



TECNOLOGICO DE ESTUDIOS SUPERIORES DE ECATEPEC

DIVISION DE INGENIERIA EN SISTEMAS COMPUTACIONALES

**NEBULIZADOR GERMICIDA AUTOMÁTICO CON APLICACIÓN WEB
DE MONITOREO**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN SISTEMAS COMPUTACIONALES**

P R E S E N T A N:

**VANESSA ABIGAIL JUAREZ RAMIREZ
LUIS ANTONIO QUESADA ALONSO**

DIRECTOR(A): DRA. GRISELDA CORTÉS BARRERA,
CODIRECTOR: M EN C EDGAR CORONA ORGANICHE,
CODIRECTOR: DR. FRANCISCO JACOB ÁVILA CAMACHO

ECATEPEC DE MORELOS, EDO. DE MÉXICO, 2022.



GOBIERNO DEL
ESTADO DE MÉXICO

DICTAMEN DE LIBERACIÓN Y CUMPLIMIENTO DE
REQUISITOS ACADÉMICOS PARA TITULACIÓN

FO-TESE-DA-78
TIT05



DIRECCIÓN ACADÉMICA
DIVISIÓN INGENIERÍA EN SISTEMAS COMPUTACIONALES

TES_PRF-20222-005
09 DE NOVIEMBRE DEL 2022

Datos generales

Nombre del egresado:

JUAREZ RAMIREZ VANESSA ABIGAIL
Apellido paterno Apellido materno Nombres (s)

Matrícula: 201710604 Teléfono: 55-5835-9794 Móvil: 55-7516-8866 Correo Electrónico: shiro.dc92@gmail.com 201710604@tese.edu.mx

Opción de titulación: **TESIS PROFESIONAL**
Únicamente en las opciones I, II, VII y X

PROYECTO: **NEBULIZADOR GERMICIDA AUTOMÁTICO CON APLICACIÓN WEB DE MONITOREO**

ASESOR (ES): **DRA. GRISELDA CORTES BARRERA, M. EN C. EDGAR CORONA ORGANICHE, DR. FRANCISCO JACOB ÁVILA CAMACHO**

Dictamen:

Con base en la solicitud de opción de titulación No. (1) correspondiente al periodo escolar (2022 - 2), presentada por el alumno de referencia en fecha **21 DE FEBRERO DEL 2022** y después de haber sido desarrolladas las actividades requeridas en dicha opción, el Consejo Académico Divisional, conformado por los integrantes que firman al calce, ha determinado que SI (X) NO () han sido cubiertos en su totalidad los requisitos que permitan a esta División Académica liberar al alumno de los trámites académicos para tal efecto e iniciar el trámite de titulación en la Unidad de Registro y Certificación de este Tecnológico de Estudios Superiores de Ecatepec.

Observaciones:

Consejo Académico Divisional

M. EN C. IRVING CARDIEL ALCOCER GUILLERMO
Presidente de la Academia de Ciencias Básicas

LIC. MARÍA REBECA GARCÍA PONCE
Presidenta de la Academia de Ciencias Sociales

M. EN D. DULCE ANAHÍ FLORES NUÑEZ
Presidenta de la Academia de Ciencias Económico y
Administrativas

M. EN C. ROSA MARÍA HERNÁNDEZ ROJAS
Presidenta de la Academia de Ciencias de la Ingeniería

I.S.C. FRANCISCO JAVIER HERNÁNDEZ BARAJAS
Presidente de la Academia de Diseño de la Ingeniería

M. EN C. XÓCHITL RAQUEL WONG COHÉN
Presidenta de la Academia de Ingeniería Aplicada

Elaboró:	Revisó y Visto Bueno:
M. EN C. MERCEDES FLORES FLORES SECRETARIO DEL CONSEJO ACADÉMICO DIVISIONAL	ING. JOSÉ LUIS CHÁVEZ ROJAS ENCARGADO DE LA DIVISIÓN ISC



GOBIERNO DEL
ESTADO DE MÉXICO

DICTAMEN DE LIBERACIÓN Y CUMPLIMIENTO DE
REQUISITOS ACADÉMICOS PARA TITULACIÓN

FO-TESE-DA-78
TÍTOS



DIRECCIÓN ACADÉMICA
DIVISIÓN INGENIERÍA EN SISTEMAS COMPUTACIONALES

TES_PRF-20222-006
09 DE NOVIEMBRE DEL 2022

Datos generales

Nombre del egresado:

Apellido paterno QUESADA Apellido materno ALONSO Nombres (s) LUIS ANTONIO

Matrícula: 201710721 Teléfono: 55-2617-6743 Móvil: 55-4477-4085 Correo Electrónico: laquesada29@gmail.com 201710721@tese.edu.mx

Opción de titulación: **TESIS PROFESIONAL**
Únicamente en las opciones I, II, VII y X

PROYECTO: **NEBULIZADOR GERMICIDA AUTOMÁTICO CON APLICACIÓN WEB DE MONITOREO**

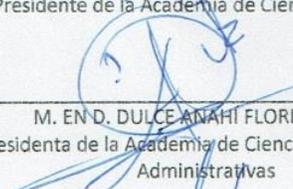
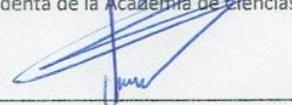
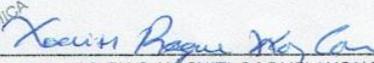
ASESOR (ES): **DRA. GRISELDA CORTES BARRERA, M. EN C. EDGAR CORONA ORGANICHE, DR. FRANCISCO JACOB ÁVILA CAMACHO**

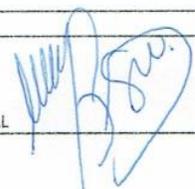
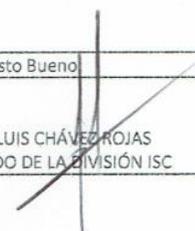
Dictamen:

Con base en la solicitud de opción de titulación No. (1) correspondiente al periodo escolar (2022 - 2), presentada por el alumno de referencia en fecha **21 DE FEBRERO DEL 2022** y después de haber sido desarrolladas las actividades requeridas en dicha opción, el Consejo Académico Divisional, conformado por los integrantes que firman al calce, ha determinado que SI (X) NO () han sido cubiertos en su totalidad los requisitos que permitan a esta División Académica liberar al alumno de los trámites académicos para tal efecto e iniciar el trámite de titulación en la Unidad de Registro y Certificación de este Tecnológico de Estudios Superiores de Ecatepec.

Observaciones:

Consejo Académico Divisional

 M. EN C. IRVING CARDIEL ALCÁZAR GUILLERMO Presidente de la Academia de Ciencias Básicas	 LIC. MARÍA REBECA GARCÍA PONCE Presidenta de la Academia de Ciencias Sociales
 M. EN D. DULCE ANAHI FLORES NUÑEZ Presidenta de la Academia de Ciencias Económico y Administrativas	 M. EN C. ROSA MARÍA HERNÁNDEZ ROJAS Presidente de la Academia de Ciencias de la Ingeniería
 I.S.C. FRANCISCO JAVIER HERNÁNDEZ BARAJAS Presidente de la Academia de Diseño de la Ingeniería	 M. EN C. XOCHITL RAQUEL WONG COHÉN Presidenta de la Academia de Ingeniería Aplicada

Elaboró:  M. EN C. MERCEDES FLORES FLORES SECRETARIO DEL CONSEJO ACADÉMICO DIVISIONAL	Revisó y Visto Bueno  ING. JOSÉ LUIS CHÁVEZ ROJAS ENCARGADO DE LA DIVISIÓN ISC
--	--

Resumen

La presente tesis muestra el desarrollo del prototipo nebulizador germicida automático promovido por el Tecnológico de Estudios Superiores de Ecatepec debido a las problemáticas que causó la pandemia del covid-19, por lo cual, nuestro objetivo es desarrollar un prototipo de un dispositivo con sistema de monitoreo capaz de realizar la nebulización de un líquido de manera automática y empleando un módulo de comunicación para el envío de información a una página web que monitoreará el rendimiento del dispositivo. Con el desarrollo de un prototipo nebulizador germicida con página web reducirá drásticamente dicha exposición al virus al mantener los espacios desinfectados. El proyecto se desarrolló en 4 etapas, en la primera etapa se planteó el análisis de los requerimientos del proyecto y se definió la arquitectura general de la solución. En la segunda etapa se realizaron los diseños requeridos para el desarrollo del prototipo y la aplicación web relacionados con el esquema del circuito, la estructura de la carcasa, la base de datos y esquemas de la página web. En la tercera etapa se realizó el desarrollo del prototipo nebulizador y la aplicación web, se crearon las pistas del circuito con una maquina CNC, se instalaron los componentes al circuito y se realizó la programación del Arduino, el servidor en Nodejs y la aplicación web de monitoreo, para finalizar esta etapa se integraron los módulos del prototipo, la aplicación web y el servidor. En la última etapa se realizaron pruebas de funcionamiento para cada uno de los componentes del prototipo y la aplicación web de monitoreo. En los resultados de las pruebas se comprobó que el prototipo puede realizar una nebulización en un lapso de 4 segundos agotándose el líquido alrededor del disparo 60 y alcanzando por cada disparo en al menos 2 segundos una distancia mínima de 8 cm y una máxima de 43 con una cobertura de 17 cm². Para probar el envío de datos se mandaron 500 mensajes de los cuales 477 fueron recibidos almacenados y mostrados correctamente en la aplicación web. En las pruebas se detectaron problemas de conectividad GPS cuando el prototipo estaba en movimiento también se comprobó que el área de detección del sensor ultrasónico era en algunas situaciones inconsistente, a su vez la efectividad de la nebulización se veía afectada enormemente por el ambiente siendo inconsistente cuando había corrientes de aire.

Aun así, el dispositivo realizaba la nebulización y lograba enviar datos consistentemente a la aplicación web quien las mostraba de manera adecuada y en tiempo real.

índice

Capítulo I Introducción	10
1.1 Antecedentes	11
1.2 Planteamiento del problema	15
1.3 Objetivo general	17
1.4 Objetivos específicos	17
1.5 Justificación	18
1.6 Alcances y limitaciones	19
1.6.1 Alcances	19
1.6.2 Limitaciones	19
1.7 Hipótesis	20
Capítulo II Marco Teórico	21
2.1 Consecuencias generadas durante la pandemia	22
2.2 Importancia de sistemas electrónicos para la desinfección de espacios y prevención de contagios	23
2.3 Conceptos básicos de electrónica para para la implementación de componentes electrónicos	24
2.3.1 Ley de ohm	25
2.3.2 Corriente continua, corriente alterna y potencia	25
2.3.3 Señales digitales y señales analógicas.....	27
2.4 Componentes eléctricos básicos utilizados en los circuitos electrónicos	27
2.4.1 Fuentes de alimentación eléctrica.....	27
2.4.2 Resistencias y potenciómetros	29
2.4.3 Capacitores o condensadores y su funcionamiento.....	29
2.4.4 Inductores	30
2.4.5 Transistores	31

2.4.6 Circuitos integrados	32
2.5 Microcontroladores.....	33
2.5.1 Sistemas electrónicos embebidos.....	33
2.5.2 Hardware Arduino, historia y características.....	34
2.6 Sensores para la detección de temperatura y distancia.....	38
2.6.1 Sensores de temperatura	38
2.6.2 Sensores ultrasónicos.....	39
2.7 Aerosoles y Nebulizadores	41
2.7.1 Nebulización ultrasónica con disco piezoeléctrico de cerámica, principio de operación y aplicaciones.....	42
2.8 Comunicación vía redes móviles GSP/GPRS	43
2.8.1 GSM (Global System for Mobile communications) y su Arquitectura.....	44
2.8.2 El módulo SIM800L.....	47
2.9 Sistema de posicionamiento GPS con GY-NEO 6M.....	49
2.9.1 Seguimientos del sistema GPS.....	50
2.9.2 Obtención de coordenadas.....	51
2.9.3 Fuentes de error del GPS	52
2.9.4 Módulo GPS GY-NEO-6M	53
2.10 Aplicaciones Web.....	54
2.10.1 Arquitecturas de las aplicaciones web	54
2.10.2 Diseño de páginas web.....	59
2.10.3 Bases de datos SQL y NoSQL	64
2.10.4 Framework de Angular para el desarrollo de aplicaciones web.....	65
2.11 Computación en la nube	70
2.11.1 Características de la computación en la nube	70
2.11.2 Modelo de servicios para la computación en la nube	71
2.11.3 Plataformas de computación en la nube.....	72
2.12 Recolección de requisitos para el prototipo Nebulizador inteligente automotriz germicida del SARS-CoV-2.....	80
2.13 Análisis de las necesidades y requerimientos para la aplicación de monitoreo	81
Capítulo III Metodología	86
3.1 Arquitectura de solución.....	87

3.1.1 Diseño general del dispositivo nebulizador	88
3.1.2 Capa de datos	95
3.1.3 Aplicación web	97
3.2 Construcción del prototipo	100
3.3 Desarrollo de la aplicación web, base de datos y funciones en la nube de Firebase	104
3.4 Pruebas de funcionamiento del dispositivo y la aplicación web.	109
Capítulo IV Análisis e interpretación de resultados	113
4.1 Prototipo final	114
4.2 Aplicación web final.....	115
4.3 Resultados de las pruebas de funcionamiento del prototipo nebulizador y la aplicación web	118
4.3.1 Resultados de las pruebas al envío de datos	126
4.3.2 Resultados de las pruebas al sensor de proximidad y temperatura	127
4.3.3 Resultados de las pruebas al consumo energético.....	129
4.3.4 Resultados de las pruebas del componente de nebulización	130
4.3.5 Resultados de las pruebas de datos de ubicación y el dispositivo receptor	131
Capítulo V Conclusiones y recomendaciones	133
Bibliografía	135

Índice de figuras y tablas

Tabla 1. Tipos de potencia	26
Tabla 2 Especificaciones Arduino Uno.....	35
Tabla 3 Especificaciones LM35.....	39
Tabla 4 Referencias HC-SR04	40
Tabla 5 Especificaciones mist maker	43
Tabla 6 Especificaciones SIM800L	48
Tabla 7 Especificaciones GPS GY-NEO 6M	53
Tabla 8 Datos recopilados del dispositivo	81

Tabla 9 Datos enviados por el dispositivo nebulizador.....	82
Tabla 10 Relación de criterios para la activación de una alerta	83
Tabla 11 Niveles de alerta.....	99
Tabla 12 muestra de 10 datos de la relación entre datos recibidos y mostrados....	119
Tabla 14 Comparación de registro de datos de ubicación entre google maps y la aplicación	125
Tabla 15 Pruebas de medición del sensor de temperaturas 1	128
Tabla 16 Pruebas de medición del sensor de temperaturas 2	128
Tabla 17 Resultados de proceso de extracción de datos.....	131
Fig. 1 Código de colores para las resistencias (Crespo, Pendino, & Reldán, 2002) .	29
Fig. 2 Arduino Uno	34
Fig. 3 LM35	38
Fig. 4 HC-SR04.....	40
Fig. 5 Temporización Sensor ultrasónico (leantec, 2021)	41
Fig. 6 Disco piezoeléctrico Mist Maker	42
Fig. 7 Esquema de arquitectura GSM (Gutiérrez , Maigualida, & Piña , 2008)	45
Fig. 8 SIM800L.....	48
Fig. 9 GPS Gy-neo 6m	53
Fig. 10 Modelo cliente/servidor.....	55
Fig. 11 Arquitectura de angular	67
Fig. 12 Arquitectura general de la solución	87
Fig. 13 Diagrama de bloques	89
Fig. 14 Diagrama de bloques	90
Fig. 15 Conexión HC-SR04.....	91
Fig. 16 Conexión LM35	92
Fig. 17 Conexión GPS GY-NEO 6M.....	93
Fig. 18 Conexión SIM800L.....	93
Fig. 19 Diagrama de circuito para el componente de nebulización	94
Fig. 20 Diseño vista de estado general	98
Fig. 21 Diseño vista de alertas	98

Fig. 22 Diseño vista Datos generales, historial y ubicación.....	100
Fig. 23 Circuito general	101
Fig. 24 PCB Arduino.....	102
Fig. 25 PCB Nebulizador.....	102
Fig. 26 Carcasa 3D	103
Fig. 27 Realtime Database de Firebase	105
Fig. 28 Cloud Functions	106
Fig. 29 Estructura de archivos.....	107
Fig. 30 Configuración de MapBox	108
Fig. 31 Prototipo de dispositivo nebulizador.....	114
Fig. 32 Prototipo de dispositivo receptor	115
Fig. 33 Vista final Indicadores	116
Fig. 34 Vista final de alertas	117
Fig. 35 Vista final datos generales, historial y mapa	117
Fig. 36 Relación de datos enviados y recibidos	119
Fig. 37 Relación entre datos recibidos y mostrados.....	120
Fig. 38 Primer prueba de alertas.....	121
Fig. 39 Segunda prueba de alertas	122
Fig. 40 Tercer prueba de alertas	123
Fig. 41 Última prueba de alertas	123
Fig. 42 Datos del historial de eventos	124
Fig. 43 Ultimas alertas registradas por la base de datos.....	124
Fig. 44 Resultados de datos enviados y recibidos	126
Fig. 45 Resultados de pruebas al sensor de proximidad.....	127
Fig. 46 Grafica de relación de temperaturas registradas a lo largo de una semana	128
Fig. 47 Grafica de comparación de consumo energético	129
Fig. 48 Contenedor de desinfectante	130

Capítulo I

Introducción

En este Capítulo se abordarán los temas asociados a la problemática causada por la pandemia del covid- 19, las medidas sanitarias para la prevención del contagio por el virus, así como la propuesta desarrollada por la división de sistemas computacionales del Tecnológico de Estudios Superiores de Ecatepec sobre un dispositivo que realice la desinfección de espacios y de manera automática.

1.1 Antecedentes

En Wuhan se produjo el primer brote del nuevo coronavirus (SARS-CoV-2) en una ciudad de la provincia de Hubei, en China. se registró una rápida propagación a escala comunitaria, regional e internacional, con un aumento de casos y muertes. El 30 de enero del 2020, el director general de la OMS declaró que el brote de SARS-CoV-2 era una emergencia de salud pública de importancia internacional de conformidad con el Reglamento Sanitario Internacional (2005). El primer caso en la Región de las Américas se confirmó en Estados Unidos el 20 de enero del 2020, y Brasil notificó el primer caso en América Latina y el Caribe el 26 de febrero del 2020. Desde entonces, la SARS-CoV-2 se ha propagado a los 54 países y territorios de la Región de las Américas. En América Latina el número de casos confirmados de SARS-CoV-2 ascendía a 450.000 y las muertes relacionadas con el mismo sumaban más de 25.000 hasta el 15 de mayo de 2020. A falta de una cura o una vacuna, los países de la región han recurrido a las cuarentenas y al distanciamiento social para reducir la tasa de infección y evitar el colapso de los sistemas sanitarios. Desde la aparición de dicho virus, las políticas públicas en relación con la ciencia, tecnología e innovación apoyaron el financiamiento de proyectos de investigación e innovación orientados a atender las necesidades generadas por la pandemia de SARS-CoV-2. A su vez, las empresas, universidades y centros de investigación reorientaron sus esfuerzos a la investigación y producción para hacer frente a la actual pandemia de SARS-CoV-2 (Álvarez Buylla, 2020)

En el trabajo escrito por Jennifer L. Cadnum. En el año 2020 titulado “Evaluación de una tecnología de desinfectante en aerosol electrostático para la descontaminación rápida de equipos portátiles y grandes áreas abiertas en la era del SARS-CoV-2”, probó la efectividad de una nueva tecnología de desinfectante en aerosol que utiliza un rociador electrostático para aplicar un desinfectante esporicida a las superficies después de una mínima limpieza previa. Se enfocaron en artículos o áreas que son difíciles de limpiar y desinfectar manualmente, incluidas sillas de ruedas, equipos portátiles y áreas de espera para pacientes. El dispositivo está diseñado para usarse con una variedad de desinfectantes. (Candum, y otros, 2020)

En el año 2020, en el trabajo titulado “Desinfectantes y antisépticos frente al coronavirus: Síntesis de evidencias y recomendaciones”, Sintetizaron la evidencia disponible sobre el uso de antisépticos y desinfectantes ante la infección de COVID-19, Se identificaron 47 documentos (sin duplicados: 36) de los que se seleccionaron 17 documentos. Entre las principales aportaciones de los documentos analizados caben destacar tres líneas temáticas: Información relacionada con la permanencia del coronavirus en las superficies, distintos antisépticos que se utilizan para la desinfección de dichas superficies y antisépticos para la limpieza de manos. (Joaquín León Molina, 2020)

En el año 2021, en el trabajo titulado “Diseño y fabricación de robot rociador desinfectante para covid-19” crearon un sistema de desinfección móvil capaz de ayudar con la tarea de desinfección dentro de un hospital, la unidad de desinfección portátil está unida a la parte superior del robot móvil. El sistema integra un mecanismo de rociadores y se utiliza para distribuir aire y mezcla de fluido desinfectante. Los componentes principales del robot móvil consisten en un motor de CC, módulo Bluetooth, Arduino, controlador de motor, bomba sumergible, rociador, batería, convertidores de CC que se utilizan. El sistema es capaz de desinfectar los suelos de los hospitales. El área de aplicación puede incluir pasillos de hospitales, tiendas médicas, quirófano, senderos para caminar, sala de médicos, centro de pruebas y sala de pacientes. El robot de desinfección móvil autónomo se está volviendo más útil en el hospital COVID-19 Ambientes. Reduce la intervención humana en la higienización. El sistema diseñado es muy compacto, tan fácilmente puede transportar este robot a cualquier lugar. La pandemia de COVID-19 presenta incluso más razones para utilizar robots móviles para una limpieza segura en zonas de cuarentena. El modelo propuesto es fabricado y probado en un entorno hospitalario. (G Sundar raju, 2021)

Todas estas tecnologías muestran que los métodos de desinfección son bastante requeridos en la actualidad por lo que el presente proyecto nace y se destina a brindar una solución en el ámbito de la desinfección de espacios, utilizando un método de nebulización eficaz ya que algunos desinfectantes mejoran su rendimiento al ser aplicados por aproximadamente 5 minutos en las superficies.

En cuanto a las aplicaciones de monitoreo en sistemas automatizados, se presenta que en la actualidad los sistemas de monitoreo han sido de gran importancia para las industrias brindando soluciones a sus necesidades para la gestión y el mantenimiento de sus sistemas, Desde el 2007 En el trabajo titulado “Sistema electrónico de monitoreo y mando vía web a través de internet”, realizado por Elizabeth Parra Bautista, Indica la necesidad de un sistema de control y gestión de procesos, cuya demanda es creciente en el mundo moderno y que requiere de circuitos que se auto gobiernen, ya que en ciertas operaciones es necesaria la supervisión y mando manual con personal cualificado. Si ese personal, gerencial, operativo o de mantenimiento, que en muchas ocasiones puede ser altamente calificado, tiene que estar físicamente en el lugar del proceso, se subutiliza su potencial intelectual y se producen costos de viáticos. Para brindar solución en lo anterior, en este mismo trabajo se planteó el diseño de un sistema web en comunicación con un microcontrolador para realizar el monitoreo para aplicaciones de robótica, Domótica y biomedicina entre otros. (Bautista., 2007)

En cuanto a las tecnologías de monitoreo y comunicación en 2015 en el trabajo titulado “Estación de monitoreo para cultivos, usando red GSM, internet y microcontroladores de 32 bits” por realizado por Juan Carlos Vargas Lombana, indico que un sistema de monitoreo para la producción agraria ayudo al agricultor a tener una nueva perspectiva del manejo del campo, la reducción de costos, optimización y aumento en la productividad, mejorar la administración de los recursos y la protección del medio ambiente, además de permitir que el agricultor tome decisiones más racionales sobre sus cultivos. Para su comunicación se desarrolló un sistema de comunicación de largo alcance mediante el uso de Celular inteligente, redes GSM e internet, con el fin de realizar una estación de monitoreo para un invernadero desde cualquier dispositivo inteligente que cuente con conexión a internet. El funcionamiento de la estación de monitoreo para cultivos sobrepaso sus expectativas iniciales ya que dio la posibilidad de realizar un control en tiempo real de diferentes cultivos de la región con el fin de aumentar y mejorar la producción de alimentos además se logró diseñar un sistema de comunicación de largo alcance mediante el uso de Celular inteligente, redes GSM

e internet, con el fin de realizar una estación de monitoreo para un invernadero desde cualquier dispositivo inteligente que cuente con conexión a internet. (Lombina, 2015)

Tres años más tarde en el trabajo titulado como “Sistema de monitoreo energético y control domótico basado en tecnología internet de las cosas” realizado por Erick Escobar Gallardo y Alex Villazón en 2018, nos muestra que los sistemas de control y monitoreo en tiempo real son importantes en múltiples áreas y aplicaciones, gracias a que otorgan la habilidad de poder monitorear la eficiencia del sistema analizado, además de poder diagnosticar errores y fallas presentes en estos. Estas fallas se ven reflejadas en las múltiples herramientas disponibles en los sistemas de monitoreo en tiempo real a través de gráficas de las variables monitoreadas. Dentro de los sistemas de control, el IoT permite la computación distribuida de los eventos generados por los sensores. En este trabajo se instaló el sistema completo e hicieron pruebas de validación de los diferentes componentes del sistema. El sistema se encuentra totalmente funcional y los datos recolectados están disponibles en el portal de ChuchusMOTE donde se puede interactuar con el sistema domótico, analizar los datos recolectados durante varios meses, y visualizar los datos en tiempo real los diferentes datos generados por distintos sensores. Para la validación del sistema, que duró 4 semanas, se verificó que los nodos electrónicos comunican correctamente los datos al servidor, son guardados en la base de datos, y que los valores calculados corresponden a las especificaciones de los dispositivos en observación. (Villazón, 2018)

Para la geolocalización del dispositivo, en el trabajo llamado como “Sistema de geolocalización de vehículos a través de la red GSM/GPRS y tecnología Arduino”, realizado por Jhon, Alejandro Castro Correa, Sergio Basilio Sepúlveda Mora, Byron Medina Delgado, Dinael Guevara Ibarra, Oriana López-Bustamante en 2019, resuelve el problema donde la implementación de dispositivos de geolocalización y sus procesos hace imprescindible conocer, en tiempo real, datos como: longitud, latitud, velocidad y curso, que son elementos necesarios para establecer la ubicación y trayectoria de un objeto; asimismo, de poderse aplicar e integrar con software que permita manipular, tratar y mostrar los datos obtenidos. El prototipo desarrollado fue

expuesto a diversas pruebas de funcionamiento, así como a un estudio en el cual se comparó la información obtenida desde el GPS con datos adquiridos a través de la función de geolocalización de HTML5 indicando que es viable un sistema de monitoreo que indique la posición en tiempo real. (Jhon Alejandro Castro Correa, 2019)

En el 2020 en el trabajo titulado como “Desarrollo y test de un software de monitoreo de mensajes de texto (SMS) para pacientes con insuficiencia cardíaca aguda descompensada” escrito por Leticia López Pedraza, João Ricardo Wagner de Moraes, Eneida Rejane Rabelo Silva se mostró que la tecnología móvil, en particular el servicio de mensajes cortos (SMS), está emergiendo como una plataforma prometedora para el manejo de enfermedades crónicas en poblaciones en vías de desarrollo puesto que tienen altas tasas de utilización entre estos grupos socioeconómicos. Los investigadores de salud pública han tratado de sacar provecho de esta modalidad de comunicación que puede cambiar el rol mediante el desarrollo y prueba de intervenciones de SMS diseñadas para proporcionar información que resulte en mejores resultados o cambios en los comportamientos de salud. En poco más de una década de investigación innovadora, se han realizado docenas de estudios y más de 20 revisiones sistemáticas y metaanálisis para explorar el potencial de los mensajes SMS para mejorar la respuesta de salud pública. Por último, indica los beneficios que tiene el uso de tecnologías de monitoreo en conjunto con mensajes SMS para la recopilación y envío de datos. (Leticia Lopez Pedraza, 2020).

1.2 Planteamiento del problema

Desde la aparición de la pandemia del covid-19 la economía mundial sufrió graves afectaciones, este virus tiene un amplio espectro de manifestaciones clínicas durante su curso y se han encontrado al menos 55 efectos a largo plazo que incluyen signos, síntomas y parámetros clínicos por lo que se buscan soluciones para emprender acciones a largo plazo referentes a protección de espacios libres del virus durante y después de la pandemia.

Una de las soluciones que ayudan a la desinfección de espacios son los nebulizadores, estos son una alternativa adecuada para ayudar a controlar la propagación de virus y desinfectar espacios o habitaciones grandes o inaccesibles, a su vez logran evitar la contaminación cruzada al no tener la necesidad de manipular las superficies para su desinfección y pueden ser implementados en sistemas automatizados para la desinfección ambiental. Sin embargo, para el desarrollo de un nebulizador que sea capaz de desinfectar de manera adecuada un espacio y eliminar el virus del SARS-CoV-2 debe considerar aspectos ambientales como el volumen del área a desinfectar, entradas y salidas de aire, etc., el desinfectante debe ser el adecuado para funcionar correctamente mediante nebulización y no debe dañar las superficies o poner en riesgo la salud de las personas , además, la nebulización requiere de un tiempo específico de aplicación y densidad de líquido por superficie para garantizar que la nube desinfectante logre eliminar los virus en el área aplicada.

Por otra parte después de la declaración oficial de la pandemia por parte de la Organización Mundial de la Salud, las políticas públicas en relación con la ciencia, tecnología e innovación apoyaron el financiamiento de proyectos de investigación e innovación orientados a atender las necesidades generadas por la pandemia de SARS-CoV-2. A su vez, las empresas, universidades y centros de investigación reorientaron sus esfuerzos a la investigación y producción para hacer frente a la actual pandemia de SARS-CoV-2.

En dichos esfuerzos se ha determinado que en la actualidad los sistemas de monitoreo remoto han sido de gran importancia brindando soluciones a sus necesidades para la gestión y el mantenimiento de los sistemas, ya que anteriormente, un sistema de control y gestión requiere de la supervisión y operación manual mediante un personal altamente calificado y que debe estar físicamente en el lugar del proceso.

Por estos motivos el Tecnológico de Estudios Superiores de Ecatepec promueve el desarrollo de soluciones frente a dichas problemáticas, una de ellas es un dispositivo nebulizador para la desinfección automática de espacios, el cual utilice una interfaz digital de monitoreo para que las personas tengan una solución tecnológica de vanguardia que ayude a mitigar los efectos causados por la aparición del virus

1.3 Objetivo general

Desarrollar un nebulizador prototipo con monitoreo web de sus parámetros de funcionamiento y ubicación con base en el uso de tecnologías de comunicación GSM, ubicación GPS, servicios en la nube. y nebulización ultrasónica para dispensar de forma automática agentes germicidas y lograr desinfectar espacios y áreas de uso por el ser humano.

1.4 Objetivos específicos

Para el desarrollo del prototipado para el dispositivo nebulizador automático germicida del SARS-CoV-2 en conjunto con la aplicación web de monitoreo se han definido los siguientes objetivos:

- I. Identificar los requerimientos y variables para el desarrollo del prototipo nebulizador automático germicida y aplicación web de monitoreo considerando la aplicación de tecnologías en la nube y representación de datos geodésicos para generar una arquitectura de solución
- II. Obtener los diseños de funcionamiento del prototipo nebulizado germicida automático, así como los modelados de su carcasa y su circuito electrónico.
- III. Identificar los métodos para el envío, recepción y almacenamiento de los datos proporcionados por el prototipo nebulizador, así como la estructura de la base de datos considerando el manejo de eventos en tiempo real.
- IV. Obtener el diseño visual y de funcionamiento de la página web para el monitoreo del dispositivo nebulizador.
- V. Construir el prototipo del dispositivo nebulizador automático siguiendo los diseños cumpliendo con los requerimientos y variables previstos.
- VI. Desarrollar el sistema web con módulo de datos geodésicos para el monitoreo del funcionamiento y rendimiento del dispositivo nebulizador en tiempo real.
- VII. Incorporar la página web de monitoreo en tiempo real al prototipo nebulizador automático.

- VIII. Comprobar el funcionamiento del prototipo nebulizador y la aplicación web de monitoreo en tiempo real con pruebas de funcionamiento por un lapso de dos semanas.

1.5 Justificación

Las problemáticas relacionadas con la pandemia del covid-19 anteriormente mencionadas indican una relación entre la exposición del virus y la movilidad de las personas. Con el desarrollo de un prototipo nebulizador germicida con página web reducirá drásticamente dicha exposición al virus al mantener los espacios desinfectados. Al automatizar el proceso de desinfección podremos reducir el riesgo de contagio por SARS-CoV-2 y el tener un dispositivo nebulizador automático que opere de manera eficiente y realice una desinfección adecuada dentro de un espacio cerrado, se podrá considerar su aplicación en otros ámbitos en donde se requiera una desinfección automática y de esta manera tener un impacto positivo con respecto a la reciente pandemia del virus SARS-CoV-2. Además, cuenta con una aplicación de monitoreo que es capaz de indicar el estado general del dispositivo como su temperatura, cantidad de germicida restante, estado de los sensores y ubicación, esto ayuda a tener un mejor conocimiento del estado general del dispositivo. Con ello se busca ser eficiente y que los datos de funcionamiento nos brinden un respaldo y una seguridad más adecuada para el funcionamiento específico del dispositivo ya que nos ayudan a tener un conocimiento sobre su eficiencia, desempeño y funcionalidad que es de vital importancia para su aplicación por empresas de transporte o sanidad pública.

Este nebulizador prototipo minimiza la exposición al virus al dar protección y seguridad al usuario que está en contacto continuo con personas o áreas infectadas con bacterias o virus. Además, la misma arquitectura permite construir un dispositivo que sea adaptable y utilizable en el interior de los hogares, oficinas, hospitales etc. y con el sistema de monitoreo satisfacer las necesidades de en empresas orientadas al transporte público o encargadas de la gestión de espacios públicos.

1.6 Alcances y limitaciones

1.6.1 Alcances

- I. Se determinan los requerimientos para el desarrollo del prototipo nebulizador automático y la página web de monitoreo en tiempo real, así como su arquitectura general.
- II. Se realizan los diseños de circuito y modelados de carcasa para la construcción del prototipo nebulizador y los diseños de vistas para la página web de monitoreo junto con el esquema para la base de datos.
- III. Se realizan la construcción del prototipo siguiendo los diseños y modelados, además se desarrolla la página web utilizando las vistas y tecnologías previstas.
- IV. Se hace uso de servicios en la nube para el almacenamiento y manejo de la información a la aplicación web.
- V. Se hace la comunicación y el envío de datos entre el prototipo nebulizador y aplicación web de monitoreo mediante una comunicación SMS y se muestran los datos en tiempo real.
- VI. Se realizan pruebas de funcionamiento para la nebulización automática, el envío y recepción de datos junto con la representación de la información en tiempo real.

1.6.2 Limitaciones

- I. Algunos servicios en la solo se utilizan en el periodo de un mes debido a los costes que requiere la adquisición del servicio, a excepción los servicios ofrecidos gratuitamente.

- II. El disco piezoeléctrico ultrasónico solo puede ser la versión de 20 mm al ser la versión más comercial en México, a su vez que otras versiones solo son obtenidas mediante importaciones y al por mayor.
- III. Las pruebas de funcionamiento se realizan solo con agua ya que no se ha definido un desinfectante específico para su utilización mediante la nebulización.
- IV. Al no contar con un desinfectante las pruebas solo muestran los aspectos de aspersión de la nube mas no su eficiencia de desinfección.
- V. El prototipo se limita a los servicios telefónicos ofrecidos por TELCEL para la comunicación vía SMS debido a la falta de compatibilidad de dispositivo de comunicación a otras compañías telefónicas.

1.7 Hipótesis

El prototipo Nebulizador Germicida Automático con aplicación web de monitoreo logra realizar la nebulización de un líquido en un espacio cerrado accionándose de manera automática, además puede enviar datos de funcionamiento y ubicación a una aplicación web que los muestra en tiempo real.

Capítulo II

Marco Teórico

En este capítulo se revisarán los aspectos teóricos que contribuyen en el desarrollo del proyecto. Se realizará una investigación de todos los conceptos necesarios para la comprensión de los aspectos relacionados con la problemática del Covid-19 y el desarrollo del prototipo nebulizador germicida del SARS-CoV-2 con aplicación web de monitoreo junto con la descripción de las actividades realizadas en su desarrollo.

2.1 Consecuencias generadas durante la pandemia

En esta etapa de la pandemia existen estudios que han analizado las secuelas a largo plazo secundarias a infección por COVID-19. Estudiar estas secuelas es fundamental para comprender la historia natural de la enfermedad, predecir el impacto que tiene más allá de la hospitalización y mortalidad, y determinar si se debe considerar rehabilitación posterior al alta. Las secuelas y complicaciones se observaron en las epidemias previas de coronavirus causante del Síndrome Respiratorio Agudo Severo (SARS) en 2002 y del Síndrome Respiratorio de Oriente Medio (MERS) desde 2012 hasta nuestros días. En algunas personas afectadas, se observó reducción de la función pulmonar, trastorno de estrés postraumático, depresión, ansiedad y una calidad de vida reducida, sugiriendo que es probable que SARS-CoV-2 pueda tener un impacto similar. Durante el proceso fisiopatológico se genera una intensa respuesta inflamatoria afectando en primer lugar al tracto respiratorio y posteriormente al sistema cardiovascular, nervioso central y periférico, musculo esquelético, además de los efectos psiquiátricos y psicológicos que puede desencadenar. El principal potencial secuela es el desarrollo de fibrosis pulmonar. El daño agudo favorece el depósito de material hialino en las membranas alveolares y en una fase posterior los pulmones presentan depósito de fibrina e infiltración de células inflamatorias y fibroblastos, para que, finalmente, el tejido se vuelva fibrótico.

En diversos estudios se han observado lesiones residuales posteriores a la resolución del cuadro agudo mediante pruebas de imagen o autopsias. Dos estudios han observado la presencia de fibrosis tras COVID-19 en alrededor del 40% de los pacientes (39% y 44%, respectivamente). En ambos estudios, los casos con fibrosis fueron los que tuvieron clínica más grave, mayor afectación pulmonar y edad avanzada. Sin embargo, es aún pronto para determinar si estos hallazgos se podrán resolver con el tiempo o si progresarán a fibrosis pulmonar crónica. En otra serie de 110 casos, se observó deterioro en las pruebas de función pulmonar, sobre todo en la capacidad de difusión del monóxido de carbono a las 4-6 semanas desde el inicio de síntomas que observaron en el 47% de los casos. En un estudio realizado en Italia, un 43% de una serie de 143 pacientes presentaban disnea residual un mes después del

alta, aunque este síntoma se consideró de carácter multifactorial (afectación respiratoria, cardiaca y ansiedad, entre otras). (Llamosas Falcón, 2020)

2.2 Importancia de sistemas electrónicos para la desinfección de espacios y prevención de contagios

A nivel mundial los países están constantemente buscando la cura para el SARS-CoV-2 y aplicando nuevas maneras de prevención para prevenir el incremento de la curva de infectados, es así que tanto el área de medicina como el área de tecnología están haciendo su mayor esfuerzo para encontrar soluciones efectivas ante esta pandemia, los campos de inteligencia artificial y big data buscan aportar tecnológicamente usando data existente de los ciudadanos y de la que se puede tener confianza en que muchos de estos servicios tendrán una finalidad con impacto ya que han tenido avances exponenciales en las últimas décadas.

Entre los países que están implementando soluciones tecnológicas para prevención de esta enfermedad se encuentra China, la cual formo una alianza con Alibaba y Tencent, para desarrollar la app: Alipay Health Code. Esta aplicación formaba parte de un servicio existente de Alibaba y era muy utilizado por sus ciudadanos. La finalidad de esta tecnología es para monitorear y clasificar diariamente a los ciudadanos según su estado de salud, para que puedan transitar libremente, en esta aplicación se presenta un código QR que muestra tres colores: verde para tránsito libre y amarillo o rojo de aviso inmediato con cuarentena por días. (Terazona & Jefferson, 2020)

En el Ambiente Urbano, se han observado robots móviles recorriendo las calles de algunas ciudades asiáticas, los cuales, según su tipo de aplicación, pueden ser robots terrestres para el recojo de residuos potencialmente infectados, así como tareas de desinfección de aceras y caminos. Adicionalmente, se usan robots aéreos, como los Drones, para la entrega de medicamentos u otros productos de uso indispensable, además de la detección temprana de pacientes con síntomas respiratorios (tos y estornudos).

En el Ambiente Hospitalario, a fin de no exponer y proteger al personal de salud, se han implementado el uso de robots y diversos equipos biomédicos, como es el caso del monitoreo de signos vitales con brazos robóticos teleguiados para medir la temperatura. Además, en procedimientos mínimamente invasivos, se está empleando un dispositivo bio-mecatrónico llamado “Cápsula Endoscópica” controlada por magnetismo, que es desechable, la cual proporciona una nueva modalidad de diagnóstico sin contacto y libre de infección. Otras aplicaciones, consisten en el uso de robots para esterilización por luz UV de las instalaciones clínicas y material médico; también son usados para poder fabricar equipos de protección personal para con tecnologías de impresión 3D (Cornejo, Vargas, & Cornejo-Aguilar, 2020).

Estas tecnologías muestran la importancia de los componentes electrónicos y digitales en el combate contra el covid-19.

2.3 Conceptos básicos de electrónica para para la implementación de componentes electrónicos

La Electrónica es el campo de la ingeniería y de la física aplicada al diseño y aplicación de dispositivos, por lo general circuitos electrónicos, cuyo funcionamiento depende del flujo de electrones para la generación, transmisión, recepción y almacenamiento de información.

Un circuito electrónico consiste en un modelo simplificado de una instalación real o de algún dispositivo eléctrico o electrónico como por ejemplo un relevador, motor, generador, fuente de alimentación, etc. La parte experimental de los circuitos electrónicos, proporcionan las herramientas matemáticas, eléctricas y electrónicas con los que se puede calcular el valor de algunas magnitudes eléctricas como voltaje, corriente, potencia, frecuencia, etc. en cualquier punto de un sistema electrónico o en el interior de un dispositivo (sensor, generador, fuente de alimentación, osciloscopio, etc.). (Pérez Tirado, 2010)

2.3.1 Ley de ohm

La ley de Ohm, postulada por el físico y matemático alemán Georg Simón Ohm, es una ley de la electricidad; en la cual se establece que la diferencia de potencial V que aparece entre los extremos de un conductor determinado es proporcional a la intensidad de la corriente I que circula por el citado conductor. Ohm completó la ley introduciendo la noción de resistencia eléctrica; que es el factor de proporcionalidad que aparece en la relación entre V e I (Prat Viñas, 1999):

$$V = I * R$$

- I. Tensión o diferencia de potencial “voltaje”: Es una magnitud física que cuantifica la diferencia de potencial eléctrico entre dos puntos; su unidad en el Sistema Internacional de Unidades es el Volt [V].
- II. La corriente eléctrica: Es el flujo de cargas eléctricas o flujo de electrones en forma ordenada a través de un conductor eléctrico. La unidad en el Sistema Internacional de Unidades es ampere [A].
- III. Resistencia Eléctrica: Es la propiedad física de un elemento o dispositivo de impedir el flujo de corriente. La unidad en el Sistema Internacional de Unidades Ohm [Ω].

2.3.2 Corriente continua, corriente alterna y potencia

La corriente continua se refiere al flujo continuo de carga eléctrica a través de un conductor entre dos puntos de distinta tensión, que no cambia de sentido y valor con el tiempo. “Puede ser almacenada”.

En cambio, con la corriente alterna cambia su polaridad cíclicamente, siendo positiva y negativa en periodos determinados de tiempo. La forma de onda depende del generador que la produce, pero siempre se encuentra una línea de cero voltios que divide la onda en dos picos simétricos. La corriente alterna no puede almacenarse, debe utilizarse en el mismo momento que se genera.

La “potencia” se define como la energía o trabajo consumido o producido en un determinado tiempo. La unidad de medida de potencia es el vatio (W). Su definición está relacionada con la tensión aplicada y la intensidad de corriente que circula por un circuito.

Cuando se trata de corriente continua (CC) la potencia eléctrica desarrollada en un cierto instante por un dispositivo de dos terminales es el producto de la diferencia de tensión entre dichos terminales y la intensidad de corriente que pasa a través del dispositivo. Por esta razón la potencia es proporcional a la corriente y a la tensión. Cuando el dispositivo es una resistencia de valor R o se puede calcular la resistencia equivalente del dispositivo.

Para calcular la potencia de corriente alterna es un poco más compleja, ya que no solo hay una potencia, sino que hay 3 diferentes. Se comporta como la potencia para corriente continua, es el “factor de potencia” o conocido como coseno de fi “ $\cos(\varnothing)$ ” como se muestra en la Tabla 1 (UNLP, 2016).

Tabla 1. Tipos de potencia

<p>Potencia Activa</p> <p>Es la única de las 3 potencia que se transforma en energía útil, es decir es la potencia útil del receptor, que se transforma en calor o trabajo.</p>	$P = V \times I \times \cos(\varnothing)$	<p>se mide en W (vatios)</p>
<p>Potencia Reactiva</p> <p>Esta potencia es una potencia consumida por las bobinas y por los capacitores, ejemplo, motores, transformadores, etc. No genera trabajo ni calor.</p>	$Q = V \times I \times \text{seno}(\varphi)$	<p>se mide en VAR (voltio amperios reactivos)</p>
<p>Potencia Aparente</p> <p>Es la suma vectorial de las otras dos, es la potencia útil más la potencia perdida</p>	$S = V \times I \text{ s}$	<p>se mide en volt Amper (VA)</p>

2.3.3 Señales digitales y señales analógicas

Las señales analógicas son variables eléctricas que evolucionan en el tiempo en forma análoga a alguna variable física. Estas variables pueden presentarse en la forma de una corriente, una tensión o una carga eléctrica. Varían en forma continua entre un límite inferior y un límite superior. Cuando estos límites coinciden con los límites que admite un determinado dispositivo, se dice que la señal está normalizada. La ventaja de trabajar con señales normalizadas es que se aprovecha mejor la relación señal/ruido del dispositivo.

Las señales digitales son variables eléctricas con dos niveles bien diferenciados que se alternan en el tiempo transmitiendo información según un código previamente acordado. Cada nivel eléctrico representa uno de dos símbolos: 0 o 1, verdadero o falso, etc. Los niveles específicos dependen del tipo de dispositivos utilizado. Por ejemplo, si se emplean componentes de la familia lógica TTL (transistor-transistor-logic) los niveles son 0 V y 5 V, aunque cualquier valor por debajo de 0,8 V es correctamente interpretado como un 0 y cualquier valor por encima de 2 V es interpretado como un 1 (los niveles de salida están por debajo de 0,4 V y por encima de 2,4 V respectivamente). En el caso de la familia CMOS (complementary metal-oxide-semiconductor), los valores dependen de la alimentación. Para alimentación de +5 V, los valores ideales son también 0 V y 5 V, pero se reconoce un 0 hasta 2,25 V y un 1 a partir de 2,75 V. (Miyara, 2004)

2.4 Componentes eléctricos básicos utilizados en los circuitos electrónicos

2.4.1 Fuentes de alimentación eléctrica

Sabemos que es posible generar energía a través de una reacción química; entre estas energías, la que nos interesa es la Energía Eléctrica. Los sistemas capaces de almacenar Energía Eléctrica en forma de energía química son denominadas Celdas Electroquímicas. La estructura fundamental de una Celda Electroquímica consiste en

dos electrodos, metálicos en muchos casos, introducidos en una disolución conductora o electrolito: Electrodo Positivo, denominado Cátodo, y Electrodo Negativo, denominado Ánodo. Ejemplos reales de este tipo de celdas o baterías son:

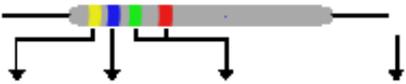
- I. Baterías de plomo que utilizamos en nuestros automóviles para el funcionamiento del arranque, iluminación y de los componentes eléctricos de un vehículo normal, a gasolina, diesel o gas.
- II. Pilas secas de zinc en sus diferentes variantes doble AA, triple AAA, etc., para el funcionamiento de varios equipos portátiles del hogar como ser radios, linternas, mandos a distancia, juguetes, etc.
- III. Baterías de litio

Las baterías de litio se caracterizan por cargarse más rápido, durar más, contar con una mayor vida útil y ofrecer más densidad energética, por lo que en menos espacio se puede obtener mayor autonomía

El funcionamiento de las baterías de ion-litio recargables o secundarias se basan en procesos denominados de inserción- desinserción de iones Litio (Li^+). Las reacciones son en estado sólido entre dos compuestos de inserción como electrodos. Uno de los compuestos es denominado Huésped (M); de naturaleza iónica, reacciona ocupando lugares vacantes (^a) en la estructura de otra especie denominada Anfitrión (A). Estas reacciones pueden ocurrir en forma reversible En las baterías primarias de Litio, el electrodo Negativo está constituido por el litio metálico y el electrodo Positivo, por un compuesto que actúa como anfitrión durante la reacción de inserción. El electrolito puede ser una disolución de una sal de litio en un solvente no acuoso. Durante el proceso de carga, se produce la reacción contraria: en el electrodo Positivo, la oxidación de la especie anfitrión y, en el electrodo Negativo, se deposita el Litio metálico (Hamel Fonseca, 2011).

2.4.2 Resistencias y potenciómetros

La resistencia es un componente pasivo, es decir, no genera corriente ni voltaje en un circuito. Su comportamiento se rige por la Ley de Ohm. Su valor se conoce por el código de colores de la Fig. 1 (también puede ir impreso en el cuerpo de la resistencia directamente). Una vez fabricadas su valor es fijo, aunque puede tener una pequeña tolerancia.



Color	1ra	2da.	Multiplicador	Tolerancia
Plata	-		$10^{-2} = 0.01$	+/- 10 %
Oro	-		$10^{-1} = 0.1$	+/- 5 %
Negro	0	0	$10^0 = 1$	-
Marrón	1	1	$10^1 = 10$	+/- 1 %
Rojo	2	2	$10^2 = 100$	+/- 2 %
Naranja	3	3	$10^3 = 1000$	-
Amarillo	4	4	$10^4 = 10000$	-
Verde	5	5	$10^5 = 100000$	+/- 0'5 %
Azul	6	6	$10^6 = 1000000$	+/- 0'25 %
Violeta	7	7	-	+/- 0'1 %
Gris	8	8	-	+/- 0'05 %
Blanco	9	9	-	-

Fig. 1 Código de colores para las resistencias (Crespo, Pendino, & Reldán, 2002)

Los potenciómetros son componentes pasivos de tres terminales, que permiten manipular la corriente que circula o la señal que hay en un circuito, por ejemplo, el volumen de un equipo de música. Normalmente el terminal central corresponde al cursor o parte móvil del componente y entre los extremos se encuentra la resistencia (De la Peña Valencia & Castillo Rodríguez, 2009).

2.4.3 Capacitores o condensadores y su funcionamiento

El capacitor o condensador es un dispositivo que posee una capacidad y sirve para almacenar cargas eléctricas. Está compuesto por dos armaduras entre las que se

encuentra un material dieléctrico no conductor como: aire, papel, vidrio, cerámica, mica, etc. Al someter una diferencia de potencial determinada, se cargan. La relación que adquieren las armaduras y la diferencia de tensión aplicada se llama “capacidad” o “capacitancia” y se mide en faradios (F), que está definida por la siguiente ecuación:

$$C = \frac{Q}{V}$$

Donde C es la capacitancia en Faradios, V el voltaje en volts y Q como constante en Coulombs. Cuando se aplica a un condensador una diferencia de tensión de CD se cargan sus armaduras hasta alcanzar la misma tensión que se les aplica. En ese punto dejan de pasar electrones desde una armadura a la otra, por eso se dice que bloquea o no deja pasar la corriente continua. En realidad, permite el paso durante un instante hasta que la carga del condensador contrarresta la diferencia de tensión. El tiempo que tarda en cargarse un condensador al conectarle una tensión de corriente continua es proporcional al producto de su capacidad por la resistencia del circuito (De la Peña Valencia & Castillo Rodríguez, 2009).

2.4.4 Inductores

El inductor es una Espiral o hélice de alambre de cobre arrollado sobre un núcleo ferromagnético o sobre un soporte de material aislante. Su propiedad más importante es la de comportarse como un imán cuando circula por ella una corriente eléctrica. Frente a la corriente continua la bobina se comporta como un hilo conductor de muy baja resistencia eléctrica, ya que en la corriente continua no se producen efectos de autoinducción. Con CD la bobina funciona al revés que el condensador, ya que la deja pasar en lugar de bloquearla. Por el contrario, si se aplica una corriente alterna varía el magnetismo que genera y corta a la misma bobina produciéndose el fenómeno de autoinducción (Magnitud física con símbolo L, que indica la aptitud de un circuito eléctrico para crear un campo magnético), la cual presenta una cierta resistencia al

paso de la corriente alterna y que recibe el nombre de reactancia inductiva, X_L , cuyo valor se puede calcular con la siguiente fórmula: $X_L = 2\pi fL$. En consecuencia, una bobina presenta mayor resistencia cuanto mayor sea la frecuencia, inversamente al condensador (De la Peña Valencia & Castillo Rodríguez, 2009).

2.4.5 Transistores

Un transistor es un dispositivo electrónico que posee tres bornes (Base, Emisor y Colector) cuya principal característica es que permite controlar la corriente que se establece en el circuito Colector-Emisor mediante la corriente que circula por el circuito de Base. Dependiendo de la aplicación, la corriente de base será en general decenas o cientos de veces más chica que la de colector. El transistor puede trabajar en tres regiones o zonas diferentes. Su comportamiento, dependerá de la zona en la cual se encuentre operando. Se distinguen tres zonas posibles de trabajo que identifican las distintas maneras de actuar del transistor (Crespo, Pendino, & Reldán, 2002):

- *Zona de Corte:* La tensión base-emisor V_{BE} es menor a 0,7 V, lo que implica que la corriente de base I_B es cero y por lo tanto la corriente por el colector I_C también será cero (o bien será extremadamente pequeña). La tensión colector-emisor V_{CE} dependerá de la configuración del circuito. El transistor se comporta como una llave abierta.
- *Zona de Saturación:* La tensión base-emisor V_{BE} es 0,7 V, por lo que circula una corriente de base I_B , lo que implica una circulación de corriente I_C por el colector. La relación entre la corriente de base y la de colector es $I_B \approx I_C/10$. La tensión colector-emisor V_{CE} será aproximadamente cero. El transistor se comporta como una llave cerrada.
- *Zona Lineal:* La relación entre la corriente de base y la corriente de colector es igual a una constante llamada β , siendo la de colector la mayor de las dos. El transistor es utilizado como amplificador y en el circuito de colector la corriente sigue las variaciones de la corriente de base.

2.4.6 Circuitos integrados

Un circuito integrado (CI), también conocido como chip, microchip, es una estructura de pequeñas dimensiones de material semiconductor, normalmente silicio, de algunos milímetros cuadrados de superficie (área), sobre la que se fabrican circuitos electrónicos generalmente mediante fotolitografía y que está protegida dentro de un encapsulado de plástico o de cerámica. El encapsulado posee conductores metálicos apropiados para hacer conexión entre el Circuito Integrado y un circuito impreso. Los CI se hicieron posibles gracias a descubrimientos experimentales que mostraban que artefactos semiconductores podían realizar las funciones de los tubos de vacío, así como a los avances científicos de la fabricación de semiconductores a mediados del siglo XX. La integración de grandes cantidades de pequeños transistores dentro de un pequeño espacio fue un gran avance en la elaboración manual de circuitos utilizando componentes electrónicos discretos. La capacidad de producción masiva de los circuitos integrados, así como la fiabilidad y acercamiento a la construcción de un diagrama a bloques en circuitos, aseguraba la rápida adopción de los circuitos integrados estandarizados en lugar de diseños utilizando transistores discretos. Los circuitos integrados son usados en prácticamente todos los equipos electrónicos hoy en día, y han revolucionado el mundo de la electrónica. Computadoras, teléfonos móviles, y otros dispositivos electrónicos que son parte indispensables de las sociedades modernas, son posibles gracias a los bajos costos de los circuitos integrados. (De la Peña Valencia & Castillo Rodríguez, 2009)

2.4.6.1 Circuito integrado LM555

El temporizador 555 es un circuito integrado muy versátil que tiene un gran número de aplicaciones en los circuitos electrónicos, sobre todo para generar intervalos de tiempo. También se utiliza para construir temporizadores, generadores de impulsos, multivibradores, alarmas, etc... Este temporizador fue introducido en el mercado mundial por la SIGNETICS (filial de PHILLIPS) en 1972. La gran aceptación que tuvo impulsó a numerosas industrias de semiconductores a fabricarlo (bajo licencia de la

SIGNETICS). Conviene señalar que el 555 es una aplicación de los AOP's, ya que su circuito interno consta de dos comparadores. El circuito integrado temporizador 555, puede funcionar lo mismo como multivibrador astable (oscilación libre) que como multivibrador monoestable (one shot) (Carrera, 2019).

2.5 Microcontroladores

Un microcontrolador es un computador completo de limitadas prestaciones, que está contenido en un chip de un circuito integrado y se destina a gobernar una sola tarea. Para que el microcontrolador realice las operaciones que deseamos es necesario grabar en su memoria de programa un conjunto de instrucciones que constituyen el programa de aplicación. En primer lugar, debemos conocer perfectamente las especificaciones de la tarea que debe ejecutar. Hay que tener en cuenta que podemos llegar al mismo resultado con distintos programas, siendo óptimo aquél que esté mejor estructurado y no realice operaciones innecesarias, ahorrando de esta forma espacio en la memoria de programa y tiempo (Apaza Condori, 2010).

2.5.1 Sistemas electrónicos embebidos

Un sistema embebido (SE) o sistema empotrado se define como un sistema electrónico diseñado específicamente para realizar unas determinadas funciones, habitualmente formando parte de un sistema de mayor entidad. La característica principal es que emplea para ello uno o varios procesadores digitales (CPUs) en formato microprocesador, microcontrolador o DSP lo que le permite aportar 'inteligencia' al sistema anfitrión al que ayuda a gobernar y del que forma parte. En el diseño de un sistema embebido se suelen implicar ingenieros y técnicos especializados tanto en el diseño electrónico hardware como el diseño del software. A su vez también se requerirá la colaboración de los especialistas en el segmento de usuarios de tales dispositivos, si hubiese lugar a ello. (Ubeda, 2009)

2.5.2 Hardware Arduino, historia y características

Arduino se refiere a tres cosas: Una placa hardware libre que incorpora un microcontrolador reprogramable y una serie de pines-hembra (los cuales están unidos internamente a las patillas de E/S del microcontrolador) que permiten conectar de forma muy sencilla y cómoda diferentes sensores y actuadores. La “placa hardware” se refiere en concreto a una PCB (del inglés “printed circuit board”, o bien, placa de circuito impreso). Las PCBs son superficies fabricadas de un material no conductor (normalmente resinas de fibra de vidrio reforzada, cerámica o plástico) sobre las cuales aparecen laminadas (“pegadas”) pistas de material conductor (normalmente cobre). No obstante, cuando se refiere a “placa Arduino”, se debe especificar el modelo concreto, ya que existen varias placas Arduino oficiales, cada una con diferentes características (como el tamaño físico, el número de pines-hembra ofrecidos, el modelo de microcontrolador incorporado y como consecuencia, entre otras cosas, la cantidad de memoria utilizable, etc.). El diseño hardware de la placa Arduino está inspirado originalmente en el de otra placa de hardware libre preexistente, la placa Wiring Esta placa surgió en 2003 como proyecto personal de Hernando Barragán, estudiante por aquel entonces del Instituto de Diseño de Ivrea (lugar donde surgió en 2005 precisamente la placa Arduino).

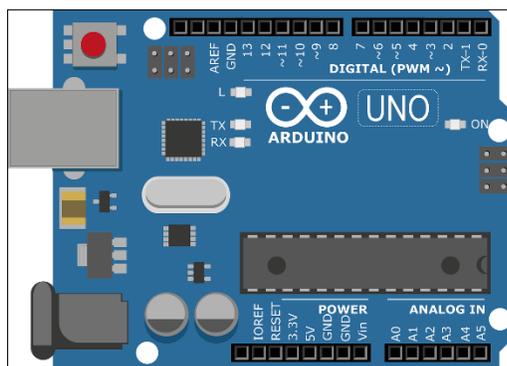


Fig. 2 Arduino Uno

Arduino Fig. 2 y Tabla 2 nació en el año 2005 en el Instituto de Diseño Interactivo de Ivrea (Italia), centro académico donde los estudiantes se dedicaban a experimentar con la interacción entre humanos y diferentes dispositivos (muchos de ellos basados en microcontroladores) para conseguir generar espacios únicos, especialmente artísticos. Arduino apareció por la necesidad de contar con un dispositivo para utilizar en las aulas que fuera de bajo coste, que funcionase bajo cualquier sistema operativo y que contase con documentación adaptada a gente que quisiera empezar de cero. La idea original fue fabricar la placa para uso interno de la escuela.

Tabla 2 Especificaciones Arduino Uno

Microcontrolador	ATmega 328
Voltaje de operación (Nivel lógico)	5v
Voltaje de entrada (Recomendado)	7-12V
Pines de entrada y salida digitales	16
Pines de entrada analógica	8
Memoria flash	16kb de los cuales 0.6kb usados por el Bootloader
SRAM	1kb
EEPROM	512bytes
Velocidad de reloj	16hz

Entre sus características contamos con:

Arduino tiene una gran comunidad: muchas personas lo utilizan, enriquecen la documentación y comparten continuamente sus ideas.

Su entorno de programación es multiplataforma: se puede instalar y ejecutar en sistemas Windows, Mac OS X y Linux. Esto no ocurre con el software de muchas otras placas.

Su entorno y el lenguaje de programación son simples y claros: son muy fáciles de aprender y de utilizar, a la vez que flexibles y completos para que los usuarios avanzados puedan aprovechar y expresar todas las posibilidades del hardware. Además, están bien documentados, con ejemplos detallados y gran cantidad de proyectos publicados en diferentes formatos (Torrente Artero, 2013).

2.5.2.1 El modelo del microcontrolador ATmega 328

El microcontrolador que lleva la placa Arduino UNO es el modelo ATmega328P de la marca Atmel. La “P” del final significa que este chip incorpora la tecnología “Picopower” (propietaria de Atmel), la cual permite un consumo eléctrico sensiblemente menor comparándolo con el modelo equivalente sin “Picopower”, el Atmega328 (sin la “P”). De todas formas, aunque el ATmega328P pueda trabajar a un voltaje menor y consumir menos corriente que el Atmega328 (especialmente en los modos de hibernación), ambos modelos son funcionalmente idénticos. Al igual que ocurre con el resto de los microcontroladores usados en otras placas Arduino, el ATmega328P tiene una arquitectura de tipo AVR, arquitectura desarrollada por Atmel y en cierta medida “competencia” de otras arquitecturas como por ejemplo la PIC del fabricante Microchip. Más concretamente, el ATmega328P pertenece a la subfamilia de microcontroladores “megaAVR”. Otras subfamilias de la arquitectura AVR son la “tinyAVR” (cuyos microcontroladores son más limitados y se identifican con el nombre de ATtiny) y la “XMEGA” (cuyos microcontroladores son más capaces y se identifican con el nombre de ATxmega), pero no las estudiaremos ya que las placas Arduino no incorporan microcontroladores de esas familias (Torrente Artero, 2013).

2.5.2.2 Las memorias del microcontrolador SRAM, Flash y EEPROM

Otra cosa que hay que saber de los microcontroladores son los tipos y cantidades de memoria que alojan en su interior. En el caso del ATmega328P tenemos:

Memoria Flash: memoria persistente donde se almacena permanentemente el programa que ejecuta el microcontrolador (hasta una nueva reescritura si se da el caso). En el caso del ATmega328P tiene una capacidad de 32KB.

Memoria SRAM: memoria volátil donde se alojan los datos que en ese instante el programa (grabado separadamente en la memoria Flash) necesita crear o manipular para su correcto funcionamiento. Estos datos suelen tener un contenido variable a lo largo del tiempo de ejecución del programa y cada uno es de un tipo concreto (es decir,

un dato puede contener un valor numérico entero, otro un número decimal, otro un valor de tipo carácter, también pueden ser cadenas de texto fijas u otros tipos de datos más especiales). Independientemente del tipo de dato, su valor siempre será eliminado cuando se deje de alimentar eléctricamente al microcontrolador. En el caso del ATmega328P esta memoria tiene una capacidad de 2KB.

Memoria EEPROM: memoria persistente donde se almacenan datos que se desea que permanezcan grabados una vez apagado el microcontrolador para poderlos usar posteriormente en siguientes reinicios. En el caso del ATmega328P esta memoria tiene una capacidad de 1 KB, por lo que se puede entender como una tabla de 1024 posiciones de un byte cada una (Torrente Artero, 2013).

2.5.2.3 El gestor de arranque del microcontrolador

La memoria Flash del microcontrolador incluido en las placas Arduino viene pregrabado de fábrica un pequeño programa llamado “bootloader” o “gestor de arranque”, que resulta imprescindible para un cómodo y fácil manejo de la placa en cuestión. Este software (también llamado “firmware”, porque es un tipo de software que raramente se modifica) ocupa, en la placa Arduino UNO, 512 bytes de espacio en un apartado especial de la memoria Flash, el llamado “bootloader block”, pero en otros modelos de placas Arduino puede ocupar más (por ejemplo, en el modelo Leonardo ocupa 4 Kilobytes). La función de este firmware es gestionar de forma automática el proceso de grabación en la memoria Flash del programa que queremos que el microcontrolador ejecute. Lógicamente, el bootloader realizará esta grabación más allá del “bootloader block” para no sobrescribirse a sí mismo. Más concretamente, el bootloader se encarga de recibir nuestro programa de parte del entorno de desarrollo Arduino (normalmente mediante una transmisión realizada a través de conexión USB desde el computador donde se está ejecutando dicho entorno hasta la placa) para proceder seguidamente a su correcto almacenamiento en la memoria Flash, todo ello de forma automática y sin que nos tengamos que preocupar de las interioridades electrónicas del proceso. Una vez realizado el proceso de grabación, el bootloader termina su ejecución y el microcontrolador se dispone a procesar de inmediato y de

forma permanente (mientras esté encendido) las instrucciones recientemente grabadas (Torrente Artero, 2013).

2.6 Sensores para la detección de temperatura y distancia

2.6.1 Sensores de temperatura

En la actualidad hay muchas formas de medir la temperatura con todo tipo de sensores de diversas naturalezas. La ingeniería de control de procesos ha inventado, perfeccionado e innovado a la hora de disponer de sensores que les ayuden a controlar los cambios de temperatura en procesos industriales por ejemplo las termocuplas. Las termocuplas son los sensores de temperatura eléctricos más utilizados en la industria. Una termocupla se hace con dos alambres de distinto material unidos en un extremo, al aplicar temperatura en la unión de los metales se genera un voltaje muy pequeño, del orden de los milivolts el cual aumenta con la temperatura. (De la Peña Valencia & Castillo Rodríguez, 2009)

2.6.1.1 Sensor LM35 y principio de operación

El sensor de temperatura LM35 es un sensor de circuito integrado con una salida lineal proporcional a la escala Celsius. Tiene un alcance de $-55\text{ }^{\circ}\text{C}$ a $+150\text{ }^{\circ}\text{C}$ y una exactitud aproximada de $0.25\text{ }^{\circ}\text{C}$. En la Fig. 3 y Tabla 3 se muestran sus conexiones, su carcasa y características. (Asini, 2021)

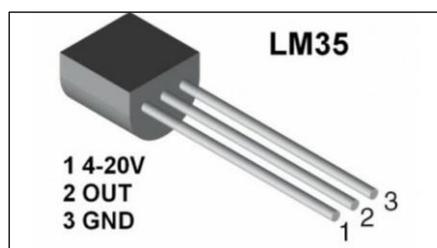


Fig. 3 LM35

Tabla 3 Especificaciones LM35

Detalles	Valores
Voltaje de trabajo	4v
Corriente de trabajo	97.5mA
Rango de precisión de sensibilidad	+/-1.0°C
Rango de temperatura	0°-150°C
Número de pines	3

El ajuste de salida de voltaje del LM35 es proporcional al de su entrada, y por cada valor de temperatura este aumenta o disminuye 10 mV. El LM35 puede funcionar con una fuente de alimentación continua de 5 V hasta 20 V, y la corriente de reserva es inferior a 60 uA.

Dependiendo del tipo de encapsulado y presentación puede variar en su escala de temperatura, como se aprecia a continuación:

- LM35-LM35A: Lectura de temperatura entre -55 a +150°C.
- LM35C-LM35CA: Lectura de temperatura entre -40 a +110°C.
- LM35D: Lectura de temperatura entre 0 a 100°C.

2.6.2 Sensores ultrasónicos

El sonido viaja en forma de ondas acústicas que golpean el aire, haciendo que se comprima y expanda millones de veces, las ondas se expanden debido a que el medio es elástico y los sonidos que se escuchan son en su mayoría una mezcla de dichas ondas. Por encima de la frontera de 20 kHz, el sonido se transforma en ultrasonido y llega a extenderse hasta altísimas frecuencias, las ondas ultrasónicas exhiben una característica que las hace extremadamente útiles, cuando viajan por un medio,

cualquiera que sea, son reflejadas si existe una discontinuidad o alguna sustancia extraña en su trayectoria, estos ecos son recogidos, analizados y la causa descubierta.

Los ultrasonidos son simples vibraciones mecánicas de frecuencia superior a los 20 kHz, se producen en forma natural o artificial y no son perceptibles por el oído humano, el cual percibe frecuencias entre 10 kHz y 18 kHz aproximadamente. El estudio y aplicación de estas vibraciones recibe el nombre de Ultrasónica. Los sensores ultrasónicos utilizan este principio para detectar ondas acústicas. (Armijos Llerena, 2000)

2.6.2.1 Sensor HC-SR04 y su principio de operación

El sensor HC-SR04 Fig. 4 y Tabla 4 este compuesto por un emisor y un receptor de ultrasonidos. Estos nos ayudan a medir la distancia a la que se encuentra un objeto justo frente a él, enviando un pulso de ultrasonidos y midiendo el tiempo que transcurre hasta que vuelve dicho pulso.

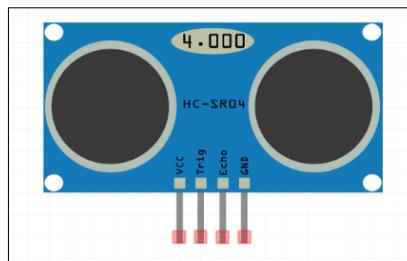


Fig. 4 HC-SR04

Tabla 4 Referencias HC-SR04

Detalles	Valores
Voltaje de trabajo	5v
Corriente de trabajo	15mA
Frecuencia de trabajo	40khz
Rango de funcionamiento	2 a 500 cm
Angulo de detección	15 a 20 grados

En el diagrama de temporización Fig. 4 se aprecia como solo es necesario aplicar un pulso de 10uS en el pin trigger para comenzar con la medición. A continuación, el sensor envía una serie de 8 pulsos de 40KHz y pone el pin de Echo a nivel alto. El pin Echo permanecerá a nivel alto hasta que se reciba el eco de los pulsos de 40KHz. Para saber a la distancia a la que se encuentra el objeto, solo hay que medir el tiempo al que está el pin Echo a nivel alto y aplicar la siguiente formula. (leantec, 2021)

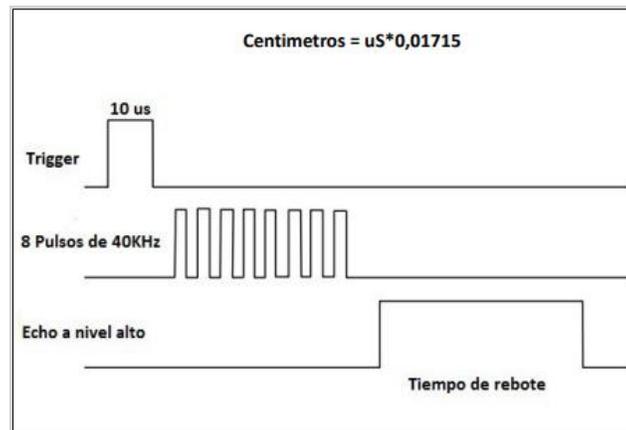


Fig. 5 Temporización Sensor ultrasónico (leantec, 2021)

2.7 Aerosoles y Nebulizadores

En la práctica diaria el término “aerosol” denota el uso de un nebulizador. Un aerosol es una suspensión de líquido o de partículas sólidas en un gas transportador y no necesariamente un atomizador líquido solamente. Hay tres mecanismos habitualmente citados por los cuales se deposita una partícula en aerosol: impactación inercial, caída gravitacional (sedimentación) y difusión. La impactación inercial ocurre con las partículas más grandes y veloces. La caída gravitacional es una función del tamaño particular y del tiempo, con una tasa de asentamiento proporcional al tamaño de la partícula. La Difusión ocurre con las partículas más pequeñas que 1 μm . Varios fabricantes han desarrollado dispositivos de aerosol que utilizan un acoplamiento o una placa con múltiples aberturas para producir un aerosol líquido. Este principio de funcionamiento utiliza una placa de abertura unida a un material piezoeléctrico que

vibra a alta frecuencia. Esta vibración a alta velocidad crea una acción de bombeo para producir el aerosol desde la solución líquida. Alternativamente, la solución puede ser forzada a través del acoplamiento para crear aerosol. Estos dispositivos pueden generar aerosoles con una fracción de partícula altamente fina, que puede resultar en una entrega más eficiente del líquido comparada con los nebulizadores convencionales. El aerosol es generado como una niebla fina, y no se requiere ningún sistema de baffle o distorsionador interno. Tienen un alto índice de producción de aerosol, son portátiles y funcionan con batería propia. (Hess, Myers, & Rau, 2005)

2.7.1 Nebulización ultrasónica con disco piezoeléctrico de cerámica, principio de operación y aplicaciones

El disco de malla piezoeléctrico mostrado en la Fig. 6 y Tabla 5 que está hecho de una placa de acero perforada. La capa exterior de la placa de acero es una placa vibratoria de cerámica. La hoja atomizadora micro-porosa se usa ampliamente en pequeños humidificadores. El agua es succionada por la varilla de succión de la esponja y rociada a través del orificio fino en el medio de la hoja atomizadora. La frecuencia y el voltaje de trabajo de la hoja atomizadora son relativamente pequeños y no necesita trabajar en agua. Su trabajo de chorro se expulsa a través de los micro agujeros en el medio (Hernández Menéndez, Alaffita Hernández, & Olivares Galván, 2018).



Fig. 6 Disco piezoeléctrico Mist Maker

El disco de malla piezoeléctrica utiliza el principio de ondas ultrasónicas para convertir el agua en pequeñas gotas de agua, generando así niebla y logrando la atomización.

Tabla 5 Especificaciones mist maker

Diámetro	20mm
Tensión de funcionamiento	3,7-12V
Frecuencia	113kHz
Potencia nominal	2,5W

Se suele utilizar en:

- Humidificación en alimentos refrigerados y plantas de aire acondicionado.
- Equipos de inhalación y desinfección
- Humidificación en el control de procesos industriales
- para lubricación, revestimiento, etc.

2.8 Comunicación vía redes móviles GSP/GPRS

Se dice que una comunicación es móvil cuando existe al menos un terminal cuya ubicación se desplaza, requiriéndose servicio durante ese desplazamiento. Las comunicaciones móviles se pueden clasificar como sigue (Asini, 2021):

- Sistemas vía satélite (INMARSAT, IRIDIUM).
- Redes de área extensa de transmisión de datos (WATM).
- Redes móviles privadas (Wireless Ethernet).
- Redes de telefonía celular públicos (GSM, GPRS, UMTS).
- Redes de telefonía sin hilos (DECT).
- Redes domésticas (Home RF).
- Redes de área personal (Bluetooth).
- Clasificación de redes inalámbricas.
- Redes globales: IRIDIUM, GLOBASTAR, UMTS.

- Redes de área extensa: WATM, GSM, UMTS

2.8.1 GSM (Global System for Mobile communications) y su Arquitectura

GSM es el estándar en telecomunicaciones móviles más extendido en el mundo, con un 82% de los terminales mundiales en uso. GSM cuenta con más de 3000 millones de usuarios en 159 países distintos, siendo el estándar predominante en Europa, América del Sur, Asia y Oceanía, y con gran extensión en América del Norte. Las siglas GSM se corresponden al nombre en inglés del Sistema Global de Comunicaciones Móviles. Se trata de un estándar muy utilizado desde principios de siglo y también se conoce como 2G debido a que supuso un salto de las comunicaciones analógicas a las digitales.

La banda de frecuencia en la que opera el GSM difiere según el territorio. En Europa se utiliza el espectro radioeléctrico de 900 y 1800 MHz, mientras que en Estados Unidos la banda es la de 1900. Esto hace que no todos los móviles GSM puedan funcionar en todo el mundo, a no ser que su tecnología esté preparada para conectarse a todas las bandas. (Asini, 2021)

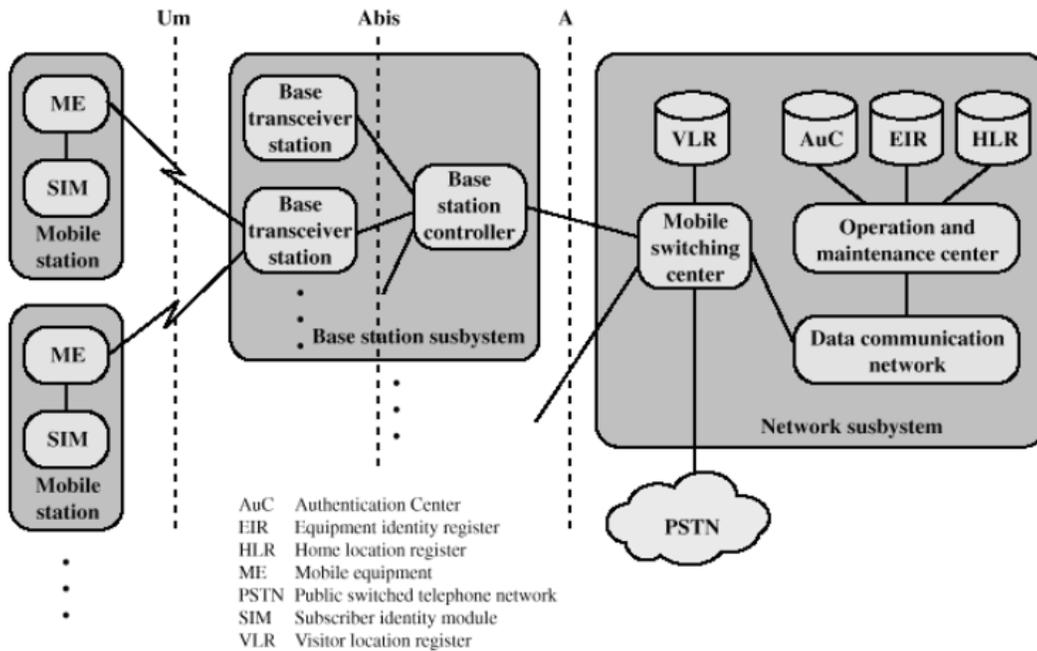


Fig. 7 Esquema de arquitectura GSM (Gutiérrez , Maigualida, & Piña , 2008)

La red GSM está dividida en tres grandes subsistemas:

1. Sistema de Conmutación (Switching System, SS)
2. Sistema de Estación Base (Base Station System, BSS)
3. Sistema de Operación y Soporte (Operation and Support System, OSS)

Una red GSM se encuentra formada por los siguientes componentes que integran la red pública móvil terrestre (PLMN-Public Land Mobile Network):

1. La estación móvil (MS:Mobile Station). Es el punto de entrada a la red móvil inalámbrica. Es el equipo físico usado por el usuario GSM para acceder a los servicios proporcionados por la red.
2. El módulo de identidad del abonado (SIM:Subscriber Identity Module). GSM distingue entre la identidad del abonado y la del equipo móvil. El SIM está asociado con el abonado, se trata de un chip que el usuario debe introducir en el terminal GSM.

3. La estación transmisora-receptora de base o estación transreptora de base (BTS-Base Transceiver Station). Se encarga de proporcionar, vía radio, la conectividad entre la red y las estaciones móviles.
4. El controlador de estaciones base (BSC-Base Station Controller). Se encarga de todas las funciones centrales y de control del subsistema de estaciones base (BSS:Base Station Subsystem) que está constituido por el BSC y las BTSs.
5. La unidad de Transcodificación (TRAU-Transcoding Rate and Adaptation Unit). Se encarga de comprimir la información en el interfaz aéreo cuando se hace necesario. La TRAU forma parte del subsistema BSS. Permite que tasas de datos GSM (8,16,32 Kbps) puedan ser enviadas hacia la interfaz RDSI del MSC que sólo acepta tasas de 64 Kbps.
6. El centro de conmutación de servicios móviles o centro de conmutación de móviles (MSC-Mobile Services Switching Center). Se encarga de enrutar el tráfico de llamadas entrantes y salientes, y de la asignación de canales de usuario en la interfaz entre el MSC y las BSC.
7. El registro general de abonados (HLR-Home Location Register). Es una base de datos que contiene y administra la información de los abonados, mantiene y actualiza la posición del móvil y la información de su perfil de servicio.
8. El registro de abonados itinerantes (VLR-Visitor Location Register). Diseñado para NO sobrecargar el HLR. Guarda localmente la misma información que el HLR, cuando el abonado se encuentra en modo de itinerancia (roaming).
9. El centro de autenticación (AuCAuthentication Center). Genera y almacena información relativa a la seguridad, genera las claves usadas para autenticación y encriptación.
- 10.Registro de Identidad de Equipos (EIR: Equipment Identity Register). Los terminales móviles tienen un identificador único, el IMEI (International Mobile Equipment Identity), el EIR se utiliza para mantener una relación de las

identidades de los equipos abonados; a través de él resulta posible identificar aquellos usuarios autorizados.

2.8.1.1 SMS (Short Message Service)

Significa "Servicio de mensajes cortos". SMS se utiliza para enviar mensajes de texto a teléfonos móviles. Los mensajes generalmente pueden tener hasta 160 caracteres de longitud, aunque algunos servicios usan el modo 5-bit, que admite caracteres 224. SMS fue creado originalmente para teléfonos que usan comunicación GSM (Sistema Global para Móvil), pero ahora todos los principales sistemas de telefonía celular lo admiten (Nicola, 2016).

2.8.1.2 Los problemas del GSM

GSM es un sistema de conmutación de circuitos: Necesita de mantener una conexión abierta durante todo el tiempo de vida de una transferencia de información (voz o datos). Para el flujo constante (voz), se reservan tantos recursos como el necesario. No se desperdician recursos. Implica que para flujo variable (datos, www), desperdicio ancho de banda. En aplicaciones como WWW he de realizar establecimiento de llamada por link a acceder.

Para el tráfico de datos es mejor la conmutación de paquetes: Cada paquete de datos se transmite de manera independiente. No se reservan recursos para sesiones de datos. Todas las fuentes son multiplexadas por la red. No hay establecimiento de llamada, siempre se está conectado a la red y no hay retardos iniciales. (Nicola, 2016)

2.8.2 El módulo SIM800L

Es un dispositivo quad-band GSM/GPRS Fig. 8 y Tabla 6 que trabaja en las frecuencias GSM850MHz, EGSM900MHz, DCS1800MHz y PCS1900MHz. Este módulo de telefonía celular que permite utilizar las tecnologías de voz, texto, datos y

SMS. Esta versión cuenta con un conector uFL. Utiliza el mismo chip SIM800L que el módulo FONA de Adafruit, por lo que se pueden utilizar las mismas librerías (SIMcom, 2013).

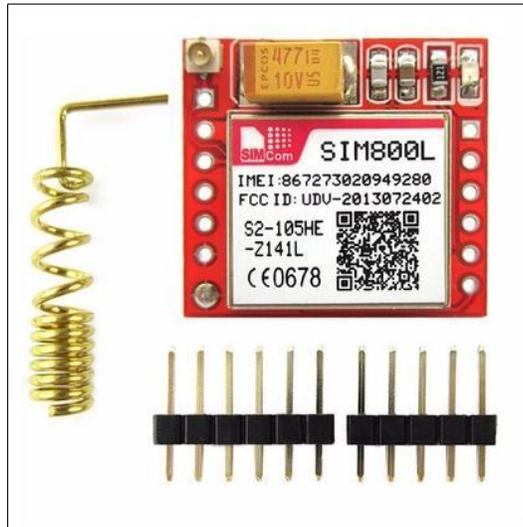


Fig. 8 SIM800L

Tabla 6 Especificaciones SIM800L

Voltaje de operación	3.4-4.4v
Voltaje Nivel lógico	3-5v
Consumo de corriente	500mA
Interfaz	Serial UART
Tamaño de tarjeta SIM	Micro SIM
Quad-band 850/900/1800/1900MHz	
Envío y recepción de mensajes SMS	
Envío y recepción de datos GPRS	
Controlador por Comandos AT	
Velocidades de transmisión serial desde 1200bps hasta 115200bps	

Por sí solo, este módulo no puede hacer nada. Se requiere un microcontrolador con una UART puede enviar y recibir comandos a través de los pines RX/TX. También es necesario un chip SIM 2G.

2.9 Sistema de posicionamiento GPS con GY-NEO 6M

El sistema de posicionamiento global mediante satélites (GPS: Global Positioning System) supone uno de los más importantes avances tecnológicos de las últimas décadas. Diseñado inicialmente como herramienta militar para la estimación precisa de posición, velocidad y tiempo, se ha utilizado también en múltiples aplicaciones civiles. Por razones de seguridad, las señales GPS generadas para uso civil se someten a una degradación deliberada, al tiempo que su emisión se restringe a una determinada frecuencia. A pesar de ello, las aplicaciones civiles siguen proliferando a un ritmo exponencial gracias a la incorporación de las técnicas diferenciales (DGPS). En este artículo se introducen los principios que rigen el funcionamiento del sistema GPS, los principales errores que afectan a la localización y las técnicas de corrección de estos. Se presenta un conjunto de aplicación actuales y futuras con gran repercusión económica y social para finalizar con comentarios sobre la posible evolución de los sistemas de posicionamiento global (Huerta, Manguiterra, & Noguerra, 2005).

El sistema GPS tiene por objetivo calcular la posición de un punto cualquiera en un espacio de coordenadas (x,y,z) , partiendo del cálculo de las distancias del punto a un mínimo de tres satélites cuya localización es conocida. La distancia entre el usuario (receptor GPS) y un satélite se mide multiplicando el tiempo de vuelo de la señal emitida desde el satélite por su velocidad de propagación. Para medir el tiempo de vuelo de la señal de radio es necesario que los relojes de los satélites y de los receptores estén sincronizados, pues deben generar simultáneamente el mismo código. Ahora bien, mientras los relojes de los satélites son muy precisos los de los receptores son osciladores de cuarzo de bajo coste y por tanto imprecisos. Las distancias con errores debidos al sincronismo se denominan pseudodistancias. La desviación en los relojes de los receptores añade una incógnita más que hace necesario un mínimo de cuatro satélites para estimar correctamente las posiciones (Pozo Ruz, y otros, Madrid).

2.9.1 Seguimientos del sistema GPS

El sistema GPS Está constituido por tres segmentos fundamentales:

- I. Espacial
- II. De control
- III. Del usuario

El seguimiento espacial hace referencia al sistema de satélites lanzados alrededor del globo los cuales se identifican por su órbita y utilizan relojes atómicos para su operación y seguimiento (Huerta, Manguiterra, & Noguerra, 2005).

Para las funciones principales del segmento de control, denominado internacionalmente con las siglas OCS (Operational Control Segment) son:

- I. Monitoreo y control permanente de los satélites con el objeto de determinar y predecir las órbitas y los relojes de a bordo
- II. Sincronización de los relojes de los satélites con el tiempo GPS
- III. Transmisión, a cada satélite, de la información procesada

Por último, el seguimiento del usuario está constituido por los instrumentos utilizados para deceptonar y procesar la señal emitida por los satélites. Estos instrumentos están integrados esencialmente por una antena y un receptor. Un equipo complementario es usado, en ocasiones, para transferir datos entre receptores.

La antena está conectada por cable al receptor o en otros casos forman una sola unidad. Las coordenadas que se calculan corresponden al centro radioeléctrico de la antena. El receptor consta de un mínimo de 4 canales (generalmente 10 o 12) que permiten recibir y procesar simultáneamente la señal de cada satélite.

Un microprocesador interno con el software correspondiente calcula las coordenadas de la antena y la velocidad y acimut si el aparato está en movimiento. Posee además una memoria para almacenar observaciones. La capacidad de esta memoria varía de acuerdo con el tipo de receptor, pudiendo llegar a almacenar información durante

varias decenas de horas. Todo equipo adiciona una unidad de alimentación eléctrica que deberá brindar al receptor la autonomía necesaria. Los equipos están en continuo desarrollo y su evolución es comparable a la experimentada en informática durante las últimas décadas para los ordenadores personales (Huerta, Manguiterra, & Noguerra, 2005).

2.9.2 Obtención de coordenadas

Para determinar la posición de un punto en el espacio, es suficiente conocer las distancias a tres puntos de coordenadas conocidas. Se trata de una intersección espacial inversa. Es un problema geométrico relativamente simple, más allá de las dificultades que su cálculo suponga. Se trata, en definitiva, de una pirámide de base triangular.

Para el sistema GPS un solo satélite transmite una señal de microondas que es captada por un dispositivo GPS y se utiliza para calcular la distancia desde el dispositivo GPS al satélite. Dado que un dispositivo GPS solo proporciona información sobre la distancia desde un satélite, un solo satélite no puede proporcionar mucha información sobre la ubicación. Los satélites no proporcionan información sobre los ángulos, por lo que la ubicación de un dispositivo GPS podría estar en cualquier punto de la superficie de una esfera.

Cuando un satélite envía una señal, crea un círculo con un radio medido desde el dispositivo GPS al satélite. Cuando agregamos un segundo satélite, crea un segundo círculo, y la ubicación se reduce a uno de los dos puntos donde los círculos convergen. Con un tercer satélite, la ubicación del dispositivo se puede determinar definitivamente, ya que el dispositivo está en la intersección de los tres círculos (Huerta, Manguiterra, & Noguerra, 2005)

2.9.3 Fuentes de error del GPS

A continuación, se describen las fuentes de error que en la actualidad afectan de forma significativa a las medidas realizadas con el GPS (Pozo Ruz, y otros, Madrid):

- I. Perturbación ionosférica. La ionosfera está formada por una capa de partículas cargadas eléctricamente que modifican la velocidad de las señales de radio que la atraviesan.
- II. Fenómenos meteorológicos. En la troposfera, cuna de los fenómenos meteorológicos, el vapor de agua afecta a las señales electromagnéticas disminuyendo su velocidad. Los errores generados son similares en magnitud a los causados por la ionosfera, pero su corrección es prácticamente imposible.
- III. Imprecisión en los relojes. Los relojes atómicos de los satélites presentan ligeras desviaciones a pesar de su cuidadoso ajuste y control; lo mismo sucede con los relojes de los receptores.
- IV. Interferencias eléctricas imprevistas. Las interferencias eléctricas pueden ocasionar correlaciones erróneas de los códigos pseudo-aleatorios o un redondeo inadecuado en el cálculo de una órbita. Si el error es grande resulta fácil detectarlo, pero no sucede lo mismo cuando las desviaciones son pequeñas y causan errores de hasta un metro.
- V. Error multi-sonda. Las señales transmitidas desde los satélites pueden sufrir reflexiones antes de alcanzar el receptor. Los receptores modernos emplean técnicas avanzadas de proceso de señal y antenas de diseño especial para minimizar este error, que resulta muy difícil de modelar al ser dependiente del entorno donde se ubique la antena GPS.
- VI. Interferencia "Disponibilidad Selectiva S/A". Constituye la mayor fuente de error y es introducida deliberadamente por el estamento militar.
- VII. Topología receptor-satélites. Los receptores deben considerar la geometría receptor-satélites visibles utilizada en el cálculo de distancias, ya que una determinada configuración espacial puede aumentar o disminuir la precisión de las medidas. Los receptores más avanzados utilizan un factor multiplicativo que modifica el error de medición de la distancia (dilución de la precisión geométrica)

2.9.4 Módulo GPS GY-NEO-6M

El módulo receptor GPS NEO-6M que se muestra en la Fig. 9 y con las características de la Tabla 7 utiliza la comunicación USART para comunicarse con el microcontrolador o terminal de PC. El módulo recibe información como la latitud, longitud, altitud, hora UTC, etc. de los satélites en forma de cadena NMEA (u-blox, 2021).



Fig. 9 GPS Gy-neo 6m

Tabla 7 Especificaciones GPS GY-NEO 6M

Voltaje de alimentación	3.3-5V
Comunicación	UART
Precisión horizontal GPS	2.6m
Precisión de velocidad	0.1m/s

El módulo GPS GY-NEO al igual que los sensores puede operar con un voltaje de entrada de 5 V como se ve en la Tabla 7 y a su vez tiene salidas RX y TX que estarán conectadas en los pines 2 y 3 de la placa de Arduino. Estas salidas se encargaran de comunicarse comunicándose como un puerto serial para el envío de los comandos AT.

2.10 Aplicaciones Web

El desarrollo de aplicaciones Web ha sido una de las industrias más evolutivas en la ingeniería de software reciente. Esta evolución también se ha asociado a la aparición de nuevos lenguajes de programación, herramientas y metodologías para el desarrollo de aplicaciones Web, aunque representan preocupaciones adicionales que los desarrolladores de software deben abordar. La administración de la complejidad del software y la adecuada selección de herramientas de desarrollo se convierten en vitales para el proceso de gestión del cambio asociado al desarrollo de software Web. Esto se debe a que un proceso defectuoso puede conducir a problemas de productividad y mantenimiento. (Velarzo Pardo, Honores, Gómez Moreno, & Vincés Sánchez , 2018)

Las aplicaciones Web son aquellas herramientas donde los usuarios pueden acceder a un servidor Web a través de la red mediante un navegador determinado. Por lo tanto, se define como una aplicación que se accede mediante la Web por una red ya sea intranet o Internet. Por lo general se menciona aplicación Web a aquellos programas informáticos que son ejecutados a través del navegador. (Luján Mora, 2002)

2.10.1 Arquitecturas de las aplicaciones web

Las aplicaciones web se encuentran dentro de las arquitecturas cliente/servidor: un ordenador solicita servicios (el cliente) y otro está a la espera de recibir solicitudes y las responde (el servidor). Una aplicación web es proporcionada por un servidor Web y utilizada por usuarios que se conectan desde cualquier punto vía clientes Web (browsers o navegadores).

La arquitectura de un Sitio Web tiene tres componentes principales:

- I. Un servidor Web
- II. Una conexión de red
- III. Uno o más clientes

El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de archivos y la envía de vuelta al navegador que la solicitó. Las aplicaciones Web están basadas en el modelo Cliente/Servidor que gestionan servidores web y que utilizan como interfaz páginas web (Reyes , 2017)

a. Modelo cliente servidor

Como se muestra en la Fig. 10 el modelo cliente/ servidor cuenta con procesos que ofrecen servicios (servidores) y procesos que utilizan o consumen dichos servicios (Clientes). Se valen de un mecanismo de comunicación basado en pregunta-respuesta, siendo siempre el cliente quien inicia dicha comunicación.

Los servicios pueden ser implementados por varios procesos en diferentes computadoras que los alojan (host). Esto significa que tanto servidores como clientes pueden residir en diferentes máquinas, siendo el mecanismo de comunicación lo que pone en evidencia la distribución

Este mecanismo de comunicación conforma un tercer componente o middleware, que establece la manera en que los procesos (tanto clientes como servidores) van a interactuar, determinando si la comunicación es sincrónica o asincrónica. (Bazán, 2017)

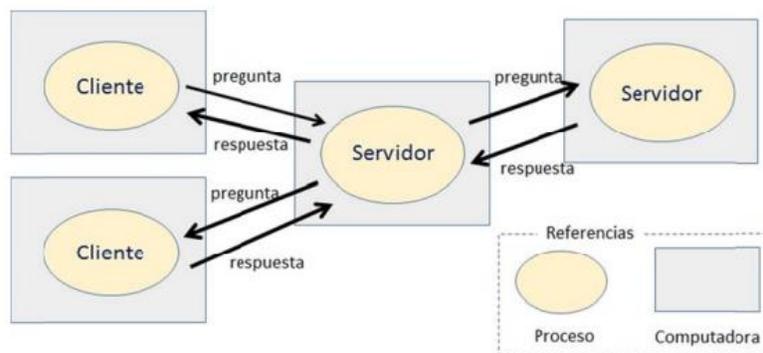


Fig. 10 Modelo cliente/servidor

b. Características del modelo cliente/servidor

El modelo cliente servidor tiene las siguientes características (Vignaga & Perovich, 2019):

- I. Cliente/servidor es fundamentalmente una relación entre procesos ejecutados en computadores distintos. El proceso del servidor hace de éste un proveedor de servicios. El cliente es un consumidor de servicios.
- II. Un servidor puede atender a muchos clientes al mismo tiempo y regular su acceso a recursos compartidos.
- III. Entre cliente y servidor se establece una relación de n a 1. Son siempre los clientes los que inician el diálogo al solicitar un servicio. Y los servidores aguardan pasivamente las solicitudes de los clientes
- IV. El servidor es un proceso que puede residir en el mismo equipo que el proceso cliente o en un equipo distinto a lo largo de una red. Suele ocultarse a los clientes la ubicación del servidor mediante el redireccionamiento de las llamadas de servicio en caso de ser necesario. Un programa puede ser un cliente, un servidor o ambos a la vez.
- V. El software ideal de cliente/servidor es independiente del hardware y de la plataforma de software del sistema operativo. Se debe poder mezclar e igualar las plataformas del cliente y del servidor.
- VI. Clientes y servidores tienen bajo acoplamiento e interactúan a través de un mecanismo determinado de transmisión de mensajes. El mensaje es el mecanismo de entrega para las solicitudes y de respuestas de servicios. (se considera como mensaje tanto a la solicitud como a la respuesta).
- VII. Un mensaje le indica al servidor que servicio se solicita. Los servidores pueden ser sustituidos sin afectar a los clientes, siempre y cuando la interfaz implementada por el servidor no cambie
- VIII. Los sistemas cliente/servidor pueden escalarse horizontal o verticalmente. Por escalabilidad horizontal se entiende el agregar o eliminar estaciones de trabajo de cliente con apenas un ligero impacto en la performance. Escalabilidad vertical significa migrar el servidor actual a un servidor más potente o a múltiples servidores.

- IX. El código del servidor y los datos se conservan centralmente, lo que resulta en un mantenimiento de menor costo y en la protección de los datos de la integridad de los datos compartidos. A su vez, los clientes mantienen su individualidad e independencia.

c. Modelos de distribución en aplicaciones cliente/servidor

Dependiendo de las funciones que se distribuyan, distinguiremos los tres modelos de distribución siguientes (Mendez Aranda, 1999):

- I. **Presentación Distribuida:** En este modelo la parte de la aplicación que se ocupa de la presentación de los datos está situada remotamente respecto de la lógica de negocio y los datos de la aplicación. La lógica de presentación se desarrolla generalmente con alguna de las herramientas de desarrollo Cliente/Servidor. La presentación distribuida en su forma más simple puede consistir en una interfaz gráfica del usuario para acceder a programas de proceso de transacciones ya existentes, alternativamente, se pueden realizar, como parte de la Presentación Distribuida, todos los diálogos con el usuario, incluyendo la validación de datos y la iniciación de la lógica de negocio y acceso a datos.
- II. **Datos Distribuidos:** Cuando se distribuyen los datos su acceso y manipulación quedan separados del resto de la aplicación, como Archivos Distribuidos y Bases de Datos Distribuidas.
- III. **Función Distribuida:** Este modelo proporciona la máxima flexibilidad y permite a los departamentos de desarrollo un control total sobre donde situar las funciones en la red. Es necesario evitar que la distribución de las funciones de la aplicación quede fija por diseño. Cuando esto sucede, para volver a redistribuir funciones y datos casi siempre son necesarias importantes modificaciones o volver a diseñar la aplicación. Un proceso cliente invoca a un proceso servidor que reside en el mismo procesador o en otro procesador de la red. El servidor puede realizar funciones relativas al negocio y acceder a archivos o bases de datos, devolviendo el resultado al proceso cliente que lo ha invocado. El cliente recibe el resultado sin saber si procede del mismo

procesador o de otro distinto. Un servidor de aplicaciones puede, a su vez, actuar como cliente, solicitando funciones de otros servidores de la red.

d. Ventajas y desventajas de la arquitectura cliente/servidor

Entre las ventajas que proporciona la implementación del modelo cliente/servidor (Vignaga & Perovich, 2019):

- I. Posibilidad de reducir costos de desarrollo.
- II. Mejores herramientas de desarrollo.
- III. Modificabilidad es compatible con buen diseño.
- IV. Flexibilidad en el cliente.
- V. Permite un mejor control sobre permisos de acceso a información.
- VI. Escalabilidad. Si la empresa cambia en lo que respecta al tamaño, alcance, requerimientos de información, la escala de la arquitectura cliente/servidor puede ser cambiada.
- VII. La tecnología disponible es tan variada que puede utilizarse la que mejor se adecue a la realidad de la empresa.
- VIII. El software puede ser desarrollado desde varios lugares, lo que reduce el costo de desarrollo.
- IX. Soporta los conceptos de comercio electrónico.

Entre las desventajas proporcionadas por el modelo cliente/ servidor se encuentran

- I. Falta de profesionales calificados.
- II. Necesidad de reimplementación de software existente, ya que el software monolítico no suele ser compatible con la arquitectura cliente/servidor.
- III. Necesidad de entrenamiento de usuarios, debido al cambio de las aplicaciones existentes.

2.10.2 Diseño de páginas web

La World Wide Web (WWW) o también conocida como “telaraña mundial” es un sistema de intercambio y distribución de información ubicado en Internet. Esta información, a la que acceden miles de usuarios desde diferentes puntos del planeta, debe estar perfectamente ordenada y ser fácilmente accesible. Para ello, se utiliza como base el concepto de página web. Las páginas web están compuestas de un conjunto de elementos que van desde texto e imágenes hasta todo tipo de vídeos, sonidos y animaciones, que permiten enriquecer la experiencia final del usuario y su interacción con las mismas (Barzanallana Asenso, 2015).

La cualidad más importante que atesoran las páginas web es su capacidad hipertexto. Esta característica va a permitir al navegante de Internet interactuar no solo con la página web actual, sino también, mediante los denominados hipervínculos, enlaces o links, referenciar otros elementos o páginas web, quedando así, toda la información vinculada y relacionada. Queda en manos del desarrollador de la página web el decidir qué textos, imágenes o elementos actuarán como enlaces activos, de manera que, mediante un simple clic de ratón, permitan al usuario navegar de la página actual a una página o elemento diferente. Para el desarrollo de las páginas web, los programadores emplean los denominados lenguajes de marcas.

El origen de estos lenguajes se remonta a los años setenta, en los que la empresa IBM (Industrial Business Machines) crea el lenguaje GML (generalized markup language) o lenguaje generalizado de marcas, entre 1969 y 1970. IBM pretendía establecer de forma descriptiva, y mediante el uso de marcas, el formato de un determinado documento con base en las partes que lo componen. La idea era separar la presentación de la estructura de los documentos, para evitar así los problemas asociados a los diferentes dispositivos y plataformas existentes de por aquel entonces. Posteriormente, GML se convierte, de la mano de ISO (International Organization for Standardization) en un estándar, concretamente en el ISO 8879, de manera que GML se transforma en SGML (standard generalized markup language) en base al cual se desarrollaron los actuales HTML (hypertext markup language) y XML (extensible markup language). (Paredes Colmenar & Millanes Santos, 2020)

2.10.2.1 Estructura de un sitio web

La estructura lógica o de navegación define como un visitante se va a mover (navegar) de una zona a otra de un sitio web. La estructura lógica y la física son totalmente independientes, aunque se pueden definir de forma que una sea una proyección de la otra. Según el tipo de navegación que se permita en un sitio web, los usuarios tendrán una sensación u otra

En cuanto a la estructura física de un sitio web es la forma en que se almacenan los distintos recursos (ficheros) que forman un sitio web en el sistema de archivos del servidor web. Una estructura física eficiente reduce los costes de mantenimiento, cada vez que se tenga que actualizar el sitio web, el tiempo necesario para localizar el recurso que se desea actualizar se reducirá. La estructura física se debe de planificar antes de iniciar el desarrollo del sitio web. Modificar la estructura física una vez que se han creado muchas páginas es costoso y propenso a cometer errores, ya que para ello hay que actualizar los enlaces entre las páginas, las referencias a las imágenes, etc., para que reflejen la nueva estructura del sitio web. Si el sitio web contiene un número pequeño de ficheros (5 o 10 ficheros), quizás no suponga ningún problema tenerlos todos juntos en el mismo directorio. Pero conforme aumente el número de ficheros, si se ordenan en subcategorías se simplificará su localización y su cambio. No existe una forma perfecta de organizar físicamente un sitio web. (Luján Mora, 2002)

2.10.2.2 Lenguajes de programación para desarrollo de entornos web

Por desarrollo web entendemos aquellas disciplinas, técnicas y herramientas tecnológicas que permiten la creación de sitios o aplicaciones web que funcionan en un ambiente web y que se acceden a ellas a través de un navegador. Las tendencias en desarrollo web, frameworks y lenguajes de programación son muy cambiantes y existen muchas alternativas para cada propósito, por tal razón, los desarrolladores web deben actualizarse constantemente en herramientas robustas y modernas para desarrollar productos aceptables y funcionales en la web.

Las tecnologías que los programadores deben dominar en la actualidad y a futuro, se enfocan en características como: división entre el Back-End y Front-End, las webs simple page, cuyo propósito es crear contenido en una sola página con peticiones asíncronas mediante Ajax con frameworks de Javascript novedosos como: Angular, React y Vue o mediante el mismo lenguaje puro en su versión Vanilla JS. Se introduce un nuevo estándar conocido como API Rest para crear servicios a nivel de Back-End respetando la arquitectura http, ejemplo de ello tenemos: autenticación por Facebook, la autenticación en los servicios de Google. (Zelaya Reyes, 2020)

Principalmente las herramientas que se necesitan para desarrollar una aplicación web se orientan a tres aspectos principales: orientados a definir la estructura de la página como el lenguaje de marcas de hipertexto HTML, lenguajes y frameworks que definen el aspecto de la página como CSS y Bootstrap y por último aquellos lenguajes que ayudan a dar funcionalidad a las aplicaciones web como lo es el JavaScript/TypeScript:

a. Lenguaje de marcas de hipertexto Html

HTML, (HyperText Markup Language, Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo, Javascript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML. Un documento HTML comienza con la etiqueta, <html> y termina con </html>. Dentro del documento (entre las etiquetas de principio y fin de html), hay dos zonas bien diferenciadas: el encabezamiento, delimitado por <head> y </head>, que sirve para definir diversos valores válidos en todo el documento; y el cuerpo, delimitado por <body> y </body>, donde reside la información del documento.

La utilidad del encabezamiento es la directiva <title> y</title>, que permite especificar el título de un documento HTML. Este título no forma parte del documento en sí: no aparece, por ejemplo, al principio del documento una vez que este se presenta con un

programa adecuado, sino que suele servir como título de la ventana del programa que nos la muestra. Por ejemplo:

```
<title>Título de HTML</title>
```

El cuerpo de un documento HTML contiene el texto que, con la presentación y los efectos que se decidan. Dichos efectos se especifican exclusivamente a través de directivas. Esto quiere decir que los espacios, tabulaciones y retornos de carro que se introduzcan en el fichero fuente no tienen ningún efecto a la hora de la presentación final del documento.

b. CSS

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc. (Eguíluz Pérez, 2009)

c. Bootstrap

Bootstrap es una colección de herramientas gratuita y de código abierto para crear sitios web y aplicaciones web. Contiene plantillas de diseño basadas en HTML y CSS para tipografía, formularios, botones, navegación y otros componentes de la interfaz, así como extensiones de JavaScript opcionales. Su objetivo es facilitar el desarrollo de

sitios web y aplicaciones web dinámicos. Bootstrap es un Front End, es decir, una interfaz para el usuario, a diferencia del código del lado del servidor que reside en el backend. (Cimo, 2015).

d. JavaScript y el ECMAScript

JavaScript es un lenguaje de programación de scripts (secuencia de comandos) orientado a objetos. Esta descripción es un poco rudimentaria, hay varios elementos que vamos a diseccionar. Javascript actualmente es principalmente utilizado en internet, junto con las páginas web (HTML o XHTML). Javascript está directamente incluido en la página web (o en un archivo externo) y mejora una página HTML, añadiendo interacción del usuario, animación, ayudas a la navegación, tales como:

- I. Mostrar / ocultar el texto
- II. Deslizamiento de imágenes
- III. Crear presentaciones de diapositivas
- IV. Crear burbujas de información.

Se dice que es un lenguaje del lado del cliente, es decir que los scripts son ejecutados por el navegador del usuario (cliente). Esto difiere de los llamados lenguajes de script del lado del servidor que son ejecutados por el servidor web. Este es el caso de lenguajes como PHP.

ECMAScript específicamente es el estándar que a partir del año 2015 a la actualidad se encarga de regir como debe ser interpretado y funcionar el lenguaje JavaScript, siendo este (JS – JavaScript) interpretado y procesado por multitud de plataformas, entre las que se encuentran los navegadores web, NodeJS u otros ambientes como el desarrollo de aplicaciones para los distintos sistemas operativos que actualmente existen en el mercado. Los responsables de dichos navegadores y JavaScript deben encargarse de interpretar el lenguaje tal como lo fija ECMAScript.

Javascript está diseñado para ser usado en conjunción con HTML, el lenguaje ha evolucionado desde entonces hacia otros destinos. Javascript es regularmente utilizado para hacer extensiones para diferentes programas, como los scripts codificados en Lua o Python. JavaScript también se puede utilizar para construir aplicaciones. Mozilla Firefox es el ejemplo más famoso: la interfaz del navegador se

crea con una especie de HTML llamado XUL y JavaScript que se utiliza para animar la interfaz. (Barzanallana Asenso, 2015)

e. TypeScript

TypeScript es un lenguaje de programación moderno que permite crear aplicaciones web robustas en JavaScript. TypeScript no requiere de ningún tipo de plugin, puesto que lo que hace es generar código JavaScript que se ejecuta en cualquier navegador, plataforma o sistema operativo. TypeScript es un "transpilador", es decir, un compilador que se encarga de traducir las instrucciones de un lenguaje a otro, aquí lo llamaremos también pre-compilador ya que este realmente intenta realizar las funciones de un compilador más las funciones de un traductor de instrucciones.

TypeScript es un lenguaje el cual será compilado finalmente a JavaScript, la versión del JavaScript en la cual será compilado junto con otras configuraciones estará en el archivo tsconfig, TypeScript nos proporciona una serie de ventajas sobre JavaScript, o ES2016, ya que tiene una serie de características que JavaScript no suele tener, como, por ejemplo: interfaces, clases y tipado fuerte. (Serrano Sánchez, 2020)

2.10.3 Bases de datos SQL y NoSQL

Las bases de datos utilizadas en la actualidad en organizaciones y empresas se dividen en cuatro grandes categorías: relacionales, heredadas, in memory ("en memoria") y NoSQL. A esta clasificación clásica es preciso añadirle las bases de datos avanzadas, que son extensión de las bases de datos relacionales, tales como bases de datos MPP y bases de datos de memoria caché. Los sistemas gestores de bases de datos (SGBD) posibilitan el almacenamiento y acceso a los datos, de modo rápido y estructurado. El SGBD de una empresa, es un conjunto de programas que se encargan de administrar y gestionar la información que tiene una base de datos.

Las bases de datos relacionales constituyen hoy en día el corazón de la mayoría de las plataformas distribuidas. Algunos ejemplos de soluciones típicas son: Oracle Database , DB2 de IBM, Microsoft SQL Server, SAP Sybase y MySQL. El mercado de las bases de datos relacionales tradicionales está liderado por Oracle, Microsoft (con

los productos SQL Server y Access, IBM, seguido de Teradata, SAP, Sybase e Informix. Entre los proveedores de bases de datos relacionales libres y abiertas (open source) destacan MySQL, PostgreSQL y María.

El lenguaje de consulta estructurada de las bases de datos relacionadas es SQL (Structural Query Language) que permite acceder a la gestión de dichas bases de datos y realizar tareas típicas como: recoger, eliminar, agregar o modificar información. Las bases de datos NoSQL (Not only SQL) son una categoría de Sistemas de Gestión de Bases de Datos que no utilizan SQL como lenguaje de consulta principal. Estas bases de datos no requieren esquemas de tablas fijas y no soportan operaciones Join. Están optimizadas para operaciones de lectura/escrituras escalables en lugar de pura consistencia. Asimismo, constituyen un ecosistema de información y se están convirtiendo en alternativas viables a las bases de datos relacionales para muchas aplicaciones. (Joyanes Aguilar, 2019)

2.10.4 Framework de Angular para el desarrollo de aplicaciones web

Angular es una plataforma que permite desarrollar aplicaciones web en la sección cliente utilizando HTML y JavaScript para que el cliente asuma la mayor parte de la lógica y descargue al servidor con la finalidad de que las aplicaciones ejecutadas a través de Internet sean más rápidas. El hecho de estar mantenido por Google, así como una serie de innumerables razones técnicas, ha favorecido su rápida adopción por parte de la comunidad de desarrolladores. Permite la creación de aplicaciones web de una sola página (SPA: single-page application) realizando la carga de datos de forma asíncrona.

Hoy en día, la mayoría de las aplicaciones web disponen de una parte cliente dinámica con el objetivo de disminuir la comunicación con el servidor y mejorar la fluidez y experiencia del usuario. Sin embargo, según hasta donde se lleve esta técnica, podemos hablar de grandes patrones de diseño:

- I. Multi page web applications (MPA).
- II. Single-page application (SPA).

Las aplicaciones multi page web applications (MPA) corresponderían al tipo tradicional de aplicación web. Su característica principal es que la mayoría de las acciones de usuario se transforman en solicitudes al servidor de páginas completas (incluyendo diseño, css, JavaScript y contenido).

Por otra parte, el patrón de diseño single-page application (SPA) es una evolución del patrón de diseño MPA + AJAX, pero llevando al extremo el uso de AJAX. Hasta el punto de que en el cliente se carga una única página que se modifica desde el propio cliente (navegador) según las acciones de usuario. Por tanto, toda la navegación por las distintas pantallas o interfaces de la aplicación se realizará sin salir de esa única página. Una de las principales ventajas de las aplicaciones SPA respecto las MPA es la mejora de experiencia de usuario debido a la reducción en el tiempo de respuesta ante las acciones del usuario. Esto se consigue gracias a que:

- I. Ya no se crean páginas completas por cada acción del usuario.
- II. Solo se intercambia la información necesaria con el servidor.

En las aplicaciones SPA, la responsabilidad del aspecto de la aplicación recae principalmente en la parte cliente, mientras que el servidor tiene la función de ofrecer al cliente una API de servicios para dar acceso a la base de datos de la cual se alimenta la aplicación. Los datos intercambiados entre cliente y servidor suelen estar en formato JSON, que es un formato óptimo que el tradicional XML. (Boada Oriolis & Gómez Gutiérrez, 2019)

2.10.4.1 Elementos de una aplicación en Angular

En la web de Angular podemos observar el diagrama de arquitectura que muestra cómo se relacionan los elementos principales de una app los cuales son los mostrados en la Fig. 11:

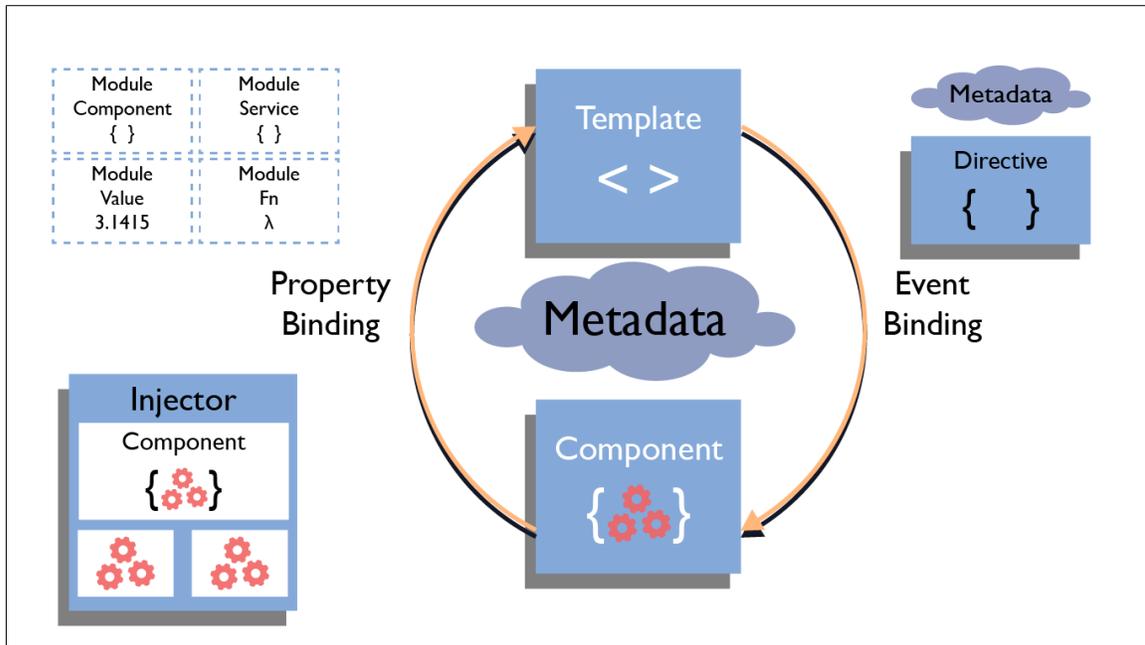


Fig. 11 Arquitectura de angular

Entre los componentes que conforman una aplicación en angular se encuentran: modules, components, templates, Metadata, data binding, directives, services, dependency injection.

Un módulo o NgModule declara un contexto de compilación para un conjunto de componentes. Un NgModule puede asociar sus componentes con código relacionado, como servicios, para formar unidades funcionales. Cada aplicación generada con Angular cuenta con un root module o módulo de raíz llamado convencionalmente AppModule, el cual provee el mecanismo de arranque que inicia nuestra aplicación.

Cada aplicación de Angular tiene al menos un componente. Al igual que el root module, existe el root component que conecta una jerarquía de componentes con el DOM. Cada componente define una clase que contiene data y lógica, y está vinculada a nuestro template HTML. (Boada Oriolis & Gómez Gutiérrez, 2019).

2.10.4.2 Patrón Modelo-Vista-VistaModelo (MVVM)

El patrón MVVM tiene por objetivo simplificar las tareas de desarrollo y mantenimiento del software escrito a través de la división de ocupaciones.

- I. **Modelo:** El modelo, dentro de MVVM es el encargado de representar el modelo del negocio, suministrando de esta manera la base necesaria para la manipulación de los datos de la aplicación, además parte del modelo se lo puede usar como clases POCO (Plain Old CLR Objects).
- II. **Vista:** La vista es la encargada de la parte visual de nuestra aplicación, no se tiene que ocupar en ningún momento en el manejo de datos. En MVVM la vista tiene un rol activo, esto significa que en algún momento la vista recibirá o manejará algún evento (Clic en un botón, alguna tecla presionada, etc.) y tendrá que comunicarse con el modelo, para poder cumplir el requerimiento.
- III. **ViewModel:** El ViewModel (modelo de vista en español) es el encargado de ser la capa intermedia entre el modelo y la vista, procesando todas las peticiones que tenga la vista hacia el modelo, además de tener que ocuparse de manejar las reglas del negocio, la comunicación con aplicaciones externas o consumir datos desde alguna fuente.

2.10.4.3 Componentes, Servicios, Módulos y Rutas

a. Componentes

Elemento que controla una zona de espacio de la pantalla que representa la “vista”. Define las propiedades y métodos que usa el propio template y contiene la lógica y la funcionalidad que utiliza la vista. Pueden tener atributos tanto de entrada (decorador @ Input) como de salida (decorador @Output). También podemos definir al componente como:

Componente = template + metadatos + clase

b. Servicios

Son clases que permiten realizar acciones concretas que ponen a disposición de cualquier componente. Por ejemplo, permiten obtener datos, resolver lógicas especiales o realizar peticiones a un servidor.

c. Módulos

Conjunto de códigos dedicados a resolver un objetivo que generalmente exporta una clase. Cada aplicación posee al menos una clase de módulo que denominamos “módulo raíz” y que, por defecto, posee el nombre de AppModule. En el mismo podemos apreciar una clase con un decorador (@NgModule) y una serie de propiedades entre las que destacamos las siguientes:

- I. Declarations: define las vistas que pertenecen a este módulo.
- II. Exports: declaraciones que han de ser visibles para componentes de otros módulos.
- III. Imports: clases que otros módulos exportan para poderlas utilizar en el módulo actual.
- IV. Providers: servicios usados de forma global y accesibles desde cualquier parte de la aplicación.
- V. Bootstrap: la vista principal o componente raíz, que aloja el resto de las vistas de la aplicación. Propiedad establecida solo por el módulo raíz.

Un módulo puede agrupar diversos componentes relacionados entre sí.

d. Rutas

Se usan las rutas para definir la manera en la que los usuarios navegan a través de la aplicación. Se pueden agregar rutas para definir cómo los usuarios navegan en una aplicación de una parte a otra. También se pueden configurar las rutas para proteger la aplicación contra comportamientos inesperados o no autorizados. (Boada Oriolis & Gómez Gutiérrez, 2019)

2.11 Computación en la nube

La computación en la nube (cloud computing) es un paradigma que permite ofrecer servicios de computación a través de Internet a los que se pueden acceder desde cualquier lugar teniendo la autorización. En este tipo de computación todo lo que puede ofrecer un sistema informático se ofrece como servicio, de modo que los usuarios puedan acceder a los servicios disponibles "en la nube de Internet" sin conocimientos en la gestión de los recursos que usan. "Cloud computing" es un nuevo modelo de prestación de servicios de negocio y tecnología, que permite al usuario acceder a un catálogo de servicios estandarizados y responder a las necesidades de su negocio, de forma flexible y adaptativa, en caso de demandas no previsibles o de picos de trabajo, pagando únicamente por el consumo efectuado (Hernandez Quintero & FLopez Fuente, 2014).

Una de las bases para el origen de la computación en la nube es la virtualización. Este es un mecanismo que permite realizar de una manera no física algunas actividades en el área de la computación y la informática, como, por ejemplo, simular computadores virtuales dentro de un computador físico, simular procesos de negocios, simular procesos médicos, etc. Otro aspecto que facilitó el nacimiento de la computación en la nube es la aparición de la Web 2.0. Esta tecnología permite a los usuarios finales interactuar con las aplicaciones web. Inicialmente, las páginas web fueron estáticas, es decir que no existía la interacción en tiempo real con los usuarios. (Morales Caluña & Altamirano Capelo, 2016)

2.11.1 Características de la computación en la nube

El rápido crecimiento del cloud computing pone a disposición un sinnúmero de servicios con características avanzadas. A continuación, se detallan las principales: (Morales Caluña & Altamirano Capelo, 2016).

- I. Disponibilidad: cloud computing puede considerarse como un servicio full time o 24/7. Es un servicio que está disponible en todo momento y lugar con requerimientos mínimos de conexión.
- II. Accesibilidad: tiene varias formas y mecanismos de accesos. Es posible ingresar desde un computador que únicamente tenga como software un navegador web hasta con equipos avanzados como tablets, celulares, etc.
- III. Tecnología verde: la utilización de cloud computing como política en las áreas de software de las empresas públicas y privadas ayudaría a reducir las emisiones de CO2 a la atmósfera, en vista de que se evitaría mantener los equipos de cómputo encendidos las 24 horas del día.
- IV. Servicio bajo demanda: los usuarios pueden solicitar recursos hardware y software de acuerdo con sus necesidades. Cuando nos referimos a hardware puede ser la solicitud de mayor capacidad de almacenamiento mientras que en software puede ser la petición de nuevos servicios o aplicaciones informáticas.
- V. Reducción de costes: las organizaciones pueden ahorrar dinero al utilizar cloud computing en vista de que pagan por la utilización del servicio y no por los equipos hardware. Esto es transparente para el usuario.
- VI. Adaptabilidad: las aplicaciones en la nube son capaces de ajustarse a cualquier sistema sobre el que estén implementadas. Son capaces de soportar alto número de usuarios concurrentes.
- VII. Nubes libres: el mercado actual brinda aplicaciones en la nube de libre utilización como es el caso de Google Drive. Esta plataforma brinda varios servicios de forma gratuita; su utilización es recomendable para actividades académicas.

2.11.2 Modelo de servicios para la computación en la nube

Tres modelos arquetípicos y sus combinaciones derivadas describen la prestación de los servicios en la nube. A menudo se hace referencia a los tres modelos individuales como el “Modelo SPI,” donde “SPI” hace referencia a Software, Plataforma e Infraestructura respectivamente y se definen del siguiente modo:

- I. Software as a Service (SaaS)
- II. Platform as a Service (PaaS)
- III. Infraestructura as a Service (IaaS)

Por otra parte, se tienen los modelos de despliegue que se pueden implementar en las organizaciones: la nube privada, comunitaria, pública e híbrida, entre otras. Aunque el modelo de nube comunitaria propuesta por el NIST no ha sido muy aceptado por la industria. (Hernandez Quintero & FLopez Fuente, 2014)

a. Software como servicio (SaaS)

Se encuentra en la capa más alta y consiste en la entrega de aplicaciones completas como un servicio. El proveedor de tecnologías de información y comunicación (TIC) ofrece el SaaS (Software as a Service). Para ello dispone de una aplicación que se encarga de operar y mantener y que frecuentemente es desarrollada por el mismo. Con ella se encarga de dar servicio a multitud de clientes a través de la red, sin que estos tengan que instalar ningún software adicional. La distribución de la aplicación tiene el modelo de uno a muchos, es decir, se elabora un producto y el mismo lo usan varios clientes. Los proveedores de SaaS son responsables de la disponibilidad y funcionalidad de sus servicios no dejando de lado las necesidades de los clientes que finalmente son los que usaran el software. Las actividades son gestionadas desde alguna ubicación central, en lugar de hacerlo desde la sede de cada cliente, permitiendo a los clientes el acceso remoto a las aplicaciones mediante la web. Igualmente, las actualizaciones son centralizadas, eliminando la necesidad de descargar parches por parte de los usuarios finales. Un ejemplo claro es la aplicación para el manejo del correo electrónico por medio de un navegador de Internet. (Ábila Megía, 2021)

2.11.3 Plataformas de computación en la nube

Con independencia del modelo de servicio utilizado (SaaS, PaaS, IaaS,) hay cuatro formas principales en los que se despliegan los servicios en la nube y se caracterizan con modelos de despliegue adicionales que afrontan requisitos específicos:

- I. Nube pública: La infraestructura de nube se pone a disposición del público en general o de un gran grupo industrial y es propiedad de una organización que vende los servicios en la nube.
- II. Nube privada: La infraestructura de nube se gestiona únicamente para una organización. Puede gestionarla la organización o un tercero y puede existir tanto en las instalaciones como fuera de ellas.
- III. Nube comunitaria: La infraestructura de nube la comparten diversas organizaciones y soporta una comunidad específica que tiene preocupaciones similares (p.ej., misión, requisitos de seguridad, políticas y consideraciones sobre cumplimiento normativo). Puede ser gestionada por las organizaciones o un tercero y puede existir en las instalaciones y fuera de ellas.
- IV. Nube híbrida: La infraestructura de nube es una composición de dos o más nubes (privada, comunitaria o pública) que se mantienen como entidades separadas pero que están unidas por tecnología estandarizada o propietaria que permite la portabilidad de datos y aplicaciones. (Hernandez Quintero & FLopez Fuente, 2014)

a. Computación en la nube publica

La nube publica consiste en que el proveedor ofrece el mismo servicio a muchos clientes desde el mismo centro de datos de forma que comparten recursos (de almacenamiento, de proceso, etc.). Esto hace posible una gran escalabilidad y eficiencia y generalmente un precio asequible

Entre sus ventajas se encuentran

- I. Escalabilidad
- II. Ahorro de tiempo y costes
- III. Mayor eficiencia de los recursos

Entre sus desventajas

- I. La infraestructura es compartida

- II. Hay poca transparencia para el cliente de cloud ya que no se sabe el resto de los recursos que puede estar compartiendo

Los clientes utilizan los servicios que son procesados en el mismo servidor y pueden compartir espacio en disco u otras infraestructuras de red con otros clientes.

Son ejemplos los servicios de almacenamiento o correo que se ofrecen generalmente en formato gratuito o premium, es decir, el servicio básico es gratuito, pero se paga por un servicio avanzado, por ejemplo, más espacio de almacenamiento o más cuentas de correo. Se reconocen porque en estos casos el contrario y el ANS (Acuerdo de Nivel de Servicio) suelen ser cerrados e innegociables. Es decir, tenemos que aceptar las condiciones del proveedor. (Instituto, 2019)

a. Firebase y sus servicios de cloud computing

Es una aplicación web desarrollada por Firebase Inc. en el año 2011 y tres años más tarde fue adquirida por Google. Se considera como plataforma de aplicación web la cual ayuda a los desarrolladores a la construcción de aplicaciones de buena calidad. La información es almacenada en formato de notación de objetos JavaScript JSON6 la cual no utiliza consultas para insertar, actualizar, eliminar o agregar datos. El backend de un sistema se utiliza como base de datos para almacenar información. Su función esencial es hacer más sencilla la creación de tanto aplicaciones webs como móviles y su desarrollo, procurando que el trabajo sea más rápido, pero sin renunciar a la calidad requerida.

Sus herramientas son variadas y de fácil uso, considerando que su agrupación simplifica las tareas de gestión a una misma plataforma. Las finalidades de estas se pueden dividir en cuatro grupos: desarrollo, crecimiento, monetización y análisis. Es especialmente interesante para que los desarrolladores no necesiten dedicarle tanto tiempo al backend, tanto en cuestiones de desarrollo como de mantenimiento. (Khawas, 2018)

b. Herramientas de desarrollo de Firebase

Entre las herramientas de desarrollo se encuentran: (Khawas, 2018)

- I. Firebase Analytics: Brinda información correspondiente a la aplicación, permite la participación de usuarios, y permite al desarrollador comprender como los usuarios utilizan el servicio
- II. Firebase Cloud Messaging: Se basa en un servicio de pago multiplataforma para mensajes y notificaciones para Android, aplicaciones web e IOS.
- III. Firebase Auth: Los desarrolladores administran mediante correo electrónico y contraseña el inicio de sesión almacenada en Firebase para así autenticar a los usuarios.
- IV. Real-Time Database: Brinda acceso a la DataBase para proporcionar un API que será de utilidad para enlazar los datos de los usuarios y almacenarla en la sección correspondiente a Firebase.
- V. Firebase Storage: Este servicio permite la transferencia de archivos de manera fácil y segura, independientemente de la calidad de las aplicaciones de Firebase, el servicio de almacenamiento de datos está respaldado por Google Cloud.
- VI. Firebase Test Lab for Android: Servicio de pruebas Firebase como capturas de pantalla. Los videos y registros están disponibles en la consola de Firebase ya que proporciona una infraestructura basada en la nube para probar las aplicaciones.
- VII. Firebase Crash Reportinf: Servicio donde se detallan los errores que se crean dentro de la aplicación y corregirlos a tiempo para su correcto funcionamiento en la plataforma.
- VIII. Firebase Notifications: Servicio que envía notificaciones de usuario para las aplicaciones móviles.

c. Bases de datos en tiempo real de firebase

Firebase Realtime Database es una base de datos alojada en la nube. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado. Cuando compilas apps multiplataforma con las SDK de plataformas de

Apple, Android y JavaScript, todos los clientes comparten una instancia de Realtime Database y reciben actualizaciones automáticamente con los datos más recientes.

- I. Tiempo real: En lugar de solicitudes HTTP típicas, Firebase Realtime Database usa la sincronización de datos (cada vez que cambian los datos, los dispositivos conectados reciben esa actualización en milisegundos).
- II. Sin conexión: Las apps de Firebase continúan respondiendo, incluso sin conexión, dado que el SDK de Firebase Realtime Database hace que tus datos persistan en el disco. Cuando se restablece la conexión, el dispositivo cliente recibe los cambios que faltaban y los sincroniza con el estado actual del servidor.
- III. Acceso desde dispositivos cliente: Se puede acceder a Firebase Realtime Database directamente desde un dispositivo móvil o un navegador web; no se necesita un servidor de aplicaciones. La seguridad y la validación de datos están disponibles a través de las reglas de seguridad de Firebase Realtime Database: reglas basadas en expresiones que se ejecutan cuando se leen o se escriben datos. (Firebase inc, 2021)

d. Funciones en la nube de firebase

Cloud Functions para Firebase es un framework sin servidores que permite ejecutar de forma automática el código de backend en respuesta a las solicitudes HTTPS. El código JavaScript o TypeScript se almacena en la nube de Google y se ejecuta en un entorno administrador sin necesidad de administrar ni escalar los servidores.

Entre sus funciones se encuentran:

- I. Integración de plataformas de firebase: Las funciones pueden responder a eventos generados por varias funciones de Firebase y Google Cloud, desde activadores de Firebase Authentication hasta activadores de Cloud Storage.
- II. Sin mantenimiento: se puede implementar código de JavaScript o TypeScript en los servidores mediante la línea de comando. Después firebase aumenta los recursos de procesamiento automáticamente según los patrones de uso de los usuarios.

Después de escribir e implementar una función, los servidores de Google comienzan a administrarla de inmediato. Se puede activar la función directamente con una solicitud HTTP o, en el caso de funciones en segundo plano, los servidores de Google detectarán los eventos y ejecutarán la función cuando se active.

A medida que la carga aumenta o disminuye, la respuesta de Google es escalar con rapidez la cantidad de instancias de servidor virtual necesarias para ejecutar la función. Cada función se ejecuta sin ayuda, en su propio entorno y con su propia configuración. (Firebase inc, 2021)

2.11.3.1 Mapbox y sus aplicaciones de geoposicionamiento

Mapbox GL JS es una biblioteca de JavaScript del lado del cliente para crear mapas web y aplicaciones web con la moderna tecnología de mapeo de Mapbox. Se puede usar Mapbox GL JS para mostrar mapas de Mapbox en un navegador web o cliente, agregar interactividad del usuario y personalizar la experiencia del mapa en la aplicación. (MapBox inc, 2022)

Con mapbox se puede realizar:

1. Visualización y animación de datos geográficos
2. Consultar y filtrar entidades en un mapa
3. Colocar datos entre capas de un estilo de Mapbox
4. Visualización dinámica y estilo de datos personalizados del lado del cliente en un mapa
5. Animaciones y visualizaciones de datos en 3D
6. Agregar marcadores y ventanas emergentes a los mapas mediante programación

2.11.3.1.1 Herramientas de mapbox

Las API de servicios web de Mapbox permiten acceder mediante programación a las herramientas y servicios de Mapbox. Se pueden usar estas APIs para recuperar

información sobre la cuenta en uso, cargar y cambiar recursos y usar las herramientas principales de Mapbox.

Las API de Mapbox se dividen en cuatro servicios distintos: Mapas, Navegación, Búsqueda y Cuentas. Estas páginas de descripción general se dividen en las API individuales que componen el servicio. Un punto final o End-Point es un método específico dentro de una API que realiza una acción y se encuentra en una URL específica. (MapBox inc, 2022)

- I. Map service: El servicio Mapbox Maps incluye varias API para crear y solicitar mapas, ya sea interactuando con una API directamente o usando un SDK. Esto incluye servicios para solicitar mosaicos de mapas, solicitar imágenes estáticas, cargar datos en su cuenta de Mapbox, consultar datos en un conjunto de mosaicos, etc.
- II. Navigation service: El servicio de navegación de Mapbox incluye varias API para obtener indicaciones o crear experiencias de navegación, ya sea interactuando directamente con las API del servicio de navegación o usando un SDK móvil. Esto incluye servicios para solicitar direcciones, crear isócronas, hacer coincidir rutas con la red de carreteras en un mapa y más.
- III. Search service: El servicio de búsqueda de Mapbox incluye dos API: la API de codificación geográfica de Mapbox y la API de búsqueda de Mapbox Japón. La API de codificación geográfica de Mapbox permite realizar operaciones de codificación geográfica directa e inversa. Forward Geocoding toma el texto en forma de dirección o lugar y lo convierte en coordenadas geográficas (latitud/longitud). La geocodificación inversa toma coordenadas geográficas (latitud/longitud) y las convierte en una dirección o lugar en forma de texto.
- IV. Accounts Service: Se utiliza el servicio de cuentas de Mapbox para interactuar con las cuentas de usuario mediante programación. El servicio de cuentas incluye una API: la API de tokens de Mapbox. (MapBox inc, 2022)

a. APIs para la representación de mapas

El servicio Mapbox Maps incluye varias API para crear y solicitar mapas, ya sea interactuando con una API directamente o usando un SDK. Esto incluye servicios para solicitar mosaicos de mapas, solicitar imágenes estáticas, cargar datos de una cuenta de Mapbox, consultar datos en un conjunto de mosaicos, etc.

Entre las APIs utilizadas se encuentran:

- I. Vector tiles api
- II. Raster tiles Api

b. Vector Tiles Api

Una tesela vectorial es un formato de datos liviano para almacenar datos vectoriales geoespaciales, como puntos, líneas y polígonos. Las teselas vectoriales se utilizan para crear conjuntos de teselas vectoriales de Mapbox. Los mosaicos vectoriales se pueden usar como fuentes de estilos o se pueden consultar directamente para crear experiencias interactivas con Mapbox Mobile SDK y Mapbox GL JS.

Las teselas vectoriales codifican información geográfica de acuerdo con la especificación de teselas vectoriales. (MapBox inc, 2022)

c. Ráster Tiles Api

Los datos ráster son un formato de datos basado en píxeles en el que los datos se almacenan en una estructura de cuadrícula. Cada píxel, o unidad de información, tiene el mismo tamaño y forma, pero varía en valor. Todas las fotografías digitales se almacenan en este formato, incluidas las imágenes de satélite como las que se utilizan en el conjunto de fichas de Mapbox Satélite. Los conjuntos de teselas ráster se utilizan en las herramientas de Mapbox para mostrar mapas como una cuadrícula de imágenes que se pueden cargar en el mapa. (MapBox inc, 2022)

2.12 Recolección de requisitos para el prototipo Nebulizador inteligente automotriz germicida del SARS-CoV-2

El dispositivo nebulizador inteligente germicida del SARS-CoV 2 es un sistema propuesto para la desinfección de interiores de un espacio específico de manera automática, siendo activado mediante la presencia de un usuario. Es por ello que el sistema requiere de diversos elementos de automatización y tecnologías de esparcimiento de líquidos para realizar las tareas de desinfección. A su vez debe ser contemplada la implementación de un sistema de monitoreo ajeno del dispositivo que ayude en las tareas de mantenimiento y operación de este al representar los datos de su funcionamiento, por lo que el dispositivo deberá de ser capaz de recopilar datos de su funcionamiento y mandarlos a través de un canal de comunicaciones.

Entre los requerimientos para el desarrollo dispositivo se encuentran;

- I. El dispositivo deberá tener un sistema que realice la desinfección dentro de un espacio de manera óptima.
- II. El dispositivo podrá reconocer la presencia de un usuario para accionar el sistema de desinfección solo cuando el mismo se encuentre presente. Estos criterios pueden ser alterado por la naturaleza del desinfectante a utilizar.
- III. El dispositivo recopilará la información de funcionamiento de cada uno de sus componentes que son:
 - a. Temperatura
 - b. Cantidad de desinfectante
 - c. Carga de la batería
 - d. Estado del dispositivo accionador
 - e. Ubicación
- IV. Cada uno de los datos podrán ser enviados mediante un módulo de comunicación. Estos datos serán enviados conforme al formato descrito en la Tabla 8.
- V. Estas funciones de accionamiento y envío de datos serán automáticos siendo el usuario responsable solo del mantenimiento del dispositivo.

- VI. El dispositivo deberá ser seguro para el usuario al no comprometer la integridad del mismo ni el entorno en el que operará el dispositivo nebulizador.

Tabla 8 Datos recopilados del dispositivo

Título	Tipo de dato	Descripción
Sensor de detección	Bool [True, False]	Es el dato referente a la detección del usuario al accionar el dispositivo que puede ser verdadero o falso.
Batería	Integer [0,100]	Este dato representa la cantidad de batería restante del dispositivo en porcentaje e 0% al 100%
Desinfectante	Integer [0,100]	Este dato representa la cantidad de desinfectante restante en el contenedor del dispositivo en porcentaje e 0% al 100%
Temperatura	Double [-55.0, 150.0]	Son los datos de temperatura que puede enviar el dispositivo en grados definidos por las especificaciones técnicas del sensor de temperatura
Latitud	Double [-90.0, 90.0]	Estos son los datos referentes a la posición espacial del dispositivo en formato de latitud y longitud
Longitud	Double [-180.0,180.0]	

2.13 Análisis de las necesidades y requerimientos para la aplicación de monitoreo

Es importante para el desarrollo del prototipo nebulizador germicida del SARS-CoV-2 incluir un sistema de monitoreo que ayude con las operaciones de supervisión y mantenimiento además de operar de manera remota y a tiempo real. Dicho sistema de monitoreo deberá recibir la información de funcionamiento transmitida por el dispositivo en tiempo real y a su vez almacenarla para que los datos puedan ser vistos, consultados y verificados. Para ayudar a los procesos de mantenimiento el sistema deberá alertar al usuario los aspectos relevantes sobre el funcionamiento y operación del dispositivo. En cuanto al proceso de consulta y verificación la información deberá ser mostrada de manera visual y de fácil entendimiento, sin embargo, deberá ser capaz de presentar la información detallada de los eventos registrados por el sistema y los

datos generales del dispositivo. Considerando los aspectos anteriormente mencionados, las funcionalidades generales del sistema son las siguientes:

- I. Recepción de datos enviados por el dispositivo y almacenamiento de datos en tiempo real.
- II. Sistema de alertas para el mantenimiento del dispositivo
- III. Presentación de los datos generales de funcionamiento de manera visual y de manera detallada

El prototipo nebulizador germicida del SARS-CoV-2 envía periódicamente datos extraídos por cada uno de sus sensores referentes a su funcionamiento, ubicación y operación siendo la información recibida la mostrada en la Tabla 9.

Tabla 9 Datos enviados por el dispositivo nebulizador

Relación de datos enviados		
Título	Tipo de dato	Descripción
Sensor de detección	Bool [True, False]	Es el dato referente a la detección del usuario al accionar el dispositivo que puede ser verdadero o falso.
Batería	Integer [0,100]	Este dato representa la cantidad de batería restante del dispositivo en porcentaje e 0% al 100%
Desinfectante	Integer [0,100]	Este dato representa la cantidad de desinfectante restante en el contenedor del dispositivo en porcentaje e 0% al 100%
Temperatura	Double [-55.0, 150.0]	Son los datos de temperatura que puede enviar el dispositivo en grados definidos por las especificaciones técnicas del sensor de temperatura
Latitud	Double [-90.0, 90.0]	Estos son los datos referentes a la posición espacial del dispositivo en formato de latitud y longitud
Longitud	Double [-180.0,180.0]	

Estos datos deben de ser almacenados siguiendo el formato descrito en la Tabla 9 y actualizados cada que el dispositivo envíe información de manera periódica.

Según los datos enviados por el dispositivo nebulizador descritos en la Tabla 9 y recibidos por el sistema se determinarán alertas que indiquen aspectos esenciales del dispositivo y que ayuden a determinar elementos críticos que afecten el funcionamiento del sistema. Los criterios para el accionamiento de una alerta están descritos en la relación de la Tabla 10.

Tabla 10 Relación de criterios para la activación de una alerta

Tipo de alerta	Evento	Descripción
Alerta grave	No hay conexión con el dispositivo	Esta alerta se debe enviar cuando no se reciban los datos provenientes del dispositivo en un tiempo determinado.
	Desinfectante agotado	Para garantizar la presencia de líquido en el contenedor del dispositivo esta alerta se enviará cuando se detecte una cantidad menor al 5%
	La batería está a punto de agotarse	Para garantizar que el dispositivo se encuentre siempre en operación se enviará esta alerta la cual será accionada cuando la batería se encuentre por debajo del 5%
	Falla en el suministro de energía	Se enviará esta alerta cuando se detecte la falta de energía suministrada por las baterías de los componentes alternos del dispositivo.
Alerta media o aviso	Se detectan altas temperaturas	Para proteger la integridad del dispositivo y sus componentes se accionará esta alerta cuando la temperatura recibida del dispositivo sea mayor a 50°
	El contenedor de desinfectante se encuentra por debajo del 40% de su capacidad	Para prevenir al usuario sobre la posible falta de líquido en el contenedor del dispositivo se enviará esta alerta cuando se detecte una cantidad de líquido menor al 40%
	La batería se encuentra por debajo	Para prevenir al usuario sobre la posible falta de alimentación del dispositivo se accionará esta alerta cuando se detecte una capacidad energética del 20% de su capacidad.

	del 20% de su capacidad	
	Error en los datos de posicionamiento	Cuando se detecte la falta de datos coherentes de posicionamiento [lat: 0, lon:0] se accionará esta alerta
Alerta normal o Evento	Se ha accionado el dispositivo	Cada que se accione el dispositivo se tendrá un registro y se mostrará esta alerta.
	Contenedor del desinfectante al 100% de su capacidad	Cuando el contenedor llegue al 100% de su capacidad se mandará este evento.
	La batería se encuentra al 100% de su capacidad	Cuando la batería se cargue por completo o se encuentre con una capacidad del 100% se mandará esta alerta.

Las alertas que el sistema debe mandar son las descritas en la Tabla 10 y las mismas deben ser almacenadas en un historial de eventos. Dicho historial de eventos se actualizará en tiempo real mostrando siempre la información resiente de operación y funcionamiento del dispositivo.

Para la presentación de los datos generales se utilizará la información descrita en la Tabla 9 y Tabla 10 tomando en cuenta los datos enviados por el dispositivo. De esta forma se deberán mostrar los datos provenientes del dispositivo, así como el registro completo de eventos relacionados con el dispositivo y los datos de geoposicionamiento referentes a su ubicación cada uno de ellos siendo actualizados en tiempo real.

En general la aplicación deberá representar la información actualizada y en tiempo real de los siguientes aspectos:

- I. Estado general del dispositivo:
 - a. Estado de la batería: Representando el porcentaje de carga de la batería.
 - b. Estado del contenedor: Representando el porcentaje de líquido restante en el contenedor:
 - c. Estado del usuario: Representando si se encuentra presente el usuario o no.
 - d. Estado de la temperatura: Representando a la temperatura en la que se encuentra trabajando el dispositivo.

- II. Alertas de funcionamiento.
 - a. Alertas referentes al accionamiento y conectividad del dispositivo.
 - b. Alertas referentes al estado del contenedor de líquido desinfectante.
 - c. Alertas referentes al estado de la batería
 - d. Alertas referentes a los datos de ubicación.
 - e. Alertas referentes a la temperatura de operación.

- III. Datos detallados del dispositivo
 - a. Datos generales de funcionamiento.
 - b. Historial de eventos relacionados con las alertas.
 - c. Datos de ubicación.

Capítulo III

Metodología

En este punto se presentarán las actividades realizadas durante el desarrollo del proyecto, siguiendo una metodología de desarrollo tradicional en cascada con las fases de análisis, diseño, desarrollo y pruebas de funcionamiento. Se describirán las necesidades y funcionalidades requeridas, los diseños realizados, la implementación y el resultado de la implementación de las tecnologías para el desarrollo de una aplicación web funcional para el prototipo nebulizador germicida del SARS-CoV-2.

3.1 Arquitectura de solución

En la Fig. 12 se muestra el diseño de la arquitectura de solución general para la aplicación web de monitoreo en conjunto con el dispositivo nebulizador inteligente automotriz del SARS-CoV-2 los cuales estarán operando de manera simultánea.

La arquitectura de solución general propuesta muestra el funcionamiento general del dispositivo complementado con los elementos necesarios para el funcionamiento con una aplicación web. El usuario interactuará tanto con el dispositivo como la aplicación web para el monitoreo, el dispositivo enviará los datos mediante un mensaje SMS a un receptor que mediante un servidor en nodeJs podrá enviar los datos a una nube de Firebase y mostrarlos en la aplicación web.

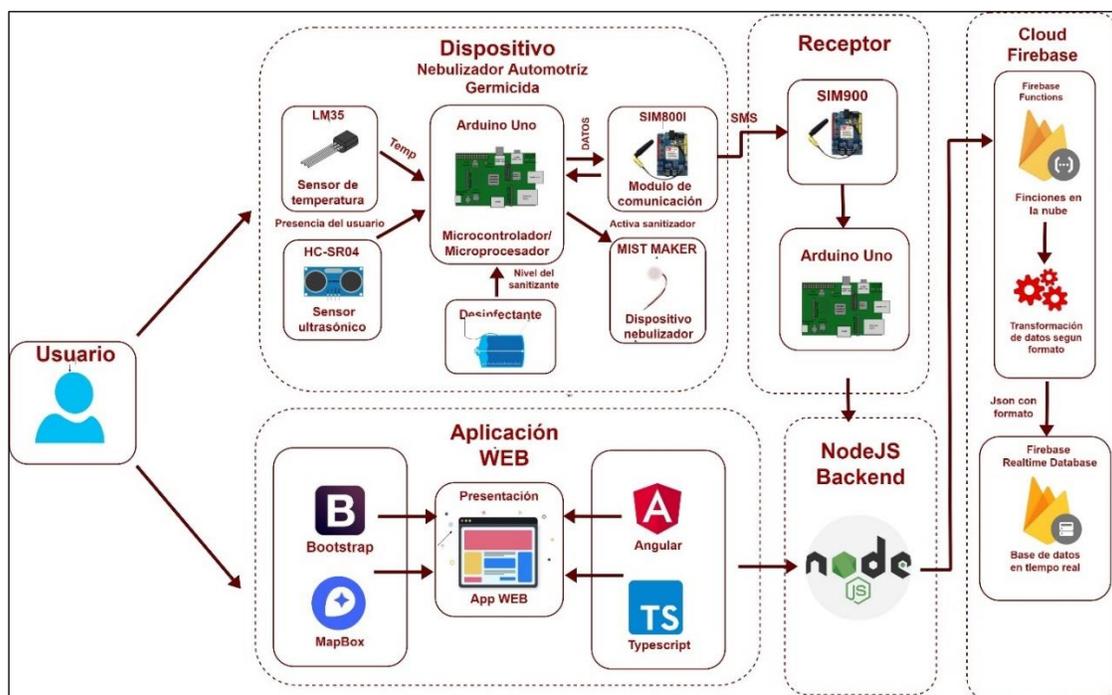


Fig. 12 Arquitectura general de la solución

3.1.1 Diseño general del dispositivo nebulizador

El dispositivo nebulizador es aquel que se encarga de realizar la desinfección del entorno mediante un agente desinfectante. Esto se llevará a cabo utilizando una nebulización con el disco piezoeléctrico de 113kHz Mist Maker mostrado en la Fig. 6 y utilizará una serie de sensores para la obtención de datos sobre su funcionamiento, para el control del dispositivo nebulizador se utilizará el microcontrolador Arduino Uno. La placa de Arduino uno tiene todo lo necesario para realizar el control y configuración de cada uno de los elementos del dispositivo nebulizador, así como determinar la presencia del usuario, accionar el componente de nebulización y mandar los datos recopilados de cada uno de los siguientes sensores:

- I. Sensor de proximidad para la detección del usuario
- II. Sensor de temperatura
- III. Sensor para la medición del desinfectante
- IV. Sensor para la medición de la carga de batería.
- V. Modulo GPS para la obtención de datos de posicionamiento

Los datos serán enviados utilizando el módulo de comunicación GSM SIMS1800I mostrado en la Fig. 8, este módulo nos permitirá mandar los datos obtenidos usando como canal de comunicación el servicio de mensajes simple SMS. por lo tanto, los datos serán enviados siguiendo el siguiente formato:

bateria=[x]&sanitizante=[x]&sensor=[x]&temperatura=[x]&longitud=[x]&latitud=[x]

siendo [x] reemplazado por el dato correspondiente y siguiendo el formato de la Tabla 8.

3.1.1.1 Diagrama de bloques

En la Fig.12. se muestra la solución general del proyecto, pero para tener un entendimiento general sobre el funcionamiento del hardware empleado en el diseño

del dispositivo nebulizador, se lo ha dividido en varias partes otorgando una mejor visión de este. La Fig. 13 indica el diagrama de bloques del hardware realizado.

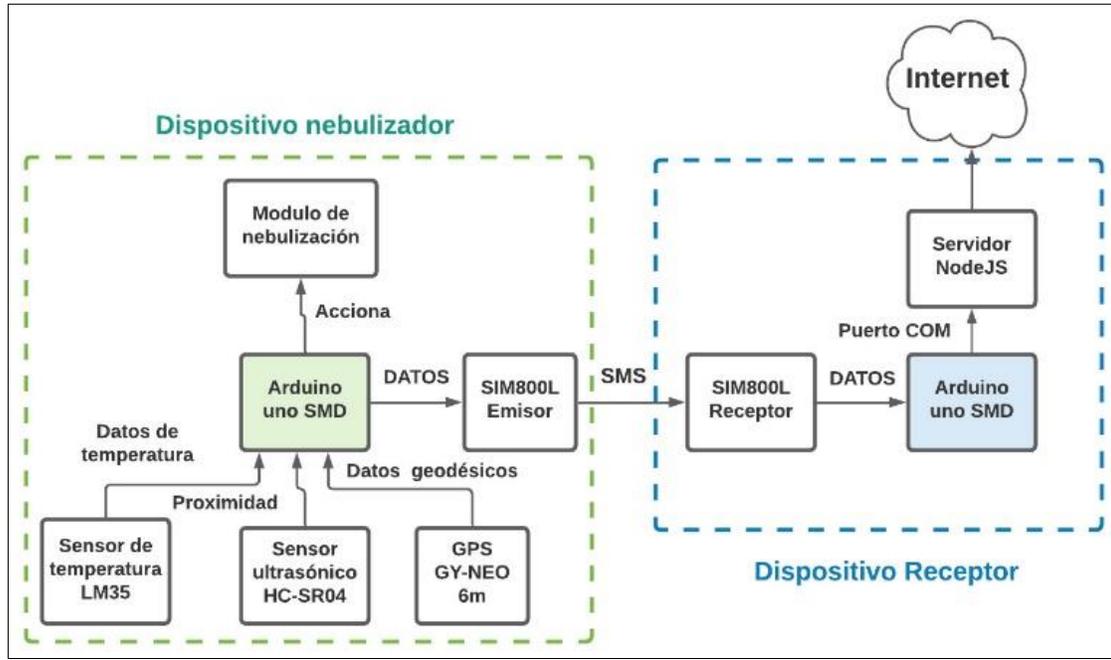


Fig. 13 Diagrama de bloques

De acuerdo con la Fig. 13 el hardware consta de las siguientes partes:

- I. Un primer microcontrolador Arduino uno que se encargara de realizar el control del dispositivo del nebulizador germicida, accionando el mecanismo de nebulización al detectar la presencia del usuario con el señor ultrasónico. Además, recopila los datos de su funcionamiento y posicionamiento de los demás módulos que serán enviados mediante el módulo SIM800 emisor estos datos están representados en la Tabla 8.
- II. Se cuentan con dos módulos GSM/GPRS sim 800L uno conectado al dispositivo nebulizador que se encargará de enviar los datos de el mismo mediante mensajes SMS, El segundo modulo SIM800L recibirá el mensaje y lo transmitirá a su respectivo microcontrolador Arduino UNO

El segundo Arduino UNO recibirá los datos de su respectivo módulo de comunicación SIM800 y se encargará de adecuar los datos recibidos mediante el mensaje de texto. Este módulo estará conectado mediante el puerto serial COM a un servidor en nodeJS que se encargará de enviar los datos a internet

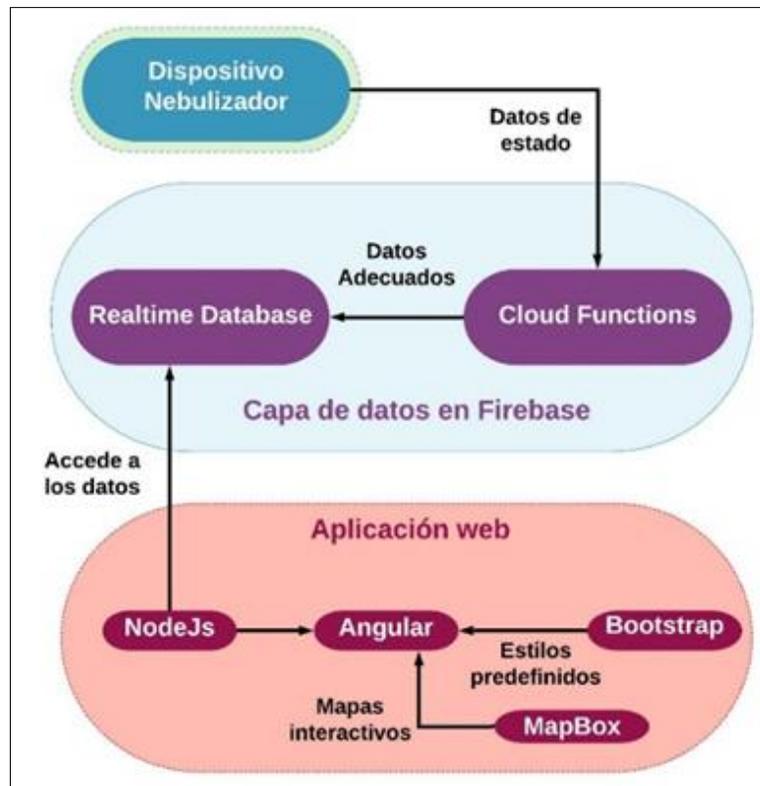


Fig. 14 Diagrama de bloques

Como se muestra en la Fig. 14 también se extrajo la arquitectura elemental para el funcionamiento de una aplicación web de monitoreo utilizando un diagrama de bloques y especificando su comportamiento. Se contemplan tres aspectos generales para el funcionamiento; El dispositivo nebulizador quien mandará la información, una capa de datos utilizado la nube de firebase quien recibirá los datos y los almacenará en su base de datos y por último una aplicación web, quien accederá a la base de datos para mostrarlos en tiempo real.

3.1.1.2 Diagramas de conexión del dispositivo nebulizador

Para el desarrollo del dispositivo y siguiendo el diagrama de bloques de la Fig. 13 comenzaremos por los sensores de control y monitoreo constituidos por un sensor de temperatura, uno de proximidad, un módulo GPS para la obtención de datos geodésicos y por último el módulo de comunicación Sim 900L.

Comenzando con el sensor de proximidad se utilizará el HC-SR04 cuyas especificaciones se encuentran en la Tabla 4. Como se muestra en la Fig. 15 Este sensor requiere de una alimentación de 5v para funcionar los cuales serán proporcionados por la placa de Arduino y un pin Triger y Echo para el control y medición de las distancias, los cuales estarán conectados a las terminales digitales del Arduino Uno, 9 y 10 respectivamente .

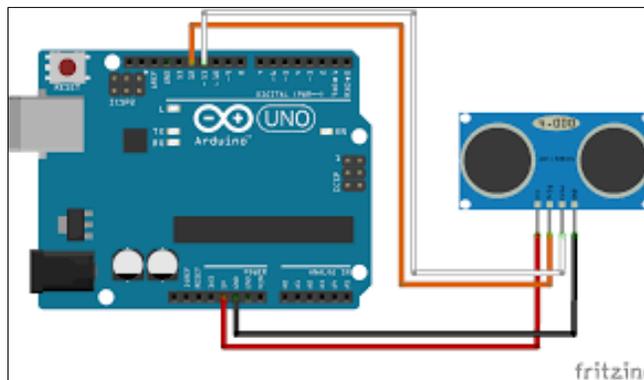


Fig. 15 Conexión HC-SR04

Una vez conectado se deberá escribir la programación para traducir los valores recibidos a metros o centímetros utilizando la siguiente formula:

$$Distancia (m) = \frac{\left\{ (Tiempo del pulso ECO) * \left(Velocidad del sonido = \frac{340m}{s} \right) \right\}}{2}$$

Para el sensor de la temperatura se optó por utilizar el sensor LM35, como se ve en la Tabla 3, su salida es analógica, es decir, te proporciona un voltaje proporcional a la temperatura y tiene un rango desde -55°C a 150°C . además tiene la peculiaridad de que no es necesario de un microprocesador o microcontrolador para medir la temperatura.

Ya que el valor recibido es un voltaje proporcional a la temperatura se tiene que utilizar el siguiente modelo para poder convertir el voltaje a milivolts y posteriormente a grados centígrados usando:

$$\text{Milivolts} = \frac{\text{Valor Entrante}}{1024} * 500$$
$$\text{Centrigrados} = \frac{\text{Milivolts}}{10}$$

El sensor de temperatura requiere de una alimentación de al menos 4V para su funcionamiento por lo que se conectará como se ve en la Fig. 16, conectando el pin de alimentación al suministro de 5v de la placa de Arduino uno y su salida al pin analógico A0.

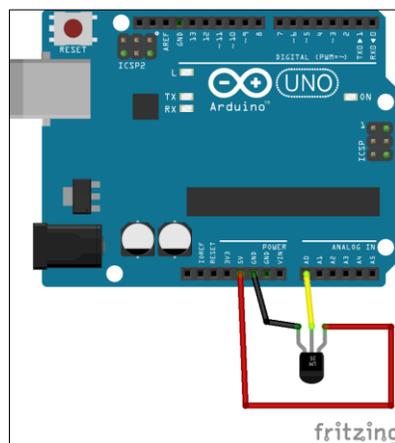


Fig. 16 Conexión LM35

Para el módulo GPS NEO-6M utiliza la comunicación USART para comunicarse con el microcontrolador o terminal de PC. El módulo recibe información como la latitud, longitud, altitud, hora UTC, etc. de los satélites en forma de cadena NMEA. Esta cadena necesita ser analizada para extraer la información que queremos usar.

El módulo GPS GY-NEO al igual que los sensores puede operar con un voltaje de entrada de 5 V como se ve en la Tabla 7 y a su vez tiene salidas RX y TX que estarán conectadas en los pines 2 y 3 de la placa de Arduino como se muestra en la Fig. 17.

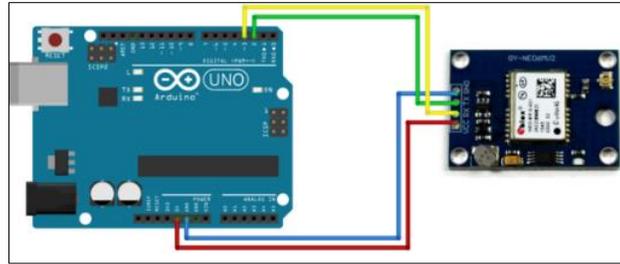


Fig. 17 Conexión GPS GY-NEO 6M

Por último, para garantizar el envío de datos se utilizará un medio de transmisión de datos utilizando mensajería móvil SMS ya que su red de transmisión aporta una gran cobertura y permite mandar datos de manera segura en un vehículo en movimiento. Es por ello que se realizará la implementación del módulo GSM/GPRS SIM800L es un dispositivo quad-band GSM/GPRS que trabaja en las frecuencias GSM850MHz, EGSM900MHz, DCS1800MHz y PCS1900MHz.

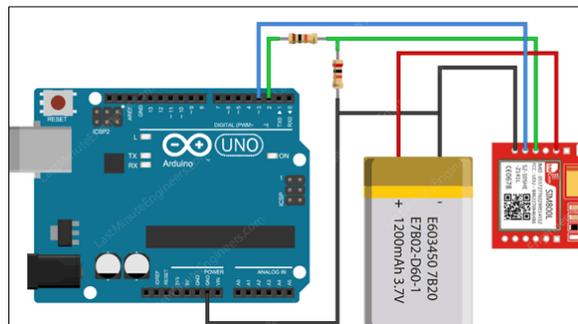


Fig. 18 Conexión SIM800L

Para la conexión con el SIM800L se realizará como se ve en la Fig. 18, se implementó una batería externa de 4.3V debido a que se necesita este voltaje para el correcto funcionamiento del módulo SIM800L como lo indica la Tabla 6 es por ello que se alimentara directamente de la batería a los sus pines correspondientes, en cuanto al pin RT y XT del módulo sim 800L se conectarán a los pines 1 y 2 respectivamente de las entradas digitales del Arduino. Para que la transmisión de datos funcione

adecuadamente se deberá conectar la tierra del Arduino junto con la tierra de la batería.

3.1.1.3 Diagrama de conexión del componente nebulizador con disco piezoeléctrico

Para el funcionamiento del disco piezoeléctrico se requiere un circuito que adecue la corriente suministrada para que funcione correctamente el disco piezoeléctrico, se requiere una onda senoidal con una frecuencia de 113khz como se ve en la Tabla 5 es por ello que se realizó el siguiente circuito que cumpla dicha función.

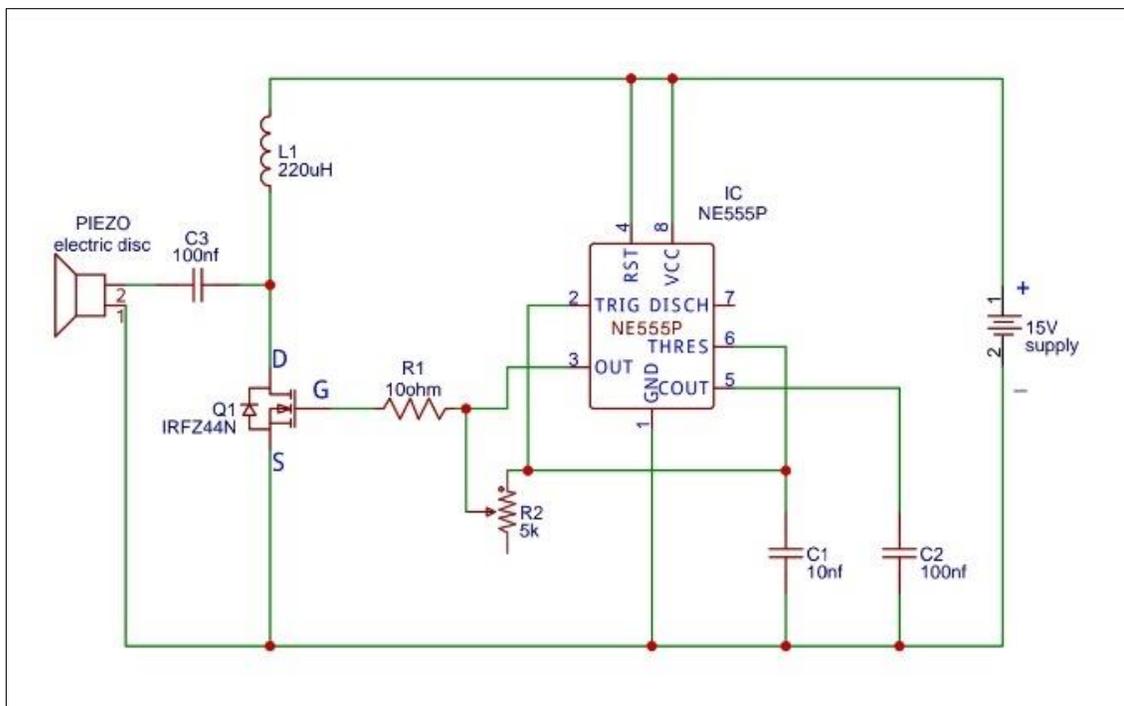


Fig. 19 Diagrama de circuito para el componente de nebulización

El circuito utiliza los siguientes componentes:

- I. 1x NE555
- II. 1x IRLZ44N
- III. 1x 113kHz Piezoelectric Disc

- IV. 1x 5k Ω Trimmer
- V. 1x 10 Ω Resistor
- VI. 1x 220 μ H Inductor IRL
- VII. 2x 100nF, 1x 10nF Capacitor

3.1.2 Capa de datos

3.1.2.1 Dispositivo receptor

Para recibir los datos provenientes del dispositivo nebulizador se requerirá de otro dispositivo Arduino uno con su respectivo módulo de comunicación SIM800I que está conectado tal y como se muestra en la Fig. 18. Esto con la finalidad de recibir un mensaje de texto y enviarlo a través de la web. Por si solo el Arduino no es capaz de realizar una conexión a través de la web. Es por lo que utilizaremos NodeJS que es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con entrada y salida de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Entre las tareas que realiza el servidor se encuentran las siguientes:

- I. El servidor logrará comunicarse con la placa Arduino UNO utilizando el cable serial y configurando el puerto COM, de esta forma todos los datos enviados por Arduino a su puerto Serial serán recibidos por el servidor de nodeJS.
- II. Los datos recibidos por el dispositivo Arduino serán separados y adecuados para ser enviados a internet ya que los mismos se encuentran en el formato visto en la obtención de requisitos.
- III. Al ser separados serán enviados a internet conforme a los requerimientos de la nube de firebase para las Cloud Functions

3.1.2.2 Diseño de la base de datos para la aplicación web

Siguiendo con los elementos vistos en la Fig. 14 para la capa de datos se utilizarán las bases de datos de Firebase alojadas en la nube. Estas son No SQL y almacenan los datos en un formato JSON, además permiten alojar y disponer de los datos e

información de la aplicación en tiempo real, manteniéndolos actualizados, aunque el usuario no realice ninguna acción. Firebase envía automáticamente eventos a las aplicaciones cuando los datos cambian, almacenando los datos nuevos y aunque no hubiera conexión por parte de un usuario, sus datos estarían disponibles para el resto y los cambios realizados se sincronizarían una vez restablecida la conexión.

Dado que los datos se almacenan como JSON la estructura de los datos almacenados son diferentes a los usados comúnmente en bases de datos SQL por lo que se tendrá que definir la estructura de los datos de la siguiente forma:

```
{ID-Dispositivo}
  |-> alertas
    |   |-> {id alerta}
    |     |-> Activo: bool
    |     |-> Evento: string
    |     |-> Fecha: fulldate
    |     |-> hora: time(hh:mm)
    |     |-> id: num
    |     |-> Latitud: double
    |     |-> Longitud: double
    |     |-> Nivel: string
  |-> estado
    |-> Dispositivo
      |-> Bateria: int
      |-> Conductor: string
      |-> Desinfectante: int
      |-> Latitud: double
      |-> Longitud: double
      |-> Temperatura: int
```

Los datos almacenados provenientes del dispositivo referentes a su estado como la cantidad de la batería, la presencia del conductor, la latitud, longitud y la temperatura serán almacenados y dependiendo de los datos recibidos se mandará una alerta cuyos datos serán almacenados e identificados con su respectivo id. Para recibir los datos y almacenarlos en la base de datos se hará uso de las cloud functions o funciones en la nube del servicio de firebase, quienes se encargarán de recibir los datos enviados por el dispositivo y almacenarlas siguiendo el formato anteriormente mencionado. Al recibir los datos las funciones alojadas en la nube procederán a determinar si se lanzará una alerta al usuario siguiendo los criterios descritos en la Tabla 10, de cumplir con dichos criterios la función en la nube procederá a enviar la alerta y registrar el evento en la base de datos almacenando el título del evento, la fecha y la hora, la descripción del evento, su ubicación en términos de latitud y longitud, el nivel de gravedad, un identificador y una variable que defina si la alerta se encuentra activa o no.

3.1.3 Aplicación web

Para el último elemento visto en la Fig. 14 se definirá la estructura visual y los elementos que constituye la aplicación web teniendo 3 aspectos generales; la vista de estado general, la vista de alertas y por último la vista de datos específicos y de ubicación.

1. Vista de estado general

La vista general representa los datos esenciales del estado del dispositivo siendo estos; su temperatura, la cantidad de desinfectante restante, la carga de batería y la detección del usuario

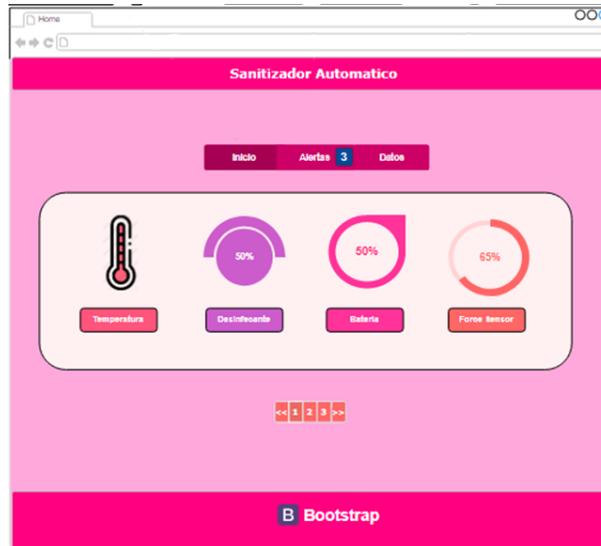


Fig. 20 Diseño vista de estado general

Como se ve en la Fig. 20 cada uno de los indicadores se mostrarán de manera visual utilizando una gráfica animada, además tendrá la capacidad de cambiar de estado en tiempo real conforme se reciban los datos del dispositivo.

2. Diseño de vista general

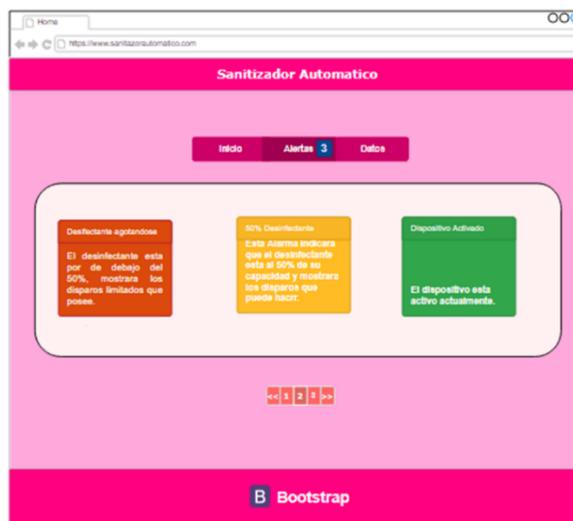
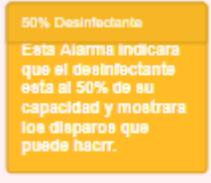
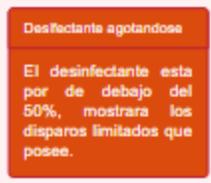


Fig. 21 Diseño vista de alertas

Para el diseño de para la vista de la Fig. 21 se mostrarán cada una de las alertas activas almacenadas en la base de datos y considerando algunos aspectos definidos en la Tabla 10 referentes a los criterios de activación de la alerta.

Una alerta proporciona información esencial del estado del dispositivo, estas alertas pueden indicar fallas potenciales del dispositivo, fallas existentes o eventos relacionados con la funcionalidad del dispositivo. Por lo tanto, se manejarán 3 tipos de alertas como se muestra en la Tabla 11.

Tabla 11 Niveles de alerta

Ejemplo	Grado	Descripción
	Evento (Alerta normal)	Este tipo de alertas se muestran con eventos que no impliquen problemas de funcionamiento o de operación.
	Aviso (Alerta media)	Este grado de alerta se muestra con eventos en donde se encuentran problemas potenciales para el funcionamiento del dispositivo, pero no llegan a un nivel crítico.
	Alerta (Alerta grave)	Este grado corresponde a los eventos críticos del dispositivo. Cuando se encuentran problemas en donde una o mas funciones no operen correctamente o van a dejar de operar,

Cada una de las alertas vistas en la tabla Tabla 11 se mostrará junto con su respectivo evento almacenado en la base de datos y que se encuentre activo, a su vez el usuario tendrá la capacidad de eliminar la alerta que desee.

3. Diseño visto de historial

En esta vista se tendrán los datos generales que se tienen sobre el funcionamiento del dispositivo, es una vista detallada de los datos de funcionamiento, su ubicación y los eventos reaccionados con el dispositivo.

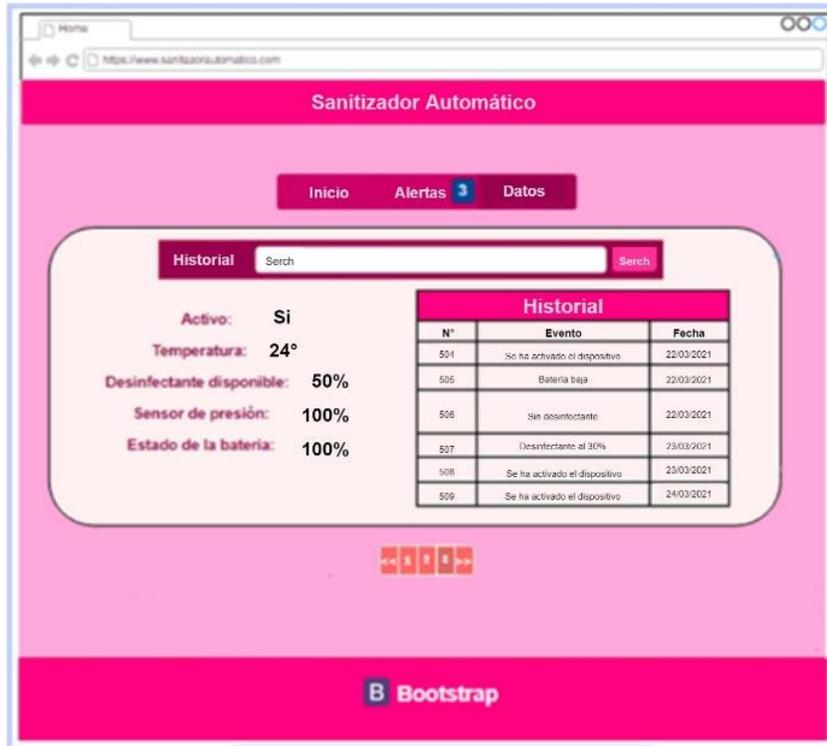


Fig. 22 Diseño vista Datos generales, historial y ubicación

Dentro de la vista relacionada se tiene un apartado de los elementos generales del dispositivo siendo los explicados en la Tabla 9, un apartado para la presentación de eventos mostrado como un historial que contiene los datos del evento y su fecha con su respectivo identificador, estos datos podrán ser filtrados utilizando la fecha registrada.

Por último, hará el uso de un mapa en el cual se podrán mostrar los datos de geoposicionamiento del dispositivo de manera visual.

3.2 Construcción del prototipo

En esta fase de desarrollo se procederá a describir y mostrar el proceso de construcción del dispositivo siguiendo los diseños descritos en la fase de diseños e

indicando la programación de ambos dispositivos, el nebulizador y el dispositivo receptor, así como el servidor en nodeJs que se encargará de enviar la información a través de internet.

En la figura Fig. 23 se muestra el esquema resultante de la implementación de cada uno de los módulos. Se realizó la conexión exitosa de los sensores de temperatura y proximidad además de lograr recibir datos de geoposicionamiento del módulo GPS y enviar mensajes a través del dispositivo SIM800L, a su vez se logró realizar el circuito para el funcionamiento del disco piezoeléctrico de 113khz realizando la nebulización.

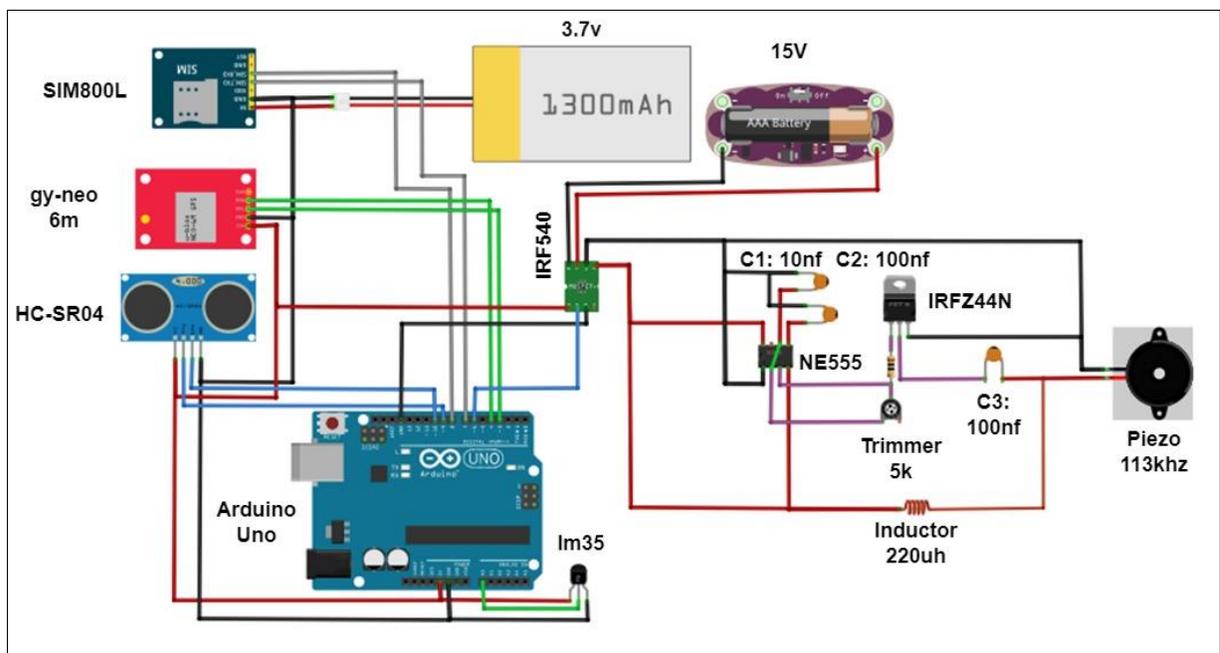


Fig. 23 Circuito general

Se realizó el diseño para la impresión de la placa PSB siguiendo el circuito general utilizando el programa de proteos del módulo de desinfección y otra placa para el Arduino uno y los sensores como se ve en la Fig. 24 y Fig. 25.

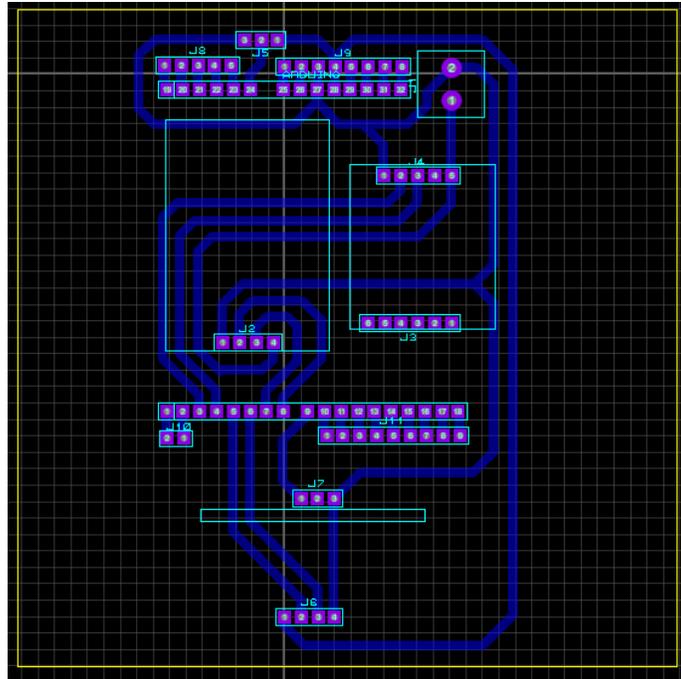


Fig. 24 PCB Arduino

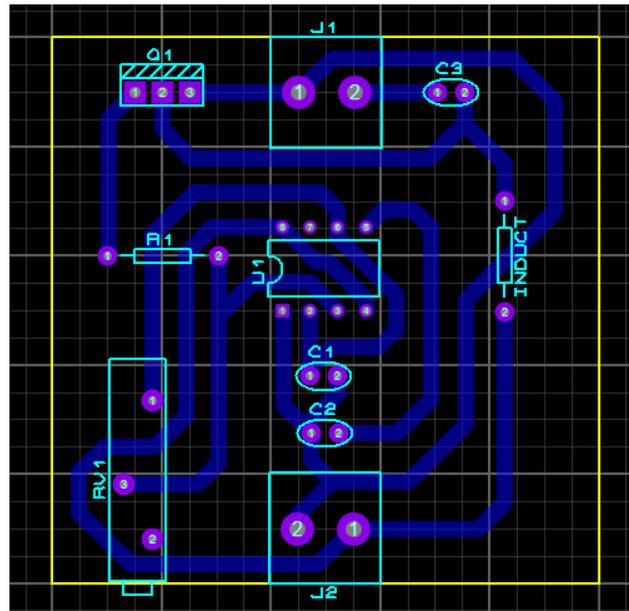


Fig. 25 PCB Nebulizador

A su vez se realizó el diseño en 3d mediante el software de Fusion 360 para la carcasa del dispositivo como se ve en la Fig. 26 considerando las medidas de cada uno de los componentes que se instalarán dentro de la carcasa.

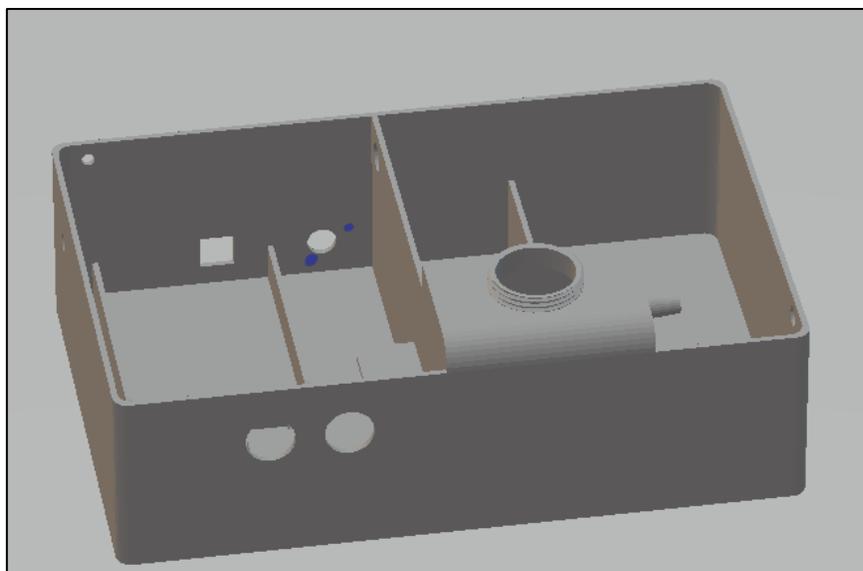


Fig. 26 Carcasa 3D

Se realizaron la programación correspondiente para el correcto funcionamiento de los sensores, la recepción de datos GPS, y el envío de estos a través del módulo SIM800L, de tal forma que el programa cumple con las siguientes tareas:

- I. Detecta la presencia del usuario mediante el sensor ultrasónico HC-SR04, establece la distancia utilizando el modelo de conversión anterior mente descrito y ajustándolo a un rango de 40cm. Si detecta al usuario se acciona el dispositivo nebulizador mandando una señal al pin de control del dispositivo.
- II. Recibir la información analógica del sensor Im35 y utiliza el modelo previamente descrito para convertir los datos de milivolts a grados centígrados.
- III. Recibe los datos de posición del módulo GPS referentes con la latitud y longitud.
- IV. Recopilará todos los datos en una cadena de texto para ser enviados mediante SMS.
- V. Cada 10 min enviará un mensaje de texto con los datos al siguiente módulo Arduino UNO

Se realizo la programación del dispositivo Arduino para la recepción de datos y envió a través de internet el cual solo recibe las siguientes tareas:

- I. Utiliza los comandos AT y configura el dispositivo sim a recepción de mensajes SMS con los comandos; AT+CMGF=1 y AT+CNMI=1,2,0,0,0.
- II. Arduino recibe la información entrante y la organiza los datos para ser enviados al internet.
- III. Se establece un puerto serial con un baud rate de 9600. Se leen los datos recibidos por el SIM800 en formato SMS y los imprime en el puerto serial COM 8.

Se realizo la programación del servidor en NodeJS para la interconexión con el microcontrolador Arduino mediante el puerto Serial COM el cual realiza las siguientes tareas:

- I. Establece una conexión mediante el puerto serial utilizando el puerto COM 8 y con un baud-rate de 9600 que es igual al establecido en el serial del microcontrolador.
- II. El servidor se mantendrá escuchando a los datos enviados por el dispositivo al puerto serial los cuales serán almacenados en una variable.
- III. Los datos recibidos serán separados por valor y adecuados para mandarlos a través de internet ya sea por medio de una petición HTTP o una conexión directa a la nube.

3.3 Desarrollo de la aplicación web, base de datos y funciones en la nube de Firebase

En esta fase se realizado el desarrollo e implementación de los elementos vistos en la obtención de requisitos. Arquitectura de solución e implementando cada uno de los diseños previamente definidos.

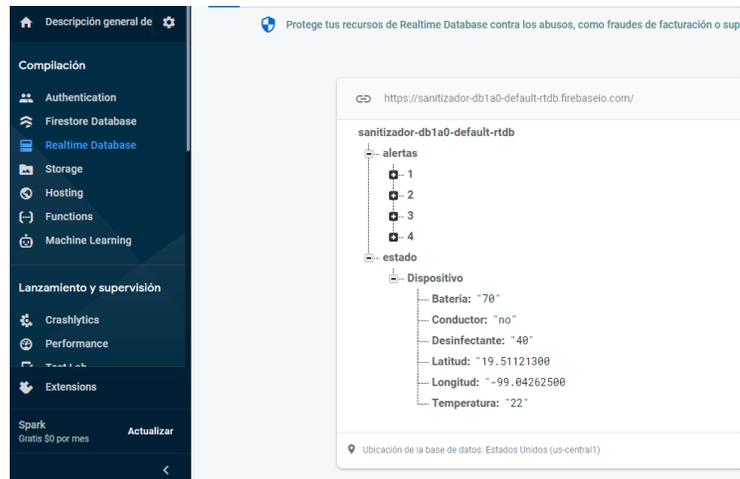


Fig. 27 Realtime Database de Firebase

Para la creación de la base de datos se creó un nuevo proyecto en firebase en el cual se implementó una base de datos en tiempo real y se realizó una estructura como se muestra en la Fig. 27. En ella definimos el valor padre Alertas el cual contendrá cada una de las alertas activas y no activas emitidas por el dispositivo, almacenando los valores siguientes valores.

- I. Estado de la alerta activa/inactiva
- II. Título del evento
- III. Descripción del evento
- IV. Fecha
- V. Hora del evento
- VI. Identificación del evento
- VII. Longitud
- VIII. Latitud
- IX. Nivel rojo/amarillo/verde

A su vez tenemos una relación de estado con el dispositivo, en donde se encuentran los datos referentes al estado actual del dispositivo, estos datos deben ser únicos y no deben replicarse por lo que se actualizan constantemente y se ven reflejados en la página en tiempo real. Los valores almacenados son:

- I. Estado de la batería 0-100
- II. Presencia del conductor Si-No
- III. Desinfectante restante 0-10
- IV. Latitud
- V. Longitud
- VI. Temperatura 0-50

Cada cierto tiempo se recibirán datos provenientes del dispositivo y estos se actualizan automáticamente por la base de datos para ser mostrados por el dispositivo, cuando se recibe una alerta esta se agrega y permanece, las alertas no son eliminadas para mantener el historial de eventos y el único campo alterado es la variable que indica si es una alerta activa.



The screenshot shows the Google Cloud Functions console interface. At the top, there are tabs for 'Panel', 'Estado', 'Registros', and 'Uso'. Below the tabs, there is a warning message: 'Protege tus recursos de Functions contra los abusos, como fraudes de facturación o suplantación de identidad.' with a link to 'Configurar la Verificación de aplicaciones'. The main content is a table with the following columns: 'Función', 'Activador', 'Región', 'Tiempo de ejecución', 'Memoria', and 'Tiempo de espera'. There are two rows of functions listed:

Función	Activador	Región	Tiempo de ejecución	Memoria	Tiempo de espera
arduinoRfidciber	HTTP Solicitud https://us-central1-arduinoRfidciber.firebaseio.com/functions/net/arduinoRfidciber	us-central1	Node.js 14	256 MB	60s
helloWorld	HTTP Solicitud https://us-central1-arduinoRfidciber.firebaseio.com/functions/net/helloWorld	us-central1	Node.js 14	256 MB	60s

Fig. 28 Cloud Functions

Para recibir los datos y almacenarlos en la base de datos sin la necesidad de la instalación de un servidor, se optó por implementar las funciones en la nube de firebase, las cuales nos permiten realizar el envío de datos mediante una petición para. Dentro de esta petición se mandarían los datos específicos de la batería, temperatura, presencia del conductor, desinfectante restante, latitud y longitud.

Cuando la cloud function recibe los datos de la petición verifica y adecua los datos para su almacenado en la base de datos en tiempo real identificando real. Para el desarrollo de la aplicación se utilizó el framework de angular por lo que se instaló el framework y se diseñó cada una de las vistas definidas Angular como framework hace

el uso de componentes, servicios, módulos y rutas para su funcionamiento y por lo que se definió la siguiente estructura de archivos.

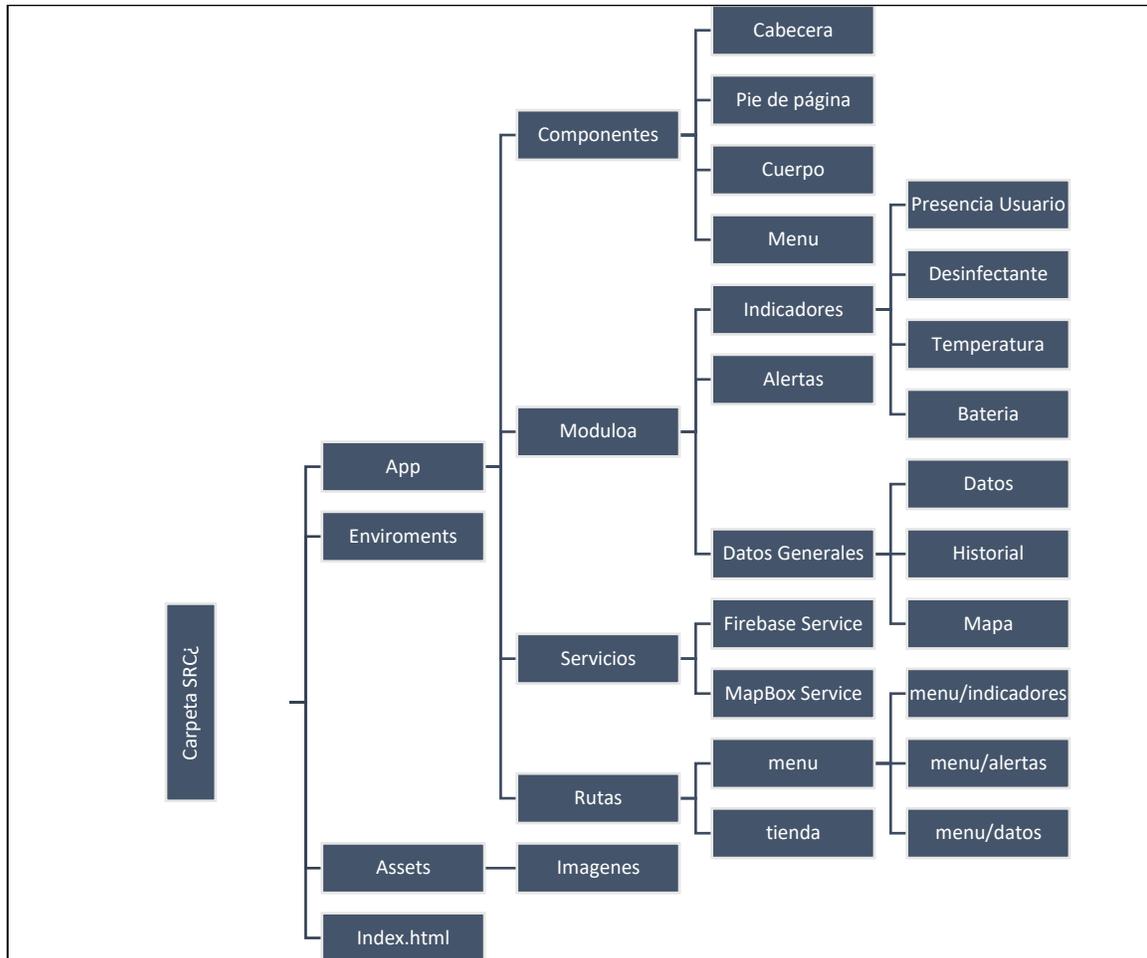


Fig. 29 Estructura de archivos

Con la estructura que se muestra en la Fig. 29 se realizaron los elementos requeridos para la aplicación siendo SRC la carpeta fuente predefinida por Angular, en donde se encuentra el archivo *index.html* y *main.ts*.

En la carpeta enviromens se encuentran archivos que contienen variables definidas para el funcionamiento del entorno, como api-keys, en la carpeta Assets se encuentran archivos como imágenes o fuentes que requiramos a lo largo del desarrollo y en la App se encuentran todas las subcarpetas relacionadas con el desarrollo de la aplicación,

siendo la carpeta Componente la que tenga aquellos componentes esenciales mostrados en la aplicación y la carpeta módulos como aquella que contiene todos los subcomponentes que le dan funcionalidad a los aspectos generales de la aplicación como las alertas los indicadores, el historial y los mapas.

Los indicadores fueron diseñados usando CSS cuyo lenguaje nos permite realizar las animaciones para las transiciones entre valores, cambiar colores, dar transparencia etc. Cada uno de los estados de los indicadores corresponde a su valor emitido por el dispositivo nebulizador y almacenado en la base de datos. Las alertas y los demás componentes se realizaron con la ayuda de Bootstrap quien brinda diseños predefinidos para algunas de las características de la página web.

Para la visualización de los mapas se hizo uso de mapbox, en donde se solicitó un token gratuito el cual te permite realizar la carga de 50 000 mapas previos al monto. Este token fue implementado en la página web junto con su librería de representación de mapas compatible con angular.

```
mapa(lng:any,lat:any): void {
  (mapboxgl as any).accessToken = environment.Mapbox.mapbox_key ;
  this.map = new mapboxgl.Map({
    container: 'mapa',
    style: 'mapbox://styles/mapbox/streets-v11',
    center: [lng,lat],
    zoom: 15
  });
  new mapboxgl.Marker()
    .setLngLat([lng,lat])
    .addTo(this.map)
}
```

Fig. 30 Configuración de MapBox

El mapa solo requiere de los atributos de latitud y longitud para funcionar, pero además el módulo de MapBox permite la configuración del mapa para poner marcadores, establecer el zoom y el centro del mapa por lo que establecimos la configuración que se muestra en la Fig. 30.

3.4 Pruebas de funcionamiento del dispositivo y la aplicación web.

Para lograr comprobar el funcionamiento de cada uno de los elementos se realizarán pruebas para determinar su funcionamiento, para ello se pondrá a prueba el dispositivo nebulizador en conjunto con el dispositivo receptor a lo largo de una semana en donde se estará accionando constantemente el dispositivo nebulizador y enviando datos al dispositivo receptor. En general se realizarán pruebas de funcionamiento a los siguientes elementos:

- I. Dispositivo nebulizador
 - a. Fiabilidad del sensor de proximidad
 - b. Fiabilidad del sensor de temperatura
 - c. Fiabilidad de los datos de ubicación
 - d. Fiabilidad del dispositivo nebulizador
- II. Dispositivo receptor
 - a. Datos recibidos
 - b. Conversión de datos por el servidor en nodeJS

Las pruebas que se realizarán al dispositivo nebulizador serán las siguientes:

- I. Se realizará el envío de datos durante una semana para comprobar la fiabilidad a la hora de enviar la información y comprobar los datos enviados y recibidos determinando la cantidad de datos perdidos,
- II. Se comprobará la fiabilidad de acción del sensor de proximidad de detección al usuario al realizar las detecciones en el rango estimado por la programación del dispositivo. Se comprobará el rango real de activación y la fiabilidad de la aplicación.
- III. Se comprobará la fiabilidad del sensor de temperatura al realizar cinco mediciones en cada día durante una semana y las mismas serán comprobadas en relación con las obtenidas por un termómetro atmosférico que se encontrará en el mismo entorno que el dispositivo.

- IV. Se realizará la fiabilidad de las baterías utilizadas, midiendo los tiempos de carga y descarga de las baterías que alimentan al módulo de comunicación y al Arduino uno con sus sensores. A su vez se revisará el tiempo de consumo del dispositivo Arduino al utilizar sus sensores y al no usarlos.
- V. Se realizarán pruebas a la eficiencia de nebulización comprobando la distancia a la que llega el vapor nebulizado, el área a la que llegó el vapor, la cantidad de disparos realizados a lo largo de la semana, la cantidad de disparos que se pueden realizar con el contenedor y el líquido consumido por cada disparo.
- VI. Se comprobará la fiabilidad de los datos obtenidos por el módulo GPS realizando una medición de los datos GPS en un recorrido de 2 kilo metros, el tiempo de actualización de los datos y el margen de error de lectura.

Las pruebas que se realizarán al dispositivo receptor serán las siguientes:

- I. Se realizará la recepción de los datos durante el periodo de una semana, comprobando los datos emitidos por el dispositivo al enviar mensajes hacia un dispositivo móvil y al dispositivo receptor y así determinar la recepción de los datos.
- II. Se realizarán pruebas al servidor de nodej al enviar datos y comprobar la coincidencia entre los datos enviados y los datos extraídos de la cadena por el servidor

En cuanto a la aplicación web se realizarán pruebas a cada uno de los elementos funcionales de la aplicación, haciendo énfasis en las muestras de datos, el almacenamiento y el tiempo de respuesta de cada uno de los componentes.

Para comprobar la recepción de datos en conjunto con el dispositivo se tomará un tiempo de prueba de una semana en los cuales el dispositivo nebulizador inteligente automotriz germicida del SARS-CoV-2 estará enviando información a las funciones en la nube de firebase, en este periodo se hará el registro del total de datos enviados por el dispositivo que fueron recibidos y almacenados por el sistema de monitoreo. A

su vez se hará el ingreso de datos manuales para la comprobación del funcionamiento del sistema en situaciones específicas.

Para el funcionamiento de las funciones en la nube y la base de datos de firebase se realizará lo siguiente:

- I. Durante una semana se hará el envío de mensajes utilizando el dispositivo nebulizador y su sistema de envío de mensajes mediante SMS y el servidor.
- II. De los mensajes enviados se verificarán cuantos mensajes llegaron y se almacenaron o actualizaron correctamente verificando que los datos emitidos y recibidos sean los mismos.
- III. Según los datos enviados se verificarán las alertas emitidas por las funciones en la nube y se buscarán alertas faltantes o alertas que no debieron ser emitidas.
- IV. Se medirá el tiempo de respuesta entre la llegada de la alerta y la actualización de la base de datos.

Para las pruebas de funcionamiento sobre los datos de estado e indicadores mostrados por la aplicación web se realizará lo siguiente:

- I. Se hará el envío de 100 mensajes en los cuales se utilizarán todos los valores posibles que pueden tener cada uno de los indicadores de estado.
- II. Se comprobará que la información almacenada y la información mostrada por los indicadores sea la adecuada
- III. Se comprobará la velocidad de actualización de los indicadores, así como la fluidez de la animación.

Para las pruebas de funcionamiento sobre las alertas mostradas por la aplicación web se realizará lo siguiente:

- I. Se comprobará que las alertas almacenadas en la base de datos y las alertas mostradas por la aplicación sean las adecuadas.

- II. Se realizará el envío de mensajes utilizando datos para accionar cada una de las alertas.
- III. Se medirá el tiempo de reacción de cada alerta

Para las pruebas relacionadas con el historial de eventos y la representación de datos geodésicos se realizarán las siguientes pruebas:

- I. Se comprobarán los datos mostrados por el historial y los almacenados en la base de datos.
- II. Se comprobará que el total de datos registrados por el historial sean los mismos que la base de datos
- III. Se realizará una búsqueda utilizando las fechas registradas por la base de datos
- IV. Para la presentación de datos geodésicos se hará una comparación entre sus datos mostrados y los mostrados por la plataforma de Google maps utilizando las mismas coordenadas.
- V. Se medirá el tiempo de respuesta de la aplicación web para actualizar el mapa.

Capítulo IV

Análisis e interpretación de resultados

En este capítulo se mostrarán los resultados obtenidos a partir del desarrollo de los puntos anteriores. Se mostrará el resultado de todas las implementaciones realizadas para generar el dispositivo nebulizador en conjunto con la aplicación web además de las pruebas de funcionamiento que se realizaron para determinar la calidad y fiabilidad del sistema.

4.1 Prototipo final

Como resultado de los procesos de análisis de requisitos, diseño del prototipo y construcción se lograron construir los prototipos para el nebulizador inteligente automotriz germicida del SARS-CoV 2 y el dispositivo receptor de mensajes SMS como se muestran en las Fig. 31 y Fig. 32 respectivamente.

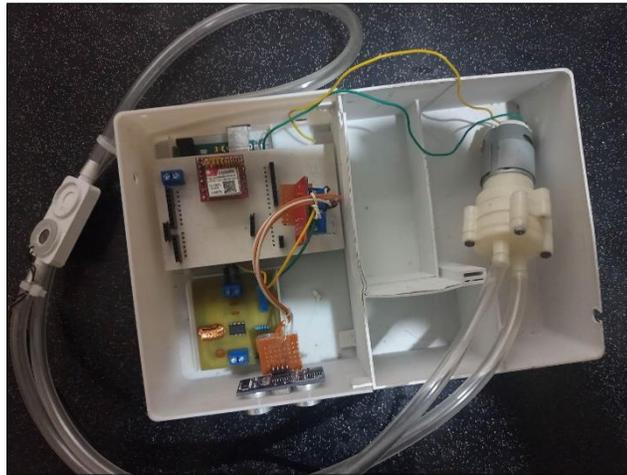


Fig. 31 Prototipo de dispositivo nebulizador

Entre las funcionalidades logradas por el dispositivo nebulizador de la Fig. 31 se encuentran:

- I. La lectura de objetos presentes a menos de 30 cm del sensor de proximidad.
- II. Las lecturas de temperatura obtenidas por del sensor de temperatura y la recepción de los datos de ubicación proporcionados por el módulo GPS
- III. La nebulización de un líquido mediante el uso del disco piezoeléctrico Mist Maker y su accionamiento a través del sensor de proximidad.
- IV. La recopilación y el envío de datos a través de mensajes SMS con el módulo GSM SIM800L.

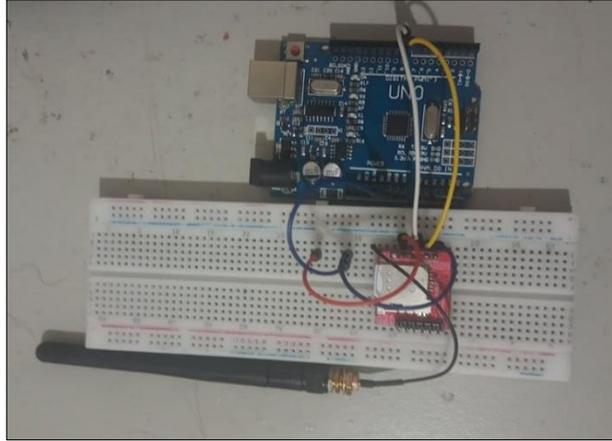


Fig. 32 Prototipo de dispositivo receptor

Entre las funcionalidades logradas por el dispositivo receptor de mensajes SMS visto en la Fig. 32 y su servidor nodeJS se encuentran:

- I. La recepción de mensajes SMS mediante el módulo SIM800L
- II. La conexión entre el dispositivo receptor y el servidor de nodeJS usando el puerto serial del Arduino y realizando la transferencia de información.
- III. El manejo de la cadena de datos para separarlas en variables manejables y prepararlas para su envío a internet.

4.2 Aplicación web final

Como resultado del desarrollo de la aplicación web que se realizó para el componente de indicadores siguiendo los diseños de la Fig. 20 tal se obtuvo la vista de la Fig. 33 que es capaz de representar los datos en tiempo real siendo estos consultados y actualizados a través de la aplicación de firebase realtime database y utilizando graficas para representar los datos.

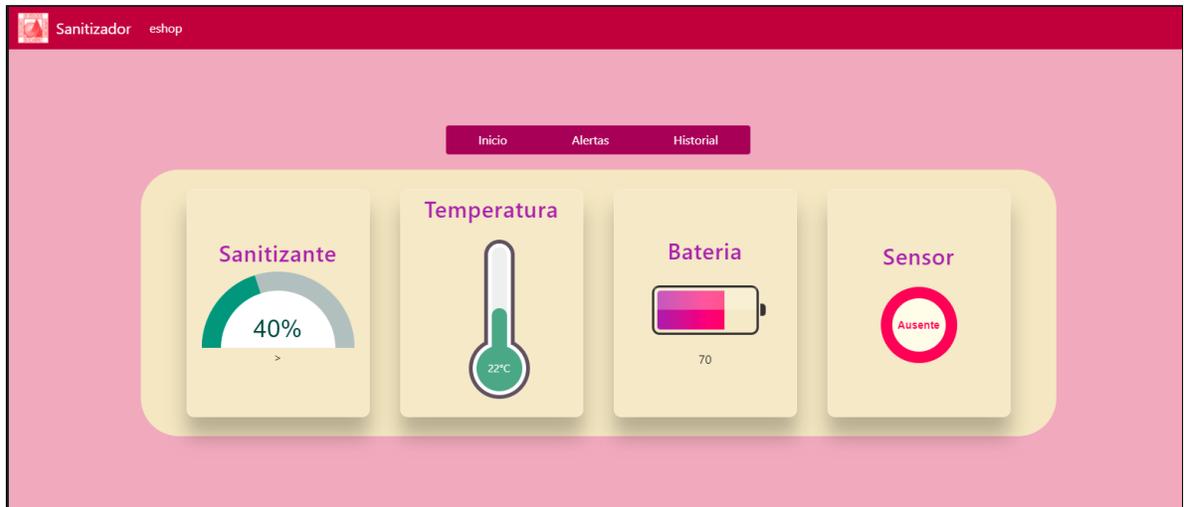


Fig. 33 Vista final Indicadores

Cada uno de los datos representados en la vista pueden ser actualizados y la vista podría actualizar los indicadores en tiempo real sin la necesidad de recargar la página realizando animaciones que fueron programadas mediante CSS.

En cuanto al componente de alertas que se muestra en la Fig. 34 también fue realizado siguiendo los diseños de la Fig. 21 cada una de las alertas representa un registro activo de la base de datos, estas a su vez se actualizan en tiempo real siempre mostrando las alertas activas en pantalla. Además, el usuario es capaz de eliminar una alerta en su respectivo botón de borrar y dicho evento será reflejado en la página y en la base de datos de manera automática y sin necesidad de refrescar la página.



Fig. 34 Vista final de alertas

Por último, la vista general de datos, historial y ubicación mostrados en la Fig. 35 muestra los datos en bruto almacenados en la base de datos en tiempo real, estos no incluyen las alertas y tienen la capacidad de cambiar sus valores en tiempo real sin necesidad de actualizar la página.

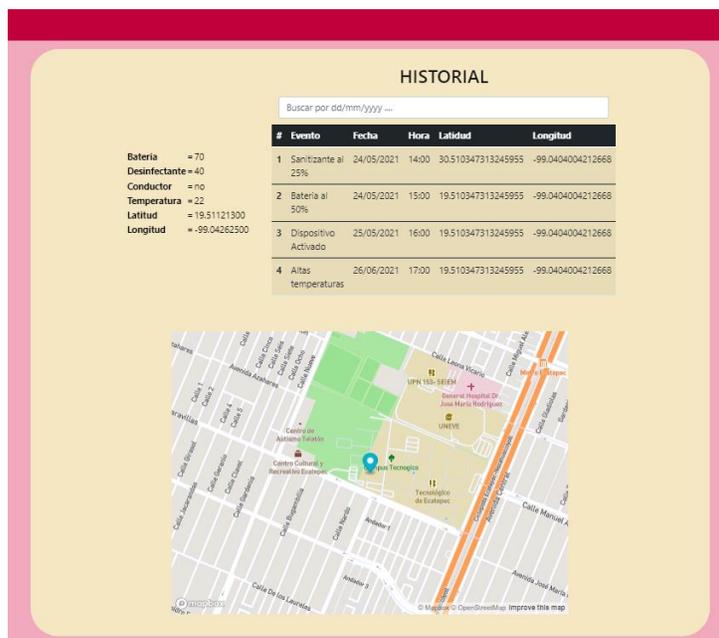


Fig. 35 Vista final datos generales, historial y mapa

El historial de eventos muestra cada uno de los eventos sucedidos por el dispositivo. Estos eventos son extraídos desde la base de datos y actualizados mostrando el número de evento, el título del evento, su fecha y hora y la ubicación en términos de latitud y longitud. Pudiendo realizar una búsqueda indicando el título, la fecha u el número de evento.

Por último, el mapa muestra la ubicación actual del dispositivo siguiendo las coordenadas marcadas por el estado del dispositivo y siendo extraído de la base de datos. Dentro de esta vista el usuario puede ver la ubicación e interactuar con el mapa, pero no puede reposicionar e indicador o agregar uno nuevo. Cabe notar que cada uno de los datos se actualiza en tiempo real y no pueden ser modificados por el usuario.

4.3 Resultados de las pruebas de funcionamiento del prototipo nebulizador y la aplicación web

Se desarrollo una página web funcional con la capacidad de recibir alertas, mostrar datos almacenados en la nube de firebase y actualizar los mismos en tiempo real. Para poder comprobar el correcto funcionamiento de la aplicación web se realizaron pruebas en un periodo de una semana a los componentes esenciales como la representación de datos en tiempo real, actualización de los datos, manejo de alertas y representación de los datos de geoposicionamiento.

El primer aspecto que se comprobó fue la capacidad de recepción de datos en el periodo de una semana, en el cual a lo largo de una semana se enviaron 500 mensajes desde el dispositivo nebulizador automático germicida del SARS-COV-2, de los cuales 477 fueron recibidos correctamente por el servidor y enviados a la base de datos, estos datos a su vez se actualizaron en su campo correspondiente.



Fig. 36 Relación de datos enviados y recibidos

En la Fig. 36 se muestran los resultados de la prueba de envío de datos en donde muestra la pérdida de 33 datos, se determinó que al recibir más de un mensaje en un lapso menor a un segundo el servidor no registra correctamente las alertas correspondientes siendo registrados únicamente los últimos datos enviados en ese corto periodo de tiempo, sin embargo para los datos enviados con lapsos mayores no se encontraron problemas de registro de alertas y teniendo un tiempo de respuesta menor a un segundo desde las cloud functions y todos los datos recibidos correspondían con los datos emitidos.

Tabla 12 muestra de 10 datos de la relación entre datos recibidos y mostrados

Datos Recibidos				Datos Mostrados			
Desinfectante	Temperatura	Batería	Proximidad	Desinfectante	Temperatura	Batería	Proximidad
100	24	100	50	100	24	100	50
99	22.1	97	10	99	22.1	97	10
98	25.2	96	50	98	25.2	96	50
97	23.1	93	10	97	23.1	93	10
96	22.5	91	50	96	22.5	91	50
95	22.4	95	10	95	22.4	95	10
94	24	94	50	94	24	94	50
93	24.4	93	10	93	24.4	93	10

92	22.5	92	50	92	22.5	92	50
91	23.2	91	10	91	23.2	91	10

El segundo aspecto que se comprobó fue la fiabilidad de los indicadores comprobando los diversos estados posibles que un indicador puede tener al ingresar todos sus estados posibles para verifica su comportamiento. En la Tabla 12 tenemos 10 muestras de 100 realizadas para cada estado. Cada una de las muestras coincidió con la representada en el indicador y la almacenada como se muestra en la Fig. 37 donde se representó las 10 muestras. De igual forma al realizar la actualización de los datos se comprobó que las animaciones transcurren en tiempo real sin necesidad de recargar la aplicación y se efectúan de manera fluida realizando el cambio de estado de manera automática.

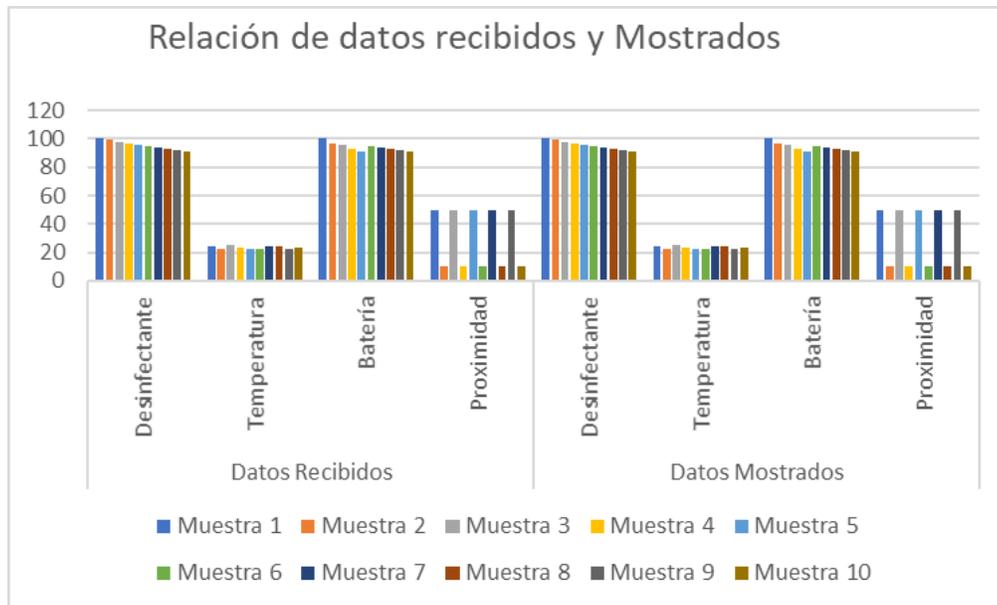


Fig. 37 Relación entre datos recibidos y mostrados

El tercer aspecto al que se le realizaron las pruebas fue al componente de alertas se comprobó la fiabilidad de cada tipo de alerta registrada en la Tabla 10 en donde se

detalla cada una de las alertas. Como se ve en las siguientes figuras se realizó una prueba al ingresar los datos correspondientes para cada tipo de alerta, esta se visualizó correctamente en la vista de alertas al tener su color, título y breve descripción correspondiente. Dichas alertas se pudieron remover correctamente de la pestaña actualizando su estado correspondiente en la base de datos.

Para la primera prueba de alerta se mandó la siguiente cadena de datos para accionar las alertas correspondientes al accionamiento del dispositivo, batería cargada y contenedor del desinfectante a 100% de su capacidad. Las alertas accionadas se muestran en la Fig. 38.

- I. batería = 100
- II. sanitizante = 100
- III. sensor = si
- IV. temperatura = 24.2
- V. longitud = -99.046720
- VI. latitud = 19.502127
- VII. fecha = 17/01/2022
- VIII. hora = 03:16:10



Fig. 38 Primer prueba de alertas

Para la segunda prueba se mandaron los siguientes datos correspondientes a las altas temperaturas del dispositivo, el contenedor de desinfectante o la batería están

por agotarse y la falla en los datos de posicionamiento, las alertas que se accionaron en la página web se muestran en la Fig. 39 .

- I. batería = 100
- II. sanitizante = 38
- III. sensor = no
- IV. temperatura = 51
- V. longitud = 0.0
- VI. latitud = 0.0
- VII. fecha = 17/01/2022
- VIII. hora = 03:25:08



Fig. 39 Segunda prueba de alertas

Para la tercera prueba se mandaron los datos referentes a la falta de desinfectante en el dispositivo y la falta de energía en la batería. Los siguientes datos enviados accionaron las alertas mostradas en la Fig. 40.

- I. batería = 2
- II. sanitizante = 1
- III. sensor = no
- IV. temperatura = 23
- V. longitud = -99.046720
- VI. latitud = 19.502127
- VII. fecha = 17/01/2022
- VIII. hora = 03:30:39



Fig. 40 Tercer prueba de alertas

Por último, se suspendió el envío de datos para accionar la alerta referente a la falta de conexión del dispositivo, la cual se actualiza al no recibir datos por 20 minutos o más, la alerta accionada se muestra en la Fig. 41.



Fig. 41 Última prueba de alertas

El último aspecto funcional al que se le realizaron las pruebas fue al componente de historial de eventos y representación de datos geodésicos. En relación con los datos mostrados con la tabla de los 477 datos obtenidos se obtuvieron sus 355 alertas con su respectivo evento, estos datos fueron correctamente almacenados en una tabla que muestra 5 elementos como se ve en la Fig. 42. También se comprobó que los datos mostrados por el historial sean los almacenados en la base de datos como se ve en la Fig. 43.

#	Evento	Fecha	Hora	Latitud	Longitud
351	Se detectan altas temperaturas	17/01/2022	03:31:44	19.502127	-99.046720
352	No hay conexión con el dispositivo	17/01/2022	03:31:44	19.502127	-99.046720
353	El contenedor de desinfectante se encuentra por debajo de su 40% de capacidad	17/01/2022	03:31:44	19.502127	-99.046720
354	La batería se encuentra por debajo del 20%	17/01/2022	03:31:44	19.502127	-99.046720
355	Batería 100% cargada	17/01/2022	03:31:44	19.502127	-99.046720

< Anterior 1 ... 67 68 69 70 **71** Siguiente >

Fig. 42 Datos del historial de eventos

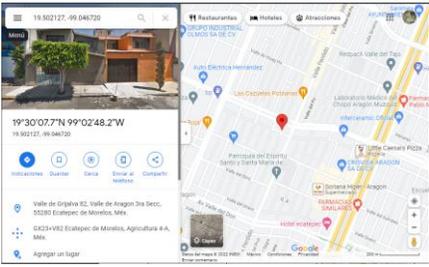
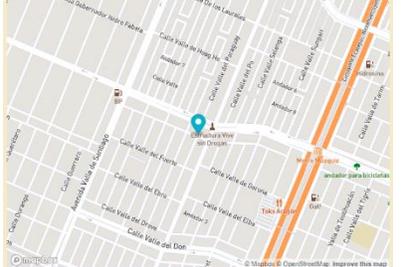
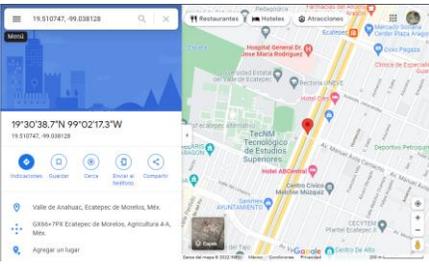
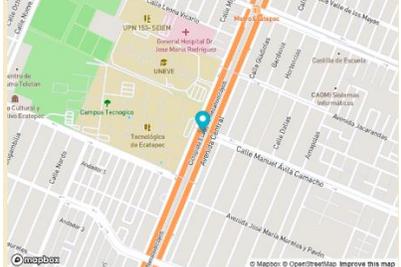
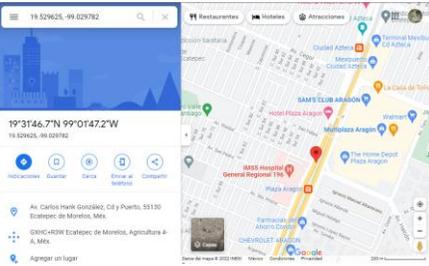
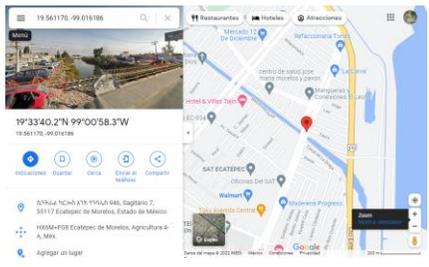


Fig. 43 Últimas alertas registradas por la base de datos

Por último, se realizó el seguimiento de los datos geodésicos utilizando el servicio de MapBOX, se enviaron datos geodésicos en formato de latitud y longitud para poder representar dicha posición en el mapa como se ve en la Tabla 13. El mapa mostró la ubicación del prototipo en tiempo real y su tiempo de actualización fue entre 3

segundos como mínimo y 5 como máximo. Estos datos fueron comparados con las ubicaciones obtenidas por las mismas coordenadas en Google maps.

Tabla 13 Comparación de registro de datos de ubicación entre google maps y la aplicación

Coordenadas [latitud, longitud]	Ubicación de Google Maps	Ubicación de la aplicación web
[19.502127, -99.046720]		
[19.510747, -99.038128]		
[19.529625, -99.029782]		
[19.561170, -99.016186]		

Al realizar la prueba podemos denotar la precisión para mostrar los datos y la facilidad de manejo, ya que se podía interactuar con el mapa realizando acercamientos y alejamientos además de moverlo mediante el cursor.

4.3.1 Resultados de las pruebas al envío de datos

Se realizaron 500 envíos en el transcurso de la semana estos datos contenían información del funcionamiento del dispositivo y fueron mandados desde el dispositivo nebulizador y recibidos por el dispositivo receptor. Al realizar la prueba la mayoría de los datos fueron recibidos y enviados correctamente por el servidor teniendo que de los 500, 477 fueron correctamente recibidos y solo 23 se perdieron en la transferencia. Al finalizar la prueba se determinó que la mayoría de las pérdidas se vieron influenciadas por el agotamiento de la batería que le suministraba al módulo SIM800I ocasionando problemas de comunicación.



Fig. 44 Resultados de datos enviados y recibidos

A su vez se realizó el envío de 100 mensajes en un recorrido corto de 2 kilómetros para medir las posibles fallas con el dispositivo en movimiento y como se ve en la Fig. 44 se determinó que de los 100, 93 llegaron satisfactoriamente siendo difícil determinar

la causa de la falla al no tener un aspecto evidente para la razón de los problemas de comunicación.

4.3.2 Resultados de las pruebas al sensor de proximidad y temperatura

En cuanto al sensor de proximidad se intentó medir la distancia y la fiabilidad de detección encontrando una alta precisión a la hora de detectar los objetos enfrente del sensor a una distancia de 30cm además se encontraron problemas con algunos tipos de objetos como telas o superficies blandas. Al realizar las pruebas se encontraron problemas en los cuales el sensor emitía señales erradas de detección sin tener objetos enfrente o dentro de su rango de detección. Como se ve en la Fig. 45 se realizaron un total de 50 pruebas de detección de las cuales 39 se detectaron correctamente 14 fueron señales enviadas por error y 11 no fueron detectadas teniendo un total de 53 señales de detección enviadas por el sensor.

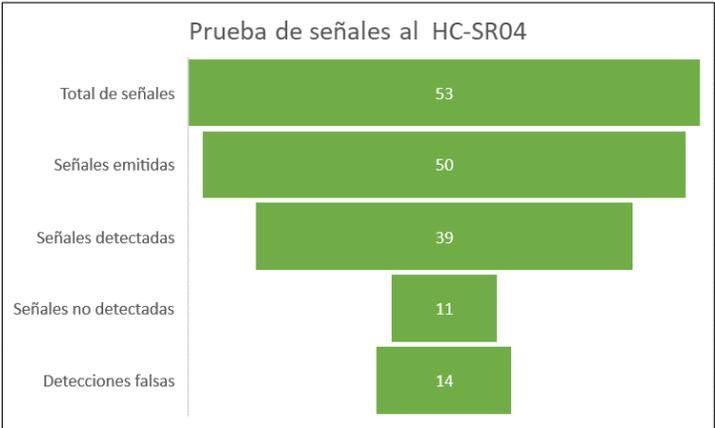


Fig. 45 Resultados de pruebas al sensor de proximidad

Para las pruebas realizadas al sensor de temperatura se registraron algunos de los datos de temperatura en el transcurso de cada uno de los días a lo largo de la semana utilizando el sensor de dispositivo y un termómetro ambiental analógico con los que

conseguimos los datos de las tablas Tabla 14 y Tabla 15 en donde se colocaron ambos termómetros en el mismo ambiente.

Tabla 14 Pruebas de medición del sensor de temperaturas 1

	Lunes		Martes		Miércoles		Jueves	
Hora	Sensor	Muestra	Sensor	Muestra	Sensor	Muestra	Sensor	Muestra
12:00	24.4	24.7	22	22.1	23.1	23.2	22.4	22.3
14:00	23.4	23.6	24.9	25.2	24.3	24.2	24	23.7
16:00	20	19.5	23	23.1	24.3	24.3	24.4	24.6
18:00	17.5	17.5	22.3	22.5	22	22.1	22.5	22.5
20:00	16.7	16.8	18.1	17.9	22	19.8	16.6	18.6

Tabla 15 Pruebas de medición del sensor de temperaturas 2

	Viernes		sábado		Domingo	
Hora	Sensor	Muestra	Sensor	Muestra	Sensor	Muestra
12:00	21.5	21.4	22	21.9	21	20.8
14:00	23.3	23.2	23.5	23.5	22.4	22.3
16:00	22	21.8	22.5	22.4	21	20.8
18:00	20	20.2	19.2	19.3	20	19.9
20:00	18.6	18.6	18.1	17.9	16.9	16.8

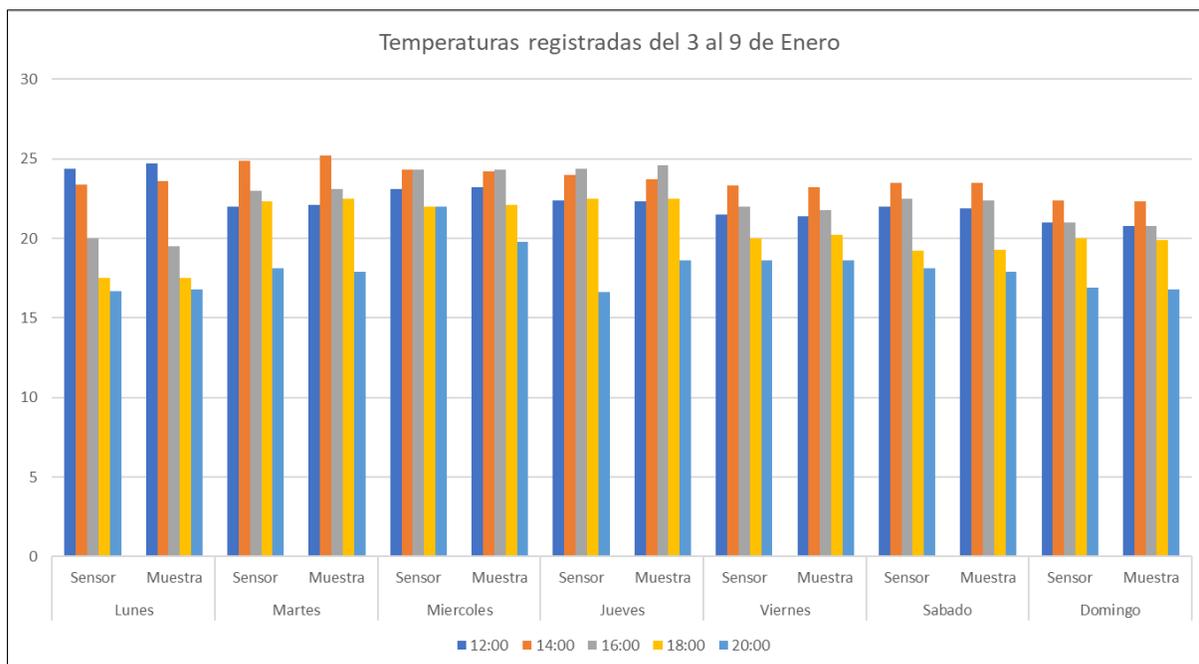


Fig. 46 Grafica de relación de temperaturas registradas a lo largo de una semana

La tabla nos muestra una comparativa de los datos del sensor y del termómetro ambiental. Al comparar los datos observamos una gran similitud entre los datos obtenidos por el sensor instalado en el prototipo y el termómetro ambiental como lo muestra la gráfica de la Fig. 46. .

4.3.3 Resultados de las pruebas al consumo energético

Para las pruebas relacionadas al consumo energético a lo largo de una semana los dos dispositivos estuvieron conectados a baterías nuevas de litio de 2200mAh a 3.7 v conectados directamente a los módulos de comunicación SIM800 los cuales requieren de dicha alimentación externa, en el transcurso se comprobó que el dispositivo nebulizador comenzó a presentar problemas de conectividad al tercer día, debidos a insuficiencia en el suministro de energía teniendo que recargar la batería, sin embargo el valor de la batería referente al dispositivo receptor no presento problema hasta el cuarto día. Teniendo que recargar dos veces la batería del dispositivo nebulizador a lo largo de su semana de uso.

En cuanto al consumo del Arduino Uno y sus sensores se utilizaron pilas de 9v zinc-carbono no recargable para la alimentación con una capacidad de 300mAh.

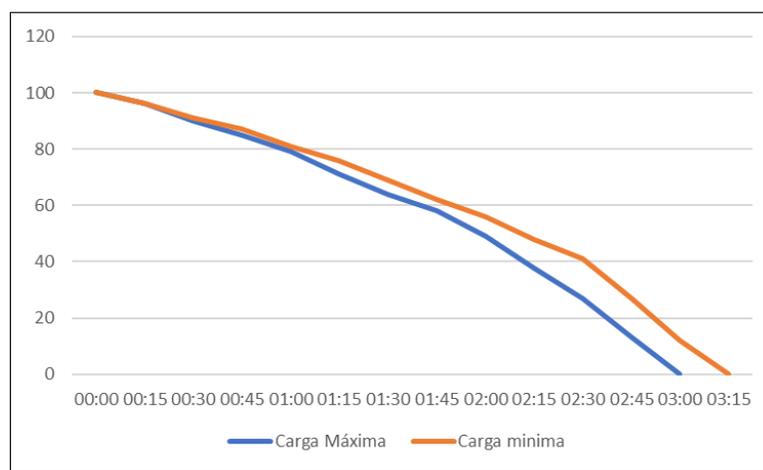


Fig. 47 Grafica de comparación de consumo energético

Como se ve en la Fig. 47 se ve la diferencia de tiempo de la batería utilizando al dispositivo en carga con los sensores de temperatura, proximidad y el módulo GPS recibiendo datos de los satélites en comparación del dispositivo sin carga, en donde solo el sensor de temperatura manda señal, con le modulo GPS sin conexión a los servidores y sin señales provenientes del sensor de proximidad. Se obtuvo una ligera disminución en el tiempo de consumo siendo una diferencia de entre 15 y 30 minutos con un tiempo de descarga máximo de 3 horas con 15 minutos aproximadamente y un consumo mínimo de entre 3 horas aproximadamente.

4.3.4 Resultados de las pruebas del componente de nebulización

Para el componente de nebulización se comprobó la eficiencia de su funcionamiento del nebulizador realizando las mediciones de área de aspersion colocando el dispositivo frente a un vidrio y accionando el componente de nebulización. Como resultado tenemos que mandamos 50 señales de aspersion de las cuales la mínima distancia alcanzada fueron de 8 cm con una cobertura de 10 cm de diámetro y una distancia alcanzada máxima de 43 cm, con un diámetro de cobertura de aproximadamente de 17cm. Al realizar las pruebas dentro de un ambiente cerrado sin corrientes de aire la nube salía de manera recta y consistente pudiendo realizar las mediciones de manera más detallada, pero al realizar las pruebas en un ambiente abierto la nube se veía altamente influenciada por las leves corrientes de aire pudiendo disipar la misma con una gran facilidad.

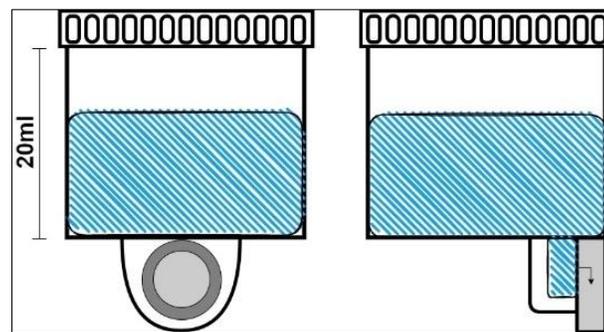


Fig. 48 Contenedor de desinfectante

Para probar la cantidad de líquido utilizado en una nebulización se utilizó el contenedor compatible con el disco piezoeléctrico de la Fig. 48 con capacidad de 20ml de líquido desinfectante. Al realizar la prueba con un disparo sostenido el líquido se agotaba al minuto 4, con los disparos realizados por el dispositivo se obtuvo que se lograron 60 disparos hasta que se agote los 20ml del contenedor.

4.3.5 Resultados de las pruebas de datos de ubicación y el dispositivo receptor

Se realizaron pruebas de conectividad durante la semana de prueba del dispositivo y leyendo los datos de posicionamiento recibidos por el módulo GY-NEO-6m recibiendo datos de posición cercanos a la ubicación del dispositivo. Con una variación aproximada de 5m al realizar las pruebas encontramos una dificultad para realizar la conectividad con los satélites ya que se requería estar en un espacio moderadamente abierto tardando en realizar una conexión de aproximadamente 5 min. Se intento registrar el movimiento de dispositivo en una ruta de 2km para ver los datos recibidos pero el dispositivo se desconectaba de los satélites muy fácilmente marcando solo tres puntos a través de la ruta.

En cuanto a las pruebas realizadas al dispositivo receptor de mensajes SMS se hizo el envío manual de cadenas que serán procesadas a la aplicación, para ello utilizamos las cadenas mostradas en la Tabla 16.

Tabla 16 Resultados de proceso de extracción de datos

Cadena	Variables extraídas
'bateria=99&sanitizante=98&sensor=si&temperatura=50&longitud=-99.046720&latitud=19.502127'	batería = 100 sanitizante = 98 sensor = si temperatura = 50 longitud = -99.046720 latitud = 19.502127

bateria=76&sanitizante=75&sensor=2i&temperatura=22&longitud=-92.116486&latitud=13.52170'	batería = 100 sanitizante = 75 sensor = 2i temperatura = 22 longitud = -92.116486 latitud = 13.52170
'bateria=24&sanitizante=2&sensor=no&temperatura=21&longitud=-98.675186&latitud=21.32173	batería = 100 sanitizante = 2 sensor = no temperatura = 21 longitud = -98.675186 latitud = 21.32173

Como muestra la Tabla 16, cuatro de los veinte datos enviados al servidor fueron extraídos correctamente las variables aun si se mandaban algunos datos con errores como en la segunda y tercera prueba esto indica que la función para dividir los datos de la cadena opera de manera adecuada.

Capítulo V

Conclusiones y recomendaciones

En este capítulo se presentarán las conclusiones y recomendaciones obtenidas a partir de la etapa de pruebas al haber desarrollado un prototipo de dispositivo nebulizador automático con aplicación web de monitoreo y ponerlo en operación.

El prototipo de dispositivo nebulizador automático para el SARS-CoV-2 fue capaz de cumplir con sus tareas esenciales de nebulización, recopilación, envío y recepción de los datos y funcionamiento automático definidos en los objetivos de desarrollo del prototipo. En cuanto a la aplicación de monitoreo fue capaz de mostrar los datos de su funcionamiento en tiempo real haciendo el uso de alertas, indicadores, tablas y mapas para mostrar los datos generales del dispositivo. Para la realización del proyecto se utilizaron tecnologías de vanguardia como el framework de angular en conjunción con el servicio de MapBox y tecnologías en la nube como la implementación de firebase y sus respectivas herramientas.

Se realizaron pruebas al sistema de monitoreo y funcionamiento del dispositivo en el lapso de una semana en funcionamiento con el dispositivo y se lograron obtener 477 datos. Las pruebas nos indican gran funcionalidad en cada una de las características de la aplicación de monitoreo, al realizar sus tareas de representación y actualización de los datos tanto en la aplicación web como en la base de datos y el servidor, salvo la representación de los datos de geoposicionamiento, cuyo tiempo de presentación era más lenta en comparación a los otros componentes de la aplicación. Los indicadores y las alertas funcionan de manera esperada, así como la vista de los datos, en cuyo caso la tabla logra mostrar cada uno de los eventos realizados en la prueba con su respectiva paginación.

Además, las pruebas de funcionamiento nos mostraron que algunas características deben ser mejoradas para su correcta implementación dentro del interior del automóvil como lo es la optimización del sistema de nebulización, ya que debe realizar una aspersion del líquido de una manera más consistente y cuya nube no se debe de ver alterada por leves corrientes de aire, además de facilitar el ajuste para lograr la frecuencia óptima de operación. El sensor de temperatura funciona adecuadamente mostrando los datos de temperatura con un bajo nivel de error, pero en cuanto al sensor de proximidad funciona, pero presentaba inconsistencias en las lecturas de proximidad por lo que se puede optimizar o utilizar una alternativa para la detección del usuario. Sin embargo, el funcionamiento de los módulos de comunicación mediante SMS mostró que es una forma fiable de envío de datos teniendo pocas perdidas de

datos justificadas por problemas de energía, por lo tanto, se comprueba que es un método de transmisión que tiene ventajas frente a otros tipos de transmisión de datos al tener mayor cobertura con la desventaja de un aumento en los costes de implementación y utilización. En cuanto a la aplicación web encontramos algunos problemas para la obtención de datos geodésicos por parte del dispositivo ya que el componente GPS no lograba conectarse a los satélites en ciertas circunstancias y su reconexión podía tardar hasta 5 min además se encontraron pérdidas al recibir los datos del dispositivo perdiendo de esta manera 23 mensajes por problemas de conexión o un lapso muy corto entre mensajes

Estas pruebas denotaron algunas fallas de optimización del dispositivo, pero también muestran el potencial de esta tecnología y su viabilidad para resolver la problemática planteada en la solución

Bibliografía

Firebase inc. (15 de 1 de 2021). *Firestore realtime database*. Obtenido de <https://firebase.google.com/docs/database?hl=es>

Ábila Megía, Ó. (2021). *Computación en la nube*. Depto. de ingeniería Eléctrica. UAM-I.

Álvarez Buylla, E. (22 de Junio de 2020). Comunicado 161. *Fueron aprobadas las primeras 102 propuestas de la Convocatoria Covid-19*. Ciudad de México, México: Comunidad del Conacyt.

Apaza Condori, D. (2010). *MICROCONTROLADORES PIC*. San Francisco: UASF.

Ardino. (2021). *Arduino/docs*. Recuperado el 8 de Diciembre de 2021

Armijos Llerena, J. L. (2000). *CONDICIONAMIENTO DE SENSORES DE ULTRASONIDO PARA APLICACIONES DE ROBOTICA MOVIL*. Quito.

Asini, N. (2021). *Sensor de temperatura lm35*. Recuperado el 8 de Diciembre de 2021, de <http://www.electronicapty.com/sensor-de-temperatura-lm35-detail?tmpl=component&format=pdf>

Barzanallana Asenso, R. (2015). *Desarrollo de aplicaciones web*. UMU.

Bautista., E. P. (1 de Junio de 2007). Sistema electrónico de monitoreo y mando vía web a través de internet. International Latin American and Caribbean Conference for Engineering and Technology.

- Bazán, P. (2017). *Aplicaciones, servicios y procesos distribuidos*. Buenos Aires: Edulp.
- Boada Oriolis, M., & Gómez Gutiérrez, J. A. (2019). *El gran libro de angular*. México: AlfaOmega.
- Candum, J. L., Jencson, A. L., Livingston, S. H., LI, D. F., Redmond, S. N., Pearlmutter, B., . . . Dobskey, C. J. (6 de Junio de 2020). *NCBI*. Recuperado el 22 de Junio de 2021, de <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7275188/>:
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7275188/>
- Carrera, J. M. (2019). *El temporizador 555*. Universidad Técnica Federico Santa María.
- CDC. (14 de Abril de 2020). *Centros para el control y la prevención de enfermedades*. Obtenido de Covid19: <https://espanol.cdc.gov/coronavirus/2019-ncov/community/organizations/disinfecting-transport-vehicles.html>
- Cimo, F. (2015). *Bootstrap Programing Coockbock*. WebCodeGeecks.
- Cornejo, J., Vargas, M., & Cornejo-Aguilar, J. A. (Diciembre de 2020). *Aplicaciones innovadoras de la robótica y biomédica en la salud pública durante la pandemia DEL Covid-19*. Obtenido de Scielo.
- Crespo, M., Pendino, C., & Reldán, G. (2002). *Componentes electrónicos*. Universidad Nacional de Rosario.
- De la Peña Valencia, B. G., & Castillo Rodríguez, F. D. (2009). *Elementos de electrónica*. Cuautitlán: FES-CUAUTITLAN.
- Díaz Castrillon, F. J., & Toro Montoya, A. I. (2020). SARS-CoV-2/COVID-19: el virus, la enfermedad y la pandemia. *Med. Lab*.
- Dimas Huacuz, B. (2020). *ABC de la COVIC-19*. México: INPI.
- Eguíluz Pérez, J. (2009). *Introducción a CSS*. AutoEdición.
- Enriquez, A., & Sáenz, C. (2021). *Primeras lecciones y desafíos de la pandemia de COVID-19 para los países del SICA*. México: CEPAL.
- G Sundar raju, K. S. (2021). Design and fabrication of sanitizer sprinkler robot for covid- 19 hospitals.
- Gobierno de la ciudad de México. (2021). *LINEAMIENTOS DE MEDIDAS DE PROTECCIÓN A LA SALUD QUE DEBERÁ CUMPLIR EL SECTOR DEL TRANSPORTE PÚBLICO PARA REANUDAR ACTIVIDADES HACIA UN REGRESO SEGURO A LA NUEVA NORMALIDAD EN LA CIUDAD DE MÉXICO*. Ciudad de México.
- Gutiérrez , L., Maigualida, M., & Piña , j. (Junio de 2008). *Tecnología GSM*. Obtenido de Universidad Yacambu: http://www.geocities.ws/johanna_pina28/fase2/t3.html
- Hamel Fonseca, J. (2011). CELDAS, PILAS Y BATERÍAS DE IONLITIO UNA ALTERNATIVA PARA....???. *JOURNAL BOLIVIANO DE CIENCIAS*, 40-47.
- Hernández Menéndez, A., Alaffita Hernández, F., & Olivares Galván, J. (2018). Análisis y diseño de la automatización del. *Identidad Energética*, 1.

- Hernandez Quintero, N. L., & Florez Fuente, A. S. (2014). Computación en la nube. 46-51.
- Huerta, E., Manguiterra, A., & Noguerra, G. (2005). *GPS posicionamiento satelital*. Rosario: UNR Editora.
- Instituto, n. d. (2019). Cloud computing. *Una guía de aproximación para el empresario*, 5-31.
- Jhon Alejandro Castro Correa, C. C. (2019). Sistema de geolocalización de vehículos a través de la red GSM/GPRS y tecnología Ardiono.
- Joaquín León Molina, E. A.-C. (21 de mayo de 2020). I. Desinfectantes y antisépticos frente al coronavirus: Síntesis de evidencias y recomendaciones .
- Joyanes Aguilar, L. (2019). *Inteligencia De Negocios Y Analítica De Datos*. Alfaomega.
- Khawas, C. (2018). Application of Firebase in Android App Development-A Study. *International Journal of Computer Applications*, 0975 – 8887.
- leantec. (2021). *Datasheet HC-SR04*. Recuperado el 8 de Diciembre de 2021, de <https://leantec.es/wp-content/uploads/2019/06/Leantec.ES-HC-SR04.pdf>
- Leticia Lopez Pedraza, J. R.-S. (2020). Desarrollo y test de un software de monitoreo de mensajes de texto (SMS) para pacientes con insuficiencia cardíaca aguda descompensada. *Revista Latino-Americana de Enfermagem*.
- Llamosas Falcón, L. S. (2020). SECUELAS A LARGO PLAZO DE COVID-19. *Revista Española de Salud Pública*.
- Lombina, J. C. (2015). Estación de monitoreo para cultivos, usando red GSM, Internet y microcontroladores. *Ingeniare, Revista chilena de ingeniería*.
- Luján Mora, S. (2002). *Programación de aplicaciones web*. España: Club Universitario.
- Maguiña Vargas, C., Gastelo Acosta, R., & Tequen Bernilla, A. (2020). El nuevo coronavirus y la pandemia del Covid-19. *Rev Med Hered*, 125-131.
- Maguiña Vargas, C., Gastelo Acosta, R., & Tequen, B. A. (Junio de 2020). El nuevo Coronavirus y la pandemia del Covid-19. *The new Coronavirus and Covid-19 pandemic*. Madrid, España: Revista Medica HHereditaria.
- MapBox inc. (16 de 1 de 2022). *Documentación MapBox GL*. Obtenido de <https://docs.mapbox.com/mapbox-gl-js/guides/>
- Mendez Aranda, J. R. (1999). *Estudio y evaluación del Modelo Cliente/Servidor en la migración de redes y aplicaciones en las empresas, dentro del marco definido por el concepto Middleware*. Monterrey.
- Miyara, F. (2004). *Electronica III conversores D/A y A/D*. Argentina: Universidad Nacional de Rosario.
- Morales Caluña, E. R., & Altamirano Capelo, F. X. (2016). *Computación en la nube*. Ecuador: ESPOCH.
- Nicola, F. E. (2016). *Redes selulares*. UNIVERSIDAD NACIONAL DE ROSARIO.

- Organización Mundial de la Salud. (2020). *Limpieza y desinfección de las superficies del entorno inmediato en el marco de Covid.19*. OMS.
- Organización Mundial de la salud. (2020). Transmisión del SARS-CoV-2: repercusiones sobre las precauciones en materia de prevención de infecciones .
- Paredes Colmenar, M. d., & Millanes Santos, j. (2020). *Aplicaciones web* . Madrid: Sintesis.
- Pérez Tirado, Á. (2010). *Manual de laboratorio de electrónica básica*. Colombia: UNIVERSIDAD DEL ATLÁNTICO.
- Pozo Ruz, A., Ribeiro, A., García Alegre, M. C., García, L., Guinea, D., & Sandoval, F. (Madrid). SISTEMA DE POSICIONAMIENTO GLOBAL (GPS): DESCRIPCIÓN, ANÁLISIS DE ERRORES, APLICACIONES Y FUTURO. 54-59.
- Prat Viñas, L. (1999). *Circuitos y dispositivos electrónicos* . Edicions UPC.
- Reyes , J. (2017). *Técnicas de Programación para Internet*.
- Ruiz Bravo, A., & Jiménez Valera, M. (20 de Julio de 2020). SARS-CoV-2 and acute respiratory síndrome pandemic (COVID-19). *SARS-CoV-2 y pandemia de síndrome respiratorio agudo (COVID-19)*. Granada, España.
- Serrano Sánchez, C. (2020). *TypeScript, Curso practico*. Ra-MA.
- SIMcom. (2013). SIM800L hardware design . 1.
- Terazona, A., & Jefferson, C. (Diciembre de 2020). *La tecnología como aliado para contrarrestar la pandemia del Covid-19*. Obtenido de Scielo.
- Torrente Artero, Ó. (2013). *Arduino Curso práctico de formación*. Ciudad de México: Alfaomega.
- Ubeda, B. (2009). *Sistemas Embebidos*. España: DIIC.
- u-blox. (2021). *neo 6 u-blox 6 gps modules data sheet*. Recuperado el 6 de Diciembre de 2021
- UNLP. (2016). *Electricidad*. Universidad Nacional de la plata.
- Velarzo Pardo, M. R., Honores, T. J., Gómez Moreno, A. S., & Vines Sánchez , L. (2018). Comparación de tendencias tecnológicas en aplicaciones WEB. *Glosas de Innovación aplicadas a la pyme*, 28-49.
- Vignaga, A., & Perovich, D. (2019). Arquitecturas y tecnologías para el desarrollo de aplicaciones web . *Universidad de la República, Facultad de Ingeniería, Instituto de*, 51.
- Villazón, E. E. (2018). Sistema de monitoreo energético y control Momótico basado en tecnología internet de las cosas.
- Zelaya Reyes, C. J. (2020). *Nuevas Tendencias en Desarrollo Web*.