreo pretende brindar una solución para el monitoreo de condiciones ambientales en cuartos de ambiente controlado para la fabricación de dispositivos médicos, es común que en estas instalaciones se cuente con tecnologías para el acceso a la red como LAN, la cual utiliza tecnologías de red estándar como Ethernet o IEEE 802.11 Wi-Fi.

Con el fin de reducir el cableado, la tecnología de acceso inalámbrico a utilizar en el sistema de monitoreo corresponde al estándar IEEE 802.11 Wi-Fi el cual tiene un alcance de cobertura de rango medio que se adapta a las necesidades del sistema. La placa ESP32 a su vez cuenta con la capacidad de conectarse a velocidades de hasta 150 Mbps mediante Wi-Fi 802.11b/g/n, permitiendo así la comunicación entre el microcontrolador y el punto de acceso a internet.

Para lograr la conexión del microcontrolador al punto de acceso de red se debe configurar en el *sketch* de Arduino IDE el ESP32 de manera que inicie el proceso de conexión Wi-Fi con la red. Para esto se debe conocer las credenciales de la red como; SSID de la red y la contraseña.

A continuación, se muestra el código que debe ser utilizado para lograr la configuración del microcontrolador ESP32 para iniciar el proceso de conexión inalámbrica con el punto de acceso:

Paso 1: incluir en el *sketch* la librería “WiFi.h” para agregar el conjunto de funciones necesarias para la configuración de la conectividad Wi-Fi en el ESP32.

Ejemplo de código:

```
#include <WiFi.h> // Funciones para conectividad por Wi-Fi
```

Paso 2: definir las variables constantes que no cambiaran durante el tiempo después de su inicialización. Correspondientes al SSID de la red y su contraseña.

Ejemplo de código:

```
const char* ssid = "Nombre de red Wi-Fi"; // Corresponde a la variable
"ssid" que contiene el nombre de la red Wi-Fi a la cual el ESP32 debe
conectarse
const char* password = "Contra123"; // Corresponde a la variable
"password" que contiene la contraseña de la red Wi-Fi
```

Paso 3: en la sección del *Sketch* “void setup ()” se debe inicializar la función de conexión a Wi-Fi. En la sección de “void loop ()” se debe programar los comandos para que el microcontrolador realice la conexión al punto de acceso.

Ejemplo de código:

Sección de “void setup ()”

```
void setup() {

// Se inicializa el bloque de conexión a Wi-Fi

conectarWiFi(); // Bloque que controla la función de conexión a Wi-Fi
```

Sección de “void loop ()”

Acá se define el bloque llamado “conectarWiFi” y se colocan las instrucciones para realizar la conexión del ESP32 a Wi-Fi. Inicialmente se debe colocar en el void loop la condición de que si no se está conectado a la red Wi-Fi, se debe ejecutar el bloque “conectarWiFi”.

```
void loop() {
```

```

// Se coloca la condición de verificar la conexión del ESP32 a la red,
sino para intentar reconectarlo ejecutando el bloque "conectarWiFi"

if (WiFi.status() != WL_CONNECTED) { //condición que verifica la
conectividad a la red.

connectWiFi(); // Si se cumple la condición anterior, se debe ejecutar el
bloque "conectarWiFi"
}

void conectarWiFi() {

WiFi.mode(WIFI_OFF); // Se apaga módulo Wi-Fi, para ser reiniciado
delay(1000); // retraso de 1 segundo para apagar completamente el módulo

WiFi.mode(WIFI_STA); // Se configura el módulo Wi-Fi en modo estación,
actúa como cliente

WiFi.begin(ssid, password); // Se inicializa el proceso de conexión a la
red con las credenciales.

Serial.println("Conectandose a WiFi"); // Se utiliza el monitor serial de
Arduino IDE para ver estado de conexión (opcional)

// El siguiente Bucle (opcional) se ejecuta solo mientras el ESP32 intenta
conectarse a la red
// Se imprime un punto cada 0.5 Segundos para indicar que está intentando
la conexión a la red.
while (WiFi.status () != WL_CONNECTED) {
    delay(500);
    Serial.print("."); // Se imprime un punto cada 0.5 Segundos
}

// Las siguientes líneas imprimen en el monitor serial la conexión exitosa
a la red
Serial.print ("Conectado a: "); Serial.println (ssid); //Indica a que red
se conectó
Serial.print ("Dirección IP: "); Serial.println (WiFi.localIP()); //Indica
la dirección IP asignada al ESP32}

```

Utilizando el código anterior es posible lograr la configuración del microcontrolador ESP32, para que establezca una conexión por medio de WiFi, con el punto de acceso a la red o enrutador. Es

importante tomar aspectos en cuenta como: asegurarse que el ESP32 esté configurado como un cliente, de manera que le permita obtener acceso a internet a través de una red Wi-Fi existente, asegurarse que los puntos de muestreo donde se coloque el ESP32 se encuentren dentro del rango de alcance del punto de acceso a internet, finalmente tomar en cuenta la configuración del punto de acceso, para que los filtros y protocolos de seguridad permitan el acceso al microcontrolador ESP32.

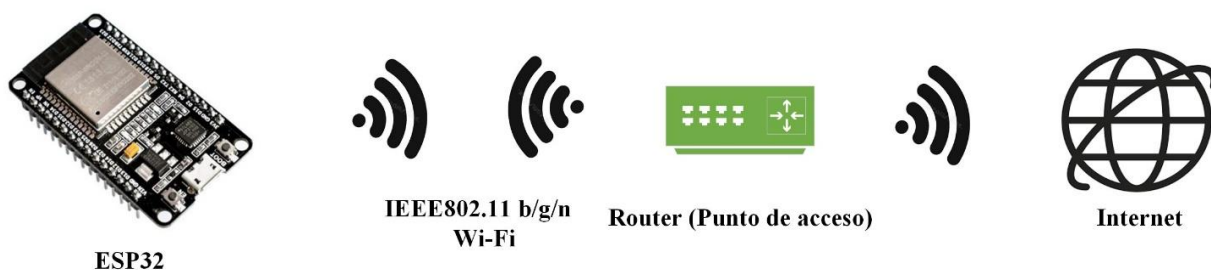


Figura 36. Conexión inalámbrica del microcontrolador.

Fuente: Propia.

### Aspectos por tomar en cuenta en caso de que el *router* falle

Es importante considerar como se establecerá la comunicación entre el ESP32 y el servidor local en caso de que el punto de acceso o *router* presente una falla que no permita la transferencia de datos. Para estos casos se deben agregar redundancias a la red. Una manera de agregar redundancia a la red es configurando el ESP32 para que pueda cambiar automáticamente entre dos *routers* garantizando la conexión incluso si uno de los *routers* falla.

Tomando en cuenta que dos *routers* se encuentran conectados a la misma red interna, de esta manera se puede garantizar que el servidor local mantendrá la misma dirección IP a la que el ESP32 enviará los datos independientemente si un *router* falla o no. Esto se puede lograr conectando físicamente ambos *routers* a un *switch* común configurando a un *router* como principal y el otro solo como extensor de la red o configurando en modo puente al *router* secundario haciendo que ambos *routers* compartan la misma red y que el *router* principal asigne las direcciones IP. A continuación, se listan los pasos a tomar en cuenta:

### Paso 1

Asignar una IP estática al servidor local para que siempre tenga la misma dirección IP, sin importar si se conecta al *router* principal o secundario, además de configurar el servidor local para que pueda conectarse automáticamente mediante Wi-Fi al *router* disponible en caso de que uno falle.

### Paso 2

Configurar el ESP32 para que sea capaz de detectar la pérdida de conexión con un *router* y automáticamente intentar conectarse al segundo *router*.

#### Paso 2.1

Se definen las constantes de SSID y contraseña del segundo *router*. Además de la variable de estado de conexión a red 1.

Ejemplo de código:

```
const char* ssid2 = "Nombre de red Wi-Fi"; // Corresponde a la variable
"ssid2" del segundo router.
const char* password2 = "Contra123"; // Corresponde a la variable
"password2" del segundo router.
bool conectadoRed1 = true; // estado de conexión a red principal.
```

#### Paso 2.2

En la sección de “void loop ()” se colocan las instrucciones para realizar la conexión del ESP32 a Wi-Fi. En esta ocasión se detalla un ejemplo de la programación del ESP32 en caso de que se pierda la conexión con uno de los *routers*.

Ejemplo de código:

```
//Si se desea utilizar redundancias con 2 routers se debe agregar el
siguiente código.
if (WiFi.status() != WL_CONNECTED) { //condición que verifica la
conectividad a la red.
reconectarWiFi(); // Intenta reconectar si se pierde la conexión
}
```

#### Paso 2.3

Se deben definir los bloques para establecer las conexiones con ambos *routers* en caso de que la conexión se pierda con alguno. Se agregan los bloques de “reconectarWiFi()”, “esperarConexion()” y además se modifica el bloque “conectarWiFi()” para agregar las redundancias.

Ejemplo de código:

```
void conectarWiFi() {

//Se establece conexión con la red 1
WiFi.mode(WIFI_STA);
Serial.println("Conectando a la red 1...");
WiFi.begin(ssid, password);

// Se espera 10 segundos para conectarse a la red 1 sino se intenta la
conexión con la red 2
if (esperarConexion(10000)) {
Serial.println("Conectado a la red 1");
conectadoRed1 = true;
} else {
Serial.println("No se pudo conectar a red 1, intentando red 2...");
WiFi.begin(ssid2, password2);

// Se espera 10 segundos para conectarse a la red 2 sino se indica que
falla la conexión.

if (esperarConexion(10000)) {
if (esperarConexion(10000)) {
Serial.println("Conectado a red 2");
conectadoRed1 = false;
} else {
Serial.println("No se pudo conectar a ninguna red");
}}}

// Si el estado de conexión del bloque "esperarConexion" no es el estado
WL_CONNECTED, según se configuro antes, se ejecuta el bloque
reconectarWiFi().

void reconectarWiFi() {
if (conectadoRed1) {
Serial.println("Reintentando conexión a red 1...");
WiFi.begin(ssid, password);

if (!esperarConexion(5000)) {
```

```

Serial.println("Fallo en red 1, conectando a red 2...");
WiFi.begin(ssid2, password2);
conectadoRed1 = false;
esperarConexion(5000);
}
} else {
Serial.println("Reintentando conexión a red 2...");
WiFi.begin(ssid2, password2);

if (!esperarConexion(5000)) {
Serial.println("Fallo en red 2, conectando a red 1...");
WiFi.begin(ssid, password);
conectadoRed1 = true;
esperarConexion(5000);
}}}

// se ejecuta el bloque para establecer el tiempo de espera en las
conexiones.
bool esperarConexion(int tiempoEspera) {
int inicio = millis();
while (WiFi.status() != WL_CONNECTED && millis() - inicio < tiempoEspera)
{
delay(500);
Serial.print(".");
}
return WiFi.status() == WL_CONNECTED; }

```

Mediante los pasos anteriores es posible definir como el sistema solventará la necesidad de transmisión de los datos obtenidos por los sensores mediante redundancias en caso de que alguno de los *routers* falle, asegurando así que se mantenga el registro de las mediciones.

## Topología de conectividad

El principal objetivo de la tecnología IoT es lograr la interconexión de dispositivos y conectarlos a una red informática como lo es el internet. Para el presente diseño del sistema de monitoreo, los sensores serán los encargados de recopilar la información del cuarto limpio para luego ser transmitido a la red. El encargado de llevar los procesos de transmisión, recolección y conexiones de los sensores será la placa del microcontrolador ESP32.

Dado que el sistema de monitoreo requerirá de varios puntos de muestreo ubicados a lo largo del cuarto limpio, según se explicó anteriormente se hará uso del estándar IEEE 802.11 Wi-Fi, con la finalidad de hacer más flexible la instalación de cada punto de muestreo. La placa ESP32 se utilizará como nodo (un nodo por cada punto de muestreo), para la recopilación y transmisión de los datos obtenidos de los sensores. Al mismo tiempo la placa ESP32 cumplirá la función de *gateway*, ya que facilitará la comunicación entre los sensores y la red Wi-Fi.

La topología de red que se utilizará corresponde a la topología en estrella, donde cada nodo (punto de muestreo) estará conformado por la placa ESP32 y esta a su vez estará conectada a los sensores. De esta manera cada sensor se comunicará con su respectivo ESP32 en una relación maestro/esclavo de modo que cada nodo es independiente y no requiere comunicarse con algún otro nodo para enviar la información. El microcontrolador ESP32 será el punto central encargado de enviar los datos recopilados de manera independiente a la tabla asignada en la base de datos.

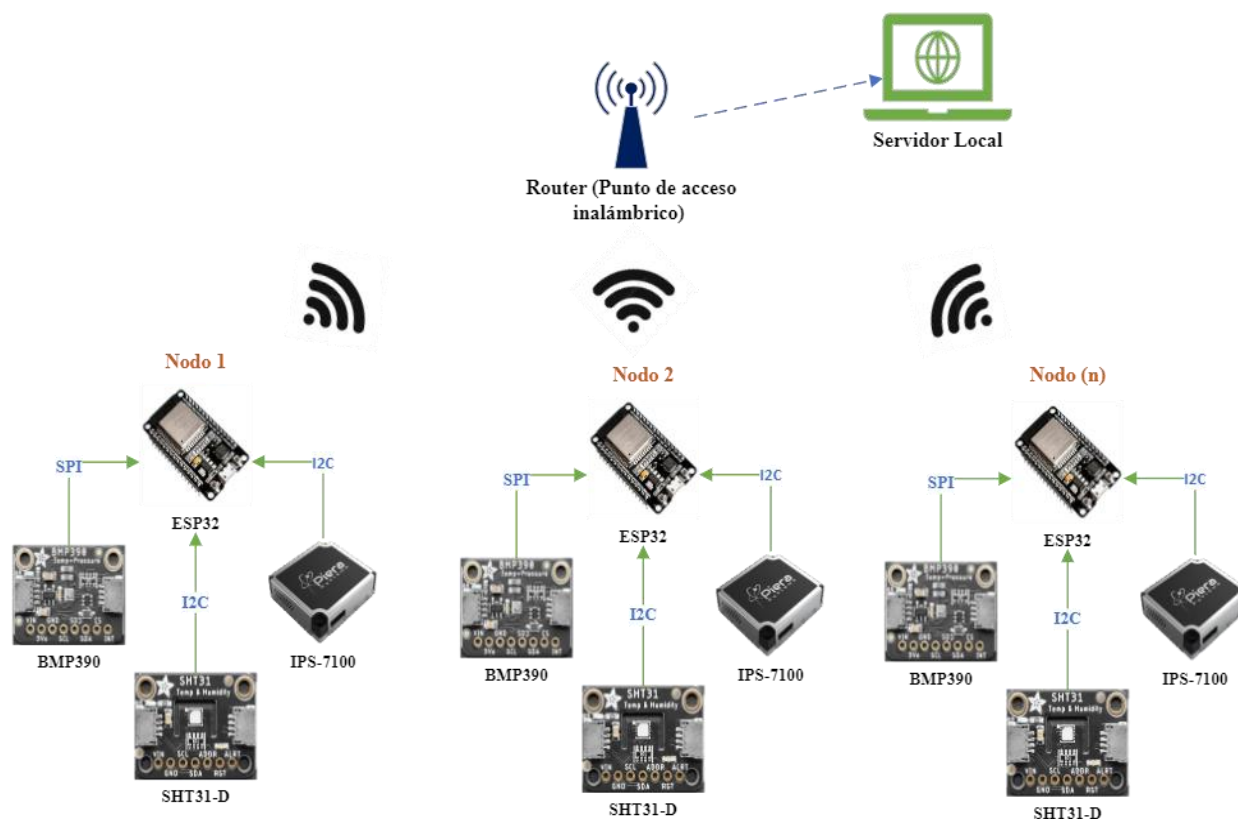


Figura 37. Topología de red del sistema de monitoreo

Fuente: Propia

## Protocolo de comunicación

Para lograr la transmisión de los datos entre los objetos inteligentes y otros sistemas, como lo requiere una red IoT, es necesario establecer el protocolo de comunicación. Para el presente diseño del sistema de monitoreo, el protocolo de comunicación a utilizar entre el microcontrolador ESP32 y el servidor donde serán almacenados los datos de los sensores corresponde al protocolo HTTP (protocolo de transferencia de hiper texto, por sus siglas en inglés). HTTP es un protocolo de capa de aplicación que utiliza TCP/IP para transportar los datos de sus mensajes.

Según se explicó en capítulos anteriores, HTTP se compone de un cliente y un servidor. Para lograr que el cliente, o sea el microcontrolador ESP32 pueda enviar mensajes al servidor, se requiere establecer la conexión TCP/IP entre el cliente y el servidor mediante direcciones del protocolo de internet (IP).

Para lograr obtener la dirección IP, se utiliza el URL (del inglés Uniform Resource Locator o Localizador Uniforme de Recursos) lo cual describe la ubicación/dirección IP específica de un recurso en un servidor particular y cómo acceder a él.

Para el presente diseño, se pretende utilizar la multiplataforma XAMPP de código abierto que funciona en múltiples sistemas operativos. Esta plataforma se utiliza para crear un entorno de servidor local en una computadora, incluyendo un servidor web Apache, un sistema de gestión de bases de datos MySQL y *scripts* para lenguajes PHP.

El servidor web Apache, incluido en XAMPP que se encuentra instalado en la computadora, corresponde a un servidor web HTTP, lo cual ayuda a recibir solicitudes HTTP de los clientes (en este caso el microcontrolador ESP32 de cada punto de muestreo) y procesar las solicitudes. Mediante el uso del URL es posible definir la dirección exacta donde se requieren enviar las solicitudes HTTP del cliente para luego ser procesadas.

El servidor Apache brinda la opción de ejecutar los *scripts* escritos en PHP, por lo que el lenguaje de programación del servidor (o sea la computadora que tiene instalado XAMPP) mediante los *scripts* de PHP, procesará las solicitudes HTTP para interactuar con la base de datos donde se desee almacenar los registros del sistema de monitoreo.

A continuación, se muestra el código que debe ser utilizado para lograr la configuración del microcontrolador ESP32 para iniciar el proceso de comunicación entre el cliente y el servidor mediante HTTP:

Paso 1: incluir en el *sketch* la librería “HTTPClient.h” para agregar el conjunto de funciones necesarias para las solicitudes HTTP.

Ejemplo de código:

```
#include <HTTPClient.h> // Librería para realizar solicitudes HTTP
```

Paso 2: definir la dirección URL en la que se realizarán las solicitudes de “post” con el servidor local.

Ejemplo de código:

```
String URL = "http://dirección_IP_computadora/Nombre_Carpeta_en_XAMPP  
/Nombre_documento_PHP.php"; //Se define la variable URL, como la dirección  
de servidor local.
```

Paso 3: en la sección de “void loop ()” se debe programar los comandos para realizar las solicitudes HTTP entre el microcontrolador y el servidor local.

Ejemplo de código:

```
void loop() {

HTTPClient http;
http.begin(URL); //Inicializar conexión con URL
http.addHeader("Content-Type", "application/x-www-form-urlencoded");
//Encabezado de la solicitud http.
int httpCode = http.POST(postData); // Realiza solicitud HTTP Post con los
datos contenidos en la variable "post data"
String payload = http.getString(); //Devuelve código de respuesta http,
valor "200" indica solicitud exitosa.

// Se imprime en el monitor serial URL, Datos enviados y código http de
respuesta del servidor.
Serial.print("URL: "); Serial.println(URL);
Serial.print("Data: "); Serial.println(postData);
Serial.print ("httpCode: "); Serial.println(httpCode);
Serial.print ("payload: "); Serial.println (payload);
Serial.println ("-----");

}
```

Utilizando el código anterior es posible lograr la configuración del microcontrolador ESP32 para realizar solicitudes HTTP al servidor local, permitiendo así publicar en el servidor los

datos obtenidos por el sistema de monitoreo. Es importante recordar que el sistema de monitoreo enviará los datos a una computadora que contiene instalado el programa XAMPP. La computadora estará configurada de manera que su dirección IP no cambie con el tiempo, incluso si fuese apagada, de esta manera se asegura que el servidor contará con una dirección IP estática y no será necesario actualizar constantemente la dirección IP en el código de cada nodo del sistema de monitoreo.

En la siguiente figura se muestra el diagrama de flujo para la conectividad entre el cliente y el servidor local mediante HTTP.



Figura 38. Diagrama de flujo de conectividad HTTP cliente-servidor local

Fuente: Propia

## Sistema de almacenamiento

Para el presente diseño del sistema de monitoreo, se requiere definir el sistema encargado del almacenamiento de los datos obtenidos por los sensores los cuales fueron enviados mediante el protocolo HTTP desde la placa microcontroladora hasta el servidor local. Como se mencionó anteriormente, se utilizará la multiplataforma XAMPP, la cual estará instalada en la computadora que cumple la función de servidor local, y se encargará de gestionar la base de datos MySQL.

En la plataforma XAMPP es posible diseñar la base de datos, en la que serán almacenados los datos de los sensores. A continuación, se listan los pasos para diseñar la base de datos:

Paso 1. Designar un nombre para la base de datos

Paso 2. Crear la tabla en las que se almacenaran los datos. En esta sección, es posible crear una tabla para almacenar los datos de cada posición (nodo) de muestreo en el cuarto limpio. De esta

manera es posible manejar tablas de datos específicos para cada posición, permitiéndose posteriormente realizar análisis de tendencias de datos de manera específica por posición como a nivel general del cuarto limpio.

En la siguiente figura se muestra, como se crea la base de datos en XAMPP, además de cómo se crean las tablas para cada nodo o punto de muestreo.

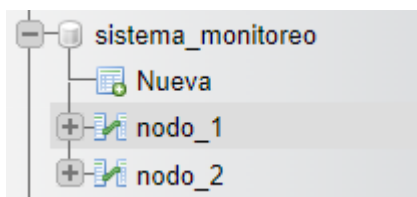


Figura 39. Esquema de base de datos “Sistema\_monitoreo”.

Fuente: Propia

Paso 3. Se debe diseñar en el programa XAMPP cada una de las tablas, indicando los nombres de las columnas y el tipo de dato asociado a la variable que se está midiendo (ya sea si corresponde a una fecha, texto, valores enteros o decimales, entre otros).

En la siguiente imagen se muestra un ejemplo de cómo se vería la tabla con los nombres asignados a cada columna para las variables que se desea monitorear con el sistema.

ID	Temperatura	Humedad	Presion	PM0p5	PM1p0	PM5	PC0p5	PC1p0	PC5	Fecha
168	27	65	98.07	22027	21979	50830	110207	156874	176154	2024-06-30 18:49:12
169	27	62	98.07	22008	21972	50771	110171	156871	176381	2024-06-30 18:50:12

Figura 40. Diseño de base de datos.

Fuente: Propia

Paso 4. Se debe crear entre las carpetas de instalación del programa XAMPP el código PHP, el cual actúa como intermediario entre las solicitudes HTTP y la base de datos MySQL. El código PHP se encarga de procesar y ejecutar las consultas a la base de datos. A continuación, se muestra cómo se relaciona el código PHP con el código en Arduino IDE instalado en el ESP32.

Paso 4.1. Primeramente, en el código de Arduino IDE se deben ingresar las solicitudes HTTP para publicar en la base de datos los valores obtenidos por los sensores.

Ejemplo del código:

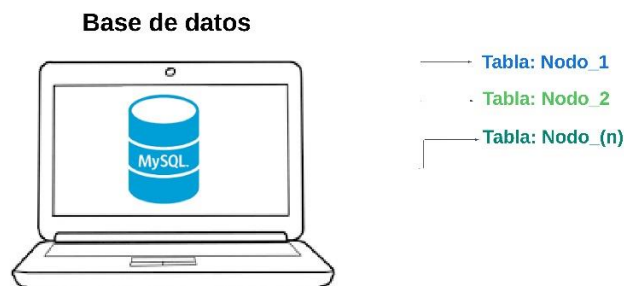
```
//Se realiza la solicitud Postdata a la base de datos.
String postData = "Temperatura=" + String(temperatura) + "&Humedad="
+ String(humedad) + "&Presion=" + String(presion) + "&PM0p5=" +
String(PM0p5) + "&PM1p0=" + String(PM1p0) +
"&PM5=" + String(PM5) + "&PC0p5=" + String(PC0p5) + "&PC1p0=" +
String(PC1p0) + "&PC5=" + String(PC5);
```

Paso 4.2. Seguidamente, se deben agregar las instrucciones en el código PHP, para procesar las solicitudes HTTP. Ver anexo 2 para código PHP completo.

Ejemplo del código:

```
//Se realiza la función de insertar en la tabla nodo_1, los valores
de las variables definidas para cada columna de la tabla de la base
de datos.
$sql = "INSERT INTO nodo_1 (Temperatura, Humedad, Presion, PM0p5,
PM1p0, PM5, PC0p5, PC1p0, PC5) VALUES (". $t.", ". $h.", ". $p.",
". $pm05.", ". $pm1.", ". $pm5.", ". $pc05.", ". $pc1.", ". $pc5.")";
```

Una vez completados los pasos anteriores mediante el diseño de la base de datos, creación de los códigos en Arduino IDE y PHP, es posible conectar el sistema de monitoreo a un servidor local que almacenará los datos de manera permanente, llevando un registro de las fechas y tiempos en que se efectuaron las mediciones para cada nodo. Además, que la plataforma XAMPP, ofrece las funciones de exportar los datos de la base de datos a hojas de Excel o *softwares* de análisis estadísticos, para luego poder ser filtrados, analizados e incluso graficados para estudiar el comportamiento de las condiciones ambientales del cuarto a lo largo del tiempo.



*Figura 41. Distribución de base de datos.*

*Fuente: Propia*

## **Visualización de los datos**

Como parte del diseño del sistema de monitoreo, se selecciona la plataforma de código abierto Grafana diseñada para la visualización de métricas y datos. Su versatilidad permite generar diversos tipos de paneles de control, conocidos como *dashboards* en inglés.

La plataforma Grafana debe ser instalada en el servidor local, de manera que le sea posible extraer los datos almacenados en la base de datos diseñada anteriormente en XAMPP. Para el diseño de los *dashboards* es posible definir a cada gráfico que columna/variable de la base de datos corresponde, además que brinda la opción de graficar datos en un rango de periodo de fechas y asignar límites de control para establecer ayudas visuales, cuando se obtengan valores fuera de las especificaciones predefinidas por el usuario.

En la plataforma de Grafana, será posible observar un *dashboard* para las variables monitoreadas dentro del cuarto limpio en cada uno de los nodos/puntos de muestreo, que contiene la siguiente información:

- Promedio de temperatura, humedad relativa, presión barométrica. Promedio de concentración de partículas de 0.5  $\mu\text{m}$ , 1.0  $\mu\text{m}$ , 5  $\mu\text{m}$  y cantidad de partículas de 0.5  $\mu\text{m}$ , 1.0  $\mu\text{m}$ , 5  $\mu\text{m}$ .
- Indicación del valor más alto obtenido en un periodo de tiempo definido por el usuario para las variables de temperatura, humedad relativa, concentración de partículas de 0.5  $\mu\text{m}$ , 1.0  $\mu\text{m}$ , 5  $\mu\text{m}$  y cantidad de partículas de 0.5  $\mu\text{m}$ , 1.0  $\mu\text{m}$ , 5  $\mu\text{m}$ .
- Indicación del valor más bajo obtenido en un periodo de tiempo definido por el usuario para la variable de presión barométrica en los puntos que se desee medir.

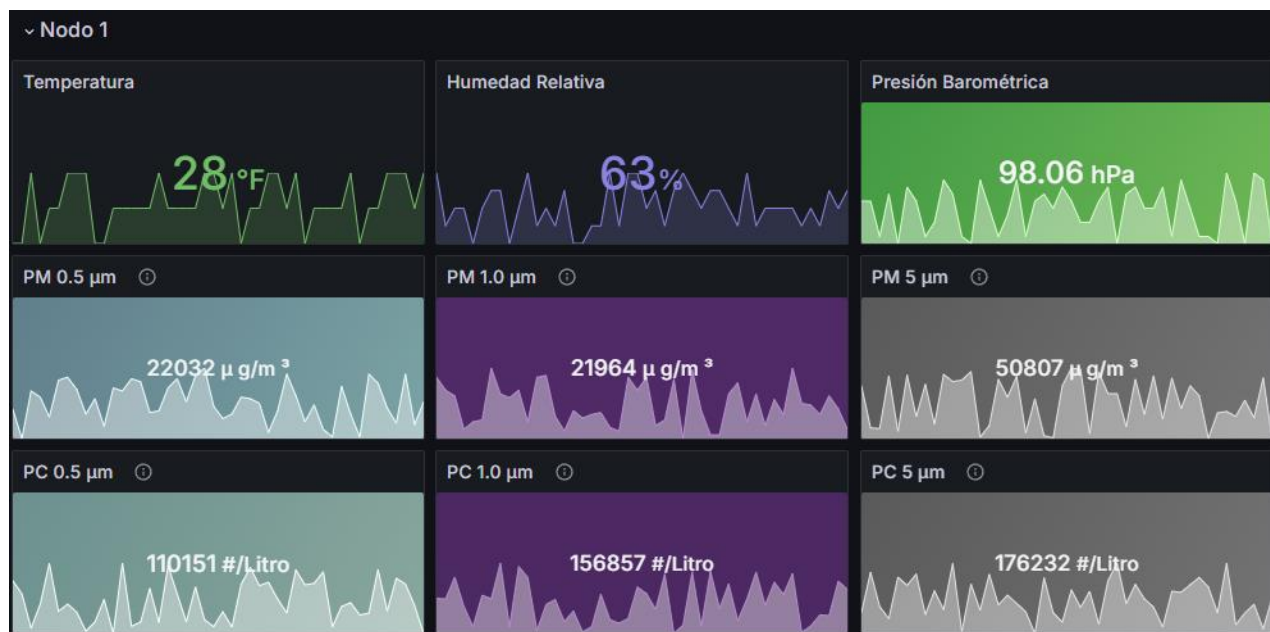


Figura 42. Dashboard de Promedios para nodo 1.

Fuente: Propia

## Análisis financiero

Como parte del análisis financiero del sistema de monitoreo, se considerará primeramente los costos de implementación del sistema de monitoreo, tomando en cuenta componentes y materiales requeridos para el montaje de un único nodo o punto de muestreo.

### Materiales requeridos para cada nodo o punto de muestreo

Componente	Proveedor	Cantidad	Precio en (\$)	Precio en (₡)
Sensor de temperatura y humedad Relativa, SHT31-D.	Adafruit	1	13,95	7323,75
Sensor de presión barométrica, BMP390.	Adafruit	1	10,95	5748,75
Sensor contador de partículas en el aire, IPS-7100.	Piera Systems	1	71,10	37327,5
Microcontrolador ESP32-WROOM-32	CRCibernética	1	14,95	7848,75
Escudo de terminales de tornillo para microcontrolador ESP32	CRCibernética	1	11,95	6273,75

<b>Componente</b>	<b>Proveedor</b>	<b>Cantidad</b>	<b>Precio en (\$)</b>	<b>Precio en (₡)</b>
<b>Adaptador de corriente, cargador USB 5V DC, 2A</b>	Amazon	1	7,99	4194,75
<b>Cable USB Tipo micro-B</b>	Amazon	1	6,99	3669,75
<b>Tornillos Torlak punta broca 8 x 1/2</b>	Constru Plaza	1	0,74	400
<b>Total</b>			138,62	72787

*Tabla 23. Costo de componentes requeridos para el diseño del sistema de monitoreo.*

*Fuente: Propia elaborado en Excel.*

En la tabla 23 es posible observar que el costo de implementación de un punto de muestreo es de 138.62 \$ o su equivalente a ₡72 787 (setenta y dos mil setecientos ochenta y siete colones) tomando en cuenta el tipo de cambio del dólar a la fecha 7 de julio del 2024, de ₡525. Además, es importante considerar que inicialmente se debe hacer la inversión de una computadora que funcionará como servidor local para todos los nodos y la implementación de un *switch* y al menos 2 *routers* para las redundancias de la red. La computadora, *switch* y *routers* seleccionados corresponden a los que se muestran en la siguiente tabla:

<b>Componente</b>	<b>Proveedor</b>	<b>Cantidad</b>	<b>Precio en (\$)</b>	<b>Precio en (₡)</b>
<b>Computadora ACER Aspire 5, Core i5, 16GB RAM, 512GB Memoria</b>	extremetech	1	588,57	309 000
<b>SWITCH RG-ES208GC GIGABIT - 8 PUERTOS</b>	RUIJIE	1	40	21 000
<b>ROUTER EX220 4 Puertos Gigabit LAN Ethernet. Compatible con VLAN, Dynamic IP/Static.</b>	TP-LINK	2	122	64 000

*Tabla 24. Computadora utilizada como servidor local.*

*Fuente: Propia elaborado en Excel.*

### Costos de programación e instalación del sistema de monitoreo

Con el objetivo de detallar los costos asociados a la programación e instalación de cada punto de muestreo, se ha consultado con el personal de una empresa de manufactura de dispositivos médicos. Esto incluye a un ingeniero especializado en ciencias de la computación y a un técnico en automatización, quienes proporcionaron una cotización detallada para la implementación de un nodo o punto de muestreo individual. En la tabla 25 se encuentra el detalle de la cotización.

Como dato importante, el salario promedio del técnico de automatización de la empresa corresponde a ₡550 000 (quinientos cincuenta mil colones) mensuales, y revisando según su contrato laboral el pago por hora es de ₡2750 (dos mil setecientos cincuenta colones). El técnico asegura que por cada nodo y dependiendo de su ubicación y configuración, en promedio le tomaría 4 horas de trabajo instalar un solo punto de muestreo. En la siguiente tabla se muestra el desglose:

Actividad	Proveedor	Precio en (\$)	Precio en (₡)	Precio en (\$)	Precio en (₡)
		Costo Inicial		Costo por cada nodo adicional	
Programación del microcontrolador y sensores en Arduino IDE	Profesional de ciencias de la computación	571,43	300000	19.05	10000
Diseño de la base de datos en XAMPP		761,90	400000	38.10	20000
Diseño de la interfaz gráfica en Grafana		190.48	100000	15.24	8000
Instalación del nodo o punto de muestreo	Técnico en automatización	20.95	11000	20.95	11000
<b>Total</b>		1544.76	811000	93.34	49000

Tabla 25. Costos de programación e instalación del sistema de monitoreo.

Fuente: Propia, elaborado en Excel.

### Costos por consumo eléctrico del sistema de monitoreo

Según se calculó anteriormente, el consumo eléctrico de cada nodo o punto de muestreo del sistema de monitoreo es de 1.525 W. Por lo que utilizando la ecuación (6), es posible determinar el consumo en Kilowatts-hora (KWh) para cada nodo.

$$E_{(KWh)} = P_{(KW)} \times t_{(h)}$$

$$E_{(KWh)} = \left( \frac{1.525W}{1000} \right) \times 1 \text{ hr}$$

$$E_{(KWh)} = 0.0015 \text{ KWh}$$

Tomando en cuenta las tarifas del Instituto Costarricense de Electricidad (ICE) publicados en su página web, establece que la tarifa T-IN Industrial corresponde a ₡127.11 por cada KWh. (Instituto Costarricense de Electricidad, 2023)

<b>Para consumos menores o iguales que 3 000 kWh</b>	
Por cada kWh	₡ 127.11
<b>Para consumos mayores de 3 000 kWh</b>	
Cargo por energía, por cada kWh	₡ 76.07
Cargo por potencia, por cada kW	₡ 12 571.65

Figura 43. Tarifas de consumo eléctrico, ICE.

Fuente: Instituto Costarricense de Electricidad, 2023

Tomando en cuenta la información anterior, el costo por consumo eléctrico del sistema de monitoreo es de ₡0.19 (diecinueve céntimos de colones) por KWh, suponiendo que va a operar las 24 horas del día, su costo mensual suponiendo un mes de 30 días es de ₡137.30 (ciento treinta y siete colones con treinta céntimos). Esto corresponde a un costo anual de ₡1647.6 (mil seiscientos cuarenta y siete colones con sesenta céntimos)

### Costo actual del sistema de monitoreo

Los costes de implementación total actual del sistema de monitoreo para un único nodo de monitoreo corresponden a ₡1 277 787 (un millón doscientos setenta y siete mil setecientos ochenta y

siete colones), sin embargo, corresponde solo al costo inicial dado que para los siguientes nodos adicionales el costo se reduce por cada uno hasta ¢121 787 (ciento veinte mil setecientos ochenta y siete colones). En la siguiente tabla se muestra el desglose de los costos:

Actividad	Costo inicial (Total)		Costo por cada nodo adicional (Total)	
	Precio en (\$)	Precio en (¢)	Precio en (\$)	Precio en (¢)
<b>Componentes requeridos</b>	138.62	72787	138.6	72787
<b>Computadora (servidor local)</b>	588,57	309000	N/A	N/A
<b>1 Switch</b>	40	21 000	N/A	N/A
<b>2 Routers</b>	122	64 000	N/A	N/A
<b>Programación e instalación</b>	1544.76	811000	93.34	49000
<b>Total</b>	2433.95	1277787	231.94	121787

Tabla 26. Costos actuales totales del sistema de monitoreo.

Fuente: Propia, elaborado en Excel.

### Costo de mantenimiento del sistema de monitoreo

Para el mantenimiento del sistema de monitoreo, se ha consultado con el ingeniero en ciencias de la computación y técnico en automatización, por el costo de correcciones o actualizaciones del código instalado en cada sensor, además del costo por el mantenimiento o ajustes a la base de datos y el sistema de visualización en Grafana. En la siguiente tabla se muestra el desglose de los costos:

Mantenimientos	Proveedor	Costo bimensual	
		Precio en (\$)	Precio en (¢)
Actualización en código de Arduino IDE para los sensores	Profesional de ciencias de la computación	5.71	3000
Mantenimiento de base de datos en XAMPP		19.05	10000
Ajustes/Actualizaciones de interfaz gráfica en Grafana		5.71	3000
Ajustes/limpieza de cada nodo, realizado por técnico en automatización	Técnico en automatización	2.62	1375
<b>Total</b>		33.09	17375

*Tabla 27. Costos de mantenimiento bimensual del sistema de monitoreo.*

*Fuente: Propia, elaborado en Excel.*

En la tabla anterior es posible observar que el costo por mantenimiento bimensual es de ₡17 375 (diecisiete mil trescientos setenta y cinco colones) por lo que los costos anuales por mantenimiento corresponden a ₡104 250 (ciento cuatro mil doscientos cincuenta colones).

### **Cálculo del VAN y TIR**

Para el cálculo del valor actual neto (VAN), es importante considerar los costos actuales que pueden tener un impacto positivo mediante la implementación del sistema de monitoreo.

Entre las ventajas de la implementación del sistema de monitoreo se sabe que se pueden reducir los tiempos utilizados por el personal para la toma y registro de los resultados de las mediciones de temperatura, humedad relativa, presión y cantidad de partículas suspendidas en el aire para los tamaños de 0.5  $\mu\text{m}$ , 1.0  $\mu\text{m}$ , 5  $\mu\text{m}$ . Según se observó en una empresa de manufactura de dispositivos médicos que no cuenta con un sistema de monitoreo automatizado y se identifica que utilizan dos operarios de producción para la toma de cantidad de partículas suspendidas en el aire, donde un operario se encarga de operar el medidor de partículas y otro se encarga de registrar los datos. Por cada punto de muestreo se tardan al menos 10 minutos. Tomando en cuenta que el salario por hora de cada operario es de ₡2350 (dos mil trescientos cincuenta colones), representando un costo por los dos operarios en una sola medición de ₡783.33 (setecientos ochenta y tres colones con treinta y tres céntimos).

Además, se identifica que al final de mes se debe asignar el recurso de un ingeniero de calidad para verificar y registrar en un Excel los resultados de las mediciones realizadas, para posteriormente ser aprobadas por el ingeniero. Consultando en la empresa, el ingeniero tiene un salario por hora de ₡5200 (cinco mil doscientos colones), e informa que le toma al menos 4 minutos verificar y registrar un solo punto de muestreo, lo cual es equivalente a ₡346 (trescientos cuarenta y seis colones) por cada punto de muestreo.

Tomando en cuenta los costos anteriores, si se implementara el sistema de monitoreo para tomar el registro de las variables en un solo nodo, en un total de 4 veces al día con una diferencia de 6 horas entre cada medición, actualmente representaría un ahorro anual de ₡1 626 235 (un millón seiscientos veinte seis mil doscientos treinta y cinco colones) por cada nodo, tal y como se desglosa a continuación:

$$[(\text{Costo de 2 operarios} \times 4 \text{ mediciones/día}) + (\text{Costo de Ing Calidad} \times 4 \text{ mediciones/día})] \times 30 \text{ días}$$

$$\text{Ahorro mensual} = [(783.33 \times 4) + (346 \times 4)] \times 30 \text{ días} = \text{¢}135519.6$$

$$\text{Ahorro anual} = 135519.6 \times 12 \text{ meses}$$

$$\text{Ahorro anual} = \text{¢}1\,626\,235$$

Con los valores anteriores y tomando como referencia los datos del Banco de Costa Rica (BCR), si se colocara el monto en un fondo de inversión a plazo en un periodo previsto de un año, la tasa de interés de inversión a plazo corresponde a 5.69% en el BCR.

Para el cálculo del VAN, colocando los valores anteriores se obtiene lo siguiente:

$$\text{VAN} = -1\,277\,787 + \frac{1\,626\,235 - 104\,250 - 1\,647.6}{(1 + 0.0569)^1} = 160700.46$$

$$\text{VAN} = 160\,700.46$$

Al obtenerse un valor positivo para el valor actual neto, se evidencia que el presente proyecto es viable y permite recuperar la inversión de cada nodo en un plazo menor a un año.

Con el fin de evaluar la rentabilidad y viabilidad del sistema de monitoreo, en la siguiente fórmula se muestran los cálculos de la tasa interna de retorno (TIR), que ayudará a determinar el potencial de retorno de la inversión.

$$\text{TIR} = 0 = -1\,277\,787 + \frac{1\,626\,235 - 104\,250 - 1\,647.6}{(1 + i)^1}$$

$$\text{TIR} = 18.9\%$$

Para el presente análisis financiero, los resultados obtenidos confirman la viabilidad del diseño del sistema de monitoreo en términos de costos iniciales de implementación. Los cálculos indican que es posible recuperar los costos de inversión en menos de un año para cada nodo o punto de muestreo instalado. Además, con una tasa interna de retorno del 18.9% significativamente superior a las tasas de interés del 5.69% del banco, se demuestra la rentabilidad financiera del proyecto.

## CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

- El sistema de monitoreo propuesto representa una solución tecnológica para el monitoreo de cuartos limpios utilizados para la fabricación de dispositivos médicos. Incluyendo la capacidad para verificar variables de temperatura, humedad relativa, presión, y cantidad de partículas suspendidas en el aire para los tamaños de 0.5  $\mu\text{m}$ , 1  $\mu\text{m}$  y 5  $\mu\text{m}$ .
- El estudio de mercado y análisis técnico de los sensores y microcontroladores disponibles en el mercado actual permitió definir las características y capacidades de los sensores para ser adaptados entre sí para formar la red de monitoreo. Permitiendo así definir los protocolos de comunicación utilizados para la transmisión de los datos y programación de estos.
- Entender las necesidades del sistema de monitoreo para cuartos limpios fue clave para definir y establecer una topología de red IoT. Esta topología permite la interconexión de sensores y ofrece un diseño totalmente escalable, adaptable a diferentes puntos de monitoreo según las necesidades del usuario. Además, que al utilizar tecnologías de acceso a internet como Wi-Fi, lo vuelve un diseño versátil reduciendo el cableado para la transmisión de los datos.
- Para satisfacer la necesidad de un servidor local al que se puedan enviar los datos obtenidos por los sensores, se propone utilizar la plataforma XAMPP en el diseño. Basado en esta selección, se definió el uso del protocolo de comunicación HTTP para el envío de datos desde cada nodo al servidor local. La plataforma XAMPP permite diseñar y administrar la base de datos en MySQL, proporcionando así un sistema de almacenamiento en el que se puedan consultar los registros.
- De acuerdo con el objetivo planteado, se seleccionó la plataforma Grafana para extraer los registros de la base de datos y mostrarlos en *dashboards* que pueden ser diseñados y ajustados de acuerdo con las necesidades del usuario, facilitando la visualización de las condiciones ambientales del cuarto, tanto en tiempo real como en periodos de tiempo definidos por el usuario.
- Los costos de implementación del sistema de monitoreo fueron calculados, obteniéndose un costo inicial de ₡1 277 787 para cada nodo o punto de muestreo. Los

análisis del VAN y TIR indican que es viable recuperar la inversión en menos de un año por cada nodo o punto de muestreo instalado.

### **Recomendaciones**

- El diseño actual del sistema de monitoreo permite monitorear la presión barométrica actual en cada uno de los nodos o puntos de muestreo, registrándolo en su respectiva tabla en la base de datos y permitiendo graficar el valor histórico más bajo y el promedio de los valores en un periodo de tiempo determinado. Si se desea registrar la presión diferencial entre puntos del cuarto limpio directamente en la base de datos, es posible realizarse en la plataforma XAMPP. Sin embargo, se recomienda solicitar ayuda de un profesional en el área que se encargue de configurar el código de la base de datos, asegurando cuestiones importantes como el hecho que se tomen las mediciones en ambos nodos al mismo tiempo.
- Dado que la plataforma de XAMPP permite extraer registros de la base de datos en periodos definidos por el usuario, se recomienda como buena práctica crear una plantilla en Excel o utilizar *software* especializado para el análisis de datos. Esto facilitará el análisis detallado de los datos para realizar cálculos estadísticos y fórmulas predefinidas que simplifiquen el proceso de análisis para mejorar la automatización del sistema de monitoreo.
- Dado que la plataforma de Grafana permite desarrollar un sistema de alarmas para monitorizar datos en tiempo real, además que permite su integración con otros gestores de alarmas que permitan la recepción, clasificación y notificación de alarmas. Se recomienda trabajar en conjunto con el equipo de tecnología de la información (TI) y Calidad de la empresa para establecer los límites de control y configurar canales de notificación de alarmas. Esto podría lograrse enviando notificaciones a través del correo institucional de la empresa, dirigidas a las personas responsables del monitoreo de los cuartos limpios.
- Mediante la implementación de TCP/IP y la integración de protocolos como Modbus o SNMP (Protocolo Simple de Gestión de Red, por sus siglas en inglés) es posible permitir al sistema que se acceda y controle de manera remota permitiendo la monitorización desde ubicaciones remotas a la empresa.

## CAPÍTULO VI: PROPUESTA

La propuesta del presente proyecto de graduación corresponde al diseño de un sistema de monitoreo para las variables de temperatura, humedad relativa, presión y cantidad de partículas dispersas en el aire, para el monitoreo de cuartos de ambiente controlado utilizados para la fabricación de dispositivos médicos, en los que es de suma importancia mantener registros de las variables, según los requerimientos del usuario.

El presente sistema de monitoreo IoT ofrece las siguientes ventajas:

- Diseño totalmente escalable: esto permite ser adaptable a distintos tamaños de cuartos controlados mediante la implementación de nodos de sensores, que funcionan de manera independiente y son fáciles de integrar con otros módulos.
- Monitoreo de áreas específicas: la implementación de nodos de sensores, permite obtener datos específicos de distintas áreas del cuarto controlado, por lo que es posible identificar los puntos en los que se compromete las condiciones ideales, permitiendo así optimizarlo.
- Tecnología de acceso a la red: el sistema de monitoreo mantiene una conectividad inalámbrica por medio de Wi-Fi entre sus nodos y el servidor local, siendo fácil de implementar en empresas simplificando la instalación al reducir el cableado.
- Sistema de almacenamiento de datos: al utilizar la plataforma XAMPP de código abierto es posible diseñar y administrar la base de datos en un servidor local como lo puede ser una computadora donde sea posible consultar y exportar la información de los registros obtenidos del sistema para análisis estadísticos y tendencias de datos.
- Sistema de visualización: mediante la plataforma de código abierto Grafana es posible obtener *dashboards* representativos de los datos almacenados en la base de datos en tiempo real, además que permite filtrar los periodos de tiempo en que se consultan los registros.
- Plataformas de código abierto: al implementar plataformas de código abierto como Arduino IDE, XAMPP y Grafana permite adaptar el sistema de monitoreo a las necesidades de cada usuario. Además, que reduce la limitación de pago de licencias para su uso.

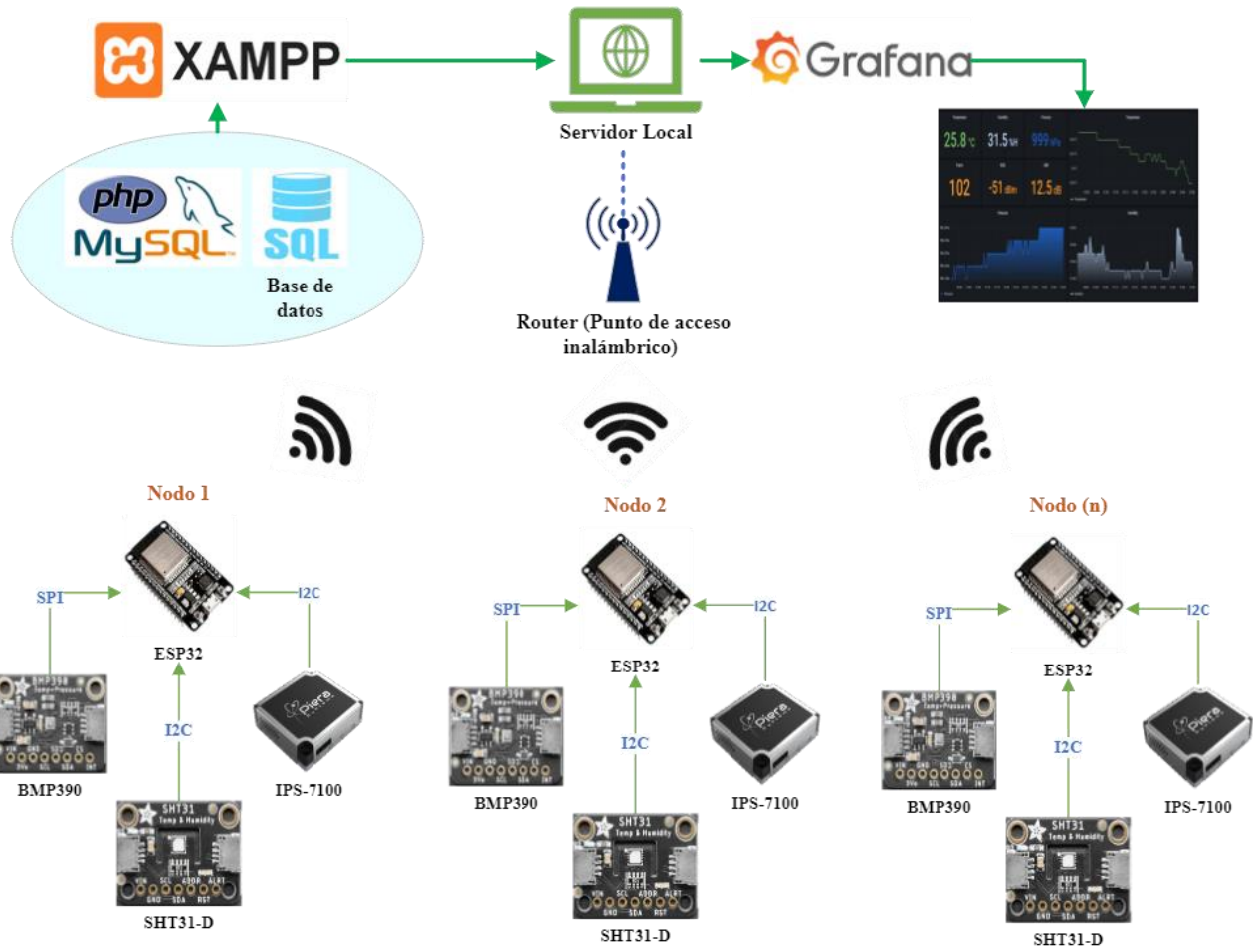


Figura 44. Propuesta de sistema de monitoreo.

Fuente: Elaboración propia en Visio.

## Referencias bibliográficas:

- Adafruit. (8 de diciembre de 2015). *Arduino Code*. Learn.adafruit. <https://learn.adafruit.com/adafruit-sht31-d-temperature-and-humidity-sensor-breakout/wiring-and-test>
- \_\_\_\_\_. (Octubre de 2022). Adafruit LPS22 Pressure Sensor. Adafruit. <https://www.adafruit.com/product/4633>
- \_\_\_\_\_. (Noviembre de 2022). Adafruit BMP280 I2C or SPI Barometric Pressure y Altitude Sensor. Adafruit. <https://www.adafruit.com/product/2651>
- \_\_\_\_\_. (18 de setiembre de 2023). Adafruit BME280 I2C or SPI Temperature Humidity Pressure Sensor. Adafruit. <https://www.adafruit.com/product/2652>
- \_\_\_\_\_. (6 de febrero de 2023). *Adafruit BMP390 - Precision Barometric Pressure and Altimeter - STEMMA QT*. Adafruit. <https://www.adafruit.com/product/4816>
- Allen, T. (2003). *Powder Sampling and Particle Size Determination*. ELSEVIER.
- Amador Álvarez, A. (2017). Diseño de un sistema electrónico inalámbrico de adquisición de datos para la toma de decisiones en un sistema hidropónico de lechuga en un invernadero del TEC sede San Carlos [tesis de licenciatura, Tecnológico de Costa Rica]. [https://repositoriotec.tec.ac.cr/bitstream/handle/2238/9857/dise%C3%B1o\\_sistema\\_electr%C3%B3nico\\_inal%C3%A1mbrico\\_adquisici%C3%B3n\\_de\\_datos\\_toma\\_de\\_decisiones\\_sistema\\_hidrop%C3%B3nico\\_lechuga\\_invernadero\\_tec\\_sede\\_san\\_carlos.pdf?sequence=1&isAllowed=y](https://repositoriotec.tec.ac.cr/bitstream/handle/2238/9857/dise%C3%B1o_sistema_electr%C3%B3nico_inal%C3%A1mbrico_adquisici%C3%B3n_de_datos_toma_de_decisiones_sistema_hidrop%C3%B3nico_lechuga_invernadero_tec_sede_san_carlos.pdf?sequence=1&isAllowed=y)
- AOSONG. (2020). *AM2301 temperature and humidity sensor*. Guangzhou Aosong Electronic Co. <http://www.aosong.com/en/products-28.html>
- AOSONG. (2021). *Guangzhou Aosong Electronics Co*. [www.aosong.com](http://www.aosong.com)
- Arduino. (05 de febrero de 2018). *What is Arduino?*. Arduino. <https://www.arduino.cc/en/Guide/Introduction>
- \_\_\_\_\_. (16 de octubre de 2023). *Overview of the Arduino UNO Components*. Arduino . [https://docs.arduino.cc/tutorials/uno-rev3/intro-to-board?queryID=63f901eec8b897e42522d6ec3cec425e&\\_gl=1\\*14vdl5n\\*\\_ga\\*NzA1OTgzMzc5LjE2OTMwOTYxOTk.\\*\\_ga\\_NEXN8H46L5\\*MTcwMjAxMDEwNS4xMi4xLjE3MDIwMTA3MzkuMC4wLjA.\\*\\_fplc\\*bTBBbjB2V3E4YmlCTDVnUW5YMWVmNXhHeGhHaTINcGd](https://docs.arduino.cc/tutorials/uno-rev3/intro-to-board?queryID=63f901eec8b897e42522d6ec3cec425e&_gl=1*14vdl5n*_ga*NzA1OTgzMzc5LjE2OTMwOTYxOTk.*_ga_NEXN8H46L5*MTcwMjAxMDEwNS4xMi4xLjE3MDIwMTA3MzkuMC4wLjA.*_fplc*bTBBbjB2V3E4YmlCTDVnUW5YMWVmNXhHeGhHaTINcGd)
- \_\_\_\_\_. (16 de octubre de 2023). *Arduino Integrated Development Environment (IDE) v1* . Arduino. <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics>
- \_\_\_\_\_. (2023). *Downloads. Arduino IDE 2.3.3*. Arduino. <https://www.arduino.cc/en/software>
- \_\_\_\_\_. (2023). *¿Qué es Arduino?*. ARDUINO.cl. <https://arduino.cl/que-es-arduino/>
- Banzi, M., y Shiloh, M. (2015). *Make: Getting Started with Arduino*. (3ra). Maker Media.
- Beuchat, R., Depraz, F., Guerrieri, A., y Kashani, S. (2021). *Fundamentals of System-on-Chip Design on Arm Cortex-M Microcontrollers*. arm Education Media.

- Berrocal Flores, J. (2021). Implementación de microcontroladores electrónicos de bajo costo en la adquisición de deformaciones unitarias en estructuras [tesis de licenciatura, Universidad de Costa Rica]. <https://repositorio.sibdi.ucr.ac.cr/server/api/core/bitstreams/3121d532-f444-4f02-a350-b813dcc7c85f/content>
- Bhatti, S. (2022). *HashDork*. What is Mesh Network Topology?. <https://hashdork.com/what-is-mesh-network-topology/>.
- BOSCH. (Marzo de 2021). Pressure sensor BMP390. Bosch Sensortec. <https://www.bosch-sensortec.com/products/environmental-sensors/pressure-sensors/bmp390/>
- \_\_\_\_\_. (Enero de 2022). Humidity sensor BME280. Bosch Sensortec. <https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/>
- \_\_\_\_\_. (Octubre de 2021). *BMP280 Digital Pressure Sensor*. Bosch Sensortec. <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmp280-ds001.pdf>
- Carvajal González, C. (2015). Sistema electrónico para el control de riego y adquisición de datos en la monitorización de temperatura, humedad y luminosidad de un invernadero [tesis de licenciatura, Instituto Tecnológico de Costa Rica]. <https://repositoriotec.tec.ac.cr/handle/2238/7059>
- Charles, A., y Matthew, S. (2013). *Fundamentos de circuitos eléctricos*. McGRAW-HILL. <https://www.latecnicalf.com.ar/descargas/material/electrotecnia/Fundamentos%20de%20circuitos%20el%C3%A9ctricos,%205ta.%20Edici%C3%B3n%20-%20Charles%20K.%20Alexander.pdf>
- Corona, L., Abarca, G., y Mares, J. (2014). *Sensores y Actuadores, Aplicaciones con Arduino*. GRUPO EDITORIAL PATRIA.
- CRCibernética. (2023). Optical Dust Sensor - GP2Y1010AU0F. CRCibernética. <https://www.crcibernetica.com/optical-dust-sensor-gp2y1010au0f/>
- CRCibernética. (2023). ESP32 Wifi and Bluetooth Dev Board. CRCibernética. <https://www.crcibernetica.com/esp32-wifi-and-bluetooth-dev-board/>
- Elmasri, R., y Navathe, S. (2016). *Fundamentals of Database Systems*. (7ma). Pearson. <https://www.auhd.edu.ye/upfiles/elibrary/Azal2020-01-22-12-28-11-76901.pdf>
- ESPRESSIF. (s.f.). *ESP32-DevKitC V4 Getting Started Guide*. ESPRESSIF. <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html>
- Garg, V. K. (2007). *Wireless Communications and Networking*. Elsevier. <https://www.iqytechnicalcollege.com/Wireless%20Communications%20and%20Networking.p>
- Guillen Gomez, Irani. (2021). Modelo de implementación para el monitoreo y control de condiciones ambientales basados con tecnologías de Arduino y Raspberry [tesis de maestría, Universidad Nacional de San Agustín]. [https://alicia.concytec.gob.pe/vufind/Record/UNSA\\_6cff92455f72d4c52ae6d56d567b2de0/Details](https://alicia.concytec.gob.pe/vufind/Record/UNSA_6cff92455f72d4c52ae6d56d567b2de0/Details)

- Hanes, D., Salgueiro, G., Grossetete, P., Barton, R., y Henry, J. (2017). *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things*. Cisco Press.  
[https://ptgmedia.pearsoncmg.com/images/9781587144561/samplepages/9781587144561\\_CHO8.pdf](https://ptgmedia.pearsoncmg.com/images/9781587144561/samplepages/9781587144561_CHO8.pdf)
- Hernandez Sampieri, R., Fernández Collado, C., y Baptista Lucio, M. (2014). *Métodología de la Investigación*. (6ta.) MC Graw Hill. <https://www.esup.edu.pe/wp-content/uploads/2020/12/2.%20Hernandez,%20Fernandez%20y%20Baptista-Metodolog%C3%ADa%20Investigacion%20Cientifica%206ta%20ed.pdf>
- Honeywell. (2009). *ASDX Series Silicon Pressure Sensor*. Honeywell. <https://sps-support.honeywell.com/s/article/AST-Analog-ASDX-Datasheet?tabset-3dec3=2>
- Honeywell. (2021). HPM Series Particulate Matter Sensors . Honeywell. <https://n9.cl/zyxga>
- Instituto Costarricense de Electricidad. (21 de diciembre de 2023). *Instituto Costarricense De Electricidad*. Alcance(257), Gaceta (237). <https://www.grupoice.com/wps/wcm/connect/669c79e9-84c9-4682-b63b-136238ebc7e2/Tarifas+actuales.pdf?MOD=AJPERES>
- LABS, S. (31 de mayo de 2022). *I 2C HUMIDITY AND TEMPERATURE SENSOR*. SILICON LABS. <https://www.silabs.com/documents/public/data-sheets/Si7021-A20.pdf>
- Liárraga Osuna, D. (2018). Sistema de administración de energía utilizando Internet de las Cosas (IoT)[tesis de maestría, Instituto Tecnológico y de Estudios Superiores de Monterrey]. <https://repositorio.tec.mx/handle/11285/629928>
- LoRa. (5 de enero de 2024). Telegraf & InfluxDB & Grafana. *LoRa.vsb.cz*. <https://lora.vsb.cz/index.php/telegraf-influxdb-grafana/>
- Moreno, L. G. (2023). *Topología de estrella*. Visual Paradigm. <https://online.visual-paradigm.com/es/community/share/topologia-de-estrella--mykolfo4r>
- Morel Linares, E. (2017). Diseño de un sistema Automático de supervisión de parámetros de un cuarto limpio [tesis de licenciatura, Universidad UNAPEC]. [https://bibliotecaunapec.blob.core.windows.net/tesis/TESIS\\_C2\\_IEL\\_01\\_2017\\_ET170559.pdf](https://bibliotecaunapec.blob.core.windows.net/tesis/TESIS_C2_IEL_01_2017_ET170559.pdf)
- Nova Fitness Co., L. (20 de mayo de 2021). *Laser PM2.5 Sensor Specification, model SDS011*. Nova Fitness. [https://components101.com/sites/default/files/component\\_datasheet/SDS011-Sensor-Datasheet.pdf](https://components101.com/sites/default/files/component_datasheet/SDS011-Sensor-Datasheet.pdf)
- Novatronic. (2020). *DHT21/AM2301 Sensor capacitivo de temperatura y humedad (Digital)*. Novatronic. <https://novatronic.com/index.php/product/dht21-am2301-sensor-capacitivo-de-temperatura-y-humedad-digital/>
- Pallas Areny, R. (2003). *Sensores y Acondicionadores de señal*. MARCOMBO, S.A. [https://www.academia.edu/37282240/Sensores\\_y\\_Acondicionadores\\_de\\_Se%C3%B1al\\_Ramon\\_Pallas\\_Areny](https://www.academia.edu/37282240/Sensores_y_Acondicionadores_de_Se%C3%B1al_Ramon_Pallas_Areny)

- Parra, L. (2012). *Microprocesadores*. RED TERCER MILENIO.  
<https://www.aliat.click/BibliotecasDigitales/sistemas/Microprocesadores.pdf>
- Payan de Tejada Alonso, A. (2021). Sistema de monitorización de variables ambientales de bajo coste basado en Arduino [tesis de doctorado, Universidad Politécnica de Madrid].  
<https://oa.upm.es/69277/>
- Peña, C. (2020). *Arduino IDE: configuración y uso*. Plandos S.A.  
[https://books.google.co.cr/books/about/Arduino\\_IDE.html?id=Xgv2DwAAQBAJ&redir\\_esc=y](https://books.google.co.cr/books/about/Arduino_IDE.html?id=Xgv2DwAAQBAJ&redir_esc=y)
- Piera Systems (2021). *Intelligent Particle Sensors*. Piera Systems. <https://pierasystems.com/intelligent-particle-sensors/>
- Proyectos con Arduino (2023). *Sensores de Polvo y Partículas en el Aire*. Arduino.  
[https://proyectosconarduino.com/sensores/polvo-particulas/?expand\\_article=1](https://proyectosconarduino.com/sensores/polvo-particulas/?expand_article=1)
- Quiel Hidalgo, S. (2021). Diseño de sistema de medición y monitorización de la cantidad de material de un basurero de reciclaje [tesis de licenciatura, Instituto Tecnológico de Costa Rica]. <https://repositoriotec.tec.ac.cr/handle/2238/13447>
- Rembor, K. (14 de septiembre de 2022). Adafruit PMSA003I Air Quality Breakout. Adafruit.  
<https://learn.adafruit.com/pmsa003i>
- Rocha Boza, F. (2019). Aprovechamiento de tecnologías basadas en internet de las cosas y su relación con el agotamiento de protocolos IPV4 en la zona de San Carlos hasta el año 2019 [tesis de licenciatura, Universidad Técnica Nacional].  
<https://repositorio.utn.ac.cr/server/api/core/bitstreams/5a3d1c5a-03f9-42b5-aade-7c94af59c404/content>
- Ruonala, J. (2023). Comprender el desempeño y las especificaciones de medición. VAISALA.  
<https://www.vaisala.com/es/expert-article/understanding-measurement-performance-and-specifications>
- Sabalza Mejía, M. (2016). Control seguro de una red de sensores/actuadores domésticos [tesis de grado, Universidad Tecnológica de Bolívar].  
<https://repositorio.utb.edu.co/handle/20.500.12585/3228?locale-attribute=en#page=2>
- SENSIRION. (Mayo de 2015). Datasheet SHT3x-DIS Humidity and Temperature Sensor. SENSIRION.  
[https://cdn-shop.adafruit.com/product-files/2857/Sensirion\\_Humidity\\_SHT3x\\_Datasheet\\_digital-767294.pdf](https://cdn-shop.adafruit.com/product-files/2857/Sensirion_Humidity_SHT3x_Datasheet_digital-767294.pdf)
- \_\_\_\_\_. (2020). *PM2.5 Sensor for HVAC and air quality applications SPS30*. SENSIRION.  
<https://www.sensirion.com/products/catalog/SPS30/>
- SparkFun. (2024). *SparkFun Atmospheric Sensor Breakout - BME280*. SparkFun.  
<https://www.sparkfun.com/products/13676>
- Salazar Xirinachs, J. (2022). El sector/clúster de dispositivos médicos de Costa Rica: estudio de caso. Biblioteca Felipe Herrera Banco Interamericano de Desarrollo.

- <https://publications.iadb.org/es/publications/spanish/viewer/el-sector-cluster-dispositivos-medicos-costa-rica.pdf>
- STMicroelectronics. (29 de junio de 2017). *LPS22HB MEMS nano pressure sensor: 260-1260 hPa absolute digital output barometer*. STMicroelectronics. <https://www.st.com/en/mems-and-sensors/lps22hb.html>
- Systems, E. (2023). *ESP32-WROOM-32*. Espressif. [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf)
- Tiwary, A., y Williams, I. (2019). *Air Pollution Measurement, Modelling and Mitigation*. (4ta.) [https://api.pageplace.de/preview/DT0400.9781498719469\\_A37409788/preview-9781498719469\\_A37409788.pdf](https://api.pageplace.de/preview/DT0400.9781498719469_A37409788/preview-9781498719469_A37409788.pdf)
- Villanueva Martinez, S. (2014). *Diseño de un sistema de captura y procesamiento de señales* [tesis de Grado, Universitat Politècnica de Valencia]. <https://riunet.upv.es/handle/10251/53674>
- Waterhouse Morales, M. (2021). *Implementación de un sistema de monitoreo de partículas no viables en un ambiente controlado de Boston Scientific* [tesis de licenciatura, Universidad de Costa Rica]. <https://repositorio.sibdi.ucr.ac.cr/server/api/core/bitstreams/1236c19a-f1fe-4bef-8163-46f956ccd98b/content>
- Yong, Z. (26 de febrero de 2018). *PMSA003I series data manual*. PLANTOWER. [https://cdn-shop.adafruit.com/product-files/4632/4505\\_PMSA003I\\_series\\_data\\_manual\\_English\\_V2.6.pdf](https://cdn-shop.adafruit.com/product-files/4632/4505_PMSA003I_series_data_manual_English_V2.6.pdf)
- Zhendong, Z., & Yong, Z. (20 de Febrero de 2019). *About us?*. PLANTOWER. [www.plantower.com](http://www.plantower.com)

## Anexo 1

### Ejemplo de código del sistema de monitoreo implementado en la plataforma Arduino IDE.

```

#include <Adafruit_SHT31.h> //Librería del sensor de temperatura y humedad SHT31-D
#include <Adafruit_BMP3XX.h> //Librería del sensor de presión BMP390
#include <Wire.h> //funciones para comunicación I2C
#include <SPI.h> //funciones para comunicación SPI
#include <IpsI2C.h> //Librería del sensor de partículas en el aire IPS-7100
#include <WiFi.h> // Funciones para conectividad por Wi-Fi
#include <HTTPClient.h> // Librería para realizar solicitudes HTTP

//Ejemplo de URL, servidor local
String URL=
"http://dirección_IP_computadora/Nombre_Carpeta_en_XAMPP/Nombre_documento_PHP.php";
//Se define la variable URL, como la dirección de servidor local.

#define BMP_SCK 18 //Definir pin SCL(SCK) para BMP390
#define BMP_MISO 19 //Definir pin SDO (MISO) para BMP390
#define BMP_MOSI 23 //Definir pin SDA (MOSI) para BMP390
#define BMP_CS 5 //Definir pin CS para BMP390

#define SDA_PIN 21 //Definir pin SDA para IPS-7100
#define SCL_PIN 22 //Definir pin SCL para IPS-7100

const char* ssid = "Nombre de red Wi-Fi"; // Corresponde a la variable "ssid" que contiene el
nombre de la red Wi-Fi a la cual el ESP32 debe conectarse
const char* password = "Contra123"; // Corresponde a la variable que contiene la contraseña de la
red Wi-Fi

//Las siguientes líneas se deben colocar en caso de que se desee utilizar
redundancias con 2 routers.
const char* ssid2 = "Nombre de red Wi-Fi"; // Corresponde a la variable "ssid2"
del segundo router.
const char* password2 = "Contra123"; // Corresponde a la variable "password2" del
segundo router.
bool conectadoRed1 = true; // Asume que empieza conectado a la red principal

```

```

Adafruit_BMP3XX bmp;// Se declara como "bmp" para BMP3XX
Adafruit_SHT31 SHT31 = Adafruit_SHT31();//Se declara como "SHT31" para sensor SHT31,
además de inicializarse el sensor.
IpsSensor ips_sensor;// Se declara como "ips_sensor" para sensor IPS7100

float temperatura = 0; // Variable equivale a cero cuando el sensor no realiza la medición
float humedad = 0; // Variable equivale a cero cuando el sensor no realiza la medición
float presion = 0; // Variable equivale a cero cuando el sensor no realiza la medición

//Las siguientes variables equivalen a cero cuando el sensor no realiza la medición

float PM0p5 = 0; // Concentración de partículas de 0.5 micrómetros
float PM1p0 = 0; // Concentración de partículas de 1.0 micrómetros
float PM5 = 0; // Concentración de partículas de 5 micrómetros
float PC0p5 = 0; // Conteo de partículas de 0.5 micrómetros
float PC1p0 = 0; // Conteo de partículas de 1.0 micrómetros
float PC5 = 0; // Conteo de partículas de 5 micrómetros

//Función "millis" para verificar que cuando transcurran 6 horas se genere un
nuevo registro
unsigned long millisAnteriores= 0; // En la variable "millisAnteriores" se
almacena el tiempo anterior.
const long periodicidad = 21600000; // En la variable "periodicidad" se realiza
un registro cada 6 horas.

void setup() {

// Se inicializan los sensores
SHT31.begin(0x44); //inicializar sensor SHT31 con la dirección I2C
bmp.begin_SPI (BMP_CS, BMP_MOSI, BMP_MISO, BMP_SCK); //inicializar sensor BMP390
ips_sensor.begin(SDA_PIN, SCL_PIN); // inicializar sensor IPS7100

// Se inicializa el bloque de conexión a Wi-Fi
conectarWiFi(); //bloque que controla la función de conexión a Wi-Fi
// Configuración del sensor BMP390
bmp.setIIRFilterCoeff(BMP3_IIR_FILTER_COEFF_3); //función que configura al sensor para
filtrar las variaciones rápidas

}

void loop() {

```

```

//Función millis para obtener el tiempo actual

unsigned long millisActuales = millis(); //se obtiene el tiempo actual
if (millisActuales - millisAnteriores >= periodicidad)
{ millisAnteriores = millisActuales; //Actualizar el tiempo actual

// ***Se coloca la condición de verificar la conexión del ESP32 a la red, sino para intentar
reconectarlo ejecutando el bloque "conectarWiFi"
if (WiFi.status() != WL_CONNECTED) { //condición que verifica la conectividad a la red.
    conectarWiFi(); // Si se cumple la condición anterior, se debe ejecutar el bloque "conectarWiFi"
}

//Si se desea utilizar redundancias con 2 routers se debe agregar el siguiente
código en vez de la línea anterior con ***
if (WiFi.status() != WL_CONNECTED) { //condición que verifica la conectividad a
la red.
reconectarWiFi(); // Intenta reconectar si se pierde la conexión
}

// Obtener valores de variables y partículas.
variables ();
particulas ();

//Se realiza la solicitud Postdata a la base de datos.
String postData = "Temperatura=" + String(temperatura) + "&Humedad=" + String(humedad) +
"&Presion=" + String(presion) + "&PM0p5=" + String(PM0p5) + "&PM1p0=" + String(PM1p0) +
"&PM5=" + String(PM5) + "&PC0p5=" + String(PC0p5) + "&PC1p0=" + String(PC1p0) +
"&PC5=" + String(PC5);

//codigo para manejar las solicitudes http
HTTPClient http;
http.begin(URL); //Inicializar conexión con URL
http.addHeader("Content-Type", "application/x-www-form-urlencoded"); //Encabezado de la
solicitud http.

int httpCode = http.POST(postData); // Realiza solicitud HTTP Post con los datos contenidos en
"post data"
String payload = http.getString(); //Devuelve código de respuesta http, valor "200" indica solicitud
exitosa.

// Se imprime en el monitor serial URL, Datos enviados y código http de respuesta del servidor.
Serial.print("URL: "); Serial.println(URL);
Serial.print("Data: "); Serial.println(postData);
Serial.print ("httpCode: "); Serial.println(httpCode);

```

```

Serial.print ("payload: "); Serial.println (payload);
Serial.println ("-----");

}

}

void variables () {

    temperatura = SHT31.readTemperature(); // Leer el valor de temperatura en el sensor
    humedad = SHT31.readHumidity(); // Leer el valor de humedad en el sensor
    presion = (bmp.pressure / 100.0); // Leer el valor de presión en hPa en el sensor

//En caso de que fallen los sensores en la medición se registrará un valor de
cero.

if (isnan(temperatura) || isnan(humedad)|| isnan(presion)) { // Sino se obtienen los valores de las
variables, estas toman el valor de cero.
    Serial.println("Error en leer variables!");
    temperatura = 0;
    humedad = 0;
    presion = 0;
}

//Sino se cumple la condición anterior se imprime en el monitor serial los valores de las variables
Serial.println("Temperatura:", temperatura);
Serial.println("Humedad:", humedad);
Serial.println("Presion:", presion);
delay(2160000); // pausa la ejecución del bloque "void variables" por 6 horas

}

void particulas () {

{
ips_sensor.update(); // Actualiza las variables y mediciones internas del sensor IPS-7100

PM0p5 = ips_sensor.getPM05(); // Concentración de partículas de 0.5 micrometros
PM1p0 = ips_sensor.getPM10(); // Concentración de partículas de 1.0 micrometros
PM5 = ips_sensor.getPM50(); // Concentración de partículas de 5 micrometros
PC0p5 = ips_sensor.getPC05(); // Conteo de partículas de 0.5 micrometros
PC1p0 = ips_sensor.getPC10(); // Conteo de partículas de 1.0 micrometros
PC5 = ips_sensor.getPC50(); // Conteo de partículas de 5 micrometros

//En caso de que fallen los sensores en la medición se registrará un valor de cero.

```

```

if (isnan(PM0p5) || isnan(PM1p0) || isnan(PM5) || isnan(PC0p5) || isnan(PC1p0) ||
isnan(PC5)) {
Serial.println("Error en leer partículas!");
PM0p5 = 0;
PM1p0 = 0;
PM5 = 0;
PC0p5 = 0;
PC1p0 = 0;
PC5 = 0;

// Imprimir valor de concentración de partículas obtenido por IPS-7100

Serial.print("PM0.5: "); // Se define nombre a imprimir para la variable "PM0p5"
Serial.println(PM0p5); // imprimir valor leído por el sensor.
Serial.print("PM1.0: "); // Se define nombre a imprimir para la variable "PM1p0"
Serial.println(PM1p0); // imprimir valor leído por el sensor.
Serial.print("PM5.0: "); // Se define nombre a imprimir para la variable "PM5"
Serial.println(PM5); // imprimir valor leído por el sensor.

// Imprimir valor de coneteo de partículas obtenido por IPS-7100

Serial.print("PC0.5: "); // Se define nombre a imprimir para la variable "PC0p5"
Serial.println(PC0p5); // imprimir valor leído por el sensor.
Serial.print("PC1.0: "); // Se define nombre a imprimir para la variable "PC1p0"
Serial.println(PC1p0); // imprimir valor leído por el sensor.
Serial.print("PC5.0: "); // Se define nombre a imprimir para la variable "PC5"
Serial.println(PC5); // imprimir valor leído por el sensor.

}

delay(21600000); // pausa la ejecución del bloque "void variables" por 6 horas

}

void conectarWiFi() {

WiFi.mode(WIFI_OFF); // Se apaga módulo Wi-Fi, para ser reiniciado
delay(1000); // retraso de 1 segundo para apagar completamente el módulo

WiFi.mode(WIFI_STA); // Se configura el módulo Wi-Fi en modo estación, actua como cliente

WiFi.begin(ssid, password); // Se inicializa el proceso de conexión a la red con las credenciales.
Serial.println("Conectandose a WiFi"); // Se utiliza el monitor serial de Arduino IDE para ver estado
de conexión (opcional)

```

```

// El siguiente Bucle (opcional) se ejecuta solo mientras el ESP32 intenta conectarse a la red
// Se imprime un punto cada 0.5 Segundos para indicar que está intentando la conexión a la red.
while (WiFi.status () != WL_CONNECTED) {
    delay(500);
    Serial.print("."); // Se imprime un punto cada 0.5 Segundos
}

// Las siguientes líneas imprimen en el monitor serial la conexión exitosa la red
Serial.print ("Conectado a : "); Serial.println (ssid); //Indica a que red se conectó
Serial.print ("Dirección IP : "); Serial.println (WiFi.localIP()); //Indica la dirección IP asignada al
ESP32.
}

//El siguiente código se ejecutará en el bucle conectarWiFi(), reconectarWiFi(),
esperarConexion() en caso que se quieran añadir redundancias.

void conectarWiFi() {

//Se establece conexión con la red 1
WiFi.mode(WIFI_STA);
Serial.println("Conectando a la red 1...");
WiFi.begin(ssid, password);

// Se espera 10 segundos para conectarse a la red 1 sino se intenta la conexión
con la red 2
if (esperarConexion(10000)) {
Serial.println("Conectado a la red 1");
conectadoRed1 = true;
} else {
Serial.println("No se pudo conectar a red 1, intentando red 2...");
WiFi.begin(ssid2, password2);

// Se espera 10 segundos para conectarse a la red 2 sino se indica que falla la
conexión.

if (esperarConexion(10000)) {
if (esperarConexion(10000)) {
Serial.println("Conectado a red 2");
conectadoRed1 = false;
} else {
Serial.println("No se pudo conectar a ninguna red");
}}}

```

// Si el estado de conexión del bloque "esperarConexion" no es el estado WL\_CONNECTED, según se configuro antes, se ejecuta el bloque reconectarWiFi().

```
void reconectarWiFi() {
  if (conectadoRed1) {
    Serial.println("Reintentando conexión a red 1...");
    WiFi.begin(ssid, password);

    if (!esperarConexion(5000)) {
      Serial.println("Fallo en red 1, conectando a red 2...");
      WiFi.begin(ssid2, password2);
      conectadoRed1 = false;
      esperarConexion(5000);
    }
  } else {
    Serial.println("Reintentando conexión a red 2...");
    WiFi.begin(ssid2, password2);

    if (!esperarConexion(5000)) {
      Serial.println("Fallo en red 2, conectando a red 1...");
      WiFi.begin(ssid, password);
      conectadoRed1 = true;
      esperarConexion(5000);
    }
  }
}
```

// se ejecuta el bloque para establecer el tiempo de espera en las conexiones.

```
bool esperarConexion(int tiempoEspera) {
  int inicio = millis();
  while (WiFi.status() != WL_CONNECTED && millis() - inicio < tiempoEspera) {
    delay(500);
    Serial.print(".");
  }
  return WiFi.status() == WL_CONNECTED; }
}
```

## Anexo 2

### Ejemplo de código PHP, del sistema de monitoreo implementado en la plataforma XAMPP.

```
<?php
```

```
$hostname = "localhost"; // Se indica el nombre del servidor
```

```
$username = "root"; // Se indica el nombre de usuario de la base de datos
```

```
$password = ""; // se puede ingresar el valor de la contraseña de la base de datos
```

```
$database = "sistema_monitoreo"; // Se debe indicar el nombre de la base de datos a la que se  
enviaran los registros
```

```
$conn = mysqli_connect($hostname, $username, $password, $database); // Se establece conexión a  
la base de datos.
```

```
// verifica el estado de la conexión y si la conexión falla se envía el mensaje de conexión fallida
```

```
if (!$conn) {
```

```
die("Conexion Fallida: ". mysqli_connect_error ());
```

```
}
```

```
// Se imprime el mensaje en caso de que la conexión fue exitosa
```

```
echo "Conexion exitosa a la base de datos";
```

```
print_r($_POST); // Imprime el contenido de la solicitud Post
```

```
// Se verifica que todas las variables del sistema de monitoreo estén incluidas en la solicitud POST.
```

```
if (isset($_POST["Temperatura"]) && isset($_POST["Humedad"])&&
```

```
isset($_POST["Presion"])&& isset($_POST["PM0p5"])&& isset($_POST["PM1p0"]) &&
```

```
isset($_POST["PM5"])&& isset($_POST["PC0p5"])&& isset($_POST["PC1p0"])&&
```

```
isset($_POST["PC5"])) {
```

```
// Se nombran variables a cada solicitud de Post
```

```
$t = $_POST["Temperatura"];
```

```

$h = $_POST["Humedad"];
$p = $_POST["Presion"];
$pm05= $_POST["PM0p5"];
$pm1= $_POST["PM1p0"];
$pm5= $_POST["PM5"];
$pc05= $_POST["PC0p5"];
$pc1= $_POST["PC1p0"];
$pc5 = $_POST["PC5"];

```

//Se realiza la función de insertar en la tabla nodo\_1, los valores de las variables definidas para cada columna de la tabla de la base de datos.

```

$sql = "INSERT INTO nodo_1 (Temperatura, Humedad, Presion, PM0p5, PM1p0, PM5,
PC0p5, PC1p0, PC5)
VALUES (".$.t.", ".$.h.", ".$.p.", ".$.pm05.", ".$.pm1.", ".$.pm5.", ".$.pc05.", ".$.pc1.", ".$.pc5.");
}

```

// Se verifica si la consulta SQL fue exitosa. En caso de ser exitosa se imprime el mensaje indicando que se añadió un nuevo registro a la base de datos, de lo contrario se envía el mensaje de error.

```

if (mysqli_query ($conn, $sql)) {
echo "Nuevo registro creado";}
else {
echo "error: " . $sql . "<br>" . mysqli_error($conn);
}

```

?>

### Anexo 3

#### Tarifa del Instituto Costarricense de Electricidad (ICE), sector industrial.

Publicado en el Alcance N°257 Gaceta N°237 del 21 de diciembre del 2023

##### Tarifa T-IN Industrial

A. Aplicación: Para el suministro de energía y potencia a servicios eléctricos servidos a media o baja tensión clasificados en el sector industrial según la clasificación de actividades económicas (código CIU) utilizada por el Banco Central de Costa Rica (BCCR)

##### B.- Características de servicio:

Suministro de energía y potencia a servicios eléctricos servidos en media o baja tensión y clasificados como B1, B2, B3, B4, B5, B6, B7, M1, M2, M3, M4, M5, M6, M7 o M8, conforme a lo especificado en el artículo 26 de la norma técnica regulatoria AR-NT-SUCOM "Supervisión de la comercialización del suministro eléctrico en baja y media tensión".

##### C. Precios mensuales:

<b>Para consumos menores o iguales que 3 000 kWh</b>	
Por cada kWh	¢ 127.11
<b>Para consumos mayores de 3 000 kWh</b>	
Cargo por energía, por cada kWh	¢ 76.07
Cargo por potencia, por cada kW	¢ 12 571.65

## Anexo 4

## Tasas de interés de inversión a plazo en colones en el BCR.

**COLONES**  
**TASA NOMINAL**

Plazo en días	Físico negociable en Bolsa	Bancobcr y Desmaterializado-negociable BCR
1-6	N/A	0.18%
7-13	N/A	0.20%
14-20	N/A	0.62%
21-29	N/A	1.21%
30-59	3.01%	3.11%
60-89	3.45%	3.55%
90-119	3.88%	3.98%
120-149	4.53%	4.63%
150-179	5.01%	5.11%
<b>180*-209</b>	5.19%	5.29%
210-239	5.19%	5.29%
240-269	5.44%	5.54%
270-299	5.45%	5.55%
300-329	5.49%	5.59%
330-359	5.54%	5.64%
360-539	5.59%	5.69%
540-719	5.69%	5.79%
720-1079	6.04%	6.14%
1080-1439	6.24%	6.34%
1440-1799	6.38%	6.48%
De 1800	6.52%	6.62%

\*Interés capitalizable

**NOTA:** Se debe reducir 15% de Impuesto de Renta sobre los intereses, según se define en el artículo 23 de la Ley No. 7092 Ley del Impuesto sobre la Renta y su Reglamento, reformados por la Ley de Fortalecimiento de Finanzas Públicas No. 9635, a partir del día del 01 de julio de 2019.