



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“DISEÑO DE UN SISTEMA DE CONTROL Y MONITOREO
POR MEDIO DE SENSORES PARA LOS DEPARTAMENTOS
DE VACUNAS DEL DISTRITO DE SALUD 09D07”**

INFORME DE PROYECTO INTEGRADOR

Previo a la obtención del Título de:

LICENCIADO EN REDES Y SISTEMAS OPERATIVOS

BRAYAME LUIS CRIOLLO ORTIZ

IVAN RICARDO MUÑOZ RIVERA

GUAYAQUIL – ECUADOR

AÑO: 2018

AGRADECIMIENTOS

Mis más sinceros agradecimientos a mis padres, mi mama, Janeth Patricia Guevara Ortiz y mi papa, Emilio Alfonso criollo España, debido al sacrificio, consejos y guía de ellos yo pude optar por una formación profesional.

Agradezco a los docentes de la Escuela Superior Politécnica del Litoral (ESPOL), por compartir sus conocimientos y consejos a lo largo de mi carrera académica.

Brayame Luis Criollo Ortiz

AGRADECIMIENTOS

Agradezco a Dios por permitirme cumplir esta meta y guiarme día a día en la realización de mis actividades universitarias.

Agradecer a mis padres: Ivan Muñoz Bolaños y Juana Rivera Rivera, por haberme inculcado buenos valores y haberme guiado a realizar este sueño, su apoyo incondicional fue de gran valor para mí.

Agradecer enormemente a mi esposa Jenny Naula, quien ha sido el artífice de este sueño, con su apoyo incondicional y motivación diaria ayudo a cumplir esta meta que se había dificultado.

Agradecer a los docentes de la Escuela Superior Politécnica del Litoral, por haber compartido sus conocimientos tanto profesionales como éticos a lo largo de mi formación en mi carrera universitaria.

Ivan Ricardo Muñoz Rivera

DEDICATORIA

El presente proyecto lo dedico a mis padres, que con mucha paciencia siempre me brindaron su ayuda en todos los aspectos posibles de mi vida, también a mis hermanos que estuvieron presentes cuando tenía cualquier duda o inquietud, ellos estaban prestos a brindarme su apoyo y guía.

Además, también dedico este proyecto a todas las personas que estuvieron conmigo en el transcurso de este periodo para brindarme sus consejos, sin los cuales no hubiera podido tener éxito.

Brayame Luis Criollo Ortiz

DEDICATORIA

El presente proyecto de titulación lo dedico a Dios por darme la fuerza necesaria y el conocimiento para cumplir con este objetivo, pero especialmente le dedico este título a mis hijos: Nora y Mathias Muñoz quienes han sido mi mayor motivación, son quienes me dan la fuerza necesaria para seguir adelante en los momentos difíciles, quiero demostrarles que a pesar de las adversidades las personas pueden cumplir sus metas y sueños sin excusa alguna, dedicárselo también a toda mi familia como son: mi esposa, mis padres, mis hermanos, quienes son parte fundamental de mi vida, a ellos le dedico este pequeño esfuerzo y este objetivo que pude alcanzar, gracias por su apoyo incondicional.

Ivan Ricardo Muñoz Rivera

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

Brayame Luis Criollo Ortiz

Ivan Ricardo Muñoz Rivera

TRIBUNAL DE EVALUACIÓN

Msig. Robert Andrade Troya

PROFESOR DE MATERIA
INTEGRADORA

Mg. María Angélica Santacruz

TUTOR ACADÉMICO

RESUMEN

En este proyecto se diseñó un sistema de monitoreo de temperatura de frigoríficos, el cual permite visualizar en tiempo real sus temperaturas, además en este documento encontraremos la metodología usada para el desarrollo del proyecto.

En el área de vacunación del distrito de salud 09D07 Pascuales 1 existen constantes variaciones de temperatura en los frigoríficos, ocasionado por el mal estado de algunos de estos, lo que conlleva a que el personal del departamento tenga que estar vigilando físicamente los frigoríficos en un intervalo de tiempo determinado, además causa que el personal se atrase en sus actividades diarias.

Se ha diseñado un sistema de sensores conectados a un micro controlador para la recolección de datos de sensores de temperatura, este a su vez se conecta de forma inalámbrica a un raspberry pi 3, de modo que pueda transmitir los datos recogidos de los sensores hacia una base de datos para ser almacenada.

Una vez almacenada esta información puede ser visualizada en una plataforma web y en caso de cambios de temperaturas en los frigoríficos se realiza una notificación, esto con la finalidad de poder migrar a tiempo los suministros médicos que se encuentren dentro del frigorífico afectado y así prevenir daños de los componentes químicos.

Palabras Clave: NodeMCU ESP8266, raspberry pi3, Node-RED, Sensor DHT22.

ABSTRACT

In this project, a refrigeration temperature monitoring system was designed, which allows to visualize these temperatures in real time, and in this document we will find the methodology used for the development of the project.

The vaccination area of health district 09D07 Pascuales 1 does not have a temperature monitoring system for its refrigerators, which means that the department staff has to be physically monitoring the refrigerators in a certain time, also causing the personnel fall behind in their daily activities, this happens because constant temperature variations occur in the refrigerators caused by poor condition of some of them.

A system of sensors connected to a NodeMCU microcontroller has been designed, this in turn connects wirelessly to a raspberry pi 3, so that it can transmit the data collected from the sensors to a database to be stored.

Once stored this information can be viewed on a web platform and in case of temperature changes in the refrigerators a notification is made, this with the purpose of being able to migrate on time the medical supplies that are inside the affected refrigerator and thus prevent damages of the chemical components.

ÍNDICE GENERAL

RESUMEN.....	I
ABSTRACT.....	II
ÍNDICE GENERAL.....	III
ABREVIATURAS.....	V
SIMBOLOGÍA.....	VI
ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABLAS.....	IX
CAPÍTULO 1.....	1
1. Introducción.....	1
1.1 Descripción del problema.....	1
1.2 Objetivos.....	4
1.2.1 Objetivo General.....	4
1.2.2 Objetivos Específicos.....	4
1.3 Marco Teórico.....	5
1.4 Justificación.....	6
CAPÍTULO 2.....	7
2. Metodología.....	7
2.1 Empatizar.....	7
2.2 Definir.....	8
2.3 Idear.....	9
2.4 Prototipo.....	13
2.5 Testear.....	13

CAPÍTULO 3.....	15
3. Diseño de la solución.....	15
3.1 Subsistema de recolección de datos.....	16
3.1.1 Instalación del sensor DHT22	16
3.1.2 Configuración de NodeMCU	19
3.1.3 Protocolo MQTT.....	22
3.2 Subsistema de almacenamiento y presentación de la información.....	24
3.2.1 Access Point.....	24
3.2.2 Base de datos.....	26
3.2.3 Almacenamiento de los datos en raspberry pi.....	27
3.2.4 Visualización de los datos.....	30
3.2.5 Sistema de Notificación.....	34
3.3 Características de los equipos utilizados.....	40
3.3.1 Sensor DHT22.....	40
3.3.2 Microcontrolador NodeMCU ESP8266	41
3.3.3 Raspberry pi 3	42
CAPÍTULO 4.....	43
4. Presupuesto y planificación	43
4.1 Presupuesto.....	43
4.2 Planificación del proyecto.....	44
Conclusiones.....	45
Recomendaciones	46
BIBLIOGRAFÍA	47
ANEXOS	49

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
TIC	Information and Communication Technology
AP	Access Point
SSID	Service Set Identifier
MySQL	My Structured Query Language
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
DHCP	Dynamic Host Configuration Protocol
MQTT	Message Queue Telemetry Transport
WPA	Wi-Fi Protected Access
PSK	Pre-Shared Key
API	Application Programming Interface
HTML	HyperText Markup Language
UPS	Uninterruptible Power Supply

SIMBOLOGÍA

°C:	Grados Centígrados
C++:	Lenguaje de Programación
Ghz:	Gigahercio.
GND:	Tierra.
Hz:	Hercio.
802.11:	Estándar de Redes Inalámbricas.
mm:	Milímetro.
m/s:	Metro por segundo.
Rx:	Recepción.
Trig:	Desencadenar.
Tx:	Transmisión.
Vcc:	Voltaje de corriente continua
W:	Watt.

ÍNDICE DE FIGURAS

Figura 1.1. Captura de la Ubicación geografica del distrito 09D07.....	1
Figura 1.2. Entrada principal del distrito de salud 09D07	2
Figura 2.1. Matriz de importancia - dificultad.....	12
Figura 2.2. Prototipo del diagrama de sistema de monitoreo.....	13
Figura 3.1. Diseño general.....	15
Figura 3.2. Prensa estopa	16
Figura 3.3. Instalación de prensa estopa.....	17
Figura 3.4. Instalación de los sensores en los frigoríficos.....	18
Figura 3.5. Conexión del sensor entre el NodeMCU.....	19
Figura 3.6. Código de programación del NodeMCU parte 1.....	20
Figura 3.7. Código de programación del NodeMCU parte 2	21
Figura 3.8. Código de programación del NodeMCU parte 3	21
Figura 3.9. Código de programación del NodeMCU parte 4	22
Figura 3.10. Funcionamiento del protocolo MQTT	23
Figura 3.11. Configuración de los archivos dhcpd y dnsmasq	25
Figura 3.12. Configuración de Access Point	25
Figura 3.13. Información de la tabla monitoreo.....	26
Figura 3.14. Información de datos del personal.....	27
Figura 3.15. Captura de Node-RED en Raspbian.....	28
Figura 3.16. Despliegue de nodos en Node-RED parte 1	28
Figura 3.17. Configuración del protocolo MQTT en Node-RED.....	29
Figura 3.18. Query para el ingreso de datos en Node-RED.....	29
Figura 3.19. Configuración del nodo MySQL en Node-RED.....	30
Figura 3.20. Despliegue de nodos en Node-RED parte 2	31
Figura 3.21. Configuración del nodo Timestamp y SQL Query en Node-RED.....	31
Figura 3.22. Programación del nodo function (Código) en Node-RED	32
Figura 3.23. Captura de configuración del nodo Chart en Node-RED.....	33
Figura 3.24. Captura de interfaz de monitoreo en Node-RED.....	34
Figura 3.25. Sistema de notificación en Node-RED	34
Figura 3.26. Consultas a la base de datos en Node-RED.....	35

Figura 3.27. Configuración de los Topic en Node-RED	35
Figura 3.28. Configuración de Nexmo en Node-RED.....	36
Figura 3.29. Configuración de Correo en Node-RED.....	37
Figura 3.30. Captura de correo de notificación.....	37
Figura 3.31. Interface de acceso a la aplicación.....	38
Figura 3.32. Interfaz de administración de usuarios.....	39
Figura 3.33. Sensor DHT22.....	40
Figura 3.34. NodeMCU ESP8266.....	41
Figura 3.35. Raspberry pi 3.....	42
Figura 4.1. Planificación del proyecto.....	44

ÍNDICE DE TABLAS

Tabla 2.1. Punto de vista de la encargada del departamento.....	9
Tabla 2.2. Lluvia de ideas	9
Tabla 2.3. Matriz de decisión.....	11
Tabla 4.1. Presupuesto para la implementación del diseño.....	43



Figura 1.2. Entrada principal del distrito de salud 09D07.

Este distrito está encargado de administrar y proveer insumos médicos a 6 centros de salud y a la Penitenciaría del Litoral, existen diferentes departamentos entre ellos se encuentran TIC, recursos humanos, vacunación y otros.

Específicamente en el departamento de vacunación del distrito mencionado, existe una oficina que dispone de un área de refrigeración la cual tiene 6 frigoríficos, donde se almacenan las vacunas y otros suministros médicos a una temperatura de entre -8°C a 2°C (grados centígrados), una de sus funciones es almacenar dichos productos y mantenerlos en buen estado con la finalidad de distribuirlos a los centros de salud que lo soliciten y posteriormente ser administrado a los pacientes.

El problema del departamento radica porque en el área de refrigeración existen constantes variaciones de temperatura, específicamente en los frigoríficos que almacenan dichos suministros médicos por motivos de variación de energía y equipos defectuosos, llegando al punto de tener que controlar y monitorear la temperatura personalmente en cada frigorífico para verificar el estado de las vacunas y los suministros, en caso de que los medicamentos fuesen afectados por variaciones de temperatura no podrían distribuirse ni administrarse a ningún paciente, lo cual afectaría al distrito en pérdida económica y disponibilidad de medicamentos.

La forma de cómo detectar daños en las vacunas es por medio de una técnica llamada la prueba de agitación, esta prueba consiste en agitar por unos segundos el vial, luego se debe observar a detalle el comportamiento del componente químico, si el vial de prueba sedimenta más lentamente que un vial de referencia se puede usar caso contrario se desecha la vacuna.

1.2 Objetivos

1.2.1 Objetivo General

Diseñar un sistema de monitoreo y notificaciones, sobre las variaciones de temperaturas de los frigoríficos, para prevenir el daño a los componentes químicos de las vacunas y otros suministros optimizando el tiempo del personal encargado de vigilar el área de refrigeración.

1.2.2 Objetivos Específicos

- Prevenir el daño en los componentes químicos de las vacunas y otros suministros médicos.
- Optimizar el tiempo del personal encargado de vigilar el área de refrigeración.
- Monitorear y notificar las variaciones de temperaturas de los frigoríficos.

1.3 Marco Teórico

Se encontraron algunos proyectos que emplean sensores de temperatura para diferentes áreas, pero nada específico en lo que respecta al área de refrigeración para el control de medicamentos. Los proyectos son similares debido a que utilizan un sistema de monitoreo para diferentes situaciones como son la calidad del agua o la temperatura corporal pero no se emplean a la problemática de la variación de temperatura en frigoríficos.

Entre ellos está:

Diseño de un sistema de sensores dedicado al control y monitoreo de la calidad del agua en piscinas de crianza de camarones ubicada en la Isla Santa Rosa – Guayaquil [2].

Diseño e implementación de una red de sensores inalámbricos para el monitoreo de temperatura corporal [3].

1.4 Justificación

El distrito de salud 09D07 se encarga de proveer vacunas y suministros médicos a diferentes centros de salud, para que esto ocurra los suministros deben estar en óptimas condiciones, debido a las constantes variaciones de temperatura, el personal se encarga de monitorear estos suministros de forma manual, lo cual en algunos casos por falta de tiempo e imprecisión provoca que los suministros sufran daños en su composición química y tengan que desecharse.

Con la finalidad de prevenir esto, es necesario diseñar un sistema que monitoree las temperaturas y pueda notificar en caso de que ocurra una variación, algunos beneficios que esta alternativa podría brindar son los siguientes:

- Optimizar el tiempo del personal.
- Monitorear las temperaturas de los frigoríficos de forma precisa.
- Evitar que los suministros sean afectados y posteriormente desechados.

CAPÍTULO 2

2. METODOLOGÍA

Para conocer las problemáticas y necesidades del distrito 09D07 pascuales¹ se consideraron varias herramientas de la metodología Design Thinking.

2.1 Empatizar

Primero se realizó una investigación de escritorio para conocer datos del distrito como, por ejemplo:

- ¿Dónde se encuentra ubicado el distrito 09D07?
- ¿Qué tipo de actividades se realizan en este distrito?
- ¿Cuáles son las personas encargadas administrativamente en este distrito?

Luego de la investigación de escritorio se procedió a realizar entrevistas a varias personas del distrito, entre ellos el director del distrito de salud responsable de gestión de provisión, los encargados del departamento TIC y la encargada del departamento de vacunación, las preguntas de las entrevistas se las puede encontrar en el Anexo A Después de estas entrevistas con las personas mencionadas se pudo obtener varias problemáticas como, por ejemplo:

- No disponen de cámaras de seguridad en los centros de salud.
- Falta de control y políticas de seguridad en red LAN.
- La atención técnica a los usuarios del personal del distrito se realiza de forma desordenada, debido a que son solicitadas mediante SMS y llamadas telefónicas.
- Variación de temperatura en los frigoríficos donde se almacenan vacunas y suministros médicos.
- No existe un control de inventario de entrada y salida de los suministros médicos.
- Los generadores de energía del distrito se encuentran en mal estado.

Teniendo estas problemáticas se utilizó la herramienta P.O.V (Punto de vista) esta herramienta nos permite analizar las necesidades y preocupaciones de cada departamento, de esta manera se llegó a la conclusión con el personal de TIC que se debía resolver los problemas con los frigoríficos donde se almacenan suministros médicos.

2.2 Definir

Teniendo en cuenta los resultados se procedió a realizar la redefinición del problema centrado en el departamento de vacunación. Actualmente en el departamento de vacunación existen variaciones de temperatura en los frigoríficos, donde se almacén las vacunas y otros suministros médicos, con esto se definieron causas y efectos para un mayor entendimiento del problema.

Causas:

- Variación de la energía eléctrica en el distrito.
- Frigoríficos en mal estado.
- Evaluación de suministros médicos mediante procesos manuales.

Efectos:

- Daño en los componentes químicos de las vacunas y otros suministros médicos.
- La pérdida de los suministros médicos genera gastos al distrito.
- El exceso de carga laboral provoca que el personal cometa errores al momento de revisar los suministros médicos y por ende se da la pérdida de medicamentos.

Con la redefinición del problema se volvió a utilizar la herramienta de punto de vista, esta vez se centró en la Lcda. Ruth Quimi quien es la persona encargada del área de vacunación, este punto de vista se lo puede apreciar en la tabla 2.1.

Usuario	Necesidad	Punto de vista
Lcda. Ruth Quimi encargada del departamento de vacunación	<ul style="list-style-type: none"> • Monitorear la temperatura de los frigoríficos. • Optimizar el tiempo de trabajo de su personal de área. 	Aliviar la carga del personal, evitando el proceso manual de revisión de la temperatura de los frigoríficos y suministros médicos.

Tabla 2.1. Punto de Vista de la encargada del departamento.

2.3 Idear

Para esta fase se utilizó la herramienta lluvia de ideas, que nos permitió estudiar y analizar posibles soluciones a las problemáticas establecidas.

En la tabla 2.2 se muestran las posibles soluciones planteadas que se generaron al usar esta herramienta.

Lluvias de Ideas

Diseñar un sistema que monitoree la temperatura	Renovar los frigoríficos	Implementar windows server
Utilizar el CLOUD de Hikvision	Implementar una base de datos para la generación de tickets	Implementar una base de datos para el inventario

Tabla 2.2. Lluvia de ideas.

Adicional a la lluvia de ideas se utilizó la matriz de decisión que nos permite evaluar las posibles soluciones que más se ajusta a las necesidades de nuestro cliente, al aplicar esta herramienta algunas soluciones destacaron más que otras permitiéndonos visualizar la mejor opción.

Entre algunas necesidades del cliente destacaron el aplicar una solución de bajo costo, esto sucede debido a que el departamento no dispone de un presupuesto elevado, esta necesidad influyo mucho a la hora de seleccionar una solución, puesto que a pesar de tener que ser de bajo presupuesto debe realizar múltiples tareas para sustentar las necesidades solicitadas. Esto lo podemos observar en la tabla 2.3.

Necesidades	Posibles soluciones					
	Diseñar un sistema que monitoree la temperatura	Renovar los frigoríficos	Implementar windows server	Utilizar el CLOUD de Hikvision	Implementar una base de datos para la generación de tickets	Implementar una base de datos para el inventario
Monitorear la temperatura de los frigoríficos	✓	X	X	X	X	X
Notificar los cambios de temperaturas de los frigoríficos	✓	✓	X	X	X	X
Generación de ticket para atención del departamentos de TIC a los centros de salud	X	X	✓	X	✓	X
Control e implementación de políticas de seguridad para la red LAN	X	X	✓	X	X	X
Monitorear de manera remota las cámaras de seguridad de los centros de salud	X	X	X	✓	X	X
Automatizar el control del inventario de entrada y salida de los suministros médicos	X	X	✓	X	X	✓
Implementación de bajo costo	✓	X	X	✓	X	X
Disminuir la carga laboral del personal del distrito	✓	X	✓	✓	✓	✓
Total	4	1	4	3	2	2

Aplica	✓
No aplica	X

Tabla 2.3. Matriz de decisión.

Por último, usamos la herramienta matriz de importancia-dificultad, esta matriz nos permite clasificar las ideas en base a los aspectos negativos tales como tiempo, costo, esfuerzo, riesgo, complejidad entre otros y los aspectos positivos como son ingresos, ahorro de costo, riesgo de no hacerlo, etc.

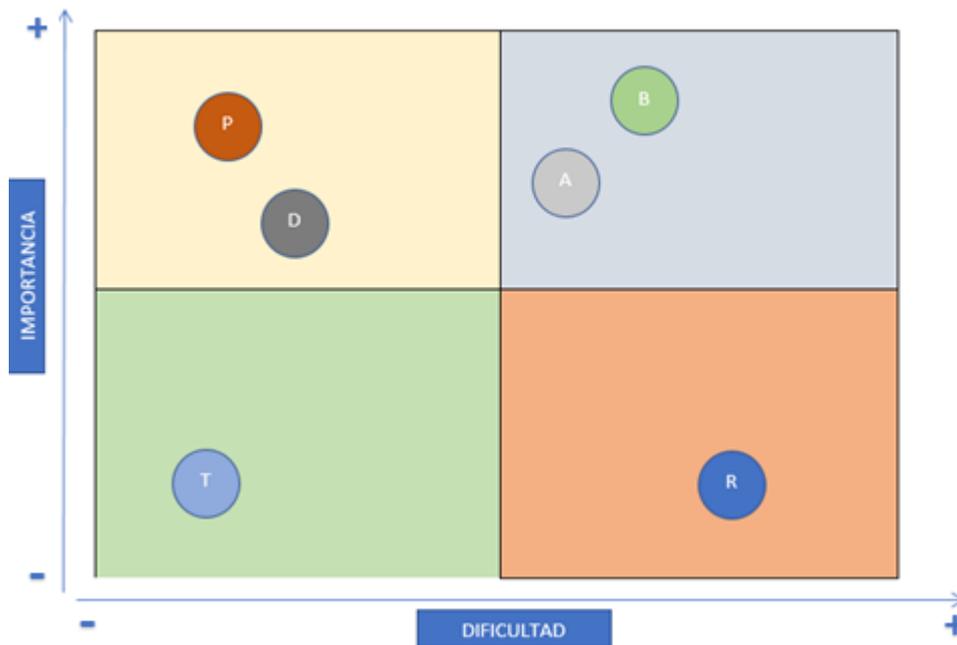


Figura 2.1. Matriz de importancia-dificultad.

- A: Diseñar un sistema que monitoree la temperatura.
- B: Implementar Windows server.
- D: Implementar una base de datos para el inventario.
- R: Renovar los frigoríficos.
- P: Implementar una base de datos para la generación de tickets.
- T: Utilizar la nube (del inglés *Cloud*) de Hikvision.

Después de utilizar las diferentes herramientas del Design Thinking, se seleccionó la problemática más importante dentro del distrito, la cual se torna en base a las variaciones de temperatura en los frigoríficos que almacenan las vacunas.

2.4 Prototipo

Para el prototipo se utilizaron sensores que permiten medir la temperatura de los frigoríficos con la finalidad de almacenar esta información en una base de datos, los sensores se conectarán a un microcontrolador el cual será el encargado de recolectar los datos y enviarlos de forma inalámbrica al servidor, los dispositivos tomados en cuenta para este prototipo son los sensores DHT22, microcontrolador NodeMCU ESP8266 y el servidor raspberry pi 3. En la figura 2.2 se muestra el prototipo de sistema de monitoreo.

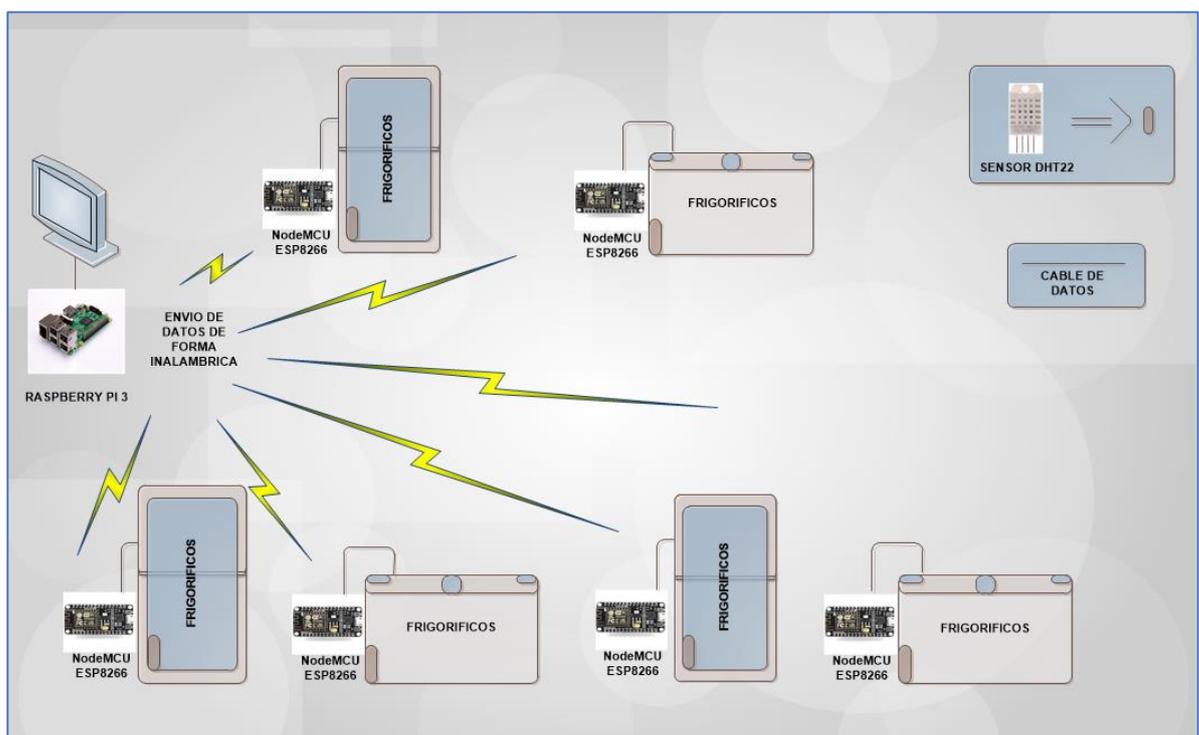


Figura 2.2. Prototipo del diagrama de sistema de monitoreo.

2.5 Testear

Después de la presentación del prototipo se obtuvieron algunas observaciones por parte del cliente y se replanteo algunas ideas del diseño. Unas de las observaciones fue el hecho de que en cada frigorífico solo existía un sensor de temperatura lo cual genero preocupación, debido a que si el sensor deja de funcionar este no tiene ningún otro que pueda monitorear la temperatura, por este motivo se plantea integrar dos sensores en cada frigorífico.

Otra observación presentada, fue la creación de una interfaz para el administrador del sistema de notificaciones, con la finalidad de gestionar el correo y número de celular del personal, debido a que se planteó enviar una notificación cuando se presente una variación de temperatura.

Esta observación se dio porque los datos del personal del departamento de vacunación irían fijos en el código y solo podrían modificarse por el desarrollador lo que conllevaría a una pérdida de tiempo y dinero.

Siguiendo todas las observaciones presentadas, se plantea la elaboración de un diseño de alto nivel el cual se detallará en el siguiente capítulo.

CAPÍTULO 3

3. DISEÑO DE LA SOLUCION

Este diseño fue creado en base a la necesidad que existe en el distrito de salud 09D07, donde no existe un sistema de monitoreo de las temperaturas en los frigoríficos que almacenan las vacunas, estas se deben almacenar a una temperatura de entre -8°C a $+2^{\circ}\text{C}$. El diseño de la solución está formado por 2 subsistemas, el primero se encargará de recolectar los datos de temperatura de los frigoríficos por medio de los sensores y enviar estos datos de forma inalámbrica al servidor, el segundo subsistema almacenará dichos datos y los mostrara mediante una interfaz para efectos de monitoreo, además el usuario recibirá una notificación cuando exista una variación de temperatura. En la figura 3.1 se muestra un diseño general.

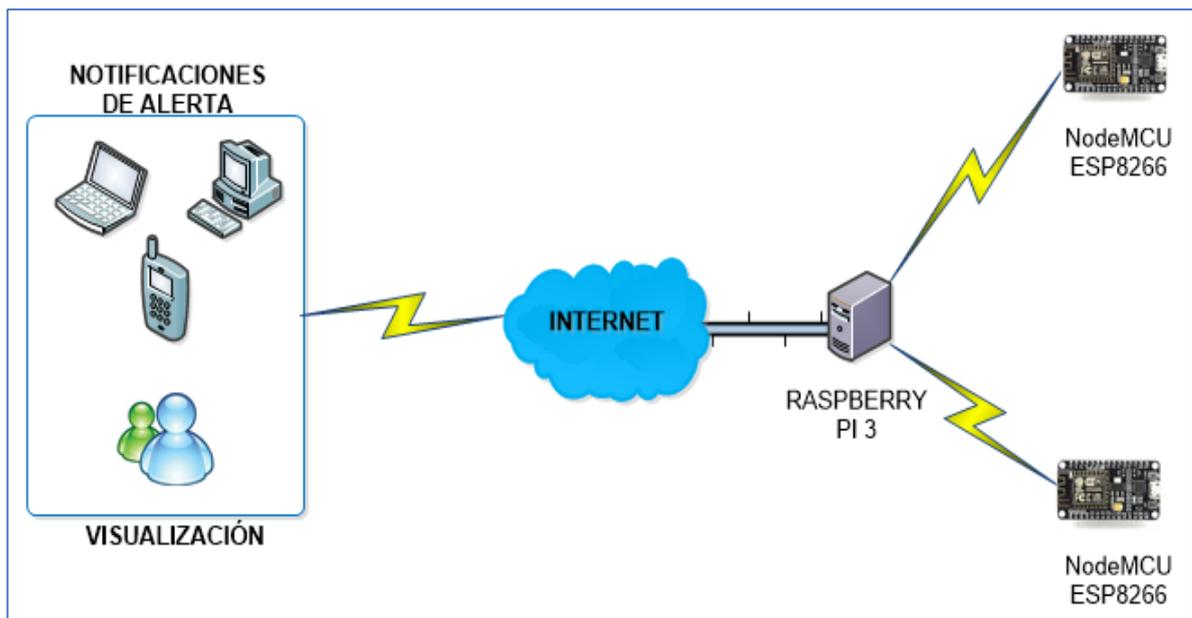


Figura 3.1. Diseño general.

3.1 Subsistema de recolección de datos

Para este proceso se utilizó el sensor DHT22 que permite medir los niveles de humedad mediante un sensor capacitivo y los niveles de temperatura con un termistor que mide el aire circundante [4], además se utilizó el micro controlador NodeMCU ESP8266 que se encargará de recibir los niveles de temperatura y humedad para posteriormente enviar los datos de forma inalámbrica utilizando el estándar IEEE 802.11, además se usó el protocolo MQTT (del inglés *Message Queue Telemetry Transport*) que permite establecer la comunicación con el raspberry pi 3.

3.1.1 Instalación del sensor DHT22

Para la instalación de los sensores se necesita hacer un orificio en el frigorífico, se escogió la puerta para realizar dicha instalación debido a que en las puertas no pasa ninguna tubería de refrigeración ni cableado eléctrico.

La perforación de la puerta se la realizará con un taladro de broca 5mm., se realizarán múltiples perforaciones circularmente con el fin de redondear el orificio en el frigorífico para luego proceder a instalar la prensa estopa, que contiene una pestaña con una banda de goma esto produce que al enroscar haga presión al cable evitando cualquier tipo de fuga. La prensa estopa la podemos observar en la siguiente figura 3.2



Figura 3.2. Prensa estopa [5].

Podemos observar el resultado final de la instalación en la siguiente figura 3.3.



Figura 3.3. Instalación de prensa estopa [5].

El cable que usaremos para la conexión del sensor al NodeMCU será el cable UTP categoría 5e para exteriores, el cual dispone de cuatro pares de cable, de dichos 4 pares solo utilizaremos 3 hilos para cada sensor que clasificaremos como energía, tierra y datos.

El departamento de vacunación cuenta actualmente con 6 frigoríficos, a cada frigorífico se le instalará 2 sensores DHT22 con la finalidad de crear redundancia, en caso de que un sensor dejase de funcionar el otro seguirá recolectando la temperatura y la humedad. La figura 3.4, muestra un ejemplo de instalación de dos frigoríficos.

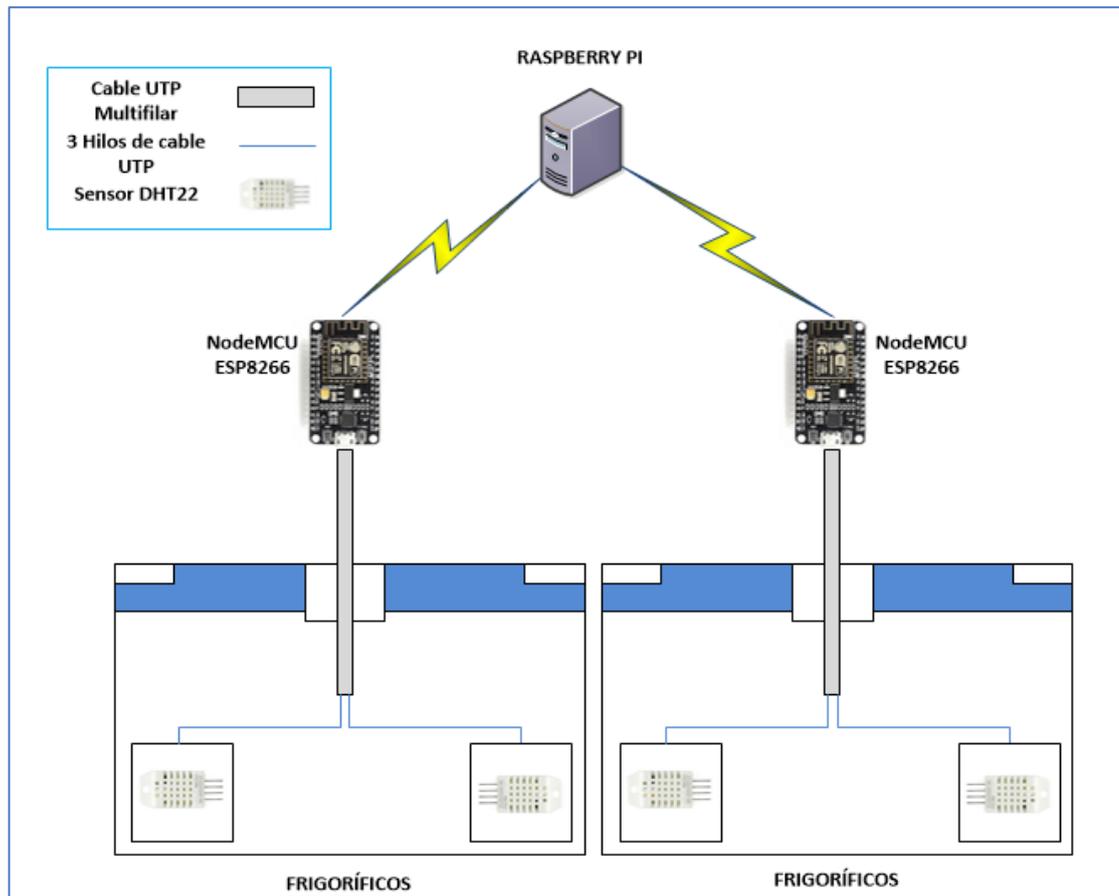


Figura 3.4. Instalación de los sensores en los frigoríficos.

Para la conexión del sensor al NodeMCU se utilizaron los siguientes pines de conexión.

- Hilo conductor de energía con el pin 5V
- Hilo conductor de tierra con el pin GND
- Hilo conductor de transmisión de datos en el pin D5

Podemos observar la conexión física entre el sensor DHT22 y el NodeMCU en la figura 3.5.

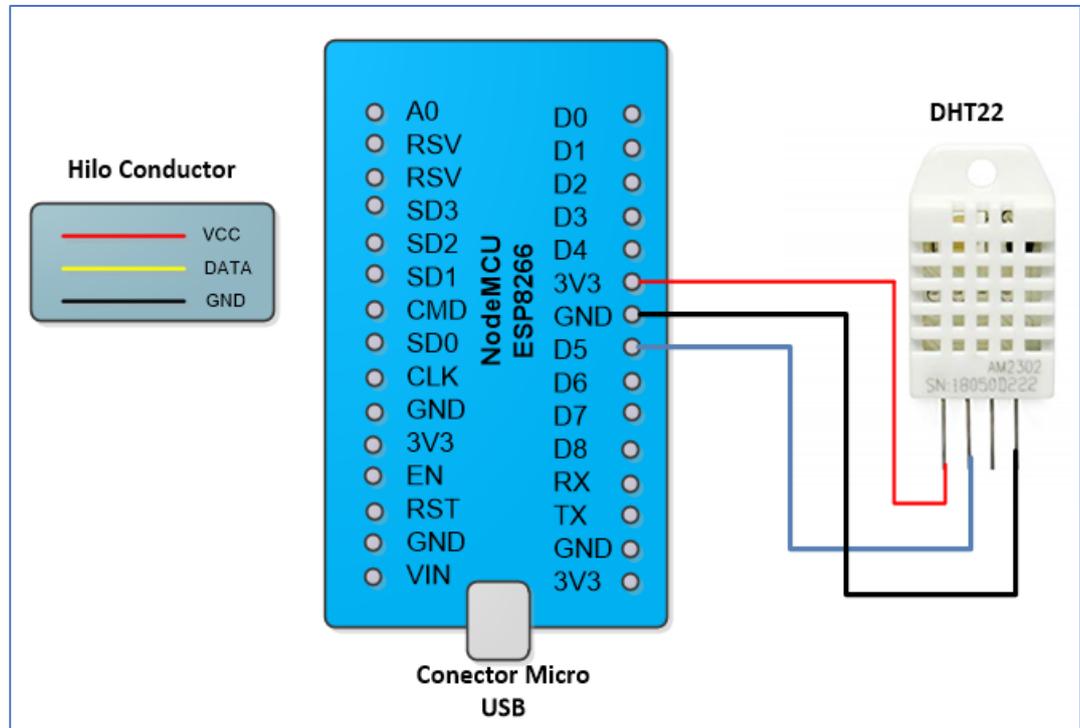


Figura 3.5. Conexión del sensor entre el NodeMCU.

3.1.2 Configuración de NodeMCU

El NodeMCU utiliza el lenguaje de programación C++, con dicho lenguaje aplicaremos el código para la recepción de datos de los sensores y el envío de datos al raspberry, recordemos que el envío de datos se realizará de forma inalámbrica [6].

Para definir los sensores, su tipo o modelo se necesitan las librerías Adafruit_Sensor.h y DHT, para el envío de datos y conexión inalámbrica se necesitan las librerías ESP8266WiFi.h y PubSubClient que es la librería de funcionamiento del protocolo MQTT que nos permitirá realizar la función de cliente servidor.

Para empezar la lectura de temperatura se definen los pines al cual está conectado el sensor y también se define el tipo de sensor. Esto lo podemos observar en la figura 3.6.

```
/******LIBRERIAS*****/  
#include <ESP8266WiFi.h>  
#include <PubSubClient.h>  
#include <Arduino.h>  
#include "DHT.h"  
  
/******DEFINIR TIPO Y MODELO DEL SENSOR*****/  
#define PIN2 D2  
#define TIPO_SENSOR DHT22  
  
#define PIN3 D3  
#define TIPO_SENSOR DHT22  
  
DHT dht2(PIN2, TIPO_SENSOR);  
DHT dht3(PIN3, TIPO_SENSOR);
```

Figura 3.6. Código de programación del NodeMCU parte 1.

En esta parte del código se configura la velocidad de datos en bits por segundo para la transmisión de datos en serie. Para poder comunicarse con la computadora en este caso se eligió 115200 bits por segundo.

También se procedió a definir las variables para identificar los dispositivos, como la ubicación del NodeMCU y los sensores, esto se lo hizo con el propósito de identificarlos en caso de que presenten fallas y se realice el mantenimiento, también para identificarlos cuando se presente una variación de temperatura dado que existirá un sistema de notificación que enviara la posición del sensor para su respectiva revisión.

Adicionalmente se configuraron las variables para la lectura de temperatura y humedad. Esto se puede apreciar en la figura 3.7.

```

/*****CONFIGURACION DE SENSOR*****/
Serial.begin(115200);
dht2.begin();
dht3.begin();
/*****/
}
void loop() {
  /*****CONFIGURACION DE LECTURA DEL SENSOR*****/
  int IdNodeMCU = 1;//IDENTIFICACION DEL MICROCONTROLADOR
  int Idsensor1 = 1;//IDENTIFICACION DEL SENSOR
  int Frigorifico = 5;//IDENTIFICACION DEL FRIGORIFICO
  int Sensor2h = dht2.readHumidity();//LECTURA DE HUMEDAD
  int Sensor2t = dht2.readTemperature();//GRADOS CENTIGRADOS
/*****/

```

Figura 3.7. Código de programación del NodeMCU parte 2.

Una de las funciones que emplea el raspberry pi es la configuración como Access point, esto se realiza con la finalidad de que el NodeMCU pueda conectarse de forma inalámbrica por medio de una solicitud para la asignación de una IP. Adicionalmente se configuro el número de caracteres que se trasmiten de forma inalámbrica al raspberry, para evitar cualquier tipo de error pusimos 800(char msg), esto representa el número de caracteres que se pueden transmitir al raspberry. Este código se muestra en la figura 3.8.

```

/*****CONEXION AL ACCESP POINT*****/
const char* ssid = "Monitoreo";
const char* password = "distrito09d07";
const char* mqtt_server = "192.168.1.1";

/*****NUMERO DE CARACTERES QUE SE ENVIAN EN MSG*****/
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[800];

int value = 0;
void setup_wifi() {

```

Figura 3.8. Código de programación del NodeMCU parte 3.

En el siguiente código mostrara la IP que asigno el raspberry pi al momento de conectarse a este. Este código se lo puede observar en la siguiente figura 3.9.

```
/******SOLICITUD DE IP AL RASPBERRY*****//  
delay(10);  
Serial.println();  
Serial.print("Connecting to ");  
Serial.println(ssid);  
  
WiFi.begin(ssid, password);  
  
while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
}  
randomSeed(micros());  
Serial.println("");  
Serial.println("WiFi connected");  
Serial.println("IP address: ");  
Serial.println(WiFi.localIP());  
}
```

Figura 3.9. Código de programación del NodeMCU parte 4.

3.1.3 Protocolo MQTT

MQTT es un protocolo muy utilizado en el mundo del internet de las cosas debido a que nos permite reducir el consumo de ancho de banda y ser utilizado en equipos de bajo nivel que posean pocos de recursos de memoria RAM (del inglés *Random Access Memory*).

MQTT también puede ser utilizado como un servicio de publicación/suscripción TCP/IP sencillo y sumamente ligero. Se basa en el principio cliente/servidor.

El servidor, llamado bróker, recopila los datos que los publishers (los objetos comunicantes) le transmiten. Ciertos datos recopilados por el broker se enviarán a determinados publishers que previamente así se lo hayan solicitado.

En nuestro caso se lo utilizo el protocolo MQTT para transmitir los datos entre el NodeMCU Y el raspberry pi. la comunicación entre ambos es por medio de topic (tema), los cuales son creados por el cliente de tal manera que los nodos que deseen recibirlos deben subscribirse a él [7].

El topic que utilizamos se llama MONITOREO, lo que permite que el bróker envíe los datos a todos los clientes con esta suscripción, en este caso raspberry pi, esto lo podemos observar en la siguiente figura 3.10.

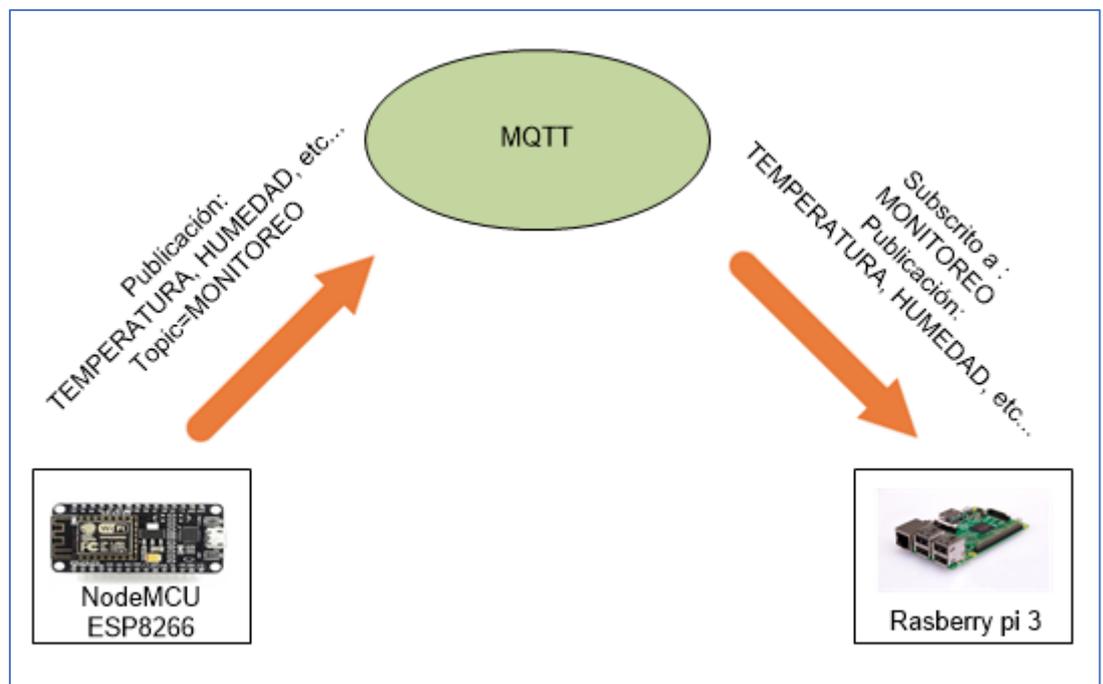


Figura 3.10. Funcionamiento del protocolo MQTT.

3.2 Subsistema de almacenamiento y presentación de la información

Para este subsistema se utilizó el raspberry pi 3, que es un ordenador de tamaño reducido y es catalogado como computador de placa simple de bajo costo, funciona con su propio sistema operativo llamado raspbian, este sistema está basado en la distribución de Linux Debian.

Para este proyecto se lo utilizó en múltiples funciones como:

- Access point
- Base de datos
- Interfaz de visualización
- Sistema de notificación

3.2.1 Access point

El raspberry pi 3 para nuestro proyecto se lo utilizo como un AP (Access Point), esto se lo realizó con la finalidad de que los micros controladores NodeMCU se puedan conectar al raspberry pi de manera inalámbrica y así poder transmitir los datos recogidos de la temperatura en los frigoríficos [8].

Se instaló los paquetes hostapd y dnsmasq, que nos ayudarán a poder configurar el raspberry pi en modo AP.

Se configuró una IP estática para el dispositivo, dicha IP fue la 192.168.1.1/24, adicionalmente se agregó un pool DHCP utilizando el rango de IP 192.168.1.2/24 - 192.168.1.254/24, con esto se realiza la asignación de IP a los micros controladores NodeMCU, cada configuración se la realiza en sus respectivos directorios como se muestra en la figura 3.11.

```
GNU nano 2.7.4 Fichero: /etc/dhcpd.conf
#interface eth0
#fallback static_eth0

interface wlan0
    static ip_address=192.168.1.1/24
    nohook wpa_supplicant

GNU nano 2.7.4 Fichero: /etc/dnsmasq.conf

interface=wlan0
    dhcp-range=192.168.1.2,192.168.1.254,255.255.255.0,24h
```

Figura 3.11. Configuración de los archivos dhcpd y dnsmasq.

Se configuró el *SSID* (del inglés *Service Set Identifier*) que permite identificar el nombre de una red inalámbrica Wifi y la contraseña. El SSID de la red inalámbrica se la nombró monitoreo, adicionalmente se configuraron parámetros como la seguridad donde se utilizó el cifrado WPA 2- PSK y el estándar inalámbrico IEEE 802.11 tal como se muestra en la figura 3.12.

```
GNU nano 2.7.4 Fichero: /etc/hostapd/hostapd.conf

interface=wlan0
driver=nl80211
ssid=Monitoreo
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=distrito09d07
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Figura 3.12. Configuración de Access Point.

3.2.2 Bases de datos

Para el sistema de gestión de base de datos se trabajó con MySQL, una de las ventajas es que se puede ejecutar en varias plataformas como Windows y Linux.

Para este proyecto se instaló el motor de base de datos MySQL mediante la consola de comando, en dicha base se crearon dos tablas con sus respectivos atributos, también se le configuró su respectivo usuario y contraseña para poder acceder a dicha base de datos. Las características técnicas se la especifican a continuación:

- Nombre de la base datos: BASE_DE_DATOS_SENSORES
- Nombre de la tabla: MONITOREO
- Atributos de la monitoreo: ID, FECHA, HUMEDAD, TEMPERATURA, NUMSENSOR, NODEMCU, FRIGORIFICO
- Nombre de la tabla: DATOS_DEL_PERSONAL
- Atributos de datos_del_personal: ID, CEDULA, NOMBRE, APELLIDO, CORREO, CELULAR

Los atributos de la tabla monitoreo se lo puede observar en la figura 3.13.

Field	Type	Null	Key	Default	Extra
ID	bigint(20)	NO	PRI	NULL	auto_increment
FECHA	datetime	YES		NULL	
TEMPERATURA	decimal(10,0)	YES		NULL	
HUMEDAD	decimal(10,0)	YES		NULL	
IDSENSOR	int(4)	YES		NULL	
IDNODEMCU	int(4)	YES		NULL	
FRIGORIFICO	int(4)	YES		NULL	

Figura 3.13. Atributos de la tabla monitoreo.

La siguiente tabla contiene los atributos de los datos del personal, véase la figura 3.14.

```
MariaDB [BASE_DE_DATOS_SENSORES]> SELECT * FROM DATOS_DEL_PERSONAL;
```

ID	CEDULA	NOMBRE	APELLIDO	CORREO	CELULAR
1	0987474334	brayane	criollo	blcrioll@espol.edu.ec	0987261912
2	0999999999	ivan	munoz	imunoz@espol.edu.ec	0986157586

2 rows in set (0.00 sec)

Figura 3.14. Atributos de tabla datos del personal.

3.2.3 Almacenamiento de los datos en raspberry pi

Para almacenar las lecturas de temperatura, optamos por utilizar la herramienta Node-RED [9], que es un motor de flujos con enfoque IoT (Internet de las Cosas), que permite programar mediante nodos conectados entre sí, además de integrar API (del inglés Application Programming Interface) de terceros y servicios en línea.

Es una herramienta visual muy ligera que puede ejecutarse en dispositivos tan limitados como raspberry pi. El editor de flujos de Node-RED consiste en una sencilla interfaz en HTML, accesible desde cualquier navegador, en la que arrastrando y conectando nodos entre sí, es posible definir un flujo que ofrezca un servicio. En las últimas versiones del sistema operativo raspbian ya viene instalado de forma predeterminada como se muestra en la siguiente figura 3.15.

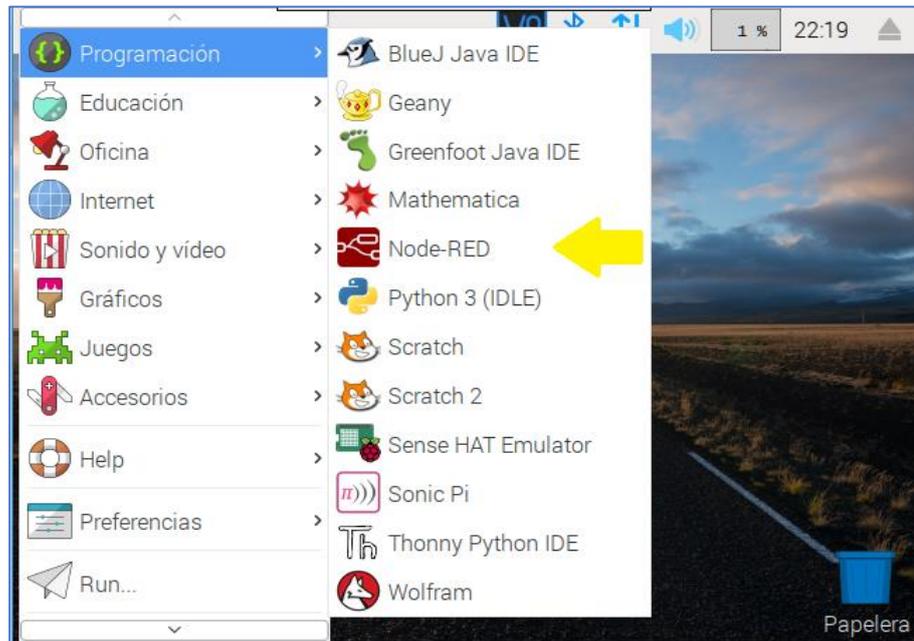


Figura 3.15. Captura de Node-RED en raspbian.

Dentro de la plataforma de Node-RED se necesitó los nodos (MQTT, json, function y MySQL), para poder realizar la configuración, cada uno de los nodos están configurados con sus respectivos parámetros [10]. El despliegue de los nodos se puede observar en la figura 3.16.



Figura 3.16. Despliegue de nodos en Node-RED parte 1.

El nodo MQTT ya está configurado para recibir los datos, esto lo podemos observar en la siguiente figura 3.17.

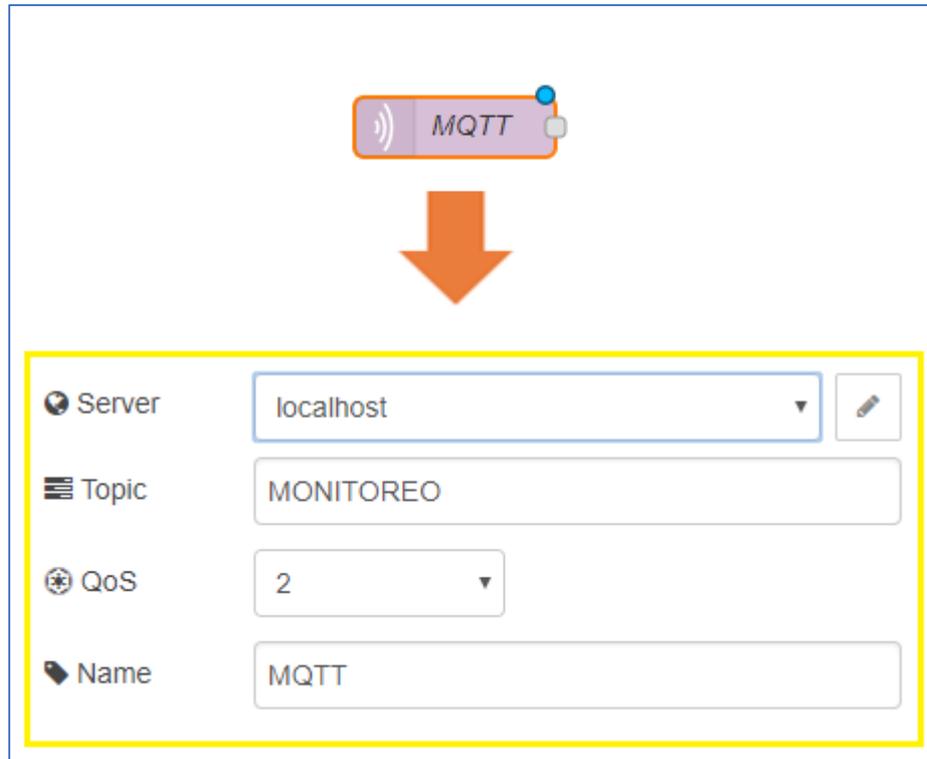


Figura 3.17. Configuración del protocolo MQTT en Node-RED [11].

El nodo json se lo utiliza para convertir los datos recibidos de formato json a objetos en JavaScript, lo que dará una mayor facilidad al programar, esto sucede debido a que se puede acceder una propiedad de un objeto con una simple notación de puntos (NombreObjeto.Nombrepropiedad).

El nodo ingreso de datos (Function) se encarga de ingresar los datos a la tabla de monitoreo mediante una consulta a la base de datos (query), este nodo trabaja con lenguaje de programación JavaScript, esto se lo puede observar en la siguiente figura 3.18.

```

1 var now = new Date();
2 var isonow = new Date(now.getTime()-(now.getTimezoneOffset()*60000)).toISOString();
3
4 msg.topic =
5   "INSERT INTO MONITOREO(FECHA, TEMPERATURA, HUMEDAD, IDSENSOR, IDNODEMCU, FRIGORIFICO) VALUES" +
6   " ('"+isonow+" ', '"+msg.payload.Temperatura+" ', '"+msg.payload.Humedad+" ', '"+msg.payload.IDsensor+" ', ' " +
7   " '"+msg.payload.IDnodeMCU+" ', '"+msg.payload.Frigorifico+" ' )";
8 return msg;

```

Figura 3.18. Query para el ingreso de datos en Node-RED.

El nodo MONITOREO (MySQL) está configurado para conectarse a la base de datos creada en el raspberry pi, como se puede observar en la siguiente figura 3.19.

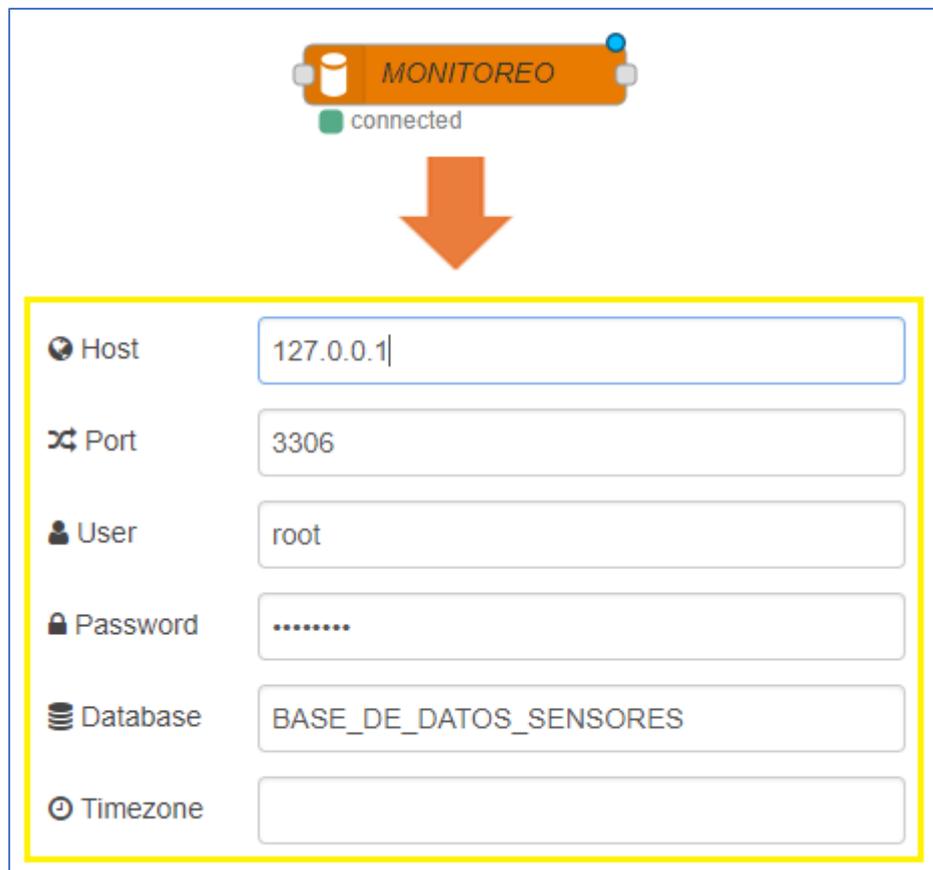


Figura 3.19. Configuración del nodo MySQL en Node-RED [12].

3.2.4 Visualización de datos

Para la interfaz de visualización se optó por utilizar la herramienta de Node-RED mencionada anteriormente.

Dentro de la plataforma de Node-RED se necesitó los nodos (MySQL, inject, function, y chart) para poder realizar la configuración. Cada uno de los nodos están configurados con sus respectivos parámetros, el despliegue de estos se puede observar en la figura 3.20.

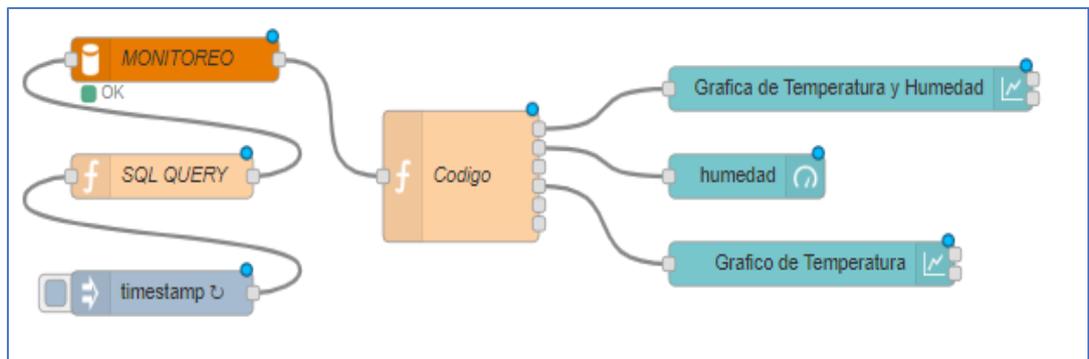


Figura 3.20. Despliegue de nodos en Node-RED parte 2.

El nodo timestamp lo usamos para enviar peticiones cada cierto tiempo según el requerimiento [13], esta petición va acompañada del nodo (SQL QUERY) [14], para ser enviada al nodo (MONITOREO) que conecta con la base de datos, el cual proporciona los últimos 100 datos de la tabla, como podemos observar en la siguiente figura 3.21.

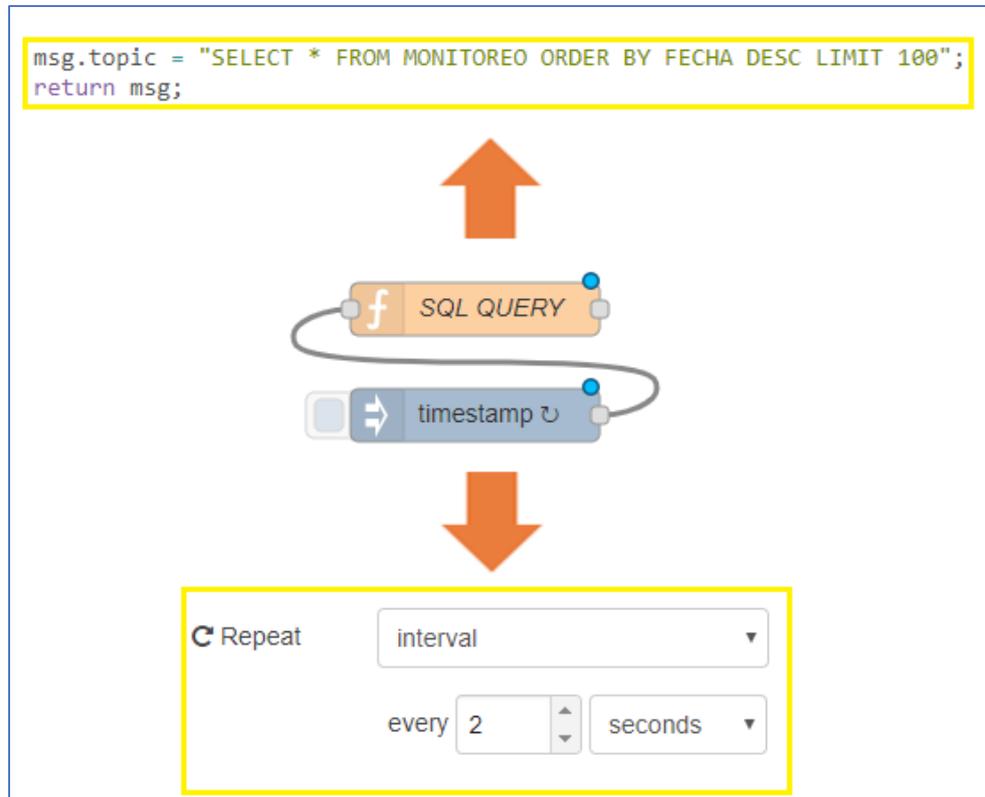


Figura 3.21. Configuración del nodo timestamp y SQL QUERY en Node-RED.

Nodo function (Código), contiene la programación para mostrar los datos de temperatura y humedad, su función es recorrer el mensaje entregado del nodo (MONITOREO) para poder enviar los datos al nodo (Gráfica de temperatura y humedad) el cual traza los valores de entrada en un gráfico, parte del código los podemos observar en la figura 3.22.

```
1 //objetos :
2 var newmsg1 = {};
3 var newmsg2 = {};
4
5 var DEVICE = {};
6 //arrays
7 var temp = [];
8 var time = [];
9
10 var hum = [];
11 var datasensor = [];
12 var ARDUINO = [];
13 var fecha = [];
14
15 var long, i;
16 long = msg.payload.length;
17
18 for (i=0;i<=long-1;i++){
19     var temperatura = {};
20     var humedad = {};
21
22     if (msg.payload[i].IDSSENSOR===4){
23         temperatura.x=Date.parse(msg.payload[i].FECHA);
24         temperatura.y=msg.payload[i].TEMPERATURA;
25         temp[i] = temperatura;
26
27         humedad.x=Date.parse(msg.payload[i].FECHA);
28         humedad.y=msg.payload[i].HUMEDAD;
29         hum[i] = humedad;
30     }
31 }
32 datasensor[0] = temp;
33 datasensor[1] = hum;
34
35 DEVICE.series = ["GRAFICA TEMP","GRAFICA HUM"]
36 DEVICE.data = datasensor;
37 DEVICE.labels=["ETIC1","ETIC2"];
38 ARDUINO [0]=DEVICE
39 newmsg1.payload = ARDUINO;
40 newmsg2.payload = long;
41
42 return [newmsg2,null,null,newmsg1,null,null];
```

Figura 3.22. Programación del nodo function (Código) en Node-RED.

Los nodos debug sirven para visualizar los mensajes que están pasando al momento de ejecutar el programa, en caso de ocurrir algún error en la programación nos muestra un mensaje que nos permite identificar el nodo afectado y el error. Este no necesita ninguna configuración, solo lo despliega y conecta al nodo que se desea analizar.

Los nodos chart (Gráfica de temperatura y humedad), recogen el mensaje que genera el nodo function (Código) anteriormente mencionado y nos permite mostrar los datos en una gráfica que genera este [15], la configuración la podemos observar en la siguiente figura 3.23.

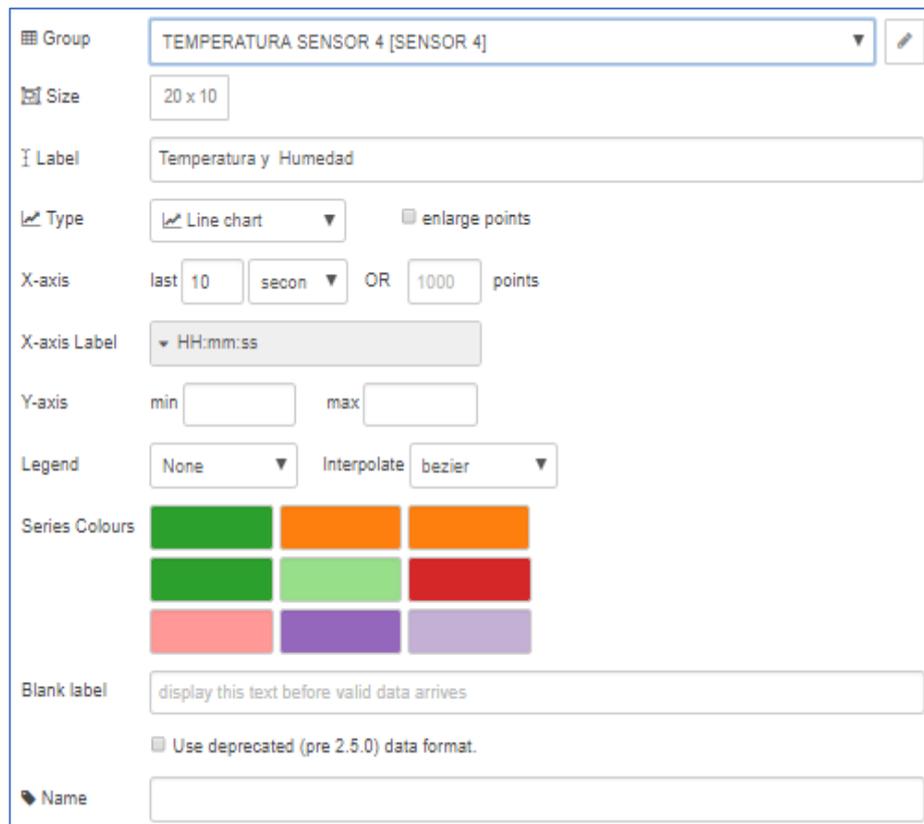


Figura 3.23. Captura de configuración del nodo chart en Node-RED.

interfaz donde se podrá visualizar las temperaturas de los frigoríficos, lo podemos observar en la figura 3.24.

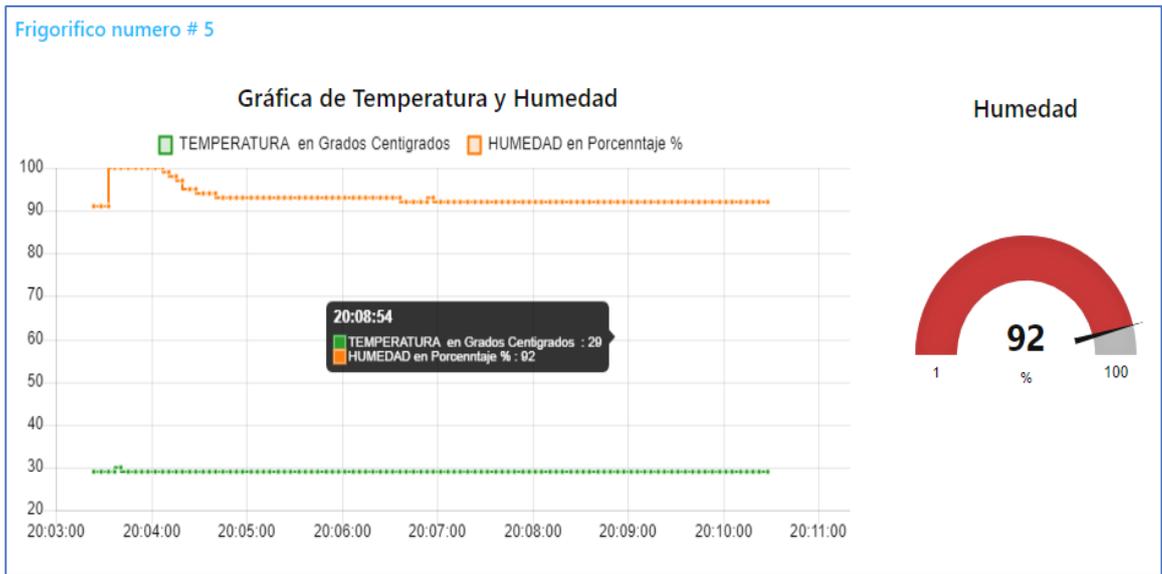


Figura 3.24. Captura de interfaz de monitoreo en Node-RED.

3.2.5 Sistema de notificación

En esta parte enviaremos las notificaciones al correo y celulares del personal encargado del departamento, esto sucederá cuando ocurra una variación de temperatura en los frigoríficos con la finalidad de mantener en alerta al personal. Para este sistema se utilizaron los nodos email, nexmo(SMS), delay, trigger. Podemos observar el despliegue de nodos en la siguiente figura 3.25.

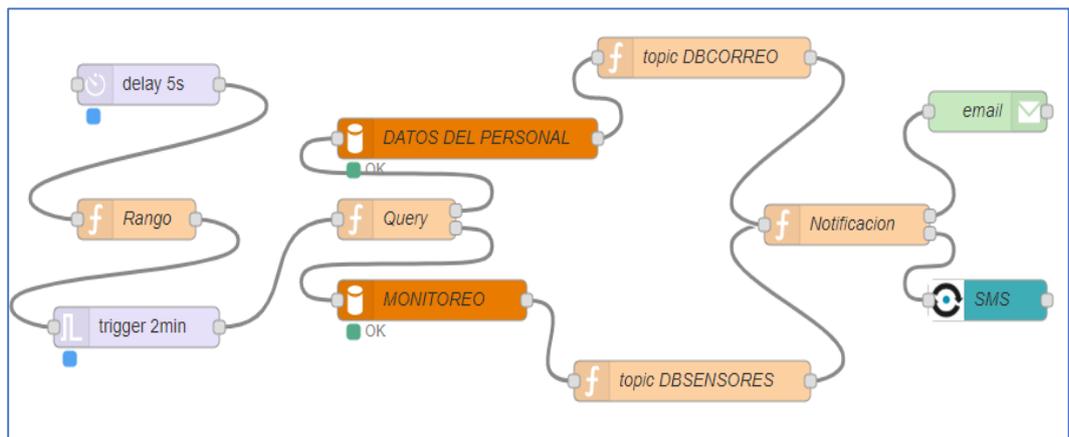


Figura 3.25. Sistema de notificación en Node-RED.

El nodo (deley) recibe los datos cada 5 segundos, este nodo conecta con el nodo function (Rango), el cual establece un rango de temperatura que si sobrepasa retorna true al nodo (trigger 2min), este nodo está configurado para que cuando recibe true se espere 2 minutos hasta enviar el siguiente dato al nodo (Query).

El nodo (Query) se encarga de realizar consultas a la base de datos (datos del personal y monitoreo). La primera consulta se encarga de devolver los datos de temperatura que sobrepasen el rango establecido, la segunda consulta se realiza para devolver los parámetros de nombre, correo y número celular del personal, como se puede observar en la siguiente figura 3.26.

```
1 msg1 ={};
2 msg2 ={};
3 msg1.topic = "SELECT CORREO ,NOMBRE, CELULAR FROM DATOS_DEL_PERSONAL ";
4 msg2.topic = "SELECT FECHA,TEMPERATURA,HUMEDAD,IDSENSOR,IDFRIGORIFICO FROM MONITOREO"+
5 "WHERE TEMPERATURA < -8 OR TEMPERATURA > 2 ORDER BY FECHA DESC LIMIT 1 ";
6
7 return [msg1, msg2];
```

Figura 3.26. Consultas a la base de datos en Node-RED.

Los nodos de la base de datos (DATOS DEL PERSONAL y MONITOREO) envían mensajes simultáneamente al nodo function (Notificaciones), para que este puede distinguir dichos mensajes se procedió a implementar dos nodos function (topic DBSENSORES y topic DBCORREO) para poder cambiar la etiqueta de los mensajes. Esto se lo puede apreciar en la figura 3.27.

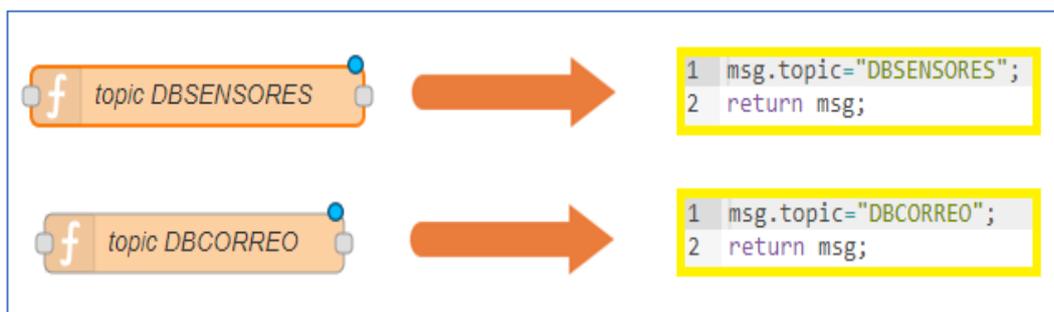


Figura 3.27. Configuración de los Topic en Node-RED.

El nodo (Notificaciones) se encargará de recorrer los mensajes entregados de los diferentes nodos, para proceder a realizar el envío de las notificaciones a las cuentas de correo y celular (SMS) del personal [16].

Para enviar las notificaciones se creó una cuenta de correo en Gmail, y también una cuenta en la plataforma nexmo, este nos permite el envío de SMS entre otras funciones, también cuenta con un nodo en Node-RED, en este se debe configurar las claves que se obtienen al momento de crear una cuenta, esto lo podemos observar en la figura 3.28.

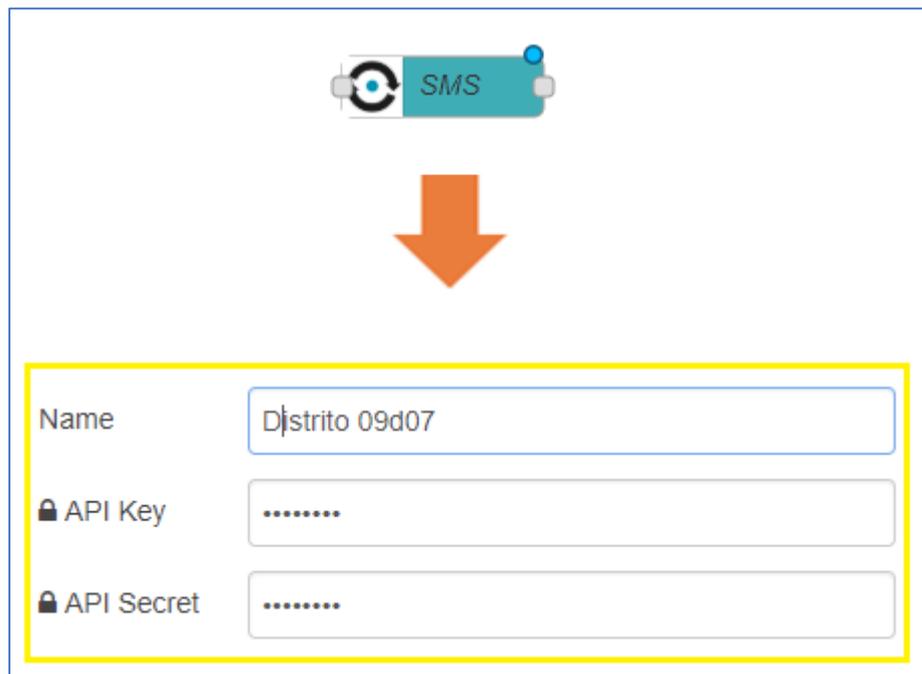


Figura 3.28. Configuración de Nexmo en Node-RED [17].

Adicionalmente se debe configurar la cuenta creada de Gmail, como podemos observar en la siguiente figura 3.29.

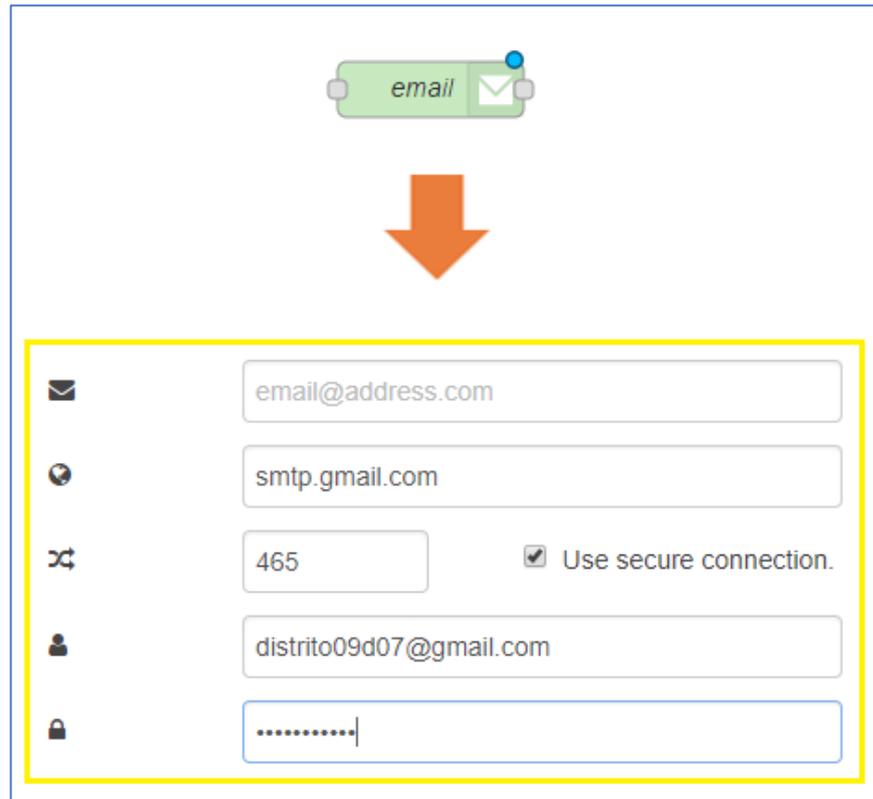


Figura 3.29. Configuración de correo en Node-RED [18].

Se configuró el rango de temperatura establecido en el distrito, con la finalidad de recibir un correo que contenga la información de la temperatura y otros parámetros, lo podemos observar en la siguiente figura 3.30.



Figura 3.30. Captura de Correo de notificación.

Para que el administrador de la base de datos tuviera mayor facilidad al ingresar, eliminar y buscar usuarios se realizó una aplicación en NetBeans que es una herramienta de programación para el desarrollo de aplicaciones, por motivos de seguridad solo puede ejecutarse desde el raspberry pi, la interfaz de inicio se la puede observar en la siguiente figura 3.31.



The image shows a login interface with a light gray background. At the top center is the logo of the Ministerio de Salud Pública (MSP), which consists of a stylized 'm' and 'p' in yellow and red, with a blue wave-like element between them. Below the logo, the text 'Ministerio de Salud Pública' is displayed. Underneath, there are two input fields: the first is labeled 'Usuario:' and the second is labeled 'Contraseña:'. At the bottom of the interface, there are two buttons: 'Ingresar' on the left and 'Salir' on the right.

Figura 3.31. Interface de acceso a la aplicación.

Luego de iniciar sesión aparecerá otra interface, en la cual se podrá añadir a usuarios y sus respectivos datos como son cédula, nombre, apellido, correo y celular, adicionalmente contara con la opción de eliminar y editar, en caso de cualquier error al ingresar los datos como se puede apreciar en la siguiente figura 3.32.

ID	CEDULA	NOMBRE	APELLIDO	CORREO	CELULAR

Id:

Cédula: **Correo:**

Nombre: **Celular:**

Apellido:

Figura 3.32. Interfaz de administración de usuarios.

3.3 Características de los equipos utilizados

Los equipos que más se ajustaron a la necesidad de nuestro proyecto fueron los siguientes:

- Sensor DHT22
- Micro controlador NodeMCU ESP8266
- Raspberry pi 3

3.3.1 Sensor DHT22

El sensor DHT22 permite medir los niveles de humedad mediante un sensor capacitivo y los niveles de temperatura con un termistor que mide el aire circundante. El sensor DHT22 se lo puede apreciar en la figura 3.33.

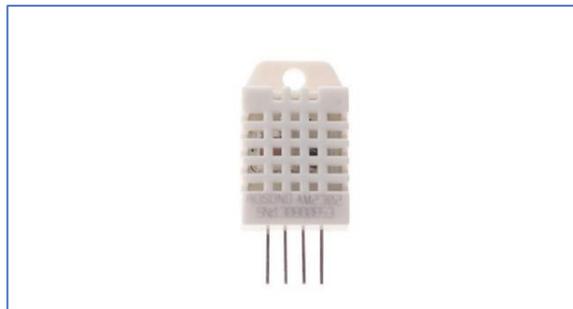


Figura 3.33. Sensor DHT22 [19].

A continuación, se detalla las características del sensor.

- Muy bajo costo.
- Compatible con sistemas electrónicos operando a 3 y 5 volts.
- Corriente máxima de 2.5 mA cuando se realiza la conversión.
- 0 – 100 % Humedad relativa, precisión del 2%.
- -40°C a 80 °C de temperatura, precisión $\pm 0.5^{\circ}\text{C}$.
- Frecuencia de muestreo: No más de 0.5 Hz (una vez cada 2 segundos).
- 4 pines de conexión espacio de 0.1 pulgadas.
- No requiere componentes activos externos.

3.3.2 Micro controlador NodeMCU ESP8266

El micro controlador NodeMCU ESP8266 tendrá la función de mostrar los datos obtenidos a través del sensor DHT22, para esto se requiere utilizar un código de programación basado en lenguaje C++. El NodeMCU ESP8266 se lo puede apreciar en la figura 3.34.



Figura 3.34. NodeMCU ESP8266 [20].

Características:

- Voltaje de Alimentación (USB): 5V DC
- Voltaje de Entradas/Salidas: 3.3V DC
- CPU: Tensilica Xtensa LX3 (32 bit)
- Frecuencia de Reloj: 80MHz/160MHz
- Instruction RAM: 32KB
- Data RAM: 96KB
- Memoria Flash Externa: 4MB
- Pines Digitales GPIO:
- Pin Analógico ADC: 1 (0-1V)
- Chip USB-Serial: CP2102
- Antena en PCB
- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Stack de Protocolo TCP/IP integrado
- Potencia de salida de +19.5dBm en modo 802.11b

3.3.3 Raspberry pi 3

Utilizamos el raspberry pi para implementar una base de datos MySQL con la finalidad de mostrar los parámetros en una interfaz gráfica, también funcionara como Access point con su respectivo SSID para que los NodeMCU se conecten y puedan transmitir los datos. El raspberry pi 3 se lo puede apreciar en la figura 3.35.



Figura 3.35. Raspberry pi 3 [21].

Características:

- CPU: Quad-Core Cortex A7 a 900MHZ
- GPU: VideoCore IV de doble núcleo
- RAM: 1GB DDR2
- Puertos:
- 4 x USB 2.0
- 1 x 40 GPIO pin
- 1 X HDMI 1.4
- 1 x Ethernet
- 1 x Combo audio/mic
- 1 x Interfaz de cámara (CSI)
- 1 X Interfaz de Pantalla (DSI)
- 1 x Micro SD
- 1 x Núcleo Grafico 3D
- Módulo Bluetooth
- Módulo de Wi-Fi b/g/n en la banda de 2.4GHz

CAPÍTULO 4

4. PRESUPUESTO Y PLANIFICACION

4.1 Presupuesto

En la siguiente tabla 4.1 se muestra el presupuesto y materiales utilizados para la implementación del proyecto.

Cotización			
Nombre	Cantidad	Precio Unitario	Precio Total
Raspberry pi 3	1	\$79.00	\$79.00
NodeMCU ESP8266	6	\$10.00	\$60.00
Tarjeta de memoria 256 GB	1	\$76.00	\$76.00
Sensores DHT22	12	\$8.00	\$96.00
Cables de conexión para los sensores (10 metro)	1	\$5.00	\$5.00
Cinta doble faz	1	\$6.00	\$6.00
Instalación de Puntos eléctricos	4	\$30.00	\$120.00
Rollo de cable eléctrico (12 AWG)	3	\$26.00	\$78.00
Tomacorriente dobles	4	\$1.50	\$6.00
Prensa estopa	6	\$2.50	\$15.00
Cinta aislante	3	\$1.00	\$3.00
Breaker	1	\$5.00	\$5.00
Case de NodeMCU	6	\$5.36	\$32.16
Mantenimiento	1	\$250.00	\$250.00
Instalación	1	\$800.00	\$800.00
Total			\$1,631.16

Tabla 4.1. Presupuesto para la Implementación del diseño.

El mantenimiento se lo realizará cada 6 meses para verificar el estado de los equipos y realizar las respectivas actualizaciones. Se le entregará un manual de configuración de los equipos al departamento técnico del distrito y se le dará la respectiva capacitación sobre nuestro sistema de monitoreo.

Cabe recalcar que los puntos de eléctricos serán instalados por un contratista que será contratado por nosotros.

4.2 Planificación del proyecto

En la figura 4.2 se detalla la planificación del proyecto, este tuvo una duración de aproximadamente 1 mes.

Nombre de tarea	Dura	Comienzo	Fin	Nombres de los recursos
Planificación del proyecto	51 días	jue 10/4/18	jue 12/13/18	Brayame criollo,Ivan Muñoz
Primera Actividad	5 días	jue 10/4/18	mié 10/10/18	Brayame criollo,Ivan Muñoz
Investigacion de escritorio	4 días	jue 10/4/18	mar 10/9/18	Brayame criollo,Ivan Muñoz
Entrevistas	1 día	mié 10/10/18	mié 10/10/18	Brayame criollo,Ivan Muñoz
Segunda Actividad	5 días	jue 10/11/18	mié 10/17/18	Brayame criollo,Ivan Muñoz
Analisis de las problematicas	4 días	jue 10/11/18	mar 10/16/18	Brayame criollo,Ivan Muñoz
Entrevistas	1 día	mié 10/17/18	mié 10/17/18	Brayame criollo,Ivan Muñoz
Tercera Actividad	5 días	jue 10/18/18	mié 10/24/18	Brayame criollo,Ivan Muñoz
Definicion de las problematicas	4 días	jue 10/18/18	mar 10/23/18	Brayame criollo,Ivan Muñoz
entrevistas	1 día	mié 10/24/18	mié 10/24/18	Brayame criollo,Ivan Muñoz
Cuarta Actividad	5 días	jue 10/25/18	mié 10/31/18	Brayame criollo,Ivan Muñoz
Investigacion de las posibles soluciones	4 días	jue 10/25/18	mar 10/30/18	Brayame criollo,Ivan Muñoz
solucion definida	1 día	mar 10/30/18	mar 10/30/18	Brayame criollo,Ivan Muñoz
Quinta Actividad	30 días	vie 11/2/18	jue 12/13/18	Brayame criollo,Ivan Muñoz
Prototipo de bajo costo	6 días	vie 11/2/18	vie 11/9/18	Brayame criollo,Ivan Muñoz
Observaciones del prototipo	5 días	lun 11/12/18	vie 11/16/18	Brayame criollo,Ivan Muñoz
Mejoras del prototipo	5 días	vie 11/16/18	jue 11/22/18	Brayame criollo,Ivan Muñoz
Prototipo de alto nivel	15 días	jue 11/22/18	mié 12/12/18	Brayame criollo,Ivan Muñoz

Figura 4.1. Planificación del proyecto.

CONCLUSIONES Y RECOMEDACIONES

Conclusiones

La implementación de un sistema de notificaciones mediante sensores y microcontroladores, permite tener un mayor control sobre las lecturas de temperatura y humedad en los frigoríficos, optimizando el tiempo de respuesta del usuario.

La implementación del protocolo MQTT facilita la comunicación entre los dispositivos mediante su sistema de publicación y suscripción, permitiendo entregar los datos de manera segura entre dispositivos y evitando los tiempos de respuesta del servidor.

La implementación del raspberry pi para el uso de varias funciones como acces point y bases de datos, permitió un considerable ahorro de dinero debido a que cumplió los estándares deseados del sistema de monitoreo, sin necesidad de implementar una costosa infraestructura.

Finalmente se plasmó lo investigado en nuestro proyecto proporcionándonos la satisfacción de haber culminado este proyecto.

Recomendaciones

Se recomienda utilizar UPS para la conexión de los NodeMCU y raspberry pi, con esto los dispositivos tendrían mayor protección ante variaciones de energía y darían un margen de tiempo hasta que esta se restablezca la energía.

Es recomendable pagar un servicio de Nexmo para el envío de SMS, debido a que en la actualidad solo se puede enviar SMS a un usuario.

Se recomienda adquirir una IP pública para poder ser asignada al raspberry pi 3, con la finalidad de visualizar de forma remota las temperaturas desde cualquier lugar del mundo.

Se recomienda diseñar una aplicación móvil para el monitoreo de las temperaturas, esto con la finalidad de poder seguir monitoreando desde un dispositivo móvil en caso de que el usuario se encuentre fuera del distrito o lejos de su estación de trabajo.

BIBLIOGRAFÍA

- [1] Ubicación geográfica del distrito de salud 09D07, 2018. Google Maps. [Online]. Disponible en: <https://www.google.com/maps/search/subcentro+de+salud++09d07/@-2.0839108,-79.9348509,16z>.
- [2] Diseño de un sistema de sensores dedicado al control y monitoreo de la calidad del agua en piscinas de crianza de camarones, 2018. DSpace. [Online]. Disponible en: <https://www.dspace.espol.edu.ec/handle/123456789/44718>.
- [3] Diseño e implementación de una red de sensores inalámbricos para el monitoreo de temperatura corporal, 2015. DSpace. [Online]. Disponible en: <https://www.dspace.espol.edu.ec/handle/123456789/30312>.
- [4] Tutorial sensor de temperatura y humedad DHT11 y DHT22, 2018. VITEL. [Online]. Disponible en: https://naylampmechatronics.com/blog/40_Tutorial-sensor-de-temperatura-y-humedad-DHT1.html.
- [5] Prensa Estopa PVC, 2018. Naylamp. [Online]. Disponible en: <https://www.vitel.cl/prensa-estopa-pvc/>.
- [6] Cómo programar NodeMCU con el IDE de Arduino, 2018. Programar fácil.com. [Online]. Disponible en: <https://programarfácil.com/esp8266/como-programar-nodemcu-ide-arduino/>.
- [7] MQTT, 2018 Digital dimensión. [Online]. Disponible en: <http://www.digitaldimension.solutions/es/blog-es/opinion-de-expertos/2015/02/mqtt-un-protocolo-especifico-para-el-internet-de-las-cosas/>.
- [8] Setting up a Raspberry Pi as an access point, 2018. Raspberr Pi. [Online]. Disponible en: <https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md>.
- [9] ¿Qué es Node-RED?, 2018. Sofia2 IoT Platform Blog. [Online]. Disponible en: <https://about.sofia2.com/2016/11/16/que-es-nodered/>.
- [10] Flow-based programming for the Internet of Things, 2018. Node-RED. [Online]. Disponible en: <https://nodered.org/>.
- [11] Connect to an MQTT Broker, 2018. Node-RED Cookbook. [Online]. Disponible en: <https://cookbook.nodered.org/mqtt/connect-to-broker>.

- [12] Node-node-node-mysql, 2018. Node-RED. [Online]. Disponible en: <https://flows.nodered.org/node/node-red-node-mysql>.
- [13] Timestamp to current date and time, 2018. Node-RED. [Online]. Disponible en: <https://discourse.nodered.org/t/timestamp-to-current-date-and-time/873>.
- [14] Writing Functions, 2018. Node-RED. [Online]. Disponible en: <https://nodered.org/docs/writing-functions>.
- [15] Node-RED dashboards – multiple lines on a chart, 2018. Node RED Programming Guide. [Online]. Disponible en: <http://noderedguide.com/tutorial-node-red-dashboards-multiple-lines-on-a-chart/>.
- [16] Set Email and SMS Notifications, 2018. Tuts Codingo. [Online]. Disponible en: <https://tuts.codingo.me/set-email-and-sms-notifications-that-are-triggered-by-sensor-values%20>.
- [17] Node-red-contrib-nexmo, 2018. npm. [Online]. Disponible en <https://www.npmjs.com/package/node-red-contrib-nexmo>.
- [18] Node-red-node-email, 2018. Node-RED. [Online]. Disponible en <https://flows.nodered.org/node/node-red-node-email>.
- [19] DHT22 Sensor de temperatura y humedad relativa, 2018. Geek factory. [Online]. Disponible en <https://www.geekfactory.mx/tienda/sensores/dht22-sensor-de-temperatura-y-humedad/>.
- [20] Introducción al ESP8266 y NodeMCU, 2018. GitHub. [Online]. Disponible en <https://github.com/jaimelaborda/Planta-Twittera/wiki/1.-Introducci%C3%B3n-al-ESP8266-y-NodeMCU>.
- [21] Raspberry Pi 3 Modelo B, 2018. Raspberry Pi. [Online]. Disponible en <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.

ANEXOS

Anexo A

Preguntas para la entrevista

Departamento de TICS

- ¿Qué significado tienen las siglas TICS?
- ¿Cuántos sistemas de registros usan para los pacientes?
- ¿Tienen un sistema de online para atención a los pacientes?
- ¿Tienen un sistema de emergencia para la atención del paciente?
- ¿El sistema que usan es centralizado o por cada distrito?
- ¿Qué protocolos de seguridad aplican para los usuarios conectados en la red?
- ¿Cómo se gestiona el ancho de banda?
- ¿Qué problemáticas existen en el departamento de TICS?
- ¿El departamento TICS solo gestiona un centro de salud?
- ¿Cómo se movilizan a otro centro de salud?
- ¿En cada centro de salud hay disponibilidad de Internet?

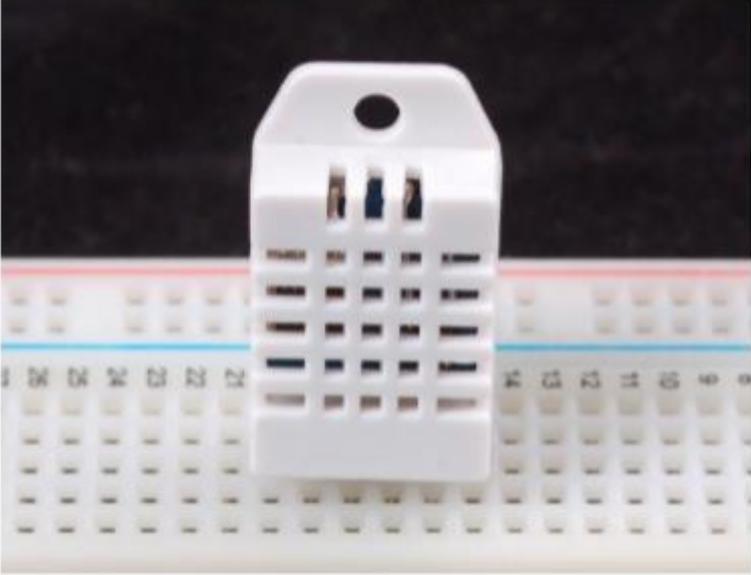
Departamento de vacunación

- ¿Cuál es la función del departamento de vacunación?
- ¿Qué sistemas usan en el departamento de vacunación?
- ¿Qué tipo de seguridad usan para el ingreso al banco de vacunas?
- ¿Cómo manejan la temperatura adecuada en los congeladores?
- ¿Las neveras tienen algún tipo de alarma que indique cuando una de esta falla o está por debajo de la temperatura?
- ¿Qué pasaría si la energía falla, tienen alguna fuente de energía alterna?

Anexo B

Sensor DHT22

Connecting to a DHTxx Sensor
Luckily it is trivial to connect to these sensors, they have fairly long 0.1"-pitch pins so you can plug them into any breadboard, perfboard or similar.



AM2302 (wired DHT22) temperature-humidity sensor

\$15.00
IN STOCK

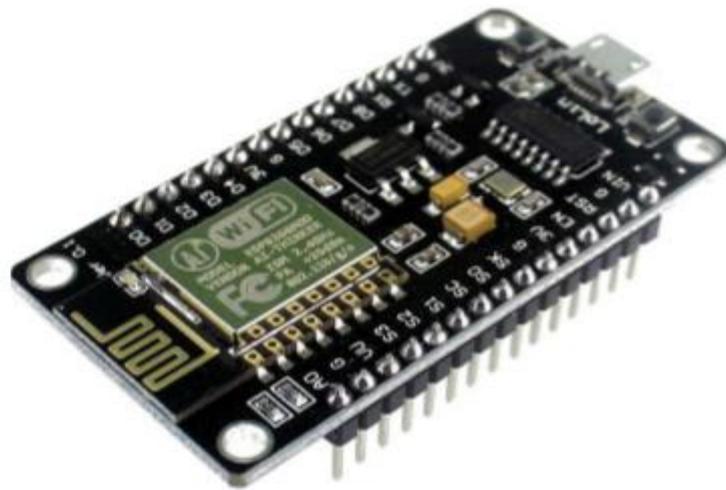
ADD TO CART

Fuente: <https://www.mouser.com/ds/2/737/dht-932870.pdf>

Anexo C

NodeMCU ESP8266

ESP8266 NodeMCU WiFi Devkit



The ESP8266 is the name of a micro controller designed by Espressif Systems. The ESP8266 itself is a self-contained WiFi networking solution offering as a bridge from existing micro controller to WiFi and is also capable of running self-contained applications.

This module comes with a built in USB connector and a rich assortment of pin-outs. With a micro USB cable, you can connect NodeMCU devkit to your laptop and flash it without any trouble, just like Arduino. It is also immediately breadboard friendly.

Fuente: https://www.handsontec.com/pdf_learn/esp8266-V10.pdf