



**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE GUAYAQUIL**

**CARRERA DE INGENIERÍA ELECTRÓNICA**

Trabajo de titulación previo a la obtención del título de:  
**“INGENIERO ELECTRÓNICO”**

**“DISEÑO Y DESARROLLO DE UN PROTOTIPO DE SISTEMA DE  
MONITOREO DE CALIDAD DE AIRE BAJO UNA TEMPERATURA LOT  
EN LA NUBE PARA LA DETECCIÓN DE LOS NIVELES DE  
CONTAMINACIÓN DE AIRE EN LAS SALAS DEL HOSPITAL DE NIÑOS  
LEÓN BECERRA”**

Autor:

Trino Gerónimo Párraga Intriago

Tutor:

Ing. Teddy Jhennse Negrete Peña

GUAYAQUIL – ECUADOR

2020

## **Certificado de Responsabilidad y Autoría**

Los conceptos desarrollados en el presente proyecto de titulación, el desarrollo del tema, el análisis y respectivas conclusiones de este, corresponden exclusivamente a **Trino Gerónimo Párraga Intriago** con cédula de identificación 0924780349 y el patrimonio intelectual del mismo a la **Universidad Politécnica Salesiana**.



---

Trino Gerónimo Párraga Intriago

C.I: 0924780349

## Certificado de Cesión de Derechos de Autor

Yo, **Trino Gerónimo Párraga Intriago** con cédula de identificación 0924780349, manifiesto mi voluntad y cedo a la **Universidad Politécnica Salesiana** la titularidad sobre los derechos del actual proyecto de titulación en virtud de que soy el autor del tema: “**DISEÑO Y DESARROLLO DE UN PROTOTIPO DE SISTEMA DE MONITOREO DE CALIDAD DE AIRE BAJO UNA TEMPERATURA LOT EN LA NUBE PARA LA DETECCIÓN DE LOS NIVELES DE CONTAMINACIÓN DE AIRE EN LAS SALAS DEL HOSPITAL DE NIÑOS LEÓN BECERRA**”, mismo que ha sido para la obtención del título de: **INGENIERO ELECTRÓNICO**, quedando la universidad facultada para ejercer y usar los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual del Ecuador, en mi condición de autor, me reservo los derechos morales de la obra anteriormente citada.

En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la **Universidad Politécnica Salesiana**.



---

Trino Gerónimo Párraga Intriago

C.I: 0924780349

## **Certificado de Dirección del Trabajo de Titulación**

Yo, Msc. Teddy Negrete certifico que bajo mi dirección y asesoría fue desarrollado el proyecto técnico de titulación “**DISEÑO Y DESARROLLO DE UN PROTOTIPO DE SISTEMA DE MONITOREO DE CALIDAD DE AIRE BAJO UNA TEMPERATURA LOT EN LA NUBE PARA LA DETECCIÓN DE LOS NIVELES DE CONTAMINACIÓN DE AIRE EN LAS SALAS DEL HOSPITAL DE NIÑOS LEÓN BECERRA**”, realizado por el señor **Trino Gerónimo Párraga Intriago** para la obtención del título de **Ingeniero Electrónico**.

Por medio de la presente certifico que el documento cumple con los requisitos establecidos en el Instructivo para la Estructura y Desarrollo de Trabajos de Titulación para pregrado de la Universidad Politécnica Salesiana.

En virtud de lo anterior, autorizo su presentación y aceptación como una obra auténtica y de alto valor académico.



---

Teddy Jhennse Negrete Peña

## **Dedicatoria**

Dedico este proyecto de titulación principalmente a Dios, porque creo firmemente que en cada paso que he dado en la consecución de este logro Él ha estado conmigo. A Esther, mi esposa, por ser el pilar fundamental de mi vida y el motor que me impulsa cada día a seguir creciendo en todo sentido. A mis padres, por darme siempre su amor y apoyo incondicional, pero, sobre todo, porque me hicieron creer que era posible, sin importar el tiempo y las dificultades que se presentaran. A Adrián, mi hijo, puesto que su llegada a este mundo me dio el coraje necesario para prepararme y ser lo mejor posible para él.

**Trino Gerónimo Párraga Intriago**

## **Agradecimiento**

Me van a faltar páginas para agradecer a todas las personas que de una u otra forma se han involucrado en la realización de este trabajo. Sin embargo, agradezco especialmente a mis Padres por el gran ejemplo que me han dado de superación y perseverancia.

A mi esposa, por sus palabras de aliento y por la confianza que depositó en mí. Espero que en el futuro podamos seguir cosechando más triunfos como estos, trabajando como el gran equipo que somos.

A mis amigos, en especial a la Econ. Zaida Sáenz Abril porque con sus consejos y palabras oportunas pude tomar decisiones acertadas en la consecución de este logro.

Finalmente, a mis profesores de la carrera de ingeniería electrónica, particularmente al Ing. Teddy Negrete que, como mi director de Tesis, me ayudó con los consejos y correcciones necesarias para culminar con éxito este trabajo. Al Ing. Diego Freire, Ing. Carlos Bósquez, Ing. Leonel Vásquez, Ing. Gino Alvarado, Ing. Mónica Miranda e Ing. Néstor Zamora, porque con sus enseñanzas me ayudaron a poder crecer cada día como un profesional con calidad y calidez humana.

**Trino Gerónimo Párraga Intriago**

## **Agradecimiento Institucional**

Este agradecimiento va exclusivamente al Hospital de Niños León Becerra y a quienes trabajan en él, por haberme permitido implementar el proyecto en sus instalaciones, facilitándome los espacios y tiempos necesarios para realizar las mediciones de variables en el ambiente interior.

**Trino Gerónimo Párraga Intriago**

## Resumen

Se diseñó y desarrolló un sistema de monitoreo de calidad del aire para detectar los niveles de contaminación de las salas del Hospital de Niños León Becerra, mediante la implementación de una plataforma IOT, para almacenar las mediciones en la nube y acceder a ellas a través de un aplicativo web. Se escogió para su implementación una tarjeta Wasmote junto con sensores para medir las variables de CO, temperatura y humedad, una tarjeta Raspberry para enviar los datos a internet, la plataforma de base de datos Firestore de Google, y el diseño de una Aplicación Web Progresiva (PWA), siendo seleccionados estos elementos por su afinidad con proyectos IoT, además de su robustez, capacidad de respuesta rápida y excelente desempeño en todas las pruebas realizadas.

Las variables a medir en el aire fueron determinadas en base a las recomendaciones de la Agencia de Protección ambiental (EPA) y la Administración de Seguridad y Salud Ocupacional (OSHA), a más de las normativas establecidas por la Unión Española de Normalización en su norma UNE 100713:2005 para los niveles adecuados de temperatura y humedad en hospitales. Las mediciones fueron realizadas en 9 salas, estableciéndose 3 visitas para cada una de ellas, recolectando un total de 14504 muestras en 38 horas de funcionamiento. La forma de medición fue realizada a distintos niveles de proximidad con respecto al paciente y a distintos niveles de altura.

En el análisis de las mediciones se detectaron valores permitidos de monóxido de carbono en todas las salas, no así de las variables de temperatura y humedad, las cuales en algunos lugares estuvieron por encima de lo recomendado. En el análisis de la medición por niveles de proximidad y altura se encontró diferencias al medir las variables en el piso, a 1 metro y 2 metros de altura, resultados que pueden ser analizados en proyectos futuros.

Con los resultados obtenidos se concluyó que el equipo es un aporte significativo para proyectos IoT relacionados al monitoreo de la calidad del aire interior, logrando cumplir con las exigencias requeridas en cuanto a confiabilidad, portabilidad y autonomía. Presenta como punto crítico el limitado acceso a algunos de los elementos del equipo en el mercado nacional, incrementando sus costos y tiempos de espera para adquirirlos.



## Abstract

An air quality monitoring system was designed and developed to detect pollution levels in the rooms of the León Becerra Children's Hospital, through the implementation of an IOT platform, to store the measurements in the cloud and access them through a web application. A Waspnote card was chosen for its implementation along with sensors to measure the variables of CO, temperature and humidity, a Raspberry card to send the data to the internet, the Google Firestore database platform, and the design of a Progressive Web Application (PWA), these elements being selected for their affinity with IoT projects, in addition to their robustness, rapid response capability and excellent performance in all tests performed.

The variables to be measured in the air were determined based on the recommendations of the Environmental Protection Agency (EPA) and the Occupational Health and Safety Administration (OSHA), in addition to the regulations established by the Spanish Standardization Union in its standard UNE 100713: 2005 for adequate temperature and humidity levels in hospitals. The measurements were made in 9 rooms, establishing 3 visits for each of them, collecting a total of 14504 samples in 38 hours of operation.

The measurement form was performed at different levels of proximity to the patient and at different levels of height. In the analysis of the measurements, allowed values of carbon monoxide were detected in all rooms, but not in the temperature and humidity variables, which in some places were above the recommended. In the analysis of the measurement by levels of proximity and height, differences were found when measuring the variables on the floor, at 1 meter and 2 meters, results that can be analyzed in future projects.

With the results obtained, it was concluded that the equipment is a significant contribution to IoT projects related to indoor air quality monitoring, managing to meet the required requirements in terms of reliability, portability and autonomy. It presents as a critical point the limited access to some of the elements of the equipment in the national market, increasing its costs and waiting times to acquire them.

## Índice General

Certificado de Responsabilidad y Autoría.....	ii
Certificado de Cesión de Derechos de Autor .....	iii
Certificado de Dirección del Trabajo de Titulación.....	iv
Dedicatoria.....	v
Agradecimiento.....	vi
Agradecimiento Institucional.....	vii
Resumen .....	viii
Abstract.....	ix
Índice General.....	x
Índice de Figuras .....	xi
Índice de Tablas .....	xiii
1. Introducción.....	1
1.1 Descripción del Problema.....	1
1.2 Antecedentes.....	2
1.1 Importancia .....	3
1.2 Alcance .....	4
1.3 Delimitación.....	4
1.3.1 Espacial.....	4
1.3.2 Temporal.....	5
1.3.3 Académica. ....	5
1.4 Objetivos.....	5
1.4.1 Objetivo general.....	5
1.4.2 Objetivos específicos.....	5
2. Fundamentos Teóricos .....	5
2.1 Internet de las Cosas (IoT).....	5
2.2 Elementos de IoT .....	6
2.2.1 Hardware. ....	6
2.2.2 Software.....	9
2.2.3 Internet.....	12
2.3 Aplicaciones IoT.....	25
2.3.1 Reducción de contaminación acústica en Cuenca.....	25
2.3.2 Sistema de monitoreo de calidad del aire en Mallorca.....	26
2.3.3 Sistema de alerta temprana para evitar inundaciones en ríos. ....	27
2.4 Calidad del Aire .....	28
2.4.1 Normativas de Calidad del Aire.....	28
2.4.2 Calidad del Aire en Hospitales. ....	30
3. Fundamentos Metodológicos .....	32
3.1 Selección de Equipos y Protocolos .....	33
3.1.1 Elección de Hardware. ....	33
3.1.2 Elección de Software.....	41
3.1.3 Protocolos de comunicación utilizados .....	47
3.2 Diseño Lógico.....	48
3.3 Diseño Físico .....	50
3.3.1 Etapa de medición. ....	50
3.3.2 Etapa de comunicación a internet. ....	52
3.3.3 Etapa de almacenamiento. ....	54
3.3.4 Etapa de visualización. ....	56
3.3.5 Tratamiento de los datos obtenidos.....	59

3.4 Pruebas y Funcionamiento.....	63
4. Análisis y Resultados.....	64
4.1 Análisis de las Mediciones en el HLB .....	64
4.1.1 Quirófanos. ....	65
4.1.2 Unidad de Cuidados Intensivos. ....	66
4.1.3 Postoperatorio. ....	66
4.1.4 Central de esterilización. ....	66
4.1.5 Emergencia. ....	67
4.1.6 Observación. ....	67
4.1.7 Pensionado de primera. ....	68
4.1.8 Administración.....	68
4.2 Análisis del Método de Medición .....	69
4.3 Análisis de la Implementación del Hardware .....	71
4.4 Análisis de la Implementación del Software .....	71
Conclusiones.....	72
Conclusiones específicas .....	72
Recomendaciones.....	73
Líneas futuras.....	74
Proyectos de Investigación Vinculados.....	74
Bibliografía .....	75
Anexos.....	78
Acta de Entrega-Recepción .....	<b>¡Error! Marcador no definido.</b>
Presupuesto.....	78
Cronograma de Actividades .....	79
Monóxido de Carbono (Página Oficial de OSHA) .....	80
Norma Técnica de Ventilación General en Hospitales .....	83
Implementación del Hardware del Sistema de Monitoreo .....	89
Programación de la PWA .....	90
Componente principal .....	90
Mediciones en el Hospital de Niños León Becerra.....	93

## Índice de Figuras

<i>Figura 1.</i> Personas que transitan semanalmente en el HLB.....	3
<i>Figura 2.</i> Áreas del HLB consideradas para monitorear la calidad del aire .....	3
<i>Figura 3.</i> Ubicación del Hospital de Niños León Becerra .....	4
<i>Figura 4.</i> Sensores. ....	6
<i>Figura 5.</i> Microcontrolador PIC16F877A. ....	7
<i>Figura 6.</i> Shield de ethernet para Arduino uno. ....	7
<i>Figura 7.</i> Placa Waspote. ....	8
<i>Figura 8.</i> Funcionamiento de un gateway .....	8
<i>Figura 9.</i> Arquitectura genérica de una solución de IoT. ....	9
<i>Figura 10.</i> Tarjeta Raspberry .....	9
<i>Figura 11.</i> Logo de la iniciativa open source.....	10
<i>Figura 12.</i> Arquitectura de un sistema operativo .....	10
<i>Figura 13.</i> Linus Torvalds.....	11
<i>Figura 14.</i> Arquitectura de Android .....	11
<i>Figura 15.</i> Principales distros Linux.....	12
<i>Figura 16.</i> Sistema Operativo Debian.....	12
<i>Figura 17.</i> Comunicación HTTP .....	14

<i>Figura 18.</i> Ejemplo de programa escrito en HTML.....	14
<i>Figura 19.</i> Diagrama de carga de sitios web.....	15
<i>Figura 20.</i> Estructura del CSS.....	15
<i>Figura 21.</i> Tienda de aplicaciones de Google.....	16
<i>Figura 22.</i> Comparación entre aplicaciones web y nativas.....	17
<i>Figura 23.</i> Lenguajes de programación más usados.....	18
<i>Figura 24.</i> Typescript un superset de Javascript.....	18
<i>Figura 25.</i> Timeline de EcmaScript.....	19
<i>Figura 26.</i> Transpilación vs compilación.....	20
<i>Figura 27.</i> Transpilación de una función lambda.....	20
<i>Figura 28.</i> Edición de código fuente en HTML y JavaScript.....	20
<i>Figura 29.</i> Funcionamiento de Webpack.....	21
<i>Figura 30.</i> Pug vs Html.....	22
<i>Figura 31.</i> Comparativa entre SASS, SCSS y CSS.....	22
<i>Figura 32.</i> Reactividad de Vue.....	23
<i>Figura 33.</i> Sistema de componentes de Vue.....	23
<i>Figura 34.</i> Ventajas de las PWA.....	24
<i>Figura 35.</i> PWA de Instagram.....	24
<i>Figura 36.</i> Tiempo de espera por respuestas.....	25
<i>Figura 37.</i> Diagrama de instalación del nivel de sensor de ruido.....	26
<i>Figura 38.</i> Plataforma de sensores instalados.....	26
<i>Figura 39.</i> Diagrama de comunicación del proyecto.....	27
<i>Figura 40.</i> Contaminantes de criterio.....	29
<i>Figura 41.</i> Rango óptimo de Humedad Relativa.....	31
<i>Figura 42.</i> Normativa UNE 100713:2005.....	32
<i>Figura 43.</i> Diagrama general del sistema de monitoreo de calidad de aire.....	33
<i>Figura 44.</i> Sensor Fígaro TGS2442.....	34
<i>Figura 45.</i> Estructura interna del sensor Fígaro TGS2442.....	34
<i>Figura 46.</i> Sensor BME280.....	35
<i>Figura 47.</i> Placa de sensores de gases.....	36
<i>Figura 48.</i> Ubicación de sockets y sensores permitidos.....	37
<i>Figura 49.</i> Componentes tarjeta Waspnote (parte superior).....	38
<i>Figura 50.</i> Componentes tarjeta Waspnote (parte posterior).....	39
<i>Figura 51.</i> Raspberry Pi 3.....	40
<i>Figura 52.</i> Componentes de la tarjeta Raspberry Pi3.....	40
<i>Figura 53.</i> Batería de iones de litio de 3.7v recargable.....	41
<i>Figura 54.</i> Batería portátil Klip Xtreme KBH-155GD.....	41
<i>Figura 55.</i> IDE Waspnote y sus partes.....	41
<i>Figura 56.</i> Servicios ofrecidos por Firebase.....	42
<i>Figura 57.</i> Consumo de Firestore.....	42
<i>Figura 58.</i> Opción de agregar PWA en la pantalla de inicio de Android.....	43
<i>Figura 59.</i> Versiones de Javascript.....	43
<i>Figura 60.</i> Tabla de soporte a la función trim.....	44
<i>Figura 61.</i> Ejemplo de una transpilación de código usando Babel.....	45
<i>Figura 62.</i> LESS vs CSS.....	45
<i>Figura 63.</i> Ejemplo de código PUG.....	46
<i>Figura 64.</i> Comparación entre Pug y HTML.....	46
<i>Figura 65.</i> Ejemplo de inserción de código Javascript mediante un tag HTML.....	47
<i>Figura 66.</i> Descarga de una página web desde el navegador.....	47
<i>Figura 67.</i> Esquema general de la etapa de medición.....	49

<i>Figura 68.</i> Esquema general de la etapa de comunicación a internet .....	49
<i>Figura 69.</i> Esquema general de la etapa de almacenamiento.....	50
<i>Figura 70.</i> Esquema general de la etapa de visualización.....	50
<i>Figura 71.</i> Configuración de la placa para la lectura de datos .....	51
<i>Figura 72.</i> Código para la lectura de datos de los sensores .....	51
<i>Figura 73.</i> Conexión entre Waspote y Raspberry Pi. ....	51
<i>Figura 74.</i> Envío de datos a través del puerto serial.....	52
<i>Figura 75.</i> Envío de datos medidos a través del puerto serie .....	52
<i>Figura 76.</i> Conexión de Raspberry Pi con el puerto serie .....	53
<i>Figura 77.</i> Inicialización de librerías de Firebase .....	53
<i>Figura 78.</i> Envío de datos a Firestore desde la Raspberry Pi.....	54
<i>Figura 79.</i> Panel de Firestore en Firebase. ....	54
<i>Figura 80.</i> Colección de configuraciones .....	55
<i>Figura 81.</i> Colección de sensores .....	55
<i>Figura 82.</i> Icono de la PWA en la pantalla de inicio de Android .....	56
<i>Figura 83.</i> Pantalla principal de la aplicación .....	56
<i>Figura 84.</i> Selección del lugar de medición.....	57
<i>Figura 85.</i> Formatos de visualización de datos .....	57
<i>Figura 86.</i> Selector de cantidad de muestras para visualizar .....	58
<i>Figura 87.</i> Pestañas de visualización de mediciones .....	58
<i>Figura 88.</i> Exportación de los valores visualizados en la aplicación .....	59
<i>Figura 89.</i> Opción de detener y reanudar las mediciones .....	59
<i>Figura 90.</i> Formato para el análisis de los datos, elaboración propia.....	60
<i>Figura 91.</i> Rango de valores que serán analizados en las mediciones .....	60
<i>Figura 92.</i> Gráfico para valores medidos de CO.....	61
<i>Figura 93.</i> Gráfico para valores medidos de temperatura.....	61
<i>Figura 94.</i> Gráfico para valores medidos de humedad. ....	62
<i>Figura 95.</i> Niveles de medición dentro del quirófano .....	64
<i>Figura 105.</i> Valores obtenidos medición por niveles en sala de Administración	70
<i>Figura 106.</i> Promedio de Mediciones en el Hospital de Niños León Becerra .....	70
<i>Figura 107.</i> Acoplamiento de sensores con lector de gases y tarjeta Waspote. ....	89
<i>Figura 108.</i> Conexión del equipo de medición a la batería de litio. ....	89
<i>Figura 109.</i> Conexión del equipo de medición al gateway (RaspberryPi3). ....	89
<i>Figura 110.</i> Equipo de medición conectado y operando.....	89
<i>Figura 111.</i> Componente principal de la aplicación.....	90
<i>Figura 112.</i> Cada pestaña de la PWA. ....	90
<i>Figura 113.</i> Navbar de la aplicación.....	91

## **Índice de Tablas**

<i>Tabla 1.</i> Principales servicios web y sus puertos.....	13
<i>Tabla 2.</i> Características del Monóxido de Carbono.....	30
<i>Tabla 3.</i> Niveles de exposición de CO, según OSHA.....	30
<i>Tabla 4.</i> Características de funcionamiento del sensor Fígaro TGS2442. ....	34
<i>Tabla 5.</i> Condiciones de funcionamiento del sensor Fígaro TGS2442.....	35
<i>Tabla 6.</i> Características eléctricas del sensor BME280. ....	36
<i>Tabla 7.</i> Sensores soportados por la placa de sensores de gases. ....	37
<i>Tabla 8.</i> Especificaciones y valores operaciones de la tarjeta Waspote. ....	39
<i>Tabla 9.</i> Implementación de ECMAScript 5 en los principales navegadores. ....	44

<i>Tabla 10.</i> Estimado de consumo de Firestore.....	55
<i>Tabla 11.</i> Parámetros ambientales y su normalización para hospitales. ....	62
<i>Tabla 12.</i> Salas en las que se realizará la medición de calidad del aire. ....	63
<i>Tabla 13.</i> Cronograma de visitas al HLB para las mediciones.....	63
<i>Tabla 14.</i> Cantidad de datos recolectados por cada sala del HLB.....	64
<i>Tabla 15.</i> Promedio de mediciones en Quirófanos. ....	65
<i>Tabla 16.</i> Promedio de mediciones en Unidad de Cuidados Intensivos. ....	66
<i>Tabla 17.</i> Promedio de mediciones en Postoperatorio. ....	66
<i>Tabla 18.</i> Promedio de mediciones en Central de Esterilización. ....	67
<i>Tabla 19.</i> Promedio de mediciones en Emergencia. ....	67
<i>Tabla 20.</i> Promedio de mediciones en Observación. ....	68
<i>Tabla 21.</i> Promedio de mediciones en Pensionado de Primera. ....	68
<i>Tabla 22.</i> Promedio de mediciones en Administración. ....	69
<i>Tabla 23.</i> Presupuesto para implementación. ....	78

## **1. Introducción**

En la actualidad, el manejo de datos e información en tiempo real ha crecido enormemente, al punto de llegar a tener carreras dedicadas al análisis, manipulación y predicción de datos, además de existir tecnologías capaces de administrar esta gran cantidad de información. El “big data”, de la mano de Cloudera o Hortonworks, se han vuelto el estándar por defecto para la industria hoy en día.

Una de las aplicaciones que se le puede dar al manejo de datos en tiempo real es el monitoreo de variables en el entorno en el que se desenvuelven las personas, lo que permite visualizar la información obtenida, analizarla y poder realizar acciones a partir de ella. Es aquí donde entra en juego el Internet of Things (IoT) o Internet de las Cosas.

El Hospital de Niños León Becerra de la ciudad de Guayaquil (HLB), que tiene como principal foco de atención la calidad de servicio y la excelente atención a los pacientes, requiere que estas tecnologías sean aplicadas para mejorar la calidad del aire en sus instalaciones. Esto permitirá que, a futuro, este y otros centros hospitalarios de la ciudad puedan obtener información temprana sobre posibles anomalías en el aire que respiran sus pacientes y el personal que labora en ellos, a fin de establecer políticas que favorezcan el mejoramiento de la calidad del aire interior.

En el capítulo 1 del presente documento se explicará el problema que motiva a la elaboración de este trabajo, junto con los objetivos, alcance y delimitación del mismo. En el capítulo 2 revisaremos la literatura relacionada a términos y conceptos que utilizaremos, al igual que proyectos que se relacionan con este. En el capítulo 3 se plantea una metodología para abordar el problema planteado, a más del diseño del sistema y su implementación. Finalmente, en el capítulo 4, analizaremos los resultados obtenidos en la puesta en marcha del sistema de medición.

### **1.1 Descripción del Problema**

El aire es de vital importancia para el ser humano. En virtud de ello, el mismo debe ser de buena calidad, sin contaminantes ni gases tóxicos. Una mala calidad del aire puede tener efectos perjudiciales en la salud de las personas a corto, mediano y largo plazo.

Existen organismos que regulan los niveles de contaminación del aire, tanto en ambientes outdoor (al aire libre) como en interiores. La Agencia de Protección Ambiental de Estados Unidos y la Administración de Seguridad y Salud Ocupacional (EPA y OSHA, por sus siglas en inglés, respectivamente) son algunos de ellos, ya que proveen de importante información a la hora de saber qué niveles de contaminación son permitidos para una buena salud ambiental.

Es importante que todas las edificaciones, en especial los hospitales y centros de salud, tengan en consideración estas normas y regulaciones, a más de realizar estudios que monitoreen constantemente el ambiente en sus instalaciones, ya que a ellas acuden personas con problemas de salud muchas veces críticos, y una mala calidad del aire podría afectar seriamente su proceso de recuperación.

Tomando en cuenta estos datos, se ha decidido analizar qué niveles de contaminación pudieran presentarse en las instituciones hospitalarias, en especial las que atienden a pacientes con procesos infecciosos, problemas digestivos y respiratorios. Se ha escogido al HLB como el lugar idóneo para dicho análisis, puesto que cumple con los requerimientos antes mencionados y además se ha constatado que no cuenta con un sistema funcional de monitoreo de calidad del aire en sus ambientes interiores.

En función de esta necesidad se plantea un proyecto que pueda detectar los niveles de calidad de aire de este centro hospitalario, en áreas tales como quirófanos, sala de emergencia, pensionados, unidad de cuidados intensivos, etc., mediante una plataforma en la nube para almacenar datos, manteniendo la información actualizada y a la mano para posteriores consultas.

## **1.2 Antecedentes**

La Agencia de Protección Ambiental de Estados Unidos (EPA) asevera que las personas están cerca del 90% del tiempo en espacios interiores, y se exponen de esta forma a contaminantes del aire presentes en estos lugares, como son el material particulado, monóxido de carbono, entre otros (EPA, 2016). Esta agencia, junto con otros organismos (OSHA, por ejemplo), establece parámetros de calidad de aire, los cuales permiten saber qué niveles de contaminación son permitidos para el correcto desempeño de las personas en los distintos ambientes.

En los centros hospitalarios existen, en cambio, organismos internacionales que regulan todo lo relacionado a la climatización de sus instalaciones. Algunos de estos son: la Asociación Española de Normalización (UNE) en su normativa 100713:2005 (Instalaciones de acondicionamiento de aire en hospitales), la Sociedad Americana de Ingenieros de Calefacción, Refrigeración y Aire Acondicionado (ASHRAE, por sus siglas en inglés), entre otros. Estos determinan las condiciones de temperatura, humedad y presión que debe tener cada una de sus áreas.

De acuerdo a un estudio realizado por la Universidad Politécnica Salesiana en marzo de 2019 en el HLB, se constató que dicha institución hospitalaria no tiene una herramienta que les permita obtener información de manera directa de los niveles de contaminación del aire, lo que genera un factor de riesgo para las personas que transitan diariamente por sus áreas y salas (Vasquez, Leonel, 2019). En la *Figura 1* se puede apreciar la cantidad aproximada de personas que transitan diariamente en este hospital, según el estudio ya mencionado, llegando a un total de 528 entre pacientes, familiares, personal que labora en el centro hospitalario, proveedores de insumos, entre otros.



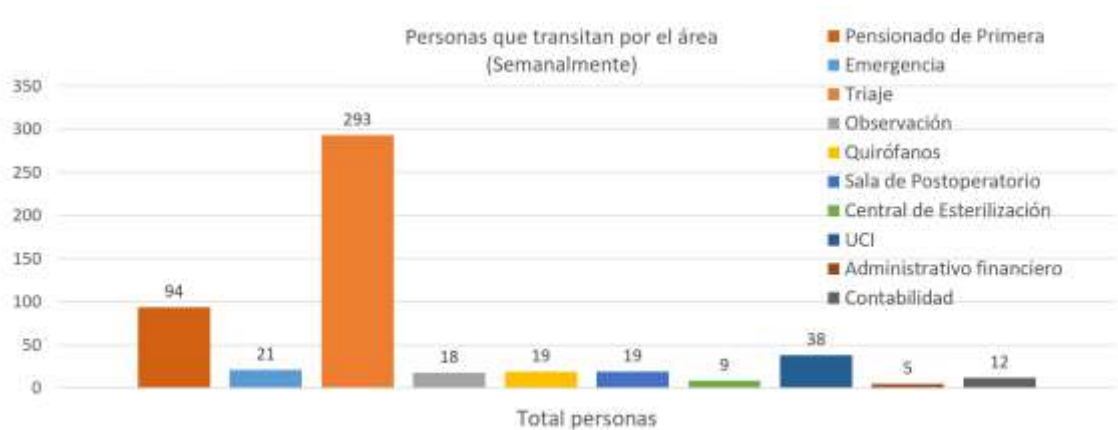


Figura 1. Personas que transitan semanalmente en el HLB. Fuente: elaboración propia.

En la Figura 2 se observa las áreas que se consideran importantes para monitorear la calidad del aire, según una encuesta realizada dentro del mismo estudio (Vasquez, Leonel, 2019). El eje vertical muestra la cantidad de personas que participan dando su opinión por cada sala, según la escala que va desde *innecesario* (color negro) hasta *Urgente* (color rojo).



Figura 2. Áreas del HLB consideradas para monitorear la calidad del aire. Fuente: elaboración propia.

Por otro lado, en el campo del Internet de las Cosas, en el país se ha observado un bajo conocimiento acerca de la importancia de utilizar tecnologías para monitorear en tiempo real variables ambientales en interiores, así como también la poca existencia de equipos y sistemas de medición directa de estos datos; lo que genera un problema de limitación de uso, análisis, detección y presentación de parámetros ambientales para controlar y mejorar el confort y la calidad de aire en ambientes indoor.

### 1.1 Importancia

Este proyecto es importante porque brinda elementos para obtener información técnica ambiental con la cual indicar niveles de contaminación en las salas seleccionadas del HLB. Al utilizar una plataforma en la nube y una aplicación web, se puede acceder en tiempo real a dicha información, lo que abre la posibilidad de establecer políticas de mejora continua, contribuyendo así al mejoramiento de la calidad del servicio que el hospital ofrece a sus pacientes y al personal que labora en él.

El proyecto en cuestión tiene una importancia relevante, ya que será un primer paso para acciones o proyectos futuros con impacto social y tecnológico, puesto que acoge tecnologías de código abierto, siendo este un vehículo apropiado para dinamizar las auditorías ambientales internas en el hospital, así como la adopción temprana de tecnologías, sostenibilidad y reconocimiento nacional.

## 1.2 Alcance

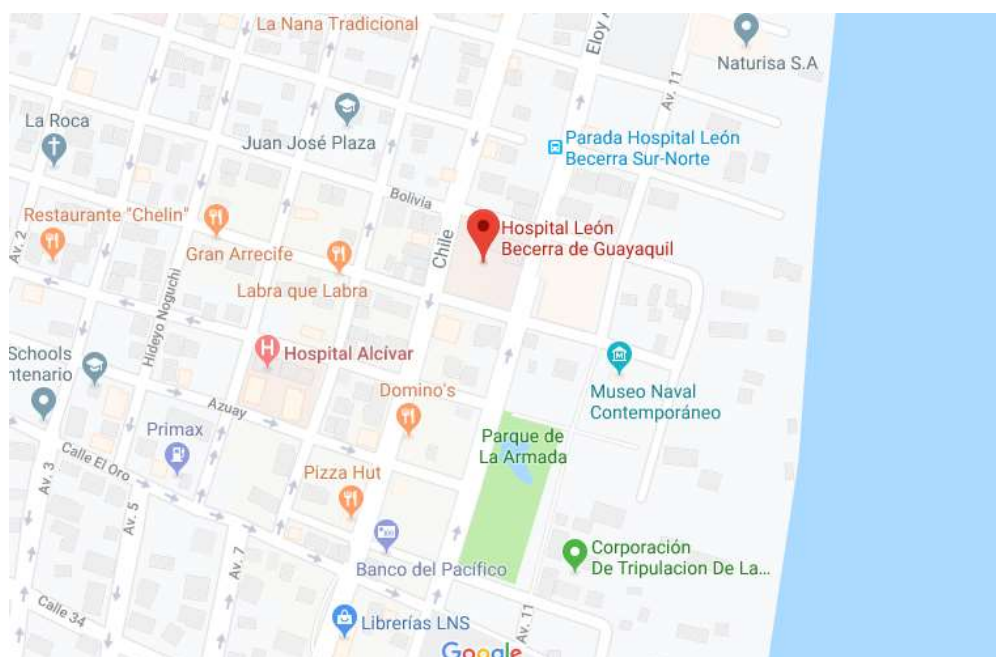
El alcance del presente proyecto de titulación es:

- Implementar una placa capaz de medir el CO, temperatura y humedad del aire a través de sensores.
- Crear un sistema de comunicación entre la placa de sensores y un gateway para enviar los datos obtenidos a internet.
- Almacenar y administrar los datos enviados por medio de una base de datos en la nube.
- Crear una aplicación web para visualizar los resultados almacenados.
- Confrontar la información visualizada con los parámetros establecidos para una buena calidad del aire en ambientes interiores.

## 1.3 Delimitación

### 1.3.1 Espacial.

La implementación del presente proyecto de titulación será realizada en las instalaciones del Hospital de Niños León Becerra, el cual se encuentra ubicado en las calles Chile entre Bolivia y Vacas Galindo, tal como se muestra en la *Figura 3*.



*Figura 3.* Ubicación del Hospital de Niños León Becerra. Fuente: Google Maps, 2019.

### **1.3.2 Temporal.**

EL presente proyecto de titulación será efectuado en el periodo de noviembre del 2018 a noviembre del 2019.

### **1.3.3 Académica.**

El presente proyecto de titulación requiere tener los conocimientos necesarios en sistemas de microprocesadores, electrónica analógica y digital, así como también en redes de computadoras, conocimientos adquiridos durante el flujo académico de la Universidad Politécnica Salesiana, en la carrera de Ingeniería en Electrónica, con énfasis en Telecomunicaciones.

## **1.4 Objetivos**

### **1.4.1 Objetivo general.**

Diseñar y desarrollar un prototipo de sistema de monitoreo de calidad de aire bajo una temperatura lot en la nube para la detección de los niveles de contaminación de aire en las salas del Hospital de Niños León Becerra.

### **1.4.2 Objetivos específicos.**

- Realizar las mediciones de las variables en las áreas del HLB con los sensores de humedad, temperatura y CO.
- Diseñar e implementar la estructura de red inalámbrica de sensores y la conexión a la red IP.
- Diseñar e implementar el sistema de monitoreo de calidad de aire en el HLB usando la plataforma IoT.
- Desarrollar la base de datos que contendrá toda la información.
- Analizar los datos obtenidos y contrastarlos con los parámetros de calidad de aire establecidos por organismos competentes.

## **2. Fundamentos Teóricos**

### **2.1 Internet de las Cosas (IoT)**

El Internet of Things (IoT), conocido en español como el internet de las cosas, consiste de forma simple en dotar de conectividad a internet a cualquier dispositivo.

Antiguamente, cuando se tenía un dispositivo de medición, este almacenaba la información en el mismo dispositivo para posteriormente copiarla a alguna computadora. Este mecanismo se vuelve complicado de manejar cuando se tiene grandes cantidades de información y se desea realizar su análisis.

Al dotar de internet a un dispositivo (como un sensor, por ejemplo), se puede mandar la información obtenida en tiempo real a una base de datos en la nube. Esta dispone de un almacenamiento mucho más grande y robusto en la cual se puede almacenar más información y de forma más segura.

La principal ventaja de centralizar toda la información en la nube es que los datos se encuentran en un solo lugar y siempre disponibles desde cualquier parte del mundo.

En la actualidad existen muchos dispositivos que se conectan a internet para enviar información a servidores remotos (en la nube) con la finalidad de poder brindar una mejor experiencia de usuario.

## 2.2 Elementos de IoT

La tecnología IoT requiere de ciertos elementos para poder funcionar. Estos elementos se basan en la interacción de un objeto que comúnmente está presente en nuestra vida cotidiana con un dispositivo electrónico (*hardware*), configurado por medio de *software* para una tarea específica, y que pueda conectarse a Internet, de manera que podamos visualizar su desempeño.

### 2.2.1 Hardware.

#### 2.2.1.1 Sensores.

Los sensores son dispositivos capaces de leer magnitudes del mundo real y convertirlas en magnitudes digitales, es decir puede digitalizar estas señales medidas para su posterior manipulación.

Los sensores *Figura 4* existen de diversos tipos, formas y mecanismos de medición, pero en términos simples se tratan de dispositivos que varían su voltaje o corriente en forma proporcional a su magnitud física o química, (llamadas variables de instrumentación), es decir producen una señal medible por un microcontrolador la cual es proporcional a una variable del mundo real.



*Figura 4.* Sensores. Fuente: ShopMaster  
(<https://shop.master.com.mx/product/model/AR-KITSENSOR>)

#### 2.2.1.2 Microcontroladores.

Los microcontroladores (*Figura 5*) son circuitos integrados programables que almacenan una serie de instrucciones en su memoria para ejecutarlo posteriormente. Los microcontroladores están compuestos principalmente por una CPU, memoria y periféricos de entrada y salida.



*Figura 5.* Microcontrolador PIC16F877A. Fuente: Quantica.

(<https://quanticaelectronics.com/catalogo/semiconductores-es-es/pic16f877-cmos-microcontrolador-microchip/>)

### **2.2.1.3 Arduino.**

Arduino es una compañía open source (2.2.2.1) y open hardware que fabrica placas microcontroladas llamadas también Arduino, las cuales se adaptan según la necesidad del usuario. Cuenta también con un entorno de desarrollo que tiene su propio lenguaje de programación de alto nivel (llamado igualmente Arduino), el cual abstrae todas las complejidades de los lenguajes para microcontroladores.

El lenguaje de programación de Arduino es un derivado de Wring, que a su vez es un lenguaje para microcontroladores basado en C/C++. Las principales ventajas que ofrece Arduino son:

- Bajo costo
- Multiplataforma
- Gran comunidad
- Ecosistema de librerías y placas
- Código Abierto

Al ser un proyecto Open Source y Open Hardware permite conocer cómo está hecho y cómo funciona cada uno de sus componentes, lo que permite que muchas personas puedan diseñar sus propias placas compatibles con Arduino, conocidas como placas expansoras o “shields”, como se observa en la *Figura 6*:



*Figura 6.* Shield de ethernet para Arduino uno. Fuente: Alselectro.

(<http://www.alselectro.com/ethernet-shield.html>)

### **2.2.1.4 Waspnote.**

Waspnote (*Figura 7*) es una placa microcontrolada basada en Arduino, que ha sido creada por la empresa Libelium para su uso en escenarios reales (ciudades, edificaciones y proyectos ambientales, entre otros), a diferencia de las placas Arduino que buscan ser más didácticas, empleándose mayormente en proyectos de bajo costo y de uso más bien doméstico.

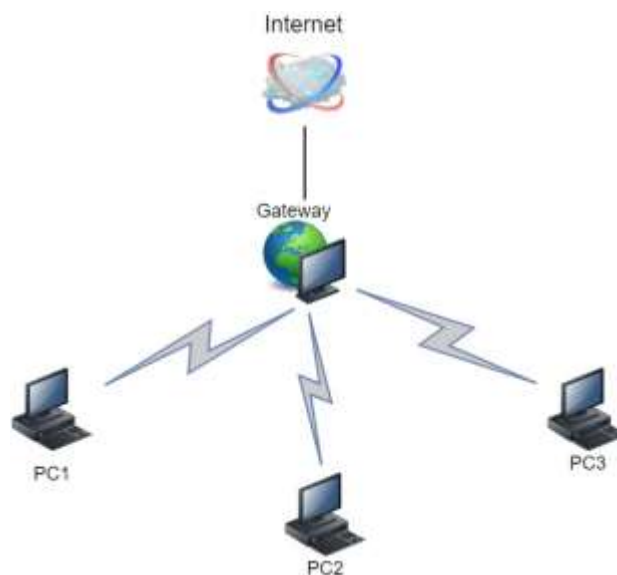


*Figura 7.* Placa Wasp mote. Fuente: Libelium.  
([http://www.libelium.com/expansion\\_radio\\_board/](http://www.libelium.com/expansion_radio_board/))

Wasp mote usa el mismo lenguaje de Arduino para su programación, lo que hace que sea compatible con casi todas las librerías existentes de Arduino. Esto le dota de un ecosistema muy grande de librerías y placas expansoras, facilitando su uso.

#### **2.2.1.5 Gateway.**

Un gateway, también llamado puerta de enlace, es un dispositivo de frontera en una red. Esto quiere decir que tiene como principal función comunicar una red con otra u otras, sin importar los protocolos y arquitectura que puedan usar estas redes. Por lo general, el gateway conecta redes LAN (domésticas) a la red de internet. En la *Figura 8* se puede apreciar el funcionamiento básico de un gateway en una topología estrella.



*Figura 8.* Funcionamiento de un gateway. Fuente: Superuser.  
(<https://superuser.com/questions/1390533/virtualbox-gateway-connection>)

Es importante conocer este término, ya que para el presente proyecto se utilizará un dispositivo que funcione como puerta de enlace para comunicación a internet. En la *Figura 9*, se observa como diversos sensores se conectan con un dispositivo central (gateway) usando diferentes protocolos. Este gateway es el encargado de enviar la información a la nube a través de internet, la cual será almacenada en una base de datos para su posterior consumo desde un aplicativo web en el dispositivo final (PC).



*Figura 9.* Arquitectura genérica de una solución de IoT. Fuente: Influxdata. (<https://www.influxdata.com/blog/building-an-influxdb-iot-edge-data-collection-device/>)

### **2.2.1.6 Raspberry**

Raspberry (*Figura 10*) es una fundación nacida en el Reino Unido con la finalidad de llevar las ciencias de la computación a países de escasos recursos gracias a sus sistemas embebidos, los cuales cuentan con grandes prestaciones y costo reducido. Esta minicomputadora por sí sola es capaz de resolver problemas de cálculo, reproducir videos, procesar palabras, entre otras operaciones. En conjunto con otras placas microprocesadas (con Arduino, por ejemplo) se convierte en una opción viable para realizar proyectos más complejos, siendo una de las características más sobresalientes para el trabajo que realiza, el poder conectarse a internet y transmitir datos de manera inalámbrica.



*Figura 10.* Tarjeta Raspberry. Fuente: Raspberrypi (<https://www.raspberrypi.org>).

## **2.2.2 Software.**

### **2.2.2.1 Open Source.**

La iniciativa open source (*Figura 11*) es un movimiento fundado en 1998 que busca que el software sea completamente abierto, de manera que cualquier persona pueda consultar la forma en la que está hecho e incluso mejorarlo.



Figura 11. Logo de la iniciativa open source. Fuente: Open Source.  
(<https://opensource.org/logo-usage-guidelines>)

Existen programas que quizá alguna vez hemos utilizado y que han sido creados con open source o código abierto como son Mozilla Firefox, Open Office, 7-Zip, entre otros.

#### 2.2.2.2 Kernel.

El kernel de un sistema operativo es el programa de más bajo nivel que existe en una computadora, este software es el responsable de la interacción el hardware como se puede apreciar en la Figura 12.

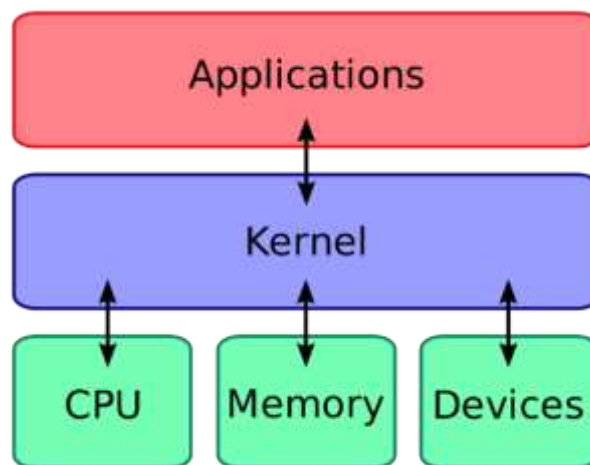


Figura 12. Arquitectura de un sistema operativo. Fuente: Digilentinc.  
(<https://blog.digilentinc.com/demystifying-the-linux-kernel/>)

Cuando un software desea hacer una vídeo llamada, el kernel (mediante los drivers) es el responsable de acceder a la cámara del dispositivo y leer los datos de audio y vídeo para que el software solicitante los pueda usar, de la misma forma también regula el acceso al disco duro, así como a la memoria RAM.

#### 2.2.2.3 Linux.

El kernel Linux es un kernel open source el cual fue creado por Linux Torvalds (Figura 13) durante su etapa universitaria. Este kernel surgió como un proyecto personal del sr. Torvalds, el cual buscaba tener más control sobre el software que se ejecutaba en una PC.





Figura 13. Linus Torvalds. Fuente: The New York Worker.

(<https://www.newyorker.com/science/elements/after-years-of-abusive-e-mails-the-creator-of-linux-steps-aside>)

Linux es actualmente el kernel más usado a nivel de servidores y teléfonos móviles en todo el mundo, esto es debido principalmente a su naturaleza de open source. Esto quiere decir que cualquiera puede personalizarlo para su propio uso sin tener que adquirir una licencia para ello; esto permitió que las grandes empresas descarguen Linux y lo personalicen para tener un uso más óptimo de acuerdo a su caso de uso.

Android es un sistema operativo para móviles basado en el kernel de Linux. Esto es debido a que Google “recortó” el kernel eliminando el soporte para periféricos, como lectores de cd, entre otros, y agregando otros soportes, como por ejemplo para pantallas táctiles y demás. En la *Figura 14* se puede apreciar la arquitectura de Android.



Figura 14. Arquitectura de Android. Fuente: How To Geek.

(<https://www.howtogeek.com/189036/android-is-based-on-linux-but-what-does-that-mean/>)

#### 2.2.2.4 Distro.

Distro es la abreviatura de la palabra distribución, y esta hace referencia a un sistema operativo basado en Linux. Dado que el ecosistema Linux es muy grande por su misma naturaleza, cualquier persona puede descargar el kernel de Linux y crear su propia distro a partir de cualquier cambio, por muy pequeño que sea.

En la *Figura 15* se puede apreciar algunas de las principales distros Linux; estas distros aparecen y desaparecen de forma esporádica. Sin embargo, existe una pequeña cantidad de distros que sobreviven al paso de los años, siendo cada vez más estables y usadas a medida que Linux gana terreno en el mundo del usuario final.



*Figura 15.* Principales distros Linux. Fuente: Nixwindows.  
[\(https://nixwindows.wordpress.com/2015/02/09/linux-distros/\)](https://nixwindows.wordpress.com/2015/02/09/linux-distros/)

### **2.2.2.5 Debian.**

Debian es un sistema operativo basado en Linux. Es uno de los sistemas operativos open source más antiguos y robustos que existe, siendo soportado por la Debian Foundation, fundación que se encarga del desarrollo de nuevas funcionalidades, así como del mantenimiento de las funciones existentes. En la *Figura 16* se puede observar su interfaz.



*Figura 16.* Sistema Operativo Debian. Fuente: Brian Linkletter.  
[\(https://www.brianlinkletter.com/installing-debian-linux-in-a-virtualbox-virtual-machine/\)](https://www.brianlinkletter.com/installing-debian-linux-in-a-virtualbox-virtual-machine/)

### **2.2.2.6 Raspbian.**

Es el sistema operativo que usa la Raspberry Pi, el cual está basado en Debian, que a su vez es una distro Linux. Este sistema operativo es de uso general, es decir no solo se puede usar con la Raspberry Pi, sino que se puede instalar en cualquier computadora de escritorio o servidor normal.

## **2.2.3 Internet.**

### **2.2.3.1 Nube.**

Se conoce a la nube como una red mundial de servidores en los cuales se mueven grandes cantidades de información. Estos servidores forman un ecosistema de servicios, los cuales permiten almacenar y administrar datos, ejecutar aplicaciones, entre otros.

Existen nubes públicas que ofrecen servicios gratuitos como Dropbox, ofreciendo 5Gb de almacenamiento gratuito; y luego hay servicios como Netflix que ofrecen un catálogo de películas y series (almacenadas en la nube) a cambio de una suscripción mensual a sus servicios.

### 2.2.3.2 Servidores web.

Un servidor web o servidor HTTP/HTTPS es un programa que se ejecuta en un servidor, el cual recibe peticiones web y delega a un aplicativo definido por el usuario la responsabilidad de generar una respuesta.

Para poder acceder a la información a través de internet, es necesario que los servidores donde se encuentran los aplicativos tengan instalado un software llamado servidor web. Este software es el responsable de exponer los servicios a la web; de allí viene su nombre de servidor web.

El navegador realiza peticiones HTTP y HTTPS, las cuales por defecto van al puerto 80 y 443 respectivamente. El servidor web recibe estas peticiones y genera respuestas para cada una de estas peticiones.

Los puertos web, en cambio, son canales de comunicación por los cuales viaja la información. Un servidor puede tener hasta 65536 puertos disponibles, debido a una limitante del diseño de las computadoras. Este valor se justifica a raíz de que el espacio designado para la especificación de un puerto es de un “word” o palabra (2 bytes). Como sabemos que cada byte contiene 8 bits, tenemos un total de 16 bits para especificar el puerto a través del cual se realiza la conexión, y 2 elevado a la 16 da como resultado los 65536 puertos disponibles.

Todos los puertos inferiores al 1024 son puertos considerados como “conocidos”, es decir, puertos que son usados por el sistema operativo y que no deben ser utilizados por los aplicativos del proyecto sino por servicios previamente definidos. Los puertos del 1024 al 49151 son puertos que se conocen como “registrados”, ya que son puertos que pueden ser usados por cualquier aplicación. Finalmente, los puertos desde el 49151 hasta el 65535 son los puertos dinámicos, siendo estos puertos asignados dinámicamente a las aplicaciones de tipo cliente que inician algún tipo de conexión con un servidor.

Algunos de los principales servicios web se pueden ver en la *Tabla 1*.

*Tabla 1.* Principales servicios web y sus puertos.

Protocolo	Puerto	Descripción
21	ftp	Puerto del Protocolo de transferencia de archivos
22	Ssh	Servicio de shell seguro
23	telnet	El servicio Telnet
25	Smtp	Protocolo simple de transferencia de correo
80	http	Protocolo de transferencia de hipertexto para los servicios del World Wide Web
109	pop2	Protocolo Post Office versión 2

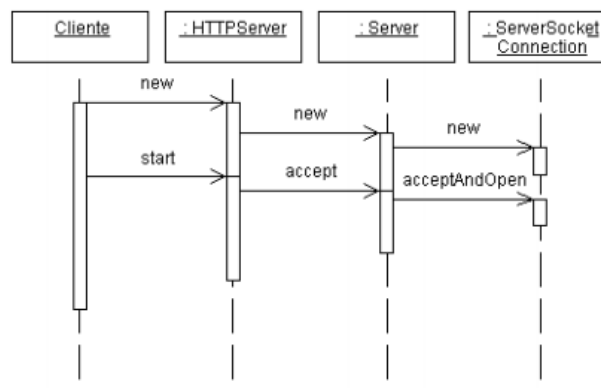
110	pop3	Protocolo Post Office versión 3
123	Ntp	Protocolo de tiempo de red
143	Imap	Protocolo de acceso a mensajes de Internet
161	snmp	Protocolo simple de administración de redes
443	https	Protocolo de transferencia de hipertexto seguro

Fuente: elaboración propia.

### 2.2.3.3 Peticiones HTTP/HTTPS.

Para comunicarse a través de internet todos los clientes y servidores deben tener un protocolo de comunicación en común, para que se puedan entender entre sí. Para comunicaciones web el protocolo estándar es el protocolo HTTP, siendo este protocolo el que define cómo se envían y se reciben mensajes del servidor.

El protocolo HTTP es un protocolo sin sesión, es decir, el servidor procesa la petición recibida y genera una respuesta. Luego de esto inmediatamente termina la petición, tal como se puede apreciar en la *Figura 17*.



*Figura 17.* Comunicación HTTP (Guillermo Diez-Andino Sancho, 2003).

### 2.2.3.4 HTML.

HTML es un lenguaje de etiquetas muy similar a XML, como se puede apreciar en la *Figura 18*. HTML, más que un lenguaje de programación, es un lenguaje de marcado.

```

<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
  
```

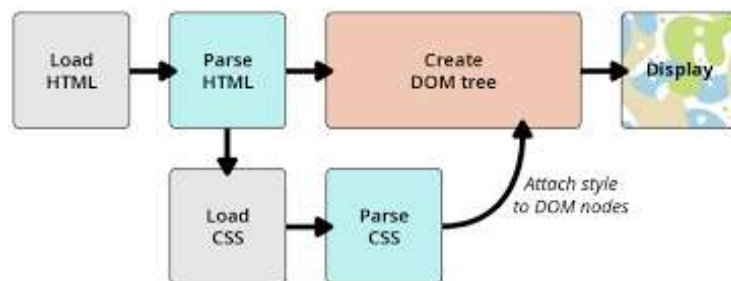
*Figura 18.* Ejemplo de programa escrito en HTML. Fuente: elaboración propia.

HTML permite definir la estructura de una página web mediante sus principales atributos (header, body, etc.). Solo se usa para definir la estructura de un sitio web, mas no para modificar el aspecto visual del mismo.

### 2.2.3.5 CSS.

CSS son las siglas de Cascade Style Sheets, conocido en español como hojas de estilos en cascadas. Es un lenguaje de estilos, siendo el responsable de toda la parte visual de un sitio web. Esto quiere decir que los colores y formas que se ven en un sitio web están definidos en los archivos de estilo CSS.

Cuando se abre un sitio web, el navegador descarga el HTML que forma dicho sitio. En el HTML vienen las referencias a los archivos CSS, es decir, el navegador primero pasa el HTML y con este HTML listo, procede a cargar el CSS. Mientras descarga el CSS, el navegador crea el árbol de estructura o “DOM tree”, tal como se puede apreciar en la *Figura 19*.



*Figura 19.* Diagrama de carga de sitios web. Fuente: developer.mozilla.org. ([https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction to CSS/How CSS works](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction%20to%20CSS/How%20CSS%20works))

El CSS es parte fundamental del desarrollo web moderno, ya que hoy más que nunca la apariencia de un sitio web resulta indispensable para “enganchar” a los usuarios. El CSS se forma de un selector (el HTML) sobre el cual se aplicará el estilo, de una o varias propiedades y de valores para aplicar a este estilo, tal como se puede apreciar en la *Figura 20*.



*Figura 20.* Estructura del CSS. Fuente: support.wordpress.com. (<https://en.support.wordpress.com/custom-design/css-basics/>)

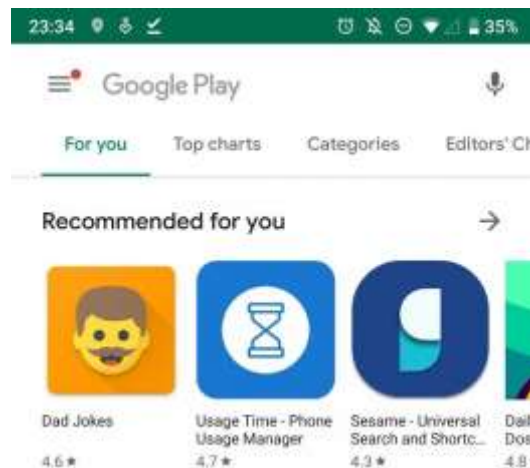
### 2.2.3.6 Javascript.

Javascript es el lenguaje de programación más utilizado para la web. Esto se debe principalmente a que es el único lenguaje soportado de forma nativa por los navegadores web. Es el complemento ideal para CSS y HTML, ya que estos son los

responsables del estilo y estructura de un sitio web, mientras que javascript es el encargado de brindar la interactividad al sitio en cuestión.

### 2.2.3.7 Aplicaciones Nativas.

Se conoce como aplicaciones nativas a las aplicaciones que se ejecutan directamente por el sistema operativo sin la interacción de un intermediario, es decir, se instalan y ejecutan directamente en el equipo. Un ejemplo de ello son las apps que se descargan de la tienda de aplicaciones de los dispositivos móviles, como las que se puede apreciar en la *Figura 21*.



*Figura 21.* Tienda de aplicaciones de Google. Fuente: elaboración propia.

Las aplicaciones nativas siempre han tenido una ventaja respecto a otro tipo de aplicaciones, ya que tienen acceso total a las funciones del dispositivo. Las principales ventajas de las aplicaciones nativas son:

- Mayor velocidad de ejecución
- Mayor capacidad de acceso a funcionalidades del dispositivo
- Mejor segmentación del mercado al tener aplicativos independientes para cada plataforma.

### 2.2.3.8 Aplicaciones Web.

Las aplicaciones web son todas las aplicaciones que se ejecutan desde el navegador web de cualquier dispositivo.

Las aplicaciones web ofrecen la ventaja de que solo es necesario escribir un programa para que el navegador las ejecute, mientras que las aplicaciones nativas tienen que ser programadas para cada plataforma sobre la que se espera que se ejecuten (IOS, Mac, Windows, Linux, entre otras).

En muchos casos, hacer que una aplicación nativa se ejecute entre diferentes sistemas operativos requiere reescribir casi al 100% el aplicativo. Incluso entre varias versiones del sistema operativo muchos aplicativos dejan de funcionar; esto hace que estas aplicaciones sean más difíciles y costosas de desarrollar. Debido a esto en la actualidad las empresas comienzan a adoptar aplicativos webs.

Como se puede apreciar en la *Figura 22* las principales ventajas de las aplicaciones web son:

- Bajo coste de desarrollo en relación a las aplicaciones nativas
- Reutilización de programación entre diferentes plataformas
- Mayor facilidad para implementar nuevas funcionalidades.



*Figura 22.* Comparación entre aplicaciones web y nativas. Fuente: SinLinos. (<https://sinlios.com/2013/06/07/apps-nativas-vs-web-apps/>)

Las aplicaciones web permiten un uso mucho más sencillo del software, puesto que estos aplicativos no requieren instalación, sino que se ejecutan a través del navegador web, permitiendo que los usuarios los prueben sin ningún tipo de inconvenientes.

En una página web el usuario puede simplemente ver su contenido, mientras que, en una aplicación web, el usuario puede interactuar, guardar sus preferencias y ver un contenido diferente en base a su perfil. Un ejemplo claro de ello son las redes sociales, donde cada usuario puede ver sus fotos y las fotos de sus amigos, pero como cada usuario tiene un círculo social diferente, todos los usuarios terminan viendo una sección de noticias con contenido diferente.

### 2.2.3.9 NodeJS.

Anteriormente javascript solo se ejecutaba en el navegador, es decir, era un lenguaje del lado del cliente. Sin embargo, debido al gran crecimiento de las aplicaciones web, crecieron también los desarrolladores de javascript y las librerías open source del mercado. Este crecimiento impulsó a Google a separar el motor v8 de javascript del navegador web y convertirlo en un software independiente, permitiendo que, instalar este motor en un servidor, sea suficiente para poder ejecutar javascript del lado del servidor y ya no solo del navegador web o cliente.

A partir de ese cambio, el motor v8 de javascript pasó a llamarse NodeJS y es lo que permite usar el mismo lenguaje de programación tanto del lado del servidor como del lado del cliente. Javascript es en la actualidad el lenguaje más utilizado para desarrollar software, como se puede observar en la *Figura 23*. Esto es debido principalmente a que con javascript se puede crear:

- Aplicaciones de escritorio con Electrón,
- Aplicaciones web con Vue,
- Aplicaciones móviles con React Native,
- Aplicaciones en el servidor con NodeJS, entre otros.



Figura 23. Lenguajes de programación más usados. Fuente: Theapicompany.com. (<https://trends.google.com/trends/explore?geo=US&q=%2Fm%2F02p97,%2Fm%2F060kv,%2Fm%2F06ff5>)

### 2.2.3.10 *TypeScript.*

TypeScript es un superset de javascript creado por Microsoft con la visión de traer la programación orientada a objetos a las aplicaciones web.

TypeScript no es un lenguaje de programación nuevo, es un superset de javascript, como se puede apreciar en la Figura 24. Esto quiere decir, que todo lo que se escribe en TypeScript al final es convertido a javascript para ser ejecutado por NodeJS o por el navegador. Este proceso de conversión es llamado transpilación y dentro del mundo de las aplicaciones web es un proceso muy común.

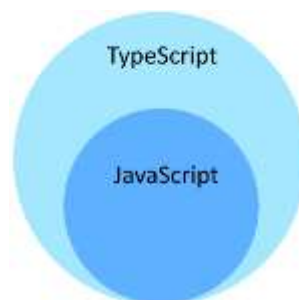


Figura 24. Typescript un superset de Javascript. Fuente: Los techies. (<https://lostechies.com/derekgreer/2016/08/30/exploring-typescript/>).

Javascript es un lenguaje orientado a eventos, ya que usa estructuras de datos sencillas para almacenar la información. TypeScript permite que tengamos en javascript lo mejor de la programación orientada a objetos, teniendo entre sus principales ventajas las que se enumeran a continuación:

- Clases
- Interfaces
- Enums
- Herencia
- Tipado de datos

### 2.2.3.11 *ECMAScript 6.*

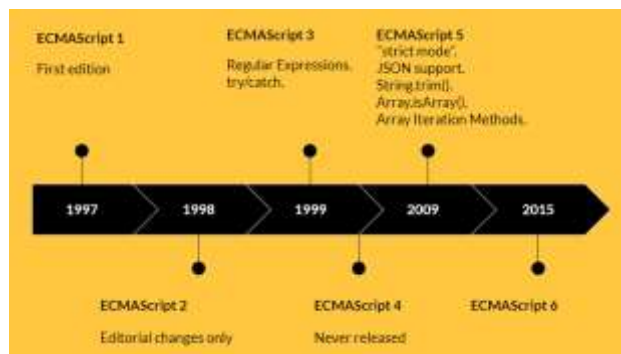
En el mundo de las aplicaciones web existen muchos navegadores web y no existe algún ente u organismo que los regule. Esto hace que cada navegador implemente las



funcionalidades que quiere y como las quiere de forma arbitraria. Ante este tipo de problemáticas se crearon los estándares, siendo ECMAScript el estándar de javascript y ECMAScript 6, su versión más reciente.

ECMAScript no es más que una definición de cómo debe funcionar javascript; y aunque ya existe ECMAScript 9, el cual implementa cientos de mejoras muy significativas, en la actualidad todos los navegadores implementan únicamente ECMAScript en su versión 5.1.

Como se puede ver en la *Figura 25* la versión 5 apareció en el 2009, por lo que actualmente este estándar se encuentra 10 años atrasado. Por esta razón es indispensable la transpilación y así poder usar las últimas bondades de ECMAScript 6 en navegadores que no implementan este estándar aún.



*Figura 25.* Timeline de EcmaScript. Fuente Blog.quadiontech.com.  
(<https://blog.quadiontech.com/have-you-heard-about-ecmascript-6-find-out-whats-new-in-javascript-es6-part-1-13a82a97adc5?gi=ca9c9cc09b1d>)

### 2.2.3.12 Transpilación.

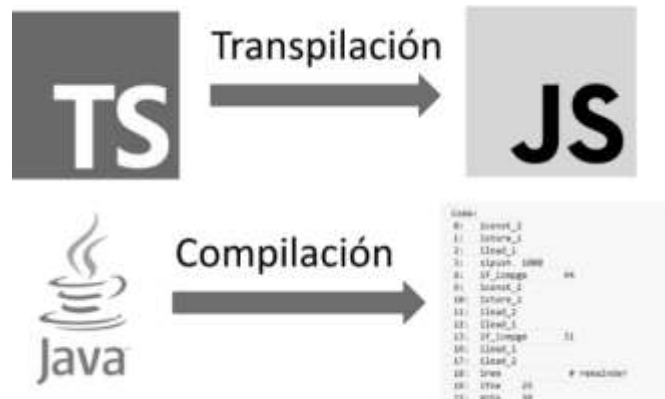
En el mundo de las aplicaciones web, donde los únicos lenguajes que soporta el navegador son HTML, CSS y javascript, los desarrolladores se encuentran muy limitados en cuanto a las opciones para desarrollar programas. Debido a esto surgieron formas de hacer código compatible con los navegadores, como es la transpilación, proceso por el cual se escribe código en un lenguaje para posteriormente “convertirse” a otro lenguaje.

En la actualidad existen cientos de opciones de lenguajes que se transpilan a HTML, CSS y javascript, aportando cada uno con ventajas y desventajas. Sin embargo, es importante para los desarrolladores tener un abanico de opciones para escoger la que mejor se adecúe a sus necesidades. Las opciones más comunes de transpilación en la actualidad son:

- Typescript a Javascript
- ECMAScript 6 a Javascript
- Pug a HTML
- SCSS a CSS

La diferencia entre transpilación y compilación es principalmente que la compilación convierte un programa a lenguaje de máquina, mientras que la

transpilación convierte un lenguaje de programación a otro del mismo tipo, tal como se puede apreciar en la *Figura 26*.



*Figura 26.* Transpilación vs compilación. Fuente: Ingenieriadesoftware. (<https://ingenieriadesoftware.es/diferencia-transpilacion-compilacion/>)

### 2.2.3.13 Babel.

Babel es el transpilador de ECMAScript 6 a Javascript más usado actualmente, puesto que permite usar las últimas novedades de javascript en navegadores que aún no implementan este estándar.

En la *Figura 27* se muestra un ejemplo de cómo se realiza la conversión de una función lambda a la función tradicional que soportan todos los navegadores web. Esta función lambda es una novedad de ECMAScript 6



*Figura 27.* Transpilación de una función lambda. Fuente: SessionStack. (<https://blog.sessionstack.com/how-javascript-works-the-internals-of-classes-and-inheritance-transpiling-in-babel-and-113612cdc220>)

### 2.2.3.14 Webpack.

En la antigüedad cuando se hacía un proyecto web, se escribían diferentes módulos de CSS y Javascript y estos eran incluidos en el proyecto de forma individual como se puede ver en la *Figura 28*.

```

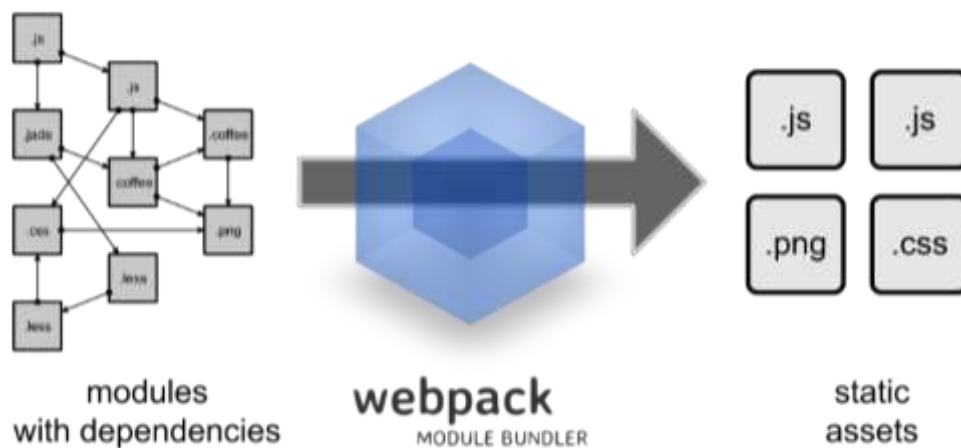
14 <html <?php language_attributes(); ?>>
15 <head>
16     <meta charset="<?php bloginfo( 'charset' ); ?>" />
17     <meta name="viewport" content="width=device-width, initial-scale=1" />
18     <script type="text/javascript" src="/scripts/myJavaScript.js"></script>
19     <link rel="profile" href="https://gmpg.org/xfn/11" />
20     <?php wp_head(); ?>
21 </head>

```

*Figura 28.* Edición de código fuente en HTML y JavaScript. Fuente: WP Buffs. (<https://wpbuffs.com/edit-source-code-in-wordpress/>)

Conforme creció la web, se creaban muchas librerías javascript y un proyecto web podía estar formado por hasta 50 librerías. Al ser el navegador el responsable de descargar de forma individual todas estas librerías, se producía un retraso muy grande a la hora de cargar la página, entorpeciendo la experiencia de usuario.

Debido a esto aparece Webpack, el cual es responsable de unir todos los javascript y CSS en uno solo (o varios dependiendo de la configuración) para que el navegador no necesite realizar varias descargas sino que los pueda descargar todos de una sola vez, tal como se puede apreciar en la *Figura 29*. A más de esto, mediante plugins como el de Babel se puede agregar el transpilado de código, es decir, que transpile todos los archivos que no sean ECMAScript 5 y luego de eso unirlos en un solo archivo para ser enviado al navegador.



*Figura 29.* Funcionamiento de Webpack. Fuente [Webpack.github.io](https://webpack.github.io/).  
(<https://webpack.github.io/>)

### 2.2.3.15 Pug.

Antiguamente la única forma de describir la estructura de las páginas web era HTML. Este lenguaje, al ser similar al XML resulta extenso de escribir; sin embargo, su principal problema era que no se podía reutilizar de forma eficiente, ya que la única forma de hacerlo era copiando y pegando el programa de un lugar a otro.

Cuando se escriben muchos sitios web, resulta imposible evitar la reutilización de programas. Sin embargo, HTML no permite usar módulos, haciendo de la reutilización una tarea imposible. Ante todos estos problemas surge Pug como una alternativa a la hora de simplificar la forma en que se escribe HTML, usando tabulaciones como lo hace Python, en lugar de abrir y cerrar tags (como lo hace tradicionalmente el HTML).

Como se puede apreciar en la *Figura 30*, en Pug es más compacto y sencillo escribir. Pero su principal ventaja es que se puede crear módulos y reutilizarlos entre los proyectos, con la finalidad de evitar rescribir el mismo código o funcionalidad en varias partes de nuestra aplicación.

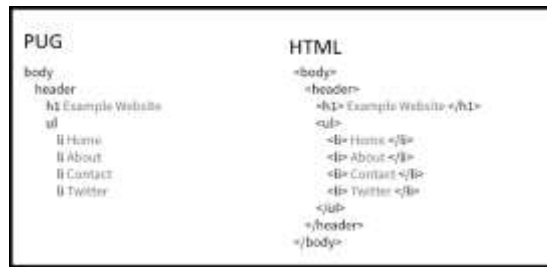


Figura 30. Pug vs Html. Fuente: Medium. ([https://medium.com/@\\_phzn/roots-a-delightful-static-site-generator-get-started-in-20-minutes-f7aaab43056d](https://medium.com/@_phzn/roots-a-delightful-static-site-generator-get-started-in-20-minutes-f7aaab43056d))

El navegador web no soporta de forma nativa pug, pero gracias a babel y webpack se puede usar esta tecnología, transpilándola a HTML.

### 2.2.3.16 SCSS.

De forma similar a lo que es pug para el HTML, tenemos Scss para el Css. Este es un nuevo lenguaje que, gracias a webpack y babel, se transpilará para que el navegador lo pueda entender, ya que no es soportado de forma nativa por él.

Css, a diferencia del HTML, sí permite la reutilización de programa, lo que permite crear módulos de Css independientes y utilizarlos dentro de diferentes proyectos. Sin embargo, es un lenguaje estático, es decir, no soporta variables ni lazos o funciones, por lo que, si se desea hacer que una acción se ejecute 6 veces, se la debe escribir 6 veces; o si se desea que varias partes del programa tengan un valor común, se debe escribir ese valor en todas las partes del programa.

Scss trae al Css variables, lazos y condicionales que permiten escribir un mejor y más eficiente Css, dándole más flexibilidad y modularidad. De la misma forma que con Pug, el Scss es transpilado a Css por webpack y babel.

En la *Figura 31* se puede apreciar un mismo ejemplo escrito en SASS, SCSS y CSS.



Figura 31. Comparativa entre SASS, SCSS y CSS. Fuente: Medium. (<https://medium.com/@raaechelb/css-precompilers-which-one-is-better-5e6ba831c0ed>)

### 2.2.3.17 Framework.

Un framework se puede traducir al español como marco de trabajo. Es un conjunto de reglas que forman un esquema o estructura para desarrollar y organizar un proyecto.

Cada implementación de un framework puede ser diferente, ya que no hay una definición de él como tal. El framework simplemente define las reglas de cómo se debe estructurar el proyecto que se va a llevar a cabo.

### 2.2.3.18 Vue.

Vue es un framework para desarrollar aplicaciones web. Este framework cambia la metodología de programación tradicional mediante la introducción de nuevos paradigmas de programación, como la reactividad, componentes, entre otros.

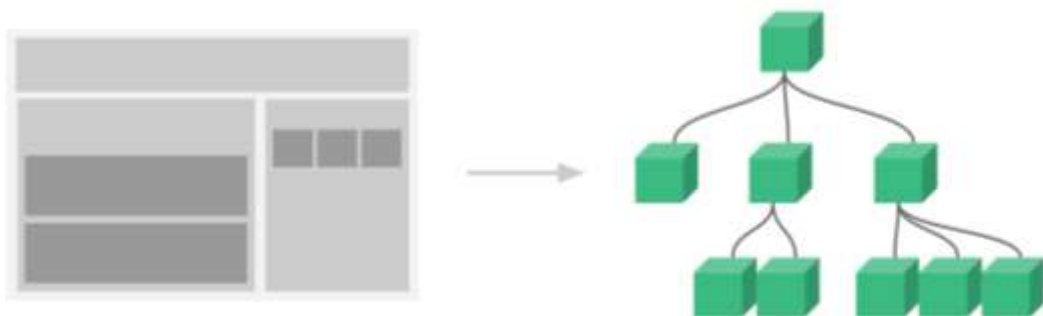
Vue implementa el mecanismo llamado reactividad, permitiendo mediante una serie de eventos que cuando se actualice el valor de una variable, se actualice el contenido de una vista de la aplicación, tal como se puede apreciar en la *Figura 32*.



*Figura 32.* Reactividad de Vue. Fuente: SAP Blogs.  
(<https://blogs.sap.com/2017/06/14/two-way-data-binding-ui5-vs-vue/>)

Este concepto facilita el desarrollo, ya que no es necesario modificar las vistas de la aplicación de forma manual cuando se realiza algún cambio, sino que Vue automáticamente actualiza las vistas cuando es necesario.

Además de la reactividad, vue implementa un sistema de componentes, los cuales permiten separar la interfaz en diferentes elementos. Estos elementos se estructuran como un diagrama de árbol, en una relación de padre e hijo. En la *Figura 33* se puede apreciar lo explicado:



*Figura 33.* Sistema de componentes de Vue. Fuente: VueJs.  
(<https://vuejs.org/v2/guide/components.html>)

### 2.2.3.19 Progressive Web App (PWA).

PWA son las siglas de Progressive Web Application, lo que en español significa Aplicación Web Progresiva. Este término hace referencia a las aplicaciones web que se ven y funcionan como aplicaciones nativas en un teléfono móvil o en una computadora.

La PWA es una tecnología que permite a las aplicaciones web tener casi todas las ventajas de una aplicación nativa y a su vez, casi todas las ventajas de una aplicación web. En pocas palabras, una PWA une lo mejor de ambos mundos.

En la *Figura 34* se puede observar las principales ventajas de una PWA.



*Figura 34.* Ventajas de las PWA. Fuente: Analista SEO.

(<https://www.analistaseo.es/posicionamiento-buscadoreseo-progressive-web-apps-pwa/>)

En la actualidad muchas empresas están lanzando aplicaciones PWA a la par que las aplicaciones nativas. Aplicaciones como Instagram, por ejemplo, no usan ninguna funcionalidad que solo se encuentre disponible de forma nativa, sino que pueden perfectamente ser una PWA sin que el usuario note la diferencia. En la actualidad Instagram dispone de una PWA, la cual se puede usar si se accede a Instagram desde la web como se aprecia en la *Figura 35*.



*Figura 35.* PWA de Instagram. Fuente Aglowiditsolutions.

(<https://aglowiditsolutions.com/blog/progressive-web-apps-development/>)

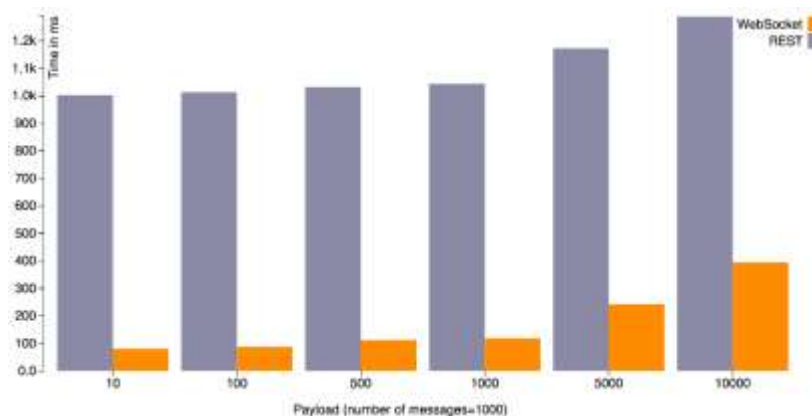
### 2.2.3.20 Sockets web.

Los sockets son mecanismos bidireccionales de comunicación entre cliente y servidor. Esto quiere decir que son un camino desde el cual el cliente puede realizar consultas al servidor y el servidor pueda enviar notificaciones al cliente.

Los sockets, de la misma forma que las peticiones HTTP, se usan para comunicar cliente y servidor. Sin embargo, los sockets a diferencia de las peticiones HTTP son un canal de comunicación bidireccional sobre el cual pasa información, mientras que en el protocolo HTTP cada vez que se desea enviar o recibir datos se debe establecer un nuevo camino de comunicación para dicha transacción.

Debido a que en los sockets ya hay un canal abierto, no es necesario el intercambio de mensajes de inicio y finalización de sesión como si se hace en HTTP. En su lugar, las respuestas llegan mucho más rápido y no es necesario crear nuevas conexiones.

Como se puede ver en la *Figura 36*, los tiempos de respuesta a las peticiones web mediante Sockets son mucho menor al de las peticiones HTTP mediante un API REST.



*Figura 36.* Tiempo de espera por respuestas. Fuente: Arungupta. (<http://blog.arungupta.me/rest-vs-websocket-comparison-benchmarks/>)

La desventaja de los sockets frente al tradicional HTTP es que mantener conexiones abiertas con el servidor resulta en un mayor consumo de memoria por parte del servidor, esto hace que se necesite más y mejores equipos para miles de conexiones sobre sockets de lo que haría falta en conexiones HTTP.

## 2.3 Aplicaciones IoT

A nivel de aplicaciones, la tecnología IoT ha dado pasos acelerados, logrando incursionar en múltiples campos, favoreciendo la interacción de herramientas de uso diario con las personas. Esta tecnología también busca solucionar de manera eficiente problemas reales que se suscitan en el entorno, algunos de los cuales se mencionan a continuación.

### 2.3.1 Reducción de contaminación acústica en Cuenca.

Este proyecto da solución a un problema que afecta a muchas ciudades del país y del mundo: la contaminación acústica. Por medio de una red de sensores inalámbricos,

se estructuró un sistema de medición, colocando 7 nodos en distintas partes de la ciudad para crear un mapa de ruido. Se configuraron y compararon los sensores del proyecto con un sensor de ruido certificado para verificar que la información a obtener sea confiable. Se envió la información adquirida a una página web para que cualquier persona pueda visualizarla, incluso los servidores públicos a los que les compete determinar acciones en base a los resultados obtenidos (Libelium, 2019).



Figura 37. Diagrama de instalación del nivel de sensor de ruido. Fuente: Libelium World ([http://www.libelium.com/libelium-helps-to-reduce-noise-pollution-in-the-city-of-cuenca-ecuador/#!prettyPhoto-img\[46120\]/1/](http://www.libelium.com/libelium-helps-to-reduce-noise-pollution-in-the-city-of-cuenca-ecuador/#!prettyPhoto-img[46120]/1/)).

### 2.3.2 Sistema de monitoreo de calidad del aire en Mallorca.

Este proyecto está dirigido a monitorear la calidad del aire en el puerto de Palma, Mallorca, realizando análisis de monóxido de carbono, material particulado, ozono, dióxido de carbono, dióxido de azufre y ruido, en intervalos de 10 minutos, por medio de la instalación de plataformas de sensores en distintos puntos de la isla (Figura 38).



Figura 38. Plataforma de sensores instalados. Fuente: Libelium World (<http://www.libelium.com/wp-content/uploads/2018/12/mallorca12.jpeg>)



Después de más de un año de poner en funcionamiento los nodos, se llegó a la conclusión de que los niveles de contaminación no se deben en su mayoría a los cruceros (como suponían en un principio), sino a las embarcaciones más pequeñas que tienen fines comerciales o de transporte de pasajeros.

El proyecto contribuyó al establecimiento de políticas para corregir los niveles de contaminación del aire como, por ejemplo, facilitarles energía a las embarcaciones a menor costo, entre otras (Libelium, 2018).

### 2.3.3 Sistema de alerta temprana para evitar inundaciones en ríos.

El sistema tiene como principal objetivo monitorear el comportamiento de los ríos colombianos de La Liboriana, La Clara y Barroso, ya que, en el año 2015, ocurrió el desbordamiento del primero, causando la muerte de alrededor de 80 personas y muchas pérdidas materiales.

El proyecto cuenta con una red de 5 sensores alimentados con paneles solares que monitorean el nivel de agua de los ríos, junto con la temperatura del aire, tomando muestras cada 5 minutos. Este sistema cuenta además con sirenas que alertan a la población en caso de necesidad, para que puedan evacuar a tiempo los sectores aledaños. En la *Figura 39* se puede observar el diagrama de comunicación del proyecto, que incorpora una Raspberry Pi para enviar datos a la nube (Twilio), además de un sistema de comunicación propio, puesto que el área contaba con cobertura 3G, presentando muchas limitaciones a la hora de transmitir los datos (Libelium, 2017).

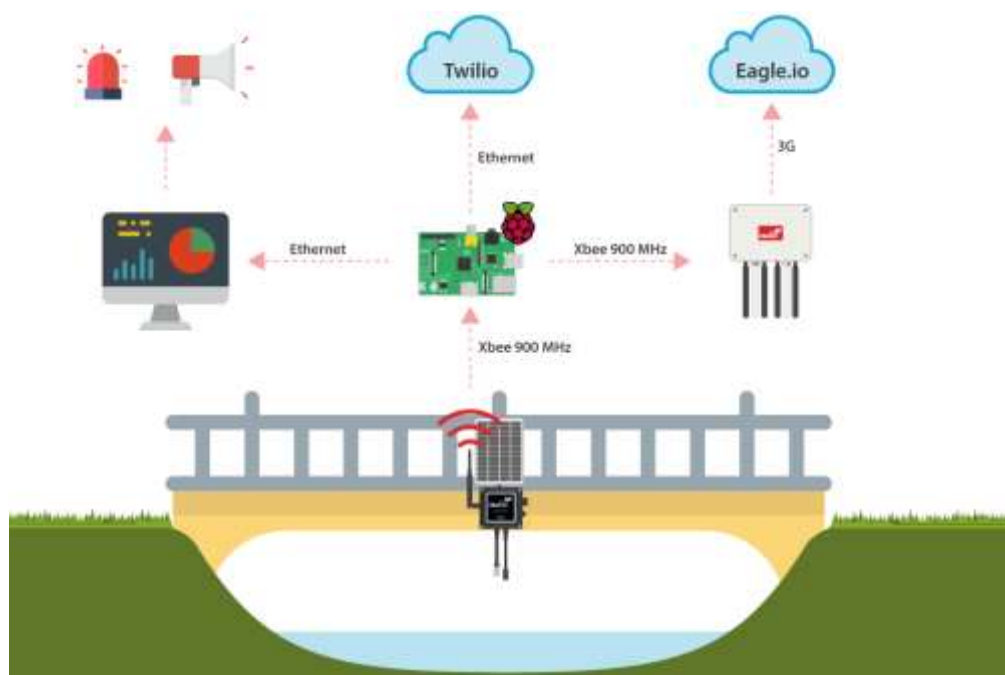


Figura 39. Diagrama de comunicación del proyecto. Fuente: Libelium World ([http://www.libelium.com/wp-content/uploads/2017/12/diagrama\\_colombia\\_1100.png](http://www.libelium.com/wp-content/uploads/2017/12/diagrama_colombia_1100.png)).

## **2.4 Calidad del Aire**

En la actualidad, todos los países del mundo conviven con un problema que afecta en gran medida a sus habitantes: la contaminación ambiental. Dicho problema repercute, tanto en los países del primer mundo como en los que se encuentran en vías de desarrollo. (Ministerio del Ambiente, 2012).

Los países desarrollados tienen este problema por su alto índice de industrias y empresas dedicadas a la producción, a más de la gran cantidad de vehículos y maquinarias. Los países en vías de desarrollo, en cambio, lo tienen porque no han asumido tecnologías que contribuyan a un correcto desarrollo de sus pequeñas o medianas empresas. (Ministerio del Ambiente, 2012).

### **2.4.1 Normativas de Calidad del Aire.**

Un número considerable de países alrededor del mundo han establecido políticas que velan por el recurso aire, logrando que este elemento presente niveles bajos de contaminación. Ecuador ha dado pasos importantes en este aspecto, contando con un marco referencial de la calidad del aire de las ciudades con mayor población y crecimiento económico, incluyendo una normativa que establece el porcentaje de contaminantes y gases tóxicos permisibles en el ambiente. Sin embargo, no existen organismos que velen porque dicha normativa se cumpla, lo que ocasiona que, en muchas de estas grandes ciudades, los niveles de contaminación sean muy elevados. A esto se añade el hecho de que no se cuenta con estudios serios que indiquen efectos de la salud ocasionados por la contaminación del aire (Ministerio del Ambiente, 2012).

Es por esto que se hace necesario regirnos a organizaciones internacionales, las cuales llevan muchos años aportando en lo concerniente a la calidad del aire que respiran las personas, delimitando los efectos que los contaminantes presentes en el ambiente pudieran generar en la salud y, generando a la vez, propuestas para mejorar su calidad de vida.

#### **2.4.1.1 Agencia de Protección Ambiental (EPA).**

Creada en el año 1970, la Agencia de Protección Ambiental (EPA, por sus siglas en inglés) es una entidad de los Estados Unidos que busca generar proyectos, estudios y normas para la protección de la salud y del medio ambiente. Entre los aportes legislativos en los que interviene la EPA están el aire, el agua, el suelo, residuos peligrosos, especies en riesgo de extinción, entre otros. (EPA, 2017).

Aunque el estudio de la calidad del aire en ambientes interiores no es su principal objeto de estudio (se centra más en ambientes outdoor), contribuye de forma relevante a través de sus investigaciones e informes sobre los cuidados que se debe tener para mejorar la calidad del aire en lugares cerrados, así como también, información sobre los efectos negativos que tienen los contaminantes comunes del aire.

La EPA en su ley del aire limpio, establece 188 sustancias que son consideradas peligrosas para el ser humano si están presentes en el aire (EPA, 2016). Sin embargo, existen algunos contaminantes que habitualmente se encuentran presentes. Estos se los conoce como *contaminantes de criterio* y se los puede observar en la *Figura 40*:



Figura 40. Contaminantes de criterio. Fuente: CeMCAQ (<http://www.cemcaq.mx/contaminacion/contaminacion-ambiental>)

#### 2.4.1.2 Administración de Seguridad y Salud Ocupacional (OSHA).

Creada con el objetivo de velar por la seguridad y salud de los trabajadores en el ejercicio de sus labores, la Administración de Seguridad y Salud Ocupacional (OSHA, por sus siglas en inglés), es una organización que pertenece al Departamento de Trabajo de los Estados Unidos. Tiene como principales funciones el establecer y dar seguimiento a normas relacionadas con el bienestar y comodidad de los trabajadores, a más de brindar capacitaciones de diversa índole (OSHA, 2013).

Dentro de los múltiples estudios y aportes que OSHA ha realizado, están los de calidad del aire en ambientes interiores, los cuales indican que la exposición prolongada de una persona a niveles altos de contaminación en lugares cerrados puede ocasionar problemas en el organismo, a corto, mediano y largo plazo. Estos problemas surgen por múltiples factores, como pueden ser la temperatura del ambiente, la humedad, falta de ventilación, proliferación de bacterias o exposición a sustancias químicas (OSHA, 2016).

#### 2.4.1.3 Monóxido de carbono (CO).

El monóxido de carbono es un gas incoloro e inodoro muy peligroso para la salud. Su toxicidad para el ser humano se debe a que dificulta el transporte de oxígeno al cuerpo cuando se mezcla con la hemoglobina de la sangre, sustituyéndolo (OSHA, 2015).

Se origina por medio de la combustión incompleta de algunas sustancias como carbón, petróleo, gasolina, madera, entre otras. En la *Tabla 2* se puede visualizar algunas características de este compuesto químico:

Tabla 2. Características del Monóxido de Carbono.

	Características
Nombre	Monóxido de Carbono
Estado común	Gaseoso
Fórmula molecular	CO
Densidad	1,184 kg/m <sup>3</sup>
Masa molecular	28.0 g/mol
Punto de fusión	68 K (-205 °C)
Punto de ebullición	81 K (-192 °C)

Fuente: elaboración propia.

Los efectos en la salud que pudiera presentar la exposición a este gas en pequeñas concentraciones están: confusión mental, dolor de cabeza, vértigo, debilidad, náuseas y pérdida del conocimiento. En cambio, una exposición a altas concentraciones, puede generar complicaciones a nivel del sistema nervioso y en poco tiempo puede llevar a la muerte (OSHA, 2015).

OSHA establece niveles de exposición al monóxido de carbono en ambientes interiores, siendo el límite permitido el de 50 partes por millón (ppm) para 8 horas de exposición. A partir de las 200 ppm es necesario que el personal que se encuentre en el lugar se retire inmediatamente y ventilar el espacio contaminado. Si el nivel de contaminación por CO se encuentra alrededor de las 1000 ppm, supone un riesgo mortal. La *Tabla 3* muestra con mayor detalle los parámetros establecidos:

Tabla 3. Niveles de exposición de CO, según OSHA

PPM	Exposición	Índice
0-50	8 horas	Moderado
51-100	8 horas	Alerta
101-200	30 minutos	Severo
700-1000	30 minutos	Peligroso

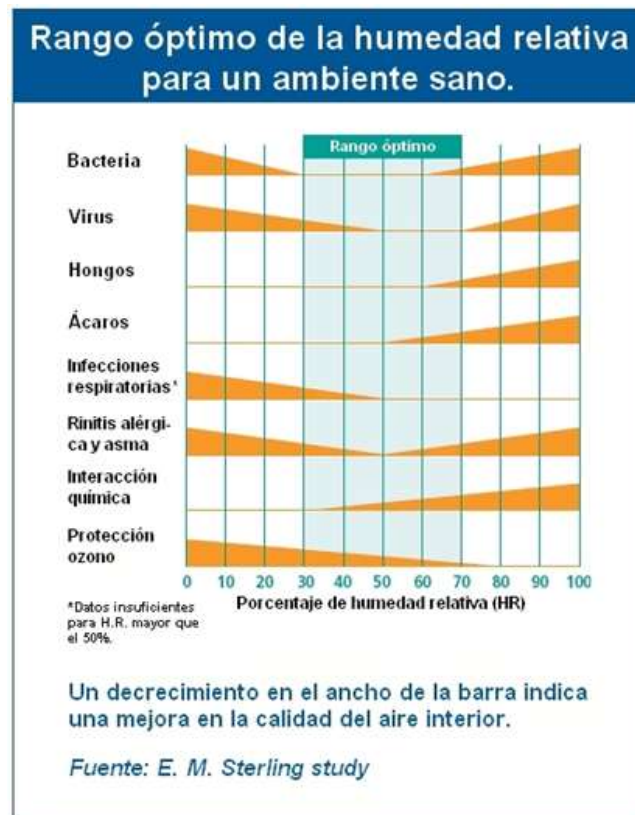
Fuente: elaboración propia.

#### 2.4.2 Calidad del Aire en Hospitales.

Si bien es cierto, un ambiente libre de contaminantes resulta indispensable cuando se habla de mejorar la calidad de vida de las personas, es doblemente indispensable cuando se trata de edificaciones que acogen a personas vulnerables, como son los hospitales. Es por esto que, en ellos, las políticas y normativas que implican su cuidado deben ser asumidas con mayor atención.

La calidad del aire en los hospitales requiere de cuidados específicos, basados en prevenir problemas ocasionados por sustancias químicas que pudieran estar presentes en el ambiente interior y que son nocivas para el ser humano, pero sobre todo en crear un ambiente hostil que detenga la proliferación de bacterias, virus y demás

microorganismos, los cuales, en la mayoría de los casos, son dañinos para el ser humano. En la *Figura 41* se puede observar cómo un ambiente controlado a nivel de humedad relativa (%HR) disminuye considerablemente el crecimiento de estos microorganismos, impidiendo incluso que se generen otros problemas.



*Figura 41.* Rango óptimo de Humedad Relativa. Fuente: Fisair (<https://fisair.com/es/control-de-humedad-en-hospitales/>).

#### 2.4.2.1 Normativa UNE.

La Asociación Española de Normalización (UNE) es un organismo de normalización en España, creado en 1986 y designado por el Ministerio de Economía, Industria y Competitividad ante la Comisión Europea (UNE, 2018). Su objetivo es el de contribuir a la sociedad en la protección a personas y empresas, favoreciendo su competitividad y seguridad.

A nivel del medio ambiente, pretende establecer normas y políticas que ayuden al mejoramiento de los recursos ambientales; es en este aspecto donde se vuelve relevante para el presente proyecto, ya que se encarga de facilitar normativas para mejorar el confort de las personas que pasan la mayor parte del día en el interior de establecimientos públicos y privados, entre ellos los del sector de la salud.

La normativa UNE 100713-2005 (Instalaciones de acondicionamiento de aire en hospitales), revisada y actualizada en 2015, es la que facilita los procedimientos que deben realizarse en centros hospitalarios, teniendo como principales factores la temperatura, presión y humedad. En la *Figura 42* se puede observar los valores que establece para cada sala de hospital en los parámetros de temperatura y humedad, variables con las que trabajaremos en este proyecto:

# Normativa: UNE 100713-2005

Área del hospital		Rango temperatura (- C)	Rango de humedad (%HR)
Quirófanos	Quirófanos tipo A y B, incluso accidentes y partos	22-26	45-55%
	Pasillos, almacén estéril, entrada y salida		
	Sala despertar		
	Otros locales		
Partos	Paritorios	24-26	
Medicina intensiva	Habitaciones con camas, incluso eventual antesala		
	Habitaciones para pacientes con riesgo de contraer infecciones		
	Para el resto de pacientes		
	Sala de urgencias		
Cuidados especiales	Habitaciones con camas		
	Sala de urgencias		
	Pasillos		
Infecciosos	Habitaciones con camas, incluso eventual antesala		
Prematuros	Habitaciones con camas		
Recién nacidos	Habitaciones con camas		
Otras	Habitaciones con camas para hospitalización		
Quemados	Unidad de quemados (Fuera UNE)	24-26	< 30%
Fisioterapia	Piscinas y baños terapéuticos	A definir	< 65%

Figura 42. Normativa UNE 100713:2005. Fuente: Fisair (<https://fisair.com/es/control-de-humedad-en-hospitales/>)

### 3. Fundamentos Metodológicos

Para dar solución al problema descrito en la sección 1.1 de este documento, se realizó la implementación de un sistema de monitoreo de calidad del aire interior para las salas seleccionadas del HLB. Se utilizó la metodología Top-Down, que es usada en múltiples áreas como son: el desarrollo de productos, software, diseño de circuitos y redes, entre otros (Restrepo Muñoz, 2010). Esta metodología basa su funcionamiento en la realización del trabajo por partes, tanto a nivel de hardware como de software (bloques de programación), subdividiéndolas cada vez en partes más pequeñas, para al final acoplarlas en un solo sistema. Esta forma de funcionamiento facilita, entre otras cosas, la labor de revisión en caso de que se genere algún error o problema de funcionamiento.

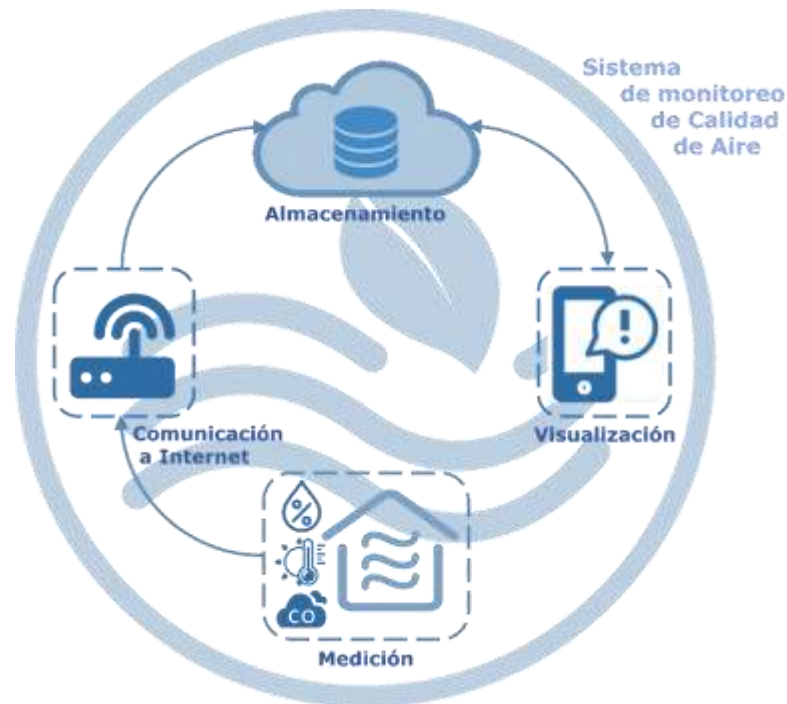
Top-Down fue seleccionada porque se adecuaba muy bien al presente proyecto en todas sus etapas y sirve como una guía estructurada sobre cómo se debe ir trabajando paso a paso. Permite, además la integración de muchos métodos de investigación como son: el análisis y síntesis, investigación exploratoria y descriptiva, investigación cuasi-experimental y longitudinal, investigación aplicada, entre otras.

Esta metodología se divide en 4 fases, la primera es la fase de *identificación de necesidades* donde se realiza el análisis de los objetivos que se quieren lograr y de las limitaciones técnicas que pudieran existir. Se realiza, así mismo, la selección de equipos y tecnologías que contribuyan a la implementación del producto. La segunda es la fase de *diseño lógico*, en la que se realiza el diseño del producto a nivel de estructura, tanto de manera general, como también de cada parte que lo conforma. La fase de *diseño físico* implica la implementación del sistema a nivel de hardware y software (configuración de equipos, acoplamiento de dispositivos, utilización de estándares y protocolos, etc.) Finalmente, en la fase de *prueba y funcionamiento* se

realizan pruebas del producto en simuladores o escenarios reales para corregir posibles errores antes de la puesta en marcha.

### 3.1 Selección de Equipos y Protocolos

El sistema de monitoreo de calidad del aire para ambientes indoor basa su funcionamiento en la toma de datos del ambiente por medio de sensores. Estos datos son enviados a la nube a través de una conexión a internet, se almacenan en una base de datos y se visualizan por medio de una aplicación web. En la *Figura 43* se puede apreciar el diagrama general de la forma como se va a implementar dicho sistema, pudiéndose identificar 4 bloques de trabajo bien definidos:



*Figura 43.* Diagrama general del sistema de monitoreo de calidad de aire. Fuente: elaboración propia.

Para llevar a cabo la implementación observada en la figura anterior, se realizó una selección de equipos y tecnologías, tanto de la parte del hardware como del software, haciendo un análisis en función de su utilización frecuente en sistemas de IoT, además de la calidad y prestigio que tienen las empresas que los fabrican. Se explicará, así mismo, los protocolos que utilizaremos en la fase de desarrollo físico.

#### 3.1.1 Elección de Hardware.

##### 3.1.1.1 Sensor de monóxido de carbono.

Para medir el nivel de contaminación del aire por monóxido de carbono, se ha seleccionado el sensor Fígaro TGS2442. Este es un sensor semiconductor elaborado de óxido de metal, con alta selectividad al CO y buena resistencia a ambientes húmedos. Entre sus ventajas está el bajo consumo de energía que genera, ya que utiliza calentamiento por pulsos (Figaro, 2007). En la *Figura 44* se aprecia el sensor TGS2442 de la marca Fígaro.



Figura 44. Sensor Figaro TGS2442. Fuente: Figaro  
<http://www.produktinfo.conrad.com/datenblaetter/175000-199999/183303-da-01-en-GASSENSOR FIGARO TGS2442 B00.pdf>

Entre sus principales aplicaciones están el de detectar niveles de CO en lugares residenciales y comerciales, controlar la calidad del aire y controlar la ventilación para estacionamientos en lugares cerrados o bajo techo (Figaro, 2007). En la *Figura 45* se muestra su estructura interna, y en la *Tabla 4* se puede apreciar sus características.

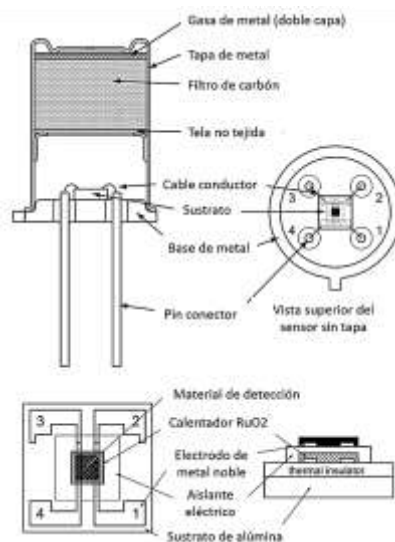


Figura 45. Estructura interna del sensor Figaro TGS2442 Fuente: Figaro  
<http://www.produktinfo.conrad.com/datenblaetter/175000-199999/183303-da-01-en-GASSENSOR FIGARO TGS2442 B00.pdf>

Tabla 4. Características de funcionamiento del sensor Figaro TGS2442.

Características	Descripción
Número de modelo	TGS 2442
Tipo de elemento sensor	M1
Paquete estándar	Lata de metal TO-5
Gases objetivo	Monóxido de carbono
Rango de detección típico	30 – 1000 ppm

Fuente: Ficha técnica Sensor Figaro.

Si bien es cierto, el rango típico de detección es de 30 a 1000 partes por millón, es posible reajustar los valores modificando los datos de resistencia en los distintos puntos de calibración para obtener un rango mucho más amplio. Esto se hace mediante



programación, la cual se detalla en la sección *anexos*. Cabe recalcar que los niveles de monóxido de carbono serán dados por el sensor en partes por millón (ppm), lo que facilita el trabajo de análisis de datos, ya que los organismos medioambientales que normalizan las emisiones de este elemento en espacios interiores, lo hacen precisamente en esta unidad de medida.

En la *Tabla 5* se observan las condiciones de operación del sensor que se deben garantizar para un funcionamiento estable del mismo.

*Tabla 5.* Condiciones de funcionamiento del sensor Fígaro TGS2442.

Condiciones	Detalle	Símbolo	Descripción
Condiciones de circuito estándar	Ciclo de voltaje del calentador	VH	VHH=4.8V±0.2V DC, 14ms VHL=0.0, 986ms
	Ciclo de voltaje del circuito	VC	VC=0V para 995ms, VC=5.0V±0.2V DC para 5ms variable (≥10kΩ)
	Resistencia de carga	RL	
Características eléctricas en condiciones de prueba estándar.	Resistencia del calentador	RH	17 ± 2.5Ω a temperatura ambiente.
	Corriente del calentador	IH	203mA (en caso de VHH) aprox.
	Consumo de energía del calentador	PH	14mW (ave.) aprox.
	Resistencia del sensor	RS	13.3kΩ ~ 133kΩ en 100 ppm de monóxido de carbono.
	Sensibilidad (relación de cambio de Rs)	B	0.13 ~ 0.31

Fuente: Ficha técnica Sensor Fígaro.

### 3.1.1.2 Sensor de temperatura y humedad.

Para medir la temperatura y humedad de los ambientes interiores del hospital se ha seleccionado el sensor digital BME280. Este es un sensor de potencia baja, precisión alta y abarca un rango bastante amplio de medición (Libelium, 2017). En la *Figura 46* se puede apreciar el sensor y su aspecto físico.



*Figura 46.* Sensor BME280. Fuente: Libelium ([http://www.libelium.com/downloads/documentation/gases\\_sensor\\_board\\_3.0.pdf](http://www.libelium.com/downloads/documentation/gases_sensor_board_3.0.pdf)).

Este sensor mide tres parámetros en el ambiente: temperatura, humedad y presión. El primero presenta una gran eficiencia, ya que tiene una resolución alta y bajo ruido. El segundo aporta significativamente en ambientes donde se necesita tiempo de respuesta rápido y es preciso en un rango considerable de temperatura. En la *Tabla 6* se puede apreciar sus características eléctricas, detalle que no se debe dejar de lado al momento de poner a funcionar el sensor.

*Tabla 6.* Características eléctricas del sensor BME280.

Características	S. Temperatura	S. Humedad
Tensión de alimentación	3,3 V	3,3 V
Corriente de sueño típica	0.1 $\mu$ A	0.1 $\mu$ A
Máxima corriente de reposo	0.3 $\mu$ A	0.3 $\mu$ A
Precisión	$\pm 1$ °C	$\pm 3\%$ HR
Consumo típico	1 $\mu$ A	1.8 $\mu$ A
Tiempo de respuesta	1.65 segundos	1 segundo

Fuente: Guía de gases de Wasmote.

### 3.1.1.3 Placa de sensor de gases.

Para recolectar los datos medidos por los sensores se ha seleccionado la placa de sensor de gases Wasmote, creada por la empresa Libelium. Esta placa ha sido diseñada para monitorear variables en el ambiente, tales como humedad, temperatura, presión atmosférica, entre otros diferentes tipos de gases. Por medio de esta tarjeta es posible medir la calidad del aire al mismo tiempo con 7 sensores de gases, los mismos que pueden ser configurados mediante programación, al conectar la placa con la tarjeta de Wasmote (Libelium, 2017). En la *Figura 47* se puede apreciar la placa de sensores de gases que utilizaremos en el proyecto, y en la *Tabla 7* se puede encontrar el listado de sensores soportados por placa.



*Figura 47.* Placa de sensores de gases. Fuente: elaboración propia.

Tabla 7. Sensores soportados por la placa de sensores de gases.

Sensor	Abreviatura
Monóxido de carbono	CO
Dióxido de carbono	CO2
Oxígeno molecular	O2
Ozono	O3
Óxido nítrico	NO
Dióxido nítrico	NO2
Dióxido de azufre	SO2
Amoníaco	NH3
Metano	CH4
Sulfuro de hidrógeno	H2S
Material particulado – polvo	(PM1 / PM2.5 / PM10)
Temperatura	C
Humedad	%
Presión	Bar

Fuente: Guía de gases de Waspnote.

La placa de sensores de gases cuenta con entradas o sockets en donde van colocados los sensores. Cada socket está configurado de fábrica para recibir información de un grupo de sensores específicos, tal como se muestra en la *Figura 48*.

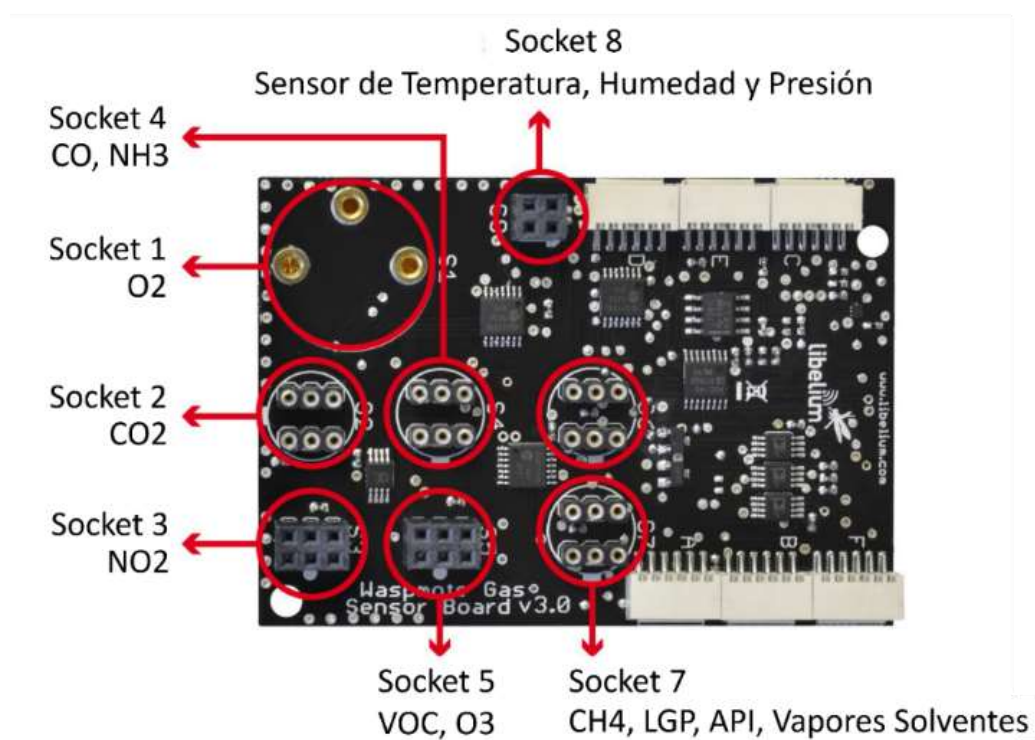


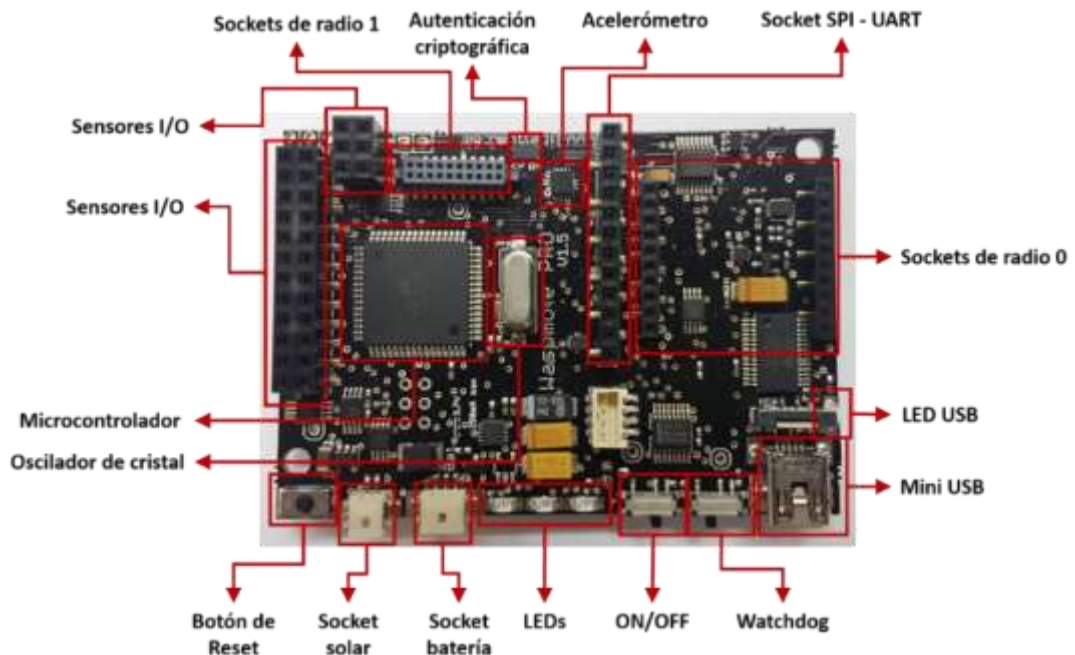
Figura 48. Ubicación de sockets y sensores permitidos, Fuente: Libelium ([http://www.libelium.com/downloads/documentation/gases\\_sensor\\_board\\_3.0.pdf](http://www.libelium.com/downloads/documentation/gases_sensor_board_3.0.pdf)).

### 3.1.1.4 Tarjeta Waspote PRO v1.5.

Para configurar los sensores y posteriormente enviarlos a internet se ha seleccionado la tarjeta Waspote. Esta tarjeta ha sido creada con un diseño modular, lo que permite integrar los módulos o accesorios necesarios para cada proyecto implementado (Libelium, 2019). Entre los módulos que se pueden integrar con la tarjeta Waspote y que se encuentran en el manual de usuario del fabricante están:

- Módulos ZigBee / 802.15.4 XBee (2.4 GHz, 868 MHz, 900 MHz)
- Módulo LoRaWAN (433/868/900 MHz)
- Módulo LoRa (868/900 MHz)
- Módulo Sigfox (868/900 MHz)
- Módulo GPRS (Cuatribanda: 850/900/1800/1900 MHz)
- Módulo 3G (WCDMA / UMTS de doble banda 900/2100 MHz)
- Módulo 3G (GSM / GPRS / EDGE de tres bandas 850/900/1800 MHz)
- Módulo 4G (versiones para Europa / Brasil, América y Australia)
- Módulo WiFi
- Módulos Bluetooth: Bluetooth Low Energy y Bluetooth Pro
- Módulo NFC / RFID
- Módulo GPS
- Módulos de sensores (el que usaremos en el proyecto)
- Módulo de almacenamiento: tarjeta de memoria SD

En la *Figura 49* y *Figura 50* se puede apreciar los componentes de la tarjeta Waspote y en la *Tabla 8* se observan las especificaciones y valores operacionales, los cuales sirven a la hora de poner a funcionar el proyecto en las condiciones de operación adecuadas.



*Figura 49.* Componentes tarjeta Waspote (parte superior). Fuente: Libelium ([http://www.libelium.com/downloads/documentation/waspote\\_technical\\_guide.pdf](http://www.libelium.com/downloads/documentation/waspote_technical_guide.pdf)).

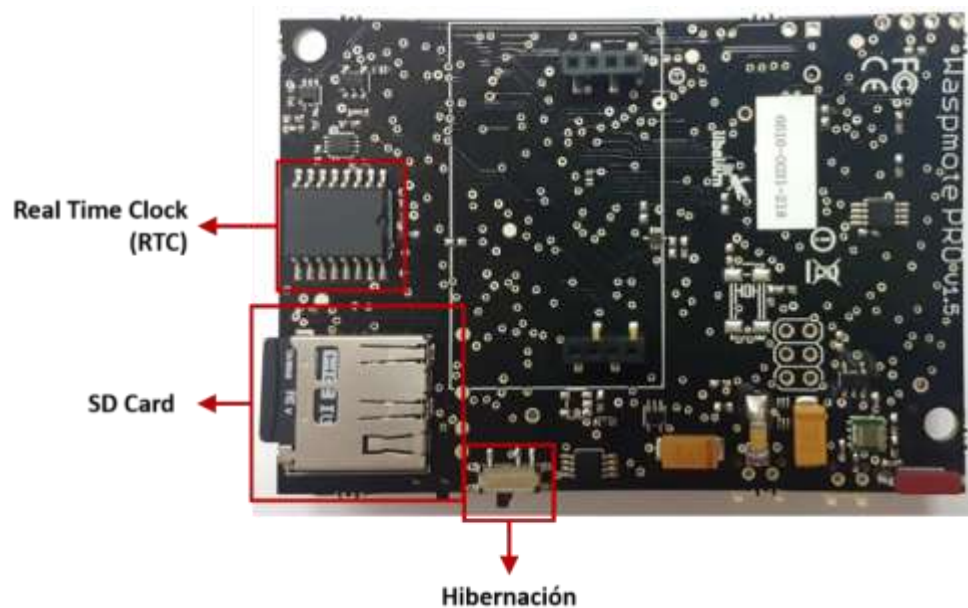


Figura 50. Componentes tarjeta Wasp mote (parte posterior). Fuente: Libelium (<http://www.libelium.com/downloads/documentation/wasp mote technical guide.pdf>).

Tabla 8. Especificaciones y valores operaciones de la tarjeta Wasp mote.

Especificaciones		Valores operacionales	
Microcontrolador	ATmega1281	Voltaje operativo mínimo de batería	3.3 V
Frecuencia	14.7456 MHz	Voltaje operativo máxima de batería	4.2 V
SRAM	8 kB	Voltaje de carga USB	5 V
EEPROM	4 kB	Voltaje de carga del panel solar	6-12 V
FLASH	128 kB	Corriente de carga de batería	100 mA
Tarjeta SD	16 GB	Corriente de carga de panel solar	300 mA
Peso	20 g		
Dimensiones	73.5 x 51 x 13 mm		
R. de temp.	-30 °C, +70 °C		

Fuente: Guía técnica de Wasp mote.

### 3.1.1.5 Raspberry Pi3.

Para poder enviar a internet la información obtenida por los sensores, se necesita un dispositivo capaz de conectarse a internet a manera de gateway, siendo seleccionada para cumplir esta tarea la tarjeta Raspberry Pi3 (Figura 51). Esta tarjeta es el primer modelo de la tercera generación de minicomputadoras de la empresa Raspberry, reemplazando a modelos de versiones anteriores (Pi2 modelo B, Pi1 modelo B+, entre otros) (RaspberryPi, 2016). Actualmente no es la última versión que ha salido al mercado, pero cuenta con excelentes prestaciones, las cuales se muestran a continuación:

- Un procesador (CPU) QuadCore de 1.2Ghz Broadcom BCM2837 de 64 bits
- Una memoria RAM de 1Gb de capacidad.
- Una tarjeta Wireless LAN y Bluetooth Low Energy (BLE) BCM43438

- Dispone de 40 pines de uso general GPIO
- 4 puertos USB
- Salida HDMI.
- Como unidad de almacenamiento una ranura MicroSD
- Puerto de alimentación micro USB con soporte de 5 voltios hasta 2.5 amperios.



Figura 51. Raspberry Pi 3. Fuente: RaspberryPi (<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>)

El cambio más notorio de esta versión en comparación con las anteriores y que para el proyecto presenta gran relevancia es su eficiente conectividad a internet, ya que cuenta con la tecnología Wi-Fi BCM43143. En la *Figura 52* se puede apreciar los componentes de esta tarjeta y sus ubicaciones dentro de la misma.

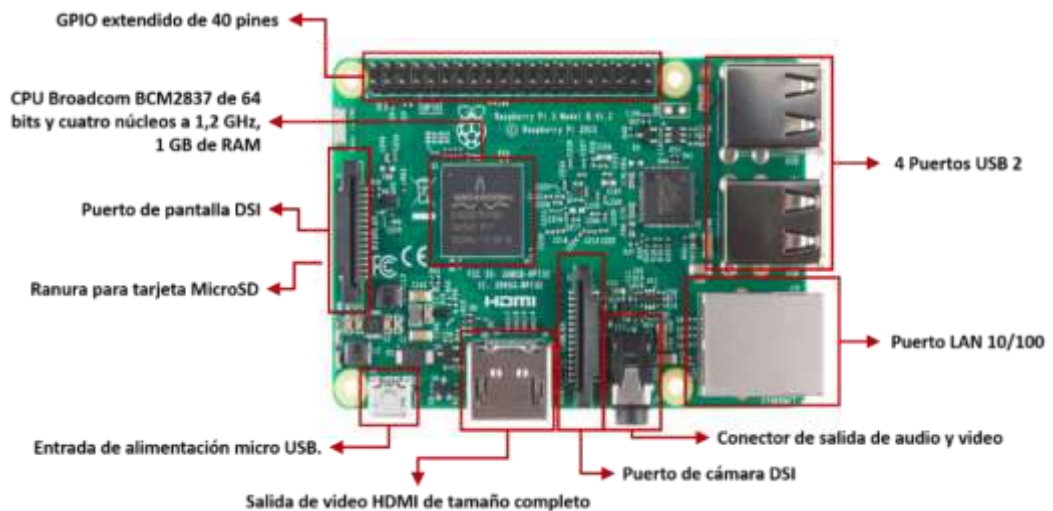


Figura 52. Componentes de la tarjeta Raspberry Pi3. Fuente: elaboración propia.

### 3.1.1.6 Fuentes de energía.

Para que el equipo de medición sea modular, es decir, que pueda trasladarse de un lugar a otro mientras realiza las mediciones, es necesario contar con fuentes de energía portátiles. Para este proyecto se ha seleccionado una batería de iones de litio recargable de 3.7 voltios, para alimentar a la tarjeta Waspote (*Figura 53*), y una batería portátil Klip Xtreme KBH-155GD 5000mAh Enox5000, para alimentar a la tarjeta Raspberry Pi3 (*Figura 54*)



Figura 53. Batería de iones de litio de 3.7v recargable. Fuente: elaboración propia.



Figura 54. Batería portátil Klip Xtreme KBH-155GD. Fuente: AibiTech (<https://www.aibitech.com/telefonos-celulares-desbloqueados/bateria-externa/cargador-bateria-portatil-klip-xtreme-kbh-155gd-5000mah-enox5000-oro-klip-xtreme-17569.html>).

Estas baterías son de excelente calidad y almacenan un valor considerable de corriente, lo cual permite un desenvolvimiento totalmente autónomo del proyecto, llegando a funcionar ininterrumpidamente hasta por 12 horas.

### 3.1.2 Elección de Software.

#### 3.1.2.1 IDE Wasmote.

El entorno de desarrollo integrado (IDE) es el software que se necesita para configurar los programas y acciones que deben ejecutarse en el proyecto en su fase de inicio, es decir, recibir la información que proporcionan los sensores y posteriormente enviarla al dispositivo con conexión a internet (Libelium, 2017). En la Figura 55 se puede observar el entorno de desarrollo y sus partes.



Figura 55. IDE Wasmote y sus partes. Fuente: Libelium ([http://www.libelium.com/downloads/documentation/wasmote\\_technical\\_guide.pdf](http://www.libelium.com/downloads/documentation/wasmote_technical_guide.pdf)).

### 3.1.2.2 Firebase.

Los datos que se leen de los sensores deben enviarse a la nube para ser almacenados, por lo que se necesita contar con una base de datos en internet. Para esta tarea se ha seleccionado una plataforma gratuita de Google llamada Firebase. Esta plataforma está formada por diferentes servicios especialmente diseñados para el IoT como, por ejemplo:

- Base de datos en tiempo real
- Monitoreo de aplicaciones
- Analítica de uso de las aplicaciones
- Almacenamiento en la nube

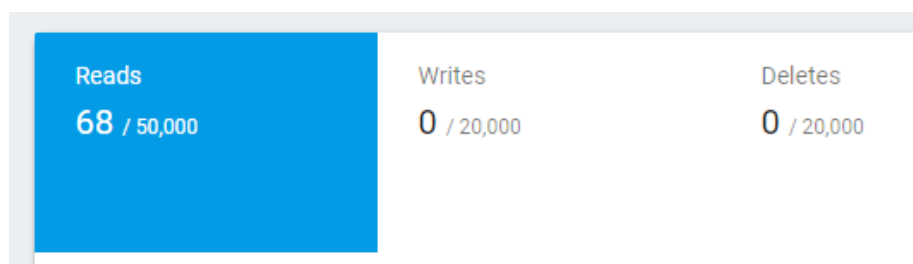
En la *Figura 56* se puede apreciar los principales servicios de Firebase de Google.



*Figura 56.* Servicios ofrecidos por Firebase. Fuente: Hackernoon. (<https://hackernoon.com/introduction-to-firebase-218a23186cd7>)

Para el almacenamiento de los datos de las mediciones vamos a usar el servicio de bases de datos en tiempo real llamado Firestore, este servicio es una base de datos NoSQL, la que mediante sockets mantiene comunicación en tiempo real con todos los dispositivos conectados.

Firestore ofrece de forma gratuita 50 mil lecturas, 20 mil escrituras y 20 eliminaciones de datos, así como 10Gb de almacenamiento como se puede apreciar en la *Figura 57*. Si fuese necesario consumir más de eso, se debe contratar un plan de pago, sin embargo, para el proyecto, las cantidades que ofrece el servicio gratuito son más que suficientes.



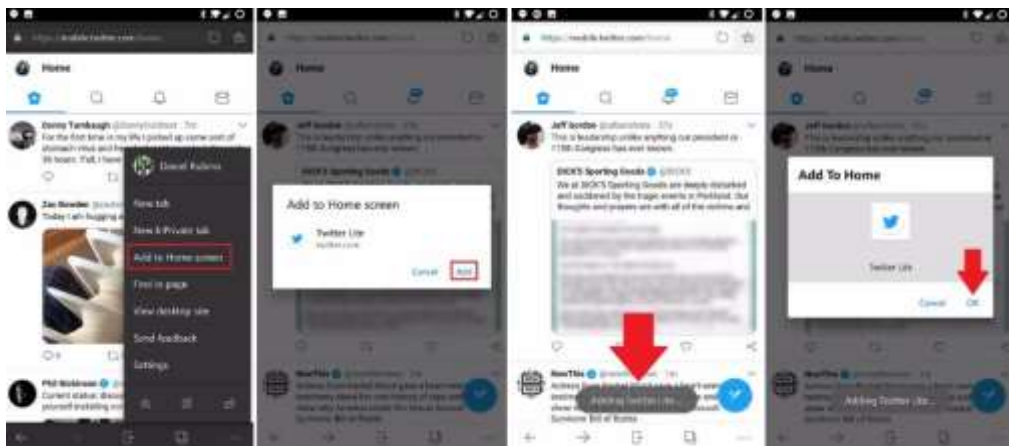
*Figura 57.* Consumo de Firestore. Fuente: elaboración propia.



### 3.1.2.3 PWA.

Para la visualización de los datos almacenados en la base de datos se creó una aplicación web progresiva (PWA), la cual tiene las ventajas de funcionar como una aplicación nativa en los teléfonos móviles y como una aplicación web tradicional en los navegadores. En otras palabras, permite hacer un solo programa que funciona en múltiples plataformas. Como las PWAs se ejecutan sobre un navegador web, cualquier dispositivo con un navegador web será capaz de ejecutar la PWA.

Otra de las ventajas que tiene la PWA, sobre todo para el presente proyecto, es que al abrirla en un teléfono móvil, el navegador ofrece la opción de agregar esta página al menú de inicio, tal como se puede apreciar en la *Figura 58*:



*Figura 58.* Opción de agregar PWA en la pantalla de inicio de Android. Fuente: Windows Central. (<https://www.windowscentral.com/how-to-pwa-edge-android>)

#### 3.1.2.3.1 Versiones de Javascript.

Las PWA son desarrolladas con tecnologías web como lo son HTML, CSS y Javascript, y aunque HTML y CSS son tecnologías que no liberan nuevas versiones con mucha frecuencia, javascript si está en constante mejora.

Como se puede apreciar en la *Figura 59* existen diferentes versiones de javascript y cada navegador web es libre de implementar la versión que ellos deseen, es decir, no están obligados a trabajar con la última versión. Esto ha ocasionado que el desarrollo de aplicaciones web sea muy complejo y que no se puedan disfrutar de las últimas novedades que ofrece javascript a los desarrolladores.



*Figura 59.* Versiones de Javascript. Fuente: Adrián Mejía (<https://adrianmejia.com/overview-of-javascript-es6-features-a-k-a-ecmascript-6-and-es2015/>).

Actualmente en la documentación de Javascript existe una sección en cada función, la cual se detalla a partir de la versión soportada por el navegador, tal como se puede apreciar en la *Figura 60*.

**Browser Support**

The numbers in the table specify the first browser version that fully supports the method.

Method	Chrome	Internet Explorer	Firefox	Safari	Opera
trim()	10.0	9.0	3.5	5.0	10.5

**Syntax**

```
string.trim()
```

*Figura 60.* Tabla de soporte a la función trim. Fuente: Benchresources. (<http://www.benchresources.net/remove-leading-and-trailing-whitespace-from-javascript-string/>)

Como se dijo anteriormente, esto hace que programar aplicaciones web sean muy complejo. Sin embargo, existe una versión que en la actualidad cualquier navegador implementa y es ECMAScript 5. Esta versión fue lanzada hace varios años, específicamente en 2009, lo que da a entender que es una versión relativamente antigua.

En la Tabla 9 se puede observar a partir de que versión del navegador está disponible este estándar, siendo Internet Explorer el primero en adoptarlo y Ópera el último en hacerlo.

*Tabla 9.* Implementación de ECMAScript 5 en los principales navegadores.

<b>Navegador</b>	<b>Versión</b>	<b>Fecha</b>
Chrome	23	Sep-12
Firefox	21	Apr-13
IE	9	Mar-11
IE / Edge	10	Sep-12
Safari	6	Jul-12
Opera	15	Jul-13

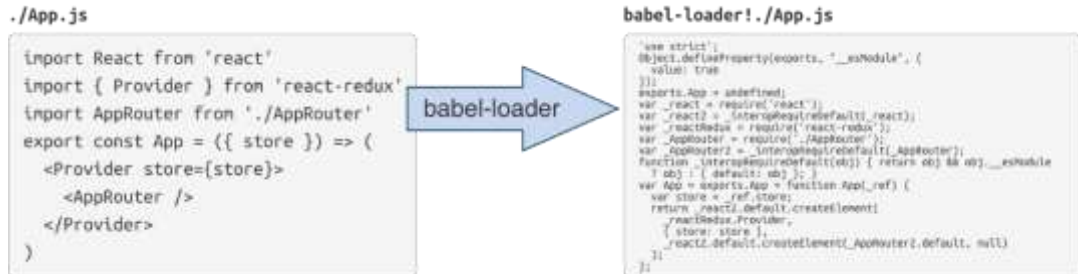
Fuente: Elaboración propia.

Todo esto ayuda a comprender que para hacer una aplicación compatible con todos los navegadores web se debe usar ECMAScript 5. El único inconveniente es que este estándar fue lanzado en el año 2009, es decir, existen 10 años de mejoras y nuevas funcionalidades que no podríamos usar.

### 3.1.2.3.2 *Transpilación de código con Babel.*

Ante la problemática anteriormente descrita surgieron varias soluciones, siendo la principal la capacidad de escribir código en diferentes versiones de ECMAScript y que estas se conviertan o “Transpilen” a otra versión de ECMAScript, con la finalidad de poder usar las últimas novedades del lenguaje.

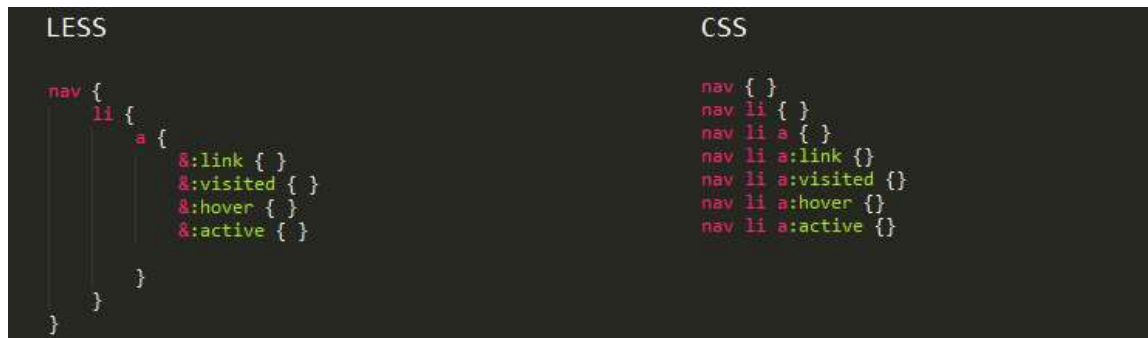
En sus inicios existieron muchos transpiladores de código, pero el más usado en la actualidad es Babel. En la *Figura 61* se ver cómo se escribe líneas de código más sencillas y compactas y estas son transpiladas a ECMAScript 5. Esta transpilación es realizada de forma automática por babel y permite usar todas las nuevas funcionalidades de javascript durante el desarrollo.



*Figura 61.* Ejemplo de una transpilación de código usando Babel. Fuente: Github (<https://github.com/webpack/docs/wiki/usage>).

### 3.1.2.3.3 Transpilación de CSS.

La aparición de Babel abrió un gran abanico de posibilidades gracias a su modularidad. Babel no se creó simplemente para convertir diferentes versiones de ECMAScript, sino que mediante su sistema de plugins es capaz de convertir cualquier entrada en cualquier salida. Esto significa que no es necesario escribir código CSS, sino que se puede definir el CSS desde javascript como lo hace el preprocesador de CSS llamado LESS. En la *Figura 62* se puede apreciar como LESS permite anidar clases CSS como si fuesen objetos de javascript para que posteriormente babel se encargue de convertirlo a CSS mediante un plugin de LESS.



*Figura 62.* LESS vs CSS. Fuente: elaboración propia.

Esta metodología de transpilación de código se volvió el estándar del desarrollo web moderno ante la poca importancia que le daban los navegadores web a la implementación de los últimos estándares de javascript.

### 3.1.2.3.4 Transpilación de HTML.

De la misma forma que con CSS y JavaScript, escribir código HTML se torna largo y tedioso debido a las propias limitantes del lenguaje. Esto no permite declarar estructuras condicionales y repetitivas, ni mucho menos facilita la reutilización de componentes entre diferentes páginas web.

Así como apareció LESS, aparecieron otros preprocesadores, pero para HTML en este caso usaremos de ejemplo PUG, el cual es un preprocesador que cambia la clásica sintaxis de HTML (similar al XML), por un estilo más similar a Python como se puede apreciar en la *Figura 63*.

```
html(lang="en")
  head
    meta(charset="UTF-8")
    meta(name="viewport", content="width=device-width, initial-scale=1.0")
    meta(http-equiv="X-UA-Compatible", content="ie=edge")
    link(rel="stylesheet", href="main.scss")
    title Testing!!!
  body
    h1 Testing out Pug with Parcel
    script(src="./index.js")
```

*Figura 63.* Ejemplo de código PUG. Fuente: elaboración propia.

La principal ventaja de PUG es que permite escribir menos código gracias a su sintaxis más limpia, además de la facilidad para reutilizar código previamente escrito como si se tratase de programación funcional. En la *Figura 64* se puede apreciar la diferencia entre escribir código PUG y escribir las mismas funcionalidades en HTML.



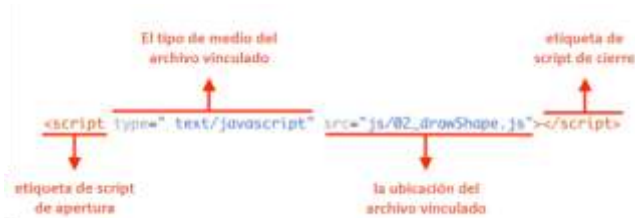
*Figura 64.* Comparación entre Pug y HTML. Fuente: elaboración propia.

### 3.1.2.3.5 Webpack.

En el momento en que apareció Babel, surgieron múltiples tecnologías web modernas que transpilan a los clásicos HTML, CSS y Javascript. Estas tecnologías destacan por su modularidad, es decir permite crear componentes individuales fáciles de reutilizar.

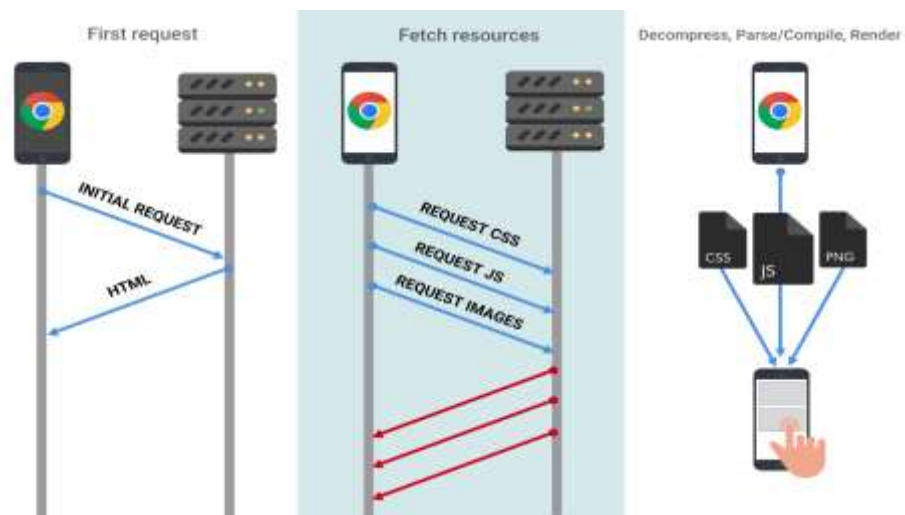
Esta modularidad facilita enormemente el trabajo a los desarrolladores, pero para el navegador web que debe descargar uno a uno cada uno de estos pequeños componentes es un trabajo extra muy pesado, ya que un aplicativo web se puede componer fácilmente de cientos de librerías y módulos reutilizables.

El navegador web es el responsable de descargar primero el HTML principal que forma el sitio web. Dentro de ese HTML existen directivas LINK y SCRIPT las cuales indican al navegador que debe descargar archivos Javascript y CSS externos. En la *Figura 65* se puede apreciar un tag SCRIPT, que permite incluir un archivo javascript en el sitio web.



*Figura 65.* Ejemplo de inserción de código Javascript mediante un tag HTML.  
Fuente: Protohelp (<https://protohelp.zendesk.com/hc/en-us/articles/208154766-Link-JS-and-CSS-files-in-HTML-file-In-DevSpace>).

El navegador descarga inicialmente el HTML del sitio web y luego uno por uno cada LINK y SCRIPT que se encuentre en el HTML, procurando hacerlo en el orden exacto en que aparecen, tal como se puede apreciar en la *Figura 66*.



*Figura 66.* Descarga de una página web desde el navegador. Fuente: Google Developers.  
(<https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/javascript-startup-optimization/>)

### 3.1.3 Protocolos de comunicación utilizados

#### 3.1.3.1 Comunicación serial

La comunicación serie es un medio de comunicación entre dispositivos electrónicos para transmitir o recibir datos. Consiste en el envío de dichos datos bit por bit a través de un único pin para la transmisión y un único pin para la recepción, permitiendo de esta forma una comunicación bidireccional. Para el presente proyecto contaremos con la comunicación por medio del protocolo USB, ya que este cuenta con algunas características que permite transmitir datos entre los equipos del sistema. Estas características son:

- Mayor velocidad de transmisión y recepción.
- Múltiple conexión de dispositivos (hasta 127).
- Facilidad de manejo.
- Simplicidad en la conexión.

### **3.1.3.2 Comunicación inalámbrica**

La comunicación inalámbrica hace referencia al envío y recepción de información por un medio distinto del físico, es decir, la información viaja en la forma de ondas de radio a través del espacio vacío. Ejemplos de este tipo de comunicación son el GSM, la conexión bluetooth y WiFi, siendo esta última la que utilizaremos en el sistema de monitoreo para enviar los datos a la nube. El protocolo utilizado será 802.11n, creado por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, por sus siglas en inglés), ya que es uno de los más conocidos y presenta algunas ventajas en comparación con sus antecesores, las cuales mencionamos a continuación:

- Mayor velocidad de transmisión y recepción de paquetes de datos.
- Uso de dos bandas de frecuencia: 2.4 GHz y 5 GHz
- Mayor cantidad de canales de comunicación.
- Mejor recepción de la señal.

### **3.1.3.3 Comunicación a la web**

La comunicación a la web hace referencia a la transmisión de paquetes de datos a internet, utilizando protocolos de direccionamiento, seguridad, entre otros. Uno de los que usaremos en el sistema de monitoreo es el Protocolo de Transferencia de Hipertexto (HTTP, por sus siglas en inglés), ya que este tiene como principal función definir la forma como el cliente se comunica con el servidor, es decir, solicita la ubicación de una página o base de datos de internet y el servidor responde mostrando la información solicitada.

## **3.2 Diseño Lógico**

Como se puede observar en la *Figura 43*, el sistema de monitoreo de calidad del aire está conformado por 4 etapas bien marcadas: medición, comunicación a internet, almacenamiento y visualización. Cada una de estas etapas o procesos cuenta con sus respectivos componentes a nivel de hardware, software y protocolos de comunicación.

La *etapa de medición* inicia a partir de los sensores de CO, temperatura y humedad que toman los datos del entorno y los envían a la placa Waspote por medio de la tarjeta lectora de gases. La placa, mediante programación en el IDE, envía los datos medidos por medio del puerto serial (USB). La batería recargable proporciona la energía necesaria para el correcto funcionamiento de la tarjeta Waspote. Este proceso se lo puede observar en la *Figura 67*:

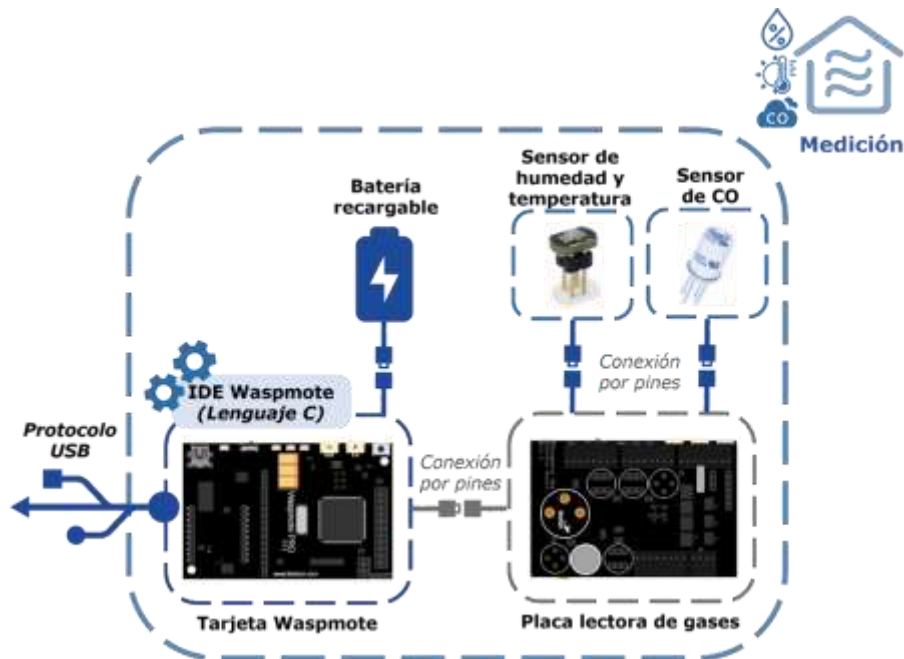


Figura 67. Esquema general de la etapa de medición. Fuente: elaboración propia.

En la Figura 68 se observa la *etapa de comunicación a internet*, donde la tarjeta Raspberry Pi3 recibe los datos enviados desde la etapa de medición y los envía a la base de datos en la nube utilizando el protocolo HTTP, dentro de la programación realizada en Typescript. La tarjeta Raspberry es alimentada por medio de una batería recargable para que pueda funcionar.

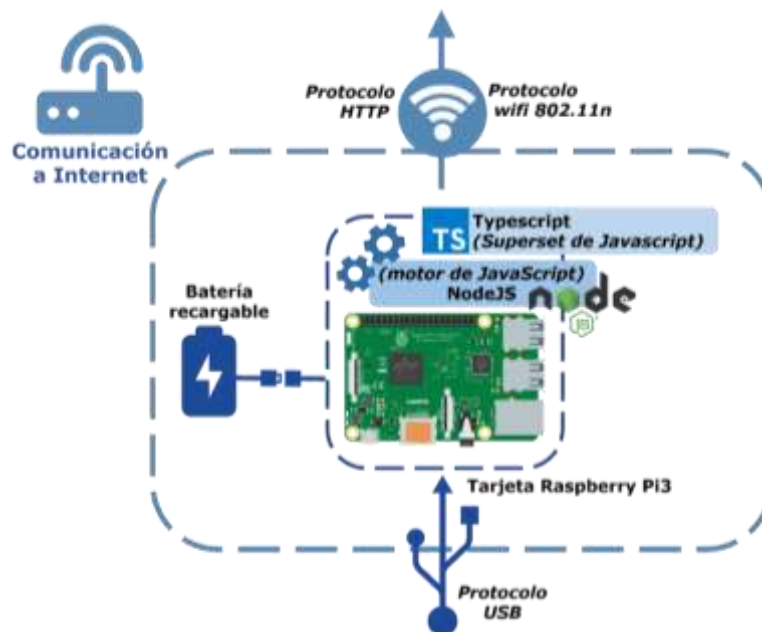


Figura 68. Esquema general de la etapa de comunicación a internet. Fuente: elaboración propia.

La Figura 69 muestra la *etapa de almacenamiento*, en donde los datos que han sido enviados en la etapa anterior se alojan en la plataforma de base de datos Firestore, que pertenece a Firebase de Google. Se muestra también el servidor estático Apache, ya que este será el que guarde las configuraciones de la base de datos de internet.

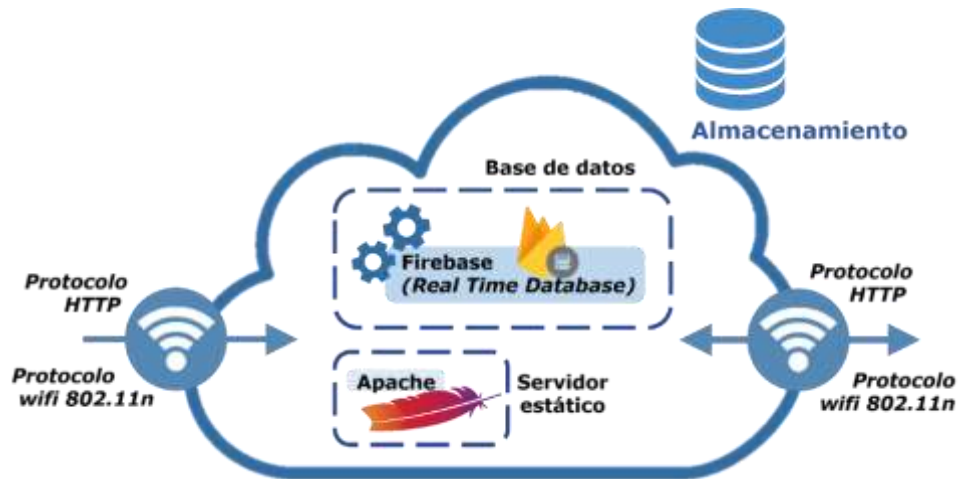


Figura 69. Esquema general de la etapa de almacenamiento. Fuente: elaboración propia.

La última etapa es la de *visualización* (Figura 70) en la que los datos almacenados en la base de datos de Firestore son solicitados por la PWA, para mostrarlos al usuario final en un dispositivo. Esta PWA es creada a partir de VueJs, contando además con Babel para la transpilación y Webpack para el envío en un solo paquete de los datos al navegador web.

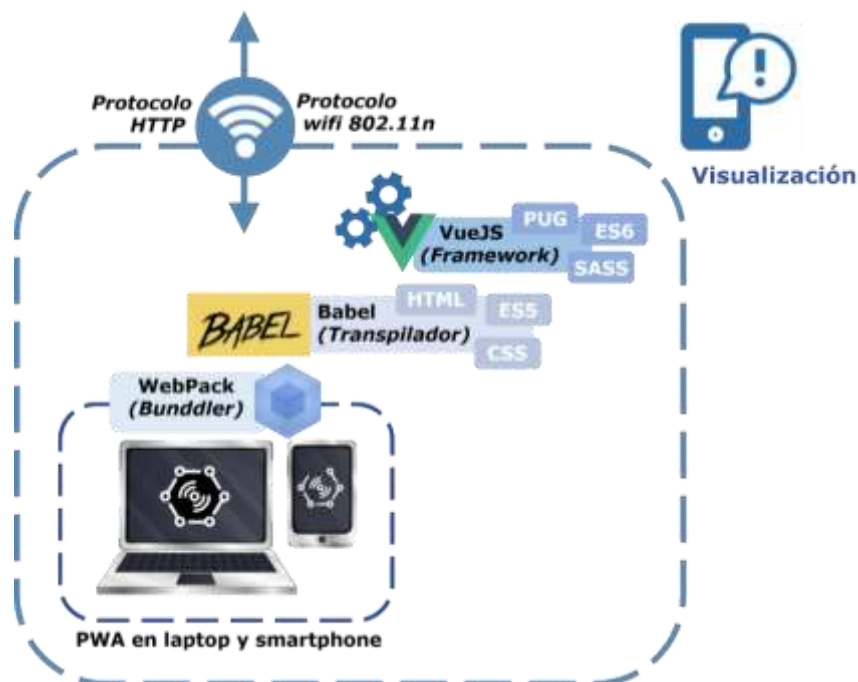


Figura 70. Esquema general de la etapa de visualización. Fuente: elaboración propia.

### 3.3 Diseño Físico

#### 3.3.1 Etapa de medición.

El sistema de monitoreo debe medir la calidad del aire para ambientes interiores del HLB en 3 parámetros: nivel de contaminación por *monóxido de carbono* (CO) en partes por millón (ppm), *humedad* en porcentaje de humedad relativa (% RH) y *temperatura* en grados centígrados (°C).



Para que la placa de sensores de gases pueda leer los datos, se deben encender los módulos de lectura de datos y comunicación necesarios, como se puede apreciar en la *Figura 71*.

```
void setup()
{
  USB.ON();
  RTC.ON();
  COSensor.setCalibrationPoints(resValues, concentrations, numPoints);
  Gases.ON();
  COSensor.ON();
  bme.ON();
  SD.ON();
}
```

*Figura 71.* Configuración de la placa para la lectura de datos. Fuente: elaboración propia.

La siguiente acción a realizar en el IDE es la lectura de datos de los sensores, tal como se puede apreciar en la *Figura 72*. El IDE de Wasmote, al ser basado en Arduino, tiene una función repetitiva, es decir, que lo que se configura y programa, termina y vuelve a iniciar de forma indefinida.

```
float COVol = COSensor.readVoltage();
float CORes = COSensor.readResistance();
float COPPM = COSensor.readConcentration();
float temperature = bme.getTemperature();
float humidity = bme.getHumidity();
float batttery = PWR.getBatteryVolts();
char* rtcTime = RTC.getTime();
```

*Figura 72.* Código para la lectura de datos de los sensores. Fuente: elaboración propia.

El mecanismo de comunicación entre la tarjeta Wasmote y la Raspberry Pi3 es el protocolo serie, ya que viene implementado de fábrica en ambas placas, lo que facilita su utilización. En la *Figura 73* se puede apreciar la comunicación entre ambos equipos por medio del cable USB.



*Figura 73.* Conexión entre Wasmote y Raspberry Pi. Fuente: elaboración propia.

El envío de información por comunicación serial entre estos equipos se puede lograr por medio de la función “*print*” del módulo USB del Wasmote como se puede apreciar en la *Figura 74*.

```

USB.print("{ \"temperature\": \"\"");
USB.print(temperature);
USB.print("\", \"covoltage\": \"\"");
USB.print(COVol);
USB.print("\", \"coresistance\": \"\"");
USB.print(CORes);
USB.print("\", \"humidity\": \"\"");
USB.print(humidity);
USB.print("\", \"coppm\": \"\"");
USB.print(COPPM);
USB.print("\", \"power\": \"\"");
USB.print(battery);
USB.print("\", \"day\": \"\"");
USB.print(day);
USB.print("\", \"date\": \"\"");
USB.print(date + 1);
USB.print("\", \"time\": \"\"");
USB.print(time + 1);
USB.println("\}");

```

Figura 74. Envío de datos a través del puerto serial. Fuente: elaboración propia.

Cada medición realizada por el Wasmote está digitalizada dentro de la placa. En otras palabras, está almacenada en formatos de unos y ceros, los cuales son enviados a través del puerto serial, uno a continuación del otro, desde el transmisor hasta el receptor por protocolo serie. A continuación, el receptor recoge los datos y confirma que han llegado; si algún dato se pierde, el protocolo serie automáticamente los vuelve a solicitar, de esta manera siempre estamos seguros de que la información enviada llega en su totalidad. En la Figura 75 se puede observar los datos enviados por protocolo serie, visualizados desde una herramienta con la que cuenta el IDE, llamada *Monitor Serie*:

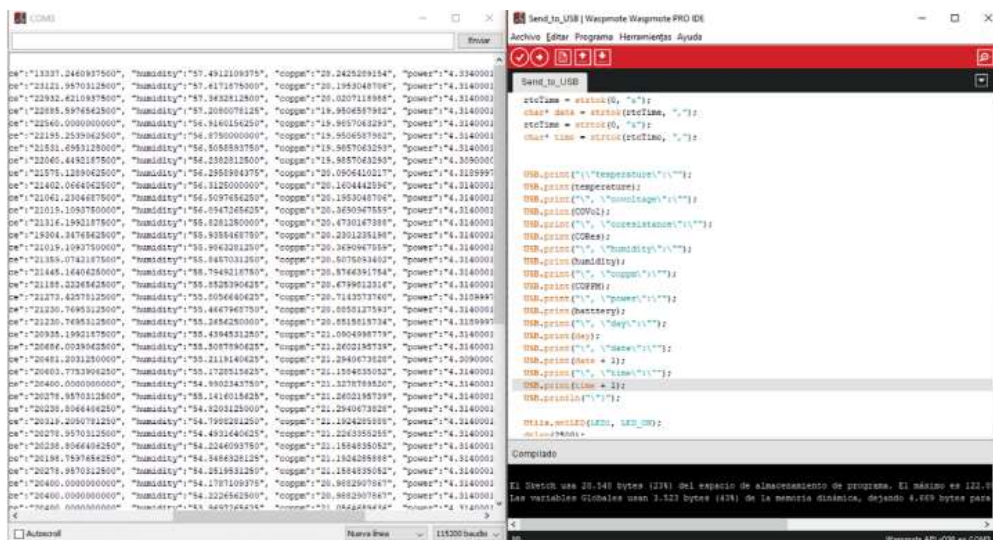


Figura 75. Envío de datos medidos a través del puerto serie. Fuente: elaboración propia.

### 3.3.2 Etapa de comunicación a internet.

Los datos de los sensores son leídos por la tarjeta Wasmote, sin embargo, al carecer esta de conexión a internet, no se los puede subir a la nube. Es por esta razón que se utiliza la tarjeta Raspberry Pi3 para que funcione como gateway.

La Raspberry Pi, al ser una mini computadora, dispone de acceso a internet. Esta conexión la puede lograr de forma inalámbrica mediante wifi o a través de un cable por medio de su puerto de red (ethernet). Sin importar el mecanismo usado para dotar de internet a la Raspberry Pi, el programa está diseñado para enviar los datos a través de cualquier conexión de red existente.

Para la lectura de datos desde el puerto serie, la Raspberry Pi3 tiene un script en NodeJS (*Figura 76*). Este script es responsable de la lectura de los datos del puerto serial. El proceso de lectura de datos consiste en encontrar los puertos seriales activos, es decir, todos los dispositivos conectados para luego establecer comunicación al primer puerto serial activo. Como la tarjeta Waspote es el único dispositivo conectado a la Raspberry Pi, será el primero que aparezca en la lista.

```
console.log('awaiting for ports')
const ports : any[] = await Serial.findActivePorts()
if(ports.length === 0) {
  |   throw Error('No waspmote dected')
}
console.log('ports ready\n')

console.log('awaiting for waspmote')
const port = ports[0]
const serial : Serial = new Serial(port['comName'], {
  |   baudRate : 115200
})
await serial.waitForOpen()
console.log('waspote ready\n')
```

*Figura 76.* Conexión de Raspberry Pi con el puerto serie. Fuente: elaboración propia.

El siguiente paso es enviar la información recibida a la base de datos en la nube. Para esto es necesario instalar las librerías para NodeJS desarrolladas por Google e incluirlas en el proyecto. Como se puede apreciar en la *Figura 77*, estas librerías darán todas las funcionalidades necesarias para el uso de Firebase en la aplicación.

```
import Firebase from 'firebase'
import FirebaseAdmin from 'firebase-admin'
import FirebaseConfigurations from './FirebaseConfigurations'
import ServiceAccountKey from './ServiceAccountKey.json'

const App : Firebase.app.App = Firebase.initializeApp(FirebaseConfigurations)
const AdminApp : FirebaseAdmin.app.App = FirebaseAdmin.initializeApp({
  credential: FirebaseAdmin.credential.cert({
    projectId: ServiceAccountKey.project_id,
    clientEmail: ServiceAccountKey.client_email,
    privateKey: ServiceAccountKey.private_key,
  }),
  databaseURL: 'https://iot-sensors-b2da9.firebaseio.com'
})

export const Database : Firebase.database.Database = App.database()
export const Firestore : FirebaseAdmin.firestore.Firestore = AdminApp.firestore()
export default App
```

*Figura 77.* Inicialización de librerías de Firebase. Fuente: elaboración propia.

Con las librerías instaladas, se hace el envío de los datos de las mediciones a Firestore mediante las librerías de Firebase, tal como se puede apreciar en la *Figura 78*.

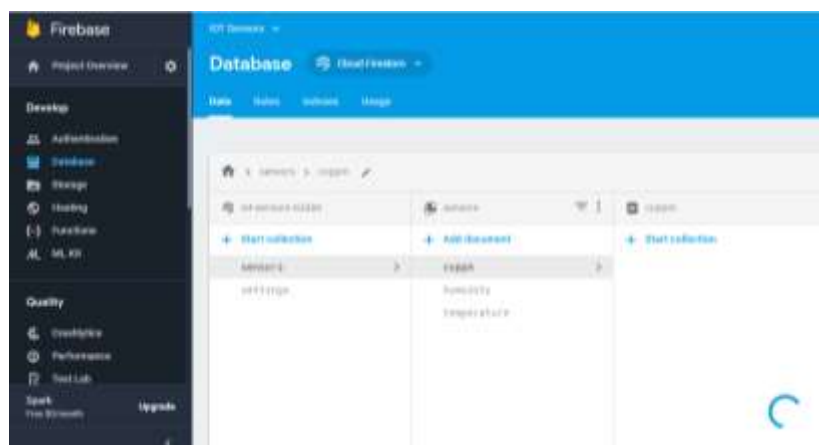
```
console.log('saving to server')
coppmSensor.update({
  measurements : firestore.FieldValue.arrayUnion({
    created_at : firestore.Timestamp.fromDate(new Date()),
    value : parseFloat(json['coppm']),
    measurementPlace : settings.data.measurementPlace
  })
})
humiditySensor.update({
  measurements : firestore.FieldValue.arrayUnion({
    created_at : firestore.Timestamp.fromDate(new Date()),
    value : parseFloat(json['humidity']),
    measurementPlace : settings.data.measurementPlace
  })
})
temperatureSensor.update({
  measurements : firestore.FieldValue.arrayUnion({
    created_at : firestore.Timestamp.fromDate(new Date()),
    value : parseFloat(json['temperature']),
    measurementPlace : settings.data.measurementPlace
  })
})
})
```

*Figura 78.* Envío de datos a Firestore desde la Raspberry Pi. Fuente: elaboración propia.

### 3.3.3 Etapa de almacenamiento.

La base de datos donde se almacena la información que envía la Raspberry es el servicio de Google llamado Firebase. Este servicio de almacenamiento de datos ofrece un SDK con todas las funcionalidades necesarias para el envío y recepción de datos a Firestore en tiempo real.

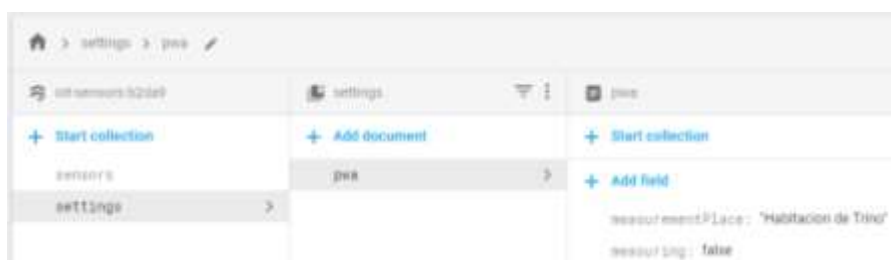
En el panel de Firestore se puede ver, editar y eliminar los datos almacenados mediante una interfaz web como se aprecia en la *Figura 79*. Para el sistema de monitoreo se usan dos colecciones (así es como se llaman las bases de datos en Firebase), una de sensores y otra de configuraciones.



*Figura 79.* Panel de Firestore en Firebase. Fuente: elaboración propia.

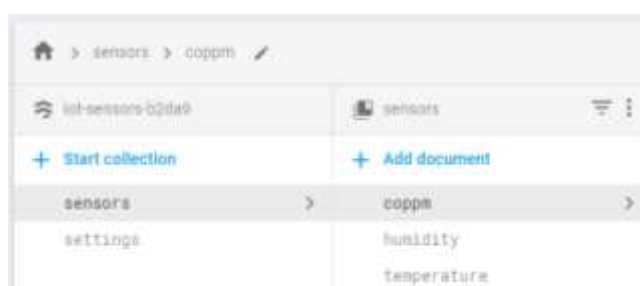
En la colección de configuraciones “settings”, almacenamos las configuraciones de la aplicación web, como por ejemplo el estado “measuring”, el cual indica a la placa

Wasmote si debe o no leer los datos de los sensores, así como el lugar donde se está midiendo. Este lugar se llama “measurementPlace”, y se puede apreciar en la *Figura 80*.



*Figura 80.* Colección de configuraciones. Fuente: elaboración propia.

De la misma forma, en la *Figura 81*, se puede apreciar los sensores disponibles en la aplicación, siendo estos el sensor de monóxido de carbono “coppm”, el sensor de humedad “humidity” y el sensor de temperatura “temperature”.



*Figura 81.* Colección de sensores. Fuente: elaboración propia.

Firestore ofrece un margen de almacenamiento gratuito, así como también una opción de pago por consumo con la cual brinda mayores prestaciones en el servicio. Sin embargo, siempre se indicará oportunamente cuando la cuota gratuita ha sido utilizada, lo que aporta mucha flexibilidad y escalabilidad a las aplicaciones que se usen.

El sistema de monitoreo está configurado para que tome muestras del ambiente y envíe los datos a la nube cada 10 segundos, esto da un total de 8640 escrituras a la base de datos diariamente. Esto quiere decir, que el consumo de escrituras a la base de datos por parte del Wasmote no supera la cuota que ofrece Firestore, por lo que se puede usar de forma gratuita. En la *Tabla 10* se puede apreciar este dato con mayor claridad.

*Tabla 10.* Estimado de consumo de Firestore.

Detalle	Cantidad equivalente
Equivalencia de 1 día en horas	24 horas
Equivalencia de 1 día en minutos	1440 minutos
Equivalencia de 1 día en segundos	86400 segundos
Intervalo entre cada medición almacenada	10 segundos
Mediciones almacenadas	8640 mediciones
Mediciones permitidas por Firestore	50000 mediciones

Fuente: Elaboración propia.

### 3.3.4 Etapa de visualización.

Como se mencionó anteriormente, la PWA es un híbrido entre página Web y aplicación móvil. Esta característica hace que en un smartphone o tablet no sea necesario abrir el navegador y escribir la URL del sitio web cada vez que se quiere visualizar los datos, sino que en su lugar se guarda el icono en la pantalla de inicio del móvil como si se tratase de una aplicación nativa instalada, tal como se puede apreciar en la *Figura 82*. Nuestra aplicación tiene el nombre de *IoT Sensors*.



*Figura 82.* Icono de la PWA en la pantalla de inicio de Android. Fuente: elaboración propia.

#### 3.3.4.1 Interfaz de la aplicación.

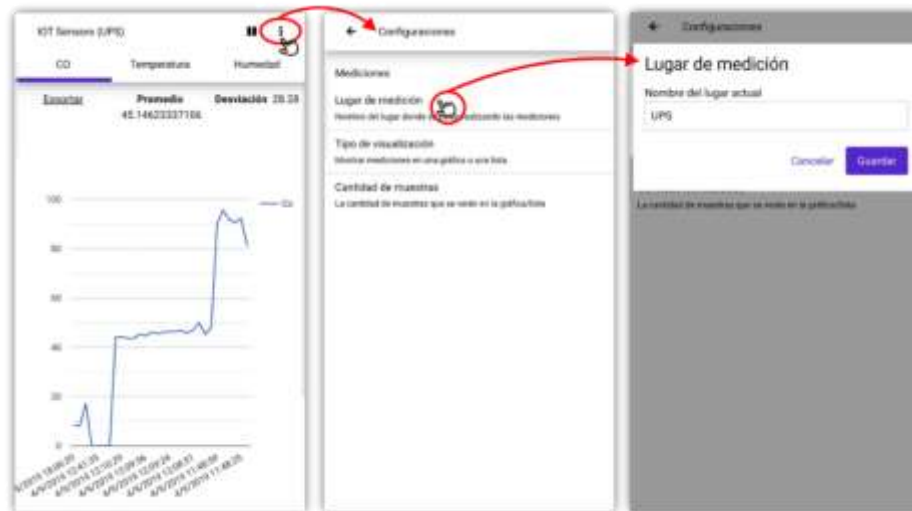
La aplicación presenta una interfaz sencilla pero funcional, con algunas características y opciones que permiten la interacción entre el equipo modular de medición y el usuario. En la *Figura 83* se puede observar la pantalla principal de la aplicación y los elementos que contiene.



*Figura 83.* Pantalla principal de la aplicación. Fuente: elaboración propia.

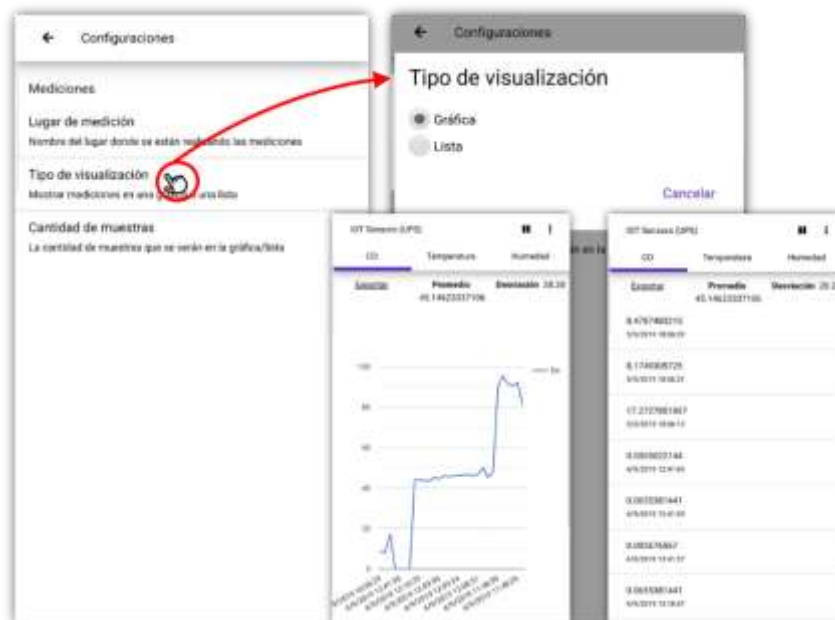
*Lugar de medición.* La aplicación cuenta con un menú de opciones, el cual permite poner un nombre al lugar actual de medición, de manera que los datos almacenados

sean referenciados a esta ubicación y poder analizar los niveles de calidad de aire en cada espacio por separado. En la *Figura 84* se puede observar con detalle este proceso:



*Figura 84.* Selección del lugar de medición. Fuente: elaboración propia.

*Tipo de visualización.* En la *Figura 85* se puede observar los dos tipos de visualización de datos con los que cuenta la aplicación. En el lado inferior izquierdo se observan los datos en formato de gráfica y en el inferior derecho los que están en formato de lista. Estos formatos permiten tener una mejor visualización de los datos para cada situación. Es importante recalcar que ambas formas de visualización se presentan junto con el registro de fecha y hora de toma de datos del ambiente.



*Figura 85.* Formatos de visualización de datos. Fuente: elaboración propia.

*Cantidad de muestras.* Los datos se visualizarán en la PWA en función de la cantidad de muestras que configuremos en esta opción. En la *Figura 86* se observa con más detalle el procedimiento, pudiendo visualizarse en la parte inferior izquierda la gráfica con 30 muestras y en la inferior derecha con 10 únicamente:

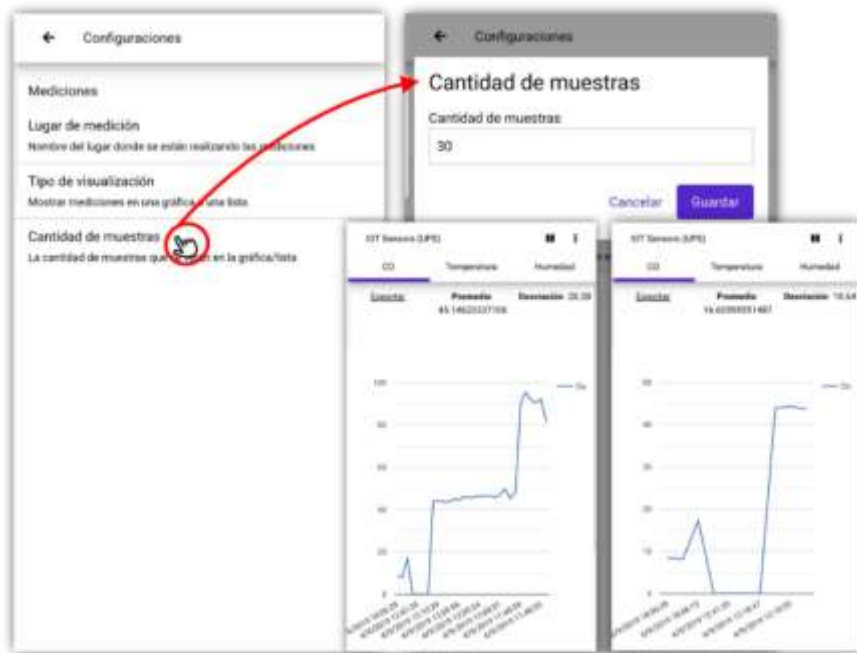


Figura 86. Selector de cantidad de muestras para visualizar. Fuente: elaboración propia.

*Pestañas de visualización de mediciones.* La aplicación PWA cuenta con 3 pestañas para poder visualizar de forma independiente las mediciones de cada sensor. Debajo de estas pestañas tenemos una sección donde se observa el valor promedio de las mediciones y su desviación estándar. Debajo de esta sección encontramos las gráficas de las mediciones que hemos seleccionado, tal como se muestra la Figura 87 :

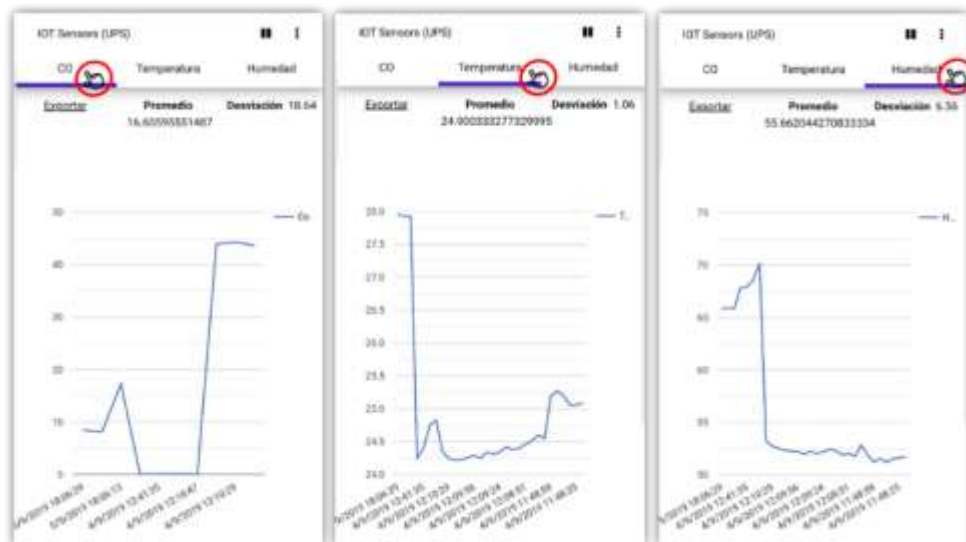


Figura 87. Pestañas de visualización de mediciones. Fuente: elaboración propia.

*Opción de exportar.* En la misma sección donde visualizamos el promedio y la desviación estándar de las mediciones, tenemos la opción de exportar (descargar) los valores obtenidos hasta el momento. Los valores se guardan en un archivo de formato CSV (Valores Separados por Comas), tal como se muestra en la Figura 88. Esto permite tener los datos a la mano para poder analizarlos.



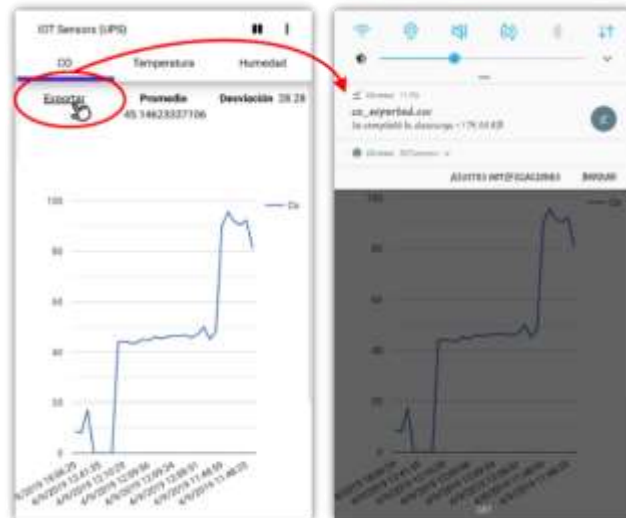


Figura 88. Exportación de los valores visualizados en la aplicación. Fuente: elaboración propia.

*Detener y reanudar las mediciones.* La aplicación cuenta con un botón de pausar y continuar las mediciones, tal como se muestra en la Figura 89. Este botón es muy útil a la hora de cambiar el lugar de medición, más aún cuando los lugares en donde se quiere hacer el monitoreo de calidad del aire se encuentran muy distanciados entre sí y se quiere evitar tomar datos mientras se moviliza el equipo.

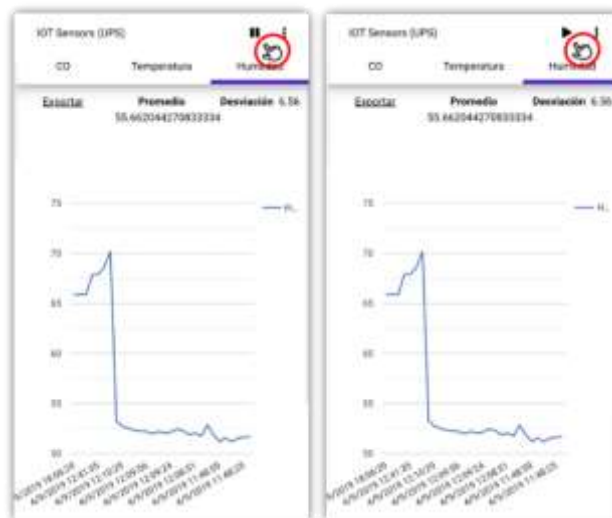


Figura 89. Opción de detener y reanudar las mediciones. Fuente: elaboración propia.

### 3.3.5 Tratamiento de los datos obtenidos.

La información que se obtenga del aire ambiente de cada uno de los lugares seleccionados será organizada en un documento en formato XLS, el cual es una extensión que permite el tratamiento de datos, sobre todo numéricos. Aquí se los clasificará en columnas con los encabezados correspondientes al día de la semana, fecha y hora de cada una de las variables tomadas. A esta clasificación se suma la categorización por niveles de medición, método que se explicará más adelante, en la sección 3.4 de este documento. La Figura 90 muestra el formato donde se almacenarán los datos para su posterior análisis:

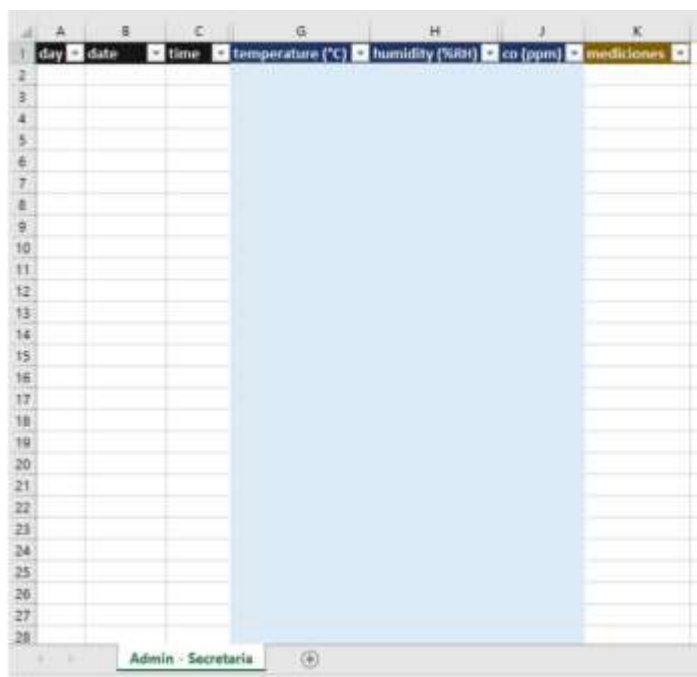


Figura 90. Formato para el análisis de los datos, elaboración propia. Fuente: elaboración propia.

Una vez organizada la información en el documento XLS, se procederá a escoger una muestra de los datos medidos en cada nivel. Esto quiere decir que se seleccionarán entre 20 y 25 valores de lo que se ha obtenido, por ejemplo, al medir la calidad del aire a 1 metro del piso en el centro de la sala. Este procedimiento se hace debido a que los sensores, a pesar de tener una capacidad de respuesta rápida frente a las variaciones de temperatura, humedad y monóxido de carbono, pudieran realizar lecturas erróneas al momento de cambiar de un nivel de medición a otro. En la *Figura 91* se puede observar que en las columnas donde se colocarán los datos medidos, hay 3 recuadros marcados en negro; ese será el rango que utilizaremos para el respectivo análisis.

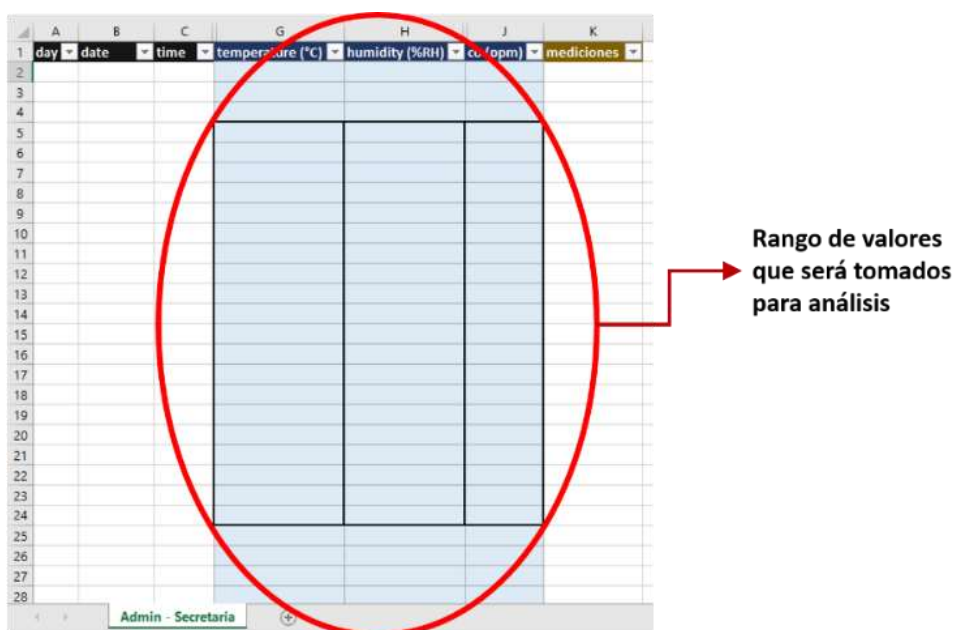
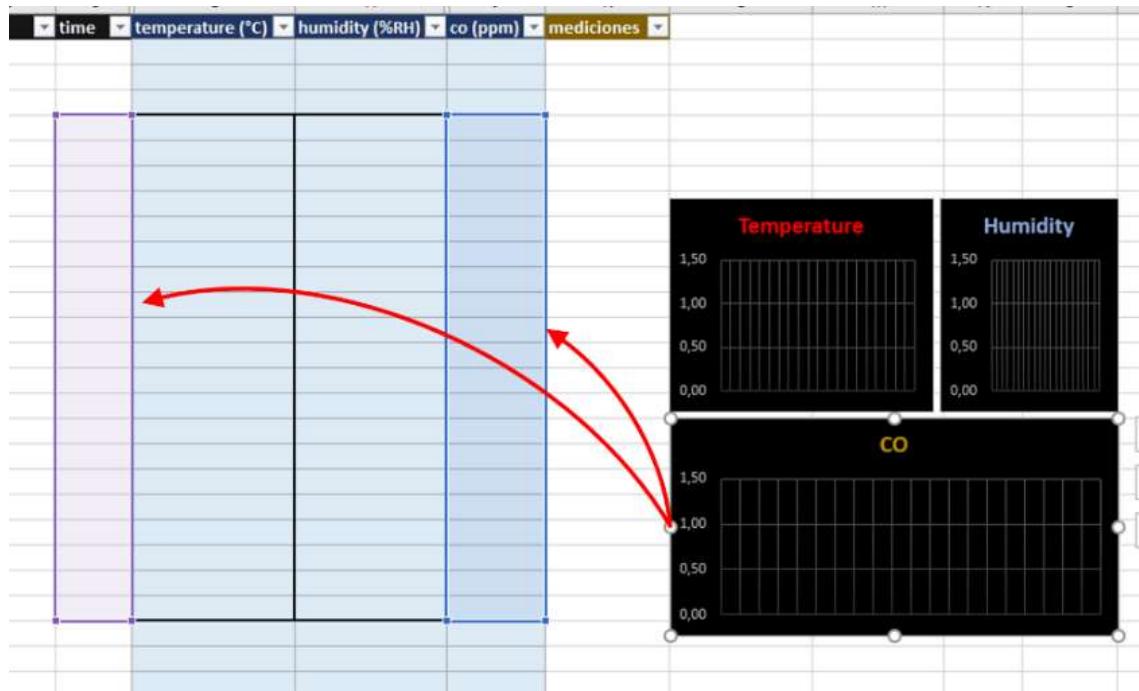
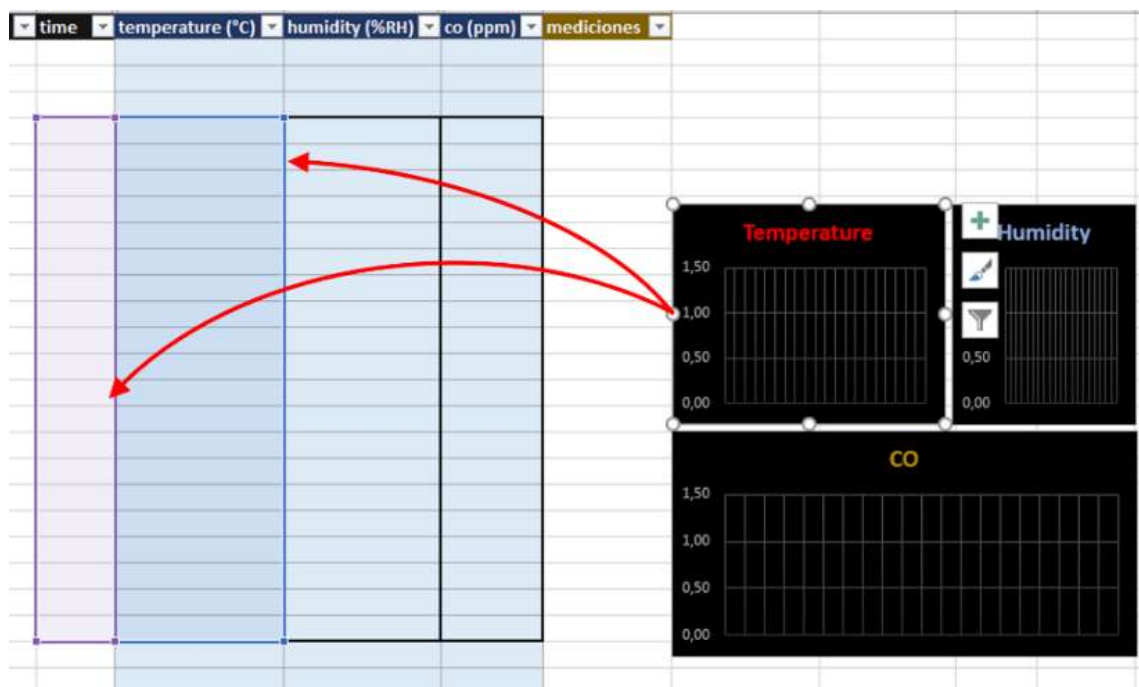


Figura 91. Rango de valores que serán analizados en las mediciones. Fuente: elaboración propia.

En el tratamiento de los datos contaremos también con tres gráficos, los cuales mostrarán las fluctuaciones de cada una de las variables ambientales que vamos a medir y que se encuentran en el rango establecido. En la *Figura 92*, *Figura 93* y *Figura 94* se puede apreciar los gráficos y las columnas de datos que se visualizarán en ellos:



*Figura 92.* Gráfico para valores medidos de CO. Fuente: elaboración propia.



*Figura 93.* Gráfico para valores medidos de temperatura. Fuente: elaboración propia.

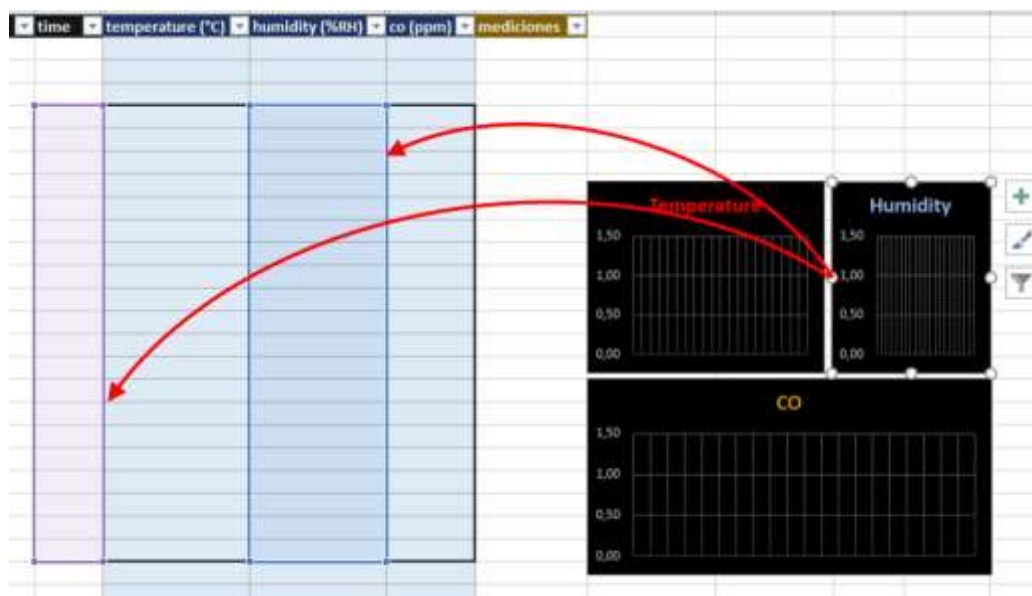


Figura 94. Gráfico para valores medidos de humedad. Fuente: elaboración propia.

Finalmente, los valores dentro del rango de cada nivel de medición serán mostrados en una tabla donde aparecerá su promedio y desviación estándar. Para las celdas que muestran el promedio se ha hecho una configuración de colores, lo que permite visualizar si el valor promediado se encuentra dentro de los márgenes establecidos por los organismos normalizadores de los parámetros de calidad del aire. En la *Tabla 11* se puede observar el código de colores y los niveles permitidos por las instituciones que realizan recomendaciones para los espacios interiores y centros hospitalarios (OSHA y UNE):

Tabla 11. Parámetros ambientales y su normalización para hospitales.

Variable	Organismo	Lugar	Rango	Índice	Color
CO	OSHA	Todos	0 – 50 ppm	Moderado	Verde
		Todos	51 – 100 ppm	Alerta	Amarillo
		Todos	101 – 200 ppm	Severo	Rojo
		Todos	201 – 1000 ppm	Peligroso	Rojo
Temperatura	UNE	Quirófano	0 – 21°C	No adecuado	Amarillo
		Quirófano	22 – 26°C	Adecuado	Verde
		Quirófano	27 – 30°C	No permitido	Rojo
		Todos	0 – 23°C	No adecuado	Amarillo
		Todos	24 – 26°C	Adecuado	Verde
Humedad	UNE	Todos	0 – 44 %HR	No permitido	Rojo
		Todos	45 – 55 %HR	Adecuado	Verde
		Todos	56 – 100 %HR	No permitido	Rojo

Fuente: Elaboración propia.

### 3.4 Pruebas y Funcionamiento

El sistema de monitoreo de calidad del aire entró en funcionamiento en el Hospital de Niños León Becerra del 19 al 29 de marzo de 2019, realizando las mediciones de los parámetros ambientales de monóxido de carbono, temperatura y humedad en 9 salas, las cuales se detallan en la *Tabla 12*, junto con las habitaciones o compartimientos internos y el número de personas que transitan semanalmente en ellas:

*Tabla 12.* Salas en las que se realizará la medición de calidad del aire.

Salas del HLB	Divisiones	Beneficiarios
Quirófanos	6	19
Central de Esterilización	2	9
Postoperatorio	2	19
Unidad de Cuidados Intensivos	6	38
Emergencia	1	21
Triaje	2	293
Observación	2	18
Pensionado de Primera	23	94
Departamento Administrativo	4	5

Fuente: Elaboración propia.

Estas mediciones se realizaron en 3 días distintos para cada lugar y en diferentes horarios para tener mayor confiabilidad en los resultados obtenidos. En la *Tabla 13* se puede apreciar el cronograma de las visitas y el tiempo que se estuvo realizando las mediciones en cada lugar:

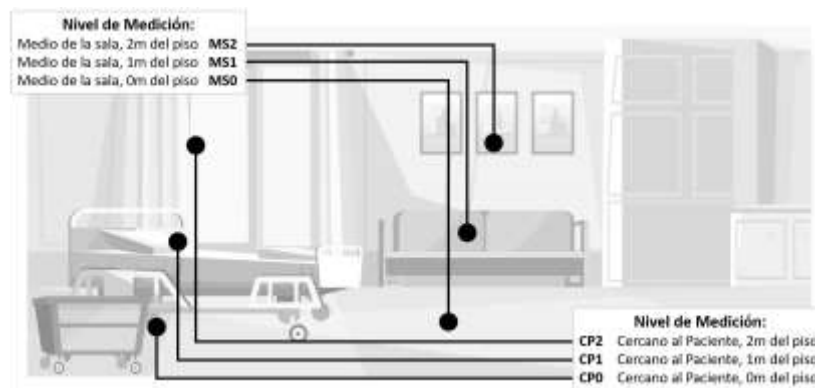
*Tabla 13.* Cronograma de visitas al HLB para las mediciones.

Fecha	Hora	Salas	Tiempo
19/03/19	15h00	Administración	90 min
19/03/19	17h00	Unidad de Cuidados Intensivos	130 min
20/03/19	14h00	Emergencia – Triaje – Observación	180 min
21/03/19	14h30	Quirófanos – C. Esterilización – Postoperatorio	210 min
22/03/19	08h30	Emergencia – Triaje – Observación	180 min
22/03/19	12h45	Unidad de Cuidados Intensivos	130 min
22/03/19	15h45	Pensionado de primera	150 min
23/03/19	09h00	Quirófanos – C. Esterilización – Postoperatorio	210 min
25/03/19	13h45	Administración	90 min
25/03/19	15h45	Emergencia – Triaje – Observación	180 min
26/03/19	07h00	Pensionado de primera	150 min
26/03/19	16h00	Quirófanos – C. Esterilización – Postoperatorio	140 min
28/03/19	13h30	Pensionado de primera	150 min

28/03/19	16h15	Unidad de Cuidados Intensivos	130 min
29/03/19	08h00	Quirófanos	70 min
29/03/19	13h15	Administración	90 min

Fuente: Elaboración propia.

La toma de muestras del ambiente se realizó cada 10 segundos durante un tiempo de 5 minutos a distintos niveles de proximidad (con respecto al paciente) y a distintas alturas (con respecto al piso). En relación al paciente, las mediciones se hicieron entre 0.5 y 1 m de distancia, y luego se realizaron en el centro de la sala. Para ambas mediciones se ubicará el dispositivo a 0m, a 1m (altura a la que normalmente se encuentra las camillas de los pacientes y escritorios del personal que labora en el lugar) y a 2m del piso. Este método de toma de muestras fue propuesto en el proyecto de extensiones universitarias realizado en el año 2018 por la Universidad Politécnica Salesiana, con el título: *Detección de los niveles de contaminación de aire en las salas del Hospital de niños León Becerra mediante un sistema de monitoreo de calidad del aire*. Para una mejor comprensión, en la *Figura 95* se muestra la nomenclatura de cada uno de estos niveles junto con su respectiva ubicación.



*Figura 95.* Niveles de medición dentro del quirófano. Fuente: elaboración propia. Autor de dibujo: Katemangostar (<https://www.freepik.es/fotos-vectores-gratis/fondo>)

## 4. Análisis y Resultados

### 4.1 Análisis de las Mediciones en el HLB

El tiempo de toma de datos en las salas seleccionada del hospital fue de 38 horas en total. Durante este tiempo se recolectaron 14504 muestras, tal como se puede observar en la *Tabla 14*:

*Tabla 14.* Cantidad de datos recolectados por cada sala del HLB.

Lugar	Cantidad de muestras
Administración	1702
C. Esterilización	563
Emergencia	1245
Observación	2248
Pensionado	2954

Postoperatorio	820
Quirófanos	2670
UCI	2298
<i>Total</i>	<i>14500</i>

Fuente: Elaboración propia.

Los datos medidos se clasificaron por medio de tablas, mostrando el promedio de dichas mediciones. A continuación, se realizará el análisis de los resultados obtenidos en cada uno de los lugares. Es conveniente mencionar que los datos en el formato original se anexarán a este proyecto, donde se podrá apreciar de mejor manera el código de colores que determina los rangos permitidos en cada una de las variables.

#### 4.1.1 Quirófanos.

En la *Tabla 15* se observa el resultado de las mediciones en cada uno de los quirófanos. A nivel de monóxido de carbono se puede constatar que ninguno presenta niveles por encima de las 50 ppm, siendo el nivel máximo registrado el del quirófano 5 con 7 ppm. En cuanto a nivel de humedad, el quirófano 5 es el único que se encuentra dentro del rango permitido en los 3 días de medición y el quirófano 4 es el que más problemas tiene en ese sentido, mostrando en los 3 promedios niveles por encima del rango establecido por la norma. Para los niveles de temperatura se presentan resultados aceptables, ya que se encuentran dentro del rango ideal para su correcto funcionamiento.

*Tabla 15.* Promedio de mediciones en Quirófanos.

Lugar	Fecha	Temperatura (°C)	Humedad (%)	CO (ppm)
Quirófano 1	21/03/2019	23.79	60.21	0.66
	23/03/2019	24.05	53.83	6.45
	26/03/2019	22.83	59.51	4.52
Quirófano 2	21/03/2019	23.32	59.18	0.27
	23/03/2019	23.16	54.46	5.40
	26/03/2019	26.14	51.52	0.53
Quirófano 3	21/03/2019	23.42	58.96	0.63
	23/03/2019	23.10	53.29	5.23
	26/03/2019	23.55	52.49	3.33
Quirófano 4	21/03/2019	23.26	62.58	1.12
	23/03/2019	22.68	56.04	4.46
	26/03/2019	22.66	56.21	3.29
Quirófano 5	21/03/2019	24.14	52.97	6.66
	23/03/2019	21.79	52.73	3.34
	26/03/2019	24.14	51.06	5.86

Fuente: Elaboración propia.

#### 4.1.2 Unidad de Cuidados Intensivos.

Por medio de la *Tabla 16* se puede constatar los resultados de las mediciones en el área de Cuidados Intensivos. Del parámetro Temperatura se observa que los niveles se encuentran por debajo de lo permitido, siendo en el sector de Aislamiento donde se registran los valores más bajos. Esto se puede explicar debido a que las bajas temperaturas impiden la proliferación de bacterias dañinas. Esto se confirma con el registro de niveles de humedad, ya que en el área de aislamiento los valores se encuentran dentro del rango adecuado. El promedio máximo de humedad detectado es de 55 % HR y el mínimo es de 42 %. En cuanto a las partículas de monóxido de carbono se registran valores por debajo de las 50 ppm, siendo el rubro más alto el de 7.

*Tabla 16.* Promedio de mediciones en Unidad de Cuidados Intensivos.

Lugar	Fecha	Temperatura (°C)	Humedad (%)	CO (ppm)
UCI	19/03/2019	22.58	45.47	7.45
	22/03/2019	21.47	47.11	2.48
	28/03/2019	23.00	41.76	3.25
Hall de aislamiento	19/03/2019	15.96	54.99	0.41
	22/03/2019	19.46	48.06	2.18
	28/03/2019	18.86	47.28	0.95
Aislamiento 2	19/03/2019	15.53	53.80	0.31
	22/03/2019	18.29	51.79	0.75
	26/03/2019	15.40	53.19	5.86

Fuente: Elaboración propia.

#### 4.1.3 Postoperatorio.

La *Tabla 17* muestra los resultados de las mediciones en la sala de Postoperatorio, siendo notorio que los valores de temperatura y humedad sobrepasan ligeramente los rangos permitidos. La medición más alta registrada de temperatura es de 26 °C y el más alto porcentaje de humedad es de 57 %HR. Para la variable de monóxido de carbono no se presentan valores por encima de 50ppm, siendo su punto más alto el registrado en la tercera visita con 9 ppm y el más bajo en la primera con 4 ppm.

*Tabla 17.* Promedio de mediciones en Postoperatorio.

Lugar	Fecha	Temperatura (°C)	Humedad (%)	CO (ppm)
Postoperatorio	21/03/2019	26.32	56.08	3.58
	23/03/2019	25.19	54.97	7.03
	26/03/2019	26.46	56.66	8.55

Fuente: Elaboración propia.

#### 4.1.4 Central de esterilización.

Los datos medidos en la Central de esterilización (*Tabla 18*) son los siguientes:



Los niveles promediados de temperatura están entre 27 y 28 °C, lo cual está por encima de los valores permitidos. La misma situación se presenta en los porcentajes de temperatura (entre 59 y 63 %HR). Esto se puede explicar al tomar en cuenta que en este lugar se realiza la esterilización de los implementos médicos y demás elementos que se utilizan en el hospital y que requieren purificarse constantemente, lo que hace que los valores referidos se mantengan altos la mayor parte del tiempo. Los niveles de monóxido de carbono se mantienen por debajo de las 50 ppm, siendo el punto más alto el de 4, detectada en la visita del 26 de marzo.

*Tabla 18.* Promedio de mediciones en Central de Esterilización.

Lugar	Fecha	Temperatura (°C)	Humedad (%)	CO (ppm)
Central de Esterilización	21/03/2019	27.60	62.10	0.29
	23/03/2019	27.30	62.60	0.49
	26/03/2019	27.44	58.65	4.31

Fuente: Elaboración propia.

#### 4.1.5 Emergencia.

A continuación, se observa los valores promediados en el área de Emergencia, específicamente en el sector de Triage y salas con camillas (*Tabla 19*). En Triage se detectan niveles de temperatura y humedad por encima del rango que establece la normativa UNE (entre 28 y 29 °C y 60 a 72 %HR, respectivamente). La sala con camillas mantiene valores permitidos para temperatura, no así para humedad, teniendo un porcentaje de humedad relativa que va de 59 a 66 %. Su nivel de monóxido de carbono se encuentra por debajo de las 50 ppm en ambos sectores, siendo su punto más alto el de la sala de camillas con 9 partes por millón.

*Tabla 19.* Promedio de mediciones en Emergencia.

Lugar	Fecha	Temperatura (°C)	Humedad (%)	CO (ppm)
Triage	20/03/2019	28.55	71.76	0.72
	22/03/2019	28.35	67.73	0.49
	25/03/2019	28.65	59.61	8.36
Camillas	20/03/2019	25.71	65.13	0.75
	22/03/2019	25.22	65.72	0.22
	25/03/2019	25.79	58.82	8.72

Fuente: Elaboración propia.

#### 4.1.6 Observación.

La *Tabla 20* muestra los resultados en el área de Observación. Los valores promediados de CO en los 3 días están dentro del rango permitido por OSHA, siendo su punto más alto el de 7. A nivel de humedad, los datos indican que no se cumple con la normativa UNE, ya que mantiene porcentajes que van desde 56 a 70 % de humedad relativa. Para temperatura, los valores se encuentran debajo del rango permitido, siendo su punto más bajo 22 °C y el más alto 23 °C.

Tabla 20. Promedio de mediciones en Observación.

Lugar	Fecha	Temperatura (°C)	Humedad (%)	CO (ppm)
Observación	20/03/2019	23.03	61.70	6.93
	22/03/2019	23.21	70.44	0.07
	25/03/2019	22.22	55.84	4.66

Fuente: Elaboración propia.

#### 4.1.7 Pensionado de primera.

En la *Tabla 21* se muestran los valores obtenidos en las mediciones en Pensionado de primera. A nivel de monóxido de carbono se puede observar que son los niveles más bajos detectados en las salas del hospital donde se hicieron las mediciones, siendo su punto más alto el de 1 ppm en la segunda visita, específicamente en recepción y en una de las habitaciones del pasillo derecho. En contraparte, en este lugar se detectan los niveles más altos de temperatura y humedad (32 °C y 71 %HR, respectivamente). Esto se explica por el hecho de que el hospital, en este sector, no cuenta con sistema de acondicionamiento de aire, lo que permite que los niveles de temperatura y humedad no sean los adecuados para tener un funcionamiento óptimo. El punto más bajo de temperatura es de 29 °C y para humedad 63 %HR.

Tabla 21. Promedio de mediciones en Pensionado de Primera.

Lugar	Fecha	Temperatura (°C)	Humedad (%)	CO (ppm)
Recepción	22/03/2019	30.37	69.76	0.17
	26/03/2019	29.31	68.75	1.28
	28/03/2019	29.41	69.13	0.17
Habitación pasillo derecho	22/03/2019	32.40	63.43	0.39
	26/03/2019	31.25	64.13	0.65
	28/03/2019	30.73	66.41	0.28
Habitación pasillo izquierdo	22/03/2019	31.55	67.43	0.34
	26/03/2019	30.30	65.34	0.49
	28/03/2019	30.05	68.34	0.24
Baño general	22/03/2019	30.69	69.63	0.31
	26/03/2019	29.73	70.25	0.37
	28/03/2019	29.61	71.15	0.22

Fuente: Elaboración propia.

#### 4.1.8 Administración.

Los niveles promediados en esta área del hospital se los puede observar en la *Tabla 22*. Se han detectado valores de temperatura superiores al rango permitido, ya que estos valores oscilan entre 26 y 27 °C. La medición de porcentaje de humedad arroja valores por encima del límite permitido, siendo el nivel máximo 62 %HR. En cuanto a monóxido de carbono se puede ver que los niveles no sobrepasan el límite permitido

de 50 ppm; sin embargo, se observa el nivel más alto de este gas medido en el hospital, llegando a 10ppm, y un promedio de 9,3 en esta área en la segunda visita.

Tabla 22. Promedio de mediciones en Administración.

Lugar	Fecha	Temperatura (°C)	Humedad (%)	CO (ppm)
Recepción	19/03/2019	26.41	60.05	0.83
	25/03/2019	26.07	57.68	8.79
	29/03/2019	25.53	60.51	0.61
Vicepresidencia	19/03/2019	26.66	59.95	2.10
	25/03/2019	26.27	58.25	9.31
	29/03/2019	25.76	61.94	0.37
Financiero	19/03/2019	26.78	54.53	8.39
	25/03/2019	27.18	53.48	9.73
	29/03/2019	25.76	61.94	0.37

Fuente: Elaboración propia.

De los resultados obtenidos del sistema de monitoreo de calidad del aire se puede indicar que:

- El nivel de monóxido de carbono medido en las salas durante los 3 días no supera el límite permitido por OSHA que es de 50 ppm. Sin embargo, el nivel recomendado para este contaminante disuelto en el aire es de 1 a 2 ppm, por lo que se considera que el aire ambiente en algunas salas no es lo suficientemente puro.
- El nivel de temperatura medido en algunas salas del hospital no se encuentra entre el rango permitido por UNE en su normalización 100713-2005, ya que se detectaron valores que llegan hasta los 32 °C. Esto debe motivar a que se establezcan políticas dentro del centro hospitalario que contribuyan al mejoramiento del ambiente en aquellos lugares que no cuentan con sistemas de climatización.
- El nivel de humedad obtenido en determinadas salas, sobrepasa por mucho los valores permitidos por la normalización UNE 100713-2005, teniendo porcentajes mayores a 70. Esto debe ser tomado en cuenta por las autoridades del hospital, ya que un nivel elevado de humedad puede ocasionar proliferación de bacterias que pueden ser dañinas para el ser humano.

#### 4.2 Análisis del Método de Medición

Como se mencionó en la sección 3.3.1 de este documento, los parámetros de calidad del aire se midieron en base al método por niveles. En la *Figura 96*, se observan los resultados obtenidos en las 3 visitas realizadas en la sala de Administración.

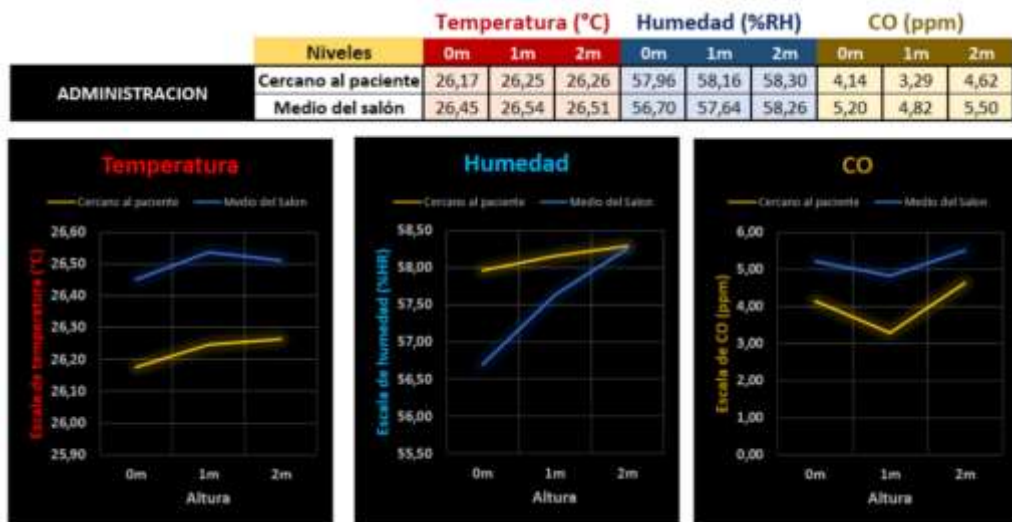


Figura 96. Valores obtenidos en la medición por niveles en la sala de Administración. Fuente: elaboración propia.

Si bien los valores medidos a distintos niveles no varían considerablemente, sí presentan cambios, los cuales podrían ser objeto de un análisis más técnico en estudios futuros. A nivel de temperatura existe una variación de 0.09 °C en los valores medidos a nivel del paciente, desde el piso hasta los 2 m de altura. El sensor de humedad, en cambio, muestra una variación de 0.34 %HR; y para la medición de partículas de CO la variación va de 3.29 a 4.62 ppm. Este último dato es interesante, ya que a 1m del piso los niveles de monóxido de carbono se reducen en comparación con los medidos a 0m y a 2m de altura.

En la *Figura 97* se puede observar el promedio general de mediciones en todas las áreas del HLB que fueron seleccionadas, pudiendo constatarse que, en términos generales, las concentraciones de monóxido de carbono en el ambiente a 1 metro de altura son más bajas en comparación con las que se miden en el piso y a 2 metros de altura.

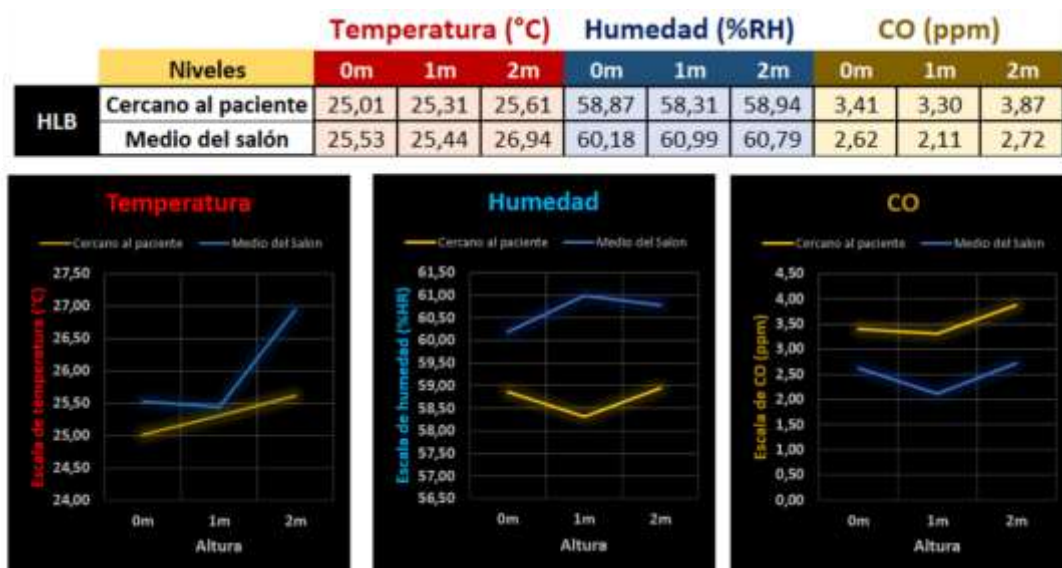


Figura 97. Promedio de Mediciones en el Hospital de Niños León Becerra. Fuente: elaboración propia.

### **4.3 Análisis de la Implementación del Hardware**

*Operatividad.* Este equipo ha estado en funcionamiento desde 15 de marzo de 2019, obteniendo un tiempo de operación de más de 60 horas, entre mediciones de pruebas de sensores, mediciones en el hospital y otros procedimientos para asegurar el correcto funcionamiento del mismo.

*Portabilidad.* El equipo implementado cuenta con las dimensiones (188mm x 113mm x 60mm) y peso (245g) adecuados para poder movilizarlo sin inconvenientes hacia cualquier lugar.

*Conectividad a internet.* La tarjeta Raspberry Pi3 hace la función de gateway en el sistema. Se comprobó en múltiples pruebas que tarda un promedio de 18 segundos en conectarse a la red wifi más cercana, desde el momento en que se enciende.

*Autonomía.* Las dos baterías, conjuntamente con el bajo consumo de cada una de las partes del equipo (su potencia total, entre las placas Wasmote y la Raspberry, suman 5.9w), proporcionan un nivel de autonomía aceptable para lo que se espera del sistema de medición. Se ha sometido a pruebas de medición continua, logrando hacer lecturas cada 10 segundos de manera ininterrumpida por 10 horas.

*Costos y accesibilidad.* El sistema está diseñado para su uso, tanto doméstico como a nivel industrial, ya que cuenta con componentes de excelente calidad. Por esta razón los costos de cada elemento son relativamente elevados (se adjunta el detalle de costos en la sección de anexos). A nivel de accesibilidad, actualmente no se cuenta con muchas empresas en el mercado nacional que comercialicen estos equipos, siendo InnovGroup la única en el país que trabaja con productos de Wasmote hasta la fecha.

### **4.4 Análisis de la Implementación del Software**

*IDE.* El entorno de programación de la tarjeta de Wasmote es bastante amigable, ya que está basado en lenguaje C, siendo este un lenguaje de programación muy conocido. Adicional a esto, la empresa Wasmote en su página oficial, presenta muchos ejemplos de códigos y librerías que son útiles a la hora de hacer pruebas de configuración de la tarjeta en el IDE.

*Base de datos en la nube.* Firebase de Google ha respondido convenientemente en la implementación del sistema de medición, almacenando la información de manera segura y sin riesgos de pérdida de datos. Sin embargo, es recomendable que de ser posible se realice la contratación del servicio pagado, ya que esto implica mayor capacidad de almacenamiento de datos y mayor ancho de banda.

*PWA.* La aplicación permite visualizar los datos medidos e interactuar con el usuario de manera conveniente. El diseño de la aplicación ha requerido de un alto conocimiento en programación, sobre todo a nivel de estructuras de código y procesos de conversión de un lenguaje de programación sencillo a otro u otros más complejos que son los que entiende la computadora.

## **Conclusiones**

Este proyecto es un aporte significativo a la formación de ingenieros en Electrónica y Automatización con alta capacidad profesional, socialmente responsables, capaces de favorecer y aportar en el cambio de la matriz productiva del país, mediante la integración de las tecnologías en electrónica, instrumentación y de ordenadores, sus aplicaciones y adaptaciones para la innovación y desarrollo con compromiso social y ambiental.

La implementación de este equipo modular de medición de calidad del aire en ambientes interiores, contribuye al mejoramiento del servicio que ofrece el HLB a sus usuarios y personal que labora en sus instalaciones, facilitando el monitoreo permanente de contaminantes que puedan afectar la salud de quienes transitan diariamente por el centro hospitalario.

Las ventajas que ofrece el mundo del Internet de las Cosas son considerables, ya que contribuyen al desarrollo de tecnologías aplicadas a muchas áreas importantes para el ser humano, siendo una de ellas el sector de la salud.

### **Conclusiones específicas**

El diseño e implementación del sistema de monitoreo de calidad del aire a través de sensores ha sido posible gracias al acoplamiento de múltiples dispositivos, los cuales han cumplido con los requerimientos exigidos en todas sus etapas. Este sistema entró en funcionamiento en las salas seleccionadas del Hospital de Niños León Becerra, por un periodo de tiempo de 38 horas, contribuyendo a que exista el primer estudio técnico de la calidad del aire interior en el centro hospitalario.

Los sensores utilizados para el diseño e implementación del sistema, acoplados a la tarjeta Wasmote, pudieron configurarse fácilmente para obtener los datos del ambiente en intervalos programables. Esto gracias al Entorno de Desarrollo Integrado de Wasmote, creado a partir del lenguaje C, estando presente este un lenguaje de programación en el pensum de las carreras universitarias afines.

Gracias a la tecnología wifi incorporada en la tarjeta RaspberryPi3, el envío a la nube de los datos medidos en el ambiente fue realizado con éxito, dado que esta tarjeta sirvió como gateway entre la información que midieron los sensores y la base de datos en internet.

Firestore de Google y su plataforma de base de datos, hicieron que sea posible el almacenamiento, administración y procesamiento de la información que se obtuvo de las mediciones del ambiente en un porcentaje del 100%. Es importante, sin embargo, que se pueda a futuro contratar el servicio completo de Firestore, para así tener acceso a todas las características y ventajas que dicha plataforma ofrece.

La implementación de una aplicación web progresiva (PWA) fue uno de los principales aciertos de este proyecto, puesto que permitió visualizar en cualquier dispositivo electrónico (PC, tablet o smartphone) la información almacenada en la base de datos en la nube, gracias a su interfaz amigable y funcional para el usuario.

Los datos ambientales obtenidos por medio de los sensores ofrecieron información relevante del nivel de contaminación en las salas seleccionadas del hospital. Estos datos indican que algunas de las áreas presentan niveles no permitidos en cuanto a temperatura y humedad. Lo expuesto debe motivar a que las autoridades del centro hospitalario establezcan medidas pertinentes para mejorar las condiciones de dichos ambientes.

Las mediciones realizadas a distintos niveles del suelo y a distancias variables con respecto al paciente, ofrecen una nueva forma de medir los contaminantes que pudieran estar presentes en el aire ambiente interior, ya que se constató que, al medir en distintas posiciones, los niveles de temperatura, humedad y monóxido de carbono varían. Esto puede ser el inicio para que a futuro se realicen estudios más minuciosos en este aspecto.

Si bien el sistema de monitoreo logró cumplir con los requerimientos establecidos en el presente proyecto, la adquisición de algunos de los equipos (tarjeta lectora de gases, placa Waspote, entre otros) es limitada en el país, trayendo como consecuencia que se tengan que importar los equipos y por ende los costos de implementación del sistema sean relativamente elevados.

## **Recomendaciones**

Para el funcionamiento del equipo de medición:

Es necesario que el equipo de medición tenga acceso a la red wifi del lugar que va a monitorear, ya que, si está conectada a una red más lejana, podría presentar inconvenientes para enviar los datos a la nube y estos podrían perderse. Por otro lado, es recomendable crear una red wifi independiente para los sensores, con la finalidad de evitar sobrecargas en la red de datos.

Es importante evitar que la batería externa que alimenta el dispositivo alcance niveles muy bajos de energía, ya que esto puede ocasionar que el equipo funcione de forma irregular. Si se va a monitorear una sala por un tiempo mayor a 8 horas, es mejor que el equipo esté conectado a una toma de corriente.

Se debe evitar que el equipo permanezca mucho tiempo en zonas donde el porcentaje de humedad sea muy alto, ya que los componentes electrónicos pueden averiarse. De igual manera, es recomendable no extraer la tarjeta MicroSD ni los sensores cuando el equipo está midiendo, puesto que se podrían alterar los datos obtenidos o pueden no grabarse en la tarjeta si se la inserta de nuevo.

Para el hospital:

Establecer dentro de centro hospitalario planes de mejora continua, realizando las mediciones de calidad de aire regularmente. Esto ayudaría a evitar que los contaminantes presentes en el ambiente se encuentren en niveles no saludables para las personas que transitan diariamente por el lugar.

## **Líneas futuras**

- Complementar el sistema de monitoreo de calidad del aire incorporando nuevos parámetros a medir en el ambiente, como por ejemplo ruido y material particulado.
- Agregar a la PWA otras funciones para el tratamiento de los datos visualizados, a más de las que cuenta, como podría ser: descargar en una hoja de cálculo los gráficos obtenidos en las mediciones, visualizar niveles máximos y mínimos detectados, visualizar valores con códigos de colores, etc.
- Automatizar el método de medición por niveles, haciendo uso de una plataforma móvil que pueda desplazar el equipo a diferentes alturas.
- Profundizar el análisis de las variables en el ambiente, relacionándolas con otras variables, como podría ser la presión atmosférica y su influencia en los niveles de monóxido de carbono presentes en los hospitales de provincias de altura.

## **Proyectos de Investigación Vinculados**

- Implementación de un prototipo IoT con semáforos inteligentes a través de una infraestructura de computación en la niebla, Valle Pazmiño Sebastián
- Metamodelo para la integración del internet de las cosas y redes sociales, Rodríguez Molano José Ignacio
- El internet de las cosas y las consideraciones de seguridad, Cuzme Rodríguez Fabián Geovanny
- Estudio del modelo de referencia del internet de las cosas (IoT), con la implementación de un prototipo domótico, Peña Merizalde Janneth Lilitiana
- Calidad de servicios en la nube en combinación con el internet de las cosas: revisión sistemática de la literatura y modelo de calidad, Pérez Zúñiga Juan Gabriel
- Diseño e implementación de una aplicación IoT en la nube de Azure para el análisis de imagen, Mora Rodríguez Alejandro
- Método para la Integración de IoT y Cloud en el desarrollo de aplicaciones SIG, Rodríguez López Daniel Alberto
- Detección de los niveles de contaminación de aire en las salas del hospital de niños León Becerra mediante un sistema de monitoreo de calidad de aire, Vásquez Cevallos Leonel
- Análisis de interoperabilidad de plataformas IoT aplicado al desarrollo de un sistema de monitoreo de polución de aire para la ESPOCH, Coronel Silva Valeria Estefanía, Tenelanda Santillán Dennis Miguel



## Bibliografía

- Arduino. (2008). *Arduino Nano*. Obtenido de arduino.cc: <https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>
- B. Navya Rupa, G. K.-h. (2015). Test Report Generation Using JSON. *International Journal of Software Engineering and Its Applications*.
- Banzi, M. (2008). *Getting Started with Arduino*.
- Batini, C. (s.f.). *Conceptual Database Design, An Entity-relationship Approach*. 1992: Benjamin/Cummings.
- Ben Everard, E. U. (2014). *Learning Computer Architecture with Raspberry Pi*.
- Bliznakoff Del Valle, D. (14 de Noviembre de 2014). IoT: Tecnología, usos, tendencias y desarrollo futuro (tesis, maestría). Barcelona, España: Universidad Abierta de Cataluña.
- Bliznakoff Del Valle, D. (6 de Noviembre de 2014). IoT: Tecnologías, usos, tendencias y desarrollo futuro.
- Blum, C. B. (2013). *Sams Teach Yourself Python Programming for Raspberry Pi in 24 Hours*.
- Boxall, J. (2013). *Arduino Workshop: A Hands-On Introduction with 65 Projects*.
- Brock Craft, J. E. (2015). *Raspberry Pi Projects For Dummies*.
- C.Calderón, M. C. (2014). Desarrollo de una Aplicación Cliente/Servidor para un Wall View en base a la. *REVISTA EPN*, 7. Obtenido de [https://revistapolitecnica.epn.edu.ec/ojs2/index.php/revista\\_politecnica2/articloe/view/155/pdf](https://revistapolitecnica.epn.edu.ec/ojs2/index.php/revista_politecnica2/articloe/view/155/pdf)
- Ceballos, E. L. (2013). Diseño web adaptativo o responsivo. *Revista Digital Universitaria*.
- Elio, E. M. (2015). *Microcontrolador Arduino*. Universidad Cristobal Colón.
- EPA. (22 de Febrero de 2016). *Centro de Información sobre Contaminación de Aire (CICA)*. Recuperado el 12 de Septiembre de 2019, de [https://www3.epa.gov/ttnatc1/cica/help/haqshaps\\_s.html](https://www3.epa.gov/ttnatc1/cica/help/haqshaps_s.html)
- EPA. (19 de Enero de 2017). *Historia de la EPA*. Recuperado el 2 de Septiembre de 2019, de <https://www.epa.gov/history>
- Fernando Bizzarro, A. H. (2016). *The V-Dem Party Institutionalization Index: a new global indicator (1900-2015)*.
- Figaro. (2007). *Technical Information for TGS2442*. Recuperado el 23 de Agosto de 2019, de [http://www.produktinfo.conrad.com/datenblaetter/175000-199999/183303-da-01-en-GASSENSOR\\_FIGARO\\_TGS2442\\_B00.pdf](http://www.produktinfo.conrad.com/datenblaetter/175000-199999/183303-da-01-en-GASSENSOR_FIGARO_TGS2442_B00.pdf)
- Flores Carbajal, E. E. (30 de Octubre de 2012). *Redes de Sensores Inalámbricas Aplicado a la Medicina*. Cantabria, España.
- Guillermo Diez-Andino Sancho, R. M. (2003). Desarrollo de un servidor HTTP para dispositivos móviles en J2ME. *Departamento. Ingeniería Telemática - Universidad Carlos III de Madrid*. Obtenido de <http://www.it.uc3m.es/celeste/papers/ServidorHTTP.pdf>
- Javier, L. S. (2001). Base de Datos Distribuidas Estudio de Actualización de Réplicas. *Facultad de Informática - UNLP*. Obtenido de [http://sedici.unlp.edu.ar/bitstream/handle/10915/3872/Documento\\_completo\\_.pdf-PDFA1b.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/3872/Documento_completo_.pdf-PDFA1b.pdf?sequence=1)
- Jayesh Umre, K. B. (2014). Comparative performance analysis of MySQL and SQLite relational database management systems in Windows 10 environments. *International Journal of Latest Trends in Engineering and Technology*.

- Jimy Alexander Cortés Osorio, F. A. (2010). *Sistemas de seguridad basados en biometría*.
- Kilicdagi, A. (2014). *Laravel Design Patterns and Best Practices*.
- Libelium. (12 de Diciembre de 2017). *Libelium World*. Recuperado el 9 de Septiembre de 2019, de <http://www.libelium.com/early-warning-system-to-prevent-floods-and-allow-disaster-management-in-colombian-rivers>
- Libelium. (Febrero de 2017). *Smart Gases 3.0: Technical Guide*. Recuperado el 13 de Mayo de 2019, de [http://www.libelium.com/downloads/documentation/gases\\_sensor\\_board\\_3.0.pdf](http://www.libelium.com/downloads/documentation/gases_sensor_board_3.0.pdf)
- Libelium. (12 de Diciembre de 2018). *Libelium World*. Recuperado el 9 de Septiembre de 2019, de <http://www.libelium.com/balearic-islands-to-become-a-smart-tourist-destination-by-expanding-iot-sensor-networks/>
- Libelium. (5 de Septiembre de 2019). *Libelium World*. Recuperado el 9 de Septiembre de 2019, de <http://www.libelium.com/libelium-helps-to-reduce-noise-pollution-in-the-city-of-cuenca-ecuador/>
- Libelium. (Junio de 2019). *Technical Guide Waspnote*. Recuperado el 29 de Agosto de 2019, de [http://www.libelium.com/downloads/documentation/waspnote\\_technical\\_guide.pdf](http://www.libelium.com/downloads/documentation/waspnote_technical_guide.pdf)
- Margolis, M. (2011). *Arduino Cookbook*.
- María F. Maldonado, A. C. (2008). Implementación de un sistema Web para manejo de datos meteorológicos del Laboratorio de Energías Alternativas y Eficiencia Energética de la Escuela Laboratorio de Energías Alternativas y Eficiencia Energética de la Escuela Politécnica Nacional. Obtenido de <http://bibdigital.epn.edu.ec/bitstream/15000/4923/1/PAPER%20Implementaci%C3%B3n%20de%20un%20sistema%20Web%20para%20manejo%20de%20datos%20meteorol%C3%B3gicos%20del%20Laboratorio%20de%20Energ%C3%ADas%20Alterna.pdf>
- Ministerio del Ambiente. (2012). *Plan Nacional de la Calidad del Aire*. Recuperado el 23 de Enero de 2019, de <http://www.ambiente.gob.ec/wp-content/uploads/downloads/2012/10/libro-calidadaaire->
- Mishra, A. (2014). Critical Comparison Of PHP And ASP.NET For Web Development. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 3,*. Obtenido de <http://www.ijstr.org/final-print/july2014/Critical-Comparison-Of-Php-And-Aspnet-For-Web-Development.pdf>
- Molloy, D. (2016). *Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux*. Obtenido de <https://www.element14.com/community/community/raspberry-pi?src=raspberrypi>
- Monk, S. (2012). *Programming the Raspberry Pi: Getting Started with Python*.
- Mulloy, B. (2012). *Web API Design, Crafting Interfaces that Developers Love*. apigee.
- Nguyen, Q. H. (2015). Building a web application with LARAVEL 5. *Oulu University of Applied Sciences*.
- Oracle. (2011). *PHP Scalability and High Availability Database Resident Connection Pooling and Fast Application Notification*.
- OSHA. (2013). *Sobre OSHA*. Recuperado el 10 de Septiembre de 2019, de <https://www.osha.gov/aboutosha>

- OSHA. (2015). *Envenenamiento por Monóxido de Carbono*. Recuperado el 12 de Septiembre de 2019, de <https://www.osha.gov/Publications/OSHA-carbon-monoxide-spanish.html>
- OSHA. (2016). *Calidad del aire interior*. Recuperado el 10 de Septiembre de 2019, de <https://www.osha.gov/SLTC/indoorairquality/>
- Peck, A. (2017). *Jumpstarting the Raspberry Pi Zero W*.
- Philbin, C. A. (2013). *Adventures in Raspberry Pi*.
- Purdum, J. J. (2012). *Beginning C for Arduino: Learn C Programming for the Arduino*.
- RaspberryPi. (Febrero de 2016). *Raspberry Pi 3 Modelo B*. Recuperado el 1 de Septiembre de 2019, de <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- Restrepo Muñoz, V. P. (22 de Diciembre de 2010). Aplicación y comparación de la metodología de diseño Top Down y Bottom Up. Medellín, Colombia. Obtenido de [https://repository.eafit.edu.co/bitstream/handle/10784/8830/VeronicaPauline\\_RestrepoMunoz\\_2010.pdf?sequence=2&isAllowed=y](https://repository.eafit.edu.co/bitstream/handle/10784/8830/VeronicaPauline_RestrepoMunoz_2010.pdf?sequence=2&isAllowed=y)
- Scott Trent, M. T. (2008). *Performance Comparison of PHP and JSP as Server-Side Scripting Languages*. Obtenido de [https://www.researchgate.net/publication/225161349\\_Performance\\_Comparison\\_of\\_PHP\\_and\\_JSP\\_as\\_Server-Side\\_Scripting\\_Languages](https://www.researchgate.net/publication/225161349_Performance_Comparison_of_PHP_and_JSP_as_Server-Side_Scripting_Languages)
- Spectator, C. (2016). *Comparativa de rendimiento de la red entre distintos proveedores Cloud desde diferentes localizaciones geográficas*.
- UNE. (2018). *La Asociación Española de Normalización*. Recuperado el 16 de Septiembre de 2019, de <https://www.une.org/la-asociacion/historia>
- Vasquez, Leonel. (26 de Abril de 2019). Proyecto de Extensiones Universitarias. *Detección de los niveles de contaminación de aire en las salas del Hospital de niños León Becerra mediante un sistema de monitoreo de calidad del aire*. Guayaquil, Guayas, Ecuador: Universidad Politécnica Salesiana.

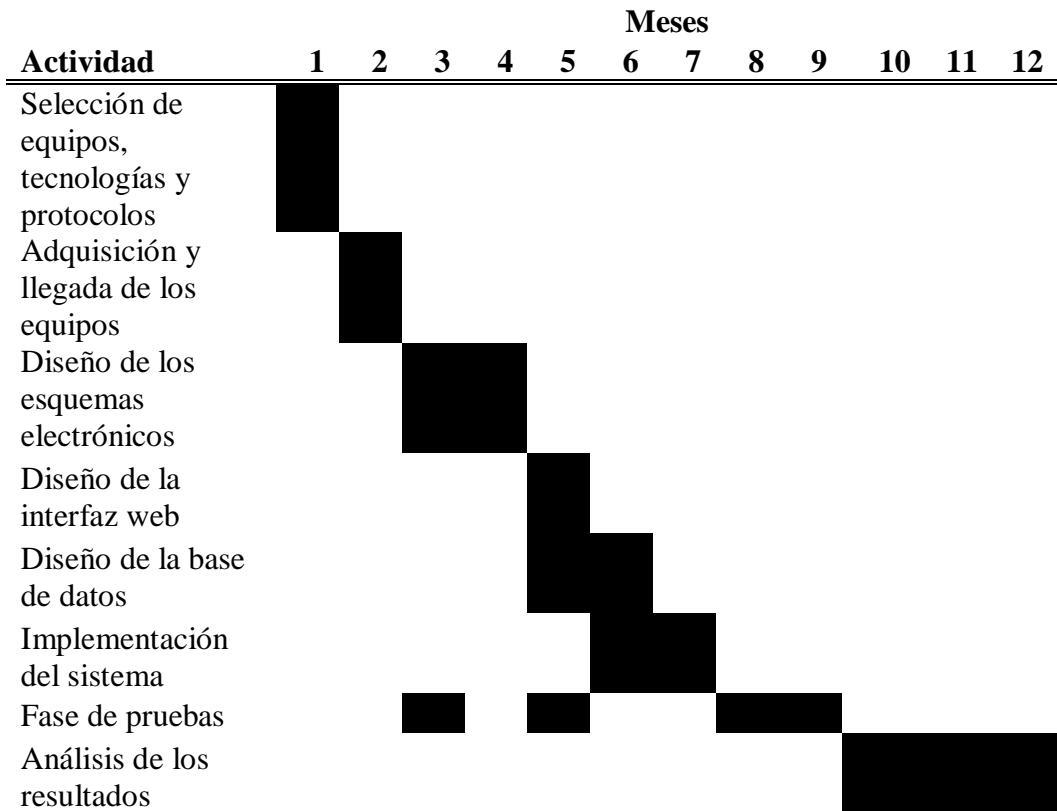
## Anexos

### Presupuesto


Tabla 23. Presupuesto para implementación.


<b>Descripción</b>	<b>Cantidad</b>	<b>Precio</b>	<b>Subtotal</b>
Tarjeta MicroSD de 8GB	2	\$ 12.00	\$ 24.00
Raspberry Pi 3	1	\$ 65.00	\$ 65.00
Waspote Starter Kit	1	\$ 450.00	\$ 450.00
Cable Mini-USB a USB	1	\$ 5.00	\$ 5.00
Estuche para guardar equipo	1	\$ 8.20	\$ 8.20
Batería portátil 5000 amperios	1	\$ 8.00	\$ 8.00
Curso de diseño web con W3CSS	1	\$ 200.00	\$ 200.00
Curso de desarrollo en NodeJS	1	\$ 250.00	\$ 250.00
<b>Total</b>			<b>\$ 1010.20</b>

## Cronograma de Actividades



# Monóxido de Carbono (Página Oficial de OSHA)


UNITED STATES DEPARTMENT OF LABOR



Occupational Safety and Health Administration
CONTACT US   FAQ   A TO Z INDEX   ENGLISH   ESPAÑOL

OSHA ▾   STANDARDS ▾   TOPICS ▾   HELP AND RESOURCES ▾

OSHA Occupational Chemical Database / Report Page Label information & descriptions

## CARBON MONOXIDE (and CO by COHb)†

Chemical Identification			
<b>CAS#</b>	630-08-0		
<b>Formula</b>	CO		
<b>Synonyms</b>	carbon oxide, flue gas, monoxide, carboxyhemoglobin/COHb		

Physical Properties			
<b>Physical description</b>	Colorless, odorless gas.		
<b>Boiling point</b>	-313°F	<b>Molecular weight</b>	28.0
<b>Freezing point/melting point</b>	-337°F	<b>Vapor pressure</b>	<35 mm
<b>Flash point</b>		<b>Vapor density</b>	0.97
<b>Specific gravity</b>	0.791 at -312.7°F	<b>Ionization potential</b>	14.01 eV
<b>Lower explosive limit (LEL)</b>	12.5%	<b>Upper explosive limit (UEL)</b>	74%
<b>NFPA health rating</b>	3	<b>NFPA fire rating</b>	4
<b>NFPA reactivity rating</b>	0	<b>NFPA special instruction</b>	

Monitoring Methods Used by OSHA			
<b>Analyte code (IMIS no.)</b>	0000	CF30 (COHb)	
<b>Sampling group</b>			
<b>Sampler/Sampling media</b>	Five layer aluminum gas sampling bag (Calibrated instruments GSS-P110)	Contact the SLTC Physical Measurements Team Leader in cases where CO poisoning affects more than 4 individuals. The most practical use of resources sometimes requires evaluating the most severely affected employees.	
<b>Sampling time<sup>a</sup></b>			
<b>Sampling volume (TWA)<sup>b</sup></b>	2-5 L		
<b>Sampling flow rate (TWA)<sup>b</sup></b>	0.01-0.05 L/min		
<b>Sampling volume (STEL/Peak/C)<sup>c</sup></b>	2-5 L		
<b>Sampling flow rate (STEL/Peak/C)<sup>c</sup></b>	0.3 L/min		
<b>Analytical method instruments</b>	GC-DID	COHb analysis is a routine measurement during the medical response to a suspected CO-poisoning incident or from an autopsy after a suspected CO-poisoning incident. As a result, the COHb analysis is performed using a blood-gas instrument that is part of a quality system. The COHb analysis is not performed at the SLTC.	
<b>Method reference</b>	OSHA ID-210 (July validated)	Carbon monoxide (CO) exposure is determined by calculation based on carboxyhemoglobin (COHb) measurements provided by medical professionals as described below. OSHA uses a modified Coburn, Forster, Kane (CFK) Calculation. The documented SLTC modifications (printed in the supplemental report that accompanies the results) use a more accurate version of the CFK equation than the closed-form [e.g., ppm CO = f(x,y,z,...)] version used in the 1972 NIOSH criteria document for the purpose of calculating the	

		<p>8-hr TWA ppm REL that would result in a 5% COHb level [U.S. Department of Health, Education, and Welfare. Criteria for a Recommended Standard - Occupational Exposure to Carbon Monoxide. Pub. No. HSM 73-11000. NIOSH, Cincinnati, OH (1972)] The calculation also provides an incident-specific sampling and analytical error (SAE) based on a Monte-Carlo approach designed to deal with the uncertainties in the data. The calculation is performed at the SLTC and the results are critically assessed for correctness by the SLTC experts prior to reporting. The SLTC is available to assist the CSHD with acquiring the data and interpreting the results.</p>	
<b>Notes</b>		<p>The exposure incident is documented on a worksheet. Personal information is used to assess the effect of any tobacco smoking behavior (before, during and after the exposure), blood volume, and activity levels (during and after the exposure). Post-exposure delay to sampling is preferable less than approx 6 hours for living persons. Longer delays may result in significantly higher SAEs or the inability to distinguish an overexposure from baseline for an individual. Contact Scott Jones at 801-233-4962 [OSHA staff only].</p>	
<b>Special requirements</b>			

\* All sampling instructions above are recommended guidelines for OSHA Compliance Safety and Health Officers (CSHOs); please see the corresponding OSHA method reference for complete details.

Wipe Method	
<b>Sampler/Sampling media</b>	

Bulk Method	

On-Site Screening Techniques	
<b>Device</b>	CMS Chip
<b>Model/Type</b>	Carbon monoxide, 5-150 ppm range
<b>Sampling information (see manufacturer instructions)</b>	

Exposure Limits							
OSHA PEL 8-hour TWA (ST) STEL (C) Ceiling Peak		NIOSH REL Up to 10-hour TWAE (ST) STEL (C) Ceiling		ACGIH TLV® 8-hour TWAE (ST) STEL (C) Ceiling		California PEL 8-hour TWA (ST) STEL (C) Ceiling Peak	
<b>PEL-TWA</b>	50 ppm (66 mg/m³)	<b>REL-TWA</b>	35 ppm (46 mg/m³)	<b>TLV-TWA</b>	25 ppm (1889)	<b>PEL-TWA</b>	25 ppm (29 mg/m³)
<b>PEL-STEL</b>		<b>REL-STEL</b>		<b>TLV-STEL</b>		<b>PEL-STEL</b>	200 ppm
<b>PEL-C</b>		<b>REL-C</b>	200 ppm (229 mg/m³)	<b>TLV-C</b>		<b>PEL-C</b>	
<b>Skin notation</b>	N	<b>Skin notation</b>	N	<b>Skin notation</b>	N	<b>Skin notation</b>	N
<b>Notes:</b>	See 29 CFR 1910.1001 Table Z-1	<b>Notes:</b>		<b>Notes:</b>	DC18	<b>Notes:</b>	
<b>Health factors:</b>	See NLM-NIH TOXNET	<b>IDLH ppm</b>	1200 ppm				
<b>Carcinogenic classifications:</b>	Not listed	<b>Notes:</b>					
<b>AHA emergency response planning guidelines - ERPG-1/ERPG-2/ERPG-3: 200 ppm/500 ppm/500 ppm</b>							

#### Additional Resources and Literature References

**NOAA: GAMED Chemicals** - Carbon monoxide

**NIOSH: Pocket Guide to Chemical Hazards** - Carbon Monoxide

#### Literature References

- ACGIH. *Documentation of the Threshold Limit Values (TLVs) and Biological Exposure Indices (BEIs)* - Carbon monoxide. See annual publication for most recent information.
- *Occupational Health Guidelines for Chemical Hazards*. DHHS/NIOSH Publication No. 81-123. (January 1981). Provides a table of contents of guidelines for many hazardous chemicals. The files provide technical chemical information, including chemical and physical properties, health effects, exposure limits, and recommendations for medical monitoring, personal protective equipment (PPE), and control procedures.
- ATSDR. *Toxicological profile for carbon monoxide*. June 2012.
- EPA. *Integrated Science Assessment for Carbon Monoxide (Final Report)*. U.S. Environmental Protection Agency, Washington, DC. EPA/600/R-09/019F. January 29, 2010.
- NIOSH. *Criteria Document for a Recommended Standard - Occupational Exposure to Carbon Monoxide*. 1972.
- Sari I, Zengin S, Ozer O, Davutoglu V, Yildirim C, Aksoy M. Chronic carbon monoxide exposure increases electrocardiographic P-wave and QT dispersion. *Inhal Toxicol*. 20(9): 679-84, July 2008.

Last Updated: 02/09/2018



UNITED STATES  
DEPARTMENT OF LABOR

Occupational Safety and Health  
Administration  
200 Constitution Ave NW  
Washington, DC 20210  
☎ 800-321-6742 (OSHA)  
TTY  
www.OSHA.gov

#### FEDERAL GOVERNMENT

White House  
Severe Storm and Flood Recovery  
Assistance  
Disaster Recovery Assistance  
DisasterAssistance.gov  
USA.gov  
No Fear Act Data  
U.S. Office of Special Counsel

#### OCCUPATIONAL SAFETY AND HEALTH

Frequently Asked Questions  
A - Z Index  
Freedom of Information Act  
Read the OSHA Newsletter  
Subscribe to the OSHA Newsletter  
OSHA Publications  
Office of Inspector General

#### ABOUT THE SITE

Freedom of Information Act  
Privacy & Security Statement  
Disclaimers  
Important Website Notices  
Plug-Ins Used by DOL  
Accessibility Statement

Para mayor información, dé clic en el siguiente enlace: <https://www.osha.gov/chemicaldata/chemResult.html?recNo=462>



# Norma Técnica de Ventilación General en Hospitales

## Ventilación general en hospitales

General ventilation in hospital  
Ventilation générale dans des hôpitaux

### Redactores:

M. Gracia Rosell Farrás  
Ingeniero Técnico Químico

CENTRO NACIONAL DE  
CONDICIONES DE TRABAJO

Adriano Muñoz Martínez  
Técnico Superior de Prevención

HOSPITAL SANT PAU. BARCELONA

*En los hospitales la ventilación tiene que cubrir las necesidades clínicas y proporcionar las condiciones higiénicas adecuadas con el fin de proteger a los pacientes y a los profesionales que realizan sus tareas en éste ámbito y a su vez, realizar el tratamiento térmico del ambiente. Desde el punto de vista de la prevención de riesgos laborales, la ventilación de los lugares de trabajo es una medida de protección colectiva que permite eliminar o reducir el contenido de agentes contaminantes que puedan estar presentes en el ambiente. En esta Nota Técnica de Prevención (NTP) se comentan los criterios de las diferentes normativas aplicables para obtener una buena calidad del aire.*

Las NTP son guías de buenas prácticas. Sus indicaciones no son obligatorias salvo que estén recogidas en una disposición normativa vigente. A efectos de valorar la pertinencia de las recomendaciones contenidas en una NTP concreta es conveniente tener en cuenta su fecha de edición.

### 1. INTRODUCCIÓN

En prevención de riesgos laborales la ventilación es una herramienta que permite mantener unas condiciones de trabajo seguras y saludables reduciendo o eliminando los contaminantes ambientales generados en el lugar de trabajo. Si además el aire de ventilación se climatiza permite trabajar en condiciones confortables. En los centros sanitarios la ventilación y el acondicionamiento del aire ha que cumplir con una serie de requisitos especiales, inherentes con las propias funciones y considerando la susceptibilidad de los pacientes. Además del mantenimiento del clima ambiental, uno de los cometidos específicos de la instalación de acondicionamiento de aire es la reducción de la concentración de agentes contaminantes, tales como microorganismos, polvo, gases narcóticos, desinfectantes, sustancias odoríferas u otras sustancias contenidas en el ambiente.

Existen distintas normas específicas aplicables al diseño, control y mantenimiento de los sistemas de aire acondicionado de los centros sanitarios, entre las que cabe destacar la norma UNE-100713:2005, sobre instalaciones de acondicionamiento de aire en hospitales, la UNE-EN ISO 14644, sobre salas limpias y locales anexos, el HVA Desing Manual for Hospital and Clinics, publicado por la American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE), los Criterios CDC, así como recomendaciones de tipo general para

la prevención de infecciones quirúrgicas, redactadas y promovidas por la autoridad laboral competente de las diferentes comunidades autónomas donde este ubicado el centro sanitario.

En la presente NTP se comentan los criterios que deben cumplir los sistemas de acondicionamiento de aire según la norma UNE-100713:2005 y otras normas relacionadas, partiendo de la base de que los conocimientos básicos sobre ventilación general por dilución y las especificaciones y clasificación de tipos de aire son conocidos. Para una mayor información sobre el tema se puede consultar la NTP nº 742 sobre ventilación general de edificios.

### 2. ASPECTOS GENERALES

Las unidades técnicas de acondicionamiento del aire deben de estar diseñadas para asegurar la circulación del caudal de aire necesario entre los diferentes locales, disponiendo de accesos fáciles, que permitan llevar a cabo las tareas de limpieza, desinfección, mantenimiento y cambio de filtros. El sentido de la circulación del aire será desde las zonas más limpias hacia las zonas más sucias, y las condiciones termo-higrométricas han de ser las adecuadas para cada lugar, y todo ello, sin que el nivel de presión sonora supere los 40dB(A) en el peor de los casos. En las tablas 1 y 2 se indican los valores

Local	UNE 100713:2005				ASRAE			
	Temperatura		Humedad relativa (HR)		Temperatura		Humedad relativa (HR)	
	Máxima	Mínima	Máxima	Mínima	Máxima	Mínima	Máxima	Mínima
En todo el centro sanitario	26 °C	24 °C	55%	45%	24 °C	21 °C	60%	30%
Quirófanos	26 °C	22 °C	55%	45%	24 °C	20 °C		

Tabla 1. Condiciones termo-higrométricas

mínimos y máximos de temperatura y humedad, según la norma UNE 100713:2005 y ASHRAE, así como los valores máximos de presión sonora según norma UNE 100713:2005.

Local	Presión sonora máxima
Sala de reanimación de quirófanos	35 dB(A)
Habitaciones (todas) cuidados intensivos	
Resto del hospital	40 dB(A)

Tabla 2. Nivel de presión sonora

### 3. CLASIFICACIÓN DE LOS SECTORES DEL HOSPITAL Y EXIGENCIAS DE CADA ZONA

Según la norma UNE 100713:2005 los locales se clasifican en dos clases:

- Locales clase I con exigencias muy elevadas de asepsia.
- Locales de clase II con las exigencias habituales.

Los primeros disponen de tres niveles de filtración del aire y los segundos de dos. Las clases de filtros requeridas en cada caso se indican en la tabla 3.

La circulación de aire entre locales de diferentes clases sólo es admisible cuando se realiza desde locales con requisitos de calidad de aire elevados con respecto a la presencia de gérmenes, hacia locales con requisitos menos exigentes. Por este motivo, han de preverse exclusas de aire en los siguientes casos:

- Entre locales de clase I y de clase II
- Entre locales de clase I y el exterior
- Entre locales de la misma clase I, cuando sea necesario (por ejemplo: entre quirófanos y zonas de cuidados intensivos)

Para que la función de estas exclusas quede asegurada es necesario que nunca puedan estar abiertas a la vez la puerta de entrada y salida.

### 4. TOMAS DE AIRE EXTERIOR Y SALIDAS DE EXPULSIÓN

Tanto por exigencias técnicas como higiénicas, la situación de las rejillas de toma y expulsión de aire son importantes para la calidad del aire interior y para evitar riesgos de contaminación hacia edificios colindantes. Las distancias recomendadas por la norma UNE-100713:2005 están descritas en la tabla 4. Son distancias mínimas

que deben ser objeto de estudio en cada caso debido a la variabilidad de las condiciones meteorológicas, a la orografía del terreno y a las estructuras de los edificios colindantes.

Fuente de contaminación	Distancia mínima en m
Lugar de circulación de vehículos	10
Cubiertas o tejados	2,5
Terreno	2,5

Tabla 4. Distancias mínimas de separación

Las distancias mínimas que debe haber entre la descarga de aire contaminado y la toma de aire dependerán, del grado de contaminación del aire descargado, de la velocidad de la descarga y del caudal del efluente. Para su cálculo se utiliza la siguiente ecuación:

$$d = 0,004 \cdot \sqrt{Q} \left( \sqrt{f} \pm \frac{V}{2} \right)$$

*d*: la distancia mínima de separación, en m

*Q*: caudal del efluente, en l/s

*f*: factor de dilución.

*V*: velocidad de la descarga, en m/s.

Según la dirección de la descarga, el signo que se toma será positivo o negativo y dependerá de si la dirección de la descarga con respecto a la toma de aire es a favor o en contra. El factor de dilución varía según la clase de efluente según se indica en la tabla 5.

### 5. QUIRÓFANOS

Dadas las altas exigencias de calidad ambiental exigidas por el tipo de trabajo que se realiza en los quirófanos, éstos reciben un tratamiento diferenciado en función de las características específicas de su utilización.

#### Clasificación

La norma UNE 100713:2005 clasifica los quirófanos como locales de clase I, lo que indica, como ya se ha comentado anteriormente, que se trata de locales con altas exigencias respecto a la presencia de gérmenes. Según la experiencia, en los quirófanos, es necesario impulsar un caudal mínimo de aire de 2400 m<sup>3</sup>/h para garantizar una concentración media de gérmenes en el aire ( $k_r$ )

Nivel de filtración (clase de local)	Clase de filtro (eficiencia)	Norma	Localización
1 (I, II)	F5 (40% < 60%)	UNE-EN 779	En la toma de aire exterior, si la longitud del conducto es >10m, sino, en la entrada de aire de la central de tratamiento o después de la eventual sección de mezcla.
2 (I, II)	F9 (>95%)	UNE-EN 779	Después de la unidad de tratamiento de aire y al comienzo del conducto de impulsión
3 (I)	H13 (99,95%)	UNE-EN 1822-1	Lo más cerca posible del local a tratar. En locales clase I, en la propia unidad terminal de impulsión de aire.

Tabla 3. Clases de filtros localización

Clase de efluente	Procedencia del aire	Valor de f
1	Espacios sin fuentes de contaminación definidas como oficinas, aulas, salas de conferencias, tiendas, habitaciones de hoteles	5
2	Espacios que pueden tener una leve contaminación, salas de fotocopiadoras, comedores, restaurantes, vestuarios, aseos de accesos restringidos cocinas de viviendas, aire procedente de locales donde se fuma ocasionalmente	10
3	Espacios con significados indicios de contaminación, aseos públicos, aseos de hospitales, ventilación general de cocinas comerciales, laboratorios y lavanderías en seco, piscinas, ventilación primaria de redes de evacuación de aguas fecales, aire de locales donde se fuma moderadamente	15
4	Espacios con gases o humos molestos o tóxicos, cabinas de pinturas, garajes, campanas de cocinas, campanas de laboratorios, almacén de productos químicos, salas de maquinaria frigorífica, salas de almacenamiento de ropa sucia, chimeneas de aparatos que queman combustibles gaseosos, locales donde se fuma mucho	25
5	Aire que contiene concentraciones elevadas de partículas, bio-aerosoles o gases peligrosos como chimeneas de calderas de combustibles sólidos o líquidos, campanas de humos sin tratamientos, torres de refrigeración y condensadores evaporativos	50

Tabla 5. Clases de efluentes

producidos exclusivamente en el propio quirófano, sin que existan grandes diferencias entre distintos puntos del mismo cuando están dotados de un sistema por difusión de mezcla de aire y un mínimo de 20 movimientos /h. A este caudal de aire se le llama caudal de referencia.

El grado de contaminación  $\mu_s$  se define por la relación entre  $\frac{K_s}{K_r}$  donde:

$k_s$  es la concentración media de gérmenes en el área de protección, considerando como tal la zona de operaciones y mesa de instrumental.

$k_r$  es la concentración media de gérmenes en el aire del quirófano con un caudal de aire impulsado igual al de referencia.

La norma UNE-EN ISO 14644-1:2000, sobre salas limpias, se ajusta a las necesidades de calidad del aire requeridas en el quirófano y en consecuencia se ha utilizado como referencia para la clasificación de los diferentes tipos de quirófanos. La clasificación se basa en las características de la intervención y en el riesgo de infección postoperatorio. De acuerdo con esta norma los quirófanos se clasifican en: quirófanos tipo A, que son los que tienen un nivel de asepsia más elevado, quirófanos tipo B que corresponden a un nivel intermedio en cuanto a la exigencia de asepsia y quirófanos tipo C cuando el

nivel de exigencia frente a la contaminación por agentes biológicos es menos exigente. En la tabla 6 se especifican las diferentes clasificaciones de los quirófanos según las distintas normas.

### Ventilación

Tomando como base la norma UNE-EN ISO 14644-1:2000 y desde un punto de vista preventivo frente a las infecciones quirúrgicas en los quirófanos, la ventilación puede ser de flujo unidireccional o turbulento y el número de aportaciones de aire exterior tiene que ser igual o superior a 20 renovaciones hora.

En los quirófanos de clase A, se admite tanto el sistema unidireccional como turbulento, aunque se recomienda el sistema unidireccional. En este caso se admite la recirculación del aire, que debe ser del mismo quirófano y tratada de igual manera que el aire exterior. Se recomienda un mínimo de 35 movimientos/h. En los quirófanos de clase B y C se admite el régimen turbulento y en caso de los quirófanos de clase B los movimientos del aire deben ser de 20 por hora, siendo el aire del 100% exterior. En los quirófanos de clase C los movimientos de aire tienen que ser iguales o superiores a 15 por hora y también con un aire 100% exterior.

Tipo quirófano	UNE 100713:2005	UNE-EN ISO 14644-1:2000	Denominación quirófano	Tipo de intervención
A	Clase I	ISO clase 5	Quirófanos de alta tecnología. Cirugía especial.	Transplantes de órganos, cirugía cardíaca, cirugía vascular, cirugía ortopédica con implantes, neurocirugía,...
B	Clase I	ISO clase 7	Quirófanos convencionales	Cirugía convencional y de urgencias. Resto de operaciones quirúrgicas.
C	Clase I	ISO clase 8	Quirófano de cirugía ambulatoria	Cirugía ambulatoria. Salas de partos.

Tabla 6. Tipos de quirófano y clasificación del mismo según norma UNE 100713:2005 y UNE-EN ISO 14644-1:2000

La norma UNE 100713:2005 indica que aunque con las tres etapas de filtración se obtiene una buena calidad del aire, se debe impulsar un caudal mínimo de aire exterior de 1200 m<sup>3</sup>/h, para mantener la concentración de los gases de anestesia y desinfectantes dentro de un nivel ambiental aceptable ( $\leq 0,4$  ppm) en locales clase I, como es el caso de los quirófanos. Esta norma indica, también, que para quirófanos con altas exigencias respecto a gérmenes, es necesario impulsar un mínimo de 2400 m<sup>3</sup>/h cuando están dotados de sistema de difusión por mezcla de aire, con un mínimo de 20 movimientos por hora.

La norma UNE-EN 13779:2008, sobre la ventilación de edificios no residenciales, indica como requisito para la climatización la recirculación del aire interior, y que el caudal de aire exterior aportado debe establecerse en función de la tasa de aportación de contaminante al ambiente de quirófano y del valor límite ambiental (ventilación por dilución).

Aunque, según las distintas normas, existen disparidad de criterios no se exige que el aire de ventilación de quirófanos sea totalmente exterior. Deberá serlo en la medida que se evite una concentración de contaminantes químicos por encima de los niveles establecidos como límites ambientales, como son los valores límite de exposición profesional que se pueden consultar en la NTP nº 606 sobre exposición laboral a gases anestésicos.

En la tabla 7 se resumen los caudales de ventilación, así como las condiciones termo-higrométricas de sobre-presión y tipos de filtro según la clase de quirófano.

## 6. EXIGENCIAS DE CLIMATIZACIÓN EN DIFERENTES ZONAS DEL HOSPITAL

En la tabla 8 se detallan diferentes tipos de locales, su clasificación y caudales de aire de ventilación indicados por la norma UNE 100713:2005 y se comparan con los indicados por la American Society of Heating and Air-Conditioning Engineers (ASHRAE). Además, se indican las referencias de otros criterios.

## 7. MANTENIMIENTO Y CONTROL DE LAS INSTALACIONES DE ACONDICIONAMIENTO DE AIRE

Para que todo funcione correctamente se debe prestar atención a la limpieza sistemática, y cuando proceda, a la desinfección de los humectadores de aire, incluyendo calderines de agua, baterías de calefacción o refrigeración y la bandeja de recogida de aguas de condensación. Para ello los usuarios llevarán a cabo una serie de controles técnicos e higiénicos que se resumen en la tabla 9.

Tipo de quirófano	Caudal mínimo de aire impulsado	Movimientos/hora (MH)	Temperatura Humedad	Presión	Filtros
Clase A	2400 m <sup>3</sup> /hora 1200 m <sup>3</sup> /hora (aire exterior)	Mínimo 30	18°C-26°C 45- 55 % de humedad	+ 20 Pa a + 25Pa	F5/F9/H14
Clase B		Mínimo 20	22°C-26°C 45- 55 % de humedad		F5/F9/H13
Clase C	Mínimo 15				

Tabla 7. Caudal de ventilación, movimientos hora del aire, temperaturas, presión y filtros indicados para quirófanos.

Controles técnicos	Al sustituir los filtros absolutos, comprobar la ausencia de fugas a través del filtro y la estanqueidad en la junta
	Verificar periódicamente el sentido del flujo del aire. Prestar especial atención a la posible reducción del caudal de aire por la variación de la pérdida de carga en los filtros. El caudal de aire siempre debe permanecer constante
Controles higiénicos	Comprobación higiénica de las instalaciones de acondicionamiento de aire especialmente en los ambientes clase I
	Contaje de partículas y medición de los microorganismos contenidos en el aire ambiente después de la sustitución de las células filtrantes del 3er nivel de filtración
	Presencia del higienista tras las reparaciones que puedan tener posibles efectos de carácter higiénico
	Todas las comprobaciones deben incluir análisis microbiológicos

Tabla 9. Tipos de control que debe llevar a cabo el usuario

Área hospitalaria	UNE 100713: 2005		ASHRAE				Otros criterios
	Clasificación local	Caudal mínimo aire exterior (m <sup>3</sup> /h.m <sup>2</sup> )	Presión	Renovaciones aire exterior/hora	Renovaciones totales/hora	Todo existido	
<b>Quirófanos y áreas críticas</b>							
Quirófanos clase A y B	I	Ver tabla 7	positiva	5	25		no
Sala de partos	I	15	positiva	5	25		UNE-EN ISO 14644-1:2000. Salas limpias
Sala de reanimación	II	15		2	6		no
Cuidados intensivos	II	30		2	6		no
Neonatos	II	15		2	6		no
Urgencias traumatología			positiva	3	15		no
Endoscopia digestiva	I	30	negativa	2	6		no
Broncoscopia	I	30	negativa	2	12	si	no
Sala de urgencias	II	30	negativa	2	12	si	
Sala de criba			negativa	2	12	si	
Sala de espera radiología			negativa	2	12	si	
Sala de procedimientos			positiva	3	15		no
<b>Hospitalización</b>							
Habitación hospitalización	II	15		2	6		
Baño habitación	II		negativa		10	si	
Habitación hematología, protección paciente	I	30	positiva	2	12		no
Habitación aislamiento	II	10	negativa	2	12	si	no
Antesala habitaciones aislamiento	II	10	negat/pos		10	si	no
Pasillos	II	15	-	2	2		no
<b>Diagnóstico por imagen, Radioterapia, Medicina nuclear</b>							
Radiología intervencionista hemodinámica			positiva	3	15		no
Radiología convencional	II	15			6		UNE-EN ISO 14644-1:1000 1 Salas limpias
Cuarto obscuro /sala revelar			negativa		10	si	no
Sala de exploración, Despacho de visita					6		
Sala de tratamiento					6		
<b>Estérilización central</b>							
Sala de equipos de esterilización	II		negativa		10	si	
Sala de limpieza-decontaminación	II		negativa	2	6	si	no
Sala estéril, Óxido de etileno	II		negativa		10	si	no
Sala lámpa	I		positiva	2	4		no
Almacén material estéril	II		positiva	2	4		no
<b>Laboratorios</b>							
Biología	II		positiva	2	6		no
Anatomía, patológica, Histología, Citología	II		negativa	2	6	si	NTP 373: La ventilación general en el laboratorio. INSHT. Norma UNE-EN 13779:2008.
Microbiología	II		negativa	2	6	si	no

Tabla 8. Comparación entre los criterios de ventilación de la norma UNE 100713:2005 y los criterios ASHRAE y referencias de otros criterios.

**BIBLIOGRAFIA**

---

- (1) UNE 100713:2005.  
**Instalaciones de acondicionamiento de aire en hospitales.**
- (2) EN-ISO 14644-1:2000.  
**Salas limpias y locales controlados.**
- (3) UNE-EN 13779:2008.  
**Ventilación de edificios no residenciales.**
- (4) ASHRAE  
**HVAC Desing Manual for Hospitals and Clinics.**  
*American Society of Heating, Refrigerating and Air-Conditioning Atlanta, GA 30329 (2003)*
- (5) INSTITUTO NACIONAL DE SEGURIDAD E HIGIENE EN EL TRABAJO  
**Límites de exposición profesional para agentes químicos en España.**  
*INSHT, 2008*
- (6) CENTERS FOR DISEASE CONTROL AND PREVENTION (CDC).  
**Guidelines for Environmental Infection Control in Health-Care Facilities. Recommendations of CDC and the Healthcare Infection Control Practices Advisory Committee (HICPAC).**  
*U.S. Department of Health and Human Services Atlanta, GA 30333(2003)*

## Implementación del Hardware del Sistema de Monitoreo



*Figura 98.* Acoplamiento de sensores con el lector de gases y la tarjeta Wasp mote.



*Figura 99.* Conexión del equipo de medición a la batería de litio.



*Figura 100.* Conexión del equipo de medición al gateway (RaspberryPi3).



*Figura 101.* Equipo de medición conectado y operando.

## Programación de la PWA

### Componente principal

El componente principal de la aplicación consiste en la declaración de la Navbar o barra superior, así como de cada uno de los tabs que mostrarán las respectivas mediciones, tal como se muestra en la *Figura 102*.

```
<template lang="pug">
  div
    Navbar(
      v-bind:tabs='tabs'
      v-bind:options='options'
      v-bind:title=`${t("appTitle")} (${this.settingsMeasurementPlace})`
      v-bind:actions='actions'
      v-on:onOptionSelected='onOptionSelected'
      v-on:onActionSelected='onActionSelected'
      v-on:onChangeTab='(newTab) => activeTab = newTab.id')
    transition(name='replace' mode="out-in")
    TabCo(v-if='activeTab === 0')
    TabTemperature(v-if='activeTab === 1')
    TabHumidity(v-if='activeTab === 2')
</template>
```

*Figura 102.* Componente principal de la aplicación.

De la misma forma cada una de las pestañas donde se muestra el contenido de las mediciones es un componente aparte, el cual se encarga de graficar las mediciones, tal como se puede apreciar en la *Figura 103*.

```
<template lang="pug">
  #tabCo
    template(v-if='isDownloadingData')
      .w3-display-container(style='height: 200px')
        .w3-display-middle
          Spinner
    template(v-else)
      .w3-row
        .w3-col.w3-center.s4
          a(v-bind:href='exportedData' download='co_exported.csv') Exportar
        .w3-col.w3-center.s4
          strong Promedio &nbsp;
          | {{average}}
        .w3-col.w3-center.s4
          strong Desviación &nbsp;
          | {{standardDeviation}}
      template(v-if="dataset['x'].length === 0")
        h3.w3-center {{t('noMeasures')}}
      template(v-else)
        template(v-if="settingsMeasurementDisplay === 'graph'")
          GoogleCharts(v-if='dataset' v-bind:datasets='[dataset]')
          br
          br
          br
          br
          br
          br
          br
          br
          br
          br
        template(v-if="settingsMeasurementDisplay === 'list'")
```

*Figura 103.* Cada pestaña de la PWA.



La aplicación cuenta con una barra superior en la cual se ve el nombre del lugar donde se están realizando las mediciones, y muestra el botón de play y pause, así como el botón de menú. Para esto se usa el código de la *Figura 104*.

```

<template lang="pug">
  div
    .ui-top.ui-color-primary.ui-card-4
      .ui-bar.ui-padding
        .ui-bar-item.ui-button(v-on:click="onBack" v-if="backButton")
          i.fas.fa-arrow-left
        .ui-bar-item {{title | cut}}
        .ui-right.ui-dropdown-menu(v-if="options.length > 0")
          button.ui-button
            i.fas.fa-ellipsis-v
          .ui-dropdown-content.ui-bar-block.ui-border(style="right: 4px")
            .ui-bar-item.ui-button(v-for="option in options" v-on:click="onOptionSelected(option)")
            .ui-right(v-if="actions.length > 0")
              button.ui-button(v-for="action in actions" v-on:click="onActionSelected(action)")
                i.fas(v-bind:class="{action.icon}")
        .ui-row
          .ui-col.ui-center.ui-padding(v-on:click="onChangeTab(tab)" v-for="tab, index in tabs" v-bind:cl

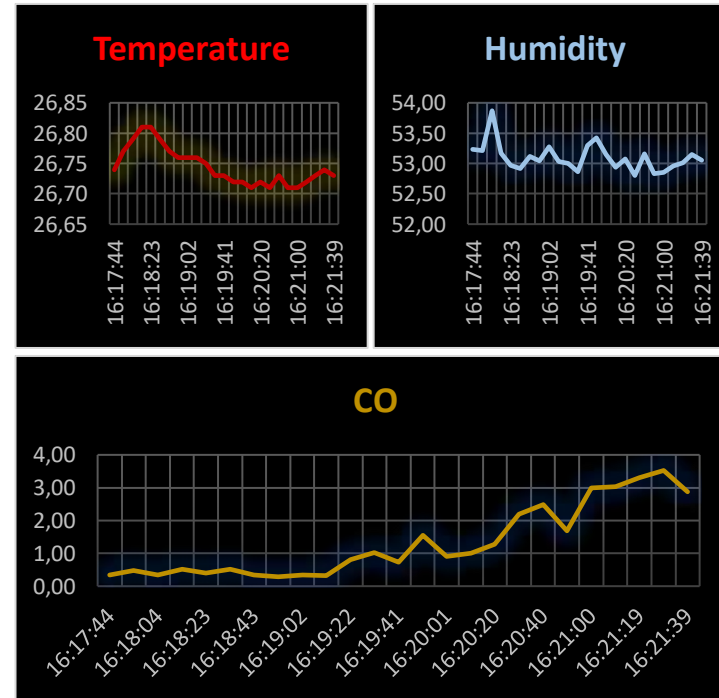
```

*Figura 104.* Navbar de la aplicación.

### Mediciones en el Hospital de Niños León Becerra

day	date	time	temperature (°C)	humidity (%RH)	co (ppm)	mediciones
Tue	19/03/2019	16:17:25	26.75	54.48	1.17	CP1
Tue	19/03/2019	16:17:34	26.75	53.66	0.35	CP1
Tue	19/03/2019	16:17:44	26.74	53.23	0.35	CP1
Tue	19/03/2019	16:17:54	26.77	53.21	0.48	CP1
Tue	19/03/2019	16:18:04	26.79	53.87	0.35	CP1
Tue	19/03/2019	16:18:13	26.81	53.17	0.51	CP1
Tue	19/03/2019	16:18:23	26.81	52.97	0.39	CP1
Tue	19/03/2019	16:18:33	26.79	52.92	0.51	CP1
Tue	19/03/2019	16:18:43	26.77	53.12	0.35	CP1
Tue	19/03/2019	16:18:52	26.76	53.04	0.29	CP1
Tue	19/03/2019	16:19:02	26.76	53.28	0.35	CP1
Tue	19/03/2019	16:19:12	26.76	53.03	0.33	CP1
Tue	19/03/2019	16:19:22	26.75	53.00	0.81	CP1
Tue	19/03/2019	16:19:32	26.73	52.86	1.03	CP1
Tue	19/03/2019	16:19:41	26.73	53.30	0.73	CP1
Tue	19/03/2019	16:19:51	26.72	53.42	1.54	CP1
Tue	19/03/2019	16:20:01	26.72	53.15	0.91	CP1
Tue	19/03/2019	16:20:11	26.71	52.94	1.00	CP1
Tue	19/03/2019	16:20:20	26.72	53.07	1.27	CP1
Tue	19/03/2019	16:20:30	26.71	52.80	2.19	CP1
Tue	19/03/2019	16:20:40	26.73	53.16	2.49	CP1
Tue	19/03/2019	16:20:50	26.71	52.83	1.69	CP1
Tue	19/03/2019	16:21:00	26.71	52.86	2.99	CP1
Tue	19/03/2019	16:21:09	26.72	52.96	3.03	CP1

	temperature (°C)	humidity (%RH)	co (ppm)
CP1	27	53	1
D. Estándar	0.03	0.22	1.10



Tue	19/03/2019	16:21:19	26.73	53.02	3.30
Tue	19/03/2019	16:21:29	26.74	53.15	3.53
Tue	19/03/2019	16:21:39	26.73	53.05	2.88
Tue	19/03/2019	16:22:47	26.74	53.20	3.38
Tue	19/03/2019	16:22:57	26.74	53.10	3.48
Tue	19/03/2019	16:23:07	26.74	52.81	4.95
Tue	19/03/2019	16:23:17	26.75	53.48	4.98
Tue	19/03/2019	16:23:26	26.75	53.36	5.18
Tue	19/03/2019	16:23:36	26.73	52.92	4.20
Tue	19/03/2019	16:23:46	26.72	52.96	3.48
Tue	19/03/2019	16:23:56	26.73	53.21	4.08
Tue	19/03/2019	16:24:06	26.73	52.77	3.70
Tue	19/03/2019	16:24:16	26.73	53.21	6.70
Tue	19/03/2019	16:24:25	26.68	53.19	5.98
Tue	19/03/2019	16:24:35	26.56	53.20	5.90
Tue	19/03/2019	16:24:45	26.58	53.35	6.85
Tue	19/03/2019	16:24:55	26.55	53.47	5.50
Tue	19/03/2019	16:25:05	26.57	53.88	7.00
Tue	19/03/2019	16:25:14	26.60	53.41	7.15
Tue	19/03/2019	16:25:24	26.64	53.40	5.50
Tue	19/03/2019	16:25:34	26.66	53.44	7.20
Tue	19/03/2019	16:25:44	26.68	53.56	5.50
Tue	19/03/2019	16:25:54	26.66	53.27	7.25
Tue	19/03/2019	16:26:04	26.66	53.23	5.68
Tue	19/03/2019	16:26:13	26.63	53.16	6.35
Tue	19/03/2019	16:26:23	26.67	53.48	7.60

CP1

CP1

CP1

CP0

CP0

CP0

CP0

CP0

CP0

CP0

CP0

CP0

CP0

CP0

CP0

CP0

CP0

CP0

CP0

CP0

CP0

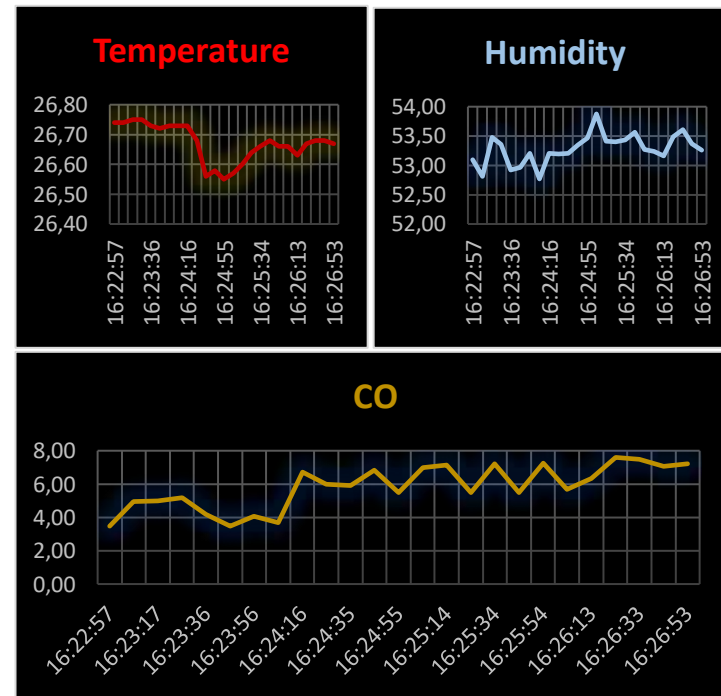
CP0

CP0

CP0

CP0

	temperature (°C)	humidity (%RH)	co (ppm)
CP0	27	53	6
D. Estándar	0.06	0.25	1.32



Tue	19/03/2019	16:26:33	26.68	53.61	7.50
Tue	19/03/2019	16:26:43	26.68	53.37	7.05
Tue	19/03/2019	16:26:53	26.67	53.26	7.20
Tue	19/03/2019	16:28:31	26.51	54.05	8.05
Tue	19/03/2019	16:28:41	26.41	55.03	10.10
Tue	19/03/2019	16:28:51	26.37	54.97	11.75
Tue	19/03/2019	16:29:01	26.33	55.08	11.60
Tue	19/03/2019	16:29:10	26.40	55.23	9.95
Tue	19/03/2019	16:29:20	26.48	55.29	9.80
Tue	19/03/2019	16:29:30	26.59	57.97	11.40
Tue	19/03/2019	16:29:40	26.63	55.57	10.40
Tue	19/03/2019	16:29:50	26.66	55.06	9.70
Tue	19/03/2019	16:30:00	26.67	54.34	8.35
Tue	19/03/2019	16:30:10	26.70	54.25	9.20
Tue	19/03/2019	16:30:19	26.69	53.89	8.55
Tue	19/03/2019	16:30:29	26.70	53.44	8.05
Tue	19/03/2019	16:30:39	26.70	53.93	8.35
Tue	19/03/2019	16:30:49	26.70	53.72	8.30
Tue	19/03/2019	16:30:59	26.71	53.67	11.10
Tue	19/03/2019	16:31:09	26.72	53.66	9.70
Tue	19/03/2019	16:31:18	26.73	53.64	10.30
Tue	19/03/2019	16:31:28	26.73	53.69	10.25
Tue	19/03/2019	16:31:38	26.73	53.51	9.50
Tue	19/03/2019	16:31:48	26.74	53.50	11.95
Tue	19/03/2019	16:31:58	26.74	53.74	9.30
Tue	19/03/2019	16:32:08	26.73	53.78	10.85

CP0

CP0

CP0

CP2

CP2

CP2

CP2

CP2

CP2

CP2

CP2

CP2

CP2

CP2

CP2

CP2

CP2

CP2

CP2

CP2

CP2

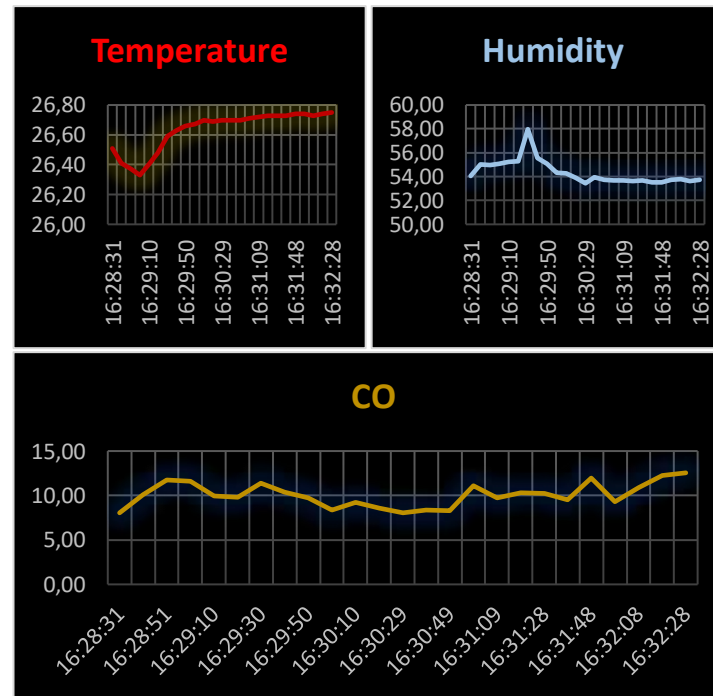
CP2

CP2

CP2

CP2

	temperature (°C)	humidity (%RH)	co (ppm)
CP2	27	54	10
D. Estándar	0.13	1.01	1.37



Tue	19/03/2019	16:32:18	26.74	53.64	12.25
Tue	19/03/2019	16:32:28	26.75	53.74	12.55
Tue	19/03/2019	16:33:27	26.77	53.44	10.80
Tue	19/03/2019	16:33:37	26.76	53.35	12.55
Tue	19/03/2019	16:33:47	26.79	53.75	10.75
Tue	19/03/2019	16:33:56	26.79	53.44	12.15
Tue	19/03/2019	16:34:06	26.80	53.24	12.40
Tue	19/03/2019	16:34:16	26.77	53.31	12.50
Tue	19/03/2019	16:34:26	26.79	53.45	10.75
Tue	19/03/2019	16:34:36	26.85	54.87	12.60
Tue	19/03/2019	16:34:46	26.84	53.89	10.85
Tue	19/03/2019	16:34:56	26.83	53.25	11.05
Tue	19/03/2019	16:35:06	26.84	53.32	10.85
Tue	19/03/2019	16:35:15	26.82	53.02	12.60
Tue	19/03/2019	16:35:25	26.81	52.85	10.75
Tue	19/03/2019	16:35:35	26.80	53.04	12.60
Tue	19/03/2019	16:35:45	26.81	52.73	10.85
Tue	19/03/2019	16:35:55	26.81	53.20	9.60
Tue	19/03/2019	16:36:05	26.82	53.24	12.60
Tue	19/03/2019	16:36:15	26.81	52.94	12.60
Tue	19/03/2019	16:36:25	26.81	52.84	12.45
Tue	19/03/2019	16:36:34	26.82	53.22	9.65
Tue	19/03/2019	16:36:44	26.80	52.71	9.55
Tue	19/03/2019	16:36:54	26.81	52.86	9.70
Tue	19/03/2019	16:37:04	26.81	53.29	9.65
Tue	19/03/2019	16:37:14	26.82	52.84	12.05

CP2

CP2

MS1

MS1

MS1

MS1

MS1

MS1

MS1

MS1

MS1

MS1

MS1

MS1

MS1

MS1

MS1

MS1

MS1

MS1

MS1

MS1

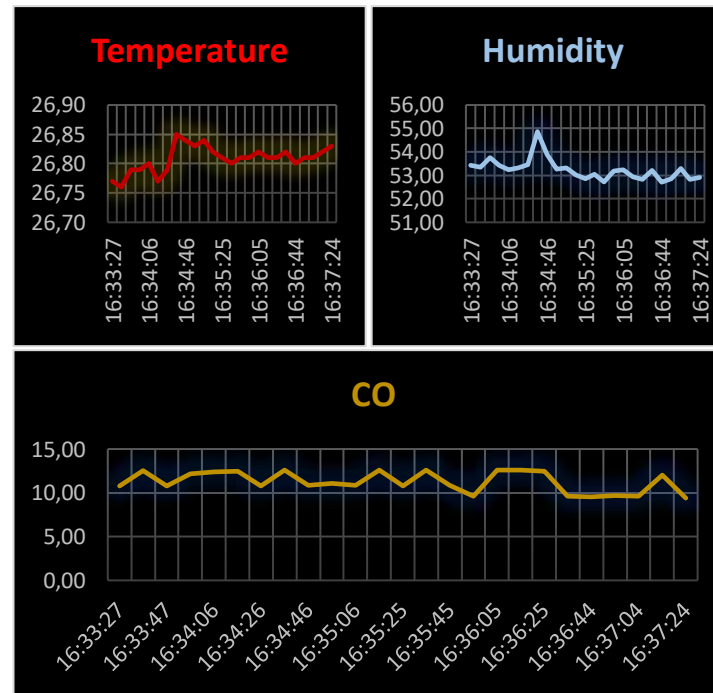
MS1

MS1

MS1

MS1

	temperature (°C)	humidity (%RH)	co (ppm)
MS1	27	53	11
D. Estándar	0.02	0.45	1.20



Tue	19/03/2019	16:37:24	26.83	52.92	9.40
Tue	19/03/2019	16:38:53	26.81	53.00	9.10
Tue	19/03/2019	16:39:03	26.82	53.00	9.20
Tue	19/03/2019	16:39:13	26.80	53.05	11.15
Tue	19/03/2019	16:39:23	26.85	53.82	9.60
Tue	19/03/2019	16:39:33	26.86	53.49	12.89
Tue	19/03/2019	16:39:42	26.87	53.33	11.00
Tue	19/03/2019	16:39:52	26.88	53.51	12.60
Tue	19/03/2019	16:40:02	26.87	53.43	11.25
Tue	19/03/2019	16:40:12	26.87	53.51	12.81
Tue	19/03/2019	16:40:22	26.83	54.52	11.35
Tue	19/03/2019	16:40:32	26.82	54.36	13.17
Tue	19/03/2019	16:40:42	26.81	54.29	13.09
Tue	19/03/2019	16:40:52	26.81	55.33	12.89
Tue	19/03/2019	16:41:02	26.81	55.53	11.30
Tue	19/03/2019	16:41:12	26.77	56.32	13.09
Tue	19/03/2019	16:41:22	26.78	56.35	11.00
Tue	19/03/2019	16:41:31	26.80	56.35	11.10
Tue	19/03/2019	16:41:41	26.81	56.30	12.90
Tue	19/03/2019	16:41:51	26.77	56.32	11.20
Tue	19/03/2019	16:42:01	26.78	56.41	12.75
Tue	19/03/2019	16:42:11	26.80	56.31	11.15
Tue	19/03/2019	16:42:21	26.79	56.53	12.60
Tue	19/03/2019	16:42:31	26.80	56.53	11.00
Tue	19/03/2019	16:42:41	26.81	56.27	11.00
Tue	19/03/2019	16:42:51	26.81	56.62	10.85

MS1

MS0

MS0

MS0

MS0

MS0

MS0

MS0

MS0

MS0

MS0

MS0

MS0

MS0

MS0

MS0

MS0

MS0

MS0

MS0

MS0

MS0

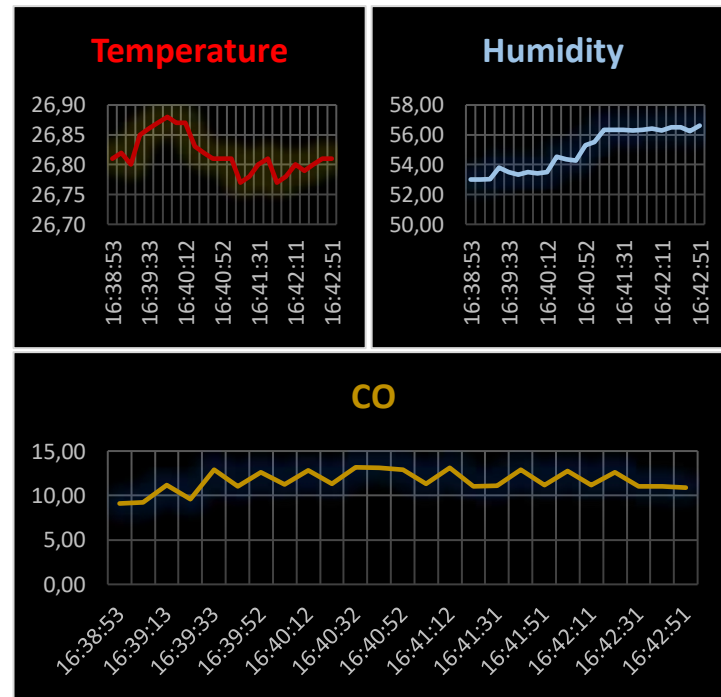
MS0

MS0

MS0

MS0

	temperature (°C)	humidity (%RH)	co (ppm)
MS0	27	55	12
D. Estándar	0.03	1.42	1.22



Tue	19/03/2019	16:45:00	26.83	59.31	10.70
Tue	19/03/2019	16:45:10	26.88	58.51	12.60
Tue	19/03/2019	16:45:20	26.91	57.98	9.90
Tue	19/03/2019	16:45:30	26.92	57.95	12.60
Tue	19/03/2019	16:45:40	26.94	57.83	9.55
Tue	19/03/2019	16:45:50	26.97	57.76	11.50
Tue	19/03/2019	16:46:00	26.98	57.82	9.75
Tue	19/03/2019	16:46:09	27.00	57.88	11.60
Tue	19/03/2019	16:46:19	27.01	57.88	11.30
Tue	19/03/2019	16:46:29	27.02	57.93	11.70
Tue	19/03/2019	16:46:39	27.03	57.93	9.30
Tue	19/03/2019	16:46:49	27.03	57.95	11.15
Tue	19/03/2019	16:46:59	27.04	58.10	11.30
Tue	19/03/2019	16:47:09	27.05	58.01	8.95
Tue	19/03/2019	16:47:19	27.06	58.35	9.55
Tue	19/03/2019	16:47:29	27.06	58.42	10.30
Tue	19/03/2019	16:47:39	27.05	58.20	10.75
Tue	19/03/2019	16:47:49	27.07	58.31	9.75
Tue	19/03/2019	16:47:59	27.09	58.44	8.95
Tue	19/03/2019	16:48:09	27.10	58.51	9.65
Tue	19/03/2019	16:48:19	27.10	58.53	9.95
Tue	19/03/2019	16:48:29	27.10	58.62	9.75
Tue	19/03/2019	16:48:39	27.11	58.65	9.00
Tue	19/03/2019	16:48:49	27.12	58.45	8.20
Tue	19/03/2019	16:48:59	27.14	58.87	8.55

MS2

MS2

MS2

MS2

MS2

MS2

MS2

MS2

MS2

MS2

MS2

MS2

MS2

MS2

MS2

MS2

MS2

MS2

MS2

MS2

MS2

MS2

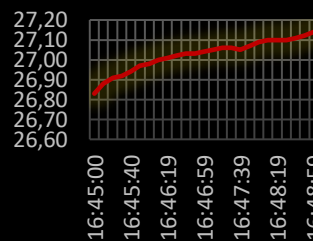
MS2

MS2

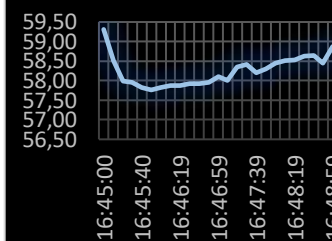
MS2

	temperature (°C)	humidity (%RH)	co (ppm)
MS2	27	58	10
D. Estándar	0.08	0.38	1.21

### Temperature



### Humidity



### CO

